

Muthucumaru Maheswaran  
Elarbi Badidi *Editors*

# Handbook of Smart Cities

Software Services and Cyber  
Infrastructure

 Springer

# Handbook of Smart Cities

Muthucumaru Maheswaran • Elarbi Badidi  
Editors

# Handbook of Smart Cities

Software Services and Cyber Infrastructure

 Springer

*Editors*

Muthucumaru Maheswaran  
School of Computer Science  
McGill University  
Montreal, QC, Canada

Elarbi Badidi  
CIT  
UAE University  
Al Ain, UAE

ISBN 978-3-319-97270-1      ISBN 978-3-319-97271-8 (eBook)  
<https://doi.org/10.1007/978-3-319-97271-8>

Library of Congress Control Number: 2018961431

© Springer Nature Switzerland AG 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

Cities around the world are under immense pressure to accommodate growing populations and address environmental challenges. One idea many cities have started pursuing to address this problem is to leverage Internet technologies and the forthcoming 5G to develop a suite of city-scale cyber-physical infrastructures dubbed smart cities. The major objective of smart city projects is to create sustainable, environmentally friendly cities that can provide services to the residents in the most efficient manner. Smart cities could improve on existing services such as transportation or provide new types of services such as governance that were not available before. The scope of smart cities is ever-changing as new projects around the world are deployed and experiences are gleaned from those projects. Cities manage many critical infrastructures such as transportation, waste management, water resource management, and building services that are ripe for enhancement in a smart city context.

Smart city concept is gaining momentum because of the widespread availability of many technological building blocks such as cloud computing, superfast wireless networking, and Internet of Things. In smart city deployments, sensors and actuators that generate and consume massive volumes of data under diverse formats and ontologies will be integrated into the overall system. The data created by the participating devices need to be appropriately classified and related so that duplication and conflicts can be minimized.

This book provides a glimpse of the research projects that are underway in smart cities and examination of the critical issues relevant for smart cities. The material is targeted toward researchers, developers of smart city technologies, and graduate students in the fields of communication systems, computer science, and data science. The book includes 14 chapters from researchers working on various aspects of smart city-scale cyber-physical systems.

The first three chapters deal with infrastructures for smart cities. Chapter 1 provides a survey of the Internet of Things and computer networking technologies at the smart city scale. It describes some typical smart city projects that apply IoT to provide smart services mainly in three areas: smart mobility, smart sustainability, and smart living. In particular, it discusses the unique challenges posed when

devices and sensors are connected in massive numbers to cloud computing backend services and the security challenges of IoT deployments in smart cities. Chapter 2 examines the roles of 5G and IoT in future smart city scenarios and postulates some future challenges that could emerge in that context. Chapter 3 describes the use of clouds and sensor-based devices for monitoring and managing smart facilities like bridges and other smart applications.

The emerging multidisciplinary field of urban informatics is the focus of Chap. 4. A variety of issues ranging from IoT infrastructure, mobile crowdsensing, big data management, knowledge management of IoT applications, to IoT security and privacy are discussed in this chapter.

Casinos are a major part of city-scale entertainment. The evolution of casinos in a 5G smart city scenario is discussed in Chap. 5. In particular, it presents a design of an integrated casino and entertainment architecture called 5G ICEMO, which relies on the future 5G micro-operators model. It proposes a business model for Integrated Casinos and Entertainment (ICE) in future smart cities and analyzes how the technologies of cloud computing, fog computing, analytics, access control, security handling, virtual reality, robots, etc. can be used to develop ICE micro-operators. Chapter 6 reports on a small-scale prototype smart parking deployment using IoT hardware and cloud computing. The chapter describes the experiment and the experiences obtained through the experiment.

Vehicular crowdsensing is the focus of Chap. 7. It examines two types of vehicular crowdsensing: public and private. In public crowdsensing, a global database is created using the sensing activity, whereas with private crowdsensing individual queries are mapped to the participants who could solve the crowdsensing tasks. The chapter describes a model for personalized vehicular crowdsensing.

Context-sensitive computing is a key smart city technology. In Chap. 8, a new architecture for deploying context-sensitive computing at the scale of smart cities is described. Chapter 9 describes intelligent mobile message support for smart cities based on reinforcement learning. Chapter 10 describes the data integration problem with urban data streams.

Large-scale interoperability is a fundamental problem in smart cities. It could be tackled in many different ways including the creation and adoption of standardized protocols. In Chap. 11, an interesting idea called asymmetric interoperability is presented to tackle the interoperability problem. The chapter addresses the issue of services interoperability in the context of smart cities and the Internet of Things where services are implemented in the IoT devices. These devices typically interact in large numbers while exhibiting different characteristics.

Video surveillance is already gaining popularity. With smart cities, they could see even broader deployment. Chapter 12 surveys technologies and infrastructures for video surveillance management in a smart city. It provides typical examples of smart cities' applications that use video surveillance. It also describes on-premises and cloud-based solutions and the experimental testbed used to evaluate the performance of both solutions in terms of CPU, memory, storage, and bandwidth usage.

Chapters 13 and 14 focus on transportation in smart cities. The management of electric vehicles in smart cities is the focus of Chap. 12, while information

presentation to nudge citizens toward greener transportation choices is examined in Chap. 13. Chapter 12 surveys the research efforts concerning electric vehicles charging by focusing on the issue of selection of suitable charging stations. It describes three main configurations: centralized, distributed, and hybrid. It also describes how mobile edge computing could be used for the selection of charging stations.

Smart buildings play an important role in smart cities. They have a significant impact on the overall functioning of the smart cities. In Chap. 15, a novel energy harvesting approach to deploy wireless sensing is described. Approaches like this could pave the way toward creating maintenance-free remote deployments in smart buildings and possibly in smart cities.

Montreal, QC, Canada  
Al Ain, UAE

Muthucumaru Maheswaran  
Elarbi Badidi

# Contents

<b>Internet of Things (IoT) Infrastructures for Smart Cities</b> .....	1
Quang Le-Dang and Tho Le-Ngoc	
<b>The Role of 5G and IoT in Smart Cities</b> .....	31
Attahiru Sule Alfa, Bodhaswar T. Maharaj, Haitham Abu Ghazaleh, and Babatunde Awoyemi	
<b>Leveraging Cloud Computing and Sensor-Based Devices in the Operation and Management of Smart Systems</b> .....	55
Shikharesh Majumdar	
<b>Mobile Computing, IoT and Big Data for Urban Informatics: Challenges and Opportunities</b> .....	81
Anirban Mondal, Praveen Rao, and Sanjay Kumar Madria	
<b>5G Wireless Micro Operators for Integrated Casinos and Entertainment in Smart Cities</b> .....	115
Da-Yin Liao and XueHong Wang	
<b>An IoT-Based Urban Infrastructure System for Smart Cities</b> .....	151
Edna Iliana Tamariz-Flores, Kevin Abid García-Juárez, Richard Torrealba-Meléndez, Jesús Manuel Muñoz-Pacheco, and Miguel Ángel León-Chávez	
<b>Vehicular Crowdsensing for Smart Cities</b> .....	175
Tzu-Yang Yu, Xiru Zhu, and Muthucumaru Maheswaran	
<b>Towards a Model for Intelligent Context-Sensitive Computing for Smart Cities</b> .....	205
Salman Memon, Richard Olaniyan, and Muthucumaru Maheswaran	
<b>Intelligent Mobile Messaging for Smart Cities Based on Reinforcement Learning</b> .....	227
Behrooz Shahriari and Melody Moh	



<b>Asymmetric Interoperability for Software Services in Smart City Environments</b> .....	255
José C. Delgado	
<b>Management of Video Surveillance for Smart Cities</b> .....	285
Nhat-Quang Dao, Quang Le-Dang, Robert Morawski, Anh-Tuan Dang, and Tho Le-Ngoc	
<b>Intelligent Transportation Systems Enabled ICT Framework for Electric Vehicle Charging in Smart City</b> .....	311
Yue Cao, Naveed Ahmad, Omprakash Kaiwartya, Ghanim Puturs, and Muhammad Khalid	
<b>Green Transportation Choices with IoT and Smart Nudging</b> .....	331
Anders Andersen, Randi Karlsen, and Weihai Yu	
<b>Energy Harvesting in Smart Building Sensing: Overview and a Proof-of-Concept Study</b> .....	355
Aristotelis Kollias, Colton Begert, and Ioanis Nikolaidis	
<b>Building a Data Pipeline for the Management and Processing of Urban Data Streams</b> .....	379
Elarbi Badidi, Nouf El Neyadi, Meera Al Saeedi, Fatima Al Kaabi, and Muthucumar Maheswaran	
<b>Index</b> .....	397

# Internet of Things (IoT) Infrastructures for Smart Cities



Quang Le-Dang and Tho Le-Ngoc

**Abstract** Smart City promises to enhance resource utility, cost-effectiveness, sustainability and living conditions in urban environments by utilizing Internet-of-Things (IoT) infrastructures. This chapter presents a comprehensive survey on the architectural design and key wireless communication technologies that enable Smart City applications. In addition, with the adoption and installation of IoT devices on a city-wide scale, securing these devices and the associated communications networks becomes an important issue. As a result, this chapter then continue with a survey to discuss potential security threats for IoT devices in a Smart-City environment, possible countermeasures and open research issues.

**Keywords** Smart City Architecture · Smart City Security · Smart City Enabling Infrastructures

## 1 IoT and Smart Cities: An Introduction

The last few decades have witnessed an unprecedented trend of the world population moving to live in urban areas. The year 2008 marks a big milestone that for the first time, more people are living in cities than in the rural areas<sup>1</sup> and it is predicted that 2050, two out of every three people will be metropolitan inhabitants.<sup>2</sup>

This concentration of population in small areas brings about many consequences including scarcity of natural resources, pollution, insufficiency of infrastructure, and public safety management. As a result, sustainable development is the central

---

<sup>1</sup>Urban population – World Bank: <http://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS?end=2015&start=1960&view=chart>

<sup>2</sup>World Population Prospects – UN: <http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html>

Q. Le-Dang · T. Le-Ngoc (✉)  
McGill University, Montréal, QC, Canada  
e-mail: [quang.le2@mail.mcgill.ca](mailto:quang.le2@mail.mcgill.ca); [tho.le-ngoc@mcgill.ca](mailto:tho.le-ngoc@mcgill.ca)

theme that many cities and governments focus on recently. However, sustainable development in any city can only be realized through innovations that place sustainability, resilience, flexibility, adaptive capacity and efficiency as fundamental considerations. Driven by those motivations, the idea of smart cities emerges to protect the environment, utilize the resources and infrastructures more efficiently and more importantly, place at the heart of this idea, to improve citizens' quality of life, answering their needs. With the technological advancements in recent years, many novel developments in environmental sciences, intelligent monitoring, big-data analytics, distributed signal processing and advanced communications technologies are centralized around this theme. Among these innovations, the integration of the Internet-of-Things (IoT) [1–4] and advanced Information and Communications Technology (ICT) [5, 6] has been recently considered as the key element to develop smart/sustainable urban infrastructures.

Through its extensive sensor networks, IoT deployment can supply a huge amount of data throughout the city in real-time, providing insights into city operations and policymaking process [7]. Additionally, a unified network infrastructure of IoT can facilitate communications and interaction among multiple applications and services, such as smart grid, intelligent traffic, waste management, environment monitoring, and smart surveillance [8], allowing more effective resource sharing and information exchange. It is especially beneficial when data in a specific domain can potentially support smart applications in other domains as well [9], greatly reducing the infrastructure deployment cost and functionality overlapping. Thus, applying IoT concept and practices is extremely appealing to city administrators for their Smart City.

Within this effervescent context, this chapter provides a concise survey on the turnkey technologies and infrastructures that enable Smart City services/applications as well as the security aspects. In particular, the chapter is organized as follows. Section 2 provides the background on current trends of applying IoT technologies for Smart Cities and notable Smart City applications and their impacts. Section 3 describes the architectural design of IoT-based Smart Cities and enabling technologies. Section 4 provides a survey on key enabling wireless communications technologies for Smart Cities. Section 5 surveys the security threats, counter measures and open issues. Finally, Sect. 6 concludes the chapter.

## 2 Background

### 2.1 *Current trends in Internet of Things for Smart Cities*

Among the broad facets of Smart City, the applications of IoT technologies typically have greatest contributions in three aspects: smart mobility, smart sustainability, and smart living<sup>3</sup>.

---

<sup>3</sup>Other Smart City aspects, such as smart governance, smart people, and smart economy [10] also benefit from data produced by IoT applications, but are beyond the scope of this chapter.

Smart mobility refers to the use of IoT applications to enhance transportation and logistics operations, improve commuting efficiency, and reduce gas emissions, thus improving travel experience for citizens throughout the city as well as reducing the pollution caused by vehicles. Examples of such applications encompass IoT-enhanced public transportation systems, smart parking monitoring and effective traffic management.

Smart sustainability aims to reduce environmental impact of energy consumption and pollution from city operations [1–3, 6, 7]. The emphasis is put on monitoring, managing, and distributing resources such as waste, water, and electricity efficiently and dynamically according to demand. IoT applications such as intelligent lighting, environment monitoring, smart irrigation system, waste management, and smart grids are good examples of this category.

Smart living often involves using IoT devices for enhancing quality of life, preventing and minimizing the risk and impact of adverse events such as crimes, accidents, as well as making the Smart City safer and more attractive to residents. Applications of this category include video surveillance for public security and interactive kiosks that provide users with location-based information feeds and context-based advertisements.

## *2.2 Smart City Applications Around the World*

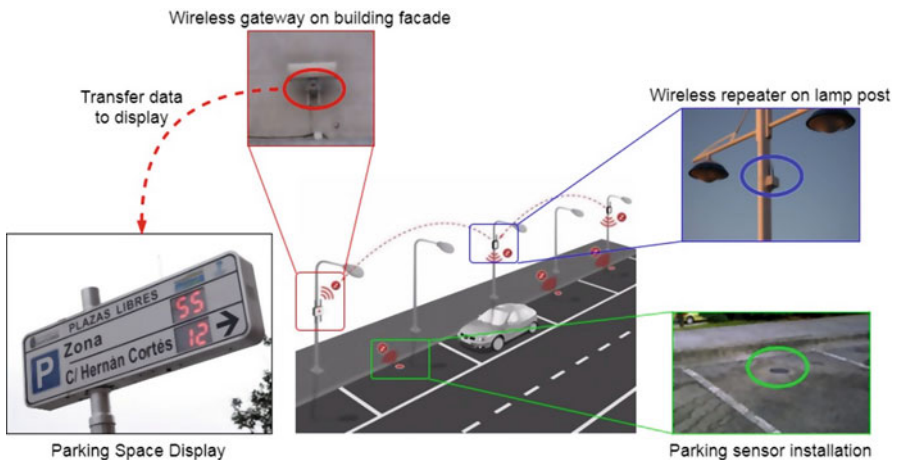
Due to these several fold benefits, Smart City concept over an IoT platform has been adopted and implemented in many metropolitan areas around the globe. With the huge number of smart cities that have implemented one or more IoT-based systems, it is impossible to have a comprehensive survey. Instead, in this section, some representative examples of smart cities around the world will be discussed to highlight how their smart applications utilize IoT technology to achieve the objectives of smart mobility, smart sustainability, and smart living and their impacts.

### **2.2.1 Smart Mobility Projects**

**Barcelona, Spain** Barcelona is one of the first cities pushing for Smart City model with many IoT-based initiatives; particularly, the mobility projects aim to improve traveling experience for citizens through the use of sensors technologies. Parking spaces are equipped with wireless sensors to notify and guide drivers to available spaces through mobile apps, reducing unnecessary commuting to find parking. In terms of public transport, city buses are equipped with wireless GPS sensors to monitor their locations, which can be accessed through interactive electronic displays mounted at smart bus stops to present information automatically to passengers about bus arrival and departure timing, thus improving customer experience while waiting for a bus [11].

**Beijing, China** A smart traffic management system with 157 high-definition cameras on Beijing's surrounding expressways was deployed to automatically count vehicles and provide traffic flow statistics, as well as automatically record any events such as accidents when they occur and alarm authorities as necessary [12]. At the same time, tens of thousands of traffic flow detectors are installed in expressways and near intersections to automatically collect traffic flow, speed, and density data, which is processed by a traffic control system to automatically adjust traffic signals according to the number of cars on the road. Overall, the smart traffic management system has reduced Beijing's traffic congestion by up to 60% and doubled its road capacity, reducing driving time and fuel consumption. In particular, Beijing's 5th Ring Road, 80% of intersections are regulated by this system, which has increased its road capacity by 15% [12].

**Santander, Spain** The Smart Santander project started in 2010, enabled installations of thousands of devices throughout the city of Santander to provide an urban city-scale real-world IoT experimentation facility [13, 14]. One highlight application of the project is a smart outdoor parking management service that deployed over 400 ferromagnetic wireless sensors to monitor parking availability in the city center area. The sensors, equipped with 802.15.4 radio module for communication, are buried under the asphalt at each parking space, which limits the devices exclusively to battery power and wireless communication. Thus, wireless relay nodes and gateways are deployed in the area to provide connectivity to the Internet so parking occupancy data can be delivered quickly to drivers and traffic control system. Sensor data can be displayed on a smart phone app or electronic panels located at street intersections to indicate how many free parking spaces are on that street as shown in Fig. 1. The project offers a city-scale testbed for



**Fig. 1** Smart parking system in smart Santander project. (Draw based on [13])

experimentation with IoT technologies as well as many insights in deployment and management of a large-scale IoT infrastructure.

### 2.2.2 Smart Sustainability Projects

**Amsterdam, Netherlands** Since 2006, many projects have been initiated in Amsterdam, Netherlands, progressing toward Amsterdam Smart City. One of the notable and highly successful projects is a smart lighting project in collaboration with Philips, Cisco, and Aliander to install a connected lighting system and public Wi-Fi connection at Hoekenrodeplein, near Amsterdam Arena [15]. With lighting accounting for 19% of all electricity consumed, the goal is to apply adaptive lighting with smart controllers, allowing automatic adjustment based on different circumstances, such as dimming during low traffic hours, or boosting for public events, thus optimize energy usage and reduce unnecessary consumption [16]. The project is expected to generate about 130 billion euro in energy savings while providing better utility for citizens [17]. Another high impact project in Amsterdam is Climate Street initiative which aims at improving energy management and the efficiency of public services such as waste collection. The waste bins are equipped with wireless level sensors for status report so that the waste is only collected when the bins are full. The city is also equipped connected electricity meters for dynamically matching of electricity demands. The project helps to reduce the annual CO<sub>2</sub> emission of the commercial district by 62% in 2 years [18].

**Padova, Italy** A proof-of-concept IoT application was deployed as a collaboration between the University of Padova and the city of Padova to promote early adoption of urban IoT solutions in public administration for "Padova Smart City" [7]. The target application consists of a system for collecting environmental data and monitoring public street lighting using various wireless sensors mounted on street light poles. The implementation uses Temperature, humidity, light intensity, and benzene sensors to monitor weather conditions and air quality. Data was collected over a period of 7 days which observes regular pattern of measurements corresponding to day and night cycles, which can be used to detect anomalies due to traffic congestion and weather conditions, such as rainstorms, so that system adjustments can be made automatically.

**Las Vegas, USA** Located in the desert with limited waters supply, Las Vegas looks to optimize its water distribution and management system to be as efficient as possible. The Las Vegas Valley Water District (LVVWD) employs wireless leak detection devices that periodically listen for sounds or vibrations that may be caused by water seeping from the system. In total, 13 permanent acoustic sensors are installed monitoring 3 miles of the aging pipeline under Las Vegas Boulevard [19]. The technology enables preventive maintenance by detecting small leaks that may go unnoticed and reduce leakage loss. Though LVVWD has committed to reduce water consumption per capita to 199 liters in 2035, consumption has fallen to 205

liters per person/day at the end of 2014 thanks to its proactive leak control and maintenance programs [20].

### 2.2.3 Smart Living Projects

**New York City, USA** An interactive platform called City24/7 is launched in collaboration with Cisco and the City of New York to deliver information from open government programs, local businesses, and events to citizens as requested [21]. Information is displayed on durable, easy-to-use Smart Screen kiosks that replace unused public assets such as payphone and posting boards typically located at key locations such as bus stops, train stations, and shopping malls. These kiosks incorporate touch, voice, and audio technology to deliver wide variety of local news, services, and advertisements to all types of users, including people with disabilities. The kiosks can also be equipped with sensors to support a citywide sensing network that can monitor and alert people environmental and weather conditions, as well as cameras for video surveillance applications. The initiative aims to deploy 250 Smart Screens throughout New York City with future plans to expand toward to Los Angeles, London, Boston, and many other cities in the United States and around the world [21].

**Seoul, South Korea** Seoul has operated the u-Seoul Safety Service since 2008 using advanced location-based services and CCTV technologies to notify authorities and family members of emergencies involving children, the disabled, or the elderly [22]. A dedicated smart device was developed to notify its holder when they leave a designated safe zone and provide an emergency call button that can notify guardians and emergency respondents when pressed. The device also includes a GPS tag to monitor its holder's location, whose data can be complemented by real-time CCTV networks to quickly locate missing children. Seoul is expected to have 50,000 users registered to the service with multiple efforts to provide necessary devices to low-income and vulnerable groups.

**Medellin, Colombia** Medellin created the Integrated Emergency and Security System (SIES-M) in 2013 to bring together more than 10 different government agencies responsible for responding to emergencies [23]. The system enables different services such as police, medical, and fire departments to mount a coordinate response to citizen reports through a single security and emergency number, 123. Information from citizen reports is cross referenced with data from 823 video surveillance cameras distributed throughout the city. Each camera covers a 120-meter radius area, and is linked to fiber optics and wireless network to transmit high-definition video to integrated control center. The SIES-M has enabled optimization of resources for city security organizations and increased timeliness of services to citizens in emergency and security situations.

As summarized in Table 1, this survey shows that Smart City initiatives are gaining traction all over the world, with many cities already develop fully opera-

**Table 1** Notable examples of smart city applications around the world

Category	City	Application	Instruments	Outcome/impact
Smart mobility	Barcelona [11]	Parking Notification, public transport monitoring	Wireless parking sensors, GPS sensors	Encourages public transportation and improves customer experience
	Beijing [12]	Smart traffic management	HD cameras, Traffic flow detectors, traffic signals	Reduces congestion by 60% and doubled road capacity
	Santander[13, 14]	Smart parking management	Parking sensor, wireless communications	Experimental application to notify drivers of available parking to reduce commuting
	Amsterdam[15–18]	Smart Lighting, smart public service	Light sensors, wireless electricity meters, wireless smart bins	Reduces energy consumption cost by 130 billion euro Reduces the CO2 emission by 62% in 2 years
SmartLiving	Padova [7]	Smart Lighting, environmental monitoring	Wireless temperature, humidity, light, benzene sensors	Proof-of-concept implementation to observe light intensity and weather conditions
	Las Vegas [20]	Smart water leak detection	Wireless acoustic sensors	Enables preventive maintenance and reduces water loss to leakage
	New York [21]	Smart information kiosks	Smart screen kiosks with video, video capabilities and sensors	Delivers information and advertisements on-demand to citizens Environment monitoring and surveillance
	Seoul [22]	Smart safety	GPS monitor, CCTV cameras	Enhances security for children and vulnerable citizens
	Medellin [23]	Smart public safety	HD cameras	Enables coordinated and enhanced emergency response



tional smart services while others begin pilot projects. In these initiatives, it is clear that IoT technology plays an integral role in many applications of Smart City to collect comprehensive data about the cities and their residents. While most designs utilize dedicated sensors to collect specific data for their smart services, devices producing multi-purpose data like cameras are becoming more common and often seen in many applications across multiple domains, such as high-definition cameras being used in both smart traffic management and public safety applications. As such, it is more efficient in terms of both deployment and maintenance cost to reuse generic data from these devices for various Smart City applications. Moreover, it is also observed that wireless communication also gains its popularity as the glue, connecting the low-power, low-capacity IoT devices to the core network.

### 3 Sensing Architectures for Smart Cities

Due to the infancy of Smart City technologies, many architectural models were proposed in the literature. The proposals includes the 3-layer [24, 25], 5-layer [26–28] and 7-layer [29, 30] architectural models. Basically, the more layers the model has, the more separation of functionalities for the development but more complex to understand and less correlated between the architecture model and the real IoT components. To strike a balance between these two extremes, we propose a 4-layer architecture based on the well-known IoTWF [29] and ITU-T architecture [31] for IoT applications in general and Smart City in specific. We believe that this architecture provides a good abstraction for understanding the Smart City system as a whole as well as enough details for implementation. This IoT architecture for Smart City is divided into four layers: the data acquisition and control layer, the connectivity layer, the data management layer and the application layer as illustrated in Fig. 2.

**The data acquisition & control layer** is where all data gathering in Smart City takes place. Enormous amount of information is collected by various types of IoT sensors. This layer also composes of actuators such as traffic lights, street panels to react according to some predefined configurations or explicit directions from the human in charge. In fact, the components of this layer vary the most in type, capability, and quantity according to the requirements of Smart City applications. RFID and sensor networks are considered the basic building blocks of this layer [32], mainly due to the low cost hardware. In order to connect constrained devices (without IP capability) to the upper layer, gateways are used for protocol translations, security and management. Recently, new applications like smart video surveillance and traffic control become more common in Smart Cities [33], as discussed in Chapter 11 “Management of Video Surveillance for Smart Cities”. These applications demand sensing capability and data rate much higher than that supported by sensor networks. These applications normally demand real-time high-definition (HD) images from Internet protocol (IP) cameras, which requires very high-speed and low-delay data transmission. To accommodate the new

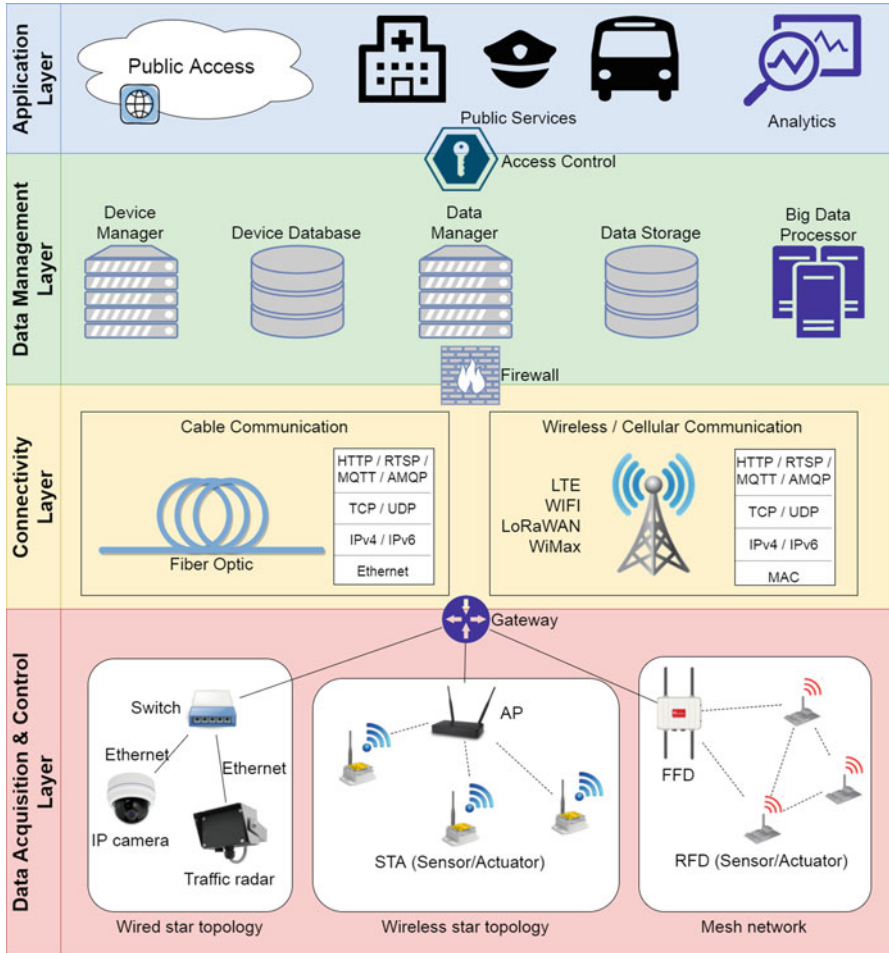


Fig. 2 Smart city IoT multi-layered architecture

requirements, IEEE 802.11 (Wi-Fi) are commonly used to support streaming of live videos. It is important to acknowledge that this layer does not only facilitate sensing plain environment data but can also collect citizen related information such as people satisfactions, their behaviors and opinions.

**The connectivity layer** transports data collected from IoT devices to the data center and delivers commands from remote control operator to the IoT devices. It is the backbone network infrastructure that holds the Smart City devices together and bridges the communications between the data acquisition and control layer and the upper layers. The communication channel is either wired network (e.g., optical fiber) or wireless/cellular network (e.g., LTE, LoRaWAN), depending on the availability of such technologies and requirements of Smart City applications.

Serving as a backbone infrastructure, wired communication technologies such as fiber optic networks are normally used at this layer due to their stability, large bandwidth, and low delay characteristics. However, if the IoT components are mobile, for instance, mounting on a service truck, wireless communication technologies need to be used due to their flexibility and mobility support. In these cases, depending on the traffic volume, different scenarios can be envisioned. First, for low bandwidth IoT components (GPS, temperature sensors, etc.), LTE and LoRaWAN can be used. LTE networks have wide coverage and higher bandwidth; however, LTE radios and subscriptions are quite expensive so it is only suitable for a limited number of mobile IoT devices. On the other hand, LoRaWAN networks offer wide coverage at low cost but at narrow bandwidth. For large bandwidth IoT components (such as cameras on road inspection vehicles), wireless communications are normally not adequate or too expensive for real-time streaming and offline storage is required for latter upload.

**The data management layer** is the central point that oversees the operation of Smart City devices and data. This is where the collected data from the entire city is stored and processed. It also provides interface to remotely monitor, configure, and control IoT devices in the data acquisition layer. Usually, the logical components of the management layer are separated into two subsystems: device management (control plane) and data management (data plane). Device management system provide facilities for collecting, storing status reports from deployed devices as well as a uniform platform for controlling devices with different capabilities from different vendors. Monitoring device availability, firmware update and reconfiguration are some of the typical functionalities on this device management system. Data management system facilitates data collection, storage, and visualization from all the IoT devices in the network. Since there are multiple devices from different vendors, with different capabilities, sensing different environmental factors, data should be translated and stored in a unified format for ease of access and usage afterwards. As this layer is critical to the Smart City infrastructure, it must be protected against threats from the Internet, such as via the use of firewalls. Additionally, the management layer maintains access control over devices and data to prevent misuse of information and services provided by Smart City IoT infrastructure.

To facilitate these functionalities, two main approaches can be identified based on the placement of the data management platform: On-premise and cloud-based approach. In the On-premise approach, all the above functionalities are hosted on local computing facilities such as a data center. In this case, the operational, administration and maintenance of these facilities will be on the shoulder of the data center's operator. This approach offers the benefits of performance and data privacy. Since the data sink is placed close to its sources, the performance of data archiving in terms of bandwidth and delay can be guaranteed easily. This is especially advantageous for real-time data. In addition, since data is stored locally, the operator has the complete control over the data, how to use it and its privacy control. However, this approach can be costly to set up, maintain and scale up this

back-end infrastructure for supporting the growing Smart City IoT network and applications.

Cloud computing is a trendy solution which brings about the availability of virtually unlimited storage, processing capacity and computing resources at low cost that can be virtualized and leased to users on demand [34]. In cloud-based approach, many of the management functionalities are designated to the cloud computing facility, which is also referred as Cloud of Things [35] or CloudIoT [36]. In this case, the Cloud abstracts away the complex hardware back-end server details that would have to be carefully planned in the On-premise approach. To catch up with the IoT integration and Smart City trends, many cloud providers such as Amazon [37], IBM [38], and Microsoft [39] are racing to adopt their cloud computing platforms for delivering various IoT services over the Internet.

**The Application Layer** is where the vast amount of collected data is processed and analysed in order to extract useful information about the current state of the Smart City. The applications can be provided in this layer are what make up the attractiveness of IoT and Smart City. Depending on the analytic functions applied on the data, the value added information could be as raw as counting the number of vehicles passing an intersection or as complicated as the recommendations for planning the city infrastructure for the next 10 years. Moreover, the readings of the collected data could raise an alarm or warning on some critical events and depending on the applied algorithms, these events could in turn trigger an actuator to react quickly to the situation. Furthermore, each iteration of processing produce more valuable information that may be applicable to other application domains. The possibilities for IoT applications in Smart City, are virtually endless, though they generally aim to achieve one of the following 3 purposes: presentation, optimization, and prediction. **Presentation** enables human operators to observe and comprehend a situation through data collected from IoT devices. These applications can visualize incoming data in real-time or combine it with historical data to identify trending patterns or suspicious activities. **Optimization** uses advanced analytics to automatically optimize the operations and utilization of resources in Smart Cities. These applications (such as smart lighting and waste management) have active control over IoT components such as actuators, to adjust the system dynamically in response to the observed conditions through the collected data. **Prediction** applies statistical analysis to predict the likelihood of events in the future. These applications normally incorporate historical data from many sources to determine the prospect of important events such as traffic jams, floods, power outage, etc.

As shown in Fig. 2, the presence of the data acquisition and control layer imposes a significant challenge to the implementation of Smart Cities. On one hand, since the IoT networks may span a wide coverage area, the question of how to facilitate the communication to and from this vast number of devices is a hard question. A general answer to this question should be the use of wireless communications, but how to choose which technology to use in a certain scenario is still not clear. To answer this question, the next section will present a survey on enabling wireless communication technologies for Smart Cities. On the other hand, as the IoT device are exposed to the public, how to secure these devices is a critical issue. Unattended

IoT devices are vulnerable to physical attacks, vandalism [40], DoS attacks [41] or can be used as a pedal to penetrate the Smart City network. The effects of these attacks can be very devastating from breaking into a house, altering medical records to endangering public safety of a city by manipulating the actuators in Smart City. To give an answer to this issue, a survey on current approach will be presented in Sect. 5.

## **4 Smart City Enabling Wireless Communication Technologies**

In the context of a Smart City, the deployment of massive number of devices over a city-wide area makes communications challenging. To facilitate data transfer, the traditional approach of connecting devices directly to the wired network is either too expensive or not adequate due to the lack of existing wired infrastructure. In this case, the amendment of wireless communications to extend the reach of the existing wired infrastructure is not only unavoidable but also one of the key factors that make a Smart City program a success. This section aims at reviewing the existing predominant enabling wireless communications technologies for Smart Cities. For the ease of organization, these technologies are grouped by the coverage area in which the possible use cases for these technologies will be discussed. At the end of this section, a brief summary and a discussion on the open research issues will be discussed.

### ***4.1 Short-range Communications***

Short-range communications refer to the communications range of a few meters or less. With a short-range data transfer, this kind of communications has unique useful characteristics such as low cost/disposable hardware, low power, low interference, and secure (hard to eavesdrop). In a Smart City scenario, short-range communications can be benefit in various applications such as assets management, access control, intelligent transportation (vehicle identification, tracking, and road toll collection), etc.

Radio-Frequency Identification (RFID) is a wireless communications technology that is designed to identify and track physical objects over a short distance [42]. A RFID system generally consists of RFID tags, RFID readers and a management system. RFID tags consists of a microchip connecting to an antenna, depending on the presence of a dedicated battery, RFID tags are classified into passive tags and active tags. Active RFID tags uses a dedicated battery to power the microchip and necessary communications. On the other hand, in passive RFID tags, there is no battery and the tag has to harvest the energy from the transmitted signals from the

reader to power itself and communicate with the reader. Generally speaking, passive tags are cheaper, smaller, and can be used without the hassle of changing battery; however, the tradeoff would be shorter communications range in comparison to the active ones.

Each RFID tag is embedded with a unique identification number. When attached to an object, RFID tags can be queried by the readers and hence can support object identification and tracking.

RFID systems operate in one of the four unlicensed spectrums as follows: 125–134KHz (Low Frequency – LF), 13.56 MHz (High Frequency – HF), 865-956 MHz (Ultra-High Frequency - UHF), 2.4GHz–5.8GHz (Microwave). The operational range of RFID is in the range of 10 cm (LF) to 10 m (MW) and the transmission rate is from 1Kbps (LF) to 500Kbps (MW) [43].

In a Smart City, RFID can be found in many useful applications such as vehicle tracking, toll collecting, access control, and assets management [8]. The main advantage of RFID are its low cost, durability, reusability, small size, non-contact communication and no-battery operation (passive RFID).

## ***4.2 Medium-range Communications***

Medium-range communications refer to communications range of less than a couple of hundreds of meters. Medium-range communications can be considered as the extension of the existing wired infrastructure, providing the flexible, quick and low cost deployment of wireless communications. In a Smart City scenario, medium-range communications can be used in many scenarios including Smart City data collection, device controls and providing large bandwidth connection for applications such as video surveillance.

### **4.2.1 802.15.4-based Technologies**

The IEEE 802.15.4 standard [44] is one of the most well-known and widely used standards for low data rate Wireless Personal Area Network (LR-WPAN). The goal of IEEE 802.15.4 standard is to provide a low rate monitor and control applications for small devices with low energy consumption in a small area at low cost. The standard defines the physical layer (PHY) and specifies a sub-layer for media access control (MAC). At the physical layer, the standard supports communications over two unlicensed frequency bands: the sub-1GHz band (868 MHz for Europe and the 915 MHz for American) and the popular 2.4GHz band. The standard combines the direct sequence spread spectrum (DSSS) technology with the Binary Phase Shift Keying (BPSK) and Offset Quadrature Phase Shift Keying (O-QPSK) modulation schemes to facilitate wireless communications. The supported bit rate was originally at 20, 40 and 250 kbps on the 868, 915 MHz and 2.4GHz frequency bands,

respectively [45]; however, the revision in 2006 improves the maximum bit rate for all bands to 250Kbps [46].

At the MAC layer, 802.15.4 defines the data frame structure for transmission through the wireless physical channel. The maximum frame size of 802.15.4 is limited at 127 bytes. For channel access, 802.15.4 employs Carrier-sense multiple access with collision avoidance (CSMA/CA). The protocol is claimed to support a large network of up to 65 k nodes [28].

The higher layers are not defined by this standard which enables many protocols to be built upon. Zigbee, 6LoWPAN, and Thread are some of the protocols that use IEEE 802.15.4 as their foundation for wireless communications.

## Zigbee

Zigbee [47] is the most popular and most widely used communication protocol based on IEEE 802.15.4. It features a complete protocol stack for IoT applications built on top of IEEE 802.15.4. Zigbee aims at low power, low bandwidth, and low complexity applications deployed on low cost devices such as home automation, lighting control, telemetry etc. Zigbee supports multiple topologies, including star, tree and mesh topologies. With very low operation duty cycle, a Zigbee device can be powered on coin cell batteries for years. Latest Zigbee devices can make use of energy harvesting for battery-less operations [48]. Being based on IEEE 802.15.4, Zigbee can support the bit rate up to 250Kbps; however, due to the protocol and signaling overhead, the actual achievable data throughput is only a fraction of this number. Moreover, in the mesh topology, as the number of hops between the source and destination node increases, the achievable data throughput decreases rapidly. Table 2 shows the throughput test results that the authors did in an urban area around McGill University. The throughput was measured based on loopback tests with encryption enabled on Zigbee Pro radios at 50 m intervals. The result illustrates that the 2.4GHz Zigbee radios cannot establish a stable connection with any distance further than 50 m while the Sub-GHz Zigbee can reach over 415 m and is more suitable for Smart City applications.

Since Zigbee devices are not IP-compatible, to support data transmission between a Zigbee network and the Internet, a gateway is required. The gateway runs the Zigbee and the TCP/IP protocol stack in parallel and conduct protocol translation between the two networks.

**Table 2** Throughput test results on Zigbee radios (2.4GHz and 915 MHz)

Distance	Zigbee 2.4GHz (Peak/average)	Zigbee 915 MHz (peak/average)
1 m	8.9/8.9 Kbps	16.4/16.4 Kbps
50 m	8.9/7.3 Kbps	16.4/16.4 Kbps
415 m	N/A	11/2.5 Kbps

## IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)

6LoWPAN [49] is a fairly new standard for IoT which aims at combining the attractive characteristics of low power, low cost and mesh network capability of the IEEE 802.15.4 with the benefits of IP communications of low cost devices. In its essence, 6LoWPAN is an adaptation layer for IP support over IEEE 802.15.4. The standard only supports IPv6 due to its larger addressing space and address auto configuration capability. In order to operate with IPv4 networks, an IPv6-to-IPv4 protocol translation gateway is required. In order to support IP over the relatively short 802.15.4 packet, header compression methods and packet fragmentation schemes were defined. In the best case, an IPv6 header can be compressed to only 2 bytes [50].

The IP-native support is an interesting features and is a step-up in comparison to Zigbee as it enable IP-based end-to-end direct access to the device from the Internet or vice versa, as well as direct communications between devices across the Internet without the need of any protocol translation gateway.

## Thread

Thread [51] is a latest protocol standard added to the IEEE 802.15.4 family. In fact, thread can be considered as a complete IoT stack that uses IEEE 802.15.4 for wireless connectivity, 6LoWPAN for IP adaptation. On top of these two layers, Thread adds a routing layer and User Datagram Protocol (UDP) or Datagram Transport Layer Security (DTLS) for data transport and security provisioning.

Although Thread was only introduced recently (2015), the standard is backed by big players in the market such as ARM, Samsung, LG, Philips etc. which illustrates its potential of competing against Zigbee and become a dominant standard for IP-based IoT in the future.

### 4.2.2 WiFi

WiFi is a set of standards built by the IEEE 802.11 working groups [52] for the lowest two layers of the OSI model, namely the Physical and Datalink layers. The original standard was built as a wireless counterpart for the wired IEEE 802.3 Ethernet. WiFi primarily operates in the licensed-free 2.4 and 5GHz frequency band with various types of channel bandwidths (from 20 MHz to several hundreds of MHz) and modulation schemes (DSSS, FHSS, OFDM, MIMO-OFDM, etc.). WiFi technology can offer very high bit rate of up to several Gbps (802.11 ac). However, this high bit rate comes at the cost of a fairly large power consumption which makes it not very attractive to battery-based IoT applications.

At Datalink layer, WiFi uses random access methods based on CSMA/CA protocol for managing user access. Although random access relax the stringent synchronization of other channel access methods such as Time Division Multiple



Access (TDMA), it performs poorly when the number of devices scales up. This is another drawback of WiFi as an enabling technology for IoT as an IoT network generally has to be able to support thousands of devices.

Coverage range is another drawback of WiFi, in an urban area, WiFi signal is easily absorbed by the thick concrete walls and a WiFi network can only cover an area of about several tens of meters in diameter. As a result, a huge cost has to be invested to provide a city wide coverage IoT network.

On the other hand, it is acknowledged that WiFi is the cheapest technology that can provide high bitrate wireless connectivity. For applications such as video surveillance, WiFi maybe the only wireless solution that can provide both installation flexibility and enough throughput for video streams at reduced costs (as presented in Chapter 11 on “Management of Video Surveillance for Smart Cities”). However, since the WiFi spectrum is normally overcrowded in urban areas, careful site survey has to be done regularly

### **4.2.3 Bluetooth Low Energy (BLE)**

Bluetooth [53] is a wireless communication standard that was originally designed for hands-free calls between a mobile phone and cordless headsets. Due to the low cost of hardware and the usefulness of the application, Bluetooth is so successful that a Bluetooth connection has become an integrated part of almost all mobile phones today. As technology evolves, more functionalities have been added to the standard including high-definition music streaming and fitness tracker accessory. As the smart phones becomes an important part in one’s daily life and IoT applications such as smart home start to lift off, the expansion to IoT support is a natural move. As a result, a new standard was developed in 2010, namely Bluetooth Low Energy or Bluetooth Smart for effective connectivity with devices with small batteries (IoT devices) as well as the existing large devices (such as smart phones). BLE operates over the 2.4 GHz frequency band over the set of 40 2 MHz channels, using frequency hopping to reduce the interference with coexisting devices in this frequency band. The supported bit rate in BLE is from 125 Kbps to up to 2 Mbps and the transmission range is up to 100 m. However, as the 2.4 GHz frequency can be absorbed easily by concrete structures, the transmission range in practical cases should be surveyed carefully. With highly customizable operation duty cycle, a BLE IoT device running on a coin cell battery can operate for a couple of years. Initially BLE only support point-to-point and star topologies but a mesh profile was recently added to the specifications to enable more use scenarios of BLE.

## **4.3 Long-range Communications**

Long-range communications refer to the communications range of over hundreds of meters, with the capability of covering/connecting the entire city wirelessly. As

the Smart City deployment expands, more IoT devices on a larger area need to be connected. In this case, the medium-range communications standards struggle as with a mesh setup, expanding the network to a wide area may require the deployments of devices to provide the connectivity, which can be very costly. In addition, the mesh setup may cause many undesirable effects such as performance degradation in terms of throughput, delay, and packet loss due to the multi-hop communications and unbalance power consumption due to un-optimal routing. For example, it is reported that the 6LoWPAN throughput after 3 hops could drop to as low as 0.8 Kbps [54], falling very far from the theoretical figure of 250 Kbps.

With the observations above, the idea of a long-range, low power communications scheme is very attractive in the context of IoT and Smart City as it allows the quick and arbitrary deployment of IoT devices without worrying about the provisioning of connectivity. In order to achieve these benefits, many modifications have to be made [54]. Firstly, to enable long range communication, the operational frequency bands of these schemes have to shift to the Sub-1GHz band. Secondly, narrow band communications are used to enhance the signal-to-noise ratio, spectral efficiency, and allowing more devices. Thirdly, light-weight medium-access control, simple infrastructure and complexity offloading from the IoT devices are utilized to reduce the power consumption. However, despite all the modifications, many tradeoffs have to be made, including low bit rate, high latency, infrequent and small data exchange, and low mobility. In a Smart City scenario, these tradeoffs may not be a big issues to many wide-area, low-data-rate sensing applications, such as environment monitoring, smart metering.

### 4.3.1 LoRaWAN

LoRaWAN is one of the most popular technologies used for long-range communication in IoT. The technology is based on the proprietary chirp spread spectrum [55] which was designed and patented by Semtech. The spreading factor of this modulation can be varied to allow the tradeoffs between throughput, coverage and energy consumption. LoRaWAN operates at Sub-1GHz frequency, mainly at the 915 MHz and the 868 MHz bands but also can operate at 433 MHz and 169 MHz frequencies. LoRaWAN devices can use different bandwidths from 7.8 KHz to 500 KHz depending on the devices' use cases and the advertised bit rate is from 0.3 to 50 Kbps.

The rest of the LoRaWAN stack is open and is governed by LoRa Alliance. At the MAC layer, LoRaWAN is built based on the star topology, where end devices connect to a centralized server via gateways. In this topology, end devices only have to associate with the LoRa server but not with any certain gateway; a message can be received and forwarded by many gateways and duplicated messages will be filtered only at the server. This setup greatly simplify the network management, allowing end devices to move freely between gateways without the signaling overhead as in cellular networks. The channel access mechanism in LoRaWAN is based on the ALOHA protocol.

LoRaWAN specifies three device classes. Class A devices are used for uplink applications (such as monitoring) and only allow a certain amount of time after uplink transmission for downlink communication. Class B devices are used for downlink applications (actuators), allowing downlink data at specific time windows. Class C devices are devices with no constraint on energy consumption and can keep the receiving window always open.

Through various field experiments, the coverage of LoRaWAN in an urban environment can achieve up to 1.2 km [56]. With this range, the whole city can be fully covered with only tens of LoRaWAN gateways, which is very promising for Smart City applications.

### 4.3.2 Sigfox

Introduced in 2009, Sigfox [57] is one of the first technologies available for long range IoT communications. Sigfox operates on the 915 MHz and the 868 MHz frequency bands. Utilizing ultra-narrow bands of only 100 Hz, the technology can achieve very low noise level, low power consumption at the cost of very low bit rate, only around 100 bps. Similar to LoRaWAN, in Sigfox networks, end devices connect directly to base stations in a star topology. For communication range, Sigfox claims to achieve 30–50 km in rural area and 3–10 km in urban scenarios. However, the Sigfox protocols is proprietary and closed. In fact, the business model of Sigfox is to provide IoT connectivity services as a provider, similar to the role of a cellular provider. As a result, deployments using Sigfox depend heavily on whether the Sigfox infrastructure is already available or not and cannot just set up a private infrastructure based on this technology such as in the case of LoRaWAN.

## 4.4 Open Issues

Table 3 provides the summary of the key characteristics of the communications technologies described in Sect. 4.3. It is illustrated that for Smart City applications, there is no one-fit-all solution and tradeoffs have to be made between the coverage, bandwidth, and battery life. Thus, for a complete Smart City solution, there should be multiple technologies co-existing in the same setup. Moreover, it is noted that although these technologies can facilitate connectivity for many current Smart City applications, further researches need to be done to their limitations for a full-scale Smart City deployment. In this section, some of the key open issues for wireless communication in a Smart City scenario are presented.

**Scalability** As the number of connected devices grow to millions of devices, providing an adequate resource allocation scheme for channel access can be very tricky for these devices. The issue becomes more complex when the connected devices do not share the same characteristics in terms of traffic demands, power

**Table 3** Comparison of communication technologies for Smart City applications

	Coverage	Frequency bands	Bandwidth	Bit rate	Topology	Energy consumption
RFID	<10 m	125–134 KHz, 13.56 MHz, 865–956 MHz, 2.4–5.8GHZ	Varies	1–500 Kbps	Star	Low, battery-less
IEEE 802.15.4-based (Zigbee, 6LoWPAN, thread)	Up to a few hundreds of meters	2.4 GHz, 915 MHz, 868 MHz		20–250 kbps	Star, mesh	Low
WiFi	Up to 100 m	2.4 GHz, 5 GHz	20–160 MHz	1 Mbps–3.5 Gbps	Star	High
BLE	Up to 100 m	2.4 GHz	2 MHz	125 Kbps–2 Mbps	Star, mesh	Low
LoRaWAN	Rural 21 km Urban 1.2 km	915 MHz, 868 MHz, 433 HMz, 166 MHz	7.8–500 KHz	0.3–50 Kbps	Star	Low
Sigfox	Rural 30–50 km Urban 3–10 km	915 MHz, 868 MHz	100 Hz	100 bps	Star	Low

demands and quality of service. In this context, an efficient media access protocol has to be scalable to adapt to the dynamicity of the network with nodes joining and leaving at any time but still can maintain a high throughput and fairness in the network. With many of the nodes in the network are battery operated, the protocol has to be also energy efficient and having low complexity.

**Interference** In the Smart City scenario with many wireless technologies co-existing on the same frequency band, interference is inevitable, especially for those using ISM license-free bands. As efficient spectrum sharing is the key factor, adaptive communication schemes should be researched to further increase the spectrum usage efficiency. To this end, regulatory bodies should also take part in regulating the spectrum usage and encourage cooperation between devices/technologies.

**Improvement in bit rate, coverage and battery life** As shown in Table 3, there is no one-fit-all solution for Smart City, for instance, the long-range communications standards can provide a good coverage but their bit rates are so low to accommodate all types of sensors that a Smart City needs. As an example, in our test at McGill University, an average traffic sensor requires a data rate of 500Kbps which cannot be supported by long-range communication standards. It is expected that the requirements for future applications will only increase, better modulation techniques need to be researched to support higher bit rate at longer distance. Battery life is another issue. At a full scale deployment, millions of sensors will be deployed. Maintenance tasks, such as replacement of battery, may be very costly. As a result, longer battery life is expected. In fact, battery-less operation using energy harvesting techniques would be the ideal solution and will be a very interesting and useful research direction.

## 5 Data Security in Smart Cities

As the deployment of Smart Cities become popular and Smart City devices are installed at every street corner, it opens up a new dimension for hackers. On one hand, hackers can make use of these sensors to harvest, modify collected information or to change the behavior of actuators. For example, if the street displays or traffic lights are taken over, the consequences could be very disastrous. On the other hand, these devices can also serve as a pedal for the attackers to levitate the attack to compromise the whole Smart City infrastructure. Compromised IoT devices can also be used for further attacks to other systems, for example, in the DYN DDoS attack in Oct 2016, hundreds of thousands of IoT compromised devices were used to achieve the immense rate of tetra bits per second [58].

As shown in Fig. 2, it is apparent that the only difference in terms of architectural design between the IoT-based Smart City and other ICT systems is the presence of the Data Acquisition and Control layer consisting of sensors and actuators with different capabilities. As a result, the main focus of this section is on the security

measures at this layer as the security at other layers can be realized using available standardized mechanisms.

### ***5.1 Examples of IoT Security Vulnerabilities in Smart Cities***

In [59], the authors explore the vulnerabilities in transportation systems using RFID cards. The investigated systems include car parking tokens, bike renting cards and bus-tram-metro cards. The research highlights several issues in RFID card configuration as well as system structure that allows the attacker to abuse the system by cloning the cards for free rides, and parking.

In [60], the authors reveal several security vulnerabilities which can be exploited in smart city's infrastructure. The types of devices under test range from smart terminals in government offices, airports and bike rental facilities to street cameras. The study shows that the terminals that allow user interactions are not well implemented and attackers can *escape the kiosk mode* (the GUI that constraints users interaction to only designed functionalities, preventing users from accessing the core OS) quite easily through several techniques to enter commands and change configurations. It is also shown that many of these devices store sensitive data such as login credentials in *clear text*, making it easy for retrieving the logins, passwords, payment details or further penetrations. The authors also tested speed cameras installed on the streets and found that the cameras' streams are open without any password protection; moreover, the authors showed the possibility to change the camera configuration to disable detection functions on certain lanes, making the trustworthiness and privacy protection of these cameras questionable.

In [61], the authors show how traffic sensors can be manipulated using wireless connection. Using the *open Bluetooth* connection, the authors demonstrate that sensor's firmware can be changed remotely. Moreover, the studies also show that configuration settings of these devices as well as the related data content can be altered through several techniques, making readings through these devices unreliable.

Through the above examples, it is shown that although each and every new technology and innovation bring new benefits, they also come along with new challenges and problems. In the context of Smart City, dealing with security is a hard problem. Since devices may be exposed to the environment, these devices have a significant disadvantage of lacking physical security. With so much complexity and interdependency between systems, Smart Cities exhibit a large attack surfaces and it is difficult to make sure that everything is secure. Moreover, as the new technologies in Smart City are not quite mature, thorough security tests may not be done properly by the vendors. To add more complexity into the picture, in Smart Cities, there is usually the coexistence of old and new devices and technologies, assuring both interoperation and security at the same time is a difficult task.

## 5.2 *IoT Security Threats, Countermeasures and Research Issues*

As illustrated above, security is a critical aspect in realization of Smart Cities. As a result, this subsection will focus on providing understandings regarding various threats, countermeasures against these threats and issues that are still required further research.

### 5.2.1 IoT Security Threats

The first step to design a secure Smart City is to have a throughout understanding of the attack surfaces and their associated security threats.

#### Physical Vulnerabilities

Due to the nature of a Smart City, in order to collect valuable information and execute appropriate actuations, the IoT network in a Smart City is normally exposed to the public. As a result, attackers may have direct physical access to the IoT devices, which makes these devices vulnerable to various types of attacks.

**Physical Tampering** Having physical access to the device allows the attackers to make modifications on existing devices. Such modifications may involve obtaining access to the collected data, exploiting the actuating functionality of the device, disable the device, or gain remote access to the device for latter control.

**Firmware tampering** Besides the physical modifications, the device can be modified in terms of software/firmware to change the normal operation of the hardware, or insert a backdoor for remote access later on. This type of attack is very hard to detect as the IoT device may be both physically and operationally similar to a normal device.

**Information extraction** In some cases, either the hardware or the software of the IoT device is not the interest of the attacker but the information it contains such as cryptographic information, login credentials, and device identities. Such information could be very critical to the integrity of the network as a whole. For example, the attacker could exploit these information to escalate his attack to the system core to gain more access and control.

**Node replication** Having physical access to the device, the attacker can replicate the IoT devices to add malicious nodes to the system. Being low cost, it could be assumed that replication of the IoT nodes is not very hard nor expensive. With just a few malicious node inserted, the attacker can conduct several types of attacks [62] to significantly degrade the performance of the network such as insert malicious packets, corrupt or send duplications of legit packets.

## Vulnerabilities in Communications

Even in the case of not having the physical access to the IoT devices, having constant communication transactions to and from the devices, most of the cases through wireless communications, still gives a wide surface for the attackers.

**Eavesdropping** When the communication is on wireless media, it is very hard to restrict the reach of the communications. As a result, the attacker can sniff the ongoing conversations to gain critical information about the device and the network, especially when the communication is unencrypted. Even when the communication is encrypted, there is not a 100% peace of mind as the vulnerabilities in the security protocols could be exploited, such as the exploit of the famous WPA2 protocol for WiFi recently [63].

Even if the communications channel is encrypted, by listening to the ongoing conversations, and monitoring the electromagnetic spectrum, critical information can be collected. For example, the leaked electromagnetic signature of a device might help to reveal information about the devices [64]. This type of attack is a very critical threat in health care system as patients' privacy might be seriously compromised.

**Denial of Service (DoS) attacks** In wireless communications, as the data transfer happens over the air, it is hard to prevent any interaction from the attacker over this open media. As illustrated below, there are several variations of this type of attack.

*Battery draining* Battery-powered IoT devices rely on a very short duty cycle to maintain their long operation life on the field. In this type of attack, the attacker tries to disturb the duty cycle of the device to quickly drain the battery of the device, making it unavailable. One way to implement this attack is to bombard the target node with fabricated packets to keep the device awake or to disturb its sleeping schedule.

*Jamming attack* In this type of attack, the attacker sends interference signals to occupy the channel, preventing any communication over it. A variation of this attack is to jam the channel intermittently so that the jam behaves similar to packet loss and is harder to detect and prevent.

*Replay attack* Even in cases where communication are encrypted, the attacker can still degrade the network performance by capturing legit packets and then re-insert these legit packets into the network, with or without modification. This type of attack can be used along with battery draining attack or exploiting the vulnerabilities in the security protocols.

**Routing attack** If the attacker inserts malicious nodes into the network, many types of attacks can be escalated. The attacker can inject falsified information to disrupt the network routing, creating routing loops, disturb packet forwarding in the network.



*Selective forwarding* In this type of attack the malicious node randomly drops some of the packets in the network. The dropped packets can be targeted to a selected node or group of nodes.

*Black Hole, Gray Hole attacks* In Black Hole and Gray Hole attacks, the malicious node alters the routing information to route all the packets in the network to itself for further processing or dropping them all together or randomly. This action creates congestion and increase energy consumption of other nodes in the network.

*Worm Hole attack* In this type of attacks, the malicious nodes confuse the routing mechanisms by advertising a shorter route to the destination node. The attacking nodes capture the packets and relocate them to another location via a tunnel.

*Hello flood attack* In this type of attack, the malicious node uses a high transmission power to send Hello messages to the target network to claim to be the neighbor of all the nodes in the network. This type of attack can be conducted even if the node is not a member of the target network by capturing and replay the Hello messages. In this case, the routing information will be disrupted and the attack can be used as a form of jamming and battery draining attack.

### 5.2.2 Countermeasures and Research Issues

With the understanding of the various threats to the Smart City infrastructure, many studies have looked into methods to mitigate these threats.

#### Countermeasures and Research Issues for Physical Security

**Physical tampering** When the circuitry of an IoT device is tampered with, the new elements will affect the operation of the original circuit in one way or another. Due to this, testing the characteristics of the device can be used to detect the presence of the malicious elements. These tests are based on the timing, power and the heat distribution of the device [30]. Among those tests, the timing and power tests are promising as they can be done remotely just with a firmware update. However, it is remarked that the operational of the circuitry may change with time depending on the working conditions which could raise false alarms. Also, after a new firmware is updated, the testing results have to be updated to accommodate the changes.

**Firmware tampering, Information extraction** In order to protect the information/firmware of the IoT device, many schemes can be applied. A cryptographic co-processor can be integrated to the circuit for security hardening. On one hand, this co-processor will aid the central processing unit in processing sophisticated cryptography operations. On the other hand, this piece of hardware can be used to store and protect cryptographic information such as keys and credentials as well as the hash signature of the hardware and software configurations. When the system boots up, the hash signature will be verified, if the verification fails, the system

will be halted. As such, this hardware can be used both to verify the integrity of the system as well as to protect sensitive information. One example of this device is the Trusted Platform Module (TPM) which was standardized by International Organization for Standardization [65]. However, it is noted that security at the hardware level might hinder the flexibility of the system in the future. Also, once the technology is compromised, mass replacement of the deployed hardware must be performed, which may be very costly.

**Secure re-configurations** During the life cycle of an IoT device, it is necessary for the device to be reconfigured and updated regularly to catch up with the functionality requirements and to apply security patches. An attacker can exploit this batch operation to inject malicious modifications into the devices' firmware. Although few studies have looked at this issue [66], a secure framework for mass reconfiguration is still open for future research. Furthermore, most of the IoT devices allows local firmware update through the use of JTAG, USB or COM ports. Defining a secure framework for authenticating the use of local update and checking the authenticity and integrity of the new firmware is still open for research.

**Node replication** Preventing insertion of malicious nodes in its essence is the question of device authentication. There are two main approaches to deal with this issue.

*Multi-factor authentication (MFA)* Traditional authentication of devices are normally based on symmetric keys, in some cases authenticating the joining of nodes in a network is governed by only one key, which makes it easy to be compromised. The idea of MFA is to use different authentication factors for checking the device identity. One notable example of this scheme is the use of Physical Unclonable Function (PUF) [67]. PUF is a noisy function that is embodied in to an integrated circuit. During the authentication process, PUF generates a response for a challenge based on its unique physical structure. However, this approach requires modifications of the chip manufacturing process which may be costly. Moreover, MFA can be conducted based on other information such as the context of the device (location, neighbor devices, etc.). Although MFA is already widely used in everyday applications such as banking, the use of MFA on IoT is still limited and still open for research.

*Blockchain* While traditional authentication process is based on a centralized database, either stored on a server or at nodes as symmetric keys, blockchain provides a different approach by creating a distributed database over a peer-to-peer network. Since the database is distributed, in order to "hack" the system, the attacker has to "hack" more than half of the nodes in the system which is extremely difficult. As such, when a device is compromised, this compromised device can be eliminated easily. Although the concept is quite successful in crypto-currency such as Bitcoin [68], applying this idea to IoT world is still an open research question.

## Countermeasures and Research Issues for Communication Security

**Eavesdropping** Using encryption schemes is a common way to protect a communication channel. However, most of the encryption schemes for wireless communications are based on symmetric key algorithms which could be vulnerable in case the key is compromised. Although various studies already tried to implement public key encryption to IoT devices, the limited processing power, memory and battery capacity of these constrained devices is still an obstacle to overcome [69]. As a result, designing a light-weight but secure encryption algorithm based on public key and digital certificates for constrained devices is still an open research issue.

**Routing attack** To protect from routing attacks, secure routing protocols should be designed. Through the use of random numbering, encryption, and message authenticity check, protection from falsified information could be achieved. For instance, in Routing Protocol for Low-power and Lossy Networks (RPL) [70], the use of random numbers and hash values can enable protection against replay attacks and data authenticity check.

## Countermeasures Against Complex, Future Attacks

Security is a very dynamic domain, new threats and types of attacks are evolving daily. To cope up with these attacks that are not yet documented, new mechanisms of security should be developed using data analytics and machine learning to quickly identify the new attack and come up with a counter measure to protect the system and its data.

Intrusion Detection System (IDS) [71] is one of the technologies that relies on pre-defined rules or signatures to identify the abnormal events in the network. This type of IDS can detect known attacks effectively and reliably, however, it is not effective against new threats as the signatures of these are not yet learnt. To overcome this limitation, IDSs based on anomaly detection are proposed to identify new attacks by learning the normal behaviors of the devices and systems in the past. This approach can identify many types of DoS attacks as well as new attacks; however, since any abnormal events may trigger an alarm, this approach has high false alarm rate. In addition, to be able to learn, IoT devices must be equipped with adequate memory space and processing power, which may hinder the cost of deployment as well as their battery life. As a result, effective, adaptive but efficient self-learning framework to identify and proactively react to new security threats on resource-constrained devices still requires a lot of research efforts.

## 6 Conclusion

This chapter provides an overview of IoT infrastructures for enabling Smart Cities. In particular, current trends and examples of Smart Cities are discussed. Then, a four-layer architecture is proposed to accommodate the general framework for building Smart Cities. As wireless communications will be the key enabling technology for providing communications and connections of a massive number of IoT devices, a survey on wireless communication standards for Smart Cities is presented along with discussions on their pros and cons, and use cases. With IoT devices installed in public areas, security becomes a critical consideration. Acknowledging this issue, a survey is provided to highlight the security threats, countermeasures and open issues.

## References

1. S. P. Mohanty, U. Choppali, E. Kougianos, "Everything you wanted to know about smart cities: The Internet of things is the backbone", *IEEE Consumer Electronics Magazine*, Vol. 5 (3), 2016.
2. L. Hou, S. Zhao, X. Xiong, K. Zheng, P. Chatzimisios, M. S. Hossain, W. Xiang, "Internet of Things Cloud: Architecture and Implementation", *IEEE Communications Magazine*, Vol. 45 (12), pp. 32-39, 2016.
3. R. Mohideen, R. Evans, "Shaping Our Technological Futures", *IEEE Technology and Society Magazine*, Vol. 34 (4), pp. 83-86, 2015.
4. H. Yue, L. Guo, R. Li, H. Asaeda, Y. Fang, "DataClouds: Enabling Community-Based Data-Centric Services Over the Internet of Things", *IEEE Journal on Internet of Things*, Vol. 1 (5), pp. 472-482, 2014.
5. V. Zdraveski, K. Mishev, D. Trajanov, L. Kocarev, "ISO-Standardized Smart City Platform Architecture and Dashboard", *IEEE Pervasive Computing*, Vol. 16 (2), pp. 35-43, 2017.
6. C.-I Chang, C.-C. Lo, "Planning and Implementing a Smart City in Taiwan", *IEEE IT Professional*, Vol. 18 (4), pp. 42-49, 2016.
7. A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities", *IEEE Internet of Things Journal*, Vol. 1 (1), pp. 22-32, 2014.
8. J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications", *IEEE Internet of Things Journal*, Vol. PP (99), pp. 1-17, 2017.
9. C. Yin, Z. Xiong, H. Chen, J. Wang, D. Cooper, "A literature survey on smart cities", *Science China Information Sciences*, Vol. 58 (10), pp. 1-18, 2015.
10. S. Pellicer, G. Santa, A. L. Bleda, R. Maestre, A. J. Jara, et al. "A Global Perspective of Smart Cities: A Survey". *IEEE Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 439-444, 2013.
11. S. Madakam and R. Ramachandran, "Barcelona Smart City: The Heaven on Earth (Internet of Things: Technological God)", *ZTE Communications*, Vol. 13 (4), pp. 3-9, 2015.
12. P. Liu and Z. Peng, "China's Smart City Pilots: A Progress Report", *Computer*, Vol. 47 (10), pp. 72-81, 2014.
13. L. Sanchez, L. Mun˜oz, J. A. Galache, P. Sotres, J. R. Santana, "SmartSantander: IoT experimentation over a smart city testbed", *Computer Networks*, Vol. 61, pp. 217-238, 2014.
14. J. Guti´errez Bayo, "International Case Studies of Smart Cities: Santander, Spain", *Inter-American Development Bank*, 2016.

15. A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, "Smart Cities: A Survey on Data Management, Security and Enabling Technologies", *IEEE Communications Surveys & Tutorials*, Vol. PP (99), 2017.
16. S. Talari, M. Shafie-khah, P. Siano, V. Loia, A. Tommasetti, "A Review of Smart Cities Based on the Internet of Things Concept", *Energies*, Vol. 10 (4), 2017.
17. S. Mitchell, N. Villa, M. Stewart-Weeks, and A. Lange. "The Internet of Everything for Cities: Connecting people, Process, Data, and Things to Improve the 'Livability' of Cities and Communities". *Cisco*, pp. 1–21, 2013.
18. GSMA – "Guide to Smart Cities: The Opportunity for Mobile Operators, GSMA Smart Cities"
19. M. Khan, "Las Vegas Valley Water District using IoT technology for water infrastructure management" – Available online: <https://www.iot-now.com/2015/07/08/34535-las-vegas-valley-water-district-using-iot-technology-for-water-infrastructure-management-and-leak-detection/>
20. M. Bouskela, "The Road toward Smart Cities: Migrating from Traditional City Management to the Smart City", *Inter-American Development Bank*, 2016.
21. J. Frazier, T. Touchet, "Transforming the City of New York", *Cisco*, 2012.
22. J.-S. Hwang and Y. H. Choe, "Smart Cities Seoul: a case study", *ITU-T Technology Watch Report*, pp. 1–20, 2013.
23. D. Amar Flo´rez "International Case Studies of Smart Cities: Medellin, Colombia", *Inter-American Development Bank*, 2016.
24. J. Gubbi, R. Khan, R. Zaheer, S. Khan, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", *Future Generation Computer Systems*, Vol. 29 (7), pp. 1645-1660, 2013.
25. R. Mahmoud, T. Yousuf, I. Zualkernan, "Internet of Things (IoT) Security: Current status, Challenges, and prospective measures", *IEEE Internet Technology and Secured Transactions Conference*, 2015.
26. L.-M. Ang, K. P. Seng, A. M. Zungeru, G. K. Ljemarku, "Big Sensor Data Systems for Smart Cities", *IEEE Internet of Things Journal*, Vol.4 (5), pp. 1259-1271, 2017.
27. J. Granjal, E. Monteiro, J. S. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues", *IEEE Communications Surveys and Tutorials*, Vol. 17 (3), pp. 1294-1312, 2015.
28. A. A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, "Internet of Things: A Survey on Enabling technologies, Protocols, and Applications", *IEEE Communications Surveys and Tutorials*, Vol. 17 (4), pp. 2347-2376, 2015.
29. "The Internet of Things reference model", Cisco, 2014 – Available online: [http://cdn.iotwf.com/resources/71/IoT\\_Reference\\_Model\\_White\\_Paper\\_June\\_4\\_2014.pdf](http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf)
30. A. M. Nia, N. K. Jha, "A Comprehensive Study of Security of Internet-of-Things", *IEEE Transaction on Emerging Topics in Computing*, Vol. PP (99), pp. 1-1, 2017.
31. ITU-T Y.2060 – Overview of the Internet of things, 2012.
32. M. Presser and A. Gluhak, "The internet of things: Connecting the real world with the digital world". *EURESCOM The Magazine for Telecom Insiders* 2, 2009.
33. A. Kandhalu, A. Rowe, R. Rajkumar, C. Huang, and C.-C. Yeh, "Real-Time Video Surveillance over IEEE 802.11 Mesh Networks", *IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 205-214, 2009.
34. M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing", *Journal of Network and Computer Applications*, Vol. 67, pp. 99-117, 2016.
35. M. Aazam, I. Khan, A. A. Alsaar, and E.-N. Huh, "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved", *International Bhurban Conference on Applied Sciences & Technology (IBCAST)*, pp. 414-419, 2014.
36. A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey", *Future Generation Computer Systems*, Vol. 56, pp. 684-700, 2016.
37. Amazon Web Services. Accessed 2017. 2006. url: <https://aws.amazon.com/>
38. IBM Bluemix. Accessed 2017. 2014. url: <https://www.ibm.com/cloud-computing/bluemix/>

39. Microsoft Azure. Accessed 2017. 2014. url: <https://azure.microsoft.com/>
40. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey", *Computer Networks*, Vol. 54(15), pp. 2787-2805, 2010.
41. A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey", *Future Generation Computer Systems*, Vol. 56, pp. 684-700, 2016.
42. R. Want, "An Introduction to RFID Technology", *IEEE Pervasive Computing*, Vol. 5 (1), pp.25-33, 2006.
43. A. Luvisi, G. Lorenzini, "RFID-plants in the smart city: Applications and outlook for urban green management", *Urban Forestry & Urban Greening*, Vol.13 (4), pp. 630-637, 2014.
44. IEEE 802.15.4 Std. - 2015 - Available Online: <https://standards.ieee.org/findstds/standard/802.15.4-2015.html>
45. IEEE 802.15.4 Std. - 2003 - Available Online: <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>
46. IEEE 802.15.4 Std - 2006 - Available Online: <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>
47. Zigbee Alliance - <http://www.zigbee.org/>
48. J. Song, Y. K. Tan, "Energy consumption analysis of ZigBee-based energy harvesting wireless sensor networks", *IEEE International Conference on Communication Systems*, 2012.
49. IETF RFC 4944, "Transmission of IPv6 Packets over IEEE 802.15.4 networks", 2007.
50. IETF RFC 6282, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-based Networks", 2011.
51. Thread Group - <https://threadgroup.org/About>
52. IEEE 802.11 working group - <http://www.ieee802.org/11/>
53. Bluetooth SIG - <https://www.bluetooth.com/>
54. M. Centenaro, L. Vangelista, A. Zanella, M. Zorzi, "Long-Range Communication in Unlicensed Bands: The Rising Star in the IoT and Smart City Scenarios", *IEEE Wireless Communications*, Vol. 23 (5), pp. 60-67, 2016.
55. N. Sornin, M. Luis, T. Eirich, T. Kramp, O. Hersent, "LoRaWAN Specification", *LoRa Alliance*, 2015.
56. M. Centenaro, L. Vangelista, A. Zanella, M. Zorzi, "Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios", *IEEE Wireless Communications*, Vol. 23 (5), pp. 60-67, 2016.
57. Sigfox - <http://www.sigfox.com/en/>
58. E. Bertino, N. Islam, "Botnets and Internet of Things Security", *IEEE Computer*, Vol. 50 (2), pp. 76-79, Feb. 2017.
59. Abusing smart cities – Available Online: [http://securingsmartcities.org/wp-content/uploads/2016/09/BECCARO\\_abusing-smart-cities.pdf](http://securingsmartcities.org/wp-content/uploads/2016/09/BECCARO_abusing-smart-cities.pdf)
60. Fooling the "Smart City" – Available Online: [http://securingsmartcities.org/wp-content/uploads/2016/09/Fooling-smart-city\\_in\\_template.pdf](http://securingsmartcities.org/wp-content/uploads/2016/09/Fooling-smart-city_in_template.pdf)
61. Trick traffic sensors – Available Online: <https://securelist.com/how-to-trick-traffic-sensors/74454/>
62. E. Leloglu, "A Review of Security Concerns in Internet of Things", *Journal of Computer and Communications*, Vol. 5, pp. 121-136, 2017.
63. M. Vanhoef, F. Piessens, "Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2", *ACM Conference on Computer and Communications Security*, 2017.
64. A. M. Nia, S. S-Kolay, A. Raghunathan, N. K. Jha, "Physiological Information Leakage: A New Frontier in Health Information Security", *IEEE Transaction on Emerging Topics in Computing*, Vol. 4 (3), pp. 321-334, 2016.
65. Trusted Platform Module – ISO/IEC 11889-1:2015.

66. B.-C. Choi, S.-H. Lee, J.-C. Na, J.-H. Lee, "Secure firmware validation and update for consumer devices in home networking", *IEEE Transactions on Consumer Electronics*, Vol 62 (1), 2016
67. C. Wachsmann, A.-R. Sadeghi, "Physical Unclonable Functions (PUFs): Applications, models, and future directions", *Synthesis Lectures on Information Security, Privacy, and Trust*, Vol. 5 (3), pp. 1-91, 2014.
68. Bitcoin - <https://bitcoin.org/en/>
69. S. L. Keoh, S. S. Kumar, H. Tschofenig, "Securing the Internet of Things: A Standardization Perspective", *IEEE Internet of Things Journal*, Vol. 1 (3), pp. 265-275, 2014.
70. J. Granjal, E. Monteiro, J. S. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues", *IEEE Communication Surveys & Tutorials*, Vol. 17 (3), pp. 1294-1312, 2015.
71. B. B. Zarpelao, R. S. Miani, C. T. Kawakani, S. C. de Alvarenga, "A survey of Intrusion detection in Internet of Things", *Journal of Network and Computer Applications*, Vol. 84, pp. 25-37, 2017.

# The Role of 5G and IoT in Smart Cities



**Attahiru Sule Alfa, Bodhaswar T. Maharaj, Haitham Abu Ghazaleh,  
and Babatunde Awoyemi**

**Abstract** Smart Cities are envisioned to offer its citizens a plethora of services that are aimed at optimizing the use of public assets and improving the quality/efficiency of the regular activities. The performance of these services are reliant on the inter-networking of information between the different and disjointed systems with overlapping purposes. The information exchange process will likely involve the communication of large volumes of data with a range of complex requirements that are application-specific. In this chapter, the role of the 5G systems and Internet of Things (IoT) for meeting such stringent requirements are discussed along with the essential constituents of a successful Smart City infrastructure.

## 1 Introduction

The current and future vision of Smart Cities is that of modern urban development that integrates information and the Internet of Things to manage and control most of the activities and assets of the cities. Examples of assets and activities include transportation systems, hospitals and health care systems, water supply, waste management, schools, law enforcement, local communication systems, etc. Some

---

A. S. Alfa (✉)

Department of Electrical and Computer Engineering, University of Manitoba,  
Winnipeg, MB, Canada

Department of Electrical Electronic and Computer Engineering, University of Pretoria,  
Pretoria, South Africa

e-mail: [Attahiru.Alfa@umanitoba.ca](mailto:Attahiru.Alfa@umanitoba.ca)

B. T. Maharaj · B. Awoyemi

Department of Electrical Electronic and Computer Engineering, University of Pretoria,  
Pretoria, South Africa

e-mail: [sunil.maharaj@up.ac.za](mailto:sunil.maharaj@up.ac.za)

H. Abu Ghazaleh

Department of Engineering and Computer Science, Tarleton State University,  
Stephenville, TX, USA



popular examples of smart city features in existence include parking systems that communicate in all of a city's main car parks displaying information about free space locations, and smart waste bins that let local garbage collection companies know when they need to be collected. General activities and assets are closely intertwined, such as the local road traffic systems and law enforcement that are closely linked. Another example is enabling vehicles to communicate with other systems such as traffic signals, which can assist with changing the light sequence in real time and in response to the vehicular traffic flow, thus improving the overall traffic flow in a city network. One feature that has gradually been introduced in several cities (now a Smart City idea) is making available to smart phones the state of emergency waiting room queues, information that can be used by a prospective emergency patient to decide which facility to go to in order to minimize the waiting time for service.

The key items that would make smart cities succeed are excellent sources of data collection, transmission of data, analysis, inference, management and control. Given that the expected growth of the number of devices to be connected to the mobile network is about 50 billion by 2020 [1], a Smart City of the future would need to be implemented using fifth generation (5G) networks, which would provide higher and faster transmission capabilities, larger bandwidth and higher data capacities. Most of the communication would be device-to-device. The high transmission speeds are an essential feature of these devices to allow them to rapidly respond to observations so a quick decision-making process by the devices is enhanced, as most of the monitoring and response systems are in real time. Therefore, it is clear that 5G networks and the Internet of Things (IoT) will play a huge role in the development of the Smart City infrastructure.

The aims of this chapter are:

- To formally define and present the expected features of a Smart City.
- To present examples of activities carried out in a Smart City system.
- To present the current state of the art and examples of well-functioning Smart Cities.
- To identify the current state of how the IoT works and assists in achieving a Smart City, and to identify how it can further enhance Smart Cities.
- To identify the state of communication technologies in cities and how the new 5G technology would enhance the operation of Smart Cities.
- To consider resource allocation in cognitive radio networks and its application to Smart Cities.

The next section presents the commonly-used definitions for a Smart City along with our perception of what truly defines a Smart City. Some of the key components of the Smart City infrastructure and an example of a Smart City system are also covered in the same section. An overview of the Internet of Things and Wireless Sensor Networks, as well as their relevance in Smart Cities, are given in Sect. 3. The highlights of 5G networks and their importance in Smart Cities are reviewed in Sect. 4. The need for 5G networks in the Internet of Things and their roles in the development of a Smart City infrastructure are given in Sects. 5 and 6, respectively.

We conclude this chapter in Sect. 7 with a few research ideas and future directions for achieving an efficient and effective Smart City infrastructure.

## 2 Smart Cities

In this section, we present our all-encompassing version for the definition of a Smart City (SC). This would assist the reader in understanding the context in which we present our ideas and thoughts. First, we point out that we have surprisingly not been able to find a single clear definition of an SC. Most of the definitions presented by others seem to explain how it is achieved rather than what it is. For example, Comarch [2] explains an SC by presenting it as *“using modern technology to improve urban space; interacting with citizens to improve quality of life.”* Examples of activities listed by Comarch include smart parking and mobility, smart services, smart citizenship, smart retail, smart entertainment and smart payment. The idea of “smart” simply implies the involvement of a two-way interaction between parties at the least, such as between the citizens and the parking facilities, which may also involve the exchange of information. OpenDataSoft [3] explains SC as follows; *“The idea behind the Smart City is motivated namely by the optimization of costs, organization, and well being of city residents.”* Again, this is an explanation rather than a definition. However, it does emphasize a different aspect, the “optimization of costs, organization and well being of city residents”. This definition, again, captures the interaction aspect, but with the additional emphasis on ensuring that it is done optimally. Finally, a lengthy but clearer definition of SC (given in Wikipedia [4]) and what is expected to be achieved is quoted as follows: *“a smart city is an urban development vision to integrate information and communication technology (ICT) and Internet of things (IoT) technology in a secure fashion to manage a city’s assets. These assets include local departments’ information systems, schools, libraries, transportation systems, hospitals, power plants, water supply networks, waste management, law enforcement, and other community services. A smart city is promoted to use urban informatics and technology to improve the efficiency of services. ICT allows city officials to interact directly with the community and the city infrastructure and to monitor what is happening in the city, how the city is evolving, and how to enable a better quality of life. Through the use of sensors integrated with real-time monitoring systems, data are collected from citizens and devices then processed and analyzed. The information and knowledge gathered are keys to tackling inefficiency.”* Part of this definition is based on the writings of Musa [5], in which he states: *“A smart city is defined as a city that engages its citizens and connects its infrastructure electronically. A smart city has the ability to integrate multiple technological solutions, in a secure fashion, to manage the city’s assets that include, but not limited to, local departments’ information systems, schools, libraries, transportation systems, hospitals, power plants, law enforcement, and other community services. The goal of building a smart city is to improve the quality of life by using technology to improve the efficiency of services and*

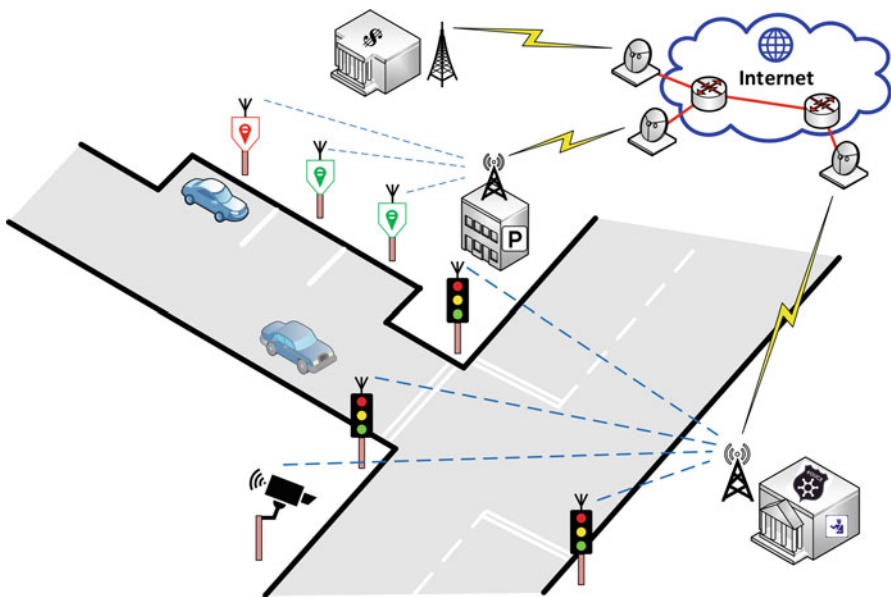
*meet residents' needs. Business drives technology and large-scale urbanization drives innovation and new technologies. Technology is driving the way city officials interact with the community and the city's infrastructure. Through the use of real-time control systems and sensors, data are collected from citizens and sensors and then processed in real-time (Poslad, Athen, Zhenchen & Haibo, [6]). The information and knowledge gathered are keys to tackling inefficiency, which leads to optimizing systems. A smart city offers technological solutions to tell what is happening in the city, how the city is evolving, and how to enable a better quality of life."*

We gather from all these definitions and explanations that a well-operating SC depends on the following:

- **Effective wireless sensor networks (WSN)** We require that the sources of sensed data together with the wireless network that supports them and that forms a WSN to be effective. The sensing and data collection have to be carried out as frequently as needed to obtain dynamic time-current information. Not only would the data collected have to be current, but also quickly analyzed and passed on to the appropriate destination nodes where they would be used to fully understand the important and related phenomena and make effective decisions.
- **Superior, reliable and secure communication networks** An exceptional network is needed for the WSN functions to perform extremely well in terms of latency and high capacity for excellent throughput. Reliability is another very critical requirement of the network in that we should be confident that it is always, or almost always, available. While gathering and sending data for decision-making purposes, it is critical that the information in the data is not inadvertently tampered with in any form, especially by a malicious entity. This could lead to information (especially sensitive information) falling into wrong hands or data being modified, leading to incorrect interpretations of the information that they are carrying. Hence, the security of the information data that are collected is another important aspect.
- **Big data** With the abundance of data collection and interpretations to be made for the purpose of analyzing the system for useful information and for making certain decisions, there is bound to be a big amount of data involved in the process. While the size of the data is a factor, more important are the ways to analyze them. The Big data issue is a major research area in itself and will not be addressed in this chapter. However, its relevance for SCs should not be overlooked.
- **Cooperating citizens and municipal authorities** To explain this requirement better, consider the example of one of the facilities that can be smartly controlled in an SC, namely the vehicular traffic light system in a city. Suppose a city traffic department is interested in incorporating a smart traffic lights network to augment its current and traditional traffic lights system (most likely a vehicle-actuated one). In such systems, the durations of the green, yellow and red lights are altered in response to the amount of traffic crossing the particular actuators placed on the street legs close to the intersection. Additional sensors placed near the intersections and along the streets (done by the municipality) would allow

for the dynamic collection and real-time processing of traffic flow information for estimating the duration of wait times at the intersections. The municipality would then be able to compute and pass on these estimates to cooperating citizens (drivers) through their devices. This would allow the drivers to switch their routes based on their own perception. Hence, strong cooperation between citizens and the municipality would be necessary for implementing an SC in this case. This example can be transferred to all other aspects of SC.

An example of smart parking and mobility was presented earlier as one of the expected features of SCs. Let us explore this further. In a smart parking system there would be sensors located around parking facilities that can sense the presence (or absence) of a parked vehicle at a given location. An online database of such information should be made available to citizens on a continuous basis, making it easier for them to determine the locations of vacant parking spots. Also, once a citizen parks a vehicle, information about the vehicle should be made available immediately to the central (municipal) office so as to update the information on available and unoccupied parking spots, as well as the start of metering to determine how much a user is being charged for the parking. A further interesting feature would be to have embedded in this smart parking system a smart financial system. Hence, several inter-woven features can be involved in a smart system, such as the example shown in Fig. 1 for the smart parking system. After the user vacates the parking spot, the municipality would immediately be aware of the newly available spot and can proceed with the transaction to bill the user instantly, which can be



**Fig. 1** An example of a smart city solution with inter-woven systems for smart parking

carried out using the smart payment system. It is quite clear that this smart parking system would have various requirements to function properly, such as very efficient and effective WSNs with sensors at the parking area, a strong communication system for the municipality, along with the users having powerful and personal devices with multiple capabilities (e.g. for receiving messages). For this smart parking system, it is easy to see why and how 5G technology and the IoT will play a major role. The details of such roles will be covered in later sections.

Even though SC has no clear definition and has different meanings to different people, a common element agreed upon by all is that the aim of having an SC is to drive economic growth and improve the quality of life of its citizens through enabling the technologies that help to achieve those goals. However, one question that still needs to be answered is “what constitutes a Smart City?”, keeping in mind that an SC does not necessarily need to be developed from scratch. Most parts of the infrastructure will be the result of retrofitting existing systems to achieve it. Furthermore, a single aspect of an SC may be sufficient to declare that a city is a smart city, or qualify it to be named as one. For example, can a city with a smart traffic system discussed earlier be declared an SC, or just called a city with a smart traffic system? If this city now also has a smart parking system, would the overall system constitute an SC or a city with two smart systems? There are currently no clear distinctions or boundaries in terms of the number and types of smart systems a city is required to have in order to be labeled an SC. Thus, we prefer to use a simple descriptor, such as “a city with smart solutions”. The following list contains examples of some of the current smart solutions:

- E-governance and citizen services
- Waste management
- Water management
- Energy management
- Urban mobility

Based on the earlier discussions, the following are some of the main issues that need to be considered and clearly addressed:

- What constitutes an SC?
- What benefits do SCs bring to our society and quality of living?
- What elements do we need to ensure the interoperability of the different systems for achieving an SC?
- What will be the greatest challenges to be faced in establishing an SC?

While citizens, municipalities and system developers have a strong desire to ensure the success of an SC, some of the greatest challenges that will be faced at the earlier stages of the development include providing sufficient network capacities, maintaining low latencies, ensuring high reliability and security for efficient systems operation. Furthermore, it is essential that we try to minimize direct manual (human) involvement in a reliable and efficient SC. This is equivalent to introducing semi-automation capabilities into the various systems and allowing them to inter-operate with minimal physical contact or remote access. As pointed out earlier, the number

of devices expected to be involved in such systems will be growing fast. Even though the gathering of data by the devices will not be an issue in such systems, the processing of the data along with the prompt delivery of the results to the appropriate nodes in the system will be one of the major challenges. The ability to achieve such tasks in real-time will greatly depend on the network that interconnects these different systems. Therein lies the importance of the role that 5G and IoT will have in turning SC into a reality, as discussed in the subsequent sections.

### **3 Internet of Things and Wireless Sensor Networks**

The Internet has, and continues to serve a vast number of users in a variety of ways. It has transformed the manner in which we conduct many of our regular activities, which has helped with efficiently utilizing our valuable resources (such as time) and thereby improving our quality of life. This was made possible by the plethora of network applications available for diverse operations and purposes. It has further aided rapid advancements in certain fields such as health care and education. The invention of the Internet was seen as the next evolution after the development of electronic computers in the 1950s. It provided the mechanism to inter-connect the various computing systems for the purpose of exchanging information and media collaboration, regardless of their geographic locations.

Our society, in general, has been heavily reliant on Internet services. Based on the data reported in [7], the number of Internet users has been steadily increasing since its adoption by the global community. The rate of that increase has also risen, especially in the last decade, and many have predicted the trend to continue in future years. Only 0.4% of the global population were connected to the Internet in 1995, whereas 49.5% of the total population in 2016 were utilizing the same and improved services, with a wider variety of applications. The data support the evidence of the higher rate in the growth of Internet adoption when compared with the increase in the world population. The advent of wireless communication technologies also furthered the widespread usage of Internet services. This has facilitated “anywhere and anytime” connectivity to the Internet and promoted the development of numerous mobile applications. According to a forecast report published by Cisco [8], there were 8.0 billion mobile devices with Internet connectivity in 2016, with an estimated increase to 11.6 billion devices by 2021 (exceeding the world’s projected population at that time of 7.8 billion). This does not include the number of objects, or “things”, that will be connected to the Internet as part of the SC infrastructure.

People have ordinarily been the main contributors to the data traffic on the Internet. The generation and consumption of data on the network have typically been initiated by humans through the use of computers. However, there has been a proliferation of other smart devices with computing capabilities in recent years, many of which include the means for connecting to a wired and/or wireless network. These smart devices are expected to be coupled with or embedded within objects for the purpose of monitoring and/or enhancing the performance of the

system surrounding the objects. Several solutions have been proposed that require interaction between such devices for establishing a smart environment. The Internet is seen as the best solution to facilitate the connectivity and interaction between those smart objects. Hence, in this post-PC era, we will witness the Internet's next phase of evolution from being an "Internet of computers" to the "Internet of Things" (IoT) [9] that will facilitate seamless connectivity between numerous smart devices in an SC.

Many have visualized the IoT to be conceived of various parts and solutions. However, the prevailing perception is that the IoT will provide the common infrastructure for heterogeneous and smart devices, thus unifying their services for the ultimate goal of improving the quality of life of its users. The Internet can allow for both people and objects to communicate their data for the sake of attaining a better, safer and harmonious environment for our society. Hence, the Internet will not only have to support man-to-machine data transfer, but will also need to sustain machine-to-machine communication, with the latter expected to be the dominant source of data traffic in the IoT. In addition to the expected growth in the number of users accessing the Internet, the IoT is further estimated to add a large number of smart devices to the network, especially ones with wireless communication capabilities. While the IoT is not limited to wireless devices, such devices are expected to be the dominant types of objects because of our dynamic lifestyles.

Various devices are being developed for the purpose of supporting and improving many of our traditional tasks that have typically been done manually. Such devices include electronic health trackers and smart environmental control systems. The trend has been to incorporate intelligence and wireless connection capabilities into these devices to enhance their performance and user experiences. This intelligence is made possible through the inclusion of sensing technologies that permit the devices to become more aware of their surroundings, and to aid further in any of the actions to be decided by the device. In the early proposals of IoT, it was envisioned that *"we need to empower computers with their own means of gathering information, so they can hear, see, and smell the world for themselves, in all its random glory"* [10]. Hence, sensor devices and their technological advancements are expected to play a leading role in the realization of the IoT and SC. Other objects are also expected to be a significant part of IoT, such as Radio-Frequency Identification (RFID) technologies [11]. The RFID systems consist of tags and readers used for automating the tracking of people and objects within a certain geographic area. Given the recent (and continuing) advances in the areas of microelectronics and communications, a combination of sensing and RFID technologies are expected to be embedded in a multitude of devices, where the sensing process is likely to be more active than the others.

Sensing devices have been employed in a range of applications such as in military operations, environmental monitoring systems, health monitoring systems and home automation. Their most popular form of deployment includes expanding their functionality with wireless connectivity, thus enabling them to network their activities with other sensor devices and within their system surroundings. This further allows

for extending their operations to broader and remote locations. The networking of the wireless sensors has been an attractive solution for many applications because of their scalability, reliability, flexibility and ease in deployment [12]. In a WSN, these sensing devices can readily communicate their data in a tetherless and ad hoc manner for the purpose of sharing their sensed data with neighboring nodes, or relaying those results to designated nodes known as “sinks”.

WSNs have been provisioned in a variety of services, with many new applications emerging owing to their popularity in addressing several and complex challenges [13]. WSNs have commonly been used for sensing and tracking ambient conditions, such as temperature, pressure, water and seismic levels. The sensing devices can be deployed in large and/or inaccessible regions, with their data being used for monitoring and forecasting certain environmental conditions or natural disasters. There is also growing interest from the civil sector in incorporating WSNs into public structures such as roads, bridges and towers. This enables the “health monitoring” of the structures for discovering any potential weaknesses or faults in time. Another typical application that has benefited from the deployment of WSNs is in the field of health care for monitoring the different vital signs of a mobile patient, which may also aid with early emergency response. In general, the devices used in WSNs have some constraints that limit their performance, such as limited energy capacities, short transmission ranges, and low processing capabilities. These challenges must be addressed in order to help expand the functionality and operations of WSNs. Furthermore, and given the wide range of applications that are expected to include WSNs, the communication technologies of the future will need to be developed in a manner that will readily support these systems.

WSNs are traditionally deployed autonomously, without any direct interaction with other networks. In certain systems, such as military applications, the isolation of the WSN from other systems is vital for reasons such as security and safety. However, there are many services that can potentially be improved if data were shared among multiple and disjointed WSNs. For example, the WSNs deployed for vehicular networks, traffic light systems and smart parking systems can together alleviate traffic congestion problems and improve users’ driving experience if the sensed results from these WSNs were collectively made available to the different networks of sensors. However, there is no current mechanism to assist with the sharing of information among the different WSNs. Any sharing of data that is needed today between the different WSNs is done manually and requires human interaction. This manual process of data distribution can be problematic since it is prone to both delays and errors. A better alternative would be to permit the sensing devices to communicate their results directly with the devices in the other WSNs and without any human intervention. Hence the need for SC solutions.

The Internet has been identified as the ideal global infrastructure to facilitate the collaboration of data between many WSNs. This is due to its widespread availability almost everywhere. This would help to enable cooperation between different WSNs and empower the sensing devices to seek supporting information from devices on other networks. Accordingly, the Internet of the future will not only need to support



the exchange of information between people and their computers, but also the sharing of data between objects that will predominantly be comprised of sensing devices. These devices can either operate individually, or as part of a collection of sensing devices in a WSN, where the latter is destined to be the most popular mode of operation. Hence, the IoT is also viewed as a “network of networks” that is largely for supporting machine-to-machine communication between diverse sensor devices and networks [14].

In the previous section, it was deduced that the ultimate aim of an SC architecture is to promote the quality of the many services offered to the community. In turn, this can lead to more efficient use of public resources, while also increasing certain rewards and reducing various costs. The rapid growth in population density in cities have prompted the need to seek systems that are capable of intelligently handling the limited resources available to communities. Such gains can only be achieved if these systems were empowered to offer “smart” choices. Information from numerous WSNs can collectively enhance the decision-making process of the many systems through the sharing of relevant results between sensing devices in the different networks. Hence, WSNs are a key component in the SC architecture. Moreover, the collaboration between several WSNs is required to support the smart decisions made by the systems, which are reliant on the IoT. A reliable, scalable and secure network architecture for the IoT is further needed to reinforce the systems in SCs, which must also be capable of supporting the connectivity to a large number of heterogeneous and smart devices with substantial volumes of (or big) data. Therefore, an SC is only conceivable with a vast number of WSNs that operate cooperatively through the IoT.

The WSNs operating within the IoT in SCs promise to provide numerous benefits to the social, economic, safety and environmental prosperity of communities [15, 16]. These technologies can also help to shape and stimulate the progress of society better. There are numerous ways in which such systems can help to accomplish these developments. One example involves integrating the information between sensors in smart parking systems, vehicular networks and traffic monitoring systems. The sensor data from the different systems can collectively assist individuals with finding and reserving the optimal and available parking lot that is nearest to their intended destination. This form of smart service can shorten the time taken to drive around seeking an empty lot, which consequently helps to reduce traffic on the roads. In addition, this solution can further lessen the fuel consumption of drivers, which can lead to lower pollution levels and a safer environment, thus promoting a healthier city.

The previous example assumes the existence of mutual agreements on the sharing of information between different WSNs. However, there may be other scenarios in which certain systems should be selective about how their data are shared with other networks. An example of such a scenario could be the communication between the security and surveillance systems managed by law enforcement agencies and vehicular networks. Collaboration between the two systems can assist law officers with their pursuit of reckless drivers and vehicles, as well as alerting vehicles to any potential dangers on the road. In this example, the sensed information from the

vehicular networks are expected to be shared with others, while the surveillance network may be limited to only communicating alerts to the vehicles and the sensed information being restricted to authorized personnel.

Various challenges are being faced in the development of the future IoT that will provide the fundamental infrastructure for SCs [17, 18]. Some of these issues are hardware-centric such as the energy efficiency and processing capabilities of the devices. However, more profound concerns are related to the integration of the prospective heterogeneous systems and networks with extreme reliabilities and quality of service. The future IoT must be capable of supporting inter-communication among a large number of devices and at a growth rate that is several magnitudes higher than the existing rate. Furthermore, the newly anticipated applications are expected to involve the sharing of large volumes of data across the networks. Some operations may even have strict requirements in terms of their data rates and latency. Ultimately, the IoT should be adept at managing an abundance of bandwidth for a large number of devices, and scalable at supporting the growing service demands with the needed quality. Other and necessary advancements involve developing an infrastructure that is capable of protecting the privacy of the information flowing in the network, as well as securing the network against failures. A further development that is being investigated is the proposal of empowering the objects in the IoT and SC with cognitive capabilities [19], such as self-learning and self-understanding abilities. Overall, many of these challenges cannot be undertaken with the current networking technologies, but can be addressed with the advent of the 5G communication network.

## 4 5G Networks

The Fifth Generation (5G) network is the newly evolving paradigm of wireless communications. 5G networks – in form, content and reach – are making laudable promises of providing applications that would yield very high social and economic values, thus having a far-reaching impact on our world [20]. The promises of 5G are quite expansive and broad, especially in terms of its potential to bridge the digital divide. As a result, the hype and hysteria about 5G have been astonishing. Fundamentally, 5G will achieve better performance in terms of capability, capacity, speed, latency, etc., than current technologies such as WiMAX, LTE and LTE-Advanced [21]. At its core, 5G will provide a platform for connecting new industries and devices, empowering new user experiences, enabling new services and delivering new levels of efficiency in ways we have never experienced before [22]. It will enhance mobile broadband, improve response time and capabilities during emergencies or mission-critical situations and will make the inter-connecting of millions, if not billions, of objects/devices possible over the Internet. Unequivocally then, 5G is distinguishing itself as the most invaluable technology in the development of SCs and for achieving the highly anticipated hyper-connected world.

The need for 5G is premised on the continued exponential growth in broadband wireless mobile communication demands, both in developed and developing countries across the globe. By 2011, the number of Internet users already exceeded 2.2 billion [7]. It is estimated that by 2019, mobile communication through phones alone will have reached a staggering 5.07 billion of the world population [23]. It is estimated that globally, by 2020 the number of connected devices relying on wireless connectivity for their operations will have risen in the order of tens of billions [24]. Therefore, it is clear that the current demands for wireless connectivity now and in the near future cannot be adequately catered for by prevalent technologies, such as WiMAX, LTE/LTE-Advanced, despite their attempts at improving their impact and reach. A new technology that can meet this high demand is needed, and 5G is emerging as the answer to that demand.

Another key reason for a new and improved technology in 5G is that expectations of mobile communication have continued to change very rapidly over the years. From analog phone calls that provided only voice mobility services, to digital phone calls and messaging that brought about mass penetration, to Internet experience and then broadband with lower latency, the way in which people use mobile devices continues to change dramatically. More recently, the combination of mobile broadband networks and smartphones has significantly enhanced the mobile Internet experience. Already across the world, various reports have shown that LTE customers consume around double the monthly amount of data of non-LTE users, and in some cases three times as much [25]. The data demand is expected to keep rising as new applications emerge and SCs take shape across the globe. Only 5G has the scale and scope needed to meet and sustain future expectations of wireless mobile communications and inter-connectivity between smart devices in an SC.

The improvements in wireless communication experience promised by 5G are quite remarkable, necessitating the description of 5G as a generational shift in communication. Even though the exactness in specifications is still an ongoing debate, recent works on 5G are providing some clarity, with sufficient compromises and consensus being reached. The most consistent highlights of 5G, in terms of specifications and goals, are summarized as follows [25, 26]:

- **Speed** 5G networks should be capable of driving communication at speeds in order of 1–10 Gbps.
- **Latency** The latency requirement is less than 1 ms for 5G networks.
- **Coverage** In terms of a perception of 100% coverage.
- **Density** 1000 times bandwidth per unit area.
- **Connectivity** Values of 10–100 times more connected devices than what is currently obtainable through LTE networks.
- **Availability** A perception of 99.999% availability.
- **Reliability** 100% reliability of all 5G networks.
- **Energy/cost efficiency** 90% reduction in network energy usage and up to 10 years of battery life for low-power, machine-type devices.

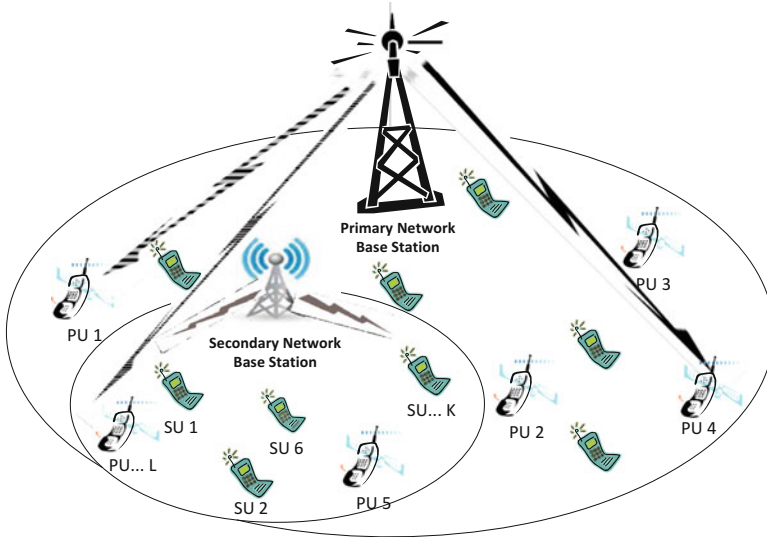
It is important to note that some of these 5G specifications/goals are conflicting and it would be herculean to achieve them simultaneously within a single communica-

tion network. No use case, service or application has been identified that requires all the above-mentioned performance specifications across an entire network at the same time. Indeed, some of these specifications cannot actually be linked to use cases or services, but are instead aspirational statements of how networks should be built, independent of service or technology [25].

As promising as 5G is, its workability and effectiveness will rely heavily on the availability of wireless communication resources. Indeed, 5G will have a massive number of connected devices, including the many smart devices in SCs, that will require large bandwidths and more spectrum resources than current technologies. A potential limitation to the prospects of 5G networks is thus the scarcity of spectrum and other resources needed for driving 5G operations. Note that addressing the resource limitation issues would consequently improve multiple aspects of the network, such as throughput, latency and capacity. Spectrum (resource) management and optimization are therefore some of the most important aspects of 5G. While resource allocation problems in wireless communications are not entirely new, the size and scale of 5G pose new and/or unique resource allocation challenges which may not have been identified or pronounced in current technologies such as LTE/LTE-Advanced. For instance, the thousand-fold increase in traffic demand of 5G for SC applications may be impossible to be handled by frequency slicing methods such as the OFDMA being used in LTE/LTE-Advanced networks. Therefore, non-orthogonal multiple access methods are currently being considered and developed for 5G. One other important approach being investigated for addressing resource allocation problems in 5G is the development of solution models that are capable of spectrum sharing and/or simultaneous double usage of a spectrum space. A good example of this spectrum-sharing arrangement is cognitive radio networks (CRNs) developed for 5G applications.

In a typical CRN developed for 5G, a primary network is made to operate alongside a secondary network over the same spectrum space in an underlay, overlay or hybrid mode [27]. In the underlay mode, the users in the secondary network (called secondary users (SUs)) are allowed to use the entire spectrum space of the primary network as long as they do not cause interference to the primary users (PUs) above a certain predetermined and permissible value. In an overlay mode, the SUs use the spectrum opportunistically. The SUs use the spectrum when the PUs are not transmitting but must sense the presence of the PUs and vacate the spectrum once the PUs arrive and are ready to transmit. The hybrid mode combines both underlay and overlay to achieve even better results, though the system becomes more complex to develop, analyze and implement. A practical scenario that fits the CRN arrangement for 5G is when the primary network is designed as a macrocell, while the secondary network is designed as a femtocell or picocell (or a combination of femtocell and picocell) and both networks are made to work simultaneously to improve the coverage or the quality of service experience of the users/clients.

A typical CRN arrangement with a primary-secondary network for 5G is shown in Fig. 2. With such arrangements, a marked improvement in spectrum usage and overall productivity of 5G can be achieved. The implication of such primary-secondary network arrangements is the immediate provision of the additional



**Fig. 2** A model of primary-secondary networking for 5G applications

capacity needed to sustain the multitude of wireless smart devices that are part of the SC infrastructure.

5G will be a key player in the design and eventual realization of smart cities. It would not be an overstatement to claim that most of the newly emerging and rapidly evolving smart services (such as e-health, e-transport, e-banking, e-farming, and e-security) that are currently being developed to drive the realization of SCs will rely heavily on the successful roll-out of the 5G network. Some of the specific ways in which 5G will help drive the realization of SCs are highlighted as follows:

1. **5G will be instrumental in connecting a massive number of devices** 5G will connect wireless networks to billions of devices, such as cars, home appliances, machinery and wearable technology. Innovative localities will use SC technologies such as connected sensors and data to provide municipal services more efficiently and effectively [28]. For example, monitors in dumpsters will communicate with sanitation trucks when the dumpsters are full and should be emptied. Most of the communication needs of smart cities will rely on 5G to be met.
2. **The economic impact of 5G in SCs will be enormous** As telecommunication operators expand their networks, they are expected to invest across the country. It has been projected that in the United States alone, telecommunication operators will invest approximately \$275 billion over the next 7 years to deploy 5G wireless technology, with trials beginning as early as 2017 in selected cities [29]. 5G is expected to create 3 million new jobs and boost annual gross domestic product (GDP) by \$500 billion, driven by the projected \$275 billion investment from telecommunication operators. Thus, the investments in 5G and SCs will drive

innovation and increase productivity, leading to massive job creation and GDP growth across the globe.

3. **New and improved services will be provided through 5G** In smart cities, transportation, distribution, finance and energy services will be connected to networks and will be able to interact together and provide more reliable, convenient and environmentally aware new services. Moreover, residents in SCs will have seamless access to these services without needing to know about the networks on which they are based [30].
4. **5G will be useful for mission-critical applications in smart cities** Mission-critical applications such as automotive, robotics and health services will require ultra-low latency, high availability, high reliability and strong security for their effectiveness in SCs [22]. 5G will be that platform to drive such applications in SCs.

## 5 Role of 5G in IoT

While 5G is predominantly the newly emerging communication standard, IoT is the Internet reality for the near future. IoT describes the possibility of simultaneously and seamlessly inter-connecting several objects or things (devices, machines, structures, buildings, gadgets etc.) through the Internet to facilitate the provision of effective and efficient autonomous services, with as little human intervention and/or participation as possible [31]. In differentiating the traditional Internet from the newly developing IoT, the main distinguishing characteristic is that “things” in IoT is a broader and more encompassing term. Unlike in the regular Internet where the connecting devices over which inter-networking occurs are mostly computers, “things” in IoT are not limited to traditional computers, but include all matters with which it is possible to connect, and/or such matters between which the exchange of information and communication is achievable. The inter-connection of IoT “things”, coupled with the embedding of software that collects data and analyzes results in a timely manner, makes it possible for IoT to provide intelligent services beyond what is achievable by the traditional internet [32].

Although 5G and IoT are distinct technologies, one significantly impact the other. Essentially, while 5G is strictly a telecommunications paradigm, IoT is a much broader technology which, to a large extent, encapsulates major aspects of 5G. IoT can be viewed, in fact, as a progressive combination of the Internet, broadband wireless mobile communication (of which 5G is its newest development), WSNs, and heterogeneous networks (HetNet) to form a single but powerful inter-connected communication network. From the foregoing, it can be established that although the goal and prospects of IoT cannot be single-handedly realized by 5G technology, 5G will certainly be one of the greatest enablers and a key driver for the realization of viable and vibrant IoT networking. This is due to the promises of 5G providing higher coverage and availability, increased network density in terms of cells and devices, along with improved data rates and latency [33]. Therefore, it is important

to note that this interplay between 5G and IoT technologies, as well as the striking linkage in their prospects and goals, makes these technologies the most potent duo for the achievement of the much-anticipated SCs and a hyper-connected world.

It has already been established that spectrum and other resources needed to drive 5G and IoT are scarce and limited. The limitation in resource availability is a major threat to these new and daring telecommunication paradigms. The reality of non-ubiquity in resource availability for accomplishing the goals of 5G and IoT is therefore an underlying problem of these technologies. To overcome the limitation of resource scarcity, appropriate resource optimization models/schemes for 5G and IoT are needed. Resource allocation (RA) optimization in communication networks describes mechanisms for achieving the utmost productivity for the networks, thus overcoming the limitation in resource availability. For 5G and IoT networks, the goal of RA is to coordinate the distribution and utilization of the limited resources efficiently so as to achieve an overall optimal performance for these networks. The two most powerful tools, optimization and queueing analysis, can be employed for developing appropriate RA models for 5G and IoT (e.g. see [34] and [35]), which ultimately improves the performance of the smart services in SCs.

Several kinds of data traffic are generated and transmitted in typical 5G and IoT environments. Moreover, most 5G and IoT services have real-time and reliability expectations. Long delays and/or packet losses in the course of transmitting data can have a significant, undesirable impact on the overall performance of the smart services that are supported by the 5G and IoT infrastructure. Network links in 5G and IoT are often limited in the amount of bandwidth over which transmission could occur, while servers in the processing layer might also have finite capacities. This is usually the case in practical IoT networks such as in machine-to-machine communication. In such cases, machine-type devices seeking to gain access to the network may at times experience delays, loss of packets and/or connectivity. Therefore, optimizing congestion control and queue management is imperative in IoT, if the performance requirements of the service provisioning are to be met. Congestion control and queue management in 5G and IoT can only be achieved by employing appropriate queueing models. Some of the previous work such as those carried out in [35] and [36] are examples of how queueing models can be employed in addressing RA and access control to mitigate network performance degradation caused by congestion and other queueing-related occurrences.

In general, the following aspects of 5G and IoT networking are identified as open-ended problems that still require active and adequate research/investigation, if the prospects of 5G and IoT are to be fully realized.

1. **Spectrum availability for service provisioning** Radio-frequency spectrum is a necessary resource for effective 5G and IoT networking. The limitation in spectrum availability poses a big challenge to 5G and IoT functionality. New attempts at improving spectrum availability and usability, such as those currently being developed in CRNs, are imperatives for successful 5G and IoT networking. Similarly, investigating the most appropriate optimization techniques that can

help in improving spectrum allocation for 5G and IoT networks is an important area of research.

2. **Improving sensing capabilities of “things”** The efficient deployment of the IoT will rely heavily on the capability of its “things” to sense accurately and timeously for making quick decisions. Poor sensing will greatly hamper the development of IoT. Hence, a major research focus is that of designing and providing very reliable sensing mechanisms for connected objects in IoT. Importantly, more investigations of optimization techniques that can be employed to improve the time, resources, and the number of sensors required for an effective IoT realization are necessary.
3. **HetNet integration, coordination and management** IoT can be viewed as an example of a HetNet. This is true not only from the perspective of the computation capability of the various objects within the network, but also from the viewpoint that networks and communication technologies used for inter-connecting their objects and the services they offer are going to be heterogeneous in nature [37]. Both 5G and IoT will run on a number of HetNet designs and applications, as well as a deep reliance on cloud computing, with the aim of achieving swift, smooth and seamless communications between the different devices and services. However, HetNet in itself has been undergoing a great deal of research and development in recent years. Establishing the appropriate linkages between macrocells, microcells, picocells and femtocells, together with combining wireless communication standards (such as Wi-Fi, 3G, 4G, and LTE-Advanced) with other technologies such as fibre optics and relay networks is what HetNets seek to accomplish.
4. **Optimizing system functionality that incorporates scalability and robustness requirements** Efficient deployment of application components in 5G and IoT networks has often been viewed as a multi-objective optimization problem. In many cases, the objectives are at conflict with one another (e.g. performance versus energy consumption). In such cases, obtaining unique solutions that simultaneously optimize the multiple objectives can be a challenging task, and resource trade-offs have to be made. It is likely that Pareto optimization would have to be employed in addressing such problems. However, because IoT is essentially an open-ended ecosystem of heterogeneous resources, this makes the crisp definition of Pareto-optimal solutions difficult because of an incomplete view of the external factors and uncertain circumstances that might influence the optimality of such solutions.
5. **Improving system security** The importance of security is paramount in the deployment of both 5G and IoT. Arguably the most demanding of concerns and/or requirements for the widespread realization of many of the IoT visions is its security [38]. There are a number of threat implications concerning an expanding IoT, as described in [39]. Hence, it is imperative that the IoT technologies such as the RFID and sensors be adequately protected from malicious attacks. There are also security issues at the transmission or transport layer. Of serious concern is the possibility of cross-heterogeneous network attacks. The application layer also poses security issues, such as being able to select the same database content



according to different access, and providing user privacy information protection. Strengthening the safety and security of the various parts of the 5G and IoT infrastructure, such as the transport layers cross-domain authentication and cross-network authentication, is a crucial problem to be addressed.

- 6. Addressing the problems of large communication overheads and efficient data management** The authors in [40] argued that the traditional approach of migrating raw data to centralized points for data storage and analysis was likely to incur debilitating communication and energy costs, which could affect the environment negatively in the future. Developing models to optimize communication overhead and storage mechanisms are therefore necessary, as these two usually have the most significant impact on the amount of energy the network consumes.

## 6 The Role of 5G and IoT in Smart Cities

Cities worldwide have been undergoing rapid growth in their population and this trend is expected to continue in the years to follow. This is mainly owed to the influx of dwellers from rural areas because of the many amenities offered by urban life. The increase in city residents require the expansion, and possibly the advancement, of current public infrastructure and services to sustain this rise in population. Hence, urban planners will be faced with the challenge of finding efficient solutions to meet the needs of city residents, along with refining the quality of living standards. Several ideas have been proposed in the literature [15, 41] that seek to enhance various parts of the city's infrastructures and services, such as waste management, traffic congestion mitigation, city energy consumption monitoring and vehicle parking management. These solutions are intended for smartly managing and administering public resources and services with the aim of offering residents the ultimate experience, thus transforming the region into an SC.

As stated earlier, an SC is envisioned as an assortment of smart environments, each managed by its own system, but with the ability to operate and/or communicate with other systems. An example of such a system includes Smart Homes for monitoring various characteristics and controlling relevant apparatus. The network of smart devices within a particular household could further be inter-connected with other Smart Home systems for security purposes and as part of a Smart Neighborhood Watch system [42]. This inter-connectivity between the different networks is to be provided by the IoT infrastructure. SC systems have been implemented in different regions and environments for the purpose of examining the potential of new smart system applications, along with identifying the infrastructure needs for supporting these future services [15, 41]. The studies from these initiatives concluded the strong need for a framework that is capable of supporting a very large number of devices with high volumes of data to be communicated across many systems, both reliably and seamlessly.

