

Chapter 1

Mobile Distributed Systems: Networking and Data Management

1.1 Introduction

Convergence of wireless networking, Internet, embedded processing, and cloud computing led to a multi dimensional shift in computing paradigms. Foundations were laid through advancements in wireless networking, miniaturization, and low power embedded processing. Wireless networking itself triggered a revolution in communication landscape. It became possible to move without wire while maintaining connectivity with the network and the Internet. Miniaturization reduced the form factor of the devices. The advances in battery technology coupled with low power embedded processing capabilities made the devices light weight (such as phones, laptops, PDA), portable and powerful. With cloud computing, it was possible to implement “pay per service” model of payment as well as accessibility of personalized services any where, at any time.

Besides personal communication devices, embedded technology led to emergence of self organizing low cost Wireless Sensor and Network (WSN). Low power wireless standards such as ZigBee, 6LoWPAN, RPL, and COAP implementing IPv6 made it possible to integrate WSN with the Internet. Through WSNs, it became possible to build unattended distributed data centric networks for global information dissemination. The information gathered by WSNs could be disseminated to mediating systems for the detection of important events and to trigger appropriate responses from corresponding actuation systems. Figure 1.1 illustrates a generic diagram of a WSN based patient monitoring. In this book, our goal is to deal with two important areas namely, wireless networking protocols and mobile data management. Our aim is to focus on the foundations of enabling technologies rather than the applications. Enabling technologies provide necessary capabilities for designing and implementing innovative mobile pervasive applications. The book has been conceived and written in this context. The book consists of 16 chapters including the current one. In this chapter our aim is to examine the issues that commonly arise in building mobile distributed applications and services instead of trying to provide chapter wise summarization of the contents.

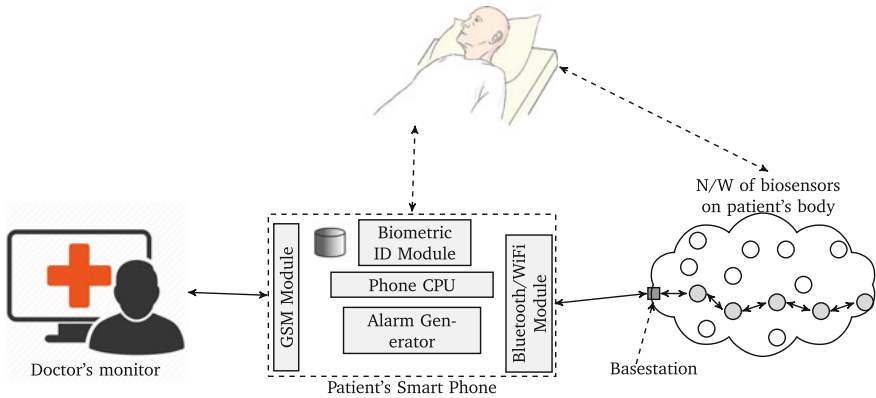


Fig. 1.1 WSN for patient monitoring

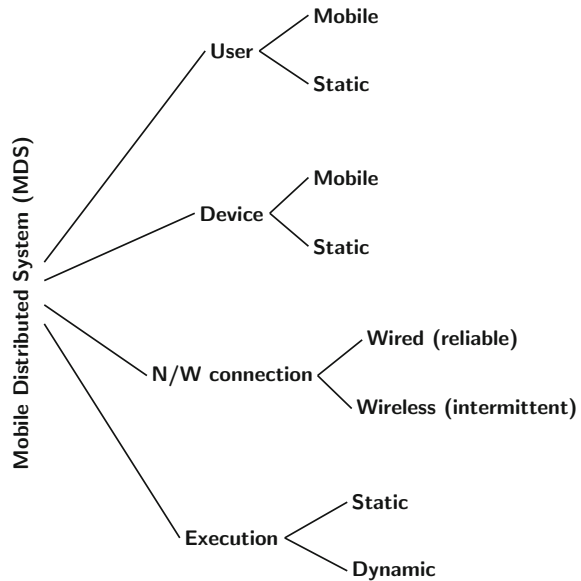
1.2 Mobile Pervasive and Ubiquitous Computing

Pervasive computing and *ubiquitous computing* are often used interchangeably in connection with the applications developed using mobile distributed systems. But there are subtle differences. Pervasiveness implies “anything whose presence is diffused”. Ubiquitous, on the other hand, means “anything which is omnipresent”. Pervasive computing tends to lean more towards mobile computing while ubiquitous computing is more close to embedded processing, intelligence and richness in interface experience. In some sense, ubiquitous computing subsumes a number of overlapping ideas from pervasive computing. It extends the idea of computers and networking being closely interwoven with our environment which do not require any human attention. Persons or inhabitants of the environment are important central theme of pervasive computing. In contrast, community and environment are dominant themes of ubiquitous computing where person centric computing is indirectly subsumed. Weiser [26] visualized the convergence between communication and computing in terms of availability of an omnipresent, and omniscient infrastructure servicing information. However, the ultimate goal of both computing paradigms is to bring comfort to human lives.

