

# Big Data Processing using Internet of Software Defined Things in Smart Cities

Murad Khan<sup>1</sup> · Javed Iqbal<sup>1</sup> ·  
Muhammad Talha<sup>2</sup> · Muhammad Arshad<sup>3</sup> ·  
Muhammad Diyan<sup>4</sup> · Kijun Han<sup>4</sup>

Received: 2 February 2018 / Accepted: 19 April 2018 / Published online: 25 April 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract** Software Defined Networks (SDN) has been attracting researchers, scientist, and technology experts from both academia and industry to enhance the current ICT stakes and networking paradigm. The beauty of SDN is the division of Control and Data planes and make it easy for the engineers to modify the networking protocols without visiting onsite devices. Similarly, smart cities concept has been coined recently, where a plethora of smart devices will be connected and providing tons of services to the citizens, officials, and governmental departments. The Internet of Things (IoT) plays a vital role in guaranteeing such services. Few efforts have been made

---

✉ Kijun Han  
kjhan@knu.ac.kr

Murad Khan  
murad.csit@suit.edu.pk

Javed Iqbal  
javed.ee@suit.edu.pk

Muhammad Talha  
mnaseem@ksu.edu.sa

Muhammad Arshad  
amuhammad@soharuni.edu.om

Muhammad Diyan  
m.diyana@netopia.knu.ac.kr

- <sup>1</sup> Department of Computer Science, Sarhad University of Science and Information Technology, Peshawar, KPK, Pakistan
- <sup>2</sup> Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia
- <sup>3</sup> Department of Computer Science, Sohar University, Sohar, Oman
- <sup>4</sup> School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea

to merge SDN and IoT with the sole purpose of efficient Data retrieval and achieve remotely configurable networks. In this paper, we explicitly define the Internet of Software Defined Things architecture and bring it to Smart Cities as a use-case. Our 3-tier architecture consists of Data Collection, Data Management, and Application levels that are further connected via two intermediate levels working on SDN principles. Followed by the potentials of SDN and IoT for Smart Cities, we evaluated our proposed architecture using Spark and GraphX with Hadoop Ecosystem and the results shows that efficient transfer of Data over SDN for real-time processing is achieved.

**Keywords** Software Defined Networks · Big data analytics · Internet of Things · Smart devices · Smart city

## 1 Introduction

Information and Communications Technologies (ICT) started with the simple concept of communications and became the necessity and part of our everyday lives. ICT has a vital role to enable ubiquitous connectivity to the users with the services as well as the things around them. These services include health, transportation, emergency response, shopping, utilities, economy, weather, etc., and referred as smart services in this paper. Information related to smart services are ubiquitously made available to the citizens through varying underlying technologies such as Internet of Things (IoT) that make these citizens smart to pre-act any forthcoming situations. In addition, the inter-service information exchange is requisite to make these services smarter to shape a smart city. Having said that, a large volume of data (Approx. 90 percent as reported by IBM in 2012) has been generated recently in the last two years, which is drawing consideration from both public and private sectors [1]. It has given rise to the new research field, i.e. Big Data. Big data has become one of the hottest topics in both academia and industry. Big data represents data sets so large and complex that traditional data management tools or processing methods are inadequate to deal with it. Big data is mostly characterized by “5Vs” (initially it was referred to as “3Vs”; two have been added recently): volume (size of data set), variety (range of data type and source), velocity (speed of data in and out), value (how useful the data is), and veracity (quality of data) [2]. It is very challenging to store, analyse, gather and process the data by using the present methods because it is comprised of high velocity, highly dynamic, immense volume and numerous types of information.

Meanwhile, the traditional network gained great success by adopting a hierarchical structure. But for the closed systems of network devices in smart cities, we have to configure many devices with high complexity when business requirements change. At the same time, researchers also cannot deploy new protocols in the real environment especially when we are focusing on a plethora of embedded devices. With the rapid growth of Internet traffic (expected global traffic to reach  $1.6 \times 10^{21}$  B [3]), users desire greater bandwidth and various new services. It is a big challenge, so we need a high-performance and high-stability network architecture that can be configured flexibly to support smart city applications. Almost half decade ago, the aforementioned issues

led Nick McKeown to present OpenFlow [4] and based on OpenFlow, the concept of software-defined networking (SDN) was presented in [5].

The main idea of SDN is to detach the control plane from the forwarding plane, to break vertical integration, and to introduce the ability to program the network. SDN allows logical centralization of feedback control, and decisions are made by the “network brain” with a global network view, which eases network optimization. In SDN, data plane elements become highly efficient and programmable packet forwarding devices, while the control plane elements are represented by a single entity, the controller. Compared to traditional networks, it is much easier to develop and deploy applications in SDN. In addition, with the global view in SDN, it is straightforward to enforce the consistency of network policies. SDN represents a major paradigm shift in the evolution of networks, introducing a new pace of innovations in networking infrastructure.