The extraordinary rise in the expected number of smart devices that will be inter-connected across the Internet is primarily due to the continuing advances in Information and Communications Technologies (ICTs), along with our growing reliance on electronic equipment for managing/improving our regular activities. The majority of these devices are likely to have wireless capabilities and to be designed to utilize any of the commonly available communication networks such as WiFi, WiMAX, Bluetooth, ZigBEE and cellular networks. The heterogeneity in medium access prevents these devices from directly communicating with other devices that are operating in different networks. However, the IoT framework can provide the infrastructure needed to permit the communication between these heterogeneous devices, with the Internet being the common networking technology for establishing the link between all these devices. Numerous sensor network solutions currently exist whose services can be enhanced if they are empowered with inter-networking capabilities (i.e. communicating with other sensor networks), which could further influence the development of newer and advanced solutions.

The increase in the number of smart devices and solutions in the near future will also lead to a surge in the volumes of data communicated across the devices and network. The data transfer patterns are also predicted to digress from today's models of man-to-man communication owing to the forthcoming services and devices using machine-to-machine communication. Hence, the traditional networks should be augmented to have increased capacities along with supporting high data-rate and low latency transmissions. Even though the Internet will provide the common infrastructure for linking the devices on different networks, the Internet alone cannot sustain this growth. Moreover, many of these devices are expected to utilize the wireless medium as their primary means of data communication. 5G wireless networks are anticipated to support a diverse range of service demands and be capable of maintaining those demands for a huge number of devices. The 5G network would also provide the foundation for linking the myriad of wireless devices on disjointed networks, which ultimately provide the backbone for the fixed and mobile devices in the various sensor networks. Therefore, the adoption of 5G networking technologies into the IoT is necessary for bolstering efficient and reliable communication between diverse devices and applications. This will help foster the environment of smart devices, thus enabling the enhancement of countless services and experiences by permitting those devices to coordinate with other systems, such as linking vehicular networks and smart traffic systems with smart parking systems.

Some of the highly anticipated features of 5G network connectivity are high data-rate transmissions and very low latencies. The services in an SC are envisioned to have the capability of retrieving data in a timely manner and of making a decision at the appropriate times. Many of the smart devices will rely on receiving their data in real-time for their decision-making process that is related to the operations within the city, and for aiding its citizens, with minimal delay. It is therefore crucial for the network to efficiently handle the real-time data for these smart devices that may be needed for executing urgent actions. Hence, success in the deployment of SC solutions will be contingent on the IoT providing the necessary infrastructure

for merging the different systems, with the 5G networking technology facilitating the reliable inter-communication for a large number of devices. However, not all services within the SC environment require their data to be communicated at high rates and lower latencies, with some applications being content with modest data-rates. In such applications, the data may either be retrieved for processing delayed actions, or be accumulated at a remote server for performing offline data analysis. The results from the data analysis can further be used for the benefit of urban planning [41]. Despite not all applications requiring high data-rate transmissions with low latencies, the exuberant features of the 5G wireless networks will aid in supporting efficient and reliable inter-networking services for a much larger set of smart devices.

In essence, the SC ecosystem can only be realized with a networking infrastructure that can support inter-communication services for tens of billions of devices. Furthermore, this infrastructure must be competent at delivering data across the network at high speeds with extremely low latencies to support those devices that need to execute real-time decisions. The majority of these smart devices currently operate within a closed system. In other words, they are limited to communicating with other and similar devices in the same network. The IoT will aim to furnish the common framework for uniting the different networks and systems under a single platform using the Internet. This would entail regulating a common communication protocol between the heterogeneous smart devices, e.g. IP, or an enhanced version of it that is tailored for IoT. But the IoT alone is insufficient at granting the interoperability between the various smart devices and systems due to their utilization of networking technologies with limited ranges and capacities, such as WiFi and WiMAX. Furthermore, the success of the IoT is contingent on the availability of a network anywhere and anytime for the contiguous intercommunication between the smart devices, such as cellular networks, or 5G.

The development of 5G networking technologies is aimed at addressing one of the primary factors that continues to limit the performance of cellular networks, and that is spectrum shortage. This requirement is essential for the IoT and for supporting the intercommunication between the anticipated large numbers of smart devices. Given that the majority of these smart devices will be untethered, their data communications are expected to be done wirelessly, with the upcoming 5G network being the prominent technology for conducting these wireless transmissions. The 5G wireless networks are favored for their promising features of supporting a large number of connections with high data-rates and low latencies. Hence, the objects in the SCs of the future will be developed with the capabilities of communicating their data across the Internet, with the 5G network providing the wireless connectivity at the edges of the network. Smart devices are also expected to be developed with cognitive radio capabilities that aim to exploit some of the under-utilized parts of the radio spectrum. This can further aid with increasing the throughput of the devices and effectively expanding the capacity of the network. Other initiatives are in place for optimizing the various parts of the SC systems, such as incorporating cloud computing into the infrastructure, with the intent of relieving the devices from some of their data processing powers and for energy-saving purposes.

## 7 Conclusions and Future Directions

In this chapter, the role of 5G and IoT in the SCs have been discussed. One aspect that comes to light from this is that achieving a smart city involves the integration of several smart systems. In essence, an SC is considered to involve the inter-networking of several modules called smart systems. A key future direction is to focus the research on efficient and effective modules, the interconnection of the modules and access to information and interaction by users (citizens). Ultimately, the aim should be to achieve the following:

- Design each smart system as a self-contained unit that can interact with other systems efficiently. This would require optimizing the performance of each system in terms of ensuring that the available resources are efficiently used with minimal latency for a rapid centralized data accumulation and decision-making process.
- Design a communication network that interconnects and passes messages and data among several smart systems. In passing messages and data to other smart systems the communication network will be competing with other networks users. It is known that the growth of network users (Internet and wireless communication networks users) will increase at an alarming rate and that is why the idea of 5G is promising. It is in this context that we see 5G having a major impact on SCs. However, excellent RA models are needed for it to be successful.
- Ensure that citizens, who will be using all the services provided in SCs, have easy access to the information generated and using sufficiently powerful applications with convenient viewing capabilities, i.e. good displays. As the number of users and devices will be increasing over time, the demands on the system will be huge, with a further need to ensure that resources are made available to manage these demands and manage them well. Hence, the need for an effective IoT.

Given the immense volumes of data that are expected to be involved in the communications between the various smart systems, the development efforts of both the IoT and 5G systems must address the limited RA issues for such systems. This can be achieved through the improvement of the existing allocation schemes, the proposal of new schemes, spectrum sharing, data coding and compression strategies, as well as many others. These efforts are necessary for expanding the future network's capacity and throughput capabilities, which are needed for supporting the range of services for a large number of users and systems. This would further strengthen the performance of the network and systems, thereby ensuring the efficient operation of the smart services. In addition to the field testing of the proposed developments, considerable efforts are needed to model the different proposals mathematically in order to examine both their short-term and long-term potentials. The results from such analyses can aid with determining further improvements along with optimizing certain operations in the systems with the aim of maximizing their performance. Other aspects that are of prime importance for realizing SCs include the following:

- optimizing the heterogeneity aspects of 5G and IoT networking,
- minimizing the latency of data transfers across the networks,
- efficiently managing large volumes of (big) data,
- expanding on the applications and inter-networking of wireless sensor networks,
- energy savings and harvesting in smart devices,
- enhancing the performance of the embedded systems in smart devices, and
- economic implications of 5G and IoT deployment.

The 5G and IoT systems are planned for supporting the growing demands of wireless mobile communications and the sharing of large volumes of data between different devices. These two different systems are inter-related and cannot be developed independently. Therefore, the development of both the 5G and IoT infrastructure should be done in unison. Moreover, SC solutions and systems are anticipated to be the major users of these systems in the near future. Hence, these development efforts should jointly factor in the supporting of SC applications. This would help to ensure the achievement of a desired level of performance for network users, and more importantly SC systems.

**Acknowledgements** This research is partly funded by the Advanced Sensor Networks SARCHI Chair program, co-hosted by University of Pretoria (UP) and Council for Scientific and Industrial Research (CSIR), through the National Research Foundation (NRF) of South Africa.

## References

1. “The Internet of Things [Infographic],” <https://blogs.cisco.com/diversity/the-internet-of-things-infographic>, 2011.
2. Comarch, [[https://smartcity.comarch.com/\\$#\\$about](https://smartcity.comarch.com/$#$about)].
3. Opensdatasoft, [<https://www.opendatasoft.com/smart-cities-solution/>].
4. Wikipedia, [[https://en.wikipedia.org/wiki/Smart\\_City](https://en.wikipedia.org/wiki/Smart_City)].
5. S. Musa, “Smart Cities - A Roadmap for Development”, *Journal of Telecommunications Systems and Management*, 2016, 5:144. <https://doi.org/10.4172/2167-0919.1000144>.
6. S. Poslad, A. Ma, Z. Wang, and H. Mei, “Using a smart city IoT to incentivise and target shifts in mobility behaviour - Is it a piece of pie?”, *Sensors*, Volume 15, Issue 6, pp. 13069–13096, 2015.
7. S. C. Mukhopadhyay and N. K. Suryadevara, “Internet of Things: Challenges and Opportunities,” *Internet of Things: Challenges and Opportunities*, Springer International Publishing, pp. 1–17, 2014.
8. “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper,” <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>, February 2017.
9. F. Mattern and C. Floerkemeier, “From the Internet of Computers to the Internet of Things,” *From Active Data Management to Event-Based Systems and More*, pp. 242–259.
10. K. Ashton, “That ‘Internet of Things’ Thing: In the real world, things matter more than ideas,” *RFID Journal*, June 2009.
11. A. K. Evangelos, D. T. Nikolaos and C. B. Anthony, “Integrating RFIDs and smart objects into a Unified Internet of Things architecture,” *Advances in Internet of Things*, Volume 1, pp. 5–12, 2011.

12. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, Volume 38, Issue 4, pp. 393–422, March 2002.
13. P. Rawat, K. D. Singh, H. Chaouchi and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of Supercomputing*, Volume 68, Issue 1, pp. 1–48, April 2014.
14. L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, Volume 54, Issue 15, pp. 2787–2805, October 2010.
15. A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, Volume 1, No. 1, February 2014.
16. H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah and P. Siano, "IoT-based smart cities: A survey," *IEEE 16th International Conference on Environment and Electrical Engineering*, Florence, 2016, pp. 1–6.
17. J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, Volume 29, Issue 7, pp. 1645–1660, September 2013.
18. S. Hammoudi, Z. Aliouat, and S. Harous, "Challenges and research directions for Internet of Things," *Telecommunication Systems*, Springer Science+Business Media, pp. 1–19, July 2017.
19. Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, and K. Long, "Cognitive Internet of Things: A New Paradigm Beyond Connection," *IEEE Internet of Things Journal*, Volume 1, No. 2, April 2014.
20. A. Gohil, H. Modi, and S. K. Patel, "5G technology of mobile communication: A survey," *2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*, pp. 288–292, March 2013.
21. S. Kumar, G. Gupta, and K. R. Singh, "5G: Revolution of future communication technology," *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 143–147, October 2015.
22. Qualcomm, "Leading the world to 5G," *Qualcomm Technologies, Inc.*, February 2016.
23. Statista, "Number of mobile phone users worldwide from 2013 to 2019;" [<http://www.statista.com/statistics/274774/forecast-of-mobile-phone-usersworldwide/>], 2016.
24. K. Katzis and H. Ahmadi, "Challenges Implementing Internet of Things (IoT) Using Cognitive Radio Capabilities in 5G Mobile Networks," *Internet of Things (IoT) in 5G Mobile Technologies*, Springer International Publishing, pp. 55–76, 2016.
25. D. Warren and C. Dewar, "Understanding 5G: Perspectives on future technological advancements in mobile," *gSMA Intelligence*, The Walbrook Building, 25 Walbrook, London EC4N 8AF, December 2014.
26. A. Alnoman and A. Anpalagan, "Towards the fulfillment of 5G network requirements: technologies and challenges," *Telecommunication Systems*, Uolume 65, No. 1, pp. 101–116, May 2017.
27. B. S. Awoyemi, B. T. Maharaj, and A. S. Alfa, "Resource allocation for heterogeneous cognitive radio networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1759–1763, March 2015.
28. Y. Getachew, A. Montoya-Boyer, and S. Overton, "5G, smart cities and communities of color," *Joint Center for Political and Economic Studies*, June 2017.
29. M. A. Amine, K. Mathias, and T. Dyer, "How 5G can help municipalities become vibrant smart cities," [<http://www.accenture.com/strategy>], 2017
30. K. Hamaguchi, Y. Ma, M. Takada, T. Nishijima, and T. Shimura, "Telecommunication systems in smart cities," *Hitachi Review*, Volume 61, No. 3, pp. 152, 2012.
31. Z. Yu and W. Tie-ning, "Research on the visualization of equipment support based on the technology of internet of things," *Second International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, pp. 1352–1357, December 2012.
32. P. Pereira, J. Eliasson, R. Kyusakov, J. Delsing, A. Raayatnezhad, and M. Johansson, "Enabling cloud connectivity for mobile internet of things applications," *IEEE 7th International Symposium on Service Oriented System Engineering (SOSE)*, pp. 518–526, March 2013.

33. N. Al-Falahy and O. Y. Alani, "Technologies for 5G networks: Challenges and opportunities," *IT Professional*, Volume 19, No. 1, pp. 12–20, January 2017.
34. L. Lei, D. Yuan, C. K. Ho, and S. Sun, "Joint optimization of power and channel allocation with non-orthogonal multiple access for 5G cellular systems," *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, December 2015.
35. C. Oh, D. Hwang, and T. Lee, "Joint access control and resource allocation for concurrent and massive access of M2M devices," *IEEE Transactions on Wireless Communications*, 2015.
36. J. Huang, D. Du, Q. Duan, Y. Sun, Y. Yin, T. Zhou, and Y. Zhang, "Modeling and analysis on congestion control in the internet of things," *IEEE International Conference on Communications (ICC)*, pp. 434–439, June 2014.
37. A. Athreya and P. Tague, "Network self-organization in the internet of things," *IEEE International Workshop of Internet-of-Things Networking and Control (IoT-NC)*, pp. 25–33, June 2013.
38. T. Xu, J. Wendt, and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 417–423, November 2014.
39. M. Covington and R. Carskadden, "Threat implications of the internet of things," *5th International Conference on Cyber Conflict (CyCon)*, pp. 1–12, June 2013.
40. N. Ali and M. Abu-Elkheir, "Data management for the internet of things: Green directions," *2012 IEEE Globecom Workshops*, pp. 386–390, December 2012.
41. M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the Internet of Things using Big Data analytics," *Computer Networks*, Volume 101, pp. 63–80, June 2016.
42. X. Li, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart Community: An Internet of Things Application," *IEEE Communications Magazine*, Volume 49, Issue 11, pp. 68–75, November 2011.

# Leveraging Cloud Computing and Sensor-Based Devices in the Operation and Management of Smart Systems



**Shikharesh Majumdar**

**Abstract** Cloud computing and sensor-based internet of things are two important technologies that are driving the realization of smart cities. This chapter focuses on the use of clouds and sensor-based devices in monitoring and managing smart facilities such as bridges, industrial and aerospace machinery and smart applications. Three different roles of cloud computing are discussed. The first concerns the unification of diverse resources required for collecting and analyzing the data monitored by sensors associated with a smart facility. The second focuses on resource management in platforms executing data analytics applications while the third discusses its use in information dissemination and control in the operation of smart applications. The chapter includes case studies on sensor-based bridges, a cloud-based collaboration platform, popular cloud-based platforms for performing data analytics and two sensor-based smart applications, a museum touring system and a restaurant management system.

**Keywords** Smart facilities · Sensor-based bridges · Sensor-based machinery · MapReduce platform · Storm platform · NFC enabled mobile devices · Resource management · Service level agreement (SLA) · Constraint programming

## 1 Introduction

With a continued advancement in the state of the art in Information and Communications Technology (ICT) the vision of a smart city that integrates and manages its various assets in a secure and efficient manner is becoming a reality. Effective utilization of these advancements is leading to the emergence of smart cities in different parts of the world. Cloud computing and Internet of things (IoT) are among the forefront of technologies that are driving this realization of the smart city vision.

---

S. Majumdar (✉)

Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada

e-mail: [majumdar@sce.carleton.ca](mailto:majumdar@sce.carleton.ca)

© Springer Nature Switzerland AG 2018

M. Maheswaran, E. Badidi (eds.), *Handbook of Smart Cities*,

[https://doi.org/10.1007/978-3-319-97271-8\\_3](https://doi.org/10.1007/978-3-319-97271-8_3)



A smart city is built of smart systems (components) that include smart infrastructure, smart transportation, smart payment, smart health care, smart machinery and smart entertainment to name a few. Clouds and sensor-based internet enabled devices play a crucial role in the operation and management of these components. Cloud computing is a very popular and well used paradigm whereas the number of internet-enabled devices supported by the Internet of Things (IoT) technology is growing continuously and is expected to reach tens of billions by 2020 [37]. The use of sensor-based applications and facilities that include cyber physical assets of a city has been increasing and is expected to become more prevalent as the cost of sensors and associated technologies for the monitoring and management of the facilities decrease as we move ahead in the emerging world of IoT.

Various systems that garner the advancements in Information and Communication Technology (ICT) and sensor technologies for improving the operation and management of the different assets of a city and improve the quality of life of its inhabitants are important components of a smart city. Smart infrastructures and machineries are examples of some of such physical assets. Additionally, ICT based smart systems that contribute to the entertainment and cultural experience of the public and lead to the reduction in cost of operation and maintenance of the respective facilities are important contributors to the smart city experience. Smart museums discussed in the literature and restaurants that automate various operations such as food ordering and table booking are examples of such facilities. This chapter reports on state of the art research that focuses on the use of cloud computing and sensor-based devices supported by the IoT technology in the management of *smart facilities* such as sensor-based bridges and smart industrial/aerospace machinery and in *smart entertainment* systems that include a sensor-based museum touring system and a smart restaurant management system.

Monitoring and intelligent management of the smart infrastructure and machinery can significantly reduce the maintenance cost as well as prevent failures leading to accidents that may result from the inability to detect faults in a timely manner. The smart entertainment systems reduce operational cost, improve system efficacy and enhance the quality of experience for the users. Cloud computing and sensor-based devices play multiple important roles in the operation and management of such smart systems that constitute a smart city. These include the following:

- Role 1: Unifying multiple resources and making them available on demand.
- Role 2: Providing a platform and effective management of platform resources for performing analytics on data related to the facilities.
- Role 3: Facilitating information dissemination and control for smart applications.

It is important to note the relationship between the first two roles. Role 1 and Role 2 are important in the context of monitoring and management of smart facilities. Unification of resources (role 1) such as data analysis tools and servers scattered across the country for example are crucial for monitoring the health of sensor-based bridges and industrial machinery. The large volume of stored data for an infrastructure or the streaming data from sensors used in monitoring a power generator or aero-space machinery requiring real time analysis often need support

from platforms for big data analytics. Facilitating the hosting of such platforms on a cloud and the management of resources for such platforms (role 2) is thus important in the context of smart facility management. Discussion of these three roles forms a major component of this chapter. Because of their diversity, each role is discussed in detail in a separate section. Examples and use cases for a specific role are presented in subsections within the respective section.

The rest of the chapter is organized as follows. Background information on cloud computing is presented in the next section. Roles 1, 2 and 3 are discussed in Sects. 3, 4 and 5 respectively. Section 6 concludes the chapter and presents a summary of the highlights of the discussion presented.

## 2 Cloud Computing

Cloud computing that is based on resources acquired on demand is popular among service providers and consumers as well as researchers and system builders. The rise in interest in cloud computing is also reflected in a significant and continuous annual growth in the virtualization market. A number of financial institutions and market research organizations have predicted a multi-billion dollar market for the cloud computing industry [11]. As pointed out in [16], the annual world-wide enterprise Information Technology (IT) spending on cloud computing is expected to increase significantly with time. The primary reasons for the popularity of this distributed system infrastructure include the following.

- *Low IT investment:* With clouds the service consumer does not need to purchase resources. The “pay as you go” feature of the cloud allows users to acquire resources on demand for an hourly resource rental fee for example. This is of great value for both small start up companies as well as larger enterprises.
- *Elasticity:* A cloud lets the computing demand of a service consumer grow and shrink dynamically in accordance with its current workload. This provides a significant benefit for handling temporary increases in resource usage for the service consumer.
- *Green Computing:* By consolidating the IT operations of multiple consumers at a single data centre, an effective resource sharing is achieved leading to a reduction in power consumed by the computing, storage and cooling equipments.

Various types of clouds are in operation today. These include *public clouds* such as the Amazon Elastic Compute Cloud (EC2) [2] and Microsoft’s Windows Azure [31] comprising shared resources that can be used by the public at large. A *private cloud* on the other hand is accessible to the members of a given group only. An *enterprise cloud* that serves the employees of a given company and a *research and engineering cloud* that unifies resources located in multiple institutions are two variants of the private cloud. Irrespective of the type of a cloud effective resource management is crucial for harnessing the power of the underlying distributed hardware and achieving a high system performance. A survey presented in [17] on

cloud computing shows that security and performance are the two top priorities for cloud service consumers. As in the case of grids [15], a predecessor of cloud, quality of service (QoS) remains an important issue. Service level agreement, an important characteristic of clouds [7] often requires the handling of an Advance Reservation (AR) request that is characterized by a deadline for completion.

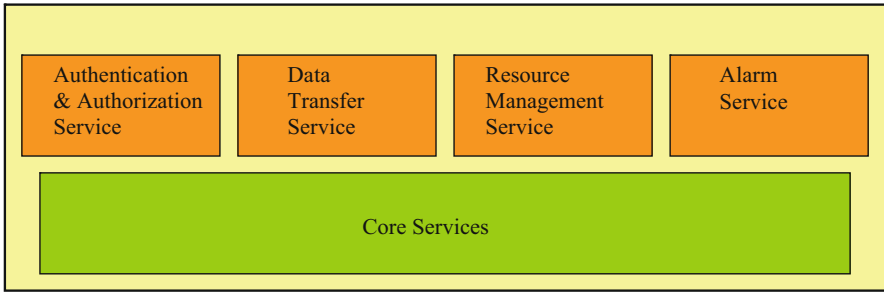
Clouds and clouds integrated with sensors or sensor-based systems are important components of smart systems that range from sensor-based smart infrastructures to smart health care. The availability of a mature cloud computing technology and the availability of sensors at a reasonable price are causing a great deal of interest in building smart systems that are important components of a smart city.

This chapter focuses on the use of clouds and sensors in smart city components. The different roles of clouds and the leveraging of sensors and IoT technology in various types of smart systems are discussed in the following subsections.

### 3 Unification of Resources

The resources required for managing smart facilities are often distributed in different geographic locations. Such resources include computing and storage resources, machines generating data and wireless sensor networks monitoring different parts of a smart facility such as a bridge or a smart building for example. A heterogeneous cloud serves as the glue and unifies the diverse set of resources to make them available on demand. Such a cloud often deploys a middleware comprising of multiple services for aiding the access, allocation and reservation of resources by a user (see Fig. 1). The roles of a number of such important services provided by this middleware are described next. These services are typically accessed through predefined Application Programming Interfaces (APIs).

- *Authentication and Authorization Service*: is responsible for ensuring that only actual users who are authorized to use the system have access to the system. It also allows a given authenticated user to only access a designated set of resources and perform a predefined set of operations on each of these resources.
- *Data Transfer Service*: is responsible for transferring data from one resource to another. Two types of data transfers may be performed. *Bulk data transfer* is used when a large volume of stored data, a file stored on a server for example, needs to be transferred to another server. Bulk data transfers are typically done on a best effort basis whereas a deadline may be associated with a *real time data transfer* that is required by some smart systems. Such a data transfer may require an advance reservation of all the resources that lie in the end-to-end data path between the data source and destination.
- *Resource management service*: concerns the allocation of resources to requests and scheduling of requests that are allocated to the same resource. A request is often associated with an earliest start time and deadline for completion [22]. When a request arrives the resource manager needs to select one or more



**Fig. 1** Middleware architecture

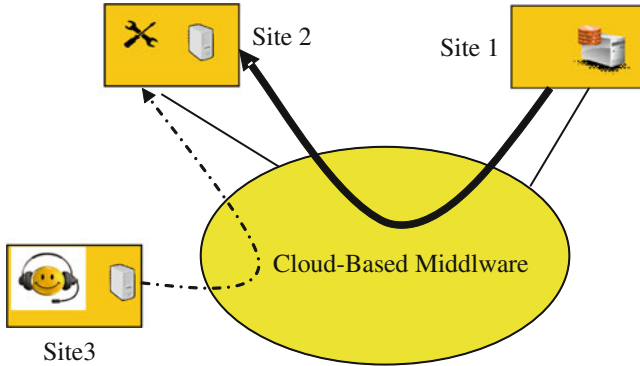
resources requested and determine the time at which the request will be scheduled to execute on each resource. Both compliance to user provided start time and deadline parameters and effective utilization of resources are important. Such resource management techniques are discussed in [7, 25] for example.

- *Alarm Service*: is used to make the system administrator aware of the occurrence of a predetermined system state that may indicate a system fault or an emergency situation.

The core services at the bottom layer in Fig. 1 provide support for core functionalities that include security and messaging to the services in the higher layer. A cloud-based solution for such unifying of resources in the context of sensor-based bridge management is described in [22, 28] and in the context of a collaboration platform for smart facilities management in [30]. A short discussion of each of them is provided in the following subsections.

### 3.1 *Cloud-Based Middleware for Sensor Equipped Bridges*

Bridges are important infrastructures providing means of transportation within a city as well as between adjacent cities. Maintaining the structural health of bridges is crucial for public safety. There are approximately 55,000 bridges in Canada and many of these bridges are reaching the end of their design service life [22]. There are many more bridges in the US. Multi-billion dollars are spent in both of these countries for their maintenance and repair. In spite of the efforts in performing routine maintenance there are still occasional failures of bridges. Such a failure results in millions of dollars in repair and have also led to loss of human lives. One of the problems associated with the current bridge maintenance practice is the dependence on human involvement in the semi-automatic process of maintenance and repair. Incorporating sensors for monitoring of structural health in the newly designed bridges and embedding sensors in existing bridges can mitigate some of the problems that arise from dependence on visual inspection by humans



**Fig. 2** Example of using the cloud-based middleware for tool invocation

[22]. Sensor data repositories, archival databases containing the history of bridge maintenance, data analysis tools and computing resources required for running these tools are often scattered across the country. A cloud-based middleware that serves as a glue and unifies these resources and makes them available on demand to bridge operators, researchers and engineers is described in [22, 28]. Using this middleware an authorized user of the system can acquire a tool and data available at two different locations and use the tool to process the data (see Fig. 2). The middleware provides such a resource access seamlessly leading to a user experience that is similar to the situation in which both the tool and data are co-located. Additionally, the middleware simplifies data analysis by making tools and data sets available on demand. This is discussed in the following paragraph.

Consider for example the Signal Processing Platform for Analysis of Structural Health (SPPLASH) tool [22] that was developed for the assessment of the structural condition of the 12.9 KM long Confederation Bridge, the longest bridge in Canada. The bridge data may be stored in Site 1 whereas the tool may be available at Site 2 and the user (bridge professional) planning to use the tool may be at Site 3 (see Fig. 2). The user starts by authenticating herself/himself. After a successful authentication a bulk data transfer service needs to be initiated for transferring the data from Site 1 to Site 2. The user then invokes the reservation service that reserves the tool at Site 2 for the time period requested by the user. The bulk data transfer service is then activated to transfer the data from Site 1 to Site 2. The user then invokes the tool through its Graphical User Interface and performs the data analysis needed. At the end of the reservation period the tool is released and can be used by another user. The mobilization of the tool, data and applications on demand greatly reduces the time required to obtain the results of the data analysis [22].

In addition to the analysis of sensor data by a tool such as SPPLASH, archived data on a bridge that includes its entire maintenance history over a long period of time may need to be analyzed for making decisions regarding its next scheduled maintenance. Big data platforms hosted on clouds that employ parallel processing

techniques may be useful in performing the analysis of such very large volumes of data in a timely manner. Such data analytics platforms are useful in the context of bridges as well as other assets of a smart city and are discussed in more detail in Sect. 4.

*Management of multiple bridges* The cloud-based middleware discussed earlier can be used to manage multiple bridges in the city. This will enable the sharing of common data analysis tools and resources in the monitoring and management of multiple bridges. Such an approach can thus reduce the cost of bridge management as well as provide uniformity in the process used for management of all the bridges in the city.

### 3.2 *Cloud-Based Platform for Research Collaboration*

This section focuses on a cloud platform that facilitates research collaboration. Similar to the cloud-based middleware for bridge management discussed in the previous section *Research Platform for Smart Facilities Management (RP-SMARF)* unifies heterogeneous resources available at different locations and makes them available on demand to a requesting member of the collaboration team [30]. A distinguishing feature of RP-SMARF is that it supports *multi-tenancy*. Multiple collaboration teams can use the same cloud-based platform. A user that belongs to team A can use the resources for its team without interfering with the operations of another collaboration team. The resources of a given collaboration team are typically visible to members of the respective collaboration team and are not accessible to other collaboration teams. Currently, RP-SMARF supports two research collaboration teams. The first comprises primarily of Civil Engineers researching structural health assessment of bridges whereas the second comprises of Mechanical Engineers and Aerospace machinery researchers that focus on the monitoring and maintenance of sensor-based aerospace machinery.

A common and popular operation performed on RP-SMARF is the use of specific data analysis tools. The RP-SMARF platform provides a Graphical User Interface (GUI) that can be used for setting up a tool. The GUI walks a user through the steps necessary to run the tool: tool selection, parameter selection, viewing the status of the tool after it starts running and collecting the tool output. These common steps are shown for the example SPPLASH tool in Fig. 3. The screen displayed in the figure is for showing the results of vibration analysis for a bridge performed by SPPLASH for a given input file storing pre-recorded sensor data for the bridge. A tool can use any dataset contained in file/folders that are accessible through the platform. This simplifies and speeds up the setting up and running of a research tool by a collaborator although the tool may be newly created by another researcher.

Built with the help of a grant from Canadian Network for the Advancement of Research, Industry and Education (CANARIE), RP-SMARF is one of the first platforms for collaboration among researchers of smart facility management [29].

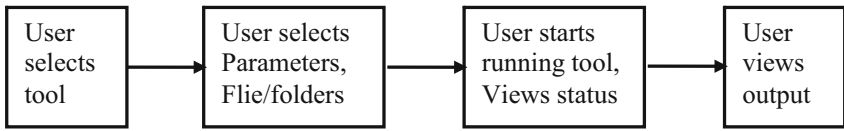
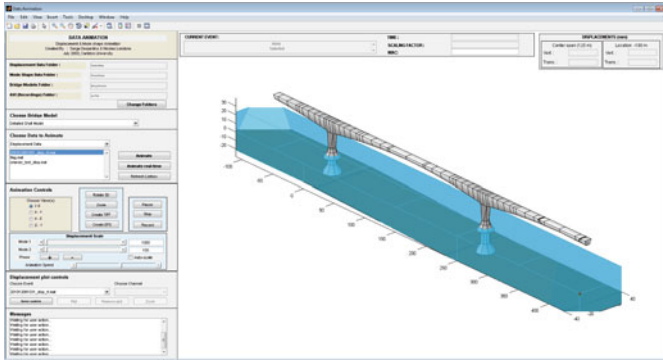


Fig. 3 Example of tool usage

The key features of RP-SMARF are summarized next. Further discussion of its components and operation are available from [30].

*Multi-Tenancy* The platform supports multiple user communities and the members and resources of each community are typically isolated from those of the other communities.

*Unification of Heterogeneous Resources* Unlike a data centre cloud that typically provides computing and storage resources, RP-SMARF unifies geographically dispersed diverse resources ranging from computing and storage nodes, data processing tools as well as sensor and archival data sets and make them available on demand to an authenticated user.

*Automatic Data Movement* In order to support tools that use local data files RP-SMARF implicitly moves data from a remote location to the local site for the tool when required.

*Support for Batch and Interactive Modes* The platform enables the use of a tool in a batch mode where the data file is processed in the background and the user is notified when the processing is complete. It also supports an interactive mode in which a user can interact with the tool in real time.

*Tools and Dataset discovery* Resources and users are grouped into communities. When a user wants to gain visibility of tools and data sets in another community her/his request needs to be granted by the owner of the respective dataset/tool in the other community.

*Processing Real-Time Data Streams* RP-SMARF supports the processing of data in real time. This enables the streaming of data resulting from an experiment to be fed into a tool that can process it and provide the results of the processing to the user.

## 4 Data Analytics Platforms and Resource Management

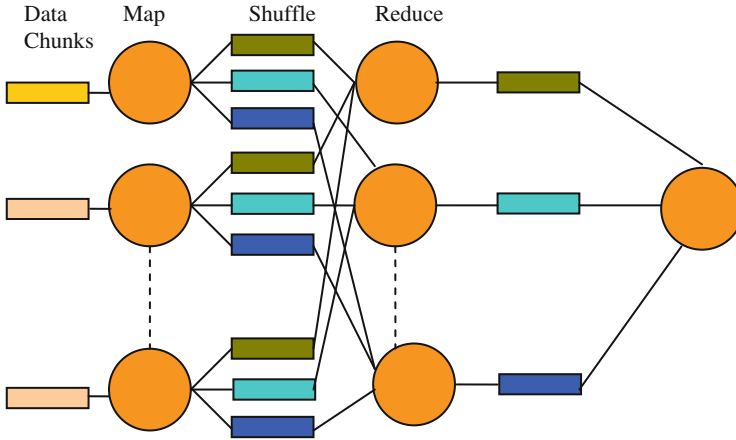
Monitoring of smart infrastructures such as bridges was discussed earlier. Analyses of both stored as well as streaming data are important in the context of smart systems. Examples of stored data on a bridge that includes both the sensor data stored over a period of time as well as archival data on the maintenance history of the infrastructure are often very large in volume requiring special parallel processing platforms for performing their analysis in a timely manner. Sensor data collected continuously on a power generator for example often needs special real time techniques for predicting impending failures and performing proactive maintenance of the system to avoid such failures. This section focuses on data analytics platforms that are used for performing analyses of such data collected on the respective smart systems and the management of resources in these platforms. Platforms and resource management techniques for analyzing stored data are discussed first. This is followed by a discussion of the streaming data analytics platforms and their management.

Analysis of large volumes of stored data (both sensor data as well as archived data on management histories of the respective facilities) from smart systems will benefit from parallel processing platforms using multiple computing resources provided by a cloud for performing the analysis in a timely manner. Effective resource management (resource allocation and task scheduling) techniques are required for harnessing the power of the large resource pool in the cloud. Examples include resource management techniques for MapReduce/Hadoop applications with deadlines for completion. A short introduction to MapReduce is presented first. This is followed by a discussion of resource management techniques that are required to execute the MapReduce applications such that they meet the user specified deadlines.

### 4.1 *MapReduce/Hadoop*

MapReduce is a parallel processing framework proposed by Google [40] for processing large volumes of data. The overall approach is to divide the large data file into chunks that are processed in parallel. MapReduce is characterized by three phases of operations: Map, Shuffle and Reduce (see Fig. 4). The outputs of the concurrently running map tasks, each of which typically runs on its own processing node, referred to as key value pairs in the MapReduce literature are finally combined





**Fig. 4** The map reduce framework

in the reduce phase by a set of parallel reduce tasks each of which can run on its own processing node. In the shuffle phase that follows the map phase data is redistributed in such a way that all data belonging to a specific key is located on the same processing node, thus improving system performance. Apache Hadoop is a popular open source implementation of the MapReduce framework that also incorporates a file system referred to as the Hadoop File System (HDFS). Hadoop employs a Master/slave architecture. A typical Hadoop cluster that often runs on a cloud contains a single master node and multiple slave nodes. The master node maintains HDFS and assigns the different map and reduce tasks to slave nodes for execution [40]. The operations performed by the map and reduce tasks depend on the application using the MapReduce framework.

Although Hadoop provides a platform for the parallel execution of a data analytics applications effective resource management is crucial for meeting the quality of service (QoS) requirements of clients. Often the QoS requirements are captured in a service level agreement (SLA) between the client and Hadoop service provider. Such an SLA specified by the user submitting the job may include the earliest start time for the MapReduce job, estimated task execution times and a completion time deadline. Accepting the job binds the service provider to comply with the SLA requirements specified by the client.

A discussion of resource management in MapReduce platforms is presented next. Processor allocation and scheduling are known to be computationally hard even for sequential jobs characterized by a single thread of execution. Running multi-phase parallel jobs such as MapReduce makes the problem even harder. Resource management for MapReduce jobs that are executed on a best effort basis has been well researched. Associating an SLA that includes a deadline for completion with MapReduce jobs has started receiving attention more recently. Such a deadline is

important when a real time or near real time response is required in live business intelligence and real time analysis of event logs [38]. Associating an SLA with a job makes the problem challenging and is discussed in the following subsections. A discussion of resource management in a closed system processing a given batch of MapReduce jobs is presented first. This is followed by a description of resource management techniques for open systems subjected to job arrivals.

#### 4.1.1 Closed Systems

Dong et al. [12] propose an algorithm for scheduling of workloads comprising two types of jobs: MapReduce jobs with deadlines (real-time jobs) and jobs with no deadlines (non-real-time jobs). They incorporate the Tasks Forward Scheduling (TFS) and the Approximately Uniform Minimum Degree of Parallelism (AUMD) techniques into the Hadoop scheduler that performs two-level scheduling for scheduling both real-time and non-real-time jobs. An abstraction of the MapReduce matchmaking and scheduling problem is by formulated as an optimization problem in [9]. Linear programming is used for determining a schedule that minimizes the overall completion time of the jobs in the batch. A job execution cost model that considers parameters such as the execution times of map and reduce tasks is devised for scheduling MapReduce jobs with deadlines in the work presented in [19]. Using a Schedulability test on the model the algorithm determines whether or not the deadline for the given job can be met with the available resources in the cluster that is being shared with a given number of other jobs. If the test fails the user has the option of resubmitting the job with a new deadline requirement.

Various optimization techniques have been used in resource management for batches of MapReduce jobs. A comparison between two optimization techniques in the context of resource management for MapReduce jobs with deadlines are considered by Lim et al. in [25] that presents techniques for resource allocation and scheduling on systems subjected to a batch of MapReduce jobs with deadlines. Two different approaches, one based on Mixed Integer Linear Programming (MILP) and the other on Constraint programming (CP) are presented. The MILP based resource management algorithm is implemented in LINGO [26] whereas the CP-based algorithm has two versions one using Gecode [13] and the other IBM CPLEX [18]. Through a simulation-based analysis the authors demonstrate the superiority of the CP-based technique that uses IBM CPLEX over the MILP-based technique especially for larger workloads comprising 10s of MapReduce jobs with each job comprising up to 100 map and reduce tasks [25]. The simulation results indicate the superiority of the CP-based technique both in terms of batch completion time as well as the resource management overheads for such large workloads.

Batch systems comprise a fixed number of jobs whereas an open system is subjected to a stream of job arrivals. Resource management for open systems that are characterized by continuous job arrivals is often needed for use in a data centre environment and is the subject of attention for the next section.

### 4.1.2 Open Systems

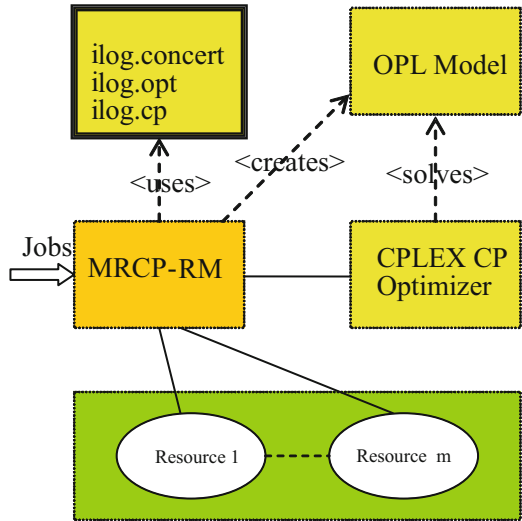
Resource management in open systems subjected to MapReduce jobs each of which is characterized by an SLA has been receiving attention from researchers. Examples include [23, 38]. The authors of [38] have proposed two resource management algorithms based on earliest deadline first (EDF). The Minimum Resource Quota Earliest Deadline First (MinEDF), allocates the minimum number of resources for completing a job before its deadline. The second technique called Minimum Resource Quota Earliest Deadline First with Work-Conserving Scheduling (MinEDF-WC) improves upon MinEDF by incorporating the ability to dynamically allocate/deallocate resources from jobs. This allows spare resources that were previously allocated to a job that does not need them any longer to be effectively utilized by another job.

An algorithm for matchmaking and scheduling on an open system subjected to arrivals of MapReduce jobs with SLAs is described in [23]. The SLA for a job comprises an earliest start time for the jobs, user estimates of task execution times and a deadline for job completion. The resource management performed by the MapReduce Constraint Programming Based Resource Manager (MRCP-RM) is based on a Constraint Programming approach. A set of constraints used to describe the relationships among the various system and workload parameters, as imposed by the precedence relationships among the map and reduce tasks and the SLA parameters is programmed in IBM's Optimization Programming Language (OPL) and solved by the IBM CPLEX system [18]. The objective function is the minimization of the number of late jobs (or jobs that miss their deadlines). Jobs upon arrival at the system are enqueued (see Fig. 5 that is based on [24]). If the resource manager MRCP-RM is not busy it is invoked for processing the jobs in the system. MRCP-RM performs matchmaking and scheduling of the newly arrived jobs as well as tasks that have already arrived on the system but have not started execution. MRCP-RM uses IBM CPLEX which generates a model by using OPL that is an implementation of the CP model comprising the constraints corresponding to the tasks that have not started their execution. The CP Optimizer solving engine in IBM CPLEX is then used to solve this OPL model and the allocation of tasks to resources as well their start time for execution are determined.

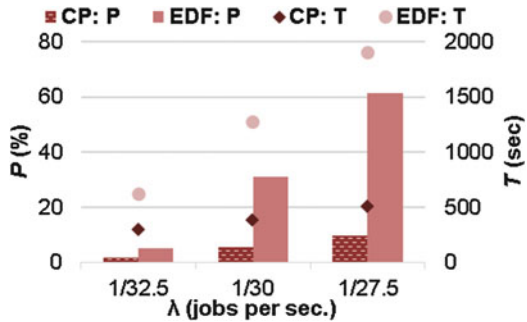
A Hadoop Constraint Programming based Resource Manager (HCP-RM), an implementation of MRCP-RM on a Hadoop cluster is reported in [23]. A rigorous performance analysis based on prototyping and measurement is reported. A sample set of results is presented in Fig. 6. The figure presents the measurement data from an experiment made on a Hadoop cluster subjected to a stream of jobs arriving on the system.

Two performance metrics, the proportion of jobs that miss their deadlines (P) and the average turn around time of jobs (T) are captured in Fig. 6. The performance of HCP-RM is compared to that of an Earliest Deadline First (EDF) algorithm that associates a higher priority with jobs that have shorter deadlines. A detailed description of EDF and the details of the system and workload parameters used in

**Fig. 5** The MRCP-RM system



**Fig. 6** Performance Analysis of HCP-RM (From ref [23])



the experiment are provided in [23]. As shown in the figure the CP-based algorithm (HCP-RM) outperforms EDF by a large margin for both the performance metrics.

### 4.1.3 Energy Aware Resource Management

Although MapReduce frameworks shorten the completion time for data analytics applications they can give rise to significant energy consumption through their execution on multiple machines. Energy consumption in data centers in which these machines run is an important concern: 1.1% and 1.5% of the total worldwide annual energy consumption in 2010 can be attributed to data centres [21]. As a result energy aware resource management techniques in general (see [10] for example) and for platforms supporting MapReduce applications in particular have started receiving attention. A novel resource management approach that combines allocation and scheduling algorithms for meeting client SLAs with dynamic voltage

and frequency scaling (DVFS) is reported in [14]. It uses a constraint programming based technique for allocation and scheduling. The algorithm adjusts the CPU frequency for minimizing energy usage while ensuring the SLA requirements of the MapReduce jobs are met. The algorithm is observed to give rise to substantial energy savings in comparison to a non-energy aware technique that focuses only on meeting SLAs. A detailed discussion of the technique and a survey of other related techniques are presented in that paper.

## 4.2 Streaming Data Analytics

Data streaming is an important phenomenon the popularity of which is growing continuously. Further growths in interest are expected with an increase in the use of the Internet of Things (IoT) technology. Examples of data streams that need analysis include computer network traffic, sequence of ATM transactions, financial data e.g. stock pricing information, web server logs, data centre logs, and sensor data streaming from IoT devices. Example of sensor data streams include data generated by smart facilities such as sensor equipped bridges, smart industrial/aero-space machinery and buildings as well as bio-medical equipment (e.g. Electro Cardiogram (ECG) machines taking periodic ECGs on patients being monitored in an Intensive Care Unit). Streaming data needs to be analyzed both in real time as well stored for a more detailed analysis at a later point in time. Clouds that can provide the desired number of resources on demand can be effectively utilized for the storage and processing of streaming data. Processing of stored data was discussed in the preceding section. This section focuses on platforms for streaming data analytics.

Both Apache *spark* and more recently Apache *storm* are popular platforms for performing streaming data analytics. This section focuses on Apache *storm* that is described next.

*Storm* is a well-known platform for performing streaming data analytics [4]. Such a platform that is often referred to as a storm cluster (see Fig. 7) comprises a *nimbus* node and multiple *supervisor* nodes [8]. The *nimbus* node runs on an independent machine. Users submit jobs referred to as *topologies* to the *nimbus* node that distributes the tasks in the topology to the worker processes in the cluster. There are multiple supervisor nodes in the cluster with each supervisor node hosting multiple worker processes. In addition to the *nimbus* and the supervisor nodes a storm cluster uses an Apache *zookeeper* [5] for synchronization and for managing the state of the storm cluster.

A storm topology comprises *spouts* and *bolts*. Spouts connect the topology to external streaming data sources that may include individual IoT devices as well as message brokers such as Apache *Kafka* [3]. A spout provides the application logic for connecting to the message source and forwarding the stream of *tuples* to other processing elements. Application code for stream processing that depends on the type of data analytics performed is contained in a bolt. A running instance of a bolt or spout is referred to as a task.

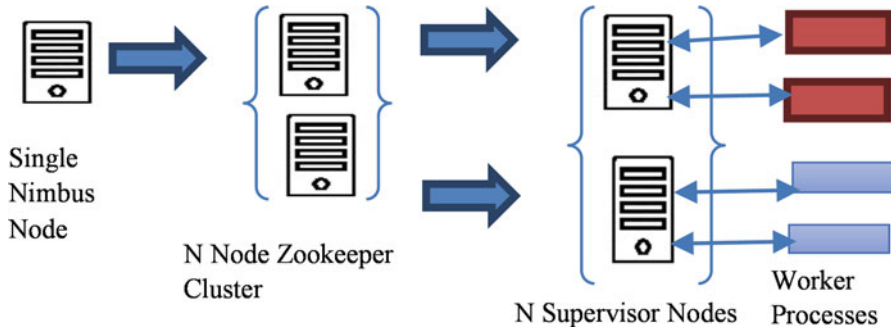


Fig. 7 Example of a storm cluster (from Ref [8])

Allocation of nodes to multiple topologies running in a *multi-tenant* storm cluster is an important function of a resource management system. The default resource manager in storm is called the *isolation scheduler* that allocates resources to the multiple topologies in such a way that no resource is shared between any two given topologies. Isolation scheduler, however, can give rise to starvation of some of the topologies in a resource constrained environment [8]. This occurs because this scheduler allocates the resources requested to the respective topologies in the order in which they get submitted. As result in a resource constrained environment in which there is a dearth of resources topologies that are submitted later may not have enough resources to run their essential components. Moreover, isolation scheduler does not support priorities for topologies. Research presented in [8] focuses on addressing these shortcomings of the isolation scheduler through two priority-based schedulers. A priority based scheduler allocates a higher proportion of resources to a higher priority topology in comparison to a low priority topology. In the first, called the static priority scheduler (SPS) every topology is given a fixed priority that remains unchanged throughout the lifetime of the topology. In the second, called the dynamic priority scheduler (DPS) the priority of a topology can change dynamically during execution. The authors show that both of these schedulers can handle the resource constrained environments that can not be handled effectively by the isolation scheduler. Performance of DPS and SPS are analyzed by the authors for a synthetic workload [8]. A comparison the performance of SPS and DPS is performed on an Amazon EC2 cloud with 26 c4.large type EC2 nodes [2] in the storm cluster. One of the nodes is used for running nimbus, another for running a user interface, and 24 nodes are used for running the supervisor nodes where each supervisor node hosts 2 worker processes.

A use case considered in [8] is the detection of *complex events* that has been a subject of interest in stream processing systems [27]. A complex event corresponds to the occurrence of multiple raw events in a given sequence with each raw event corresponding to a phenomenon being monitored by a sensor in a smart system exceeding a given threshold for example. The complex event considered in [8] is the simultaneous occurrence of Raw Event1 and Raw Event2 each corresponding to

one of the two data streams coming into the system. Four storm topologies are used to process the batches of tuples coming into the system. Two of these topologies are used to infer complex Events while the other two are used to generate event logs for the two streams that are stored on the system. The raw events of interest for this use case are discussed next.

- Raw Event 1 corresponds to the situation in which a predefined proportion of the tuples in the first stream containing measurement readings for the first phenomenon in a batch has crossed a predefined threshold.
- Raw Event 2 corresponds to the situation in which a predefined proportion of the tuples in the second stream containing measurement readings for the second phenomenon in a batch has crossed a predefined threshold.

The topologies *EventTP1* and *EventTP2* are used to detect Raw Event1 and Raw Event 2 respectively. Simultaneous occurrence of Raw Event 1 and Raw Event 2 signifies an onset of a complex event. *EventTP1* and *EventTP2* need to assume a higher priority when the user specified trigger condition that corresponds to the first appearance of Raw Event 1 or Raw Event 2 occurs so that the respective topologies get enough resources to detect the complex event in a timely manner. The results of an experiment comparing SPS with DPS is presented in Fig. 8. In the experiment  $S_{high}$ , the mean execution time for the high priority topologies, *EventTP1* and *EventTP2* is varied. The execution time for processing a tuple by the two low priority topologies,  $S_{low}$ , is set to  $S_{high}/2$  [8]. The performance metric of interest is the complex event processing latency  $T_E$ . The details of the other experimental parameters are available in the paper. Fig. 8 that is based on [8] shows that DPS outperforms SPS and the performance difference between the two scheduling strategies increases with an increase in  $S_{high}$ . The superiority of DPS can be attributed to its ability to vary the topology priorities dynamically, after one of the raw events is detected on the system.

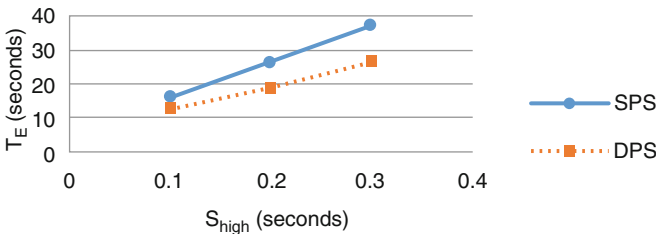


Fig. 8 Comparison of the performance of SPS and DPS

## 5 Information Dissemination and Control for Smart Applications

Sensor-based devices such as smart phones are being used in many systems that require human interaction. The sensors in such phones are used to sense various phenomena that can range from a person's health related data in a smart health application to the identification of an exhibit in a smart museum. A cloud is often used for providing a repository for storing data and disseminating it to the various components of the respective system and coordinating their operations. Research in the area including two case studies, one on a smart museum touring system and the other on a smart restaurant management system is discussed.

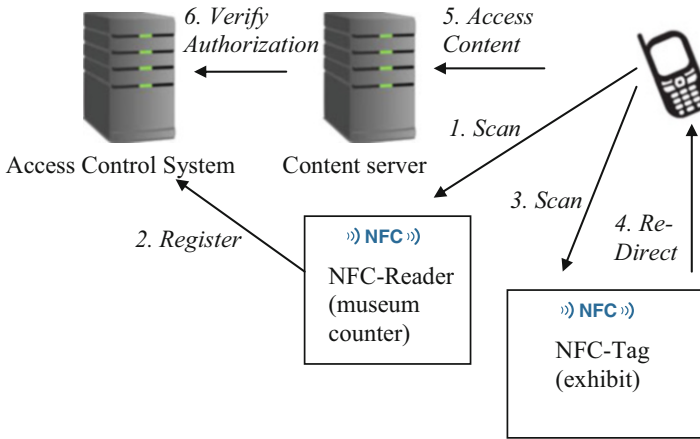
As discussed in Sect. 1 both of these systems contribute effectively to the entertainment and cultural experience of the users of a smart city. Both systems are based on sensors that use the Near Field Communication (NFC) technology. A brief introduction to NFC is provided. Near Field Communication is a set of standards developed by the NFC Forum [35] and is based on Radio Frequency Identification (RFID). NFC is a set of communication protocols that let two devices such as a smart phone and a smart tag or sticker to communicate by bringing them in close proximity (typically 4 cm or less) of each other. Applications such as mobile payment supported by companies such as Google, BlackBerry, Visa, Bell, Rogers, and Telus are fuelling the growth in the popularity of NFC devices.

### 5.1 *Museum Tour Guide System*

This section focuses on an NFC-based museum touring system [39]. The system effectively utilizes the computing and multimedia capabilities of NFC enabled mobile devices such as smart phones and tablets that can enhance the visitor's experience by accessing multimedia content on their mobile devices. The system described in the paper is a proposed replacement of the conventional "audio guides" used typically at museums. With this NFC-based system the number of clients using the system is thus not constrained by the fixed number of audio guides available at a given museum location. Moreover, replacing electro-mechanical device based information dissemination with web-based content delivery can also reduce the maintenance cost for such systems significantly. The key features of the system are summarized [39].

- The system can work with any Android based NFC sensor-equipped mobile device.
- The cost for maintaining the conventional audio guide systems incurred by the museum is eliminated.
- The absence of kiosks for picking up and returning audio guide systems decreases the cost incurred by the museum owners and leads to a higher customer





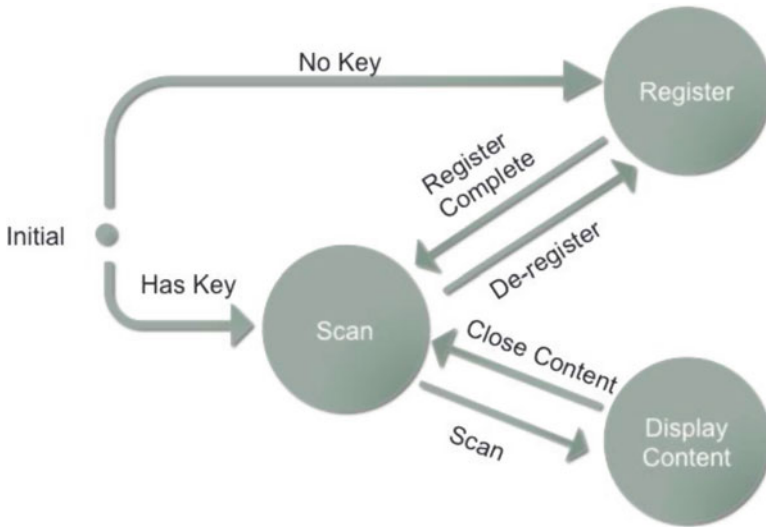
**Fig. 9** The NFC enabled museum touring system

convenience resulting from the elimination of queues. Moreover, the situation that a customer is unable to rent a system because all the guide systems in the museum are already rented is avoided.

- By using a webpage based information dissemination it is possible to deliver a richer multimedia content associated with an exhibit to the customer.

Museum exhibits in the proposed system are equipped with NFC tags that are placed near the respective exhibits. In order to use this system the visitor needs to download a free app provided by the museum. Upon arrival at a museum the visitor needs to pay the required fee associated with using the system and register her/his mobile device. Once registered, the visitor can tap the device on a given tag and receive multimedia content on the exhibit on the visitor's smart phone. A high level diagram of the system for explaining the sequence of steps required for enabling such an experience by a visitor is captured in Fig. 9 that is based on [39]. A short summary of the operations performed in each step 1–6 is presented next.

1. Visitor scans the NFC reader at the cash counter at the entry to the museum with her his smart phone.
2. The NFC reader registers the device with the access control system running on a server and provides a unique ID to the phone.
3. The visitor tours the museum and then scans an NFC tag at a desired exhibit.
4. The URL information encoded in the smart tag is transferred to the phone.
5. The phone contacts a content server with the URL pushed by the tag via the museum Wifi and sends the unique ID received during registration.
6. The content server verifies the authorization of the smart phone for accessing the content by sending the unique id received to an access control server. Once authorized the desired content is sent from the content server to the user.



**Fig. 10** State diagram for the android application. (From Ref. [39])

Note that the first 2 steps are performed only once whereas steps 3–6 are repeated for each exhibit visited by the user. The operations performed by the app running on the smart phone are explained with the help of Fig. 10 that displays a state diagram for the app.

Upon start the application is in the initial state in which it checks whether or not it has a key. It has a key if the device was registered with the museum signifying that the visitor has paid the requisite fee for using the museum touring system. If a key is found it checks whether the key is still valid by inspecting the expiry date and time associated with the key. If there is no key or the key has expired the application moves to the register state where it waits for the smart phone to be tapped on the NFC reader at the museum kiosk. Once the tapping is complete and a key is acquired it makes transition to the scan state where it waits for the phone to be tapped on a smart tag placed near an exhibit. Once tapped the application receives the URL containing information about the exhibit and moves to the display content state in which it presents the multimedia content stored at the URL to the visitor. After the visitor closes the content screen it moves back to the scan state where it waits to be tapped on another smart tag. From the scan state the app can also move to the register state if the user wants to deregister the device or the time duration for which the key is valid expires. Once in the register state the application waits for the next registration event.

*Implementation Technology* A summary of the technologies used in implementing the system is presented next. The implementation reported in [39] describes an Android based application written in Java running on the mobile device. The registration program (see Fig. 9) that runs on a workstation in the museum is a

Java based application. It is used by the person at the museum kiosk to register the mobile devices. The content server shown in Fig. 9 is achieved as an Apache web server serving PHP web pages whereas a MySQL [33] database server is deployed for performing the operations of the access control server. The mobile device used was an Asus Nexus 7 [6] tablet that was chosen because it was NFC enabled and had the capability of running the latest version of the Android at the time of system implementation. Further discussion of system implementation is beyond the scope of this chapter. The interested reader is referred to [39] that presents a more detailed discussion of the implementation of the museum touring system.

*Handling multiple museums* By deploying the content server and the access control server in a cloud a single entity can handle multiple museums that may be located within the same city or in multiple cities. This opens up a business case for a service provider who can provide a museum touring service for multiple museums.

## 5.2 Restaurant Management System

Using wireless devices to order in restaurants has started receiving attention. Examples include Wireless Ordering System (WOS) [20], a hand-held ordering tablet [32] and rich media menu displays [34]. All these systems focus on the food ordering and displaying components of a restaurant visit. A discussion of a smart restaurant management system (SRMS) [36] that concerns the various operations in the end-to-end restaurant visit experience is presented. The system leverages NFC tags, cloud and smart phone technologies and is motivated by the improvement of the quality of experience for the customer and lowering of the cost for the restaurant owner as well as reduction of the work performed by the restaurant employees. A short description of SRMS is provided next.

A high level overview of SRMS is captured in Fig. 11. The system is characterized by four components: the Android application running on the customer's mobile device, a web application running on the restaurant server, a parking subsystem as well as a web server and a MySQL database [33] running on a public cloud such as the Amazon EC2 cloud. The hosting of the web server and the database on a cloud is a design choice made by the authors who state that such a system partitioning facilitates the integration of various system components. The cloud-based sub system maintains communication among the restaurant server, the parking sub-system and the user mobile device. By decoupling the restaurant server from the parking sub system and the user's android device can reduce the security threats for the restaurant server and allows the possibility of using a parking lot maintained by a separate service provider that is distinct from the restaurant operator. The figure shows the four options available to the customer: Find Parking, Static menu, Interactive Menu and Checkout & pay. Selecting any of these options displays the different operations corresponding to the selected option that can be invoked by the customer. The web application is responsible for performing the operations invoked by the restaurant staff through the respective webpage (see Fig. 11). The

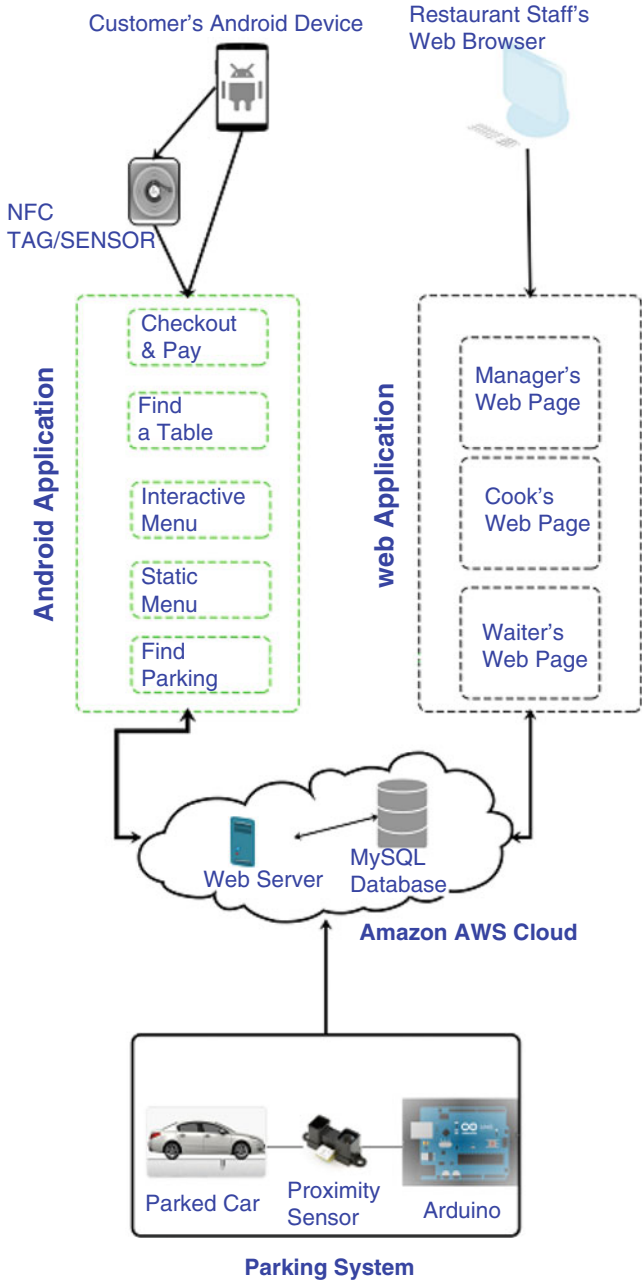


Fig. 11 The SRMS system. (From Ref. [36])

Manager's, Cook's and the Waiter's web pages contain information that is relevant to the specific restaurant staff. The web server and the database are used for communication with the customer's Android app and the restaurant server and for storing the information provided by the customers, the restaurant staff and the parking component. The parking subsystem interfaces proximity sensors each monitoring a specific parking space and reporting its status (free/busy) to an Arduino Uno R3 microcontroller [1]. This controller communicates with the cloud and stores the status (free or busy) of each parking spot maintained in the "Parking Spots" table in the database. The operations performed by the system are categorized into three phases a short discussion for which is provided.

*Phase 1* The first phase focuses on helping the customer find a parking spot in the restaurant garage. By tapping the smart phone on the NFC tag placed near the garage entrance the Android app running on the mobile device gets connected to the MySQL database located on the cloud. The database sends information regarding free parking spots that get displayed on the mobile device. The parking spots are equipped with proximity sensors that are used to maintain the status of each spot. Finding an appropriate table inside the restaurant for the customer is also performed in this phase. On tapping the smart device on an NFC tag placed at the restaurant entrance the information on available tables is displayed and the customer can choose one that suits her/his needs. If all the tables are occupied the customer has the option to leave a request for the desired table size and her/his phone number that is used to call the customer when an appropriate table becomes available.

*Phase 2* This phase concerns providing the customer with an interactive menu, receiving the order and dispatching it to the restaurant kitchen. The interactive menu is reached by a customer by tapping her/his smart phone or tablet on an NFC tag on the table. Customers can then choose dishes they want to order and then view the dishes ordered on a "summary of the order page". She/he can move back to the ordering screen if a change needs to be made. On touching the "send order" button on the GUI displayed on the touch screen of the device the order gets dispatched and becomes visible to the kitchen staff.

*Phase 3* The payment of the bill by the customer is handled in Phase 3. Through the GUI displayed on her/his mobile device the customer can get the final bill, make a credit card payment and get a receipt. SRMS provides the customer with various additional features including the splitting of the bill among multiple payees. A detailed description is provided in [36].

*Restaurant Management Service* The cloud-based deployment of the data base and the web servers opens up the possibility of using the common resources for multiple restaurants. This presents an opportunity for a restaurant management service. A service providing company can provide a smart restaurant management service to multiple restaurants thus contributing to the economic development of the smart community.

## 6 Summary and Conclusions

A smart city comprises smart components that need to be managed in an efficient, secure and reliable manner. This chapter has focused on the use of sensors and clouds in the operation and management of such smart components. Sensors are often embedded in the smart facility such as a bridge or machine for monitoring its state of operation and or health. The monitored data is then analyzed to make decisions regarding the set of possible actions regarding the respective facility. A cloud can be used for unifying the various geographically dispersed resources that are required for the monitoring and management of a given facility. Two examples in which the cloud is used to unify of resources including tools, data sets and computing resources and making them available on demand to a user were discussed. The first focuses on the monitoring of the structural health of sensor-based bridges and the second on a multi-tenant platform that enabled resource sharing among the members of a collaboration team. The multi-tenancy supported by the platform enabled multiple research teams each with its unique set of users and resources to use the same cloud-based platform concurrently. The systems discussed are based on a single cloud. Utilizing a multi-tiered architecture in which data processing is distributed among various levels from edges close to the facilities to a backend cloud warrants further investigation.

Analysis of data on a smart facility is crucial for its maintenance and management. A MapReduce platform that uses parallel processing for analyzing stored data and resource management algorithms for the platform that enabled the analysis of data in a timely manner were discussed. Resource management for platforms executing applications characterized by a higher number of task levels in comparison to MapReduce need investigation. Such resource management algorithms should also be robust for handling error associated with the estimates of task execution times provided as part of the SLA by the user. Platforms such as storm for handling streaming data and scheduling algorithms for storm were discussed. Resource management in clouds that support both types of platforms one running analytics on stored data and the other on streaming data forms an interesting direction for future research.

Smart applications such as a smart museum touring system and a restaurant management system were described in the chapter. Both cloud and NFC technology can be effectively utilized in the construction of such systems. A general framework that includes a number of reusable components performing common operations such as sensing of NFC tags and interfacing with various classes of mobile devices such as Android phones and tablets is worthy of investigation. Such a framework can speedup the development of various NFC based applications.

**Acknowledgments** This chapter is based on the results of a number of research projects. The author is grateful to the Natural Sciences and Engineering Research Council of Canada (NSERC), the Ontario Centre of Excellences (OCE), CANARIE, Huawei Canada, Solana Networks and Cistel for their support in the respective research projects. Thanks are also due to the students and research staff for their participation in the research.

## References

1. Arduino Uno, <http://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf>, Accessed: November 13, 2015.
2. Amazon, Amazon Elastic Cloud, <http://aws.amazon.com/ec2>, Accessed November 15, 2017.
3. Apache Software Foundation, Apache Kafka, June 2016. <https://kafka.apache.org/>, Accessed November 15, 2017.
4. Apache Software Foundation, Apache Storm, <http://Storm.apache.org/>, Accessed: November 15, 2017.
5. Apache Software Foundation, Apache Zookeeper, <https://zookeeper.apache.org/>, Accessed November 15, 2017.
6. Asus, Nexus 7, [https://www.asus.com/us/Tablets/Nexus\\_7/](https://www.asus.com/us/Tablets/Nexus_7/), Accessed: November 14, 2017.
7. R. Buyya, S.K. Garg, and R.N. Calheiros, "SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions," in *Proceedings of the International Conference on Cloud and Service Computing (CSC 2011)*, Hong Kong, China, December 2011, pp.1-10.
8. R. Chakraborty, S. Majumdar, "Priority Based Resource Scheduling Techniques for a Resource Constrained Stream Processing System". in *Proceedings of the 4th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT 2017)*, Austin, USA, December 2017, pp. 21-31.
9. H. Chang, M. Kodialam, R.R. Kompella, T.V. Lakshman, M. Lee, and S. Mukherjee, "Scheduling in MapReduce-Like Systems for Fast Completion Time", in *Proceedings of the IEEE INFOCOM 2011*, Shanghai, China, 10-15 April 2011, pp. 3074-3082.
10. Y. J. Chiang, Y. C. Ouyang and C. H. Hsu, "An Efficient Green Control Algorithm in Cloud Computing for Cost Optimization," *IEEE Transactions on Cloud Computing*, Vol.: 3, No.:2, June 2015, pp.145-155.
11. R. Cohen, "Gartner Announces 2012 Magic Quadrant for Cloud Infrastructure as a Service", Forbes, October, 2012, <http://www.forbes.com/sites/reuvencohen/2012/10/22/gartner-announces-2012-magic-quadrant-for-cloud-infrastructure-as-a-service/>, Accessed: November 15, 2017.
12. X. Dong, Y. Wang, and H. Liao, "Scheduling Mixed Real-Time and Non-real-Time Applications in MapReduce Environment", in *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS 2011)*, December 2011, pp.9-16.
13. Gecode, Generic Constraint Development Environment., <http://www.gecode.org/>, Accessed: November 15, 2017
14. A. Gregory, S. Majumdar, "Resource Management for Deadline Constrained MapReduce Jobs for Minimizing Energy Consumption", *International Journal of Big Data Intelligence*, Vol. 5, No.: 4, 2018, pp. 270-287.
15. I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International Journal of Supercomputer Applications*, vol.15, no.3, 2001, pp. 200-222
16. F. Gens, "IT Model in the Cloud Computing Era," *IDC Enterprise Panel*, August 2008.
17. F. Gens, "IT Cloud Services Forecast – 2008, 2012: A Key Driver of New Growth," *IDC Exchange*, October 2008.
18. IBM, "Detailed Scheduling in IBM ILOG CPLEX Optimization Studio with IBM ILOG CPLEX CP Optimizer", White Paper. IBM Corporation, 2010.
19. K. Kc and K. Anyanwu, "Scheduling Hadoop Jobs to Meet Deadlines", in *Proceedings of the International Conference on Cloud Computing Technology and Science (CloudCom 2009)*, Indianapolis, USA, December 2010, pp. 388-392

20. K. Khairunnisa, J. Ayob, A. Mohd. Helmy, M. Wahab, M. Erdi Ayob, M. Izwan Ayob, A. Ayob., "The Application of Wireless Food Ordering System", *MASAUM Journal of Computing*, vol. 1, Issue: 2, 2009, pp. 178-184.
21. J. Koomey. "Growth in data center electricity use 2005 to 2010", *Analytics Press*, August, 2011.
22. D.T. Lau, J. Liu, S. Majumdar, B. Nandy, M. St-Hilaire, C.S. Yang, "A Cloud-Based Approach for Smart Facilities Management", in *Proceedings of the IEEE Conference on Prognostics and Health Management (PHM 2013)*, Gaithersburg, United States, June 2013, pp. 1-8.
23. N. Lim, S. Majumdar, P. Ashwood-Smith, "MRCP-RM: a Technique for Resource Allocation and Scheduling of MapReduce Jobs with Deadlines", *IEEE Transactions on Parallel and Distributed Systems*, Vol.: 28, No: 5, 2017, pp. 1375-1389.
24. N. Lim, S. Majumdar and P. Ashwood-Smith, "A Constraint Programming Based Hadoop Scheduler for Handling MapReduce Jobs with Deadlines on Clouds", in *Proceedings of the ACM/SPEC International Conference on Performance Engineering (ICPE 2015)*, Austin, United States, February 2015, pp. 111-122.
25. N. Lim, S. Majumdar, P. Ashwood-Smith, "Engineering Resource Management Middleware for Optimizing the Performance of Clouds Processing MapReduce Jobs with Deadlines", in *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*, Dublin, Ireland, March 2014, pp. 161-172.
26. Lindo Systems Inc., "Lindo Systems – Optimization Software", <http://www.lindo.com/>, Accessed: November 15, 2017.
27. D. C. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, 3 ed., Addison-Wesley Longman Publishing Co.,USA, 2001.
28. S. Majumdar, M. Asif, J.O. Melendez, R. Kanagasundaram, D.T. Lau, B. Nandy, M. Zaman, P. Srivastava, N. Goel, "Middleware Architecture for Sensor-Based Bridge Infrastructure Management", in *Proceedings of the 15th Communications and Networking Symposium (CNS 2012)*, Boston, USA, Article No. 8, pp, 1-10
29. Market Wired, "Smarter Cities, Better Vaccines, Greener Buildings: CANARIE Invests in Research Software Tools that Drive Innovation", June 2014, <https://finance.yahoo.com/news/smarter-cities-better-vaccines-greener-175508147.html>, Accessed Oct 24, 2017.
30. A. McGregor, D. Bennett, S. Majumdar, B. Nandy, J.O. Melendez, M. St-Hilaire, D.T. Lau, J. Liu, "A Cloud-Based Platform for Supporting Research Collaboration", in *Proceedings of the 8th IEEE International Conference on Cloud Computing (CLOUD 2015)*, New York, United States, July 2015, pp. 1-4.
31. Microsoft, Windows Azure, <http://www.windowsazure.com/en-us/>, Accessed: November 15, 2017.
32. Micro Works, "Take a Stroll Around Your Dining Room", <http://www.microworks.com/products/Handheld-Ordering.htm>, Accessed: November 14, 2017.
33. MySQL, <https://www.mysql.com/>, Accessed: November 15, 2017.
34. NEXTSTEP, "Dynamic Menu Displays Application", <http://www.nextstepsystems.com/dynamic-menu-displays>, Accessed: November 14, 2017.
35. NFC Forum, "What is NFC?", <https://nfc-forum.org/what-is-nfc/>, Accessed: November 15, 2017.
36. H. Saeed, A. Shouman, M. Elfar, M. Shabka, S. Majumdar, C.H. Lung, "Near-Field Communication Sensors and Cloud-Based Smart Restaurant Management System", *the Proceedings of the IEEE 3rd World Forum on Internet of Things (WF-IoT 2016)*, Reston, USA, December 2016, pp 686-691.
37. Spectrum "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated", *IEEE Spectrum*, Aug 2016, IEEE <https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated>, Accessed: November 15, 2017.



38. A. Verma, L. Cherkasova, V.S. Kumar, and R.H. Campbell, "Deadline-Based Workload Management for MapReduce Environments: Pieces of the Performance Puzzle", in *Proceedings of the Network Operations and Management Symposium (NOMS 2012)*, Maui, USA, April 2012, pp. 900-905
39. L. Wang, S. Majumdar, C.-H. Lung, "A Near Field Communication Based Access Control and Information Dissemination System" (invited paper), *Computer Society of India Journal of Computing*, Vol. 3, No.: 1, 2017, pp.1-12.
40. T. White, *Hadoop: The Definitive Guide, 2nd Edition*, O'Reilly Media, Inc., USA, 2011.

# Mobile Computing, IoT and Big Data for Urban Informatics: Challenges and Opportunities



Anirban Mondal, Praveen Rao, and Sanjay Kumar Madria

## 1 Introduction

Over the past few decades, the population in the urban areas has been increasing in a dramatic manner. Currently, about 80% of the U.S. population and about 50% of the world's population live in urban areas and the population growth rate for urban areas is estimated to be over one million people per week [1, 2]. By 2050, it has been predicted that 64% of people in the developing nations and 85% of people in the developed world would be living in urban areas [1, 2]. Such a dramatic population growth in urban areas has been placing demands on urban infrastructure like never before [1]. Furthermore, the growth of urban infrastructure can be reasonably expected to lag far behind the population growth in urban areas due to practical considerations and cost issues. Thus, it is becoming increasingly important to make *efficient* use of existing urban infrastructure in order to cater to the needs of the population living in urban areas.

In this regard, urban informatics is emerging as a new multi-disciplinary field for obtaining an in-depth understanding as well as valuable insights about providing key services to the residents of urban areas. Urban informatics has been defined in various ways in existing literature. For example, urban informatics has been defined in [3, 4] as “the study, design, and practice of urban experiences across

---

A. Mondal  
Ashoka University, Sonapat, Haryana, India  
e-mail: [anirban.mondal@ashoka.edu.in](mailto:anirban.mondal@ashoka.edu.in)

P. Rao  
University of Missouri-Kansas, Kansas City, MO, USA  
e-mail: [raopr@umkc.edu](mailto:raopr@umkc.edu)

S. K. Madria (✉)  
Missouri University of Science and Technology, Rolla, MO, USA  
e-mail: [madrias@mst.edu](mailto:madrias@mst.edu)

different urban contexts”. According to the study in [2], “ Urban informatics uses data to better understand how cities work”. Furthermore, Wikipedia defines urban informatics as “an interdisciplinary field which pertains to the study and application of computing technology in urban areas” [5]. In essence, urban informatics concerns applying information technology to improve the lives of citizens in urban areas.

According to a study by McKinsey [6], there are three key themes across today’s urban informatics initiatives, namely (a) the use of existing city data for improving efficiency (b) the creation of new data for facilitating decision-making about city planning as well as the operational aspects of cities (c) and improving the engagement of the public towards problem-solving in cities. Thus, today, mobile computing, IoT (Internet-of-Things), and big data are enabling state-of-the-art technologies for advancing urban informatics [7]. For example, New York City produces a terabyte of raw data every day about electricity to parking tickets [2]. The City’s Department of Transportation analyzes hundreds of gigabytes of taxi trip data using an interactive analytics system called TaxiVis [8]. Furthermore, huge amount of data is also generated from IoT e.g., the data from Geographic Positioning Systems (GPS), car accelerometers, temperature/pressure sensors, heart monitoring implants and several other types of sensors.

Incidentally, the ever-increasing popularity of mobile devices and the dramatic technological advances in their embedded sensors (e.g., GPS, accelerometers, gyroscopes, high-resolution cameras etc.) coupled with the prevalence of social media have significantly contributed to urban informatics initiatives becoming increasingly *people-centric*. In fact, mobile crowdsourcing/crowdsensing [9–11] embodies the people-centric approach in that it can be perceived as an anonymized user information system, where mobile users use their devices to contribute data (typically in lieu of some **incentives**) about a wide gamut of city-related events such as broken pavements, dysfunctional street lights and illegal garbage dumps. Interestingly, large amounts of mobile crowdsourced/crowdsensed data are also generated when mobile users respond to queries such as: (a) *Where are the top-k cheapest available parking spots within 3 km of my current location?* (b) *What are the most desired fuel efficient routes between 5 PM and 7 PM on Saturdays in Atlanta to reach airport?*

Observe how the proliferation of mobile devices has unlocked tremendous potential opportunities for mobile crowdsensing in urban informatics. Moreover, mobile crowdsensing is cost-efficient since it leverages the existing mobile devices and sensors of a large number of mobile users, thereby reducing the need to install a massive number of dedicated sensors. Since the users are mobile, data can effectively be collected across a wide gamut of locations. Furthermore, the large number of users ensures the collection of data at a scale that has never been seen before. Given that people are the *producers* as well as the *consumers* of information, we believe that the importance of the people-centric mobile crowdsourcing/sensing angle in urban informatics will continue to grow dramatically in the future. Figure 1 depicts a high-level overview of mobile crowdsensing and its applications, while Fig.2 depicts the mobile sensing system with its components and key functionalities.

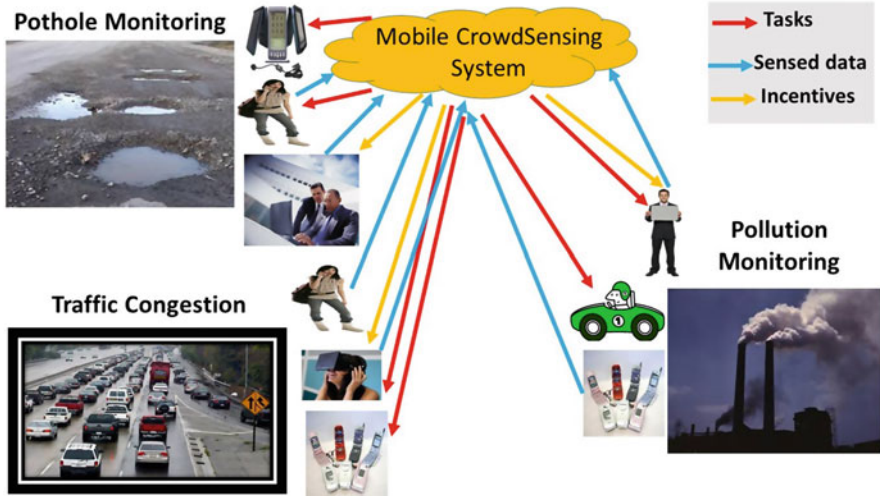


Fig. 1 High-level overview of mobile crowdsensing and its applications

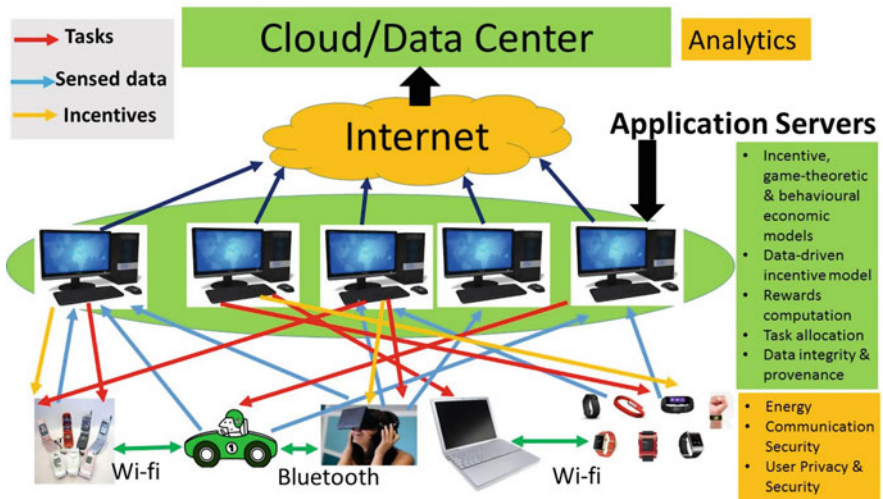


Fig. 2 Mobile crowdsensing system: components and key functionalities

The large-scale nature of data collected by city governments necessitates the use of big data technologies, which constitute the backbone of many successful companies. Such data can have tremendous value for obtaining actionable insights towards significantly enhancing applications for creating novel next-generation mobile applications for urban informatics. For example in transportation and logistics, the data can significantly facilitate traffic congestion monitoring, traffic navigation, traffic accident response and road conditions monitoring. In pollution

control, the data can help in identifying illegal garbage dumps or toxic waste disposal at sensitive places (e.g., near hospitals) and by detecting when garbage bins overflow. In a similar vein, in law enforcement, the data can quickly alert police officers to street crimes, traffic violations and illegal car parking. In healthcare, the data can be used for monitoring the health of at-risk individuals (e.g., elderly or disabled persons), detecting fake pharmaceutical products and understanding disease epidemiology. The data can also help municipal governments in monitoring street-lights, potholes, pavements and water-pipes.

Intuitively, we can understand that urban informatics systems can add a significant amount of value to the lives of people living in urban areas. However, there are several research challenges associated with realizing such urban informatics systems. These challenges include (but are not limited to) scalability, data security and privacy, context-aware analytics, personalization, free-riding and the design of sustainable incentive schemes (for encouraging mobile user participation in contributing data to the system) and mobile resource constraints (e.g., energy and bandwidth). In this article, we highlight research issues and ideas with respect to effectively gathering, analyzing and querying the massive and heterogeneous amounts of data that are generated from IoT environments and user interactions on mobile devices.

The remainder of this chapter is organized as follows. Section 2 presents an overview of infrastructure and frameworks for smart city management. Section 3 describes systems and approaches for facilitating effective city management in different domains. Section 4 describes incentive-based data collection and management. Section 5 discusses big data management and analytics in mobile and IoT environments. Section 6 discusses knowledge management issues for IoT applications. Section 7 highlights the security and privacy issues in IoT environments. Finally, our concluding remarks are presented in Sect. 8.

## **2 Infrastructure and Frameworks for Smart City Management**

Several efforts have been made to develop infrastructure and frameworks for smart city management. The work in [12] discusses a vision for the Future Internet (FI) and its relationship with smart cities in terms of its components such as Internet of Things (IoT) and Internet of Services (IoS). The goal here is to create a unified urban-scale open innovation platform for smart city management. At the infrastructure level, IoT will support the functioning of heterogeneous sensors that are deployed in urban areas. On the other hand, at the service level, IoS will act as a suite of open and standardized enablers for supporting interoperable services in a smart city. Furthermore, the work in [13] presents a vision for IoT, including a Cloud-centric vision for the realization of IoT. It also discusses the key enabling technologies as well as application domains that would likely assume importance in

future research on IoT. Moreover, it presents the open research challenges associated with implementing a Cloud-centric IoT.

The work in [14] presents a city planning framework, which is designated as the Smart City Reference Model. The goal of the model is to facilitate city planners in conceptualizing how to build smart city innovation ecosystems. Thus, city planners can use the model as a guide to defining the conceptual layout of a given city. In particular, the framework comprises a number of layers such as green, interconnected, open, integrated, intelligent and innovating layers. Given that cities can come with a wide gamut of characteristics, the layered nature of the model makes it flexible enough to be adopted towards designing smart city policies for different types of cities. Furthermore, the Smart City architecture presented in [15] is an event-driven architecture for monitoring public spaces in a given city. In particular, the architecture facilitates the management and cooperation of heterogeneous sensors for effectively monitoring the public spaces. The architecture is also capable of detecting anomalous city-related events.

The Smart Cities Critical Infrastructure Response (SCCIR) framework described in [16] is aimed at protecting critical infrastructure of a given city and ensuring the continuity in the operations of those infrastructures. Hence, the SCCIR framework provides a response strategy to first responders in emergency scenarios (e.g., when fires need to be extinguished). The response strategy is based on information flows of smart cities. Additionally, for handling scenarios where information flows have been compromised due to failures in critical infrastructures, the SCCIR framework proposes a robust infrastructure state preservation system.

Incidentally, traditional Complex Event Processing (CEP) systems, such as Esper, lack scalability in dealing with huge amount of data and are generally not adequately flexible to adapt to unexpected changes in the monitored environments since they depend upon static user-defined rules [17]. Hence, the framework in [17] aims at scalable and dynamic processing of complex event data in cities. In particular, the proposed solution combines Esper with the Storm stream processing framework for ensuring scalability by parallelizing the data processing. Furthermore, the system is able to dynamically adapt to the changes of the monitored environment by doing analytics on the historical data and then dynamically updating the rules based on the results of the analysis. The framework has been applied to monitored traffic environments. However, the framework is also generally applicable to a wide range of monitored city environments.

The CityZen platform [18] facilitates event reporting and analytics for engaging residents towards city management through incentives. Authenticated residents report events in a given city using the CityZen mobile application, which interfaces with the CityZen platform. CityZen supports differentiated incentive management based on types and priorities of events, quality and timeliness of event reports as well as resident intent. Moreover, CityZen has a social dashboard for searching events and subscribing to event alerts.

The SmartCrowd framework [19] is aimed at optimizing task assignment in knowledge-intensive crowdsourcing for smart cities. SmartCrowd formulates the worker-to-task assignment problem as an optimization problem and takes into

consideration human factors such as worker expertise, wage requirements and worker availability. Additionally, detailed theoretical analysis of the task assignment optimization problem has been presented. Furthermore, both optimal and approximation algorithms with guarantees have been discussed in [19].

The work in [20] discusses frameworks and techniques for intelligent and effective urban data monitoring for smart cities. The framework considers issues such as data volume, velocity and veracity in addition to aspects such as data quality, privacy, security and resilience. Notably, the system and framework and solutions developed in [20] are currently being utilized by the city of Dublin. The work in [21] discusses a smart cities platform that leverages the capabilities of both cloud computing as well as IoT. The platform uses cloud computing capabilities for facilitating ubiquitous connectivity in order to satisfy the needs of real-time applications and services for smart cities. Data are obtained from a wide variety of heterogeneous data sources, including sensors and smart devices. Management and analytics on the data is performed by distributed cloud-based services.

A detailed survey of software architectures for smart cities can be found in [22].

### **3 Systems and Approaches for Facilitating Smart City Applications**

Given that cities typically face a wide gamut of management and maintenance problems across several important and diverse domains, there have been several efforts for facilitating effective city management. This section discusses various systems and approaches for facilitating city management in different domains such as transportation, waste management and environmental monitoring, and retail.

#### ***3.1 Transportation***

Given that transportation is one of the most critical aspects of smart city management, several systems have been proposed for sensing and analyzing road and traffic conditions [23–30], traffic navigation [31, 32] and car parking [33].

##### **3.1.1 Sensing Road Conditions**

The work in [23] aims to explain the underlying reasons concerning traffic conditions based on relevant information of events such as dysfunctional traffic lights, accidents or road maintenance works. Hence, static data are collected from event providers, while dynamic data about events are collected through social media. Thus, users are provided with real-time information about the causes of traffic

congestion, thereby enabling them to visualize traffic conditions and understand the reasons for traffic congestion so that they can make better routing decisions. Furthermore, the Sipresk platform [24] is a big data analytics platform for enabling smart transportation. The platform performs analytics on urban transportation data for obtaining insights into traffic patterns. The platform has a layered architecture comprising data, analytics and management layers. Moreover, the platform uses the Cloud for achieving scalability and reliability as well as for both online and retrospective analysis.

The Ushahidi platform [25] is being used for urban transportation planning in Beijing. The Ushahidi platform uses crowdsourcing through a web-based platform as well as by means of smartphone applications, thereby enabling urban planners to obtain feedback directly from the users. Thus, Ushahidi enables urban planners towards effectively identifying areas of improvement in urban transportation. The FixMyStreet platform [26] involves residents in city management by encouraging them to report road conditions such as broken pavements, potholes and dysfunctional street lights. In particular, the FixMyStreet platform is an open source project, which is used worldwide for facilitating the reporting of road conditions and city-related problems. Moreover, the FixMyStreet platform enables users to view reports concerning problems and additionally allows users to subscribe to alerts.

The Nericell system [27] uses smartphone sensors (e.g., accelerometer, GPS sensors, microphone) to sense road conditions such as potholes and bumps. In particular, Nericell focusses on traffic flow patterns in developing regions, where traffic conditions are generally more complex and chaotic. Nericell also emphasizes energy efficiency e.g., it performs localization and honk detection in an energy efficient way. The speed-breaker early warning system proposed in [28] uses a mobile application to notify the driver when her vehicle is coming close to a speed-breaker. In particular, the system uses smartphone accelerometers to detect speed-breakers. The system is primarily aimed at developing regions, where speed-breakers are not always visible due to lack of warning signs, especially at night or during other low-visibility conditions such as fog. In such situations, vehicles approaching a given speed-breaker at a relatively high speed can cause accidents. The work in [29] focuses on analyzing anomalous road conditions by using a mobile phone with a tri-axial accelerometer for collecting data concerning acceleration of motorcycle riders. Understandably, road anomalies, such as uneven road surfaces, can be extremely unsafe especially for motorcycle riders. Hence, the work in [29] uses both supervised and unsupervised machine learning approaches for detecting road anomalies.

Furthermore, the RoadEye system [30] enables personalized retrieval of dynamic road conditions. Notably, awareness of dynamically changing road conditions is of paramount importance for a safe and quality driving experience, as well as for augmenting trip planning. The goal of RoadEye is to keep users informed in a timely and personalized manner about road conditions arising from both scheduled and ad hoc events. For enabling quick and personalized retrieval of user-queried road conditions, RoadEye proposed the  $\psi$ R-tree, which is an R-tree-based index



augmented with linked lists that store information about various types of events. Thus, users can query the RoadEye system about various kinds of road conditions. Notably, the R-tree [34] is a popular spatial index structure. In the R-tree, each spatial data object is represented by a Minimum Bounding Rectangle (MBR). Non-leaf nodes of the R-tree contain entries of the form (ptr, rect) where ptr is a pointer to a child node in the R-tree and rect is the MBR that covers all of the MBRs in the child node. On the other hand, leaf nodes of the R-tree contain entries of the form (oid, rect) where oid is a pointer to the object in the database and rect is the MBR of the object.

Incidentally, traffic events and road conditions can be detected using the sensor network-based infrastructure, but the implicit assumption here is that the sensors are reliable. Hence, for reliably sensing traffic conditions, the work in [35] focuses on real-time detection of faulty traffic sensors. In particular, the approach proposed in [35] performs a real-time resolution of the source of irregular sensor readings by exploring the correlation among neighbouring sensors as well as by means of crowdsourcing.

### 3.1.2 Facilitating Traffic Navigation and Parking

Systems, such as Waze [31] and IBM InfoSphere Streams [32], facilitate traffic navigation. Waze [31] is a popular crowdsourcing-based traffic navigation application. Millions of drivers share real-time information about traffic conditions (e.g., traffic accidents, road blockages, traffic congestion) using Waze, thereby enabling safe and efficient trip planning. Given Waze's huge user base, it is able to provide its users with updated and real-time information about traffic conditions. Waze also offers additional services such as facilitating its users in locating nearby cheap gas stations by maintaining crowdsourced data about fuel prices. IBM InfoSphere Streams [32] is a scalable stream processing platform for facilitating intelligent transportation services. The platform uses real-time location-based data from a wide gamut of sources and amalgamates this information with static map databases for answering various kinds of transportation-related one-time queries as well as continuous queries from users. The platform performs various kinds of detailed analysis on the dynamic data (real-time traffic information) and the static data (map data) to answer user queries such as finding shortest-time routes.

For facilitating efficient parking, ParkNet [33] is a mobile system consisting of vehicles. While driving by, the vehicles collect information about parking space occupancy. For collecting data about parking space occupancy, each ParkNet vehicle is fitted with a passenger-side-facing ultrasonic rangefinder and a GPS receiver. The data is then transmitted to a centralized server, where data aggregation is performed to create a real-time map of parking space availability. Thus, drivers can query the ParkNet system when they are trying to find parking space in a given location.

### 3.2 *Waste Management and Environmental Monitoring*

Waste management in cities involves managing waste collection, distribution and routing [36]. Under the traditional approach, waste collection trucks need to physically go to dumpsters to check the trash levels at fixed times. However, the problem with the traditional approach to waste collection is that the waste collection times do not have any alignment to the rate at which the trash bins get filled up. Thus, if the waste collection timings are spaced too far apart, potential overflows may occur in the trash bins. On the other hand, if waste collection is performed too frequently, the trash bins may be under-filled, thereby making the process inefficient as well as unnecessarily wasting time and fuel. Hence, determination of the optimal intervals of time during which the waste collection needs to be done by the trucks becomes critical. Hence, the waste management systems described in [37–40] and companies, such as Enevo [41] and SmartBin [42], use sensors/IoT in conjunction with visualization dashboards for determining the frequency with which to perform waste collection.

The waste management system described in [37, 38] uses intelligent sensor-based containers for estimating the waste content in order to optimize the waste collection. The system uses distributed sensor technology and geographical information systems. In particular, it uses a network of sensorized waste containers that can communicate with a data management system. Thus, the data management system obtains information about the amount (in terms of both waste weight and volume) and type of waste material at different collection points, thereby enabling it to optimize the routing of waste collection vehicles as well. Additionally, the system tries to predict the amount of solid waste content to be collected based on residential population, season and consumer index. The system has been tested in the Pudong New Area, Shanghai.

Furthermore, the solid waste bin monitoring system in [39] uses wireless sensor network technology in conjunction with ZigBee and GSM/GPRS communication technologies for monitoring solid waste bins on a real-time basis. The system has a three tier architectural structure. The lowest tier has the sensorized bins and it uses an energy-efficient sensing algorithm to collect the waste-related bin data. The bin-related status data is transmitted to the middle tier. The middle tier has the communication gateway for transmitting the bin data to the uppermost tier, which consists of a centralized control station. The control station stores the data and performs analytics on the data. Moreover, it uses the data as input to a decision support system for facilitating the optimization of waste collection vehicle routes.

The versatile and scalable smart waste-bin system discussed in [40] can be used in conjunction with common-type waste-bins to determine their fill-level estimates, while maintaining energy-efficiency. The system comprises low-cost embedded components and the sensing units are based on ultrasonic sensors. Moreover, the system uses RFID technologies with active RFID tags for retrieving the information. In a similar vein, the work in [43] discusses the concept, methodology and architecture of an integrated node for supporting smart-city applications, while conserving

energy and using low-cost components. For performing fill-level estimation of the waste bins, the sensing units used are ultrasonic sensors. The integrated node uses RFID technology.

Enevo [41] is an innovative waste management company. Enevo has a proprietary dumpster sensor and software system that is placed on the lids of garbage receptacles. Thus, the system is capable of communicating regarding the trash level of a given garbage container. This facilitates waste collection agencies to follow the fill levels of the waste containers, thereby enabling effective streamlining of their waste collection operations. Enevo also performs predictive analytics to predict when a dumpster will be full, thereby facilitating the route planning of the waste collection trucks in advance. Thus, Enevo provides a data-driven way to improve the efficiency of waste collection management and operations.

SmartBin [42] provides intelligent remote monitoring solutions for waste collection and management. It uses IoT-based Smart monitoring by means of Ultrasonic Level Sensor (UBi), which is among the most widely deployed fill-level sensors. These sensors are deployed across a wide gamut of waste containers. In particular, the plug-and-play nature of the IoT devices significantly facilitates the reporting of real-time data to the SmartBin Live platform. This enables waste management agencies in keeping track of the fill-level of the waste containers and eliminates the possibility of over-filling the waste containers. Furthermore, SmartBin Live is the IoT game changer for collectors and distributors in that it not only helps in monitoring waste containers, but also plans for route optimization of the waste collection trucks. In particular, optimized routes are sent directly to the drivers on their smartphones, thereby improving the efficiency of the waste collection operations. Additionally, the web-based platform has the capability of tracking the drivers of the waste collection trucks.

A decision support system for efficient IoT-enabled waste collection in smart cities was proposed in [44]. The system uses a model for sharing data between truck drivers on a real-time basis for performing waste collection and dynamic route optimization. Moreover, the system also supports waste collection in relatively inaccessible areas by using surveillance cameras to capture the inaccessible areas so that the municipal authorities can be made aware of such areas. A survey on the challenges and opportunities of IoT-enabled waste management systems can be found in [45].

Additionally, environmental monitoring is becoming increasingly important for smart cities. In this regard, PEIR (Personal Environmental Impact Report) [46] is a participatory sensing application. PEIR samples location data from users' mobile phones and utilizes this data for computing estimates of environmental pollution. Server-side processing in PEIR also includes activity recognition and classification e.g., for identifying modes of transportation. Thus, PEIR essentially uses mobile crowdsourcing towards facilitating city governments in pollution monitoring.

### ***3.3 IoT Technology and Retail Industry***

The work in [47] examines the business and social impact of IoT on the retail industry from the viewpoints of security, reliability, integration, discoverability and interoperability. It explores new ideas for business profitability for the retail industry using IoT technologies, the key focus areas being on embedded systems, cyber physical systems and generic sensors. It predicts that the focus of leveraging IoT technologies for the retail industry will shift from data collection to knowledge creation.

The case study in [48] explores the impact of RFID (radio frequency identification) technology and the electronic product code (EPC) network on mobile B2B eCommerce. In particular, the case study indicates insights concerning business processes such as shipping. Furthermore, the case study suggests that the RFID-EPC network can foster more effective information sharing between the members of the retail supply chain and the RFID-EPC network needs to be integrated into the broader retail business strategy. Furthermore, the work in [49] explores the value created by Big Data for the retail industry. In particular, it focuses on predictive analytics of the Big Data generated in the retail industry for making decisions about pricing and merchandising.

The work in [50] focuses on enabling the experience of shoppers by trying to ensure the on shelf availability (OSA) of the shopper's desired products. Hence, it proposes an Internet of Things/Internet of Everything (IoT/IoE) enabled framework for enhancing the OSA. Moreover, it examines various sensor configurations that can be used to improve the OSA in the retail industry. These sensors in conjunction with big data analytics facilitate in creating information concerning out-of-stock products on the retail store shelves, thereby enabling store managers to stock products on those shelves. Prediction of trends about out-of-stock product scenarios is also performed. Observe how the framework improves operational efficiencies and the profitability of retail stores.

The Android-based mobile app, designated as SmartMart [51], aims at making it easier for shoppers to locate their desired products in a given large-scale supermarket. Given that retailers often lose out on sales and profitability due to shoppers being unable to easily locate their desired products, providing assistance to shoppers towards locating their desired products has indeed become a necessity for the retail industry. SmartMart leverages IoT-based technologies to enable store products to automatically register their location information in an information retrieval system, thereby providing an IoT-based in-store mapping of products. This enables shoppers to use their mobile devices in searching and locating their desired products within the large supermarket.

In a similar vein, for assisting shoppers in a shopping mall, a PDA-based system was implemented in [52]. The PDA-based system provides the shoppers with directions in a shopping mall based on the shopper's stated product preferences, the shopper's location as well as the products purchased previously by a given

shopper. The system uses a decision-theoretic planning approach for optimizing the expected utility of a shopper in terms of the time required for each purchase and the uncertainty associated with the shopper being actually able to locate the desired products in a given location within the shopping mall.

### 3.3.1 Discussion and Insights

The infrastructure and frameworks discussed in Sect. 2 and the smart city applications discussed in this section could either be loosely or tightly coupled. An infrastructure could be proprietary i.e., only a single organization could build applications on top of the infrastructure and framework, thereby creating a tight coupling between the infrastructure and the applications. On the other hand, the infrastructure could be loosely coupled i.e., it could be used as a foundation based on which applications could be designed by any interested stakeholders. In effect, one could envisage the infrastructure and the applications as part of a layered hierarchy that is either loosely or tightly coupled. For example, CityZen could form a substrate for some of the smart city applications that we have discussed.

## 4 Incentive-Based Mobile and IoT Data Collection and Management

Data collection in mobile and IoT environments typically occurs by means of implicit sensor inputs (e.g., dedicated temperature/pressure sensors in buildings, car accelerometers etc.) as well as explicit user inputs (i.e., mobile crowdsourcing). While dedicated sensor and IoT data sources provide advantages in terms of data reliability and standardized data formats, explicit user inputs enable the collection of newer kinds of valuable context-rich data, which are inherently better aligned to people's needs; this is an essential pillar in the increasingly popular *people-centric* approach to urban informatics.

Incidentally, in order to realize a mobile crowdsensing system for facilitating urban informatics, effective large-scale data collection becomes critical. The key research challenge here arises from the design of *sustainable incentive* mechanisms to entice mobile user's participation towards contributing data to the mobile crowdsensing system. Hence, we will now highlight research issues and ideas with respect to effectively designing and implementing sustainable incentive mechanisms for mobile crowdsensing.

#### ***4.1 Incentive-Based Practical Infrastructure for Mobile Crowdsensing***

Today, mobile devices come equipped with a wide gamut of inherent hardware capabilities in sensing, storing, processing, communication and visualization interfaces. Moreover, mobile apps add value to such capabilities. However, all of these capabilities of mobile devices due to hardware and mobile apps do not provide any direct support for incorporating incentive mechanisms into their sensing. Since the incorporation of incentive schemes is critical for mobile users to be enticed adequately to participate effectively in sensing, infrastructure and middleware for incentive-based mobile crowdsensing become critical.

In this regard, the work in [53] discusses the design and implementation of Medusa, a programming framework for mobile crowdsensing. It incorporates support for incentives, privacy and security in addition to facilitating support for humans-in-the-loop for triggering actions associated with crowdsensing or for reviewing results. Moreover, it provides high-level abstractions for the steps that are typically needed to complete a crowdsensing task. Its distributed runtime system coordinates the execution of the crowdsensing tasks between the mobile devices and the Cloud.

Moreover, the work in [54] proposes a mobile crowdsourcing approach, which is designated as CrowdMAC. In CrowdMAC, mobile users create a marketplace for mobile Internet access. Users that have residual capacity in their data plans, share their access with other users in their proximity in lieu of a small fee. In its middleware framework, CrowdMAC incorporates incentive mechanisms for admission control, service selection, and mobility management. CrowdMAC was implemented and evaluated using Android phones as a testbed. Furthermore, the work in [55] discusses a privacy-preserving truth discovery (PPTD) framework for mobile crowdsensing systems. PPTD is a Cloud-enabled framework, which is capable of protecting both the users' sensory data and their reliability scores; such reliability scores are derived by the truth discovery approaches. The PPTD framework performs weighted aggregation on users' encrypted data using homomorphic cryptosystem. For handling big data, PPTD is parallelized with MapReduce framework.

We observe that while the infrastructure for mobile crowdsensing [53–55] provides some support for incorporating incentive mechanisms, they do not provide support for facilitating the incorporation of more complex incentive mechanisms that are based on game theory and behavioral economics. As such, they do not provide any high-level abstractions for supporting the more complex incentive mechanisms. A solution could be to include new modules and high-level abstractions of the typical steps that are required in case of complex incentive schemes into the existing infrastructure. We perceive this as an extensible and rich library of modules for supporting a wide gamut of incentive mechanisms, from which developers could select one or more mechanisms and build the newer and more complex mechanisms on top of them.

## 4.2 *Security and Privacy of Incentive Mechanisms for Mobile Crowdsensing*

Security and privacy is of utmost importance in mobile crowdsourcing due to many facets of the computing and networking infrastructure involved right from the distribution of tasks to users, their devices, data collections and responses, locations, wireless communication and computing platforms. Many of the algorithms as discussed use incentives, but users should not be able to exploit the incentive mechanisms to increase their own utility without making the requested contributions or should not provide false reports to earn incentives. Adversaries can launch Sybil type of attacks, where they can impersonate as multiple users and try to earn incentives multiple times for the same task. They should adhere to the security and privacy policies defined by the system so that data, patterns, location, user anonymity, and linkability of tasks and social relationships can be protected.

The work in [56] observes that auction-based incentive mechanisms suffer from the drawback of failing to preserve the privacy of each user's bid. They assert that since a given user's bid generally includes her private information, the privacy of such private information should be preserved. They design an incentive mechanism, which is based on the single-minded reverse combinatorial auction. Their incentive mechanism is privacy-preserving, approximately truthful and computationally-efficient. However, malicious users can interfere with data collection processes and tasks by putting false reports and may also eavesdrop on the communication.

Moreover, the work in [57] also proposes a privacy-preserving incentive scheme for mobile crowdsensing. They discuss a reverse auction mechanism, which lets users determine their own price for the data that they contribute, while also incentivizing the submission of data with better quality. Their auction protocol ensures bidders' anonymity by allowing users to claim their rewards without linking them to the data contributed by them.

The work in [58] proposes two credit-based privacy-conscious incentive schemes for mobile crowdsensing systems. In both schemes, mobile users earn credits by contributing data without divulging any information about the data that they have contributed. The first scheme uses the notion of an online trusted third party (TTP) to preserve user privacy and to prevent malicious attacks. On the other hand, the second scheme deals with those cases, where online TTPs are not available. Thus, it uses blind and partially blind signatures as well as an extended version of the Merkle Hash Tree technique for preserving user privacy and to prevent attacks by malicious users.

Furthermore, the work in [59] observes that in a Mobile P2P (M-P2P) network, selfish nodes can drop packets, thereby impacting the efficiency of the whole network. Hence, they propose a mechanism based on virtual currency to identify selfish nodes in the network. Each node sends a receipt to its broker for providing evidence that it has performed a forwarding service. Each broker examines the receipts that it has received, and rewards well-behaved nodes with virtual currency, while penalizing selfish nodes for dropping packets. Observe that the virtual

currency-based incentive mechanism in [59] to identify selfish nodes in the network can also be applied in mobile crowdsensing applications for dealing with malicious users.

Given the security and privacy-preserving solutions proposed, there is a need to provide improved solutions which can scale (users leaving and joining) and are energy-efficient in terms of handling large number of users, tasks, devices etc. They should be applied to a large class of incentive-based auction mechanisms, secure biddings, secure payments, dynamic reputation and credibility of users. The solutions should be decentralized, light-weight and both computation- and communication-efficient to run also on the mobile devices.

### ***4.3 Energy-Efficient Incentive Mechanisms for Mobile Crowdsensing***

Conserving the energy of the mobile devices and sensors is of critical importance in enabling effective crowdsensing in a sustainable manner. Mobile devices typically suffer from resource constraints, and one of those key resource constraints is energy. Although today's smartphones, tablets and other mobile devices have more battery power than ever before, the demands on their energy resources have also been increasing like never before. For example, activities such as live streaming of video using mobile devices impose significantly on the limited energy resources of the mobile devices; such activities have indeed become extremely commonplace and popular in today's world. Hence, it has become a necessity to design energy-efficient incentive mechanisms for mobile crowdsensing.

Now let us discuss some of the existing works, such as [60, 61], which concern energy-efficient incentive mechanisms for crowdsensing. Notably, demand response (DR) alludes to dynamic demand mechanisms, whose objective is to manage electricity consumption in response to supply-side signals [60]. In this regard, the work in [60] examines how to incentivize the widespread adoption of DR in the residential sector. They assert that behavioral incentive mechanisms for DR would require to encourage the desired energy consumption behavior among residents, while also being effective at maintaining the long-term engagement of residents. To this end, they performed a crowdsourcing experiment with the goal of exploring appropriate behavioral incentive mechanisms for DR. In particular, they collected 55 ideas from 27 different users, and classified them based on Fogg's Behavior Model [62]. The detailed findings and results of their experiment can be found in [60].

Interestingly, works on incentive schemes in Mobile-P2P networks can also be applied to mobile crowdsensing. For example, the work in [61] discusses the E-ARL incentive scheme for Mobile-P2P networks. In E-ARL, each data item has a price in virtual currency. The price of a data item depends on factors such as its access frequency, its expiry time and the energy of its host peer. E-ARL requires a query



issuing peer to pay the price of its requested data item to the peer serving its request and a commission to each relay peer in the successful query path. E-ARL's incentive scheme conserves the energy of low-energy mobile peers by increasing the message relay commissions at a given peer as its energy decreases. This facilitates network connectivity since query-issuing peers would likely prefer lower cost query paths for obtaining their requested data items. Furthermore, as the remaining energy of a peer decreases, the price of accessing data items at that peer increases. Thus, peers would be less likely to access data items at low-energy peers if they are able to obtain them at higher-energy MPs, where item prices would be lower. This conserves the energy of low-energy host peers.

Energy-efficient crowdsensing opportunities to satisfy crowdsourcing requirements are challenging as incentives associated with data quality requirements may need to use greedy approaches, but they may consume more energy on the mobile devices/sensors and therefore, more incentives would need to be distributed. If clients wait for responses for the next better opportunity to get better quality data, it may delay pending tasks to sense, upload, or apply computation to the data and thus, may reduce the incentives earned. Thus, algorithms and benchmarks are needed to measure the data quality, latency and incentive models to optimize the energy usage with respect to various application domains.

#### ***4.4 Game Theory and Behavioral Economics for Incentive Mechanism Design***

Mobile crowdsensing is essentially a people-centric paradigm, hence the success or failure of a given mobile crowdsensing campaign fundamentally depends upon whether it is able to incentivize users towards contributing to the system. However, human beings are complex in that their behavior with respect to their likely responses to different incentive mechanisms may not necessarily follow any easily deterministic or predictable model. Thus, a comprehensive understanding of human behavior should play a key role in designing incentive mechanisms for mobile crowdsensing. Hence, works, such as the ones in [10, 63, 64], have used game theory and behavioral economics to model different aspects of human behavior for incentive mechanism design.

The work in [10] proposes two incentive models, namely the *platform-centric model* and the *user-centric model*, for crowdsensing using mobile phones. In the platform-centric model, the platform gives a reward that is shared by the contributing users and the incentive mechanism is designed based on a Stackelberg game with the platform as the leader and the users as the followers. The computation of the Stackelberg Equilibrium, where the platform's utility is maximized and no user is able to improve its utility by deviating from its current strategy, is also described. On the other hand, the user-centric model uses an auction-based incentive mechanism, which is both rational and truthful.

Furthermore, the work in [63] discusses a reward-based collaboration mechanism, in which the crowdsensing platform announces a total reward that is to be shared among collaborating users. The collaboration is deemed to be successful if there is an adequate number of users, who are willing to collaborate. In particular, they propose a Stackelberg game-based incentive mechanism. Their proposed mechanism is capable of handling asymmetrically incomplete information between users and the crowdsensing platform.

However, these incentive models suffer from some serious drawbacks from the perspective of sustainably incentivizing users over the long-term. These drawbacks are often a consequence of not taking into consideration the relevant human factors in mobile crowdsensing. For example, game-theoretic incentive models [10, 63] typically assume rational behavior on the part of the users. In other words, they assume that users will try to maximize their expected utility. However, this assumption does not necessarily hold good in practice because in reality, crowdworkers often do not have a clear understanding of how to select those sensing tasks, which would maximize their expected utility.

Moreover, users may also be intrinsically motivated towards performing certain mobile crowdsensing tasks. For example, if a user Alice feels strongly about eradicating graffiti on the walls in the streets of her city, she may send her crowdsensing report to the Cloud server (which would in turn report it to municipal authorities) even if she is not provided with any incentives to do so. Hence, she would be indifferent to maximizing her expected utility in this case. Herein lies the importance of behavioral economics theories in the design of incentive models. Works on behavioral economics examine different ways to model human behavior, while considering that human behavior is not based solely on expected utility.

Incidentally, incentive models based on behavioral economics theories have also been proposed in the crowdsourcing/crowdsensing literature, and the ideas are applicable to mobile crowdsensing. For example, the proposal in [64] examines situations when users need to decide under uncertainty about the crowdsourcing/crowdsensing tasks that they want to undertake. In particular, instead of modeling user behavior by means of the traditional expected utility theory (where users try to choose tasks with the goal of maximizing their expected utility), they model user behavior by means of prospect theory.<sup>1</sup> Then they explore the design of incentive models for crowdsourcing/crowdsensing assuming that user behavior in selecting tasks follows prospect theory. Notably, in behavioral economics, prospect theory states that people estimate their gains and losses using heuristics instead of trying to make optimal decisions when they are required to choose between probabilistic alternatives.

However, works on incentive model design, which try to model human behavior based on one of the classical behavioral economics models (e.g., the prospect model) suffer from a serious drawback in practice. While humans may sometimes behave based on a specific classical behavioral model, human behavior is generally far too complex to be made amenable to modeling by any combination of such models.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Prospect\\_theory](https://en.wikipedia.org/wiki/Prospect_theory)

Moreover, humans also have inherent biases, which further exacerbates the problem of modeling human behavior.

Interestingly, studies in behavioral psychology [65] have demonstrated that human behavior can be better incentivized when the rewards are unexpected or scheduled in a variable and unpredictable manner. In other words, providing the same level of rewards for performing a given task repeatedly does not necessarily incentivize humans over the long-term. This occurs because humans get used to the rewards. However, no incentive model can practically keep increasing the rewards for comparable tasks on a continual basis because it would pose severe impediments to cost-efficiency as well as sustainability.

Drawing from studies in behavioral psychology [65], a solution to this problem could be to incorporate an element of randomness and unpredictability into the design of the incentive model such that the crowdworkers would not have any pre-defined expectations or even notions of utility for completing crowdsensing tasks. Thus, the system could sometimes pay additional unexpected rewards to the crowdworkers, thereby encouraging their long-term engagement in the future; and at times, the rewards could be lower as well. In other words, this solution would always keep the crowdworkers guessing about the rewards that they would earn by completing certain sensing tasks. Such an element of chance would be more akin to a game, which would continue to keep the incentive model interesting as well as engaging to crowdworkers over the long-term. Incidentally, we assume here that crowdworkers are not relying on crowdsensing as their primary means of income, but they are performing crowdsensing to earn some additional money; this assumption would indeed be valid in the vast majority of real-world scenarios.

#### ***4.5 Data-Driven Incentive Mechanisms for Mobile Crowdsensing***

Given that the paradigm of mobile crowdsensing heavily relies on the willingness of the users in contributing to the system, a key challenge that typically arises in mobile crowdsensing campaigns is that there is generally little or no a priori knowledge about the participating users. As discussed in the previous section, the users' willingness to contribute may depend upon a wide variety of contextual factors including their preferences and belief systems. Hence, it becomes absolutely critical to obtain data about such contextual factors associated with any given user so that the incentive mechanisms can be designed so as to be driven by such data, thereby increasing the chances of success for the crowdsensing campaign as a whole. Hence, works, such as [66, 67] have proposed data-driven incentive mechanisms for mobile crowdsensing.

The work in [66] examines the design of optimal incentive-based mobile crowdsensing campaigns for maximizing the quality of user contributions when individual user preferences are unknown. They use logistic-regression techniques

from machine learning for learning user preferences from the past data. Then, for determining the tasks and payments, which should be optimally offered to each user, they formulate non-linear optimization problems. Furthermore, the work in [67] proposes a framework, which probabilistically models the impact of incentives on the users' choice about whether to contribute to a given crowdsensing task. They formulate a convex optimization problem for the allocation of incentives with the objective of maximizing the data quality, while considering constraints such as task budget constraints as well as the locations of the users and the tasks [67]. They use a two-step iterative algorithm where the first step allocates incentives to a set of users and the second step refines the initial allocation of incentives by revoking portions of the initially allocated incentives.

