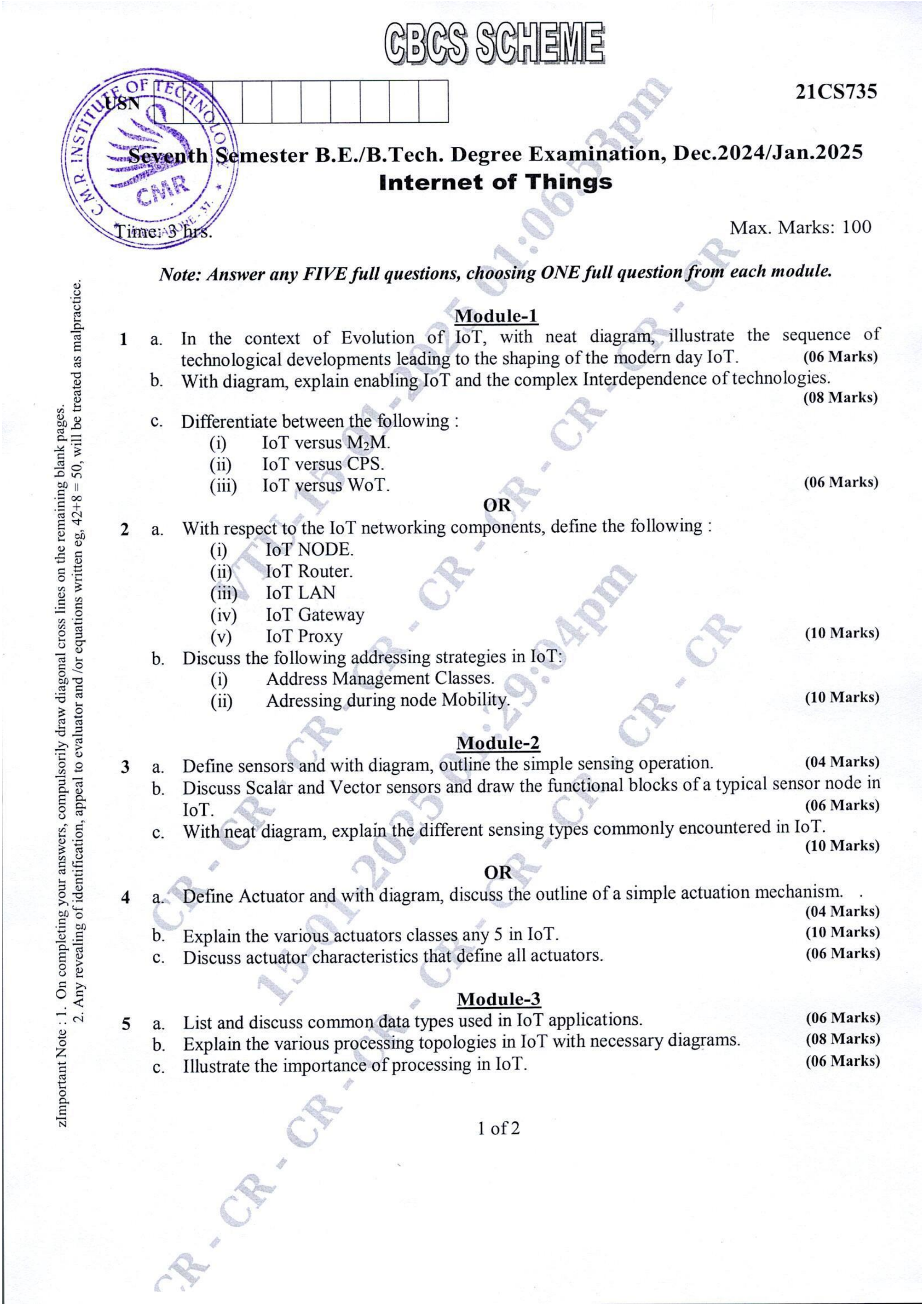


CBCS SCHEME

USN □□□□□□□□□□ **21CS735**

Seventh Semester B.E./B.Tech. Degree Examination, Dec.2024/Jan.2025

Internet of Things

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. In the context of Evolution of IoT, with neat diagram, illustrate the sequence of technological developments leading to the shaping of the modern day IoT. **(06 Marks)**

b. With diagram, explain enabling IoT and the complex Interdependence of technologies. **(08 Marks)**

c. Differentiate between the following :
(i) IoT versus M_2M.
(ii) IoT versus CPS.
(iii) IoT versus WoT. **(06 Marks)**

OR

2 a. With respect to the IoT networking components, define the following :
(i) IoT NODE.
(ii) IoT Router.
(iii) IoT LAN
(iv) IoT Gateway
(v) IoT Proxy **(10 Marks)**

b. Discuss the following addressing strategies in IoT:
(i) Address Management Classes.
(ii) Adressing during node Mobility. **(10 Marks)**

Module-2

3 a. Define sensors and with diagram, outline the simple sensing operation. **(04 Marks)**

b. Discuss Scalar and Vector sensors and draw the functional blocks of a typical sensor node in IoT. **(06 Marks)**

c. With neat diagram, explain the different sensing types commonly encountered in IoT. **(10 Marks)**

OR

4 a. Define Actuator and with diagram, discuss the outline of a simple actuation mechanism. . **(04 Marks)**

b. Explain the various actuators classes any 5 in IoT. **(10 Marks)**

c. Discuss actuator characteristics that define all actuators. **(06 Marks)**

Module-3

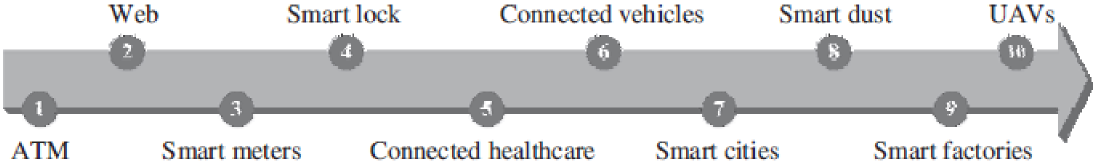
5 a. List and discuss common data types used in IoT applications. **(06 Marks)**

b. Explain the various processing topologies in IoT with necessary diagrams. **(08 Marks)**

c. Illustrate the importance of processing in IoT. **(06 Marks)**

zImportant Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Explain how evolution of IoT take place



- ATM: ATMs or automated teller machines are cash distribution machines, which are linked to a user’s bank account. ATMs dispense cash upon verification of the identity of a user and their account through a specially coded card. The central concept behind ATMs was the availability of financial transactions even when banks were closed beyond their regular work hours. These ATMs were ubiquitous money dispensers. The first ATM became operational and connected online for the first time in 1974.
- Web: World Wide Web is a global information sharing and communication platform. TheWeb became operational for the first time in 1991. Since then, it has been massively responsible for the many revolutions in the field of computing and communication.
- Smart Meters: The earliest smart meter was a power meter, which became operational in early 2000. These power meters were capable of communicating remotely with the power grid. They enabled remote monitoring of subscribers’ power usage and eased the process of billing and power allocation from grids.
- Digital Locks: Digital locks can be considered as one of the earlier attempts at connected home-automation systems. Present-day digital locks are so robust that smartphones can be used to control them. Operations such as locking and unlocking doors, changing key codes, including new members in the access lists, can be easily performed, and that too remotely using smartphones.
- Connected Healthcare: Here, healthcare devices connect to hospitals, doctors, and relatives to alert them of medical emergencies and take preventive measures. The devices may be simple wearable appliances, monitoring just the heart rate and pulse of the wearer, as well as regular medical devices and monitors in hospitals. The connected nature of these systems makes the availability of medical records and test results much faster, cheaper, and convenient for both patients as well as hospital authorities.

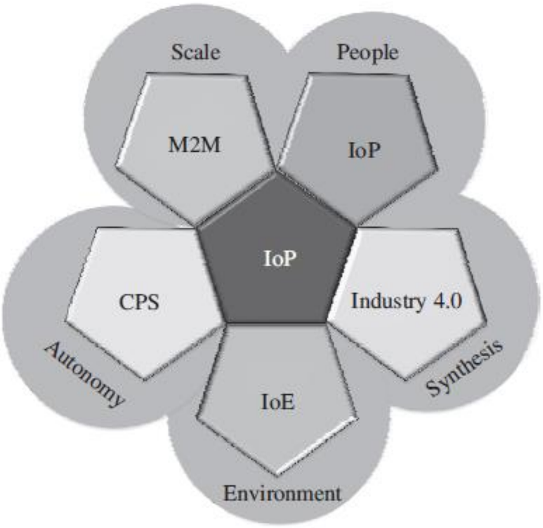
- Connected Vehicles: Connected vehicles may communicate to the Internet or with other vehicles, or even with sensors and actuators contained within it. These vehicles self-diagnose themselves and alert owners about system failures.
- Smart Cities: This is a city-wide implementation of smart sensing, monitoring, and actuation systems. The city-wide infrastructure communicating amongst themselves enables unified and synchronized operations and information dissemination. Some of the facilities which may benefit are parking, transportation, and others.
- Smart Dust: These are microscopic computers. Smaller than a grain of sand each, they can be used in numerous beneficial ways, where regular computers cannot operate. For example, smart dust can be sprayed to measure chemicals in the soil or even to diagnose problems in the human body.
- Smart Factories: These factories can monitor plant processes, assembly lines, distribution lines, and manage factory floors all on their own. The reduction in mishaps due to human errors in judgment or unoptimized processes is drastically reduced.
- UAVs: UAVs or unmanned aerial vehicles have emerged as robust public domain solutions tasked with applications ranging from agriculture, surveys, surveillance, deliveries, stock maintenance, asset management, and other tasks.