## 2 Potentials of SDN and Big Data in Smart Cities: A Birds Eye View

SDN architecture has three folded advantages. First, the open architecture of SDN realizes the centralized control and automatic management of networks. Managers can design, deploy, operate, and maintain networks on a centralized SDN controller rather than configure a lot of heterogeneous devices.

Second, the network operating system and network applications can be deployed on servers that adopt X86 architecture and can control data forwarding by Open-Flow. Thus, SDN can provide various open APIs to flexible program networks.

Third, SDN decouples the data plane and control plane by using OpenFlow and virtualizes networks. A network becomes a logical resource that can be configured through software. For these advantages, the core idea of SDN has been used in the field of routers to build an open, flexible, and modularized reconfigurable router [6].

While some excellent work has been done on big data and SDN, these two important areas have traditionally been addressed separately in most previous works. However, on the one hand, SDN, as an important networking paradigm, will have a significant impact on big data applications. In particular, several good features (e.g., separation of the control and data planes, logically centralized control, global view of the network, ability to program the network) can greatly facilitate big data acquisition, transmission, storage, and processing. For example, big data is usually processed in cloud data centers. Compared to traditional data centers, SDN-based data centers can have better performance by dynamically allocating resources in data centers to different big data applications to meet the service level agreements (SLAs) of these big data applications [7–15].

On the other hand, big data, as an important network application, will have a profound impact on the design and operation of SDN. Specifically, with the global view of the network, the logically centralized controller in SDN can obtain big data from all the different layers (i.e. from physical to application layers) with arbitrary granularity. From experience in cross-layer design, we have learned that although sharing information among different layers can improve network performance, the network becomes so complex that traditional approaches are inadequate to design and optimize

such networks. Fortunately, big data analytics, which leverages analytical methods to obtain insights from data to guide decisions, can help the design and operation of SDN. For example, with big traffic data analytics, it is easier for the controller to perform traffic engineering to improve the performance of SDN.

In this paper, we propose the integration of IoT with programmable devices (SDN-IoT) to aid a variety of Big applications in smart cities followed by the potentials of SDN in building smart cities. The proposed architecture starts with data gathering process from various smart city-enabling technologies such as Smart Homes, Smart Grids, Intelligent Transportation System (ITS), weather forecast intelligent systems, and so on. After the data aggregation as a second level, we present a detailed Data Processing and Management (DPM) level where Big Data analysis has been used to filter the useful data. We also have evaluated the performance of our architecture using Hadoop Ecosystem. On top of DPM, we have also identified the Applications level along with varying data flow that is supported by our evaluated Big Data architecture. We have considered the Future Internet architecture recently named as Named Data Networking (NDN) aka Content-Centric Networking (CCN) at the application level. Finally, we provide open issues that provide a roadmap to follow by the active researchers in the said domain.

### 3 Proposed Scheme

The storage and processing systems can be further extended with the help of multi-level active storage architecture. The proposed layering architecture is broadly divided into four different layers of various functionalities such as enabling read and write operations. In the following sections, the proposed layering architecture is presented with each functionality of the layer for enabling high-performance computing. After presenting the proposed layering architecture, the design and working of the proposed system is elaborated with the help of different case studies.

#### 3.1 Proposed Layered Architecture

As we know that Big Data is composed of huge amount of data obtained from various heterogeneous sources. Therefore, analyzing such huge amount of data in real-time and using single level processing can lead to severe problems such as processing of data in real-time, disseminating data with the citizen in real-time, etc. The proposed model uses IoT based smart environment where different entities and objects interact with each other. Similarly, the proposed model integrates a variety of data obtained from health-centers, hospitals, etc. for designing and planning smart hospitals, mobile health centers, case of security, healthcare, elderly age people and kids, and transportation system, machine-to-machine network, wireless sensor network, and vehicular network, etc. Figure 1 shows that the proposed layering architecture is consisted of four different layers. The discussion on these four different layers is shown in following section.

*Level I* This level collects data from various sources and object connected with the internet. The level I aggregates data and uses different filters to normalize data