1.3 Characterizing Mobile Distributed System

A mobile system by definition consists of portable devices which run on batteries and have wireless communication interface for networking. Using mobile portable devices, the users can remain connected while moving from one place to the other. This introduces following two new elements in a distributed system: (i) dynamicity in networking, and (ii) asymmetry in the capabilities of the nodes. An architectural

Fig. 1.2 Architectural taxonomy of DS including mobility



taxonomy of Distributed Systems including mobility is shown in Fig. 1.2. To understand the implications of the mobility, first we need to study the characteristics of a Distributed System (DS), and then examine how mobility complicates the computing scenario in a DS.

In defining a distributed system, one is reminded of Leslie Lamport's celebrated definition of a distributed system [14]:

Definition 1.1 (Lamport [14]) A distributed system is the one in which your computer becomes unusable due to failure of a computer you did not know existed.

The definition embodies the concept that a distributed system presents a single coherent system though it may typically consists of a number of geographically separated autonomous but stationary computers connected by high bandwidth network links. Some of the basic characteristics of such a computing model are:

1. An application is typically partitioned into a number of smaller tasks, and the responsibility of executing these tasks is evenly shared on by a set of participating computers.
2. Every computer is assumed to have enough resources to complete a task assigned to it in the expected duration of time.
3. Communication among computers is carried over high bandwidth wired links and mostly hidden from the users.
4. The users interact with a distributed application through a single console regardless of the location of the remote computer where interaction actually takes place.
5. Scaling is inherent, and a distributed system is extensible.
6. Partial failures can be masked.

7. Computer systems can be replaced, repaired without user being aware of occurrences of such problems during the time a distributed application is being executed.
8. Network connection is assumed to be robust.
9. The computers are assumed to be connected to main power supply.

Decoupling the location of computing [9] from the method of computing, a unified framework can be defined for a distributed system which supports mobility of all kinds. Mobility can be supported at different levels of distributed computing such as: the users, the devices, the access network, the backbone network, and the data/application nodes. Basically, it means that the location of computing is conceptually defined by realizing that every bit of information and every operation has a well-defined location at the scale of geographical experience [9]. The user's location, the location of data in the hard drive or the memory, the location of computing resources, the link location, etc., are all well-defined. The important set of locations include [9]:

1. The location of the user, where the result have to finally be intimated.
2. The location of the user's computer which may be a fixed location such as on the user's office desk, or it may be co-located and changes with the location of the user.
3. The location of network used to transmit information to or from the user's computer.
4. The location, where the information gets processed according to the user's request.
5. The location, where the necessary data is stored.
6. The location, where the input data is provided to the network or the storage locations.
7. The location, where the data get interpreted, processed, compiled or otherwise prepared for storage.
8. The location, where the data are measured.
9. The location, that are representable, the special case of geographic data.

The last pair of locations, stated above, pertain to the geographic data as these locations are always defined in the sense there is no dynamicity.

The computing in a uniprocessor system is the one in which every location except the last are the same. The only possibility of changing location is offline storing of data. The offline transfer of data to the location of computing incurs considerable cost. In the case of wired distributed systems, the locations can be widely separated and the cost for going from one location to other is low due to availability of high bandwidth (wired) network links. In the case of MDS, the cost of location change may be considerable. The cost increases across all the locations stated above, specially if a mobile user changes his/her location too frequently. In a MDS, the mobility and the portability nature of the devices together with the dynamicity of the network topology affect the cost across all the top seven locations listed above.

User's mobility and terminal mobility:

It relates to the top two locations in the list. At different stages of a computation, tracking a user's location itself may become quite expensive. Terminal mobility is normally decoupled from the user's mobility. Because a user may use a PDA, or a mobile phone, or a laptop or even a static workstation at different times. In Chap. 10 of the book, we specifically address the problem of tracking terminal mobility.

Another aspect of personal and terminal mobility is the possibilities of newer and more innovative person centric applications. We discuss a few of these applications. Important generic ideas in the context of these applications are presented in Chaps. 13–16.

Wireless and wired network:

Due to user's mobility and terminal mobility, the location of network links change very frequently. Specially, when wireless links are used, the cost shipping data increases tremendously. The mobile nodes are resource poor. Therefore, the execution cost of any computation on a mobile portable device more compared to the execution cost of the same computation on a desktop or a workstation. The inherent asymmetry in computing capabilities of fixed and mobile nodes calls for relocation of computation for minimizing computation cost. The asymmetry of costs in the fixed part versus the dynamic part of the network, necessitates a paradigm shift in the design of algorithm for mobile distributed environment. Chapter 11 of the book deals with the algorithm design issues in MDS environment in extensive details.

Both cost and interoperability are important issues in MDS due to heterogeneity of wireless network standards. A sound understanding network protocols is a foundational requirement in this respect. Therefore, Chaps. 2–9 of this book focus on various protocols and standards of wireless networks.

1.4 Mobile Cloud Computing

This book is not about cloud technology. However, as stated in the previous section, the running mobile applications and services exclusively on mobile devices is not possible, as they face severe resource constraints in terms of energy, memory, and computing power. Offloading complex computations to the fixed part of the network enables the mobile devices to conserve energy significantly and operate longer. Such an approach cuts down the response time.