Q 1 b)

Figure shows the various technological interdependencies of IoT with other domains and networking paradigms such as M2M, CPS, the Internet of environment (IoE), the Internet of people (IoP), and Industry 4.0. Each of these networking paradigms is a massive domain on its own, but the omnipresent nature of IoT implies that these domains act as subsets of IoT. The paradigms are briefly discussed here:

(i) **M2M**: The M2M or the machine-to-machine paradigm signifies a system of connected machines and devices, which can talk amongst themselves without human intervention. The communication between the machines can be for updates on machine status (stocks, health, power status, and others), collaborative task completion, overall knowledge of the systems and the environment, and others.

(ii) **CPS**: The CPS or the cyber physical system paradigm insinuates a closed control loop—from sensing, processing, and finally to actuation—using a feedback mechanism. CPS helps in maintaining the state of an environment through the feedback control loop, which ensures that until the desired state is attained, the system keeps on actuating and sensing. Humans have a simple supervisory role in CPS-based systems; most of the ground-level operations are automated.



(iii) IoE: The IoE paradigm is mainly concerned with minimizing and even reversing the ill-effects of the permeation of Internet-based technologies on the environment [3]. The major focus areas of this paradigm include smart and sustainable farming, sustainable and energy-efficient habitats, enhancing the energy efficiency of systems and processes, and others. In brief, we can safely assume that any aspect of IoT that concerns and affects the environment, falls under the purview of IoE.

(iv) **Industry 4.0**: Industry 4.0 is commonly referred to as the fourth industrial revolution pertaining to digitization in the manufacturing industry. The previous revolutions chronologically dealt with mechanization, mass production, and the industrial revolution, respectively. This paradigm strongly puts forward the concept of smart factories, where machines talk to one another without much human involvement based on a framework of CPS and IoT. The digitization and connectedness in Industry 4.0 translate to better resource and workforce management, optimization of production time and resources, and better upkeep and lifetimes of industrial systems.

(v) **IoP**: IoP is a new technological movement on the Internet which aims to decentralize online social interactions, payments, transactions, and other tasks while maintaining confidentiality and privacy of its user's data. A famous site for IoP states that as the introduction of the Bitcoin has severely limited the power of banks and governments, the acceptance of IoP will limit the power of corporations, governments, and their spy agencies

Q 1 C) Differentiate between

i) IoT versus M2M

M2M or the machine-to-machine paradigm refers to communications and interactions between various machines and devices. These interactions can be enabled through a cloud computing infrastructure, a server, or simply a local network hub. M2M collects data from machinery and sensors, while also enabling device management and device interaction. Telecommunication services providers introduced the term M2M, and technically emphasized on machine interactions via one or more communication networks (e.g., 3G, 4G, 5G, satellite, public networks). M2M is part of the IoT and is considered as one of its sub-domains, as shown in Figure 4.7. M2M standards occupy a core place in the IoT landscape. However, in terms of operational and functional scope, IoT is vaster than M2M and comprises a broader range of interactions such as the interactions between devices/things, things, and people, things and applications, and people with applications; M2M enables the amalgamation of workflows comprising such interactions within IoT. Internet connectivity is central to the IoT theme but is not necessarily focused on the use of telecom networks.

ii)IoT versus CPS

Cyber physical systems (CPS) encompasses sensing, control, actuation, and feedback as a complete package. In other words, a digital twin is attached to a CPS-based system. As mentioned earlier, a digital twin is a virtual system–model relation, in which the system signifies a physical system or equipment or a piece of machinery, while the model represents the mathematical model or representation of the physical system's behavior or operation. Many a time, a digital twin is used parallel to a physical system, especially in CPS as it allows for the comparison of the physical system's output, performance, and health. Based on feedback from the digital twin, a physical system can be easily given corrective directions/commands to obtain desirable outputs. In contrast, the IoT paradigm does not compulsorily need feedback or a digital twin system. IoT is more focused on networking than controls. Some of the constituent sub-systems in an IoT environment (such as those formed by CPS-based instruments and networks) may include feedback and controls too. In this light, CPS may be considered as one of the sub-domains of IoT

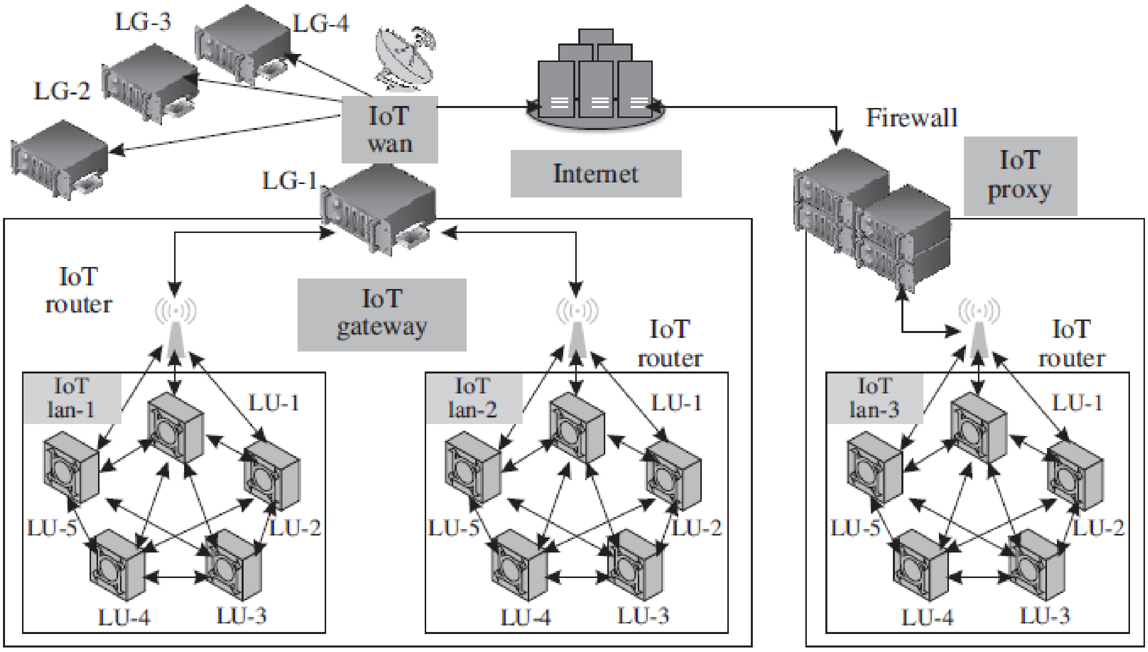
iii) IoT versus WoT

From a developer's perspective, the Web of Things (WoT) paradigm enables access and control over IoT resources and applications. These resources and applications are generally built using technologies such as HTML 5.0, JavaScript, Ajax, PHP, and others. REST (representational state transfer) is one of the key enablers ofWoT. The use of RESTful principles and RESTful APIs (application program interface) enables both developers and deployers to benefit from the recognition, acceptance, and maturity of existing web technologies without having to redesign and redeploy solutions from scratch. Still, designing and building the WoT paradigm has various adaptability and security challenges, especially when trying to build a globally uniform WoT. As IoT is focused on creating networks comprising objects, things, people, systems, and applications, which often do not consider the unification aspect and the limitations of the Internet, the need for WoT, which aims to integrate the various focus areas of IoT into the existing Web is really invaluable. Technically, WoT can be thought of as an application layer-based hat added over the network layer. However, the scope of IoT applications is much broader; IoT also which includes non-IP-based systems that are not accessible through the web.

Q 2 a)

IoT Networking Components

An IoT implementation is composed of several components, which may vary with their application domains. Various established works such as that by Savolainen et al. [2] generally outline five broad categories of IoT networking components. However, we outline the broad components that come into play during the establishment of any IoT network, into six types: 1) IoT node, 2) IoT router, 3) IoT LAN, 4) IoT WAN, 5) IoT gateway, and 6) IoT proxy. A typical IoT implementation from a networking perspective is shown in Figure 4.9. The individual components are briefly described here:



(i) **IoT Node**: These are the networking devices within an IoT LAN. Each of these devices is typically made up of a sensor, a processor, and a radio, which communicates with the network infrastructure (either within the LAN or outside it). The nodes may be connected to other nodes inside a LAN directly or by means of a common gateway for that LAN. Connections outside the LAN are through gateways and proxies.

(ii) **IoT Router**: An I oT router is a piece of networking equipment that is primarily tasked with the routing of packets between various entities in the IoT network; it keeps the traffic flowing correctly within the network. A router can be repurposed as a gateway by enhancing its functionalities.