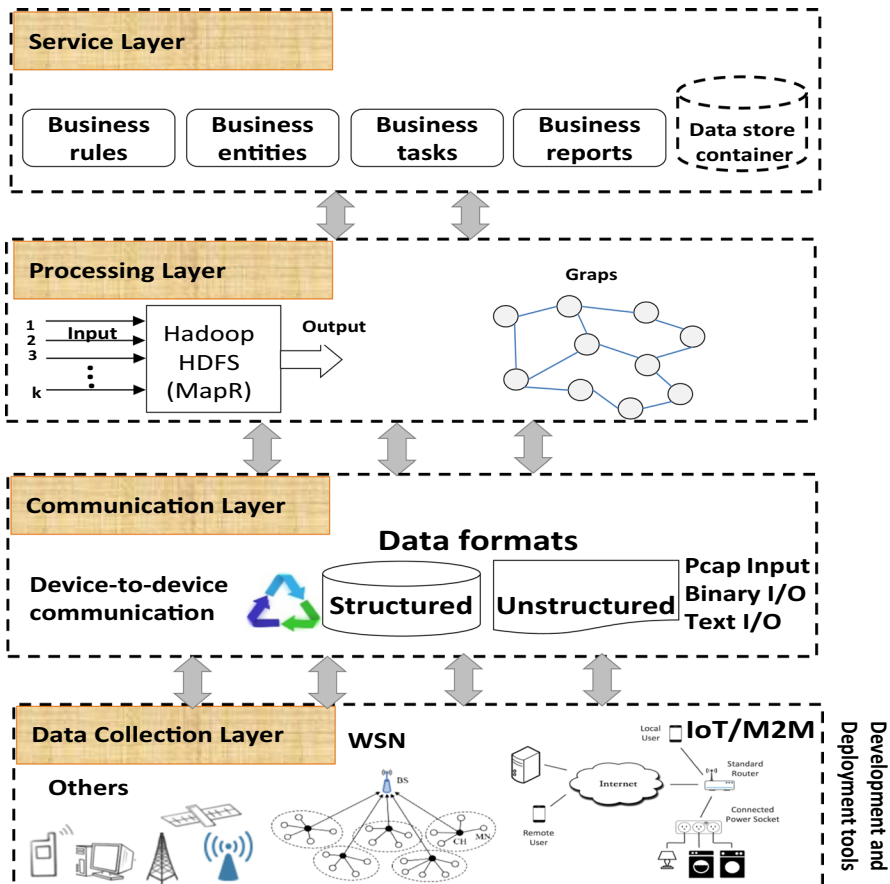


Fig. 1 Four-tier communication model

and presented in suitable and meaningful form to further layers. As we know, data is generated from various sources, therefore, it consists of various formats, different point of collection, time stamps, and periodicity. Similarly, every big data has various requirements such as security, privacy, versatility, and quality, etc. Moreover, the meta data always contains higher data than the actual measures. Thus, pre-filtration, normalization and registration techniques are applied at this layer to present the data in clear format and meaning form. In addition, the redundancy is also removed at this layer.

*Tier-II* This layer is responsible for end to end delivery between various devices. The data is aggregated at various aggregated points that are installed in different locations. The data is then presented and converted in suitable and manageable format for further processing.

*Tier-III* This layer is used to process, and store data based on various inputs from user. Since Hadoop is used to process in offline, we need to integrate Hadoop with a real-time data processing system such as SPARK, STORM, VoltDB, etc. Moreover,

a MapReduce paradigm is implemented in the processing of data using Hadoop with a real-time system. Similarly, at the same layer a data storing system such as HDFS, HIVE, HBASE, etc. is used to store data for future analysis and further processing and dissemination with citizen of a community or a municipality authority.

*Tier-IV* The service is layer is responsible for providing the possible interfaces to the proposed system for injecting and collecting information. Similarly, the entire system can be managed and edit with the help of such system. Further, this system is implemented at various locations, cloud, and sites to enable a human or a program to perform necessary tasks such as update, modify, etc. Moreover, each entity in the system is assigned with a global ID that can be used to manage each item and entity in the system individually. Furthermore, the entity in the system can perform self-actions such as self-configuration, self-healing, etc. for better performance and working. Such actions are controlled by installing a system called vendor control to maintain and update the information to a central management station for the record. As the data size is quite huge and it is difficult for multiple users and objects to interact with the data, therefore, an intelligent system is adopted to control the involvement of humans, users, and other objects at this layer. This intelligent module is able to perform various tasks such as request generator, session initiating, setting up communicating rules, interact with heterogeneous objects and terminating the session.

### 3.2 Proposed System Architecture

The execution flow of the proposed mechanism consists of three main levels and two intermediate levels. The data from various IoT-enabled embedded devices is collected and send to the data processing level using the SDN-enabled networking. In addition, the data is passed to the respective users after dealing out in data processing level. Figure 2 shows the flow diagram of the proposed framework.

#### 3.2.1 Data Collection and Intermediate Levels

A number of smart services of a smart city are considered for data aggregation and collection purpose i.e. smart health services, smart transportation services, etc. The sensors are attached with these services to efficiently gather data and pass it to the upper levels using the Intermediate Level 1 (IL1). The IL1 is further consists of various Aggregator Points (AP). The APs is used to reliably transfer the sensed data over the SDN-enabled network. The APs are further classified into three levels i.e. zone, local, and global APs to reduce the congestion level on the SDN-core network. The Zone Level APs (ZLAP) are in charge of aggregating high detail information from each sensor of a smart city unit. The Local Level APs (LLAP) is responsible for aggregating sensor data from similar units such as hospital data from e-health services, transportation data from roads, etc. Finally, the Global Level APs (GLAP) gathers data from the LLAP and send it to the SDN-core network.