While existing data-driven incentive mechanisms for mobile crowdsensing [66, 67] consider payments to the users as a fundamental way of incentivizing them, an alternative solution is to look at incentives in terms of **value-add** to the users as opposed to that of expected payments/utility. Although such value-add may seem somewhat intangible, the notion of **value** often goes well beyond utility/currency. For example, if a user Alice uses a mobile crowdsensing app for uploading various city events (e.g., traffic congestion, road blockage, broken pavements etc.) to a Cloud-based system, it would add value to her if the system regularly sends her information about the extent of traffic congestion in her daily traffic route from her home to her office. Additionally, the system could provide her with an interface to enable her to issue queries about traffic congestion on other routes as well, thereby adding considerably to her convenience.

In essence, this is about providing value-added services to the users as opposed to providing them with monetary rewards. We could also classify users into multiple levels (e.g., gold, silver, bronze etc.) depending upon the contribution of the user towards providing data to the system. The level of services would increase as a given user's level increases, thereby providing different levels of value-added services. For example, a user at the bronze level would get alerts pertaining only to her regular home-to-office route. On the other hand, a user at the gold level would be able to receive alerts pertaining to any route of her choice. Here, the level of value-added services could also be based on the spatial region under consideration. For example, while a user at the bronze level would receive alerts about traffic congestion only within a radius of 5 km of her house, a user at the gold level would receive such alerts for the region within a radius of 100 km from her house. Observe how this scheme incentivizes users to improve their contributions to the system so that they would be able to obtain better services from the system.

Incidentally, having a fine-grained understanding of a given user's preferences and beliefs can also play a prominent role in creating sustainable incentive models. For example, if a given user has strong views against cars illegally parked on the street or graffiti on public property, it would indeed provide her with satisfaction in sensing and reporting such events to the Cloud-based server, regardless of any tangible financial rewards. Herein lies the importance of **personalization** based on people's belief systems because incentives typically do not follow a "one-size-fits-all" paradigm. Thus, when the incentives align well with a given user's belief system

or causes that she truly cares about, it can be a tremendous value-add from her perspective. In such cases, the users may be moved more by such value-add aspects rather than mere utility or currency.

Interestingly, the concept of marginal significance [68] in economics lends further credence to the idea of rewarding users in a value-based manner as opposed to a currency-based way. Although marginal significance has considerable ramifications for the design of incentive models, existing incentive models mostly fail to take this important concept into consideration. As an example, an incentive model may reward a user with \$0.50 to report a traffic congestion event. However, depending upon the user's level of wealth, \$0.50 may not be adequate to incentivize her to report the event. On the other hand, \$50 would probably serve as an adequate incentive. However, this kind of incentive would be too expensive for the system and it would not be sustainable in the long run. Hence, in such cases, we could emphasize value-based incentives such as incentivizing the social reputation of the contributing users.

In particular, we observe that the problem concerns determining the amount of currency, which would be significant enough (i.e., **marginally significant**) for the user to expend some effort towards providing data to the system. Note that given the bandwidth costs in sending messages, the currency incentives should definitely exceed those costs; the question is "*By how much?*" This is a challenging issue, which is exacerbated by the fact that incomes as well as purchasing power can vary significantly across the user population. To aggravate the issue further, even users having similar purchasing power may mentally assign different value to the currency incentive, depending upon their belief systems, spending habits, socio-cultural expectations and the cost of living in their respective regions.

A solution to this issue could be to determine different levels (ranges) of currency incentives for different regions, depending upon an estimated cost of living in those regions. (Reliable estimates for cost of living data across various cities in the world are publicly available online.) Then using the user's estimated income or purchasing power as a guide, the currency incentive could be set at one of those incentive levels i.e., higher-income individuals get less currency incentives and vice versa. Observe that this requires the system to have some knowledge about a user's income. Such information could be provided to the system by a given user in the form of a range (e.g., annual income of \$40000–\$50000) for privacy reasons as opposed to an exact value of income.

Notably, this solution lacks fairness in the sense that lower-income individuals get more currency incentives for performing comparable crowdsensing tasks as those of higher-income individuals. It is possible that it could result in decreased participation of high-income users in providing data to the system, while increasing the participation of low-income users. However, one also has to note that the marginally significant currency incentive amount, which would entice high-income users to participate, would be much higher than that of low-income individuals. As a consequence, paying those high amounts of currency incentives would not be sustainable for the system in the long-run. However, for high-income individuals,

participation could be increased by adding value (as discussed earlier) instead of trying to entice them with currency-based incentives.

#### **4.5.1 Discussion and Insights**

There are privacy, trust, and security ramifications of letting each application deal with issues such as incentive engineering and data security. Given the smart city context, it can be reasonably expected that a trusted entity should be in control of the infrastructure and the policies associated with the applications. The trusted entity could be the city government, a highly trusted corporate organization with a good track record of CSR (Corporate Social Responsibility) activities or a well-established and highly trusted non-profit organization. As such, the policies associated with the applications should be shaped by social forces in order to address the people-centric nature of smart city applications. Thus, the success of the applications would depend significantly upon the design of appropriate feedback mechanisms for citizens to provide information about their pain points as well as any obstacles that they may be encountering in terms of gaining maximum benefit from the applications.

## **5 Big Data Management and Analytics for IoT Applications**

Big data management and processing on the Cloud has attracted significant attention due to an increasing trend towards performing analytics on the big data for obtaining actionable insights that would benefit existing applications as well as usher in novel next-generation applications. This entails several research challenges such as data filtering, validation and integration, spatial indexing to facilitate location-dependent services, personalizing the analytics results w.r.t. the relevant stakeholders, big data replication and partitioning approaches on the Cloud, security of the big data and uncertainty.

Data filtering, validation and integration become necessary because the data generated from mobile and IoT environments is typically heterogeneous, multi-modal (e.g., text, photos, audio and video), unstructured, noisy and often incomplete. For example, when users take a photo/video using a mobile app before sending it to the system, they often do not add any meta-data/tags. Here, incentives could be provided to mobile users for encouraging them to add tags to the data. Moreover, the unstructured text data sent by mobile users do not generally have any pre-defined format, thereby posing considerable challenges to the interpretation and understanding of the data. A possible solution could be to carefully design a set of absolutely non-overlapping categories (e.g., traffic event, fire event etc.) and then enable the users to select from those categories to establish a context for the text data that they provide.

The reliability of data, which has been collected by means of crowdsourcing, is an important issue. Hence, the fine-grained truth discovery model, designated as *FaitCrowd* [69], aims at aggregating conflicting data that has been collected by means of crowdsourcing. *FaitCrowd* estimates the reliability of the data source by considering the topical expertise of the crowdworkers in a probabilistic manner, thereby reducing the error rates in crowdsourced data aggregation. Furthermore, the work in [70] discusses an approach, designated as *REquEST*, for performing crowdsourced event detection in smart cities in a reliable manner. In particular, the *REquEST* approach selects a small subset of human sensors for performing tasks that require human judgment (e.g., tasks associated with ratings), thereby enabling reliable identification of the crowdsourced events even in the face of noisy data collected from human users as well as the biases of individual users.

Indexing of the crowdsourced data becomes a necessity to reduce query response times. In this regard, the work in [71] studies the fundamental problem of performing tree-based indexing for crowdsourced databases. It also discusses new algorithms for fundamental index operations such as search, insert and update. Moreover, it investigates how different tree and crowdsourcing parameters impact the quality of indexing operations. Incidentally, spatial indexing becomes critical to facilitate effective location-dependent services in urban settings. Spatial indexes, such as the R-tree and its variants and the quad-tree, have been extensively researched. However, a critical open research question concerns making these indexes more context-aware and personalized in terms of incorporating more contextual information corresponding to every stakeholder and its possibly unique requirements.

Given the typically diverse needs of various stakeholders in urban informatics, a possible solution could be to maintain different versions of the same index i.e., one for each stakeholder. However, this solution suffers from the drawback of having too many indexes in silos, thereby resulting in “tunnel vision”. Several indexes in silos become cumbersome to maintain over a period of time, even for low update rates and more so for update-intensive applications. To reduce the number of indexes, we could cluster the stakeholders based on their requirements and then maintain separate indexes i.e., one for every clustered group of stakeholders. However, this solution significantly reduces the possibilities of performing integrated analytics across multiple stakeholders and consequently, across multiple domains. Thus, the open research question here concerns the creation of an integrated and highly update-efficient index, which can effectively address the varying needs of diverse stakeholders in a context-aware and personalized manner.

Incidentally, personalization is critical to performing actionable analytics in urban informatics because stakeholders have varying information needs. For example, a traffic congestion or traffic accident event would be of interest to the traffic police department; a sudden fire event would concern the firefighting department; a toxic garbage dump would be relevant to pollution and healthcare agencies and so on. It is critical not to create information overload for any given stakeholder, but to precisely route the analytics results to a given stakeholder based on its needs. A possible solution would be to effectively capture the needs of various

stakeholders and create a mapping between the data and the relevant stakeholders. Existing classification techniques can be used to create such a mapping, while additionally considering domain-specific constraints. Furthermore, visualization interfaces should be carefully designed to present the analytics results to the relevant stakeholders in a navigation-friendly manner and the results should be prioritized based on the weighted scores of aspects such as importance, deadline constraints and number of citizens impacted.

The widespread use of IoT devices and the real-time availability of user-location information necessitate the development of new personalized, location-based applications and services (LBSs). Such applications require multi-attribute query processing, scalability for supporting millions of users, real-time querying capability and analyzing large volumes of data. Key-value stores were designed to extract value from very large volumes of data while being highly available, fault-tolerant and scalable, thereby providing much needed features to support SBSs with respect to IoT data. More complex queries on multi-dimensional data need to be processed efficiently. Systems such as M-Grid [72], a unifying indexing framework, enable key-value stores to support multi-dimensional queries. It organizes a set of nodes as an overlay network, which provides fault-tolerance and efficient query processing. It uses Hilbert Space Filling Curve based linearization technique to preserve the data locality to efficiently manage multi-dimensional data in a key-value store. It supports queries like range and  $k$ -nearest neighbor ( $k$ NN) on linearized values supporting location-based IoT applications.E

## 6 Knowledge Management for IoT Applications

Knowledge management will play a pivotal role in the success of IoT applications. Semantic Web technologies have gained popularity for knowledge management on the Web and in domain-specific applications. They have the potential to enable data-intensive applications in urban informatics to truly harness the value of mobile and IoT data and gain actionable insights through semantic reasoning. The inherent heterogeneity and scale of data dealt by these applications pose new challenges in data integration and knowledge management. First, there is need for a common vocabulary of terms to describe the data using the Resource Description Framework (RDF) [73]. While several popular vocabularies can be employed (e.g., Dublin Core,<sup>2</sup> Simple Knowledge Organization System (SKOS),<sup>3</sup> DBpedia Ontology<sup>4</sup> [74], Schema.org,<sup>5</sup> Linked Open Vocabularies,<sup>6</sup> Wikidata<sup>7</sup> [75], IoT-Lite Ontology [76]),

---

<sup>2</sup><http://dublincore.org>

<sup>3</sup><https://www.w3.org/2004/02/skos>

<sup>4</sup><http://wiki.dbpedia.org/services-resources/ontology>

<sup>5</sup><http://schema.org>

<sup>6</sup><http://lov.okfn.org/dataset/lov/>

<sup>7</sup><https://www.wikidata.org>



new concepts/properties arising in urban informatics use cases must be considered. Second, there is a need for new RDF data management techniques to achieve efficient query processing, which in turn will enable fast semantic reasoning. Finally, the power of cluster computing must be exploited to cope with high-throughput, massive RDF datasets that can naturally arise in IoT applications.

### 6.1 Background on RDF and SPARQL

RDF is the standard model for data representation on the Web [73]. An RDF statement is a set of terms represented as a (subject, predicate, object) triple. The subject and predicate in a triple are IRIs; the object can be an IRI or a literal. Consider an RDF dataset shown in Fig. 3a. Note that this dataset can be represented as a directed, labeled graph as shown in Fig. 3b.

SPARQL is a popular query language for RDF [77]. A SPARQL query can be posed on the dataset to extract relevant RDF terms via graph pattern matching. The variables in a query (prefixed by “?”) will be bound to RDF terms in the dataset during query processing. For example, the SPARQL query in Fig. 4a finds the IoT devices with temperature higher than 20 °C and their publishers. The output of the query, i.e., bindings for the variables, is shown in Fig. 4b.

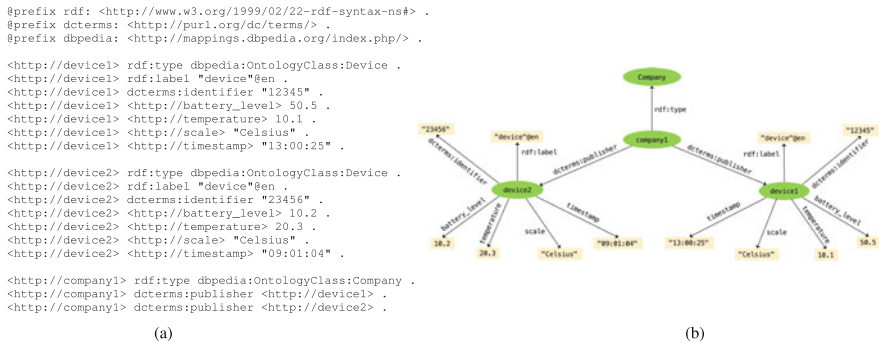


Fig. 3 Example. (a) A dataset containing RDF triples describing IoT devices. (b) A graph representation

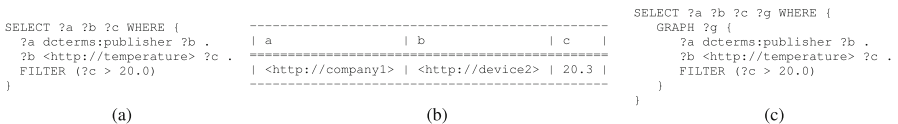


Fig. 4 (a) A SPARQL query on triples. (b) Output of the query in (a). (c) A SPARQL query on named graphs

## 6.2 SPARQL Query Processing on RDF Triples

Most of the techniques developed for RDF query processing have focused on supporting datasets containing triples. While it was popular to store triples in a single relational table and apply self-joins during query processing [78], the cost of self-joins became a serious bottleneck on large RDF datasets. Researchers addressed this performance issue by using a column-oriented DBMS [79] or via exhaustive indexing and the use of merge joins to speed up query processing [80, 81]. Subsequently, techniques were developed to reduce the size of intermediate results during join processing [82] and the size/number of indexes and the overall cost of join processing [83, 84].

More recent approaches have focused on parallel SPARQL query processing in a cluster to deal with even larger RDF datasets [85–89]. At the core of these approaches, lies the concept of partitioning the input RDF dataset (e.g., placement of RDF triples) on a set of cluster machines using techniques like vertex partitioning and graph partitioning. Queries were partitioned when needed and executed in parallel. To ensure the correctness of query processing, data replication was done. This also minimized the overhead of communication during parallel query processing. The techniques differ in their strategy to partition the data and extent of replication to avoid the communication overhead. Different query optimization techniques have been proposed to reduce the size of intermediate results as well as the communication cost. These approaches are briefly discussed next.

Huang et al. [85] developed a parallel SPARQL query processing approach by partitioning graphs on vertices and placing triples on different machines. Using  $n$ -hop replication of triples in partitions, they avoided communication between partitions during query processing. Later, Trinity.RDF [86] was developed, where RDF graphs were stored in a distributed in-memory key-value store. Using graph exploration and optimization techniques, the size of intermediate results was reduced leading to faster query execution. H2RDF+ [87] was proposed later and built eight indexes using HBase.<sup>8</sup> It used Hadoop to perform sort-merge joins during query processing. Another approach called TriAD [88] used asynchronous inter-node communication for scalable SPARQL query processing. It outperformed distributed RDF query engines that rely on Hadoop to perform joins during query processing. DREAM [89] proposed a new paradigm where queried were partitioned instead of data and different number of machines were selected to execute different SPARQL queries based on their complexity. More recently, S2RDF [90] was developed using Apache Spark<sup>9</sup> and employed a new vertical partitioning strategy for storing RDF triples along with query optimization techniques.

---

<sup>8</sup><http://hbase.apache.org>

<sup>9</sup><http://spark.apache.org>

### 6.3 RDF Quads for IoT Applications

An RDF statement of the form (subject, predicate, object, context) is called a quad. Context is useful for modeling provenance information,<sup>10</sup> especially when data integration is performed across multiple data sources. Datasets containing RDF quads are becoming popular on the Web (e.g., Billion Triples Challenge,<sup>11</sup> Bio2RDF<sup>12</sup>). RDF quads can be very powerful in urban informatics applications to capture the original/provenance of the mobile/IoT data. For example, the RDF dataset in Fig. 3a can be represented as quads by adding the context <http://iot.org/missouri> to each triple. The dataset now becomes a named graph, which can be queried using the GRAPH keyword in SPARQL. An example of a query on named graphs is shown in Fig. 4c. The output of this query will contain the bindings for ?a, ?b, ?c, and ?g.

While several approaches have been proposed for query processing on RDF triples, very little research has been done in the context of RDF quads. Given the inherent advantage of modeling provenance information for IoT applications using quads, there is a need to advance the state-of-the-art in query processing over RDF quads. A reification approach can be used to map every quad into four triples and then transforming the SPARQL query on quads to a SPARQL query on triples. The transformed query can be executed using any existing approach for RDF triples. However, the drawback is a four-fold increase in the size of the database, which can lead to poor performance during query processing.

Motivated by these reasons, RIQ was developed for fast query processing on RDF quads in a local environment [91–93]. RIQ employed a *decrease-and-conquer* approach for efficient query processing on over one billion quads. Using a novel filtering index, RIQ pruned away a major portion of the named graphs that do not contain matches for a query early on during query processing. The remaining candidate named graphs were further processed by applying query optimization and rewriting techniques to produce the final output. As IoT applications can potentially generate billions of RDF statements, further research is needed to enable parallel SPARQL query processing on RDF quads/named graphs using a commodity cluster. While it appears that RDF named graphs can be easily partitioned across a cluster of nodes, the key challenge is to group them based on some similarity metric to achieve efficient query processing. To enable parallel processing in a cluster, Apache Spark is a natural choice due to its widespread usage in many data-intensive real-world applications.

---

<sup>10</sup><https://www.w3.org/TR/prov-dm>

<sup>11</sup><http://km.aifb.kit.edu/projects/btc-2012>

<sup>12</sup><http://bio2rdf.org>

## 7 IOT Security and Privacy

Security and privacy of the big data generated by IoT applications in a cloud computing environment remains a major research challenge. IoT uses several devices, cloud, software, different networks, etc. and therefore all the information management, security vulnerabilities and governance become more complicated. Although cloud service providers provide some security, there remains an absence of clear understanding of the extent of security coverage for enterprise applications for IoT data, thereby in effect causing an impediment for organizations about complete adoption of cloud [94]. These security concerns apply to the Cloud-based processing of urban informatics data collected from mobile and IoT environments. For example, enterprise applications associated with various municipal government departments often need to ensure a high extent of security coverage, depending upon the application. Similarly, IoT applications from Healthcare domain need to protect the data privacy of the people involved right from the patients to the healthcare workers as combining IoT data can produce misusable intelligence.

IoT data has to be made available to others through the cloud for efficient dissemination and storage. Many different consumers with varied roles and responsibilities will access IoT data and thus, fine granularity access control over IoT data needs more efficient solutions. Most of IoT data is generated by sensors and cell phones carried by people. Therefore, data provenance is also needed to track the origin of data for supporting critical IoT applications e.g., in transportation and healthcare. IoT's personal and business data stored in the cloud will be passed back and forth through thousands of IoT devices that may have several security vulnerabilities in their applications. Thus, many enterprises may be reluctant to put IoT applications on the cloud. Therefore, risk assessment is needed before off-loading their applications to the cloud and to the mobile devices running those IoT applications.

A possible solution is to use third-party services for performing security risk assessment of IoT applications, but it depends on the expertise of the third-party evaluators. Hence, there exists the critical need for an offline risk assessment framework to facilitate enterprises towards selecting suitable secure cloud service providers based on the enterprise applications' security requirements and a cost-benefit (cost of off-loading vs. the security benefits) tradeoff analysis [94] for also offloading those application components to IoT devices.

Ensuring user privacy while collecting mobile crowdsourced data is critical in ensuring user cooperation in providing data. Hence, the work in [95] discusses a crowdsourcing approach, which performs data quality control while preserving the privacy of the crowdworkers. They note that it is possible to infer personal information about a given user based on the results provided by him. Thus, they propose a worker-private latent class protocol that preserves crowdworkers' privacy, while enabling requestors to infer high-quality results from the data provided by the crowdworkers. This is achieved by decentralizing the computation as well as performing the computation in a secure manner.

Observe that user privacy not only concerns location privacy, but also several other important attributes (e.g., user preferences, user history of interactions etc.) that relate to the context of the user in general. In this regard, several efforts have concerned  $k$ -anonymity and location privacy. However, given the increasing prevalence of social media applications on mobile devices, the issue of obtaining a fine-grained understanding of the context of a given user remains an open question. A possible solution to this issue is to semantically understand a given user's context based on her social groups and interactions thereof. This could add significant value towards recommendation applications in urban informatics by incorporating missing data about user preferences using techniques such as association rule mining and matrix factorization.

## 8 Conclusion

While mobile computing, IoT, and big data are enabling technologies for advancing the state-of-the-art in urban informatics, new challenges are posed by the diversity, uncertainty, and scale of mobile and IoT data. Moreover, the people-centric nature of urban informatics applications raises non-trivial challenges in the collection, management, security and privacy, analysis and reasoning over such data. Incentives and privacy play an important role in the crowdsourcing way of data collection in many applications supporting urban informatics. Crowdsourced data integrated with cloud and IoT environment can provide a wide variety of data for performing meaningful analytics to support applications in many different domains. Knowledge management is another important issue for IoT applications, and existing standards for the Semantic Web can be employed for IoT data. We presented possible research directions that should be explored and discussed in future research supported by cloud computing, big data, and IoT.

## References

1. S. E. Koonin, "Urban informatics: Putting big data to work in our cities [online]," <http://data-informed.com/urban-informatics-putting-big-data-to-work-in-our-cities/>, 2016.
2. "NYU Center for Urban Science and Progress [online]," <http://cusp.nyu.edu/urban-informatics/>.
3. "Beyond smart cities - people first approach [online]," <http://www.urbaninformatics.net/>.
4. M. Foth, J. H. Choi, and C. Satchell, "Urban informatics," in *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, 2011, pp. 1–8.
5. "Wikipedia article on Urban Computing [online]," [https://en.wikipedia.org/wiki/Urban\\_computing](https://en.wikipedia.org/wiki/Urban_computing).
6. "McKinsey Report [online]," <http://mckinseysociety.com/emerging-trends-in-urban-informatics/>.
7. A. Mondal, P. Rao, and S. K. Madria, "Mobile computing, internet of things, and big data for urban informatics," in *International Conference on Mobile Data Management (MDM)*, vol. 2, 2016, pp. 8–11.

8. N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, "Visual exploration of big spatio-temporal urban data: A study of New York City taxi trips," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, Dec. 2013.
9. J. He, K. Kunze, C. Lofi, S. K. Madria, and S. Sigg, "Towards mobile sensor-aware crowdsourcing: Architecture, opportunities and challenges," in *Proc. DASFAA Workshops*, 2014, pp. 403–412.
10. D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012, pp. 173–184.
11. A. Jian, G. Xiaolin, Y. Jianwei, S. Yu, and H. Xin, "Mobile crowd sensing for Internet of Things: A credible crowdsourcing model in mobile-sense service," in *Proceedings of the IEEE International Conference on Multimedia Big Data*, 2015, pp. 92–99.
12. J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz, J. A. Galache, M. Presser, L. A. H. Gómez, and J. Pettersson, "The Future Internet," 2011, ch. Smart Cities at the Forefront of the Future Internet, pp. 447–462.
13. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.
14. S. Zygiaris, "Smart city reference model: Assisting planners to conceptualize the building of smart city innovation ecosystems," *Journal of the Knowledge Economy*, vol. 4, no. 2, pp. 217–231, Jun 2013.
15. L. Filippini, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, "Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors," in *Fourth International Conference on Sensor Technologies and Applications*, 2010, pp. 281–286.
16. A. Attwood, M. Merabti, P. Fergus, and O. Abuelmaatti, "SCCIR: Smart cities critical infrastructure response framework," *Proceedings of 4th International Conference on Developments in eSystems Engineering*, 2011.
17. N. Zygouras, N. Zacheilas, V. Kalogeraki, D. Kinane, and D. Gunopulos, "Insights on a scalable and dynamic traffic management system," *Proceedings of EDBT*, 2015.
18. T. Mukherjee, D. Chander, A. Mondal, K. Dasgupta, A. Kumar, and A. Venkat, "CityZen: A cost-effective city management system with incentive-driven resident engagement," in *IEEE 15th International Conference on Mobile Data Management, MDM*, 2014, pp. 289–296.
19. S. Basu Roy, I. Lykourantzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das, "Task assignment optimization in knowledge-intensive crowdsourcing," *The VLDB Journal*, vol. 24, no. 4, pp. 467–491, Aug. 2015.
20. N. Panagiotou, N. Zygouras, I. Katakis, D. Gunopulos, N. Zacheilas, I. Boutsis, V. Kalogeraki, S. Lynch, and B. O'Brien, "Intelligent urban data monitoring for smart cities," *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pp. 177–192, 2016.
21. G. Suci, A. Vulpe, S. Halunga, O. Fratu, G. Todoran, and V. Suci, "Smart cities built on resilient Cloud Computing and secure Internet of Things," *Proceedings of the International Conference on Control Systems and Computer Science*, pp. 513–518, 2013.
22. W. M. da Silva, A. Alvaro, G. H. R. P. Tomas, R. A. Afonso, K. L. Dias, and V. C. Garcia, "Smart cities software architectures: A survey," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13, 2013, pp. 1722–1727.
23. E. M. Daly, F. Lecue, and V. Bicer, "Westland row why so slow?: Fusing social media and linked data sources for understanding real-time traffic conditions," in *Proceedings of the ACM International Conference on Intelligent User Interfaces*, 2013, pp. 203–212.
24. H. Khazaei, S. Zareian, R. Veleda, and M. Litoiu, "Sipresk: A big data analytic platform for smart transportation," *First EAI International Summit*, pp. 419–430, 2016.
25. "Ushahidi Platform [online]," <https://blog.ushahidi.com/2012/06/05/ushahidi-beijing/>.
26. "Fixmystreet platform [online]," <https://www.fixmystreet.com/>.
27. P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, pp. 323–336.

28. M. Jain, A. P. Singh, S. Bali, and S. Kaul, "Speed-breaker early warning system." in *Proc. USENIX/ACM Workshop on Networked System for Developing Regions*, 2012.
29. Y.-c. Tai, C.-w. Chan, and J. Y.-j. Hsu, "Automatic road anomaly detection using smart mobile device," in *conference on technologies and applications of artificial intelligence*, 2010.
30. A. Mondal, A. Sharma, K. Yadav, A. Tripathi, A. Singh, and N. M. Piratla, "RoadEye: A system for personalized retrieval of dynamic road conditions," in *IEEE 15th International Conference on Mobile Data Management, MDM*, 2014, pp. 297–304.
31. "Waze traffic navigation app [online]," <https://www.waze.com/en-GB/>.
32. A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. Koutsopoulos, and C. Moran, "IBM Infosphere Streams for scalable, real-time, intelligent transportation services," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2010, pp. 1093–1104.
33. S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: drive-by sensing of road-side parking statistics," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010, pp. 123–136.
34. A. Guttman, *R-trees: A dynamic index structure for spatial searching*, 1984, vol. 14.
35. N. Zygouras, N. Panagiotou, N. Zacheilas, I. Boutsis, V. Kalogeraki, I. Katakis, and D. Gunopulos, "Towards detection of faulty traffic sensors in real-time," *CEUR Workshop Proceedings*, vol. 1392, 2015.
36. "Smart waste management [online]," <http://www.link-labs.com/smart-waste-management/>.
37. A. Rovetta, F. Xiumin, F. Vicentini, Z. Minghua, A. Giusti, and H. Qichang, "Early detection and evaluation of waste through sensorized containers for a collection monitoring application," *Waste Management*, vol. 29, no. 12, pp. 2939–2949, 2009.
38. F. Vicentini, A. Giusti, A. Rovetta, X. Fan, Q. He, M. Zhu, and B. Liu, "Sensorized waste collection container for content estimation and collection optimization," *Waste Management*, vol. 29, no. 5, pp. 1467–1472, 2009.
39. M. A. A. Mamun, M. A. Hannan, A. Hussain, and H. Basri, "Wireless sensor network prototype for solid waste bin monitoring with energy efficient sensing algorithm," *Proceedings of the International Conference on Computational Science and Engineering*, pp. 382–387, 2013.
40. A. Papalambrou, D. Karadimas, J. Gialelis, and A. G. Voyiatzis, "A versatile scalable smart waste-bin system based on resource-limited embedded devices," *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, 2015.
41. "Waste management using Enevo [online]," <https://www.enevo.com/>.
42. "Waste management using SmartBin [online]," <https://www.smartbin.com/>.
43. D. Karadimas, A. Papalambrou, J. Gialelis, and S. Koubias, "An integrated node for smart-city applications based on active RFID tags; use case on waste-bins," *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–7, 2016.
44. A. Medvedev, P. Fedchenkov, A. Zaslavsky, T. Anagnostopoulos, and S. Khoruzhnikov, "Waste management as an iot-enabled service in smart cities," *Proceedings of the Internet of Things, Smart Spaces, and Next Generation Networks and Systems, ruSMART*, pp. 104–115, 2015.
45. T. Anagnostopoulos, A. Zaslavsky, K. Kolomvatsos, A. Medvedev, P. Amirin, J. Morley, and S. Hadjieftymiades, "Challenges and opportunities of waste management in iot-enabled smart cities: A survey," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 3, pp. 275–289, 2017.
46. M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, 2009, pp. 55–68.
47. P. Shankara, P. Mahanta, E. Arora, and G. Srinivasamurthy, "Impact of internet of things in the retail industry," in *Proceedings of On the Move to Meaningful Internet Systems: OTM Workshops*, 2015, pp. 61–65.

48. S. Fosso Wamba, L. A. Lefebvre, Y. Bendavid, and I. Lefebvre, "Exploring the impact of RFID technology and the EPC network on mobile B2B eCommerce: A case study in the retail industry," in *International Journal of Production Economics*, vol. 112, 2008, pp. 614–629.
49. H. Belarbi, A. Tajmouati, H. Bennis, and M. El Haj Tirari, "Predictive analysis of Big Data in Retail industry," in *Proceedings of the International Conference on Computing Wireless and Communication Systems*, 2016.
50. R. Vargheese and H. Dahir, "An IoT/IoE enabled architecture framework for precision on shelf availability: Enhancing proactive shopper experience," in *Proceedings of the IEEE International Conference on Big Data*, 2014, pp. 21–26.
51. D. Hicks, K. Mannix, H. M. Bowles, and B. J. Gao, "SmartMart: IoT-based in-store mapping for mobile devices," in *Proceedings of the IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2013, pp. 616–621.
52. T. Bohnenberger, A. Jameson, A. Krüger, and A. Butz, "Location-aware shopping assistance: Evaluation of a decision-theoretic approach," in *Proceedings of Human Computer Interaction with Mobile Devices*, 2002, pp. 155–169.
53. M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 337–350.
54. N. Do, C.-H. Hsu, and N. Venkatasubramanian, "CrowdMAC: a crowdsourcing system for mobile access," in *Proceedings of the 13th International Middleware Conference*, 2012, pp. 1–20.
55. C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015, pp. 183–196.
56. H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2016, pp. 344–353.
57. T. Dimitriou and I. Krontiris, "Privacy-respecting auctions as incentive mechanisms in mobile crowd sensing," in *IFIP International Conference on Information Security Theory and Practice*, 2015, pp. 20–35.
58. Q. Li and G. Cao, "Providing privacy-aware incentives in mobile sensing systems," *IEEE Transactions on Mobile Computing*, vol. 15, pp. 1485–1498, 2016.
59. H. Meka, S. K. Madria, and M. Linderman, "Incentive based approach to find selfish nodes in mobile p2p networks," in *Proceedings of the IEEE Performance Computing and Communications Conference (IPCCC)*, 2012, pp. 352–359.
60. T. K. Wijaya, M. Vasirani, and K. Aberer, "Crowdsourcing behavioral incentives for pervasive demand response," Tech. Rep., 2014.
61. A. Mondal, S. K. Madria, and M. Kitsuregawa, "E-arl: An economic incentive scheme for adaptive revenue-load-based dynamic replication of data in mobile-p2p networks," *Distributed and Parallel Databases*, vol. 28, pp. 1–31, 2010.
62. B. Fogg, "A behavior model for persuasive design," in *Proceedings of the 4th International Conference on Persuasive Technology*. ACM, 2009, pp. 40:1–40:7.
63. L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 1701–1709.
64. D. Easley and A. Ghosh, "Behavioral mechanism design: Optimal crowdsourcing contracts and prospect theory," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 2015, pp. 679–696.
65. C. B. Ferster and B. F. Skinner, "Schedules of reinforcement," *Appleton-Century-Crofts*, 1957.
66. M. Karaliopoulos, I. Koutsopoulos, and M. Titsias, "First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2016, pp. 271–280.
67. P. Micholia, M. Karaliopoulos, and I. Koutsopoulos, "Mobile crowdsensing incentives under participation uncertainty," in *Proceedings of the 3rd ACM Workshop on Mobile Sensing, Computing and Communication*, 2016, pp. 29–34.



68. L. Pritschet, D. Powell, and Z. Horne, "Marginally significant effects as evidence for hypotheses: Changing attitudes over four decades," *Psychological Science*, vol. 27, no. 7, pp. 1036–1042, 2016.
69. F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han, "Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 745–754.
70. I. Boutsis, V. Kalogeraki, and D. Guno, "Reliable crowdsourced event detection in smartcities," in *1st International Workshop on Science of Smart City Operations and Platforms Engineering (SCOPE) in partnership with Global City Teams Challenge (GCTC) (SCOPE - GCTC)*, 2016, pp. 1–6.
71. A. Mahmood, W. G. Aref, E. Dragut, and S. Basalamah, "The Palm-tree index: Indexing with the crowd," *Proc. DBCrowd*, pp. 26–31, 2013.
72. S. Kumar, S. Madria, and M. Linderman, "M-Grid: a distributed framework for multidimensional indexing and querying of location based data," *Distributed and Parallel Databases*, vol. 35, pp. 55–81, 2017.
73. "Resource Description Framework," <http://www.w3.org/RDF>.
74. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - a crystallization point for the Web of data," *Journal of Web Semantics*, vol. 7, no. 3, pp. 154–165, September 2009.
75. D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Comm. of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
76. M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "IoT-Lite: A Lightweight Semantic Model for the Internet of Things," in *2016 International IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, July 2016, pp. 90–97.
77. "SPARQL 1.1," <http://www.w3.org/TR/sparql11-query/>.
78. E. I. Chong, S. Das, G. Eadon, and J. Srinivasan, "An efficient SQL-based RDF querying scheme," in *Proc. of the 31st VLDB Conference*, 2005, pp. 1216–1227.
79. D. Abadi, A. Marcus, S. Madden, and K. Hollenbach, "SW-Store: A vertically partitioned DBMS for semantic web data management," *The VLDB Journal*, vol. 18, no. 2, pp. 385–406, 2009.
80. T. Neumann and G. Weikum, "The RDF-3X engine for scalable management of RDF data," *The VLDB Journal*, vol. 19, no. 1, pp. 91–113, 2010.
81. C. Weiss, P. Karras, and A. Bernstein, "Hexastore: Sextuple indexing for Semantic Web data management," *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 1008–1019, 2008.
82. M. Atre, V. Chaoji, M. J. Zaki, and J. A. Hendler, "Matrix "Bit" loaded: A scalable lightweight join query processor for RDF data," in *Proc. of the 19th WWW Conference*, 2010, pp. 41–50.
83. M. A. Bornea, J. Dolby, A. Kementsietsidis, K. Srinivas, P. Dantressangle, O. Udrea, and B. Bhattacharjee, "Building an efficient RDF store over a relational database," in *Proc. of 2013 SIGMOD Conference*, 2013, pp. 121–132.
84. P. Yuan, P. Liu, B. Wu, H. Jin, W. Zhang, and L. Liu, "TripleBit: A fast and compact system for large scale RDF data," *Proc. VLDB Endow.*, vol. 6, no. 7, pp. 517–528, 2013.
85. J. Huang, D. J. Abadi, and K. Ren, "Scalable SPARQL querying of large RDF graphs," *Proc. of VLDB Endow.*, vol. 4, no. 11, pp. 1123–1134, 2011.
86. K. Zeng, J. Yang, H. Wang, B. Shao, and Z. Wang, "A distributed graph engine for web scale RDF data," *Proc. VLDB Endow.*, vol. 6, no. 4, pp. 265–276, 2013.
87. N. Papailiou, D. Tsoumakos, I. Konstantinou, P. Karras, and N. Koziris, "H2RDF+: An Efficient Data Management System for Big RDF Graphs," in *Proc. of the 2014 ACM SIGMOD Conference*, Snowbird, Utah, USA, 2014, pp. 909–912.
88. S. Gurajada, S. Seufert, I. Miliaraki, and M. Theobald, "TriAD: A Distributed Shared-nothing RDF Engine Based on Asynchronous Message Passing," in *Proc. of the 2014 ACM SIGMOD Conference*, Snowbird, Utah, USA, 2014, pp. 289–300.

89. M. Hammoud, D. A. Rabbou, R. Nouri, S.-M.-R. Beheshti, and S. Sakr, "DREAM: Distributed RDF Engine with Adaptive Query Planner and Minimal Communication," *Proc. VLDB Endow.*, vol. 8, no. 6, pp. 654–665, Feb. 2015.
90. A. Schätzle, M. Przyjaciel-Zablocki, S. Skilevic, and G. Lausen, "S2RDF: RDF Querying with SPARQL on Spark," *Proc. VLDB Endow.*, vol. 9, no. 10, pp. 804–815, Jun. 2016.
91. V. Slavov, A. Katib, P. Rao, S. Paturi, and D. Barenkala, "Fast processing of SPARQL queries on RDF quadruples," in *Proc. of WebDB '14*, 2014, pp. 1–6, <https://arxiv.org/pdf/1506.01333v1.pdf>.
92. A. Katib, V. Slavov, and P. Rao, "RIQ: Fast processing of SPARQL queries on RDF quadruples," *Journal of Web Semantics*, vol. 37, no. C, pp. 90–111, 2016.
93. A. Katib, P. Rao, and V. Slavov, "A tool for efficiently processing SPARQL queries on RDF quads," in *Proc. of the 16th International Semantic Web Conference (ISWC 2017)*, Vienna, Austria, Oct. 2017, pp. 1–4, <http://ceur-ws.org/Vol-1963/paper472.pdf>.
94. S. K. Madria, "Security and risk assessment in the Cloud," *IEEE Communications Magazine*, 2016.
95. H. Kajino, H. Arai, and H. Kashima, "Preserving worker privacy in crowdsourcing," *Data Mining and Knowledge Discovery*, vol. 28, pp. 1314–1335, 2014.

# 5G Wireless Micro Operators for Integrated Casinos and Entertainment in Smart Cities



Da-Yin Liao and XueHong Wang

**Abstract** Smart cities aim to improve the quality of citizen's life by infusing technology into every part of operations in the cities. The next generation mobile network, 5G, is considered as a potentially key driver for the emerging global IoT to support smart cities where the various indoor/small cell operations create a new 5G business model—the Micro Operators. This chapter deals with the design and applications of future 5G wireless micro operators for integrated casinos and entertainment (5G ICEMO) in smart cities. We first propose a Concentric Value Circles (CVC) model for analysis of 5G ICEMO and develop the business model. A 5G Cloud-enabled ICEMO system and wireless network architectures are developed. Three illustrating cases of mega jackpot, anti-counterfeiting lottery, and autonomous transport are studied to convey the proposed 5G ICEMO architecture. Benchmarking among Las Vegas, Macao, and Singapore is analyzed to validate how the proposed 5G micro operator framework can be exploited to integrated casinos and entertainment in smart cities.

**Keywords** Micro operator · 5G · Next generation wireless · Integrated casinos and entertainment · Smart cities · IoT · Cloud RAN · Concentric value circles · Small cells · Mobile edge computing

## 1 Introduction

The drastic growth in urbanization over the world requires sustainable, efficient, and smart solutions for governance, transportation, environment, and quality of life. *Smart Cities* solutions use networking and integrated management of information

---

D.-Y. Liao (✉)

Straight & Up Intelligent Innovations Group Co., San Jose, CA, USA

e-mail: [eliao@necksoft.com](mailto:eliao@necksoft.com)

X. Wang

Center of Lottery Studies in China, Peking University, Beijing, China

© Springer Nature Switzerland AG 2018

M. Maheswaran, E. Badidi (eds.), *Handbook of Smart Cities*,

[https://doi.org/10.1007/978-3-319-97271-8\\_5](https://doi.org/10.1007/978-3-319-97271-8_5)

to create values for dealing with the problems precipitated by urbanization and growing population. Smart cities aim to improve the quality of life for their citizens by infusing technology into every part of its operations and optimizing the efficiency of services for citizens to meet their changing needs for smarter living. The Smart Cities paradigm is a vision for future cities centered around the concept of connectivity, with tight integration among citizens, devices and service providers. Smart Cities are based on a strong, reliable communication network for applications and services [46]. Smart Cities will increase efficiency, productivity, ecological awareness; and it will reduce pollution and improve quality of life. Smart cities integrate the whole range of services a city needs and wants to offer in a way that follows state-of-the-art public administration requirements, including the use of the most recent technology. It is a new type of city development based on in-depth exploration and wide applications of new generation of information and communications technologies (ICT). Future smart cities will be mainly based on offered services. In order to enable these services, the communication networks in smart cities should embrace the concepts of broadband wireless, green and energy saving, reconfigurability, replication, machine-to-machine communications, and quality of experience.

The Internet of Things (IoT) is expected to substantially support sustainable development of future smart cities. With the interconnection of massive electronic devices, as well as of *anything* belonging to people's environments such as homes, offices, and cities, the IoT paradigm gives the promise to revolutionize the way we live and work by means of a wealth of new services, based on seamless interactions between a large amount of heterogeneous devices. The vision of IoT is an integral part of the Internet, where the objects of everyday life will be able to communicate with each other and with the users. An urban IoT is a communication infrastructure that provides unified, simple, and economical access to a plethora of public services, thus unleashing potential synergies and increasing transparency to the citizens [58]. The most relevant technical issue of smart cities consists in the interoperability of the heterogeneous technologies currently used in city and urban development. The urban IoT becomes the building block to realized a unified urban-scale ICT platform, thus unleashing the potential of smart cities [28, 36]. On the other hand, the explosion in devices will pose serious energy consumption concerns. Thus an urgent need for energy management for IoT devices has emerged so that the concept of smart cities can be realized in a sustained manner. Energy-efficient solutions of IoT-enabled Smart Cities include *lightweight protocols*, *scheduling optimization*, *predictive models for energy consumption*, *cloud-based approach*, *low-power transceivers*, and *cognitive management framework* [21].

The advent of next generation mobile networks, a.k.a. the fifth generation (5G) wireless systems, with the availability of a connectivity technology which is at once truly ubiquitous, reliable, scalable, and cost-efficient, is considered as a potentially key driver for the emerging global IoT. By offering low cost, low energy consumption and supports for very large number of connecting devices, 5G is ready to enable the vision of a truly global IoT. However, due to the limited computation and storage on IoT devices, moving the computing and storage functions to the

cloud, i.e., cloud computing, is considered as a promising paradigm to provide elastic resources to application on those devices. In spite of attempts of augmenting IoT applications with the power of cloud, there are still problems unsolved in that IoT applications usually require mobility support, geo-distribution, location-awareness and low latency. One solution is fog computing which is an extension of the cloud computing paradigm from the core of network to the edge of the network. Fog computing is a highly virtualized platform that provides computation, storage, and network services between end devices and traditional cloud servers [10]. Other similar concepts are mobile cloud computing (MCC) and mobile edge computing (MEC)—MCC refers to the moving of the computing power and storage from mobile phones to the cloud, providing applications and mobile computing to not only mobile phone users but also a much broader range of mobile subscribers; while MEC adopts a cloud server running at the edge of a mobile network and performs specific tasks that could not be accomplished with traditional network infrastructure.

In addition to the concerns of unacceptable latency due to congestion and overloading of radio access and core networks, many machines or devices are geographically located in a very confined and coverage-limited area which might be located in challenging positions like indoors or at the cell-edge. A role of cellular IoT provider is needed to provide reliable and accountable connectivity and dependent on the coverage and rollout strategy of a telecomm operator. This is an important added value to both consumers and business relying on IoT technology [40]. To meet the technical and economical requirements for exponentially growing machine-type communications (MTC) traffics, the concept of small cells is emphasized to handle the massive and dense MTC rollout. The adoption of 5G would slowly shift from providing data pipes to rather controlling an ensemble of data pipes. There would be an important shift for operators from business-to-consumer (B2C) driven business to rather business-to-business (B2B) driven one, which is an important opportunity to scale sales as cellular servicing is extended to any available wireless system. Another big changes and challenges to the success of 5G for global IoT are business models related. Due to various and different applications and deployment of small cells and enormous amount of MTC devices, no one-size-fits-all 5G solutions can be expected.

The development towards 5G mobile communications demands small cells and indoors coverages as well as local services in different public, commercial, industrial premises or spaces. Small cells are able to fulfill Smart Cities requirements in terms of interoperability, robustness, limited power consumption and multi-modal access with improved quality of experience [18]. Different from existing 3G and 4G network services provided by telecom giants, the various indoor/small cell operations create a new 5G business model—the *Micro Operators* ( $\mu$ O), featuring by (1) provisioning of mobile connectivity combined and locked with specific, local services, (2) being spatially confined to either its premises or to defined area of operations, and (3) being dependent on appropriate available spectrum resources [2]. The emerging business opportunities of  $\mu$ O's have to serve economic-scaled users with a specific and necessary purpose. Potential applications of micro operators include universities, hospitals, factories, shopping malls, and casinos and

entertainment. Despite some discussions on 5G use cases can be found in the literature, to our best knowledge, business models and their use case scenarios of  $\mu$ Os for casinos and entertainment are still a neglected one.

In addition to the popular attractions of casinos in Las Vegas and Atlantic City, for the past two decades, gaming industry have become a key industry in some cities like Macao and Singapore, and many geographical areas over the world. Gaming is a strong engine to the economic growth. In United States, gaming industry generates \$240 billion annually in total economic impact, as much as the total state budgets combining New York and Texas [4]. Gaming increases employment, grows local retail sales, and generates great tax revenues to both state and local governments. Gaming boosts industries across all sectors, from lodging, food & beverage, entertainments, transportation, banking, to government and healthcare. Casinos employ more people than the airlines industry and support more than 1.7 million jobs throughout the United States.

Gaming industry is changing profoundly, creating major new sources of revenue and value beyond casino gambling and lodging. Modern gaming industry has evolved into a multi-functioned place combining with resort hotels, restaurants, retail shopping, theaters, spa/fashion/beauty, tourist attractions, live entertainment, sports and racing events, convention and exhibition, and theme parks. An integrated city of casinos and entertainment projects a city of progress and people with a spirit to grow. It is a place for fun, more than just for gambling. In the city, people enjoy relaxation, happiness, family travel, social interaction, and surprises. An Integrated Casinos and Entertainment (ICE) city is a Smart City that demands smart governance, clean and smart environment, smart transportation, smart IT (Information Technology) & communications, smart health, smart education, and smart buildings.

Nowhere is the expectation to change greater among customers than in casinos and entertainment. Driven by mobile and intelligent technologies, gaming industry is entering an era of unprecedented change, demanding innovation and creativity across all their services. The ICE that is able to optimize their physical and cyber operations will win the game. With excellence in ubiquitous customer engagement, strategic demand satisfaction, and seamless operations effectiveness, the ICE can tackle challenges and stay ahead of competition. It is an opportunity for gaming industry to advance their mobile and intelligent ambitions with 5G as a catalyst for innovation and creativity. Potential benefits of 5G technology for ICE include running casinos live and simple, improved control and decision-making based on live insights, driving profitability with clear perspectives.

This chapter deals with the design and applications of future 5G micro operators for integrated casinos and entertainment (ICE) in smart cities. We first outlook the requirements and opportunities of future 5G ICE and propose a Concentric Value Circles (CVC) Model for these business opportunities to ICE in future smart cities, based on which the 5G ICEMO business model is developed. We analyze the critical considerations in building the 5G *Integrated Casinos-and-Entertainment Micro Operators* (ICEMO) platform, including integrated cloud/fog computing and analytics, cash management, ticketing, access control, security handling and sensor

fusion of casino operations, wearables, and other advanced technologies such as virtual reality, robots, autonomous vehicles, and droned devices. The applications and functions of future 5G ICEMO impose challenging technology needs to support efficiently a heterogeneous set of services in ultra-dense networking environments.

We adopt two technology trends of network function virtualization (NFV) [35] and software defined networking (SDN) technologies in design and development of the 5G micro operator network services. We propose a SDN-based architectural framework to fulfill functional and performance requirements of 5G ICEMO in smart cities. Key features of the proposed framework include accuracy, scalability, openness, and transparency. The NFV rapidly introduces targeted and tailored services based on customer needs in the smart cities. The open interface and programmability of network elements in the SDN architecture facilitate the decoupling of the network intelligence to separate software-based controllers in diverse devices and machines used in daily ICE operations. Three illustrating case studies of 5G ICEMO applications—mega jackpot, anti-counterfeiting lottery, and autonomous transport, are discussed. Benchmarks on three casino cities of Las Vegas, Macao, and Singapore are conducted to validate the feasibility and effectiveness of the proposed 5G micro operator framework for Smart Cities.

The remaining of this chapter is organized as below. In Sect. 2, we overview the technologies of smart-cities solutions and services, and review the 5G wireless initiatives and their recent advancements, including the micro operator concept. Sect. 3 first reviews the evolutions of small cells and the cellular RAN architecture and then discusses the need of different Micro Operator ( $\mu$ O) roles in future 5G deployment. The requirements, visions, and applications to integrated casinos and entertainment in smart cities are presented and modeled in Sect. 4. Section 5 proposes the design and architecture of the SDN-based framework for 5G micro operators to fulfill both vertical and horizontal functions of integrated casinos and entertainment in smart cities. Sect. 6 presents three illustrating cases studies for 5G ICEMO. Benchmarks of Las Vegas, Macao, and Singapore are analyzed to validate how the proposed 5G micro operator framework can be exploited to integrated casinos and entertainment smart cities in Sect. 7. Finally, Sect. 8 provides the conclusions to the paper, highlighting the foreseen benefits of 5G ICEMO and identifying several open issues.

## 2 Literature Survey

Smart City has been a buzzword for a lot of years and is sometime an alternative or a synonymy by term of “digital city”, “wireless city”, “cyber city”, or “future city”, from the perspective of technologies and components in different literatures and markets [47]. The concept of smart cities arose in late 1990s and was adopted lately by technology companies for the application of computers and information systems to integrate the operation of urban infrastructure and services of buildings, transportation, utility, and public safety. The global smart cities market size is

projected to grow from 424 billion dollars in 2017 to 1202 billion dollars by 2022 [33], at a compound annual growth rate of 23.1%. The market springs from the synergic interconnection of key service sectors, such as smart living, smart governance, smart mobility, smart society, and smart environment [3].

While Las Vegas is among 78 applicants for the designated US Smart City [54], Singapore is striving to be the first smart city or smart nation in the world [12] and Macao is developing into a world tourism and leisure city, which is livable, safe, healthy, smart, culturally enriched and under good governance [25]. Las Vegas is no stranger to rapid change and innovation. The city is leading not only in business and entertainment, but in pioneering the use of smart city technologies to enhance city operations, particularly with regard to traffic, safety and efficiency. Singapore is taking the smart city to a whole new level, a Smart Nation. Government-deployed sensors will collect and coordinate an unprecedented amount of data on daily life in the city. The main issue with Smart Nation is that there may be too much government control over it right now for real innovation to take place. Poor IT infrastructure is a major constraint on smart city development in Macao. While the rest of the world is developing 5G mobile networks, the city's 4G mobile networks, launched in 2015, are still not widely used. All the focal points of the smart cities development in Las Vegas, Singapore, and Macao are on the investments in new infrastructures. As both technological change and changes in meanings are radical [55], acceptance of new technologies and practices for these cities can be troublesome if lack of solid business models to meet the expectation of change from citizens, customers of casinos and entertainment, business, and government. This chapter develops a novel business model for smart cities with a dominant paradigm of integrated casinos and entertainment.

Smart cities bring together mixed traffic of machines and humans, generated by various citywide infrastructures and hybrid networks, and thus introduce plethora of opportunities as well as challenges [19]. The next-generation wireless networks, 5G, are mission-critical, hybrid networks that connect machines and humans to provide large portfolio of applications and services through reliable, ultra-low latency and broadband communications. 5G demands quicker data speeds, higher throughputs, and more reliable service. Though 5G is still in the planning stages, the first commercial 5G networks are expected to operate by the early 2020s. The development of 5G is not an incremental advance on 4G. Indeed, 5G will need to be a paradigm shift that includes very high carrier frequencies with massive bandwidths, extreme dense base stations and devices, and unprecedented numbers of antennas [5, 7]. While it is not yet clear what technologies will do the most for 5G in the future, a few favorites have emerged, including millimeter waves (mmWaves), small cells, massive multi-input multi-output (MIMO), full duplex, and beamforming [1, 6, 9, 14, 16, 24, 30, 43, 44, 48, 59].

Smart cities scenarios pose several challenges in the management of network resources. A smart city environment is composed of various systems operated by multiple tenants and under heterogeneous scenarios where different types of devices and services co-exist in heterogeneous deployments and have heterogeneous traffic patterns. Such heterogeneity in smart cities requires quick reconfiguration of



network parameters and deployment to cope with disruptive needs of the networks. The design of 5G networks are featured by cognition and programmability through technologies of Network Function Virtualization (NFV) and Software-Defined Networking (SDN), and cloud and edge computing paradigms of the end-to-end chain of the heterogeneity of services, devices and access networks [27, 52]. The architectural flexibility from NFV and SDN enables multiple tenants to share a common physical infrastructure. Smart cities scenarios can benefit significantly from multi-tenant design.

The technology of software-defined networking composes of the decoupling of the control plane and data plane, a programmable network and virtualization, which enables network infrastructure sharing and the software-defined network functions. SDN is a centralized paradigm in which network intelligence or control plane is moved up to a logically centralized SDN controller. The data plane consists of simple forwarding devices that are controlled by the SDN controller through programmable interfaces [38]. NFV is a complementary technology of SDN, which allows to build a virtual-based, end-to-end network infrastructure and to enable the consolidation of many heterogeneous network devices onto industry standard high-volume services, switches and storage [20, 57]. NFV can potentially reduce capital investment and energy consumption by consolidating networking appliances, decrease the time to market of a new service by changing the typical innovation cycle to network operators through software-based service deployment, and rapidly introduce targeted and tailored services based on customer needs [26].

To achieve the goals of high spectral and energy efficiency in 5G systems, radio access networks (RAN) should evolve with advanced radio access technologies and all-Internet Protocol (IP) open Internet network architectures. A significant and advanced baseband computation is required to meet the complex requirements of new solutions such as large-scale cooperative signal processing in physical layer to enable the ultra-dense, heterogeneous nodes to work efficiently [31]. To overcome the above challenges, cloud radio access networks (C-RAN) have shown as an evolved system paradigm by both the operators and equipment vendors [42]. C-RAN is a breakthrough of emerging technologies to improve both spectral and energy efficiencies based on the centralized cloud principle of sharing computing and storage resources via virtualization. To increase the capacity of cellular networks in dense areas with high traffic demands, low power nodes (LPN) are used to serve for pure data-only service with high capacity in heterogeneous networks (HetNet). The advantage of HetNet is to decouple the control plane and user plane. LPNs have only the control plane. The control channel overheads and cell-specific reference signals of LPNs are moved to macro base stations (MBS). Advanced coordinated multi-point (CoMP) techniques are adopted to suppress both intra- and inter-tier interferences among LPNs and MBS. To fulfill new breakthroughs anticipated in 5G systems, the cloud computing technologies are embedded into HetNet as the heterogeneous cloud radio access networks (H-CRAN) to realize the large-scale cooperative signal processing and networking functionalities [41]. And thus both the spectral and energy efficiencies are substantially improved beyond the existing HetNet and C-RAN.

The expectations from the broad, various 5G use cases and business models make it impossible to design a standard or one-size-fits-all 5G network. A new flexible architecture is needed to support an adaption to specific combinations of use cases, business models, and value propositions [53]. In 5G, a mobile network operator (MNO) can play one or more roles of a connectivity provider, an asset provider, or a partner service provider. The concept of micro operators ( $\mu$ O) calls for new business models for 5G local service delivery to build indoor/small cell communication infrastructure and offer context-related services and contents [2, 34]. Examples of  $\mu$ O business models include universities and hospitals of public  $\mu$ Os, shopping malls and mass events of commercial  $\mu$ Os, and manufacturing and construction of industrial  $\mu$ Os. Business opportunities for the  $\mu$ O concept may include the provisioning of hosted local connectivity to all MNOs in specific locations, operation of secure networks for vertical section specific use, and offering of locally tailored content and services.

Business models and business model innovation should align technological development and economic value creation [17, 39]. Business models bridge the gap between abstract strategies versus the practical decisions and actions. Value and revenue creation is the central point of business models. The 4C business model [56] is adopted to describe the structure and interaction in the  $\mu$ Os from the business model perspective [2]. The business model canvas developed by Osterwalder and Pigneur [39] is a proven approach in practice in describing, visualizing, assessing and changing business models. We adopt both the 4C and canvas approaches to develop our  $\mu$ O business models for casinos and entertainment.

Kibria *et al.* [32] propose a complementary business model that leverages a single wireless infrastructure to mutually benefit  $\mu$ O (a third-party service provider), the owner of the space and facility, and MNOs. The model consists of a  $\mu$ O slice, which delivers a venue with customized wireless services tailored to its local service requirements, and an MNO slice, which facilitates improved wireless coverage to visitors and end users with subscriptions to different MNOs.

### 3 Micro Cell and 5G Micro Operator

#### 3.1 Small Cells in C-RAN

The classic design of cellular radio access networks (RAN) is based on a single base station, covering a circular area centered by the base station. Several cell structures can be defined on the basis of the complexity of the base station and the size of the coverage area. In RAN, macro cells are high-powered radio access nodes, with the widest range of cell sizes of about 20 miles in diameter, and small cells low-powered radio access nodes, with a range of a few meters to a mile in diameter. Small cells are the last-mile broadband access infrastructure of Smart Cities that demand a wide plethora of ubiquitous and pervasive services readily available anytime and

everywhere. Small cells should provide very high capacity and reliability as well as the capability of configuration around the real users' needs.

Small cells are becoming an integral part of wireless networks and are used to complement coverage from larger macro-cell base stations, providing incremental capacity to a small geographic area. Small cells bring network closer to user, where and when needed, and are key part of the solutions to meet the anticipated high data rate demands. In a heterogeneous network, small cells work in conjunction to provide uninterrupted coverage for end users. Small cells can increase capacity in areas with high user densities, improve customer experiences through integrated broadband, and extend handset battery life by reduced power consumption. Small cells have advantages of saving in total cost of ownership (TCO) than macrocells while enhancing coverage and capacity. But small cells need integration and orchestration across different systems and groups. Deploying small cells are to enhance coverage and capacity. Small cells are widely deployed to provide coverage infill, intended coverage at cell edges of macro cells, and an alternative to cell split. The needs to deliver the right QoE (Quality of Experience) is driving worldwide investments in small cells. Small cells are poised to not only supplement, but also substitute macro networks to bridge the gap between capacity and demand for data.

There are several types of small cells including femtocells, picocells, and micro-cells, ranging from smallest to largest. Femtocells are user-installed to improve coverage area within a small vicinity, such as home, home, or a dead zone within a building. Femtocells can support only a handful of users and is only capable of handling a few simultaneous connections. Picocells offer greater capacities and coverage areas and can support up to 100 users covering an area ranging 250 yards. Picocells are deployed indoors to improve poor wireless and cellular coverage within a building, such as an office floor or retail space. Microcells can cover an area up to a mile in diameter. Microcells are frequently used temporarily in anticipation of high-traffic with a limited area, such as sporting events or concerts. Microcells can be also installed as a permanent feature of mobile cellular networks for university and hospital campus, large shopping malls, and Casinos and entertainment parks. Both microcells and picocells are usually installed by network operators. Fig. 1 depicts a cellular RAN architecture for 5G ICEMO.

A mobile network operator is judged by the QoE it provides to its subscribers. This is not new, but the context is evolving. The business case for a high QoE network that is *always on, and available anywhere, any time* is driving the network evolution toward an integrated heterogeneous network (HetNet), allowing for capacity expansion to be based on actual demands in traffics. The fundamental challenge for an operator is to leverage its strengths in macro networks and extend them to small cells. Operator' strengths in subscriber mix and in cost structure of its network infrastructures frequently determine the appropriate small cell strategy. An holistic small cell rollout strategy should focus on reducing network costs and explore new revenue opportunities. Most small cell strategies follow a four-phased approach—from reducing costs to serve (Phase I), to increasing traditional revenues

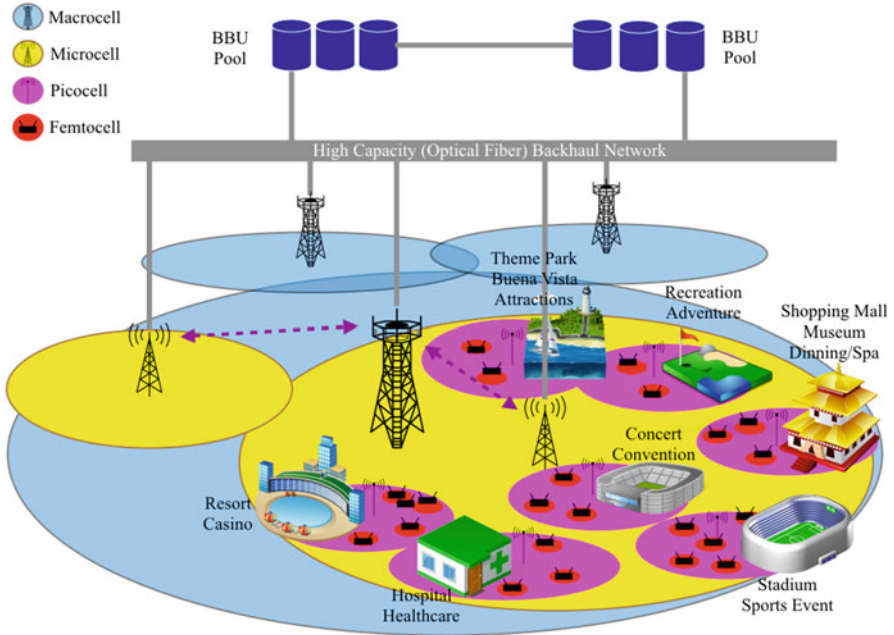


Fig. 1 Cellular RAN architecture

(Phase II), to improving customer experiences (Phase III), and finally to innovating business models (Phase IV) [8].

A new technology that is generating a lot of interests among wireless operator is C-RAN. C-RAN uses centralization, virtualization and pooling of base station baseband processing to drive efficiencies. A typical cell site currently must accommodate antennae and a base station. With C-RAN technology, antennae are deployed at macro and small cell sites, as they are today, to provide coverage to a particular area, but base stations are removed and aggregated into a centralized location. The centralized base stations are provisioned in software on virtual machines, on commercial servers, typically housed in carrier-grade data centers. The computing power required for the centralized base station is not dedicated to specific cell sites, but instead can be pooled to provide baseband processing power to the sites that most need it at any given time. Removal of the base station means simpler site acquisition and reduce space and power costs. Due to stringent requirements on bandwidth, delay and synchronization, fiber-based frontal connectivity is required to connect antennae and base stations. Wireless operators and network providers are working together to realize the immense benefits of C-RAN networks.

### 3.2 5G Micro Operator

The 5G business perspective and context are continuously evolving. Use cases for 5G networks are extremely diverse so that traditional mobile operators cannot provide all the requirements or they are weakly-suited to providing it. New players such as functions developers and facility managers enter the 5G market while the role of existing ones will drastically change. The services involved may be so specific or limited to a certain location that a MNO finds it difficult or even impossible to offer them. Telecom operators have to find ways to compete against or work with the Over-The-Top (OTT) players that offer innovative services on their networks and services. New market opportunities arise thanks to lower barrier entries. Vertical applications and services with demanding requirements are emerging but there are several challenges remain to be addressed prior to their deployment and successful adoption. These challenges appear not only in technical aspects but also in societal and economic considerations.

Economically, switching cost is a one-time cost that a buyer pays when switching from one provider to another. Switching cost builds an entry barrier since it reflects the monopoly power of incumbent firms. For high switching cost markets, an entrant firm should attract new customers by subsidizing customers' switching costs. For low switching cost market, competition is more dynamic and a new firm can more easily enter the market. For a long time, mobile users were unable to exploit patio-temporal differences between individual mobile network operators (MNOs) for a variety of reasons and high switching cost was one of the reasons. Reduction of switching costs can encourage the entrance of new operators. Specially if switching costs are low enough and end users can efficiently switch from one network to another, the economic scale of an operator decreases. Such cases are especially interesting for new network deployments such as small cells, M2M (Machine-to-Machine) and more generally IoT. This new type of operator is know as a Small Cell Operator or *Micro Operator* ( $\mu$ O). Due to the comparatively small size and limited resources,  $\mu$ Os with required scale and scope of operations are most likely to emerge if either the  $\mu$ Os serve a specific and necessary purpose and/or they own a large enough user base [2]. Scale of the  $\mu$ Os should be big enough to represent a lucrative business opportunity for MNOs. A 5G ICE microcell that covers a 2 km-radius area with dense human activities and machine communications is a good candidate for  $\mu$ Os.

The Small Cell concept is not new and has become central in today's 4G LTE-A network [18]. Small Cells provide superior cellular coverage, capacity and applications for residential, enterprise as well as dense metropolitan and rural public spaces [45] and are considered as an enabler in the wireless networking design with the aim of building service platforms for Smart Cities [37]. The 5G Infrastructure Public Private Partnership (5GPPP) SESAME Project is an innovative effort to design and develop a novel 5G platform based on small cells, featuring multi-tenancy between network operators. The idea of multi-tenancy plays a vital role

in 5G networks. 5G network infrastructures should provide rich virtualization and multi-tenant capabilities, not only in term of partitioning network capacity among multiple tenants, but also offering dynamic processing capabilities on demand, optimally deployed close to the user. Such an approach increases dynamic service capabilities and reduces the cost-of-ownership and overall energy consumption. Its potential benefits attract the interest of Communications Service Providers (CSPs) such as Mobile Network Operators (MNOS), Mobile Virtual Network Operators (MVNOs) and Over-The-Top (OTT) content and service providers, encouraging them to share in the 5G market by pursuing emerging business models.

#### **4 Integrated Casinos and Entertainment (ICE) Innovation in Smart Cities**

Many researchers have devoted to the study of 5G technology for the past years. Vendors and operators have started the testing of 5G components and trial test networks to offer commercial 5G services by 2020. However, there are many significant hurdles to clear on the way to the 5G starting line. One of the things standing in the way is that there are not enough different kinds of service providers, use cases and business models for densification or small cells in 5G. Small cells are a key ingredient for the mainstream rollout of 5G, which means operators need to ensure they have the right partnership, dominant use cases and strong business models in place to address site-specific innovation. The need for new and innovative business in 5G small cells continues to emerge.

A critical challenge for ICE businesses today is that purely technocentric innovation is less sustainable now than ever. The problem is that it tends to situate innovation within a paradigm dominated by technology, neither coming from markets nor pushing radically new meanings for products and solutions [11]. Business requirements encompass economic, physical, social, legal, and political factors that affect business activities. Significant changes in any of these factors are likely to create business pressures or opportunities on the individual business or even the entire industry. Today's casinos and entertainment business faces three types of opportunities and pressures—market, technology, and societal pressures. Market opportunities and pressures come from changing global economy and strong competition, customer loyalty and preferences, and changing nature of workforce. Technology opportunities and pressures are mainly from adoption of new technology, technological innovation and obsolescence, and information overload. Societal opportunities and pressures include social responsibility, government regulation and deregulation, protection against terrorist attacks, and ethical issues. Before diving into business modeling, we have to outlook to the future ICE business requirements and opportunities.

## ***4.1 Outlook to Future Integrated Casinos and Entertainment***

Integrated Casinos and Entertainment (ICE) are based heavily on service—it is the very experience that attracts guests and tourists to the area. Gaming, food, lodging, and entertainment are all service-based industries, and the casinos and entertainment operations reflect on the importance of quality of service. Key stakeholders in the ICE industry value chain include (1) casino service delivery, (2) physical environment, (3) equipment vendors, (4) technology vendors, and (5) supervisory agencies, detailed as follows:

### **4.1.1 Casino Service Delivery**

Customer service is a crucial ingredient for success in a casino and a resort hotel. This makes casino employees and their service an important primary investment for operators—when guests have a great experience and are treated well by employees and their service, they are more likely to return to that casino and recommend it to friends.

### **4.1.2 Physical Environment**

The comfort, design and appearance of a casino, a theatre, or a theme park are also critical for providing a superior guest experience. Aspects such as ambient conditions, floor layout, interior decor, cleanliness, and seating comfort encourage guests to stay longer and typically spend more money at the gaming tables. Supply chain stakeholders in this area include architects and designers, ventilation suppliers, mechanics, power suppliers, and casino employees.

### **4.1.3 Equipment Vendors**

Casinos and entertainment rely on the latest gaming machines and innovations to attract guests. Working with reputable gaming equipment vendors who adhere to the regulations surrounding gambling is an essential step in the supply chain.

### **4.1.4 Technology Vendors**

Modern casinos and entertainment are heavily technology-oriented. Most casinos employ technology systems for Customer Requirements Management (CRM), Point of Sale (POS), in-room entertainment systems, customer tracking, slot card systems, and more. Casinos rely on both hardware and software technology vendors as a crucial part of their supply chain. The adoption and deployment of 5G

technology in ICE will shape the competitive edge and create the optimum values to their visitors and customers.

#### 4.1.5 Supervisory Agencies

The gaming industry is highly regulated, which makes the gaming commission and gaming control a key stakeholder in the service-focused casino supply chain. Without authorized permission, casinos are unable to provide any service.

Potential benefits of 5G technology for the value chain in ICE industries include

- reduced costs of overall transactions and IT infrastructure,
- irrevocable and tamper-resistant transactions,
- ability to store and define ownership of any tangible or intangible asset,
- reduction of systemic risks (credit and liquidity risks),
- consensus in a variety of transactions,
- increased accuracy of trade data and reduced settlement risk,
- near-instantaneous clearing and settlement,
- improved security and efficiency of transactions, and
- enabling effective monitoring and auditing by participants, supervisors, and regulators.

Table 1 depicts the 5G Opportunities and Innovations in Integrated Casinos and Entertainment Smart Cities.

## 4.2 Concentric Value Circles Model for 5G ICEMO

We propose a Concentric Value Circles (CVC) model for analysis of 5G ICEMO. The development of a CVC model starts from identifying the ultimate value of the business. We then create an innermost value circle for that ultimate value. Any values that directly contribute to the ultimate value are identified and placed in a larger value circle, the second value circle, concentric to the innermost value circle. A third concentric value circle is created where each of its values directly contribute to one or more values in the second concentric value circle. The CVC model continues to grow as more concentric value circles are added to the model. The feature of the CVC model is that each value in the larger concentric value circle contribute directly to one or more values in its next smaller concentric value circle.

Our 5G ICEMO CVC model is created as follows. Customers spending their time and money in casinos and entertainment are looking for *fun*—the ultimate value of casinos and entertainment, or hospitality industry. The innermost concentric value circle of the 5G ICEMO CVC model is for fun. Its next larger concentric value circle is for *relaxed, friendly, cost-effective, convenient, safe and fair* values that all contribute directly to fun, the ultimate value of casinos and entertainment. For



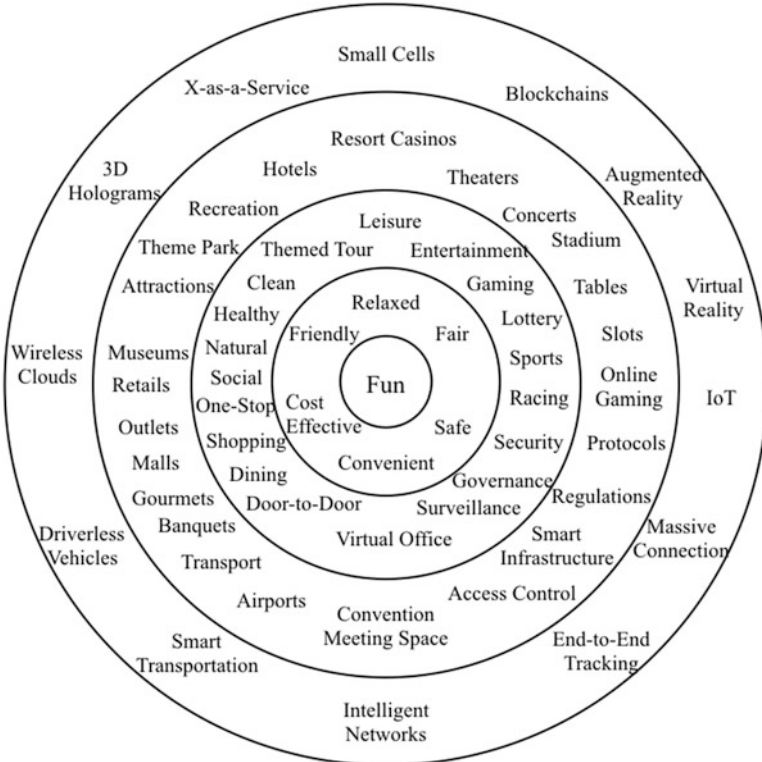
**Table 1** 5G opportunities and innovations in ICE smart cities

5G enabling technologies		Opportunities and innovations in ICE smart cities		
3D holograms Augmented and virtual reality Block chains Driverless vehicles End-to-end tracking Internet of things Intelligent networks Smart transportation Massive connection Small cells Wireless clouds X-as-a-service	Customer service	VIP/junket	Seamless, personalized door-to-door arrangement	
	Management	Business		Wireless clouds office, augmented reality meeting
		Tourist		Localized and personalized information sharing and service, latency-less, boundary-less mobility
		Logistics		Real-time tracking, smart planning of foods/beverage/fruits
		Internal control		Asset management with RFIDs and wireless sensors, revenue tracking in cage, tables and slot machines.
		Human resource		Real-time CRM, smart staff scheduling, integrity check
		Floor operations		Driverless vehicles, welcome robot, context-aware smart operations, fraud prevention
		Transportation		Autonomous transport—Indoors, outdoors, inter- and intra-campus
		Networks		Efficient and scalable, intelligent network management
		Equipment		Flexible, cooperative gaming machines and innovations

(continued)

**Table 1** (continued)

5G enabling technologies	Opportunities and innovations in ICE smart cities
Activity	Private events (wedding, ceremony, family reunions)
	Public events (convention, exhibition, sports, concert)
	Entertainment
Marketing	Advertising
Environment	Privacy
	Security
	Safety
	Health
	Natural
	Social media, event planning and management
	Massive connections, smart parking and traffic control
	eTicketing, broadcasting, social networks
	Targeted advertising, personalized advertising
	Protection of personal data and user identifiers
	Self-adaptive, intelligent security controls
	Terrorist attack prevention, integrated emergency service, surveillance, secure transaction
	Massive wireless sensors for PM2.5, CO, O3, SO2
	Energy saving



**Fig. 2** Concentric value circles model for 5G ICEMO

customers, these values are important and fundamental to enjoy funs in casinos and entertainment. To contribute these values, values in the third concentric value circle are identified—they are *leisure, entertainment, themed tour, clean, healthy, natural, social, one-stop service, shopping, dining, door-to-door service, virtual office, surveillance, security, racing, sports, lottery, and gaming*. The fourth outer concentric value circle is made to contribute the values in the third value circle. Values in the fourth outer value circle are *resort casinos, hotels, theaters, concerts, stadiums, recreation, theme park, attractions, museums, retailers, outlets, malls, gourmets, banquets, transport, airports, convention and meeting space, access control, smart infrastructure, regulations, protocols, and tables, slots, and online games*. The fifth value circle, the outermost concentric value circle of our 5G ICEMO CVC model, represents the values from 5G deployment to customers in casinos and entertainment. The values are generated from *small cells, X-as-a-Service, 3D holograms, wireless clouds, driverless vehicles, smart transportation, intelligent networks, end-to-end tracking, massive connection, IoT, blockchains, augmented and virtual reality*. Fig. 2 depicts the proposed 5G ICEMO CVC model.

### **4.3 5G ICEMO Business Model**

In the proposed 5G ICEMO CVC model, in addition to the main player, ICEMO, many players and actors are involved in the 5G ICE ecosystem. According to their types and relevance, the players and actors can be divided and categorized into the following target groups:

- *User*
  - End Users
  - Subscribers
- *Operator*
  - ICE Micro Operator (ICEMO)
  - Virtual ICE Micro Operator (VICEMO)
  - Macro Network Operator (MNO)
- *Device, Equipment, Facility and Infrastructure Vendor/Owner/Manager*
  - IT and Network Equipment Vendors
  - Small Cell Vendors
  - CPE (Customer Premises Equipment), IoT Device and Hardware Vendors
  - Facility and Equipment Manager
  - ISP and Fixed Telecomm Providers
  - Spectrum Owners
  - Venue Owner
- *Solution Provider*
  - OTT and e-Service Providers
  - Content Providers
  - Content Aggregators
  - Network Function Providers (NFPs) and Software Providers
- *Vertical Industry*
  - Gaming & Casinos
  - Media & Entertainments
  - Tourism & Leisure
  - Hotels and Resorts & Hospitality
  - Supply Chains & Logistics
  - Healthcare & Clinics
- *Advertising Agency*
- *Broker and Regulator*
- *Distributor*
- *Smart Cities*

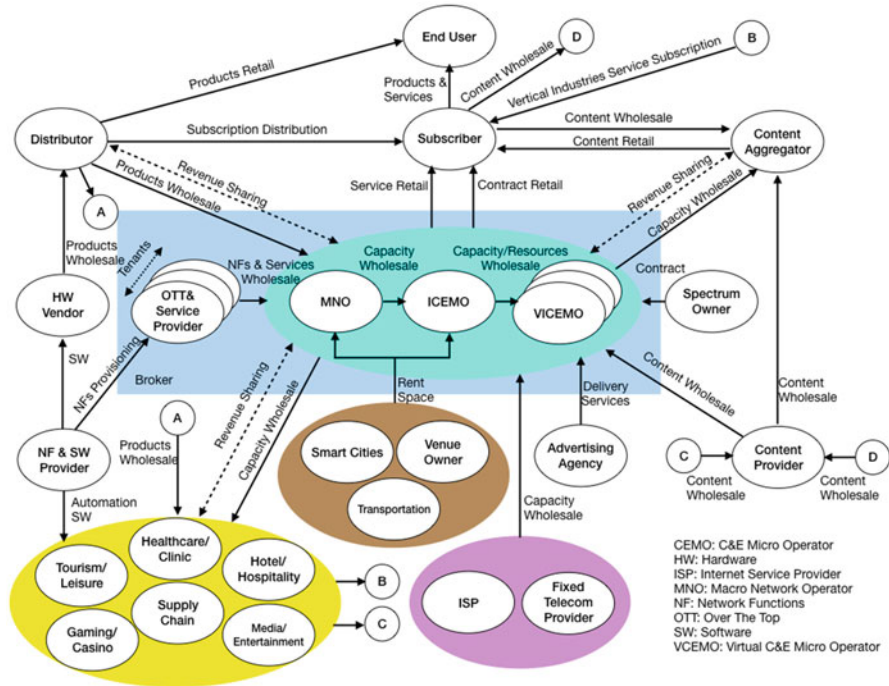


Fig. 3 5G ICEMO business model

Figure 3 depicts the 5G ICEMO business model, which is based on the basic SESAME business model [17] and brings the 5G ICEMO features into the model. The business model integrates all the above participating players and actors, illustrates the relationship among the players/actors, and identifies the revenue streams. In the model, operators and providers act as the main responsible players towards the subscriber by providing network services. Contracts or Service Level Agreements (SLAs) are made between the subscriber and the provider for the usage of services. End users may subscribe for different services from one or many operators/ $\mu$ O/providers. Some of the vertical industry’s services are offered to the end users through a distributor. The provisioning of contents are either directly by content providers or aggregators or by the  $\mu$ Os. Examples of contents include news, entertainment, video, audio, and so on. Value-added services can be provided through the  $\mu$ O by other players, including location-awareness contents and services, banking, sports update, travel and transportation, weather, contests and voting, online gaming and lottery, and so on. Collaboration with vertical industries generates further revenue sharing between the  $\mu$ Os and players providing services in vertical markets.

## 5 5G Cloud-Enabled ICEMO

Many advanced technologies like cloud radio access network (C-RAN) have been presented as potential 5G solutions to reduce both capital and operating expenditures. To increase the capacity of cellular networks, low-cost remote radio heads (RRHs) are moved closer to the users and serves for the pure data-only service, while the baseband processing is centralized at the BaseBand Units (BBU) pool. C-RAN is sometimes referred to as Centralized-RAN, that separates the radio function unit, the RRH, from the digital function unit, or BBU by fiber. A Centralized RAN system puts many processing resources together to form a pool in a central data center. Such aggregation of processing and storage resources in shared locations not only reduces deployment costs, but also leverages low latency connections between different RAN processing units.

In C-RAN architecture, all the baseband processing is made by BBU at centralized data centers, and radio signals are exchanged with the RRHs over high-speed, low-latency connections that constitute the mobile fronthaul. Radio frequency (RF) signals have to travel through a long cable, a few hundreds meters or a few kilometers away, from the base station cabinet to the antenna at the top of the tower. The requirements of the high data throughput and low delays in the fronthaul makes this *fully centralized* solution nowadays feasible only when optical fibre connection are deployed, which is not available everywhere and its complete deployment entails costly investment.

Many alternatives for *partial centralization* have been proposed to explore potential advantages of C-RANs while deploying optimized services with minimum delays. Beyond the complete centralization of all the BBU functions, the current trends towards the Cloud RAN also allowing reusing the available hardware infrastructure for deploying service instances at the edge of the mobile network. To cope with more personalized and user-centric service provisioning, one of the promising technologies is from the novel Mobile Edge Computing (MEC) industry initiative [22], which is done by running applications and performing related processing tasks closer to the mobile end users so that network congestion is reduced and applications will perform better. Different from the architecture for C-RAN, MEC-driven service instances are deployed over the cloud resources available at the RAN side. The types of MEC services would be deployed in form of application programs (APPs).

Besides the logical separation of the MEC functions from the RAN system, some processing and storage resources are placed close to the RRH by using geo-distributed data centers, and thus the fronthaul delay can be significantly reduced. This is especially relevant to enable flexible deployment of Small Cells, and particularly attractive for targeting currently deployed network architectures and special limited-access scenarios. In the following, we present the solution for 5G ICEMO based on the Distributed 5G Cloud Enabled Small Cells architecture developed in the scope of the European H2020 5GPP Project—Small cELLS coordinAtion for Multi-tenancy and Edge services [50].

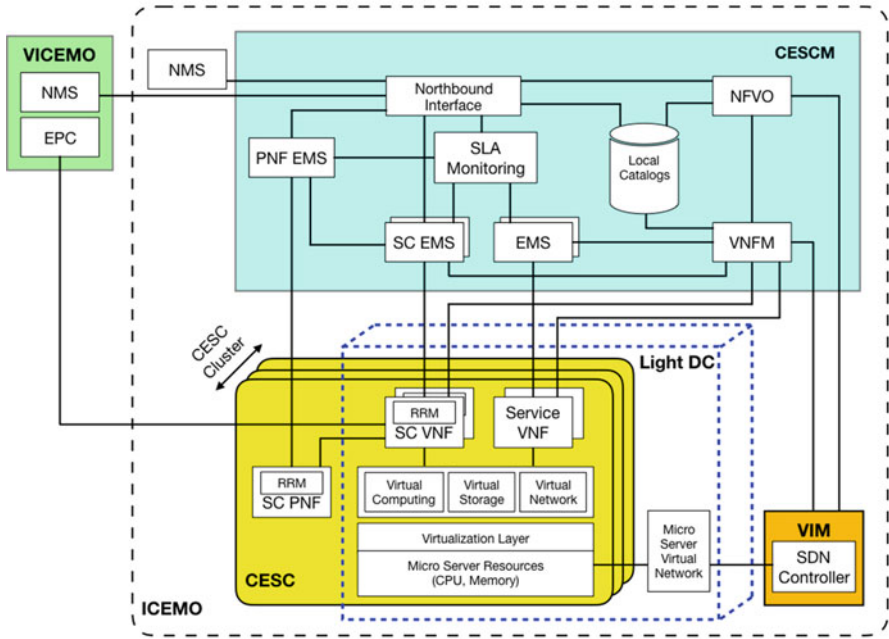


Fig. 4 5G ICEMO system architecture

### 5.1 5G ICEMO System Architecture

Our 5G ICEMO system architecture is hierarchically organized into four layers: the *NFV Infrastructure (NFVI)* layer—including the physical and virtual nodes (on-the-shelf servers, VMs, storage, switches, routers and so on) on which the services are deployed, the *NFVI Infrastructure Management* layer—including the infrastructure management entities (VIM, WICM), the *Orchestration* layer—including the Orchestrator and NF Store, and the *Applications* layer—including all the customer-facing applications and modules which facilitate multi-actor involvement and implement business-related functionality. Note that the NFVI and NFVI Infrastructure Management layers are conceptually grouped as Infrastructure Virtualization and Management (IVM). The 5G ICEMO system architecture is shown in Fig. 4.

### 5.2 5G ICEMO Wireless Network Architecture

Based on the required functionalities as well as architectural principles of the SESAME system architecture, we develop the 5G ICEMO architecture, as proposed by Fig. 5. The key innovations proposed in the SESAME architecture are the novel

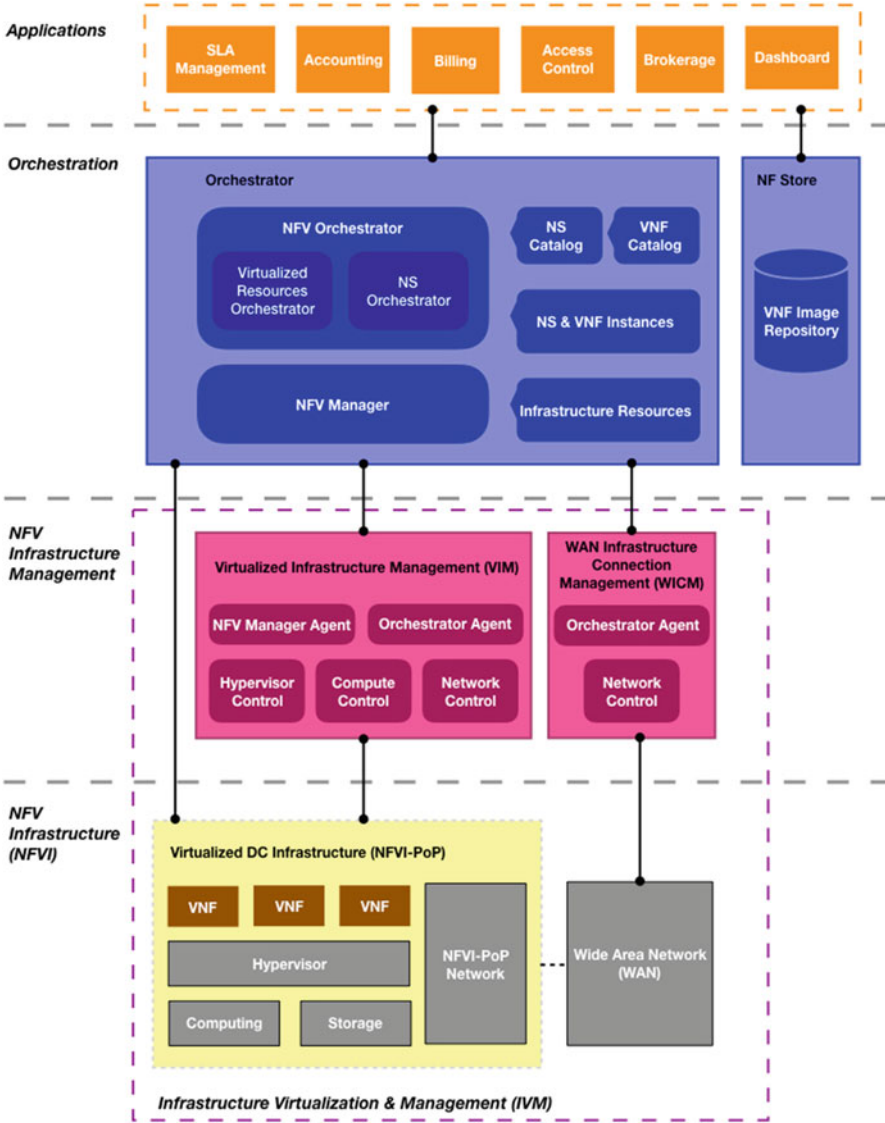


Fig. 5 5G ICEMO wireless network architecture

concepts of virtualizing Small Cell networks by leveraging the paradigms of a multi-operator enabling framework coupled with an edge-based, virtualized execution environment. SESAME falls in the scope of these two principles and promotes the adoption of Small Cell multi-tenancy, i.e., multiple network operators will be able to use the SESAME platform, each one using its now network slice. The deployment of Small Cells adopts some virtualized functions, with each Small Cell



containing a micro server through fronthaul technology. The micro servers form the Cloud-Enabled Small Cell (CESC) and a number of CESC s form the CESC Cluster, capable to provide access to a geographical area with one or more operators.

The CESC offers computing, storage and radio resources. A CESC consists of a complete Small Cell with its backhaul interface, management connection and with necessary modifications to the data model to allow Multi-Operator Core Network (MOCN) radio resource sharing. The CESC is composed by a physical Small Cell unit and an execution platform architecture, the micro server. The CESC includes components of Small Cell Physical Network Functions (SC PNF) and Small Cell Virtual Network Functions (SC VNF). The SC PNF implements the radio interface and the main protocol features, and together with the SC VNFs, provides the complete functionality of the virtualized Small Cell. SC VNFs are hosted and executed in the micro server. Different small cell functions are instantiated and managed in the micro server by the management and orchestration layer in the CESC Manager (CESCM). Both SC PNF and SC VNF have an Element Management System (EMS) for FCAPS (Fault, Configuration, Accounting, Performance and Service) purposes. Both reside in the CESCM and are connected to the Network Management System (NMS).

Cloud computing and networking is realized through the sharing of computation, storage and network resources of the micro servers present in each CESC and composing the Light Data Center (Light DC). Through virtualization, the CESC cluster can be seen as a cloud of resources which can be sliced to enable multi-tenancy. Therefore, the CESC cluster becomes a neutral host for mobile ICEMO or Virtual ICEMO (VICEMO) which want to share IT and network resources at the edge of the mobile network. In addition, cloud-based computation resources are provided through a virtualized execution platform. This execution platform is used to support the required Virtualized Network Functions (VNFs) that implement the different features and capabilities of the Small Cells (and eventually of the core network) and the cognitive management and *Self-X* operations, as well as the computing support for the mobile edge applications of the end users.

The CESC clustering enables the achievement of a micro scale virtualized execution infrastructure in the form of a distributed data center, Light DC, enhancing the virtualization capabilities and processing power at the network edge. Network Services (NS) are supported by VNFs hosted in the Light DC which is constituted by one or more CESC s. VNFs leverage on SDN and NFV functionalities that allow achieving an adequate level of flexibility and scalability at the cloud infrastructure edge. More specifically, VNFs are executed as Virtual Machines (VMs) inside the Light DC, which is provided with a hypervisor, based on Kernel-based VM (KVM), specifically extended to support carrier grade computing and networking performance.

Light DC provides a virtualized execution environment for chaining different VNFs to meet a requested network service (NS) by a mobile network operator. An NS is understood as a collection of VNFs that jointly supports data transmission between User Equipment (UE) and operators' Evolved Packet Core (EPC), with the

possibility to involve one or several service VNFs in the data path. Thus, each NS is deployed as a chain of SC VNFs and Service VNFs.

The CESC Manager (CESCM) is the central service management and orchestration component in the overall architecture. It integrates all the necessary network management elements, traditionally suggested in 3GPP, and the novel recommended functional blocks of NFV MANO (Network Functions Virtualization Management and Orchestration) [23]. A single instance of CESCM is able to operate over several CESC clusters, each constituting a Light DC, through the use of a dedicated Virtual Infrastructure Managers (VIM) per cluster.

For each instantiated VNF, an EMS, deployed in the CESCM, is responsible to carry out management functionalities. The ICEMO NMS is the central management point for the whole network of the ICEMO, while the PNF EMS and the SC EMS are in charge of the management of the physical and virtualized network functions, respectively. In particular, the PNF EMS and SC EMS include different centralized self-x functionalities to carry out the automated management of different radio parameters. A Northbound Interface is provided in the CESCM to handle all the ICEMO and VICEMO NMS communications. The lifecycle management of deployed VNFs is carried out by the VNF Manager (VNFM) in the CESCM. Service Level Agreements (SLAs) are negotiated and agreed offline between the business players and made to the CESCM SLA Monitoring. Besides management and orchestration of the above mentioned functionalities, NFV Orchestrator (NFVO) composes service chains and manages the deployment of VNFs over the Light DC to enhance the overall system performance.

## 6 Illustrative Use Cases

### 6.1 *Mega Jackpot*

Slot or poker games are fascinating not only for their flashing lights and jubilant sounds filled on casino floors, but also because they are usually banded together to offer larger jackpots than just one single machine could. The premise of a progressive jackpot network is simple. Beginning with a baseline sum, a tiny percentage of every dollar wagered within the network is diverted to a centralized jackpot. As time goes on, the jackpot total gradually grows ever higher, until a lucky player finally lands the right real combination to trigger the award. Along with the growing number of devices connected, larger and larger jackpots, or Mega Jackpots, have been awarded, which attracts more customers to try their fortune and contributes huge revenues to the casinos. Modern jackpot systems are capable of managing thousands of games connected across hundreds of sites in several states with centralized control and advanced security and analysis functions.

In addition to the concerns of single-point-of-failure and scalability, the centralized network of jackpots are more costly to operate and maintain. Each time

when a wager takes place, the information of its contributed percentage of money has to be sent to the centralized server to update the centralized jackpot which is later broadcast to each device in the jackpot network. All these transactions and processes demand low latency delays and good quality of experiences. The concept of 5G ICEMO and the blockchain technology has the ability to address these critical issues of Mega Jackpots. A blockchain is a distributed data structure that makes it possible to create a digital public ledger of data and share it among a network of independent parties. The blockchain stands as a *decentralized, trustless* mechanism of all the transactions on the network [49]. All the blockchains use cryptography to allow each participant on any given network to manage the ledger in a secure way without the need for a central authority to enforce the rules. A blockchain is a peer-to-peer system with no central authority managing data flow. To prevent the network from being corrupted, blockchains are not only decentralized but they also use a digital token, or a cryptocurrency, that has a market value and are traded on exchanges. Blockchain technology could become the seamless embedded economic layer that the Web has never had. Blockchains can serve as the technological underlay for payments, decentralized exchange, token earning and spending, digital asset invocation and transfer, and smart contract issuance and execution.

The technical implementation of the scenario of Mega Jackpots is based on *smart contracts* that are coupled with blockchains in 5G ICEMO with all its features in an autonomous way. A smart contract, or blockchain-based smart contract, is autonomous software that controls ownership and access to an asset by having it registered as a digital asset on the blockchain and having access to the private key. A smart contract is both defined by the code and executed by the code, automatically without discretion, like a vending machine. Smart contracts feature three distinct characteristics—autonomy, self-sufficiency, and decentralization [49]. In the 5G ICEMO scenario, the data of Mega Jackpot are stored as the cryptocurrency in a blockchain and all processes are implemented and managed by smart contracts. The Small Cell infrastructure of 5G ICEMO decentralizes the application interface for a set of smart contracts which interact with each other and transfer tokens from one to another. Changes in the scenario are made by a vote of control token holders. Smart contracts are stored on the blockchain which all parties have a copy of. Fig. 6 depicts the Value Network Configuration (VNC) model [13] for the employment of the blockchain technology with smart contracts in 5G ICEMO.

## 6.2 Anti-Counterfeiting Lottery

Lottery is the largest social and entertainment game in the world. Despite the popularity with over 30% of market share on the world's gambling market, lotteries today cannot provide users with a 100%-guaranteed honest lottery or offer transparency with regard to the formation and distribution of the prize fund. In addition, lottery or gaming industries always have a need for preventing counterfeiting scams. Lottery tickets or payouts for slot games, video pokers, as well as similar gaming

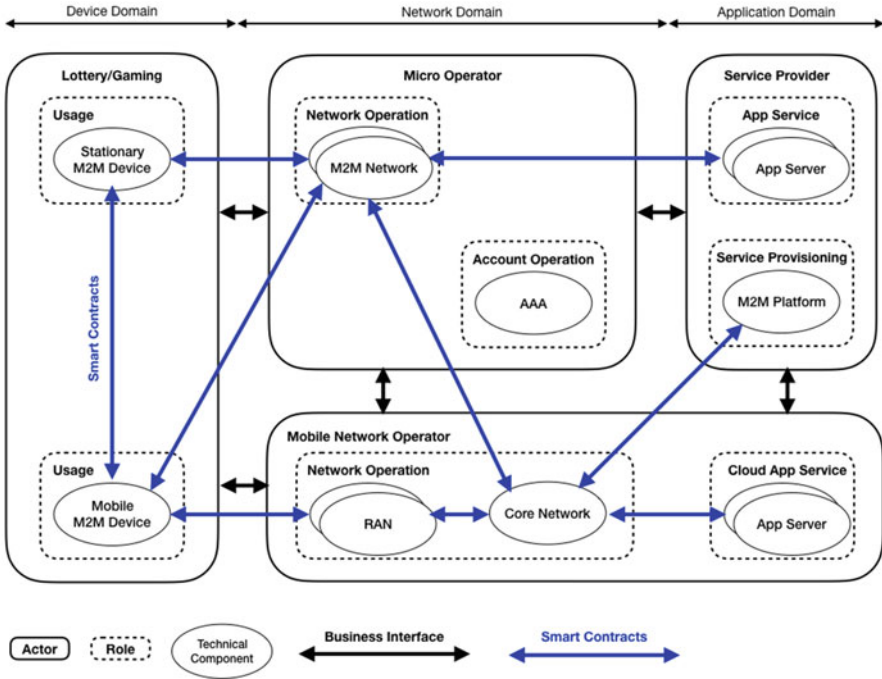


Fig. 6 5G ICEMO VNC model with blockchains and smart contracts

products popular in casinos and gaming establishments, are subject to threats of surreptitious readout, alternation, counterfeiting, and bad quality control of product. Various efforts to combat counterfeiting lottery or payout tickets have been employed, including the use of specially-selected papers and printing, applications of universal product codes or bar codes, holographic designs, and the employment of special technology embedded in the tickets. A universal problem with these various techniques is that they are either expensive or impractical to implement in retail and casino floors, especially, for printing on site.

Validation of a winning lottery ticket or a gaming machine payout ticket is through a control number. No payment is made unless the control number printed on the ticket is correct. Control numbers are ordinarily generated in computers or the embedded microprocessors by a scheme, protocol, or algorithm, in order to prevent successful counterfeiting of lottery or payout tickets and redemption of bogus tickets. To validate a control number, database queries to remote centralized computer systems are needed, whose transactions usually spend a lot of time and are vulnerable to be altered or attacked.

A scenario in which the proposed concept of 5G ICEMO can be exploited to prevent counterfeiting of lottery or payout tickets and redemption of fake tickets is to use the Small Cell features in 5G as well as the pervasive, decentralized blockchain technology. The potential of blockchain technology can improve anti-

counterfeit measures in gaming and lottery industries and have a significant positive social impact by identifying counterfeits and duplicated tickets. Furthermore, the blockchain technology can verify fraudulent transactions to assure and improve the effectiveness of risk management, control and governance processes in gaming and lottery operations. A blockchain-embedded 5G Small Cell ecosystem is a good place to build decentralized applications and services for casinos and lottery operators. Validation of control number of a lottery or payout ticket is not only based on the control number itself but also on its *chaining* (the hash—a fingerprint of this data and locked in blocked in order and time) which can be done locally, decentralized, and quickly. The encrypted pieces of the control number, the hash, tokens, code and other information are formed as smart contracts which are shared among different roles, self-verify the conditions met to execute the contract, and self-execute by releasing data.

### ***6.3 Autonomous Transport***

Industry awareness about the importance of V2X (Vehicle to Everything) is clearly gaining momentum, strongly driven by the needs in enabling driverless vehicles in the era of 5G. The automotive industry is on a transformation path, rapidly transitioning from driver-assisted systems to fully autonomous functions, which will eventually remove the driver in the loop. Autonomous vehicles are expected to make road transport safer, sustainable, and more cost-effective, while at the same time expanding its use. The next generation of Intelligent Transport Systems (ITS) will combine both types of communication so that vehicles can operate autonomously and be controlled and monitored from cloud software. In addition to cutting-edge access and a range of intelligent network functionalities, vehicle scheduling services are needed for picking up passengers or goods along a route, considering not only the locations and availability of vehicles, but also using contextual data such as information from the roadside sensors that can detect various road conditions, traffic density, accidents, and other issues, as well as data from mobile networks to provide more informed scheduling services.

The characteristics of ICE make it straightforward to embrace the use of automation in the movement of people in and around the ICE environment, for both their indoor and outdoor activities. Airports, as the major gateway to welcome and see off visitors to and from the ICE, have pioneered to automate the movement of people and luggage for decades. Monorails, airport shuttles, buses and taxis are used to take passengers between the terminals and nearby Metropolitan Rapid Transits (MRT) stations or lodging hotels/resorts, and electric club/golf carts for indoors and campus-wide movement. Autonomous transport in ICE demands not only driverless vehicles but also the integration of intelligent transport services, including fleet management, route planning, preventive maintenance, gas fill & battery charging, spare parts inventory, and so on. As most of the activities in ICE take place indoors, especially in large building complexes, indoor scenarios for robotic cars

should consider the physical environment, people interactions, the availability and reliability of sensing resources, and navigation support, all of which are different drastically from outdoor scenarios.

The autonomous transport scenario in 5G ICEMO is described as follows. Driverless shuttles, buses, and trains run for getting around the ICE and nearby gateways. For long walking distances, people use autonomous vehicles, rental cars, or taxis to their destination. Massive sensors, like wireless vehicle detection or light detection and ranging (LIDAR) sensors, are deployed in driveways, vehicles, curbsides, and more places, to make cars safer, more capable, and more foolproof. These V2X sensors collect and send data via the 5G ICEMO wireless network systems with guaranteed low end-to-end latency and reliability. The 5G ICEMO also provides the required scalability and flexibility to accommodate the requirements of rush crowds at holiday weekends and after the theater or the sports game. The scalability and flexibility come from the multi-tenancy architecture in 5G ICEMO and take advantage of 5G spectrum sharing paradigms.

## **7 Benchmarking of Las Vegas, Macao and Singapore**

The renowned cities of Las Vegas, Macao, and Singapore are benchmarked for their vision, promise and deployment to Smart Cities and the potentials to the proposed 5G ICEMO concept. Las Vegas has been famous for its title of “Gaming Capital of the World.” But since this century, the explosive growth of gaming revenues in Macao has shown the serious pressure on Las Vegas. With the openings of integrated resorts in 2010, Singapore’s gaming-integrated resorts have become the most profitable casinos in the world. As the top two Asian gambling destinations, Singapore and Macao are competitors to Las Vegas in drawing in wealthy gamblers from across the globe, especially from China, East Asia and the Pacific Rim. Macao is a city of long-standing gambling authority. Adjacent to China, Macao houses over 30 thriving casinos and draws a steady stream wealthy from Chinese gamblers, making it the most successful gambling city on the planet. The Lion City of Singapore is home to two booming casinos which attract a large number of China’s high rollers and made the country one of the top Asian VIP gambling destinations in less than 5 years.

### **7.1 Smart Cities in Las Vegas**

Las Vegas is no stranger to rapid change and innovation. The city is leading not only in business and entertainment, but in pioneering the use of smart city technologies. Las Vegas’s casinos are the early adopters of the IoT and machine learning to increase customer loyalty and satisfaction, just like smart cities want to do for citizens. In February 2016, the city of Las Vegas submitted its *Proposal*

for *Beyond Traffic: The Smart City Challenge* [51] to seek partnership with the United States Department of Transportation (USDOT) on its Smart City Challenge Grant initiative. The city of Las Vegas has already invested \$500 millions in smart infrastructure, transformed a downtown urban core into an Innovation District where a platform is established to demonstrate innovations of safe, efficient, sustainable and environmentally conscious mobility. Las Vegas is paving the way to becoming a smart city by 2025.

One of the primary aims of creating a smart Las Vegas is to improve congestion at major intersections for vehicles, pedestrians, and bicycles. Every year, more than 22,000 events like the Consumer Electronics Show (CES) and Specialty Equipment Market Association (SEMA), take place in Las Vegas, with over five million attendees. With millions of people flooding the streets and downtown in Las Vegas, the city wants to achieve zero traffic fatalities, and to improve the flow of traffic. The streets need to be smarter, the cars too, and traffic control center will be able to analyze traffic data in real time. Las Vegas has launched the country's first all-electric passenger shuttle and has proposed such initiatives as testing autonomous vehicles and smart lighting. Due to the high number of tourists that Las Vegas attracts, the city's Smart City efforts provide valuable lessons in scalability.

Las Vegas is partnering with networking giants to speed its progress towards becoming a smart connected city, while improving resident and visitor safety, reducing carbon footprint, increasing sustainability, and stimulating economic growth and diversity. Smart, connected, digital platforms and solutions are used to aggregate real-time IoT device and sensor data for actionable and immediate intelligence to empower the city to effectively reduce traffic congestion, provide faster response time to emergencies and lower operations and maintenance costs.

Las Vegas is a destination that does not just rely on gambling to attract guests. People go gambling for short periods of time. Instead, people come to Las Vegas for the full resort and entertainment vacation experience like world-class shows and high-end shopping. Las Vegas is best equipped for MICE—Meetings, Incentives, Conferences and Exhibitions, especially the best to host large trade show and convention guests who can use huge amount of meeting and exhibition space and spend money during the weekdays, in addition to the massive tourists and visitors spending their weekends or holidays there.

## 7.2 *Smart Cities in Macao*

In its first-ever Five-Year Development Plan which was released in September 2016 [25], the Government of Macao laid out a string of plans and measures aiming to convert Macao into a Smart City in the long term. The vision for Macao's future long-term development is to become *One Centre*, a true tourism and leisure city, which is resident friendly, business friendly, commuter friendly, tourist friendly, and entertainment friendly. The target is to develop Macao into a world-class tourism and leisure centre, and a city where people can enjoy international standards of

living, work, transportation, tourism and entertainment by the mid-2030s. The development strategies in Macao during 2016–2020 include improving soft and hard infrastructure and quality of tourism services, and expediting Smart City development and facilitating integration of industries and the Internet.

Towards a Smart City, Macao plans to develop “Smart Tourism” by helping enterprises to provide all-round services and information via the Internet, “Smart Transport” to improve management and quality of transportation and to enhance public’s convenient travel experiences, “Smart Healthcare” by incorporating more IT solutions into medical and health services, and “Smart Government” by implementing the e-Government and using big data for analyzing public information. One of the major tasks for building a Smart City of Macao is to accelerate the establishment of the three-network (telecommunications networks, cable TV networks, and the Internet) integration system and launch the three-network integrated services in 2019.

Macao is harnessing the cloud computing capabilities of Chinese technology giants to transform into one of Asia Pacific’s leading Smart Cities. The collaborations are in upgrading the IT infrastructure in Macao to foster digital developments in tourism, transportation, healthcare, governance and talent development. Residents and tourists visiting Macao expect to benefit from the transformation, from using artificial intelligence to optimize the management of road, air and water transportation to customizing online promotions at the airport, commercial districts, tourist spots, convenience stores and restaurants. The customized service could be based on the reams of user data from mainland China tourists, who make up almost two-thirds of all tourists entering the city.

Poor IT infrastructure is a major constraint on the Smart City development in Macao. While the rest of the world are racing on the development of 5G mobile networks, the city’s 4G mobile networks, launched in December 2015, are still not widely used. An operator is closely working with Macao Pass to launch the M-wallet service and implementing WiFi Bus solutions on over 330 buses with 4G+ technologies. The 4G+ service aims at bringing more quality, innovative and diversified 4G service experience to Macao residents. In conjunction with the fibre-optic network covering the whole city, compliments with its WiFi hotspots, data centers and cloud networks, Macao Government and telecom operators expect the 4G+ service to lay an important foundation for building the “Digital Macao” and open a new era of Macao’s telecommunications industry.

### ***7.3 Smart Nation in Singapore***

In October, 2016, two departments within Singapore Government were merged and consolidated to form the IMDA—Infocomm Media Development Authority, a new function to create a competitive Infocomm Media ecosystem that realizes the country’s 2025 masterplan, the ambition towards Smart Nation of Singapore. To address the digital divide and create digital multipliers, connection, collection



and comprehensives are emphasized. Many initiatives were launched in line with the birth of IMDA with the hope of transforming Singapore. The integration of media technologies like Augmented Reality (AR) and Virtual Reality (VR) are to apply for a variety of sectors, like education, healthcare, and entertainment. New schemes are to develop for urban logistics, aiming to disrupt the supply chain by sharing logistical resources such as vehicles, drivers and warehouses. Sensors and actuators from advanced technologies of sensing, communications, and IoT, will be widespread used and deployed. Tech-augmented security guarding is achieved by using instant image and video analytics technologies, based on the data from surveillance cameras in the streets or from the flying drones. To encourage innovation and collaboration between citizens and companies, open data is made available by Singapore Government.

As the telecommunication regulator of Singapore, IMDA encourages 5G trails conducted by mobile network operators (MNOs) in Singapore. Previous 5G trails have demonstrated promising capabilities and have achieved throughputs of more than 1Gbps with extremely low latency of less than 1 millisecond [29]. IMDA is also considering developing regulations to support the deployment of spectrum aggregation technologies while ensuring that deployment technologies such as WiFi can continue in license-exempt spectrum bands in Singapore. In Singapore, a local telecom company is working with some industry partners to start trails from 2017 for some potential 5G applications, including fleet management in the transport and logistics space. Such trials, when conducted in a real-world environment, will assist the industry in better understanding how 5G will work in Singapore's business environment and its optimum deployment scenarios.

Some mobile users in the city-state of Singapore have benefited from improved indoor mobile surfing in their 4G connection because the country's wireless operators invested and started deploying new small cells technologies in some high-traffic locations such as casinos, food courts and malls. In 2015, 4G HetNet successfully served hundreds of thousands of customers at two major annual events—SG50 Jubilee and 2015 Formula 1 Singapore Airlines Singapore Grand Prix. The operators in Singapore have deployed 4G HetNet in 2 resort casinos, 40 commercial buildings, hospitals and malls, and are continuing to test the seamless integration of enterprise-grade WiFi with macro and small cell technologies. Among many 5G trials and scenarios deployment, one Singapore operator has worked with Parallel Wireless—an innovative vRAN (Virtualized Radio Access Network) architecture for rural, public safety, enterprise, to trial small cells on buses. These innovative cases highlight how small cells are becoming sufficiently flexible to be deployed almost anywhere connectivity is needed.

## 8 Conclusion

This chapter presents the design and applications of future 5G wireless Micro Operators for Integrated Casinos and Entertainment (5G ICEMO) in smart cities. We propose a Concentric Value Circles (CVC) model for analysis of 5G ICEMO,

based on which we develop the business model. We develop a 5G Cloud-enabled ICEMO system architecture and a wireless network architecture for the operations of 5G ICEMO. In the illustrating case study of mega jackpot, we use the novel technology of smart contracts coupling with blockchains in 5G ICEMO with all its features in an autonomous way. In the illustrating case study of anti-counterfeiting lottery, we exploit the Small Cell features in 5G as well as the pervasive, decentralized blockchain technology to prevent counterfeiting of lottery or payout tickets and redemption of fake tickets. The final case study of autonomous transport illustrates how the characteristics of ICE make it straightforward to embrace the use of automation in the movement of people in and around the ICE environment, for both their indoor and outdoor activities. Benchmarks of the Smart Cities visions and activities of Las Vegas, Macao, and Singapore are analyzed and conveyed to validate how the proposed 5G micro operator framework can be exploited to integrated casinos and entertainment smart cities.

Future research directions may consider the dynamic spectrum sharing and access scenarios in the proposed ICEMO model to cope with the drastically changing demands of modern wireless applications in ICE. The quantitative measures or metrics like Quality of Experience (QoE) in the proposed ICEMO are not discussed. Future research may associate or discuss the qualitative values (like the core value of *fun*) with quantitative measures in the proposed Concentric Value Circles (CVC) model. In this research, the benchmarks in three distinct casino cities of Las Vegas, Macao, and Singapore provide some insights to the strategies making when ICE industries face the fierce competition but still have plenty of opportunities by embracing new technologies and innovations. Future research may extend the depth and width with more benchmarking studies to include new ICE competitors in the Philippines, Vietnams, and other new innovations in ICE.