(iii) **IoT LAN**: The local area network (LAN) enables local connectivity within the purview of a single gateway. Typically, they consist of short-range connectivity technologies. IoT LANs may or may not be connected to the Internet. Generally, they are localized within a building or an organization.

(iv) **IoT WAN**: The wide area network (WAN) connects various network segments such as LANs. They are typically organizationally and geographically wide, with their operational range lying between a few kilometers to hundreds of kilometers. IoT WANs connect to the Internet and enable Internet access to the segments they are connecting.

(v) **IoT Gateway**: An IoT gateway is simply a router connecting the IoT LAN to a WAN or the Internet. Gateways can implement several LANs and WANs. Their primary task is to forward packets between LANs and WANs, and the IP layer using only layer 3.

(vi) **IoT Proxy**: Proxies actively lie on the application layer and performs application layer functions between IoT nodes and other entities. Typically, application layer proxies are a means of providing security to the network entities under it ; it helps to extend the addressing range of its network.

Address management classes

As discussed previously, the IoT deployment and network topology are largely dependent on where it is deployed. Unlike traditional IPv4 networked devices, the newer IoT devices largely depend on IPv6 for address allocation and management of addresses, which again is dictated by the application and the place of deployment of the IoT solution. Keeping these requirements in consideration, the addressing strategies in IoT may be broadly differentiated into seven classes, as shown in Figure 4.11. These classes are as follows:

(i) **Class 1**: The IoT nodes are not connected to any other interface or the Internet except with themselves. This class can be considered as an isolated class, where the communication between IoT nodes is restricted within a LAN only. The IoT nodes in this class are identified only by their link local (LL) addresses, as shown in Figure 4.11(a). These LL addresses may be repeated for other devices outside the purview of this network class. The communication among the nodes may be direct or through other nodes (as in a mesh configuration).

(ii) **Class 2**: The class 1 configuration is mainly utilized for enabling communication between two or more IoT LANs or WANs. The IoT nodes within the LANs cannot directly communicate to nodes in the other LANs using their LL addresses, but through their LAN gateways (which have a unique address assigned to them). Generally, ULA is used for addressing; however, in certain scenarios, GUA may also be used. Figure 4.11(b) shows a class 2 IoT network topology. L1–L5 are the LL addresses of the locally unique IoT nodes within the LAN; whereas U1 and U2 are the unique addresses of the two gateways extending communication to their LANs with the WAN. The WAN may or may not connect to the Internet.

(iii) **Class 3**: Figure 4.11(c) shows a class 3 IoT network configuration, where the IoT LAN is connected to an IoT proxy. The proxy performs a host of functions ranging from address allocation, address management to providing security to the network underneath it. In this class, the IoT proxy only uses ULA (denoted as Lx-Ux in the figure).

(iv) **Class 4**: In this class, the IoT proxy acts as a gateway between the LAN and the Internet, and provides GUA to the IoT nodes within the LAN. A globally unique prefix is allotted to this gateway, which it uses with the individual device identifiers to extend global Internet connectivity to the IoT nodes themselves. This configuration is shown in Figure 4.11(d). An important point to note in this class is that the gateway also enables local communication between the nodes without the need for the packets to be routed through the Internet. Additionally, the IoT nodes within the gateway can talk to one another directly without always involving the gateway. A proxy beyond the gateway enables global communication through the Internet.

(v) **Class 5**: This class is functionally similar to class 4. However, the main difference with class 4 is that this class follows a star topology with the gateway as the center of the star. All the communication from the IoT nodes under the gateway has to go through the gateway, as shown in Figure 4.11(e). A proxy beyond the gateway enables global communication through the Internet. The IoT nodes within a gateway's operational purview have the same GUA.

(vi) **Class 6**: The configuration of this class is again similar to class 5. However, the IoT nodes are all assigned unique global addresses (GUA), which enables a point-to-point communication network with an Internet gateway. A class 6 IoT network configuration is shown in Figure 4.11(f). Typically, this class is very selectively used for special purposes.

(vii) **Class 7**: The class 7 configuration is shown in Figure 4.11(g). Multiple gateways may be present; the configuration is such that the nodes should be reachable through any of the gateways. Typically, organizational IoT deployments follow this class of configuration. The concept of multihoming is important and inherent to this class.

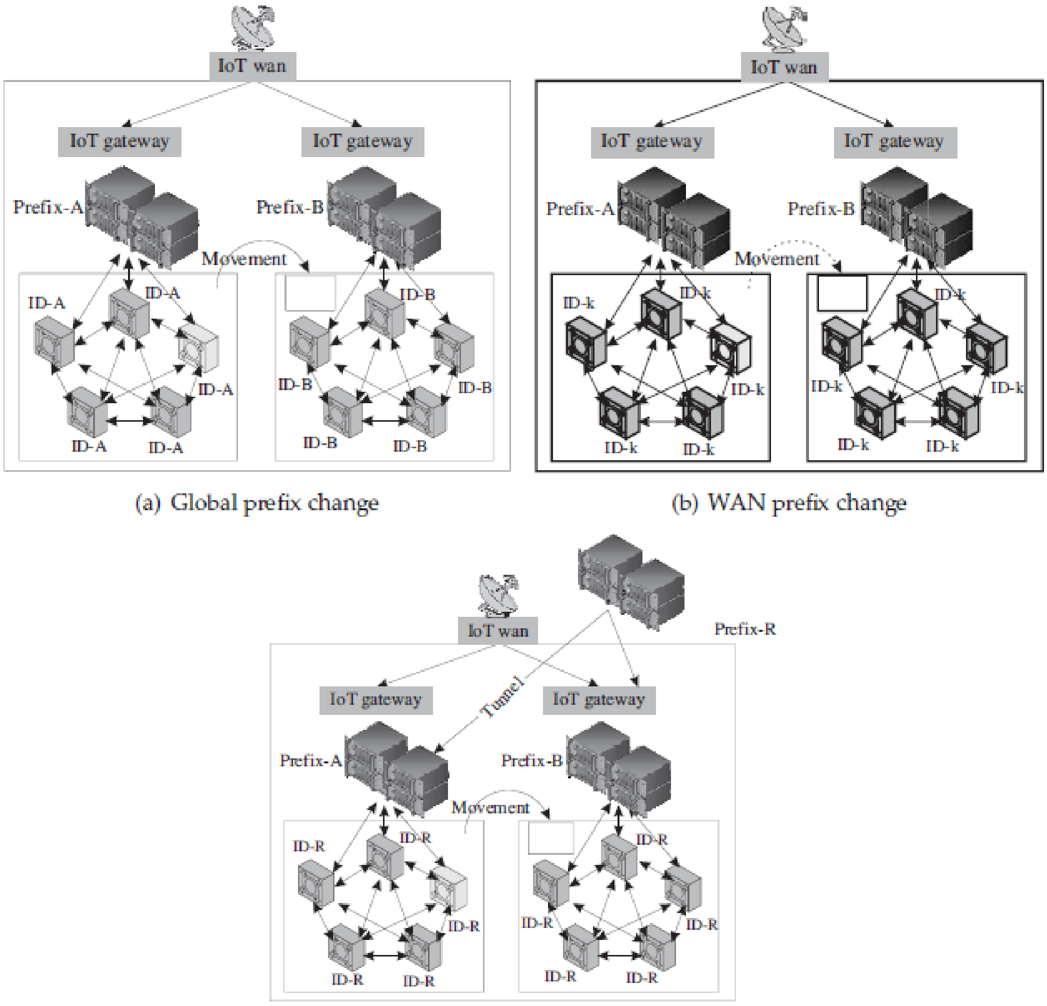
Addressing during node mobility

Traditional networks, mainly computer networks, and even paradigms such as M2M and CPS seldom take into account the need for addressing strategies when the IoT nodes are mobile. However, in a realistic scenario, especially in modernday IoT systems (which are low-power and have low form-factor), the need for addressing of mobile nodes is extremely crucial to avoid address clashes of addresses accommodating a large number of IoT nodes. One of the following three strategies may be to for ensure portability of addresses in the event of node mobility in IoT deployments [2] as shown in Figure 4.12:

(i) **Global Prefix Changes**: Figure 4.12(a) abstracts the addressing strategy using global prefix changes. A node from the left LAN moves to the LAN on the right. The node undergoing movement is highlighted in the figure. The nodes in the first LAN have the prefix **A**, which changes to **B** under the domain of the new gateway overseeing the operation of nodes in the new LAN. However, it may happen that due to movement, the device identifier may face clashes. Recall the structure of the IPv6 address (Figure 4.10). The device identifier, if allotted randomly, might face an address clash upon the node's arrival into the new LAN as there may already be a similar node identifier present in it. Typically, addresses are assigned using DHCPv6/ SLAAC; however, in this scenario, it is always prudent to have static node IP addresses to avoid a clash of addresses. This strategy is, in most cases, beneficial as the IoT nodes may be resource constrained and have low-processing resources due to which it may not be able to handle protocols such as DHCPv6 or SLAAC.