The SDN-core network consists of an SDN controller which maps the traffic from GLAP to the SDN-network. The SDN-controller is programmed to perform several tasks. For example, it can differentiate sensed data based on the sensor IDs, topol-

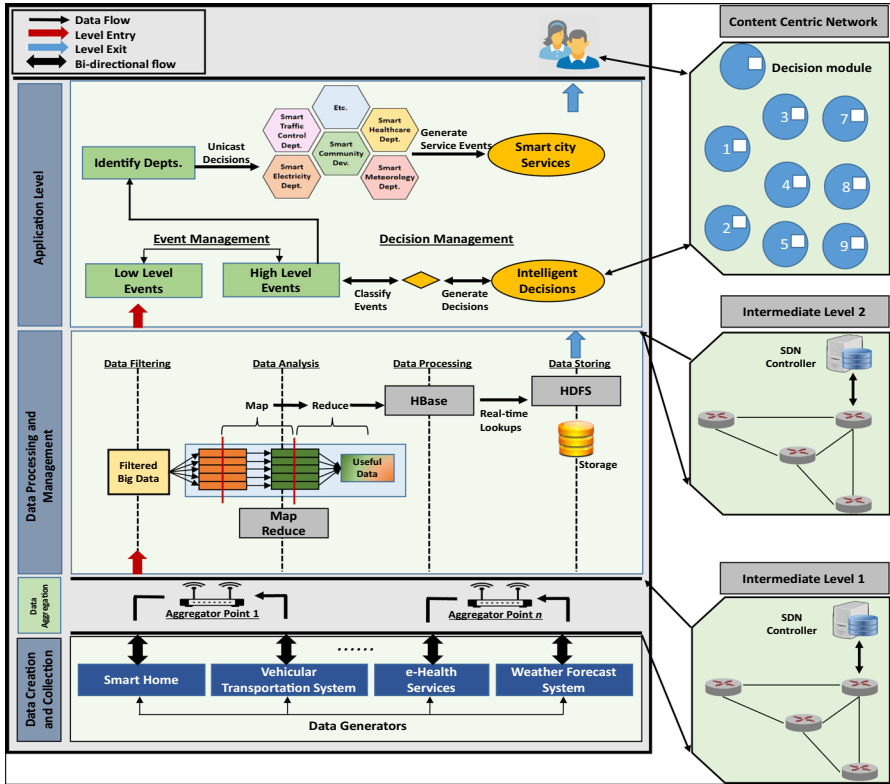


Fig. 2 The proposed architecture

ogy control, optimizes the duty cycle of sensors attached with each AP, performing routing decision based on the application requirements. In order to enable application-specific routing of data, the SDN-controller uses a priority application table in each SDN-enabled router. Moreover, with thousands of interconnected IoT-enabled devices generate a huge amount of data which ultimately slows down the network operations. The literature consists of various mechanisms to detect congestion level on a link. For example, a pre-defined threshold of link utilization is proposed by Kandula [16]. The proposed traffic mechanism declared a link as a congested link if the data traffic on the link exceeds 70% of the total capacity of the link. We also use a similar pre-defined threshold with 75% of link usage. We should mention that the design constraint of this work is to develop a framework for efficiently transfer BigData over SDN-networks. Therefore, we are using the existing traffic engineering technique to enhance the performance of the SDN-network. However, we can give some fruitful suggestion to choose a traffic engineering mechanism for controlling and routing traffic over an SDN-network. For example, the data generated by a huge number of sensors in an IoT environment require high-speed processing switches and routers. The SDN-controller follows two different mechanisms to retrieves information of the link statistics from SDN-switches i.e. the poll and push-based methods. The push-based mechanisms are

relatively fast than the poll based and, therefore, our suggestion is to use it for high-speed data routing and switching. We use the two-tier congestion handling mechanism proposed by Chen [17]. In work proposed by Chen, the SDN-controller maintain a global view of the entire network by retrieving information of links from each switch in the SDN-network. This information is further used to control the load on each link in the network.

### 3.2.2 Data Management and Processing Level

The data obtain from the IL1 is passed to data processing level to normalize the data to a meaningful form and extract the required information. For example, the road congestion data can be helpful for a smart city user to reach to a destination in a shorter amount of time. The data processing of huge amount of data always requires high time and processing power. Thus processing real-time data is a challenging job using the existing conventional methods. The proposed data processing level employed an efficient map-reduce paradigm for data analysis using Hadoop ecosystem and GraphX and SPARK for real-time processing of data. In addition, the Hadoop Distributed File System (HDFS) is used for storing and manipulation purpose. The Hadoop ecosystem uses a heterogeneous cluster Hadoop system to process a huge amount of data. The literature consists of various techniques to assign job among the Hadoop cluster system. Real-time processing requires a scheduling mechanism to split the job into sub-jobs in a map-reduce part. However, once jobs are loaded to the map-reduce system, it is not possible to change the maximum number jobs. Therefore, we used an adaptive job scheduling technique to adjust the load on the map-reduce system dynamically. Each job tracker uses two different parameters to switch the part of the job from the current node to another node i.e. CPU utilization and memory requirements. The switching of the jobs is performed in real-time based on the amount of a load of a cluster. However, in a typical Hadoop system, it is impossible to change the task from one node to another, once the Hadoop process starts operating. In the case of heterogeneous Hadoop clusters, implanting the fixed job assignment does not produce optimize results. The proposed scheduling mechanism overcome two main problems exist due to the fixed job assignment i.e. (1) a high-performance node remains in the idle state and (2) a low-performance node always remains in the busy state. This tradeoff between high and low-performance node make the system unstable for heterogeneous Hadoop clusters. Our proposed job scheduling strategy checks the load on each node in runtime. A node always demands new jobs, if its current workload drops from 75% of its total capacity. In each turn, the job scheduler checks the current load of each node and assign jobs accordingly. Thus assigning a full capacity of a node during a single turn optimizes by incorporating the load parameter. Similarly, the output of the map-reduce system is passed to HDFS module for storing and other relevant operations.