## References

1. Agiwal M, Roy A, Saxena N (2016) Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 18(3):1617–1655
2. Ahokangas P, Moqaddamerad S, Jatinmikko M, Abouzeid A, Atkova I, Gomes JF, Iivari M (2016) Future Micro Operators Business Models in 5G. *The Business and Management Review* 7(5):143–149, June 2016
3. Albino V, Berardi U, Dangelico RM (2015) Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *Journal of Urban Technology* 22(1):3–21
4. American Gaming Association (2014) When Gaming Grows, America Gains—How Gaming Benefits America. [http://www.americangaming.org/sites/default/files/research\\_files/AGA\\_EI\\_Report\\_FINAL.pdf](http://www.americangaming.org/sites/default/files/research_files/AGA_EI_Report_FINAL.pdf). Accessed 31 July 2017
5. Andrews JG, Buzzi S, Choi W, Hanly S, Lozano A, Soong AC, Zhang JC (2014) What will 5G be? *IEEE Journal on Selected Areas in Communications* 32(6):1065–1082
6. Bharadia D, Katti S (2014) Full Duplex MIMO Radio. in the *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14)*, April 2–4, 2014, Seattle, WA, USA, 359–372. <https://www.usenix.org/system/files/conference/nsdi14/nsdi14-paper-bharadia.pdf>

7. Bhusan N, Li J, Malladi D, Gilmore R, Brenner D, Damjanovic A, Sukhvasi RT, Patel C, Geirhofer S (2014) Network Densification: the Dominant Theme for Wireless Evolution into 5G. *IEEE Communications Magazine* 52(2):82–89
8. Bian YQ, Rao D (2014) Small Cell Big Opportunities, *Global Business Consulting*. Huawei Technologies Co., Ltd., February, 2014. [http://www.huawei.com/ilink/en/download/HW\\_330984](http://www.huawei.com/ilink/en/download/HW_330984).
9. Boccardi F, Heath Jr. RW, Lozano A, Marzetta TL, Popovski P (2014) Five Disruptive Technology Directions for 5G. *IEEE Communication Magazine*: 74–80
10. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog Computing and Its Role in the Internet of Things, in Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing 13–16 ACM
11. Brown T (2009) *Change by Design—How Design Thinking Transforms Organizations and Inspires Innovation*. HarperCollins
12. Calder KE (2016) *Singapore: Smart City, Smart State*. Brooking Institution Press, Washington, D.C.
13. Casey T, Smura T, & Sorri A (2010, June). Value Network Configurations in Wireless Local Area Access. In 2010 IEEE 9th Conference on Telecommunications Internet and Media Techno Economics (CTTE):1–9.
14. Chen S, Zhao J (2014) The Requirements, Challenges, and Technologies for 5G of Terrestrial Mobile Telecommunication. *IEEE Communications Magazine* 52(5):36–43
15. Chesbrough H, Rosenbloom RS (2002) The Role of The Business Model in Capturing Value from Innovation: Evidence from Xerox Corporation’s Technology Spin-off Companies. *Industrial and Corporate Change* 11(3):529–555
16. Chih-Lin I, Rowell C, Han S, Xu Z, Li G, Pan Z (2014) Toward Green and Soft: A 5G Perspective. *IEEE Communications Magazine* 52(2):66–73
17. Chochliouros IP, Spiliopoulou AS, Kostopoulos A, Papafili I, Dardamanis A, Neokosmidis I, Rocks T, Goratti L (2017) Modern Business and Market Perspectives Coming from the Progress of the SESAME Project Effort
18. Cimmino A, Pecorella T, Fantacci R, Granelli F, Rahman TF, Sacchi C, Carlini C, Harsh P (2014) The Role of Small Cell Technology in Future Smart City Applications. *Transactions on Emerging Telecommunications Technologies* 25(1):11–20
19. Condoluci M, Sardis F, Mahmoodi T (2016) Softwarization and virtualization in 5G Networks for Smart Cities. In Internet of Things. IoT Infrastructures: Second International Summit, IoT 360° 2015, Rome, Italy, October 27–29, 2015. Revised Selected Papers, Part I, 179–186. Springer
20. Demestichas P, Georgakopoulos A, Karvounas D, Tsagkaris K, Stavroulaki V, Lu J, Xion C, Yao J (2013) 5G on the Horizon: Key Challenges for the Radio-Access Network. *IEEE Vehicular Technology Magazine* 8(3):47–53
21. Ejaz W, Naem M, Shahid A, Anpalagan A, Jo M (2017) Efficient Energy Management for the Internet of Things in Smart Cities. *IEEE Communications Magazine* 84–91
22. ETSI Industry Specification Group Mobile-edge Computing, <http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing>
23. European Telecommunications Standards Institute (ETSI) (2014) Network Functions Virtualization (NFV); Management and Orchestration. ETSI GS NFV-MN 001 V1.1.1
24. Ge X, Cheng H, Guizani M, Han T (2014) 5G Wireless Backhaul Networks: Challenges and Research Advances. *IEEE Network* 28(6):6–11
25. Government of Macao Special Administrative Region (2016) The Five-Year Development Plan of the Macao Special Administrative Region (2016–2020)
26. Han B, Gopalakrishnan V, Ji L, Lee S (2015) Network Function Virtualization: Challenges and Opportunities for Innovations. *IEEE Communications Magazine*, 53(2): 90–97
27. Hawilo H, Shami A, Mirahmadi M, Asal R (2014) NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC). *IEEE Network* 28(6):18–26

28. Hernández-Muñoz JM, Vercher JB, Muñoz L, Galache JA, Presser M, Hernández Gómez LAH, Peterson J (2011) Smart Cities at the Forefront of the Future Internet. The Future Internet Assembly 447-462, Springer, Berlin, Heidelberg
29. Infocomm Media Development Authority, Singapore (2017) Facilitating 5G Developments in Singapore, Fact Sheet, May 23, 2017
30. Jungnickel V, Manolakis K, Zirwas W, Panzner B, Braun V, Lossow M, Sternad M, Apelfröjd R, T. Svensson T (2014) The Role of Small Cells, Coordinated Multipoint, and Massive MIMO in 5G. *IEEE Communications Magazine* 52(5):44-51
31. Kamel M, Hamouda W, Youssef A, Ultra-Dense Networks: A Survey. *IEEE Communications Surveys & Tutorials* 18(4):2522-2545
32. Kibria MG, Villardi GP, Nguyen K, Liao WS, Ishizu K, Kojima F, Shared Spectrum Access Communications: A Neutral Host Micro Operator Approach. *IEEE Journal on Selected Areas in Communications* 35(8):1741-1753
33. MARKETSandMARKETS (2017) Smart Cities Market by Focus Areas, Transportation (Types, Solutions, Services), Utilities (Types, Solutions, Services), Buildings (Types, Solutions, Services), Citizen Services (Types), and Region - Global Forecast 2022. Report Code: TC 3071 <http://www.marketsandmarkets.com/Market-Reports/smart-cities-market-542.html>
34. Matinmikko M, Latva-aho M, Ahokangas P, Yrjölä S, Koivumäki T (2017) Micro Operators to Boost Local Service Delivery in 5G. *Wireless Personal Communications*. 1-14, May 2017. doi:<https://doi.org/10.1007/s11277-017-4427-5>
35. Mijumbi R, Serrat J, Gorricho JL, Bouten N, Turck FD, Boutaba R (2016) Network Function Virtualization: State-of-the-Art and Research Challenges. *IEEE Communications Surveys & Tutorials* 18(1):236-262
36. Mulligan CEA, Olsson M (2013) Architectural Implications of Smart City Business Models: An Evolutionary Perspective. *IEEE Communications Magazine* 51(6):80-85
37. Nakamura T, Nagata S, Benjebbour A, Kishiyama Y, Hai T, Xiaodong S, Ning Y, Nan L (2013) Trends in Small Cell Enhancements in LTE Advanced. *IEEE Communications Magazine* 51(2):98-105
38. Nguyen VG, Do T, Kim Y (2015) SDN and Virtualization-based LTE Mobile Network Architectures: A Comprehensive Survey. *Wireless Personal Communications* 86(3):1401-1438, August 2015
39. Osterwalder A, Pigneur Y (2010) Business Model Generation—A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, Hoboken, New Jersey
40. Palattella MR, Dohler M, Grieco A, Rizzo G, Torsner J, Engel T, Ladid L (2016) Internet of Things in the 5G Era: Enablers, Architecture and Business Models. *IEEE Journal on Selected Areas in Communications* 34(3):510-527, February 2016
41. Peng M, Li Y, Zhao Z, Wang C (2015) System Architecture and Key Technologies for 5G Heterogeneous Cloud Radio Access Networks. *IEEE Network* 29(2):6-14, March 2015
42. Peng M, Sun Y, Li X, Mao Z, Wang C (2016) Recent Advances in Cloud Radio Access Networks: System Architectures, Key Techniques, and Open Issues. *IEEE Communications Surveys & Tutorials* 18(3):2282-2308
43. Rappaport TS, Sun S, Mayzus R, Zhao H, Azar Y, Wang K, Wong GN, Schulz JK, Samimi M, Gutierrez F (2013) Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!. *IEEE Access* 1: 335-349
44. Roh W, Seol JY, Park J, Lee B, Lee J, Kim Y, Cho J, Cheun K, Aryanfar F (2014) Millimeter-wave Beamforming as An Enabling Technology for 5G Cellular Communications: Theoretical Feasibility and Prototype Results. *IEEE Communications Magazine* 52(2):106-113
45. Quek TQ, Roche G, Güvenc İ, Kountouris M (Eds.) (2013) Small Cell Networks: Deployment, PHY Techniques, and Resource Management. Cambridge University Press
46. Smart Cities Committee (2013) FTTH Smart Guide Edition 1. FTTH Council Europe
47. Schaffers H, Komninou N, Pallot M, Trousse B, Nilsson M, Oliveraie A (2011), Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation, *The Future Internet*, 431-446. <https://link.springer.com/content/pdf/10.1007/978-3-642-20898-0.pdf#page=423>.

48. Siddique U, Tabassum H, Hossain E, Kim, DI (2015) Wireless Backhauling of 5G Small Cells: Challenges and Solution Approaches, *IEEE Wireless Communications*, 22(5):22–31
49. Swan M (2015) *Blockchain: Blueprint for A New Economy*, O'Reilly
50. The 5G Infrastructure Public Private Partnership (5GPPP) SESAME: Small cEllS coodinAction for Multi-tenancy and Age services. <http://www.sesame-h2020-5g-ppp.eu/>
51. The City of Las Vegas, Nevada (2016) Proposal for Beyond Traffic: The Smart City Challenge. U.S. Department of Transportation, SAM Entity Identifier No: 4R250
52. Trivisonno R, Guerzoni R, Vaishnavi I, Soldani D (2015) SDN-based 5G Mobile Networks: Architecture, Functions, Procedures and Backward Compatibility. *Transactions on Emerging Telecommunications Technologies* 26(1):82–92
53. Tudzarov A, Gelev S (2017), Requirements for Next Generation Business Transformation and Their Implementation in 5G Architecture. *International Journal of Computer Applications* 162(2): 31–35
54. US Department of Transportation (2016) Smart city Challenge Lesson Learned. <http://www.transportation.gov/smartcity>
55. Verganti R (2009) *Design-driven Innovation—Changing the Rules of Competition by Radically Innovating What Things Mean*. Harvard Business Press
56. Wirtz BW, Schilke O, Ulrich S (2010) Strategic Development of Business Models: Implications of the Web 2.0 for Creating Value on the Internet. *Long Range Planning* 43(2):272–290
57. Wood T, Ramakrishnan K, Hwang J, Liu G, Zhang W (2015) Toward A Software-based Network: Integrating Software-defined Networking and Network Function Virtualization. *IEEE Network* 29:36–41, May 2015
58. Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M (2014), Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1(1), 22-32, February 2014
59. Zhang Z, Chai X, Long K, Vasilakos AV, Hanzo L (2015), Full Duplex Techniques for 5G Networks: Self-interference Cancellation, Protocol Design, and Relay Selection. *IEEE Communications Magazine* 53(5):128-137

# An IoT-Based Urban Infrastructure System for Smart Cities



**Edna Iliana Tamariz-Flores, Kevin Abid García-Juárez,  
Richard Torrealba-Meléndez, Jesús Manuel Muñoz-Pacheco,  
and Miguel Ángel León-Chávez**

**Abstract** This chapter reports a prototype implementation of an IoT-based urban infrastructure system for smart cities. More specifically, we propose a real-time sensor network that uses the IEEE 802.15.4 standard for monitoring the available spaces in three car parking areas at City University of Autonomous University of Puebla in Mexico. The first step consists of implementing a local server and a database over a Galileo2 development board from Intel using Linux. Then, a wireless sensor network is implemented using SHARP proximity sensors and XBee modules with the 802.15.4 standard to 2.4 GHz for WSN training. The data, in frame format, are sent to the Intel Galileo2 development board, in which free software is used to install a suitable server and database for the frames of nodes, which represent each car park. Finally, a web page, which is integrated into the same Galileo board, is designed for monitoring the automatic updating of the data from any device that belongs to an external network. In this way, the architecture of the IoT is analyzed by considering 3 layers, namely, Application, Network and Perception, and a 5-layer model: Business, Application, Service Management, Object Abstraction and Objects. This analysis is performed with the purpose of using the compatibility among multiple technologies to fulfill the IoT objective. This research will not only help analyze and validate theoretical results but also enable IoT applications in real-world case studies in developing countries by using affordable hardware and free software.

**Keywords** Smart parking · IoT · Sensors · IEEE 802.15.4

---

E. I. Tamariz-Flores (✉) · K. A. García-Juárez · M. Á. León-Chávez  
Faculty of Computational Sciences, Autonomous University of Puebla, Puebla, Mexico  
e-mail: [iliana.tamariz@correo.buap.mx](mailto:iliana.tamariz@correo.buap.mx)

R. Torrealba-Meléndez · J. M. Muñoz-Pacheco  
Faculty of Electronics Sciences, Autonomous University of Puebla, Puebla, Mexico

## 1 Introduction

The principal definition of the Internet of things (IoT) [1] was proposed by Kevin Ashton in 1999 during a presentation in which he argued that by associating physical objects with RFID tags, each object could be given an identity for generating data on such things. The IERC is actively involved in the ITU\_T Study Group 13, which leads the work of the ITU on standards for next-generation networks (NGNs) and future networks and has been part of the team that formulated the following definition [2]: *“Internet of things (IoT): A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies”*. The IERC definition [3] states that the IoT is *“A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual “things” have identities, physical attributes and virtual personalities, use intelligent interfaces and are seamlessly integrated into the information network”*. In general, the IoT is the technological environment in which everyday objects are connected to the Internet and able to receive, generate and send information.

With the continued development of the Internet, it is anticipated that additional potential will be realized by a combination with related technological approaches and concepts, although they are not entirely new ideas, such as cloud computing, future Internet, big data, robotics and semantic technologies. Taking into account the growth in mobile devices to the present day, the IoT arose between 2008 and 2009 according to the IBSG Cisco [4] and each individual on earth will have more than six devices connected to the Internet by 2020.

There are many applications for the IoT as Perera et al. [4] proposed; the popular existing IoT solutions in the marketplace have been classified into five different categories: smart wearable, smart home, smart city, smart environment and smart enterprise. In this chapter, we present an application for the urban infrastructure that is based on the smart city framework, with the aim of detecting vehicle traffic during car parking to inform users about space availability.

Much of the IoT's success is due to RFID technology, which is used to tag and track objects, people and animals as Kortuem et al. [5] described, and along with other technologies for the IoT as the authors in [6, 7] presented, such as sensor networks, M2M, and mobile Internet, it can be grouped into three categories: (i) technologies that enable “things” to acquire contextual information, (ii) technologies that enable “things” to process contextual information, and (iii) technologies that improve security and privacy. The first two categories can be jointly understood as functional building blocks that are required for building “intelligence” into “things”, which are the features that differentiate the IoT from the standard Internet. The third category, although not functional, is a requirement for promoting the penetration of the IoT.

Updating on what is occurring around us can transform the way we perform everyday activities by giving applications current and detailed knowledge about

physical events. To make optimum and functional applications focused on the IoT standards and protocols, it is necessary for application programming interfaces (APIs), which are available to the developers, to communicate with one another as Welbourne et al. [8] described. Currently, there is no single or specific IoT standard [8]. However, several IoT organizations and standards have been proposed to facilitate and simplify application programming and provide better service. Among the organizations of companies are Allseen Alliance, Wize Alliance, Thread, and Industrial Internet Consortium. The groups that are in charge of providing protocols are the World Wide Web Consortium (W3C), IETF, EPCglobal, IEEE and ETSI [6, 9].

Works that are related to smart parking are presented as follows: authors in [10–14] presented an algorithm for controlling traffic congestion, where it defines multiple applications, such as finding a free parking space and using an alternative localization technique for smart parking that allows users to rapidly find a free parking spot. Authors in [15–17] presented applications for a car parking system; those articles propose a model that can regulate and manage the number of cars that can be parked in a given area using sensing devices, a smartphone application for detecting cars and calculating the number of available parking spots, and an innovative system in which a driver can make a reservation using a smart phone or tablet 30 min prior to parking and the service platform will book a parking spot using a vehicle ID. Many classifications of parking systems, algorithms, and techniques for smart parking have been proposed with the aim of solving parking problems as the authors in [18–20] proposed. In addition, the authors in [21, 22] evaluated and compared many protocols of the application layer and proposed a system that satisfies GSI global standards for smart parking. Later, the authors in [23, 24] presented a smart parking system that is based on the integration of UHF, RFID and IEEE 802.15.4 WSN technologies.

In that scenario, the main contribution of this chapter is the analysis of an IoT structure for monitoring car parking at City University of Autonomous University of Puebla, where the implementation of the prototype was carried out using SHARP proximity sensors and XBee modules with the 802.15.4 standard to 2.4 GHz for WSN training (Digi XBee-PRO S1, USA). The data, in frame format, are sent to the Intel Galileo Gen 2 development board, in which free software is used to install a suitable server and database for the frames of nodes, which represent each parking spot. Finally, in a web page that is designed on the same card, the user can monitor the automatic updating of these data from any device that belongs to an external network. In this way, the steps were established while taking into account the elements and architecture of the IoT.

The chapter is organized as follows: Sect. 2 introduces the IoT architecture by introducing the main definitions and various scenarios. Sect. 3 presents a smart parking application scenario. This section also discusses the IoT data trajectory in the different layer models of a node. The interconnection of different technologies that are used in this implementation is also discussed in this section. Sect. 4 presents the implementation of the urban infrastructure, hardware devices and network technologies. Sect. 5 presents the proposed methodology and results of



the implementation. The results that are presented in this section explain various operational features and the layers in the IoT system. Conclusions and future work are discussed in Sect. 6.

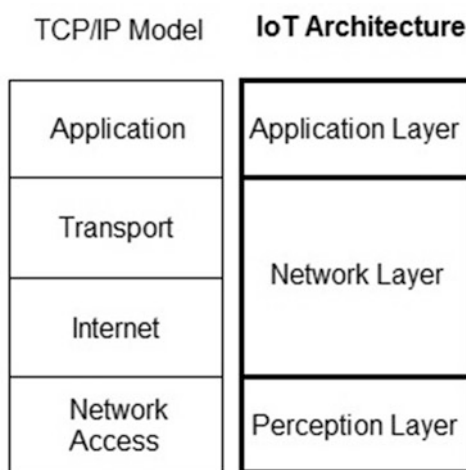
## 2 The IoT Architecture

The Transmission Control Protocol/Internet Protocol (TCP/IP) model uses four layers, namely, Application, Transport, Internet and Network Interface, which define the operation of the Internet because each layer depends on the top or bottom layer, as appropriate. In addition, each layer handles its own protocols and standards to realize satisfactory performance of the network.

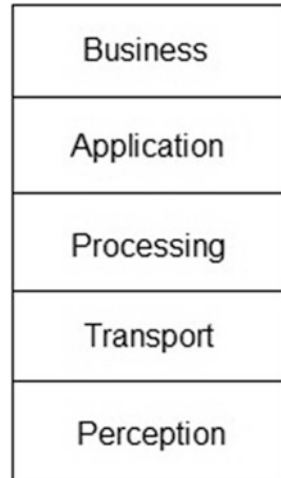
According to the definition of the IoT and taking into account the TCP/IP model, the general architecture of a basic 3-layer model [25, 26] is composed of the perception layer, network layer and application layer. From the point of view of biological systems, the Perception Layer could represent the five sensory organs of the IoT, similar to the human body, because this layer recognizes things and collects information. Therefore, the Perception Layer can include RFID tags and reader-writers, cameras, GPS, sensors, terminals, and sensor networks, among others. The Network Layer is analogous to the brain of the IoT. Its main function is transmitting and processing information that is obtained by the Perception Layer. Later, the Application Layer could emulate social interaction to demand solutions. In this manner, the dependence of these layers on the model can share the information with the community.

The relationship between the TCP/IP model and the architecture of the IoT is shown below (see Fig. 1).

**Fig. 1** Layers of the TCP/IP model against the IoT architecture



**Fig. 2** The new IoT architecture



The Transport Layer of the TCP/IP model disappears in the IoT architecture due to the use of devices such as sensors that do not allow the application of the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). In the case of the TCP, the sensors do not establish a session in the sensory network, nor do they ensure that the data arrive reliably to the user. The objective of this layer is to facilitate the final communication on the internetwork, and because of that, it was established as the main problem in the perception of things in the IoT architecture.

According to the presented models, the lack of a Transport Layer in the IoT architecture and the progress that has been made in IoT applications, a new model [7, 25] was established, which is based on 5 layers, namely, the Business Layer, the Application Layer, the Processing Layer, the Transport layer and the Perception Layer, as shown in Fig. 2.

This new model defines another name for the Network Layer, namely, the Transport layer, because in addition to transmitting data that are received from the Perception Layer to the processing center through various networks, such as Wireless or cable networks or even the enterprise local area network (LAN), this layer meets the goal of transport.

The two layers that were added to the new model were the Processing Layer and the Business Layer. The Processing Layer was derived from the Network Layer and has the functions of storing, analyzing and processing the information of objects that are received from the Transport Layer. Cloud computing and ubiquitous computing are the primary technologies in this layer. The Business Layer manages the Internet of Things. The objective of this layer is to ensure the effectiveness and the viability of the business in the long term.

### 3 Smart Parking Application Scenario

The term smart city is used to define a city with connectivity that can exchange information based on various technologies such as WSN for the IoT [27–30] and, thus, achieve the interconnection among various smart city objects to offer citizens the information that they need to improve their quality of life [31].

According to analyses of the application and use of technologies, by 2020 we will see the development of mega city corridors and networked, integrated and branded cities [32] that support various transmission technologies, such as Bluetooth, ZigBee, WLAN, and millimeter-wave and even visible light communication, on the level of heterogeneity of, for example, a cellular network [33]. Taking into account the connectivity between applications and smart devices, it is important to design new architectures and mechanisms for the IoT to provide reliability in communications [34]. For that reason, authors in [35, 36] identified several security vulnerabilities and privacy issues in the smart city.

In the smart city, applications are classified according to the type of network, scalability, coverage, flexibility, heterogeneity, repeatability, and end-user involvements. In general, these applications can be grouped into personal, home, utilities, mobile, and enterprises [37]. Several applications and services for the IoT that have already been implemented to support smart urban mobility are considered: traffic monitoring, smart parking and smart traffic lights [38].

With the introduction of the IoT, a city will act more like a living organism and will be able to respond to citizens' needs. In this context, there are numerous important research challenges for smart city IoT applications. For example, authors in [39–41] provided case studies for improving the traffic in a city.

In recent years, growth in the number of vehicles at City University in BUAP Mexico has led to the creation of new parking spaces. BUAP is considered a macro university because it has approximately ninety thousand enrolled students. In addition, several events are regularly carried out in the Seminary building, which is close to the Central Library and the Children's Circle, which is similar to a nursery school. These events provoke traffic jams in the parking spots of the largest parking area in BUAP. Therefore, it is necessary to determine the number of available places for better control and security. In this prototype, the higher-capacity parking areas in the university were considered, which were defined as P1, P2 and P3.

In that framework, this chapter reports a prototype implementation of an IoT-based urban infrastructure system for smart cities. More specifically, we propose a real-time sensor network that uses IEEE 802.15.4 for monitoring the available spaces in three car parking areas at City University of Autonomous University of Puebla in México. The first step consists of implementing a local server and a database over a Galileo 2 development board from Intel using Linux. Then, a wireless sensor network is implemented using SHARP proximity sensors and XBee modules with the 802.15.4 standard to 2.4 GHz for WSN training. The data, in frame format, are sent to the Intel Galileo 2 development board, in which free software is

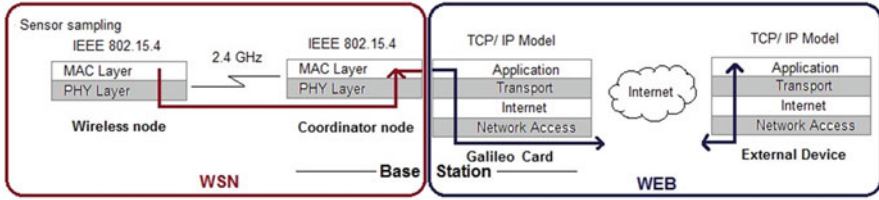


Fig. 3 The IoT data trajectory in the different layer models

used to install a suitable server and database for the frames of nodes that represent each parking spot. Finally, a web page, which is integrated into the same Galileo card, is designed to monitor the automatic updating of the data from any device that belongs to an external network.

Currently, we are developing a complete architecture of the IoT that is based on multiple technologies and applying it to various environments (see Fig. 3). We use IEEE 802.15.4 and 2.4 ISM bands to transmit signals from sensors on the car parking spaces. The Ethernet network that is based on the TCP/IP model was selected to interface our base station (BS) to the Internet.

The network model in this prototype is made up of two parts: the WSN, where the sensor data are sampled, and the Internet, over which any external device can access the server. Figure 3 shows the crossing of the sensor data in the WSN and the query of an external device to the server. In the figure, to present a more detailed diagram of the models, only one node is shown. The collected data are sent to the coordinating node using the 2.4 GHz frequency. The coordinator is connected to the Intel Galileo board, whose connection represents the base station. This is where the other part of the network is considered, since any external device can consult the website.

According to the IoT model, a comparison was made with this implemented model. The layers of the standard 802.15.4, the Physical layer and the MAC Layer only transmit the data transmission that are obtained by the sensors in a point-to-point connection by the 2.4 GHz wireless link. Thus, they represent the Perception and Transport Layers of the model of the IoT. When the data arrive at the base station, they are processed to locate the car parking or origin place and stored in the corresponding database. Thus, this represents the Processing Layer of the IoT model. The Application Layer is represented by the graphical interface, which indicates the available places in each parking area and determines which parking place is best for the user. Finally, the Business Layer must be resolved with the corresponding authorities for its implementation in the university.

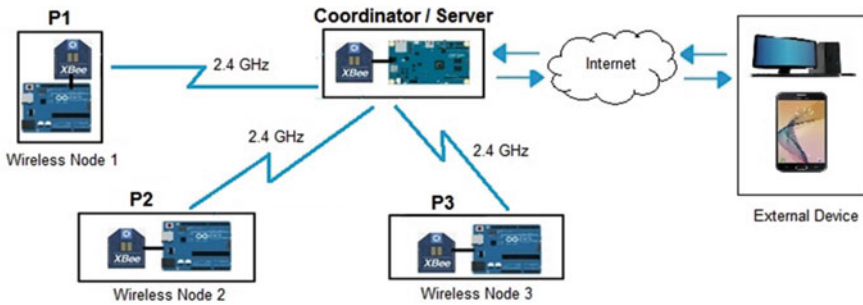


Fig. 4 The IoT implementation scheme

## 4 The Implementation of Urban Infrastructure, Hardware Devices and Network Technologies

An urban infrastructure system that was designed for smart city applications is presented in this section. Electronic hardware and software are developed for the scenario that is shown in Fig. 4. For the implementation, the WSN in this work was based on a network of wireless modules in the IEEE 802.15.4 standard [42–46], in which distance sensors were connected. These modules are capable of forming ad hoc networks without a pre-established physical infrastructure or central administration. The Sharp 2Y0A21 sensors are used herein to perform the data sampling.

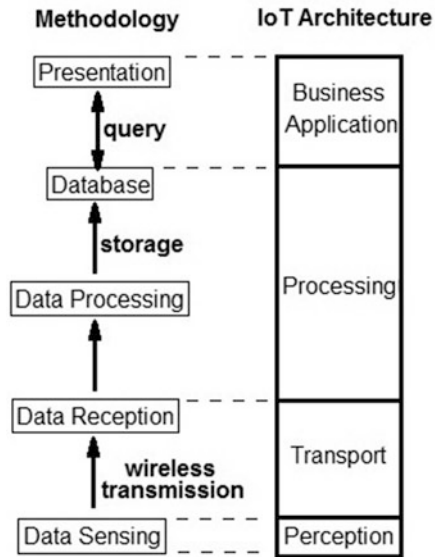
Then, the base station is represented, in addition to the coordinator node, by the connection to the Intel Galileo Gen 2 development board. In this board, the server was configured. The external device can be any device that can access the IP address that is set on the server.

The XBee modules will collect data from all sensors in the WSN. The processing and storage of these data are realized on the BS, i.e., the board, which is capable of displaying all received data on a graphical interface and storing all data in the database system.

## 5 Methodology for the IoT Oriented to Smart Parking

The proposed methodology allows us to obtain information about the available places for an external user over the Internet. Figure 5 shows the operation diagram of the proposed prototype and describes each of the stages. The layers that involve the IoT are highlighted and related to the development stages.

**Fig. 5** Methodology diagram for a smart parking scenario



According to [9, 47], it is necessary to establish the building blocks of the IoT. A brief description of each of the elements that are used for this project is shown in Table 1.

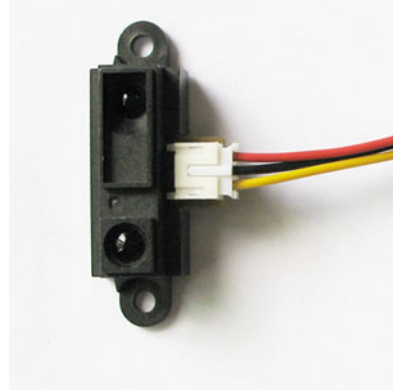
### 5.1 Data Sensing

Sensor nodes are designed to collect signals from each parking slot and detect the entry or exit of an automobile in the range in which these sensors were characterized. A sensor node performs three tasks: detecting signal via end-to-end sensing, digitizing/coding/controlling for a multi-access communication and wireless transmission via a radio transceiver technology. In addition to data acquisition and processing, the microcontroller maintains a power management scheme for controlling the distribution of the energy from the battery in an optimized manner. It should be programmed to turn battery connections OFF for blocks that are not operating (i.e., during sleep mode). An analog-to-digital conversion (ADC) stage is necessary for digital processing. Then, this signal is processed and transmitted over the air via an XBee 802.15.4 module.

*Sensor characterization* The sensors that were selected for the development were the Sharp 2Y0A21 sensors [48] (see Fig. 6). These infrared-light distance sensors are widely used in works in which high-accuracy distance measurements are required.

**Table 1** The IoT elements

IoT element	Description	Implementation
Identification	Identification methods are used to provide a clear identity for each object within the network. Each object's ID refers to its name, and the object's address refers to its address within the communication network.	Naming Parking slots
Sensing	The IoT sensing element collects data from objects within the network and sends them to a database.	Addressing IPv4
Communication	The IoT communication technologies connect objects within the network to establish specific smart services.	Sensor SHARP and RFID tag RFID and IEEE 802.15.4
Computation	The computation is the main part of the IoT. Processing and storage in the database are performed in processing units and software applications.	Hardware Arduino UNO, Intel Galileo Gen2
Services	The IoT services can be categorized into four classes (+): identity-related services, information aggregation services, collaborative-aware services and ubiquitous services.	Software Linux Debian OS Information aggregation

**Fig. 6** Sharp sensor 2Y0A21

The sensor has 3 pins: power, ground and output. The output voltage varies between 0.3 and 3.1 volts depending on the measured distance, according to the curve that is provided by the manufacturer. The sensor power must be between 4.5 and 5.5 volts and it is recommended to be as stable as possible. The equation for converting the analog output of the Sharp sensor to a distance between 10 and 80 centimeters is shown in Eq. 1.

$$Distance (cm) = \frac{4800}{V_{out} - 20} \quad (1)$$

This equation was used in the Arduino UNO to process the data that are delivered by the sensor as binary values, where '1' is the input and output of a parking car and the value is '0' otherwise. In this way, the infrared sensors were connected to the XBee end nodes. Thus, the data that were collected by these sensors were analyzed through the Arduino UNO, which served as a support for characterizing the data that were received from the sensor and transmitting them to the XBee Coordinator module. This was done by creating a serial port in the Arduino with the library "SoftwareSerial.h", in which pins 10 and 11 served as the basis for the correct operation since they correspond to the TX (transmitter) and RX (receiver), respectively.

The Arduino sends to the Coordinator module a flag, which could be 1, 3 or 5 for the inputs and 0, 2 or 4 for the outputs. The flag is interpreted by the server, which handles its processing and indicates whether the data are entry or exit data, which was of great importance for this project. In addition, the date and time of one of these input or output changes were added.



## 5.2 Receiving Data

The data reception block is an important block since it evaluates the proper reception of data in the receiver.

*Evaluation of the network* For the evaluation of the implemented network, the XBee modules are configured according the IEEE 802.15.4 standard to operate as the final node, router and coordinator. Then, the transmission of the data was performed in the transparent or serial mode and wireless mode.

The functions of the IEEE 802.15.4 modules within the WSN [49] are described as follows:

- Coordinator. The coordinator is responsible for forming the network, delivering addresses and managing the functions that define the network. There is only one per network.
- Router. This function enables the joining of existing networks, sending of information, and receiving of information, and handles the routing of information. There may be more than one router or even no router.
- Final device. This device can only send or receive information within the network and can switch to sleep mode when not in use, which helps conserve energy.

The configurations of RF modules and the tests that were performed in the network were realized by using the XCTU software, which is a free cross-platform for connecting RF Digi modules. By setting a star topology network, the final node only sent data to the coordinator in the wireless mode. This data structure, unlike the serial mode, includes important fields such as the physical address.

As an example, we send a broadcast-type transmission (see Fig. 7) whose destination physical address corresponds to “FF FF FF FF FF FF FF FF” and logical address to “FF FE”. Therefore, all nodes that belong to that network received the data, including the coordinator. We repeat the transmission for unicast and broadcast transmissions to determine whether the coordinator receives the proper data from the corresponding final node.

The graphical interface shows if a frame arrived correctly or incorrectly to the transmitter node (see Fig. 8), where ID represents the package number, Time is the time of package arrival, Length represents the frame size and Frame indicates the type of frame that is received, which, in this case, is a received packet. On the right side, there is a panel that shows the details of the frame. A received packet results in a layer 2 frame; the important fields are shown below.

The data structure is given in Table 2. The delimiter is a special sequence of bits that indicates the beginning of a data frame and its value always corresponds to 0x7E, which enables simple detection of incoming frames. The next field is Length, which specifies the total number of bytes that are included in the data field of the frame. MSB represents the most significant bit and LSB the least significant bit. Subsequently, the data field is divided into two sub-fields: the frame type and the

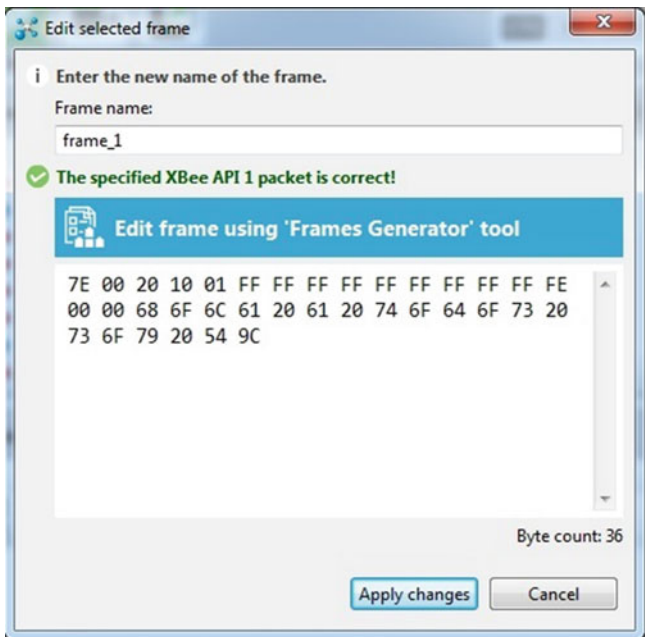


Fig. 7 Tool for generating a frame of XCTU

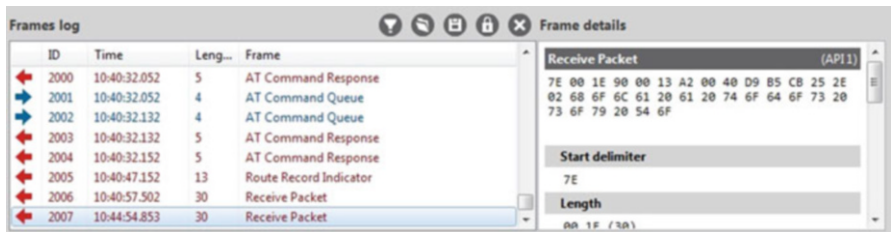


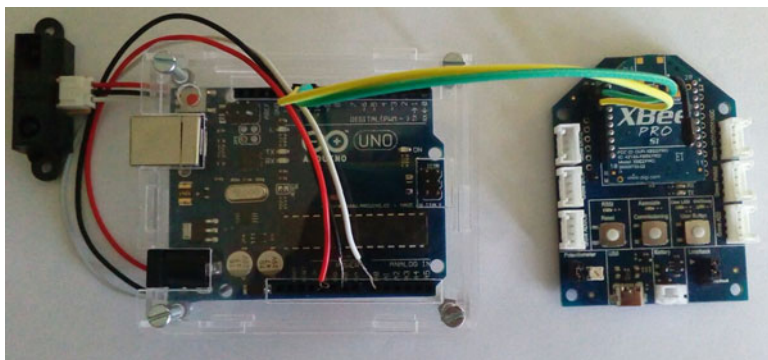
Fig. 8 Detailed view of frames in the graphic interface of XCTU

Table 2 Frame fields

Delimiter		Length		Frame data					Checksum		
		Type	Data								
1	2	3	4	5	6	7	8	9	...	n	n + 1
0X7E	MSB	LSB		Data						Simple byte	

data type. The Checksum field is the last byte and helps check the integrity of the data.

The analysis of this frame based on the standard IEEE 802.15.4 stack, which is presented in Fig. 4, was performed to justify its structure within this project and thus justify the analysis of the performed tests.



**Fig. 9** Wireless node

The Physical layer and the MAC layer provided the services of point-to-point transmission over air. Because the standard works on the ISM bands, we worked on the 2.4 GHz frequency and achieved transmission rates of up to 250 kbps. The tests were performed indoors over a distance of 45 m without considering walls and up to 20 m with obstacles, all without causing transmission errors. The MAC layer adds the physical layer device address, which corresponds to a specific field in the frame.

*Implementation of the WSN* The WSN is characterized by its easy deployment and self-configuration, which allows it to operate as an emitter and a receiver and provide routing between nodes without a direct line of sight. In addition, the energy efficient management gives it a high degree of autonomy [49, 50]. A WSN consists of wireless nodes, a gateway, and a base station.

The wireless nodes in this work are composed of an XBee wireless module with 802.15.4, an Arduino UNO and the distance sensor. This node acquired, processed and transmitted data from the environment to the receiver (see Fig. 9).

The wireless node was set as the final device in the implementation. Following the structure of the WSN, the gateway is an essential element as it allows the interconnection between the WSN and a data network (TCP/IP). The base station is the collector of the data that are obtained by the network sensors, which is implemented in this work by the Galileo Gen 2 board and an XBee wireless module (see Fig. 10).

In a normal structure, the data are routed to a server computer within a database, which users can access remotely to monitor and analyze the collected data [49].

According to the operation modes of 802.15.4 modules, the module that is connected to the Intel development board is in the coordinator node. In this way, the card was configured to recognize the wireless data that were coming from the coordinator so that these data were processed by the server and addressed to the corresponding parking database.

The implemented topology was a star-type topology in which the coordinating node is in the center and is connected to a circle of final devices. Each message



**Fig. 10** Base station

must pass through the coordinating node, which routes them as necessary. The end devices do not communicate with each other.

*Wireless transmission* After presenting the frame fields in Table 2 of this chapter, the following was used to test messages from the XCTU platform, where now the coordinator module was connected to the Intel Galileo Gen 2 board. Therefore, all received data from the three final nodes had to be displayed on the development board (see Fig. 11). The evaluated message was EDNA and the receive frame starts with bytes **7E**, which represent the frame delimiter, as discussed above; the values **00 09** represent the length of the frame; and **81** is the frame type, which indicates that the device received an RF packet. The values **ee 00** refer to the logical address, and **24** is an optional byte, while values **00 45 44 4E 41** are the data that represent the EDNA message in its character representation. Finally, value **54** corresponds to the checksum field, which helped verify that the frame arrived correctly as follows: all hexadecimal digits were added and the result in the last two digits had to be equal to **FF**. The complete operation is shown in Eq. 2.

$$81 + EE + 00 + 24 + 00 + 45 + 44 + 4E + 41 + 54 = 2FF \tag{2}$$

In this way, it was verified that the WSN worked correctly because the transmitting node of the network was identified and that the data sent was correct. In the wireless transmission part, a time interval for the sending of data was configured to

```
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
Rx: 7e 00 09 81 ee 00 24 00 45 44 4e 41 54
Msg: EDNA
```

Fig. 11 Reception frames in Galileo

define a synchronization between the nodes so that they could transmit and avoid collisions during the reception process.

### 5.3 Data Processing

In the data processing operation, the data that arrived at reception stage were analyzed to locate the corresponding parking lot. Each parking lot could have only two flags: one corresponds to an exit and the other to an entry. This helped sort the data and determine to which database table the data should be saved. In this way, the available places were updated. The libraries that were used in this part of the server were **sqlite3** for the management of the database, **serial** to obtain the data that were received by the coordinator of the WSN and **datetime** for the handling of time.

All received data were analyzed and sent to the corresponding input and output DBs on the server (see Fig. 12). The relevant data that are displayed on the Galileo board are the status, date and time; the flags are only used internally for processing.

```
('exit', '06/05/17/22:18:06')
('entrance', '06/05/17/22:18:19')
('entrance', '06/05/17/22:18:26')
('entrance', '06/05/17/22:18:28')
('exit', '06/05/17/22:18:35')
('exit', '06/05/17/22:18:37')
('entrance', '06/05/17/22:18:39')
('entrance', '06/05/17/22:18:45')
('entrance', '06/05/17/22:18:53')
('entrance', '06/05/17/22:18:55')
('entrance', '06/05/17/22:18:58')
('entrance', '06/05/17/22:19:00')
('exit', '06/05/17/22:19:03')
('exit', '06/05/17/22:19:05')
('exit', '06/05/17/22:19:06')
('exit', '06/05/17/22:19:07')
```

Fig. 12 Registration of inputs and outputs

## 5.4 Database

According to the functions of the processing layer of the IoT layer model, the server was divided into two parts: the database and the data processing part. The programming was performed in Python, as it has efficient and high-level data structures for object-oriented programming [51]. Python was used to control the components of the Intel Galileo Gen 2 board; version 2.7.3 was used.

SQLite is an open-source relational database (DB) and was designed to provide a convenient way for applications to manage data without the overhead that often comes with dedicated management system relational databases. This DB can be used on websites, operating system services, scripts and applications. It can also be considered a digital conduit, which provides easy realization of the link between applications and data [52].

SQLite is considered a useful tool for system administration because it is compact, small and elegant as a UNIX utility. SQLite works best when used with scripting languages such as Perl, Python and Ruby. It should be noted that SQLite DBs are common operating system files. Therefore, it is easy to work with them and they can easily transport the DB and be backed up, which makes it possible for the DB to be carried in a USB.

SQLite was added as part of the PHP 5 standard [52] and the important functions to apply for this DB are as follows: the Insert function adds records to the database, while the Update command allows the fields of a record in the database to be updated

	id	Parking	Available	Capacity
	Filter	Filter	Filter	Filter
1	1	P1	400	437
2	2	P2	340	356
3	3	P3	60	85

**Fig. 13** DB of available parking places

and the Delete function deletes items from the database. A single element or several elements can be deleted, which will depend on the parameters that are used. Finally, the Query function allows the user to query the database, either to display a single element of a table or a whole table in the database.

According to the instructions that were applied in this work, the data were stored in the database (see Fig. 13). According to the arrival of those data from the sensors, they identify from which parking they come and are inserted into the database. This DB consists of 4 tables: one of them (see Fig. 13) corresponds to the places that are available in the 3 parking lots and only has 3 registers. The other 3 tables correspond to each of the parking lots and store the date, time and type of data as input or output; these tables are used for statistical purposes.

## 5.5 Presentation

The Intel Galileo board can be used as a client or server, depending on the project requirements, as it is compatible with other Intel boards. The Galileo board is a free hardware. Therefore, the schemes are available to the public. Additionally, it can operate as an Arduino or as a machine with the Linux operating system [53].

The Intel Quark System on a Chip (SoC) X100 is based on the x86 architecture, which makes it possible to run an S.O. of 32 and 64 bits. This processor is designed for applications with low power consumption and high performance, which is useful when developing the IoT. In addition, the processor runs at 400 MHz, is compatible with Pentium instructions and has 16 KB of shared instructions and an L1 data cache, which increases the speed of code execution and minimizes the number of times the CPU needs to access the external memory. Finally, it has an on-chip DDR3 memory controller that facilitates access and management of external DRAM. Intel Quark SoC supports a wide variety of external memory interfaces, including micro SD, SDIO, and eMMC. The operating system that was selected is a Linux Debian distribution that was modified for the Intel board, which uses a Linux 3.8.7 kernel. It is the best option for the application development because of the free software features.

**Fig. 14** Website on a smart device



In the presentation stage, the Apache server was used, which handles the webpage of the prototype. Additionally, it has installed utilities that support the management of SQLite as a database, since its characteristics are the most efficient for this prototype. For instance, its small size reduces memory allocation because the Apache server that is installed on the Galileo Gen 2 board consumes more memory. The data are presented on a web page (see Fig. 14), which was developed with PHP and HTML. In addition, the responsive design was added so that the page can adapt to any screen size (Tablet, Smartphone, PC or Laptop) and thus maintain an aesthetic design.

The initial test was performed at the Network Laboratory of the Faculty of Computer Science of BUAP to test the connectivity of a public IP address to the Internet from an external network.



In these results, an important point is that the last two octets of the address were eliminated due to the security concerns and the responsibilities that are involved in handling the granted permits. However, it is necessary to mention that it was not possible to install the DNS, which would have completed this work better, due to compliance issues with the Linux version.

## 6 Conclusions

In this chapter, a new IoT Layer Model was introduced for the detection of available places in a car park for a smart city scenario. To develop an efficient system for the IoT, it was necessary to use advanced design concepts to obtain related hardware, software, protocols and standards. Therefore, the design methodology that was presented in this chapter can be used as the basis for any application design. Additionally, this chapter presented a structure that was defined according to an IoT layer model that was related to the operation of the Internet and WSN, in contrast to that of previous works.

Further work will be necessary to develop multimedia networks with better structures and longer lives, which could use different and innovative technologies to support advanced smart city systems.

## References

1. Weber, RH (2009) Internet of Things - Need for a New Legal Environment?. *Computer Law & Security Review*, Elsevier 25, pp. 522–527
2. Atzori L, Iera A, Morabito G (2010) The Internet of Things: A survey. *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805
3. Zhu C, Leung VCM, Shu L, Ngai ECH (2014) Green Internet of Things for Smart World. *IEEE Access*, vol. 3, pp. 2151–2162
4. Perera C, Liu CH, Jayawardena S, Chen M (2014) A Survey on Internet of Things From Industrial Market Perspective. In: *IEEE Access*, vol. 2, pp. 1660–1679
5. Kortuem G, Kawsar F, Fitton D, Sundramoorthy V (2010) Smart objects as building blocks for the Internet of Things. *IEEE Internet Comput.*, vol. 14, no. 1, pp. 44–51
6. Vermesan O, Friess P (2014) *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. Denmark: River Publishers
7. Bassi A, Bauer M, Fiedler M, Kramp T, van Kranenburg R, Lange S, Meissner S (2016) *Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model*. 1st ed., Springer Publishing Company, Incorporated.
8. Welbourne E, et al. (2009) Building the Internet of Things using RFID: The RFID ecosystem experience. In: *IEEE Internet Comput.*, vol. 13, no. 3, pp. 48–55
9. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. In: *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376
10. Pham TN, Tsai MF, Nguyen DB, Dow CR, Deng DJ (2015) A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies. In: *IEEE Access*, vol. 3, pp. 1581–1591

11. Balzano W, Vitale F (2017) DiG-Park: A Smart Parking Availability Searching Method Using V2V/V2I and DGP-Class Problem. In: The 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, pp. 698-703
12. Tsaramirsis G, Karamitsos I, Apostolopoulos C (2016) Smart parking: An IoT application for smart city. In: The third International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pp. 1412–1416
13. Roy A, Siddiquee J, Datta A, Poddar P, Ganguly G, Bhattacharjee A (2016) Smart traffic & parking management using IoT. In: IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, pp. 1–3
14. Shih C, Liang Z (2017) The development and simulation of a smart parking guidance system. In: International Conference on Applied System Innovation (ICASI), Sapporo, pp. 1736-1739
15. Bonde DJ, Shende RS, Kedari AS, Gaikwad KS, Bhokre AU (2014) Automated car parking system commanded by Android application. In: International Conference on Computer Communication and Informatics, Coimbatore, pp. 1–4
16. Zadeh NRN, Cruz JCD (2016) Smart urban parking detection system. In: The 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Batu Ferringhi, pp. 370–373
17. Hsu CW, Min Huai Shih, Hou Yu Huang, Yu Chi Shiu, Shih Chieh Huang (2012) Verification of smart guiding system to search for parking space via DSRC communication. In: The 12th International Conference on ITS Telecommunications, Taipei, pp. 77–81
18. Lin T, Rivano H, Le Mouël F (2017) A Survey of Smart Parking Solutions. In: IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99, pp. 1-25
19. Kotb AO, Shen YC, Huang Y (2017) Smart Parking Guidance, Monitoring and Reservations: A Review. In: IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 2, pp. 6–16
20. El-Seoud SA, El-Sofany H, Taj-Eddine I (2016) Towards the development of smart parking system using mobile and web technologies. In: The International Conference on Interactive Mobile Communication, Technologies and Learning (IMCL), San Diego, CA, pp. 10–16
21. Kayal P, Perros H (2017) A comparison of IoT application layer protocols through a smart parking implementation. In: The 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, pp. 331–336
22. Pham N, Hassan M, Nguyen HM, Kim D (2017) GS1 Global Smart Parking System: One Architecture to Unify Them All. In: IEEE International Conference on Services Computing (SCC), Honolulu, HI, USA, pp. 479–482
23. Alam M, Fernandes B, Almeida J, Ferreira J, Fonseca J (2016) Integration of smart parking in distributed ITS architecture. In: The International Conference on Open Source Systems & Technologies (ICOSST), Lahore, 2016, pp. 84–88
24. Liu Y, Zou X, Shi M, Zhuang G (2012) Intelligent parking guidance system based on wireless sensor networks. In: IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, Hangzhou, pp. 1076-1078
25. Wu M, Lu T, Ling F, Sun J, Du H (2010) Research on the architecture of Internet of Things. In: The 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, pp. V5-484-V5-487
26. Yang Z, Yue Y, Yang Y, Peng Y, Wang X, Liu W (2011) Study and application on the architecture and key technologies for IoT. In: International Conference on Multimedia Technology, Hangzhou, pp. 747–751
27. Dhiviya S, Sariga A, Sujatha P (2017) Survey on WSN Using Clustering. In: The Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), Tindivanam, Tamilnadu, India, pp. 121–125
28. Ali H, Chew WY, Khan F, Weller SR (2017) Design and implementation of an IoT assisted real-time ZigBee mesh WSN based AMR system for deployment in smart cities. In: IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, pp. 264–270

29. Faye S, Chaudet C (2015) Connectivity analysis of wireless sensor networks deployments in smart cities. In: *The IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Luxembourg City, pp. 1-6
30. Keramidas G, Voros N, Hübner M (2017) *Components and Services for IoT Platforms*. 1st ed. Springer Publishing Company, Incorporated
31. Yaqoob I, Hashem IAT, Mehmood Y, Gani A, Mokhtar S, Guizani S (2017) Enabling Communication Technologies for Smart Cities. In: *IEEE Communications Magazine*, vol. 55, no. 1, pp. 112–120
32. Vermesan O, Friess P (2014) *Internet of Things – From Research and Innovation to Market Deployment*. Denmark: River Publishers
33. Han T, Ge X, Wang L, Kwak KS, Han Y, Liu X (2017) 5G Converged Cell-Less Communications in Smart Cities. In: *IEEE Communications Magazine*, vol. 55, no. 3, pp. 44–50
34. Abreu, David Perez, Velasquez, Karima, Curado, Marilia, Monteiro, Edmundo, A resilient Internet of Things architecture for smart cities, *J Annals of Telecommunications*, 2017, V 72, N 1, 1958–1995, <https://doi.org/10.1007/s12243-016-0530-y>
35. Zhang K, Ni J, Yang K, Liang X, Ren J, Shen XS (2017) Security and Privacy in Smart City Applications: Challenges and Solutions. *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129
36. Khatoun R, Zeadally S (2017) Cybersecurity and Privacy Solutions in Smart Cities. In: *IEEE Communications Magazine*, vol. 55, no. 3, pp. 51–59
37. Mehmood Y, Ahmad F, Yaqoob I, Adnane A, Imran M, Guizani S (2017) Internet-of-Things-Based Smart Cities: Recent Advances and Challenges. In: *IEEE Communications Magazine*, vol. 55, no. 9, pp. 16–24
38. Angelakis V, Tragos E, Pöhls HC, Kapovits A, Bassi A (2017) *Designing, Developing, and Facilitating Smart Cities*. 1st ed. Springer Publishing Company, Incorporated.
39. Anagnostopoulos T, et al. (2017) Challenges and Opportunities of Waste Management in IoT-Enabled Smart Cities: A Survey. In: *IEEE Transactions on Sustainable Computing*, vol. 2, no. 3, pp. 275–289
40. Li Z, Shahidehpour M, Bahramirad S, Khodaei A (2017) Optimizing Traffic Signal Settings in Smart Cities. In: *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2382–2393
41. Djahel S, Doolan R, Munten GM, Murphy J (2015) A Communications-Oriented Perspective on Traffic Management Systems for Smart Cities: Challenges and Innovative Approaches. In: *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 125–151
42. Ayadi H, Zouinkhi A, Boussaid B, Abdelkrim MN, Val T (2016) Energy efficiency in WSN: IEEE 802.15.4. In: *The 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Sousse, pp. 766-771
43. Hadi A, Wahidah I (2016) Delay estimation using compressive sensing on WSN IEEE 802.15.4. In: *International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, Bandung, pp. 192–197
44. Nlom SM, Chinnapen S, Ouahada K, Ndjiongue AR, Ferreira HC, Martinez R (2015) Coexistence of IEEE802.15.4 in a practical implementation of a wireless smart home environment (WSHE) for appliances control. In: *IEEE First International Smart Cities Conference (ISC2)*, Guadalajara, pp. 1-6
45. Mainetti L, Palano L, Patrono L, Stefanizzi ML, Vergallo R (2014) Integration of RFID and WSN technologies in a Smart Parking System. In: *The 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, 2014, pp. 104–110
46. Fernandez P, Jara AJ, Skarmeta AFG (2013) Evaluation Framework for IEEE 802.15.4 and IEEE 802.11 for Smart Cities. In: *The Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, Taichung, 2013, pp. 421–426
47. Sinha SR, Park Y (2017) *Building an Effective IoT Ecosystem for Your Business*. 1st ed. Springer Publishing Company, Incorporated
48. Datasheet Sensor Sharp 2Y0A21
49. Faludi R (2010) *Building wireless sensor networks: with ZigBee, XBee, arduino, and processing*. O'Reilly Media, Inc

50. Gravina R, Palau CE, Manso M, Liotta A, Fortino G (2017) Integration, Interconnection, and Interoperability of IoT Systems (Internet of Things). 1st ed. Springer Publishing Company, Incorporated.
51. Hillar GC (2016) Internet of Things with Python. Packt Publishing Ltd
52. Owens M, Allen G (2010) SQLite. Apress LP
53. Intel Software, Developer Zone. (<https://software.intel.com/es-es/articles/when-to-use-the-intel-galileo-board>)

# Vehicular Crowdsensing for Smart Cities



Tzu-Yang Yu, Xiru Zhu, and Muthucumar Maheswaran

**Abstract** As smart vehicles begin to roam the streets, new possibilities will emerge for large-scale data acquisition tasks necessary for proactive smart cities applications. Unlike mobile devices, smart vehicles carry powerful sensors and are highly mobile; they can cover large areas and perform high quality sensing. However due to restricted reward structures and limited bandwidths of cellular and VANETs, not all vehicles can participate equally. Thus, we must find a method for selecting promising participants which can efficiently the required collect sensing information. In this chapter, we present ideas for participant selection under varying conditions from large scale crowdsensing to personalized crowdsensing. We present several algorithms using a common framework.

## 1 Introduction

Modern vehicles are equipped with increasingly powerful sensors, communication interfaces and computing resources. As such, vehicular crowdsensing is quickly becoming a new paradigm for data collection [10, 14]. Data collected from crowd sensing could become essential for a smart city to provide dynamic and proactive services. Vehicles are recruited as sensing participants for large-scale crowdsensing tasks such as urban sensing or traffic condition monitoring. Compared to conventional mobile crowdsensing, vehicles are ideal platforms to collect, store, compute, and share large amounts of sensor data. The advantages are manifold; for instance, vehicles have greater mobility and cover wider sensing area. Furthermore, the mobility patterns of vehicles are predictable due to the prevalence of navigation systems. Most importantly, the abundance of on-board resources and lack of power constraints enable complex and long-running sensing tasks.

---

T.-Y. Yu · X. Zhu · M. Maheswaran (✉)  
McGill University, Montreal, QC, Canada  
e-mail: [tzu-yang.yu@mail.mcgill.ca](mailto:tzu-yang.yu@mail.mcgill.ca); [xiru.zhu@mail.mcgill.ca](mailto:xiru.zhu@mail.mcgill.ca); [maheswar@cs.mcgill.ca](mailto:maheswar@cs.mcgill.ca)

The primary application of modern vehicular crowdsensing research is generalized and large scale monitoring such as environment and traffic monitoring, map updating, public safety, urban sensing and so on [17, 25]. As such, data collected are primarily analyzed in a cloud server and results made available for public use. Such information can be reused by multiple applications. Given its nature, large-scale sensing is dominated by enterprises or governments. We believe that the benefit of crowdsensing paradigm should be available for personal use; tasks tend to be numerous but limited in scale. We define this paradigm as Personalized Vehicular Crowdsensing (PVC) [22]; it focuses on supporting user-specific sensing tasks.

Unlike generalized crowdsensing tasks, user-specific sensing tasks catered towards users' custom requests and are unlikely to be shared with other users. For instance, different sizes of trucks require varying road width for driving and turning. However, due to construction, snow, events or even bad parking, passable roads may no longer be traversable. Hence, it is necessary to look in real time for a wide variety of road width for different size of trucks. Such road width requirement depends on type of truck; the system should allow user to tune the road width parameter as a sensing task.

One related application, Waze, also attempts to leverage vehicular crowdsensing for everyday users [7]. In Waze, participants form part of a community which gathers information such as police location, traffic or roadblocks location. However, unlike our proposal, Waze does not support variegated user inquiries; sensing tasks are predefined by the platform. In addition, Waze requires participants to actively enter information; a participant which sees an accident would need to manually enter such information while driving. In contrast, PVC does not require participants to be actively engaged. The client generate a customized sensing task as a runnable program and submit the program to selected vehicle. Selected vehicle execute the program, sends result back to the requester if the task objective is met.

Applications like Waze that rely on users manually entering sensing results may not be trustworthy if users intentionally submit faulty sensing results for their own benefit. For instance, users who want to have better traffic conditions while driving can report accidents on the road. Thus, vehicles moving in the same direction may be directed to other routes by the system. In contract, our PVC allows the sensing program to implement security policies. The participant running the sensing program must follow the policies; otherwise the task result will be rejected. Thus, a program based sensing task not only can support customized sensing tasks but can also reduce security concerns.

As crowdsensing systems leverage participants for collecting data, an incentive system is necessary to maintain participation. To incentivize more users to participate in a crowdsensing system, the platform should reward participants. Conventional mobile crowdsensing often require complex and fine-grained incentive mechanism; it requires participants use their mobile phones and actively gather sensing data from their local environment. This can lead to significant inconvenience to participants. Besides, some tasks require participants to have specific knowledge before participating [16]. For instance, a task aimed at collecting photos of rare plant species may want to recruit participants with some knowledge of Botany.

Vehicular crowdsensing, on the other hand, does not require human involvement for completing tasks. Indeed, the quality of sensing results depends on the on-board sensors rather than the human factor. The participant only lend the on-board resources to the recruiter, which can be seen as buying computational resources from cloud servers. However, given that vehicles are owned by individuals, there exists significant privacy and security issues compared to cloud servers. Thus, typical payment schemes used in cloud computing cannot be directly applied for vehicular crowdsensing.

Moreover, unlike conventional cloud computing which consists of numerous of static servers, vehicular crowdsensing system comprises a dynamic collection of vehicles (mobile servers). Because of spatial-temporal nature of moving vehicles, efficiently selecting and utilizing vehicular participants' on-board resources is one of the key challenges in building a vehicular crowdsensing service. Although many participant recruitment algorithms has been recently added to the literature, unique characteristics of PVC have introduced several new challenges in designing participant recruitment algorithms.

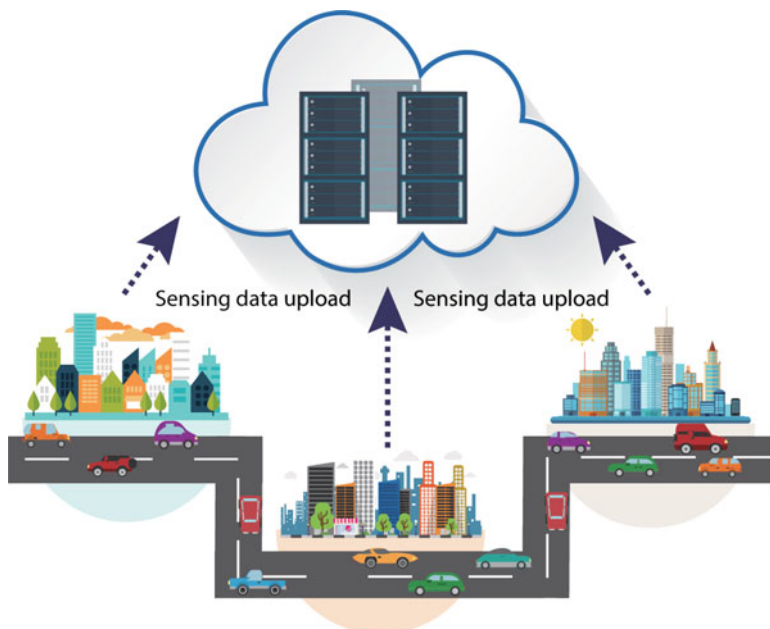
In this chapter, we focus on providing a discussion on the concept of vehicular crowdsensing for smart cities, with focus on how participant selection algorithm is used under different crowdsensing paradigms and specific application needs. We also explore the challenges of providing recruitment algorithms for PVC as well as propose and evaluate several recruitment algorithms for PVC tasks.

## **2 Background and Characteristics**

Crowdsensing refers to the outsourcing of data collection to users. This means that participants contribute to a shared pool of information with mobile sensing devices. Here, sensing devices often refer to smart phones or increasingly refer to vehicles with powerful sensors. We distinguish between Vehicular and Mobile crowdsensing given each has distinct characteristics. For instance, the mobility pattern of vehicles is predictable due to the prevalence of navigation systems. Such predicted path is often utilized for efficient participant selection for Vehicular crowdsensing tasks.

### ***2.1 Two Different Crowdsensing Paradigms***

In the literature, crowdsensing can be categorized into two type of paradigms: public crowdsensing [14] and personalized crowdsensing [22]. The primary application of public crowdsensing research is generalized and large-scale monitoring such as environment, traffic monitoring, map updating, public safety, noise pollution assessment, or urban sensing. The aggregated data is often shared to the public and can be reused by multiple applications. As such, sensed data is collected and



**Fig. 1** Conventional public vehicular crowdsensing

analyzed within the cloud server itself and then made available for public use. Given its scale, such large scale sensing are dominated by enterprises or governments.

In terms of architecture for public sensing, the server schedules optimal set of vehicular participants to cover the sensing region and collects sensing data from selected participants as shown in Fig. 1. The information sensed is often fixed, such as air quality or noise level; hence, a server would continuously request the same type of information.

In contrast, personalized crowdsensing focuses on supporting user defined sensing tasks. These tasks cater towards a specific user's requests and hence are unlikely to be requested by other users. Thus, data can also be locally processed on the sensing device; the server only need to forward the results. For instance, different size of trucks need varying road width for driving and turning. However, due to construction, snow, or events, it's possible roads width change and are no longer traversable. Hence, it is necessary for the trucker to get real time road width information ahead of his path by hiring participants. These participants are hired to perform the specific user defined sensing task of looking at road width and need to follow specific spatio-temporal constraints from the recruiter. Thus the sensing result is unlikely to be shared with other users.

The Fig. 2 depicts the architecture of a personalized crowdsensing service. As shown in Fig. 2, recruiters send task requests to the server. The server retrieves requests and assigns tasks to appropriate participants. Participants utilizes on



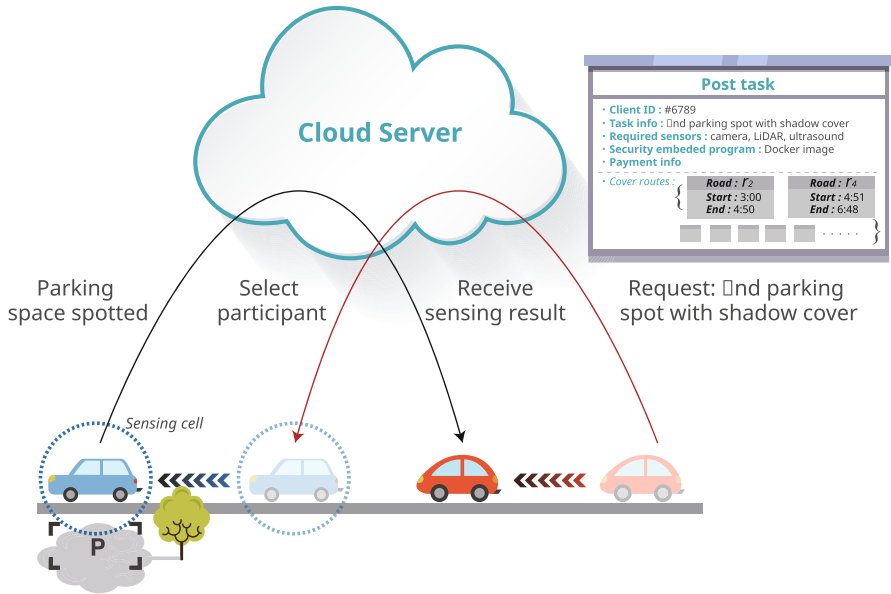


Fig. 2 Personalized vehicular crowdsensing

board resources to process sensing data. When the requested task is successfully completed, participants will send back sensing results to the corresponding recruiter via the cloud server. With personalized sensing, data requests are frequently received from multiple recruiters. The cloud server collects client requests and schedules such requests to participants to fulfill.

## 2.2 Central Server for Vehicular Crowdsensing

With vehicular crowdsensing, one major problem is the reliability of data received. We would be troubled if a participant sent purposefully misleading data. Hence, central control is necessary to maintain reliable information for the two vehicular crowdsensing paradigms introduced above. Second, there should exist a reward system for which providers of reliable and timely data are appropriately rewarded. Otherwise, it may be difficult to maintain user participation. Third, we seek to reduce redundant data. Unlike fixed sensors, vehicles move around and it's likely that sensing coverage overlap with others. Thus, the central server need to be equipped with data de-duplication mechanisms to avoid overwhelming by gathering all possible data to consumer.

## 3 Public Vehicular Crowdsensing

### 3.1 Background

In this section, we focus on participant selection for public vehicular crowdsensing. Public vehicular crowdsensing targets large scale sensing, such as air quality monitoring, traffic monitoring or even public safety. Public Vehicular Crowdsensing's purpose is to gather as much information as possible in the targeted sensing region. Given its scale, not every potential participant can and should be selected. The problems are threefold. First, each participant must be rewarded; this reward is often monetary [23]. For instance, Amazon Mechanical Turk is an example of crowdsourcing with monetary incentive where users perform tasks for small amounts of money [4]. Second, local network infrastructure can be overwhelmed when large number of vehicles continuously collect and send data. Vehicular crowdsensing requires significant overheads; Current vehicular communication technology have limitations; vehicular ad-hoc networks do not have a central server to control bandwidth and suffer from reliability issues [2]. In contrast, cellular networks are more reliable but current cellular network data usage is already growing exponentially; This risks severe network congestion in the future [1]. Third, significant sensing overlap exist between vehicles; this means selecting all vehicles would results in large amount of redundant data. For instance, sensing air quality using every vehicles in traffic would result in processing large quantity of information; selecting only a few would have been sufficient. Processing large quantity of data requires large data centers which adds costs. Thus, public vehicular crowdsensing participant selection attempts to select participants which can maximize coverage while also limiting costs.

### 3.2 System Model

Public vehicular crowdsensing is composed of a set of participant vehicles with wireless transceiver and sensor devices. Typically, participants are either smart cars or buses; both can carry onboard computing resources and sensing ability. We also assume we cannot control participants movements; participants do not actively participate in sensing tasks. Selected participants only gather data passively along their route as they pass locations of interest.

To communicate with the cloud or other vehicles, current vehicular crowdsensing relies on vehicular ad hoc networks (VANET) and LTE. VANET utilize IEEE 802.11p, Wireless Access for the Vehicular Environment as standard (WAVE). They form temporary network of vehicular clusters and road side units (RSUs) [5]. Vehicles can communicate with neighbors in the immediate cluster directly but require RSU's assistance to send data beyond the local cluster. However, VANET suffer from reliability problems [5]. To fix reliability issues, some works for VANET include LTE [18].

To store accumulated data, cloud servers are necessary. Then, applications could request for sensing information from the cloud servers. We assume the cloud servers know the participants' current location; participants could beacon that information from time to time. Many works in the literature assume knowledge of participants' predicted route; this can be information from a GPS for instance. However, some work in the literature only assume knowledge of past routing information. At set time intervals, the cloud servers can schedule a subset of available participants to collect sensing data.

When a vehicle participates in crowdsensing, it must be rewarded. Designing an incentive model which can consistently attract participant is a integral part of crowdsensing. Important aspects to be considered are costs, sensing quality, sensing coverage. Not all incentive models result in monetary gain [23]. Waze is an vehicular crowdsensing application, where users share sensing information on the platform; users are both participants and consumer of sensing information. For monetary incentive system, there exists reverse auction where participants bid for sensing tasks [11, 24]. To prevent low quality sensing, rewards based on data quality are often considered [15]. Other systems rely on game theory such as Stackelberg game. Here, a consumer proposes an offer to all participants to consider. This repeats for multiple iteration until sensing needs are met [20].

Besides minimizing cost, maximizing the amount of area covered and amount of area covered  $N$  times over the targeted region is paramount. Overall, there are two properties for coverage to consider; spatial coverage and temporal coverage. Spatial coverage simply seeks to maximize amount of area covered over a time window. The weakness of this approach is that areas covered by sensors can be unbalanced; some areas may never be covered while other areas may be always covered. Thus, popular roads will always have sensing participants on it while side roads suffer in terms of coverage frequency. However, this approach tend to work well with sparse distribution of participants [10]. In contrast, temporal coverage is about maximizing the number of regions covered at least  $N$  times. This would result in more even distribution for collected data but may reduce the overall quantity of data collected. Furthermore, coverage may not be as complete. There's little incentive to gather more data from an area already covered even though some new sensing information could be gained. A third approach exists; by combining both spatial and temporal coverage properties, we can obtain a hybrid approach [14]. In this approach, each vehicle can only gather a capped amount of information at a specific area.

### 3.3 Definitions and Assumptions

We define notations to be used when describing participant selection algorithms. We define a sensing region as  $R = \{r_1, r_2, \dots, r_m\}$ , composed of smaller areas. The exact definition and size of each area  $r_i$  differs with each approach. For instance, in [10], each area consist of graph nodes whereas in [21] they consist of 1 meter by 1 meter square areas. We define the set of vehicles as  $V = \{v_1, v_2, \dots, v_n\}$ . We

**Table 1** List of notation for public vehicular crowdsensing

Notation	Description
$T$	Sensing time window
$V$	Set of all vehicles
$P$	Set of vehicular participants selected for sensing
$R$	Set of areas for sensing
$t_i$	Time $i$ within time window $T$
$v_i$	Vehicle $i$ in $V$
$p_i$	Participant $i$ in $P$
$r_i$	Area $i$ in $R$
$r_{v_i, t_j}$	Location of vehicle $v_i$ at time $t_j$
$C(v_i)$	Function which returns the cost of participant $v_i$
$B$	The total budget of sensing task

assume each  $v_i$  has sensors and is willing to collect information. We define vehicles selected as participants as  $P = \{p_1, p_2, \dots, p_m\}$ , where  $P \subseteq V$ . We define the time window as  $T = \{t_1, t_2, \dots, t_q\}$ , where each  $t_j$  is a time unit. Let  $r_{v_i, t_j}$  be the location of vehicle  $v_i$  at time  $t_j$ , where  $r_{v_i, t_j} \in R \cup \emptyset$ . We define  $C$  as a function which when given a vehicle returns the cost of recruiting such vehicle. We define  $B$  as the budget constraint which limits the number of participants we can select. Finally, we define *coverage* as a function which takes in a participant set and sensing region and returns the coverage measure of selecting such participant set (Table 1).

### 3.4 Problem Statement

Public vehicular participant selection problem can be expressed as integer linear programming problem. We assume that the coverage function is provided. This problem has been shown to be NP hard; this can be proved by reducing the problem to a set cover problem [9, 10, 21].

**Input Values** Sensing region  $R$ , vehicle set  $V$ , cost function  $C$ , Budget  $B$ , coverage function *coverage*

**Objective Function** Find a set of participants  $P$  best fit as sensing participants.

**Maximize**  $coverage(P, R, \dots), P \subseteq V$

**Subject to**  $\sum_{p_i \in P} C(p_i) \leq B$

### 3.5 Participant Selection Algorithms

In this section, we will explore a variety of algorithms proposed for public vehicular crowdsensing participant selection. Many works in the literature have covered

mobile crowdsensing but have only recently started covering participant recruitment for vehicular crowdsensing.

Hamid et al. first proposed the idea of utilizing vehicular trajectory to best select vehicular participants [8]. In this paper, each participant has a reputation based on its past sensing results; participant commitment and quality of information provided. Participant commitment is the likelihood a participant will follow its provided trajectory. Quality of information is the quality of previous sensing collection by the participant. Hence the reputation of a participant  $v_i$  is given by the following equation where  $\alpha$  and  $\beta$  are weights and  $p$  and  $q$  are participant commitment and quality of information respectively.

$$reputation(p_k) = \alpha * p_{p_k} + \beta * q_{p_k} \tag{1}$$

The reward for each participant,  $C_{p_k}$  is based upon a fixed cost plus a variable cost based on coverage distance  $d_{p_k}$  and reputation.

$$C(p_k) = cost_{fixed} + cost_{variable} * d_{p_k} * reputation(p_k) \tag{2}$$

The problem is formulated as two step integer linear programming, one for maximizing the number of regions covered and the second for minimizing cost of participants which achieves maximum coverage. In the first step, Hamid et al. maximizes coverage by maximizing the average number of regions covered while remaining within budget  $B$ .

$$hamid\_coverage(P, R) = \sum_{t_j \in T} | \bigcup_{p_i \in P} r_{p_i, t_j} | \tag{3}$$

In the second step, the distance covered by selected vehicles is minimized while maintaining the same level of coverage in step one. This will minimize the total cost.

$$hamid\_step_2(P, R) = \sum_{p_i \in P} p_i * d_{p_i} \tag{4}$$

To solve the above equations, Hamid et al. simply utilized a integer programming solver, Gurobi 5.1. Integer linear programming is NP-hard but there are methods for approximations. However, even with approximations, the runtime of the method is bound to be a higher order polynomial. Thus, Hamid et al. method of solving vehicular participant takes too long for large-scale public sensing.

Han et al. presents two participant recruitment algorithms based on predicted trajectory [9]. Note that the cost,  $c(v_i)$  for selecting  $v_i$  is assumed to be 1 where  $\forall v_i \in V$ . The cost of selecting a vehicle is the exact same as selecting any other vehicle. The budget constraint  $B$  is the number of participant selected. The *coverage* function measure temporal coverage; it computes number of areas covered by sensors at least once.

$$han\_coverage(P, R) = \left| \bigcup_{t_j \in T} \bigcup_{p_i \in P} r_{p_i, t_j} \right| \quad (5)$$

Finally, to evaluate the algorithms, Han et al. utilized a dataset consisting of real GPS trace of 20,000 shanghai taxi from 08:42–09:42. The first algorithm, referred as the offline algorithm, assumes full knowledge of all vehicles and their trajectory within the time window  $T$ . It finds the vehicle which adds the most coverage and selects it as part of the solution. The algorithm then iterates until  $B$  vehicles have been selected. The algorithm's time complexity is  $O(B * |V| * \log(|V|))$ . In contrast, the second algorithm, referred as the online algorithm, assume no prior knowledge of a vehicle before it joins the crowdsensing system. Since the system wouldn't know of a vehicle and its trajectory before it comes into range, this may be a more realistic assumption. The algorithm decides whether to select a vehicle  $v_i$  when it joins the system by comparing the gain in temporal coverage of adding  $v_i$  with a dynamic threshold. The dynamic threshold is computed based on the number of participants already selected. The online algorithm's time complexity is  $O(B * |V|)$ .

Similarly, He et al. also proposed two participant recruitment algorithms based vehicular trajectory [10]. Both algorithms assume full knowledge of vehicles and their trajectory within time window  $T$ . The crowdsensing cost  $C$  is generated according to a normal distribution. To evaluate the solution, traffic trace dataset was obtained from TAPAS-Cologne [19], a 24 h generated vehicular trace of the city of Cologne in Germany simulated using SUMO [13]. The first algorithm consist of a greedy approximation which maximizes spatial coverage and is meant for small number of sparsely deployed participants. Here, He et al. define spatial coverage as simply the number of areas covered over a time period  $T$ .

$$he\_coverage\_1(P, R) = \sum_{t_j \in T} \left| \bigcup_{p_i \in P} r_{p_i, t_j} \right| \quad (6)$$

Similar to the offline algorithm by Han et al. this algorithm adds the most cost effective participant and iterates until the budget constraint is met. Cost effectiveness is defined as the difference in spatial coverage divided by its cost. The algorithm has a time complexity of  $O(|V|^2|T|)$  Since this algorithm does not take into account temporal coverage, this may result in unbalanced data distribution but maximizes the amount of information gathered. This is beneficial with a small number of participants sparsely deployed. In addition, this algorithm is a  $(|T| + 1)$  approximation algorithm; quality suffers when the time window is long. In contrast, the second algorithm proposed is a genetic algorithm meant for large number of densely deployed participants. It utilizes the minimum covered time for all areas as coverage.

$$he\_coverage\_2(P, R) = \min \left( \bigcup_{r_k \in R} he\_coverage\_1(P, r_k) \right) \quad (7)$$

The genetic algorithm encodes vehicle selection outcome as a binary string. Thus, with 5 possible participants, if we only select the first participant, the encoding would be “10,000”. Initially, a large number of solutions are randomly generated. At each generation, the coverage, based on temporal coverage is computed and only a top percent of the population survives. In addition, mutation and crossover operations occur as well to mimic the evolutionary processes. Crossover combines two solutions to obtain a hybrid of the two. Mutation randomly changes part of the selection outcome. Finally, solutions which violate the cost constraint are trimmed to fit. The algorithm runs until the time limit is reached or until the theoretical upper bound is reached. The main advantage of this algorithm is that it can be capped in terms of runtime which allows selection for large number of participations; on the downside they may result in weaker solutions.

Due to polynomial time complexity from many proposed algorithms, Yi et al. proposes a linear time algorithm for participant selection based on submodular property of the problem [21]. This would benefit large-scale vehicular crowdsensing where scheduling time may be tight. Coverage is defined as the maximum number of areas covered over  $T$ .

$$yi\_coverage(P, R) = \sum_{t_j \in T} | \bigcup_{p_i \in P} r_{p_i, t_j} | \quad (8)$$

The utility of a recruiter is defined as the number of coverage minus the costs of participants recruited where  $\lambda$  is a weight parameter. The objective is to maximize the utility of the recruiter. Note that there is no hard cap for number of participants;  $\lambda$  can be adjusted to serve as a soft cap. The utility function is proved to be submodular

$$utility(P, R) = yi\_coverage(P, R) - \lambda * \sum_{p_i \in P} C(p_i) \quad (9)$$

The algorithm relies on a forward and reverse greedy algorithm operating at the same time. Thus, the algorithm begins with an empty set and also a set of all participants. At each iteration, we consider whether to add a participant  $v_i$  by adding  $v_i$  to one of the empty set and removing  $v_i$  from the full set. The change in utility by adding and removing a vehicle is utilized to compute a probability of whether to include the participant as part of the solution. The runtime complexity of the algorithm is  $O(|T| * |R| * |V|)$  and thus is linear. It achieves an approximation ratio of  $1/2$ . To evaluate the algorithm, Shanghai Taxi dataset was utilized, covering the trajectory of 4316 taxis. Furthermore, a synthetic dataset was generated using Gauss-Markov mobility model. Results show that the algorithm only does slightly worse than a polynomial greedy algorithm but took considerably less time to run.

In contrast to generalized public crowdsensing approaches, Gao et al. proposes an air monitoring vehicular crowdsensing system using buses [6]. Unique to Mosaic’s approach is that unlike passenger cars, buses have fixed routes. Thus the first algorithm selects entire bus routes based on achievable coverage. To expand on the first algorithm, the second algorithm selects individual buses. Furthermore,

instead of seeking to cover an entire area, Mosaic proposes Points of Interests (POI) which are high priority sensing locations. These high priority area could be schools, hospital or other public spaces. The priority between POI are considered to be equal. The sensing region  $R$  is split into 100 by 100 m areas; this may feel somewhat rough in terms of precision but is acceptable for air quality monitoring. Each area has an importance value,  $\delta$ , attached, based on its distance to the nearest POI. Hence, an area close to a POI has higher  $\delta$  compared to an area far from any POIs. Since air quality can be inferred from nearby measurements, coverage can be defined as the number of route passing within or near the area.

$$local(r_k, R, POI) = \begin{cases} \delta(r_k, R, POI) & > 2 \text{ routes passing thru } r_k \\ 0.75 * \delta(r_k, R, POI) & 1 \text{ or } 2 \text{ routes passing thru } r_k \\ 0.5 * \delta(r_k, R, POI) & \geq 1 \text{ routes } 1 \text{ areas away} \\ 0.25 * \delta(r_k, R, POI) & \geq 1 \text{ routes } 2 \text{ areas away} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$coverage(V, R, POI) = \sum_{r_i \in R} local(r_i, V, POI) \quad (11)$$

The first algorithm proposed to find the best bus route is similar to the greedy algorithms mentioned in previous papers. It finds the route which adds the most coverage and adds it as part of the solution; this is iterated until the *Budget* is reached. However, since this algorithm does not consider the temporal dimension of coverage, a second algorithm for bus selection is proposed. In this second algorithm, coverage equation is slightly modified to include percent of time coverage. Similarly, the algorithm attempts to select the bus with highest coverage and adds it as part of the solution. It iterates until the budget  $B$  is reached. The algorithm's time complexity is  $O(B * |V| * |R|^{0.5})$  and takes about 10 s to run with 1415 buses. To evaluate Mosaic, data was collected from a  $2.9 \times 3.1 \text{ km}^2$  city area in China. It consisted of 282 bus routes and 1415 buses schedule with 72 POI consisting of school and hospital locations. Mosaic proposed a crowdsensing system which works well for air quality monitoring but would suffer when crowdsensing for other type of sensing data. Here, temporal coverage is less emphasized; air quality changes does not occur as suddenly. This paper does not consider participants selection for general sensing tasks.

The participant recruitment algorithms proposed above only considered a single type of sensing data. However, sensors on vehicles are heterogeneous, furthermore quality of sensors may differ. Hence, Liu et al. proposed a heterogeneous participant selection algorithm [14]. Unlike other works mentioned, Liu et al. utilize a time continuous markov chain mobility model rather than assume to know a participant's trajectory. Thus, given a vehicle's position, a vehicle has an average stay duration and a likelihood of transitioning to another area. The longer a vehicle stays in an area, the more data it gathers. The cost of selection is 1 for each participant and the Budget is the maximum number of vehicle which can be selected. Thus the



coverage is simply how much information all participant can gather from all sensors it possess. The objective is to maximize the total coverage while remaining within the *Budget*.

The algorithm for selection is once more a greedy algorithm which adds the best participant at each iteration and repeats until *Budget* has been reached. This is  $O(|Budget| * |V|)$  in terms of time complexity since it only seeks to find the maximum rather than sorting. To evaluate this algorithm, GPS trace of T-Drive trajectory dataset are used. This contain the trajectories of 10,357 taxis and about 15 million data points.

One weakness of current vehicular recruitment strategy is that once selected, a participant must continue sensing for a fixed period of time. However, inefficiency exists; the participant selected may be only truly cost effective for part of the time window  $T$  selected. Furthermore, trajectory provided by participants tend to be error prone in reality. Thus, Hu et al. have proposed a variable duration participant recruitment with uncertain trajectory [12].

To deal with uncertainty, Hu et al. proposes a probabilistic method for estimating location of a vehicle given a trajectory. This probability can be obtained from historical data. Let  $prob(r_k, t_j, v_k)$  be the likelihood of vehicle  $v_k$  to be at region  $r_k$  at time  $t_j$ . Thus, the position vehicle  $i$ ,  $r_{v_i, t_j}$  instead of returning a single area  $r_k$ , returns a R sized matrix of probability. Note that  $|r_{v_i, t_j}| = 1$ .

$$r_{v_i, t_j} = \bigcup_{r_k \in R} prob(r_k | v_i, t_j) \tag{12}$$

Furthermore, the solution for variable duration participant recruitment is a  $|V|$  by  $|T|$  matrix denoting whether  $v_i$  is recruited at  $t_i$  instead of a set of selected vehicles.

$$select_{v_i} = \begin{bmatrix} v_{11} & \delta_{12} & \delta_{13} & \dots & \delta_{1q} \\ v_{21} & \delta_{22} & \delta_{23} & \dots & \delta_{2q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{n1} & \delta_{n2} & \delta_{n3} & \dots & \delta_{nq} \end{bmatrix}$$

$$select_{v_i, r_j, t_k} = \begin{cases} 1 & Selected \\ 0 & Otherwise \end{cases} \tag{13}$$

In addition, the cost for selecting participants is defined by a  $|V|$  by  $|T|$  matrix, where a cost is associated for each time point for each vehicle. Thus, cost function for a participant  $v_i$  is defined as follows.

$$C(v_i) = \sum_{t_j \in T} C_{v_i, t_j} * solution_{v_i, v_j} \tag{14}$$

Finally, Hu et al. defines spatial coverage as a function of the vehicle trajectory probability and the required number of vehicles for sensing at area  $r_k$ ,  $required_{r_k}$ . Thus, the number vehicles selected at  $r_k$  at time  $t_j$  must be greater or equal than  $required_{r_k}$  or else coverage is defined as 0. Let  $P_{r_k}$  be the set of participants selected which covers region  $r_k$ .

$$SC(r_i, t_j, V) = \prod_{p_k \in P_{r_i}} prob(r_i, t_j, p_k) \quad (15)$$

$$SC_{case}(r_i, t_j, V) = \begin{cases} SC(select, r_i, t_j) & \sum_{v_k \in V} select_{r_i, t_j, v_k} > required_{r_k} \\ 0 & otherwise \end{cases} \quad (16)$$

$$coverage(select, R) = \sum_{r_i \in R} \sum_{t_j \in |T|} SC_{case}(select, r_i, t_j, V) \quad (17)$$

Note that this coverage function will penalize for overlap as the coverage probability is multiplied for every vehicle covering an area. The objective of the algorithm is to select a set of vehicles at different time points which maximizes the coverage function while staying within budget  $B$ . Given that variable duration participant selection problem is a subset of standard vehicular participant selection, the problem is NP-hard. Thus, the first step is a pruning algorithm to remove non-viable participants while the second step is a greedy algorithm to select the vehicle and time for sensing.

The pruning step computes a Pearson correlation matrix to find vehicles with significant trajectory overlap. Vehicles with significant overlap are grouped together. The vehicle with lowest average costs within the group will be preferred given they cover similar areas at similar times. As such, the algorithm can prune to only  $|V'|$  participants. The step has a time complexity is  $O(|V'|^2|V|)$ .

In the second step, vehicles are still selected at specific times. The algorithm simply seeks to iteratively recruit the best participant  $v_k$  for each area  $r_k$  at each time  $t_j$ . To find such best vehicle, Hu et al. define the following SC efficiency measure to rank participants.

$$efficient_{v_i, r_j, t_k} = SC(select_{t_k}, R) - SC(select_{t_k-1}) / C_{v_i, t_j} \quad (18)$$

This measures the marginal increase in spatial coverage for a single vehicle at time step  $t_j$ . Thus, the time complexity is  $O(|V'|^2|T|^2)$ .

To evaluate their algorithms, Hu et al. utilized a trace dataset from taxis in Rome over an area of 64 km<sup>2</sup> [3]. The duration of the dataset is over 30 days. Thus, the model proposed attempts to select participants at each  $t_i$  during a given time window. This improves coverage metrics but suffers from higher computational costs. Furthermore, the proposed model can only obtain greater coverage with the same workload by selecting more vehicles and switching off vehicles when they are no longer useful for sensing. This has a few problems; first reward system

could include a fixed cost as part of the recruitment; recruiting more vehicles would increase costs [8]. Furthermore, sensing data collected may result in highly fragmented data from multiple sources; for instance a single time window  $T$  covered can have at most  $T$  vehicles covering each area. This is a problem because of lack of continuity and varying sensor quality. Compared to single continuous data, having multiple fragments of data increases the level of noise and reduce sensing quality.

## 4 Personalized Vehicular Crowdsensing

Public vehicular crowdsensing is more appropriate for large scale sensing such as environment and traffic monitoring, public safety, and urban sensing. Beyond public sensing, the benefit of crowdsensing can be available to support more personal tasks. However, unlike conventional large scale sensing tasks, the unique characteristics of personalized vehicular crowdsensing have introduced several new challenges in the design of participant recruitment algorithms. These include:

1. Personalized crowdsensing which targets every-day users have tighter budget constraints compared to the public crowdsensing supported by large enterprises or governments.
2. Requests by users are diverse; for instance, finding parking space, checking a favorite restaurant, etc. Evidently, it would be meaningless to employ large scale sensing; the sensing region in personalized vehicular crowdsensing is both location and time specific.
3. Personalized crowdsensing tasks are time sensitive compared with large scale sensing tasks. For instance in finding parking slot scenario, the spot can be taken before the client arrives. Thus the system needs to guarantee timeliness. On the other hand, we do not need to cover all sensing region at all time; we can reduce the number of sensing participants accordingly.
4. Because clients can submit requests as needed, multiple requests can be made by a single client. Since sensing data is processed locally, overloaded participants may fail to process all requests. Thus, on-line load balancing is necessary when selecting participants.

Traditional vehicular crowdsensing participant recruitment seeks to cover an entire area over a time window. Therefore traditional approaches cannot be applied to the Personalized Vehicular Crowdsensing Participant Recruitment problem (PVC-PR) because here we only consider a particular location at a specific time. For instance, suppose a recruiter is interested in finding a parking slot near his destination. Let client's route go through three unique regions  $R = \{r_1, r_2, r_3\}$  with each region requiring 10 min to traverse. In such tasks, we do not need to know the parking status at region  $r_3$  while we are still in region  $r_1$ . This is because the parking space might be taken away before we arrive at  $r_3$  which requires a total 20 min driving time. Thus, traditional participant recruitment algorithms could suffer from over recruitment, which is very inefficient for PVC tasks. We only need

to maintain partial coverage at indicated locations instead of covering all locations all the time.

In the following sections, we propose and evaluate several algorithms specifically for PVC tasks. The main objective of these algorithms is to recruit the minimum set of vehicular participants which can complete the PVC tasks. We also ensure proper load balancing among all the participants to reduce the chance of task failures.

## 4.1 System Model and Assumptions

Similar to conventional crowdsensing systems, PVC is comprised of cloud servers and massive number of smart vehicles, and the vehicles are equipped with sensors and communication devices such as Wi-Fi and Cellular interfaces. When a vehicle begins its journey, general information such as unique vehicle ID and predicted route from navigation system are uploaded and stored in the cloud server. Vehicles need to reload predicted trajectories again if any changes occur to their planned routes. Recruiters can make queries on sensing data of interest to the server. The query needs to specify the sensing target and minimum distance from the recruiter. For instance, the recruiter is interested in finding parking space at least 1 km ahead but no longer than 5 km away. We consider such monitoring area as Monitoring Window (MW). Given the recruiter's planned route as the sensing route, the cloud server need to select a set of proper vehicular participants to complete such task. The participant selection decision is based on the provided monitoring area. As shown in Fig. 2, the monitoring area changes dynamically based on projected trajectory of the recruiter. Thus monitoring coverage should follow recruiters' movements.

## 4.2 Definitions

Before formally presenting the problem, we describe some definitions and notations. We consider the area of interest to be divided into a number of small road segments  $R = \{r_1, r_2 \dots r_m\}$ . Each road segment has a single traffic direction. Roads which have opposite traffic directions are considered as two different road segments. The area also contains a set of vehicles  $V$ , and let  $C = \{c_1, c_2 \dots c_i\}$  be the set of clients and  $P = \{p_1, p_2 \dots p_j\} \subseteq V$  be the set of participants.

Client's sensing route and participant's projected route is composed of sequence of road segments as shown in Fig. 3. Each road segment contains a time stamp specifying arrival time of such vehicle. Let  $R_c$  be the set of route segments in client's query  $c$  and let  $R_p$  be the set of predicted future road segments of a given participant  $p$ . The time stamp of a given road segment can be derived from the following functions:



Fig. 3 Future routes and sensing routes

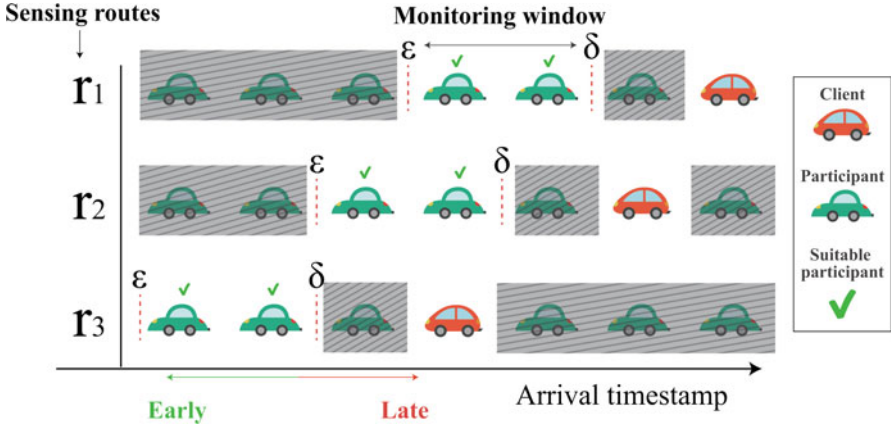


Fig. 4 Monitoring window (MW): a window for selecting useful sensing participants. For instance, a client needs a participant at least 1 min drive away  $\delta = 60$ s, but cannot be more than 5 min drive away  $\epsilon = 300$ s

$$\mathcal{T}(r_k, R) = \begin{cases} \mathbb{R}^+ & r_k \in R \\ -MAX\_INT & \text{otherwise} \end{cases}$$

In order to select valid and suitable participants for a specific client, we use the following definitions:

**Definition 1 (Common road segments)** Given a client’s query route  $R_c$  and participant’s future trajectory  $R_p$ , the set of common road segments is defined as

$$R_{c,p}^{com} = R_c \cap R_p, \quad c \in C \wedge p \in P \tag{19}$$

**Definition 2 (Data timeliness)** An important criteria to consider for participant selection is that data must arrive in a timely fashion to the client. In other word, data that arrives too early or too late is considered worthless to the client. Thus, participants need to be in a specific sensing window in order to provide useful sensing data. We refer to such window as monitoring window. Figure 4 shows how MW helps in selecting useful participants. Let  $\delta$  and  $\epsilon$  be the lower bound and upper bound for the window, respectively. The value of a participant at route  $r_k$  is represented as follow:

$$\Gamma(R_c, R_p, r_k) = \begin{cases} 1 & \delta \leq \mathcal{T}(r_k, R_c) - \mathcal{T}(r_k, R_p) \leq \varepsilon \\ & , r_k \in R_{c,p}^{com} \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

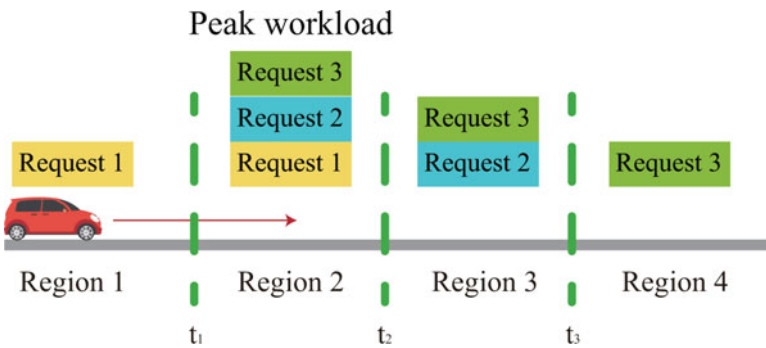
**Definition 3 (Query route coverage (QRC))** We define query route coverage as the total number of road segments a set of selected participants can cover for a single client under the associated timeliness constraint. Given a client,  $c \in C$ , and a participant,  $p \in P$ , a single participant coverage for a single client can be defined as:

$$single\_cover(c, p) = \{\forall r_k \in R_p | \Gamma(R_c, R_p, r_k) = 1\} \quad (21)$$

Thus, given a set of participants  $S \subseteq P$ , the function  $QRC$  can be defined as follows:

$$QRC(S) = \left| \bigcup_{p' \in S} single\_cover(c, p') \right| \quad (22)$$

**Definition 4 (Participant maximum load (PML))** Participants will be assigned sensing tasks by different clients along its journey. Because of the spatio-temporal nature of the PVC tasks, certain regions may not have sufficient member of participants at specific spatio-temporal locations. Hence, some participants in popular sensing regions may be overloaded. Figure 5 shows an example of participant workload with 3 clients. In the figure, the participant  $p$  services 3 clients between timestamp  $t_1$  and  $t_2$  in region 2.



**Fig. 5** Sample workload for a single participant with 3 clients; depending on clients location of interest, the workload is spatial-temporally assigned

**Table 2** List of notation for personalized vehicular crowdsensing

Notation	Description
$R$	Set of road segments
$C$	Set of clients
$P$	Set of vehicular participants
$R_c$	The query route segments of the client $c \in C$
$R_p$	The predicted future road segments of the participant $p \in P$
$\mathcal{T}(r, R)$	Function to get the timestamp of a road segment $r$ , given a set of road segments
$R_{c,p}^{com}$	The set of common road segments, obtained from $R_c \cap R_p$ .
$\delta$	Lower bound of monitoring window (MW)
$\varepsilon$	Upper bound of MW
$\Gamma(R_c, R_p, r)$	Value function return 1 if the timeliness constraints are satisfied and 0 otherwise
$single\_cover(c, p)$	The function return set of road segments such that $\forall r \Gamma(R_c, R_p, r) = 1$
$load_p^{max}$	The maximum workload the participant $p$ will have for entire journey
$\beta_c$	Maximal number of vehicles that the client $c$ can hire for doing the sensing task
$cap_p$	Soft threshold of maximum number of tasks that $p$ is able to serve simultaneously

### 4.3 Problem Formulation

The PVC participant recruitment problem can be formulated as a two stage optimization. In the first stage, we seek to find the set of participants which can maximize requested coverage. That is, given a single client  $c$  and set of participants  $P = \{p_1, p_2 \dots p_j\} \subseteq V$ . The objective function is defined as:

**Objective function of the first stage** recruit subset of vehicles  $S' \subseteq P$  maximizing coverage subject to a budget constraint  $\beta_c$ .

$$\left\{ S' \in \arg \max_S QRC(S) \mid |S'| < \beta_c \right\}$$

where  $\beta_c$  is the maximal number of vehicles that the client  $c$  can hire for doing the sensing task (Table 2).

**Objective function of the second stage** Since the first stage may return several solutions which achieves maximum coverage, we need to make sure the solution can achieve global load balance to guarantee quality of service. Thus, the objective of the second stage focuses on recruiting a set of participants from the first stage which reduces the workload among all vehicular participants given a stream of requests  $\mathbf{c} = \{c_1, c_2 \dots\}$ . The formulation of this stage is defined as below.

$$\text{Minimize } \sum_{p' \in P} \frac{\text{load}_{p'}^{\max}}{\text{cap}_{p'}}$$

Where  $\text{load}_{p'}^{\max}$  signify the maximum workload participant  $p'$  will be service for, and  $\text{cap}_{p'}$  is a soft threshold of maximum number of tasks that  $p'$  is willing to serve simultaneously.

#### 4.4 Algorithm Design

In this section, we present our online participant selection algorithm for each incoming request. To ensure our solution can maximize coverage for requested sensing routes as well as spread global load balance, we use the following score function for selecting decision.

**Definition 5 (Workload score function)** To select proper participants, our workload score function which considers participant's current maximum workload,  $\text{load}_p^{\max}$  is defined as follows:

$$\text{Score}(\text{load}_p^{\max}) \leftarrow \frac{\text{cap}_p}{\text{load}_p^{\max}}, \quad \forall \text{load}_p^{\max}, \text{ and } \forall \text{cap}_p \in \mathbb{N}^+ \quad (23)$$

Where  $\text{load}_p^{\max}$  indicates the maximum workload the participant  $p$  will have for the entire journey, and  $\text{cap}_p$  is a soft threshold of maximum number of tasks that  $p$  is willing to serve simultaneously.

The pseudocode detailed in Algorithm 1 shows how our score function is used for recruiting participants while considering load balance. In the algorithm, a participant is selected in each round, where the size of round  $S$  is capped by the size of the query routes  $|R_c|$ . The vehicle selection decision is based on the vehicle's weight  $w$ . The weight of the participant  $p'$  is calculated by its workload score times its route coverage for the clients query route,

$$w \leftarrow \text{Score}(\text{load}_{p'}^{\max}) \times \text{single\_cover}(c, p').$$

We select the vehicle which has the maximum weight  $w^{\max}$  as shown in Algorithm 1 from line 7 to line 14.

The above algorithm assume availability of perfect predictions of candidate vehicles future route. Such assumption is not realistic given high potential of prediction errors. We evaluate predicted trip error based on different traffic levels using the TAPAS Cologne simulated vehicle trace. We found that as the length of a trip increases, the error in predicted locations of participant vehicles will increased (see Fig. 6). In the light traffic, prediction error increase slowly as the length in time of trips increase, whereas prediction error grows quickly in heavy traffic. We solve this issue by proposing a windowed based scheduling approach where we only schedule participants for set time windows instead of whole trips. We set the initial

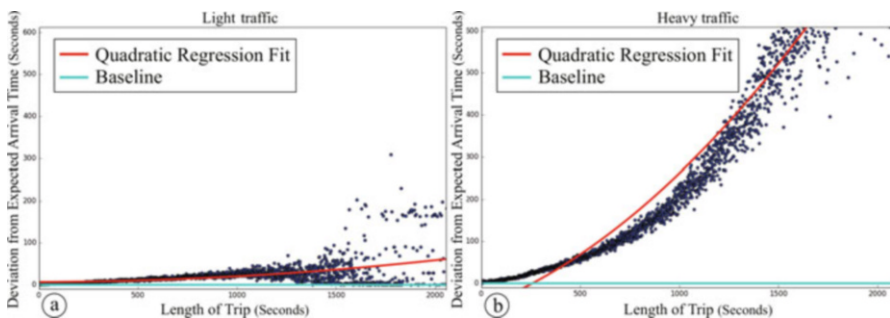


**Algorithm 1** Weight based Greedy algorithm (WBG)

```

Input  $c, P$ 
Output  $Participants$ 
1:  $Participants \leftarrow \emptyset$ 
2:  $S \leftarrow R_c$ 
3: while  $S$  is not empty do
4:    $p^{best} \leftarrow \emptyset$ 
5:    $R^{cover} \leftarrow 0$ 
6:    $w^{max} \leftarrow INT\_MIN$ 
7:   for  $p' \in P$  do
8:      $w' \leftarrow Score(load_{p'}^{max}) \times |single\_cover(c, p')|$ 
9:     if  $w' > w^{max}$  then
10:       $w^{max} = w'$ 
11:       $p^{best} \leftarrow p'$ 
12:       $R^{cover} \leftarrow \hat{R}$ 
13:     end if
14:   end for
15:
16:   if  $p^{best}$  is empty then
17:     No valid participant, break the loop
18:   end if
19:
20:    $S \leftarrow S \setminus R^{cover}$ 
21:    $update\_workload(p^{best}, R^{cover})$ 
22:    $Participants \leftarrow Participants \cup \{p^{best}\}$ 
23: end while

```



**Fig. 6** The y axis measures the difference in time from a very light traffic expected arrival time. We can see deviation from arrival time increases at different rate based on traffic

scheduling window to 300 s for each sensing task, since errors for predicted arrival times within trip length of 300 s are acceptable in both light and heavy traffic as shown in Fig. 6. However, a static window size suffers from too frequent scheduling, resulting computation resources waste and increased delays. For instance, in light traffic, a 1200 s may be better choice for the scheduling window instead of 300 s. Thus, we proposed Dynamic Window Based (DWB) solution.

**Algorithm 2** Dynamic window based scheduling (DWB)

---

```

1:  $\mathcal{P} \leftarrow$  Get participant candidates
2:  $\mathcal{C} \leftarrow$  Subscribe to the request buffer
3: for each  $c \in \mathcal{C}$  do
4:    $\{R_{c'} \in R_c \mid \forall r_k \mathcal{T}(r_k, R_c) < current\_time + SW_c\} \triangleright$  SW is calculated based on Eq. 24
5:    $S \leftarrow WBG(R_{c'}, \mathcal{P}) \triangleright$  The Weight Based Greedy Algorithm (see Algorithm 1)
6:    $Expected\_cover_c \leftarrow |QRC(S, c)|$ 
7:   Notify the client  $c$  and each  $p \in S$ 
8: end for

```

---

As shown in Algorithm 2, DWB only schedule participants for serving a client's sensing task within a specific scheduling window  $SW$ . The size of  $SW$  is calculate as the following:

$$SW_c \leftarrow \begin{cases} cover\_rate < 1 & \text{MAX} \begin{cases} SW_c \times cover\_rate_c \\ SW_{min} \end{cases} \\ \text{otherwise} & SW_c \times \theta \text{ where } \theta \in \mathbb{R}^+ \end{cases} \quad (24)$$

$\theta$  is a positive multiplier controls how fast  $SW_c$  is grow. Note that to predict the next  $SW$  size, we need to know the coverage performance of the previous  $SW$ . In such case, we assume that the client calculated the coverage rate before submit the next rescheduling request, where the coverage rate is calculated as the follow:

$$cover\_rate_c \leftarrow \frac{Actual\_cover_c}{Expected\_cover_c} \quad (25)$$

Each client has its own scheduling window due to various of driving behavior. The scheduler continuously adjust the size of  $SW$  until the corresponding sensing routes are fully scheduled. The flow of DWB scheduling is shown in Fig. 7.

## 4.5 Experiment Setup

To evaluate the performance of our proposed algorithm, we utilized the TAPAS Cologne dataset, one of the largest traffic simulation dataset. It consists of data simulated over a 24 h period for the city of Cologne, Germany. It covers over 1000 square kilometers[19]. To reduce simulation time, we restricted ourselves to a data subset which consist of 7200 time steps from 6 AM to 8 AM. During this period about 34,000 unique vehicles were part of the simulation. To obtain the trace of vehicle positions as predicted path, we utilized the SUMO traffic simulator [13]. Figure 8 shows the workflow of our experiment. For simplicity, we assume all the PVC clients are drivers, that is, we randomly select a subset of vehicles as clients from the 34,000 unique vehicles. We utilize a uniform random distribution to

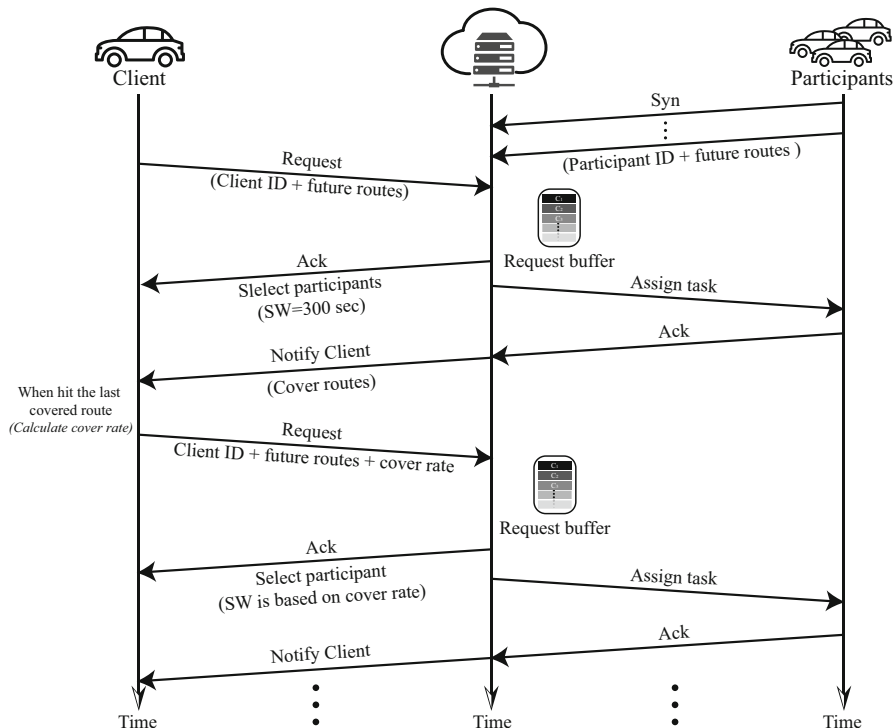


Fig. 7 Dynamic window based scheduling workflow

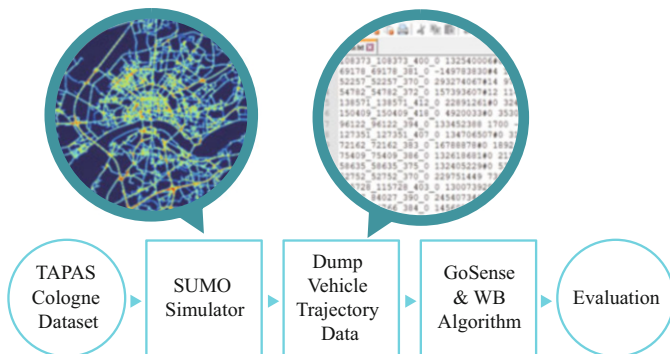


Fig. 8 Experiment flow

select 30,000 clients whereby each client’s sensing route range from 1 to 125 road segments. When clients begin their journey, client send their future sensing routes and sensing objectives to the server. The system then schedule participants to cover clients’ sensing routes in a first-in-first-serve manner. A vehicular client cannot be the participant for its own task but can be a participant for other tasks. For parameter

setting, we set the budget to be constrained by  $\beta_c = 20$ , and we set the soft parallel task constraint for each vehicle to be  $cap = 10$ . We evaluate the performance of the proposed algorithm by comparing against the PR algorithm in GoSense [22]. The simulation programs were written in Java and utilized only a single thread. When evaluating the run time of the algorithms, each algorithm was run alone. The machine utilized has i5-4590k 3.30 GHZ and 16 GB of RAM.

### 4.6 Main Results

In this subsection, we compare the performance of GoSense and our proposed weight based algorithm. We utilize the number of participants required for completing a task and maximum peak workload among all participants as metrics to evaluate the algorithms. The peak workload is defined as follows:

$$Peak\ workload \leftarrow \arg \max_{p' \in P} load_{p'}^{max}$$

As shown in Fig. 9, WBG outperforms GoSense in terms of load balance. We also evaluated how data timeliness constraints  $\delta$  and  $\epsilon$  influenced the performance. The size of the window is defined as:

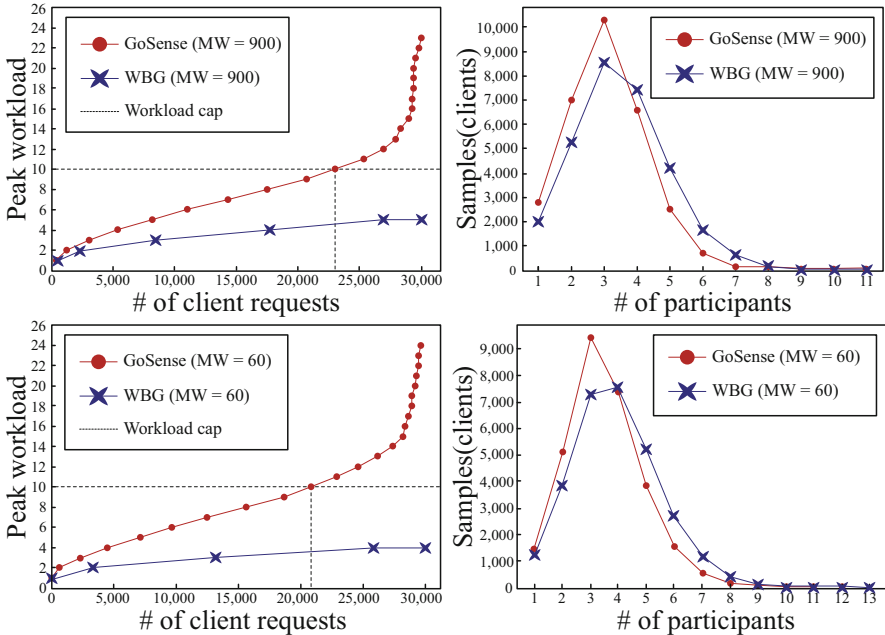


Fig. 9 Comparison of GoSense and WBG algorithm performance. Note: MW ← monitoring window (second)

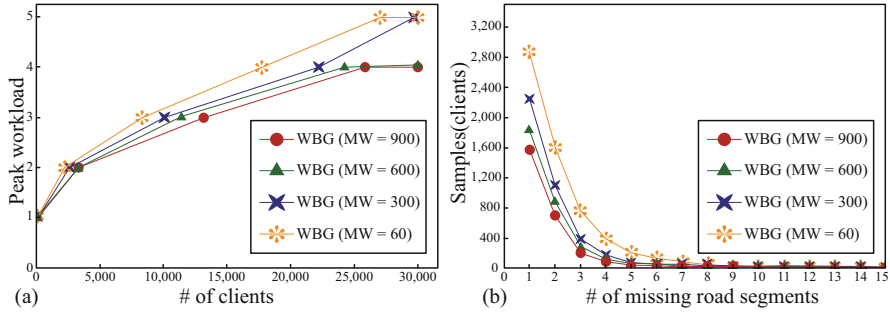


Fig. 10 Simulation results under different monitoring window (MW): second

$$MW \leftarrow \varepsilon - \delta$$

Given different sizes of MW, WBG still outperformed GoSense. We also observe that GoSense requires slightly less participants but also causes significantly heavier peak workload in comparison to WBG. Thus, we can trade off increase in participants for better load balancing.

Figure 10 depicts how MW influence the performance. As we can see in Fig. 10a, wider MW will result in smaller peak workload. That is, because wider MW will yield more potential participants for selection as shown in Fig. 3c; the workload can be more easily spread between participants. We also found that the overall number participants decreased when the size of monitoring window is decreased as shown in Fig. 9. This is because when the size of MW is small there is high chance that the region does not contains the suitable participants which result in uncovered region. As depicted in Fig. 10b, we can see that as MW narrows, the number of missing route segment covered increases. This is because the time interval which a participant is considered valid is reduced. Thus at every time step, less participants are considered.

### 4.7 The Execution Time

Next, we consider the execution time of both algorithms under the same conditions. We explore run time under different number of participants and different road length. Here, number of participants refers to the vehicles which can be selected for PVC tasks. Hence, a road length of 50 would require coverage in 50 different road segments. In general, we can see that as the number of participants or road length increases, run time increases linearly. This matches well with our complexity of  $O(S \times R \times R_p)$ . As we can see from Fig. 11a, b, when considering the number of participants, the run time of WBG is only slightly above GoSense. In contrast, the

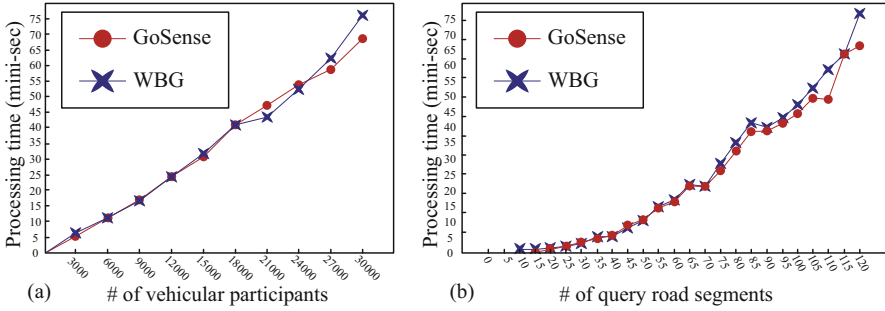
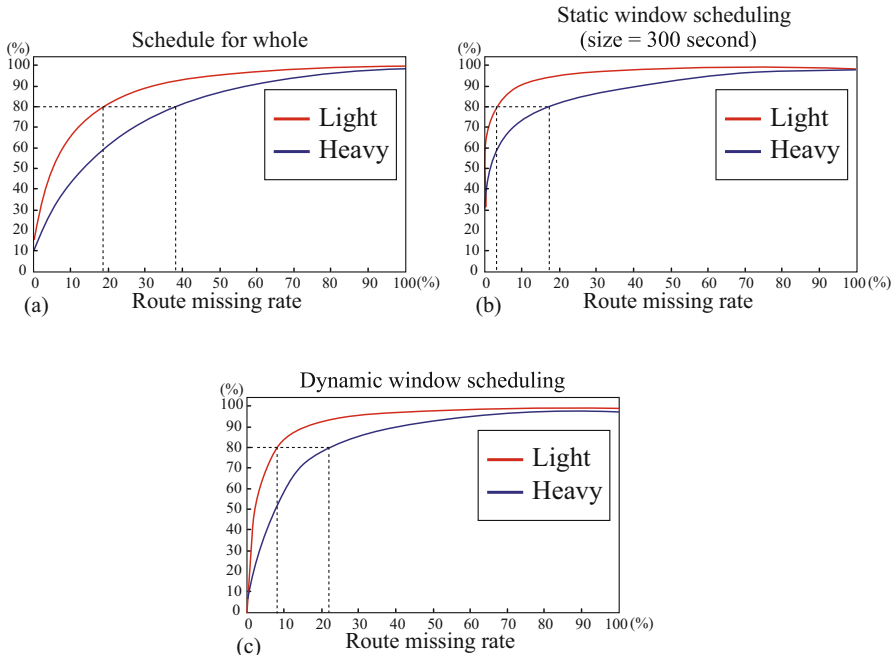


Fig. 11 Running time under different number of participants and length of request route segments

number of query routes seems to show no impact between either algorithm. Hence, we have shown that utilizing WBG does not result in a heavier overhead compared to GoSense.

### 4.8 Window Based Scheduling (Static vs. Dynamic)

In this section, we explore the improved WBG algorithm using window based scheduling. When experimenting in this section, we utilized a simple position prediction algorithm using average route speed. Hence, significant errors exist when predicting for a longer time periods. Figure 12 shows the performance in terms of route segments missing rate in Cumulative Distribution Function (CDF) graph format. Here, route segment missing rate is defined as  $route\ missing\ rate_c \leftarrow 1 - cover\_rate_c$ . As we can see, scheduling for whole route leads to the worst performance. Using 80% of vehicles as comparison point, scheduling for whole results in 20% and 40% route missing rate compared to 5% and 20% for static window scheduling and 10% and 20% for dynamic window scheduling. Overall, static window scheduling performs best for missing route rate. There is a trade off; static window scheduling requires fairly frequent scheduling and increase computational overhead. Thus, if we care about reducing the number of scheduling, dynamic window scheduling reduces the scheduling count as seen in Fig. 13. Furthermore, the performance of dynamic window scheduling remains within reach of static scheduling with more frequency scheduling. Thus, we have shown dynamic window scheduling can be a competitive option when considering heavy overhead of scheduling for all vehicles.