(ii) **Prefix Changes within WANs**: Figure 4.12(b) abstracts the addressing strategy for prefix changes within WANs. In case the WAN changes its global prefix, the network entities underneath it must be resilient to change and function normally. The address allocation is hence delegated to entities such as gateways and proxies, which make use of ULAs to manage the network within the WAN.

(iii) **Remote Anchoring**: Figure 4.12(c) abstracts the addressing strategy using a remote anchoring point. This is applicable in certain cases which require that the IoT node's global addresses are maintained and not affected by its mobility or even the change in network prefixes. Although a bit expensive to implement, this strategy of having a remote anchoring point from which the IoT nodes obtain their global addresses through tunneling ensures that the nodes are resilient to changes and are quite stable. Even if the node's original network's (LAN) prefix changes from **A** to **B**, the node's global address remains immune to this change.



(a) Global prefix change

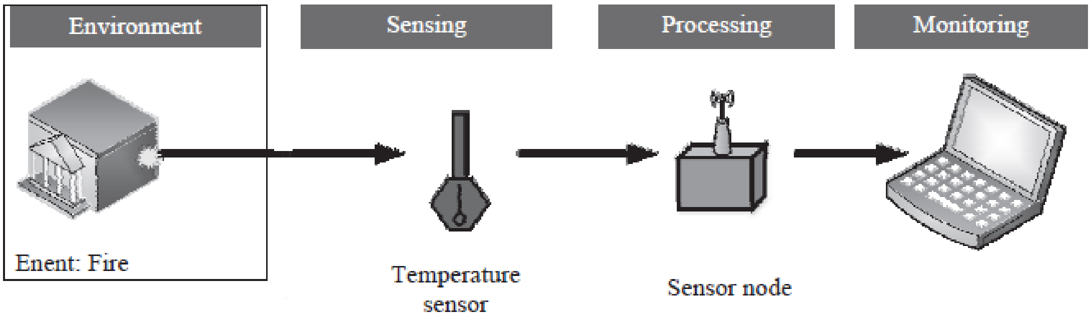
(b) WAN prefix change

(c) Remote anchoring

Q 3 a)

Sensors

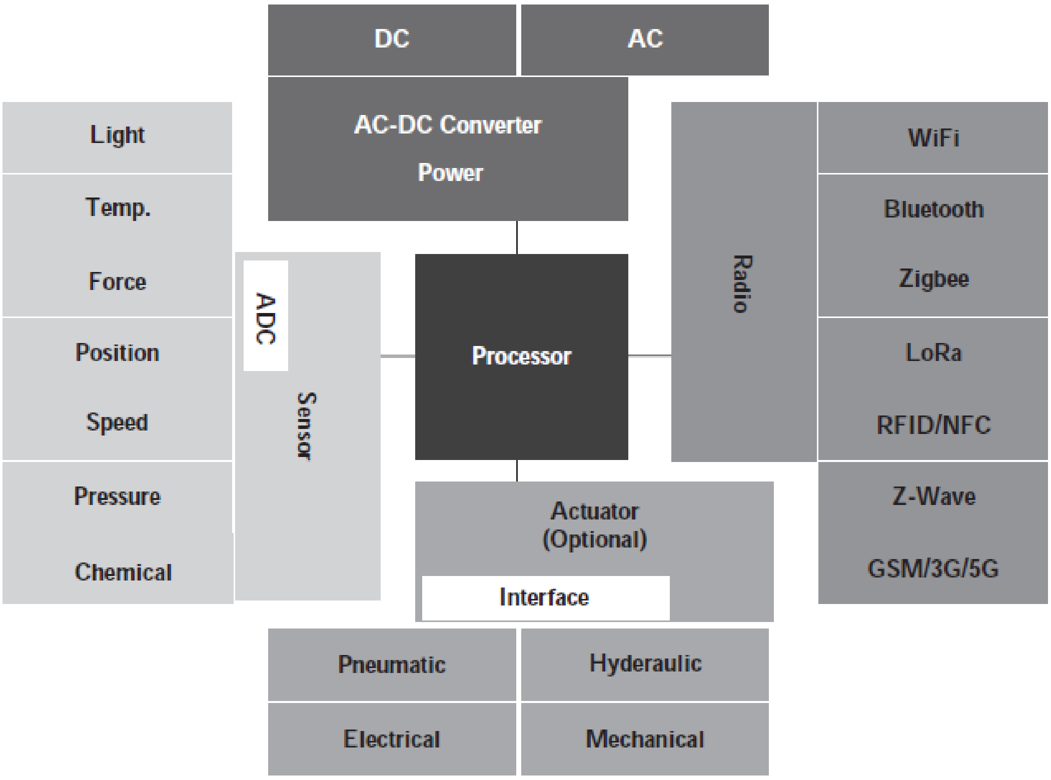
Sensors are devices that can measure, or quantify, or respond to the ambient changes in their environment or within the intended zone of their deployment. They generate responses to external stimuli or physical phenomenon through characterization of the input functions (which are these external stimuli) and their conversion into typically electrical signals. For example, heat is converted to electrical signals in a temperature sensor, or atmospheric pressure is converted to electrical signals in a barometer.



sensor is only sensitive to the measured property (e.g., a temperature sensor only senses the ambient temperature of a room). It is insensitive to any other property besides what it is designed to detect (e.g., a temperature sensor does not bother about light or pressure while sensing the temperature). Finally, a sensor does not influence the measured property (e.g., measuring the temperature does not reduce or increase the temperature). Figure 5.1 shows the simple outline of a sensing task. Here, a temperature sensor keeps on checking an environment for changes. In the event of a fire, the temperature of the environment goes up. The temperature sensor notices this change in the temperature of the room and promptly communicates this information to a remote monitor via the processor.

Q 3 b)

Measured Property: The property of the environment being measured by the sensors can be crucial in deciding the number of sensors in an IoT implementation. Some properties to be measured do not show high spatial variations and can be quantified only based on temporal variations in the measured property, such as ambient temperature, atmospheric pressure, and others. Whereas some properties to be measured show high spatial as well as temporal variations such as sound, image, and others. Depending on the properties to be measured, sensors can be of two types.
(i) Scalar: Scalar sensors produce an output proportional to the magnitude of the quantity being measured. The output is in the form of a signal or voltage. Scalar physical quantities are those where only the magnitude of the signal is sufficient for describing or characterizing the phenomenon and information generation. Examples of such measurable physical quantities include color, pressure, temperature, strain, and others. A thermometer or thermocouple is an example of a scalar sensor that has the ability to detect changes in ambient or object temperatures (depending on the sensor's configuration). Factors such as changes in sensor orientation or direction do not affect these sensors (typically).
(ii) Vector: Vector sensors are affected by the magnitude as well as the direction and/or orientation of the property they are measuring. Physical quantities such as velocity and images that require additional information besides their magnitude for completely categorizing a physical phenomenon are
categorized as vector quantities. Measuring such quantities are undertaken using vector sensors. For example, an electronic gyroscope, which is commonly found in all modern aircraft, is used for detecting the changes in orientation of the gyroscope with respect to the Earth's orientation along all three axes.



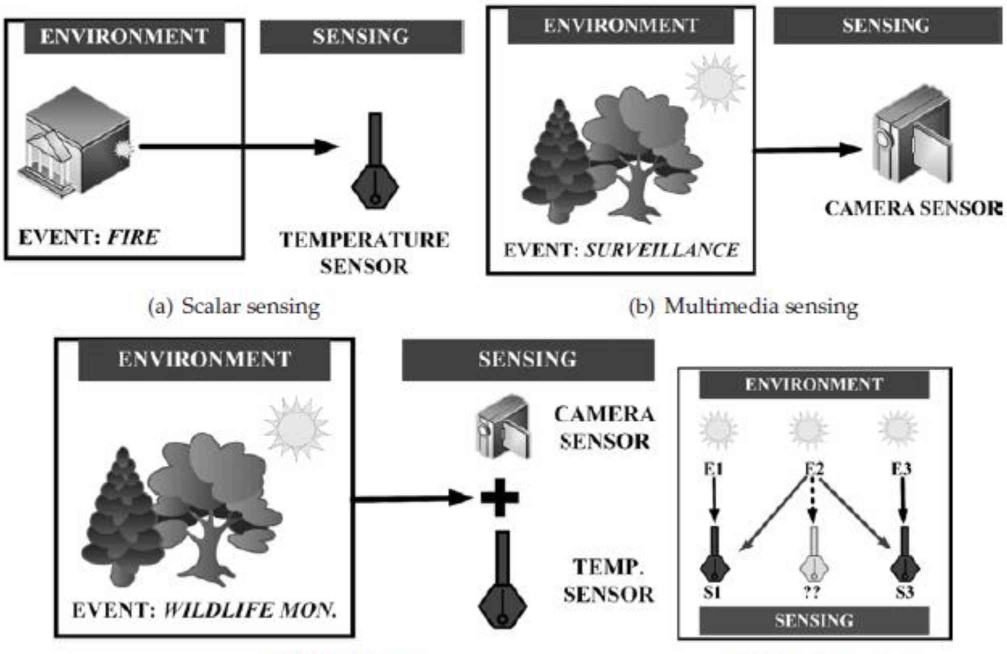
Q 3 C)

Sensing Types

Sensing can be broadly divided into four different categories based on the nature of the environment being sensed and the physical sensors being used to do so (Figure 5.4): 1) scalar sensing, 2) multimedia sensing, 3) hybrid sensing, and 4) virtual sensing—[2].

