Hindawi
Journal of Electrical and Computer Engineering
Volume 2017, Article ID 9324035, 25 pages
https://doi.org/10.1155/2017/9324035

*Review Article*

# Internet of Things: Architectures, Protocols, and Applications

**Pallavi Sethi and Smruti R. Sarangi**

*Department of Computer Science, IIT Delhi, New Delhi, India*

Correspondence should be addressed to Smruti R. Sarangi; srsarangi@cse.iitd.ac.in

Received 12 August 2016; Accepted 18 December 2016; Published 26 January 2017

Academic Editor: Rajesh Khanna

Copyright © 2017 Pallavi Sethi and Smruti R. Sarangi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Things (IoT) is defined as a paradigm in which objects equipped with sensors, actuators, and processors communicate with each other to serve a meaningful purpose. In this paper, we survey state-of-the-art methods, protocols, and applications in this new emerging area. This survey paper proposes a novel taxonomy for IoT technologies, highlights some of the most important technologies, and profiles some applications that have the potential to make a striking difference in human life, especially for the differently abled and the elderly. As compared to similar survey papers in the area, this paper is far more comprehensive in its coverage and exhaustively covers most major technologies spanning from sensors to applications.

## 1. Introduction

Today the Internet has become ubiquitous, has touched almost every corner of the globe, and is affecting human life in unimaginable ways. However, the journey is far from over. We are now entering an era of even more pervasive connectivity where a very wide variety of appliances will be connected to the web. We are entering an era of the "Internet of Things" (abbreviated as IoT). This term has been defined by different authors in many different ways. Let us look at two of the most popular definitions. Vermesan et al. [1] define the Internet of Things as simply an interaction between the physical and digital worlds. The digital world interacts with the physical world using a plethora of sensors and actuators. Another definition by Peña-López et al. [2] defines the Internet of Things as a paradigm in which computing and networking capabilities are embedded in any kind of conceivable object. We use these capabilities to query the state of the object and to change its state if possible. In common parlance, the Internet of Things refers to a new kind of world where almost all the devices and appliances that we use are connected to a network. We can use them collaboratively to achieve complex tasks that require a high degree of intelligence.

For this intelligence and interconnection, IoT devices are equipped with embedded sensors, actuators, processors, and transceivers. IoT is not a single technology; rather it is an agglomeration of various technologies that work together in tandem.

Sensors and actuators are devices, which help in interacting with the physical environment. The data collected by the sensors has to be stored and processed intelligently in order to derive useful inferences from it. Note that we broadly define the term *sensor*; a mobile phone or even a microwave oven can count as a sensor as long as it provides inputs about its current state (internal state + environment). An *actuator* is a device that is used to effect a change in the environment such as the temperature controller of an air conditioner.

The storage and processing of data can be done on the edge of the network itself or in a remote server. If any preprocessing of data is possible, then it is typically done at either the sensor or some other proximate device. The processed data is then typically sent to a remote server. The storage and processing capabilities of an IoT object are also restricted by the resources available, which are often very constrained due to limitations of size, energy, power, and computational capability. As a result the main research challenge is to ensure that we get the right kind of data at the desired level of accuracy. Along with the challenges of data collection, and handling, there are challenges in communication as well. The communication between IoT devices is mainly wireless because they are generally installed at geographically dispersed locations. The wireless channels often have high

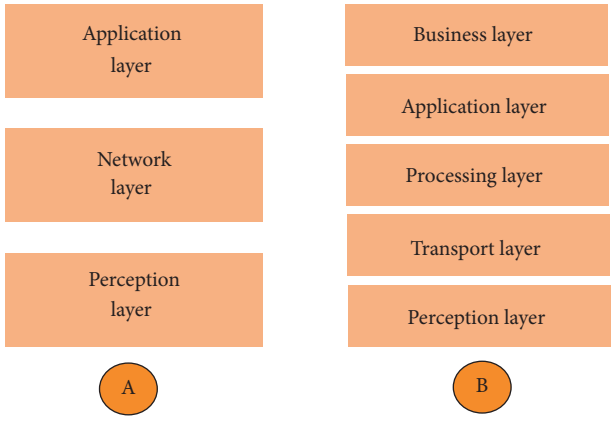



Figure 1: Architecture of IoT (A: three layers) (B: five layers).

rates of distortion and are unreliable. In this scenario reliably communicating data without too many retransmissions is an important problem and thus communication technologies are integral to the study of IoT devices.

Now, after processing the received data, some action needs to be taken on the basis of the derived inferences. The nature of actions can be diverse. We can directly modify the physical world through actuators. Or we may do something virtually. For example, we can send some information to other smart things.

The process of effecting a change in the physical world is often dependent on its state at that point of time. This is called *context awareness*. Each action is taken keeping in consideration the context because an application can behave differently in different contexts. For example, a person may not like messages from his office to interrupt him when he is on vacation.

Sensors, actuators, compute servers, and the communication network form the core infrastructure of an IoT framework. However, there are many software aspects that need to be considered. First, we need a middleware that can be used to connect and manage all of these heterogeneous components. We need a lot of standardization to connect many different devices. We shall discuss methods to exchange information and prevailing standards in Section 7.

The Internet of Things finds various applications in health care, fitness, education, entertainment, social life, energy conservation, environment monitoring, home automation, and transport systems. We shall focus on these application areas in Section 9. We shall find that, in all these application areas, IoT technologies have significantly been able to reduce human effort and improve the quality of life.

## 2. Architecture of IoT

There is no single consensus on architecture for IoT, which is agreed universally. Different architectures have been proposed by different researchers.

*2.1. Three- and Five-Layer Architectures.* The most basic architecture is a three-layer architecture [3–5] as shown in Figure 1. It was introduced in the early stages of research in this area. It has three layers, namely, the perception, network, and application layers.

(i) The *perception layer* is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.

(ii) The *network layer* is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.

(iii) The *application layer* is responsible for delivering application specific services to the user. It defines various applications in which the Internet of Things can be deployed, for example, smart homes, smart cities, and smart health.

The three-layer architecture defines the main idea of the Internet of Things, but it is not sufficient for research on IoT because research often focuses on finer aspects of the Internet of Things. That is why, we have many more layered architectures proposed in the literature. One is the five-layer architecture, which additionally includes the processing and business layers [3–6]. The five layers are perception, transport, processing, application, and business layers (see Figure 1). The role of the perception and application layers is the same as the architecture with three layers. We outline the function of the remaining three layers.

(i) The *transport layer* transfers the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 3G, LAN, Bluetooth, RFID, and NFC.

(ii) The *processing layer* is also known as the middleware layer. It stores, analyzes, and processes huge amounts of data that comes from the transport layer. It can manage and provide a diverse set of services to the lower layers. It employs many technologies such as databases, cloud computing, and big data processing modules.

(iii) The *business layer* manages the whole IoT system, including applications, business and profit models, and users' privacy. The business layer is out of the scope of this paper. Hence, we do not discuss it further.

Another architecture proposed by Ning and Wang [7] is inspired by the layers of processing in the human brain. It is inspired by the intelligence and ability of human beings to think, feel, remember, make decisions, and react to the physical environment. It is constituted of three parts. First is the human brain, which is analogous to the processing and data management unit or the data center. Second is the spinal cord, which is analogous to the distributed network of data processing nodes and smart gateways. Third is the network of nerves, which corresponds to the networking components and sensors.

*2.2. Cloud and Fog Based Architectures.* Let us now discuss two kinds of systems architectures: cloud and fog computing (see the reference architectures in [8]). Note that this classification is different from the classification in Section 2.1, which was done on the basis of protocols.

In particular, we have been slightly vague about the nature of data generated by IoT devices, and the nature of data processing. In some system architectures the data processing is done in a large centralized fashion by cloud computers. Such a cloud centric architecture keeps the cloud at the center, applications above it, and the network of smart things below it [9]. Cloud computing is given primacy because it provides great flexibility and scalability. It offers services such as the core infrastructure, platform, software, and storage. Developers can provide their storage tools, software tools, data mining, and machine learning tools, and visualization tools through the cloud.

Lately, there is a move towards another system architecture, namely, *fog computing* [10–12], where the sensors and network gateways do a part of the data processing and analytics. A fog architecture [13] presents a layered approach as shown in Figure 2, which inserts monitoring, preprocessing, storage, and security layers between the physical and transport layers. The monitoring layer monitors power, resources, responses, and services. The preprocessing layer performs filtering, processing, and analytics of sensor data. The temporary storage layer provides storage functionalities such as data replication, distribution, and storage. Finally, the security layer performs encryption/decryption and ensures data integrity and privacy. Monitoring and preprocessing are done on the edge of the network before sending data to the cloud.

Often the terms "fog computing" and "edge computing" are used interchangeably. The latter term predates the former and is construed to be more generic. *Fog computing* originally termed by Cisco refers to smart gateways and smart sensors, whereas *edge computing* is slightly more penetrative in nature. This paradigm envisions adding *smart data preprocessing capabilities* to physical devices such as motors, pumps, or lights. The aim is to do as much of preprocessing of data as possible in these devices, which are termed to be at the *edge* of the network. In terms of the system architecture, the architectural diagram is not appreciably different from Figure 2. As a result, we do not describe edge computing separately.

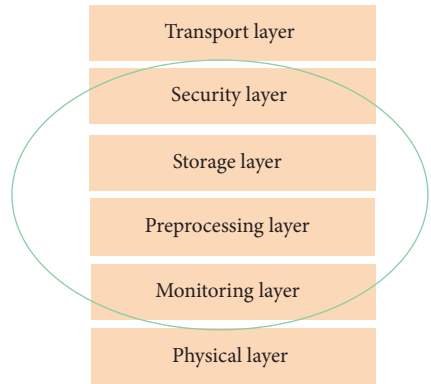



Figure 2: Fog architecture of a smart IoT gateway.

Finally, the distinction between protocol architectures and system architectures is not very crisp. Often the protocols and the system are codesigned. We shall use the generic 5-layer IoT protocol stack (architectural diagram presented in Figure 2) for both the fog and cloud architectures.

*2.3. Social IoT.* Let us now discuss a new paradigm: social IoT (SIoT). Here, we consider social relationships between objects the same way as humans form social relationships (see [14]). Here are the three main facets of an SIoT system:

(i) The SIoT is navigable. We can start with one device and navigate through all the devices that are connected to it. It is easy to discover new devices and services using such a social network of IoT devices.

(ii) A need of trustworthiness (strength of the relationship) is present between devices (similar to friends on Facebook).

(iii) We can use models similar to studying human social networks to also study the social networks of IoT devices.

*2.3.1. Basic Components.* In a typical social IoT setting, we treat the devices and services as bots where they can set up relationships between them and modify them over time. This will allow us to seamlessly let the devices cooperate among each other and achieve a complex task.

To make such a model work, we need to have many interoperating components. Let us look at some of the major components in such a system.

(1) ID: we need a unique method of object identification. An ID can be assigned to an object based on traditional parameters such as the MAC ID, IPv6 ID, a universal product code, or some other custom method.

(2) Metainformation: along with an ID, we need some metainformation about the device that describes its form and operation. This is required to establish appropriate relationships with the device and also appropriately place it in the universe of IoT devices.

(3) Security controls: this is similar to "friend list" settings on Facebook. An owner of a device might place restrictions on the kinds of devices that can connect to it. These are typically referred to as *owner controls*.

(4) Service discovery: such kind of a system is like a service cloud, where we need to have dedicated directories that store details of devices providing certain kinds of services. It becomes very important to keep these directories up to date such that devices can learn about other devices.

(5) Relationship management: this module manages relationships with other devices. It also stores the types of devices that a given device should try to connect with based on the type of services provided. For example, it makes sense for a light controller to make a relationship with a light sensor.

(6) Service composition: this module takes the social IoT model to a new level. The ultimate goal of having such a system is to provide better integrated services to users. For example, if a person has a power sensor with her air conditioner and this device establishes a relationship with an analytics engine, then it is possible for the ensemble to yield a lot of data about the usage patterns of the air conditioner. If the social model is more expansive, and there are many more devices, then it is possible to compare the data with the usage patterns of other users and come up with even more meaningful data. For example, users can be told that they are the largest energy consumers in their community or among their Facebook friends.

*2.3.2. Representative Architecture.* Most architectures proposed for the SIoT have a server side architecture as well. The server connects to all the interconnected components, aggregates (composes) the services, and acts as a single point of service for users.

The server side architecture typically has three layers. The first is the *base* layer that contains a database that stores details of all the devices, their attributes, metainformation, and their relationships. The second layer (*Component* layer) contains code to interact with the devices, query their status, and use a subset of them to effect a service. The topmost layer is the *application* layer, which provides services to the users.

On the device (object) side, we broadly have two layers. The first is the *object* layer, which allows a device to connect to other devices, talk to them (via standardized protocols), and exchange information. The *object* layer passes information to the *social* layer. The social layer manages the execution of users' applications, executes queries, and interacts with the application layer on the server.

## 3. Taxonomy

Let us now propose taxonomy for research in IoT technologies (see Figure 3). Our taxonomy is based on the architectural elements of IoT as presented in Section 2.

The first architectural component of IoT is the perception layer. It collects data using sensors, which are the most important drivers of the Internet of Things [15]. There are various types of sensors used in diverse IoT applications. The most generic sensor available today is the smartphone. The smartphone itself has many types of sensors embedded in it [16] such as the location sensor (GPS), movement sensors (accelerometer, gyroscope), camera, light sensor, microphone, proximity sensor, and magnetometer. These are being heavily used in different IoT applications. Many other types of sensors are beginning to be used such as sensors for measuring temperature, pressure, humidity, medical parameters of the body, chemical and biochemical substances, and neural signals. A class of sensors that stand out is infrared sensors that predate smartphones. They are now being used widely in many IoT applications: IR cameras, motion detectors, measuring the distance to nearby objects, presence of smoke and gases, and as moisture sensors. We shall discuss the different types of sensors used in IoT applications in Section 5.