### 3.2.3 Application Level

The data from HDFS system is passed to the application level using the SDN network via Intermedia Level 2 (IL2). The working of the IL2 is similar to IL1 except for the

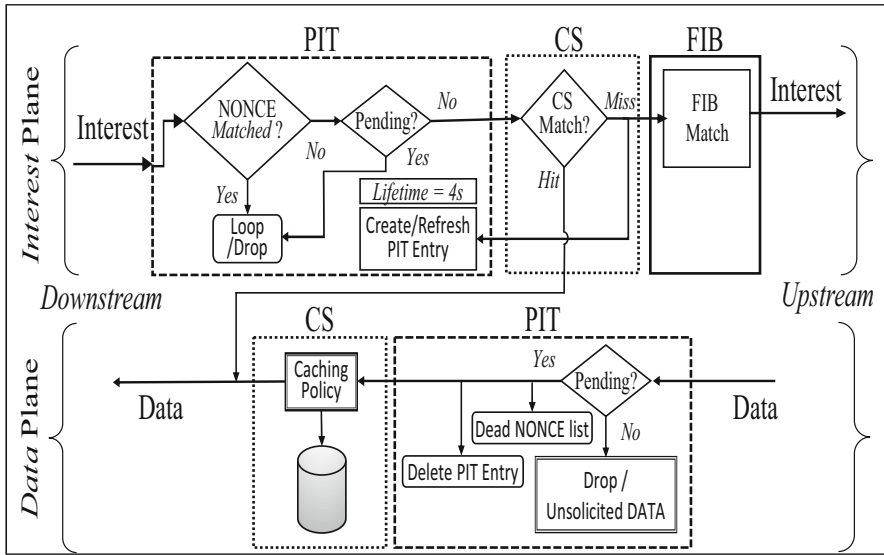


traffic level difference on the SDN-network is less compared to IL1. The application level is further divided into two parts, (1) event and decision management and (2) Named Data Network (NDN). In the event and decision module, an event is generated based on the data from the data and processing level. This event is further broadcasted to the respective departments which can process it to the concerned user. The events generated by the event management module are further classified into two groups i.e. high and low-level events. The high-level events are the most important events, and they are processed by the decision management module on a priority basis. The low-level events are stored in the decision module until a notification is sent back to the data processing level. Once the data processing level receives the notification from the decision management module. It sends back an acknowledgment to the decision module and the decision module discards the low-level event. In order to clearly understand the working of the event and decision module, we explain it through an example scenario.

Assume, the sensors deployed at a particular city collect the data of a road congestion level. The data processing level passes the data to the event and decision module. If the congestion level on the road is more than a predefined threshold, the decision module considers this is a high-level event, and it passes the data to the transportation department. However, if it is less than the pre-defined threshold, it will discard the event after sending a notification back to the data processing level. Once the decision module decides the necessary action on the data, the next step is to send data to the respective user via the respective department. Such sending of data is performed either fulfilling a user request or automatically broadcast to a group of users following the hierarchical model presented in Sect. 1. However, in both cases, we prefer ICN based networks to communicate with the user considering user's interest [18]. Therefore, we use a Name Data Network (NDN) to efficiently fulfill the user requirements either by Sub/Pub or Pull-based communications. Each decision module works as an NDN node, and it further consists of three entities i.e. Pending Interest Table (PIT), Content Store (CS), and a Forward Information Base (FIB). The PIT table consists of pending interests and their unique Nonce values to avoid Interest looping problem. The storing of incoming content and routing of content to other decision module is performed using CS and FIB, respectively [19]. Whenever a user is interested in a particular data, it generates an interest packet and sends it to the NDN network. The interest is processed, and the content is delivered from a decision module following the NDN network. The precise overview of NDN network operations is shown in Fig. 3.

#### 4 Implementation Results and Analysis results

The proposed system is implemented using Spark and GraphX with Hadoop single node setup on UBUNTU 14.04 LTS Core™ i5 machine with 3.2 GHz processor and 4 GB memory. For real-time traffic, we generated Pcap packets from the datasets by using Wireshark libraries and retransmit them towards the developed system. Hadoop-pcap-lib, Hadoop-pcap-serde, and Hadoop Pcap Input libraries are used for network packets processing and generating Hadoop Readable form (sequence file) at collection and aggregation unit so that it can be processed by Hadoop and GraphX. GraphX is



**Fig. 3** Working of NDN in the proposed architecture

used to build and process graphs with the aim of making smart transportation decisions. We have considered the massive volume of data from [9, 10, 20]. The intensity of the traffic varies from time to time on the same road. The intensity analysis at the various time of the day helps the authorities to manage and make a proper plan for the traffic on that particular time.