5.5.1 Scalar sensing

Scalar sensing encompasses the sensing of features that can be quantified simply by measuring changes in the amplitude of the measured values with respect to time [3]. Quantities such as ambient temperature, current, atmospheric pressure, rainfall, light, humidity, flux, and others are considered as scalar values as they normally do not have a directional or spatial property assigned with them. Simply measuring the changes in their values with passing time provides enough information about these quantities. The sensors used for measuring these scalar quantities are referred to as scalar sensors, and the act is known as scalar sensing. Figures 5.3(b), 5.3(d), 5.3(e), 5.3(f), 5.3(g), 5.3(h), 5.3(i), and 5.3(j) show scalar sensors. A simple scalar temperature sensing of a fire detection event is shown in Figure 5.4(a).



(a) Scalar sensing

(b) Multimedia sensing

(c) Hybrid sensing

(d) Virtual sensing

Multimedia sensing

Multimedia sensing encompasses the sensing of features that have a spatial variance property associated with the property of temporal variance [4]. Unlike scalar sensors, multimedia sensors are used for capturing the changes in amplitude of a quantifiable property concerning space (spatial) as well as time (temporal). Quantities such as images, direction, flow, speed, acceleration, sound, force, mass, energy, and momentum have both directions as well as a magnitude. Additionally, these quantities follow the vector law of addition and hence are designated as vector quantities. They might have different values in different directions for the same working condition at the same time. The sensors used for measuring these quantities are known as vector sensors. Figures 5.3(a) and 5.3(c) are vector sensors. A simple camera-based multimedia sensing using surveillance as an example is shown in Figure 5.4(b).

5.5.3 Hybrid sensing

The act of using scalar as well as multimedia sensing at the same time is referred to as hybrid sensing. Many a time, there is a need to measure certain vector as well as scalar properties of an environment at the same time. Under these conditions, a range of various sensors are employed (from the collection of scalar as well as multimedia sensors) to measure the various properties of that environment at any instant of time, and temporally map the collected information to generate new information.

For example, in an agricultural field, it is required to measure the soil conditions at regular intervals of time to determine plant health. Sensors such as soil moisture and soil temperature are deployed underground to estimate the soil's water retention capacity and the moisture being held by the soil at any instant of time. However, this setup only determines whether the plant is getting enough water or not. There may be a host of other factors besides water availability, which may affect a plant's health. The additional inclusion of a camera sensor with the plant may be able to determine the actual condition of a plant by additionally determining the color of leaves. The aggregate information from soil moisture, soil temperature, and the camera sensor will be able to collectively determine a plant's health at any instant of time. Other common examples of hybrid sensing include smart parking systems, traffic management systems, and others. Figure 5.4(c) shows an example of hybrid sensing, where a camera and a temperature sensor are collectively used to detect and confirm forest fires during wildlife monitoring.

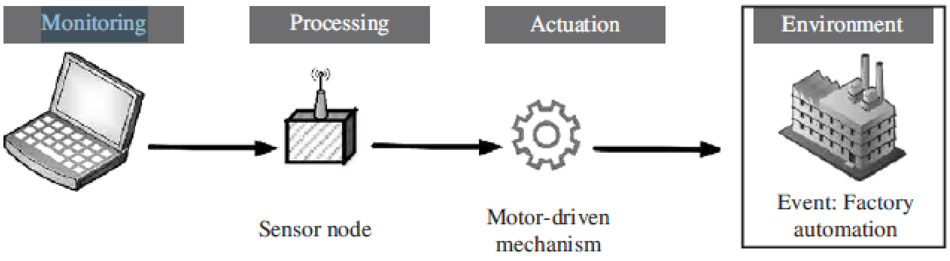
5.5.4 Virtual sensing

Many a time, there is a need for very dense and large-scale deployment of sensor nodes spread over a large area for monitoring of parameters. One such domain is agriculture [5]. Here, often, the parameters being measured, such as soil moisture, soil temperature, and water level, do not show significant spatial variations. Hence, if sensors are deployed in the fields of farmer **A**, it is highly likely that the measurements from his sensors will be able to provide almost concise measurements of his neighbor **B**'s fields; this is especially true of fields which are immediately surrounding **A**'s fields. Exploiting this property, if the data from **A**'s field is digitized using an IoT infrastructure and this system advises him regarding the appropriate watering, fertilizer, and pesticide regimen for his crops, this advisory can also be used by **B** for maintaining his crops. In short, **A** 's sensors are being used for actual measurement of parameters; whereas virtual data (which does not have actual physical sensors but uses extrapolation-based measurements) is being used for advising **B**. This is the virtual sensing paradigm. Figure 5.4(d) shows an example of virtual sensing. Two temperature sensors S1 and S3 monitor three nearby events E1, E2, and E3 (fires). The event E2 does not have a dedicated sensor for monitoring it; however, through the superposition of readings from sensors S1 and S3, the presence of fire in E2 is inferred.

Q 4 a)

Actuators

An actuator can be considered as a machine or system's component that can affect the movement or control the said mechanism or the system. Control systems affect changes to the environment or property they are controlling through actuators. The system activates the actuator through a control signal, which may be digital or analog. It elicits a response from the actuator, which is in the form of some form of mechanical motion. The control system of an actuator can be a mechanical or electronic system, a software-based system (e.g., an autonomous car control system), a human, or any other input. Figure 5.5 shows the outline of a simple actuation system. A remote user sends commands to a processor. The processor instructs a motor controlled robotic arm to perform the commanded tasks accordingly. The processor is primarily responsible for converting the human commands into sequential machine-language command sequences, which enables the robot to move. The robotic arm finally moves the designated boxes, which was its assigned task.



Q 4 b)

Actuator Types

Broadly, actuators can be divided into seven classes: 1) Hydraulic, 2) pneumatic, 3) electrical, 4) thermal/magnetic, 5) mechanical, 6) soft, and 7) shape memory polymers. Figure 5.6 shows some of the commonly used actuators in IoT applications.

5.8.1 Hydraulic actuators

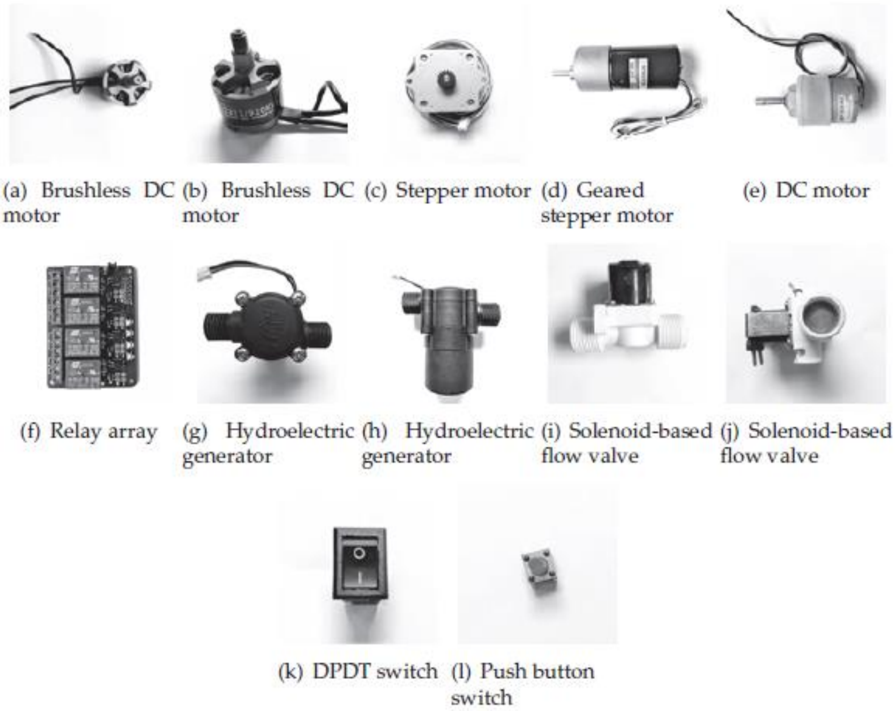
A hydraulic actuator works on the principle of compression and decompression of fluids. These actuators facilitate mechanical tasks such as lifting loads through the use of hydraulic power derived from fluids in cylinders or fluid motors. The mechanical motion applied to a hydraulic actuator is converted to either linear, rotary, or oscillatory motion. The almost incompressible property of liquids is used in hydraulic actuators for exerting significant force. These hydraulic actuators are also considered as stiff systems. The actuator's limited acceleration restricts its usage.