**Fig. 12** Results show in Cumulative Distribution Function (CDF) graph under random monitoring window. The random number is generated using the same seed. The route missing rate = number of uncover route segments  $R_c^{miss}$  divided by the total query route segments  $R_c$ . For calculating  $SW$ , we set  $\theta = 2$  and  $SW_{min} = 300$  for our dynamic window scheduling

## 5 Future Works

Previous works have proposed various methods and metrics for selecting sensing participants when the trajectory has some level of certainty. However, such sensing are considered opportunistic; no solution proposed so far seeks to actively suggest new routes to the participants to minimize sensing coverage. For instance, if a participant’s current trajectory collects information that is already known; that participant would make a small overall contribution. However, by taking a detour, the participant could greatly increase the contribution to the sensing task. The advantages are unmistakable; by rerouting participants, areas without possibility of sensing in the old model can be sensed. Overall the sensing quality can be improved. However, significant challenges lie ahead. First, a reward system must be designed to incentivize users to tolerate rerouting while minimizing costs. Unlike participants selected to sense opportunistically, rerouting consumes a participant’s time and requires direct participation. Second, a new route generation algorithm must be devised to suggest routes which increase overall sensing coverage to potential participants while minimizing overlap and costs. Finally, a complete vehicular

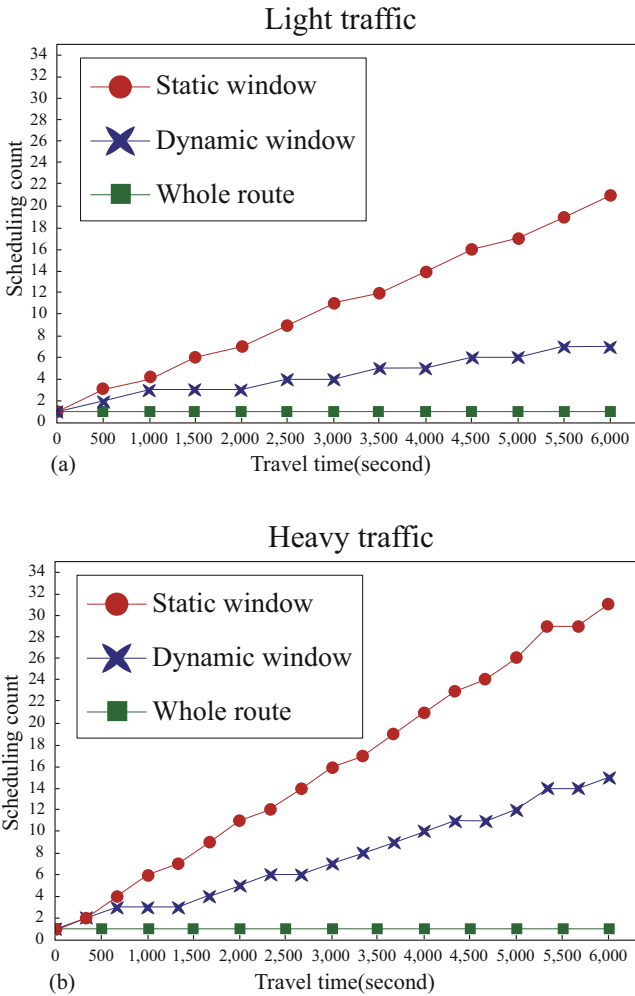


Fig. 13 Scheduling count difference under light and heavy traffic

crowdsensing system should combine both opportunistic sensing and participatory sensing. How to seamlessly incorporate both types of sensing will prove to be an intriguing challenge.

## 6 Conclusion

Vehicular crowdsensing is certain to play a major role in sensing data collection for proactive services in smart cities. Because vehicles are highly mobile, each vehicle can quickly gather sensing information over large regions. However, challenges



remain; current network infrastructure and vehicular networking technologies cannot support large-scale vehicular crowdsensing due to its bandwidth requirements. Luckily, it is unnecessary to select all vehicles as participants; sensor coverage intersects and produce duplicated data. Thus, we can minimize bandwidth use by selecting the best participants from the available ones. The selection criterion can be diverse; this includes spatio-temporal coverage, distance to point of interest, or even combined coverage of heterogeneous sensors. Thus, the selection problem consist of maximizing these measures while minimizing or staying within a budget limit. In personalized vehicular crowdsensing, load balance criteria is also included because of local processing needs. However, work still remain in the area of participatory vehicular crowdsensing; actively suggesting routes to participants for sensing can significantly improve sensing coverage.

## References

1. Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper, Mar 2017.
2. S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan. A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications*, 37:380–392, 2014.
3. L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from <https://crawdad.org/roma/taxi/20140717>, July 2014.
4. M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon’s mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.
5. C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan. A comparative survey of vanet clustering techniques. *IEEE Communications Surveys & Tutorials*, 19(1):657–681, 2017.
6. Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X. Liu, J. Bu, and C. Chen. Mosaic: A low-cost mobile sensing system for urban air quality monitoring. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.
7. Google. Waze mobile, 2017. <https://www.waze.com/>.
8. S. A. Hamid, H. Abouzeid, H. S. Hassanein, and G. Takahara. Optimal recruitment of smart vehicles for reputation-aware public sensing. In *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pages 3160–3165. IEEE, 2014.
9. K. Han, C. Chen, Q. Zhao, and X. Guan. Trajectory-based node selection scheme in vehicular crowdsensing. In *Communications in China (ICCC), 2015 IEEE/CIC International Conference on*, pages 1–6. IEEE, 2015.
10. Z. He, J. Cao, and X. Liu. High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 2542–2550. IEEE, 2015.
11. C. Hu, M. Xiao, L. Huang, and G. Gao. Truthful incentive mechanism for vehicle-based nondeterministic crowdsensing. In *Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on*, pages 1–10. IEEE, 2016.
12. M. Hu, Z. Zhong, Y. Niu, and M. Ni. Duration-variable participant recruitment for urban crowdsourcing with indeterministic trajectories. *IEEE Transactions on Vehicular Technology*, 2017.
13. D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

14. Y. Liu, J. Niu, and X. Liu. Comprehensive tempo-spatial data collection in crowd sensing using a heterogeneous sensing vehicle selection method. *Personal and Ubiquitous Computing*, 20(3):397–411, 2016.
15. D. Peng, F. Wu, and G. Chen. Pay as how well you do: A quality based incentive mechanism for crowdsensing. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 177–186. ACM, 2015.
16. S. Reddy, D. Estrin, and M. Srivastava. Recruitment framework for participatory sensing data collections. In *International Conference on Pervasive Computing*, pages 138–155. Springer, 2010.
17. L. Shao, C. Wang, Z. Li, and C. Jiang. Traffic condition estimation using vehicular crowd-sensing data. In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8, Dec 2015.
18. S. Ucar, S. C. Ergen, and O. Ozkasap. Vmasc: Vehicular multi-hop algorithm for stable clustering in vehicular ad hoc networks. In *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pages 2381–2386. IEEE, 2013.
19. S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas. Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5):1061–1075, 2014.
20. M. Wu, D. Ye, S. Tang, and R. Yu. Collaborative vehicle sensing in bus networks: A stackelberg game approach. In *Communications in China (ICCC), 2016 IEEE/CIC International Conference on*, pages 1–6. IEEE, 2016.
21. K. Yi, R. Du, L. Liu, Q. Chen, and K. Gao. Fast participant recruitment algorithm for large-scale vehicle-based mobile crowd sensing. *Pervasive and Mobile Computing*, 2017.
22. T. Y. Yu, X. Zhu, and H. Chen. Gosense: Efficient vehicle selection for user defined vehicular crowdsensing. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5, June 2017.
23. X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao. Incentives for mobile crowd sensing: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):54–67, 2016.
24. X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li. Free market of crowdsourcing: Incentive mechanism design for mobile sensing. *IEEE transactions on parallel and distributed systems*, 25(12):3190–3200, 2014.
25. D. Zhao, H. Ma, L. Liu, and X.-Y. Li. Opportunistic coverage for urban vehicular sensing. *Computer Communications*, 60:71–85, 2015.

# Towards a Model for Intelligent Context-Sensitive Computing for Smart Cities



Salman Memon, Richard Olaniyan, and Muthucumar Maheswaran

**Abstract** Smart cities is a concept that can be interpreted in many ways. One of them is to consider it as leveraging the wireless and wired Internet to streamline the operation of city-wide infrastructures to maximize their operational efficiencies and offer new services to the citizens. Many existing or ongoing smart city realizations follow this interpretation. Another more futuristic interpretation is to consider it as a large-scale context-sensitive computing infrastructure that hosts heterogeneous programs and enables the programs to interact with each other in a variety of different ways. In this chapter, we pursue such a futuristic interpretation. We are proposing a computing model for smart cities that brings together cloud computing, fogs, and mobiles to support intelligent context-sensitive computing. Our computing model has two components. The first component is a hierarchical abstract machine that spans the cloud, fogs, and devices, which can scale from a single device to many thousands of machines. The second component is an implicit learning module that observes selected data within the abstract machine to learn their characteristics. Because the implicit learning module can provide predictions based on the data in a context-sensitive manner, in certain scenarios applications can get by without explicitly incorporating machine learning into their design. In this chapter, we motivate the need for such an intelligent context-sensitive model, describe the components of the model, and present some early results.

---

S. Memon · R. Olaniyan · M. Maheswaran (✉)  
School of Computer Science, McGill University, Montreal, QC, Canada  
e-mail: [salman.memon@mail.mcgill.ca](mailto:salman.memon@mail.mcgill.ca); [richard.olaniyan@mail.mcgill.ca](mailto:richard.olaniyan@mail.mcgill.ca);  
[maheswar@cs.mcgill.ca](mailto:maheswar@cs.mcgill.ca)

© Springer Nature Switzerland AG 2018  
M. Maheswaran, E. Badidi (eds.), *Handbook of Smart Cities*,  
[https://doi.org/10.1007/978-3-319-97271-8\\_8](https://doi.org/10.1007/978-3-319-97271-8_8)

## 1 Introduction

Many countries around the world are pursuing smart city projects (e.g., City Brain in Malaysia<sup>1</sup>) that aim to develop city-scale cyber-physical infrastructures to create sustainable, environmentally friendly cities that can efficiently meet the requirements of their residents [34]. Such infrastructures need to meet several challenges; prime among them is handling the large volumes of data generated by the cities at very high rates. The applications running on smart city infrastructures will also require access to the generated data in different time granularities. For instance, video surveillance would need real-time data feeds to track an emerging scenario and air quality modeling would require access to large volumes of cross-sectional data over long-time durations. By supporting the different types of data accesses equally well, the computing infrastructure can host variety of different applications simultaneously in a common platform.

Given the large-scale of a smart city and high data volumes, cloud computing [2, 7, 21] is expected part of the mix. However, clouds aggregate all their resource capacities in few data-centres putting significant burden on the wired and wireless networks that connect the computing backend to the sensors and actuators distributed throughout the city that generate and consume the data. To address this problem, in recent years, fog computing [5, 6] that brings the resources closer to the edge has been proposed. It is expected that fog computing could bring enormous benefits to the applications by reducing the latencies of accessing the computing backend and providing context-sensitive computing (CSC) [1] tasks. By distributing the computing resources towards the edge, fog computing can significantly reduce the network traffic, which reduces network congestion and improves the network reliability.

While the introduction of fog computing has benefits for smart cities [13], it brings many interesting challenges as well. One of them is the need to develop programs that can organically support fog computing and optimally use the edge resources. In particular, with mobile clients that are prevalent in a smart city scenario, the applications need to switch the fogs as needed to benefit from closer computing resources. When switching the fogs, we need to take the computational state of the applications at the fogs and the placement of the input data objects into consideration. In addition to switching between fogs, we also need to consider switching between the fogs and clouds because some workloads are better suited for the clouds while others are better suited for the fogs. In certain cases, even a single application can have phases of the computation that are better suited for the fogs while the rest is better suited for the clouds. Therefore, the programming language and runtime should be able to support fine-grained task (function) mapping to the fogs and cloud while taking the data locations and program state into consideration.

---

<sup>1</sup><https://techcrunch.com/2018/01/29/malaysia-alibaba-city-brain/>

In a smart city scenario, a large fraction of the fog-to-device connections will be realized using a WiFi or cellular network and the rest using wired networks [25]. The cloud-to-fog connections, on the other hand, will mostly use high-speed wired networks. In certain cases where the fogs are placed in far away locations or inside moving platforms (e.g., ships or trains) cellular connections could be used to connect the fogs to the cloud. Handling disconnections between devices, fog, and cloud is an important problem that needs to be supported by the programming language runtime.

Sensing, processing, and archiving data are major operations in a smart city platform [44]. Location and time are two important characteristics of the data in smart city applications. For example, environmental parameters such as air quality and noise level will be related to given locations and times. In addition, activity measurements such as number of taxis available in a given city block or energy usage for lighting a road segment are functions of location and time as well. With the many disparate data streams describing various components of the smart cities, we need a coherent vocabulary to interconnect the data and perform data integration. With a large-scale dynamic environment like a smart city, such a standardized vocabulary or ontology is a massive undertaking. However, efforts such as READY4SmartCities<sup>2</sup> are underway. Despite the best efforts, we believe, such standardized semantic notions are only suitable for a small portion of the data that would be available in a smart city environment. Namely, it will be best suited for services offered by regular or official actors in a smart city. For example, if transportation data is collected and collated together to form an integrated database, buses, metros, and trains could be easily captured by such a standardized effort. In an evolving smart city ecosystem, we are bound have non-standard actors providing equivalent services: taxis, ride-sharing services, bike-sharing services, or private rides. It is very difficult to bring such information feeds into a regular structure as they could be available in applications, notice boards, etc.

In this chapter, we develop novel architecture for smart city cyber-physical computing infrastructure. The major difference between our architecture and existing ones is the incremental discovery scheme proposed to uncover the relations that exist among the data from different localities. The genesis of this architecture is a computing model for wide-area applications where the application shares selected data streams with the runtime. The data streams themselves contain data the application has extracted from the location where it is running. The runtime feeds the data streams into a neural network (NN) it instantiates for the given location and learns the characteristics of the data. If the NN is able to successfully learn the characteristics of the data as measured by its ability to predict the future values of the data streams, it will remain in the locality. That is, as the applications running in the devices roam in a smart city, NNs will be created in the different localities and only the ones that are able to learn the characteristics of the data will remain out of the created ones.

---

<sup>2</sup><http://ready4smartcities.eu>

We present initial results in creating such NNs in different localities and sketch out an overall architecture. Future work is necessary to develop the proposed architecture for wide-area inferences as well as local-area inferences which is possible using a given NN at the specific location. Another interesting aspect of the proposed architecture is the symbiotic relationship between the applications and the NNs maintained by the runtime. The NNs depend on the data feeds from applications that visit their localities. The application, in turn, benefit from the predictive power of the NNs they nurture by using them to predict future values.

## 2 Motivating Scenarios

In this section, we discuss some motivating scenarios for CSC in smart cities. Lets consider a future scenario where drones are providing personal transportation from a given point to another point within a city. The flight time and the ability to safely land in a given point in the city could be affected by wind, air temperature, and drone traffic congestion. The drones need to have the most accurate predictions for these parameters in the localities that intersect with their flight path with particular importance given to the beginning and ending of the flights. If the drone service relies on a global weather service that covers the whole city, its predictions may not accurately track each location because the update processing rate of the global weather predictor will limit how accurately it could track the changing local conditions. Using a local-area CSC, we could instantiate a predictor for each vicinity and such predictors can be updated very frequently to track the changing conditions. Also, in the proposed CSC design, the predictors can be updated in a parasitic manner by applications that gather data for their own purposes. That is the wind speed and air temperature readings need not be solely coming from sensors deployed by the drone transport company. By leveraging CSC the drone company can get more accurate predictions that is tolerant to failure of the drone company deployed sensors. Also, CSC predictors can be instantiated in many more locations than the drone company sensor locations.

## 3 System Architecture

The system architecture we propose here for CSC is based on the following assumptions:

- A smart city has fogs distributed throughout its area with more fogs located in areas where the device concentrations can be high.
- The devices are normally mobile although a sizable fraction of the devices could be fixed such as the sensors attached to buildings, bridges, etc.

- The fogs are connected to the cloud by a fast backhaul network that could be wired or wireless.
- The applications that get deployed on the cloud, fogs, and devices infrastructure share the available resources. They expect operating system or middleware support to divide the available resources in a fair manner among the applications that are competing for access such that all applications are progressing with their executions.
- Disconnections are possible between the fogs and cloud or fogs and the devices.

The goal of the CSC architecture is to run the applications such that the opportunistic data flows are used for learning at the edge. The CSC architecture wants to support the creation of many neural networks at different physical localities and sustain them if learning is feasible in those localities. The neural network that is instantiated in the different localities is quite generic and is supplied by the CSC; that is, the application developer is not selecting or fine tuning the neural network. Due to the automatic deployment strategy, in some localities and depending on the captured data, neural networks won't be viable because they are unable to learn the characteristics of the data and provide usable predictions.

The central issue in CSC architectures is the data sharing mechanism between the applications and the runtime that instantiates the neural network. Figure 1 shows data depots (that are implemented using in-memory key-value stores such as Redis [9]) that hold the data shared by the applications. The data depots are located at the devices, fogs, and cloud. When an application shares some of the data it acquires for its own use with the runtime, that data is routed through the data depots towards the root of the tree of machines shown in Fig. 1.

The runtime can instantiate neural networks at the devices, fogs, or the cloud. The cloud can be located far away from the application running locations where

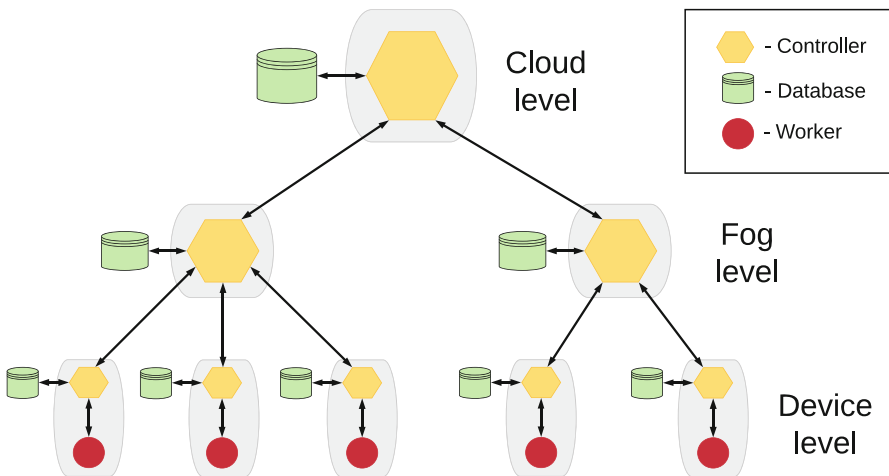


Fig. 1 Hierarchical model consisting of the cloud, fog and device levels

the actual data is generated. This means the data need to be shipped through many network hops causing long delays before processing and high transmit cost. The devices, on the other hand, while being very close to the data generation location have other disadvantages. The CSC relies on data sharing; particularly, with the neural network leveraging data feeds from many applications to autonomously learn interesting characteristics of the data for the given locality. Therefore, fogs are ideal candidate locations to host the neural networks. A fog-hosted neural network would be quickly reachable from a device so that the device could benefit from the predictions the neural network could provide.

The primary enabler for CSC is the *logger* construct provided in JAMScript [40] for sharing data between devices, fogs, and cloud. In this mechanism, the application programmer selects data variables that needs sharing between devices and fogs or between devices and fogs and the cloud. There could be different reasons for data sharing. One of them is where devices want to inform the fogs about relevant data parameters. For example, in an air quality sensing application, the device would be housing the sensor that captures the air quality. The captured values need to be periodically transmitted to the fogs for further processing and/or comparison with values sensed by other devices. Another is where the devices want to persist an important values or results in the fog so that either that device or another device could resume the computation using those values in the event of a failure. For instance, a drone could update the fog about its current coordinates using the logger construct. When the drone fails, it could obtain the resumption point from the fog.

It is important to note that the application is capturing the data values and pushing them into the logger for its own purpose. We want a neural network instantiated by the runtime to learn the characteristics of the data by observing the data feed. For instance, by observing the flight paths of the drones the neural network could predict the future locations of the drones. This could be used to regroup the drones after a disruption of one or more drone flights.

One problem that needs to be solved to connect the data feeds to the neural networks is identifying the data feeds and bin them to the appropriate neural network. For instance, air temperature at a location can be read by many different applications that label it using different terms. Therefore, we need alias resolving mechanisms that will detect similar data feeds independent of the labels and forward the data values to the right input of the right neural network. One solution is to associate descriptions with the applications on the data feeds they would share with the runtime (i.e., the loggers created by them) and match the descriptions using appropriate ontologies if necessary.

JAMScript uses a hierarchical computing model as shown in Fig. 1, where different components of a single application are connected in a tree. That is, the components running in the devices connect to corresponding components running in the fogs while the components running in the fogs connect to the cloud. A built-in discovery service is used by the components to find the corresponding components and automatically connect to them when the opportunity arises. This auto connect feature also reconnects in the event of recovery from failures. A device could find multiple options when it wants to connect to a fog. The runtime can use different



strategies in picking a fog among the many available options such as selecting the closest fog or selecting a fog closer and not used by other applications.

The hierarchical computing model followed by JAMScript makes it highly suitable for wide-area deployments that need to cover large areas such as smart cities. The applications running in a wide-area configuration can generate data feeds from many localities that are far apart. For instance, a transportation tracking application running at the bus stops and buses could gather large volumes of data that include information about a cross-section of a city. Although the CSC is focusing on localized deployment of neural networks, in future works, we could extend it to wide-area deployments by leveraging the wide-area presence of JAMScript applications. One interesting aspect of data feeds generated by JAMScript is the ability to programmatically synchronize the data capture operations. This will help the data integration steps to bring the whole data into a coherent information base.

One of the key ideas in CSC is the adaptive deployment of neural networks at different smart city locations. It is important to note that the neural networks are fed data that is captured by the applications for their own purposes. If very few applications are visiting a particular location or the neural network is unable to learn effectively from the captured data, we remove the network instance from that location. We could also transfer a trained network from another similar location to the given location.

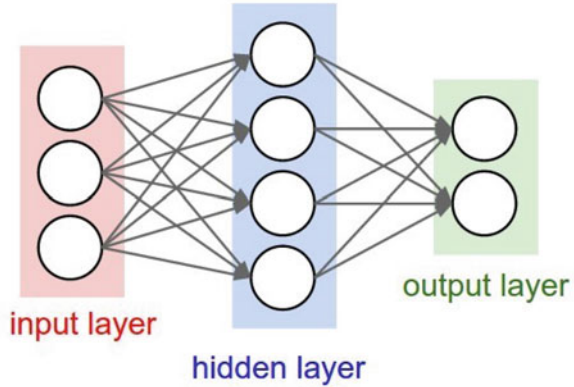
## 4 Machine Learning Models

Given the abundance of data being transmitted through a group of cooperating machines consisting of the cloud, fog and device levels in an IoT setting, an implicit learning module could be an invaluable resource for applications as well as the fog infrastructure. The models can exist at multiple levels, each learning from data with a different spatial context. Models higher up in the hierarchy would have a broader view of the data, while models lower in the hierarchy would be trained on local data. Finally, a governing application can exist at the cloud level that consolidates the learning from all the models, effectively creating a knowledge graph that can identify local and global correlations.

The implicit learning module is envisioned as a black box for the applications that instantiates the models, requiring minimal input from them other than the data and some domain-specific hyper-parameters. These models can exist for a variety of purposes that the applications can choose to utilize, such as learning patterns and characteristics to predict or classify data. As an application logs data to the different machine levels, and hence the learning module, the models learn and could be queried to produce an output for the application to make use of.

A proof of concept model was created along these lines, designed to learn time-series data streams logged by an application and forecast future values for the applications to use. The model would exist at a fog node, with the application identifying the data it wanted to train the predictive model on. Deep learning

**Fig. 2** A simple neural network with one hidden layer. Each connection is associated with a weight parameter that is adapted over the course of learning. Each circle is a neuron that transforms the input according to the activation function being used [3]

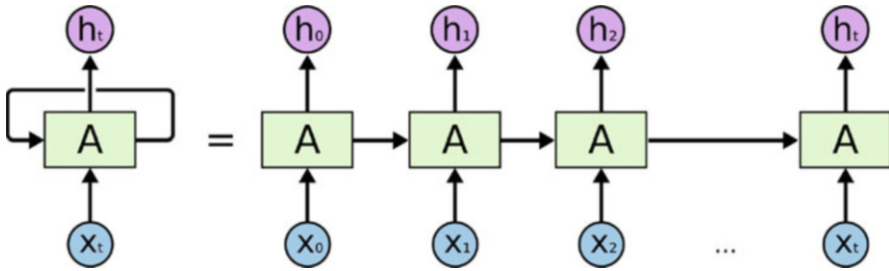


architectures have been in the limelight recently, for their prowess in learning in data-rich environments, surpassing other algorithms in complex tasks such as image recognition, speech recognition and sequence analysis and prediction [18]. This motivated us to utilize neural networks, specifically Recurrent Neural Networks (RNNs) with long short-term memory (LSTM) cells for our prediction model.

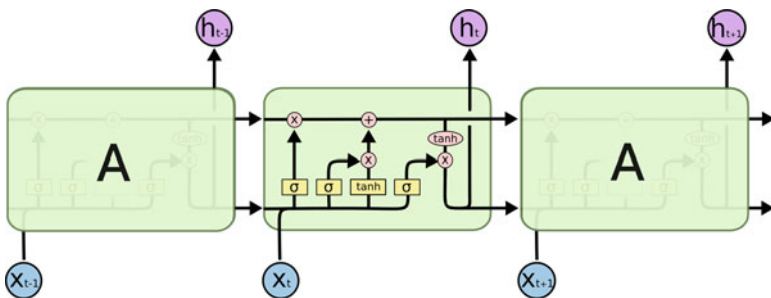
Artificial neural networks are widely used for learning functions and behaviors by vaguely mirroring the structure of biological neural networks. They consist of a number of processing units called neurons. These neurons are contained within layers and connected through weighted connections. Each neuron is further associated with an activation function that transforms the weighted inputs and can give NNs the ability to model non-linear functions. Over the course of learning, the weights are optimized through optimization algorithms such as stochastic gradient descent coupled with backpropagation, allowing the neural network to model the output behavior. Figure 2 shows a simple neural network with a single hidden layer.

Recurrent neural networks are a variant of neural networks that are expanded in time, sharing the hidden information between units. This allows them to pick up temporal patterns in the input data, making them especially well suited for modeling time-series and sequences. They are further empowered with the use of long short-term memory units, that help them retain long-term dependencies. LSTM units are complex structures, containing gates that control the influence of inputs, hidden states of previous units and outputs of previous units. They are widely used in sequence prediction tasks, especially in the domain of natural language processing. Figure 3 shows a recurrent neural network unfolded in time and Fig. 4 shows the structure of an LSTM cell.

In addition to superior performance, neural networks offer another major advantage over conventional machine learning techniques. They generally require little feature engineering and are often able to learn effectively on raw data [18]. Feature design is often highly application specific, requiring domain knowledge about the task and would have limited the types of data our model could learn without a lot of user input.



**Fig. 3** A recurrent neural network unrolled in time [4]. Each block can be a feed-forward neural network or LSTM cell. The arrows indicate a transfer of the hidden layer information from one unit to the next. This allows information from a series of inputs to persist through the network



**Fig. 4** The structure of an LSTM RNN taken from [4]. The interconnected units transfer learning forward. The units themselves consist of multiple gates that vary the influence of the unit’s input, previous unit’s hidden information and previous’ units output

### 4.1 The Dataset

The dataset used for this model was an air quality dataset from the publicly available UCI machine learning repository [11]. It contains 9358 data points, each providing a measure of 10 different pollutants observed by a roadside in a highly populated city, as well as temperature and humidity readings, averaged over every hour. The dataset was representative of what we might observe devices and applications using in a smart city scenario.

### 4.2 Model Design

Our model was designed as an autoregressive predictive model, forecasting future values as a function of past values in the time-series. The foremost phase of the model design was translating the time-series data into a supervised learning problem. Supervised learning associates each input with a known desired output, or

target. In our model, the input data consists of vectors of past values and the target the value one time-step in the future.

Choosing the right window size (the number of past values to base the prediction on) is an important parameter for prediction accuracy. Moreover, it also tends to be domain specific and should ideally be specified by the application. To identify a suitable window size, we used the autocorrelation and partial autocorrelation plots of the time series. These plots provide a measure of the correlation between a value with the past values in the time series and are considered as basic tools for time series analysis and forecasting [19]. Previous values with a high degree of correlation were included in the window size.

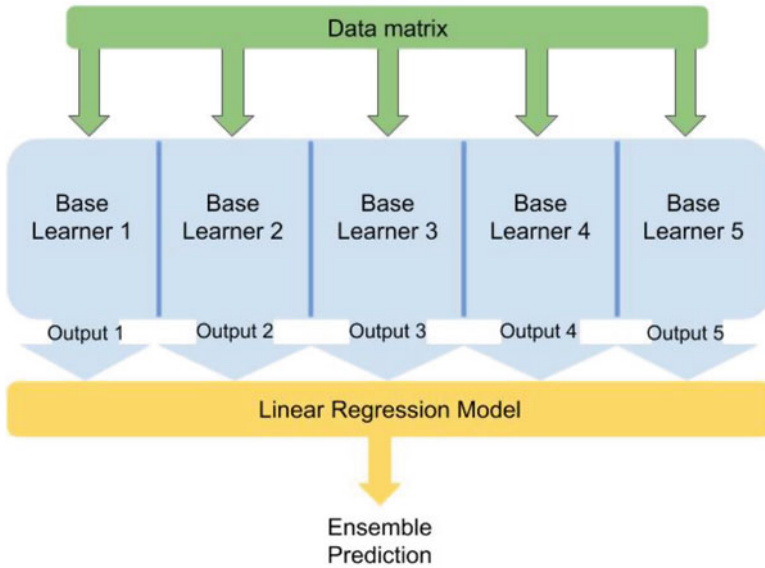
Once a window size was identified, the data stream was framed into a learning problem where each example of the input matrix was a vector the length of the window size. The target vector was the value found one time-step ahead of the window. The data was scaled between 0 and 1 which is a feature transformation know to aid in prediction performance and reduce convergence times in neural networks [31].

To maximize the performance of our model, we chose to deploy an ensemble of LSTM RNNs. An ensemble is a collection of learning models, called the base learners, producing a set of hypotheses which can then be combined to produce a result. Machine learning ensembles offer superior performance to the base learners, reducing both bias and variance [43], both of which are important considerations in machine learning. A high bias is associated with poor performance and a high error rate between the model's output and the target value. A high variance is associated with overfitting, where the learning model fits too closely to the training data and fails to perform well new data it has never seen before. In addition to the performance gain, an ensemble of learners is best-suited for deployment in a group of coordinated machines as the complexity of the data being learned is an unknown. So by having an ensemble of LSTM RNNs with different levels of complexity, we can cater to a wider range of data without the need for application-specific changes to the model's architecture.

While ensembles are quite very popular in classification problems, they are less common in regression problems such as ours. A linear regression model was conceived to combine the output of the ensemble. The training data for this model is a vector containing the output of each of the RNNs while the target is the future value, allowing the linear regression model to optimize weights to favour the models that worked the best. The structure of the ensemble is shown in Fig. 5.

### ***4.3 Implementation and Performance Analysis***

The model was implemented in python using the popular deep learning API Keras [10] with a Tensorflow backend. We choose an ensemble of five LSTM RNNs of varying complexity in terms of the number of units and drop out regularization rate. Dropout regularization [32] is a method to help avoid overfitting in a model by

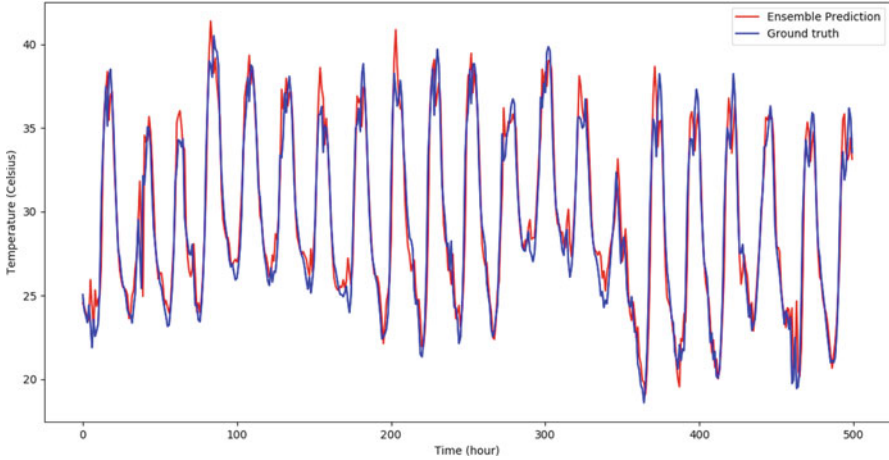


**Fig. 5** The ensemble model with five base learners. Each base learner was a double stacked LSTM RNN

disabling a percentage of random neurons from the training process. This percentage of neurons is defined by the dropout rate. As with most hyper-parameters in machine learning, the dropout rate is usually application specific and chosen using methods such as cross validation. Using different values of regularization within our base learners allowed us to avoid this step. We also chose a batch size of 256 to speed up the convergence rate for the learning. In an actual deployment, the batch size can be chosen based on the rate at which data is being logged to the AM and whether it is being buffered before learning.

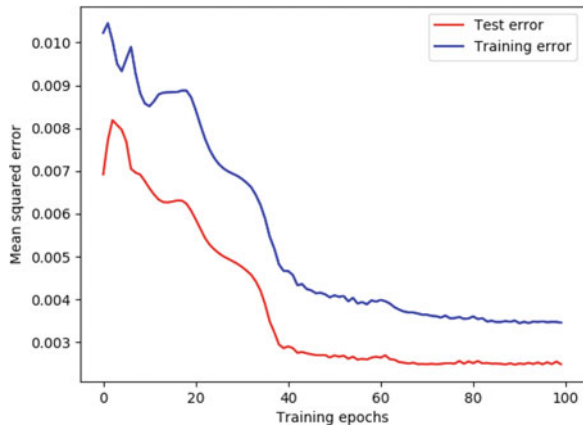
To benchmark the performance of the model, 2000 consecutive data points from the temperature feature were chosen. The data was divided into a 75/25 train-test split, with 1500 examples for training and 500 examples for benchmarking the model's performance. The Adam optimization algorithm was used for training given its low computational cost and high convergence rates, with a default learning rate of 0.001 [17]. Mean squared error was used as a cost function, typical of regression style problems. The models were trained sequentially with a 1000 epoch training horizon, although early-stopping callbacks were enabled to halt training if performance on the test data failed to improve for multiple consecutive training epochs. The results of the ensembles output on the test set is shown in Fig. 6.

Figure 7 shows the mean squared error for the training and test set over the course of training, where no overfitting was observed despite the error rate tapering off after 40 epochs. Each epoch for the ensemble consisted of 1 training epoch for each of the base-learner as well as optimizing the linear regression layer at the end. The



**Fig. 6** Prediction produced by the ensemble on a test set for the temperature data stream. Other than noise in the data, the prediction model does a good job of picking up the trends in the data stream

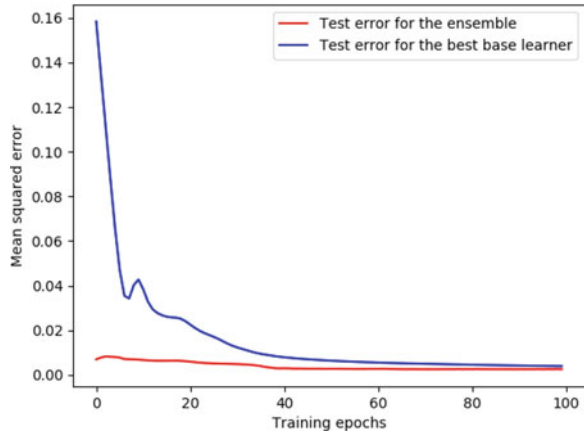
**Fig. 7** The figure shows the error rate for the train and test set for the ensemble of LSTM RNNs. The error rate tapers off after the 40th epoch. No overfitting is observed over the course of 100 training epochs



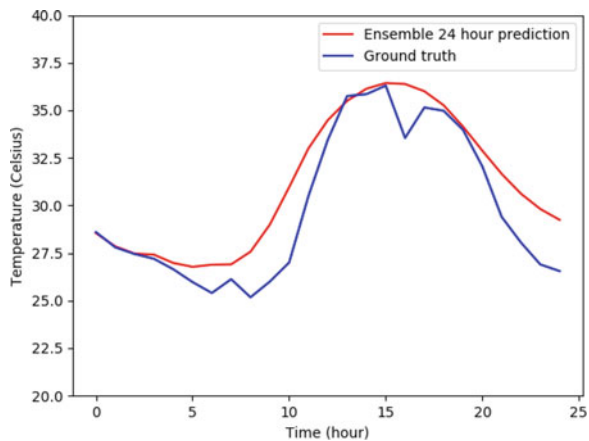
average time for each ensemble epoch was 8.13 s on a quad core 1.7 GHz CPU. In a deployment scenario, this performance can be improved many folds with more robust hardware and GPU support which can make the training process several times faster [30].

An important observation is shown in Fig. 8. The error rate for the ensemble was an order of magnitude better than the error rate for the best base learner in the ensemble in the early phases of training. This can be an invaluable feature of the ensemble model especially in latency conscious applications such as real-time analytics that would require models to perform well quickly.

**Fig. 8** A comparison of the test error rate for the best base learner with the error rate of the ensemble. The ensemble provides performance an order of magnitude better than the individual learner at the start of the training, which can be an invaluable characteristic for applications that require high quality predictions quickly, such as for real-time analytics



**Fig. 9** The figure shows the results of the model predicting 24 time steps into the future



Lastly, the performance of the model in making multi-step predictions was tested. To accomplish this, the prediction of the ensemble was inserted into the window to base the next prediction on. The results for a randomly chosen point from the test data for prediction 24 hours into the future is shown in Fig. 9. The model performed admirably for the first three prediction time steps, after which the error accumulated caused the prediction to diverge from the actual data. However, the prediction still retained the trend of the data and the results were accurate to within 2.5 °C.

While the model seems to perform reasonably well in this scenario, error would start to accumulate in the predictions over multiple cycles of prediction and eventually render the output useless. However, RNN LSTMs can be trained specifically for multi-step time series prediction by phrasing the supervised learning problem appropriately.

#### ***4.4 Multi-stream Data Prediction***

We expanded our design to the scenario where an application submits requests to learn multiple data streams. To keep the system feasible, we changed the design of the training matrix such that each sample now consisted of vectors for the window of past values for each stream. Similarly, the output matrix was also changed to correspond to a vector of targets from each stream against each training example. This allowed us to accommodate data streams with different window lengths without changes the design of our ensemble architecture. However, each data stream being learned now required a dedicated linear regression module to combine the solutions from the base-learners.

This design is more practical from a computation load point of view than having an ensemble for each stream being learned. Moreover, it suits the philosophy for fog and cloud enabled coordinated and cooperative devices as the data would be spatially and temporally correlated, allowing the learning module to take advantage of these correlations.

### **5 Challenges and Benefits of Adapting Machine Learning in Intelligent Context Sensitive Computing**

In this section, we identify challenges of adapting machine learning in smart cities and data intensive systems and highlight the benefits derived from having context sensitive learning models in real time data analytics frameworks.

#### ***5.1 Challenges of Incorporating Machine Learning***

Adapting machine learning to the data diverse environment of smart cities does have a number of challenges. A large portion of these challenges pertain to the application specific design of machine learning models. Machine learning models are generally very application specific, with hyperparameters tuned through methods such as cross validation, where multiple values of hyperparameters are evaluated on test sets. In addition to providing results that suit the particular problem, the tuning process is also slow and computationally expensive, especially for larger datasets. While having an ensemble of learners allows us to mitigate this to an extent by having a variety of base-learners with different hyperparameters, more precise tuning can yield better results in terms of reduced error rates as well as smaller training times.



Some hyperparameters are also domain specific and dependent on the nature of the data. In the case of our learning model, the size of the window for previous values to base predictions on proved to be an important hyperparameter for the model's performance. While the autocorrelation plot can certainly provide an estimate of range of suitable values, the optimal value should be determined through cross validation. This issue is further exacerbated in the multi-stream model where the optimal window sizes for each stream would have to be determined.

In some scenarios, even the architecture of the neural networks has to be changed to better suit the learning problem and the input data type. For example, convolutional neural networks are much better suited for spatially related multi-dimensional data such as images. As a result, the machine learning module would have to actively observe the nature of the data being entered for learning and assign a suitable learning model architecture. Training times and computational load could also vary radically depending of nature of the data being learned.

Missing data is also a common obstacle in machine learning and a major source of errors in models. In a smart city scenario where interruptions in the data stream would be more common, a robust strategy would be needed to deal with missing data in the training process. Although there are a variety of methods of dealing with missing data, such as estimating the underlying probability distribution and sampling the results [20], optimal results are usually found by a case by case basis.

## ***5.2 Benefits of Intelligent Context Sensitive Computing***

Integrating a context sensitive learning module offers can yield a variety of benefits for applications, devices and the communication framework. It could make real-time analytics a more feasible possibility, bringing intelligence closer to the devices and applications. It would also do so with minimal inputs from the application, so specialized knowledge of machine learning would not be a prerequisite to use the learning capabilities.

In the wider scope, where we have machine learning models dispersed throughout a smart city, it could also make transferring data much easier. A trained model from one location could provide a sufficient representation of the data without actually providing access of the data to the querier. Not only would this address privacy concerns but could also yield savings in data transmission and storage.

Moreover, transfer learning could also be employed to cut down the training time for new cloud and fog coordinated and cooperative devices. By doing a correlation analysis of the data, an existing, trained model from a related location could be deployed to the new location. This would give it prediction capabilities almost out of the box, which can radically cut down the latency associated with such applications.

## 6 Related Work

### 6.1 *Cloud Computing, Edge Processing and Smart Cities*

A decentralized and service-oriented cloud computing platform for a wide range of IoT applications was proposed in [35]. They addressed the challenge of real time large-scale data processing using cloud computing. The framework consists of an open platform for all sensors and a distributed service-based cloud platform for processing and managing data. In [26], a sensing-as-a-service model was developed for smart cities leveraging cloud computing and IoT [41] by decoupling sensors and sensing activities from service providers and sensor data consumers. Sensor owners have to publish their data to the cloud for it to be available to sensor data consumers.

Adapting edge computing to localization-based and latency-sensitive systems is becoming increasingly popular. With the rise in the quest for smart cities, it is important to employ technologies and schemes that enable fast and efficient computing as well as communication. Smart cities are expected to continuously generate a large amount of data [39], and thus the need for processing at the edge rather than cloud computing data centers. In [37], a four-tiered fog enabled architecture was proposed that leverages computing at the edge of the network. The layers are arranged bearing in mind latency of communication, data processing and storage. They argued that such an architecture will provide both localized and global services in smart cities.

A similar two-tiered architecture was developed in [36] for making cities smarter. The backend cloud provides services through cloud service providers while the edge level is controlled by mobile network operators through an edge orchestrator. They sought to make relevant data available to end devices that constantly change physical vicinity by migrating data to the appropriate edge orchestrator.

### 6.2 *Programming and Computing Models for Enabling Smart Cities*

Mobile Fog [14], a programming model for handling large scale IoT applications was developed to provide location-aware services. Components range from cloud, fog infrastructure to edge nodes that are localized to a particular physical vicinity. Tasks are location-based thus Mobile Fog serves tasks with a certain assigned neighbourhood while respecting the network hierarchical levels. Mobile fog exposes information about end devices (ranging from network topology, resource availability to location) to applications in order to perform location-based computations. Assumptions made in Mobile Fog are that fog computing infrastructure nodes are placed in the network and a programming interface is provided by the fog. Mobile Fog handles scaling by making application developers specify the scaling policy at each hierarchical level. Load balancing is based on creating on-demand fog

instances at the same level as an over-loaded fog instance. Both static and mobile edge devices are supported. Static edge devices stay attached to a single fog node while mobile edge devices change fog nodes as they move around by invoking a set of handlers.

Calvin [27] is a framework that merges IoT and cloud in a unified programming model similar to PatRICIA [24] which provides an execution environment, runtime support, development tools, policy-based automation and programming techniques at a high-level. It is an IoT programming framework that combines the ideas of actor model and flow based computing. To simplify application development, it proposes four phases to be followed in a sequential fashion: describe, connect, deploy, and manage. These phases are supported by the run time, APIs and communication protocols. The platform dependent part of Calvin runtime manages inter-runtime communication, transport layer support, abstraction for I/O and sensing mechanisms. The platform independent runtime provides interface for the actors. The scheduler of the Calvin runtime resides in this layer. Calvin runtime supports multi-tenancy. Once an application is deployed, actors may share runtime with actors from other applications.

### ***6.3 Machine Learning and Leveraging Fog Computing for Smart Applications***

While there has been some work in the domain of allowing machine learning and data analytics to take advantage of the edge computing model [38], most of the work tend to be application specific [8]. To the best of our knowledge, our context-sensitive computing model is the first to integrate machine learning services into the design of such a system.

A wide variety of fog computing applications can take advantage of machine learning, especially in the context of smart cities. Applications such as smart traffic lights [33] can benefit from the context-sensitive learning mechanism we have proposed, providing a model for the local traffic flows that caters to each traffic light. The same models can then be combined to give a global picture of the whole system. Other applications, such as smart grids can decentralize their network with fog nodes governing micro-grids [42] and the learning modules providing real-time analytics. Health care applications, such as fall detection for the elderly and aiding Parkinson's disease patients with speech disabilities [16], are another segment that can reap great benefits from the low latency and local learning offered by our model. Simple, lower powered movement detection sensors and smart watches can request their local AM for learning services through the implicit learning module. This not only improves the latency of learning module but also addresses privacy concerns of storing patient data in the cloud.

Similar works also offer solutions to some of the limitations we experienced with employing machine learning into the design of the system. We highlight that

the complexity of the machine learning model has to match the data being learned. Having overly complex models in the ensemble can increase training times and also lead to poor generalization performance. However, less complex models will tend to underfit, resulting in high error rate and poor quality prediction. One solution to this is to employ a constructive neural network algorithms such as CNNE [15]. CNNE not only caters to optimizing the number of neural networks employed in the ensemble, but also their complexity in terms of the layers and number of neurons, or units, utilized. Moreover, the algorithm starts with a minimal ensemble architecture and adding constructively, ensuring that the ensemble's complexity matches the behavior being learned. This further capitalizes on the ensemble's ability to reduce bias and variance, while giving the additional advantage of having models with the ideal complexity. The net result would be a reduction in learning times and computational load on the infrastructure.

#### ***6.4 Sensor Data Prediction Schemes***

A sensor prediction approach to underwater sensor data in [23] proposed the use of dimensionality reduction techniques to simplify the data. This technique can be utilized to reduce noise in the data stream being used for learning, and offers an automated method of doing so. It also does not require any hyper-parameter tuning such as that required for a moving average filter to reduce noise, which would require finding the optimal window size for averaging.

A recent work pertaining to forecasting sensor output shows the multitude of uses this service could provide for both applications and the infrastructure. An overview of different time series modeling techniques using deep learning is provided in [12] and gives insight into the efficacy of stacked LSTM layers, such as the ones we use in our model, in modeling time series data. The accurate predictions produced by a well trained model can then be utilized for anomaly detection, as identified in [22]. Real world observed values that diverge too much from the prediction could be used to identify problems in sensors.

The prediction mechanisms provide essential services to the sensing and communication infrastructure. Usman Raza et al. [28] proposed a prediction scheme that uses predicted values to reduce data transmissions required among wireless sensor nodes. This can produce significant energy savings for the wireless nodes which often operate on battery power. This could spell a marked increase in efficiency for smart city sensor networks where thousands of such nodes would be deployed. Similar approaches can also be applied to wearable devices, such as smart watches or body motion sensors used in health care applications, to improve their battery life and longevity. Predictions can be utilized in a similar fashion to reduce the traffic load on the network [23]. In a smart city scenario, where thousands of sensors are continuously streaming data, this can be used to help reduce transmission and reduce the load on congested networks.

This approach of reducing congestion can be further enhanced by bundling correlated streams together before the prediction module, as proposed in [29]. This approach provides a segue to exploiting the correlation of the data streams for the important task of ontology learning. The correlations determined from spatially and temporally correlated data logged into the AM can be used to create a knowledge graph of the smart city environment in conjunction with the learning module. This can provide a more natural method of ontology learning which is less dependent on the strict standardization and more reliant on the correlations in otherwise heterogeneous data.

## 7 Conclusions and Future Work

The increase in the number of devices with Internet access has caused a new revolution where more and more things and people are getting connected in smart environments. This has spurred different initiatives such as IoT and smart cities. Coordinating and managing data transfer and usage in such data intensive systems is a challenge. Issues such as data-reuse come up as we do not want redundant work and likewise there is a need for low-latency processing and computations. In this chapter, we consider a three-tiered architectural model consisting of the cloud, fogs and devices (both mobile and static) where machine learning is used to learn from data generated in specific localities and the relationship among the data generated in different localities are studied.

Location-specific neural networks are deployed to learn from data with a particular neighbourhood and predict future data. The efficiency of the neural networks are measured by how well they predict future data, and this metric is used to decide whether or not it is effective to have a neural network in a neighbourhood or not. A supervised learning autoregressive predictive model was developed to forecast future values based on previous values in a time-series. Recurrent Neural Networks empowered by Long Short-Term Memory blocks were utilized to optimize the performance of our model.

We implement the model using Python and a deep learning API with a Tensorflow backend. A host of multi-complexity LSTM RNNs were chosen with dropout regularization used to reduce the probability of overfitting by removing random neurons from the training process. We find that the error rate for the LSTM RNNs chosen at the later stages outperforms that of the best learner in the early training phases. We test the model's performance in making multi-step predictions and find that the prediction accuracy was higher at earlier time steps, but accumulates over time causing lesser prediction accuracy with the predictions still closely following the data trends within a margin of error. Training LSTM RNNs for multi-step time series prediction can be used to combat this problem.

We identify the challenges of incorporating machine learning in data intensive systems such as smart cities. The challenges include application specificness of machine learning models, hyperparameters being domain specific and highly

dependent on data, modifying the architecture of the neural networks to better fit the input data and finally, dealing with missing data. We highlight the benefits of intelligent context sensitive computing as cutting down training times for new vicinities, making data transfer easier and addressing privacy issues.

Future work include the development of context-sensitive computing platforms that incorporate many new computing paradigms and technologies to solve the problems posed by smart-city scale applications. Although many smart-city scale projects are underway, projects that incorporate many new paradigms (e.g., fog computing, machine learning, distributed resource management and self organizing) are yet to be created. It is necessary to create prototypes and study the benefits of these new paradigms in smart city scenarios.

## References

1. Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggle. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.
2. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
3. Sukhadeve Ashish. Understanding neural network: A beginner’s guide, 2017.
4. Colah’s blog. Understanding lstm networks, 2015.
5. Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. Fog computing: A platform for internet of things and analytics. In *Big data and internet of things: A roadmap for smart environments*, pages 169–186. Springer, 2014.
6. Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.
7. Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.
8. Yu Cao, Songqing Chen, Peng Hou, and Donald Brown. Fast: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. In *Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on*, pages 2–11. IEEE, 2015.
9. Josiah L Carlson. *Redis in action*. Manning Publications Co., 2013.
10. Francois Chollet. Keras, 2015.
11. S De Vito, E Massera, M Piga, L Martinotto, and G Di Francia. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757, 2008.
12. John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.
13. Andrea Giordano, Giandomenico Spezzano, and Andrea Vinci. Smart agents and fog computing for smart city applications. In *International Conference on Smart Cities*, pages 137–146. Springer, 2016.
14. Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwalder, and Boris Koldehofe. Mobile fog: A programming model for large-scale applications on the internet of things. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pages 15–20. ACM, 2013.

15. Md M Islam, Xin Yao, and Kazuyuki Murase. A constructive algorithm for training cooperative neural network ensembles. *IEEE Transactions on neural networks*, 14(4):820–834, 2003.
16. Saad Khan, Simon Parkinson, and Yongrui Qin. Fog computing security: a review of current applications and security solutions. *Journal of Cloud Computing*, 6(1):19, 2017.
17. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
18. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
19. Lon-Mu Liu, Gregory B Hudak, George EP Box, Mervin E Muller, and George C Tiao. *Forecasting and time series analysis using the SCA statistical system*, volume 1. Scientific Computing Associates DeKalb, IL, 1992.
20. Ben Marlin. *Missing Data Problems in Machine Learning*. PhD dissertation, University of Toronto, 2008.
21. Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011.
22. Puneet Misra et al. Machine learning and time series: Real world applications. In *Computing, Communication and Automation (ICCCA), 2017 International Conference on*, pages 389–394. IEEE, 2017.
23. MIM Mohamed, W Wu, and M Moniri. Data reduction methods for wireless smart sensors in monitoring water distribution systems. *Procedia Engineering*, 70:1166–1172, 2014.
24. Stefan Nastic, Sanjin Sehic, Michael Vogler, Hong-Linh Truong, and Schahram Dustdar. Patricia—a novel programming model for iot applications on cloud platforms. In *Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on*, pages 53–60. IEEE, 2013.
25. Mugen Peng, Shi Yan, Kecheng Zhang, and Chonggang Wang. Fog-computing-based radio access networks: issues and challenges. *IEEE Network*, 30(4):46–53, 2016.
26. Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Sensing as a service model for smart cities supported by internet of things. *Transactions on Emerging Telecommunications Technologies*, 25(1):81–93, 2014.
27. Per Persson and Ola Angelsmark. Calvin—merging cloud and iot. *Procedia Computer Science*, 52:210–217, 2015.
28. Usman Raza, Alessandro Camera, Amy L Murphy, Themis Palpanas, and Gian Pietro Picco. Practical data prediction for real-world wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(8):2231–2244, 2015.
29. Pedro Pereira Rodrigues and Joao Gama. Online prediction of streaming sensor data. In *Proceedings of the 3rd international workshop on knowledge discovery from data streams (IWKDDS 2006), in conjunction with the 23rd international conference on machine learning*, 2006.
30. Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
31. J Sola and Joaquin Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on nuclear science*, 44(3):1464–1468, 1997.
32. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
33. Ivan Stojmenovic and Sheng Wen. The fog computing paradigm: Scenarios and security issues. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1–8. IEEE, 2014.
34. Kehua Su, Jie Li, and Hongbo Fu. Smart city and the applications. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 1028–1031. IEEE, 2011.

35. George Suciu, Alexandru Vulpe, Simona Halunga, Octavian Fratu, Gyorgy Todoran, and Victor Suciu. Smart cities built on resilient cloud computing and secure internet of things. In *Control Systems and Computer Science (CSCS), 2013 19th International Conference on*, pages 513–518. IEEE, 2013.
36. Tarik Taleb, Sunny Dutta, Adlen Ksentini, Muddesar Iqbal, and Hannu Flinck. Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine*, 55(3):38–43, 2017.
37. Bo Tang, Zhen Chen, Gerald Hefferman, Tao Wei, Haibo He, and Qing Yang. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In *Proceedings of the ASE BigData & SocialInformatics 2015*, page 28. ACM, 2015.
38. Bo Tang, Zhen Chen, Gerald Hefferman, Tao Wei, Haibo He, and Qing Yang. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In *Proceedings of the ASE BigData & SocialInformatics 2015*, page 28. ACM, 2015.
39. Anthony M Townsend. *Smart cities: Big data, civic hackers, and the quest for a new utopia*. WW Norton & Company, 2013.
40. Robert Wenger, Xiru Zhu, Jayanth Krishnamurthy, and Muthucumar Maheswaran. A programming language and system for heterogeneous cloud of things. In *Collaboration and Internet Computing (CIC), 2016 IEEE 2nd International Conference on*, pages 169–177. IEEE, 2016.
41. Feng Xia, Laurence T Yang, Lizhe Wang, and Alexey Vinel. Internet of things. *International Journal of Communication Systems*, 25(9):1101, 2012.
42. Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. Fog computing: Platform and applications. In *Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on*, pages 73–78. IEEE, 2015.
43. Z Zhou. Ensemble learning. encyclopedia of biometrics (pp. 270–273), 2009.
44. Sotiris Zygariis. Smart city reference model: Assisting planners to conceptualize the building of smart city innovation ecosystems. *Journal of the Knowledge Economy*, 4(2):217–231, 2013.



# Intelligent Mobile Messaging for Smart Cities Based on Reinforcement Learning



Behrooz Shahriari and Melody Moh

**Abstract** Mobile messaging has become a trend in our daily lives, and is vital in supporting new services in smart cities. The current schema for messaging is to route all the messages between mobile users through a centralized server. This scheme, though reliable, creates very heavy load on the server. It is possible for users to communicate through peer-to-peer (P2P) connection, especially over urban networks characterized by heavy user traffic and dense network connectivity. P2P connections however do not provide the best user experience, as they are sometimes unreliable due to network coverage fluctuation. We propose an intelligent messaging framework based on reinforcement learning to strike a balance between reducing server load and improving user experience. The system learns and adapts in real-time to user mobility and messaging patterns. The adaptive system dynamically chooses between routing through the server and routing via P2P connection. As it does not rely on user location information, user privacy is thus preserved. Performance evaluation through simulation of user movement and messaging patterns demonstrates that the system is able to find the best messaging policy for users, achieves a well balance between heavy server load and unreliable communication, and provides a fine user messaging experience while reduces server load. We believe that this work is significant for future smart cities and urban networking where mobile messaging will be prominent among mobile users as well as mobile smart objects.

**Keywords** Mobile messaging · Reinforcement learning · SARSA · Adaptive tree · Online learning · Peer-to-peer (P2P) · User experience

---

B. Shahriari · M. Moh (✉)  
Department of Computer Science, San Jose State University, San Jose, CA, USA  
e-mail: [melody.moh@sjsu.edu](mailto:melody.moh@sjsu.edu)

© Springer Nature Switzerland AG 2018  
M. Maheswaran, E. Badidi (eds.), *Handbook of Smart Cities*,  
[https://doi.org/10.1007/978-3-319-97271-8\\_9](https://doi.org/10.1007/978-3-319-97271-8_9)

## 1 Introduction

As the number of mobile devices has surpassed the number of desktops in the world [1], mobile messaging has become the dominant way of communications [2]. Mobile users are using mobile messaging for Mobile Instance Messaging (MIM) [3] and text/image/audio/video file sharing [4], both for personal and business purposes [5]. Furthermore, mobile messaging has also enabled communications between mobile users and mobile smart objects

According to a research done by the United Nation's panel on global sustainability, by 2050, 70% of the world population will live in urban areas, which only cover 2% of the entire Earth surface, yet are responsible for 75% of the greenhouse gas emissions [36]. Based on this understanding, the concept of Smart Cities needs solution and practices to advance the development of urban environments, while also making them more sustainable. In particular, the use of Information and Communication Technologies (ICT) will provide the necessary backbone, not only for maintaining existing services but for enabling new ones. Mobile messaging, both among mobile users and between mobile users and mobile devices, is one prominent example of utilizing ICT for supporting smart cities.

The contemporary framework utilized by most messaging apps for message delivery is through a centralized messaging server, to which users must first subscribe. In centralized message communication, if connections fail hereby the mobile users can resume the connections with server shortly after failure. This centralized messaging framework however has created a heavy processing load on the server.

Alternatively, messaging data can be sent directly from one peer to another peer, or peer-to-peer (P2P), without any need for an intermediate server [6], especially when utilizing urban networking where user population and network connectivity are both high. P2P communication unfortunately is neither reliable nor robust. A group of mobile users on a P2P connection may lose their connections during the communication, and they may not reestablish the message session again due to their mobility or due to fluctuation in network coverage. Thus, utilizing user mobility profile prediction, which has been addressed in [7–10], is a way toward robust P2P connections. These schemes however usually require user location information. Yet, many users choose to hide their location to protect their privacy. The idea of generic online learning system has been explored to provide a more reliable and robust communication between mobile devices in 5G network [34].

Addressing the challenge of providing private, reliable roaming experience for mobile users without over-stressing the server, we develop an intelligent messaging system to find an equilibrium between maximizing user-experience and minimizing server load [11, 12]. Note, however, our solution to the mobile messaging problem can be extended to IoT communications, such as those supporting localized automated machinery and sensors. These include communications for smart-homes, smart-factories, self-driving cars, and drones where all the communications need to happen in real-time with minimum latency taking account the message load and

mobility of devices or bots [39–42]. For these IoT scenarios our solution provides more reliable communication the same way as it does for messaging apps.

The proposed intelligent system is based on *reinforcement learning (RL)* [13], which learns from interaction with the environment via recognition and action to achieve a goal. At each interaction step, the agent (system) chooses an action based on the state of the environment that then alters the state of the environment. A reward or punishment is then provided to the agent as a measure of the desirability of the chosen action. In other words, the agent chooses an action based on a policy, and the policy is learned through trial-and-error interactions of the agent with its environment. RL algorithms are very useful for solving a wide variety of problems especially when the model is not known in advance. A specific RL method, *SARSA (State-Action Reward State-Action)* [13], is adopted for its good convergence property. This work is a continuation of our on-going research in intelligent network systems [32–34, 37, 38, 42].

This chapter is extended from an earlier work [37]. The rest of this chapter is organized as follows. Section 2 describes the background. Section 3 presents the proposed intelligent mobile messaging architecture and learning algorithm. Section 4 illustrates the simulation model. The experimental results are presented in Section 5, followed by conclusions in Section 6.

## 2 Background: Message Types, Messaging Frameworks, and Reinforcement Learning

In this section we present message types, messaging frameworks, as well as a brief overview of reinforcement learning (RL).

### 2.1 Message Types

In general, a message has various types such as text, image, and video. Each type varies in size; thus, we define various Message Types (MT) based on their size as follow,

1. Small message (SM): Text messages with the size between 3 KB and 500 KB, which represents small text or large text message.
2. Medium message (MM): Images, short video clips, and voice messages with the size between 500 Kilo-Bytes (KB) and 10 Mega-Bytes (MB).
3. Large message (LM): Video clips, video messages, or any media files including documents with the size between 10 MB and 70 MB.

## 2.2 *Messaging Framework*

In mobile messaging, the messaging user (i.e., the sender) first sends a Message Request (MR) to the server. The MR contains sender identity, receiver identity, and message type. Then server decides on the type of the connection which is one of the followings,

### 2.2.1 **Route Framework**

It is used by top mobile messaging apps, such as Facebook messenger, WeChat, Kik, Telegram, and Line, to transmit messages between users. In this framework, a user first sends the MR to the server. Afterwards, the user starts sending the message to the server which in turn distributes the message to the designated receivers. Note that with Route Framework the connection is mostly reliable, but it adds heavy load to the server.

### 2.2.2 **P2P Framework**

Messaging apps such as Bleep use P2P connection as an alternative to Route Framework. In P2P Framework, users communicate directly with one another, while the server can act as a STUN (Session Traversal of User Datagram Protocol through Network Address Translators) server [14, 15]. After a user sends a MR, the STUN server responds with the network addresses of other users, so that the user can establish P2P connections for transmitting the message. In using P2P Framework, the server load is largely reduced, but the connections may be unreliable.

### 2.2.3 **Intelligent (Hybrid) Framework**

The first two frameworks each have their advantage and weakness. Also, P2P connection works best only when the IP version of two users is the same and Network Address Translation (NAT) of their mobile service providers [16, 17] are of compatible types. Thus, to reduce server load and to improve communication robustness, we propose the Intelligent Framework, which takes a hybrid approach. With the help of SARSA, a RL technique, in Sect. 2.3 we review RL and its learning method. With this approach, the server can make real-time decision on connection type (Route or P2P) of each MR. For simplicity in this chapter we assume that there is no group messaging, i.e., each user communicates to one single user at a time.

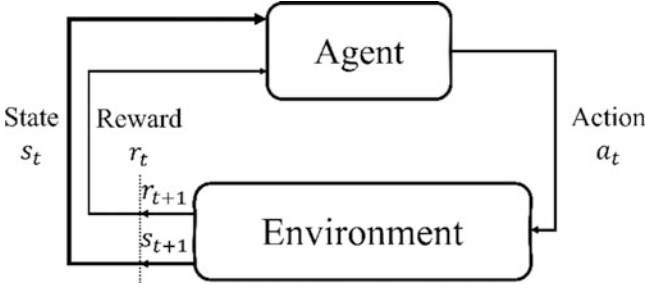


Fig. 1 Interaction of IS (agent) with environment with RL

### 2.3 Reinforcement Learning

Standard RL theories [13, 35] are based on the concept of Markov decision process (MDP). A MDP is denoted as a tuple  $\langle S, A, R, P \rangle$ , where  $S$  is the state space,  $A$  is the action space,  $R$  is the reward function (feedback), and  $P$  is the state transition probability, as shown in Fig. 1.

The goal of RL is to learn the optimal policy  $\pi^*$ , so that the expected sum of discounted reward of each state will be maximized

$$J_{\pi^*} = \max_{\pi} J_{\pi} = \max_{\pi} E_{\pi} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right] \quad (1)$$

where  $\gamma \in [0, 1)$  is the discount factor which, if set to zero, makes the agent “opportunistic” about current reward while agent with  $\gamma$  approaching to 1 strives for a long-term high reward.  $r_t$  is the reward at time-step  $t$ .  $E_{\pi}[\cdot]$  stands for the expectation with respect to the policy  $\pi$  and the state transition probabilities, and  $J_{\pi}$  is the expected total reward. A value function  $Q(s, a)$  represents the estimate of expected return attainable from executing action  $a$  in state  $s$ . Its computation repeatedly sweeps through the state-action space of MDP. The value function of each state-action pair is updated according to:

$$Q(s, a) \leftarrow \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma \max_{a'} Q(s', a')] \quad (2)$$

until the largest change  $\Delta$  in the value of any state-action pair is smaller than a preset constant threshold, where  $p(s'|s, a)$  is the probability of state transition from  $s$  to  $s'$  after executing action  $a$ , and  $r(s, a, s')$  is the corresponding reward. After the algorithm converges, the optimal policy is followed by simply taking the greedy action in each state  $s$  as

$$a^* = \operatorname{argmax}_a Q^*(s, a), \quad (\forall s \in S) \quad (3)$$

As for model-free cases where the agent has no prior knowledge of the environment,  $Q$ -learning (an RL algorithm) can achieve optimal policies from delayed rewards. At a certain time step  $t$ , the agent observes the state  $s_t$ , and then chooses an action  $a_t$ . After executing action  $a_t$ , the agent receives a reward  $r_{t+1}$  and gets into the next state  $s_{t+1}$ . Then the agent will choose the next action  $a_{t+1}$  according to the best-known knowledge and learned policy.

Let  $\alpha_t$  be the learning rate where  $\alpha_t$  equal to zero makes the agent incapable of learning anything while  $\alpha_t$  equal to one makes it consider only the most recent information. The one-step updating rule of  $Q$ -learning can be described below:

$$Q(s_t, a_t) = (1 - \alpha_t) Q(s_t, a_t) + \alpha_t (r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a')) \quad (4)$$

While  $Q$ -learning algorithm chooses the best action based on the state-action pair with highest  $Q$  value; the learning algorithm used in this chapter, SARSA (State-Action Reward State-Action) [13], chooses actions by  $\varepsilon$ -greedy policy, and the updating algorithm is described as follows,

$$Q_{t+1}(s_t, a_t) = (1 - \alpha_t) Q_t(s_t, a_t) + \alpha_t (r_{t+1} + \gamma Q'_{t+1}(s_t, a_t)) \quad (5)$$

where  $Q'$  is the  $Q$  value selected with  $\varepsilon$ -greedy policy at  $t + 1$  for  $(s_t, a_t)$ . In  $\varepsilon$ -greedy policy the action is chosen based on highest  $Q$  value with probability of  $1-\varepsilon$ , otherwise it is chosen randomly, where  $\varepsilon < 0.1$ . The  $\varepsilon$ -greedy policy allows the agent to explore the state-action space more and avoids fast convergence to local optimal solution.

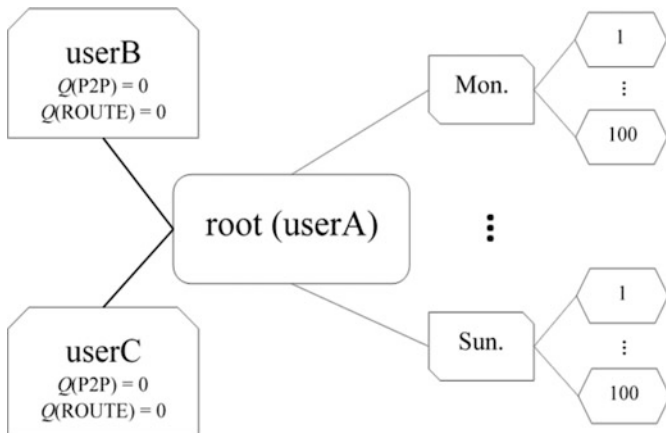
### 3 Intelligent Messaging Architecture and Learning Algorithm

In this section the proposed architecture based on SARSA [13] is described. It includes states, actions, reward values, learning policy, and adaption algorithm.

#### 3.1 States: *STree*

The visible parameters to the server are the identity of the users that want to exchange a message, the request time, and the message size. Based on these parameters there are four possible states for each MR, depending on the level of complexity.

- $SI$  ( $userA$ ,  $userB$ ): Based only on user identity, where  $userA$  wants to send data to  $userB$ .



**Fig. 2** Example of STree with no MN

- $S_2$  ( $userA, userB, DW$ ): Same as  $S_1$  but  $S_2$  relies also on day of the week ( $DW$ ), where  $userA$  wants to send data to  $userB$  on a specific day of week. This is useful when movement pattern of users depends on the day.
- $S_3$  ( $userA, userB, DW, MT$ ): Same as  $S_2$  but  $S_3$  relies also on the type of message ( $MT$ ), which as described in Section II can be  $SM, MM,$  or  $LM$ .
- $S_4$  ( $userA, userB, DW, MT, TD$ ): Same as  $S_3$  but  $S_4$  relies also on time of the day ( $TD$ ).

Note that in above,  $S_1$  is the simplest state and  $S_4$  is the most comprehensive one. A complex state such as  $S_4$  increases the complexity of the learning system and makes a large search space for finding the matching state for each MR. It is therefore important to use the appropriate state (of suitable complexity). Consider the following example: When two users always send messages to each other from their specific locations, then the server needs only  $S_1$  state for their connection; defining and using any other states would not be necessary.

Referring to Fig. 2, to define the state for each user (say  $userA$ ), we use a tree called State-Tree (STree) [18–22], where leaves of the tree are the states for a given user, e.g.  $userB$ . The nodes of STree are defined as following:

- *Root*: represents the user e.g.  $userA$ .
- *Day Node (DN)*: represents a day of week. Each root has seven DNs for each day of week.
- *Message Node (MN)*: represents the type of message, which can be  $SM, MM,$  or  $LM$ . Each DN is connected to three MNs.
- *Time-Node (TN)*: represents a period of time in a day, and each MN is connected to a hundred TNs [23].
- *QNode*: represents the leaf of STree, which is the state that contains the  $Q$  values for possible actions. *QNode* also keeps the average and standard-deviation of

changes in  $Q$  values of each action to calculate the effectiveness of  $Q$ Node for adaptation of STree.  $Q$ Node can be a child of any other node in the STree.

- *QNode Probability Links (QNP)*: each  $Q$ Node forms a connection to other  $Q$ Nodes in the same tree in order of their activation. Each connection has a weight that represents activation count (AC), every time a  $Q$ Node activates the AC of connection that it has with previous state increases by one. This way with AC, we can calculate the probability of states sequences in a STree. Based on the AC of all connections from a state to its connected, possible next states, we can calculate the probability of each state as next possible state to the current state. The system can use this probability to make a decision based on possible future states and the current states. The use of this feature has been explained in the learning algorithm described in Sect. 3.4.2, and been evaluated in the experiment reported in Sec. 5.3.

Note that it is possible that to remove MN from STree as an optimized system may not be dependent on message type. (Note also that the simulation experiments, described in Sect. 4, evaluate the system both with and without MN.)

As an example of STree, Fig. 1, we have userA with two users in its contact list (userB and userC). At the very beginning of learning, userA is the root of STree, which has two leaves ( $Q$ Node), one for each user in its contact list, and these  $Q$ Nodes describe the states with  $Q$  value for each action. At this point the type of each leaf is S1. The root of this STree also has seven DN child nodes, with each DN node having three MN children. Each MN node has a hundred TN child nodes, each representing a specific period of time during the day; note that none of these nodes has  $Q$ Node leaves yet (the initial state).

### 3.2 Actions (Choosing Connection Type)

Actions are defined based on the type of connections, thus we have two possible actions for each state,

1. *P2P*: a P2P connection will be established between the two users.
2. *ROUTE*: a route connection will be established through the server.

### 3.3 Reward

Based on the chosen action by the server, the server collects data about the connection and calculate rewards accordingly. The Connection Data (CD) contains the following information:

- Identity of sender, e.g. userA.
- Identity of receiver, e.g. userB.



- Day of the week that connection happens.
- Time of start and end of the connection in seconds.
- Size of the message in KB.
- A List of Data for each transmission epoch (LDE) of a message. During each transmission epoch the server collects information such as time in seconds, successful transmitted data and lost data in the epoch.

### 3.3.1 Server Reward

Based on each element in an LDE, the server calculates server reward and user reward. Server reward reflects the server load; the larger the server load the smaller the server reward. Server-reward is calculated as follows,

$$a_S^i = data_{transmitted}^i + data_{lost}^i \quad (6)$$

$$a_S = \frac{\sum_i \text{in LDE} (a_S^i)}{2 \times TM} \quad (7)$$

$$R_S = \begin{cases} 1 & P2P \\ 1 - a_S & ROUTE \end{cases} \quad (8)$$

Where  $a_S$  is the ratio of transmitted data for the connection, which it defined to be the sum of successful transmitted data and lost data (thus needs to be retransmitted) for each epoch divided by twice the Total Message size (TM), since in Route framework the entire message is transmitted once from userA to server then from the server to userB. Also,  $R_S$  is the server-reward where for P2P connection is one as there is no load on the server. Note that server reward may be negative if there is significant data loss.

### 3.3.2 User Reward

User reward reflects user experience based on the amount of successfully transmitted and lost data. More lost data implies a smaller user reward. To calculate user-reward, the server iterates over each element  $i$  in the LDE.

$$mtl_i = \max \left( data_{transmitted}^i, data_{loss}^i \right) \quad (9)$$

$$a_u^i = \begin{cases} \frac{data_{transmitted}^i - data_{loss}^i}{mtl_i} & mtl_i > 0 \\ 0 & O.W. \end{cases} \quad (10)$$

$$rate_i = \begin{cases} a_u^i & I \\ a_u^i - |a_u^i| \times 0.15 & II \end{cases} \quad (11)$$

Where in (10) the reward is expressed as the difference between successful transmitted data and lost data over the maximum of the two values. To calculate data transmission rate,  $rate_i$ , in (11), the following conditions are considered:

- In condition *I* where, for both userA and userB, the best cellular antenna near the user supports the user's cellular provider, then the transmission ratio is unchanged. Note that it is 1 when there is no data loss and  $-1$  if all data is lost.
- In condition *II*, unlike condition *I*, the provider of towers (closest to users) and users do not match, then there is a penalty in transmission rate, as a punishment with a constant factor.

Finally, user-reward  $R_u$  of a particular connection is defined as follows:

$$\mu_{rate} = \sum_{i \text{ in LDE}}^N rate_i / N \quad (12)$$

$$R_u = \begin{cases} -1 & I \\ 0.5 \times \mu_{rate} & II \\ \mu_{rate} & III \end{cases} \quad (13)$$

Where in (12)  $N$  is the length of LDE and in (13) we consider the following conditions,

- Condition *I* if initialization of connection between the two users fail, then user-reward is equal to  $-1$ .
- Condition *II* if the connection drops during the transmission due to bad coverage then we have a user-reward as the average rate with a penalty factor of  $0.5$ .
- Condition *III* when all the data has been transmitted successfully between users, therefore the average rate is the user-reward.

### 3.4 Learning Algorithm

The decision made by the server is the result of RL method. SARSA and  $Q$ -Learning (QL) and are two prominent learning methods, differ by the policy method for updating the  $Q$  values of each state and action [13]. In this research, we select SARSA to avoid converging to the possible erroneous state-action. In SARSA, for  $Q'$  in (5 or 14), the action selection policy for the next state-actions is the same as that of the current state.

### 3.4.1 Learning without QNP

However, in our system, if we do not use QNP then the definition of next state is ambiguous as we cannot determine the next state based on the current state-action. Thus, we modify the original adaptation of  $Q$  in SARSA for current state and action as follows,

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha [r_{t+1} + \gamma Q'_{t+1}(s_t, a_t) - Q_t(s_t, a_t)]. \quad (14)$$

$$r_{t+1} = \omega R_s^{t+1} + (1 - \omega) R_u^{t+1} \quad (15)$$

Where,

- $Q'$  is selected based on  $\varepsilon$ -greedy policy [13], which is our action selection policy for the current state and current action to maximize the exploration and exploitation. With  $\varepsilon$ -greedy policy one can guarantee that the system finds the best action policy for each state in the long-term. Note however that, as  $Q$  updating policy is changed there is no guarantee that the system converges to the best policy.
- Equation (15) is the reward that system receives from the environment, which is the weighted average of user-reward (13) and server-reward (8). When  $\omega$  is set to 0, only server-reward is considered, in other words one only intends to minimize the server-load; and for  $\omega$  equals to 1 we consider only user-reward which minimizes data loss.

### 3.4.2 Learning with QNP

In the case where we use QNP we can determine the next possible state based on the probability between QNodes. As the system calculates the probability based on the current value of ACs of state connections then, over time, the probabilities between states change depending on the user's behavior. This approach turns the learning method close to actual SARSA algorithm [25–27]. So (14) will be revised as,

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha [r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)] \quad (16)$$

where  $Q(s_{t+1}, a_{t+1})$  is the estimation of next state and its selected action with  $\varepsilon$ -greedy policy for  $s_{t+1}$ . Now, with connections between the QNodes and the probability of each connection, we can calculate the most probable next state to the current state that has the highest probability and calculate  $Q(s_{t+1}, a_{t+1})$  based on that.

*Input:* The current CD (Connection Data).

1. Find the  $Q$ Node in STree that represents the state of CD based on time, message type, and identity of users.
2. Update the  $Q$  value of state-action for  $Q$ Node of CD with (14).
3. IF (17) is satisfied, an indication of fluctuation in  $Q$  value, then
  - i) IF  $Q$ Node's parent is the root THEN move  $Q$ Node to DNs which are connected to the root.
  - ii) IF  $Q$ Node's parent is a DN THEN move the  $Q$ Node to MNs which are connected to the DN.
  - iii) IF  $Q$ Node's parent is a MN THEN move the  $Q$ Node to TNs which are connected to the MN\*.

*Output:* The adjusted STree with updated  $Q$  values for current state.

\* A STree may not have any MN, thus in step ii  $Q$ Node will be connected to TNs and step iii is ignored.

**Fig. 3** STree adaptation algorithm

### 3.5 Adaptation of STree

STree of each user is initialized with  $Q$ Nodes connected to the root for which the state is defined based on the path from the root to each  $Q$ Node e.g. (userA, userB). However, as (userA, userB) may not be a suitable description of state to make decision for userA-userB connection type, the system requires higher granularity to make a decision.

Figure 3 shows the adaptation algorithm of STree, where the system validates reliability of a state for decision making with (17).

$$2 \times \sigma_{\Delta Q} > |\mu_{\Delta Q}| \quad (17)$$

Where  $\mu_{\Delta Q}$  is the average and  $\sigma_{\Delta Q}$  is the standard-deviation (SD) of changes in  $Q$  values of each  $Q$ Node which are calculated when the  $Q$  values are updated according to (14). If for any action in any  $Q$ Node (17) is satisfied, then the action causes  $Q$  values to fluctuate quite rapidly. The  $Q$ Node is then pushed down one level in the STree. For example, if a  $Q$ Node is directly connected to the root (userA) then we remove this  $Q$ Node and add it to all the DN children of the root.

Figure 4 demonstrates the adapted STree of Fig. 2 after 20 days of learning based on STree adaptation method described in Fig. 3. For userA and userC we have that ROUTE connection leads to a better result or reward. Also, for userB and userA

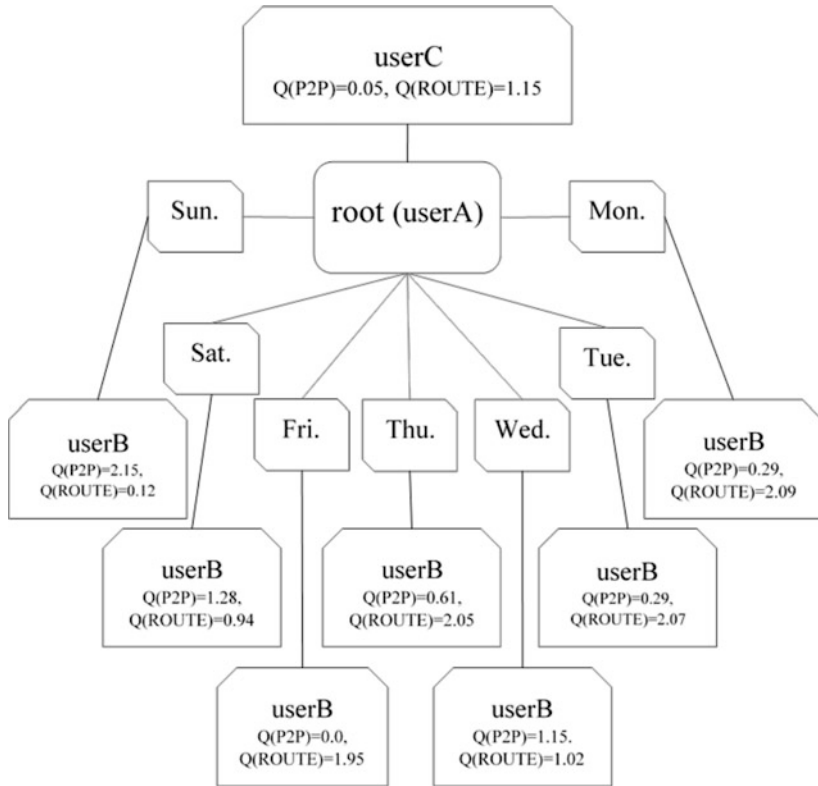


Fig. 4 Adapted STree of Fig. 2, after 20 days of learning

during weekend the P2P connection is a better choice and we hypothesis that they communicate with one another while they are at home as they are stationary with good network connection. Note that during weekdays the users are not stationary and their movements cause the server to favor ROUTE connection over P2P for most weekdays.

#### 4 Simulation Model

We use a simulation engine to mimic the user messaging and user movement pattern to evaluate our intelligent framework. The simulation is based on the assumption that users have similar patterns for each day of week [7, 8].

## 4.1 Simulation Engine Components

1. *Simulation area*, is a large area to simulate an urban city in which users live and work.
2. *Tower*, represents a real cellular mobile antenna, each supports some cellular network provider, e.g. AT&T, Verizon, Sprint, T-Mobile, etc. Also, it has the maximum coverage radius for the coverage area for which the coverage equation is a normal distribution with arbitrary parameters.
3. *User*, represents an actual mobile user which has a phone book that contains some users in the simulation platform. Each user is registered with a provider such as Sprint, Verizon, etc. Also, each user is simulated by its daily profile and communication profile that contains the following components,
  - (a) *Daily profile*, represents the movement pattern for each day of the week of a user. User follows a specific movement pattern on daily basis. Note that in our simulation user's daily movement for each day of week is mixed with substantial amount of noise so prediction of user movement pattern through statistical analysis becomes impractical. Also, engine chooses random movement patterns for some days instead of the normal pattern of the user to make the simulation realistic.
  - (b) *Contact profile*, represents the contact phone book of a user and the messaging pattern with each user in the phone book. Each user generates a MR during the steps of simulation [24] based on messaging pattern that contains the normal distributions for message type and a random number for contacting another user. The MR is generated with the assumption that all communications involve only two users.
4. *Server*, represents the actual messaging server inside the simulation engine. The server is responsible to handle each MR. It based on its intelligent framework to decide whether the connection should be routed through the server or should be a P2P connection. When the data file is transmitted or the connection has failed between userA and userB, the server collects the CD to update the  $Q$  values. Also, upon a P2P connection failure, we simplify the simulation and do not allow users to re-connect and they must make a new MR later.

## 4.2 Steps of Simulation

### 4.2.1 Initialization

The simulation engine places the towers in various places within the simulation area. The placement is made by choosing a random location in the area for the first tower, then surround the tower by 2–6 random towers. The engine places two towers near each other if and only if their coverage area has less than 15% overlap, where coverage area is modeled as a random generalized bell function [31]. Then, the

engine initializes each user and its daily movement profile and messaging profile as well as its home location. Home is defined as the start point and end point of each user during the day, after which it spends more than 6 h at home.

#### 4.2.2 Simulation at k-th Step

- If a user is not in an open communication then it randomly chooses a user from its contact list based on a random number, as well as message size to generate a MR.
- Each user updates its location and velocity based on its daily profile.
- The server receives all the MRs from all the users, it then handles each MR based on the intelligent framework described in Fig. 3.

The simulation engine finds the nearest tower with the best signal to process the data transmission of each MR based on the connection type (chosen by the server), the users' current locations and their velocities. The rate of transmission is calculated based on connection type and signal strength.

## 5 Experiments

The intelligent mobile messaging is evaluated based on the simulation of 500 users and their messaging for 150 days, for an urban area of 10 km by 10 km. For (14) we set  $\varepsilon=0.001$ ,  $\alpha=0.15$  and  $\gamma=0.85$  to ensure the adaptation of system in the long-term and to avoid opportunistic learning based on current reward.

We evaluate the intelligent system with three experiments. First two experiments include the STree with no QNP, and last experiment evaluate the system with QNP.

In experiment 1, we compare the STree against three other tests, and then in our second experiment, we evaluate the performance of our system for various  $\omega$  in (15). Table 1 describes the test 1 and 2 scenarios.

**Table 1** Description of all tests in experiments 1 and 2

Test	Description
ROUTE	Where all users use ROUTE connections.
P2P	Where all users use P2P connections.
RANDOM	Where users choose P2P or ROUTE randomly.
S	Where the server use STree with no MN, with $\omega = 0.5$ .
SM	Where the server use STree with MN.
SW	Where the server use STree with no MN and $\omega$ is a key parameter.

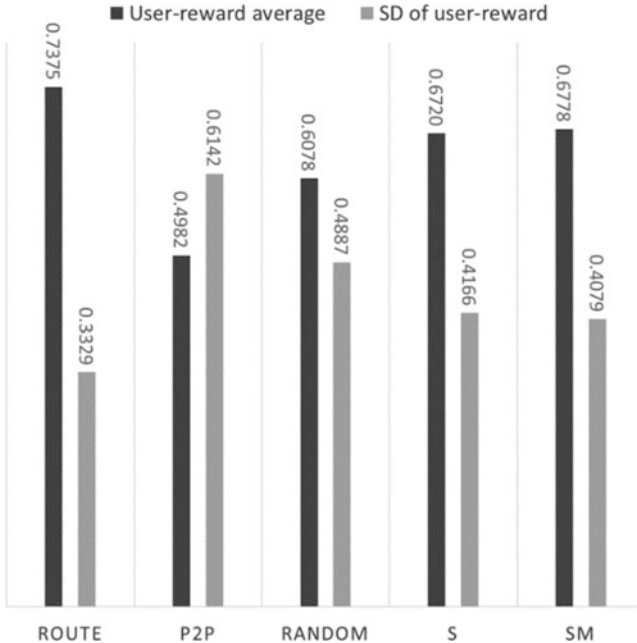


Fig. 5 Comparison of UA and its standard-deviation for experiment 1

## 5.1 Experiment 1

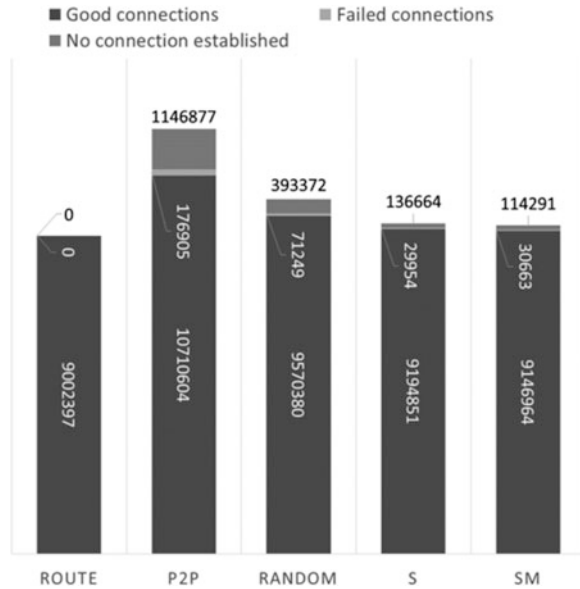
For the first experiment we set  $\omega=0.5$ . Also, according to Table 1 we evaluate the system with two type of STrees, one with MN and another without MN.

In Fig. 5, we have the comparison of **user-reward average (UA)** and SD based on (9, 10, 11, 12, and 13). Highest UA occurs when all of the connections are routed through the server, with evidently no connection drops and with low UA variance as expected. Note that when fluctuation in UA is high, then the system cannot provide good and reliable connections since the overall data loss is high. Figure 5 also shows the performance for P2P connections, where the UA is observed to be lower and variance tends to be high. Note that the RANDOM test shows a better result than P2P with lower fluctuations in UA. Ultimately, when we use STree, we observe UA value becomes similar to ROUTE with small fluctuation, which indicates that *STree with learning has an overall better performance than both P2P and RANDOM in terms of user reward.*

In Fig. 6, we have the comparison of *total number of MR connections* during the entire simulation. There are three possible final categories for each connection, (1) *good connections*: the connections remains active until the entire data is transferred, (2) *no connection established*: the connection cannot establish which will happen in P2P connections where devices of users are using incompatible IP version and cellular network, and (3) *failed connections*: the connection establishes but fails as



**Fig. 6** Comparison of all connections for experiment 1



one of the users enters an area with poor network coverage. Fig. 6 depicts that with P2P connections more MRs are generated, which is caused by lower transmission time (as shown in Fig. 8), but the total number of failed and non-established connections is higher. The total number of ROUTE connections is the smallest since it increases the transmission time, yet there is no failed or non-established connection. We see that, *in terms of failed and non-established connections, STree with learning has an overall better performance than both P2P and RANDOM.*

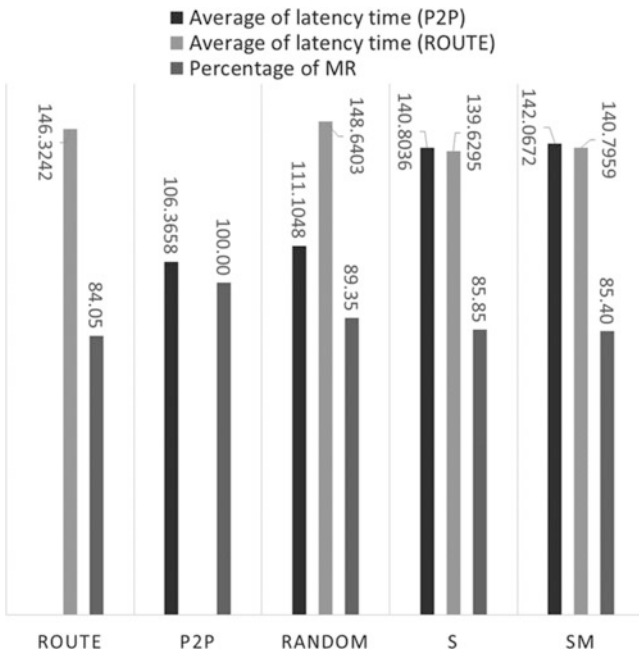
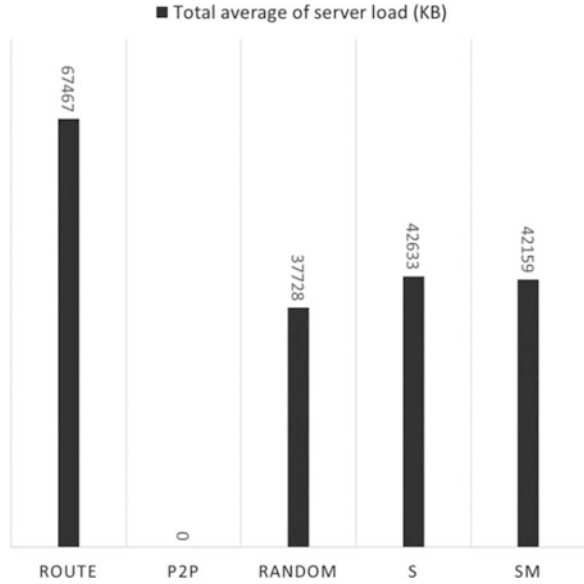
In Fig. 7, we have the comparison of server load average, where server has highest load for ROUTE test, and is 0 for P2P. Note that STree was able to reduce the load drastically by a factor of 60% over ROUTE.

Finally, in Fig. 8 we have the comparison of latency time for transmitting the entire data per connection. (Note that percentage of MR means the total number of MRs, using that P2P as 100%.) For P2P connections, we have the lowest latency time as expected, since all the data is transmitted directly between users. Note that the STree system has achieved almost equal latency for ROUTE connections and P2P connections, as it is able to make a balanced decision between the two.

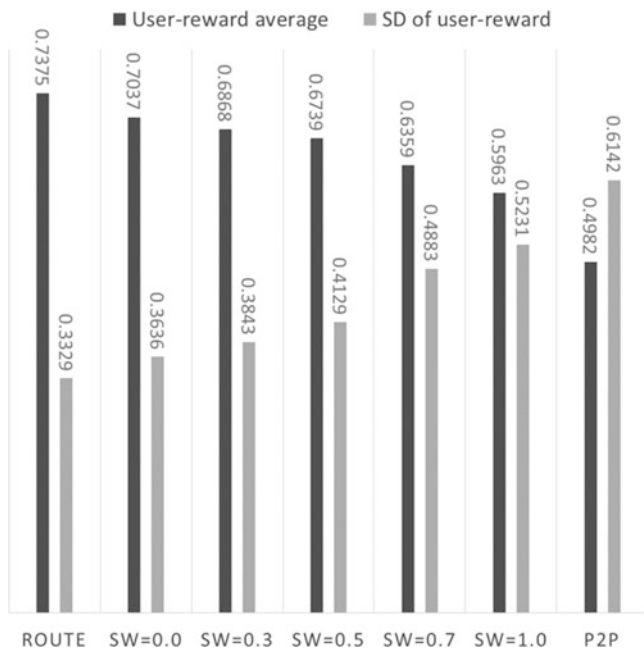
## 5.2 Experiment 2

For the second experiment, we repeat the same experiment while evaluating the effect of  $\omega$  on STree with no MN, for five different value of  $\omega$ , in  $\{0.0, 0.3, 0.5, 0.7, 1.0\}$ . Note that when  $\omega$  is 0.0, we are favoring user-reward according to (15) so we expect that result to be close to ROUTE, and when  $\omega$  is 1.0 we are favoring

**Fig. 7** Comparison of server load in experiment 1



**Fig. 8** Comparison of latency average for each case in experiment 1 per connection type against normalized rate for number of successful connections



**Fig. 9** Comparison of UA and its standard-deviation for experiment 2

server-reward so we expect to have a similar result as P2P mode where server has no load.

In Fig. 9 (corresponding to Fig. 5), we have UA (User reward average) of STree with various values for  $\omega$  against P2P and ROUTE case. Note that as  $\omega$  increases toward 1.0 the UA descends as we are favoring server-load according to (15), thus as it is expected most of the connections are P2P.

Figure 10 (corresponding to Fig. 6) shows the comparison for the number of (good, failed, and no) MR connections. Note that when  $\omega$  is 1.0 the pattern for the state of connection is similar to P2P test and when  $\omega$  is 0.0 it resembles ROUTE test. The values are not the same due to  $\epsilon$ -greedy policy, which adds randomness to the action selection method of the system.

In Figs. 11 and 12 (corresponding to Figs. 10 and 11) we have the comparison of server load and latency of each test. Based on these results we can confidently assert that STree can find the optimal solution for balancing the server-load and user-reward or user satisfaction in term of transmitting a file with less data loss.

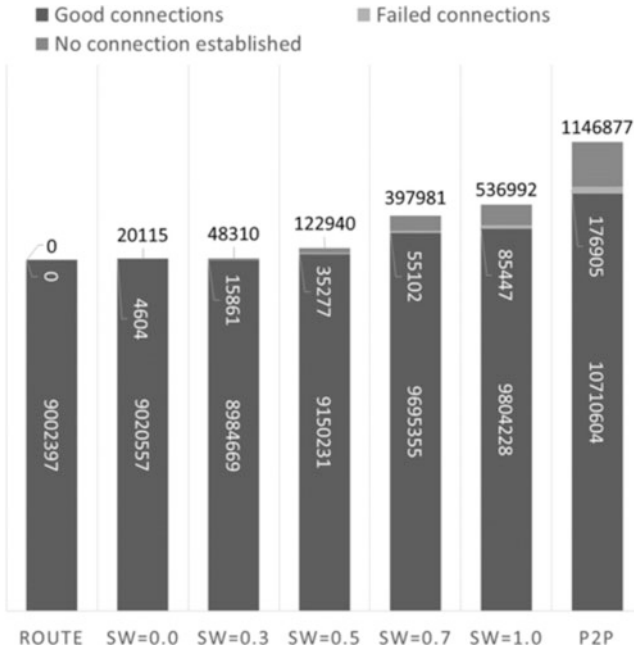


Fig. 10 Comparison of all connections for experiment 2

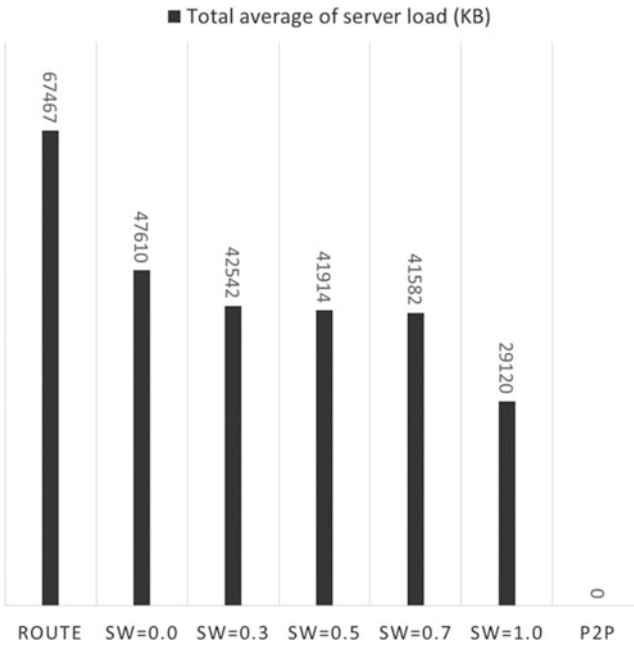


Fig. 11 Comparison of server load for experiment 2

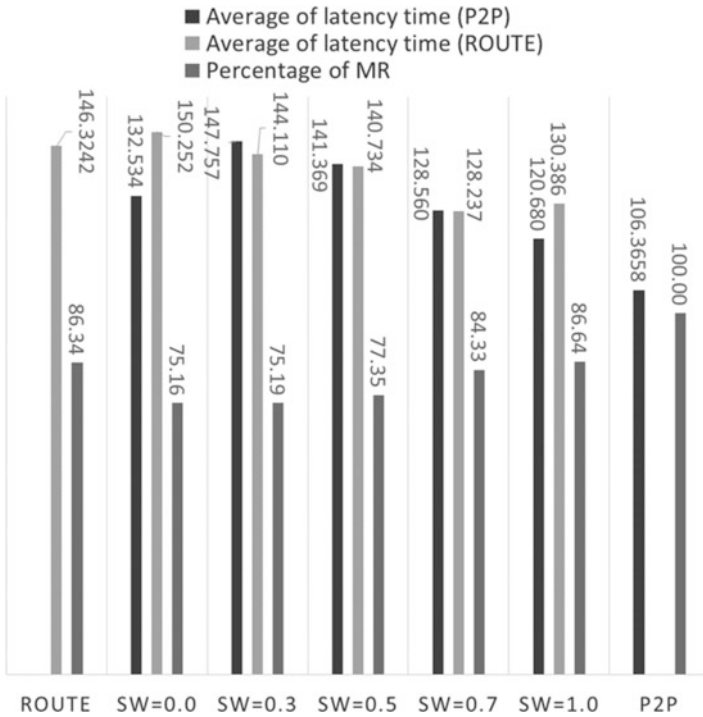


Fig. 12 Comparison of latency average for each case in experiment 2 per connection type against normalized rate for number of successful connections

### 5.3 Experiment 3

In this experiment, we evaluate the effect of QNP on performance and decision results. Recall, as explained in Sect. 3.1 and utilized in Sect. 3.4.2, QNP represents the probability that a particular state would be the next state of the current state. Using this probability would potentially improve the decision made on choosing the next state. For this purpose, we compare the STree of QNP against one with no QNP; in both case the STree has no MN, and  $\omega$  equals to 0.5. The simulation is done with 50 users and for a maximum simulation time of 300 days.

Figure 13 shows the progress of user-reward average during the simulation per 50 days. It is noticeable that utilization of QNP in STree has steadily increased the performance as time goes, while the one with no QNP does not show any improvement with longer duration of the simulation. The one with QNP has improved up to 1.2%. This is because the system can predict the future state of users more accurately.

In Fig. 14 we have the comparison for status of connections during the entire simulation. According to the chart, the number of no-connections (no connection was established between users) reduced by 5.78%, and the number of good

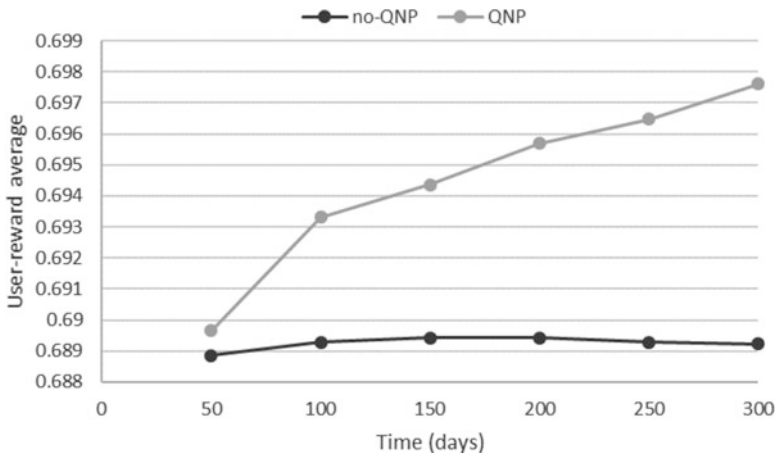


Fig. 13 The average of user-reward vs. time for both cases

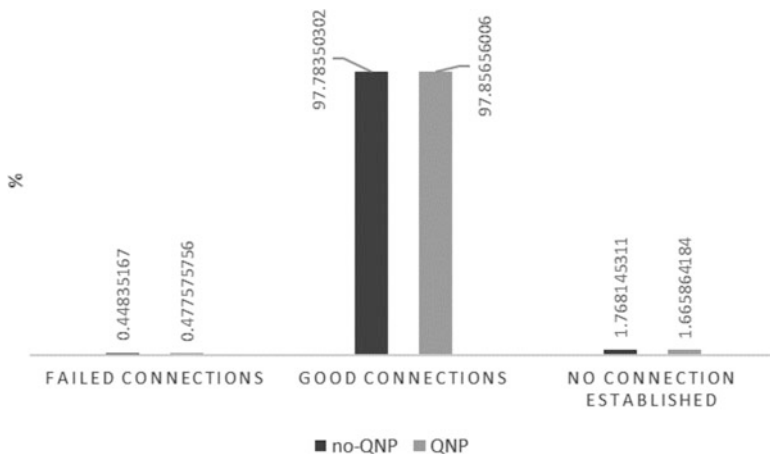
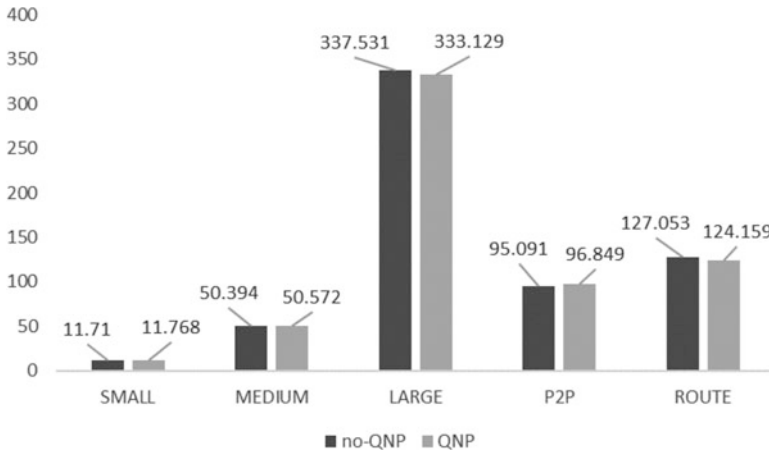


Fig. 14 Comparison of status of connections for both cases

connections (established connections between users) increased by 0.07%, Thus, we have seen some improvement for successfully established connections and good connections for QNP, which was predictable according to Fig. 13, as there is a correlation between user-reward and increase in good connections.

In Fig. 15, we have the average latency per connection type and message type. The results have been extended and shown in Table 2, the summary of average latency for various messages and connection types.

From Table 2, we can observe that system of QNP has leaned toward choosing ROUTE connection, particularly for Medium and Large sized messages, as we have 6.68% and 6.05% improvement, respectively. Also, it is noticeable that for P2P connection Medium message has the highest penalty in latency time in QNP.



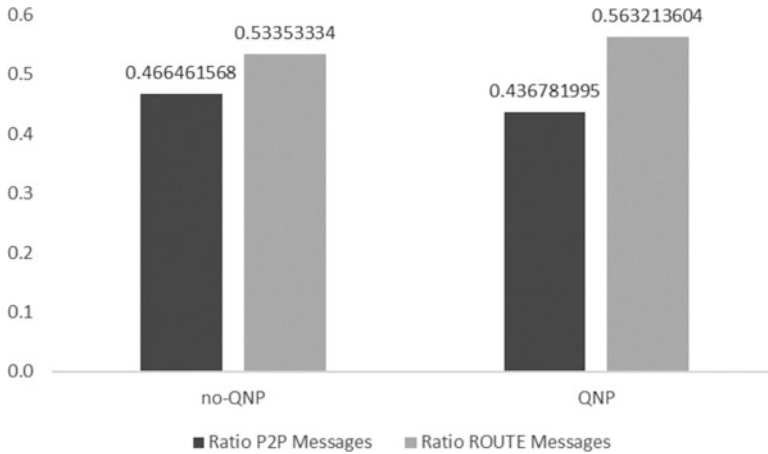
**Fig. 15** Average of latency per message type and connection type

**Table 2** Average of latency per message type and connection type

	no-QNP latency	QNP latency	Improvement percentage
Small message	11.71	11.77	-0.49
Medium message	50.39	50.57	-0.35
Large message	337.53	333.13	1.32
<b>P2P connection</b>	<b>95.09</b>	<b>96.85</b>	<b>-1.82</b>
P2P-small message	11.22	11.36	-1.19
P2P-medium message	41.09	44.98	-8.63
P2P-large message	275.65	281.95	-2.23
<b>ROUTE connection</b>	<b>127.05</b>	<b>124.16</b>	<b>2.33</b>
ROUTE-small message	12.13	12.11	0.14
ROUTE-medium message	58.22	54.57	6.68
ROUTE-large message	396.59	373.97	6.05

Based on the results of simulations the system has processed 8,641,698 messages for normal STree with no-QNP and 8,635,736 messages with QNP STree. That is, using QNP allows the system to process about 1% less messages. Figure 16 shows the ratio of messages per connection type. It is clear, and consistent with Table 2, that STree with QNP prefers to ROUTE.

In general, one can summarize that STree with QNP prefers to ROUTE (Figs. 15 and 16, and Table 2). Its performance, in general, is slightly better, with higher user reward (Fig. 13), fewer no-connections and more good connections (Fig. 14), and smaller latency (Table 2).



**Fig. 16** The ratio of messages in each simulation per connection type

## 6 Conclusion and Future Work

We have proposed an intelligent mobile messaging framework based on SARSA reinforcement learning method. Through the real-time learning method that adapts to user behavior of messaging and mobility patterns, the system has successfully reduced server load while maintaining high user-satisfaction. The learning method does not need user location information, thus preserves user privacy. Based on experimental results, the proposed system has achieved a fine balance between P2P or Route (routing through a central server) frameworks, and better than using P2P or Route framework alone, as it finds the dynamic solution for increasing user-satisfaction while reduces server load. Also, the system has a degree of freedom based on a  $\omega$  parameter that allows configuring the system to lean toward specific scenarios, such as preferring to user satisfaction or to server-load reduction. Future work would include considering additional metrics in the reward system, such as transmission time and energy efficiency [28–30], also combining the improved intelligent design used in 5G load balancing [34] for better and robust communication for P2P communication between mobile devices with this framework, and evaluate the performance of the current simulation for P2P communication.

## References

1. Mary Meeker (2015). *Internet Trends Report of 2015* [Online], Available: <http://www.kpcb.com/internet-trends>
2. R. d. A. Oliveira; W. C. Brandão; H. T. Marques-Neto, “Characterizing User Behavior on a Mobile SMS-Based Chat Service”, XXXIII Brazilian Symposium on Computer Networks and Distributed Systems (SBRC), pp 130 – 139, 2015.



3. P.-L. To, C. Liao, J. C. Chiang, M.-L. Shih and C.-Y. Chang, "An empirical investigation of the factors affecting the adoption of Instant Messaging in organizations," *Original Research Article Computer Standards & Interfaces*, vol. 30, no. 3, p. 148–156, 2008.
4. O. O. Abiona1; A. I. Oluwaranti; T. Anjali; C. E. Onime; E. O. Popoola; G. A., "Architectural model for Wireless Peer-to-Peer (WP2P) file sharing for ubiquitous mobile devices", *IEEE International Conference on Electro/Information Technology*, pp. 35-39, 2009.
5. Noor Musmayati Musa; Fauziah Redzuan, "Understanding user behavior towards mobile messaging application use in support for banking system", *3rd International Conference on User Science and Engineering (i-USER)*, pp. 269-274, 2014.
6. J. Maenpaa; V. Andersson; G. Camarillo; A. Keranen, "Impact of Network Address Translator Traversal on Delays in Peer-to-Peer Session Initiation Protocol", pp 1-6, 2010.
7. I. F. Akyildiz; Wenye Wang, "The predictive user mobility profile framework for wireless multimedia networks", *IEEE/ACM Transactions on Networking*, pp 1021 – 1035, 2004.
8. D. Barth; S. Bellahsene; L. Kloul, "Mobility Prediction Using Mobile User Profiles", *IEEE 19th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp 286 – 294, 2011.
9. S. Khokhar; A. A. Nilsson, "Estimation of Mobile Trajectory in a Wireless Network: A Basis for User's Mobility Profiling for Mobile Trajectory Based Services", *Third International Conference on Sensor Technologies and Applications*, pp. 69-74, 2009.
10. M. A. Bayir; M. Demirbas; N. Eagle, "Discovering spatiotemporal mobility profiles of cellphone users", *IEEE Int. Sym. on a World of Wireless, Mobile and Multimedia Networks*, pp. 1-9, 2009.
11. G. Gupta; R. Garg, "Minimizing the cost of mobility management: distance-based scheme as a function of user's profile", *Wireless Communications and Networking*, pp. 2075 – 2080, vol. 3, 2003.
12. T. Deng; X. Wang; P. Fan; K. Li, "Modeling and Performance Analysis of Tracking Area List-Based Location Management Scheme in LTE Networks", *IEEE Transactions on Vehicular Technology*, 2015.
13. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
14. J. Rosenberg, J. Weinberger-Dynamicsoft, C. Huitema-Microsoft, R. Mahy-Cisco, "STUN-Simple Traversal of User Datagram Protocol Through Network Address Translators," RFC-3489, 2003.
15. Ha Tran Thi Thu; Jaehyung Park; Yonggwon Won; Jinsul Kim, "Combining STUN Protocol and UDP Hole Punching Technique for Peer-To-Peer Communication across Network Address Translation", pp, 1 – 4, 2014.
16. Junnosuke Kuroda; Yasuichi Nakayama, "STUN-based connection sequence through symmetric NATs for TCP connection", *Network Operations and Management Symposium (APNOMS)*, pp. 1-4, 2011.
17. Yong Wang; Zhao Lu; Junzhong Gu, "Research on Symmetric NAT Traversal in P2P applications", *International Multi-Conference on Computing in the Global Information Technology*, 2006.
18. K. S. Hwang; Y. J. Chen; C. J. Wu, "Fusion of Multiple Behaviors Using Layered Reinforcement Learning", *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, pp. 999 – 1004, vol. 42, 2012.
19. X. Xu; C. Liu; S. X. Yang; D. Hu, "Hierarchical Approximate Policy Iteration with Binary-Tree State Space Decomposition", *IEEE Transactions on Neural Networks*, pp. 1863 – 1877, vol. 22, 2011.
20. K. S. Hwang; T. W. Yang; C. J. Lin, "Self Organizing Decision Tree Based on Reinforcement Learning and its Application on State Space Partition", *IEEE International Conference on Systems, Man and Cybernetics*, pp. 5088 - 5093, vol. 6, 2006.
21. Min Wu; A. Yamashita; H. Asama, "Rule abstraction and transfer in reinforcement learning by decision tree", *IEEE/SICE International Symposium on System Integration (SII)*, pp. 529 – 534, 2012.

22. K. S. Hwang; Y. J. Chen, "Tree-like Function Approximator in Reinforcement Learning", 33rd Annual Conference of the IEEE Industrial Electronics Society, pp. 904 – 907, 2007.
23. P. Boone; M. Barbeau; E. Kranakis, "Using time-of-day and location-based mobility profiles to improve scanning during handovers", IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, pp. 1-6, 2010.
24. A. Sokolovsky; S. I. Bross, "Attainable error exponents for the Poisson broadcast channel with degraded message sets", IEEE Transactions on Information Theory, vol. 51, pp. 364-374, 2005.
25. Le Tien Dung, T. Komeda ; M. Takagi, "Mixed Reinforcement Learning for Partially Observable Markov Decision Process", International Symposium on Computational Intelligence in Robotics and Automation, pp. 7-12, 2007.
26. L. Li; A. Scaglione, "Learning hidden Markov sparse models", Information Theory and Applications Workshop (ITA), pp. 1-13, 2013.
27. O. H. Hamid; F. H. Alaiwy; I. O. Hussien, "Uncovering cognitive influences on individualized learning using a hidden Markov models framework", Global Summit on Computer & Information Technology (GSCIT), pp. 1-6, 2015.
28. Yanwen Wang, Hainan Chen, Xiaoling Wu, Lei Shu, "An energy-efficient SDN based sleep scheduling algorithm for WSNs", Journal of Network and Computer Applications, pp. 39-45, 2016.
29. Dan Wu; Jinlong Wang; Rose Qingyang Hu; Yueming Cai; Liang Zhou, "Energy-Efficient Resource Sharing for Mobile Device-to-Device Multimedia Communications", IEEE Transactions on Vehicular Technology, vol. 63, no. 5, pp. 2093-2103, 2014.
30. Mohammad Ashraf Hoque; Matti Siekkinen; Jukka K. Nurminen, "Energy Efficient Multimedia Streaming to Mobile Devices — A Survey", IEEE Communications Surveys & Tutorials, vol. 16, 2014.
31. Jin Zhao; B. K. Bose, "Evaluation of membership functions for fuzzy logic controlled induction motor drive", IEEE 28th Annual Conference, vol. 1, pp. 229-234, 2002.
32. M. Moh, B. Chellappan, T.-S. Moh, and S. Venugopal, "Handoff mechanisms for IEEE 802.16 networks supporting intelligent transportation systems," in *Wireless Technologies for Intelligent Transportation Systems*, edited by Ming-Tuo Zhou, Yang Zhang, and Lawrence Yang, published by Nova Science Pub., 2010.
33. R. Wong, T.-S. Moh, and M. Moh, "Semi-Supervised Learning BitTorrent Traffic Detection," in *Distributed Network Intelligence, Security and Applications*, ed. by Qurban A. Memon, CRC Press - Taylor & Francis Group, USA, Apr 2013.
34. Behrooz Shahriari; Melody Moh; Teng-Sheng Moh, "Generic Online Learning for Partial Visible Dynamic Environment with Delayed Feedback: Online Learning for 5G C-RAN Load-Balancer", International Conference on High Performance Computing & Simulation (HPCS), pp. 176-185, 2017.
35. Chia-Feng Juang, and Chia-Hung Hsu, "Reinforcement Ant Optimized Fuzzy Controller for Mobile-Robot Wall-Following Control", IEEE Transactions on Industrial Electronics, Vol. 56, NO. 10. Oct. 2009.
36. United Nations Secretary-General's high-level panel on global sustainability, "Resilient People, Resilient Planet: A future worth choosing," 2012.
37. B. Shahriari and M. Moh, "Intelligent Mobile Messaging for Urban Networks – Adaptive Intelligent Messaging Based on Reinforcement Learning," Proceedings of 12th IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, October 17-19, 2016.
38. C. Tsai and M. Moh, "Load Balancing in 5G Cloud Radio Access Networks Supporting IoT Communications for Smart Communities," Proceedings of 2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Bilbao, Spain, Dec 2017.
39. Badis Hammi, Rida Khatoun, Sherali Zeadally, "IoT technologies for smart cities", IEEE IET Networks, Vol. 7, 2017.