Initially, the analysis is performed on Aarhus city traffic. The speed analysis on the intensity of traffic is carried out as shown in Fig. 4. When the intensity of traffic is more, i.e. more vehicles on the road between two points, the average speed of the vehicles is greater. The fall in some vehicles on the road results in a rise in the average speed. We can easily notice a higher number of cars, i.e. 25–30, the average speed is very low at various times of the day, shown as a red color graph. Whereas, at a lower intensity, i.e. 0–10 shown as a blue line, the average speed of the vehicles is quite higher. There are also some abnormalities exist with a lower number of vehicles the average speed is also lower. This might be because of the construction of the roads or some other incidents. Normally, the distance is conserved to measure the time to reach the destination. However, we observed that the number of vehicles and the average speed also affects the time to get to the destination. Figure 5 shows the blockage of one of the roads in Aarhus city. Based on the proposed scheme, the average speed of the vehicles is too low even when there are a minimum number of vehicles on the road. We can see that most of the road blockage is at morning times on different days. This is because of the road construction and working at morning time. Similarly, we can easily perceive, the increase in the number of vehicles on the road results in more time to reach to the other point. More traffic on the road reduces average speed of the vehicles, which results in more time to reach the destination. As a result of this

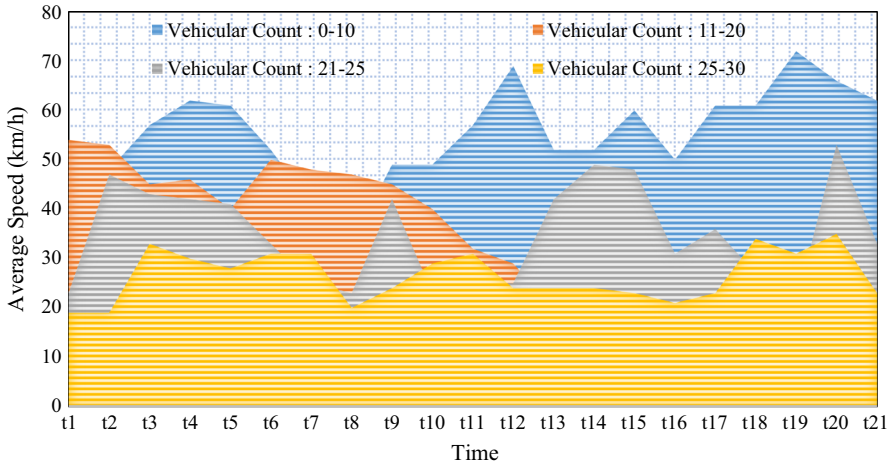


Fig. 4 The average speed of a vehicle

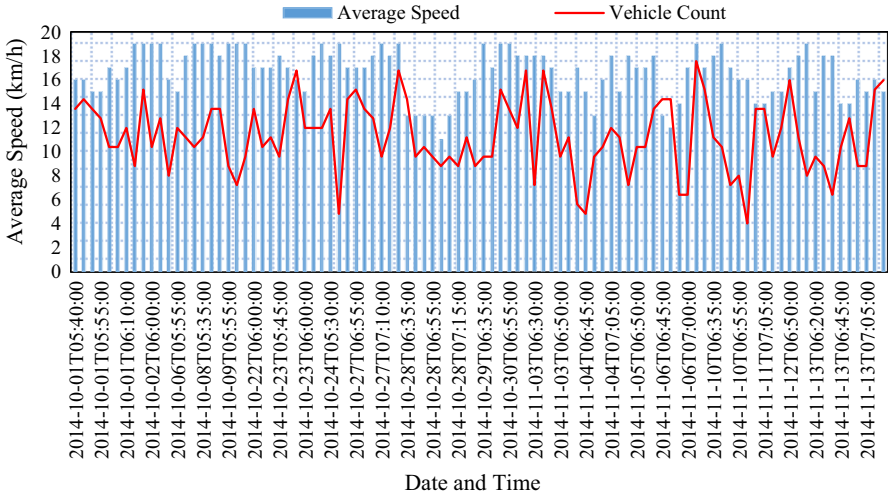


Fig. 5 Average speed during various date and time

phenomena, we take real-time traffic information to calculate the shortest and quickest path between source and destination rather than only the distance information.