5.8.2 Pneumatic actuators

A pneumatic actuator works on the principle of compression and decompression of gases. These actuators use a vacuum or compressed air at high pressure and convert it into either linear or rotary motion. Pneumatic rack and pinion actuators are commonly used for valve controls of water pipes. Pneumatic actuators are considered as compliant systems. The actuators using pneumatic energy for their operation are typically characterized by the quick response to starting and stopping signals. Small pressure changes can be used for generating large forces through these actuators. Pneumatic brakes are an example of this type of actuator which is so responsive that they can convert small pressure changes applied by drives to generate the massive force required to stop or slow down a moving vehicle. Pneumatic actuators are responsible for converting pressure into force. The power source in the pneumatic actuator does not need to be stored in reserve for its operation.

5.8.3 Electric actuators

Typically, electric motors are used to power an electric actuator by generating mechanical torque. This generated torque is translated into the motion of a motor's shaft or for switching (as in relays). For example, actuating equipments such as solenoid valves control the flow of water in pipes in response to electrical signals. This class of actuators is considered one of the cheapest, cleanest and speedy actuator types available. Figures 5.6(a), 5.6(b), 5.6(c), 5.6(d), 5.6(e), 5.6(f), 5.6(i), and 5.6(j) show some of the commonly used electrical actuators.



(a) Brushless DC motor (b) Brushless DC motor (c) Stepper motor (d) Geared stepper motor (e) DC motor

(f) Relay array (g) Hydroelectric generator (h) Hydroelectric generator (i) Solenoid-based flow valve (j) Solenoid-based flow valve

(k) DPDT switch (l) Push button switch

Thermal or magnetic actuators

The use of thermal or magnetic energy is used for powering this class of actuators. These actuators have a very high power density and are typically compact, lightweight, and economical. One classic example of thermal actuators is shape memory materials (SMMs) such as shape memory alloys (SMAs). These actuators do not require electricity for actuation. They are not affected by vibration and can work with liquid or gases. Magnetic shape memory alloys (MSMAs) are a type of magnetic actuators.

5.8.5 Mechanical actuators

In mechanical actuation, the rotary motion of the actuator is converted into linear motion to execute some movement. The use of gears, rails, pulleys, chains, and other devices are necessary for these actuators to operate. These actuators can be easily used in conjunction with pneumatic, hydraulic, or electrical actuators. They can also work in a standalone mode. The best example of a mechanical actuator is a rack and pinion mechanism. Figures 5.6(g), 5.6(h), 5.6(k), and 5.6(l) show some of the commonly available mechanical actuators.

Q 4 c)

Actuator Characteristics

The choice or selection of actuators is crucial in an IoT deployment, where a control mechanism is required after sensing and processing of the information obtained from the sensed environment. Actuators perform the physically heavier tasks in an IoT deployment; tasks which require moving or changing the orientation of physical objects, changing the state of objects, and other such activities. The correct choice of actuators is necessary for the long-term sustenance and continuity of operations, as well as for increasing the lifetime of the actuators themselves. A set of four characteristics can define all actuators:

- **Weight**: The physical weight of actuators limits its application scope. For example, the use of heavier actuators is generally preferred for industrial applications and applications requiring no mobility of the IoT deployment. In contrast, lightweight actuators typically find common usage in portable systems in vehicles, drones, and home IoT applications. It is to be noted that this is not always true. Heavier actuators also have selective usage in mobile systems, for example, landing gears and engine motors in aircraft.
- **Power Rating**: This helps in deciding the nature of the application with which an actuator can be associated. The power rating defines the minimum and maximum operating power an actuator can safely withstand without damage to itself. Generally, it is indicated as the power-to-weight ratio for actuators. For example, smaller servo motors used in hobby projects typically have a maximum rating of 5 VDC, 500 mA, which is suitable for an operations-driven battery-based power source. Exceeding this limit might be detrimental to the performance of the actuator and may cause burnout of the motor. In contrast to this, servo motors in larger applications have a rating of 460 VAC, 2:5 A, which requires standalone power supply systems for operations. It is to be noted that actuators with still higher ratings are available and vary according to application requirements.
- **Torque to Weight Ratio**: The ratio of torque to the weight of the moving part of an instrument/device is referred to as its torque/weight ratio. This indicates the sensitivity of the actuator. Higher is the weight of the moving part; lower will be its torque to weight ratio for a given power.

- **Stiffness and Compliance**: The resistance of a material against deformation is known as its stiffness, whereas compliance of a material is the opposite of stiffness. Stiffness can be directly related to the modulus of elasticity of that material. Stiff systems are considered more accurate than compliant systems as they have a faster response to the change in load applied to it. For example, hydraulic systems are considered as stiff and non-compliant, whereas pneumatic systems are considered as compliant.

Q5 a)

1 Structured data

These are typically text data that have a pre-defined structure [1]. Structured data are associated with relational database management systems (RDBMS). These are primarily created by using length-limited data fields such as phone numbers, social security numbers, and other such information. Even if the data is human or machinegenerated, these data are easily searchable by querying algorithms as well as humangenerated queries. Common usage of this type of data is associated with flight or train reservation systems, banking systems, inventory controls, and other similar systems. Established languages such as Structured Query Language (SQL) are used for accessing these data in RDBMS. However, in the context of IoT, structured data holds a minor share of the total generated data over the Internet.

2 Unstructured data

In simple words, all the data on the Internet, which is not structured, is categorized as unstructured. These data types have no pre-defined structure and can vary according to applications and data-generating sources. Some of the common examples of human-generated unstructured data include text, e-mails, videos, images, phone recordings, chats, and others [2]. Some common examples of machine-generated unstructured data include sensor data from traffic, buildings, industries, satellite imagery, surveillance videos, and others. As already evident from its examples, this data type does not have fixed formats associated with it, which makes it very difficult for querying algorithms to perform a look-up. Querying languages such as NoSQL are generally used for this data type.

Q 5 b)

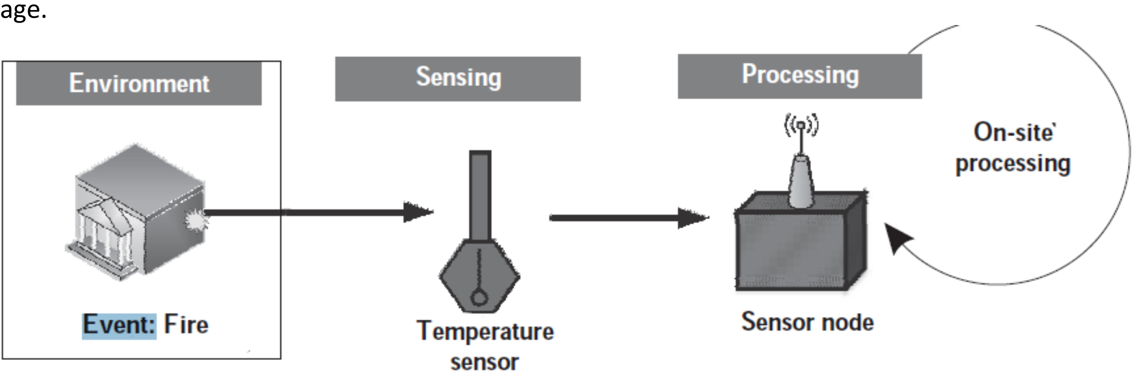
Processing Topologies

The identification and intelligent selection of processing requirement of an IoT application are one of the crucial steps in deciding the architecture of the deployment. A properly designed IoT architecture would result in massive savings in network bandwidth and conserve significant amounts of overall energy in the architecture while providing the proper and allowable processing latencies for the solutions associated with the architecture. Regarding the importance of processing in IoT as outlined in Section 6.2, we can divide the various processing solutions into two large topologies: 1) On-site and 2) Off-site. The off-site processing topology can be further divided into the following: 1) Remote processing and 2) Collaborative processing.

6.3.1 On-site processing

As evident from the name, the on-site processing topology signifies that the data is processed at the source itself. This is crucial in applications that have a very low tolerance for latencies. These latencies may result from the processing hardware or the network (during transmission of the data for processing away from the processor). Applications such as those associated with healthcare and flight control systems (realtime systems) have a breakneck data generation rate. These additionally show rapid temporal changes that can be missed (leading to catastrophic damages) unless the processing infrastructure is fast and robust enough to handle such data. Figure 6.2

shows the on-site processing topology, where an event (here, fire) is detected utilizing a temperature sensor connected to a sensor node. The sensor node processes the information from the sensed event and generates an alert. The node additionally has the option of forwarding the data to a remote infrastructure for further analysis and storage.