Cloud computing has been recognized widely as new generation of computing infrastructure that offer utility driven software services [6] under distributed settings. The integration of mobile distributed system and cloud computing, therefore, offer a “best of both” approach in building mobile application and services. The cloud service providers such as Google, Amazon, Yahoo, offer cloud services at low cost. Resources can be rapidly provisioned on-demand, and released elastically. The management functions for cloud services are provided through publicly accessible APIs.

Through these APIs, the application developers can easily integrate cloud services into their softwares.

According to MCC forum [6]:

Definition 1.2 (*MCC Fourum*) Mobile Cloud Computing at its simplest, refers to an infrastructure where both the data storage and the data processing happen outside the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and mobile computing to not just smartphone users but a much broader range of mobile subscribers.

An alternative view of MCC is: it combines mobile web with cloud computing to enable the users to access applications and services on the Internet [4, 15]. The infrastructure supported part of a Mobile Distributed System (MDS) has been viewed in terms of cloud immediately after the cloud computing technology was launched in 2007 [6].

It is possible to offer mobile services and run mobile applications on smartphones due to several supporting technologies in both hardware and software. These include, miniturization of computing devices, sensor network, phone OS, Wireless networking, and cloud infrastructure as illustrated in Fig. 1.3. Portable hand held devices (e.g., Android, Windows Mobile, iOS) are terminals where users access mobile services.



Fig. 1.3 Enabling technologies for MCC

Sensor networks are specialized low power devices for gathering contextual data that can be used for many innovative and personalized mobile services and applications. Specially, sensors are important elements for healthcare, smart home type of applications. Wireless communication is made possible through infrastructures such as GSM, WiFi, WiMAX. Taking advantage of multiple communication interfaces will need a number of supporting system softwares including mobile operating system. By integrating cloud computing with mobile distributed system through the Internet, application developers can create and deploy mobile services widely and offer mobile services anytime, anywhere and to any person. The overall architectural framework of MCC based on enabling technologies is shown in Fig. 1.4. In the general architecture of MCC, we implicitly assume that mobile devices obtain services from a remotely placed central data center. Although, functionalities of a data center are distributed over a highly parallel architecture, a mobile device has to access it through a Wide Area Network (WAN) and may at times suffer from moderate to long latency. To get around this problem, cloudlets are also deployed close to service locations. The underlying idea is to move clouds close to the mobile users by deploying cloudlets near the access points. Cloudlets can be formed dynamically on any device having resource in the LAN [24]. In the MCC architecture in Fig. 1.4, possible points where cloudlets can be placed are near AP on the LAN, near BS and close to RNC.

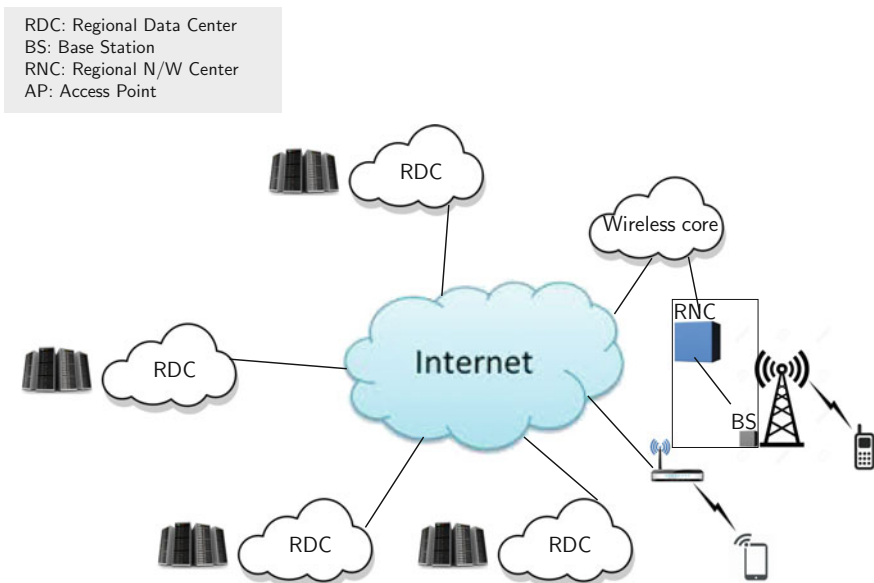


Fig. 1.4 General architecture of MCC

1.5 OS for Mobile Devices

As explained above, a critical element of MDS is Operating System (OS) support at the end devices. An OS is a set system softwares that define abstractions for optimized access of hardware capabilities of a device. In other words, OS is an interface between system architecture and the application developers or users.

Mobile OS does not include operating systems for laptops. A laptop is seen more as convenient computing device that can be carried in a bag, weighs about 2–3 kg, having a hard disk and sufficient RAM. Laptops typically run on general purpose OSes designed for desktops or workstations. However, the same may not hold strictly for a ultra thin hybrid laptop that can function both as a PC and a tablet. Though hybrid laptops are not as big as conventional laptop, the hardware design leaves a feeling of being cramped as a laptop and too bulky as tablet or PDA. These laptops have a small size keyboard with limited functionality, limited flash memory, and small display. Such a laptop switches between two OSes to provide the required functionality of the one system or the other.