Figure 6 shows the percentage of humidity inside the home. Humidity plays an important role in user behavior in case if the user is doing physical exercise or any other activity. Moreover, if there is an increase in the humidity, the usage of electricity also increases. For this case, the proposed scheme exploits the phenomena of learning mechanism. Sensors measure humidity, and this data is transferred to our proposed scheme for experiencing the level of humidity. Our proposed scheme considers several reading, and thus creating one threshold during the month of December 2016. Based on the previous knowledge, the proposed scheme will predict for the month of January

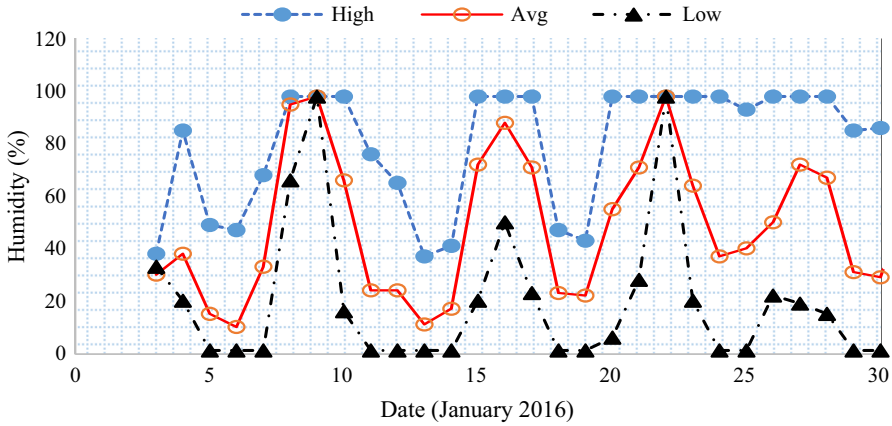


Fig. 6 Humidity inside home

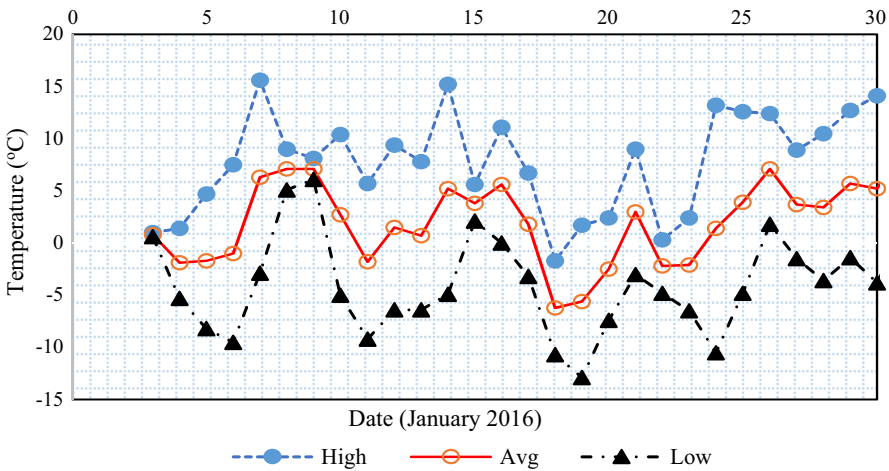
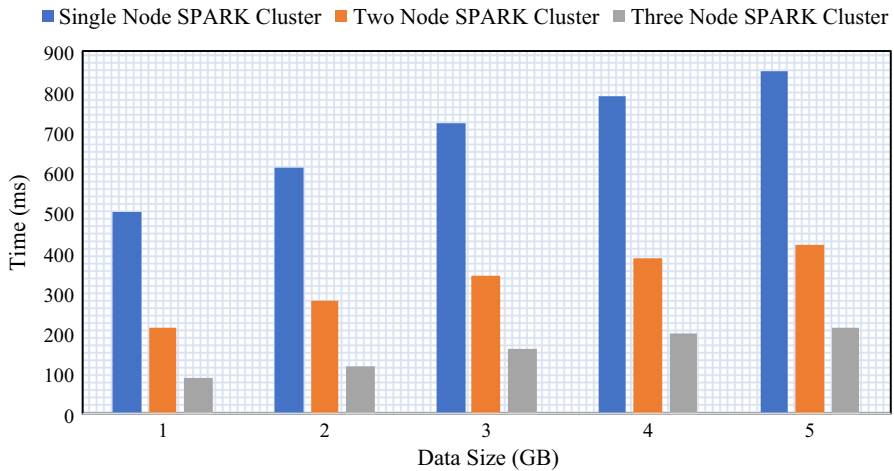


Fig. 7 Outdoor temperature

2016. Thus, the user will react accordingly if humidity is increased or decreased as shown in Fig. 6. Similarly, the same technique is followed for outside temperature as shown in Fig. 7.

The effect of processing time on increasing in the graph is also examined while evaluating the efficiency of the system. We tested the system by increasing the number of clusters, as shown in Fig. 8. In addition, the Hadoop ecosystem alongside SPARK clusters performs significantly by analysing the data with less time by increasing the number of SPARK clusters. In addition, it directly depends on the data size and the corresponding information/feature that are to be collected from the data sets. In the results of Fig. 8, the users are interested in various features such as time to destination, congestion level on road, etc. However, in future we are planning to come up with results of same vehicular traffic features but at different roads.