Subsequently, we shall discuss related work in data preprocessing. Such applications (also known as fog computing applications) mainly filter and summarize data before sending it on the network. Such units typically have a little amount of temporary storage, a small processing unit, and some security features.

The next architectural component that we shall discuss is communication. We shall discuss related work (in Section 7) on different communication technologies used for the Internet of Things. Different entities communicate over the network [17–19] using a diverse set of protocols and standards. The most common communication technologies for short range low power communication protocols are RFID (Radio Frequency Identification) and NFC (Near Field Communication). For the medium range, they are Bluetooth, Zigbee, and WiFi. Communication in the IoT world requires special networking protocols and mechanisms. Therefore, new mechanisms and protocols have been proposed and implemented for each layer of the networking stack, according to the requirements imposed by IoT devices.

We shall subsequently look at two kinds of software components: middleware and applications. The middleware creates an abstraction for the programmer such that the details of the hardware can be hidden. This enhances interoperability of smart things and makes it easy to offer different kinds of services [20]. There are many commercial and open source offerings for providing middleware services to IoT devices. Some examples are OpenIoT [21], MiddleWhere [22], Hydra [23], FiWare [24], and Oracle Fusion Middleware. Finally, we discuss the applications of IoT in Section 9. We primarily focus on home automation, ambient assisted living, health and fitness, smart vehicular systems, smart cities, smart environments, smart grids, social life, and entertainment.

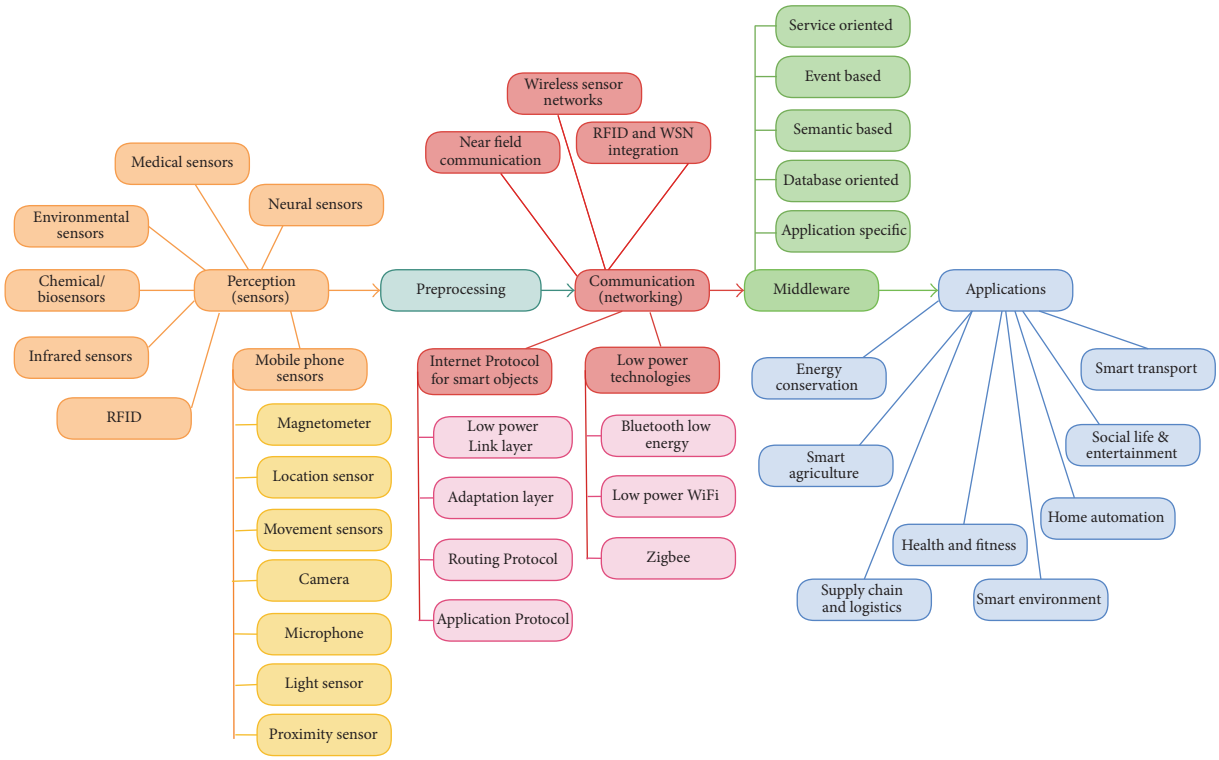



Figure 3: Taxonomy of research in IoT technologies.

## 4. Related Survey Papers

Our taxonomy describes the technologies in the IoT domain and is classified on the basis of architectural layers. We have tried to cover all subareas and recent technologies in our taxonomy. There have been many survey papers on the Internet of Things in the past. Table 1 shows how our survey is different from other highly cited surveys in the literature.

Let us first consider our novel contributions. Our paper looks at each and every layer in the IoT stack, and as a result the presentation is also far more balanced. A novel addition in our survey is that we have discussed different IoT architectures. This has not been discussed in prior surveys on the Internet of Things. The architecture section also considers newer paradigms such as fog computing, which have also hitherto not been considered. Moreover, our survey nicely categorizes technologies based on the architectural layer that they belong to. We have also thoroughly categorized the network layer and tried to consolidate almost all the technologies that are used in IoT systems. Such kind of a thorough categorization and presentation of technologies is novel to the best of our knowledge.

Along with these novel contributions our survey is far more comprehensive, detailed, and exhaustive as compared to other surveys in the area. Most of the other surveys look at only one or two types of sensors, whereas we describe 9 types of sensors with many examples. Other surveys are also fairly restricted when they discuss communication technologies and applications. We have discussed many types of middleware technologies as well. Prior works have not given middleware technologies this level of attention. We cover 10 communication technologies in detail and consider a large variety of applications encompassing smart homes, health care, logistics, transport, agriculture, environment, smart cities, and green energy. No other survey in this area profiles so many technologies, applications, and use cases.

## 5. Sensors and Actuators

All IoT applications need to have one or more sensors to collect data from the environment. Sensors are essential components of smart objects. One of the most important aspects of the Internet of Things is *context awareness*, which is not possible without sensor technology. IoT sensors are mostly small in size, have low cost, and consume less power. They are constrained by factors such as battery capacity and ease of deployment. Schmidt and Van Laerhoven [25] provide an overview of various types of sensors used for building smart applications.

*5.1. Mobile Phone Based Sensors.* First of all, let us look at the mobile phone, which is ubiquitous and has many types of sensors embedded in it. In specific, the smartphone is a very handy and user friendly device that has a host of built in communication and data processing features. With the increasing popularity of smartphones among people, researchers are showing interest in building smart IoT solutions using smartphones because of the embedded sensors [16, 26]. Some additional sensors can also be used depending

Table 1: Comparison with other surveys on the basis of topics covered.

| Survey paper | Sensors | Fog computing | Middleware | Communication | Applications | Other |
|---|---|---|---|---|---|---|
| "Internet of Things: A Survey," Atzori et al., 2010 | RFID | Not covered | Service oriented architecture | Communication standards, IEEE 802.15.4, WSN, Zigbee, 6LoWPAN, NFC, Wireless Hart, M2M, EPC global, ROLL routing | Smart home, health, logistics, transport, agriculture, social, environment | Issues related to security, privacy, naming, addressing |
| "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," Gubbi et al., 2013 | RFID | Not covered | Service oriented architecture | WSN, addressing schemes | Personal and home, enterprise, utilities, mobile | Cloud centric IoT |
| "The Internet of Things—A Survey of Topics and Trends," Whitmore et al., 2014 | RFID | Not covered | Semantic middleware | WSN, NFC, WSN | Smart infrastructure, health care, supply chains/logistics | Security and plrivacy |
| Our survey | Covered various types of sensors: environmental, medical, neural, chemical, infrared, mobile phone sensors, RFID | Fog computing/smart gateway layered architecture of IoT | Issues addressed by middleware, types of middleware: event based, service based, semantic, database, application specific | All layers of IP stack, protocols and standards of each layer, IEEE 802.15.4, 6LoWPAN, NFC, ROLL routing, COAP, MQTT, LPWAN, low energy wireless communication technologies: BLE, Zigbee, RFID-WSN integration | Smart home, health, logistics, transport, social, environment, agriculture, energy | Various architectures of IoT |

upon the requirements. Applications can be built on the smartphone that uses sensor data to produce meaningful results. Some of the sensors inside a modern smartphone are as follows.

(1) The accelerometer senses the motion and acceleration of a mobile phone. It typically measures changes in velocity of the smartphone in three dimensions. There are many types of accelerometers [27].

In a mechanical accelerometer, we have a seismic mass in a housing, which is tied to the housing with a spring. The mass takes time to move and is left behind as the housing moves, so the force in the spring can be correlated with the acceleration. In a capacitive accelerometer, capacitive plates are used with the same setup. With a change in velocity, the mass pushes the capacitive plates together, thus changing the capacitance. The rate of change of capacitance is then converted into acceleration. In a piezoelectric accelerometer, piezoelectric crystals are used, which when squeezed generate an electric voltage. The changes in voltage can be translated into acceleration.

The data patterns captured by the accelerometer can be used to detect physical activities of the user such as running, walking, and bicycling.

(2) The gyroscope detects the orientation of the phone very precisely. Orientation is measured using capacitive changes when a seismic mass moves in a particular direction.

(3) The camera and microphone are very powerful sensors since they capture visual and audio information, which can then be analyzed and processed to detect various types of contextual information. For example, we can infer a user's current environment and the interactions that she is having. To make sense of the audio data, technologies such as voice recognition and acoustic features can be exploited.

(4) The magnetometer detects magnetic fields. This can be used as a digital compass and in applications to detect the presence of metals.

(5) The GPS (Global Positioning System) detects the location of the phone, which is one of the most important pieces of contextual information for smart applications. The location is detected using the principle of trilateration [28]. The distance is measured from three or more satellites (or mobile phone towers in the case of A-GPS) and coordinates are computed.

(6) The light sensor detects the intensity of ambient light. It can be used for setting the brightness of the screen and other applications in which some action is to be taken depending on the intensity of ambient light. For example, we can control the lights in a room.

(7) The proximity sensor uses an infrared (IR) LED, which emits IR rays. These rays bounce back when they strike some object. Based on the difference in time, we can calculate the distance. In this way, the distance to different objects from the phone can be measured. For example, we can use it to determine when the phone is close to the face while talking. It can also be used in applications in which we have to trigger some event when an object approaches the phone.

(8) Some smartphones such as Samsung's Galaxy S4 also have a thermometer, barometer, and humidity sensor to measure the temperature, atmospheric pressure, and humidity, respectively.

We have studied many smart applications that use sensor data collected from smartphones. For example, activity detection [29] is achieved by applying machine learning algorithms to the data collected by smartphone sensors. It detects activities such as running, going up and down stairs, walking, driving, and cycling. The application is trained with patterns of data using data sets recorded by sensors when these activities are being performed.

Many health and fitness applications are being built to keep track of a person's health continuously using smartphones. They keep track of users' physical activities, diet, exercises, and lifestyle to determine the fitness level and give suggestions to the user accordingly. Wang et al. [30] describe a mobile application that is based completely on a smartphone. They use it to assess the overall mental health and performance of a college student. To track the location and activities in which the student is involved, activity recognition (accelerometer) and GPS data are used. To keep a check on how much the student sleeps, the accelerometer and light sensors are used. For social life and conversations, audio data from a microphone is used. The application also conducts quick questionnaires with the students to know about their mood. All this data can be used to assess the stress levels, social life, behavior, and exercise patterns of a student.

Another application by McClernon and Choudhury [31] detects when the user is going to smoke using context information such as the presence of other smokers, location, and associated activities. The sensors provide information related to the user's movement, location, visual images, and surrounding sounds. To summarize smartphone sensors are being used to study different kinds of human behavior (see [32]) and to improve the quality of human life.

*5.2. Medical Sensors.* The Internet of Things can be really beneficial for health care applications. We can use sensors, which can measure and monitor various medical parameters in the human body [33]. These applications can aim at monitoring a patient's health when they are not in hospital or when they are alone. Subsequently, they can provide real time feedback to the doctor, relatives, or the patient. McGrath and Scanaill [34] have described in detail the different sensors that can be worn on the body for monitoring a person's health.

There are many wearable sensing devices available in the market. They are equipped with medical sensors that are capable of measuring different parameters such as the heart rate, pulse, blood pressure, body temperature, respiration rate, and blood glucose levels [35]. These wearables include

Figure 4: Smart watches and fitness trackers (source:https://www.pebble.com/ and http://www.fitbit.com/).

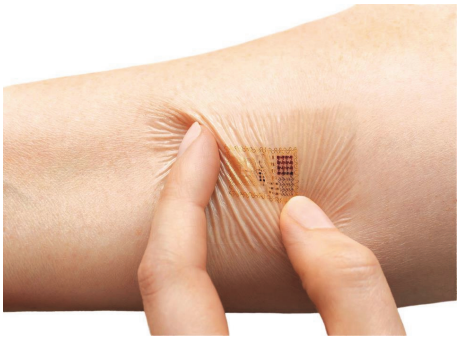


Figure 5: Embedded skin patches (source: MC10 Electronics).

smart watches, wristbands, monitoring patches, and smart textiles.

Moreover, smart watches and fitness trackers are becoming fairly popular in the market as companies such as Apple, Samsung, and Sony are coming up with very innovative features. For example, a smart watch includes features such as connectivity with a smartphone, sensors such as an accelerometer, and a heart rate monitor (see Figure 4).