A mobile communication device is equipped with multiple wireless communication interfaces such as IR, Bluetooth, WiFi, GSM-GPRS, NFC, and GPS. It runs a set of important services such as voice calls, short messaging service, camera functionality, touch screen functionality, GPS based location service. Therefore, a mobile device needs a OS that can enable it to run these services and allow the users to access the in-built communication interfaces. Advanced smartphones have many added features such as high speed CPUs, GPUs, large flash memory and multi threading capabilities which are found in conventional laptops. Therefore, mobile OS has evolved into a complex multitasking OS, and merits a closer look in the context of development of mobile applications.

Chapter 8 of this book introduces mobile OS as an abstraction layer for System on Chip (SoC) and explains rationale behind a microkernel based approach to design of mobile OS. Further, it discusses the difficulties involved in adopting desktop OSes to the mobile systems. After presenting the features of a mobile OS, the kernel features are discussed. As case studies, four mobile OSes, viz., J2ME, Symbian, Android and iOS have been included. Finally, a comparison of Android and iOS environments has been made.

1.6 Mobile Applications

To provide a platform relevant to the contents of book, let us now introduce some basic type services one expects from a mobile distributed system. Figure 1.5 illustrates a few of these application. We investigate two of these applications a little deeper in order to provide the reader with an idea about the potentials that exist. In fact, a number of venture capitalists are ready to commit large amount of investments

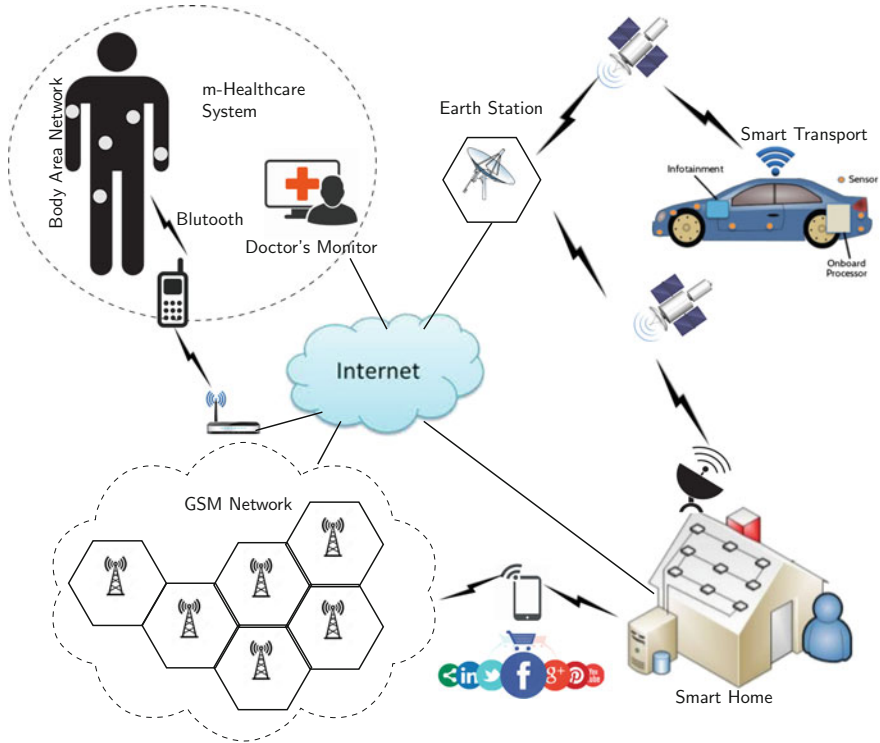


Fig. 1.5 Mobile pervasive computing

even in start up companies that come up with business plans to provide personalized services over mobile distributed systems.

1.6.1 mHealthcare

One of the very important mobile applications is round the clock health care and monitoring system for the patients with chronic diseases, and the senior citizens. Darrel West [28] cites Cisco Visual Networking Index [12] to predict that global mobile traffic may to increase by 18-folds between 2011 and 2016, and about 10 billion mobile devices could be in use world wide. According to WHO [18], the use of mobile health care is expected increase substantially in three areas:

1. Assisting the management of chronic diseases,
2. Assisting the elderly people staying alone, and
3. Assisting the pregnant mothers staying alone.

The use of mobile health apps are predominantly (over 35% of time) for health queries processed by call centers, mhealth apps are also used for other tasks such as fixing appointments with physicians by SMS, treatment reminders, telemedicine, accessing patient records, monitoring patients, physician decision support.

From technological perspectives, mHealth support is realizable for sophisticated care and monitoring [3, 10, 23]. The vision of mobile healthcare presented in [23] covers both applications and requirements of such a system. It encompasses three aspects of monitoring: (i) short term (home monitoring), (ii) long term (hospitals) and (iii) personalized monitoring. In order to provide end to end solution, the healthcare support must involve detection and management of incidences, emergency intervention, hospital transfers and treatments. Wireless network based solutions use WLAN, ad hoc wireless network, infrastructure-based wireless networks such as GSM/3G/4G/5G. The general framework of the system appears in Fig. 1.1. Varshney [23] identified that context awareness and reliability are specific challenges which have not been properly addressed in a comprehensive framework of wireless health monitoring system. The major part of context awareness can be derived from patient's medical history. But there could be also variations due to current state of the patient, the availability of specific experts and the medicines, and so on.