40. Walid Balid, Hazem H Refai, "On the development of self-powered iot sensor for real-time traffic monitoring in smart cities", IEEE SENSORS, 2017.
41. Jay Lohokare, Reshul Dani, Ajit Rajurkar, Ameya Apte, "An IoT ecosystem for the implementation of scalable wireless home automation systems at smart city level", IEEE Region 10 Conference, TENCON, 2017.
42. Su, Gary, Melody Moh. "Improving Energy Efficiency and Scalability for IoT Communications in 5G Networks." Proc. of 12th ACM Int. Conf. on Ubiquitous Information Management and Communication (IMCOM), Langkawi, Malaysia, January 2018.

# Asymmetric Interoperability for Software Services in Smart City Environments



José C. Delgado

**Abstract** Interoperability of software services is one of the main challenges of smart city environments, since there is a huge number of interconnected small devices (Internet of Things) which implement and provide a wide variety of fine-grained software services. Classical approaches, such as Service Oriented Architecture (SOA) and RESTful APIs, in which both interacting services share the same data schema, usually lead to a coupling problem, since a service cannot change the schema of its messages without changing it as well in the services with which it interacts. This chapter proposes an asymmetric interoperability approach, in which the schema used to produce a message does not have to be identical to the schema of the messages expected by the receiver. This asymmetry in interoperability is based on the concepts of structural compliance and conformance, which state that schemas need only be compatible in the message components that are actually used and not in the full message schema. This reduces service coupling and allows a service to interact with others, which send or receive messages with different schemas, and to replace another one with a new schema without impairing existing interactions. Simple models of interoperability, coupling, adaptability, and changeability are proposed to justify the usefulness of the compliance and conformance concepts. A few implementation examples, using JSON, are also presented.

**Keywords** Smart cities · Internet of Things · Interoperability · Coupling · Changeability · Compliance · Conformance · Service · Schema

## 1 Introduction

A *smart city* [1] can be defined as a high-density conglomerate of people and infrastructures that uses *software services*, based on information and communication technologies (ICTs), to improve the efficiency and competitiveness of urban

---

J. C. Delgado (✉)  
Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal  
e-mail: [jose.delgado@tecnico.ulisboa.pt](mailto:jose.delgado@tecnico.ulisboa.pt)

operations and services, the sustainability of social and environmental aspects, and generally the quality of life of its inhabitants.

Smart cities constitute a very relevant issue for the development of society. A study from the World Health Organization [2] projects “*the percentage of the world’s population living in urban areas to increase from 54% in 2015 to 60% in 2030 and to 66% by 2050*”. With two thirds of human beings living in cities, endowing them with access to software services that make their city smarter and increase their quality of life, while benefiting the economy and the environment, is of paramount importance.

Given its infrastructure density, a smart city is heavily based on a myriad of small devices, such as sensors, actuators and displays, most of which embedded in systems such as office buildings, smart homes, vehicles and a wide variety of electric appliances. Many of these devices now have Internet connectivity and constitute what is known as the Internet of Things (IoT) [3, 4].

The IoT has been experiencing an explosive growth, approximately 30% each year. Gartner [5] estimated that, by the end of 2017, around 8.4 billion IoT devices were in use, outnumbering the world’s population for the first time, with a forecast of 20.4 billion by 2020. Other predictions have hit the 50 billion mark, a figure by now recognised as overestimated. Nevertheless, several analysts still predict higher numbers than those of Gartner [6]. Independently of the numbers, the fact is that there will be a huge number of interconnected devices, from a large number of manufacturers with a wide variety of models, and all needing to interact.

Interoperability is thus one of the main challenges of IoT environments [7, 8]. The obvious solution is to define standard APIs that all devices should implement, thereby making interaction between devices an achievable goal [9]. In practice, however, *de jure* standards require time for technology to settle down, something hard to occur in such a young and vigorous field as IoT, and *de facto* standards can only be imposed by a market leader that stands out of the crowd. Again, this is not easy to achieve, given the enormous variability of manufacturers, devices, services and applications.

Without widely accepted standards, interoperability is possible if interacting devices agree on data and/or service schemas, typically based on data description languages such as Extensible Markup Language (XML) [10] and JavaScript Object Notation (JSON) [11], and on service models such as Service-Oriented Architecture (SOA) [12] and Representational State Transfer (REST) [13].

These technologies were not conceived for small devices with a weak computing power, such as those typically found in IoT, but their main disadvantage is that they are symmetric, in the sense that both the sender and receiver of a message must use the same schema. This entails more coupling than actually needed, because the interacting devices need to support all the data values valid for the schema, even if they use only a fraction of these values.

To solve this problem, we propose to use asymmetric interoperability, based on the concepts of compliance and conformance:

- The schema of the sender must comply with that of the receiver. The schema of the sender needs to include all the mandatory features of the schema of the receiver, but may or may not include the optional features and may include any additional feature, which will be ignored. Compliance allows a sender to meaningfully send a message to many receivers, not just to one that implements the same schema as the sender.
- The schema of the receiver needs to conform to the schema that the sender requires. The receiver needs to implement at least all the features that the sender uses, but can also implement others that the sender does not know about. Conformance allows a receiver to meaningfully receive messages from many senders, not just from those that implements the same schema.

Both are ways to reduce coupling between the interacting devices and to increase the interoperability range. Taking a basic API (standard, or not) as a starting point, variations to that API are allowed at both the sender and the receiver, as long as compliance and conformance hold. There is no longer the need to stick to a fixed API.

This chapter is structured as follows. Section 2 describes some of the existing work relevant to the context of this chapter. Section 3 analyses the main issues involved in the interoperability problem, providing models of interoperability, coupling, adaptability, and changeability. Section 4 discusses symmetric interoperability in existing technologies. Section 5 proposes asymmetric interoperability and provides a model of structural interoperability, based on compliance and conformance, with some examples using JSON. Finally, Sect. 6 draws some conclusions regarding the proposals made.

## 2 Background

Smart cities [14] try to improve many factors, including urban planning [15], social dimension [16], economic impact [17], energy management [18], sustainability [19], and technology [20, 21], which is the most relevant in the context of this chapter, in particular in what software services are concerned.

The IoT [3, 4] became the backbone of smart cities [1], supporting the data generated by all the devices that enable cities to be smarter [22]. Gartner [5] has asserted that the number of Internet-enabled devices is clearly growing much faster than the number of Internet human users. This means that humans no longer dominate the Internet. That role is now fulfilled by smart devices that are small computers and require technologies suitable to them and to the services they support, instead of those conventionally used for Internet browsing.

The sheer number and diversity of IoT devices entail an enormous problem in interconnecting the services running on those devices. The Internet is global, distributed and huge, while still requiring that any device, subject to specific interoperability requirements, be able to interact with any other Internet-connected entity, including humans [23].

Distributed service interoperability is not specific of the IoT contexts. It has been studied in several domains relevant to smart cities, such as enterprise cooperation [24], smart government [25], cloud computing [26], and healthcare applications [27]. Most of these domains involve applications running on full-fledged servers, not on the much simpler IoT devices, such as those involved in sensor networks [28] and vehicular networks [29].

The two main technological approaches for distributed interoperability, Web Services [30] and RESTful APIs [31], are based on data description languages such as XML [10] and JSON [11].

SOA [12] is the architectural style underlying Web Services and models real-world entities by the behaviour (services) they can offer. REST [13] is the architectural style underlying RESTful APIs and models real-world entities by the structural state (resources) that they can exhibit.

A continuing debate has been going on over the years about which architectural style, SOA or REST, is more adequate for specific classes of applications, namely IoT services [32]. There are several studies comparing these styles [33, 34], usually more on technological grounds than on conceptual or modelling arguments. There are also proposals to integrate SOA and RESTful services [35].

The main problem with these architectural styles is the significant level of coupling they entail. Interacting services need to share the same data description schema, which means that a change in one service will most likely imply a change in the other.

Several metrics have been proposed to assess the maintainability of distributed service systems, based essentially on structural features, namely for service coupling, cohesion and complexity [36]. Other approaches focus on dynamic coupling, rather than static, with metrics for assessing coupling during program execution [37]. There are also approaches trying to combine structural coupling with other levels of coupling, such as semantics [38].

Compliance [39] and conformance [40] are concepts that will be used in this chapter as foundational mechanisms to ensure partial interoperability and thus minimize coupling. These mechanisms have also been studied in specific contexts, such as choreography [41], modelling [42], programming [43], and standards [44].

Searching for a compatible service can be done by schema matching with similarity algorithms [45] and ontology matching and mapping [46]. However, this does not ensure interoperability, usually requiring manual adaptations.

### 3 Analysing the Service Interoperability Problem

#### 3.1 The Main Aspects of Interoperability

Distributed interoperability implies that interacting services can use some network to send each other messages, which both sender and receiver need to understand and take action accordingly. Smart cities exacerbate the interoperability problems by the sheer number and variety of interacting services provided by sensors, actuators, and controllers. These fine-grained services are by far more common than classic coarse-grained, enterprise-class services and thus are the most interesting ones, from the point of view of this chapter.

A service running on a device, in the role of *provider*, publishes the set of messages that it is able to respond to (its Application Programming Interface – API), which defines the interface of the functionality offered by that service. In the role of *consumer*, another service (running on some other device) may send one of the acceptable messages to the service provider and invoke the corresponding functionality. Services can also express their APIs in terms of exposed features, such as operations. Each operation can accept its own set of messages.

A typical service interaction is initiated by the consumer, which sends a request message to the provider, through the interconnecting network, which may cause the provider to answer with a response message, upon executing the request. This is illustrated by Fig. 1, in which a service implemented by a controller inquires a service running on a sensor, possibly to obtain some of its accumulated data.

The provider needs to be able to understand what the consumer is requesting, and to react and respond according to what the consumer expects. Otherwise, the service interaction will be meaningless.

In a distributed environment, interoperability cannot rely on shared data type names and inheritance hierarchies, since services evolve independently from each other. Therefore, messages cannot simply be assumed to be correct and meaningful

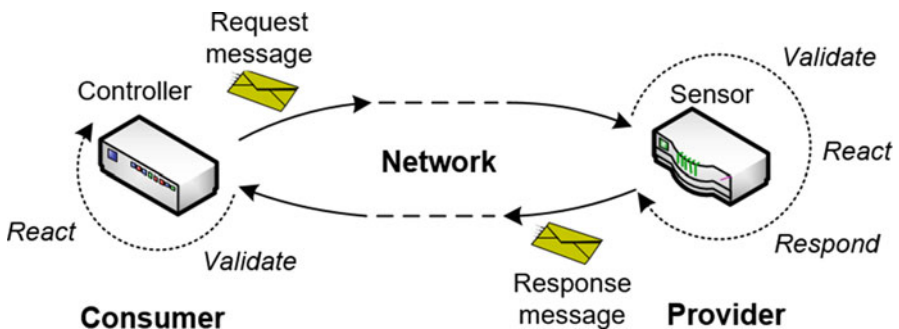


Fig. 1 Message-based interaction between two services, implemented by a controller and a sensor



in both the contexts of the interacting services. The goal of achieving a simple interaction such as the one depicted in Fig. 1 can be decomposed into the following objectives:

1. There must be an addressable interconnecting network and a message-based protocol, which allows a request message to be sent specifically to a given provider and a response message to be sent back to the original sender.
2. The request message needs to be validated by the provider, which means that it must be one of those acceptable by the provider's API. This validation must be done both at the syntactic and semantic levels.
3. The reaction of the provider and the corresponding effects, as a consequence of executing the request message, must fulfil the expectations of the consumer regarding that reaction. In other words, the provider needs to do what the consumer expects. This is pragmatics [47], a higher level than semantics.
4. The eventual response message needs to be validated by consumer, which means that it must be one of those acceptable by the consumer as a response. The validation must be done both at the syntactic and semantic levels.
5. The consumer must react to the response appropriately, fulfilling the purpose of the provider in sending that response and completing the purpose of the consumer in initiating the interaction. Again, this is pragmatics.

This means that it is not enough for a service to send a request to another one and hope that everything goes well. Both request and response need to be validated and understood (correctly reacted upon) by the service that receives it.

Service interaction is a complex issue with many factors, such as:

- *Interoperability* – Guaranteeing that a service understands the requests of another and reacts according to what is expected.
- *Coupling* – Mutual dependencies between services, with the goal of reducing them as much as possible, to avoid unnecessary constraints to the evolution and variability of services.
- *Adaptability* – Maintaining interoperability, even when interacting services change some of their characteristics.
- *Architectural style* – Choosing the way in which devices are modelled as services has a relevant impact on how devices interact.
- *Reliability* – Maintaining interoperability, even in the presence of unanticipated failures.
- *Security* – Ensuring interoperability is allowed only intentionally and with authorized and certified services.
- *Performance* – Ensuring that interactions complete faster than agreed maximum response times.
- *Scalability* – Ensuring that performance levels do not decrease substantially when the number of interacting services increases.

To limit its breadth and scope, this chapter tackles only the first four, which are detailed in the following sections.

## 3.2 A Model of Interoperability

The meaning of interoperability can vary according to the perspective, context, and domain under consideration. Although limited to information, the 24765 standard [48] provides the probably most cited definition of interoperability, as “*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*”.

In the context of software services for smart cities, and in light of Fig. 1, this definition can be interpreted as “the ability of two or more software services to exchange messages and to react to them according to some pattern or contract that fulfils the constraints and expectations of all services involved”.

Interoperability involves several abstraction layers, from low-level networking issues to high-level aspects reflecting the context of the interaction. Layering is an abstraction mechanism useful to deal with complexity. One early example is the Open Systems Interconnection (OSI) reference model [49], with seven layers, although it concentrates on the networking issues. This chapter proposes a different layering mechanism, detailing higher-level issues, as described by Table 1.

The *pragmatic* level caters for the fact that an interaction between a consumer and a provider is done in the context of a contract (even if implicit), which is implemented by a choreography that coordinates processes, which in turn implement workflow behaviour by orchestrating service invocations.

The *semantic* level expresses that interacting services must be able to understand the meaning of the content of the messages exchanged, both requests and responses. This implies compatibility in rules, knowledge, and ontologies, so that meaning is not lost when transferring a message from the context of the sender to that of the receiver.

The *syntactic* level deals mainly with form, rather than content. Each message has a structure, composed by data (primitive objects) according to some structural definition (its schema). The data in messages need to be serialised to be sent over the network, using formats such as XML or JSON.

The main objective in the *connective* level is to transfer a message from the context of one services to that of another, regardless of its content. This usually involves enclosing that content in another message with control information and implementing a message protocol over a communications network protocol, possibly involving routing gateways.

At the top of this level hierarchy, a further level can be defined to express the *symbiotic* nature of the interaction between services as a mutually beneficial agreement. This expresses the purpose of that interaction, or why it has been designed that way. However, this is more relevant at the level of interacting organizations, not at the level of devices and their fine-grained services, and will not be detailed here.

The interoperability model of Table 1 implies that all layers are involved in every interaction. Even the simplest interaction is part of a choreography, involves meaning, has a structure and needs a network to send the messages. In practice,

**Table 1** Layers of interoperability

Level	Layer	Main concern	Description
Pragmatic (reaction and effects)	Contract	Choreography	Management of the effects of the interaction at the levels of choreography, process and service
	Workflow	Process	
Semantic (meaning of content)	Interface	Service	Interpretation of a message in context, at the levels of rule, known application components and relations, and definition of concepts
	Inference	Rule base	
Syntactic (notation of representation)	Knowledge	Knowledge base	Representation of application components, in terms of composition, primitive components and their serialisation format in messages
	Ontology	Concept	
Syntactic (notation of representation)	Structure	Schema	Lower level formats and network protocols involved in transferring a message from the context of the sender to that of the receiver
	Predefined type	Primitive object	
Connective (transfer protocol)	Serialisation	Message format	
	Messaging	Message protocol	
	Routing	Gateway	
	Communication	Network protocol	
	Physics	Media protocol	

however, most of these layers are dealt with *tacitly* (based on unverified assumptions that are supported by documentation at best) or *empirically* (based on verified assumptions that are hidden by already existing specifications or tools).

The most relevant layers for message-based interactions are typically the Interface (service) and Structure (schema), although all others are present as well. The Ontology layer (concept) has gained relevance in the last few years [50], given the huge diversity of existing services and the need to resort to semantics to clarify the meaning of the schemas and of the services' interface.

### 3.3 A Model of Coupling

Some degree of previously agreed mutual knowledge is indispensable to enable services to interact and cooperate towards common or complementary objectives. However, this entails interdependencies (*coupling*), hampering services from evolving independently, when they need. Breach of an interaction contract, due to a change in one service that is incompatible with one or more of its interlocutors, is a major cause of failures in applications. This is particularly important in contexts such as smart cities, in which applications are highly distributed and involve a huge number of interacting services.

Decoupling favours adaptability, changeability and even reliability (if one fails, there is less impact on the other), but interoperability requires a minimum level of coupling so that interoperability, as described in Fig. 1, can occur. The fundamental problem of service interaction is then to achieve the maximum decoupling possible (exposing the minimum possible number of features) while ensuring the minimum interoperability requirements. Reducing coupling increases:

- The likelihood of finding suitable alternatives or replacements for a given service, faulty or discontinued.
- The set of services with which some service is compatible.

Figure 2 enlarges the scenario of Fig. 1. Now several applications, running on servers (possibly in a cloud) or user computers send messages to the controller to invoke its service functionality. In turn, the controller can inquire a different number of sensors, to obtain their data. The controller's service now acts as both a provider (regarding applications) and a consumer (regarding sensors).

In this case, coupling with respect to the controller expresses not only how much it depends on the sensors (its providers) but also how much the applications (its consumers) depend on it.

Dependency on a service can be assessed by the fraction of its features (e.g., operations) that impose constraints on other services. Two coupling metrics can be defined, from the point of view of a given service (Fig. 2):

- $C_F$  (*forward coupling*), which expresses how much a service is dependent on its providers, defined as:

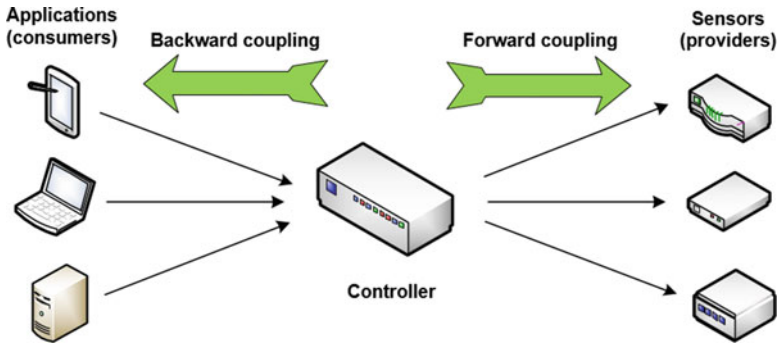


Fig. 2 Backward and forward coupling

$$C_F = \frac{\sum_{i \in P} \frac{U_{p_i}}{T_{p_i} \cdot N_i}}{|P|} \quad (1)$$

where:

- $P$  is the set of providers that this service uses.
- $|P|$  denotes the cardinality of  $P$ .
- $U_{p_i}$  is the number of features that this service uses in provider  $i$ .
- $T_{p_i}$  is the total number of features that provider  $i$  exposes.
- $N_i$  is the number of providers with which this service is compatible as a consumer, in all uses of features of provider  $i$  by this service.
- $C_B$  (*backward coupling*), which expresses how much impact a service has on its consumers, defined as:

$$C_B = \frac{\sum_{i \in C} \frac{U_{c_i}}{T_c \cdot M}}{|C|} \quad (2)$$

where:

- $C$  is the set of consumers that use this service as provider.
- $|C|$  denotes the cardinality of  $C$ .
- $U_{c_i}$  is the number of features of this service that consumer  $i$  uses.
- $T_c$  is the total number of features that this service exposes.
- $M$  is the number of known services that are compatible with this application and can replace it, as a provider.

Coupling varies from 0 (no dependencies at all) to 1 (dependency on all components).

Metric 1 expresses that the existence of alternative providers reduces the consumer's forward coupling  $C_F$ , since more services (with which the consumer is compatible) dilute the dependency.

Similarly, metric 2 concludes that the existence of alternatives to an existing provider reduces the consumers' dependency on it, since now they can choose another provider. This reduces the impact that a service may have on its potential consumers and therefore its backward coupling  $C_B$ .

In either case, increasing the number of compatible alternatives implies reducing the number of features required for compatibility. Fewer constraints will most likely mean that more services are compatible. Striving for a lower coupling is the basic tenet underlying this chapter.

### 3.4 A Structural Model of Service Adaptability and Changeability

One of the causes for the diverse variety of existing services is the evolution of specifications. An *adaptation* of a service is a set of changes made to that service due to a new specification. Several variants of the original service may thus coexist.

We assume that services can be atomic (with just one exposed feature) or structured. We consider only the structural aspects and assume that adaptations and changes to atomic services are also atomic (no intermediate stages).

The *similarity* between a service after adaptation and its previous specification is defined in terms of the similarities of its features, as:

$$S = \begin{cases} 0 & \text{changed atomic service} \\ 1 & \text{unchanged atomic service} \\ \frac{\sum_{i \in T} S_i}{|T|} & \text{structured service} \end{cases} \quad (3)$$

where  $T$  is the set of features exposed by this service and  $S_i$  is the similarity of feature  $i$  of the service. A similarity of 1 means that nothing has changed, whereas a similarity of 0 means that all features of a service have changed.

The *adaptability* of a service expresses how easily it can suffer a given adaptation. As a metric, a value of 0 in adaptability means that the service cannot be adapted and is unable to support the new intended specification, due to some limitation, and a value of 1 means that the cost or effort of adaptation is zero. It depends essentially on similarity and coupling:

$$A = S \cdot (1 - C_F) \quad (4)$$

in which  $A$  is the adaptability of a service,  $C_F$  is the forward coupling (the coupling between the service and its providers), and  $S$  is the similarity between the specification of the service before and after the adaptation.

Adaptability does not depend on which consumers use the service being adapted and reflects only the ability (*can* it be adapted?) and the cost/effort to adapt it. Many changes (low  $S$ ) or a high dependency on other services (high  $C_F$ ) reduce adaptability.

The complementary adaptation question (*may* it be adapted?) is included in its *changeability* property  $Ch$  [51], defined here as:

$$Ch = S \cdot (1 - C_F) \cdot (1 - C_B) \quad (5)$$

or

$$Ch = A \cdot (1 - C_B) \quad (6)$$

$C_B$  is the *backward coupling* between the service being adapted and its consumers, expressing the impact of the adaptation of the service. If it has a low adaptability, or if many of its consumer services are affected (high  $C_B$ ), changeability becomes lower than desirable.

All the variables in Eqs. 5 and 6 vary between 0 and 1. Any factor with a low value becomes dominant and imposes a low value on the changeability, which translates to a poor service architecture or implementation.

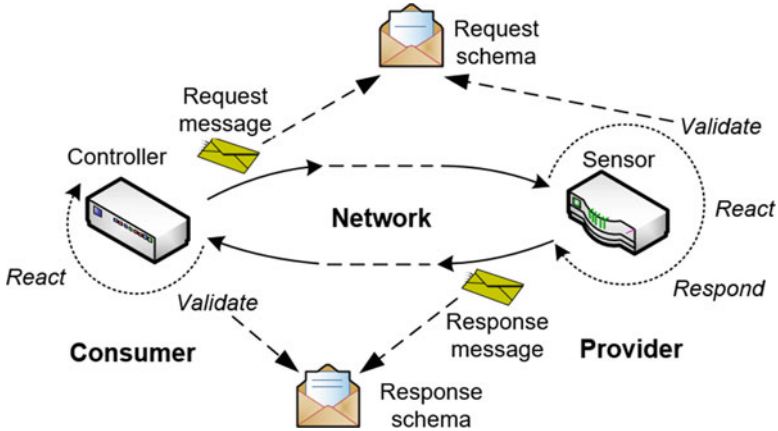
Therefore, for a given similarity, which expresses the degree of changes made, a service is more changeable (impacts less its consumers and its use of providers) if it has a lower forward and backward coupling. This is consistent with the conclusions drawn from the coupling model described in Sect. 3.3.

## 4 Existing Technologies: Symmetric Service Interoperability

Minimizing coupling as much as possible is therefore one of the main problems to deal with in order to improve interoperability. Existing technological approaches provide message-based interoperability, mainly at the Interface (service) and Structure (schema) layers, as described by Table 1, but typically use more coupling than actually required.

Messages are serialised data structures described by schemas and the typical interoperability approach used in distributed systems is to share the message's schema between the sender and the receiver of that message, as illustrated by Fig. 3. The scenario of Fig. 1 has been extended by making the schema sharing explicit, for both the request and response messages.

The same schema is used for both generating a message at the sender and validating it at the receiver. This is designated *symmetric interoperability*, since both



**Fig. 3** Schema sharing in symmetric message-based interaction

sender and receiver need to have the same knowledge about the message. The sender can produce any structured value allowed by the message schema and therefore the receiver needs to be able to read any of these values. Both sender and receiver work on the same message, with the same schema.

This is the basic mechanism used in classic document-based interoperability, using data description languages such as XML or JSON, in which a writer produces a document according to some schema and the reader uses the same schema to validate and to read the contents of that document. It has the following main drawbacks:

- The receiver needs to deal with the message using the schema of the sender, which produced the message. This usually implies endowing the receiver with a stub (interface code) that knows the schema and how to access the message components (data binding).
- The receiver needs to deal with the message using the ontology (message component names, namely) of the sender, which produced the message. An ontology mapping, between the message and the receiver, is required for the receiver to be able to interpret the message's semantics.
- The sender and the receiver are coupled for the entire range of values supported by the schema, even if only a fraction of that range is actually used in the interactions. When this happens, coupling is higher than the interacting services actually require.

In more classic, coarse-grained services, interaction is made symmetric by design, i.e., sender and receiver are designed to work together, under some common specification. This is typical of the SOA architectural style [12], in which each service has its own API, which means that a consumer using the functionality of a provider needs to know the operations and semantics of the interface of the latter. However, this hampers decoupling, changeability, and scalability, which



constitutes one of the main criticisms of SOA. This is particularly relevant in the field of smart cities, in which devices and their services are small, frequently change their configuration and exhibit an enormous variability of functionality and manufacturers.

To avoid this problem, proponents of the REST architectural style [13] contend that the basic modelling entity should be the resource, not the service, and that a client (consumer resource) should know just the link (such as a Uniform Resource Identifier – URI, in Web terms) to a server (provider resource), not its specific API. From that, a universal operation can be used to obtain the server's representation, which can contain links that can in turn be accessed, using only a fixed set of universal operations, supported by all the clients [52].

Fielding, the creator of REST, designated this as “hypermedia as the engine of application state” (HATEOAS) [53]. The basic idea is that the client (consumer) needs to know very little about the server (provider), since it only follows the links that the server provides, and that the server needs to know nothing about the client, which has the responsibility to decide which link to follow. The intended goal is to minimize coupling and to maximize scalability.

However, this is an elusive goal. Apparently, if the server changes the links it sends in the responses, the client will follow this change automatically by using the new links. The problem, however, is that this is not as general as it may seem, since the client must be able to understand the structure of the responses. It is not merely a question of blindly following all the links in a response. Moreover, just stating the data syntax (using languages such as XML or JSON) is not enough. The semantics and the actual set of names used (the schema, in fact) must be known by both client and server [54].

In the end, and in what coupling is concerned, there is no real difference from what happens in SOA, in which the schema of the service interface must be shared between consumer and provider. SOA is guided by services (behaviour), whereas REST is guided by resources (state). REST uses schemas of resources instead of services, but the consumer and the provider still need to share the schemas of the messages and the coupling is still there.

REST trades interface variability for structure variability, something that SOA lacks. REST cleanly separates the mechanism of traversing the graph of possible client-server interaction states from the processing of individual graph nodes (interaction states). Therefore, varying the structure allows changing the overall behaviour without affecting the traversal mechanism. However, this requires that all nodes are treated alike, which means that all nodes must have the same interface. This implies decomposing the SOA-style objects into their most elementary components and treat them all as first-class resources, which in turn leads to a state diagram (instead of a class diagram) modelling style.

The main problem with this is that the model is no longer guided by the static entities of the problem, in an object-oriented fashion, but rather by state, as an automaton. Although most people will find it harder to model state transitions than static entities (classes), this is not real a problem for simpler applications that can be organized in a CRUD (Create, Read, Update, Delete) approach, a natural method

when structured state is the guiding concept. The service is still there but it has a universal interface, common to all resources.

In fact, many applications are simple enough and the technologies typically used to implement REST are simpler, lighter, and in many cases cheaper than those used to implement SOA (namely, SOAP-based Web Services).

This justifies the growing popularity of RESTful applications and their APIs, but the level of resource coupling in REST is not lower than that of service coupling in SOA, since both require that the schemas used are known by both interacting applications.

It should also be noted that SOA lacks support for structured resources. Services (the set of operations supported by a resource) have just one level, offering operations but hiding any internal structured state. Structure is a natural occurrence in most problem domains and in this respect REST may constitute a better match.

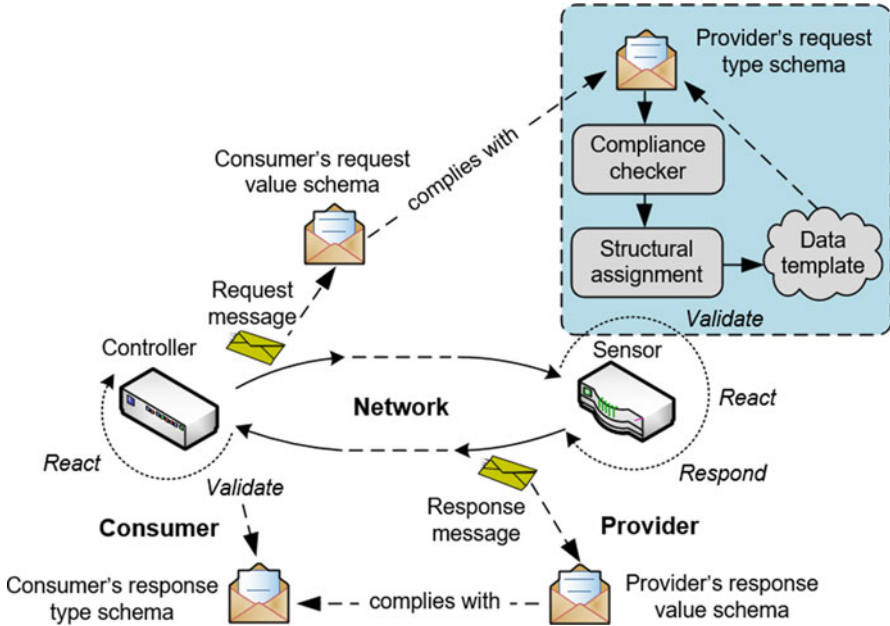
## 5 A Different Approach: Asymmetric Service Interoperability

### 5.1 What Is Asymmetric Interoperability?

In symmetric interoperability (Fig. 3), both sender and receiver share the same message schema. Figure 4 illustrates *asymmetric interoperability*, in which the schema of the message sent does not need to be the same as the one used by the receiver to read that message. The schema used by the sender needs only to be compatible with that of the receiver, but not necessarily the other way around. Hence the asymmetry.

Figure 4 details only the request message validation at the provider, for simplicity, but an identical mechanism is used by the consumer to validate the provider's response message. This mechanism can be understood in light of the following notes:

- The provider specifies and exposes a request *type schema*, which specifies the range of request message values that the provider is willing to accept.
- The request message sent by the consumer includes a *value schema*, but this is just a self-description, including no variability (unlike type schemas, which typically describe a range of structured values).
- When the request message arrives at the provider, the message's value schema is checked against the type schema of the provider, in the *compliance checker*. If the former *complies* with the latter, the message is accepted, which in practice means that the request message is one of those that satisfy the provider's request type schema.
- If compliance holds, the request message's value is *structurally assigned* to the *data template*, which is a data structure that satisfies the provider's type schema and is partly filled in with default values for the components in the type



**Fig. 4** Asymmetric message-based interaction. Only the request message validation is shown

schema that are optional, i.e., with minimum cardinality (number of occurrences) specified as zero.

- Structural assignment involves mapping the value schema of the consumer's request message to the provider's request type schema, by assigning the message to the data template, component by component (not as a whole value), according to the following basic rules:
  - Components in the message that do not comply with any component in the data template are ignored (not assigned).
  - Optional components in the data template with no counterpart in the message keep their default value.
  - Components in the data template that have counterparts in the message have their values set to the corresponding message's component values.
  - Structured components are assigned by recursive application of these rules.

After this, the components of the data template are completely populated and ready to be accessed by the receiver. Each request message received populates a new instance of the data template. This mechanism is different from the usual data binding of existing technologies, in particular in the following aspects:

- The provider deals only with its own request message schema. It does not know which is the actual schema of the request message and there is no need for a stub to deal with it.

- The mapping between the request message and the data template is done in a universal manner, by a message-based platform. It does not depend on the request schema of either the consumer or the provider.
- The structural assignment rules mean that coupling is reduced by comparison with symmetric interoperability (Sect. 4), since:
  - Only the actual components of the request message are used in the structural assignment. The missing ones use default values.
  - Additional and less stringent component matching rules are possible besides having a common name, such as by position and by type.

## 5.2 Illustrating Compliance

Compliance as described by Fig. 4 applies to data types and can be used directly in RESTful APIs, by a client that receives a resource representation from a server. Compliance means that the schema of that representation just needs to include the minimum set of features that the client expects, not actually be the latter.

To illustrate this concept, suppose that we have a device that implements a weather sensor service with a RESTful API. Since these APIs emphasize the response data, rather than the request, Listing 1 illustrates the JSON representation of the weather sensor returned upon reception of a GET request.

### Listing 1 A representation of a weather sensor, in JSON

```
{
  "temperature": 20,
  "temperature_unit": "Celsius",
  "average_temperature": 16.3,
  "humidity": 72.5
}
```

Let us now assume that the client that sent the GET request expects responses according to the type schema shown in Listing 2.

### Listing 2 A JSON Schema describing the data expected by a simple client

```
{
  "$schema": "http://json-schema.org/schema#",
  "type": "object",
  "required": ["temperature", "temperature_unit"],
  "properties": {
    "temperature": { "type": "number" },
    "temperature_unit": { "enum": ["Celsius", "Fahrenheit"] }
  }
}
```

The weather sensor's response (Listing 1) *complies* with the client's response type schema (Listing 2). The temperature property accepts any number, which encompasses the 20 stated by the weather sensor's representation. The weather sensor uses just the Celsius scale in temperature\_unit, which is also a subset of

the scales supported by the client. The client ignores the `average_temperature` and `humidity` properties of the weather sensor's representation.

Compliance means that the representation of the weather sensor can be structurally assigned to the client's data template (Fig. 4). The client's code will only have access to the properties it has declared in its own schema, and will never know that the representation returned by the weather sensor had additional properties.

This is mapping by component names and requires the same ontology (same component names on both schemas). This can be avoided by mapping by position, as shown in Listing 3, in which component names are different but the relative positions are the same, and the component types match. In this case, the `temperature` component is assigned to the `temp` component, and the `temperature_unit` component is assigned to the `unit` component.

**Listing 3 A JSON Schema describing the data expected by a client with a compatible structure, but different ontology (mapping by position)**

```
{
  "$schema": "http://json-schema.org/schema#",
  "type": "object",
  "required": ["temp", "unit"],
  "properties": {
    "temp": { "type": "number" },
    "unit": { "enum": ["Celsius", "Fahrenheit"] }
  }
}
```

Finally, mapping can still be done by type (without component names), which can even support components in different positions, as long as the mapping of component types is unambiguous, as shown in Listing 4. In this case, components have to be specified in a JSON array, since they have no name, but the rules of Table 4 support this. Note that the order of the components is not the same as in the previous listings, to show that the order is not relevant in mapping by type. However, the first component to match a type is used, and the data returned by the weather sensor has three components that match the type `number`, which means that mapping by type should be used with care and only when there is no ambiguity.

**Listing 4 A JSON Schema describing the data expected by a client, without component names (mapping by type)**

```
{
  "$schema": "http://json-schema.org/schema#",
  "type": "array",
  "minItems": 2,
  "maxItems": 2,
  "items": [
    { "enum": ["Celsius", "Fahrenheit"] },
    { "type": "number" }
  ]
}
```

Current technologies support only mapping by name, which means that Listings 3 and 4 are illustrative only.

To illustrate non-compliance, suppose that the representation returned by the weather sensor was not that of Listing 1 but instead the one depicted in Listing 5. The structure matches the schema of Listing 2, but not the type of the `temperature_unit` component. There is also no mapping by position (schema of Listing 3) or by type (schema of Listing 4), which means that the only way to support interoperability in this case would be to insert an adapter that would convert Kelvin to Celsius or Fahrenheit units.

**Listing 5 A representation of a weather sensor that does not comply with the schema of Listing 2**

```
{
  "temperature": 20,
  "temperature_unit": "Kelvin"
}
```

### 5.3 A Data Schema Model

To provide a more formal account of asymmetric interoperability and to provide further insight into this matter, it is important to realize that messages are serialised data structures that can be described by some data schema model.

A structural data interoperability mechanism needs to be based on:

- A set of built-in data types and the respective values, considered atomic (not composed of other values).
- A set of structuring mechanisms that enable the construction of arbitrarily complex structured (non-atomic) data types and the respective values.

The actual choice of these sets is not important. However, they constitute a foundation for data interoperability and must be known and shared by all interacting services. This is a principle already used by existing interoperability approaches. Table 2 illustrates possible sets of built-in types and structuring mechanisms, loosely based on those of XML.

The Union types are simply sets of types, each of which may be any of those in Table 2. Values belong to (satisfy) a Union type if they belong to at least one of its

**Table 2** Possible sets of built-in and structured data types

Data type category	Data type	Description
Built-in types	Integer	Integer numbers
	Float	Real numbers
	Boolean	True or false
	String	Strings
Structured types	Record	An unordered set of components
	List	An ordered set of components
Choice type	Union	A set of data types, any of which can be chosen

**Table 3** Attributes specified for each component of the structured types

Attribute	Letter	Description
Name	N	Name of the component, possibly qualified by some ontology (just on Records)
Position	P	Ordering number of the component (on Records, position is the order by which components appear in the specification)
Type	T	Type of the component (any of the types of Table 2)
Minimum cardinality	m	Minimum number of occurrences of components with this name
Maximum cardinality	M	Maximum number of occurrences of components with this name

member types. Contrary to many type systems, a value does not actually belong to just one type, but to all that it satisfies. Type satisfaction is explained below, in Sect. 5.4.

The Record and List structured types consist of a set of components (not necessarily belonging to the same type), each of which has the attributes described in Table 3. Attribute letters are used in Sect. 5.4.

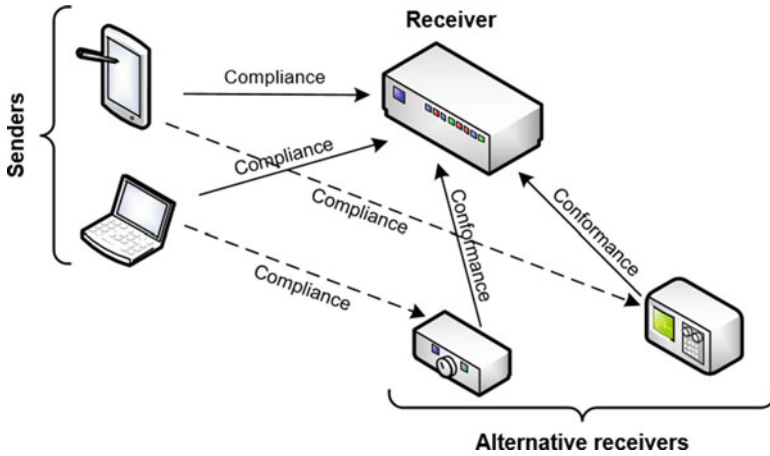
In Fig. 4, a value schema is a data type in which the type attribute of each component has been reduced to a single value and its cardinality has been fixed (the minimum and maximum cardinalities are identical). Therefore, it corresponds to the data structure of a message simply with a self-description. A type schema corresponds to what Tables 2 and 3 describe.

A data template is a type schema that additionally specifies a default value for each optional component (with a minimum cardinality of zero). If the message includes a matching component, it is assigned to the corresponding component of the data template; otherwise, the default value is used. The data template becomes fully populated after the structural assignment, even if the message lacks some components. Mandatory components (those with a minimum cardinality greater than zero) need to be present, without which the message cannot be accepted.

## 5.4 Structural Service Interoperability

In asymmetric interoperability (Fig. 4), the value schema of a message needs only to *partially* match the type schema of the receiver. This decreases coupling, since the message's value schema just needs to comply with the absolute minimum requirements of the receiver's type schema. Even components that are mandatory can be made optional, thanks to the default values of the receiver's data template.

This means that a service can use (send request messages to) various other services, as long as these messages match the relevant parts of those services' request type schemas. It also means that a service can be replaced by another one, with a different type schema, as long as it can deal with all the requests the old one



**Fig. 5** Illustration of the compliance and conformance relations

could. This can occur due to evolution of the receiver (replaced by a new version) or by resorting to a new receiver altogether.

Allowing a receiver to be able to interpret messages from different senders, and a sender to be able to send messages to different receivers, is what Eqs. 1 and 2 show that is required to decrease coupling. Note that a sender/receiver pair deals with one message, whereas a consumer/provider pair may require two sender/receiver pairs, for the request and response messages, but the considerations are valid for each of these messages.

These *use* and *replace* relationships leads to two important schema relations, which are central to asymmetric interoperability:

- *Compliance* [39]. The requests generated by the sender must satisfy (*comply with*) the *minimum* set of requirements established by the receiver's type schema to accept requests.
- *Conformance* [40]. The alternative receiver's type schema must include the *maximum* set of requirements established by the original receiver's type schema to accept requests. Therefore, the alternative receiver is able to take the form of (*conform to*) the original receiver and to continue to support any existing sender.

These relations are not symmetric (e.g., if  $X$  complies with  $Y$ ,  $Y$  does not necessarily comply with  $X$ ) but are transitive (e.g., if  $X$  complies with  $Y$  and  $Y$  complies with  $Z$ , then  $X$  complies with  $Z$ ). Figure 5 illustrates these relations between several services, running on various devices.

In semantic terms, compliance means that the set of possible message values sent by a sender is a *subset* of the set of values that satisfy the type schema of the receiver. Conformance means that the set of values that satisfy the type schema of an alternative receiver is a *superset* of the set of values that satisfy the type schema of the original receiver.



As long as compliance and conformance hold, the receiver can accept messages from different senders. In addition, a sender can start using an alternative receiver without noticing the difference with respect to the original receiver.

The compliance and conformance relations obey the following rules (denoting compliance and conformance between types  $X$  and  $Y$  by  $X \blacktriangleleft Y$  and  $X \blacktriangleright Y$ , respectively):

- Each built-in type complies with and conforms to just itself, with the exception that Integer complies with Float (subset) and Float conforms to Integer (superset).
- A Union type  $U$  complies with a built-in type  $B$  only if each member type of  $U$  complies with type  $B$ .
- A Union type  $V$  conforms to a built-in type  $C$  only if at least one member type of  $V$  conforms to type  $C$ .
- Tables 4 and 5, respectively, describe compliance and conformance between the more complex types of Table 2, structured and choice.

**Table 4** Rules for compliance of a type  $X$  with another type  $Y$ . The subscripts  $i$  and  $j$  designate component/member type and the letters designate component attributes (Table 3)

$\blacktriangleleft$	Type $Y$		
Type $X$	Record	List	Union
Record	If, for each $Y_i$ , there is a $X_j$ such that $X_{jN}=Y_{iN}$ , $X_{jT} \blacktriangleleft Y_{iT}$ , $X_{jm} \geq Y_{im}$ , and $X_{jM} \leq Y_{iM}$	If, for each $Y_i$ , there is a $X_j$ such that $X_{jP}=Y_{iP}$ , $X_{jT} \blacktriangleleft Y_{iT}$ , $X_{jm} \geq Y_{im}$ , and $X_{jM} \leq Y_{iM}$	If $X$ complies with at least one $Y_i$
List	If, for each $Y_i$ , there is a $X_j$ such that $X_{jP}=Y_{iP}$ , $X_{jT} \blacktriangleleft Y_{iT}$ , $X_{jm} \geq Y_{im}$ , and $X_{jM} \leq Y_{iM}$	If, for each $Y_i$ , there is a $X_j$ such that $X_{jP}=Y_{iP}$ , $X_{jT} \blacktriangleleft Y_{iT}$ , $X_{jm} \geq Y_{im}$ , and $X_{jM} \leq Y_{iM}$	If $X$ complies with at least one $Y_i$
Union	If all $X_j$ comply with $Y_i$	If all $X_i$ comply with $Y_i$	If each $X_j$ complies with at least one $Y_i$

**Table 5** Rules for conformance of a type  $W$  to another type  $Z$ . The subscripts  $i$  and  $j$  designate component/member type and the letters designate component attributes (Table 3)

$\blacktriangleright$	Type $Z$		
Type $W$	Record	List	Union
Record	If, for each $Z_i$ , there is a $W_j$ such that $W_{jN} = Z_{iN}$ , $W_{jT} \blacktriangleright Z_{iT}$ , $W_{jm} \leq Z_{im}$ , and $W_{jM} \geq Z_{iM}$ , and, for all remaining $W_j$ , $W_{jm} = 0$	If, for each $Z_i$ , there is a $W_j$ such that $W_{jP} = Z_{iP}$ , $W_{jT} \blacktriangleright Z_{iT}$ , $W_{jm} \leq Z_{im}$ , and $W_{jM} \geq Z_{iM}$ , and, for all remaining $W_j$ , $W_{jm} = 0$	If $W$ conforms to all $Z_i$
List	If, for each $Z_i$ , there is a $W_j$ such that $W_{jP} = Z_{iP}$ , $W_{jT} \blacktriangleright Z_{iT}$ , $W_{jm} \leq Z_{im}$ , and $W_{jM} \geq Z_{iM}$ , and, for all remaining $W_j$ , $W_{jm} = 0$	If, for each $Z_i$ , there is a $W_j$ such that $W_{jP} = Z_{iP}$ , $W_{jT} \blacktriangleright Z_{iT}$ , $W_{jm} \leq Z_{im}$ , and $W_{jM} \geq Z_{iM}$ , and, for all remaining $W_j$ , $W_{jm} = 0$	If $W$ conforms to all $Z_i$
Union	If at least one of $W_j$ conforms to $Z$	If at least one $W_j$ conforms to $Z$	If, for each $Z_i$ , there is at least one $W_j$ that conforms to it

Mapping Records to Lists and Lists to Records allow structural assignment by position instead by name, considering the position of each named component in Records as the position it occupies in its definition or declaration.

There is still another possibility, mapping by component type. In this case, components are assigned to those that comply with (or conform to) the other type. The advantage of this is to avoid needing to have exactly the same name in corresponding components. This cannot always be used, since different components can have the same type but different semantics.

These rules can lead to ambiguities, i.e., matching solutions that are not unique, in particular when unions are involved. In this case, the solution adopted can depend on the implementation. Types should be chosen to avoid ambiguities, or a compiler can check them and eventually generate an error.

Extending the compliance and conformance concepts to services, at the Interface layer (Table 1), is straightforward. Consider a service  $C$  (the consumer) and a service  $P$  (the provider).  $C$  can invoke some of the operations of  $P$ . For each invoked operation, we consider:

- $Crq$  – The value schema of the request message, sent by the consumer.
- $Prq$  – The type schema of the request message, expected by the provider.
- $Prp$  – The value schema of the response message, sent by the provider.
- $Crp$  – The type schema of the response message, expected by the consumer.

A consumer  $C$  is compliant with (can *use*) a provider  $P$  ( $C \blacktriangleleft P$ ) if, for all operations  $i$  of  $P$  that  $C$  invokes,  $Crq_i \blacktriangleleft Prq_i$  and  $Prp_i \blacktriangleleft Crp_i$ . Structural assignment is used to assign a message received (either request or response) to the data template of the receiver (Fig. 4). In a similar way, a provider  $S$  is conformant to (can *replace*) a provider  $P$  ( $S \blacktriangleright P$ ) if, for all operations  $i$  of  $P$ ,  $Srq_i \blacktriangleright Prq_i$  and  $Prp_i \blacktriangleright Srp_i$ .

## 5.5 Illustrating Conformance

Section 5.2 provides some examples of compliance, by showing that the schema of a message just needs to include the minimum set of features that the receiver of that message expects and not all the features that the client supports. Conformance means that the client can be replaced by another one that includes all the features of the original one, although it can also include different features.

To illustrate conformance, suppose that we replace the client of Listing 2 with a new version that is now able to make use of the `average_temperature` and `humidity` properties of the weather sensor's representation. Its schema is represented in Listing 6.

**Listing 6 A JSON Schema describing a new client, conformant to the previous one**

```
{
  "$schema": "http://json-schema.org/schema#",
  "type": "object",
  "required": ["temperature", "temperature_unit"],
  "properties": {
    "temperature": { "type": "number" },
    "temperature_unit": { "enum": ["Celsius",
    "Fahrenheit"] },
    "average_temperature": { "type": "number" },
    "humidity": { "type": "number" }
  }
}
```

This new client *conforms* to the old one, since it includes the properties of the latter and the additional properties are optional (not required). The reason for not allowing new mandatory features is that a transparent client replacement requires that the new client must also accept all the weather sensor representations that the old client could accept. This means that properties ignored by the old client cannot be mandatory in the new client.

Listing 7 illustrates a case of non-conformance. The new client does not support temperatures in Celsius units, which means that it will not support the sensors that return representations using these units, which would be readily accepted by the original client. The representation of Listing 5 would now comply with it, though. Therefore, the best client would be one that can accept all three temperature units.

**Listing 7 A JSON Schema describing a new client, conformant to the previous one**

```
{
  "$schema": "http://json-schema.org/schema#",
  "type": "object",
  "required": ["temperature", "temperature_unit"],
  "properties": {
    "temperature": { "type": "number" },
    "temperature_unit": { "enum": ["Kelvin", "Fahrenheit"] },
    "average_temperature": { "type": "number" },
    "humidity": { "type": "number" }
  }
}
```

Similar examples to those of Listings 1, 2, 3, 4, 5, 6, and 7 could be provided using XML Schema, but these would be more verbose. Compliance and conformance can also be defined for services, in particular for Web Services, using XML, with the rules described in Sect. 5.4 [55].

## 5.6 *Usefulness of the Approach*

The approach described in this chapter is just exploratory research and, in terms of practical implementation, there is only a preliminary implementation of the compliance and conformance algorithms. Their applicability, effectiveness and actual benefits have yet to be assessed in real-case scenarios.

The potential, however, is relevant. Whenever there is an interaction, compliance and conformance apply at each of the levels described by Table 1. Examples of applicability of the asymmetric interoperability approach include:

- Document-based interoperability. Conventionally, the writer of an electronic document also makes its schema available, and the receiver needs to produce a stub that supports the data binding necessary to be able to read the document. This means that the receiver needs to know the details of the document, leading to a high level of coupling. Asymmetric interoperability provides data binding at a universal level, according to the rules enunciated above for compliance and conformance. Any reader with which the actual document complies can read it, even if it does not support the full range of documents encompassed by its published schema. This lowers coupling and does not require specific stubs.
- Service-based interoperability. The canonical solutions at this level are based on the SOA and REST architectural styles. However, SOA requires sharing of a service description, such as a WSDL (Web Services Description Language) file, which has the same coupling drawbacks as a document's schema. REST-based technologies are simpler, but not better in terms of coupling, since they require both consumer and producer of the service to use the same schema. Forcing all services to have the same interface does not reduce coupling. In addition, SOA is based on service behaviour and REST on resource structure. Compliance and conformance enable interoperability without sharing schemas and support both behaviour and structure. This has led to the proposal of a mixed architectural style, Structural Services [56], which combines the advantages of both SOA and REST.
- Cloud interoperability. Each cloud provider has its own API, incompatible with the others, although much of the functionality they provide is basically the same. This raises a cloud interoperability problem. Unfortunately, big cloud providers are not particularly interested in adopting common standards, since that would eliminate their technological advantages. A possible solution would be to define standards for the basic, common functionalities, with additional, specific features supported by each provider. Compliance and conformance could then be used to enjoy the common interfaces while still being able to use the provider-specific features.
- Hierarchically organized specifications, such as SDN (Software Defined Networking) [57], with its application, control and data planes, and associated interfaces. Compliance and conformance, as well as backward and forward coupling, apply at each interface. Instead of requiring rigid interface specifications, a common set of features can be defined, while still being able to make use of

manufacturer-specific features, something usually appealing to big companies providing networking equipment.

## 6 Conclusion

Using symmetric interoperability, the communications interface with general-purpose programming languages, at either side of the interacting devices, is usually done with stubs, with code generated automatically from the shared schema, typically resorting to annotations. If the schema changes, the stubs have to be generated again, on both sides of the interaction.

With asymmetric interoperability, the schemas of the sender and of the receiver become independent. They just need to comply and, as long as compliance is not impaired, one can be changed without impacting the other.

The most relevant aspect of this mechanism is that the receiving service does not deal with the message, only with the data template and the components for which it has been designed. The assignment of relevant the parts of the message to the data template is done in a universal way, independently of the types of the actual message or data template. These types have become decoupled, except for the components that are really needed for the interaction (minimum coupling possible).

In addition, it should be noted that the serialisation format is not relevant. Text such as XML and JSON, or binary such as or Concise Binary Object Representation – CBOR [58] – and Efficient XML Interchange – EXI [59]). As long as messages can be parsed (deserialised) and the semantic information (namely, component names) is present, this mechanism can be implemented. Naturally, both sender and receiver need to use the same serialisation format.

These interoperability features contribute to reduce coupling and to increase the range of services that can interact with a given service. This is especially relevant in the context of smart cities and the Internet of Things, in which most of the services are implemented in small devices and interact in huge numbers and with a wide variety of characteristics. Therefore, reducing the interoperability problems, namely coupling, is of paramount importance.

## References

1. Mohanty S, Choppali U, Kougianos E (2016) Everything you wanted to know about smart cities: The internet of things is the backbone. *IEEE Consum Electron Mag*, 5.3:60–70
2. World Health Organization (2016) Global report on urban health: equitable healthier cities for sustainable development. World Health Organization. Available via [http://www.who.int/kobe\\_centre/measuring/urban-global-report/en/](http://www.who.int/kobe_centre/measuring/urban-global-report/en/). Accessed 26 Oct 2017
3. Yaqoob I, Ahmed E, Hashem I, Ahmed A, Gani A, Imran M, Guizani M (2017) Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. *IEEE Wirel Commun*, 24.3:10–16

4. Rayes A, Samer S (2017) *Internet of Things—From Hype to Reality*. Springer International Publishing, Cham, Switzerland
5. van der Meulen R (2017) Gartner Says 8.4 Billion Connected “Things” Will Be in Use in 2017, Up 31 Percent From 2016. Available via <https://www.gartner.com/newsroom/id/3598917>. Accessed 26 Oct 2017
6. Nordrum A (2016) Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated. Available via <http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated>. Accessed 26 Oct 2017
7. Ahlgren B, Hidell M, Ngai E (2016) Internet of Things for Smart Cities: Interoperability and Open Data. *IEEE Internet Comp*, 20.6:52–56
8. Gyrard A, Serrano M (2016) Connected smart cities: Interoperability with SEG 3.0 for the internet of things. In: *Proceedings of the 30th IEEE International Conference on Advanced Information Networking and Applications*, p 796–802. IEEE Computer Society Press, Piscataway
9. Gaziz V (2017) A Survey of Standards for Machine-to-Machine and the Internet of Things. *IEEE Commun Surv & Tutor*, 19.1:482–511
10. Fawcett J, Ayers D, Quin L (2012) *Beginning XML*. John Wiley & Sons, Hoboken
11. Bassett L (2015) *Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON*. O’Reilly Media, Inc, Sebastopol
12. Erl T (2016) *Service-oriented architecture: concepts, technology, and design (2nd Edition)*. Prentice Hall, Upper Saddle River
13. Pautasso C, Wilde E, Alarcon R (ed) (2014) *REST: advanced research topics and practical applications*. Springer, New York
14. Khatoun R, Zeadally S (2016) Smart cities: concepts, architectures, research opportunities. *Commun ACM*, 59:8, 46–57
15. Rathore M, Ahmad A, Paul A, Rho S (2016) Urban planning and building smart cities based on the internet of things using big data analytics. *Comp Netw*, 101:63–80
16. Doran D, Severin K, Gokhale S, Dagnino A (2016) Social media enabled human sensing for smart cities. *AI Commun*, 29:1, 57–75
17. Vinod Kumar T, Dahiya B (2017) Smart Economy in Smart Cities. In: Vinod Kumar T (ed) *Smart Economy in Smart Cities. Advances in 21st Century Human Settlements*. Springer, Singapore. In: Vinod Kumar T (ed) *Smart Economy in Smart Cities*, p 3–76. Springer, Singapore
18. Calvillo C, Sánchez-Miralles A, Villar J (2016) Energy management and planning in smart cities. *Renew Sustain Energy Rev*, 55:273–287
19. Peris-Ortiz M, Bennett D, Yábar D (eds) (2017) *Sustainable Smart Cities. Innovation, Technology, and Knowledge Management*. Springer International Publishing, Switzerland
20. Botta A, de Donato W, Persico P, Pescapé A (2016) Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, 56:684–700
21. Sun H, Wang C, Ahmad B (eds) (2017) *From Internet of Things to Smart Cities: Enabling Technologies*. CRC Press
22. Qin Y, Sheng Q, Falkner N, Dustdar S, Wang H, Vasilakos A (2016) When things matter: A survey on data-centric internet of things. *J Netw Comp Appl*, 64:137–153
23. Nitti M, Pilloni V, Colistra G, Atzori L (2016) The virtual object as a major element of the internet of things: a survey. *IEEE Commun Surv Tutor*, 18.2:1228–1240
24. Panetto H, Zdravkovic M, Jardim-Goncalves R, Romero D, Cecil J, Mezgár I (2016) New perspectives for the future interoperable enterprise systems. *Comp Ind*, 79:47–63
25. Gil-Garcia J, Zhang J, Puron-Cid G (2016) Conceptualizing smartness in government: An integrative and multi-dimensional view. *Gov Inf Q*, 33.3:524–534
26. Díaz M, Martín C, Rubio B (2016) State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *J Netw Comp Appl*, 67:99–117
27. Pramanik M, Lau R, Demirkan H, Azad M (2017) Smart health: Big data enabled health paradigm within smart cities. *Expert Syst Appl*, 87:370–383

28. Rashid B, Rehmani M (2016) Applications of wireless sensor networks for urban areas: A survey. *J Netw Comp Appl*, 60:192–219
29. Laouiti A, Qayyum A, Saad M (eds) (2016) *Vehicular Ad-Hoc Networks for Smart Cities*. Springer, Singapore
30. Zimmermann O, Tomlinson M, Peuser S (2012) *Perspectives on Web Services: Applying SOAP, WSDL and UDDI to Real-World Projects*. Springer Science & Business Media, New York
31. Pautasso C (2014) RESTful web services: principles, patterns, emerging technologies. In: Bouguettaya A, Sheng Q, Daniel F (ed) *Web Services Foundations*, p 31–51. Springer, New York
32. Guinard D, Ion I, Mayer S (2011) In search of an internet of things service architecture: REST or WS-\*? A developers' perspective. In: *Proceedings of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, p 326–337. Springer, Berlin, Heidelberg
33. Kumari S, Rath S (2015) Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration. In: *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, p 1656–1660. IEEE Computer Society Press, Piscataway
34. Bora A, Bezboruah T (2015) A Comparative Investigation on Implementation of RESTful versus SOAP based Web Services. *Int J Database Theory Appl*, 8.3:297–312
35. Sungkur R, Daiboo S (2015) SOREST, A Novel Framework Combining SOAP and REST for Implementing Web Services. In: *Proceedings of the Second International Conference on Data Mining, Internet Computing, and Big Data*, p 22–34. The Society of Digital Information and Wireless Communications, Wilmington
36. Babu D, Darsi M (2013) A Survey on Service Oriented Architecture and Metrics to Measure Coupling. *Int J Comp Scie Eng*, 5.8:726–733
37. Geetika R, Singh P (2014) Dynamic coupling metrics for object oriented software systems: a survey. *ACM SIGSOFT Softw Eng Notes*, 39.2:1–8
38. Alenezi M, Magel K (2014) Empirical evaluation of a new coupling metric: combining structural and semantic coupling. *Int J Comp Appl*, 36.1:34–44
39. Tran H, Zdun U, Oberortner E, Mulo E, Dustdar S (2012) Compliance in service-oriented architectures: A model-driven and view-based approach. *Inf Softw Technol*, 54.6:531–552. <https://doi.org/10.1016/j.infsof.2012.01.001>
40. Khalfallah M, Figay N, Barhamgi M, Ghodous P (2014) Model driven conformance testing for standardized services. In: *Proceedings of the IEEE International Conference on Services Computing*, p 400–407. IEEE Computer Society Press, Piscataway
41. Capel M, Mendoza L (2014) Choreography Modeling Compliance for Timed Business Models. In: *Proceedings of the Workshop on Enterprise and Organizational Modeling and Simulation*, p 202–218. Springer, Berlin
42. Brandt C, Hermann F (2013) Conformance analysis of organizational models: a new enterprise modeling framework using algebraic graph transformation. *Int J Info Sys Model Des*, 4.1:42–78
43. Preidel C, Borrmann A (2016) Towards code compliance checking on the basis of a visual programming language. *J Inf Technol Constr*, 21.25:402–421
44. Graydon P, Habli I, Hawkins R, Kelly T, Knight J (2012) Arguing Conformance. *IEEE Softw*, 29.3:50–57
45. Rachad T, Boutahar J (2014) A new efficient method for calculating similarity between web services. *Int J Adv Comp Sci Appl*, 5.8:60–67
46. Otero-Cerdeira L, Rodríguez-Martínez F, Gómez-Rodríguez A (2015) Ontology matching: A literature review. *Expert Sys Appl*, 42.2:949–971
47. Athanassopoulos, D. (2017, June) Self-Adaptive Service Organization for Pragmatics-Aware Service Discovery. In: *Proceedings of the IEEE International Conference on Services Computing*, p. 164–171. IEEE Computer Society Press, Piscataway

48. ISO/IEC/IEEE (2010) Systems and software engineering – Vocabulary. International Standard ISO/IEC/IEEE 24765:2010(E). First Edition (p. 186). International Standards Office, Geneva
49. ISO/IEC (1994) ISO/IEC 7498–1, Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model, 2nd edition. International Standards Office, Geneva. Available via <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>. Accessed 26 Oct 2017
50. Wang W, De S, Toenjes R, Reetz E, Moessner K (2012) A comprehensive ontology for knowledge representation in the internet of things. In: Proceedings of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, p 1793–1798. IEEE Computer Society Press, Piscataway
51. Ross A, Rhodes D, Hastings D (2008) Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. *Syst Engineer* 11.3:246–262. <https://doi.org/10.1002/sys.20098>
52. Bloomberg J, Schmelzer R (2013) Deep Interoperability: Getting REST Right (Finally!). *The Agile Architecture Revolution: How Cloud Computing, Rest-Based SOA, and Mobile Computing are Changing Enterprise IT*. John Wiley & Sons, Inc., Hoboken
53. Fielding R (2000) Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California at Irvine. Available via [http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation\\_2up.pdf](http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation_2up.pdf). Accessed 26 Oct 2017
54. Palavalli A, Karri D, Pasupuleti S (2016) Semantic Internet of Things. In: Proceedings of the IEEE Tenth International Conference on Semantic Computing, p 91–95. IEEE Computer Society Press, Piscataway
55. Delgado J (2015) Decreasing Service Coupling to Increase Enterprise Agility. In: *Achieving Enterprise Agility through Innovative Software Development*, p 225–261. IGI Global, Hershey
56. Delgado J (2016) Bridging Services and Resources with Structural Services. *Int J Inf Sys Model Des*, 7.4:83–110
57. Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015) Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103.1:14–76
58. Bormann C, Hoffman P (2013) Concise Binary Object Representation (CBOR). Available via <https://tools.ietf.org/html/rfc7049>. Accessed 26 Oct 2017
59. Schneider J, Kamiya T, Peintner D, Kyusakov R (ed) (2014) Efficient XML Interchange (EXI) Format 1.0 (Second Edition). W3C. Available via <http://www.w3.org/TR/exi/>. Accessed 26 Oct 2017



# Management of Video Surveillance for Smart Cities



Nhat-Quang Dao, Quang Le-Dang, Robert Morawski, Anh-Tuan Dang, and Tho Le-Ngoc

**Abstract** Video surveillance system is a crucial component in the development of Smart City. Video data can be used for a myriad of applications, enabling many key services of Smart City such as smart traffic management and enhanced public security. This chapter provides an overview of video management system for Smart City and its challenges. A small-scale testbed with assorted video managing services is used to demonstrate and compare performance of on-premise and cloud-based infrastructures. In addition, we present several camera deployment scenarios to illustrate the connectivity and data volume that would emerge in a city-scale implementation.

**Keywords** Smart City Architecture · Smart City Video Surveillance

## 1 Introduction

Urbanization is a major global trend that is expanding quickly as more people are moving toward city lives. Currently, 54% of global population is living in cities, with predictions suggesting up to 66% in 2050.<sup>1</sup> Rapid urban growth is driving many improvements in infrastructure, transportation system, and living conditions. A new concept of Smart City recently emerges that incorporates new computing and networking technologies such as Internet of Things (IoT), Big Data, Cloud Computing, etc. to address the challenges of urbanization [1]. The objective of

---

<sup>1</sup>World Population Prospects – UN: <http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html>.

N.-Q. Dao · Q. Le-Dang · R. Morawski · A.-T. Dang · T. Le-Ngoc (✉)  
McGill University, Montréal, QC, Canada  
e-mail: [nhat-quang.dao@mail.mcgill.ca](mailto:nhat-quang.dao@mail.mcgill.ca); [quang.le2@mail.mcgill.ca](mailto:quang.le2@mail.mcgill.ca); [robert.morawski@mcgill.ca](mailto:robert.morawski@mcgill.ca);  
[anh-tuan.dang@mcgill.ca](mailto:anh-tuan.dang@mcgill.ca); [tho.le-ngoc@mcgill.ca](mailto:tho.le-ngoc@mcgill.ca)

Smart City is to use information and communication technologies to gather data at a massive scale and facilitate many smart services and applications to enhance quality of life for the citizens.

Smart video surveillance is becoming an increasingly important component in the vision of smart cities. As image processing and analytics technologies continue to advance, video data enables not just real-time monitoring for security, but also early detection and alert for environmental hazards such as fire or chemical leakage. In addition, continuous video footage can be used to analyze behavioral patterns of citizens, providing data for urban planners to optimize traffic flows in the city.

However, implementing a citywide smart video surveillance system involves many challenges. A large number of cameras have to be deployed across the city, spanning different geographical areas and producing enormous amount of information every day. In addition, storing and enabling access to video devices and data at this large scale, in both real-time and on-demand manner, demand more computing and storage resources than traditional video surveillance system. These challenges have to be addressed to maximize the effectiveness of smart video surveillance.

This chapter presents an overview of smart video surveillance system and management platform for Smart City. In particular, the chapter is organized as follows. Section 2 introduces the background of a typical video surveillance system in Smart City, including the key building blocks, prime examples of applying such system in Smart City and outlining the challenges. Section 3 investigates two approaches in implementing video management platform, on-premise and cloud-based approaches, by experimenting on a few video management systems (VMS) and software on a small-scale deployment to establish a performance benchmark. Section 4 explores a number of camera deployment scenarios to illustrate the challenges of implementing a city-scale smart video surveillance system. Finally, Section 5 concludes the chapter.

## **2 Background**

### ***2.1 Key Components for Smart Video Surveillance System***

Smart video surveillance, like many other IoT systems, can follow the proposed layered service-oriented architecture of Smart City infrastructure as described in our previous chapter “Internet of Things (IoT) Infrastructures for Smart Cities”. The key components are divided into layers: video acquisition layer, connectivity layer, management layer, and application layer. Following closely the previously proposed layered architecture, this section focuses on the key components of a smart video surveillance system for Smart City and the enabling technologies.

### 2.1.1 Video Acquisition Layer

The main function of video acquisition devices is straightforward: to provide real-time video monitoring of activities in a target area and produce video recordings, which can then be used for a myriad of other applications in areas such as traffic management, emergency response, and public security. The quality of video data collected in this layer varies not only with device capability, but also with application-specific requirements. For example, smart traffic programs that monitor traffic patterns only need enough details to distinguish vehicle contours from the surrounding and track their movement. On the other hand, public security applications such as person tracking, license plate recognition, and crowd estimation require powerful high-definition cameras that have sufficient resolution to enable identifying and tracking facial features, reading alphanumeric characters, as well as detecting events of interest such as fire or accidents in the area.

Smart camera is an essential enabling technology of this layer. Smart cameras are cameras that have integrated processing and communication capabilities in addition to the traditional image capture function [2]. Many models of cameras on the market already come with built-in image analytics capabilities such as motion detection, motion tracking, and face detection [3]. They also include pan-tilt-zoom (PTZ) functionality, enabling directional and zooming control, both manual and automatic, to focus on details for fine-scale analysis as well as provide wider area of surveillance than traditional fixed view cameras [4, 5]. More importantly, smart cameras can be communicated and controlled remotely through the Internet that truly enables remote re-configurations and firmware upgrade for the addition/enhancement of features as often required in video surveillance systems of Smart City [6, 7].

### 2.1.2 Connectivity Layer

The connectivity layer bridges the communication between remote video acquisition devices and the upper management and application layers. For most smart cities, the backbone networking infrastructure typically employs fiber-optic network for its high-speed data transmission, though wireless and cellular communications technologies have also been used, either as backbone network or as a bridge connecting devices to the backbone network [8–10]. In addition, a large-scale video surveillance system also needs a communications infrastructure to connect the devices to the backbone optical fiber network. The high bandwidth requirement of HD cameras poses a challenge for the infrastructure to support large-scale deployments. In an ideal scenario, a wired connection would be established between a fiber drop and each camera to provide sufficient bandwidth, but such arrangement lacks scalability and flexibility for Smart City deployment scenarios. Instead, wireless technologies, specifically 802.11 standards [11] and its derivatives, are considered to facilitate high throughput, real-time transmissions of video data to fiber drop gateways, as other lightweight protocols (e.g., 802.15.4-based networks

such as Zigbee [12, 13]) often used for typical wireless sensor networks can only provide limited support due to the huge bandwidth demands.

The network topology is another major consideration for wireless deployment of smart video surveillance system. When the surveillance environment is relatively small-scale, point-to-point communications can provide better runtime video delivery [14]. However, in a full-scale deployment over a large area with many cameras deployed in all directions around a gateway, point-to-point topology is not economical due to the high number of radio devices as each camera installation requires two radios to establish a point-to-point wireless bridge. In this case, a point-to-multipoint deployment, such as star topology, where an omni-directional access point can serve all cameras within the vicinity while greatly reduce the number and cost of radio devices necessary is more desirable. In addition, signal coverage can be extended to farther locations by use of single-hop or multi-hop relay access points, which can be placed anywhere to provide stronger signal for cameras that are in blind spots or outside the range of other access points. Nevertheless, bandwidth allocation must be planned carefully in this case with some reserved bandwidth for wireless link fluctuations and interference.

### 2.1.3 Management Layer

The management layer that enables smart video surveillance system consists of many key functionalities to facilitate data and device management. At the core of a management platform are the two planes of operation: the *data plane* and the *control plane*. The data plane outlines the operation of the data management subsystem, including how data collected from video cameras is formatted and archived to be available to upper layer applications. On the other hand, the control plane describes how the device management subsystem monitors the cameras themselves to report working status, document operational history, and enable remote control and maintenance. In order to facilitate these two planes of operation, the following core services need to be implemented.

- **Data Aggregation / Device Gateway** serves as the interface between management platform and video acquisition devices, exchanging data and commands between remote cameras and management modules.
- **Data Format Transformation** converts video data from various IP cameras into a uniform format to be processed and archived.
- **Data Storage** stores all video data collected to be retrieved by request of user applications.
- **Data Display** presents live or recorded video footage directly to users via a graphical interface.
- **Device Monitor** supervises the operation of remote cameras and reports any errors or notable information to operators via display interface.

- **Device Log** records the service history of deployed cameras in a data storage device, including health statuses, connectivity, user accesses, as well as any errors and notifications that arise.
- **Device Control** translates user commands (e.g., reboot, update firmware, change configuration, etc.) for device to function calls that the device can understand, enabling operators to control and maintain deployed devices in remote locations from control center.
- **Access Control** governs how external users and applications can access the data and services provided by Smart City management platform.

The usual approach of hosting video management platform is on-premises at a local facility such as a data center. This approach offers complete control over the hardware, software, and security measures of the management platform. As data is transferred in a closed system to a private facility, performance and security can be easily guaranteed. However, unlike typical video surveillance system employed in security of buildings and properties, smart video surveillance system for smart cities requires a large-scale management infrastructure that includes numerous devices from a variety of manufacturers. Bandwidth becomes a critical issue to overcome as the huge volume of video data collected across the city is aggregated to a single facility. Upgrade and maintenance of the infrastructure can be costly as the demands of smart video surveillance system scales up with the city growth.

Cloud computing is a new, popular approach in recent years to provide virtually unlimited storage and processing capacity at low cost that can be virtualized and leased to users on demand [15]. The definition of Cloud Computing, as provided by the National Institution of Standards and Technologies (NIST) [16], is “model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” Processing intensive components, such as video management and processing services, can benefit from practically unlimited cloud-based resources to compensate for constraints in processing power, storage, and scalability. As such, many cloud providers such as Amazon, IBM, and Microsoft are racing to adopt their Cloud Computing platforms with numerous IoT development frameworks and services [17–19]. With these observations, Cloud computing can be the key enabling technology that provide the necessary resources for smart cities to deploy large-scale smart video surveillance systems.

#### 2.1.4 Application Layer

The application layer is where useful information is extracted from the massive volume of video data collected. Video data has a wide application domain that spans across many different areas. In fact, the same video footage can produce varying information according to analytic functions and context. For example, urban management applications look for vehicle patterns and optimize traffic flows

through the streets [20]. Security applications, on the other hand, identify suspicious objects and people to control access to reserved areas and monitor for crisis detection and warning [21, 22]. Data obtained from video surveillance system can be incorporated with information from other cameras and sensors to portray a more accurate representation of the situation, allowing authorities to evaluate the situation and respond in a timely manner.

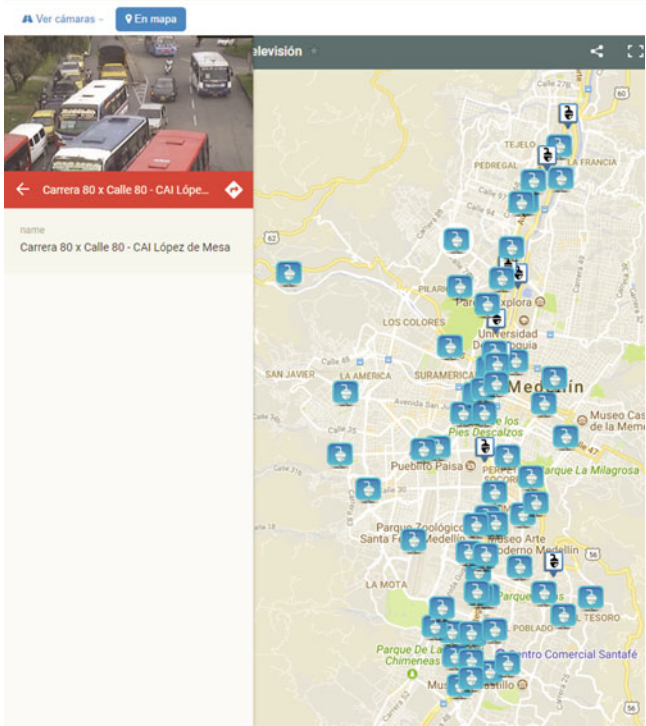
Computer vision is the driving technology behind numerous applications of smart video surveillance system. Functions such as object detection, motion detection, and optical character reader, are common features in many video analytics software. Deep learning algorithms have achieved more complex operations such as face recognition, object tracking, and optical character reading [23]. These capabilities play a crucial role in the success of smart video surveillance system as the huge volume of data needs to be automatically processed and analyzed. As more video data becomes available, visual analytic capabilities continue to progress, making smart cities smarter with a wider range of applications.

## ***2.2 Example Video Surveillance Applications Around the World***

Although video surveillance has been a staple security measure for decades to detect crimes and violations, the advance of video capturing and communications technologies have allowed video applications to extend to other domains, such as traffic analysis and environmental monitoring. This section highlights some implementations of video surveillance applications around the world to enhance city operations and improve quality of life for the metropolitans.

**Irving, Texas** The transportation department in Irving, Texas installed 70 pan-tilt-zoom (PTZ) cameras to monitor and control 175 intersections in the city [8]. The target area is divided into cells, each applies star topology where cameras are deployed around a base station. The system employs an 802.16-based wireless backbone infrastructure [24] at licensed frequencies of 18–23 GHz to provide bandwidth up to 100 Mbps. Furthermore, each individual cell can support 20–60 Mbps on-site using directional antenna with line-of-sight to the base station. The cameras transmit raw video data to a central control center, providing real-time viewing of congestion problems. The system enables immediately notifications to operators of signal problems and allows for dynamic re-timing of signals to account for special events or significant traffic accidents.

**Beijing, China** Beijing deployed a smart traffic management system that include 157 high-definition cameras on Beijing's surrounding expressways [25]. In addition to monitoring for accidents and alerting authorities, the system also augments its video data with other traffic flow sensors to automatically collect traffic statistics such as flow, speed, and density data. The analytics information is used to adjust



**Fig. 1** Medellin CCTV camera network. (Medellin CCTV camera network – <https://www.medellin.gov.co/simm/mapas/index.html?map=camarasCctv>)

traffic signals and optimize traffic flow on the road, which has reduced Beijing’s traffic congestion by up to 60% and doubled its road capacity [26].

**Medellin, Colombia** As shown in Fig. 1, the Integrated Emergency and Security System in Medellin, Colombia uses data from 823 surveillance cameras to assist numerous different government agencies such as police, medical, and fire departments [27]. Each camera covers a 120-meter radius area, providing real-time data for the authorities to verify citizen reports and coordinate emergency response in an efficient manner. The cameras are connected to optical fiber and wireless backbone network to transmit high quality video footage to integrated control center, enabling better optimization of city resources and improve response time to emergency and security situations.

Overall, video surveillance system is becoming more popular in smart cities due to its versatility. The multipurpose video data can be utilized by many different applications for both real-time monitoring and historical analysis. It can also be augmented with data from other telemetry and environmental sensors to provide more information. To fully realize the effectiveness of video surveillance systems,

especially at a large scale like in smart cities, there are many challenges to be addressed.

### 2.3 Challenges of Smart Video Surveillance System

**Connectivity** One of the key challenges to deployment of smart video surveillance system is how to provide connectivity coverage for many cameras over a large scale. This includes issues such as minimizing cost of wiring and deployment, determining good vantage points, and providing sufficient bandwidth to support high throughput video data transmission. While wired installation can easily support the bandwidth requirement, it restricts possible installations to nearby fiber drops which may not offer good vantage points for surveillance. Wireless communication technologies offer more flexibility in installation and enables wider coverage, but the performance is prone to signal degradation and interference.

**Scalability** The video surveillance system creates a large amount of data, including raw video footage, processed information, alerts, etc. This enormous volume of information is exchanged frequently among video acquisition devices, management platforms, and analytics applications. In addition, many video analytics applications require high-definition, high quality video data for face recognition, person tracking, etc. How to manage data transportation, storage, and making it accessible in a secure manner are major challenges for smart city administrators as well as smart video surveillance operators [28].

**Privacy** Privacy presents a significant barrier against implementation of smart video surveillance in smart cities [29]. Raw video data captured by surveillance cameras is often unfiltered, which makes it difficult to avoid recording unknowing citizens. In addition, analytical applications that rely on face recognition also cause security and trust concerns, as individuals cannot control how their information being collected. Indeed, it is nearly impossible to enforce privacy policy in data collection of video surveillance. This issue raises concerns about the privacy risks in relation to the use of collected data and its access, as well as the legal issue of identification and management of ownership over the data [30]. How to enable data owners to interact with data collecting and processing procedure for privacy assurance is also an open technical issue [29].

## 3 Video Surveillance Management Platform for Smart City

The management platform for video data and devices is an essential component to Smart City. While IP-enabled cameras often come with built-in management interface, managing a large number of devices across the city in this manner is



impractical. A centralized VMS can provide a more streamlined and consistent management process of IP cameras. This section explores several on-premises and cloud-based video management solutions and provide comparison on their performance on a small-scale testbed.

### **3.1 On-premises Video Management Solutions**

As video surveillance system is not a new concept, on-premises video management solutions already exist, ranging from free, open-source software to proprietary enterprise services that cost thousands of dollars. They generally include some fully developed data plane services, including data aggregation, live video display, and video data storage. Due to diversity in functionality, we selected 3 candidates at different price ranges and focus on the performance of their data plane services to support video surveillance system, which can provide a glimpse into how the system can scale to full-scale system for Smart City.

**ZoneMinder** [31] is a free, open-source VMS developed on Linux platform by a community of volunteer developers. It provides a web-based user interface for user to add new cameras, setting of recording parameters, and view live or archived camera footages. Using FFmpeg [32] (an open-source video framework) as the underline mechanism, ZoneMinder decodes an input video stream into MJPEG regardless of source video encoding format and the resulting video frames are stored as sequences of JPEG images, which can then be displayed on integrated live monitor interfaces or transcoded to H.264 format and stored as an MP4 or AVI file. In addition, ZoneMinder provides an internal motion detection mechanism by analyzing the stored JPEG image sequences.

**Xeoma** [33] is a small-scale commercial video surveillance software developed by Felena Soft. It provides functionalities such as recording, live monitoring, and motion detection as modules that can be combined in different arrangements as required by user. Xeoma also uses FFmpeg to retrieve and decode video streams from cameras, though it can archive data as transmitted by camera without transcoding. Thus, it does not support remote modification to camera configuration. Xeoma supports automatic filter to purge old recordings when storage capacity limit is reached.

**XProtect** [34] is an enterprise commercial solution developed by Milestone Systems. It is highly integrated with camera firmware, allowing remote adjustment of camera settings from an internal interface. XProtect is well optimized, allowing functions such as motion detection to be hardware accelerated for faster processing. Video stream is decoded by a proprietary mechanism and stored in the proprietary PIC format, which can only be accessed and transcoded to other video formats (e.g., MKV, AVI) by an associated proprietary media software.

One of the main limitations of an on-premises video management platform is scalability of the physical hardware. Components such as computing and storage servers are purchased and deployed to enable the VMS solutions, which can be

difficult to expand and upgrade. When a new device is added, the system may have to be taken down to properly integrate the new component. Thus, it can become costly to set up, maintain, and scale the back-end infrastructure to support growing demands of smart video surveillance systems.

### 3.2 Cloud-based Video Management Services

A key difference in the development of video management platform in recent years is the integration with Cloud Computing services. Cloud Computing addresses many issues encountered in on-premises management systems such limited resources and difficulty in hardware expansion by catering virtual computing and storage resources, which are easier to deploy and integrate for Smart City operators. Microsoft Azure [35] is a cloud computing platform for building, testing, deploying, and managing applications and services through a global network of Microsoft-managed data centers. It provides all three IaaS, PaaS, and SaaS service models and supports many different programming languages, tools, and frameworks to collaborate with both Microsoft and third-party software and systems. In this section, we use Microsoft Azure as an example cloud computing platform to illustrate the integration of cloud computing architecture and video management platform through its included multimedia management platform Azure Media Service and a third-party commercial SaaS solution Stratocast.

#### 3.2.1 Azure Media Services

Azure Media Services (AMS) [36] is an extensible cloud-based platform that enables developers to build scalable media management and delivery applications. AMS enables content providers to securely upload, store, encode, and deliver video and audio content in both on-demand and live streaming scenarios to various user devices (e.g., TV, PC, smart phones, tablets, etc.)

Fig. 2 depicts the overall data flow of Azure Media Services. Multimedia content can be uploaded to AMS through two data paths: *on-demand streaming* and *live streaming*. On-demand streaming allows video data stored on cloud to be streamed to users as requested at any time. Video files can be directly uploaded to the cloud storage through AMS web portal or REST APIs, which can then be encoded to different formats and bitrates as required by content provider. In addition, Azure media analytics can be applied on the video data in storage for face detection, motion detection, optical character recognition, etc. Analytical results are available for users to download directly as JSON file format, while video content is distributed through a streaming endpoint, which generates an URL for content retrieval.

On the other hand, live streaming allows video content to be delivered live to users through AMS in a streaming program. Each streaming program is contained in a *channel*, a pipeline for processing live-streaming content before delivering to

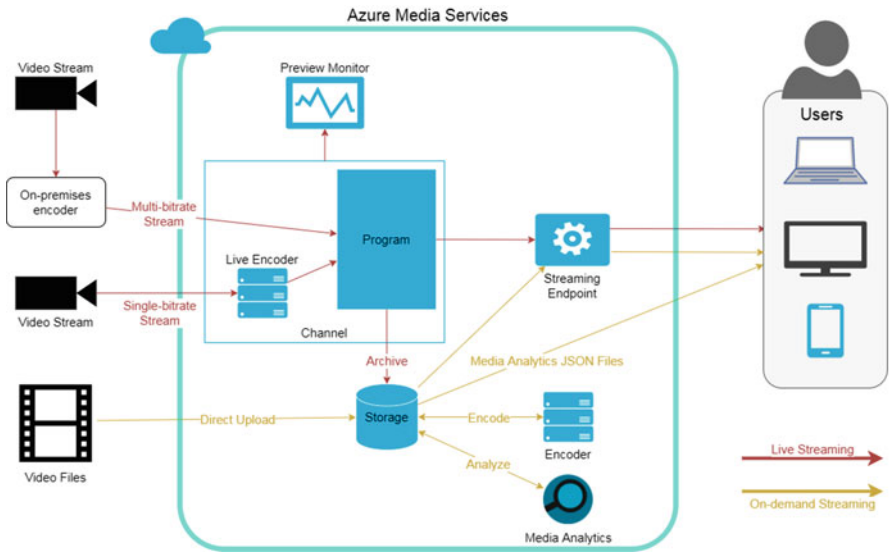


Fig. 2 Video streaming data flow in Azure media services [36]

users. Video data can be encoded using on-premises encoding software such as FFmpeg or cloud encoder provided by AMS at an additional charge. If encoded using on-premises software, the incoming stream passes through AMS to the stream endpoint and is distributed to users without any encoding. Cloud encoder allows a single-bitrate incoming stream to be encoded to different bitrates and formats to suit user requirements. Live streaming content is archived after a program is finished, allowing users to access streamed data through on-demand streaming at a later time.

However, as AMS is designed to be multimedia content delivery service, there are limitations to AMS when applying to Smart City applications such as video surveillance. Each streaming program can only run continuously for up to 25 h, which means it is difficult to provide continuous video monitoring through AMS live streaming services. On-demand streaming is only compatible with complete video files, thus continuous video streams have to be processed elsewhere into individual files before being uploaded to the cloud. In addition, live streaming is limited by AMS quotas per Azure subscription account, which reduces the scalability of this Azure platform. In particular, only 5 channels can be live simultaneously and only 1 input video stream per channel, which effectively limits support to 5 cameras streaming at any given time. As illustrated above, at its current development stage, AMS is not yet suitable for large-scale scenarios with many input video streams such as Smart City.

### 3.2.2 Stratocast

Stratocast [37] is a VMS application developed by Genetec following a Video-Surveillance-as-a-Service (VSaaS) model. Stratocast provides the key management functionalities of video management system such as video data storage, data format transformation, and data display on a cloud infrastructure. In line with the cloud computing paradigm, Stratocast enables VMS capabilities at lower cost by eliminating the need for on-premises servers and storage, reducing traditional hardware costs and installation time. To achieve this, Stratocast utilizes Microsoft Azure IaaS to support their proprietary video management and monitoring platform. In fact, the back-end management system of Stratocast employs a customized version of Security Center [38], a proprietary video monitoring and management solution, running on many virtual machines hosted on Microsoft Azure infrastructure. Live video are streamed directly to the cloud and stored in secure Azure distributed data centers, taking advantage of Azure reliable and scalable resources to ensure data protection. Being on the cloud also allows Stratocast to provide a centralized management solution accessible anywhere and anytime through the Internet from laptops, smart phones, or tablets. Stratocast services are offered in 3 separate packages, each with different video qualities (e.g., resolution, FPS) based on demand of clients. Additionally, Stratocast also offers edge recording where video recordings are stored directly on local storage devices such as NAS or SD cards, to reduce bandwidth consumption.

As shown in Fig. 3 Stratocast communicates with IP-cameras via a cloud gateway service developed by IP-camera manufacturers like Axis and Vivotek to register, monitor, and manage IP-cameras. Currently Stratocast is integrated with Axis Video Hosting System (AVHS) [39] and Vivotek Application Development Platform (VADP) [40], two cloud-based data aggregation services that enable management and monitor of Axis and Vivotek IP-cameras respectively over the Internet. However, as these are proprietary services, the main limitation of AVHS and VADP is that they are only compatible with IP cameras and recording devices from their respective vendors. Thus, Stratocast currently can only work with certain Axis and Vivotek devices, though development is ongoing to expand the list of compatible devices.

A key issue of Stratocast is that its performance depends on the service plan user is subscribed to. Stratocast standard plan only supports resolution up to  $1280 \times 720$ , 10 frames per seconds, and 600 kbps bit rate. At the highest tier service, the premium plan, Stratocast only supports up to  $1920 \times 1080$ , 15 frames per second, and 1.2 Mbps bit rate. As the framerate and especially the bandwidth restrictions are much lower than the capability of most modern smart cameras, aggressive compression must be done and hence, the quality of the collected video data is greatly reduced. This decrease in video quality may hinder some analytics applications that rely on analyzing finer image details such as license plate, facial and object recognition.

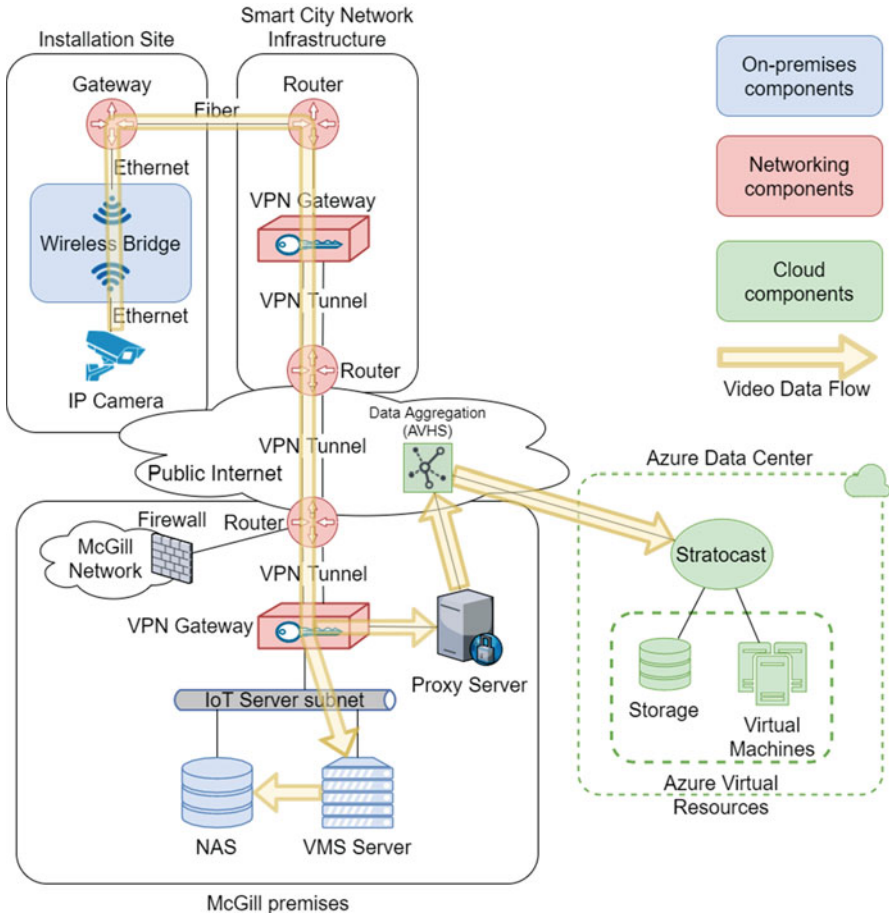


Fig. 3 Experimental testbed setup

### 3.3 Experimental Testbed

In order to compare the performance of different VMS, we implemented a small-scale testbed with 6 high-definition, IP-enabled PTZ cameras (2 Axis, 3 Hikvision and 1 Panasonic cameras) deployed at remote target locations. The cameras are connected via point-to-point wireless bridges to the fiber-optic network, from which data is transmitted through a VPN tunnel to our video management platform at McGill University. The wireless bridges are formed by pairs of Ubiquiti Nanobeam directional radios [41] that operate in the 5.8 GHz frequency and can support up to 150 Mbps.

### 3.3.1 Video Management Platform Setup

Fig. 3 shows our test video management platform with both on-premises and cloud-based systems. The on-premises test video management platform consists of a Dell Precision T7600 server hosting virtual machines that run the VMS solutions. Each virtual machine is allocated 8 Intel Xeon E5-2620 2.00-GHz CPU cores, 6 GB of RAM, and 1000 Mbps Ethernet connection. Video data archived by the VMS is stored in a network-attached storage (NAS) drive. Synology DiskStation DS212j NAS is used and is configured with a redundant array of independent disks (RAID-1) setup. The NAS is connected to VMS server using network file system (NFS) protocol to allow seamless read/write operations to the storage device. The NAS is managed by an internal proprietary operating system called DiskStation Manager, which supports user authentication, IP address filtering, and NFS permission controls as security measures.

For cloud-based test setup, only 1 Axis IP-camera that is compatible with AVHS service can be used. Registration and communication between the IP-cameras and AVHS is done by a built-in function integrated in the camera firmware, which transmits an HTTP or HTTPS request to AVHS server to establish a TCP connection. A proxy is set up to allow outbound traffic from the private network to the public Internet, enabling communication between the camera and AVHS server.

A simple video management program based on FFmpeg was developed to provide a baseline for comparison. The program, simply referred to as FFmpeg, can only record H.264 video streams from a camera and archive them as MP4 video files without transcoding. It does not support any live monitoring, camera configuration, or analytical functionality.

### 3.3.2 Performance Comparison

To investigate the capability of each VMS, 2 experiments were performed. First, the performance of each solution is evaluated when managing one video stream at different video resolutions to establish a benchmark for processing and archiving video data. Second, how their performance scales when managing multiple streaming cameras was examined.

#### Single Video Stream Performance

In this experiment, we examine each solution for their consumption of CPU, RAM, and bandwidth resources when managing one single video stream. Each on-premises VMS solution is set to continuously record 10-min video segments of a test video stream, at 3 different pixel resolutions:  $1280 \times 720$ ,  $1920 \times 1080$ , and  $3840 \times 2160$  for CPU and RAM usage measurement. The bandwidth usage for Stratocast can only be measured up to its highest supported resolution  $1920 \times 1080$ . The frame rate and bit rate of the test video stream are kept constant at 24 FPS and 16 Mbps, respectively.

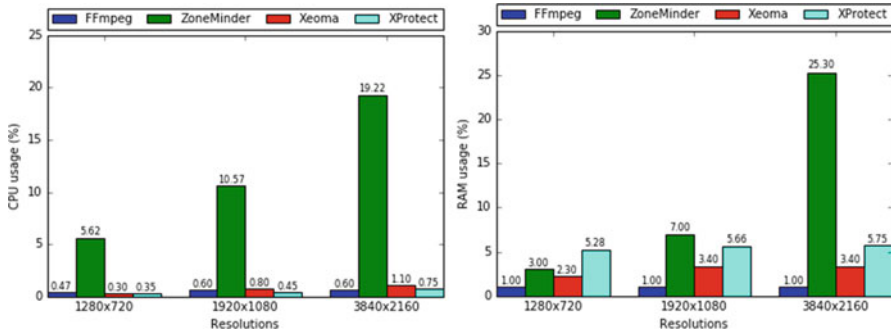


Fig. 4 CPU and RAM usage of on-premises solutions

Fig. 4 shows that there is not a big difference in CPU consumption between FFmpeg, Xeoma, and XProtect. Most notably, ZoneMinder consumes significantly more CPU and memory resources than any other candidates with increasing image resolutions. The higher resource consumption can be attributed to the video decoding process to MJPEG format implemented in ZoneMinder. This decoding process divides a video stream into JPEG still frames that are then compressed individually. As a result, MJPEG does not take advantage of redundant data contained in consecutive video frames to process more efficiently.

This inefficiency is also reflected in the bandwidth consumption of ZoneMinder in Fig. 5. Despite being based on FFmpeg, ZoneMinder only has similar incoming throughput at  $1280 \times 720$  resolution, and then lags behind due to the longer processing time required to process image frames at higher resolutions. On the other hand, FFmpeg, Xeoma, and XProtect maintain their incoming bandwidth consumptions at close to the maximum camera output bit rate of 16 Mbps.

Stratocast, on the other hand, does not behave in a similar manner as the on-premises solutions. It can be observed that while bandwidth usages of other solutions increase as resolution increases due to more data to be transmitted, bandwidth usage of Stratocast for incoming traffic is throttled at around 1.30 Mbps. As noted before, this arbitrary constraint is intended to restrict the bandwidth consumption of any camera streams so that each Azure data center hosting Stratocast can support more clients and cameras. As a result, data has to be further compressed to meet the bandwidth requirement by Stratocast, leading in poorer video quality.

Bandwidth is not the only constraint that Stratocast imposes on its managed video data. While ZoneMinder is inhibited by poorly optimized implementation, Stratocast frame rate is restricted by its service provider at 15 FPS at highest setting. The reduced bandwidth consumption, and a lower frame rate, result in lower storage requirement for Stratocast recordings. As shown in Fig. 6, the storage size of a 10-min recording for Stratocast is only 67.8 MB at  $1280 \times 720$  resolution and 102.0 MB at  $1920 \times 1080$ , significantly lower than the recording sizes for other on-premises VMS solutions at the same resolution.

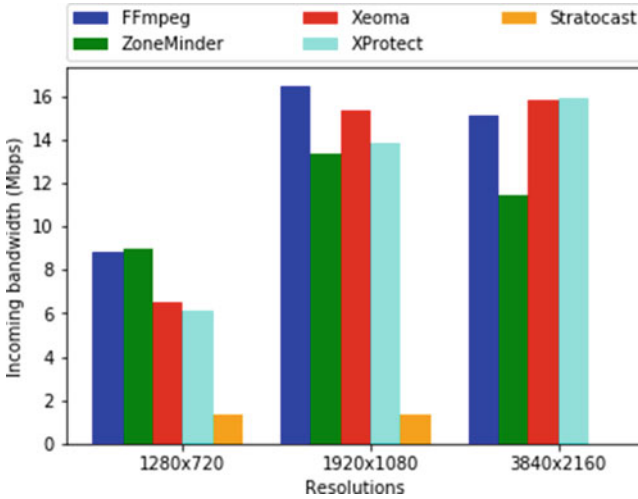


Fig. 5 Bandwidth usage of on-premises and cloud-based solutions

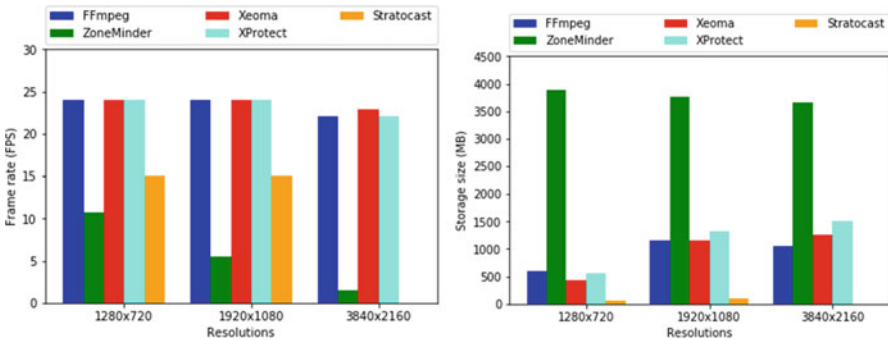


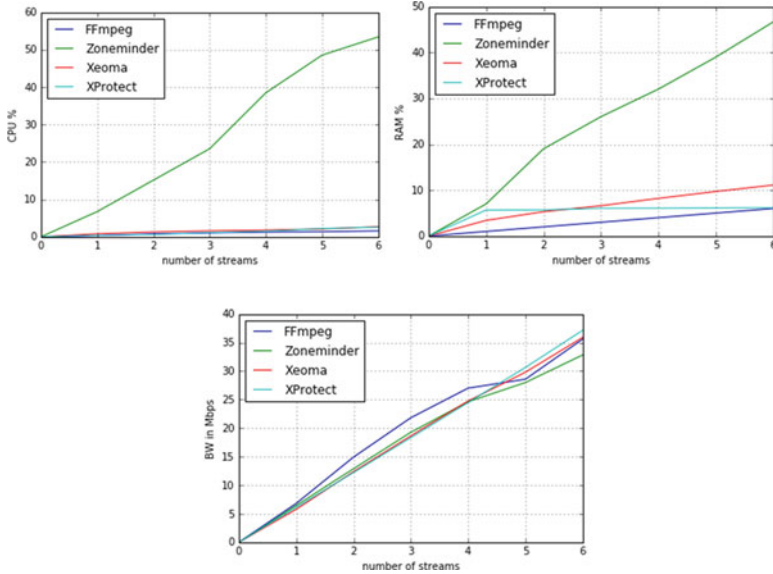
Fig. 6 Frame rate and storage size

### Multiple Video Stream Performance

To evaluate the scalability for Smart City, each VMS candidate solution manages up to 6 video streams from the 6 cameras deployed in the data acquisition layer. The result for Stratocast is not available as only one of our test cameras is compatible with the service. The video streams are fixed at 1920 × 1080 resolution and 24 FPS frame rate as this is the highest configuration supported by all of our cameras. The bit rates are limited to 6 Mbps to avoid over-congesting the VPN connection without sacrificing too much video quality. The key performance indicators are CPU, RAM, and network bandwidth usage of each VMS solution.

Fig. 7 shows the CPU and RAM consumption of each VMS candidate solution with the number of video streams. The CPU usages of all 4 solutions scale almost linearly with the number of video streams. Similar to the previous experiment,





**Fig. 7** CPU, RAM, and bandwidth usage of on-premises solutions with increasing number of video streams

there are no notable differences among FFmpeg, Xeoma, and XProtect. Likewise, ZoneMinder’s rate of CPU consumption is much higher than others due to its inefficient video decoding process. On the other hand, XProtect has higher initial memory usage than FFmpeg and Xeoma, but it increases by a negligible amount for each additional video stream. This behavior is analogous to the result shown in single stream experiment, where XProtect memory usage is stable as the number of streams increases. The results may imply that the video processing mechanism of XProtect is much more efficient with memory management, though 6 cameras is insufficient to definitively evaluate the scalability of XProtect. In contrast, FFmpeg, ZoneMinder, and Xeoma’s memory usages scale linearly with the number of video streams. The usage of FFmpeg is equal to XProtect at 6 streams while Xeoma bypasses XProtect after 3 streams. ZoneMinder, once again, consumes the most memory with almost 50% RAM usage at 6 video streams.

Overall, in terms of on-premises approach, ZoneMinder is the most inferior solution in terms of functionality. The most critical flaw of ZoneMinder is the use of MJPEG decoding, which consumes huge amount of resources to both processing video data and archiving them in JPEG images. This approach results in a large storage size while delivering very low frame rate. Meanwhile, Xeoma offers better performance in video processing and storage than ZoneMinder, but its service cost is calculated based on the number of cameras, which can become very expensive to deploy in Smart City. Finally, XProtect has the best performance out of all the candidate solutions, with much more efficient video processing and camera control mechanism than the rest. Its main limitation is its proprietary algorithms

and video archive formats, making it difficult to interoperate with other Smart City applications and services. Furthermore, its high cost may be a major financial burden to deploy on a full-scale Smart City.

As for cloud-based approach, although advertised as providing virtually unlimited resources, cloud-based solutions are restricted by manufacturer's arbitrary performance constraints that are not presented in on-premises solutions. Stratocast's service plan limitations result in lower video quality than our on-premise setup. While these limits are intended to assure quality of the service, they also place a virtual cap on the scalability of the video surveillance system and the integration with video analytics processes.

## 4 Deployment Challenges of Smart Video Surveillance

Management platform is not the only concern of a video surveillance system in Smart City. Deployment of video capturing and networking devices is also crucial to ensure that the collected video data is meaningful. There are many deployment scenarios when considering a wireless video surveillance application for a designated area, depending on numerous factors: the type of application and data to be monitored, camera deployment patterns, location of gateways, wireless technologies, etc. In this section, we focus on intersection monitoring where cameras are placed at numerous intersections and junctions in a target area to record their vehicular and pedestrian traffic activities. The objective is to demonstrate a number of camera deployment scenarios in selected Montreal areas and provide an estimation of the number of cameras and wireless devices necessary for a full-scale deployment.

### 4.1 Deployment Scenarios

The two main camera deployment patterns that we want to investigate are *high-density* deployment pattern and *low-density* deployment pattern. In a high-density deployment pattern, a camera is installed at almost every intersection in the target area to provide a nearly complete surveillance coverage of ongoing traffic, excluding some small crossroads where traffic activities are too infrequent to justify the cost. If two intersections are close enough to each other that one camera can reasonably monitor both, then a second camera does not have to be installed. In a low-density deployment pattern, only intersections with traffic lights are monitored by cameras, reducing costs of installation and maintenance while still providing surveillance at major locations in the target area.

Due to a large number of possible variations of high-density and low-density deployment patterns, scale of target areas, and distribution of gateways, we only apply one deployment scenario to each area of interest to provide a rough estimate

for as many scenarios as possible. First, in Scenario 1, the target surveillance area is approximately 0.36 km<sup>2</sup> in Quartier des Spectacles, specifically the two areas around Place-des-Arts and Berri-UQAM Metro stations, where most public activities happen. Cameras are installed on a street light pole at every intersection following the high-density pattern, excluding small alleys and intersections already in field of vision of other cameras. The wireless base APs are installed at gateway locations scattered throughout the area and are connected directly to the optical fiber network. With the small target area and the low number of cameras, wireless coverage is sufficiently provided by base APs at gateway locations that relay APs are not necessary. 802.11n-based wireless radios and APs are used to establish wireless network coverage in this scenario.

Next, we consider the larger commercial district area surrounding Quartier des Spectacles with a similar urban planning attributes (e.g., building types, traffic intersection density, etc.). Scenario 2 covers the north-eastern 1.6 km<sup>2</sup> half of the commercial district. Cameras are distributed following the high-density pattern, where a camera is installed at almost every intersection. In this scenario, base APs are installed at gateway locations, which are assumed to be deployed along the northern edge of the target area on Rue Amherst. With most camera installations being outside the range of the base APs, relay APs are distributed throughout the region to provide coverage to all cameras. They are linked to base APs via point-to-point directional wireless bridges to minimize interference. The wireless network is established using 802.11 ac-based wireless radios and APs to contrast with the 802.11n deployment in Scenario 1.

Lastly, the low-density deployment in Scenario 3 spans across the southern 2.1 km<sup>2</sup> half of the commercial district area, where cameras are only installed at intersections with traffic light. Similar to Scenario 2, base APs are installed where the gateway locations are assumed to be along Rue de Bleury that goes through the center of the target area. The wide coverage area requires relay AP to be deployed as the coverage from the gateway is not adequate, but they are more sparsely distributed as there are less cameras to be covered. The wireless network for this scenario is also based on 802.11 ac-based wireless technologies and equipment as we look to contrast the deployment density in Scenario 2. Table 1 summarizes the description of all three deployment scenarios.

In our simulation, wireless connectivity is modeled using Ubiquiti Network’s proprietary, 802.11-based airMAX wireless technology and equipment. Unlike stan-

**Table 1** Simulation scenario summary

	Scenario 1	Scenario 2	Scenario 3
<b>Area</b>	0.36 km <sup>2</sup>	1.6 km <sup>2</sup>	2.1 km <sup>2</sup>
<b>Wireless protocol</b>	802.11n	802.11 ac	802.11 ac
<b>Deployment pattern</b>	High-density	High-density	Low-density
<b>Gateway distribution in area</b>	Scattered	Along the edge	Through the center
<b>Relay APs</b>	No	Yes	Yes

standard Wi-Fi protocols, airMAX utilizes TDMA-based protocol to provide significant improvement in latency and throughput [42]. Ubiquiti wireless APs support 802.11n and 802.11 ac-based version of airMAX to imitate the performance of traditional Wi-Fi devices. With each camera average transmission rate at 20 Mbps for streaming ultra-high-definition video data, each 802.11 n-based AP can support up to 5 cameras simultaneously while an 802.11 ac-based AP can support 13 cameras. We aim to minimize the quantity of APs needed to support deployment scenarios in a large-scale smart video surveillance system to reduce cost of equipment. The simulation is done using Atoll wireless planning software [43] with Erceg-Greenstein propagation model [44], which is similar to our target environment. The terrain model details are obtained from the US Geological Survey data source [45] which has a resolution of  $30 \times 30$  m and is compatible with Atoll.

## 4.2 Deployment Estimation

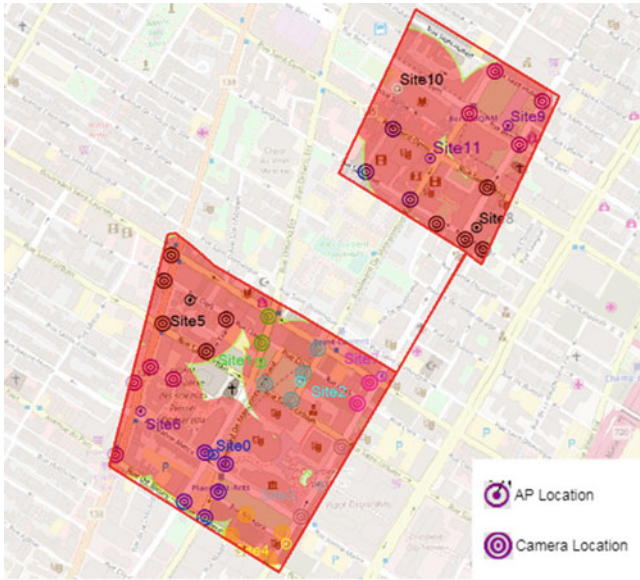
### 4.2.1 Simulation Results

From the simulated estimation results, summarized in Table 2, it can be shown that the camera deployment density of Scenario 2 is very similar to that of Scenario 1 with 13% margin. This result provides an approximate basis to estimate the number of APs needed to support a high-density deployment using 802.11 n and 802.11 ac-based implementations. In both scenarios, we are able to provide high throughput coverage of up to 150 Mbps to be shared among the cameras that connect to an AP, in the target surveillance areas, as seen in Fig. 8 and Fig. 9. The density of 802.11 n-based APs is much higher than 802.11 ac-based deployments (33 APs/km<sup>2</sup> vs 13 APs/km<sup>2</sup>, respectively), which is expected as 802.11 n supports significantly lower data rates and thus few number of cameras per AP than 802.11 ac. This difference in AP density nullifies any cost advantage that 802.11 n-based deployment has due to lower device costs, i.e., for a large-scale deployment, 802.11 ac is better both in terms of costs and device quantity.

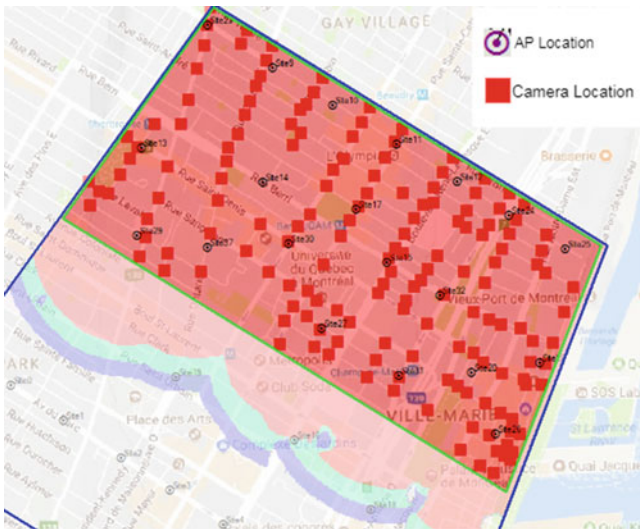
In the low-density deployment shown in Fig. 10, most cameras are covered by the base APs at gateway locations along the central street, thus fewer wireless bridges and relay APs are needed to extend coverage over the entire area. Relay APs are placed more sparsely throughout to maximize the number of stations that each AP

**Table 2** Simulation result summary

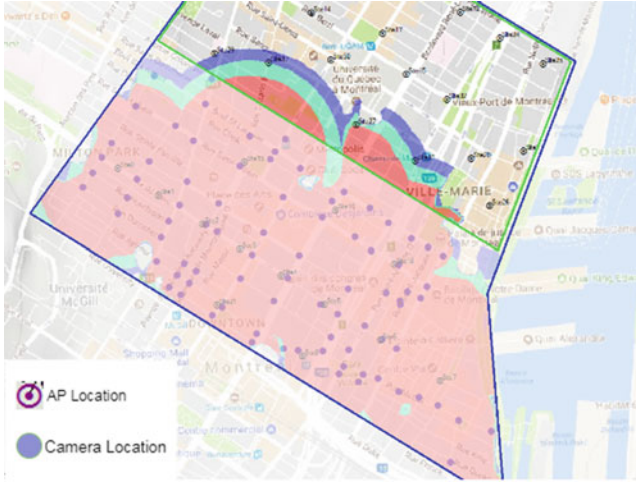
	Scenario 1	Scenario 2	Scenario 3
Number of cameras	41	161	85
Camera density	113 cameras/km <sup>2</sup>	100 cameras/km <sup>2</sup>	40 cameras/km <sup>2</sup>
Number of APs	12	21	14
AP density	33 APs/km <sup>2</sup>	13 APs/km <sup>2</sup>	7 APs/km <sup>2</sup>
Total required bandwidth	0.8 Gbps	3.2 Gbps	1.66 Gbps



**Fig. 8** Signal strength coverage of 802.11 n-based wireless networks in Scenario 1 (red areas indicate peak throughput  $\geq 150$  Mbps)



**Fig. 9** Signal strength coverage of high-density 802.11 ac-based wireless networks in Scenario 2 (red areas indicate peak throughput  $\geq 150$  Mbps)



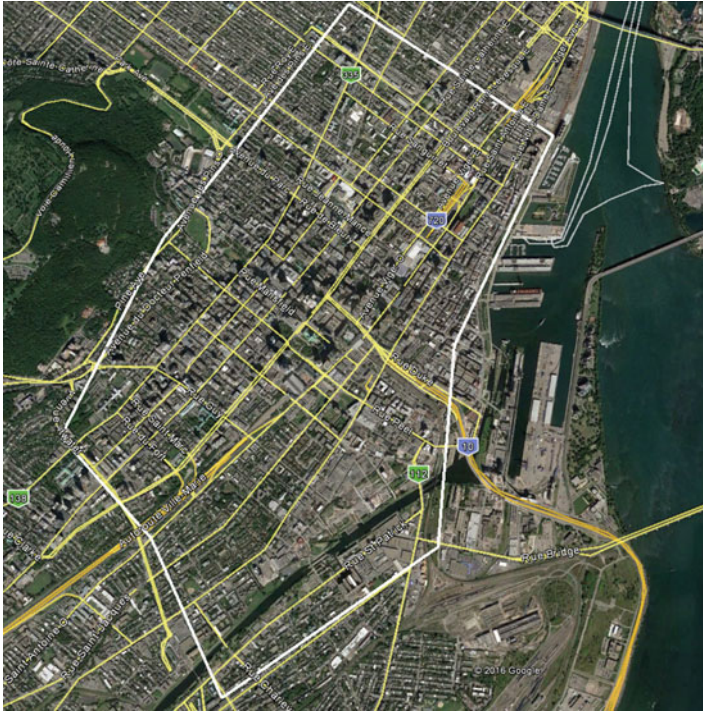
**Fig. 10** Signal strength coverage of low-density 802.11 ac-based wireless networks in Scenario 3 (red areas indicate peak throughput  $\geq 150$  Mbps)

supports, leaving a few cameras outside the maximum throughput coverage. All cameras can still achieve acceptable data rate to not warrant additional relay AP deployments. It should be noted that the uneven terrain, obtained from [45], and the narrow horizontal beam width of the omni-directional AP antennas result in irregular coverage patterns and occasional blind spots where wireless signal is weak.

In comparison to Scenario 2, the low-density deployment in Scenario 3 employs fewer cameras and APs, which scales down the throughput load on the network infrastructure and cost of deployment, making it a more viable solution for areas with limited financial and networking resources. Low-density deployment reduces bandwidth requirement by 60%, requiring only 40 cameras/km<sup>2</sup> as opposed to 100 cameras/km<sup>2</sup> in Scenario 2. Also, less APs are required to provide sufficient coverage and throughput support for the cameras, at approximately 7 APs/km<sup>2</sup>.