Another novel IoT device, which has a lot of promise are monitoring patches that are pasted on the skin. Monitoring patches are like tattoos. They are stretchable and disposable and are very cheap. These patches are supposed to be worn by the patient for a few days to monitor a vital health parameter continuously [15]. All the electronic components are embedded in these rubbery structures. They can even transmit the sensed data wirelessly. Just like a tattoo, these patches can be applied on the skin as shown in Figure 5. One of the most common applications of such patches is to monitor blood pressure.

A very important consideration here is the context [34]. The data collected by the medical sensors must be combined with contextual information such as physical activity. For example, the heart rate depends on the context. It increases when we exercise. In that case, we cannot infer abnormal heart rate. Therefore, we need to combine data from different sensors for making the correct inference.

Figure 6: Brain sensing headband with embedded neurosensors (source: http://www.choosemuse.com/).

*5.3. Neural Sensors.* Today, it is possible to understand neural signals in the brain, infer the state of the brain, and train it for better attention and focus. This is known as neurofeedback [36] (see Figure 6). The technology used for reading brain signals is called EEG (Electroencephalography) or a brain computer interface. The neurons inside the brain communicate electronically and create an electric field, which can be measured from outside in terms of frequencies. Brain waves can be categorized into alpha, beta, gamma, theta, and delta waves depending upon the frequency.

Based on the type of wave, it can be inferred whether the brain is calm or wandering in thoughts. This type of neurofeedback can be obtained in real time and can be used to train the brain to focus, pay better attention towards things, manage stress, and have better mental well-being.

*5.4. Environmental and Chemical Sensors.* Environmental sensors are used to sense parameters in the physical environment such as temperature, humidity, pressure, water pollution, and air pollution. Parameters such as the temperature and pressure can be measured with a thermometer and barometer. Air quality can be measured with sensors, which sense the presence of gases and other particulate matter in the air (refer to Sekhar et al. [37] for more details).

Chemical sensors are used to detect chemical and biochemical substances. These sensors consist of a recognition element and a transducer. The electronic nose (e-nose) and electronic tongue (e-tongue) are technologies that can be used to sense chemicals on the basis of odor and taste, respectively [38]. The e-nose and e-tongue consist of an array of chemical sensors coupled with advance pattern recognition software. The sensors inside the e-nose and e-tongue produce complex data, which is then analyzed through pattern recognition to identify the stimulus.

These sensors can be used in monitoring the pollution level in smart cities [39], keeping a check on food quality in smart kitchens, testing food, and agricultural products in supply chain applications.

*5.5. Radio Frequency Identification (RFID).* RFID is an identification technology in which an RFID tag (a small chip with an antenna) carries data, which is read by a RFID reader. The tag transmits the data stored in it via radio waves. It is similar to bar code technology. But unlike a traditional bar code, it does not require line of sight communication between the tag and the reader and can identify itself from a distance even without a human operator. The range of RFID varies with the frequency. It can go up to hundreds of meters.

RFID tags are of two types: active and passive. Active tags have a power source and passive tags do not have any power source. Passive tags draw power from the electromagnetic waves emitted by the reader and are thus cheap and have a long lifetime [40, 41].

There are two types of RFID technologies: near and far [40]. A near RFID reader uses a coil through which we pass alternating current and generate a magnetic field. The tag has a smaller coil, which generates a potential due to the ambient changes in the magnetic field. This voltage is then coupled with a capacitor to accumulate a charge, which then powers up the tag chip. The tag can then produce a small magnetic field that encodes the signal to be transmitted, and this can be picked up by the reader.

In far RFID, there is a dipole antenna in the reader, which propagates EM waves. The tag also has a dipole antenna on which an alternating potential difference appears and it is powered up. It can then use this power to transmit messages.

RFID technology is being used in various applications such as supply chain management, access control, identity authentication, and object tracking. The RFID tag is attached to the object to be tracked and the reader detects and records its presence when the object passes by it. In this manner, object movement can be tracked and RFID can serve as a search engine for smart things.

For access control, an RFID tag is attached to the authorized object. For example, small chips are glued to the front of vehicles. When the car reaches a barricade on which there is a reader, it reads the tag data and decides whether it is an authorized car. If yes, it opens automatically. RFID cards are issued to the people, who can then be identified by a RFID reader and given access accordingly.

The low level data collected from the RFID tags can be transformed into higher level insights in IoT applications [42]. There are many user level tools available, in which all the data collected by particular RFID readers and data associated with the RFID tags can be managed. The high level data can be used to draw inferences and take further action.

*5.6. Actuators.* Let us look at some examples of actuators that are used in the Internet of Things. An actuator is a device, which can effect a change in the environment by converting electrical energy into some form of useful energy. Some examples are heating or cooling elements, speakers, lights, displays, and motors.

The actuators, which induce motion, can be classified into three categories, namely, electrical, hydraulic, and pneumatic actuators depending on their operation. Hydraulic actuators facilitate mechanical motion using fluid or hydraulic power. Pneumatic actuators use the pressure of compressed air and electrical ones use electrical energy.

As an example, we can consider a smart home system, which consists of many sensors and actuators. The actuators are used to lock/unlock the doors, switch on/off the lights or other electrical appliances, alert users of any threats through alarms or notifications, and control the temperature of a home (via a thermostat).

A sophisticated example of an actuator used in IoT is a digital finger, which is used to turn on/off the switches (or anything which requires small motion) and is controlled wirelessly.

## 6. Preprocessing

As smart things collect huge amount of sensor data, compute and storage resources are required to analyze, store, and process this data. The most common compute and storage resources are cloud based because the cloud offers massive data handling, scalability, and flexibility. But this will not be sufficient to meet the requirements of many IoT applications because of the following reasons [43].

(1) Mobility: most of the smart devices are mobile. Their changing location makes it difficult to communicate with the cloud data center because of changing network conditions across different locations.

(2) Reliable and real time actuation: communicating with the cloud and getting back responses takes time. Latency sensitive applications, which need real time responses, may not be feasible with this model. Also, the communication may be lossy due to wireless links, which can lead to unreliable data.

(3) Scalability: more devices means more requests to the cloud, thereby increasing the latency.

(4) Power constraints: communication consumes a lot of power, and IoT devices are battery powered. They thus cannot afford to communicate all the time.

To solve the problem of mobility, researchers have proposed mobile cloud computing (MCC) [44]. But there are still problems associated with latency and power. MCC also suffers from mobility problems such as frequently changing network conditions due to which problems such as signal fading and service degradation arise.

As a solution to these problems, we can bring some compute and storage resources to the edge of the network instead of relying on the cloud for everything. This concept is known as *fog computing* [11, 45] (also see Section 2.2). The fog can be viewed as a cloud, which is close to the ground. Data can be stored, processed, filtered, and analyzed on the edge of the network before sending it to the cloud through expensive communication media. The fog and cloud paradigms go together. Both of them are required for the optimal performance of IoT applications. A smart gateway [13] can be employed between underlying networks and the cloud to realize fog computing as shown in Figure 7.

The features of fog computing [11] are as follows:

(1) Low latency: less time is required to access computing and storage resources on fog nodes (smart gateways).

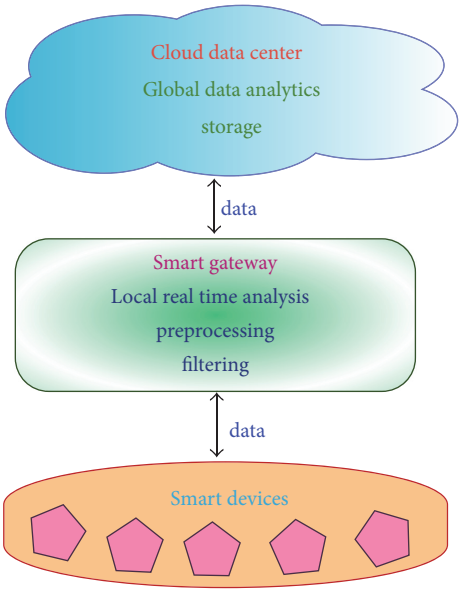



Figure 7: Smart gateway for preprocessing.

(2) Location awareness: as the fog is located on the edge of the network, it is aware of the location of the applications and their context. This is beneficial as context awareness is an important feature of IoT applications.

(3) Distributed nodes: fog nodes are distributed unlike centralized cloud nodes. Multiple fog nodes need to be deployed in distributed geographical areas in order to provide services to mobile devices in those areas. For example, in vehicular networks, deploying fog nodes at highways can provide low latency data/video streaming to vehicles.

(4) Mobility: the fog supports mobility as smart devices can directly communicate with smart gateways present in their proximity.

(5) Real time response: fog nodes can give an immediate response unlike the cloud, which has a much greater latency.

(6) Interaction with the cloud: fog nodes can further interact with the cloud and communicate only that data, which is required to be sent to the cloud.

The tasks performed by a smart gateway [46] are collecting sensor data, preprocessing and filtering collected data, providing compute, storage and networking services to IoT devices, communicating with the cloud and sending only necessary data, monitoring power consumption of IoT devices, monitoring activities and services of IoT devices, and ensuring security and privacy of data. Some applications of fog computing are as follows [10, 11]:

(1) Smart vehicular networks: smart traffic lights are deployed as smart gateways to locally detect pedestrians and vehicles through sensors, calculate their distance and speed, and finally infer traffic conditions. This is used to warn oncoming vehicles. These sensors also interact with neighboring smart traffic lights to perform traffic management tasks. For example, if sensors detect an approaching ambulance, they can change the traffic lights to let the ambulance pass first and also inform other lights to do so. The data collected by these smart traffic lights are locally analyzed in real time to serve real time needs of traffic management. Further, data from multiple gateways is combined and sent to the cloud for further global analysis of traffic in the city.

(2) Smart grid: the smart electrical grid facilitates load balancing of energy on the basis of usage and availability. This is done in order to switch automatically to alternative sources of energy such as solar and wind power. This balancing can be done at the edge of the network using smart meters or microgrids connected by smart gateways. These gateways can analyze and process data. They can then project future energy demand, calculate the availability and price of power, and supply power from both conventional and alternative sources to consumers.

## 7. Communication

As the Internet of Things is growing very rapidly, there are a large number of heterogeneous smart devices connecting to the Internet. IoT devices are battery powered, with minimal compute and storage resources. Because of their constrained nature, there are various communication challenges involved, which are as follows [19]:

(1) Addressing and identification: since millions of smart things will be connected to the Internet, they will have to be identified through a unique address, on the basis of which they communicate with each other. For this, we need a large addressing space, and a unique address for each smart object.

(2) Low power communication: communication of data between devices is a power consuming task, specially, wireless communication. Therefore, we need a solution that facilitates communication with low power consumption.

(3) Routing protocols with low memory requirement and efficient communication patterns.

(4) High speed and nonlossy communication.

(5) Mobility of smart things.

IoT devices typically connect to the Internet through the IP (Internet Protocol) stack. This stack is very complex and demands a large amount of power and memory from the connecting devices. The IoT devices can also connect locally through non-IP networks, which consume less power, and connect to the Internet via a smart gateway. Non-IP communication channels such as Bluetooth, RFID, and NFC are fairly popular but are limited in their range (up to a few meters). Therefore, their applications are limited to small personal area networks. Personal area networks (PAN) are

being widely used in IoT applications such as wearables connected to smartphones. For increasing the range of such local networks, there was a need to modify the IP stack so as to facilitate low power communication using the IP stack. One of the solutions is 6LoWPAN, which incorporates IPv6 with low power personal area networks. The range of a PAN with 6LoWPAN is similar to local area networks, and the power consumption is much lower.

The leading communication technologies used in the IoT world are IEEE 802.15.4, low power WiFi, 6LoWPAN, RFID, NFC, Sigfox, LoraWAN, and other proprietary protocols for wireless networks.

*7.1. Near Field Communication (NFC).* Near Field Communication [47–49] is a very short range wireless communication technology, through which mobile devices can interact with each other over a distance of few centimeters only. All types of data can be transferred between two NFC enabled devices in seconds by bringing them close to each other. This technology is based on RFID. It uses variations in the magnetic field to communicate data between two NFC enabled devices. NFC operates over a frequency band of 13.56 MHz, which is the same as high frequency RFID. There are two modes of operation: active and passive. In the active mode, both the devices generate magnetic fields, while in the passive mode, only one device generates the field and the other uses load modulation to transfer the data. The passive mode is useful in battery powered devices to optimize energy use. One benefit of the requirement of close proximity between devices is that it is useful for secure transactions such as payments. Finally, note that NFC can be used for two-way communication unlike RFID. Consequently, almost all smartphones in the market today are NFC enabled.

*7.2. Wireless Sensor Networks (WSN) Based on IP for Smart Objects.* Many times, data from a single sensor is not useful in monitoring large areas and complex activities. Different sensor nodes need to interact with each other wirelessly. The disadvantage of non-IP technologies such as RFID, NFC, and Bluetooth is that their range is very small. So, they cannot be used in many applications, where a large area needs to be monitored through many sensor nodes deployed in diverse locations. A wireless sensor network (WSN) consists of tens to thousands of sensor nodes connected using wireless technologies. They collect data about the environment and communicate it to gateway devices that relay the information to the cloud over the Internet. The communication between nodes in a WSN may be direct or multihop. The sensor nodes are of a constrained nature, but gateway nodes have sufficient power and processing resources. The popular network topologies used in a WSN are a star, a mesh, and a hybrid network. Most of the communication in WSN is based on the IEEE 802.15.4 standard (discussed in Section 7.3). There are clearly a lot of protocols that can be used in IoT scenarios. Let us discuss the design of a typical IoT network protocol stack with the most popular alternatives.

*7.3. IoT Network Protocol Stack.* The Internet Engineering Task Force (IETF) has developed alternative protocols for communication between IoT devices using IP because IP is a flexible and reliable standard [50, 51]. The Internet Protocol for Smart Objects (IPSO) Alliance has published various white papers describing alternative protocols and standards for the layers of the IP stack and an additional adaptation layer, which is used for communication [51–54] between smart objects.