1.6.2 Logistic and Transport Management

Logistics and transport management form one of the important area of automation due to the inherent complexity of involved operations. Big logistics firms such as DHL, having operations over 200 countries around the globe, offer integrated logistic solution for entire supply chain from the manufacturers/suppliers to the retailers to the consumers. These companies use RFID, WSN and internet technologies to automate their supply chain management operations.

The last mile connectivity for logistic supply chain management is provided by transport vehicles. Intel offers a solution blueprint for an intelligent transport system with IoT [16]. It outlines monitoring of vehicle dynamics, intelligent navigation, fleet management along with a series of value added services. According to this solution, each vehicle is to be fitted with multiple sensing units, and an on-board terminal. The terminal will consist of data storage, GPS module, vehicle condition collection module, real-time clock, wireless communication module and a data communication interface.

Once vehicles are made to speak for themselves, linking these with customer data center applications is technologically straightforward through cloud infrastructure. An overall architecture as illustrated in Fig. 1.6 has been proposed in [16].

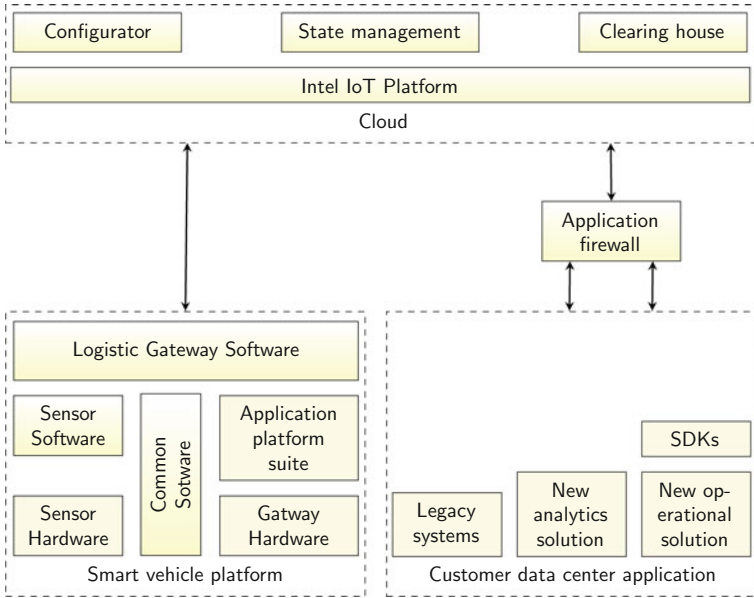


Fig. 1.6 System architecture for smart freight management [16]

1.7 Smart Environments

Smart environment is a term sometime ill-used and sometimes abused. However, in absence of proper terminology, it has been used to describe an environment where inhabitants are offered sophisticated technology based services. Weiser et al. [27] defined ubiquitous computing as:

Definition 1.3 (*Ubiquitous computing* [27]) A physical world that is richly and invisibly interwoven with sensors, actuators, displays and computational elements embedded seamlessly in everyday objects of our lives and connected through a continuous network.

We feel that the above definition succinctly captures the generic nature of any smart environment. The definition of smart environment given subsequently by Youngblood et al. [29] tries to amplify the comfort of the inhabitants over the supporting technologies describing ubiquitous computing framework.

Definition 1.4 (*Smart environment* [29]) A smart environment defined as the one that is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment.

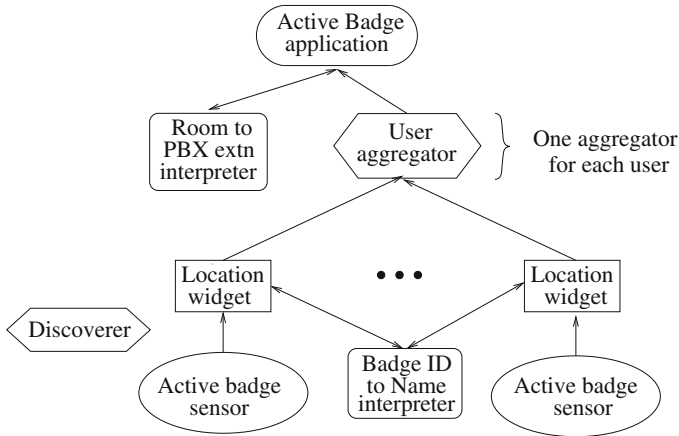


Fig. 1.7 Context aware call forwarding application [5]

Smart environment is alternatively referred to as ambient intelligence. Augusto and McCullagh [1] defined ambient intelligence as follows:

Definition 1.5 (*Ambient intelligence* [1]) A digital environment that proactively, but sensibly, supports people in their daily lives.

Definitions 1.4 and 1.5 imply that the smartness of environment is dependent on the abilities to provide added comfort to the inhabitants rather than the degree of technological sophistication. So, probably it is more appropriate to the use term *smart environment* for emphasizing physical infrastructure of sensors, actuators and network [17].

The requirements of each inhabitant vary from the other and also vary from one environment to the other. However, there is a set of common attributes which improve experience of inhabitants in a smart environment. More specifically, a smart environment

1. Optimizes productivity of inhabitants,
2. Improves ease of working,
3. Minimizes operational cost, and
4. Ensures security of inhabitants.