#### 4.2.2 Scale-up Estimation

Based on these preliminary results, we can make estimations on the number of cameras and APs, and bandwidth required to deploy over larger areas. For example, the Downtown Montreal and Quartier de l'Innovation region, which has about 9 km<sup>2</sup> in area (Fig. 11), has similar urban features and intersection distribution as the areas in our simulated scenarios. Table 3 summarizes the estimation results for the area. It should be evident that the number of devices needed to provide coverage of Downtown Montreal is very high. Even though 802.11 ac-based deployment requires less equipment overall, over 100 APs are necessary to provide connectivity for 900 cameras in high-density camera deployment, transmitting 18 Gbps of data.



**Fig. 11** Downtown and Quartier de l’Innovation area overview (outlined in white)

**Table 3** Scale-up estimation of Downtown Montreal and Quartier de l’Innovation area

Total Area $\approx 9 \text{ km}^2$	Scenario 1	Scenario 2	Scenario 3
Number of cameras	1017	900	360
Number of APs	297	117	63
Total bandwidth required	20 Gbps	18 Gbps	7.2 Gbps

Scaling up to the whole island of Montreal (almost  $506 \text{ km}^2$ ), it would require roughly 50,000 cameras, 6000 APs, and 1 Tbps data throughput. The volume of data produced by a smart video surveillance system is enormous. For the context, most fiber-optic networks can only support up to 10 Gbps, and are often congested with other types of Internet traffic.

While these preliminary estimates are very optimistic and specific to a target area, the huge throughput requirements raise non-trivial questions about the quantity of resources can Smart City support in a given area and how to distribute networking resources to numerous devices and applications. One thing is for certain: a centralized network infrastructure and data management system will not be able to scale with the demand of Smart City. Thus, designing a scalable distributed infrastructure

is crucial to enable a full-scale Smart City deployment for demanding applications such as smart video surveillance system.

## 5 Conclusion

Smart video surveillance system is becoming an integral part of smart city with its vast amount of data and versatile capability that enables a plethora of applications. The massive volume of data requires a platform that aggregates, formats, and archives data to be made accessible to numerous analytics applications to extract information. As such, video management platform is an essential component in the implementation of a smart video surveillance system.

In this chapter, two approaches of on-premises and cloud-based for implementing a video management platform for smart video surveillance are discussed. We experimented with a variety of on-premises and cloud-based video management solutions to compare their performance with a small-scale testbed and a small number of cameras. In comparison to the on-premises solutions, which offer more functionality and control over the management platform, cloud-based solutions include constraints that limit the performance of our tested software. We also demonstrated the challenges of deploying smart video surveillance through a series of simulated deployment scenarios. The results, while preliminary, show that an enormous number of devices and large bandwidth are required to provide citywide coverage, so the infrastructure of smart surveillance system and smart city as a whole must be carefully designed to support the massive volume of data to be collected.

Future contributions regarding video surveillance system will focus on efficient video analytic functions, at both the edge and on the cloud, to supplement the infrastructure with smart analytics on both offline and real-time data.

## References

1. N. Chen, Y. Chen, X. Ye, H. Ling, S. Song, and C.-T. Huang, "Smart City Surveillance in Fog Computing," *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, Springer, pp. 203–226, 2017.
2. S. Dube, K. J. Ghee, W. W. Onn, and Q. Z. Han, "Embedded user interface for smart camera," in *2017 7th IEEE International Conference on System Engineering and Technology (ICSET)*, pp. 32–37, 2017.
3. Bosch makes video analytics at the edge a new built-in standard in all their IP cameras [Online]. Available: [https://us.boschsecurity.com/en/05\\_news\\_and\\_extras\\_2/05\\_04\\_press\\_releases\\_2/2016\\_1/bosch\\_makes\\_video\\_analytics\\_at\\_the\\_edge\\_a\\_new\\_built\\_in\\_standard\\_in\\_all\\_their\\_ip\\_cameras/bosch\\_makes\\_video\\_analytics\\_at\\_the\\_edge\\_a\\_new\\_built\\_in\\_standard\\_in\\_all\\_their\\_ip\\_cameras](https://us.boschsecurity.com/en/05_news_and_extras_2/05_04_press_releases_2/2016_1/bosch_makes_video_analytics_at_the_edge_a_new_built_in_standard_in_all_their_ip_cameras/bosch_makes_video_analytics_at_the_edge_a_new_built_in_standard_in_all_their_ip_cameras)
4. S. N. Sinha, "Pan-Tilt-Zoom (PTZ) Camera", *Computer Vision*, Springer Press, 2016.



5. T. Marques, L. Lukic, J. Gaspar, "Observation Functions in an Information Theoretic Approach for Scheduling Pan-Tilt-Zoom Cameras in Multi-target Tracking Applications", *Second Iberian Robotics Conference on Robot*, 2015.
6. T. Zhang, A. Chowdhery, A. V. Bahl, K. Jamieson, S. Banerjee, "The Design and Implementation of a Wireless Video Surveillance System", *International Conference on Mobile Computing and Networking*, 2015.
7. J. Fernandez, L. Calavia, C. Baladron, J. M. Aguiar, B. Carro, A. S-Esguevillas, J. A. A-Lopez, Z. Smilansky, "An Intelligent Surveillance Platform for Large Metropolitan Areas with Dense Sensor Deployment", *Sensor*, 2013.
8. S. Leader. (2004). "Telecommunications handbook for transportation professionals—The basics of telecommunications," Federal Highway Administration, Washington, DC, USA, Tech. Rep. FHWA-HOP-04-034 [Online]. Available: [http://ops.fhwa.dot.gov/publications/telecomm\\_handbook/telecomm\\_handbook.pdf](http://ops.fhwa.dot.gov/publications/telecomm_handbook/telecomm_handbook.pdf)
9. A. Kawamura, Y. Yoshimitsu, K. Kajitani, T. Naito, K. Fujimura, and S. Kamijo, "Smart camera network system for use in railway stations", *International Conference in Systems, Man, and Cybernetics*, pp. 85–90, 2011.
10. N. Luo, "A wireless traffic surveillance system using video analytics," M.S. thesis, Dept. Comput. Sci. Eng., Univ. North Texas, Denton, TX, USA, 2011.
11. IEEE 802.11 working group - <http://www.ieee802.org/11/>
12. H. Wang, L. Dong, W. Wei, W-S. Zhao, K. Xu, G. Wang, "The WSN Monitoring System for Large Outdoor Advertising Boards Based on ZigBee and MEMS Sensor", *IEEE Sensors Journal*, Vol. 18 (3), pp.1314-1323, 2018.
13. O. S. Alwan; K. P. Rao, "Dedicated real-time monitoring system for health care using ZigBee", *IEEE Healthcare Technology Letters*, Vol. 4 (4), pp. 142-144, 2017.
14. Y. Ye, S. Ci, A. K. Katsaggelos, Y. Liu, and Y. Qian, "Wireless video surveillance: A survey," *IEEE Access*, vol. 1, pp. 646–660, 2013.
15. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey," *Future Generation Computer System*, vol. 56, pp. 684–700, Mar. 2016.
16. P. M. Mell and T. Grance, "The NIST definition of cloud computing," Gaithersburg, MD, 2011.
17. Amazon Web Services. - <https://aws.amazon.com/>
18. IBM Bluemix. - <https://www.ibm.com/cloud-computing/bluemix/>
19. Microsoft Azure. - <https://azure.microsoft.com/>
20. T. Zhang, S. Liu, C. Xu, H. Lu, "Mining Semantic Context Information for Intelligent Video Surveillance of Traffic Scenes", *IEEE Transaction on Industrial Informatics*, Vol. 9 (1), pp. 149-160, 2013.
21. T. J. Narendra Rao, G N. Girish, Mohit P. Tahiliani, Jeny Rajan, "Anomalous Event Detection Methodologies for Surveillance Application: An Insight", *Handbook of Research on Advanced Concepts in Real-Time Image and Video Processing*, 2018.
22. H-L. Eng, K-A. Toh, W-Y. Yau, J. Wang, "DEWS: A Live Visual Surveillance System for Early Drowning Detection at Pool", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18 (2), pp. 196-210, 2008.
23. L. Wang and D. Sng, "Deep Learning Algorithms with Applications to Video Analytics for A Smart City: A Survey," *CoRR*, pp. 1–8, Dec. 2015.
24. Z-T. Chou, Y-H. Lin, "Energy-Efficient Scalable Video Multicasting for Overlapping Groups in a Mobile WiMAX Network", *IEEE Transactions on Vehicular Technology*, Vol. 65 (8), pp.6403-6416, 2016.
25. Y. L. Tian, L. Brown, A. Hampapur, M. Lu, A. Senior, and C. F. Shu, "IBM smart surveillance system (S3): Event based video surveillance system with an open and extensible framework," *Mach. Vis. Appl.*, vol. 19, no. 5–6, pp. 315–327, 2008.
26. P. Liu and Z. Peng, "China's Smart City Pilots: A Progress Report", *Computer*, Vol. 47 (10), pp. 72–81, 2014.
27. D. Amar Flórez, "International Case Studies of Smart Cities: Medellin, Colombia," Washington, D.C., Jun. 2016.

28. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
29. R. H. Weber, "Internet of things: Privacy issues revisited," *Computer Law & Security Review*, vol. 31, no. 5, pp. 618–627, Oct. 2015.
30. R. C. Staudemeyer, H. C. Pöhls, and B. W. Watson, "Security and Privacy for the Internet of Things Communication in the SmartCity," in *Designing, Developing, and Facilitating Smart Cities*, Cham: Springer International Publishing, 2017, pp. 109–137.
31. ZoneMinder. - <https://zoneminder.com/>
32. FFmpeg. - <https://www.ffmpeg.org/>
33. Xeoma. - <http://felenasoft.com/xeoma/en/>
34. XProtect. - <https://www.milestonesys.com/solutions/platform/video-management-software/>
35. Microsoft Azure. url: <https://azure.microsoft.com/en-us/>
36. Azure Media Services. - <https://azure.microsoft.com/en-us/services/media-services/>
37. Stratocast. - <http://www.stratocast.com/>
38. Genetec Security Center. - <https://www.genetec.com/solutions/all-products/security-center>
39. Axis Video Hosting System- <https://www.axis.com/ca/en/products/hosted-video>
40. Vivotek Application Development Platform - <https://www.vivotek.com/website/vadp-partner/>
41. Ubiquiti NanoBeam - <https://www.ubnt.com/airmax/nanobeamm/>
42. Ubiquiti airMAX. - <https://airmax.ubnt.com/>
43. Atoll. - <http://www.forsk.com/atoll-overview>
44. V. Erceg, *et al.*, "An empirically based path loss model for wireless channels in suburban environments," *IEEE Journal on Selected Areas in Communications*, Vol. 17 (7), pp. 1205–1211, Jul. 1999.
45. U.S. Geological Survey - <https://earthexplorer.usgs.gov/>

# Intelligent Transportation Systems Enabled ICT Framework for Electric Vehicle Charging in Smart City



Yue Cao, Naveed Ahmad, Omprakash Kaiwartya, Ghanim Puturs,  
and Muhammad Khalid

**Abstract** In the future, Electric Vehicles (EVs) are expected to be widely adopted as personal, commercial, and public fleets in modern cities. The popularity of EVs will have a significant impact on the sustainable and economic development of urban city. However, compared to traditional fossil fuel vehicles, EVs have limited range and inevitably necessitate regular recharging. Thus, the provisioning of assured service quality is necessary for realizing E-mobility solution using EVs.

The design of an efficient charging management system for EVs has become an emerging research problem in future connected vehicles applications, given their mobility uncertainties. Major technical challenges involve decision-making intelligence for the selection of Charging Stations (CSs), as well as the corresponding communication infrastructure for information dissemination between the power grid and mobile EVs. This chapter introduces a number of information enabling technologies that been applied for EV charging, viewed from a transportation planning angle.

**Keywords** Electric Vehicle · Transportation Planning · Charging Management · Wireless Communication · Mobile Edge Computing · Vehicle-to-Vehicle · Mobility

---

Y. Cao (✉) · O. Kaiwartya · G. Puturs · M. Khalid  
Northumbria University, Tyne, UK  
e-mail: [yue.cao@northumbria.ac.uk](mailto:yue.cao@northumbria.ac.uk)

N. Ahmad  
University of Peshawar, Peshawar, Pakistan

© Springer Nature Switzerland AG 2018  
M. Maheswaran, E. Badidi (eds.), *Handbook of Smart Cities*,  
[https://doi.org/10.1007/978-3-319-97271-8\\_12](https://doi.org/10.1007/978-3-319-97271-8_12)

## 1 Introduction

The awareness concerning air pollution from CO<sub>2</sub> emissions has increased in recent years, and the realization of a more environment-friendly transportation system is now a worldwide goal. The idea of applying Electric Vehicle (EVs) [1] as an alternative to fossil fuel powered vehicles is gaining lot of interest, while the research and development on EVs including battery design and charging methods have attracted the attention of both commercial and academic communities over the last few years.

Unlike numerous previous works [2] which investigate charging scheduling for EVs parked at home/Charging Stations (CSs), a recent focus has been on managing the charging scenario for on-the-move EVs, by relying on public CSs to provide charging services during their journeys. The latter use case cannot be overlooked as it is the most important feature of EVs, especially for replacing traditional fossil fueled vehicles for journeys. Here, CSs are typically deployed at places where there is high concentration of EVs, such as shopping malls and parking places. On-the-move EVs will travel toward appropriate CSs for charging based on a smart decision on where to charge (referred to as CS-selection), so as to experience short waiting time for charging.

In [3–5], the decision on where to charge is made by a Global Controller (GC) in a centralized manner. Here, the GC can access the real-time conditions of the CSs under its control, through reliable channel including wired-line or wireless communications. There is an issue regarding privacy, as the status of an EV, its ID, State of Charge (SOC) or location [6, 7] will be inevitably released, when that EV sends charging request to the GC. Also there is another issue regarding system robustness. This is because that the charging service will be affected by a single point of failure at the GC. Alternatively, the CS-selection could be made by individual EVs in a distributed manner, based on historically accessed CSs condition information recorded at the EV side. One example of this is provided in [8] where EVs will decide their preferred CSs for charging, based on gathered information from Road Side Units (RSUs).

To make the best CS-selection decision making system, necessary information (such as the expected waiting time at individual CSs) needs to be disseminated between CSs and EVs. In this context, the accuracy of CSs condition information is used for managing EV charging, this plays an important role on the charging performance. Indeed, it is important to position appropriate information dissemination infrastructure to support data exchange between the EVs and power grid. In literature, the cellular network communication (it is normally assumed with ubiquitous communication range) is applied in centralized manner. Meanwhile network entities associated in Intelligent Transportation Systems (ITS) can also help to decentralize the CS-selection decision making, from the centralized GC to localized ITS entities.

## 2 Related Work for CS-Selection

### 2.1 EV Charging in “On-the-Move” Mode

As noted by the most recent survey [9], fruitful works in literature have addressed “charging scheduling” [2] (the “Parking Mode”), by regulating the EV charging, such as minimizing peak load/cost, flattening aggregated demands or reducing frequency fluctuations.

In recent few years, the “CS-selection” problem (or say the “On-the-move” Mode) has started to gain interest, from industrial communities thanks to the popularity of EVs. The works in [3–5] estimate the queuing time at CSs, such that the one with the minimum queuing time is ranked as the best charging option. The work in [3] compares the schemes to select CS based on either the closest distance or minimum waiting time. In [10], the CS with a higher capability to accept charging requests from on-the-move EVs, will propose its service with a higher frequency, while EVs sense this advertisement with a decreasing function of their current battery levels. The CS-selection scheme in [11] adopts a pricing strategy to minimize congestion and maximize profit, by adapting the price depending on the number of EVs charging at each time point.

In addition to above works that consider local status of CSs, reservation-enabled CS-selection schemes bring anticipated EVs mobility information (reservations) to estimate whether a CS will be overloaded in a near future. For example, the work in [12] concerns a highway scenario where the EV will pass through all CSs. Other works under the plug-in charging service [5, 13–15] focus on city scenario.

### 2.2 Urban Data in Intelligent Transportation Systems

Intelligent Transportation Systems (ITS) can fundamentally change urban lives at many levels, such as less pollution, garbage, disposal, parking problems and more energy savings. Exploring big data analytics via ubiquitous, dynamic, scalable, sustainable ecosystem offers a wide range of benefits and opportunities. Most of the techniques require high processing time using conventional methods of data processing. Therefore, novel and sophisticated techniques are desirable to efficiently process the big data generated from the stakeholders, in a distributed manner through ubiquitously disseminated and collected information. This will help to understand the city wide application in a whole picture.

Dedicated authorities should carefully consider which indicators were meaningful or how they should be analyzed. Here, the charging strategy in “On-the-move” Mode certainly benefits from analytics of data from CSs and EVs (that ideally should be captured ubiquitously and timely):

- CS's location condition refers to number of EVs being parked, with their required charging time [8]. A longer service queue implies a worse Quality of Experience (QoE) for incoming EVs, as they may experience additional time to wait for charging.
- Charging reservation at CS indicates which CS to charge. This includes the EV's arrival time, and expected charging time upon arrival at that CS.
- Trip destination refers that EVs would end up with journeys. Inevitably, selecting a CS that is far away from the drivers' trip destination would lead to a worse user QoE.
- Traffic condition on the road affects the EV's arrival time at CS, and energy consumed at that CS. The EV located within a certain range of traffic congestion will have to slow down its speed, whereas it will accelerate the speed once leaving from the range of traffic congestion.

### ***2.3 Communication Technologies in ITS***

ITS applications make use of wireless communications, including communications between vehicles, and between vehicles and fixed roadside installations with single-hops or multiple hops network links. Today's vehicles are no longer stand-alone transportation means, due to the advancements on Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, to access the Internet via recent technologies in mobile communications including WiFi, Bluetooth, 4G, and even 5G networks. The connected vehicles are aimed towards sustainable developments in transportation by enhancing safety and efficiency. Apart from the synchronous point-to-point communication, the topic based asynchronous communication pattern publish/subscribe (P/S) has also been investigated.

### ***2.4 Scalability of Charging System***

The CS-selection problem could be solved in different ways including:

- The centralized approach relies on a cloud based GC to advance the resource efficiency, by taking the advantage of potential economies of scale. This brings much privacy concern, as EV status (e.g., location and trip destination) included in charging request will be released to the GC.
- The decentralized approach provides a better privacy protection, where the charging management is executed by the EVs individually. This alleviates the computation burden of centralized cloud server, by using the localized information at EV to make CS-selection.
- However, the computation capability run by distributed decision makers maybe insufficient; instead, a hybrid way is desirable to enhance the computation robust-

ness, by shifting the computational extensive tasks to GC. While the network edges which are closer to EVs, would process basic information aggregation and mining tasks. The EVs thus make CS-selection decision using information obtained from RSUs.

In the following sections, we introduce recent advances on EV charging systems with their enabling ICT technologies.

### 3 Centralized Charging System

#### 3.1 Cellular Network Communication Enabled Charging System

In most of previous works, the decision on where to charge is generally made by a GC in a centralized manner. Here, the GC can access the real-time conditions of the CSs under its control, through reliable channel including wired-line or wireless communications, e.g., cellular network 3G/4G. This supports a ubiquitous and low delay interaction between EVs and GC.

Previous work [5] proposes a reservation-based EV charging scheme that periodically updates the charging reservation. Due to the traffic jam, the EVs' reservation (the arrival time at the CS, as well as the electricity consumption for travelling towards that CS) can be influenced by varied moving speed. Without reservation updating, an on-the-move EV may not reach a CS at the time it previously reserved, whereas the GC still has an obsolete knowledge that EV will reach on time. As such, the estimation on how long an incoming EV will wait for charging, is affected by the accuracy of the reservation information due to mobility uncertainty.

Based on Fig. 1, a typical procedure for our proposed EV charging management scheme is listed as follows:

- **Steps 1–2:** When an on-the-move EV needs charging service, namely  $EV_r$ , it contacts the GC with its charging request (including SOC, location, trip destination). The GC decides the appropriate CS to serve charging request (in terms of the minimized trip duration through an intermediate charging), and the decision is sent back to  $EV_r$ .
- **Steps 3:**  $EV_r$  reports its charging reservation in relation to its selected CS, including its arrival time, expected charging time and parking duration at the CS.
- **Steps 4:** When travelling towards a selected CS,  $EV_r$  periodically checks whether that CS currently selected is still the best choice, by sending a reservation update request to the GC.
- **Steps 5:** The GC then compares a cost in relation to the newly selected CS as well as that of previously selected CS. If charging at the previously selected CS cannot yield the minimum trip duration, the GC will inform  $EV_r$  about an updated arrangement with a new CS-selection.

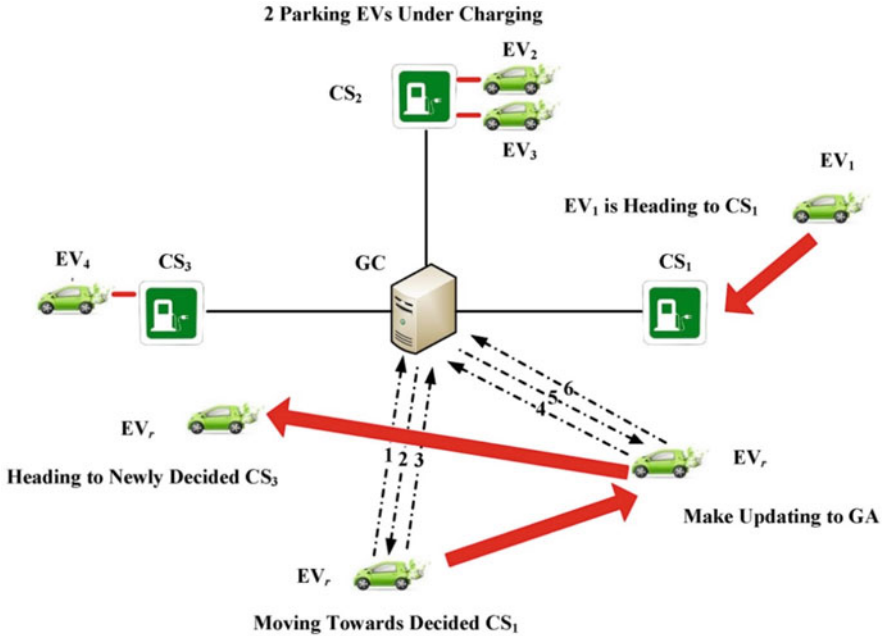


Fig. 1 Overview of reservation updating enabled EV charging

- **Steps 6:**  $EV_r$  thus cancels its reservation at the previous CS, and reports the updated reservation with the newly selected CS. Finally,  $EV_r$  changes its movement towards the location of the newly selected CS.

Steps 4–6 are repeated until  $EV_r$  reaches the newly selected CS for charging. Note that such new arrangement may change several times, depending on the frequency of reservation updating requests to trigger the computing logic detailed in [5].

### 3.2 Enabling Internet of EVs for Charging Reservations Relay

It is worth noting that reporting EVs’ charging reservations (deemed as an auxiliary service), is delay-tolerant (as the essential charging recommendation system still works, even without reservation) and independent of charging request/reply. The 3G/LTE is applied due to its ubiquitous communication deployment. However, this ubiquitous communication comes at a cost and may not be needed all the time. This is because the charging reservations are only generated when EVs need charging.

Recently, the Vehicle-to-Vehicle (V2V) communication is receiving increasing interest, thanks to the inexpensive wireless connections and flexibility of installation on vehicles. Most of the problems in Vehicular Ad hoc NETWORKS (VANETs) arise



from highly dynamic network topology, this results in the communication disruption along an end-to-end path towards destination. Here, the Delay/Disruption Tolerant Networking (DTN) [16] based routing protocols provide a significant advantage, by relying more on opportunistic communication to relay EVs' charging reservations. However, the delay due to opportunistic communication certainly has influence, on how accurate the reservation information is applied by the GC to make CS-selection decisions. E.g., a decision making based on the obsolete information that is due to long delay, may mislead the EV towards a CS in charging congestion. Envisioning for VANETs consisting of EVs, previous work [17] studies the feasibility to take the advantage of opportunistic V2V communication, mainly for the delivery of EVs' charging reservations in a multi-hop way.

When using the V2V communication, the communication cost certainly depends on the number of EVs (as explored in [18]). Whereas the communication cost when using the cellular network communication depends on the number of charging reservations. In other words, the former case is affected by the EVs density, whereas the latter case is affected by the number of service requests.

## **4 Distributed Charging System**

### ***4.1 V2I Communication Enabled Charging System***

In spite of above advanced feature facilitating the charging reservation, the centralized system has issue from privacy aspect as aforementioned in [6, 7], when that EV sends charging request to the GC. Also, concerning system robustness, the charging service will be affected by the single point of failure at the GC side. Alternatively, the CS-selection could be made by individual EV in a distributed way, or the CS-selection decision making is operated by each EV locally.

Many previous works have adopted cellular network, where the application of ITS is also of importance for the future connected vehicles. Strategically deployed Road Side Units (RSUs) can support information dissemination as used by EV charging operations, through the Vehicle-to-Infrastructure (V2I) communication. Considering the V2I communication has been mature in existing VANETs, it is worth noting that the charging system will necessitate wireless V2I communication for EV charging perspective in addition to road safety perspective. Of course, how to realize RSU functions has been discussed in many previous works [19].

In order to enabling the distributed charging management, in [8] the RSUs are introduced as an intermediate entity. They can bridge the information flow exchange between EVs and the grid infrastructure CSs, through the wireless communication technologies. The actual realization of RSUs can be based on existing wireless communication technologies, such as cellular base stations or WiFi access points. It is worth mentioning that different realization of RSUs (in particular the radio transmission coverage) would affect the actual charging information, due to the

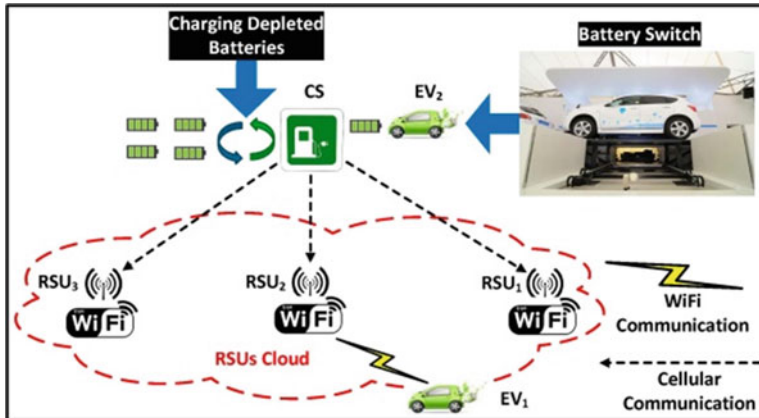


Fig. 2 V2I enabled communication network for battery switch

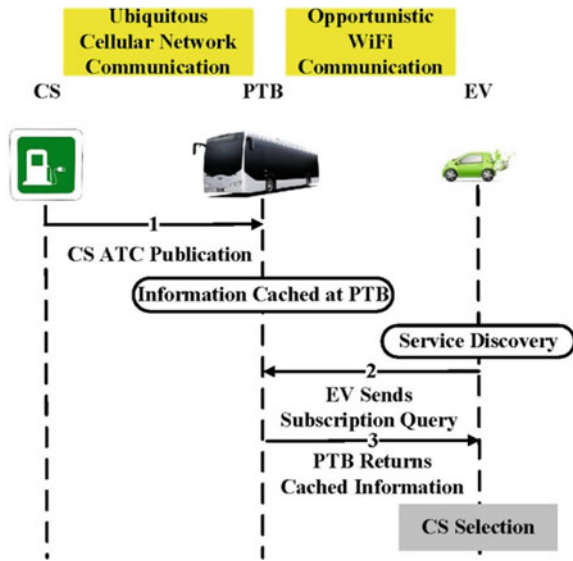
information freshness related to the data exchange between EVs and the grid. The battery switch based charging system [20] is further conducted based on this V2I communication network (Fig. 2).

#### 4.2 V2V Communication Network Enabled Charging System

In the context of new communication technologies especially for smart transportation and autonomous cars, new mechanisms have been proposed in connected vehicle environments including V2I and V2V communications. On the one hand, V2I based approaches suffer additional costs to deploy and maintain dedicated stationary infrastructures, and additionally they suffer from rigidity due to the lack of flexibility of deploying and possibly relocating fixed RSU facilities. In comparison, the V2V based approach [13] is a more flexible and efficient alternative, this supports necessary data dissemination between connected vehicles when they encounter each other.

Due to high mobility, it is difficult to maintain a contemporaneous end-to-end connection (through a synchronous communication, e.g., unicasting, multicasting etc) between the CS and EV through Public Transportation Bus (PTB). The P/S communication framework is based on caching the CS condition at PTB. Here, the previously published CS condition information can be cached at intermediate PTBs. Whenever there is a future encounter between a pairwise PTB and EV, the EV can access the cached information by sending a query.

**Fig. 3** Signalling flow of V2V communication network with PTBs



#### 4.2.1 Basic Charging System without Supporting Charging Reservation

The P/S communication framework envisioning for EV charging scenario (with PTBs to relay information) is introduced as follows, with the time sequences illustrated in Fig. 3:

- **Step 1:** Each CS periodically publishes its condition information, e.g. Available Time for Charging (ATC) using the “*ATC Update*” topic defined in Table 1, to all the designated PTBs (that are involved in message dissemination in the P/S system) through the cellular network communication. In order to make efficient usage of the cellular link equipped at the PTB side, the PTB will aggregate the information in relation to each CS, as illustrated in the payload of topic. Then the aggregated information about all CSs condition is cached in the storage of PTB. Similar to the V2I based communication framework, once a new value has been received depending on CS publication frequency, it will replace the obsolete values in the past, which are not necessarily maintained.
- **Steps 2–3:** Given an opportunistic encounter between pairwise EV and PTB, the EV could discover whether the PTB has a service to provide CSs condition, based on existing service discovery, e.g., the location based scheme [21] proposed for VANETs. Then the EV sends an explicit query to the PTB, via the same topic through WiFi communication. Upon receiving this query, that PTB then returns its latest cached CSs condition information to that EV. With this knowledge, an EV requiring charging service can make its own decision on where to charge.

Under the city scenario, each public PTB is as an intermediate entity for bridging the information flow from CSs to EVs. Note that, in Fig. 3, the role of opportunistic

**Table 1** Topic of “ATC update”

Topic name	Dissemination Mode	Publisher	Subscriber	Payload
<i>ATC update</i>	Many to Many	CSs	EVs	<“CS-1 ID”, “Publication Time Stamp”, “CS-1 ATC”> <“CS-2 ID”, “Publication Time Stamp”, “CS-2 ATC”> <“CS-3 ID”, “Publication Time Stamp”, “CS-3 ATC”> <“CS-4 ID”, “Publication Time Stamp”, “CS-4 ATC”>

WiFi is effectively used as the default radio communication technology, to enable the short-range communication between EVs and their encountered PTBs for information dissemination operations. This can be envisioned for the real-world application, where PTBs providing WiFi communication (already been applied in real world bus system), behave as mobile access points for information dissemination.

#### 4.2.2 Reservation Based Charging System

Upon above generic framework, the EV which has made its CS-selection further sends its charging reservation (including when to reach and how long its expected charging time will be at the selected CS). Apart from the information flow relayed from the CSs to EVs in Sect. 4.2.1, the charging reservation will be relayed from the EV to its selected CS, also through opportunistically encountered PTBs.

With anticipated EVs’ reservations, the charging plans of EVs can be managed in a coordinated manner. For example, if a CS has been reserved by many on-the-move EVs for charging, that CS predicts and publishes its status associated with a near future. Other EVs need charging services would identify the congestion status of CS, and thus select an alternative CS for charging purpose to avoid the congestion. Here, the CS-selection policy based on the Expected Earliest Time Available for Charging (EETAC) published from CSs, is to find the CS at which the EV would experience the shortest charging waiting time.

In Fig. 4, a typical procedure is illustrated as follows:

- **Steps 1–3:** These steps are still executed through the procedure in Sect. 4.2.1. Note that the information disseminated (e.g., the EETAC through topic “*EETAC Update*” topic in Table 2) is different from the ATC involved in Sect. 4.2.1.
- **Steps 4–5:** Based on accessed information, any EV requiring charging service can make its own decision on where to charge, and further publishes its charging reservation to any encountered PTB. Here, each PTB as subscriber

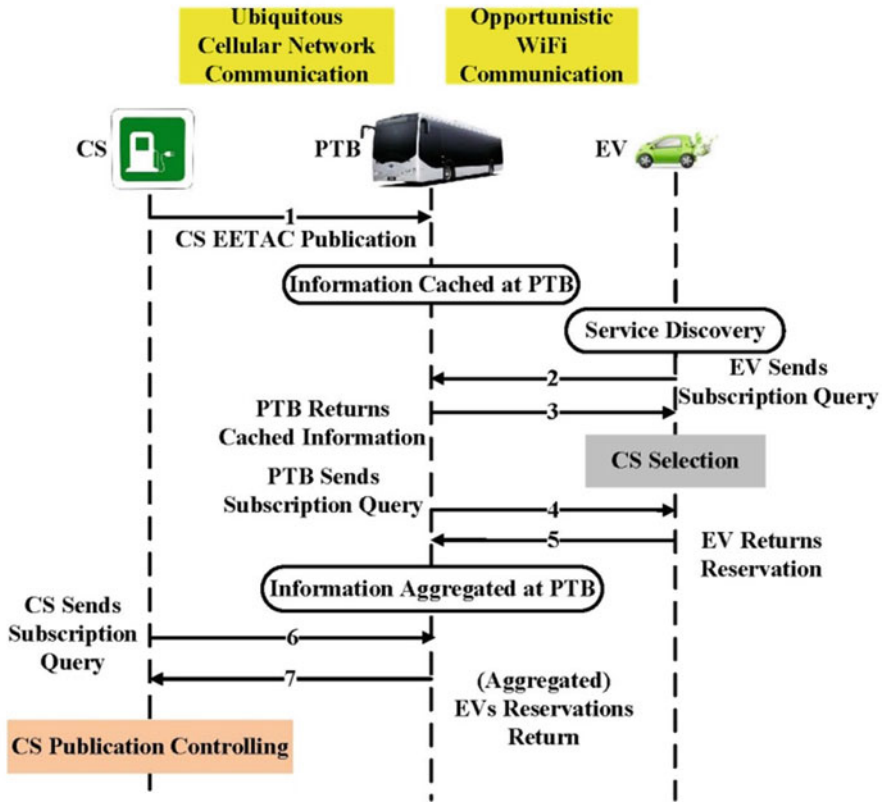


Fig. 4 Signalling flow of charging system via PTB network

Table 2 Topic of “EETAC update”

Topic name	Dissemination Mode	Publisher	Subscriber	Payload
EETAC update	Many to Many	CSs	EVs	<“CS-1 ID”, “Publication Time Stamp”, “CS-1 EETAC”>
				<“CS-2 ID”, “Publication Time Stamp”, “CS-2 EETAC”>
				<“CS-3 ID”, “Publication Time Stamp”, “CS-3 EETAC”>
				<“CS-4 ID”, “Publication Time Stamp”, “CS-4 EETAC”>

sets a “Reservations Aggregating” topic defined in Table 3, and subscribes to the reservations from encountered EVs. The number of such topics depends on number of PTBs, as each PTB uses its individual topic to collect EVs’ reservations.

**Table 3** Topic of “Reservation aggregating”

Topic name	Dissemination Mode	Publisher	Subscriber	Payload
Reservations aggregating	Many to One	EVs planning charging	PTBs	EVs' charging reservation

**Table 4** Topic of “Aggregated reservations collection”

Topic name	Dissemination Mode	Publisher	Subscriber	Payload
Aggregated reservations collection	Many to One	PTBs	CS	<Next time stamp for CS publication, aggregated EVs' charging reservations

- **Steps 6–7:** At the CS side, it accesses aggregated EVs' reservations through the “*Aggregated Reservations Collection*” topic defined in Table 4. The number of this topics depends on number of CSs, as aggregated reservations are in line with an explicit CS. Note that all aggregated EVs' reservations (in relation to an explicit CS) stored at PTBs should be published to that CS, before its next publication time stamp as given by  $(X + T)$ . Recalling that  $X$  is the time stamp for previous CS publication, while  $T$  is the CS publication frequency. Such information triggers all PTBs connected (through cellular network communication) with an explicit CS, to publish their aggregated EVs' reservations related to that CS. The CS computes its updated EETAC and releases at next publication time slot.

## 5 Hybrid Charging System

### 5.1 V2I Communication Network Enabled Charging System

In [14], a hybrid charging system is designed based on V2I communication. All CSs periodically publishes its Available Time for Charging (ATC) to RSUs. Furthermore, EVs are capable of making remote reservations to the GC through RSUs, before reaching their selected CSs. The GC then analyzes the EVs' charging reservations together with their associated CS's local condition information, to compute and notify ATC publication of that CS. The GC also schedules the amount of electricity among CSs, depending on the anticipated charging demands (identified from received EVs' charging reservations) (Fig. 5).

The system designs a closed control loop to adjust a time window within which the prediction is valid, via the analytics of EVs arrival time. Therefore, the sooner EVs will approach CSs for charging, a much tight time window should be determined for prediction and vice versa. Besides, the aggregation at RSUs benefits to communication cost involved for reservation reporting within system.

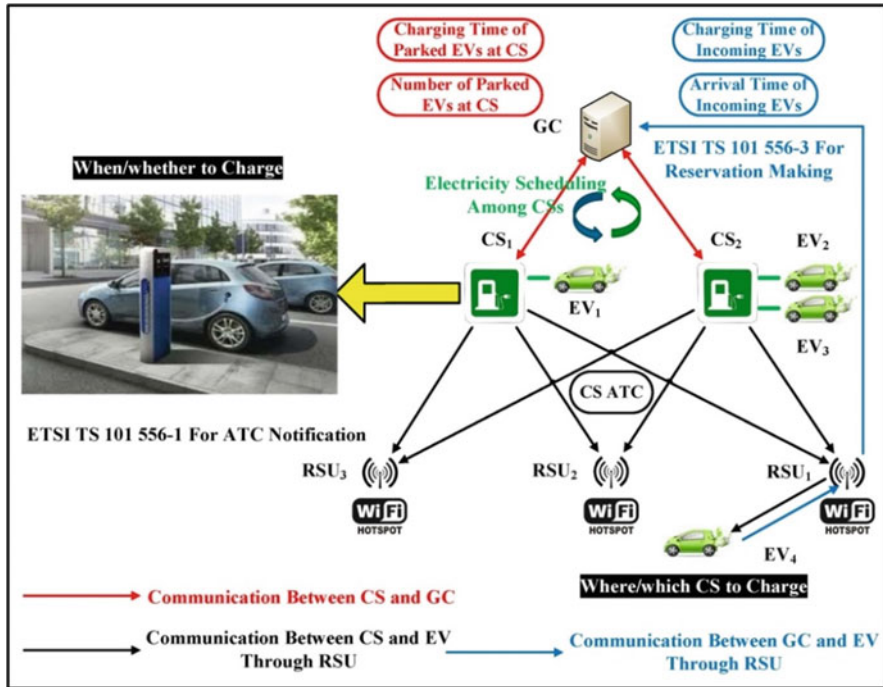


Fig. 5 Overview of hybrid charging system of V2I communication network

The “ETSI TS 101 556-1” [22] standard has been defined for the on-the-move EV charging use case. Its basic application is to notify EV drivers about the CSs’ status (e.g., ATC), such that EVs are able to select CSs for charging. In addition, the “ETSI TS 101 556-3” [23] standard enables the remote charging reservation service, from EVs to the GC. Figure 6 shows a typical procedure:

- **Step 1:** Each CS periodically (with publication interval  $T$ ) publishes its ATC to all RSUs, using its individual “*ATC Update*” topic (defined in Table 5). The RSU subscribes to the publications from all CSs, will aggregate and cache their ATC information.
- **Steps 2–3:** Given an opportunistic encounter between pairwise EV and RSU, the EV fetches the cached information from that encountered RSU. Here, the EV is aware of an updated service published from RSU (through existing service discovery protocols). As such, it only subscribes to the aggregated ATC of CSs, which is published at updated time slot using the “*Aggregated ATC Update*” topic. This reduces the redundant access signalling, particularly when the EV frequently encounters several RSUs in a short time.
- **Step 4:** The EV requiring charging service can make its own CS-selection decision on where to charge, and further publishes its charging reservation to an encountered RSU. Here, the “*Charging Reservations Report*” topic is applied,

**Table 5** Topics of hybrid charging system with V2I communication network

Topic	Dissemination nature	Publisher	Subscriber	Payload
<i>ATC update</i>	One-to-Many	CS	RSUs	<CS ID, CS's ATC, publication time slot>
<i>Aggregated ATC update</i>	Many-to-Many	RSUs	EVs	<Aggregated CS IDs and CSs' ATC, publication time slot>
<i>Charging reservations report</i>	Many-to-Many	EVs planning charging	RSUs	<EV ID, arrival time, expected charging time>
<i>Aggregated charging reservations report</i>	Many-to-One	RSUs	GC	<Aggregated EVs' reservations cached by RSUs>
<i>Local condition update</i>	Many-to-One	CSs	GC	<CS's local condition information, including number of EVs parking at CS and their charging time>
<i>ATC controlling</i>	One-to-Many	GC	CSs	<Computed ATC of each CS>

with the EV as publisher and RSUs as subscribers. Each RSU aggregates its received EVs' charging reservations and locally caches it.

- **Steps 5–6:** At the GC side, it sets two dedicated topics to collect information from CSs and RSUs. Rather than seamless operation (real-time monitoring), such collection task is only operated when the next time slot for CSs' publication is approaching. The local condition information of CSs includes the number of EVs been parked and their required battery charging time, which is accessible by sending a subscription query via the "*Local Condition Update*" topic. The GC also accesses aggregated EVs' charging reservations from all RSUs, using the "*Aggregated Charging Reservations Report*" topic.
- **Step 7:** The GC then computes the ATC related to each CS, and controls their publication at the next time publication interval, using the "*ATC Controlling*" topic.

Compared to [13], this work brings heterogeneous topics illustrated in Table 5 and enables light-weight computation at RSUs side. If using single topic for publication, there is no information merged at RSUs. In that case, an EV needs to use different subscription topics (associated to a CS) to access all CSs information from RSUs, particularly when CSs are owned by different stakeholders. In comparison, all merged CSs information at RSUs can be subscribed by an EV with unique topic. Each RSU can further verify the information of CSs and authorized information for caching, meanwhile check the time slot involved in the EV subscription.

In this system, the communication cost at CS side for information dissemination is given by  $O\left(\frac{N_{rsu}}{T}\right)$ , since there are only  $N_{rsu}$  subscribers within each  $T$  interval.



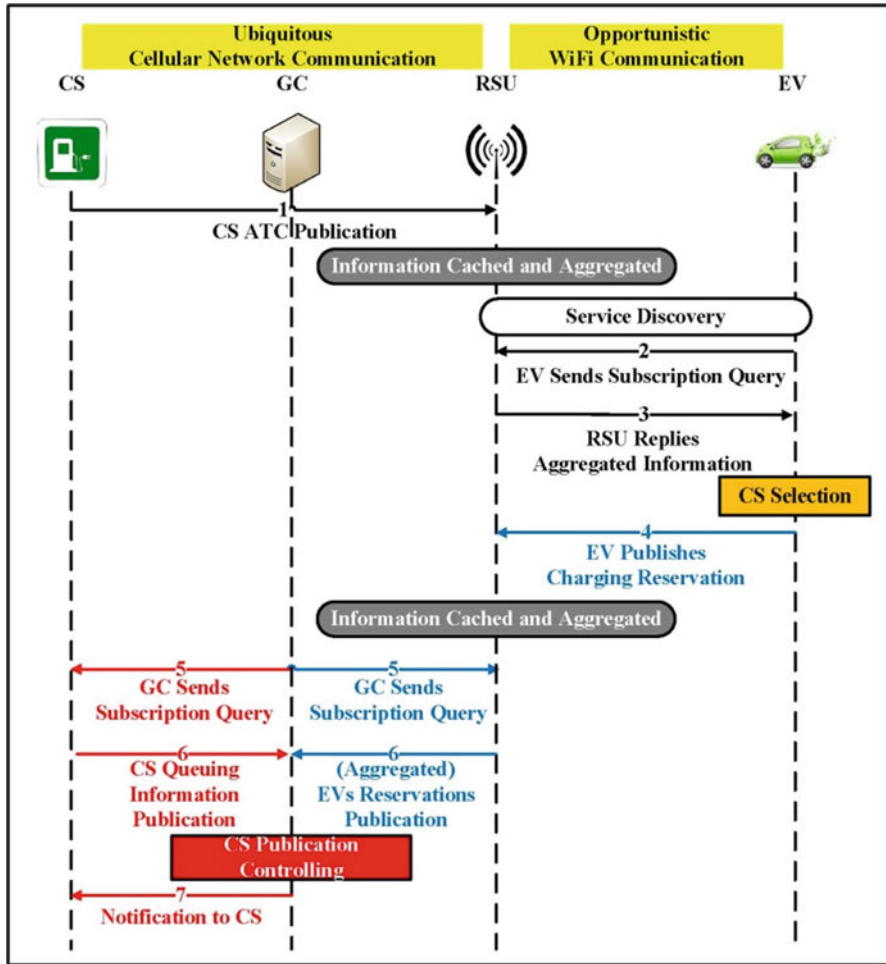


Fig. 6 Signalling flow of hybrid charging system with V2I communication network

Similarly, the cost for reservations making to the GC is given by  $O\left(\frac{N_{rsu}}{T}\right)$ , owing to the information aggregation at RSUs. In comparison, the centralized system is with cellular network communication, the cost at GC side for handling EVs' charging requests and charging reservations are both  $O(N_{ev})$ .

In reality, it is reasonable that  $(N_{rsu} \ll N_{ev})$ , while the number of charging services is larger than  $N_{ev}$  (meaning that each EV needs to charge more than once in long term). As such, the efficiency and scalability of hybrid system is achieved. Having no direct communication between service providers and clients, this system also alleviates the attack surface of network entities.

## 5.2 V2V Communication Network Enabled Charging System

The rapid growth of Internet of Vehicles (IoV) applications have placed severe demands on cloud infrastructure, which has led to moving computing and data services towards the edge of cloud, resulting in a novel Mobile Edge Computing (MEC) [24] architecture. MEC could reduce data transfer times, remove potential performance bottlenecks, and increase data security and enhance privacy while enabling advanced applications such as smart functioned infrastructure. The major difference between cloud computing and MEC, is on the location awareness to support application services (Fig. 7).

This is because the cloud server [25] normally locates in a centralized place, behaves as a centralized global manager to compute tasks (with information collected ubiquitously). Note that, MEC servers at different locations [26] are owned and managed by separate operators and owners. With the collaboration among different operators, they can form a collaborative and decentralized computing system in the wide region.

The work in [15] further extends the hybrid charging system with V2X communication network enabled, by using PTBs while with a discussion on potential of Unmanned Aerial Vehicles (UAVs). Basically, UAV are flying aircrafts which can either be controlled remotely or autonomously. Despite the fact that relatively large UAV platforms are playing increasingly prominent roles in strategic and defense programs, technological advances in the recent years have led to the emergence of smaller and cheaper UAVs.

Even though RSUs have been widely applied in VANETs, the deployment introduces additional economy cost. In addition to deployment cost, effectiveness and utilization of RSUs may also depend on the number of EVs that are presented in a given area. Although applying PTBs envisions for a more flexible way than RSUs, the bus mobility limited by regulated routes (only covers majority areas of a city) may degrade the coverage of information dissemination. Even if the mobility of UAVs is not limited by any route, the energy constraint is a primary concern for operating a large number of UAVs, where the interaction between UAVs and EVs leads to massive network overhead and can eventually undermine the UAVs' energy (thus its average lifetime). Inevitably, to frequently recharge UAVs degrades the network connectivity.

## 6 Further Discussions

### 6.1 Energy Sustainability

The wide spread of EVs experienced in recent years, must be accompanied by sufficient grid infrastructure deployment. The mismatch between EVs and infrastructures would potentially hinder the popularity of EVs. With the ever increasing

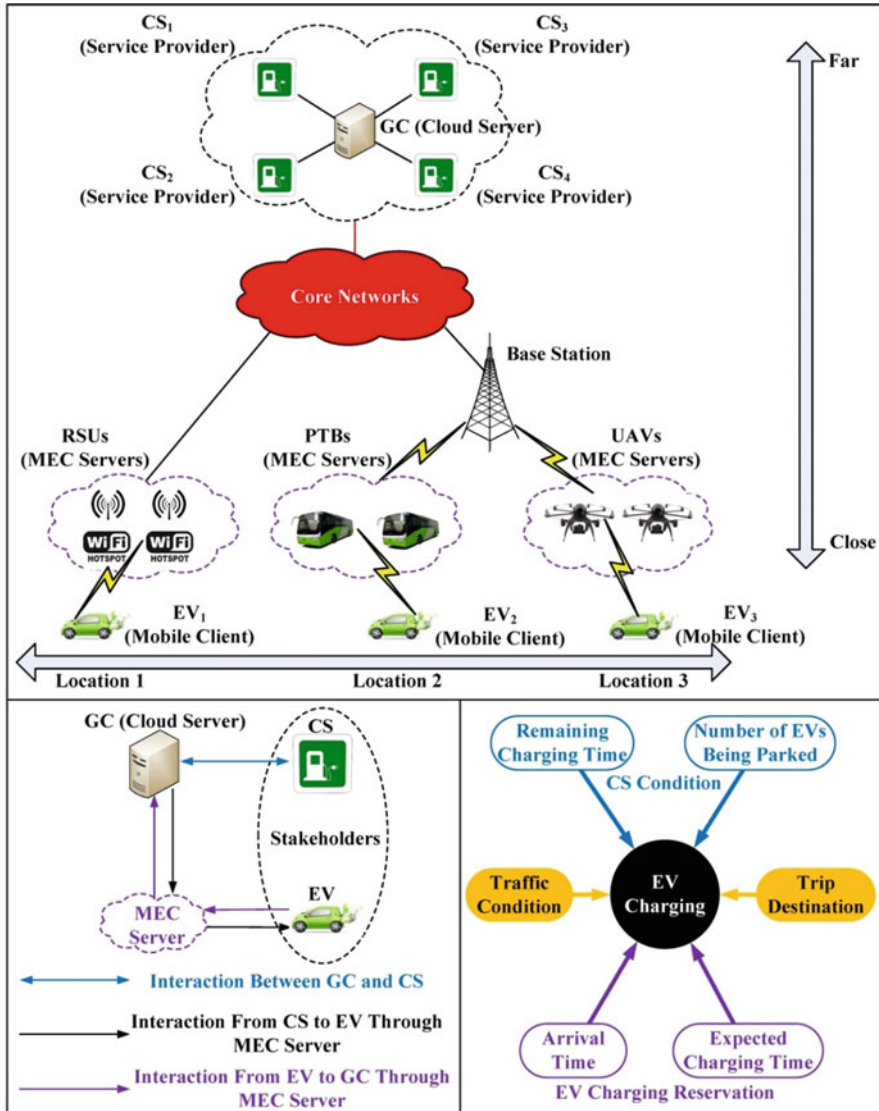


Fig. 7 Overview of MEC supporting EV charging

penetrations in EVs, the resultant charging energy imposed on the electricity network could lead to grid issues, such as voltage limits violation, transformer overloading, and feeder overloading at various voltage levels. The charging coordination with renewable energy source provides a more straightforward approach, to cope with the potential network issues. For example, the generation profile from photovoltaic coincides with the usage pattern and therefore charging profile of public charging stations.

Besides, the engagement of Vehicle-to-Grid (V2G) adapts charging points to have the capability for bidirectional power flow. With appropriate control and communication with the grid, EVs could be designed to operate as part of a “grid”, and this helps to provide supply/demand matching for energy sustainability.

## **6.2 Data Analytics**

The sustainability of EVs requires a fundamental study on data analytics on how/whether/which drivers are desirable to switch from fossil fueled vehicles to EVs. This requires the human centric data related to their routine, finance to predict and educate drivers regarding switch benefit. Also, the driving pattern of EVs will be important to guide with optimal deployment of charging infrastructures.

## **6.3 Security and Privacy**

The solution to achieve trustful messages exchange is to encrypt the sensitive information and hide the real identity. One development of the encryption involves the light-weight and highly secured encryption scheme, while another one is to design an efficient and scalable key management scheme. As for the privacy side, pseudonym is proposed to hide the identities. This includes the pseudonym changing algorithms and pseudonym reuse schemes, and both should be implemented in efficient and scalable manners. The future challenges are considered based on the nature of large number of connected EVs, high mobility, wide coverage area, heterogeneous communication systems. Security and privacy schemes will have the abilities of little bandwidth resources consumption, large number node supportable and short processing time.

## **7 Conclusion**

This chapter reviewed a number of up-to-date literature works which study the integration of ICT with EV charging. The optimization problem is scaled from transportation angle, which aims to minimize the charging waiting time. The centralized, distributed and hybrid systems in line with cellular network, V2I&V2V communication networks have been presented and integrated into EV charging systems. In summary, the centralized charging system relies on the GC to handle charging requests from EVs, and to make decision on which CS that should EV plan for charging. In the distributed charging system, each EV could make their individual decisions for CS-selection, where the RSUs and PTBs are applied to bridge the information publication from CS to EVs. The hybrid charging system

facilitates the computation advance of GC to predict and control the information dissemination in network. Meanwhile, it shifts the light-weight computation at network edge for information caching and mining to help EV for CS-selection decision making.

## References

1. Schewel, L., & Kammen, D. M. (2010). Smart transportation: Synergizing electrified vehicles and mobile information systems. *Environment*, 52(5), 24–35.
2. Mukherjee, J. C., & Gupta, A. (2015). A review of charge scheduling of electric vehicles in smart grid. *IEEE Systems Journal*, 9(4), 1541–1553.
3. Yang, S. N., Cheng, W. S., Hsu, Y. C., Gan, C. H., & Lin, Y. B. (2013). Charge scheduling of electric vehicles in highways. *Mathematical and Computer Modelling*, 57(11), 2873–2882.
4. De Weerd, M. M., Stein, S., Gerding, E. H., Robu, V., & Jennings, N. R. (2016). Intention-aware routing of electric vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(5), 1472–1482.
5. Cao, Y., Wang, T., Kaiwartya, O., Min, G., Ahmad, N., & Abdullah, A. H. (2016). An ev charging management system concerning drivers' trip duration and mobility uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
6. Yu, C. M., Chen, C. Y., Kuo, S. Y., & Chao, H. C. (2014). Privacy-preserving power request in smart grid networks. *IEEE Systems Journal*, 8(2), 441–449.
7. Lei, A., Cruickshank, H., Cao, Y., Asuquo, P., Ogah, C. P. A., & Sun, Z. (2017). Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2017.2740569>.
8. Cao, Y., Wang, N., Kamel, G., & Kim, Y. J. (2017). An electric vehicle charging management scheme based on publish/subscribe communication framework. *IEEE Systems Journal*, 11(3), 822–835.
9. Rigas, E. S., Ramchurn, S. D., & Bassiliades, N. (2015). Managing electric vehicles in the smart grid using artificial intelligence: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 1619–1635.
10. Hausler, F., Crisostomi, E., Schlote, A., Radusch, I., & Shorten, R. (2014). Stochastic park-and-charge balancing for fully electric and plug-in hybrid vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 15(2), 895–901.
11. E. Rigas, S. Ramchurn, N. Bassiliades, and G. Koutitas. (2013), "Congestion Management for Urban EV Charging Systems," Paper present at the IEEE International Conference on Smart Grid Communication, Vancouver, Canada, 21–24 October, 2013.
12. H. Qin, & W. Zhang. (2011). "Charging Scheduling with Minimal Waiting in a Network of Electric Vehicles and Charging Stations", Paper present at the ACM international workshop on Vehicular inter-networking, Las Vegas, Nevada, USA, 23 September, 2011.
13. Cao, Y., & Wang, N. (2017). Toward Efficient Electric-Vehicle Charging Using VANET-Based Information Dissemination. *IEEE Transactions on Vehicular Technology*, 66(4), 2886–2901.
14. Cao, Y., Kaiwartya, O., Wang, R., Jiang, T., Cao, Y., Aslam, N., & Sexton, G. (2017). Toward Efficient, Scalable, and Coordinated On-the-Move EV Charging Management. *IEEE Wireless Communications*, 24(2), 66–73.
15. Cao, Y., Song, H., Houbing Song, Kaiwartya, O., Zhou, B., Zhuang, Y., Cao, Y., and Zhang, X. "Mobile Edge Computing for Big Data-Enabled Electric Vehicle Charging". *IEEE Communications Magazine*. (To appear in 2017)
16. Cao, Y., & Sun, Z. (2013). Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications surveys & tutorials*, 15(2), 654–677.

17. Cao, Y., Zhang, X., Wang, R., Peng, L., Aslam, N., and Chen, X. (2017) "Applying DTN Routing for Reservation-Driven EV Charging Management in Smart Cities", Paper present at 13<sup>th</sup> IEEE International Wireless Communication and Mobile Computing Conference, Valencia, Spain, 26–30 June, 2017.
18. Cao, Y., Sun, Z., Wang, N., Cruickshank, H., & Ahmad, N. (2013). A reliable and efficient geographic routing scheme for delay/disruption tolerant networks. *IEEE Wireless Communications Letters*, 2(6), 603–606.
19. M. Rashidi, I. Batros, T. Madsen, M. Riaz, and T. Paulin, (2012) "Placement of Road Side Units for Floating Car Data Collection in Highway Scenario," Paper presented at 4<sup>th</sup> International Congress on Ultra-Modern Telecommunication and Control Systems, Petersburg, Russia, 3–5 October, 2012.
20. Cao, Y., Yang, S., Min, G., Zhang, X., Song, H., Kaiwartya, O., & Aslam, N. (2017). A Cost-Efficient Communication Framework for Battery-Switch-Based Electric Vehicle Charging. *IEEE Communications Magazine*, 55(5), 162–169.
21. Kaiwartya, O., Abdullah, A., Cao, Y., Lloret, J., Kumar, S., Aslam, N., Shah, R. "GeoLR: Geometry based Localization and Re-Location Assistance for GPS Outage in VANETs". *IEEE Transactions on Vehicular Technology*. (To appear in 2018).
22. "ETSI TS 101 556-1 (2012) v1.1.1 Intelligent Transport Systems (ITS); Infrastructure to Vehicle Communication; Part 1: Electric Vehicle Charging Spot Notification Specification," Tech. Rep.
23. "ETSI TS 101 556-3 v1.1.1 Intelligent Transport Systems (ITS); Infrastructure to Vehicle Communications; Part 3: Communications System for the Planning and Reservation of EV Energy Supply Using Wireless Networks," Tech. Rep.
24. Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3), 1628–1656.
25. Yang, Bin., Chai, Wei., Pavlou, G., Katsaros, K. (2016). "Seamless Support of Low Latency Mobile Applications with NFV-Enabled Mobile Edge-Cloud", Paper present at 5<sup>th</sup> IEEE International Cloud Networking, Pisa, Italy, 3–5 October, 2016.
26. Zhou, B., & Chen, Q. (2016). "On the Particle-assisted Stochastic Search Mechanism in Wireless Cooperative Localization", *IEEE Transactions on Wireless Communications*, 15(7), 4765–4777.

# Green Transportation Choices with IoT and Smart Nudging



Anders Andersen, Randi Karlsen, and Weihai Yu

**Abstract** Transportation and traffic have become a serious issue around the world. Congestion has severe negative consequences for public health, productivity and the environment, and less reliance on private cars can help solve many interconnected problems in these areas. Increased use of green transportation choices can also limit the need for expensive investment in infrastructure and minimize unpopular traffic regulations. One approach to achieve this is to introduce new kinds of services that provide members of the community with situational aware information of different forms. This includes information to travellers about which environmentally friendly transportation options that are available and match their needs on any given occasion, and how they can find and use these options. Thus, we should motivate people to make green transportation choices and provide them with situationally relevant details about when and how to do so. Nudging is a term from economics and political theory for influencing decisions and behaviour using suggestions, positive reinforcement and other non-coercive means, so as to achieve socially desirable outcomes. In our context, the people to influence are members of a community using various modes of transportation. The main components in such an approach are (i) *Sense*, (ii) *Analyse* and (iii) *Inform and Nudge*. With the combination of *Sense*, *Analyse* and *Inform and Nudge* we can provide smart nudging.

**Keywords** Environmental friendly · Green transportation · Context sensitive · Smart nudging · Situational aware · Personalisation






## 1 Introduction

Urban challenges of increased traffic, congestion, and air and noise pollution, have become serious issues and have to be addressed [9]. These local challenges also have an impact on global scale issues, including climate changes and global

---

A. Andersen (✉) · R. Karlsen · W. Yu  
Department of Computer Science, UiT The Arctic University of Norway, Tromsø, Norway  
e-mail: [Anders.Andersen@uit.no](mailto:Anders.Andersen@uit.no); [Randi.Karlsen@uit.no](mailto:Randi.Karlsen@uit.no); [Weihai.Yu@uit.no](mailto:Weihai.Yu@uit.no)

**Table 1** Environmental friendliness (EF) and discouraging and encouraging factors for different types of transportations. Larger EF means more environmental friendly type of transportation

Type	EF	Discouraging factors	Encouraging factors
Car		Economy (toll, parking, gas), traffic jam	Convenience
Carpool		Inconvenience, traffic jam	Economy, social
Bus		Schedule, traffic jam	Economy, priority in traffic
Bike		Time, effort, exposed to conditions	Economy, health
Walk		Time, effort, exposed to conditions	Economy, health

warming, increased health problems, and bottlenecks in logistics chains. Urban mobility has to be rethought by making alternatives to private car usage more attractive [9]. Alternatives include walking, cycling, public transportation, and the usage of motorbikes and scooters. Another approach is to optimise and limit the usage of private cars through carpooling and incentive parking (park and ride) facilities.

Green transportation choices are about choosing transportation methods that are more environmentally friendly. Since environmental friendliness can be placed on a scale, the goal is to influence people to make transportation choices that are more environmentally friendly than the default transportation choice of the person. So, for a person normally driving to work with a private car, choosing the bus is a greener transportation choice, while choosing the bike or walking is even better.

Table 1 gives examples of different types of transportation, their environmental friendliness, and their discouraging and encouraging factors.

The quantification of environmental friendliness of vehicles is a complex task. Estimates are based on a number of parameters, including number of passengers, engine type (e.g. gasoline and diesel) and size [11], tire type (e.g. summer, winter and spiked), travel distance, and travel length.

People decide how to travel according to a number of factors, including time spent for the travel, convenience, economy, and health. For example, high toll or parking fee might discourage people from driving, and traffic jam might encourage a potential car driver to take the bus. On the other hand, traffic jam might also discourage a person from taking buses and encourage the person to ride a bike instead. Even when people have decided to drive to work, there are still different options available, including sharing through carpool, and selecting alternative routes.

To collect information about the user's of different transportation choices and the current situation regarding traffic, weather, conditions, and so on, sensors play an important role. Recent development in Internet of Things (IoT), including availability of affordable microcontrollers and sensors that can operate in resource constrained environments, have improved the possibilities to sense our surroundings. This, combined with the availability of long-range low-power communication infrastructures, is an important building block in facilitating smart nudging for green transportation choices.



To enable smart nudging, where users receive relevant information and suggestions based on their current situation, sensor data needs to be collected (often from a wide range of sensors), combined in various (possibly context-determined) ways, analyzed and transformed into information and nudges that are presented to the user. This is a complex task, since sensor data may come from a wide variety of sensors, supporting e.g. heterogeneous data formats, where the combination of data may vary depending on the information need and the current situation of the user. A range of data analysis tools are available and may be chosen depending on the intended outcome. Finally, inform and nudge should be personalized to the user's needs and situation, and choices have to be made concerning nudging tools and presentation of the information.

Having a variety of sensor data and analysis tools available, and the ability to combined these, set the stage for a multitude of information and nudge services that in various ways can help the user make informed decisions, for example about the currently best transportation choice.

We will in the following first describe smart nudging, covering the topics of nudging, personalization and situation-awareness. Then a generic architecture for smart nudging is presented, and we will continue by describing aspects of each of the components in the architecture, i.e. the *Sense*, *Analyze* and *Inform and Nudge* components.

## 2 Smart Nudging

With smart nudging, we seek to present people with information relevant to their decision making, such as bus routes and schedule, real-time traffic and air pollution condition [33], so that they are encouraged to travel in more environmental friendly ways.

Smart nudging is nudging matching the current situation of the user. A better understanding of the situation of the user will make it possible to have a greater success in encouraging the user by nudging. Such knowledge is based on collecting a wide range of data and information, analysing this in the context of the user, and personalising it, before it is used to inform and nudge the user.

### 2.1 Nudging

*Nudging* is a term from economics and political theory for influencing decisions and behaviour using suggestions, positive reinforcement and other non-coercive means, so as to achieve socially desirable outcomes. The term *nudge* was first used in [37], where it was defined as

... any aspect of the choice architecture that alters people's behaviour in a predictable way without forbidding any options or significantly changing their economic incentives.

A choice architecture refers to the "*environment in which individuals make choices*". The authors also state that:

To count as a mere nudge, the intervention must be easy and cheap to avoid. Nudges are not mandates. Putting the fruit at eye level counts as a nudge. Banning junk food does not.

Nudges aim to influence people's behaviour towards decisions that are beneficial for society, but usually also in the individual's long-term interest [37]. This can, for example, be encouraging a healthier or more environmentally friendlier behaviour.

Four types of tools for nudging have been identified [23]. In the context of green transportation choices, these four tools can be described with the following examples:

1. *Simplification and framing of information*: Presenting relevant and personalised information to users (avoiding information overload and complexity), suggestions for public transportation departures, walking paths, driving routes for minimising fuel consumption and congestions, and presenting information in ways that are attractive and agreeable to the user.
2. *Changes to the physical environment*: Road planning, building easy accessible cycling and walking paths, indoor cycling parking, and easy access to shower and wardrobe at work (for cyclists, runners or walkers).
3. *Changes to the default policy*: Tourist information presents busses or walking paths as the default way to reach sights or stores, information about parking lots from where the busses depart, and presents general transportation choices ordered by environmental friendliness.
4. *Use of social norms*: Possibility to share experiences and tips regarding green transportation services within social networks (e.g. share current state of cycling and walking paths, personal achievements, and positive experiences), competition among workplaces/departments regarding the adoption of green transportation choices.

*Digital nudging*, which is our main concern, is in [40] described as

... the use of user-interface design elements to guide people's behaviour in digital choice environments.

When people make choices, they are influenced not only by the mere facts of the different choices, but also by the way these facts are presented. Digital nudging is also about selecting and combining the right set of information in the given context, so people have better and more relevant information to base their choices on. This is directly linked to the nudging tool "*simplification and framing of information*". But this is not the only tool of relevance for digital nudging approaches. Both "*changes to the default policy*" and "*use of social norms*" are nudging tools where information and communication technology can have a major role in their implementation.

The strength of nudging is the ability to influence people towards behaviour that benefit some common good, without limiting people's freedom of choice. This assumes that nudges are designed by a well-meaning party which has people's best interests in mind. There are objections to nudging, for example pointing to the potential danger of manipulating people and presenting unfair nudges [23, 37]. To counteract these concerns, it is suggested to provide transparency where the users are made aware of the nudging performed on them [37]. Criticism towards nudging is discussed in [18], which also describes a framework distinguishing between transparent and non-transparent nudges, that may be used for characterising nudges and reason about their positive and negative effects.

## 2.2 Personalisation

To influence people's transportation choices, information from various sources are needed. This includes information about traffic, public transportation, road conditions, environmental conditions, and also information about behaviour and transportation patterns of each user. To be effective, information presented in a nudge must be relevant for the specific user and tailored to the user's current transportation needs [3]. Therefore, transportation information and nudges need to be personalised according to the users' behaviour (for example where and when to travel).

Personalisation is described as the ability to provide tailored content and services to individuals based on knowledge about their preferences and behaviour [15]. With information overload and complexity, personalisation has become a valuable tool for assisting users in searching, filtering and selecting information of interest. Personalisation is well known in online stores and web based information systems, and is used in a wide range of applications and services including digital libraries, e-commerce, e-learning, health care, decision-making, search engines and for personalised recommendations of movies, music, books and news [15, 17].

Existing personalisation strategies require construction of *user profiles* that identify interests, behaviour and other characteristics of individual users. A user profile can be described through a number of dimensions. The main user profile dimensions are listed in Table 2.

To construct a user profile, information can be collected *explicitly*, through direct user participation, or *implicitly*, through automatic monitoring of user activities [16, 17]. Implicit collection of profile information is a continuous process, where the current interests and behaviour is constantly mined. The user profile can thus change over time to reflect long lasting, short term and new user characteristics.

By monitoring user activities, we can implicitly collect information about traveling characteristics, such as the locations the user is frequently traveling to or from, at which time the traveling normally is done, and the routes of the traveling. Preferences concerning transportation services can be explicitly given by the user or implicitly inferred through detection.

**Table 2** Main user profile dimensions (from [15])

Dimension	Description/examples
Personal data	Gender, age, nationality and preferred language
Cognitive style	The way in which the user process information
Device information	May be used to personalise presentation of information
Context	The physical environment where the user processes information
History	The user's past interactions
Behaviour	The user's behaviour pattern
Interests	Topics the user is interested in
Intention/Goal	Intention, goals or purposes of the user
Interaction experience	The user's knowledge on interacting with the system
Domain knowledge	The user's knowledge of a particular topic

**Table 3** Examples of user-related information useful for nudging

Information	Example	Description
Travel history	$\langle A, B, t_1, t_2, T \rangle$	Each travel is represented by departure location $A$ , destination location $B$ , departure time $t_1$ , arrival time $t_2$ , and transportation choice $T$ (e.g., car, bus, or bike). It will be used for identifying often travelled distances and times when a distance is most frequently used
Current location	$\langle A, t \rangle$	The current location is represented by the actual location $A$ and the current time $t$ . It can for example be used to compute possible routes relevant for the next nudge
Calendar events	$\langle E, t, d, A \rangle$	Each calendar event is represented by an event title $E$ , the start time of the event $t$ , the duration of the event $d$ , and the location of the event $A$ . Appointments at a different location than the current location will indicate a need for transportation
Preferences	$\langle P \rangle$	A user might have explicitly given preferences $P$ that might interfere with the choice of nudging strategies

Time and current location of the user is important to determine whether to nudge or not. Given time and location, for example  $\langle \text{AT HOME}, 7\text{PM} \rangle$ , transportation suggestions from home to work can be given. However, if the user is on holiday or have a different location, the travel-to-work nudge is not relevant and should not be given. Table 3 lists user-related information that can be useful for nudging.

By combining aggregated knowledge about the user, the current situation, next planned activities from the calendar, and explicitly given user preferences, it should be possible to give personalised (and better) green transportation suggestions to the user. A better knowledge about the personality of the user can be used to further improve the suggestions to the user. Studies have shown that a matching personality of the system providing the suggestions improves the success rate of the suggestions [13].

### 2.3 *Situational Awareness*

The current situation of the user is an important factor in smart nudging. Such situational awareness can only be achieved by sensing and analysing a wide range of data from the context of the user. It is currently possible to monitor, aggregate and predict (among others):

- car traffic in cities and on highways, relevant to services offering traffic-routing advice,
- the flow of vehicular traffic, including average speed and numbers of cars,
- levels of air pollution including carbon monoxide, nitrogen oxides, particulate matter and hydrocarbons,
- used capacity of public transportation,
- the status of footpaths and bicycle paths, and
- weather information that influences transportation choices.

The data involved are both historical data, current data, plans and predictions. *Historical data* are data collected and aggregated over time and they can be used to describe the past. *Current data* can give a snapshot partly describing the current situation. *Plans* are data describing schedules or planned activities. *Predictions* are data trying to describe the future, often based on historical data, current data, plans, and some sort of model modelling the path from present to future. Table 4 lists a few examples of historical data, current data, plans and predictions.

Plans and predictions are about the expected future and are obviously important when transport suggestions are presented to the user in her or his current situation. The suggestions should fulfill a need or expectation for the future, typically, the user’s need to be at a given location at a given time.

To achieve situational awareness, analysis is performed on the data. This can be simple calculation of expected travel time between current location and the location of the next event in the user’s calendar, or more complex calculation of expected pollution levels based on current road conditions, weather forecast, season (e.g. usage of tyres with spikes), and expected traffic flow and congestions.

**Table 4** Examples of historical data, current data, plans and predictions

Data	Examples
Historical	Historical sensor data (weather / pollution); congestion history and traffic flow; successful and unsuccessful user travel experiences; past events
Current	Current sensor data (e.g. weather / pollution); road, footpath og skitrack conditions; current pollution levels; current traffic conditions; current location of expected bus
Plans	User’s calendar events; bus and train schedules; planned infrastructure maintenance; holidays / recreation days; festivals / events influencing transportation infrastructure
Predictions	The weather the rest of the day; predicted pollution levels during the day; the deviation from the schedule of a bus leaving a nearby bus stop

### 3 Smart Nudging Architecture

The main components of the IoT-based smart nudging architecture are *Sense*, *Analyse* and *Inform and Nudge*. This is closely related to the main functions associated with Mobile Phone Sensing [20]. These components will result in a set of services that demonstrate approaches to (i) data collection using sensors, crowdsensing, third party data sources, and crowdsourcing, (ii) analysis that transforms raw data into information, and (iii) outreach to the public through information and nudging services.

An architecture for such services has to support these components in a wide range of configurations, with heterogenous devices and users, and high degree of dynamics. Included devices might be resource constrained and play different roles based on current ongoing tasks and processing. In [32], a set of requirements for IoT architectures (middleware) is outlined, and based on these requirements a comprehensive review of existing middleware is done.

In [38] an architectural approach for IoT is suggested. Their vision for the architecture includes support for both static and dynamic data, and support for integration of an actuator interface. This is of great importance in an IoT-based smart nudging architecture, where the combination of historical data (static data), current data (dynamic data), plans (static data), and predictions (dynamic data) are analysed and used to encourage a change of behaviour of the user (through an actuator interface).

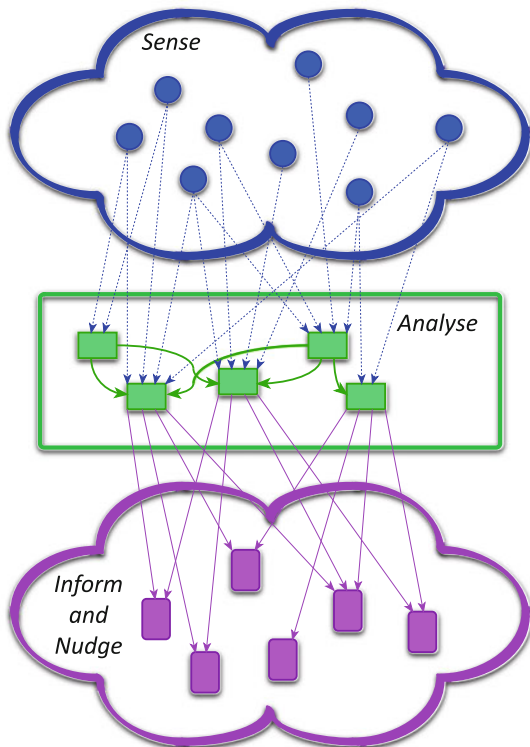
In the back-end part of the system there is nearly unlimited resource for computation and data storage (for example, by using cloud services). However, doing everything at the back-end causes latency and unnecessary transportation of large amount of data. Another issue is access to sensitive and private user data.

With fog computing [31] data processing and analysis are shifted towards leaf or edge nodes. This leads to shorter latency and lower bandwidth consumption. Other benefits include greater context awareness and higher availability. Challenges include weak edge nodes, lack of global knowledge, and the unpredictability of mobile and dynamic environments. Distributed publish-subscribe can be regarded as unifying cloud and fog computing since the brokers can be deployed either in the cloud or close to the edge.

Architectural issues include discovery and management of mobile publishers and subscribers, allocation of topics (different functions) in brokers (data cleansing, feature extraction, data aggregation and filtering etc.), context awareness, and scalability.

The IoT-based smart nudging architecture uses a publish-subscribe approach for combining services from the different components. A data collection service in the *Sense* component can be subscribed to, and used by, multiple services in the *Analyse* component. A processing and filtering service in the *Analyse* component can be subscribed to, and used by, multiple services in the *Inform and Nudge* component. Thus, two *Inform and Nudge* services can combine raw data differently, possibly using different analysis tools, producing different services to the public. This is illustrated in Fig. 1. The sense components will typically be IoT enabled

**Fig. 1** The publish-subscribe architecture with the *Sense*, *Analyse*, and *Inform and Nudge* components



sensors. However, the sense components represent all type of live and updated data, including data from users’ smart phones, external services (e.g. weather data), and data being aggregated for future usage. The analyse components will use data from the sense components, the outcome of other analyse components (e.g. pre-processed data), and data collected and aggregated in the past (including historical data, previous analytics results, and learning data). The inform and nudge components are typical the users’ smart phones, but they could also include situated public displays [35], smart-speakers and other personal assistants.

In a publish-subscribe architecture, the components can subscribe to data. When new data is produced, it is published to all components that have subscribed to the data. Such events might trigger analysis that produces new data (knowledge). And again, the data produced from analysis are published to their subscribers that might perform other analysis or inform and nudge operations. The processes are driven by the arrival of new data that can be used to produce new knowledge that are used in inform and nudge operations for the users.

IoT based sensing devices can produce new data in a periodic cycle or when certain events occur. These devices are connected in an IoT infrastructure where their data is collected, stored, and possibly forwarded to analyse components. A Low-Power Wide-Area network (LPWA network) is a wireless communication network

designed to allow long range communications at a low bit rate with better power consumption characteristics [34]. LoRaWAN [24] is the LoRa Alliance's attempts to define a global standard for LPWA networks. The low power consumption combined with good communication range make it possible to use battery powered IoT devices, and still achieve a decent operation time.

IoT devices are typically part of an IoT infrastructure where the IoT devices are managed, and the data are collected, stored and sometimes processed. The IoT devices communicate through this infrastructure using LPWA or other means of communication. Applications and services using these IoT devices connect with them through IoT infrastructure. Either by performing search for data of interest at the infrastructure, or by subscribing to data of interest using the infrastructure's publish-subscribe services. IoT devices can through such an infrastructure provide data to a wide range of applications and services.

In the following chapters, we will describe in more detail aspects of each of the components depicted in Fig. 1; *Sense*, *Analysis* and *Inform and Nudge*.

## 4 Sense

The sensing elements include existing sensors in operation in the public infrastructure and transport system, new sensors deployed for a more precise and useful situational awareness data, and sensing enabled mobile devices (e.g. smart phones) of the members of the community (the users). But in the context of IoT-based smart nudging, sense includes a wider range of data.

The data sources of interest with smart nudging for more environmental friendly transportation choices can be structured in different ways. In Table 4 we distinguish between *historical data*, *current data*, *plans*, and *predictions*. Another approach is to base it on the source of the data. The data sources are divided into five different categories: (i) IoT sensors, (ii) crowdsensing, (iii) aggregated data, (iv) crowdsourcing, and (v) static data.

These categories are related and can overlap. For example, both IoT sensors and crowdsensing will over time contribute to aggregated and static data.

### 4.1 IoT Sensors

IoT sensors represent a wide range of data sources. It is out of the scope of this text to present a complementary overview of available sensors.<sup>1</sup> Instead a few examples of IoT sensors and their usage with smart nudging for green transportation choices are discussed.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/List\\_of\\_sensors](https://en.wikipedia.org/wiki/List_of_sensors)



Weather and environment sensors have a role in many usage examples. They can be a direct part of a nudge presented for a user (e.g. current nice weather could convince a user to chose the bike), they can influence the prediction of next day pollution levels (high barometric pressure and dry weather will contribute to the possibilities for increased local pollution levels), and they can be used to produce high quality ski wax suggestions to a user that might decide to choose the ski track to work<sup>2</sup> today. Figure 2 includes two different IoT weather and environment sensors developed in our laboratory in cooperation with Telenor and their Start IoT<sup>3</sup> initiative.

Location and proximity sensors are used both to track users and vehicles. The user's location is often essential when providing situational aware services. This can be done with GPS and other location services, or using proximity sensors to detect a close proximity of a device with known location. Tracking the location of buses are used to adjust the expected arrival time at nearby bus-stops based on the current position of the bus and the expected travel time to the bus stop. More input data, like the original bus schedule, sensor data regarding current traffic flow, and historical data for travel time between these two locations on this bus route, could improve the accuracy of the prediction.

Sensors sensing traffic flow, congestions, footpath usage, available parking, and similar, are available in many forms. It is possible to use sensors to detect the conditions of roads, footpaths, bike lanes, and ski tracks. This could detect slippery roads (e.g. ice), snow shovelling status, ski track conditions (how long since last time they were prepared), available parking, and problematic routes. The IoT sensor to the right in Fig. 2 is for measuring humidity in the ground (or at a surface) and can be used to produce data about road conditions. This could, for example, be used to predict slippery conditions from wet road surface or ice. A more complex sensor setup in Fig. 3 is used to measure snow levels. This can be of interest in establishing knowledge about outdoor conditions.

## 4.2 *Crowdsensing*

Crowdsensing (also known as mobile crowdsensing) is a technique where individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest [14]. This can be seen as crowdsourcing of sensor data from mobile devices and is enabled by mobile sensing technology.

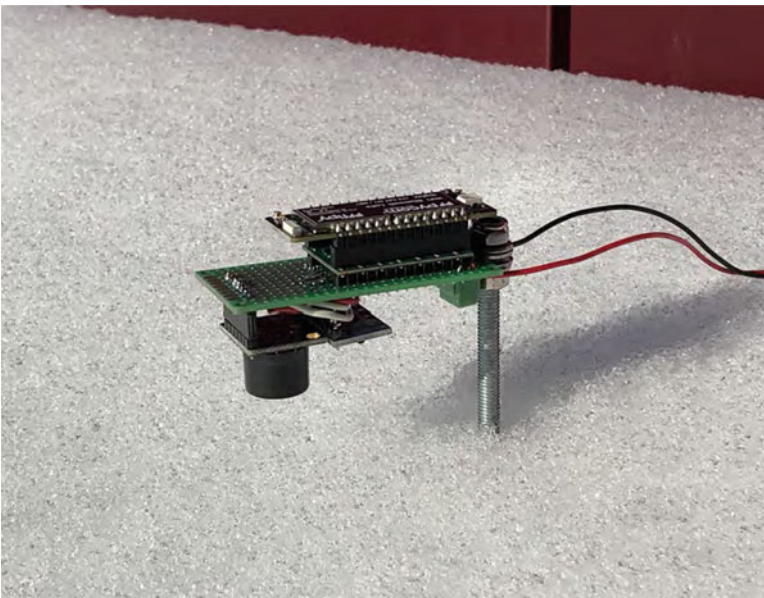
---

<sup>2</sup>The cross country ski track in Tromsø is a popular travel-to-work option at winter.

<sup>3</sup><https://startiot.telenor.com/>



**Fig. 2** Two IoT weather and environment sensors developed in our laboratory in cooperation with Telenor and their Start IoT initiative: a pH sensor to the left and a humidity sensor to the right (picture from Telenor)



**Fig. 3** A snow sensor prototype developed at our laboratory

### 4.2.1 Mobile Sensing

Mobile sensing is about using sensors on mobile devices, such as smartphones, wearables, tablets and in-vehicle sensing devices, to obtain data of both personal and common interest. This enables tracking and analysis of dynamic information that can be used in applications of various types, for example traffic and road monitoring, environment monitoring, health monitoring and human behaviour studies [8, 21, 22].

Mobile phone sensing enables large-scale sensing using ubiquitous devices with high computational power and multiple embedded sensors. For example, available smartphone sensors include accelerometer, gyroscope, gravity, GPS, proximity sensor, light sensor, compass, and general sensors such as microphone and camera. Typically, sensors can register user activity (e.g. location, movements) and environmental data (e.g. light, sound, images) [42]. Additionally, external sensors (e.g. sensors for pollution and health monitoring) can be connected to the mobile device through its communication interfaces.

GPS data provides location and movement of people, and combined with accelerometer, the mode of transportation of a user (such as walking, running or using a vehicle) can be recognized [28]. Repeated journeys can also be detected using accelerometers and gyroscopes in smartphones, by tracking manoeuvres that form a unique signature of the route [29].

Continuous monitoring of noise and air pollution can detect environmental impacts of human activity, while other environmental data (such as temperature, weather and light conditions) can be used in nudges by for example suggesting “*a nice bicycle ride in lovely weather*”.

Many vehicles have a considerable number of sensors that provide data to systems that aim to improve a vehicle’s performance, monitor its operation and status of its components, and enhance the driving experience. In [1], in-vehicle sensors are classified according to their application domain, into (i) sensors for safety, (ii) sensors for component diagnostics, (iii) sensor for driver convenience, and (iv) sensors for environment monitoring. Solutions to access this data from private cars exist. Such adapters connect to the on board diagnostics (OBD) port in cars and can be used to collect data of interest from the cars.

For our purpose, sensors for monitoring the surrounding environment, is of specific interest. This can be used for providing services that alerts about hazards on roads or report about traffic, road and weather conditions. Such information can benefit both the driver, through on-board services, and be reported to third parties and thus benefit the community. For example, data from distance sensors, detecting the distance to the preceding vehicle, can be used as an indicator of the traffic congestion level. Analysis of images taken by in-vehicle cameras can provide information about road conditions, including water and snow on the road [1].

Mobile sensing is divided in two major classes related to the awareness and involvement of people providing sensed data. In *participatory sensing*, the user is directly involved in the sensing action, for example by taking a photo of a location or event. In *opportunistic sensing*, the user is not actively involved in the sensor activity on her or his mobile device [22].

### 4.2.2 Personal Sensing and Crowdsensing

Mobile sensing systems may collect data of both personal and common interest. *Personal sensing* focus on collecting data about daily life and activities of a single user, such as tracking the user's traveling habits, frequently visited locations, transportation choices and exercise routines. Personal sensing data are typically collected for the purpose of supporting the user and are not shared with others [22]. As described in Sect. 2.2, nudges must, to be effective, be personalised to the user's specific needs. Information collected through personal sensing is crucial for continuously detecting user activity and behaviour, and is thus important for the creation of a user profile.

Personal sensing data can also be used to detect and evaluate the effect of nudges on an individual level. By recording the user's activity over time, one might detect if the user change behaviour when nudged.

Crowdsensing is described as individuals with sensing and computing devices collectively sharing data and extracting information to measure and map phenomena of common interest. Aggregated information from a crowd can be used for managing societies or cities, and is useful for monitoring large-scale phenomenon that cannot easily be measured by a single individual [14]. Examples are traffic congestion monitoring and pollution level monitoring, which can be done if a large number of individuals continuously share traffic speed information and air quality information.

The same data can be of both personal and common interest. An example is the travel history of a person, which as personal data contributes to the user profile. It also has a common interest, since an aggregated travel history of a crowd can identify travel patterns of the crowd, and thus contribute to the planning of public transportation or cycling paths.

Privacy is obviously a concern when sharing data for the common good. In this example, the travel history of each individual should not be openly shared, but rather contribute to an aggregated travel history in a privacy preserving manner [5].

## 4.3 Aggregated Data

Aggregated data is lightly processed data. It is data where a larger set is combined into a more compact form. It could be seen as extracting the essence out of a data set for a particular use case. The reason to do this is usually to avoid the need to manage (and transfer) a large data set that is not necessary for the current usage. Another possible reason is that the whole set of data should not be made available. This could for example be based on privacy concerns.

Aggregated data are dynamic data. They are continuously updated with the latest fresh data available. They can be used to detect trends and they can be used as background data for more complex analysis.

Aggregated sensor data can be used to analyse actual user behaviour and the effects of executed efforts to influence this. Such sensor data includes traffic

measurements, counts of cyclists, walkers and runners, numbers of registered cars at parking lots, weather data, pollution data, user location, speed, context etc. (from mobile devices), and location, speed, number of passengers and context (e.g. route and timetable) of public transportation entities (e.g. from sensing IoT devices on vehicles).

#### **4.4 Crowdsourcing**

The term crowdsourcing, introduced in [19], is in Merriam-Webster<sup>4</sup> defined as

the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers.

In crowdsourcing systems, tasks are distributed online to people that can contribute to collectively solving the overall task. Crowdsourcing has been used for many different purposes, including collecting ideas and opinions, annotation (e.g. of images), collective creative tasks, and data collection and sharing. Examples are information sharing systems (such as Wikipedia, Yahoo! Answers and Flickr) and geographically related information where pollution or noise can be monitored by the crowd [25, 41], idea crowdsourcing for law-reform and idea evaluation [2], and user feedback used to plan and adapt public transportation [12].

Crowdsourcing is in many cases used for solving human intelligence tasks, which are tasks designed to tackle problems that are difficult to solve for a machine but easy for humans [27]. Such problem solving uncovers the wisdom of the crowd, which is derived from aggregating individual contributions and not from averaging them [7, 36].

Crowdsourced ideas, opinions and people's experiences concerning transportation are important for urban planning and development, where data from a potentially large crowd can contribute to support, for example, changes to the physical environment, such as new paths for walking and cycling or new public transportation services [39]. The combination of traditional crowdsourcing and crowdsensing is described in for example [27], providing services such as live traffic report about congestions.

#### **4.5 Static Data**

Static data represents non-frequently updated knowledge. This could be map data, historical traffic data, public statistical data, calendars, time tables (schedules), and so on.

---

<sup>4</sup><https://www.merriam-webster.com/>

Maps are an important part of planning transportation choices. Often different maps has to be combined. For example, Google maps can provide a rough estimate of different travel time depending on different routes, means of transportation, and specified time of traveling. If this is combined with local knowledge and local maps that includes updated information about roads, footpaths and so on, we should be able to do even better predictions.

The time tables (schedules) of public transportation are used when analysing different approaches for transportation for a given user. This could be combined with the calendar of the user. With this approach, the user has to do less interactions with the service and still be able to receive suggestions based on her og his needs for transportation (recognised by the planned events in the user calendar).

Other historical data are also of interest. For example, user experiences with previously suggested travel choices could be used to improve future suggestions. And historical weather and traffic data can be used to predict pollution levels if used together with current weather forecast.

## 5 Analyse

A variety of data mining and statistical tools can be used to extract information from the raw data collected by sense components.

### 5.1 *Methods for Analysis*

A simple form of analysis is to integrate two or more data sets to enhance precision or produce new knowledge. An example is to use the current location of the user provided by the personal smart phone, combined with the next event in her or his calendar, to prepare a notification about when to start traveling to be able to be at the event on time. The event provides a location and a start time. The current location and the event location can be used to calculate travel time. Based on the start time of the event, enough information is available to produce the notification.

The example above seems to be trivial, but only if the means of transportation is already chosen and it is easy to calculate the travel time. If this is not the case, a more complex analysis might be necessary.

When the goal is to use smart nudging to encourage green transportation choices, the analysis performed might involve a wide range of data combined with complex modelling of transportation options and user behaviour. As a consequence, the analysis might include several steps of processing with final minor tweaks or adjustments based on real-time updates of user context and external data (e.g. sensors).

For example, a pre-processed set of travel options to an event in the user's calendar can be the base for a nudging service using notifications. When a delayed bus is discovered and published as a real-time change of schedule in the system,

the different travel options have to be reorganised in a reprocessing step. To be able to discover changes of relevance to the travel options of the user, the notification service will, as a consequence of the pre-processed set of travel options, subscribe to events that might influence how the different options are rated. This is also a good fit for the proposed publish-subscribe system architecture.

The complexity of producing a good set of travel options for a user can be fairly high. The methods used to perform these analyses might be based on machine learning techniques.

## 5.2 *Back-End Processing*

To provide smart nudging of green transportation choices, the data available is analysed in a back-end system. This back-end processing is a continuously running process performing a wide range of analysis. The outcome of such tasks might be used as input to another task (see *Analyse* in Fig. 1). A task is either performed periodically or when new input is received. How to decide when sufficient input is available to start processing in a task and how to handle error and failures can for example be modelled after the SNOOP<sup>5</sup> approach [4]. Tasks acting on new fresh input are subscribed to its input from a publish-subscribe service.

Back-end processing tasks can perform simple data integration and analysis or more complex data mining or machine learning based analysis. Some back-end processing tasks might produce results that are ready to be used to inform and nudge the user directly, and some back-end processing tasks might produce a pre-processed result that need more processing at the back-end and/or at the edge (e.g. at the user's mobile device).

## 5.3 *Edge Processing*

We have two cases when edge-processing is performed. The first case is when the edge is a *sense* component. In this case the sense component is typically an IoT sensor or a mobile device (a smart phone). The second case is when the edge is an *inform and nudge* component. In this case the inform and nudge component is usually the user's smart phone.

The raw data collected at IoT sensors and mobile devices as sense components tend to be redundant and noisy. Sending the raw data to the back-end could consume unnecessarily large network bandwidth. Therefore data cleansing and redundant reduction should be carried out at the computing devices as close to the edges

---

<sup>5</sup>SNOOP is an adaptive middleware for privacy aware distributed computations that includes mechanisms to ensure progress and handle errors.

(i.e. the sensor or end-user devices) as possible. Furthermore, certain data analysis and aggregation tasks, such as situation awareness, are heavily dependent on spatial and temporal contexts. These tasks should also be carried out as close to the edges as possible.

On the other hand, the edge devices might be limited with resources for computation, network communication and electric power. Therefore edge processing must respect realistic limitations [31].

When performing inform and nudge services, a mobile device has a wide range of data available. Part of this data comes from the analysis tasks in the back-end system. This pre-processed information is an important part of smart nudging, but the final processing will happen at the edge. One reason for this is that the pre-processed data will be combined with local, and possible sensitive and private, data on the user's smart phone. This includes the user's calendar, current position, user profile, and current preferences. Another reason is to be able to provide current and up to date inform and nudge services in real-time.

## **6 Inform and Nudge**

The inform and nudge component handles interaction with the user. It provides personalized information to users both through informing about transportation alternatives and through persuasive messages intended to nudge them in the direction of sustainable transportation choices.

### ***6.1 Nudging Approaches***

Studies have shown that nudging works particularly well where there are immediate, or at least short-term, benefits for the individual [26]. One challenge is thus to identify such immediate or short-term benefits, obtain the necessary information and present them in a comprehensible way to users.

Nudging for green transportation choices, can have both short and long-term benefits. For example, in areas with much congestion, cycling to work may save both time and money. If more people choose the bicycle, long term benefits will be less air pollution, carbon dioxide emissions, reduced traffic and thereby less congestion. Additionally, using the bicycle instead of driving, will represent a health benefit for the user.

It is claimed that there is no neutral way to present choices, and that online users are influenced by nudging of one kind or another, intended or unintended [37, 40]. It is therefore important to understand nudging principles and effects, to intentionally choose or reduce nudging.



Concerning system requirements, the work of [30] describes content and functionality that may be required in persuasive systems. System features are categorised as providing (i) primary task, (ii) dialogue, (iii) system credibility, or (iv) social support, and the paper presents design principles in each category. *Primary tasks* of a system may for example be simplification by reducing complex behaviour into simple tasks that lead to the desired behaviour. It also includes personalisation and tailoring to the user's potential needs, interests, personality and context. *Dialog* support includes principles for human-computer interaction that motivate the user (for example praise, rewards, reminders and suggestions). *System credibility* principles describe how to design a system so that it is more credible and thus more persuasive (including trustworthiness, authority, expertise and verifiability). *Social support* principles describe how to design the system so that it motivates users by leveraging social influence (including social comparison, cooperation and competition).

## 6.2 Personal Recommendations

Nudging is about providing tailored information to the user that is user relevant according to the user's profile and current context, but also takes into consideration the nudging goal. That is, information presented to users must both be relevant and presented so that the nudge intention is supported. Also, there is evidence of great heterogeneity in people's responses to the same influencing factor. This means that people may react differently to the same type of nudge depending on their context, role or personality [3, 23]. Personalisation is thus an important factor in nudging systems.

Nudges for a user must evolve over time and change depending on the user's current situation. It is not much use in informing the user over and over again that cycling to work is less expensive than driving. But teasing the user with the nice conditions in the cycling path may be useful.

Nudging in a familiar place (ex. the user's home town) may take different forms than nudging in an unfamiliar place. When visiting a new city, knowledge about public transportation may be lacking, and as a consequence taking a taxi is often a preferred alternative. However, if the user is provided with relevant information (and possibly help with buying a ticket), the user may choose public transportation. At home, the possibilities of relaxing with coffee and newspaper until the real-time system informs that it is time to leave, may be a reason for choosing the bus.

If a user is driving to work, it is normally not useful to try to nudge the same person to walk home. Nudging in this case, may be towards choosing the best route and/or departure time (to avoid traffic jams and thereby reduce CO<sub>2</sub> emission). If using the bicycle or walking to work, it might be that no nudge is needed for the return. Or the user can receive a recommendation for a particularly pleasant route home.

In [6], a recommendation system presents persuasive messages to commuters using a route planning application. The aim is to nudge users towards following routes that are environmentally friendly with lower CO<sub>2</sub> emissions than those they usually take. The system includes a user profile (containing the users' travel history, traveling habits obtained through a questionnaire and logged persuasion attempts), eight context probes (gathering information about available routes, emission, peer behaviour and weather), and a pool of messages that can be used for persuading the users.

### 6.3 *Interacting with the User*

No single optimal approach to interact with the user exists. The interaction should happen when the user can be encouraged to choose a more environmental friendly transportation option, and ideally without the user asking for it. This can, for example, be a notification on the personal smart phone. The timing of the notification can be important. It should not come too late, so the user is unable to choose the more environmental friendly option, and it should not come too early, so the notification is forgotten or drowned by other notifications before it is time to act. The challenge here is that *too late* and *too early* is not necessarily the same for all users. This has to be personalised based on the typical behaviour of the user. Some of it can be learned by the system over time, and some of it has to be provided by the user in explicit interactions with the system.

The personal smart phone is the obvious candidate for interacting with the user. As exemplified above, notifications on the screen, possibly accompanied with sound, is one approach to inform and nudge the user. Applications providing map services, city guides and public transportation time tables, are examples of applications that can be tailored to include smart nudging for green transportation choices. Also a day-planner or a calendar like application can include services that will nudge the user towards more environmental friendly transportation choices. The goal should be to use several of these approaches, and adapt how they are used based on what matches each user best. This might also change over time. A user that has been nudged towards an environmental friendly form of transportation for a while might not need detailed information about each step of the trip after a while. Then, maybe a simple notification when it is time to leave home is enough. Another example is failed attempts to nudge the user. The system should adapt to new nudging approaches if the current attempts fails.

We believe that the user's smart phone is the most important device to use when interacting with the user. However, a smart nudging approach could combine nudging through other devices. Examples include voice controlled smart speakers like Amazon Echo and Google Home, public displays with personalised content [10], and real-time updated public time-table screens.

## 7 Summary

The knowledge presented in the text above is based on literature studies, cooperation with partners, and our early IoT experiments.

The main cooperating partners have been *Troms County Council* (regional transport authority for public transport in Troms County), *Tromsø Municipality* (representing the local community), the *Norwegian Public Roads Administration*, *Telenor* (telecom company), local software companies (*mPower*, developer of time-table and travel apps, and *Plus Point*, a game company with an IoT interest), *Rambøll* (an engineering, design and consultancy company), and other departments at the UiT The Arctic University of Norway (*the Department of Philosophy* and *the Business School*).

The IoT experiences originated in earlier research activities in our research group, but is currently focused towards the smart lab activities at the department, and specific towards the NUDGE project initiative. In the NUDGE project initiative, experiments with smart nudging for green transportation choices are currently performed.

Telenor has through their IoT initiative, Telenor Start IoT, created an infrastructure for experimenting with IoT technology. Telenor Start IoT consists of the IoT innovation network and the Telenor AI lab. Telenor has built and operates an IoT Innovation Network based on Low-Power Wide-Area (LPWA) technologies in selected geographical locations across Norway to facilitate innovation around this network. One of the locations is Tromsø. The IoT Innovation Network and its infrastructure has been operational from early 2017. The Telenor AI lab will drive innovation and research in the fields of big data analytics and AI, through access to real data and problems and close collaboration between industry, research institutes, academia and startups. The more complex data analyses could be facilitated by the Telenor AI lab.

In early 2017 we implemented, in cooperation with Telenor, an IoT course for master level students. In the course, *Inf-3910-3 IoT services with LoRaWAN network and compatible embedded devices and sensors*, 36 students participated in IoT related experiments. At the end of the course, the students submitted 20 projects for evaluation. All projects included at least (i) one IoT sensor communicating using LoRaWAN, (ii) a data aggregation and analytics part, and (iii) a front end presenting the analysed data. The projects represented a wide range of different types of applications, including weather stations, smart waste bins, golf green monitoring (using both LoRaWAN and mesh networks), greenhouse monitoring, safety tracking of cars, epilepsy detection and alarm, wildlife camera, machine learning based activity classification on microcontroller, smart Polar Fox feeding station, air pollution monitoring, animal trap with notification, passing objects (people) counter, and a kayak safety system. The course is also given in 2018.

**Acknowledgements** We would like to thank all people involved in the NUDGE project and our smart lab activities. Thank you to Arne Much-Ellingsen from Telenor for providing insight and a long time collaboration with *all things mobile*. Thank you to the students Øysten Tveito,

Thomas Holden, and Pontus Aurdal for experimenting and documenting *all things IoT* at our laboratorium. Thank you to Telenor for long time cooperation and their support in producing the experiments needed to produce this text. Thank you to the technical staff at our department, Ken-Arne Jensen, Jon Ivar Kristiansen, Kai-Even Nilssen, and Maria Wulff Hauglann, for facilitating our laboratorium and experiments. We also have to thank Michael Morreau and Erik Lundestad at the Department of Philosophy for interesting discussions regarding nudging, crowdsourcing, and much more. And finally we would like to thank you all students at the Inf-3910-3 course in Spring 2017 and Spring 2018 for experimenting with microcontrollers, sensors, LoRaWAN network, and back-end and front-end systems.

## References

1. Abdelhamid, S., S.Hassanein, H., Takahara, G.: Vehicle as a mobile sensor. In: E.M. Shakshuki (ed.) Proceedings of the 9th International Conference on Future Networks and Communications (FNC-2014), *Procedia Computer Science*, vol. 34, pp. 286–295. Elsevier (2014). <https://doi.org/10.1016/j.procs.2014.07.025>
2. Aitamurto, T., Landemore, H., Lee, D., Goel, A.: Crowdsourced Off-Road Traffic Law Experiment in Finland: Report About Idea Crowdsourcing. No. 1/2014 in Publication of The Committee for the Future. Parliament of Finland (2014)
3. Anagnostopoulou, E., Magoutas, B., Bothos, E., Schrammel, J., Orji, R., Mentzas, G.: Exploring the links between persuasion, personality and mobility types in personalized mobility applications. In: P.W. de Vries, H. Oinas-Kukkonen, L. Siemons, N.B. de Jong, L. van Gemert-Pijnen (eds.) *Persuasive Technology: Development and Implementation of Personalized Technologies to Change Attitudes and Behaviors*, Proceedings of the 12th International Conference on Persuasive Technology (PERSUASIVE 2017), *Lecture Notes in Computer Science*, vol. 10171, pp. 107–118. Springer-Verlag, Amsterdam, The Netherlands (2017)
4. Andersen, A.: SNOOP: Privacy preserving middleware for secure multi-party computations. In: F.M. Costa, A. Andersen (eds.) Proceedings of the 13th Workshop on Adaptive and Reflective Middleware (ARM 2014). ACM, Bordeaux, France (2014)
5. Andersen, A., Karlsen, R.: Privacy preserving personalization in complex ecosystems. In: C. Linnhoff-Popien, R. Schneider, M. Zaddach (eds.) *Digital Marketplaces Unleashed*, pp. 247–261. Springer-Verlag (2017). <https://doi.org/10.1007/978-3-662-49275-8>
6. Bothos, E., Apostolou, D., Mentzas, G.: A recommender for persuasive messages in route planning applications. In: Proceedings of the 7th International Conference on Information, Intelligence, Systems and Applications (IISA 2016). IEEE (2016). <https://doi.org/10.1109/IISA.2016.7785399>
7. Brabham, D.C.: Crowdsourcing as a model for problem solving: An introduction and cases. *The International Journal of Research into New Media Technologies* **14**(1), 75–90 (2008). <https://doi.org/10.1177/1354856507084420>
8. Cao, P.Y., Li, G., Chen, G., Chen, B.: Mobile data collection frameworks: A survey. In: Proceedings of the 2015 Workshop on Mobile Big Data (Mobidata'15), pp. 25–30. ACM, Hangzhou, China (2015). <https://doi.org/10.1145/2757384.2757396>
9. Commission of the European Communities: Towards a new culture for urban mobility. Green Paper COM(2007) 551 final, European Commission, Brussels (2007)
10. Davies, N., Langheinrich, M., Jose, R., Schmidt, A.: Open display networks: A communications medium for the 21st century **45**(5), 58–64 (2012). <https://doi.org/10.1109/MC.2012.114>
11. EPA: Greenhouse gas emissions from a typical passenger vehicle. Questions and Answers EPA-420-F-14-040a, United States Environmental Protection Agency, Office of Transportation and Air Quality (2014)

12. Filippi, F., Fusco, G., Nanni, U.: User empowerment and advanced public transport solutions. *Procedia – Social and Behavioral Sciences* **87**, 3–17 (2013)
13. Fogg, B.J.: Persuasive technology: using computers to change what we think and do. *Ubiquity* (5), 89–120 (2002)
14. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine* **49**(11), 32–39 (2011). <https://doi.org/10.1109/MCOM.2011.6069707>
15. Gao, M., Liu, K., Wu, Z.: Personalisation in web computing and informatics: Theories, techniques, applications, and future research. *Information Systems Frontiers* **12**(5), 607–629 (2010). <https://doi.org/10.1007/s10796-009-9199-3>
16. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In: P. Brusilovsky, A. Kobsa, W. Nejdl (eds.) *The Adaptive Web, Lecture Notes in Computer Science*, vol. 4321, pp. 54–89. Springer-Verlag (2007). [https://doi.org/10.1007/978-3-540-72079-9\\_2](https://doi.org/10.1007/978-3-540-72079-9_2)
17. Ghorab, M., Zhou, D., OConnor, A., Wade, V.: Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction* **23**(4), 381–443 (2013). <https://doi.org/10.1007/s11257-012-9124-1>
18. Hansen, P.G., Jespersen, A.M.: Nudge and the manipulation of choice. *European Journal of Risk Regulation* **4**(1), 3–28 (2013)
19. Howe, J.: The rise of crowdsourcing. *Wired* **14**(6), 1–5 (2006)
20. Issamy, V., Mallet, V., Nguyen, K., Raverdy, P.G., Rebhi, F., Ventura, R.: Dos and don'ts in mobile phone sensing middleware. In: Proceedings of the 17th International Middleware Conference (Middleware '16), pp. 1–13. ACM, Trento, Italy (2016). <https://doi.org/10.1145/2988336.2988353>
21. Khan, W.Z., Xiang, Y., Aalsalem, M.Y., Arshad, Q.: Mobile phone sensing systems: A survey. *IEEE Communications Surveys & Tutorials* **15**(1), 402–427 (2013). <https://doi.org/10.1109/SURV.2012.031412.00077>
22. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. *IEEE Communications Magazine* **48**(9), 141–150 (2010)
23. Lehner, M., Mont, O., Heiskanen, E.: Nudging - a promising tool for sustainable consumption behaviour? *Journal of Cleaner Production* **134**(Part A), 166–177 (2015). <https://doi.org/10.1016/j.jclepro.2015.11.086>
24. LoRa Alliance: A technical overview of LoRa and LoRaWAN. Tech. rep., LoRa Alliance Technical Marketing Group (2015)
25. Luz, N., Silva, N., Novais, P.: A survey of task-oriented crowdsourcing. *Artificial Intelligence Review* **44**(2), 187–213 (2015). <https://doi.org/10.1007/s10462-014-9423-5>
26. Mont, O., Lehner, M., Heiskanen, E.: Nudging: A tool for sustainable behaviour? Report 6643, The Swedish Environmental Protection Agency (Naturvårdsverket) (2014)
27. Mrazovic, P., Matskin, M.: MobiCS: Mobile platform for combining crowdsourcing and participatory sensing. In: Proceedings of the 39th Annual IEEE Computer Software and Applications Conference Workshops (COMPSACW 2015), pp. 553–562. IEEE, Taichung, Taiwan (2015). <https://doi.org/10.1109/COMPSAC.2015.26>
28. Mun, M., Sasank Reddy, K.S., Yau, N., Jeff Burke, D.E., Hansen, M., Howard, E., West, R., Boda, P.: PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In: Proceedings of MobiSys'09, pp. 55–68. ACM (2009)
29. Nawaz, S., Efstratiou, C., Mascolo, C.: Smart sensing systems for the daily drive. *IEEE Pervasive computing* **15**(1), 39–43 (2016). <https://doi.org/10.1109/MPRV.2016.22>
30. Oinas-Kukkonen, H., Harjumaa, M.: A systematic framework for designing and evaluating persuasive systems. In: T. Ploug, P. Hasle, H. Oinas-Kukkonen (eds.) Proceedings of the Third International Conference on Persuasive Technology (PERSUASIVE 2008), *Lecture Notes in Computer Science*, vol. 5033, pp. 164–176. Springer-Verlag, Oulu, Finland (2008). [https://doi.org/10.1007/978-3-540-68504-3\\_15](https://doi.org/10.1007/978-3-540-68504-3_15)
31. Perera, C., Qin, Y., Estrella, J.C., Reiff-Marganiec, S., Vasilakos, A.V.: Fog computing for sustainable smart cities: a survey. *ACM Computing Surveys* **50**(3), 1–43 (2017)

32. Razzaque, M.A., Milojevic-Jevric, M., Palade, A., Clarke, S.: Middleware for internet of things: A survey. *IEEE Internet of Things Journal* **3**(1), 70–95 (2016)
33. Samaranyake, S., Glaser, S., Holstius, D., Monteil, J.: Real-time estimation of pollution emissions and dispersion from highway traffic. *Computer-Aided Civil and Infrastructure Engineering* **29**(7), 546–558 (2014)
34. Sanchez-Iborra, R., Cano, M.D.: State of the art in LP-WAN solutions for industrial IoT services. *Sensors* **16**(5) (2016). <https://doi.org/10.3390/s16050708>. 708
35. Storz, O., Friday, A., Davies, N.: Supporting content scheduling on situated public displays. *Computers & Graphics* **30**(5), 681–691 (2006). <https://doi.org/10.1016/j.cag.2006.07.002>
36. Surowiecki, J.: *The Wisdom of Crowds*. Anchor Books (2008)
37. Thaler, R.H., Sunstein, C.R.: *Nudge: Improving Decisions about Health, Wealth, and Happiness*. Yale University Press (2008)
38. Uckelmann, D., Harrison, M., Michahelles, F.: *An Architectural Approach Towards the Future Internet of Things*, chap. 1, pp. 1–24. Springer-Verlag (2011). <https://doi.org/10.1007/978-3-642-19157-2>
39. Wang, X., Zheng, X., Zhang, Q., Wang, T., Shen, D.: Crowdsourcing in ITS: The state of the work and the networking. *IEEE Transactions on Intelligent Transportation Systems* **17**(6), 1596–1605 (2016)
40. Weinmann, M., Schneider, C., vom Brocke, J.: Digital nudging. *Business & Information Systems Engineering* **58**(6), 433–436 (2016). <https://doi.org/10.1007/s12599-016-0453-1>
41. Yuen, M.C., King, I., Leung, K.S.: A survey of crowdsourcing systems. In: *Proceedings of IEEE International Conference on Privacy, Security, Risk, and Trust (PASSAT)*, and *IEEE International Conference on Social Computing (SocialCom)*, p. 766–773. IEEE, Boston, MA, USA (2011). <https://doi.org/10.1109/PASSAT/SocialCom.2011.203>
42. Zamora, W., Calafate, C.T., Cano, J.C., Manzoni, P.: A survey on smartphone-based crowd-sensing solutions. *Mobile Information Systems* **2016** (2016). <https://doi.org/10.1155/2016/9681842>

# Energy Harvesting in Smart Building Sensing: Overview and a Proof-of-Concept Study



Aristotelis Kollias, Colton Begert, and Ioanis Nikolaidis

**Abstract** Modern “smart” buildings require a plethora of sensors to be installed at various locations during the construction phase. Wiring costs and limited flexibility of installation make wired installations less attractive. An alternative, flexible, approach is to introduce wireless sensors and endow them with ways to harvest energy from the environment such that they attain the same “zero cost” of maintenance as their wired counterparts. The chapter reviews the sensing needs of smart buildings, and the related merits of energy harvesting to power embedded wireless sensor nodes. A proof-of-concept device exploiting thermoelectric harvesting is designed, built and tested to demonstrate how today’s wireless sensing devices enable sustained continuous operation with minor energy harvesting requirements. In multi-hop environments, the underlying optimization problems are described and simple strategies that forego the solution of the hard computation problems but appear effective are outlined.

## 1 Introduction

Wireless sensor networks (WSNs) are used in smart city applications such as structure monitoring and maintenance, to monitor human activity, to help in disaster management from, e.g., fires and earthquakes, and, in general, to collect data for scientific and business purposes, [1]. In the construction industry, sensors can be used to alert of dangerous situations, in cases where the infrastructure is critically compromised and to monitor wear and the “health” of buildings in general [2–4]. For example, humidity sensors inside wall structures can provide advance warning of excess humidity which could lead to toxic mold growth, and strain dynamometers sensors can be used to determine the response of the building during earthquakes. We remark that in most cases, the phenomena under observation are slowly varying,

---

A. Kollias · C. Begert · I. Nikolaidis (✉)  
Department of Computing Science, University of Alberta, Edmonton, AB, Canada  
e-mail: [aristote@ualberta.ca](mailto:aristote@ualberta.ca); [begert@ualberta.ca](mailto:begert@ualberta.ca); [nikolaidis@ualberta.ca](mailto:nikolaidis@ualberta.ca)

and hence the sampling rate is not particularly high. For example, ambient relative humidity measurements more than once every minute is an overkill.

A cost of adopting WSN technologies in a smart city is their upfront, but unavoidable, installation costs. Another component of the costs associated with WSNs is their long-term maintenance costs. In this chapter, we are motivated by the desire to minimize the additional long-term maintenance costs of WSNs. While, on one hand, capital costs for sensing devices are gradually decreasing, labor costs remain a considerable hindrance to sustainable long-term deployments to the extent that labor is necessary to perform tasks such as locating the WSN nodes and replacing their batteries. A powerful idea that will result in significant reduction of maintenance costs is that of designing sensing systems able to power themselves over long periods of time.

Specifically, we review the options currently available to support the multi-decade autonomous operation of the WSN. The main challenge is to provide energy for self-sufficient operation of WSN nodes. We subsequently narrow our options to thermoelectric harvesting because of the ability to observe temperature gradients due to heat transfer phenomena even in locations where, for example, photovoltaic harvesting would not have been possible due to location and/or orientation. Specifically, we are advocating the partial capture of the heat loss through walls to power WSN nodes. The main challenge is how to master the captured energy in terms of deciding how much data can be, realistically, harvested, but more directly, how much data can be transmitted based on the harvested energy. A common problem in energy harvesting research is the use of synthetic data sets. We avoid this problem by basing our work on actual measurements of heat flow through exterior walls.

We first demonstrate that, in absolute numbers, the proposed harvesting can provide sufficient energy for building applications but we also note the, inescapable, seasonality introduced by relying on thermoelectric harvesting. The particular challenge is pronounced when the WSN nodes are performing data forwarding for other nodes, i.e., when our WSN is deployed as a multi-hop network. There, routing decisions have to be adjusted to the energy harvesting potential. Simple sources of information, such as the difference of indoor vs. outdoor temperatures can be used as reliable proxies for the energy harvesting opportunities, and simple prediction mechanisms can be incorporated to make routing decisions without having to explicitly communicate between nodes, sending messages that are energy-costly in their own right (and hence eat away from the total energy available to the nodes).

It is envisioned that the building/network designer, knowing the use of a building (and possibly having a resident behavior model in mind) and the weather in the particular area, can then make an informed decision on the amount of data that such nodes could collect and forward. We make a number of observations on the time-varying nature of thermoelectric harvesting and conjecture on the links of its performance to factors such as the occupant behavior. We are encouraged by the fact that simple prediction schemes can be very accurate in terms of the near-future decisions for routing. Additionally, the use of a temporary energy storage (such as an (ultra)capacitor) is not always effective because, frequently, the topology of the



network is such that nodes with ample energy reserves are not (cannot) be used due to the routing decisions taken, e.g., when they are at the fringes of the network and not close to the sink (data collecting) node.

Section 2 provides a quick review of energy harvesting with building applications in mind, while Sect. 3 reviews a simple WSN node design with thermoelectric harvesting as its only energy source. Moreover, Sect. 3 sets the expectations of what kind of performance, in absolute numbers, one can expect from such a design using common off-the-shelf components and without optimizing the node design in any way. Section 4 summarizes the lessons we learned from investigating possible routing strategies over multi-hop WSN networks composed of nodes using thermoelectric harvesting. The relatively static topology, and the repeated seasonal trends allow us to develop simple strategies to adjust routing at various time scales. Finally, Sect. 5 provides some concluding remarks and directions for future research.