*(1) Physical and MAC Layer (IEEE 802.15.4).* The IEEE 802.15.4 protocol is designed for enabling communication between compact and inexpensive low power embedded devices that need a long battery life. It defines standards and protocols for the physical and link (MAC) layer of the IP stack. It supports low power communication along with low cost and short range communication. In the case of such resource constrained environments, we need a small frame size, low bandwidth, and low transmit power.

Transmission requires very little power (maximum one milliwatt), which is only one percent of that used in WiFi or cellular networks. This limits the range of communication. Because of the limited range, the devices have to operate cooperatively in order to enable multihop routing over longer distances. As a result, the packet size is limited to 127 bytes only, and the rate of communication is limited to 250 kbps. The coding scheme in IEEE 802.15.4 has built in redundancy, which makes the communication robust, allows us to detect losses, and enables the retransmission of lost packets. The protocol also supports short 16-bit link addresses to decrease the size of the header, communication overheads, and memory requirements [55].

Readers can refer to the survey by Vasseur et al. [54] for more information on different physical and link layer technologies for communication between smart objects.

*(2) Adaptation Layer.* IPv6 is considered the best protocol for communication in the IoT domain because of its scalability and stability. Such bulky IP protocols were initially not thought to be suitable for communication in scenarios with low power wireless links such as IEEE 802.15.4.

6LoWPAN, an acronym for IPv6 over low power wireless personal area networks, is a very popular standard for wireless communication. It enables communication using IPv6 over the IEEE 802.15.4 [52] protocol. This standard defines an adaptation layer between the 802.15.4 link layer and the transport layer. 6LoWPAN devices can communicate with all other IP based devices on the Internet. The choice of IPv6 is because of the large addressing space available in IPv6. 6LoWPAN networks connect to the Internet via a gateway (WiFi or Ethernet), which also has protocol support for conversion between IPv4 and IPv6 as today's deployed Internet is mostly IPv4. IPv6 headers are not small enough to fit within the small 127 byte MTU of the 802.15.4 standard. Hence, squeezing and fragmenting the packets to carry only the essential information is an optimization that the adaptation layer performs.

Specifically, the adaptation layer performs the following three optimizations in order to reduce communication overhead [55]:

(i) *Header compression* 6loWPAN defines header compression of IPv6 packets for decreasing the overhead of IPv6. Some of the fields are deleted because they can be derived from link level information or can be shared across packets.

(ii) *Fragmentation*: the minimum MTU size (maximum transmission unit) of IPv6 is 1280 bytes. On the other hand, the maximum size of a frame in IEEE 802.15.4 is 127 bytes. Therefore, we need to fragment the IPv6 packet. This is done by the adaptation layer.

(iii) *Link layer forwarding* 6LoWPAN also supports mesh under routing, which is done at the link layer using link level short addresses instead of in the network layer. This feature can be used to communicate within a 6LoWPAN network.

*(3) Network Layer*. The network layer is responsible for routing the packets received from the transport layer. The IETF Routing over Low Power and Lossy Networks (ROLL) working group has developed a routing protocol (RPL) for Low Power and Lossy Networks (LLNs) [53].

For such networks, RPL is an open routing protocol, based on distance vectors. It describes how a destination oriented directed acyclic graph (DODAG) is built with the nodes after they exchange distance vectors. A set of constraints and an objective function is used to build the graph with the best path [53]. The objective function and constraints may differ with respect to their requirements. For example, constraints can be to avoid battery powered nodes or to prefer encrypted links. The objective function can aim to minimize the latency or the expected number of packets that need to be sent.

The making of this graph starts from the root node. The root starts sending messages to neighboring nodes, which then process the message and decide whether to join or not depending upon the constraints and the objective function. Subsequently, they forward the message to their neighbors. In this manner, the message travels till the leaf nodes and a graph is formed. Now all the nodes in the graph can send packets upwards hop by hop to the root. We can realize a point to point routing algorithm as follows. We send packets to a common ancestor, from which it travels downwards (towards leaves) to reach the destination.

To manage the memory requirements of nodes, nodes are classified into storing and nonstoring nodes depending upon their ability to store routing information. When nodes are in a nonstoring mode and a downward path is being constructed, the route information is attached to the incoming message and forwarded further till the root. The root receives the whole path in the message and sends a data packet along with the path message to the destination hop by hop. But there is a trade-off here because nonstoring nodes need more power and bandwidth to send additional route information as they do not have the memory to store routing tables.

*(4) Transport Layer*. TCP is not a good option for communication in low power environments as it has a large overhead owing to the fact that it is a connection oriented protocol. Therefore, UDP is preferred because it is a connectionless protocol and has low overhead.

*(5) Application Layer*. The application layer is responsible for data formatting and presentation. The application layer in the Internet is typically based on HTTP. However, HTTP is not suitable in resource constrained environments because it is fairly verbose in nature and thus incurs a large parsing overhead. Many alternate protocols have been developed for IoT environments such as CoAP (Constrained Application Protocol) and MQTT (Message Queue Telemetry Transport).

(a) Constrained Application Protocol: CoAP can be thought of as an alternative to HTTP. It is used in most IoT applications [56, 57]. Unlike HTTP, it incorporates optimizations for constrained application environments [50]. It uses the EXI (Efficient XML Interchanges) data format, which is a binary data format and is far more efficient in terms of space as compared to plain text HTML/XML. Other supported features are built in header compression, resource discovery, autoconfiguration, asynchronous message exchange, congestion control, and support for multicast messages. There are four types of messages in CoAP: nonconfirmable, confirmable, reset (nack), and acknowledgement. For reliable transmission over UDP, confirmable messages are used [58]. The response can be piggybacked in the acknowledgement itself. Furthermore, it uses DTLS (Datagram Transport Layer Security) for security purposes.

(b) Message Queue Telemetry Transport: MQTT is a publish/subscribe protocol that runs over TCP. It was developed by IBM [59] primarily as a client/server protocol. The clients are publishers/subscribers and the server acts as a broker to which clients connect through TCP. Clients can publish or subscribe to a topic. This communication takes place through the broker whose job is to coordinate subscriptions and also authenticate the client for security. MQTT is a lightweight protocol, which makes it suitable for IoT applications. But because of the fact that it runs over TCP, it cannot be used with all types of IoT applications. Moreover, it uses text for topic names, which increases its overhead.

MQTT-S/MQTT-SN is an extension of MQTT [60], which is designed for low power and low cost devices. It is based on MQTT but has some optimizations for WSNs as follows [61]. The topic names are replaced by topic IDs, which reduce the overheads of transmission. Topics do not need registration as they are preregistered. Messages are also split so that only the necessary information is sent. Further, for power conservation, there is an offline procedure for clients who are in a sleep state. Messages can be buffered and later read by clients when they wake up. Clients connect to the broker through a gateway device, which resides within the sensor network and connects to the broker.

*7.4. Bluetooth Low Energy (BLE).* Bluetooth Low Energy, also known as "Bluetooth Smart," was developed by the Bluetooth

Special Interest Group. It has a relatively shorter range and consumes lower energy as compared to competing protocols. The BLE protocol stack is similar to the stack used in classic Bluetooth technology. It has two parts: controller and host. The physical and link layer are implemented in the controller. The controller is typically a SOC (System on Chip) with a radio. The functionalities of upper layers are included in the host [62]. BLE is not compatible with classic Bluetooth. Let us look at the differences between classic Bluetooth and BLE [63, 64].

The main difference is that BLE does not support data streaming. Instead, it supports quick transfer of small packets of data (packet size is small) with a data rate of 1 Mbps.

There are two types of devices in BLE: master and slave. The master acts as a central device that can connect to various slaves. Let us consider an IoT scenario where a phone or PC serve as the master and mobile devices such as a thermostat, fitness tracker, smart watch, or any monitoring device act as slaves. In such cases, slaves must be very power efficient. Therefore, to save energy, slaves are by default in sleep mode and wake up periodically to receive packets from the master.

In classic Bluetooth, the connection is on all the time even if no data transfer is going on. Additionally, it supports 79 data channels (1 MHz channel bandwidth) and a data rate of 1 million symbols/s, whereas, BLE supports 40 channels with 2 MHz channel bandwidth (double of classic Bluetooth) and 1 million symbols/s data rate. BLE supports low duty cycle requirements as its packet size is small and the time taken to transmit the smallest packet is as small as 80 $\mu$s. The BLE protocol stack supports IP based communication also. An experiment conducted by Siekkinen et al. [65] recorded the number of bytes transferred per Joule to show that BLE consumes far less energy as compared to competing protocols such as Zigbee. The energy efficiency of BLE is 2.5 times better than Zigbee.

*7.5. Low Power WiFi.* The WiFi alliance has recently developed "WiFi HaLow," which is based on the IEEE 802.11ah standard. It consumes lower power than a traditional WiFi device and also has a longer range. This is why this protocol is suitable for Internet of Things applications. The range of WiFi HaLow is nearly twice that of traditional WiFi.

Like other WiFi devices, devices supporting WiFi HaLow also support IP connectivity, which is important for IoT applications. Let us look at the specifications of the IEEE 802.11ah standard [66, 67]. This standard was developed to deal with wireless sensor network scenarios, where devices are energy constrained and require relatively long range communication. IEEE 802.11ah operates in the sub-gigahertz band (900 MHz). Because of the relatively lower frequency, the range is longer since higher frequency waves suffer from higher attenuation. We can extend the range (currently 1 km) by lowering the frequency further; however, the data rate will also be lower and thus the tradeoff is not justified. IEEE 802.11ah is also designed to support large star shaped networks, where a lot of stations are connected to a single access point.

*7.6. Zigbee.* It is based on the IEEE 802.15.4 communication protocol standard and is used for personal area networks or PANs [68]. The IEEE 802.15.4 standard has low power MAC and physical layers and has already been explained in Section 7.3. Zigbee was developed by the Zigbee alliance, which works for reliable, low energy, and cheap communication solutions. The range of Zigbee device communication is very small (10–100 meters). The details of the network and application layers are also specified by the Zigbee standard. Unlike BLE, the network layer here provides for multihop routing.

There are three types of devices in a Zigbee network: FFD (Fully Functional Device), RFD (Reduced Functional Device), and one Zigbee coordinator. A FFD node can additionally act as a router. Zigbee supports star, tree, and mesh topologies. The routing scheme depends on the topology. Other features of Zigbee are discovery and maintenance of routes, support for nodes joining/leaving the network, short 16-bit addresses, and multihop routing.

The framework for communication and distributed application development is provided by the application layer. The application layer consists of Application Objects (APO), Application Sublayer (APS), and a Zigbee Device Object (ZDO). APOs are spread over the network nodes. These are pieces of software, which control some underlying device hardware (examples: switch and transducer). The device and network management services are provided by the ZDO, which are then used by the APOs. Data transfer services are provided by the Application Sublayer to the APOs and ZDO. It is responsible for secure communication between the Application Objects. These features can be used to create a large distributed application.

*7.7. Integration of RFID and WSN.* RFID and wireless sensor networks (WSN) are both important technologies in the IoT domain. RFID can only be used for object identification, but WSNs serve a far greater purpose. The two are very different but merging them has many advantages. The following components can be added to RFID to enhance its usability:

(a) Sensing capabilities

(b) Multihop communication

(c) Intelligence

RFID is inexpensive and uses very little power. That is why its integration with WSN is very useful. The integration is possible in the following ways [69, 70]:

(a) *Integration of RFID tags with sensors:* RFID tags with sensing capabilities are called sensor tags. These sensor tags sense data from the environment and then the RFID reader can read this sensed data from the tag. In such cases, simple RFID protocols are used, where there is only single hop communication. RFID sensing technologies can be further classified on the basis of the power requirement of sensor tags as explained earlier in the section on RFIDs (active and passive) (see Section 5.5).

(b) *Integration of RFID tags with WSN nodes:* the communication capabilities of sensor tags are limited to a single hop. To extend its capabilities, the sensor tag is equipped with a wireless transceiver, little bit of Flash memory, and computational capabilities such that it can initiate communication with other nodes and wireless devices. The nodes can in this fashion be used to form a wireless mesh network. In such networks, sensor tags can communicate with each other over a large range (via intermediate hops). With additional processing capabilities at a node, we can reduce the net amount of data communicated and thus increase the power efficiency of the WSN.

(c) *Integration of RFID readers with WSN nodes:* this type of integration is also done to increase the range of RFID tag readers. The readers are equipped with wireless transceivers and microcontrollers so that they can communicate with each other and therefore, the tag data can reach a reader, which is not in the range of that tag. It takes advantage of multihop communication of wireless sensor network devices. The data from all the RFID readers in the network ultimately reaches a central gateway or base station that processes the data or sends it to a remote server.

These kinds of integrated solutions have many applications in a diverse set of domains such as security, healthcare, and manufacturing.

*7.8. Low Power Wide-Area-Networks (LPWAN).* Let us now discuss a protocol for long range communication in power constrained devices. The LPWAN class of protocols is low bit-rate communication technologies for such IoT scenarios.

Let us now discuss some of the most common technologies in this area.

*Narrow band IoT*: it is a technology made for a large number of devices that are energy constrained. It is thus necessary to reduce the bit rate. This protocol can be deployed with both the cellular phone GSM and LTE spectra. The downlink speeds vary between 40 kbps (LTE M2) and 10 Mbps (LTE category 1).

*Sigfox*: it is one more protocol that uses narrow band communication (≈10 MHz). It uses free sections of the radio spectrum (ISM band) to transmit its data. Instead of 4G networks, Sigfox focuses on using very long waves. Thus, the range can increase to a 1000 kms. Because of this the energy for transmission is significantly lower (<0.1%) than contemporary cell phones.