To this end, the environment must acquire knowledge of events and apply the same to respond to the occurrence of an event. For execution of a response, the environment should actuate actions from devices and objects in the environment which largely automate task possibly requiring minimal unobtrusive assistance from human inhabitants. It should adapt to the changes in the environment and its inhabitants. Above all privacy and security of inhabitants should be preserved. The set of generic requirements appear like a wishlist unless requirements these are seen in the context of specific environment. So, to motivate the reader we first introduce the concept of context aware computing and then follow it up with a concrete description of a smart environment.

1.7.1 Context Aware Computing

A typical application using context for its services is provided by Xerox PARC's active badge system [25, 26]. The active badge system was developed for personal emergency related communication with the wearer who is engaged in work in high security installation. Normally, considering security risks, the phone numbers are not revealed for contacting persons working in high security installations. So, establishing contact with such persons is a problem. However, by use of context awareness, the problem can be solved neatly. The entire system is explained in Fig. 1.7. The system consists of a number of infrared based sensor distributed over the installation. These sensors can detect Active Badges worn by people working inside the installation. Sensor detecting a badge then links the location of the wearer from a location widget corresponding to it. The collected information about the tag ID and its location are sent to the user's aggregator. There is a user aggregator for each user. The aggregator collects location information from each location widget for their corresponding users and shares the same to the discoverer. Every component of the application must register with the discoverer to be able to locate the users. The application then uses the discoverer to locate the user aggregator and the room to phone extension interpreter. After getting the related information, the application forwards the telephone calls. So the users can talk to each other, and in the case of emergency a person can be located and the call can be forwarded.

Dey and Abowd [5] have reported about Context Toolkit which they developed as a distributed infrastructure which provides abstractions such as widgets, interpreter, aggregators, services and discovers for building the context aware applications. The distributed infrastructure supports reliable cross platform communication between distributed objects. Simple object communication mechanisms HTTP and XML are used for encoding messages. This means TCP/IP is one of the requirement for the communication to happen. We will discuss more details about context abstractions in Chap. 16.

1.7.2 Driverless Cars

It is now common place to see people using navigation system to find routes to reach unknown addresses. The navigational applications mainly uses GPS and satellite route maps to guide people for this purpose. These software may provide some additional assistance information for safe driving. These include speed limits for the various highways and roads, weather conditions, terrain features, etc. The primary purpose is still limited to finding route from a source to a destination. However, building driverless car requires complex interplay of GPS navigation system, car cruise control and other vehicles on the road. It represents a sufficiently *smart environment* for a small number of inhabitants, i.e., the passengers of the car.

Though a number of technological challenges still remains unresolved, Google has successfully executed test runs of driverless cars on local streets [13]. It is expected by 2020, high way driving can be completely handed over to automated cruise control. Major auto makers such as Mercedes-Benz, Nissan, Volvo, BMW and Audi have already built their test models [13].

The successful realization of a driverless car is heavily dependent on integration of diverse technologies. First of all, it needs a very advanced hardware as far as car control is concerned. Two different directions of cruise controls, namely, longitudinal (break and speed), and lateral (steering and trajectory) are needed. The reaction time depends on speed of car and the colliding objects. The anticipation of colliding object's trajectory, speed assessment, etc., are important inputs to estimate how the car should apply optimum control to avoid collision or damage to self. The objects may be of different types such as a random pedestrians, or a cyclist or a speeding car or even animals. It requires a 360° view of surroundings. Inputs from multiple sensing units such as laser sensors, cameras, and radars will be needed for the purpose. Exact deployments of different sensing units needs careful simulations with normal traffic conditions on road system. The problem is to find accurate placement of sensing units so that the data collected by sensors should be able to provide enough real time data as the car is being driven. Some sensors may have to be roof mounted, others placed on the front or rear bumpers, many other underneath the car and the on different side fenders, etc. Google mapped entire California road system (172,000 miles) for a simulation purpose. It indicated that the sensors would be sending high volume and high velocity data (about 1 GB per second) [13]. It will require sophisticated big data analytics to process data and provide inputs to the autonomous driver.

Vehicle to vehicle (V2V) communication is conceptualized as peer to peer ad hoc network which can alleviate the problems by sharing information about traffic situation among vehicles in a sections of roads. V2V communication architecture consists of vehicles mounted with on board units/application units (OBU/AU), smart signals, road side units (RSU), smart street signs, and other artefacts on the street with wireless interfaces. A reference architecture is shown in Fig. 1.8. In a V2V network four communication types are possible:

1. In-vehicle communication
2. Vehicle to vehicle communication
3. Vehicle to broadband communication
4. Vehicle to RSU communication.

In-vehicle communication is for the domain inside vehicle. It typically raises alert messages related problems that is specific to a vehicle. For example, hitting low fuel, air pressure variation on wheels, driver is feeling drowsy and, so on. Vehicle to vehicle communication is used for inter vehicles domain. It may send out information related traffic situation, constructions on roads, diversion to be taken, or some vehicle needs help, and so on. Vehicle to broadband communication is related cloud processing requirements as well as infotainments. Vehicle to RSU communication enables vehicle to learn real-time updates and re-routing plans, etc.