**Fig. 8** Time to process data using different SPARK clusters

## 5 Conclusion

In this article, we present an architecture to process BigData using Hadoop ecosystem alongside Spark and GraphX. We proposed a three-level architecture for efficiently gathering data from the sensors attached to various appliances in a smart city in the data collection level and pass it to the upper levels for processing using SDN and NDN networking. The collection of huge amount of data and pass it to Hadoop ecosystem is carried out using a novel architecture of hierarchal actuator point concept. A scheduling mechanism is implanted in the Hadoop ecosystem to efficiently balance the load on the Hadoop ecosystem. The decision module perform various decisions on the data based on various thresholds. The user of the smart city have provided with the options to fetch data from the decision module based on request-response functionality and the decision module sends automatic requests if a threshold is violated.

Finally, the proposed system is tested on various data from authentic sources to test its performance. The simulation results reveal that the user is informed with the requested data in less time and accurate results. In addition, the processing time required by increasing the clusters following proposed scheduling technique efficiently enhances the working of the system.

**Acknowledgements** This work was supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea (NRF) through the Creative Human Resource Training Project for Regional Innovation (2014). This study was supported by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005). This work was supported by the Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia, through the Research Group under Project RG-1437-037.

## References

1. Gandomi, A., Haider, M.: Beyond the hype: big data concepts, methods, and analytics. *Int. J. Inform. Manage.* **35**(2), 137–144 (2015)
2. Gudivada, V.N., Baeza-Yates, R., Raghavan, V.V.: Big data: promises and problems. *IEEE Comput.* **48**(3), 20–23 (2015)
3. Index, C.V.N.: *Forecast and Methodology, 2013–2018* (2013)
4. McKeown, N., et al.: OpenFlow: enabling innovation in campus networks. In: *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74 (2008)
5. McKeown, N.: Software-defined networking. In: *INFOCOM Keynote Speech*, vol. 17, no. 2, pp. 30–32 (2009)
6. Xu, K., et al.: Toward a practical reconfigurable router: a software component development approach. *IEEE Netw.* **28**(5), 74–80 (2014)
7. Hong, W., Wang, K., Hsu, Y.-H.: Application-aware resource allocation for SDN-based cloud datacenters. In: *Proceedings of the International Conference Cloud Computing and Big Data 2013*, Fuzhou, China (2013)
8. Han, Y., et al.: Software defined networking-based traffic engineering for data center networks. In: *Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium*, Taiwan (2014)
9. Kolozali, S., Bermudez-Edo, M., Puschmann, D., Ganz, F., Barnaghi, P.: A knowledge-based approach for real-time iot data stream annotation and processing. In: *Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings 2014)*, Taipei, Taiwan (2014)
10. Gramaglia, M., Trullols-Cruces, O., Naboulsi, D., Fiore, M., Calderon, M.: Vehicular networks on two Madrid highways. In: *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, vol. 3, pp. 423–431. Singapore (2014)
11. Silva, B.N., Khan, M., Han, K.: Big data analytics embedded smart city architecture for performance enhancement through real-time data processing and decision-making. *Wirel. Commun. Mob. Comput.* **2017**, 1–12 (2017)
12. Rowley, J.: The wisdom hierarchy: representations of the DIKW hierarchy. *J. Inf. Sci.* **33**(2), 163–180 (2007)
13. Bischof, S., Karapantelakis, A., Nechifor, C.-S., Sheth, A., Mileo, A., Barnaghi, P.: Semantic modeling of smart city data. In: *Position Paper in W3C Workshop on the Web of Things: Enablers and services for an open Web of Devices*, Berlin, Germany (2014)
14. Tönjes, R., Barnaghi, P., Ali, M., Mileo, A., Hauswirth, M., Ganz, F., Ganea, S., Kjærgaard, B., Kuemper, D., Nechifor, S., Puiu, D., Sheth, A., Tsiatsis, V., Vestergaard, L.: Real Time IoT stream processing and large-scale data analytics for smart city applications. In: *Poster Session, European Conference on Networks and Communications* (2014)
15. Khan, M., Silva, B.N., Han, K.: A web of things-based emerging sensor network architecture for smart control systems. *Sensors* **17**(2), 332 (2017)
16. Kandula, S., et al.: The nature of data center traffic: measurements and analysis. In: *Proceedings of the 9th ACM SIGCOMM Conference Internet Measurement*, pp. 202–208 (2009)
17. Chen, M.-H., et al.: A low-latency two-tier measurement and control platform for commodity SDN. *IEEE Commun. Mag.* **54**(9), 202–208 (2016)
18. Ioannou, A., Weber, S.: A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Commun. Surv. Tutor.* **18**(4), 2847–2886 (2016)
19. Hou, R., Fang, L., Chang, Y., Yang, L., Wang, F.: Named data networking over WDM-based optical networks. *IEEE Netw.* **31**(3), 70–79 (2017)
20. Uppoor, S., Trullols-Cruces, O., Fiore, M., Barcelo-Ordinas, J.M.: Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Trans. Mob. Comput.* **13**(5), 1061–1075 (2014)