Again the cost is bandwidth. It can only transmit 12 bytes per message, and a device is limited to 140 messages per day. This is reasonable for many kinds of applications: submarine applications, sending control (emergency) codes, geolocation, monitoring remote locations, and medical applications.

*Weightless*: it uses a differential binary phase shift keying based method to transmit narrow band signals. To avoid interference, the protocol hops across frequency bands (instead of using CSMA). It supports cryptographic encryption and mobility. Along with frequency hopping, two additional mechanisms are used to reduce collisions. The downlink service uses time division multiple access (TDMA) and the uplink service uses multiple subchannels that are first allocated to transmitting nodes by contacting a central server. Some applications include smart meters, vehicle tracking, health monitoring, and industrial machine monitoring.

*Neul*: this protocol operates in the sub-1 GHz band. It uses small chunks of the TV whitespace spectrum to create low cost and low power networks with very high scalability. It has a 10 km range and uses the Weightless protocol for communication.

*LoRaWAN*: this protocol is similar to Sigfox. It targets wide area network applications and is designed to be a low power protocol. Its data rates can vary from 0.3 kbps to 50 kbps, and it can be used within an urban or a suburban environment (2–5 kms range in a crowded urban area). It was designed to serve as a standard for long range IoT protocols. It thus has features to support multitenancy, enable multiple applications, and include several different network domains.

*7.9. Lightweight Application Layer Protocols.* Along with physical and MAC layer protocols, we also need application layer protocols for IoT networks. These lightweight protocols need to be able to carry application messages, while simultaneously reducing power as far as possible.

OMA Lightweight M2M (LWM2M) is one such protocol. It defines the communication protocol between a server and a device. The devices often have limited capabilities and are thus referred to as *constrained devices*. The main aims of the OMA protocol are as follows:

(1) Remote device management.

(2) Transferring service data/information between different nodes in the LWM2M network.

All the protocols in this class treat all the network resources as *objects*. Such resources can be created, deleted, and remotely configured. These devices have their unique limitations and can use different kinds of protocols for internally representing information. The LWM2M protocol abstracts all of this away and provides a convenient interface to send messages between a generic LWM2M server and a distributed set of LWM2M clients.

This protocol is often used along with CoAP (Constrained Application Protocol). It is an application layer protocol that allows *constrained* nodes such as sensor motes or small embedded devices to communicate across the Internet. CoAP seamlessly integrates with HTTP, yet it provides additional facilities such as support for multicast operations. It is ideally

suited for small devices because of its low overhead and parsing complexity and reliance on UDP rather than TCP.

## 8. Middleware

Ubiquitous computing is the core of the Internet of Things, which means incorporating computing and connectivity in all the things around us. Interoperability of such heterogeneous devices needs well-defined standards. But standardization is difficult because of the varied requirements of different applications and devices. For such heterogeneous applications, the solution is to have a middleware platform, which will abstract the details of the *things* for applications. That is, it will hide the details of the smart things. It should act as a software bridge between the things and the applications. It needs to provide the required services to the application developers [20] so that they can focus more on the requirements of applications rather than on interacting with the baseline hardware. To summarize, the middleware abstracts the hardware and provides an Application Programming Interface (API) for communication, data management, computation, security, and privacy.

The challenges, which are addressed by any IoT middleware, are as follows: [20, 71, 72].

(1) *Interoperability and programming abstractions*: for facilitating collaboration and information exchange between heterogeneous devices, different types of things can interact with each other easily with the help of middleware services. Interoperability is of three types: network, semantic, and syntactic. Network interoperability deals with heterogeneous interface protocols for communication between devices. It insulates the applications from the intricacies of different protocols. Syntactic interoperability ensures that applications are oblivious of different formats, structures, and encoding of data. Semantic interoperability deals with abstracting the meaning of data within a particular domain. It is loosely inspired by the semantic web.

(2) *Device discovery and management*: this feature enables the devices to be aware of all other devices in the neighborhood and the services provided by them. In the Internet of Things, the infrastructure is mostly dynamic. The devices have to announce their presence and the services they provide. The solution needs to be scalable because the devices in an IoT network can increase. Most solutions in this domain are loosely inspired by semantic web technologies. The middleware provides APIs to list the IoT devices, their services, and capabilities. In addition, typically APIs are provided to discover devices based on their capabilities. Finally, any IoT middleware needs to perform load balancing, manage devices based on their levels of battery power, and report problems in devices to the users.

(3) *Scalability*: a large number of devices are expected to communicate in an IoT setup. Moreover, IoT applications need to scale due to ever increasing requirements. This should be managed by the middleware by making required changes when the infrastructure scales.

(4) *Big data and analytics*: IoT sensors typically collect a huge amount of data. It is necessary to analyze all of this data in great detail. As a result a lot of big data algorithms are used to analyze IoT data. Moreover, it is possible that due to the flimsy nature of the network some of the data collected might be incomplete. It is necessary to take this into account and extrapolate data by using sophisticated machine learning algorithms.

(5) *Security and privacy*: IoT applications are mostly related to someone's personal life or an industry. Security and privacy issues need to be addressed in all such environments. The middleware should have built in mechanisms to address such issues, along with user authentication, and the implementation of access control.

(6) *Cloud services*: the cloud is an important part of an IoT deployment. Most of the sensor data is analyzed and stored in a centralized cloud. It is necessary for IoT middleware to seamlessly run on different types of clouds and to enable users to leverage the cloud to get better insights from the data collected by the sensors.

(7) *Context detection*: the data collected from the sensors needs to be used to extract the context by applying various types of algorithms. The context can subsequently be used for providing sophisticated services to users.

There are many middleware solutions available for the Internet of Things, which address one or more of the aforementioned issues. All of them support interoperability and abstraction, which is the foremost requirement of middleware. Some examples are Oracle's Fusion Middleware, OpenIoT [21], MiddleWhere [22], and Hydra [23]. Middlewares can be classified as follows on the basis of their design [72]:

(1) *Event based*: here, all the components interact with each other through events. Each event has a type and some parameters. Events are generated by producers and received by the consumers. This can be viewed as a publish/subscribe architecture, where entities can subscribe for some event types and get notified for those events.

(2) *Service oriented*: service oriented middlewares are based on Service Oriented Architectures (SOA), in which we have independent modules that provide services through accessible interfaces. A service oriented middleware views resources as service providers. It abstracts the underlying resources through a set of services that are used by applications. There is

a service repository, where services are published by providers. The consumers can discover services from the repository and then bind with the provider to access the service. Service oriented middleware must have runtime support for advertising services by providers and support for discovering and using services by consumers.

HYDRA [23] is a service oriented middleware. It incorporates many software components, which are used in handling various tasks required for the development of intelligent applications. Hydra also provides semantic interoperability using semantic web technologies. It supports dynamic reconfiguration and self-management.

(3) *Database oriented*: in this approach, the network of IoT devices is considered as a virtual relational database system. The database can then be queried by the applications using a query language. There are easy to use interfaces for extracting data from the database. This approach has issues with scaling because of its centralized model.

(4) *Semantic*: semantic middleware focuses on the interoperation of different types of devices, which communicate using different formats of data. It incorporates devices with different data formats and ontologies and ties all of them together in a common framework. The framework is used for exchanging data between diverse types of devices. For a common semantic format, we need to have $N$ adapters for communication between $N$ devices because; for each device, we need adapters to map $N$ standards to one abstract standard [73]. In such a semantic middleware [74], a semantic layer is introduced, in which there is a mapping from each resource to a software layer for that resource. The software layers then communicate with each other using a mutually intelligible language (based on the semantic web). This technique allows multiple physical resources to communicate even though they do not implement or understand the same protocols.

(5) *Application specific*: this type of middleware is used specifically for an application domain for which it is developed because the whole architecture of this middleware software is fine-tuned on the basis of requirements of the application. The application and middleware are tightly coupled. These are not general purpose solutions.

### 8.1. Popular IoT Middleware

*8.1.1. FiWare.* FiWare is a very popular IoT middleware framework that is promoted by the EU. It has been designed keeping smart cities, logistics, and shop floor analytics in mind. FiWare contains a large body of code, reusable modules, and APIs that have been contributed by thousands of FiWare developers. Any application developer can take a subset of these components and build his/her IoT application.

A typical IoT application has many producers of data (sensors), a set of servers to process the data, and a set of actuators. FiWare refers to the information collected by sensors as *context information*. It defines generic REST APIs to capture the context from different scenarios. All the context information is sent to a dedicated service called a context broker. FiWare provides APIs to store the context and also query it. Moreover, any application can register itself as a context consumer, and it can request the context broker for information. It also supports the publish-subscribe paradigm. Subsequently, the context can be supplied to systems using context adapters whose main role is to transform the data (the context) based on the requirements of the destination nodes. Moreover, FiWare defines a set of SNMP APIs via which we can control the behavior of IoT devices and also configure them.

The target applications are provided APIs to analyze, query, and mine the information that is collected from the context broker. Additionally, with advanced visualization APIs, it is possible to create and deploy feature rich applications very quickly.

*8.1.2. OpenIoT.* OpenIoT is another popular open source initiative. It has 7 different components. At the lowest level, we have a physical plane. It collects data from IoT devices and also does some preprocessing of data. It has different APIs to interface with different types of physical nodes and get information from them.

The next plane is the virtualized plane, which has 3 components. We first have the scheduler, which manages the streams of data generated by devices. It primarily assigns them to resources and takes care of their QoS requirements. The data storage component manages the storage and archival of data streams. Finally, the service delivery component processes the streams. It has several roles. It combines data streams, preprocesses them, and tracks some statistics associated with these streams such as the number of unique requests or the size of each request.

The uppermost layer, that is, the application layer, also has 3 components: request definition, request presentation, and configuration. The request definition component helps us create requests to be sent to the IoT sensors and storage layers. It can be used to fetch and query data. The request presentation component creates mashups of data by issuing different queries to the storage layer, and finally the configuration component helps us configure the IoT devices.

## 9. Applications of IoT

There are a diverse set of areas in which intelligent applications have been developed. All of these applications are not yet readily available; however, preliminary research indicates the potential of IoT in improving the quality of life in our society. Some uses of IoT applications are in home automation, fitness tracking, health monitoring, environment protection, smart cities, and industrial settings.

*9.1. Home Automation.* Smart homes are becoming more popular today because of two reasons. First, the sensor and actuation technologies along with wireless sensor networks have significantly matured. Second, people today trust

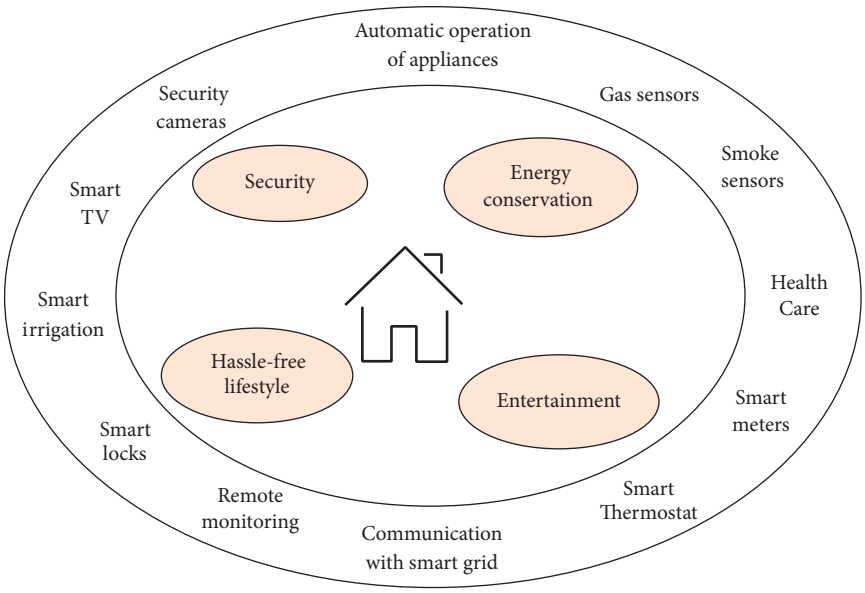



Figure 8: Block diagram of a smart home system.

technology to address their concerns about their quality of life and security of their homes (see Figure 8).

In smart homes, various sensors are deployed, which provide intelligent and automated services to the user. They help in automating daily tasks and help in maintaining a routine for individuals who tend to be forgetful. They help in energy conservation by turning off lights and electronic gadgets automatically. We typically use motion sensors for this purpose. Motion sensors can be additionally used for security also.

For example, the project, MavHome [75], provides an intelligent agent, which uses various prediction algorithms for doing automated tasks in response to user triggered events and adapts itself to the routines of the inhabitants. Prediction algorithms are used to predict the sequence of events [76] in a home. A sequence matching algorithm maintains sequences of events in a queue and also stores their frequency. Then a prediction is made using the match length and frequency. Other algorithms used by similar applications use compression based prediction and Markov models.

Energy conservation in smart homes [77] is typically achieved through sensors and context awareness. The sensors collect data from the environment (light, temperature, humidity, gas, and fire events). This data from heterogeneous sensors is fed to a context aggregator, which forwards the collected data to the context aware service engine. This engine selects services based on the context. For example, an application can automatically turn on the AC when the humidity rises. Or, when there is a gas leak, it can turn all the lights off.

Smart home applications are really beneficial for the elderly and differently abled. Their health is monitored and relatives are informed immediately in case of emergencies. Floors are equipped with pressure sensors, which track the movement of an individual across the smart home and also help in detecting if a person has fallen down. In smart homes, CCTV cameras can be used to record events of interest. These can then be used for feature extraction to find out what is going on.

In specific, fall detection applications in smart environments [78–80] are useful for detecting if elderly people have fallen down. Yu et al. [80] use computer vision based techniques for analyzing postures of the human body. Sixsmith et al. [79] used low cost infrared sensor array technology, which can provide information such as the location, size, and velocity of a target object. It detects dynamics of a fall by analyzing the motion patterns and also detects inactivity and compares it with activity in the past. Neural networks are employed and sample data is provided to the system for various types of falls. Many smartphone based applications are also available, which detect a fall on the basis of readings from the accelerometer and gyroscope data.