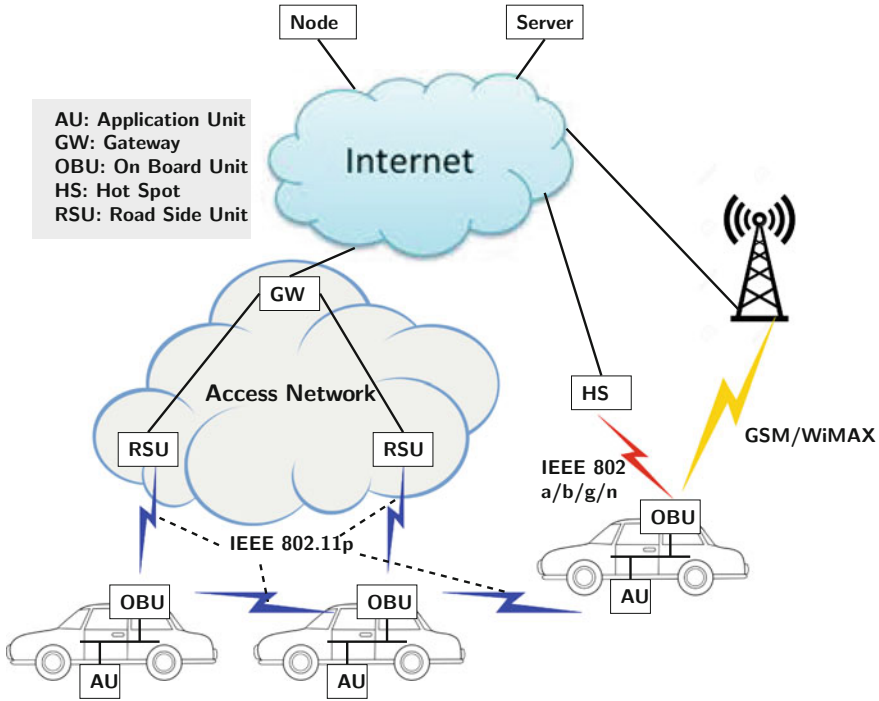


Fig. 1.8 A reference architecture for VANET [2]

The protocol architecture of V2V communication should be able to distinguish among three types of wireless communication technologies: IEEE 802.11p, conventional WLAN IEEE 802.11a/b/g/n and GPRS/UMTS. On the top these physical layer specific network layer technologies should coordinate with the transport layers. The basic reference architecture for protocols will be as shown in Fig. 1.9. However, in implementation some MAC layer could be merged into one single layer. Typically, the wireless standards for V2V network specifies a range up to 300 m. So about 5–10 hops network could disseminate information of about 1.5 km from a vehicle. Another supporting technology is cloud-based information processing. The cloud communication could be through 4G/5G with vehicle terminals.

In summary some of the research themes centered around driverless car connected to the contents of this book are:

- Modeling of wireless networks and new protocols with QoS support.
- Vehicle to vehicle ad hoc networks.
- Localization and location based services.
- Multi sensor data fusion and processing.
- Context aware computing.
- Cloud processing support with new data analytics.

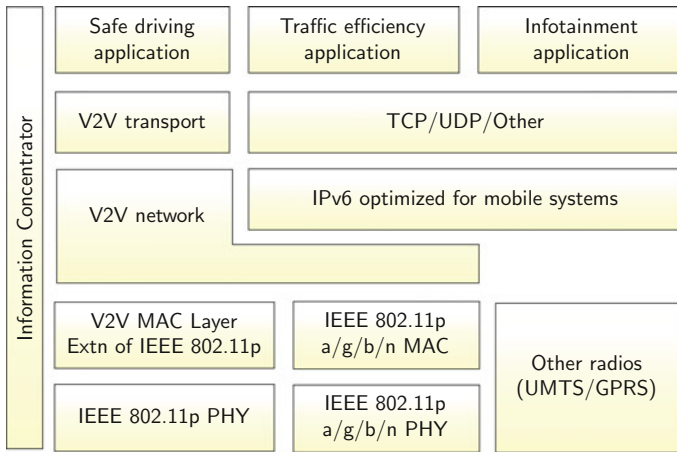


Fig. 1.9 A reference protocol architecture for VANET [2]

1.8 Organization of Book

The main purpose of the material covered in this book is to provide the reader a deeper understanding of wireless communication and data management.

The first part of the book consists of seven chapters (Chaps. 2–8), deal with different wireless networks standards, protocols, routings and applications. There are many excellent text books [8, 19, 21, 22] which cover physical layers of wireless communication system that includes mathematical theories behind characteristics and propagation of radio waves. In this book our focus is in the realm of standards, protocols, routings and applications.

A number of books addressing many aspects of mobile data management have been published in the past [7, 11, 20]. However, there are not many well known books where mobile data management appears as a central theme of the text in an organized manner. The second part of the book, consisting of eight chapters (Chaps. 9–16) dealing with mobile data management, is an attempt in this direction. We created an organization with a view that senior undergraduate students and research scholars may find it helpful to develop interests in the area.