ON-site processing

The off-site processing paradigm, as opposed to the on-site processing paradigms, allows for latencies (due to processing or network latencies); it is significantly cheaper than on-site processing topologies. This difference in cost is mainly due to the low demands and requirements of processing at the source itself. Often, the sensor nodes are not required to process data on an urgent basis, so having a dedicated and expensive on-site processing infrastructure is not sustainable for large-scale deployments typical of IoT deployments. In the off-site processing topology, the sensor node is responsible for the collection and framing of data that is eventually to be transmitted to another location for processing

Q 5 c)

Importance of Processing in IoT

The vast amount and types of data flowing through the Internet necessitate the need for intelligent and resourceful processing techniques. This necessity has become even more crucial with the rapid advancements in IoT, which is laying enormous pressure on the existing network infrastructure globally. Given these urgencies, it is important to decide—when to process and what to process? Before deciding upon the processing to pursue, we first divide the data to be processed into three types based on the urgency of processing: 1) Very time critical, 2) time critical, and 3) normal. Data from sources such as flight control systems [3], healthcare, and other such sources, which need immediate decision support, are deemed as very critical. These data have a very low threshold of processing latency, typically in the range of a few milliseconds.
Data from sources that can tolerate normal processing latency are deemed as timecritical data. These data, generally associated with sources such as vehicles, traffic, machine systems, smart home systems, surveillance systems, and others, which can tolerate a latency of a few seconds fall in this category. Finally, the last category of data, normal data,can tolerate a processing latency of a few minutes to a few hours and are typically associated with less data-sensitive domains such as agriculture, environmental monitoring, and others.
Considering the requirements of data processing, the processing requirements of data from very time-critical sources are exceptionally high. Here, the need for processing the data in place or almost nearer to the source is crucial in achieving the deployment success of such domains. Similarly, considering the requirements of processing from category 2 data sources (time-critical), the processing requirements allow for the transmission of data to be processed to remote locations/processors such

as clouds or through collaborative processing. Finally, the last category of data sources (normal) typically have no particular time requirements for processing urgently and are pursued leisurely as such.

Q 6 a) with neat diagram, explain the processing offloading paradigm for the development of IoT based solutions

Answer:

The processing offloading paradigm is important for the development of densely deployable, energy-conserving, miniaturized, and cheap IoT-based solutions for sensing tasks. Building upon the basics of the off-site processing topology covered in the previous sections in this chapter, we delve a bit further into the various nuances of processing offloading in IoT. This section on data offloading is divided into three parts: 1) offload location (which outlines where all the processing can be offloaded in the IoT architecture), 2) offload decision making (how to choose where to offload the processing to and by how much), and finally 3) offloading considerations (deciding when to offload).

Offload location

The choice of offload location decides the applicability, cost, and sustainability of the IoT application and deployment. We distinguish the offload location into four types:

- **Edge**: Offloading processing to the edge implies that the data processing is facilitated to a location at or near the source of data generation itself. Offloading to the edge is done to achieve aggregation, manipulation, bandwidth reduction, and other data operations directly on an IoT device [7].
- **Fog**: Fog computing is a decentralized computing infrastructure that is utilized to conserve network bandwidth, reduce latencies, restrict the amount of data unnecessarily flowing through the Internet, and enable rapid mobility support for IoT devices. The data, computing, storage and applications are shifted to a place between the data source and the cloud resulting in significantly reduced latencies and network bandwidth usage [8].
- **Remote Server**: A simple remote server with good processing power may be used with IoT-based applications to offload the processing from resourceconstrained IoT devices. Rapid scalability may be an issue with remote servers, and they may be costlier and hard to maintain in comparison to solutions such as the cloud [4].
- **Cloud**: Cloud computing is a configurable computer system, which can get access to configurable resources, platforms, and high-level services through a shared pool hosted remotely.

Offload decision making

The choice of where to offload and how much to offload is one of the major deciding factors in the deployment of an offsite-processing topology-based IoT deployment architecture. The decision making is generally addressed considering data generation rate, network bandwidth, the criticality of applications, processing resource available at the offload site, and other factors. Some of these approaches are **Naive Approach, Bargaining based approach, Learning based approach**

Offloading considerations

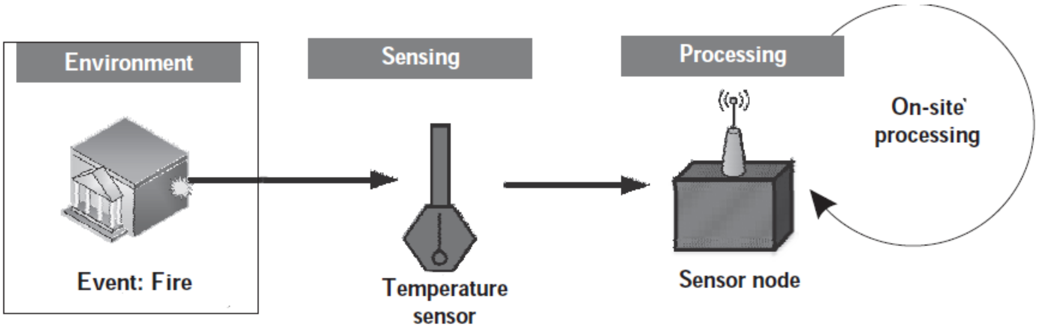
There are a few offloading parameters which need to be considered while deciding upon the offloading type to choose. These considerations typically arise from the nature of the IoT application and the hardware being used to interact with the application. Some of these parameters are as follows **Bandwidth, Latency, Criticality, Resources, Data Volume**

Q 6 b) Determine the importance of choosing the right processing topologies and associated considerations while designing IoT applications

Answer: The identification and intelligent selection of processing requirement of an IoT application are one of the crucial steps in deciding the architecture of the deployment. A properly designed IoT architecture would result in massive savings in network bandwidth and conserve significant amounts of overall energy in the architecture while providing the proper and allowable processing latencies for the solutions associated with the architecture. Regarding the importance of processing in IoT as outlined in Section 6.2, we can divide the various processing solutions into two large topologies: 1) On-site and 2) Off-site. The off-site processing topology can be further divided into the following: 1) Remote processing and 2) Collaborative processing

On-site processing

Figure shows the on-site processing topology, where an event (here, fire) is detected utilizing a temperature sensor connected to a sensor node. The sensor node processes the information from the sensed event and generates an alert. The node additionally has the option of forwarding the data to a remote infrastructure for further analysis and storage.



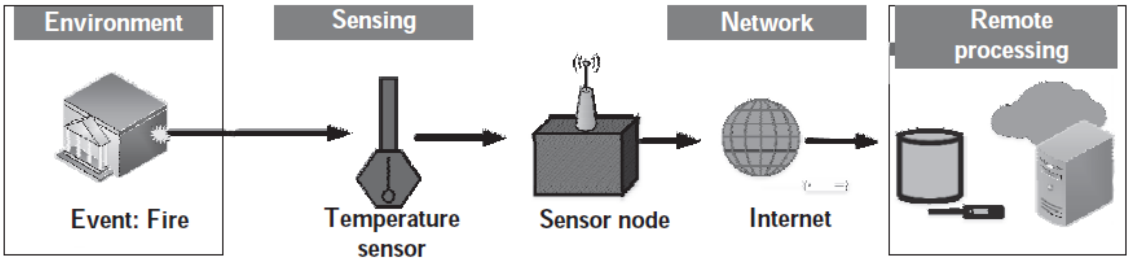
Off-site processing

In the off-site topology, the data from these sensor nodes (data generating sources) is transmitted either to a remote location (which can either be a server or a cloud) or to multiple processing nodes. Multiple nodes can come together to share their processing power in order to collaboratively process the data (which is important in case a feasible communication pathway or connection to a remote location cannot be established by a single node).

Remote processing
This is one of the most common processing topologies prevalent in present-day IoT solutions. It encompasses sensing of data by various sensor nodes; the data is then forwarded to a remote server or a cloud-based infrastructure for further processing and analytics. The processing of data from hundreds and thousands of sensor nodes can be simultaneously offloaded to a single, powerful computing platform; this results in massive cost and energy savings by enabling the reuse and reallocation of the same processing resource while also enabling the deployment of smaller and simpler processing nodes at the site of deployment

Collaborative processing
This processing topology typically finds use in scenarios with limited or no network connectivity, especially systems lacking a backbone network. Additionally, this topology can be quite economical for large-scale deployments spread over vast areas,

where providing networked access to a remote infrastructure is not viable. In such scenarios, the simplest solution is to club together the processing power of nearby processing nodes and collaboratively process the data in the vicinity of the data source itself.



Q 7 a) List common connectivity protocols in IOT

Answer: The main features of fifteen identified commonly used and upcoming IoT connectivity enablers. These connectivity technologies can be integrated with existing sensing, actuation, and processing solutions for extending connectivity to them. Some of these solutions necessarily require integration with a minimal form of processing infrastructure, such as Wi-Fi. In contrast, others, such as Zigbee, can work in a standalone mode altogether, without the need for external processing and hardware support.

Q 7 b) Explain Salient features and application scope of any 5 connectivity protocols

Answer:

IEEE 802.15.4

The IEEE 802.15.4 standard represents the most popular standard for low data rate wireless personal area networks (WPAN) [1]. This standard was developed to enable monitoring and control applications with lower data rate and extend the operational life for uses with low-power consumption. This standard uses only the first two layers—physical and data link—for operation along with two new layers above it: 1) logical link control (LLC) and 2) service-specific convergence sublayer (SSCS). The additional layers help in the communication of the lower layers with the upper layers. Figure 7.1 shows the IEEE 802.15.4 operational layers. The IEEE 802.15.4 standard was curated to operate in the ISM (industrial, scientific, and medical) band.