There are many challenges and issues with regard to smart home applications [81]. The most important is security and privacy [82] since all the data about the events taking place in the home is being recorded. If the security and trustworthiness of the system are not guaranteed, an intruder may attack the system and may make the system behave maliciously. Smart home systems are supposed to notify the owners in case they detect such abnormalities. This is possible using AI and machine learning algorithms, and researchers have already started working in this direction [83]. Reliability is also an issue since there is no system administrator to monitor the system.

### 9.2. Smart Cities

*9.2.1. Smart Transport.* Smart transport applications can manage daily traffic in cities using sensors and intelligent information processing systems. The main aim of intelligent transport systems is to minimize traffic congestion, ensure

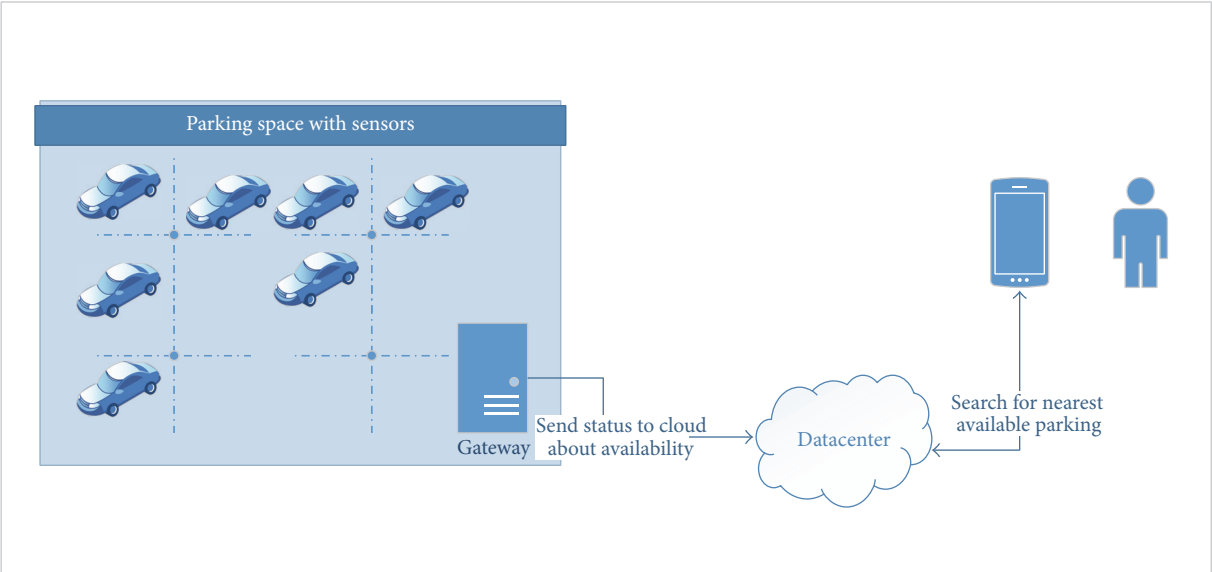



Figure 9: Block diagram of a smart parking system.

easy and hassle-free parking, and avoid accidents by properly routing traffic and spotting drunk drivers. The sensor technologies governing these types of applications are GPS sensors for location, accelerometers for speed, gyroscopes for direction, RFIDs for vehicle identification, infrared sensors for counting passengers and vehicles, and cameras for recording vehicle movement and traffic. There are many types of applications in this area (refer to [84]):

(1) Traffic surveillance and management applications: vehicles are connected by a network to each other, the cloud, and to a host of IoT devices such as GPS sensors, RFID devices, and cameras. These devices can estimate traffic conditions in different parts of the city. Custom applications can analyze traffic patterns so that future traffic conditions can be estimated. Yu et al. [85] implement a vehicle tracking system for traffic surveillance using video sequences captured on the roads.

Traffic congestion detection can also be implemented using smartphone sensors such as accelerometers [86] and GPS sensors. These applications can detect movement patterns of the vehicle while the user is driving. Such kind of information is already being collected by Google maps and users are using it to route around potentially congested areas of the city.

(2) Applications to ensure safety: smart transport does not only imply managing traffic conditions. It also includes safety of people travelling in their vehicles, which up till now was mainly in the hands of drivers. There are many IoT applications developed to help drivers become safer drivers. Such applications monitor driving behavior of drivers and help them drive safely by detecting when they are feeling drowsy or tired and helping them to cope with it or suggesting rest [87, 88]. Technologies used in such applications are face detection, eye movement detection, and pressure detection on the steering (to measure the grip of the driver's hands on the steering).

A smartphone application, which estimates the driver's driving behavior using smartphone sensors such as the accelerometer, gyroscope, GPS, and camera, has been proposed by Eren et al. [89]. It can decide whether the driving is safe or rash by analyzing the sensor data.

(3) Intelligent parking management (see Figure 9): in a smart transportation system, parking is completely hassle free as one can easily check on the Internet to find out which parking lot has free spaces. Such lots use sensors to detect if the slots are free or occupied by vehicles. This data is then uploaded to a central server.

(4) Smart traffic lights: traffic lights equipped with sensing, processing, and communication capabilities are called smart traffic lights. These lights sense the traffic congestion at the intersection and the amount of traffic going each way. This information can be analyzed and then sent to neighboring traffic lights or a central controller. It is possible to use this information creatively. For example, in an emergency situation the traffic lights can preferentially give way to an ambulance. When the smart traffic light senses an ambulance coming, it clears the path for it and also informs neighboring lights about it. Technologies used in these lights are cameras, communication technologies, and data analysis modules. Such systems have already been deployed in Rio De Janeiro.

(5) Accident detection applications: a smartphone application designed by White et al. [90] detects the occurrence of an accident with the help of an accelerometer and acoustic data. It immediately sends this information along with the location to the nearest hospital.

Some additional situational information such as on-site photographs is also sent so that the first responders know about the whole scenario and the degree of medical help that is required.

*9.2.2. Smart Water Systems.* Given the prevailing amount of water scarcity in most parts of the world, it is very important to manage our water resources efficiently. As a result most cities are opting for smart solutions that place a lot of meters on water supply lines and storm drains. A good reference in this area is the paper by Hauber-Davidson and Idris [91]. They describe various designs for smart water meters. These meters can be used to measure the degree of water inflow and outflow and to identify possible leaks. Smart water metering systems are also used in conjunction with data from weather satellites and river water sensors. They can also help us predict flooding.

*9.2.3. Examples of Smart Cities.* Barcelona and Stockholm stand out in the list of smart cities. Barcelona has a *CityOS* project, where it aims to create a single virtualized OS for all the smart devices and services offered within the city. Barcelona has mainly focused on smart transportation (as discussed in Section 9.2.1) and smart water. Smart transportation is implemented using a network of sensors, centralized analysis, and smart traffic lights. On similar lines Barcelona has sensors on most of its storm drains, water storage tanks, and water supply lines. This information is integrated with weather and usage information. The result of all of this is a centralized water planning strategy. The city is able to estimate the water requirements in terms of domestic usage and industrial usage and for activities such as landscaping and gardening.

Stockholm started way back in 1994, and its first step in this direction was to install an extensive fiber optic system. Subsequently, the city added thousands of sensors for smart traffic and smart water management applications. Stockholm was one of the first cities to implement *congestion* charging. Users were charged money, when they drove into congested areas. This was enabled by smart traffic technologies. Since the city has a solid network backbone, it is very easy to deploy sensors and applications. For example, recently the city created a smart parking system, where it is possible to easily locate parking spots nearby. Parking lots have sensors, which let a server know about their usage. Once a driver queries the server with her/his GPS location, she/he is guided to the nearest parking lot with free slots. Similar innovations have taken place in the city's smart buildings, snow clearance, and political announcement systems.

*9.3. Social Life and Entertainment.* Social life and entertainment play an important role in an individual's life. Many applications have been developed, which keep track of such human activities. The term "opportunistic IoT" [92] refers to information sharing among opportunistic devices (devices that seek to make contact with other devices) based on movement and availability of contacts in the vicinity. Personal devices such as tablets, wearables, and mobile phones have sensing and short range communication capabilities. People can find and interact with each other when there is a common purpose.

Circle Sense [93] is an application, which detects social activities of a person with the help of various types of sensor data. It identifies the social circle of a person by analyzing the patterns of social activities and the people present in those activities. Various types of social activities and the set of people participating in those activities are identified. It uses location sensors to find out where the person is and uses Bluetooth for searching people around her. The system has built in machine learning algorithms, and it gradually improves its behavior with learning.

Affective computing [94] is a technology, which recognizes, understands, stimulates, and responds to the emotions of human beings. There are many parameters, which are considered while dealing with human affects such as facial expressions, speech, body gestures, hand movements, and sleep patterns. These are analyzed to figure out how a person is feeling. The utterance of emotional keywords is identified by voice recognition and the quality of voice by looking at acoustic features of speech.

One of the applications of affective computing is Camy, an artificial pet dog [95], which is designed to interact with human beings and show affection and emotions. Many sensors and actuators are embedded in it. It provides emotional support to the owner, encourages playful and active behavior, recognizes its owner, and shows love for her and increases the owner's communication with other people. Based on the owner's mood, Camy interacts with the owner and gives her suggestions.

Logmusic [96] is an entertainment application, which recommends music on the basis of the context, such as the weather, temperature, time, and location.

*9.4. Health and Fitness.* IoT appliances have proven really beneficial in the health and wellness domains. Many wearable devices are being developed, which monitor a person's health condition (see Figure 10).

Health applications make independent living possible for the elderly and patients with serious health conditions. Currently, IoT sensors are being used to continuously monitor and record their health conditions and transmit warnings in case any abnormal indicators are found. If there is a minor problem, the IoT application itself may suggest a prescription to the patient.

IoT applications can be used in creating an Electronic Health Record (EHR), which is a record of all the medical details of a person. It is maintained by the health system. An EHR can be used to record allergies, surges in blood sugar and blood pressure.

Stress recognition applications are also fairly popular [97]. They can be realized using smartphone sensors. Wang et al. describe an application [30], which measures the stress level of a college student and is installed on the student's smartphone. It senses the locations the student visits during the whole day, the amount of physical activity, amount of sleep and rest, and her/his interaction and relationships with other people (audio data and calls). In addition, it also conducts surveys with the student by randomly popping up

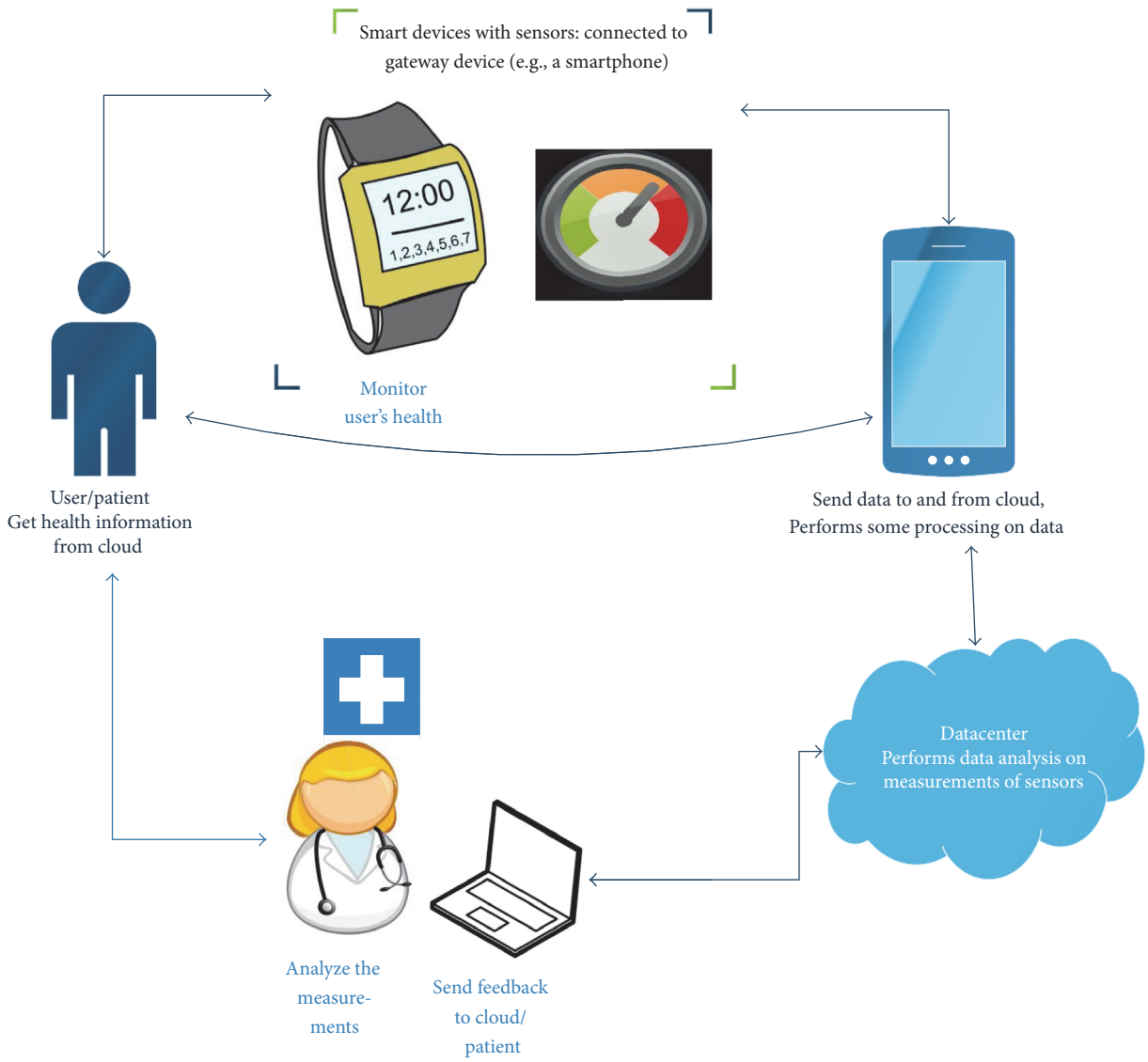



Figure 10: Block diagram of a smart healthcare system.

a question in the smartphone. Using all of this data and analyzing it intelligently, the level of stress and academic performance can be measured.