There was a slight dilemma in deciding whether the contents of Chap. 9 can be strictly classified as under mobile data management issues. Apart from sensor fusion, it deals with two network related issues in WSNs namely, routing and integration with IP. We justify our decision to classify Chap. 9 under mobile data management on the basis of following facts:

- Sensor nodes unlike IP based nodes are data centric, and WSN routing is data centric.
- Many existing deployments of WSNs interoperate with IP network through specialized sensor gateways which implement middlewares for IP integration.

- Sensor fusion techniques included in this chapter strictly represent data management issues.

References

1. J.C. Augusto, P. McCullagh, Ambient intelligence: concepts and applications. *Int. J. Comput. Sci. Inf. Syst.* **4**(1), 1–28 (2007)
2. R. Baldessari, The author team. C2C and CC system (2007). http://elib.dlr.de/48380/1/C2C-CC_manifesto_v1.1.pdf
3. A. Boukerche, Y. Ren, A secure mobile healthcare system using trust-based multicast scheme. *IEEE J. Sel. Areas Commun.* **27**(4), 387–399 (2009)
4. J.H. Christensen, Using RESTful Web-Services and Cloud Computing to Create Next Generation Mobile Applications, *The 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications (OOPSLA)*, Oct 2009, pp. 627–634
5. A.K. Dey, G.D. Abowd, A conceptual framework and a toolkit for supporting rapid prototyping of context-aware applications. *Human-Comput. Inter. (HCI) J.* **16**(2-4):97–166 (2001)
6. H.T. Dinh, C. Lee, D. Niyato, P.A. Wang, Survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Commun. Mobile Comput.* **13**(18), 1587–1611 (2013)
7. B. Reza (ed.), *Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML* (Cambridge University Press, New York, USA, 2004)
8. V.K. Garg, *Wireless Networks Evolution: 2G to 3G* (Morgan Kaufman, 2010)
9. M.F. Goodchild, D.M. Johnston, D.J. Maguire, V.T. Noronha, Distributed and Mobile Computing, *A Research Agenda for Geographic Information, Science* (2004), pp. 257–286
10. A.V. Halteren, R. Bults, K. Wack, D. Kontantas, I. Widya, N. Dokovsky, G. Koprinkov, V. Jone, R. Herzog, Mobile patient monitoring: the mobile health system. *J. Inf. Technol. Healthc.* **2**(5), 365–373 (2004)
11. T. Imielinski, H.F. Korth (eds.), *Mobile Computing* (Kluwer Academic Publishers, Norwell, MA, USA, 1996)
12. Cisco Visual Networking Index, Global Mobile Data Traffic Forecast Update, 2011–2016, February 14 2012
13. T. Jiang, S. Petrovic, U. Ayyer, S. Husain, Self-driving cars: disruptive or incremental. *Appl. Innov. Rev.* **1**, 3–22 (2015)
14. L. Lamport, Distribution. Technical report, Microsoft, 1987
15. L. Liu, R. Moulis, D. Shea, Cloud service portal for mobile device management, *The 7th IEEE International Conference on E-Business Engineering (ICEBE)*, Nov 2010, pp. 474–478
16. G. Moakley, Smart freight technology powered by internet of things. <http://www.intel.com/content/www/us/en/internet-of-things/solution-briefs/smart-freight-technology-brief.html>, Jan 2014
17. H. Nakashima, H. Aghajan, J.C. Augusto, *Handbook of Ambient Intelligence and Smart Environments* (Springer, 2009)
18. World Health Organization, mhealth: new horizons for health through mobile technologies. Second Global Survey on eHealth 3 (2011)
19. K. Pahlavan, P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*, 1st edn. (Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001)
20. E. Pitoura, G. Samaras, *Data Management in Mobile Computing* (Springer, 2012)
21. T.S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd edn. (Prentice Hall PTR, 2002)
22. J.H. Schiller, *Mobile Communications* (Addison-Wesley, 2003)
23. U. Varshney, Pervasive healthcare and wireless health monitoring. *Mobile Netw. Appl.* **12**, 113–127 (2007)

24. T. Verbelen, P. Simoens, F.D. Turck, B. Dhoedt, The Third ACM Workshop on Mobile Cloud Computing and Services (MCS '12), *Cloudlets: Bringing the Cloud to the Mobile User* (ACM, New York, NY, USA, 2012), pp. 29–36
25. R. Want, A. Hopper, V. Falcao, J. Gibbons, The active badge location system. *ACM Trans. Inf. Syst.* **10**(1), 91–102 (1992)
26. M. Weiser, Some computer science issues in ubiquitous computing. *Commun. ACM* **36**(7), 75–84 (1993)
27. M. Weiser, R. Gold, J.S. Brown, The origins of ubiquitous computing research at PARC in the late 1980s. *IBM Syst. J.* **38**(4), 693–696 (1999)
28. D. West, How mobile devices are transforming healthcare. *Issues Technol. Innov.* **18**(1), 1–11 (2012). May
29. G.M. Youngblood, D.J. Cook, L.B. Holder, E.O. Heierman, Automation intelligence for the smart environment, *The International Joint Conference on Artificial Intelligence*, pp. 1513–1514 (2005)