## 2 Energy Harvesting in Building Environments

Today's building practices include a number of subsystems to provide comfortable living environments. Among them are the distribution of heat, water, and electricity as well as the renewal of the indoor air. Each such subsystem is responsible for the transformation or transmission of energy in one form or another. Additionally, indoor living environments are characterized by diverse occupant dynamics, ranging, at one extreme, from almost completely unoccupied spaces (e.g., storage areas) to, at the other extreme, busy corridors in commercial and public buildings. With the increased interest in using sensors for data collection, it is reasonable to ask whether an entire sensor network could exploit, for its own energy needs, the energy already present in those existing subsystems.

A quick answer is to power all sensor nodes from the existing electricity distribution system, i.e., to "wire" them. Apart from the fact that wiring them obviates, for the most part, the communication problem (they no longer need to be wireless), there are several practical reasons why wiring is not a convenient solution. Currently, existing power distribution systems in buildings are designed for everyday residential high voltage AC distribution. Powering sensors typically requires a separate low voltage DC distribution wiring subsystem. While there are strong indications [5], that a move to a DC-only bus is imminent and could be efficient, the fact remains that the two systems will likely co-exist in parallel for the foreseeable future. Unless automation techniques, such as robotic building assembly becomes commonplace, wiring is expensive in terms of labor cost and materials (properly shielded and insulated copper wires). Power distribution wiring also results in a structure which is more complex than stand-alone power sources, and could be prone to faults (accidental puncture/cutting of the wires, problems with single points of failure, need for proper filtering of RF noise over long runs of wire, etc.). In short, the construction of a properly engineered low voltage power distribution system needs to be tailored to the exact environment and pre-planned

carefully. Furthermore, wired solutions are not amenable to retrofits whereby an existing structure is to be endowed with (new) sensors, at new locations, as wiring them would imply added cost and complexity. An additional wrinkle to the AC and DC co-existence is that strict regulations by national electricity codes requires at least some form of an authorized (and usually specialized) electrician to approve the design and inspect the installation. In other words, wiring is not done in an “ad-hoc” basis.

Based on the above challenges, it is tempting to consider the two remaining solutions: non-rechargeable batteries or autonomously powered, via energy harvesting, sensors. The cost of labor for replacing batteries is the main factor to reject the first option. Indeed, even if one were to propose that, with suitable energy-efficient operation, the battery-based operation could be extended to many years, there still remains the issue of (a) scale – as a typical building could include hundreds, if not thousands, of sensors, and (b) location – as it could be difficult or impossible to access some sensors to replace their batteries (e.g., sensors inside ductwork).

## ***2.1 Energy Harvesting Opportunities and Efficiencies***

Self-sustaining wireless sensor node operation is an ideal objective. In reality, even in nodes exploiting energy harvesting, energy storage is typically also used. The energy storage can be in a super-capacitor and/or a rechargeable battery. A quick review of four existing energy harvesting opportunities is as follows:

- Photovoltaic (PV) harvesters are a key technology for renewable energy and their efficiency is continuously improving with many competing, evolving, materials used for this purpose. Their reported efficiencies vary and the current state of the art is reaching an efficiency of 40%. Yet, these numbers are derived under ideal conditions, while units acquired in the market do not exceed an average of 20% efficiency. Even then, the main limiting factor with PV is their relative placement with respect to potential light sources. Indoor spaces, with little or no natural light, are not continuously illuminated. Inaccessible locations (as in the ductwork example) are simply too dark to provide any PV harvesting opportunities. Additionally, the need for a PV element to be exposed implies a need to blend the PV elements into the indoor aesthetics of a building, while the use of the space could also occlude the path of light to the PV element (when e.g., furniture is moved in front of it). In summary, PV harvesting present numerous challenges related to their placement in indoor environments.
- Piezoelectric (PE) harvesters exploit the piezoelectric property of various materials that transforms the mechanical stress causing deformation of the piezoelectric material into a short-lived electrical current. Essentially what is exploited is vibrations as they result in a sequence of excitations of the system from its resting state. The particularly complicated facet of piezoelectric harvesters is that, due to the many varieties of materials and mechanical ways to attach the PE

material (usually as part of a cantilever design) as well as the different efficiencies depending on whether the vibrations are occurring at the resonant frequencies or at off-resonance frequencies, the question of how to best characterize their efficiency is, as of yet, not a settled matter [6]. In an indoor environment, using PE harvesters is also limited to situations where occupants can, predictably, cause vibrations, as in floor tiles, doors and windows, and light switches (rockers). The challenge of floor tiles is the overall cost of installation especially if all tiles are to be PE capable. Additionally, PE harvesting using tiles, doors and light switches is idiosyncratic and highly dependent on the use patterns of the space.

- Thermoelectric (TE) harvesters exploit the Seebeck effect according to which, certain materials, develop an electric potential across a temperature gradient in proportion to the temperature gradient. The advantage of TE harvesting is that temperature gradients develop in a variety of settings during the normal operation of a building and the activity of their occupants. For example, temperature differences are intentionally caused by heating and cooling systems to enhance the comfort of the occupants. Other temperature differences are caused by the daily weather patterns and sunlight heating building surfaces. A challenge of TE harvesting is the relatively low power output compared to PV harvesters (discussed later in Sect. 3.3). Moreover, since large temperature gradients are not always available, a consideration with TE harvesters is the minimum temperature difference that can result in harvestable current. Currently, commercially available TE harvesters exist that are able to harvest even at a temperature difference of 1 °C.

In addition to the three main categories, there exist electromagnetic harvesters that exploit magnet/coil combinations bearing similarities to PE harvesters (e.g., in tiles, switches, etc.) and inherit the same limitations. Small scale wind turbines are also magnet/coil systems but they need to be exposed to air flow which makes them both aesthetically problematic and unsuitable for installation within structures where airflow does not exist.

- Radio Frequency (RF) harvesting is at its infancy and is dependent on the existence of sufficiently strong wireless transmissions in the vicinity of the harvester, thus also making the harvesting dependent on the wireless transmission patterns and, namely, on the exact antenna design, geometry, and placement. The subtle issue with RF energy harvesting is that, usually, in order to even be feasible, a special transmitter/emitter of RF energy must be built with the specific purpose of charging nearby, possibly multiple, sensors. In other words, the systems are not self-sustaining, but rather recipients of RF energy requiring a powered, and suitably engineered, transmitter. Furthermore, a truly high efficiency RF harvesting system requires beamforming which necessitates at a very minimum a specialized protocol for communication between emitter and receiver of energy to communicate the channel transfer function. Clearly, this places the additional requirement that the RF emitting node has plentiful access to energy.

Currently, the only subcategory of RF harvesting possible without special equipment is *ambient* RF energy harvesting, but results in only minuscule harvested output [7] unless one is lucky to be in close proximity to a strong commercial transmitter, e.g., for TV broadcast. Finally, recent developments have allowed standards-compliant emissions to be generated as backscatter of a different physical layer protocol, in what is called “inter-scatter” and specifically producing IEEE 802.11 beacons via Bluetooth Low Energy backscatter [8]. The process avoids the energy-demanding local oscillator signal generation, but is limited to harvesting minimal amount of energy and providing functionality equivalent to a passive RFID tag, i.e., only simple message (beacon) transmissions and extremely limited sensor-side complexity.

## 2.2 *Designing for Multi-decade Sustainable Operation*

Our presentation is primary motivated by the need for almost zero maintenance sensor networks, that will be able to operate independently and autonomously in hard-to-reach or unreachable locations. Networks like that will be able to be installed inside buildings, or in remote infrastructure and generally in difficult to maintain spots. Their primary job will be to monitor and transmit information, dealing with data collection needed for applications such as: monitoring the operational efficiency of a system, collecting usage statistics, reporting critical information e.g., structure fatigue, etc. Given the ubiquitous potential for TE harvesting, whose operation is a fundamental consequence of thermodynamics, we conjecture that it will become a de-facto means to power such sensors, as long as the power needs of the sensing platforms remain limited (a matter that we address in the next section).

The use of TE platforms for intermittent operation has been exploited in systems such as DoubleDip [9], whose application is that of monitoring the flow of water in pipes. The water flow causes, for a suitably mounted TE element, a transient temperature gradient which acts both as the source for energy harvesting as well as the actual event detection mechanism because DoubleDip’s application is specifically to monitor flow events. Due to its intermittent operation, there are two basic options: to immediately transmit the data (event) or to collect and aggregate such events until sufficient energy is available. Three constraints arise from designs such as DoubleDip. First is the limitation on the potential deployment locations; near water pipes in the case of DoubleDip. Secondly, the event-triggered nature of DoubleDip, fundamentally limits the system from being used for tasks involving regular continuous sampling. A third constraint, albeit intended to achieve a higher level of functionality than designed for, is the lack of mechanisms that would allow for multi-hop routing.

Attempts to expand and bridge the intermittent event-triggered operation to almost-continuous operation have been studied, e.g. in [10]. The reason that one cannot call for a truly continuous (hence the “almost-continuous” qualification) operation is that, under specific scenarios, it is always possible that the energy of all

nodes is depleted, and no energy harvesting opportunities exist. Such scenarios can persist across time, leading to a “dead” sensor platform. Therefore, the design space that needs to be explored for multi-decade sustainable operation should consider designs that can harvest energy even in the worst case scenarios, and even if such opportunities are sporadic. Moreover, even the best design cannot outlast the hardware components’ lifetime. Unfortunately, the planned obsolescence strategies in the consumer electronics sector distort our view of what is indeed possible. Additionally, the mean-time-between-failures (MTBF) is highly dependent on the operating conditions. In the remainder of this chapter, we assume that multi-decade hardware reliability is indeed possible, if only for a particular cost. A study of the hardware component failures is outside the scope of our presentation.

A strategy for TE harvester placement that is both flexible and quite capable for almost-continuous operation has been proposed in [11, 12]. It proposes that in-wall TE harvesters can both be “hidden” from view and be adequate to power a sensor platform embedded within walls. The concept relies on the observation that exterior wall structures in buildings are almost guaranteed to have a temperature gradient develop across them. This is both because the interior space, if inhabited, is heated (or cooled) for reasons of resident comfort, and because of exterior temperatures and other effects, e.g., convection, caused by local weather phenomena. Additionally, one could remark that the thermal mass of the interior structure of a building can store (and then radiate) heat, thus contributing to the temperature gradient on the “interface” wall between interior and exterior. Clearly, such a TE harvesting strategy results in variable output throughout the day and across seasons of the year. In the next section we outline an experiment which provides a convincing argument about the efficacy of the proposed approach. However, the broader observation is the following: TE harvesting sensors need to be placed in particular locations to attain significant power output. While DoubleDip provided a limited array of options for TE use, the work in [11, 12] expanded TE harvesting applicability to more locations. We would remiss not to add the opportunities that exist for TE harvesting from the waste heat of various engines, which further contribute to the TE harvesting “ecosystem” within buildings. Nevertheless, the existence and operation of such engines is usually not guaranteed, while a temperature gradient across exterior walls is essentially a given.

### 3 Sensor Node Design

We consider the standard model for wireless sensor network (WSN) nodes, i.e., as consisting of a transceiver, the sensor, a microcontroller and an energy source, usually a battery. The main objective is to replace the battery by an energy harvester, suitable power regulation circuitry, and a storage element (such as a supercapacitor). PV harvesters have been studied extensively in previous works e.g., [13]. Buildings with good illumination can use PV energy to power sensor modules. Instead, we consider the case of an extreme environment, such as the Canadian North, where: (a)

the potential for PV energy can be limited due to suboptimal placement, and because of long nights, as is the case at latitudes of northern continental climates, and, (b), during winter the indoor to outdoor temperature difference can reach as much as 60 °C creating unparalleled opportunities for TE harvesting. For our example data, we rely on data collected from an actual inhabited apartment complex in Fort McMurray, Alberta, Canada.

Placing TE powered sensors inside walls serves the application of powering sensors that monitor the wall and building behaviour. In climates like the one considered in our study, extreme weather conditions can cause events important to the integrity of a building, e.g., breakage of water pipes, more frequently occurring than in moderate climates. The ability to embed sensors in inaccessible locations that autonomously operate for several decades (i.e., as long as the building lasts or until a major renovation), monitoring for such events, can currently only be supported using energy harvesting.

The question of whether to use batteries as the harvester storage is answered by the fact that batteries, over several recharge cycles, slowly degrade, especially in adverse environments [14]. We therefore opt to use (super)capacitors as energy storage. While they cannot store as much energy as a battery, capacitors are ideal for harvesting applications, as they do not have a limited number of charge/discharge cycles. It has been reported, [15], that a node that operates with a battery source has a 1–2 years of lifetime, due to the amount of charge/discharge cycles, while the same system, when operating using an ultracapacitor, has a theoretical lifetime of 20 years. Finally, to keep the size of the nodes small, we assume that the capacitors used provide typical storage in order of a few Farads.

Another factor we consider is the space (surface area) needed for a TE harvester. We will consider designs whose surface area is comparable with that of the sensor platform circuit boards *inclusive* of the footprint required for the batteries they are to replace. The range of possible values is wide, and depends on how the components are stacked, but we consider as reasonable any surface area less than 25 cm<sup>2</sup> (such as a 5 × 5 cm square). Hence, when making comparisons between TE with PV harvesters, we do so under the assumption of the same (and less than 25 cm<sup>2</sup>) surface area.

### ***3.1 The Energy Requirements of Modern Ultra-low Power RF***

Currently, wireless sensor nodes vary in terms of their power needs. Even defining what constitutes a low-power platform is a matter open for discussion insofar one has to identify what features of a platform are essential to consider it a “legitimate” sensor platform. While little debate exists that the devices should be capable of wireless communication, the extent, and type, of processing performed on them varies greatly. For example, the speed of processing (tied usually to the highest processor clock rate) and on-board RAM are limiting factors to the extent that

complicated algorithmic tasks can be completed. Some define as sensor platform any platform capable of embedded (non-desktop) computing – as exemplified by platforms such as Raspberry Pi. The view taken in this chapter is that extensive processing capabilities cannot be expected of sensor network nodes, especially if powered via energy harvesting. Additionally, to fulfil the objective of high degree of integration and reduced space requirements, devices based on System-on-Chip (SoC) philosophy are to be preferred, quite literally reducing the design to a single IC, with the exception maybe of some, specialized, power management circuitry. Even under these constraints, there is a wide range of offerings, consisting of single chip integrated microcontroller and RF transceiver. As an example, Texas Instruments CC2530 is a combination of an 8051-based microcontroller and a IEEE 802.15.4, Zigbee RF subsystem. Another example, is the Nordic Semiconductors nRF52 which is a combination of an ARM Cortex-M4F and a Bluetooth Low Energy RF subsystem. CC2530 is more energy demanding when it comes to transceiver operation (at an advertised 24/29 mA for RX/TX) compared to nRF52 (5.4/5.3 mA for RX/TX) although that comes with higher RF output power for CC2530 compared to the nRF52. The issue of RF power has to be seen under the typical use case in indoor spaces. Both example platforms could be used to communicate over limited distances, i.e., in the range of 10 m. As such, in the remaining we will use CC2530 as the “worst case” design assumption, with the understanding that nRF52 can provide better results under the same, limited range, use case.

### ***3.2 Designing and Testing a TE Harvesting Node***

The TE harvester used in our studies is a TEC1-12703 Peltier module. The surface area of the module is 16 m<sup>2</sup>. The TEC1-12703 is often described as a thermoelectric cooler. Thermoelectric coolers are built on the same operating principles as thermoelectric harvesters. They use the same or similar materials and can be used both ways (as a harvester or a cooler). The main difference between the two is their optimal operating temperature. Modules that are made for harvesting tend to operate better at temperature ranges between 50 and 200 °C. The efficiency depends on their thermal and electrical conductivity which changes based on the ambient temperature. The reason that the harvesting modules are optimized for higher temperatures, is that they are typically used to harvest waste heat from machines operating at high temperatures. Another advantage of the selected module is its cost (around US\$4), compared to specialized TE harvesting modules that are normally orders of magnitude more expensive.

Because the TEC1-12703 module is designated for cooling, its datasheet is not helpful for characterizing its harvesting potential. To this end, we carried out a characterization of a typical off-the-shelf TEC1-12703 unit to determine the relation between temperature difference,  $\Delta T$ , across its two surfaces, and energy harvested

by placing the modules between a “hot” and “cold” plate to allow us control of the  $\Delta T$  up to 49 °C. The interface between the module and the hot and cold surfaces was thermally conductive by means of thermal paste with negligible thermal resistance. Within the area of interface to the module, thermistors (US Sensor USP12397), were embedded to record the exact temperature. Our findings indicate that the power output,  $W$  (in  $mW$ ), of the particular harvester relates to  $\Delta T$  (in Celsius) as  $W = 0.0096 \Delta T^2 + 0.2292 \Delta T$  (least squares fit with  $R^2 = 0.9949$  and an error of approximately 0.11  $mW$ ). All power measurements were carried with a 120 Ohm resistive load. The power output model is sufficient for determining the output current/power of a harvester but is insufficient for calculating the actual charge of an energy storage device, i.e., how fast could a (super) capacitor actually be charged. The leakage that a charging module can exhibit, and the general efficiency of said charging circuitry needs to be accounted for. To this end, we connected the harvester to a BQ25504 Battery management module from Texas Instruments responsible to charge a 1 Farad super-capacitor. After running a series of experiments we determined that the energy harvested this way was better captured by  $W = (2.57 \Delta T^2 + 5.88 \Delta T) \cdot 10^{-3}$  (also in  $mW$ ).

For the sensor platform we use NanoZ-CC2530 devices which employ the Texas Instruments CC2530 microcontroller and integrated Zigbee transceiver. A node, employing energy harvesting, communicates with another node acting as data sink, using the z-stack tool [16]. We carried out energy-exhaustion tests to determine for how long (how many bytes) of *payload* can be transmitted for the amount of energy accumulated in a 1 Farad capacitor (rated at 5 V) charged up to 3.6 V. The packets transmitted followed the standard Zigbee data frame structure. We first conducted experiments using the TEC1-12703 harvester connected to a Texas Instruments BQ25504 Evaluation Board to regulate the voltage and to determine the ability of the harvester to charge the capacitor. After the success of the first step, and in the interest of accelerating the experiments, we used a standard power supply (GPC-3030) to charge the capacitor to the same 3.6 V. The capacitor was then connected to the NanoZ module and used to send data until exhaustion. From the measurements we concluded that it was possible to transmit an average of 4103.7 bytes of payload per Joule using messages of maximum payload size (90 bytes per packet). The specific sensor modules are not energy efficient, as we measured that they consume 0.49  $mW$  in their sleep state when the processor is set to the sleep mode PM2 (only low-frequency oscillator operating).

We note that the NanoZ-2530 devices are characterized by energy losses higher than those of the CC2530 unit alone. This is a natural consequence of the losses incurred by the additional components (voltage regulators, pull resistors, LEDs, etc.) that endow the given board. Rather than attempt to re-design the platform, we retain it in its “off-the-shelf” form, noting again that it presents a “worst case” design. The inclusion of a TE harvester *not* intended for harvesting applications enhanced our confidence that the results shown here are rather conservative estimates of the capabilities of a carefully designed and optimized system.



### 3.3 Performance Results

Depending on the deployment of a sensor, i.e., whether embedded on a North-facing or South-facing external wall, or on the ground or top floor of a building, the harvested energy can be drastically different. There is also impact from the resident activity, and the setpoints of the apartment/house thermostats. The particular metric of interest is the specific heat flow (measured in Watts per  $m^2$ ) across an exterior wall unit. It is out of this heat flow that TE harvesting reclaims a fraction as electricity. No single built environment can be considered representative of all possibilities. Therefore, rather than provide a single standard of TE harvesting performance we will consider an example that demonstrates the inherent variance and seasonality of TE harvesting potential. Namely, we rely on data collected over a period of a year, from 11 different apartments within a single apartment complex in Fort McMurray, Alberta, Canada. The data collected is comprehensive, including such aspects as water flow and temperature for the water used by radiators for heating, water flow and temperature for residential water, CO<sub>2</sub> concentration, etc. For the purposes of this study we consider only the heat flow through exterior (outside-facing) walls, the indoor air temperature, and the outdoor air temperature. First, we extract a 1 year period (8th of September of 2012 to the 8th of September of 2013) which is sufficient for the purposes of capturing seasonal variations. While we have a separate indoor air temperature for each apartment, there is only a single outdoor air temperature, as acquired by the Building Automation System (BAS). It has to be noted that a single outdoor air temperature is, again, only an approximation of the locally specific outdoor wall temperature of each wall unit, since phenomena like convection can, depending on airspeed, result in different temperatures at different spots and orientations. Existing work [11] demonstrated that the indoor-vs-outdoor temperature difference  $\Delta T_{air}$  is a good proxy for the actual heat flow  $q_{hf}$  taking place via the exterior walls. Our interest in using  $\Delta T_{air}$  instead of  $q_{hf}$  is motivated also by the relatively expensive heat flow sensors, compared to the low-cost, and ubiquitous temperature sensors.

Figure 1 is a typical example of the annual variability of the heat flow, calculated as daily average, of two different apartments. Clearly, the seasonal trends are

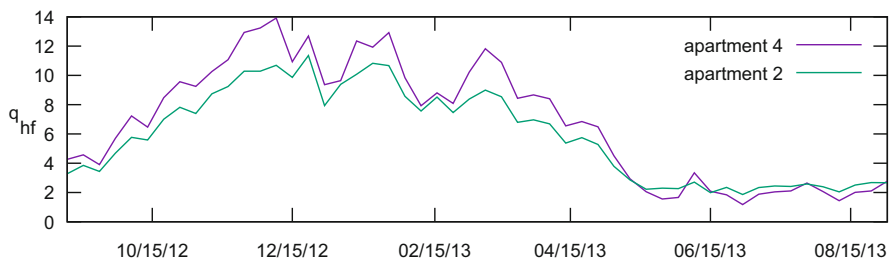
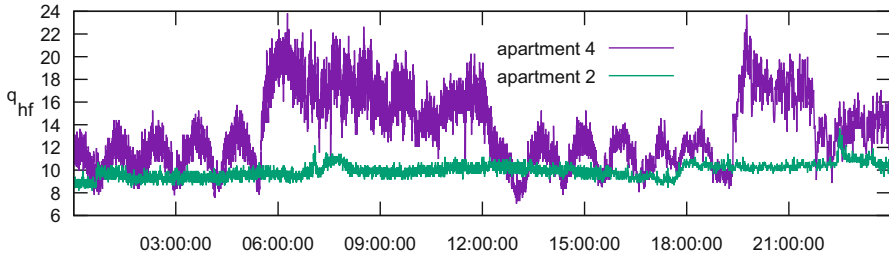


Fig. 1 Exterior wall average daily heat flow,  $q_{hf}$  (in  $W/m^2$ ), of two apartments



**Fig. 2** Exterior wall intra-day heat flow of the same apartments as in Fig. 1

evident, suggesting that when a node harvesting the heat loss through a wall receives plentiful energy from harvesting, it is very likely that neighboring nodes in the same building are in a similar position. These trends however are not devoid of outliers. Without precisely knowing the inhabitants behavior, one can only conjecture on several reasons for outliers to appear (the resident could have been using an electrical radiator close to the location of the heat flow sensor, or touching the wall at the sensor location, or had the windows open, etc.). To illustrate the differences that might show up, consider Fig. 2 depicting the heat flow during a specific day (in mid-March of 2013) during which one of the two presented apartments had almost constant heat flow (possibly the apartment was vacant that day) while the other had highly variable heat flow (the oscillations are probably due to the heating cycling around the thermostat setpoint, while higher setpoints and possible resident activity are evident from approximately 6am to 12pm and from approximately 8pm to 10pm).

Further analysis, reported in [11], of the standard deviation of the daily heat flow (based on hourly measurements) can indeed be very low, regardless of apartment, suggesting that there always exist periods that the apartments are probably vacant and the only dynamics of the heat flow are due to the setpoint of the thermostat in that apartment vs. the external temperature. We also found that the daily average of volume of payload data (i.e., net of protocol encapsulation overheads) that can be transmitted using our platform are in excess of 0.5 Mbytes per device. Clearly, the amount of data is suitable for structural integrity monitoring, and, in general, for the sampling of any slowly changing, or highly compressible, physical phenomena. Nevertheless, whether this volume of data transmission is adequate hinges on the assumption that the receiver is continuously powered and in the proximity of the transmitting sensor. We remark that built environments give the opportunity to power “sink” nodes to perform data collection, plugged into electrical outlets for their power needs. Yet, in the interest of supporting the autonomous operations of the sensors, it is imperative that we also consider support for networking them in a multi-hop network, delivering the data to sink nodes that are several hops away from where the data is collected, i.e., multi-hop environments operating (with the exception of possibly one node) using harvested energy.

In summary, the average harvested power across all apartments over a year, using the  $\Delta T_{air}$  and the characteristics of the TE harvester we described earlier, is 2.33 mW with a standard deviation of 0.21 mW. For comparison purposes, the potential for PV output from a cell of the same surface area (16 cm<sup>2</sup>), at the same geographical location, using off-the-shelf solar cells with efficiency 17% is an average daily power of 88.739 mW. While the difference is significant, this PV output is possible under specific conditions, such as, (a), a South facing wall – compared to other directions, and (b) occlusion-free placement of the PV element from the light source (sun). The TE harvesting option is not without its shortcomings as well. For example, we noticed that during the summer, the average TE power harvested could be as low as 0.113 mW (daily average). Clearly, an (ultra)capacitor is necessary to retain a sufficient charge to provide continuous operation during those periods.

## 4 Multi-hop Environments

In a multi-hop environment, each sensor collects data independently of the rest and it is tasked with using routing via its neighboring nodes to deliver the data to a pre-designated sink node. The action of receiving and forwarding data on behalf of other sensors results in an intrinsic energy cost for supporting routing. Short of using aggregation techniques to condense/compress the forwarded data, it is immediate that the energy cost paid by a node is proportional to the volume of data it is required to forward. Even in the absence of energy harvesting, the energy cost of participating in multi-hop forwarding can quickly deplete the energy of certain nodes, thus eliminating them subsequently from being forwarders of traffic and, in the worst case, resulting in a partitioned network. Multi-hop routing can result in an uneven energy burden across nodes, which is not necessarily in proportion with the harvested energy available to the nodes, hence the need for routing mechanisms to account for the varying energy availability and to adjust the routing decisions accordingly.

### 4.1 Routing Problem Definition

A complication of producing an acceptable solution to multi-hop routing is the uncertainty about the future energy harvesting. We can employ one of two straight-forward strategies: (a) a strategy whereby we have no information about the future energy harvesting outcomes and produce routing decisions based on the current energy reserves, or, (b) a strategy exploiting the likely repetition, at a particular time scale (usually a calendar year) of the possible energy harvesting. The latter relies on the assumption that entire seasons will repeat, at least in a proportional sense, to influence the routing choices in the same way each year. Thus, a year's data can be used as the template of future years. While option (a) appears reasonable

as well, it implies the frequent exchange of control message between nodes (at a time scale to be decided) such that the recalculation of the routing decisions can take place. However, such control message transmissions incur an energy cost. In the simplest centralized form, option (a) calls for forwarding to the sink the current state of charge of the nodes. The sink node solves off-line the routing problem and informs the nodes about the new routing decisions. This strategy is meaningful for networks whose topology is infrequently changing, e.g., for sensors embedded in buildings.

A third approach we will briefly comment on is to produce estimates for a limited time horizon based, e.g., on an auto-regressive model. An accurate prediction model would, at least technically, allow one to produce solutions for future routing decisions using schemes such as the time-varying splittable flow model presented in [17]. If indeed those predictions are accurate, one could then attempt to develop more efficient computational techniques to those outlined in [17]. While the prediction also requires some control messages to be exchanged, the resulting predictions can span a longer time frame, and therefore the control message exchanges can be infrequent.

## 4.2 Optimization Formulations for Multi-hop Routing with Energy Harvesting

To model the isolation of each sensor's generated data, from those of any other sensor's, we will treat them as separate commodities. The goal of every node is to forward as much data as possible with its current energy to the sink using, possibly, multiple paths. Instead of stipulating which paths are to be used and which ones are not, we pose the question as determining what fraction of traffic for each commodity should flow across each link with the purpose of either maximizing the total, or, the total concurrent volume of data delivered to the sink. By adopting a multi-commodity model, we do not force a particular routing, but rather we anticipate to observe that in optimal routing, the flows to be split to traverse across multiple paths, and we anticipate that such splitting will exhibit seasonal characteristics, i.e., a particular link will be used certain times of the year and not at others.

We consider the following multi-commodity maximization formulation of the routing problem on the  $n$  sensor nodes (with  $t$  denoting the sink):

$$\max \sum_{i=1}^n (f_i(s_i, t)) \quad (1)$$

s.t.

$$f_i(s_i, t) = \sum_{w \in \mathcal{N}_{s_i}} f_i(s_i, w) = \sum_{w \in \mathcal{N}_t} f_i(w, t) \quad (2)$$

$$\sum_{w \in \mathcal{N}_u} f_i(u, w) = \sum_{v \in \mathcal{N}_u} f_i(v, u) \tag{3}$$

$$q \left( \sum_{i=1}^n \sum_{v \in \mathcal{N}_u} f_i(u, v) \right) + p \left( \sum_{i=1}^n \sum_{w \in \mathcal{N}_u} f_i(w, u) \right) \leq c(u) \tag{4}$$

Where  $f_i(v, w)$  is the flow of commodity  $i$  from node  $v$  to node  $w$ . Exceptionally, the auxiliary notation  $f_i(s_i, t)$  indicates the total flow from the origin of commodity  $i$  (node  $s_i$ ) towards the sink  $t$  over possibly multiple hops and paths.  $\mathcal{N}_u$  indicates the neighboring (adjacent) nodes to node  $u$ . We assume the number of nodes, minus the sink, is  $n$ . Equation (2) applies to all commodities  $i$  and indicates that the total flow out of the source and into the sink must be the same and is equal to  $f_i(s_i, t)$ . Equation (3) holds for each node  $u$  (other than the sink) and commodity  $i$  and represents the flow balance equation into and out of node  $u$ . Finally, Eq. (4) represents the constraint that the energy expended at node  $u$  cannot be more than  $c(u)$  (the available energy). Here,  $q$  and  $p$  represent the ratios of energy spent per unit of flow for transmitting and receiving respectively.

By solving the above problem to determine  $f_i(v, w)$  in each cycle, using an LP solver, we derive the maximum amount of data that can be sent with the current energy levels in the network. Using the computed solution, we can determine how much energy each sensor has to spend. We subtract that from the current energy of the nodes and then we move to the next cycle, where the amount of energy harvested is added to each node, and the multi-commodity flow is solved again. We note that, as pointed out previously, the LP can be centrally solved at the sink and the results communicated to the nodes with small message overhead.

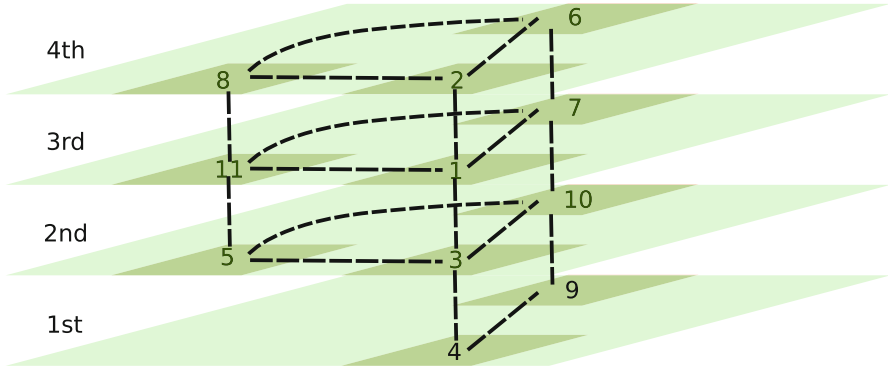
The above formulation makes no attempt to equalize the opportunities of all sensors to deliver data to the sink. A “fair” version of the routing problem can be formulated as a *concurrent multi-commodity flow problem* by changing the objective to:

$$\max \quad f_1(s_1, t) \tag{5}$$

and adding the constraint:

$$f_1(s_1, t) = f_2(s_2, t) = \dots = f_n(s_n, t) \tag{6}$$

The intuition behind it is that we maximize the total flow to the sink as long as all the commodities can deliver the same amount of data. Thus, the additional constraint ensures that no commodity is going to receive any worse service than any of the rest. Nevertheless, we expect this to happen at the detriment of the total flow. Furthermore, it should be clear that solving the concurrent flow problem does not necessarily result in the optimal use of the energy of the nodes. In the concurrent version there will always be sensors that have an excess of energy, which can lead to many different maximum solutions, some of them having wasteful energy



**Fig. 3** Network topology across four building floors – dashed lines represent communication links

expenditure, i.e., leaving drastically different (and possibly low) residual energy at the nodes. Consider the following example, taken from our sample network in Fig. 3, that illustrates the problem: consider the sink node at the 4th floor, able to receive the transmissions of any node at that floor (nodes 2, 6, and 8). Node 3 routes flow 3 to node 5. Node 5 proceeds to route this flow to node 11 which then routes it along the path to the sink (via node 8). Node 5 also routes flow 5 to node 3, which in turn routes it to node 1 to be routed along the path to the sink (via node 2). If instead of this, node 3 directly routed flow 3 over the path involving nodes 1 and 2, and node 5 routed flow 5 over the path involving nodes 11 and 8, there would be less energy spent. Nodes 11 and 1, which are closer to the sink, would still need to route one complete commodity flow each, which for them would cost the same, but node 5 and 3, instead of receiving one flow and sending two out, will now just send one flow each. The reason that this is allowed is that since the nodes 5 and 3 are closer to the edges of the network, they have a lot more residual energy, which allows them some flexibility on how to spend their energy. The problem is that unnecessary usage of energy like this, can lead to depletion of the energy of those nodes that could have been useful in future cycles.

To address this shortcoming, we optimize with respect to a secondary objective whose purpose is to maximize the residual energy of nodes in anticipation that it could be used in subsequent cycles. We remove wasteful solutions produced by the maximum concurrent formulation, by creating a second LP problem which, using the solution to the concurrent version (let's denote it by  $f^*$ ) explicitly minimizes the sum across all nodes of the consumed energy, as captured by Eq. (4). That is,

$$\min \sum_{u=1}^n \left( q \left( \sum_{i=1}^n \sum_{v \in \mathcal{N}_i} f_i(u, v) \right) + p \left( \sum_{i=1}^n \sum_{w \in \mathcal{N}_i} f_i(w, u) \right) \right) \quad (7)$$

s. t.

$$f^* = f_1(s_1, t) = f_2(s_2, t) = \dots = f_n(s_n, t) \quad (8)$$

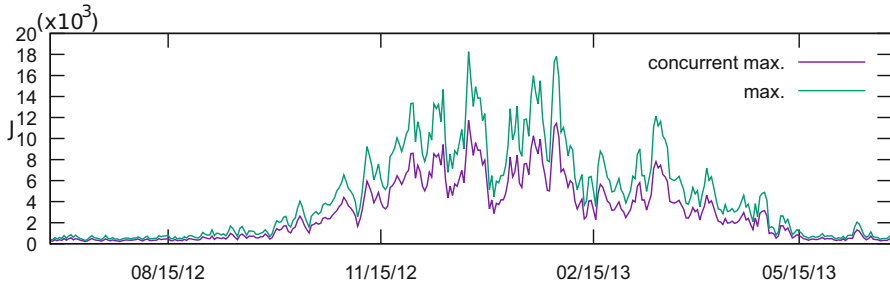
plus the additional constraints for flow conservation and source/sink flow summation we already presented. The reader should note however that this minimization takes place over the sum of energy expended across all nodes, with no specific attention to any single node.

A few technical remarks are in order: (1) we take the approach that solving off-line the optimization problem(s) and informing the nodes of the way to route data is acceptable because the topology is static and the information about the energy levels is relatively short and could be communicated to the sink (and from there to any optimization solution facility) at the beginning of the duty cycle and the nodes can be informed about the solution (delay is not a concern as the operation of the network is duty cycled anyway), and, (2) it is possible to generalize the fairness captured by the concurrent formulation to a weighted fairness by setting  $f_i(s_i, t) = w_i f_1(s_1, t)$  where  $w_i$ 's are fixed weights, in particular when it is known that certain nodes produce a constant factor more data than others by virtue of the sensing they perform.

### 4.3 Simple Routing Strategies

We refer to Fig. 3 as an example topology of the 11 apartments over four floors, and we use the data collected from the actual apartment over the period from 25th of June 2012 to 25th of June 2014. We assume one node per apartment attached to its exterior wall, and each node able to communicate with all the nodes on the same floor, as well as with the nodes at the same location on the floor plan on adjacent floors. We also consider two different sink node placements: one at the fourth floor and one at the second floor. A sink at a specific floor can communicate in one hop with all the sensors at the same floor. The reason for choosing those two floors for the sink placement is to have a location close to one extreme end (4th floor) of the building as well as one closer to the “middle” (2nd floor). Note that the optimal sink placement problem is a separate concern not addressed in this study. Specifically, we consider that the placement of sink node(s) is an input to our routing problem, as it is usually restricted by the need of the sink nodes to have adequate wired network connectivity and continuous power supply, i.e., to be parts of a wired networking infrastructure.

Figure 4 demonstrates the difference in results between the maximum multi-commodity problem and the maximum concurrent flow (for a  $q/p = 1.31$ ) assuming an infinite energy storage capacity. The results are shown in terms of total energy available for transmitting (injecting) data from all nodes in the network – hence decoupling the results from the absolute value of bits transmitted per unit of energy, which is a protocol and physical layer dependent factor. We note that the concurrent flow limits the used energy (and hence the delivered data) but this comes



**Fig. 4** Comparison of maximum vs. concurrent maximum (for  $q/p = 1.31$ ) expressed in Joules

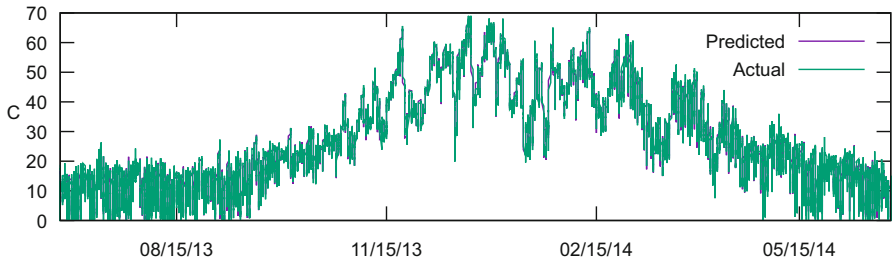
at the benefit of an equal volume of data delivered to the sink from each sensor. Regardless of which scheme is used, the sensors next to the sink (there are three of them in the example topology) are the bottlenecks to routing and ultimately it is their energy reserves that dictate the overall data transfer volume. As we have identified in [12], introducing a finite energy storage per node (e.g., a 5 Farads at 5 V capacitor) is not necessarily going to severely limit the performance of the network in terms of data delivered. This is because, depending on duty cycling (to accumulate energy before transmitting), whether a finite energy storage limits performance depends on being able to exploit the residual energy at nodes for the benefit of routing. Nodes that are not adjacent (or close) to the sink are incapable of contributing to increasing the delivered traffic, and no amount of energy reserves improves this situation. To the extent that a solution can be engineered specific to a network deployment, larger energy storage capacities should be assigned to nodes closer to the sink, assuming the location of the sink is known in advance.

Additionally, to advance the possibility of using a form of prediction of the nodes energy reserves (and limit the extent to which control information needs to be exchanged between nodes) we considered an Autoregressive Integrated Moving Average (ARIMA) model to predict upcoming energy harvesting output. The prediction is expressed as a prediction for the future  $\Delta T$  temperature difference, because  $\Delta T$  is a proxy of the TE energy output. For comparison purposes we also considered a trivial prediction method that predicts  $\Delta T$  of the next time interval to be the same as the measured one of the previous interval. The ARIMA model is an attempt to capture a more refined behavior of the physical process underlying the evolution of TE harvesting. Yet, it is also intuitively compelling that the trivial approach could be able to produce good results often enough. Both methods were chosen for their simplicity and ease of implementation and lack of complexity (on the prediction part) that could be executed at each computationally-limited node. Specifically, the ARIMA model is of the following form  $(1 - \sum_{i=1}^p \phi_i L^i)(1 - L)^d X_t = c + (1 + \sum_{i=1}^q \theta_i L^i) \epsilon_t$  where  $L$  is the lag operator and  $\phi_i$  are the parameters of the autoregressive part and  $\theta_i$  of the moving average part.  $c$  is a constant and  $\epsilon_t$  is the error. A differencing order of  $d = 1$  was found to be adequate for our time series data. The lag operator  $L$  is defined by  $X_t = LX_{t+1}$ . Table 1 summarizes the



**Table 1** ARIMA model MSE for each apartment with hourly cycle (rows 1 and 2) or twice-per-day cycle (row 3)

Apartment	1	2	3	4	5	6	7	8	9	10	11
Fit error (for hourly)	2.85	2.69	3.11	2.77	3.78	2.79	4.42	2.60	2.86	2.87	3.70
Hourly	3.64	3.13	3.09	3.37	4.43	3.24	5.45	2.75	3.03	3.01	3.79
Twice-per-day	14.81	12.00	13.02	13.74	12.45	13.22	14.49	13.64	11.72	10.14	20.83



**Fig. 5** Hourly cycle predicted  $\Delta T$  vs. actual  $\Delta T$  in Celsius for the second year for apartment 3

fitting MSE for each apartment, using a model with autoregressive coefficients at 1, 2, 4, 12, 36, 48, 60 and moving average coefficients at 1 through five, 12, 24, 48, 72. The choice for the particular model was based on the observations of the strong autocorrelation at small lags (i.e., within a few hours), the distinct antithesis between night/day (lag of 12 h) and the multi-day near-term similarities (over 24, 48, and 72 h). The coefficients were determined separately for each apartment using a year’s collected data.

When the cycle is hourly or twice per day (day/night), we calculate the next value using hourly values – in the case of hourly by predicting the next hour, and in the case of twice per day by predicting the next 12 hourly values and averaging them. In the case of a daily cycle, averaged daily data are used to make the predictions. Table 1 presents the MSE when using an ARIMA model built based on the first year’s data to predict across the second year. The furthest in the past lags in the hourly model represent 72 h in the past, relative to current time. The MSE for the hourly and the twice–day cycle are also presented for each apartment. Note that the error is larger for longer cycles which can be explained by the fact that the hourly cycle results in predictions for just the next hour, whereas the twice-daily predicts for the next 12 h, using the same data. Figure 5 highlights how close the predictions are to the real data in the case of the hourly cycle in apartment 3 (the corresponding trivial versus ARIMA predictor are shown in Table 2).

Predicting using the ARIMA model turned out to be good, but for certain time scales, e.g., the hourly cycle, less accurate than the trivial prediction. In short, the ARIMA model may be needlessly complicated and possibly overfits to the data when evaluated in terms of its next-hour prediction potential. In the case of daily and the twice–daily cycle the ARIMA was more accurate than the trivial approach. The daily cycle exhibits a substantial error no matter the prediction

**Table 2** Comparison of ARIMA versus trivial predictor MSE for apartment 3

Cycle	ARIMA	Trivial
Hourly	3.09	2.27
Twice-per-day	13.02	34.09
Once-per-day	20.88	21.69

approach used. This can be understood based on the extrinsic influence of the weather phenomena that can vary widely over such (daily) time scale. In this respect we remark that the particular location in North Alberta is not in a temperate climate and “unpredictability” of the weather is a rather common phenomenon. Moreover, the daily scale includes, 2 days in each week are the weekend days that usually underpin a different resident behavior compared to the average daily behavior.

Most impressive is the fact that a trivial method can be adequate for the hourly cycle, as temperature differences between successive hours of the day are minor. Note that the training of the model to derive the coefficients took place in a “natural” manner, i.e., using (for the 2 year period) each apartment’s first year data to build the model which was subsequently used in the predictions for the same apartment during the second year. That is, we did not use standard techniques (e.g., holdout and cross-validation schemes) because in essence we only had two sets (one for each year) and partitioning them further would impact the over-arching annual dynamics. We hasten to add that further improvement on the time series prediction are certainly possible. Our intent was to demonstrate that even well-known and well-used techniques, such as ARIMA modeling, are sufficiently powerful to guide decisions related to routing over the studied energy-harvesting WSNs. It is also remarked that we treated each apartment individually and separately because each apartment exhibited highly idiosyncratic patterns of use, depending on its resident(s) behavior – including cases where apartments were vacant for long periods of time. In future work, we will examine fitting a common Hidden Markov Model, across all, heterogeneous, apartment time series. Note however that, due to privacy restrictions, the ground truth of e.g., how many residents were inhabiting the same apartment over any period of time is information unavailable to us and hence we will have to rely on “educated guesses”.

## 5 Conclusion

Wireless networks are known for their flexibility when it comes to setting up systems of communicating peers. They free us to perform node placement guided by the application needs. In this work we have hinted that the application needs for a WSN can call for their placement inside hard-to-reach areas, e.g., wall units, and left to operate essentially “forever”. We can then see the application needs as transformed to determining where to place the nodes where they can both (a) observe the

phenomena we are interesting in sensing and, (b), be close to harvestable energy. The current continuum of options for energy harvesting put the sensors in places where the opportunity to harvest energy is maximized. However, this leaves us with very few options. The purpose of this work explained in this chapter is to significantly broaden the spectrum of placement options for TE harvesting nodes. Essentially, any exterior-facing wall is a good location. In this sense, we augment previous works on TE harvesting to a much wider range of surfaces and locations in today's built environments.

Nevertheless, the nodes need to dynamically manage their power, due to the variable availability of energy. Routing and sensing of the data should adapt to the current energy constraints. In periods of time with superfluous energy the nodes are not required to conserve anything, and they can work with no constraints. On periods with really constrained energy, the nodes need to optimize their energy usage, by sensing according to their energy reserves, routing their data through possibly different paths, according to the reserves of the other nodes of the network, and generally manage their operation according to the current energy situation of the network.

We have demonstrated that a TE harvesting-powered WSN node is a feasible design with current off-the-shelf components and able to sustain a considerable volume of data transfers. With respect to the multi-hop designs, a self-powered energy harvesting node for building applications can benefit from the relatively static network topology while being exposed to the uncertainties of the energy harvesting energy source. In essence, the work outlined in this chapter illustrates some ways to exploit the knowledge of the topology to, e.g., pro-actively calculate the splitting and routing of data flows, by having to predict the next-step energy availability. A more challenging version of the problem would be the addition of link uncertainties in the topology. Such link uncertainty can be introduced by the behavior of the residents, e.g., if they use networking equipment that interferes with the protocols and frequencies used by the sensor nodes.

A welcome observation is that simple prediction schemes can be utilized to pre-plan the routing in future steps. We also expect a worsening of the prediction, the further in the future we predict for, regardless of the prediction technique. Such long-term predictions can seriously jeopardize the application of schemes like the water-filling method in [17], even if its computational complexity could have been somehow reduced. A possible alternative is to consider cycles of length longer than a day, in the hope that the averaging of the energy harvested will be smoothed out and become more predictable. This however introduces two technical issues: first that the energy storage supported by the nodes might need to be significant (to carry over the operation during "tough" times) thus requiring appropriately sized super capacitors possibly in the order of tens of Farads, and secondly the data to be forwarded may be delayed by a significant amount of time before it can be delivered to the sink. It is debatable whether fairness across time is an absolute requirement when it comes at the cost of more expensive devices and delayed traffic collection. To this end, our study has addressed only the fairness across nodes (for the same time cycle).

## References

1. R. J. Vullers, R. Schaijk, H. J. Visser, J. Penders, and C. V. Hoof, "Energy harvesting for autonomous wireless sensor networks," *Solid-State Circuits Magazine, IEEE*, vol. 2, no. 2, pp. 29–38, 2010.
2. N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 13–24.
3. M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, "Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment," in *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*. IEEE Computer Society, 2009, pp. 277–288.
4. S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fennes, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*. IEEE, 2007, pp. 254–263.
5. E. Rodriguez-Diaz, M. Savaghebi, J. C. Vasquez, and J. M. Guerrero, "An overview of low voltage dc distribution systems for residential applications," in *Proceedings of the 2015 IEEE 5th International Conference on Consumer Electronics Berlin (ICCE-Berlin)*. IEEE, 2015, pp. 318–322.
6. Z. Yang, A. Erturk, and J. Zu, "On the efficiency of piezoelectric energy harvesters," *Extreme Mechanics Letters*, vol. 15, no. Supplement C, pp. 26–37, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2352431617300482>
7. S. Kim, R. Vyas, J. Bito, K. Niotaki, A. Collado, A. Georgiadis, and M. M. Tentzeris, "Ambient rf energy-harvesting technologies for self-sustainable standalone wireless sensor platforms," *Proceedings of the IEEE*, vol. 102, no. 11, pp. 1649–1666, Nov 2014.
8. V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith, "Inter-technology backscatter: Towards internet connectivity for implanted devices," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: ACM, 2016, pp. 356–369. [Online]. Available: <https://doi.org/10.1145/2934872.2934894>
9. P. Martin, Z. Charbiwala, and M. Srivastava, "Doubledip: Leveraging thermoelectric harvesting for low power monitoring of sporadic water use," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: ACM, 2012, pp. 225–238. [Online]. Available: <https://doi.org/10.1145/2426656.2426679>
10. B. Campbell and P. Dutta, "An energy-harvesting sensor architecture and toolkit for building monitoring and event detection," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ser. BuildSys '14. New York, NY, USA: ACM, 2014, pp. 100–109. [Online]. Available: <https://doi.org/10.1145/2674061.2674083>
11. A. Kollias and I. Nikolaidis, "In-wall thermoelectric harvesting for wireless sensor networks," in *Proceedings of the 3rd International Conference on Smart Grids and Green IT Systems*, 2014, pp. 213–221. [Online]. Available: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0004864102130221>
12. —, "Seasonally aware routing for thermoelectric energy harvesting wireless sensor networks," in *Smart Cities and Green ICT Systems (SMARTGREENS), 2015 International Conference on*. IEEE, 2015, pp. 1–11.
13. M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1602–1610.
14. R. Rao, S. Vrudhula, and D. N. Rakhmatov, "Battery modeling for energy aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, 2003.

15. F. Simjee and P. H. Chou, "Everlast: long-life, supercapacitor-operated wireless sensor node," in *Low Power Electronics and Design, 2006. ISLPED'06. Proceedings of the 2006 International Symposium on*. IEEE, 2006, pp. 197–202.
16. ti.com, "A fully compliant zigbee 2012 solution: Z-stack," <http://www.ti.com/tool/z-stack>, 2012, [Online; accessed January 2014].
17. J. Marašević, C. Stein, and G. Zussman, "Max-min fair rate allocation and routing in energy harvesting networks: Algorithmic analysis," in *Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2014, pp. 367–376.

# Building a Data Pipeline for the Management and Processing of Urban Data Streams



Elarbi Badidi, Nouf El Neyadi, Meera Al Saeedi, Fatima Al Kaabi, and Muthucumar Maheswaran

**Abstract** Urban data streams (UDS) originate from various sensors and Internet of Things (IoT) devices deployed in smart cities as well as social media sources such as Twitter and Facebook. The large volumes of urban data need to be harnessed to help smart city stakeholders and applications make informed decisions on the fly. Furthermore, effective management and governance of smart city components relies on the ability to integrate and federate their data, process urban data streams locally, and use big data analytics. Data integration and interoperability is a challenging problem that smart cities are facing today. Successful data integration is crucial for improved services and governance. This chapter describes a framework that aims to serve in building a data pipeline for the acquisition and processing of urban data streams, urban data analytics, and creation of value-added services. The framework relies on latest technologies for data processing including IoT, edge computing, data integration techniques, cloud computing, and data analytics. The proposed platform will facilitate real-time event detection, notification of alerts, mining the opinions of citizens regarding the governance of their city, and building monitoring dashboards. A prototype of the platform is being implemented using the Kafka messaging platform.

**Keywords** Smart cities · Internet of Things · Data integration · Data Interoperability · Data streams processing · Messaging Queue

---

E. Badidi (✉) · N. El Neyadi · M. Al Saeedi · F. Al Kaabi  
College of Information Technology, United Arab Emirates University, Al-Ain, United Arab Emirates  
e-mail: [ebadidi@uaeu.ac.ae](mailto:ebadidi@uaeu.ac.ae)

M. Maheswaran  
School of Computer Science, McGill University, Montreal, QC, Canada  
e-mail: [maheswar@cs.mcgill.ca](mailto:maheswar@cs.mcgill.ca)

## 1 Introduction

In modern cities, massive amounts of data are continuously collected originating from a variety of sensors and IoT devices, which monitor in real-time the operations of various city systems as diverse as water, energy, transportation, and environment. Furthermore, social media networks such as Twitter, Facebook, and Google+ are becoming a new source of real-time information regarding the activities and concerns of the citizens. Users of these social media networks are regarded as social sensors as they provide valuable information regarding the events happening in their cities [1, 9]. The data streams, in structured and unstructured formats, from these various sources, are so broad and complex to handle with traditional or conventional data management tools and methods [5]. They are currently insufficiently leveraged by planning authorities [2, 11].

By processing urban data streams, it will be possible to identify the most significant events and patterns and make appropriate decisions on them in near real-time. It will allow city stakeholders to respond to urgent situations with both speed and precision. The data processing chain may involve operations like filtering, aggregating and ultimately storing resulting data for further processing by data analytics applications. Moreover, the ability to federate and process urban data streams will permit to harness the governance of smart cities.

To address the above concerns, we propose a conceptual platform for the management and processing of urban data streams. Pre-processing data streams at the edge before sending data to the fog/cloud will reduce the usage of network bandwidth and make further processing in the fog/cloud more efficient. The platform aims at providing support for federating and processing data streams by using emerging open source tools such as Apache Storm, Kafka, Yarn, Apache Samza, WSO2 CEP, and Cassandra. It will help to find valuable insights in an overwhelming amount of urban data streams and online conversations.

The rest of this chapter is organized as follows. Section 2 provides background information on urban data streams and highlights the challenges of urban data processing. Section 3 briefly surveys the data integration techniques. Section 4 describes the two modes of data processing: batch processing and real-time processing. Section 5 describes the process of building an urban data streams pipeline. Section 6 describes our proposed framework for urban data management and processing, and a use case scenario. Section 7 discusses challenges and concerns of urban data management and processing. Finally, Section 8 concludes the chapter.

## 2 Urban Data Streams

In recent years, as a consequence of the significant decrease in the cost of sensors and the continued miniaturization of electronic devices, various kinds of sensors are proliferating in cities and businesses. It is becoming possible to deploy thousands

and even millions of sensors to sense various parameters such as humidity, temperature, sunlight and air pressure. These sensors generate huge volumes of data. They are usually capable of continuous reporting, which leads to the challenges often linked with “big data.”

In a smart city, several systems are designed to work with real-time data from sensors, meters, and many other devices used to assure the operations of the city. Sensors include not only physical hardware sensors but also software sensors and people. Software sensors can, for example, report the presence of the user detected by mouse clicks and movements. People as sensors implies the idea of users delivering direct input through social networks or dedicated interfaces.

Urban data streams originate from IoT devices and sensors, deployed in the city to monitor and report:

- Weather conditions, so that alerts and warning systems are activated to avoid and reduce traffic jams and accidents.
- Parking space availability, so that drivers avoid the lengthy searches for unoccupied spaces.
- The structural integrity of bridges, buildings, and historical monuments.
- Trash levels in waste containers, so that trash collection trucks optimize their routes.
- Night-time activity and traffic, so that adaptive smart lighting lights streets, sidewalks, and roads in an energy-efficient way.

Many cities are using advanced IoT solutions to implement smart adaptive street lighting systems. These systems aim to create safer urban environments while saving energy and protecting the environment. They light up when human activity is detected and darken when the streets are empty to reduce costs. For example, the city of San Diego recently launched a \$ 30 million Smart City IoT platform project that represents the massive deployment of the Smart City IoT platform around the world. The platform will add nearly 3200 IoT smart nodes to the current street lighting infrastructure to collect real-time sensor data across the city [15]. The collected data can be used to guide firefighters and police to accident or emergency scenes, increase security, optimize municipal systems, and develop intelligent applications that can guide drivers, for example, to available parking lots [14]. Furthermore, in recent years, the European Union has encouraged its Member States to develop smart cities and has allocated € 365 million for this initiative. Barcelona, Amsterdam, London, and Copenhagen and many other cities are leading the smart city development effort [14, 22].

A typical IoT solution is characterized by numerous IoT devices and sensors that typically use gateways to communicate over a network with a backend server that runs an IoT platform, thereby integrating the data generated in the network. One of the major challenges of current urban deployments is the non-interoperability of heterogeneous devices and technologies used in the city [17, 21]. These devices generate different types of data transmitted to a control center for storage and processing. Several works investigated the IoT interoperability issue. Aloï et al. [23, 24] proposed a smartphone-centric gateway solution to provide support for



IoT interoperability. Their proposed architecture permits to collect and forward data originating from sensors and wireless IoT devices that are using various communication interfaces and standards. Blackstock et al. [25, 27] used a hub-based approach for IoT interoperability, where hubs can aggregate things using web protocols. They proposed a staged base method to interoperability. Asensio et al. [26] proposed an IoT gateway to provide seamless connectivity and interoperability among devices. Perera et al. [28] proposed a plugin based IoT middleware on mobile devices that aims to allow collecting and processing sensor data before its transmission to the cloud for storage or further processing.

Another concern is the aggregation of the data generated by the distributed data nodes. Data aggregation deals with large volumes of data using time series analyses and data compression methods to reduce the size of raw sensory measurements [6]. It reduces communications overhead and allows for more advanced tasks in large-scale systems such as clustering or event detection. To efficiently access and use sensory data, the semantic representation of aggregations and abstractions is crucial to provide machine-readable observations for higher-level interpretations of the real context [8]. Data aggregation is common in many applications. For example, in the health sector, to analyze a patient's situation thoroughly, it is necessary to aggregate data from various IoT-based health service providers that collect data from that patient using different sensors.

Sensors are typically embedded in city equipment and physical devices to control the devices locally. They can be organized into a network to report data consistently. Wireless technology allows data to be received from individual sensors and the collected data is made available to the stakeholders and the outside world over TCP/IP and Web-based reporting. Traditional applications can store the data collected into a database to be queried later. With the power of emerging technologies for processing data streams, such as Apache Storm and Kafka, applications can perform analytical analyses on sensor data in real time and thus enable informed decision-making.

Also, the use of social media analytics in emergency interventions is particularly interesting. Social media provide a constant flow of information that can be used as an inexpensive sensing network to collect information in near real-time in an emergency. Although people post many unrelated social networking messages, any information about the emergency can be valuable to emergency response teams. Accurate data can help to obtain a correct picture of the emergency, allowing for a more efficient and timely response that can reduce overall losses and damage [10].

### 3 Urban Data Integration

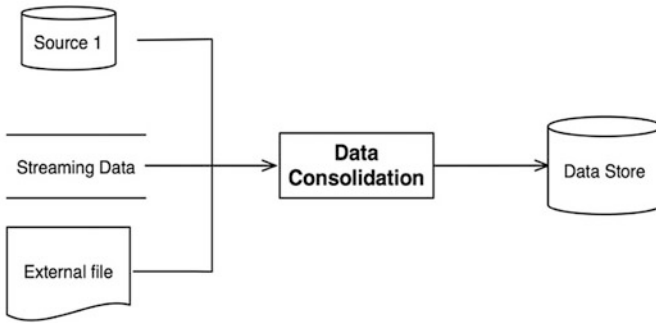
Urban data generated in different parts of a smart city requires the use of appropriate integration techniques that will provide city stakeholders with a complete picture of what is happening in the city to make informed decisions. The data integration literature identifies the following main methods that can be used to integrate urban data: data propagation, data consolidation, data federation, use of the Extensible

Markup Language (XML) and JavaScript Object Notation (JSON) as standard formats for data exchange and storage, development of controlled vocabularies and mashups.

*Data propagation* refers to the movement of data from one or more data sources to target locations. Data propagation systems typically transmit data to target locations. Most often, they are driven by events, and the propagation of data is done according to some rules (see Fig. 1c). Data updates in a source system can be propagated synchronously or asynchronously to the target system [20]. The propagation guarantees the transmission of data to the target system, regardless of the type of synchronization used. This data delivery guarantee is a key distinguishing feature of data propagation. For example, in data warehouses and operational data stores, updates involve moving large volumes of data from one system to another. The data movement is done in batches to avoid impacting the performance of data warehouse operations.

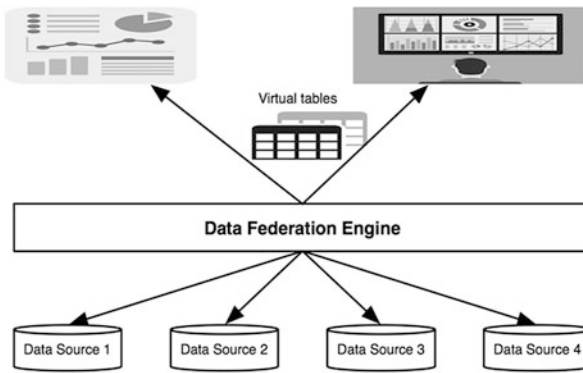
*Data consolidation* refers to collecting data from multiple sources and integrating them into a single persistent data store (see Fig. 1a). It helps to cope with data duplication and reduce the costs associated with dependency on multiple data management points and databases. It will enable organizations to report and analyze data efficiently, such as in data warehousing. The data store can serve as a data source for downstream applications, such as in an operational database system. Because data comes from multiple data sources, there is always a delay between when data is generated or updated in a data source, and when these changes appear in the data store. Depending on the underlying communication infrastructure and the nature and size of the updated data, this delay can range from a few seconds to several days.

*Data federation* refers to software resources that provide users with a single logical view for presenting and accessing data stored in one or more data sources. It represents an alternative model for the storage and use of data by organizations. This technique is also called data virtualization technology. When data sources are traditional databases, data federation leverages the native capabilities of managing and retrieving data from individual databases and creates a single, logical, and unified view of federated databases [18]. Business applications are presented with a combined data schema even though the source database schemas are distributed in many federated databases (see Fig. 1b). When a business application issues a query on this logical view, the data federation engine retrieves the data from the appropriate data source, adapts it to the virtual view, and sends the results to the requesting business application [13, 18, 20]. Federating data streams to provide a unified view is a challenging problem due to the dynamicity and heterogeneity of data streams. Because of these two factors, federation solutions developed for traditional databases are not applicable to data streams. Feng et al. [29] described the platform ACEIS, which aims to allow to discover and integrate heterogeneous and dynamic sensor data streams. Upon receiving a user request (event request), the platform discovers and composes relevant data streams to address both functional and non-functional requirements of the event request. The platform uses an ontology



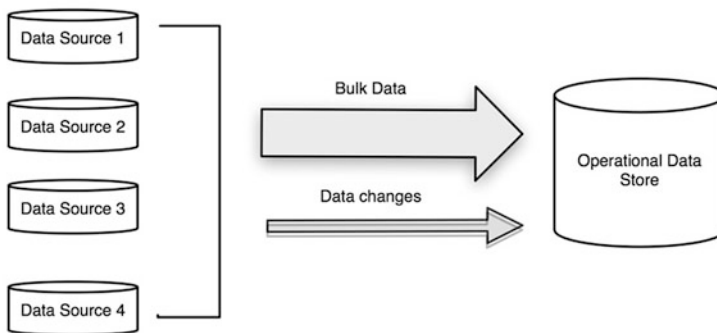
Multiple Data sources

(a)



Multiple Data sources

(b)



Multiple Data sources

(c)

**Fig. 1** Data integration techniques. (a) data consolidation. (b) data federation. (c) data propagation

for the discovery and the composition processes, which result in the generation of queries that operate on semantic data streams.

The utilization of standard formats for data exchange is essential in the data integration endeavor. XML is a markup language that facilitates the sharing of data between heterogeneous computer systems. Many databases and software applications are XML compliant. By adopting standards to represent certain types of data, XML facilitates data integration and application interoperability. JSON is an open-standard file format, which uses text to convey data objects composed of pairs of attribute values and array data types. It is a lightweight and language-independent data exchange format that is easy to read and write for humans and easy to generate and analyze for machines. JSON is becoming more and more the preferred format for exchanging and integrating data using RESTful web services.

*Controlled vocabularies* represent a form of data integration by imposing standardized terminology for data elements that appear in databases. Example of controlled vocabularies are the various ontologies developed in the context of smart cities. An ontology typically acts as a mediator for the separate schemas of different data sources and as a reference schema for federated data queries. The European project READY4SmartCities (<http://www.ready4smartcities.eu/>), which aims to increase awareness for the adoption of ICT and semantic technologies in the energy system to reduce energy consumption and CO<sub>2</sub> emissions in smart cities, maintains a catalog of many ontologies created in different smart city projects (<http://smartcity.linkeddata.es>). These ontologies concern various elements such as regulatory elements, sensors, places, people, smart buildings, etc. Kettouch et al. [30] proposed a conceptual framework to integrate data streams in smart cities on the fly using Linked Data. The goal of Linked Data is to permit the definition of links between data sources and, thus, create a single global data space, referred in the literature as the “web of data.”

A *Mashup* is a technique for creating new applications that fuse data from multiple sources to create an integrated experience. Mashup applications are typically built using open APIs, widgets, Web services, and data sources. FixMyCity [16] is an example of a smart city mashup that lets citizens report damages in public spaces to municipal departments.

## 4 Batch and Real-time Processing of Data Streams

Urban data can be processed in two ways: batch processing or stream processing. Batch processing simply means that data is processed periodically and requires separate programs for input, process, and output. In stream processing, data is not collected to reach certain quorum or timeout before processing is triggered. When the data event is received, the program immediately processes it and creates the output. This is known as event processing or real-time processing. It is often characterized by low latency. However, latency is a function of what the process is trying to do [4].

Stream processing systems (SPSs) are designed to process data streams generated by data sources to produce new results for interested consumers. The computation is defined as a set of rules or queries that are specified by the user and deployed in the system. They allow users to submit rules that are continuously evaluated to produce new results as new input data is received. The data elements in the stream are usually annotated with timestamps that indicate timing or ordering relationships. In some systems, these annotations are used to identify time patterns, such as sequences or repetitions of specific elements [7]. SPSs organize the computation into a graph of primitive operators. Depending on the actual implementation, these operators can be physically deployed on a single node, or on several connected nodes. The primary focus of SPSs is to support high volumes of input data produced at a high rate and to provide a fast response time to consumers.

Existing SPSs differ from one another on a range of aspects, including the data model used to specify the input, the language used to define processing rules, and the processing techniques adopted. Cugola et al. [3] divided these systems into two classes: Data Stream Management Systems (DSMSs), developed by the database community, and Complex Event Processing (CEP) systems, which were elaborated by the community working on event-based systems.

## 5 Building an Urban Data Pipeline

Building an urban data pipeline is a four-phase process, which includes data acquisition, data integration, data processing, and presentation (see Fig. 2).

*Data acquisition* This phase includes obtaining relevant urban data streams and social media data stream by listening to selected social media sources. These data streams can be received and sent in various data formats such as XML, JSON, Text

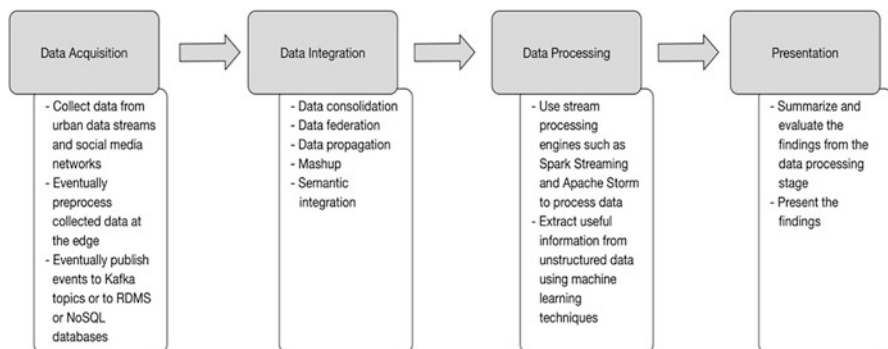


Fig. 2 Phases of an urban data pipeline

or Map format. Data can be written into data repositories such as RDBMS or Apache Cassandra. For IoT data streams, transports such as MQTT or Apache Kafka need to be supported.

*Data integration* This phase uses one or several of the techniques above for data integration to facilitate data processing and provide city stakeholders with the means to make informed decisions.

*Data processing* The results of the data processing phase will have a significant effect on the information and metrics in the presentation stage, thus the success of future decisions or actions a smart city might take. It will typically rely on a streaming processing engine such as SensorBee, WSO2 CEP, or Spark Streaming. The extraction of useful information from unstructured data often requires the use of machine learning techniques. These engines typically support various machine learning toolkits, such as Jubatus and Scikit-learn.

*Presentation and visualization* In this phase, the results from different processing approaches and analytics are summarized, evaluated, and shown to users in an easy-to-understand format. Visualization techniques may be used to present useful information; one commonly used interface design is the visual dashboard, which aggregates and displays information from multiple sources. Visualization will rely on open source tools, such as Pentaho and Helical Insight, or public or community editions of commercial tools such as Tableau public and Qlik Sense Desktop.

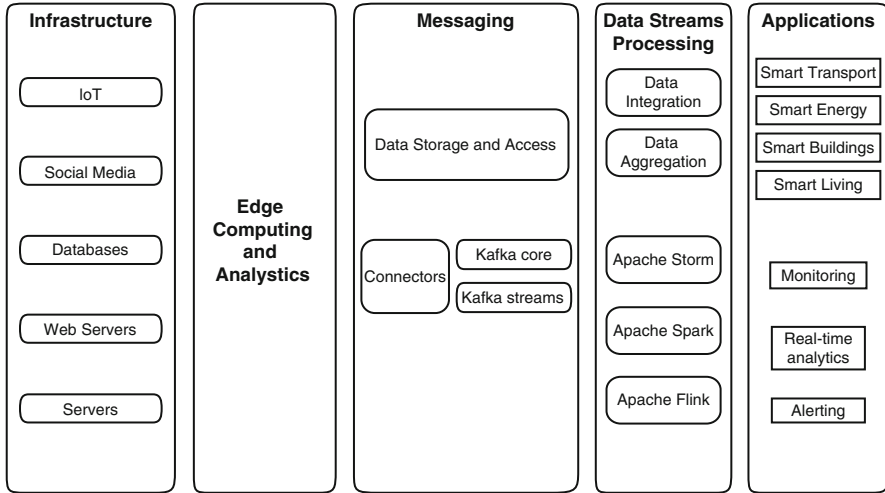
## **6 A Framework for Urban Data Management and Processing**

### **6.1 Architecture Overview**

Figure 3 depicts the proposed framework for the management and processing of urban data streams. It is composed of five tiers, namely infrastructure, edge computing and analytics, messaging, data streams processing, and applications.

#### **6.1.1 Data Sources**

The tier is made up of various smart city data sources, such as intelligent IoT devices, traditional databases, and Web servers. An IoT device detects and responds to certain inputs in its environment. The particular input may be light, motion, speed, vibration, pressure, water level, heat or any other environmental phenomenon. The reading of the device is then converted to a human-readable form or sent over a network to a gateway for further processing. An IoT device, usually with an IP address, can connect to a network to exchange data. Smart IoT devices automate a city's operations by collecting data on various physical assets (devices, equipment,



**Fig. 3** Architecture for Urban Data Streams Management and Processing

vehicles, buildings, facilities, etc.) to monitor their behavior and status, and using data collected to optimize resources and processes. IoT devices connect to gateways using WiFi or Ethernet connections in the form of a local area network (LAN) or using Bluetooth, ZigBee, and Ultra-Wideband (UWB) as a Personal Area Network (PAN).

Web server events and logs are a significant data source for the many systems in the city. Log streaming enables to resolve the connectivity issues and diagnose the causes of service interruptions. Also, clickstream analysis permits to assess the effectiveness of municipal online service delivery.

### 6.1.2 Edge Computing

As IoT sensors and smart devices generate considerable amounts of data, traditional data management systems and practices will no longer be enough to take full advantage of IoT. The basic idea behind Edge Computing (EC) is to place storage and compute resources at the edge of the network, close to the data generation location.

IDC predicted that by 2019, at least 40% of the data created by IoT would be stored, processed, analyzed and processed near or at the edge of the network [19]. Thus, the processing of urban data streams can be pushed from the cloud to the edge. EC reduces the bottleneck of traffic to the core network by processing data locally and accelerating data streams using a variety of techniques (i.e., caching and compression). Also, it allows shortening the end-to-end latency, which allows offloading heavy computation load from power constrained user equipment to the

edge. This can be very beneficial when IoT devices are deployed to remote sites with poor network coverage or when stakeholders aim to reduce the cost of expensive cellular connectivity technologies.

Edge devices, which are often battery powered, run full operating systems such as Linux, Android, or iOS. They process the raw data they receive from IoT devices and sensors, and they send commands to the actuators. They are connected to the messaging tier directly or via edge gateways. Edge gateways also run full operating systems and have unrestricted power, more CPU power, memory, and storage. They can aggregate data and support analysis at the edge of the network, and they act as intermediaries between the messaging tier and the edge devices. Both edge gateways and edge devices convey selected raw or pre-processed IoT datasets to the messaging tier components to be stored or analyzed using machine learning algorithms and analytics. They symmetrically receive commands, such as configurations or data queries, from the applications tier.

Centralized databases are essential for carrying out the different operations of the smart city systems. However, as data is constantly spreading from IoT sensors and devices to the edge, the central databases only need to handle the flow of data at a more controlled rate, such as once a minute. The use of edge servers, which typically have limited computing and storage capabilities, allows real-time data transmission and timely instructions. Data streams can be aggregated and fused at the edge, and then transported to the central databases as averages of data sensed over well-controlled time periods. Thus, moving data management from the primary databases to the edge of the network is crucial for dealing with real-time data streams.

### 6.1.3 Messaging

Similar to any distributed system, At the heart of the framework is the communication between the various machines that compose the city systems. The literature on inter-process communication describes numerous models, among which is the message queuing model. By adopting this model, the applications and services of the framework can communicate with each other by exchanging messages. The message queue provides temporary storage when the destination application is busy or not available. It provides an asynchronous communications protocol, in which an application that sends a message onto the message queue does not need an immediate response to continue its processing. Therefore, the tiers of the framework, in charge of the collection of data, will be decoupled from the other tiers, which are in charge of data processing and analytics.

The proposed architecture uses the Apache Kafka platform. Over the last few years, Apache Kafka is recognized as the most advanced messaging system capable of handling data streams in an efficient and scalable fashion. Kafka immutably stores the messages coming from multiple sources, called “producers,” in queues, called “topics,” which are organized into several partitions. Messages of a partition are indexed and saved with a timestamp. Other processes, called “consumers,” can



query the messages stored in Kafka partitions. Kafka usually runs on a cluster of one or many brokers (servers). Kafka partitions are replicated across the cluster brokers to guarantee fault tolerance.

Kafka has four APIs:

- **Producer API** – allows applications and services to publish streams of events into Kafka topics.
- **Consumer API** – allows applications and services to subscribe to Kafka topics of interest and process the stream of events.
- **Streams API** – permits to convert the input streams, stored in the topics, into other output streams that can be consumed by other applications.
- **Connector API** – permits to connect Kafka cluster with external sources such as key-value stores, and relational databases.

The messaging system and the data streams processing engines, described in next subsection, are typically deployed in a Fog server, a cluster of Fog servers, or eventually in a combination of Fog and Cloud servers. Fog-based solution are increasingly becoming attractive due to the low-latency and cost-effective services they can deliver. Also, an increasing number of cloud providers, such as Amazon, Google, and Confluent, are offering Message streaming as a Service by automating the setup, running and scaling of Apache Kafka.

#### **6.1.4 Data Streams Processing**

After enabling the efficient storage and access to of data in the messaging tier, the data streams processing tier allows city stakeholders to efficiently transform, mediate, and analyze these huge urban data streams. It is responsible for processing and analyzing the data streams stored in the Kafka topics so that applications of the smart city can use it to generate valuable insights. It provides an extensible set of established open source and commercial solutions for data processing such as Apache Storm, Apache Flink, Apache Samza, Amazon IoT, Google Cloud Dataflow, Amazon Elastic MapReduce that allows for processing both streaming and historical data, which is extremely important for current smart cities.

When deployed in conjunction with Apache Storm and Apache Spark, Apache Kafka can efficiently process data streams with the help of the APIs above.

#### **6.1.5 Applications**

The smart city applications are the consumers of the results of the data streams processing. The applications tier offers a broad set of techniques and tools for effective design, implementation, deployment, and operation of the smart city services and applications. The Service Oriented Architecture (SOA), expressed by Web services, has emerged over the last decade as the primary technology for delivering services over the Web. Web services are software systems designed to

support interoperability across platforms. They are neutral to languages, which makes them appropriate for access from heterogeneous environments. In the smart city context, Web services are vital components in the data and services integration effort because they offer a high layer of abstraction, which hides implementation details from the applications built from these services.

As an evolution of the initial concept of web services, microservices technology has recently emerged as a promising technology to support the design and implementation of scalable systems as opposed to traditional systems designed as monoliths. This technology advocates the creation of a system from a collection of small, loosely coupled services, each isolated, scalable and resilient to failure, and with its data. Services integrate with other services to create a coherent and extremely flexible system compared to the usual systems built today.

Service orientation and the microservices technology represent the main design principle to ensure interoperability between smart city systems and facilitate the provisioning of IoT integration, semantic integration, security assurance, and the creation of business processes that span several city systems. They are the basis for the development of a smart city service bus, which will be the backbone of the services of various government agencies and private businesses. They will permit the creation of new value-added services and the delivery of updated information at all times to city stakeholders, citizens, and businesses.

Smart applications could be developed in the areas of transport, energy and water consumption, resource management, safety monitoring, events and festivals monitoring, natural disaster alerting, etc. These smart applications could be used to improve the performance of different city departments, improve the efficiency of resource management, improve business profitability, save lives, minimize the risk of loss of life and resources, improve quality of life of citizens, and many more benefits.

## 6.2 Use Case Scenario

In this scenario, we use our proposed architecture to implement a real-time application to get the latest Twitter feeds concerning the **#Dubai** and **#Abu Dhabi** hashtags and process them by counting the number of accidents that are reported by the users in the cities of Dubai and Abu Dhabi.

Data streams (tweets) from Twitter have been recognized as a valuable data source for many smart cities in areas such as law enforcement, tourism, and politics (e.g., US presidential election). The Twitter Streaming API permits extracting datasets that are then used to perform sentiment analysis.

We have created a Kafka Producer that injects the Twitter feeds into the Kafka cluster. The Kafka Producer relies on the Twitter Hosebird Client (hbc) implementation. Retrieved tweets are stored in a Kafka topic. The Apache Storm and Spark components might read the messages using a Kafka consumer to inject them into their respective ecosystem.

The Twitter Hosebird is a server implementation of the Twitter Streaming API, which permits clients to receive tweets in near real-time. The Hosebird client includes two modules: hbc-core and hbc-twitter4j. The core module uses a message queue from which the consumer can poll for the raw string messages. The twitter4j module uses listeners and a data model on top of the message queue to offer a parsing layer [12].

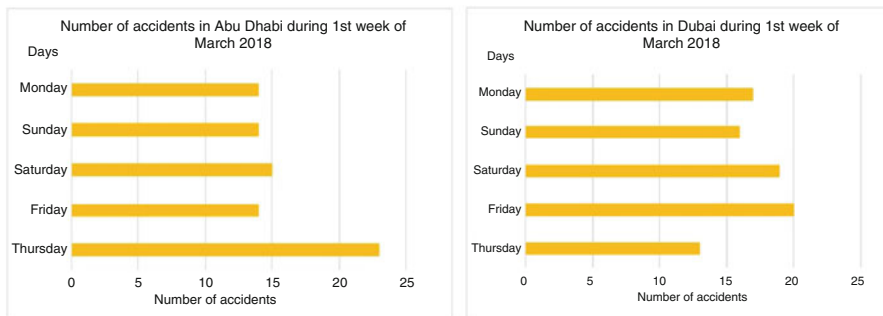
To access the Twitter Streaming API using the hbc client, we need to sign in for Twitter account to get the following OAuth authentication details:

- Customer key
- Customer secret key
- Access token key
- Access token secret key

Once the “HashTags” are received by Kafka, the Storm/Spark ecosystem can be used to process collected data. For this simple scenario, we used a Python script instead. The following steps are used for the analysis of the tweets:

1. Authenticate and connect to Twitter using Twitter API
2. Execute Kafka Producer to inject the tweets (#Dubai & #AbuDhabi) into the Kafka cluster.
3. Store collected tweets in a CSV file
4. Preprocess the gathered tweets by removing duplicates
5. Execute a Python script to process the tweets by counting the number of reported accidents on the roads of Dubai and Abu Dhabi on each day of the first week of March 2018.
6. Visualize results (see Fig. 4)

The results show that the number of reported accidents in Dubai is in general higher than the number of accidents reported in Abu Dhabi. However, it was the opposite on the last day of this investigation. The reason is that Abu Dhabi witnesses during this period of the year heavy fog than Dubai. The number of accidents during foggy days is in general higher than the average. These results indicate that



**Fig. 4** Analysis of the tweets concerning road accidents in Abu Dhabi and Dubai

users are suffering from the traffic conditions in the two cities and that municipal services should work hard to find solutions to the traffic problem, which is one of the challenges that most large cities are facing.

## 7 Discussion

Urban data integration and interoperability involve a lot of engineering effort, which requires the organization of processes for cooperation and consensus-building among all the city stakeholders concerned by the integration.

Several factors might impact the success of the urban data integration effort:

- Security and privacy issues (threats from hackers and intruders, the high cost of security applications and solutions, the privacy of personal data, etc.)
- Lack of interest in data integration by some city stakeholders and their resistance to sharing data.
- The high cost of IT experts skillful in data integration.
- Enormous effort required for the coordination of the various data resources having incompatible conceptualizations and representations.
- Lack of standards in the data integration field. Standardization would significantly lessen these challenges.

Already many initiatives, we mentioned earlier, are underway in several smart cities to integrate data obtained from multiple sources. It remains to be seen whether such initiatives can deliver promised smart services.

## 8 Conclusion

Urban data integration and interoperability is a challenging issue that modern cities are confronting today. Their data streams originate from the Internet of Things (IoT) devices and sensors, deployed in different parts of a city, as well as from social media networks such as Twitter and Facebook. The large volumes of urban data need to be harnessed to help smart city stakeholders make informed decisions on the fly. Furthermore, effective management and governance of smart city components rely on the ability to integrate and federate their data, process urban data streams locally, and use data analytics. This chapter describes our proposed framework that aims to assist in building a data pipeline for the acquisition and processing of urban data streams, urban data analytics, and creation of value-added services. The framework relies on recent technologies for data processing including IoT, edge computing, data integration techniques, cloud computing, and data analytics. The framework will facilitate real-time events detection, implementing alerting services, and mining the sentiments of citizens concerning the governance of their city together with rich visualization tools to help build monitoring dashboards.

## References

1. Anastasi G et al. (2013) Urban and social sensing for sustainable mobility in smart cities. in Proc. IFIP/IEEE Int. Conf. Sustainable Internet ICT Sustainability, Palermo, Italy, pp. 1–4.
2. Ciuccarelli P, Lupi G, Simeone L (2014) Visualizing the Data City: Social Media as a Source of Knowledge for Urban Planning and Management. Springer, Heidelberg.
3. Cugola G. and Margara A (2012) Processing flows of information: from data stream to complex event processing. *ACM Comput. Surv.* 44 (3) 15:1–15:62.
4. DataTorrent (2016) Real-Time Event Stream Processing – What are your choices? <https://www.datatorrent.com/blog/real-time-event-stream-processing-what-are-your-choices/>. Latest access on Dec. 05. 2017.
5. Hashem I A T, Chang V, Anuar N B, Adewole K, Yaqoob I, Gani A, Ahmed E, and Chiroma H (2016) The role of big data in smart city. *International Journal of Information Management*, vol. 36, no. 5, pp. 748–758.
6. Jugel U, Jerzak Z, Hackenbroich G, and Markl V (2014) M4 - A Visualization-Oriented Time Series Data Aggregation. *PVLDB*, Volume 7 Issue 10, pp. 797–808.
7. Margara A, Urbani J, van Harmelen F, and Bal H (2014) Streaming the Web: Reasoning over dynamic data. *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 25, pp. 24–44.
8. Ramparany F and Cao Q H (2016) A semantic approach to IoT data aggregation and interpretation applied to home automation. *International Conference on Internet of Things and Applications (IOTA)*, pp. 23–28.
9. Rosi A et al. (2011) Social sensors and pervasive services: Approaches and perspectives. *Proc. IEEE Int. Conf. PERCOM Workshops*, Seattle, WA, USA, pp. 525–530.
10. Tim L M V, Birte U, Vignesh S, Maria E. N (2014) Analyzing Tweets to Aid Situational Awareness. *Advances in Information Retrieval*, Vol. 8416, *Lecture Notes in Computer Science*, pp. 700–705.
11. Vaccari A, Liu L, Biderman A, Ratti C, Pereira F, Oliveira J, Gerber A (2009) A holistic framework for the study of urban traces and the profiling of urban processes and dynamics. *Proc. 12th International IEEE Conference on Intelligent Transportation Systems*, IEEE Press, New York, pp. 273–278.
12. Hosebird Client (hbc). Retrieved from: <https://github.com/twitter/hbc>
13. Barnaghi P, Tönjes R, Höller J, Hauswirth M, Sheth A T, Anantharam P. (2015) Citypulse: Real-time iot stream processing and large-scale data analytics for smart city applications. *ict-citypulse.eu Deliverable D.3.2: Data Federation and Aggregation in Large-Scale Urban Data Streams*.
14. Computing (2014) London Westminster City Council introduces smart parking system. <https://www.computing.co.uk/ctg/news/2323408/london-westminster-city-council-introduces-smart-parking-system>. Latest access on Dec. 05. 2017.
15. Diginomica.com (2017) Bright Lights. Smart City. San Diego’s pioneering IoT platform. <https://diginomica.com/2017/10/31/bright-lights-smart-city-san-diegos-pioneering-iot-platform/>. Latest access on Dec. 05. 2017.
16. Fraunhofer FOKUS Institute (2012) FixMyCity. <https://www.fokus.fraunhofer.de/04c06110dd0adebc>. Latest Access on Dec. 05. 2017.
17. Gyrard A and Serrano M (2016) Connected Smart Cities - Interoperability with SEG 3.0 for the Internet of Things. *Advanced Information Networking and Applications (AINA 2016) Workshops*.
18. Haas L M, Lin E T, and Roth M T (2002) Data integration through database federation. *IBM Systems Journal*, 41(4), pp. 578–596.
19. IDC.com (2017) IDC FutureScape: Worldwide Internet of Things 2017 Predictions. <https://www.idc.com/research/viewtoc.jsp?containerId=US40755816>. Latest Access on Dec. 05. 2017.

20. Loshin D (2009) *Data Consolidation and Integration*. Master Data Management, Elsevier, pp. 177–199.
21. Trilles S, Calia A, Belmonte Ó, Torres-Sospedra J, Montoliu R, and Huerta J (2016) Deployment of an open sensorized platform in a smart city context. *Future Generation Computer Systems*, vol. 76, pp. 221–233.
22. [Wired.com](https://www.wired.com/2015/06/copenhagenize-worlds-most-bike-friendly-cities/) (2015) The 20 Most Bike-Friendly Cities on the Planet. <https://www.wired.com/2015/06/copenhagenize-worlds-most-bike-friendly-cities/> Latest access on Dec. 05. 2017.
23. Aloï G et al. (2016) A Mobile Multi-Technology Gateway to Enable IoT Interoperability. *Proc. IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 259–264.
24. Aloï G, Caliciuri G, Fortino G, Gravina R, Pace P, Russo W, Savaglio C (2017) Enabling IoT interoperability through opportunistic smartphone-based mobile gateways. *J.Netw.Comput.Appl.* 81, pp. 73–83.
25. Blackstock M and Lea R (2014) IoT interoperability: A hub-based approach. *Proc. IEEE International Conference on the Internet of Things (IOT)*, pp. 79–84.
26. Asensio A, Marco A, Blasco R, Casas R (2014) Protocol and Architecture to Bring Things into Internet of Things. *International Journal of Distributed Sensor Networks*, Article ID 158252.
27. Blackstock M, Kaviani N, Lea R, and Friday A (2010) MAGIC Broker 2: An open and extensible platform for the Internet of Things. *Proc. the 2010 Internet of Things (IOT)*, pp. 1–8.
28. Perera C, Jayaraman P P, Zaslavsky A, Christen P and Georgakopoulos D (2014) MOSDEN: An Internet of Things Middleware for Resource Constrained Mobile Devices. *47th Hawaii International Conference on System Sciences, Waikoloa, HI*, pp. 1053–1062.
29. Feng G, Intizar Ali M, Curry E, and Mileo A (2017) Semantic Discovery and Integration of Urban Data Streams, *Future Generation Computer Systems*, Volume 76 Issue C, pp. 561–58.
30. Kettouch M, Luca C, Khorief O, Rui Wu and Dascalu S (2017) Semantic data management in Smart Cities. *International Conference on Optimization of Electrical and Electronic Equipment (OPTIM 2017) & Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP 2017)*, pp. 1126–1131.

# Index

## A

Access control, 289  
Active RFID tags, 12–13  
Adaptability  
  of service, 265–266  
  service interaction factors, 260  
Aggregated data, 344–345  
airMAX, 304  
Alarm service, 59  
Alberta (apartment complex), 362  
Amazon Elastic Compute Cloud (EC2), 57  
Amazon Elastic MapReduce, 390  
Amazon IoT, 390  
Amsterdam, Netherlands, 5  
Apache Flink, 390  
Apache Hadoop, 64  
Apache Kafka platform, 388–390  
Apache Samza, 380, 390  
Apache Storm, 380, 390  
  complex events, 69  
  dynamic priority scheduler, 69  
  isolation scheduler, 69  
  multiple supervisor nodes, 68  
  multi-tenant storm cluster, 69  
  nimbus node, 68  
  Raw Event 1 and Raw Event 2, 70  
  spouts and bolts, 68  
  static priority scheduler, 69  
Application Programming Interfaces (APIs), 58  
ARM Cortex-M4F, 363  
Asymmetric service interoperability  
  approach, usefulness of, 279  
  compliance illustration, 271–273  
  conformance illustration, 277–278

  data schema model, 273–274  
  mechanism, 269, 270  
  request message validation, 269, 270  
  structural service interoperability,  
    274–277  
ATC Update, 319–320  
Atoll wireless planning software, 304  
Authentication and authorization service, 58  
Autoregressive Integrated Moving Average  
  (ARIMA) model, 372–374  
Available time for charging (ATC), 322  
Axis Video Hosting System (AVHS), 296  
Azure Media Services (AMS), 294–295

## B

Backward coupling ( $C_B$ ), 264, 266  
Barcelona, Spain, 3  
802.15.4-based technologies  
  6LoWPAN, 15  
  Thread, 15  
  Zigbee, 14  
Beijing, China, 4  
Big data, 34  
Black Hole and Gray Hole attacks, 24  
Blockchain, 25  
Bluetooth, 16, 388  
Bluetooth Low Energy (BLE), 16  
  RF subsystem, 363  
BQ25504 Battery management, 364  
Bridges, 59–61  
Building Automation System (BAS), 365  
Built-in data types, data schema model,  
  273  
Bulk data transfer, 58

**C**

Cassandra, 380  
 CC2530, 363  
 CCTV camera network, 291  
 Centralized charging system
 

- cellular network communication enabled
  - charging system, 315–316
  - enabling internet of, 316–317

 Changeability, of service, 265–266  
 Cloud-based video management services
 

- AMS, 294–295
- experimental testbed, 297
  - multiple video stream performance, 300–302
  - single video stream performance, 298–300
- video management platform setup, 298

 Stratocast, 296, 297  
 Cloud computing, 289
 

- elasticity, 57
- 5G ICEMO, 117
- green computing, 57
- intelligent context-sensitive computing, 220
- low IT investment, 57
- middleware architecture
  - alarm service, 59
  - authentication and authorization service, 58
  - bridges, 59–61
  - data transfer service, 58
  - research collaboration, 61–63
  - resource management service, 58–59
  - for tool invocation, 60
- quality of service, 58
- security and performance, 58
- service level agreement, 58
- types, 57

 Cloud encoder, 295  
 Cloud interoperability, 279  
 Cloud of Things, 11  
 Cloud radio access networks (C-RAN), 121
 

- small cells in, 122–124

 Cognitive radio networks (CRNs), 43  
 Complex event processing (CEP) systems, 386  
 Compliance, 271–273, 275–276  
 Computer vision, 290  
 Concentric Value Circles (CVC) Model, 128, 131  
 Conformance, 275–276, 277–278  
 Connection Data (CD), 234–235  
 Connective interoperability, 261, 262  
 Connectivity
 

- layer, 287–288

- smart video surveillance system, 292

 Connector API, 390  
 Consumer, 259  
 Consumer API, 390  
 Contact profile, mobile messaging simulation, 240  
 Controlled vocabularies, 385  
 Control plane, 288  
 Coupling model, 263–265  
 Coupling, service interaction factors, 260  
 Create, Read, Update, Delete (CRUD)
 

- approach, 268–269

 CrowdMAC, 93  
 Crowdsensing, 341–344  
 Crowdsourcing, 345
**D**

Daily profile, mobile messaging simulation, 240  
 Data aggregation, 288  
 Data analytics platforms and resource management
 

- MapReduce/Hadoop
  - closed systems, 65
  - energy aware resource management, 67–68
  - open systems, 66–67
- streaming data analytics
  - complex events, 69–70
  - isolation scheduler, 69
  - SPS and DPS, 69
  - storm, 68 (*see also* Apache storm)

 Data consolidation, 383, 384  
 Data display, 288  
 Data federation, 383, 384  
 Data format transformation, 288  
 Data plane, 288  
 Data propagation, 383, 384  
 Data schema model, 273–274  
 Data storage, 288  
 Data stream management systems (DSMSs), 386  
 Data transfer service, 58  
 Day node (DN), STree, 233  
 Deep learning algorithms, 290  
 Delay/disruption tolerant networking (DTN), 317  
 Denial of Service (DoS) attacks, 23  
 Device control, 289  
 Device gateway, 288  
 Device log, 289  
 Device monitor, 288  
 Digital nudging, 334



Distributed charging system  
 V2I communication enabled charging system, 317–318  
 V2V communication network enabled charging system, 318–322  
 Document-based interoperability, 279  
 DoubleDip's application, 360, 361  
 Dropout regularization, 214

**E**

Eavesdropping, 23, 26  
 EETAC Update, 320–321  
 Electric vehicle (EVs), 312  
 centralized charging system  
 cellular network communication enabled charging system, 315–316  
 enabling internet of, 316–317  
 CS-selection  
 communication technologies, 314  
 ITS, 313–314  
 “on-the-move” mode, 313  
 scalability of, 314–315  
 distributed charging system  
 V2I communication enabled charging system, 317–318  
 V2V communication network enabled charging system, 318–322  
 energy sustainability, 326–328  
 hybrid charging system  
 V2I communication network enabled charging system, 322–325  
 V2V communication network enabled charging system, 326  
 security and privacy, 328  
 Energy-efficient crowdsensing, 95–96  
 Energy harvesting  
 in building environments  
 multi-decade sustainable operation, 360–361  
 opportunities and efficiencies, 358–360  
 multi-hop routing  
 optimization formulations, 368–371  
 routing problem, 367–368  
 simple routing strategies, 371–374  
 sensor node design  
 exterior wall average daily heat flow, 365  
 exterior wall intra-day heat flow, 366  
 modern ultra-low power RF, 362–363  
 TE harvesting node, 363–364  
 WSNs, 355–357  
 Energy sustainability, 326–328  
 Enterprise cloud, 57

Environmental friendliness (EF), 332  
 Erceg-Greenstein propagation model, 304  
 Expected earliest time available for charging (EETAC), 320  
 Experimental testbed, cloud-based video management services, 297  
 multiple video stream performance, 300–302  
 single video stream performance, 298–300  
 video management platform setup, 298  
 Extensible markup language (XML), 382–383

**F**

Facebook, 380  
 5G micro operator, 125–126  
 Fifth Generation (5G) network  
 CRN arrangement, 43  
 and IoT  
 characteristics, 45  
 communication overhead and storage mechanisms, 48  
 data traffic, 46  
 HetNet, 47  
 multi-objective optimization problem, 47  
 in Smart Cities, 48–50  
 spectrum availability, 46–47  
 system security, 47–48  
 things, sensing capabilities of, 47  
 needs, 42  
 potential limitations, 43  
 realization of SCs, 44–45  
 resource allocation problems, 43  
 specifications and goals, 42  
 spectrum management and optimization, 43  
 5G wireless micro operators for integrated casinos and entertainment (5G ICEMO)  
 anti-counterfeiting lottery, 139–141  
 autonomous transport scenario, 141–142  
 business model, 132–133  
 cloud computing, 117  
 C-RAN, 134  
 CVC model, 128, 131  
 fog computing, 117  
 gaming industry, 118  
 IoT paradigm, 116  
 MCC and MEC, 117  
 mega jackpots, 138–139  
 network function virtualization, 119  
 offered services, 116  
 SDN-based architectural framework, 119  
 small cells, 117

5G wireless micro operators for integrated casinos and entertainment (5G ICEMO) (*cont.*)  
 system architecture, 135  
 urban IoT, 116  
 wireless network architecture, 135–138  
 Fort McMurray (apartment complex), 362  
 Forward coupling ( $C_F$ ), 263–264

## G

Global controller (GC), 312  
 Google+, 380  
 Google Cloud Dataflow, 390  
 Google maps, 346  
 Green transportation  
 back-end processing, 347  
 discouraging and encouraging factors, 332  
 edge-processing, 347–348  
 inform and nudge  
 personal recommendations, 349–350  
 system credibility, 349  
 system features, 349  
 user interaction, 350  
 methods for analysis, 346–347  
 sensing elements  
 aggregated data, 344–345  
 crowdsensing, 341–344  
 crowdsourcing, 345  
 IoT sensors, 340–342  
 static data, 345–346  
 smart nudging  
 architecture, 338–340  
 nudging, 333–335  
 personalisation, 335–336  
 situational awareness, 337

## H

Hadoop File System (HDFS), 64  
 Hello flood attack, 24  
 HetNet, 121  
 Hidden Markov model, 374  
 High-density deployment pattern, 302–304  
 Hilbert Space Filling Curve based linearization technique, 103  
 Hybrid charging system  
 V2I communication network enabled charging system, 322–325  
 V2V communication network enabled charging system, 326  
 Hypermedia as the engine of application state (HATEOAS)

## I

IBM InfoSphere Streams, 88  
 Information and Communication Technologies (ICT), 228  
 Information sharing systems, 345  
 Integrated Emergency and Security System, 291  
 Intel Galileo board, 168  
 Intelligent context-sensitive computing  
 Calvin runtime, 221  
 cloud computing, 220  
 cloud-to-fog connections, 207  
 edge processing and smart cities, 220  
 fog computing, 206, 221–222  
 location and time, 207  
 machine learning models  
 advantages, 212  
 artificial neural networks, 212  
 challenges and benefits, 218–219  
 dataset, 213  
 implementation and performance analysis, 214–217  
 model design, 213–214  
 multi-stream data prediction, 218  
 proof of concept model, 211  
 recurrent neural networks, 212  
 Mobile Fog, 220  
 motivating scenarios, 208  
 sensor prediction approach, 222–223  
 system architecture  
 assumptions, 208–209  
 data sharing mechanism, 209  
 goals, 209  
 JAMScript, 210–211  
 neural networks, 209–210  
 Intelligent (hybrid) framework, 230  
 Intelligent system, mobile messaging, 229  
 Intelligent transportation systems (ITS), 312, 313–314  
 Internet-enabled devices, 257  
 Internet-of-Things (IoT), 256  
 application layer, 11–12  
 architecture vs. TCP/IP model, 154–155  
 car parking system, 153 (*see also* Smart parking application scenario)  
 classification, 152  
 connectivity layer, 9–10  
 countermeasures and research issues  
 for communication security, 26  
 IDS, 26  
 for physical security, 24–25  
 data acquisition and control layer, 8–9  
 data management layer, 10–11

- definition, 152
  - 5G ICEMO, 116
  - 5G network
    - characteristics, 45
    - communication overhead and storage mechanisms, 48
    - data traffic, 46
    - HetNet, 47
    - multi-objective optimization problem, 47
    - in Smart Cities, 48–50
    - spectrum availability, 46–47
    - system security, 47–48
    - things, sensing capabilities of, 47
  - IERC definition, 152
  - IoT Security Threats, 22–24
  - organizations and standards, 153
  - RFID technology, 152
  - security threats, 21
    - physical vulnerabilities, 22
    - wireless communications, 23–24
  - smart city management, 91–92
  - smart living, 3
    - in Medellin, Colombia, 6–8
    - in New York City, USA, 6
    - in Seoul, South Korea, 6
  - smart mobility, 3
    - in Barcelona, Spain, 3
    - in Beijing, China, 4
    - in Santander, Spain, 4–5
  - smart sustainability, 3
    - in Amsterdam, Netherlands, 5
    - in Las Vegas, USA, 5–6
    - in Padova, Italy, 5
  - waste management, 90
  - wireless communications technologies
    - issues, 18–20
    - long-range communications, 16–18
    - medium-range communications, 13–16
    - short-range communications, 12–13
  - Interoperability, 256
    - asymmetric service (*see* Asymmetric service interoperability)
    - cloud, 279
    - connective, 261, 262
    - document-based interoperability, 279
    - pragmatic, 261, 262
    - semantic, 261, 262
    - service-based, 279
    - service interaction factors, 260
    - symmetric service, 266–268
  - Intrusion Detection System (IDS), 26
  - In-vehicle sensors, 343
  - IoT applications
    - big data management and analytics, 101–103
    - knowledge management
      - RDF and SPARQL, 104
      - RDF quads, 106
      - SPARQL query processing, 105
    - security and privacy, 107–108
  - IoT-based urban infrastructure system
    - implementation, 158
    - smart parking application scenario, 156–157
  - IoT sensors, 340–342
  - IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), 15
- J**
- JavaScript Object Notation (JSON), 271–272, 278, 383
- K**
- Kafka, 380
- L**
- Large message (LM), 229
  - Large-scale video surveillance system, 287
  - Las Vegas, USA, 5–6
  - Las Vegas Valley Water District (LVVWD), 5
  - Learning algorithm, 236
    - learning without QNP, 237
    - learning with QNP, 237–238
  - Live streaming, 294, 295
  - Local area network (LAN), 388
  - Location and proximity sensors, 341
  - Long-range communications
    - LoRaWAN, 10, 17–18
    - Sigfox, 18
  - Low-density deployment pattern, 302–304
  - Low power nodes (LPN), 121
  - LTE networks, 10
- M**
- Macro base stations (MBS), 121
  - MapReduce/Hadoop
    - closed systems, 65
    - energy aware resource management, 67–68
    - open systems, 66–67
  - Markov decision process (MDP), 231
  - Mashup applications, 385
  - Mean-time-between-failures (MTBF), 361
  - Medellin, Colombia, 6–8

- Medium message (MM), 229
  - Medium-range communications
    - BLE, 16
    - WiFi, 15–16
  - Merkle Hash Tree technique, 94
  - Message(s), 266
    - ratio of, 249, 250
    - types, 229
  - Message node (MN), STree, 233
  - Microsoft's Windows Azure, 57
  - Mobile crowdsensing system
    - components and functionalities, 83
    - high-level overview, 83
    - people-centric approach, 82
  - Mobile crowdsourcing
    - advantages, 92
    - behavioral economics, 97–98
    - data-driven incentive mechanisms, 98–101
    - energy-efficient incentive mechanisms, 95–96
    - incentive-based practical infrastructure, 93
    - people-centric approach, 92
    - platform-centric model, 96
    - security and privacy, 94–95
    - Stackelberg game-based incentive mechanism, 97
    - sustainable incentive mechanisms, 92
    - user-centric model, 96
  - Mobile edge computing (MEC), 326, 327
  - Mobile Instance Messaging (MIM), 228
  - Mobile messaging, 228
    - connections and, 242–243, 245, 246–248
    - contemporary framework, 228
    - intelligent messaging architecture and learning algorithm
      - actions, 234
      - learning algorithm, 236–238
      - server reward, 235
      - STree, 232–234, 238–239, 241
      - user reward, 235–236
    - intelligent system, 229
    - latency average, 243–245, 247–249
    - message
      - ratio of, 249, 250
      - types, 229
    - messaging framework, 230
    - MN and another without MN, 241, 242
    - reinforcement learning, 231–232
    - server load, 243, 244–246
    - simulation model
      - engine components, 240
      - initialization, 240–241
      - at k-th step, 241
    - UA and SD, 242, 243, 245, 247, 248
  - Mobile sensing, 343
  - Multi-factor authentication (MFA), 25
  - Multi-hop routing, 367
    - optimization formulations, 368–371
    - routing problem, 367–368
    - simple routing strategies, 371–374
  - Museum tour guide system
    - android application, operations of, 72, 73
    - features, 71–72
    - handle multiple museums, 74
    - implementation technology, 73–74
    - NFC enabled, 72
- N**
- NanoZ-CC2530 devices, 364
  - National Institution of Standards and Technologies (NIST), 289
  - Network Address Translation (NAT), 230
  - Network-attached storage (NAS) drive, 298
  - Network file system (NFS) protocol, 298
  - Network topology, 288
  - New York City, USA, 6
  - Nordic Semiconductors nRF52, 363
- O**
- On-demand streaming, 294
  - Online learning, 228
  - On-premises video management solutions, 293–294
  - Open Systems Interconnection (OSI) reference model, 261
  - Opportunistic sensing, 343
- P**
- Padova, Italy, 5
  - Pan-tilt-zoom (PTZ) functionality, 287
  - ParkNet, 88
  - Participatory sensing, 343
  - Passive RFID tags, 13
  - Peer-to-peer (P2P), 228
    - connection type, 234
    - framework, 230
  - Performance, service interaction factors, 260
  - Personal area network (PAN), 388
  - Personal Environmental Impact Report (PEIR), 90
  - Personalized vehicular crowdsensing, 178–179
    - algorithm design, 194–196
    - characteristics, 189
    - definitions and notations, 190–193
    - execution time, 199–200
    - experiment setup, 196–198

- GoSense vs. WBG algorithm performance, 198
  - participant recruitment, 189
  - peak workload, 198
  - problem formulation, 193–194
  - system model and assumptions, 190
  - window based scheduling, 200
  - Personal sensing, 344
  - Photovoltaic (PV) harvesters, 358
  - Piezoelectric (PE) harvesters, 358–359
  - Power distribution wiring, 357
  - Pragmatic interoperability, 261, 262
  - Privacy-preserving truth discovery (PPTD) framework, 93
  - Privacy, smart video surveillance system, 292
  - Private clouds, 57
  - Producer API, 390
  - Provider, 259
  - Public clouds, 57
  - Public transportation bus (PTB), 318
  - Public vehicular crowdsensing, 177–178
    - definitions and assumptions, 181–182
    - participant selection, 180
      - budget, 186–187
      - coverage function, 183
      - genetic algorithm, 185
      - greedy algorithm, 186–188
      - offline algorithm, 184
      - online algorithm, 184
      - participant commitment, 183
      - Points of Interests, 186
      - pruning algorithm, 188
      - spatial coverage, 184, 188
    - problem statement, 182
    - system model, 180–181
- Q**
- $Q$ -learning, 232
  - QNode, 233–234
  - QNode Probability Links (QNP), 234
- R**
- Radio Frequency (RF) harvesting, 359
  - Radio-Frequency Identification (RFID), 12, 38
  - Raspberry Pi, 363
  - READY4SmartCities (European project), 385
  - Real time data transfer, 58
  - Record and list structured types, data schema model, 274
  - Redundant array of independent disks (RAID-1) setup, 298
  - Reinforcement learning (RL), 229, 231–232
  - Reliability, service interaction factors, 260
  - Representational State Transfer (REST), 258, 268, 269
  - Research and engineering cloud, 57
  - Research Platform for Smart Facilities Management (RP-SMARF), 61
    - automatic data movement, 62
    - batch and interactive modes, 62
    - GUI, 61
    - multi-tenancy, 61, 62
    - processing real-time data streams, 63
    - tools and dataset discovery, 62
    - unification of heterogeneous resources, 62
  - Reservation aggregating, 321–322
  - Resource management service, 58–59
  - RESTful web services, 385
  - RFID tags, 12–13
  - Road side units (RSUs), 312, 317
  - Root, STree, 233
  - Route framework, 230
  - Routing attack, 23–24, 26
- S**
- Santander, Spain, 4–5
  - Scalability
    - service interaction factors, 260
    - smart video surveillance system, 292
  - Selective forwarding, 24
  - Semantic interoperability, 261, 262
  - Sensors, green transportation
    - aggregated data, 344–345
    - crowdsensing, 341–344
    - crowdsourcing, 345
    - IoT sensors, 340–342
    - static data, 345–346
  - Seoul, South Korea, 6
  - Server, mobile messaging, 240
  - Server reward, 235
  - Service-based interoperability, 279
  - Service-Oriented Architecture (SOA), 256, 258, 390
  - Short-range communications, 12–13
  - Sigfox, 18
  - Signal Processing Platform for Analysis of Structural Health (SPPLASH) tool, 60
  - Simulation area, mobile messaging, 240
  - Small cells, 117
    - advantages, 123
    - C-RAN, 122–124
    - femtocells, 123
    - four-phased approach, 123–124
    - microcells, 123
    - picocells, 123

- Small message (SM), 229
- Smart camera, 287
- Smart City (SC)
  - Big data, 34
  - challenges, 36, 120
  - cooperating citizens and municipal authorities, 34–35
  - definition, 33
  - effective WSN, 34
  - environmental monitoring, 90
  - 5G use cases and business models, 122
  - infrastructure and frameworks
    - CEP systems, 85
    - CityZen platform, 85
    - IoT and IoS, 84–85
    - issues, 86
    - SCCIR framework, 85
    - smart city architecture, 85
    - smart city reference model, 85
    - SmartCrowd framework, 85–86
  - integrated casinos and entertainment (ICE) casino service delivery, 127
    - challenges, 126
    - equipment vendors, 127
    - market opportunities and pressures, 126
    - physical environment, 127
    - societal opportunities and pressures, 126
    - supervisory agencies, 128
    - technology opportunities and pressures, 126
    - technology vendors, 127–128
  - inter-woven systems for smart parking, 35
  - IoT and WSN, 37–41
  - IoT technology and retail industry, 91–92
  - issues, 36, 120
  - in Las Vegas, 120, 142–143
  - LPNs and MBS, 121
  - in Macao, 120, 143–144
  - mobile crowdsourcing
    - advantages, 92
    - behavioral economics, 97–98
    - data-driven incentive mechanisms, 98–101
    - energy-efficient incentive mechanisms, 95–96
    - incentive-based practical infrastructure, 93
    - people-centric approach, 92
    - platform-centric model, 96
    - security and privacy, 94–95
    - Stackelberg game-based incentive mechanism, 97
    - sustainable incentive mechanisms, 92
    - user-centric model, 96
  - NFV and SDN, 121
  - RAN, 121
  - in Singapore, 120, 144–145
  - smart solutions, 36
  - software services (*see* Software services)
  - superior, reliable and secure communication networks, 34
  - transportation
    - facilitating efficient parking, 88
    - sensing and analyzing road and traffic conditions, 86–88
  - video surveillance system (*see* Video surveillance system)
  - waste management
    - Enevo, 90
    - fill-level estimation, 89–90
    - intelligent sensor-based containers, 89
    - IoT-enabled waste collection, 90
    - monitoring solid waste bins, 89
    - sensors/IoT, 89
    - SmartBin Live platform, 90
- Smart living, 3
  - in Medellin, Colombia, 6–8
  - in New York City, USA, 6
  - in Seoul, South Korea, 6
- SmartMart, 91–92
- Smart mobility, 3
  - in Barcelona, Spain, 3
  - in Beijing, China, 4
  - in Santander, Spain, 4–5
- Smart nudging
  - architecture, 338–340
  - nudging, 333–335
  - personalisation, 335–336
  - situational awareness, 337
- Smart parking application system, 4, 156–157
  - data processing operation, 166
  - data reception block
    - network evaluation, 162–164
    - wireless transmission, 165–166
    - WSN implementation, 164–165
  - methodology
    - data sensing, 159–161
    - operation diagram, 159
    - presentation, 168–170
- Smart restaurant management system (SRMS)
  - bill payment, 76
  - components, 74
  - description, 74
  - interactive menu, 76
  - overview, 75
  - parking spots, 76
- Smart sustainability, 3

- in Amsterdam, Netherlands, 5
    - in Las Vegas, USA, 5–6
    - in Padova, Italy, 5
  - Smart traffic management system, 290
  - Smart video surveillance system, 286
    - application layer, 289–290
    - applications, 290–292
    - challenges of, 292
    - connectivity layer, 287–288
    - management layer, 288–289
    - video acquisition layer, 287
  - Snow sensor prototype, 342
  - Software defined networking (SDN), 118, 279
  - Software services, 255
    - asymmetric interoperability, 257
    - asymmetric service interoperability
      - approach, usefulness of, 279
      - compliance illustration, 271–273
      - conformance illustration, 277–278
      - data schema model, 273–274
      - mechanism, 269, 270
      - request message validation, 269, 270
      - structural service interoperability, 274–277
    - compliance and conformance, 258
    - distributed service interoperability, 258
    - factors, smart cities, 257
    - interoperability problem, analysis
      - aspects of, 259–260
      - coupling model, 263–265
      - model of interoperability, 261–263
      - service adaptability and changeability, structural model of, 265–266
    - SOA, 258
    - symmetric service interoperability, 266–268
  - SQLite, 167
  - State-Action Reward State-Action (SARSA), 229
  - State of charge (SOC), 312
  - Static data, 345–346
  - Stratocast, 296, 297, 299
  - Stream processing systems (SPSs), 386
  - Streams API, 390
  - STree, 232–234
    - adaptation of, 238–239
  - Structural service interoperability
    - compliance and conformance relations, 275–276
    - Mapping Records to Lists and Lists to Records, 277
  - Structured data types, data schema model, 273
  - Superior, reliable and secure communication networks, 34
  - Symmetric service interoperability, 266–268
  - Syntactic interoperability, 261, 262
  - System-on-Chip (SoC) philosophy, 363
- T**
- TEC1-12703 Peltier module, 363
  - Thermoelectric (TE) harvester, 359
  - Thread, 15
  - Time node (TN), STree, 233
  - Tower, mobile messaging, 240
  - Transmission Control Protocol/Internet Protocol (TCP/IP) model, 154
  - Transportation
    - facilitating efficient parking, 88
    - sensing and analyzing road and traffic conditions
      - FixMyStreet platform, 87
      - Nericell system, 87
      - RoadEye system, 87–88
      - Sipresk platform, 87
      - Ushahidi platform, 87
  - Twitter, 380
  - Twitter Streaming API, 392
- U**
- Ultra-Wideband (UWB), 388
  - Union types, data schema model, 273–274
  - Unmanned aerial vehicles (UAVs), 326
  - Urban data streams (UDS)
    - architecture
      - applications, 390–391
      - data sources, 387–388
      - data streams processing, 390
      - edge computing, 388–389
      - messaging, 389–390
    - batch processing, 385
    - data aggregation, 382
    - hub-based approach, 382
    - impact factors, 393
    - physical hardware sensors, 381
    - social media analytics, 382
    - software sensors, 381
    - SPSs, 386
    - Twitter HashTags on road accidents, Abu Dhabi and Dubai, 391–393
    - urban data integration, 382–385
    - urban data pipeline, 386–387
  - Urban informatics
    - themes, 82
    - Wikipedia definition, 82
  - User experience, 228, 235
  - User, mobile messaging, 240

User reward, 235–236  
 User-reward average (UA), 242, 243, 245, 247, 248

## V

Vehicle-to-Infrastructure (V2I)  
 communication, 317  
 Vehicle-to-vehicle (V2V) communication, 316  
 Vehicular Ad hoc NETWORKS (VANETs), 316  
 Vehicular crowdsensing  
 central server, 179  
 future works, 201–202  
 personalized crowdsensing, 178–179  
 public crowdsensing, 177–178  
 Video acquisition layer, 287  
 Video surveillance system, 286  
 deployment challenges of  
 scale-up estimation, 306–308  
 scenarios, 302–304  
 simulation, 304–306  
 management platform, 292–293  
 cloud-based video management  
 services, 294–297  
 on-premises video management  
 solutions, 293–294  
 smart video surveillance system,  
 components for, 286  
 application layer, 289–290  
 applications, 290–292  
 challenges of, 292  
 connectivity layer, 287–288  
 management layer, 288–289  
 video acquisition layer, 287  
 Vivotek Application Development Platform  
 (VADP), 296

## W

Waste management  
 Enevo, 90  
 fill-level estimation, 89–90

intelligent sensor-based containers, 89  
 IoT-enabled waste collection, 90  
 monitoring solid waste bins, 89  
 sensors/IoT, 89  
 SmartBin Live platform, 90  
 Waze, 88  
 Weather and environment sensors, 341, 342  
 Web Services Description Language (WSDL),  
 279  
 WiFi, 15–16  
 Wikipedia, 345  
 Wireless communications technologies  
 long-range communications, 16–18  
 medium-range communications, 13–16  
 short-range communications, 12–13  
 Wireless connectivity, 303  
 Wireless sensor networks (WSNs), 34, 39–41,  
 355–357  
 benefits and challenges, 39–41  
 decision-making process, 40  
 development, 41  
 in military applications, 39  
 Worm Hole attack, 24  
 WSN nodes, 356  
 WSO2 CEP, 380

## X

Xeoma, 293  
 XProtect, 293, 301

## Y

Yahoo, 345  
 Yarn, 380

## Z

ZigBee, 14, 388  
 ZigBee RF subsystem, 363  
 ZoneMinder, 293, 299, 301