In the fitness sector, we have applications that monitor how fit we are based on our daily activity level. Smartphone accelerometer data can be used for activity detection by applying complex algorithms. For example, we can measure the number of steps taken and the amount of exercise done by using fitness trackers. Fitness trackers are available in the market as wearables to monitor the fitness level of an individual. In addition, gym apparatus can be fitted with sensors to count the number of times an exercise is performed. For example, a smart mat [98] can count the number of exercise steps performed on it. This is implemented using pressure sensors on the mat and then by analyzing the patterns of pressure, and the shape of the contact area.

*9.5. Smart Environment and Agriculture.* Environmental parameters such as temperature and humidity are important for agricultural production. Sensors are used by farmers in the field to measure such parameters and this data can be used for efficient production. One application is automated irrigation according to weather conditions.

Production using greenhouses [99] is one of the main applications of IoT in agriculture. Environmental parameters measured in terms of temperature, soil information, and humidity are measured in real time and sent to a server for analysis. The results are then used to improve crop quality and yield.

Pesticide residues in crop production are detected using an Acetylcholinesterase biosensor [100]. This data is saved and analyzed for extracting useful information such as the sample size, time, location, and amount of residues. We can

thus maintain the quality of the crop. Moreover, a QR code can be used to uniquely identify a carton of farm produce. Consumers can scan the QR code and check the amount of pesticides in it (via a centralized database) online before buying.

Air pollution is an important concern today because it is changing the climate of the earth and degrading air quality. Vehicles cause a lot of air pollution. An IoT application proposed by Manna et al. [39] monitors air pollution on the roads. It also tracks vehicles that cause an undue amount of pollution. Electrochemical toxic gas sensors can also be used to measure air pollution. Vehicles are identified by RFID tags. RFID readers are placed on both sides of the road along with the gas sensors. With this approach it is possible to identify and take action against polluting vehicles.

*9.6. Supply Chain and Logistics.* IoT tries to simplify real world processes in business and information systems [101]. The goods in the supply chain can be tracked easily from the place of manufacture to the final places of distribution using sensor technologies such as RFID and NFC. Real time information is recorded and analyzed for tracking. Information about the quality and usability of the product can also be saved in RFID tags attached with the shipments.

Bo and Guangwen [102] explain an information transmission system for supply chain management, which is based on the Internet of Things. RFID tags uniquely identify a product automatically and a product information network is created to transmit this information in real time along with location information. This system helps in automatic collection and analysis of all the information related to supply chain management, which may help examine past demand and come up with a forecast of future demand. Supply chain components can get access to real time data and all of this information can be analyzed to get useful insights. This will in the long run improve the performance of supply chain systems.

*9.7. Energy Conservation.* The smart grid is information and communication technology enabled modern electricity generation, transmission, distribution, and consumption system [103].

To make electric power generation, transmission, and distribution smart, the concept of smart grids adds intelligence at each step and also allows the two-way flow of power (back from the consumer to the supplier). This can save a lot of energy and help consumers better understand the flow of power and dynamic pricing. In a smart grid, power generation is distributed. There are sensors deployed throughout the system to monitor everything. It is actually a distributed network of microgrids [104]. Microgrids generate power to meet demands of local sites and transmit back the surplus energy to the central grid. Microgrids can also demand energy from the central grid in case of a shortfall.

Two-way flow of power also benefits consumers, who are also using their own generated energy occasionally (say, solar, or wind power); the surplus power can be transmitted back so that it is not wasted. The user will also get paid for that power.

Some of the IoT applications in a smart grid are online monitoring of transmission lines for disaster prevention and efficient use of power in smart homes by having a smart meter for monitoring energy consumption [105].

Smart meters read and analyze consumption patterns of power at regular and peak load times. This information is then sent to the server and also made available to the user. The generation is then set according to the consumption patterns. In addition, the user can adjust her/his use so as to reduce costs. Smart power appliances can leverage this information and operate when the prices are low.

## 10. Design Considerations in an IoT System

Now, that we have profiled most of the IoT technologies, let us look at some of the design considerations for designing a practical IoT network.

The first consideration is the design of the sensors. Even though there might not be much of a choice regarding the sensors, there is definitely a lot of choice regarding the processing and networking capabilities that are bundled along with the sensors. Our choices range from small sub-mW boards meant for sensor motes to Arduino or Atom boards that consume 300–500 mW of power. This choice depends on the degree of analytics and data preprocessing that we want to perform at the sensor itself. Secondly, there is an issue of logistics also. To create a sub-mW board, we need board design expertise, and this might not be readily available. Hence, it might be advisable to bundle a sensor with commercially available embedded processor kits.

The next important consideration is communication. In IoT nodes, power is the most dominant issue. The power required to transmit and receive messages is a major fraction of the overall power, and as a result a choice of the networking technology is vital. The important factors that we need to consider are the distance between the sender and the receiver, the nature of obstacles, signal distortion, ambient noise, and governmental regulations. Based on these key factors, we need to choose a given wireless networking protocol. For example, if we just need to communicate inside a small building, we can use Zigbee, whereas if we need communication in a smart city, we should choose Sigfox or LoraWAN. In addition, often there are significant constraints on the frequency and the power that can be spent in transmission. These limitations are mainly imposed by government agencies. An apt decision needs to be made by taking all of these factors into account.

Let us then come to the middleware. The first choice that needs to be made is to choose between an open source middleware such as FiWare or a proprietary solution. There are pros and cons of both. It is true that open source middleware is in theory more flexible; however, they may have limited support for IoT devices. We ideally want a middleware solution to interoperate with all kinds of communication protocols and devices; however, that might not be the case. Hence, if we need strict compatibility with certain devices and protocols, a proprietary solution is better. Nevertheless, open source offerings have cost advantages and are sometimes easier to deploy. We also need to choose the communication

protocol and ensure that it is compatible with the firewalls in the organizations involved. In general choosing a protocol based on HTTP is the best from this point of view. We also need to choose between TCP and UDP. UDP is always better from the point of view of power consumption. Along with these considerations, we also need to look at options to store sensor data streams, querying languages, and support for generating dynamic alerts.

Finally, let us consider the application layer. Most IoT frameworks provide significant amount of support for creating the application layer. This includes data mining, data processing, and visualization APIs. Creating mashups and dashboards of data is nowadays very easy to do given the extensive support provided by IoT frameworks. Nevertheless, here the tradeoff is between the features provided and the resources that are required. We do not need a very heavy framework if we do not desire a lot of features. This call needs to be taken by the application developers.

## 11. Conclusion

In this survey paper we presented a survey of the current technologies used in the IoT domain as of 2016. Currently, this field is in a very nascent stage. The technologies in the core infrastructure layers are showing signs of maturity. However, a lot more needs to happen in the areas of IoT applications and communication technologies. These fields will definitely mature and impact human life in inconceivable ways over the next decade.

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

[1] O. Vermesan, P. Friess, P. Guillemin et al., "Internet of things strategic research roadmap," in *Internet of Things: Global Technological and Societal Trends*, vol. 1, pp. 9–52, 2011.

[2] I. Peña-López, *Itu Internet Report 2005: The Internet of Things*, 2005.

[3] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, "Choices for interaction with things on Internet and underlying issues," *Ad Hoc Networks*, vol. 28, pp. 68–90, 2015.

[4] O. Said and M. Masud, "Towards internet of things: survey and future vision," *International Journal of Computer Networks*, vol. 5, no. 1, pp. 1–17, 2013.

[5] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, "Research on the architecture of internet of things," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, vol. 5, pp. V5-484–V5-487, IEEE, Chengdu, China, August 2010.

[6] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *Proceedings of the 10th International Conference on Frontiers of Information Technology (FIT '12)*, pp. 257–260, December 2012.

[7] H. Ning and Z. Wang, "Future internet of things architecture: like mankind neural system or social organization framework?" *IEEE Communications Letters*, vol. 15, no. 4, pp. 461–463, 2011.

[8] M. Weyrich and C. Ebert, "Reference architectures for the internet of things," *IEEE Software*, vol. 33, no. 1, pp. 112–116, 2016.

[9] 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, 2013.

[10] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: a platform for internet of things and analytics," in *Big Data and Internet of Things: A Road Map for Smart Environments*, pp. 169–186, Springer, Berlin, Germany, 2014.

[11] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM MCC Workshop on Mobile Cloud Computing*, pp. 13–16, 2012.

[12] I. Stojmenovic and S. Wen, "The fog computing paradigm: scenarios and security issues," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '14)*, pp. 1–8, IEEE, Warsaw, Poland, September 2014.

[13] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proceedings of the 2nd IEEE International Conference on Future Internet of Things and Cloud (FiCloud '14)*, pp. 464–470, Barcelona, Spain, August 2014.

[14] L. Atzori, A. Iera, and G. Morabito, "SIoT: giving a social structure to the internet of things," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1193–1195, 2011.

[15] M. Swan, "Sensor mania! The internet of things, wearable computing, objective metrics, and the quantified self 2.0," *Journal of Sensor and Actuator Networks*, vol. 1, no. 3, pp. 217–253, 2012.

[16] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[17] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[18] A. Whitmore, A. Agarwal, and L. Da Xu, "The internet of things—a survey of topics and trends," *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[19] D. Zeng, S. Guo, and Z. Cheng, "The web of things: a survey," *Journal of Communications*, vol. 6, no. 6, pp. 424–438, 2011.

[20] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "Role of middleware for internet of things: a study ," *International Journal of Computer Science & Engineering Survey*, vol. 2, no. 3, pp. 94–105, 2011.

[21] J. Soldatos, N. Kefalakis, M. Hauswirth et al., "Openiot: open source internet of-things in the cloud," in *Interoperability and Open-Source Solutions for the Internet of Things: International Workshop, FP7 OpenIoT Project, Held in Conjunction with SoftCOM 2014, Split, Croatia, September 18, 2014, Invited Papers*, vol. 9001 of *Lecture Notes in Computer Science*, pp. 13–25, Springer, Berlin, Germany, 2015.

[22] A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. Campbell, and M. D. Mickunas, "Middlewhere: a middleware for location awareness in ubiquitous computing applications," in *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing Middleware 2004*, pp. 397–416, Springer, New York, NY, USA, 2004.

[23] M. Eisenhauer, P. Rosengren, and P. Antolin, "A development platform for integrating wireless devices and sensors into ambient intelligence systems," in *Proceedings of the 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops (SECON Workshops '09)*, pp. 1–3, IEEE, Rome, Italy, June 2009.

[24] T. Zahariadis, A. Papadakis, F. Alvarez et al., "FIWARE lab: managing resources and services in a cloud federation supporting future internet applications," in *Proceedings of the 7th IEEE/ACM International Conference on Utility and Cloud Computing (UCC '14)*, pp. 792–799, IEEE, London, UK, December 2014.

[25] A. Schmidt and K. Van Laerhoven, "How to build smart appliances?" *IEEE Personal Communications*, vol. 8, no. 4, pp. 66–71, 2001.

[26] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 402–427, 2013.

[27] Accelerometers, Chris Woodford, http://www.explainthatstuff.com/accelerometers.html.

[28] How Do Global Positioning Systems, or GPS, Work?, 2005, https://www.nasa.gov/audience/foreducators/topnav/materials/listbytype/How_Do_Global_Positioning_Systems.html#.VmxoY5Ph5z0.

[29] A. Anjum and M. U. Ilyas, "Activity recognition using smartphone sensors," in *Proceedings of the IEEE 10th Consumer Communications and Networking Conference (CCNC '13)*, pp. 914–919, Las Vegas, Nev, USA, January 2013.

[30] R. Wang, F. Chen, Z. Chen et al., "Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*, pp. 3–14, Seattle, Wash, USA, September 2014.

[31] F. J. McClernon and R. R. Choudhury, "I am your smartphone, and i know you are about to smoke: the application of mobile sensing and computing approaches to smoking research and treatment," *Nicotine and Tobacco Research*, vol. 15, no. 10, pp. 1651–1654, 2013.

[32] L. Pei, R. Guinness, R. Chen et al., "Human behavior cognition using smartphone sensors," *Sensors*, vol. 13, no. 2, pp. 1402–1424, 2013.

[33] N. Bui and M. Zorzi, "Health care applications: a solution based on the internet of things," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL '11)*, ACM, Barcelona, Spain, October 2011.

[34] M. J. McGrath and C. N. Scanaill, "Body-worn, ambient, and consumer sensing for health applications," in *Sensor Technologies*, pp. 181–216, Springer, 2013.

[35] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 40, no. 1, pp. 1–12, 2010.

[36] J. H. Gruzelier, "EEG-neurofeedback for optimising performance. I: a review of cognitive and affective outcome in healthy participants," *Neuroscience and Biobehavioral Reviews*, vol. 44, pp. 124–141, 2014.

[37] P. K. Sekhar, E. L. Brosha, R. Mukundan, and F. H. Garzon, "Chemical sensors for environmental monitoring and homeland security," *The Electrochemical Society Interface*, vol. 19, no. 4, pp. 35–40, 2010.

[38] N. Bhattacharyya and R. Bandhopadhyay, "Electronic nose and electronic tongue," in *Nondestructive Evaluation of Food Quality*, pp. 73–100, Springer, Berlin, Germany, 2010.

[39] S. Manna, S. S. Bhunia, and N. Mukherjee, "Vehicular pollution monitoring using IoT," in *International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2014*, ind, May 2014.

[40] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.

[41] X. Zhu, S. K. Mukhopadhyay, and H. Kurata, "A review of RFID technology and its managerial applications in different industries," *Journal of Engineering and Technology Management*, vol. 29, no. 1, pp. 152–167, 2012.

[42] E. Welbourne, L. Battle, G. Cole et al., "Building the internet of things using RFID: the RFID ecosystem experience," *IEEE Internet Computing*, vol. 13, no. 3, pp. 48–55, 2009.

[43] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: fog computing, cloud computing, and more fog computing," in *Proceedings of the IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD '14)*, pp. 325–329, Athens, Greece, December 2014.

[44] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[45] I. Stojmenovic, "Fog computing: a cloud to the ground support for smart things and machine-to-machine networks," in *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC '14)*, pp. 117–122, Melbourne, Australia, November 2014.

[46] M. Aazam, P. P. Hung, and E.-N. Huh, "Smart gateway based communication for cloud of things," in *Proceedings of the 9th IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (IEEE ISSNIP '14)*, IEEE, April 2014.

[47] P. Agrawal and S. Bhuraria, "Near field communication," *SET-Labs Bridfings*, vol. 10, no. 1, pp. 67–74, 2012.

[48] V. Coskun, B. Ozdenizci, and K. Ok, "A survey on near field communication (NFC) technology," *Wireless Personal Communications*, vol. 71, no. 3, pp. 2259–2294, 2013.

[49] K. Curran, A. Millar, and C. Mc Garvey, "Near Field Communication," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 2, no. 3, 2012.

[50] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities," *IEEE Wireless Communications*, vol. 20, no. 6, pp. 91–98, 2013.

[51] J. P. Vasseur and A. Dunkels, "Ip for smart objects," White Paper 1, IPSO Alliance, 2008.

[52] D. Culler and S. Chakrabarti, "6lowpan: incorporating IEEE 802.15. 4 into the IP architecture, IPSO Alliance," White Paper, 2009.

[53] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand, and C. Chauvenet, "Rpl: the ip routing protocol designed for low power and lossy networks," Internet Protocol for Smart Objects (IPSO) Alliance 36, 2011.

[54] J. P. Vasseur, C. P. Bertrand, B. Aboussouan et al., "A survey of several low power link layers for IP smart objects," White Paper, IPSO Alliance, 2010.

[55] J. W. Hui and D. E. Culler, "Extending IP to low-power, wireless personal area networks," *IEEE Internet Computing*, vol. 12, no. 4, pp. 37–45, 2008.

[56] W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota, "Evaluation of constrained application protocol for wireless sensor networks," in *Proceedings of the 18th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN '11)*, pp. 1–6, IEEE, Chapel Hill, NC, USA, October 2011.

[57] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," Tech. Rep., IETF, 2014.

[58] B. C. Villaverde, D. Pesch, R. De Paz Alberola, S. Fedor, and M. Boubekeur, "Constrained application protocol for low power embedded networks: a survey," in *Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS '12)*, pp. 702–707, Palermo, Italy, July 2012.

[59] D. Locke, "MQ telemetry transport (MQTT) v3. 1 protocol specification," IBM developerWorks Technical Library, 2010, http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html.

[60] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—a publish/subscribe protocol for wireless sensor networks," in *Proceedings of the 3rd IEEE/Create-Net International Conference on Communication System Software and Middleware (COMSWARE '08)*, pp. 791–798, Bangalore, India, January 2008.

[61] A. Stanford-Clark and H. Linh Truon, "MQTT for sensor networks (MQTT-S) protocol specification," International Business Machines Corporation Version 1, 2008.

[62] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: an emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.

[63] K.-H. Chang, "Bluetooth: a viable solution for IoT? [Industry Perspectives]," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 6–7, 2014.

[64] C. F. Hughes, *Bluetooth low energy [Ph.D. thesis]*, Arizona State University, 2015.

[65] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW '12)*, pp. 232–237, Paris, France, April 2012.

[66] B. Shanmuga Sundaram, "A quantitative analysis of 802.11ah wireless standard," *International Journal of Latest Research in Engineering and Technology*, vol. 2, 2016.

[67] W. Sun, M. Choi, and S. Choi, "Ieee 802.11 ah: a long range 802.11 wlan at sub 1 ghz," *Journal of ICT Standardization*, vol. 1, no. 1, pp. 83–108, 2013.

[68] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standards," *Computer Communications*, vol. 30, no. 7, pp. 1655–1695, 2007.

[69] H. Liu, M. Bolic, A. Nayak, and I. Stojmenović, "Taxonomy and challenges of the integration of RFID and wireless sensor networks," *IEEE Network*, vol. 22, no. 6, pp. 26–32, 2008.

[70] A. Mitrokotsa and C. Douligeris, "Integrated RFID and sensor networks: architectures and applications," in *RFID and Sensor Networks: Architectures, Protocols, Security and Integrations*, pp. 511–535, Auerbach Publications, 2009.

[71] M. A. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in *Proceedings of the 13th International Conference on Collaboration Technologies and Systems (CTS '12)*, pp. 21–26, Denver, Colo, USA, May 2012.

[72] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Cla, "Middleware for internet of things: a survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, 2016.

[73] Z. Song, A. A. Cárdenas, and R. Masuoka, "Semantic middleware for the internet of things," in *Proceedings of the 2nd International Internet of Things Conference (IoT '10)*, December 2010.

[74] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan, "Smart semantic middleware for the internet of things," in *Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO '08)*, pp. 169–178, Funchal, Portugal, May 2008.

[75] D. J. Cook, M. Youngblood, E. O. Heierman III et al., "MavHome: an agent-based smart home," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, pp. 521–524, March 2003.

[76] S. K. Das, D. J. Cook, A. Bhattacharya, E. O. Heierman III, and T.-Y. Lin, "The role of prediction algorithms in the MavHome smart home architecture," *IEEE Wireless Communications*, vol. 9, no. 6, pp. 77–84, 2002.

[77] D.-M. Han and J.-H. Lim, "Design and implementation of smart home energy management systems based on ZigBee," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1417–1425, 2010.

[78] N. Noury, T. Hervé, V. Rialle et al., "Monitoring behavior in home using a smart fall sensor and position sensors," in *Proceedings of the 1st Annual International IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology (MMB '00)*, pp. 607–610, Lyon, France, October 2000.

[79] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 42–47, 2004.

[80] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1274–1286, 2012.

[81] W. Keith Edwards and R. E. Grinter, "At home with ubiquitous computing: seven challenges," in *Ubicomp 2001: Ubiquitous Computing*, pp. 256–272, Springer, 2001.

[82] R. J. Robles and T.-H. Kim, "A Review on security in smart home development," *International Journal of Smart Home*, vol. 15, 2010.

[83] R. J. Robles, T.-H. Kim, D. Cook, and S. Das, "A review on security in smart home development," *International Journal of Advanced Science and Technology*, vol. 15, 2010.

[84] G. Dimitrakopoulos, "Intelligent transportation systems based on internet-connected vehicles: fundamental research areas and challenges," in *Proceedings of the 11th International Conference on ITS Telecommunications (ITST '11)*, pp. 145–151, IEEE, Saint Petersburg, Russia, August 2011.

[85] S.-H. Yu, J.-W. Hsieh, Y.-S. Chen, and W.-F. Hu, "An automatic traffic surveillance system for vehicle tracking and classification," in *Image Analysis*, pp. 379–386, Springer, 2003.

[86] M. Lv, L. Chen, G. Chen, and D. Zhang, "Detecting traffic congestions using cell phone accelerometers," in *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*, pp. 107–110, Seattle, Wash, USA, September 2014.

[87] W. Hu, X. Hu, J.-Q. Deng et al., "Mood-fatigue analyzer: towards context-aware mobile sensing applications for safe driving,"

in *Proceedings of the 1st ACM Workshop on Middleware for Context-Aware Applications in the IoT (M4IOT '14)*, pp. 19–24, ACM, Bordeaux, France, December 2014.

[88] H. Singh, J. S. Bhatia, and J. Kaur, "Eye tracking based driver fatigue monitoring and warning system," in *Proceedings of the India International Conference on Power Electronics (IICPE '10)*, pp. 1–6, New Delhi, India, January 2011.

[89] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '12)*, pp. 234–239, Madrid, Spain, June 2012.

[90] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "WreckWatch: automatic traffic accident detection and notification with smartphones," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 285–303, 2011.

[91] G. Hauber-Davidson and E. Idris, "Smart water metering," *Water*, vol. 33, no. 3, pp. 56–59, 2006.

[92] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things," *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1531–1539, 2013.

[93] G. Liang, J. Cao, and W. Zhu, "CircleSense: a pervasive computing system for recognizing social activities," in *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications (PerCom '13)*, pp. 201–206, IEEE, San Diego, Calif, USA, March 2013.

[94] R. W. Picard and R. Picard, *Affective Computing*, vol. 252, MIT Press, Cambridge, UK, 1997.

[95] Y.-K. Row and T.-J. Nam, "CAMY: applying a pet dog analogy to everyday ubicomp products," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*, pp. 63–74, Seattle, Wash, USA, September 2014.

[96] M. Lee and J.-D. Cho, "Logmusic: context-based social music recommendation service on mobile device," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*, pp. 95–98, Seattle, Wash, USA, September 2014.

[97] K. Frank, P. Robertson, M. Gross, and K. Wiesner, "Sensor-based identification of human stress levels," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '13)*, pp. 127–132, San Diego, Calif, USA, March 2013.

[98] M. Sundholm, J. Cheng, B. Zhou, A. Sethi, and P. Lukowicz, "Smart-mat: recognizing and counting gym exercises with low-cost resistive pressure sensing matrix," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*, pp. 373–382, Seattle, Wash, USA, September 2014.

[99] J.-C. Zhao, J.-F. Zhang, Y. Feng, and J.-X. Guo, "The study and application of the IOT technology in agriculture," in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT '10)*, pp. 462–465, Chengdu, China, July 2010.

[100] G. Zhao, Y. Guo, X. Sun, and X. Wang, "A system for pesticide residues detection and agricultural products traceability based on acetylcholinesterase biosensor and internet of things," *International Journal of Electrochemical Science*, vol. 10, no. 4, pp. 3387–3399, 2015.

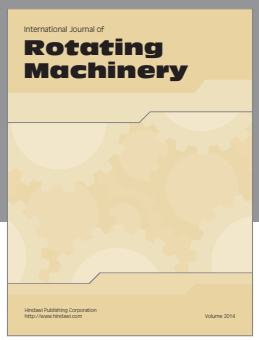
[101] P. Ferreira, R. Martinho, and D. Domingos, "Iot-aware business processes for logistics: limitations of current approaches," in *Proceedings of the Inforum Conference*, vol. 3, pp. 612–613, 2010.

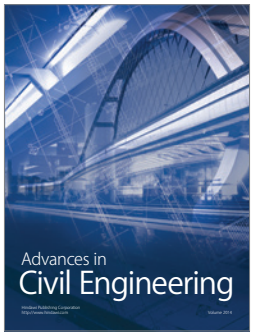
[102] Y. Bo and H. Guangwen, "Supply chain information transmission based on RFID and internet of things," in *Proceedings of the Second ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM '09)*, pp. 166–169, Sanya, China, August 2009.

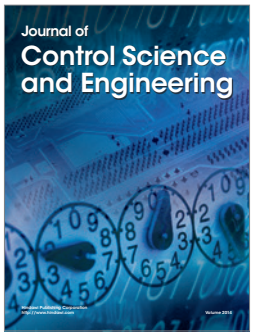
[103] S. Karnouskos, "The cooperative internet of things enabled smart grid," in *Proceedings of the 14th IEEE International Symposium on Consumer Electronics (ISCE '10)*, pp. 7–10, June 2010.

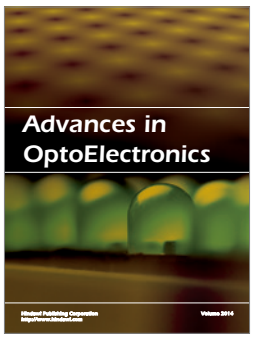
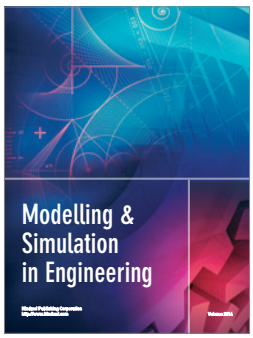
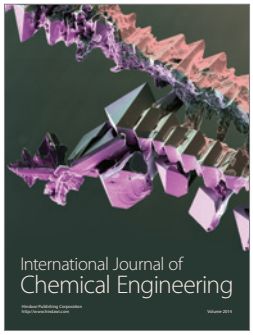
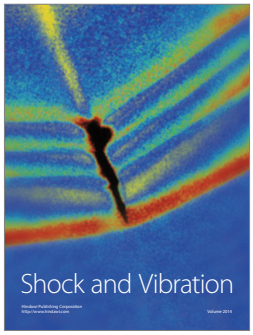
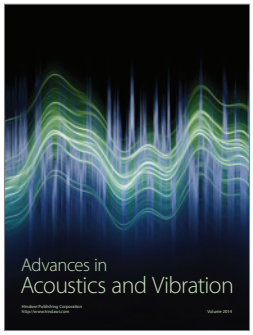
[104] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.

[105] J. Liu, X. Li, X. Chen, Y. Zhen, and L. Zeng, "Applications of internet of things on smart grid in China," in *Proceedings of the 13th International Conference on Advanced Communication Technology: Smart Service Innovation through Mobile Interactivity (ICACT '11)*, pp. 13–17, February 2011.







Hindawi

Submit your manuscripts at
https://www.hindawi.com