Zigbee

The Zigbee radio communication is designed for enabling wireless personal area networks (WPANs). It uses the IEEE 802.15.4 standard for defining its physical and medium access control (layers 1 and 2 of the OSI stack). Zigbee finds common usage in sensor and control networks [4]. It was designed for low-powered mesh networks at low cost, which can be broadly implemented for controlling and monitoring applications, typically in the range of 10–100 meters [3]. The PHY and MAC layers in this communication are designed to handle multiple low data rate operating devices. The frequencies of 2.4 GHz, 902–928 MHz or 868 MHz are commonly associated with Zigbee WPAN operations. The Zigbee commonly uses 250 kbps data rate which is optimal for both periodic and intermittent full-duplex data transmission between two Zigbee entities.

Thread

Thread is built upon the IEEE 802.15.4 radio standard; it is used for extremely low power consumption and low latency deployments [5]. Unlike Zigbee, Thread can extend direct Internet connectivity to the devices it is connected with. Thread removes the need for a mobile phone or a proprietary gateway to be in the range of devices for accessing the Internet. It is specially designed for IoT with the need for interoperability, security, power, and architecture addressed in a single radio platform.

ISA100.11A

The ISA100.11A is a very low power communication standard and has been developed and managed by ISA (International Society of Automation) [7]. Similar to the previous protocols, it uses the IEEE 802.15.4 standard as a base for building its protocol. The standard was mainly proposed for industrial plant automation systems. The ISA100.11A is characterized by an IoT compliant protocol stack, which can also be integrated with wired networks using Ethernet, support for open access protocols and device-level interoperability; it boasts of a 128-bit AES (Advanced Encryption Standard) encryption securing all communications. The security in ISA100.11A is in two layers: Transport layer and data link layer

WirelessHART

WirelessHART can be considered as the wireless evolution of the highway addressable remote transducer (HART) protocol [7]. It is a license-free protocol, which was developed for networking smart field devices in industrial environments. The lack of wires makes the adaptability of this protocol significantly advantageous over its predecessor, HART, in industrial settings. By virtue of its highly encrypted communication, wireless HART is very secure and has several advantages over traditional communication protocols. Similar to Zigbee, wirelessHART uses the IEEE 802.15.4 standard for its protocols designing.

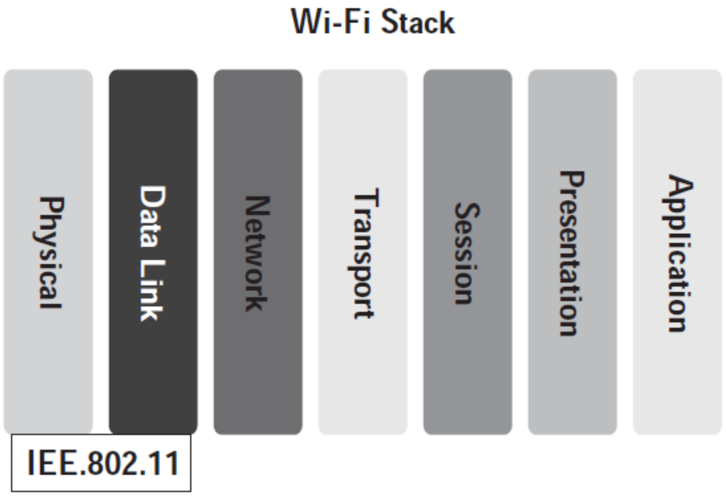
Q 7 c) Differentiate between WiFi and Bluetooth connectivity protocols in IoT

Answer:

Wi-Fi

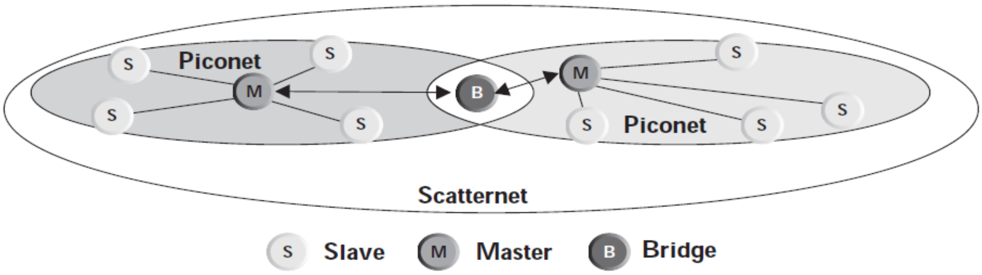
Wi-Fi or WiFi is technically referred to by its standard, IEEE 802.11, and is a wireless technology for wireless local area networking of nodes and devices built upon similar standards (Figure 7.25). Wi-Fi utilizes the 2.4 GHz ultra high frequency (UHF) band or the 5.8 GHz super high frequency (SHF) ISM radio bands for communication [16]. For operation, these bands in Wi-Fi are subdivided into multiple channels. The communication over each of these channels is achieved by multiple devices simultaneously using time-sharing based TDMA multiplexing. It uses CSMA/CA for channel access.

Various versions of IEEE 802.11 have been popularly adapted, such as a/b/g/n. The IEEE 802.11a achieves a data rate of 54 Mbps and works on the 5 GHz band using OFDM for communication. IEEE 802.11b achieves a data rate of 11 Mbps and operates on the 2.4 GHz band. Similarly, IEEE 802.11g also works on the 2.4 GHz band but achieves higher data rates of 54 Mbps using OFDM. Finally, the newest version, IEEE 802.11n, can transmit data at a rate of 140 Mbps on the 5 GHz band.



Bluetooth

Bluetooth is defined by the IEEE 802.15.1 standard and is a short-range wireless communication technology operating at low power to enable communication among two or more Bluetooth-enabled devices [17]. It was initially developed as a cable replacement technology for data communication between two or more mobile devices such as smartphones and laptops. This standard allows the transmission of data as well as voice-over short distances. Bluetooth functions on the 2.4 GHz ISM band and has a range of approximately 10 m. The transmission of data is done through frequency hopping spread spectrum (FHSS), which also reduces the interference caused by other devices functioning in the 2.4 GHz band. The data is divided into packets before transmitting them by Bluetooth. The packets are transmitted over the 79 designated channels, each 1MHz wide in the 2.4 GHz band.

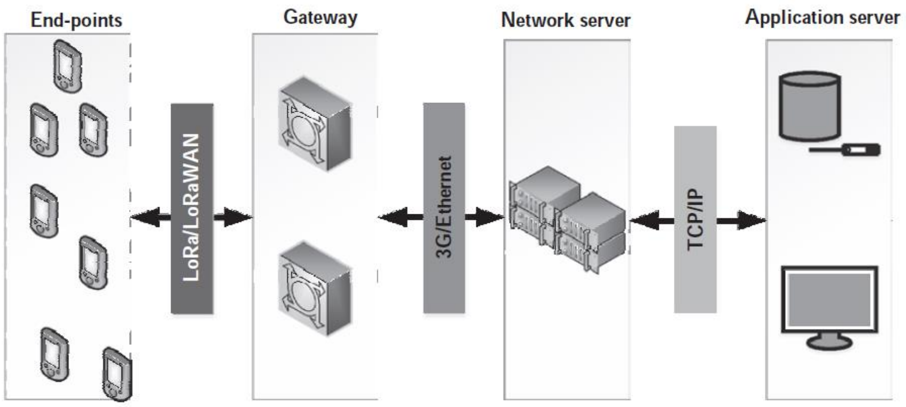


Q 8 a) With necessary diagrams, explain in detail any four connectivity protocols in IT

Answer:

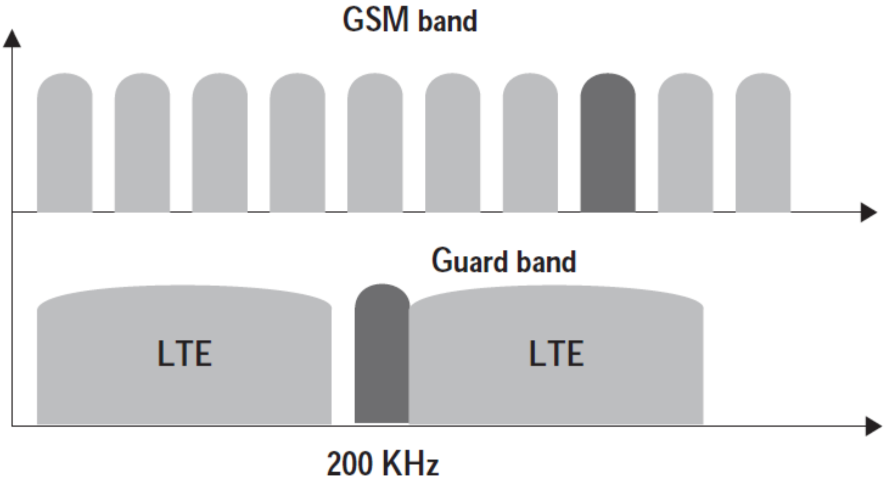
LoRa

LoRa or long range is a patented wireless technology for communication developed by Cycleo of Grenoble, France for cellular-type communications aimed at providing connectivity to M2M and IoT solutions [14]. It is a sub-GHz wireless technology that operationally uses the 169 MHz, 433 MHz, 868 MHz, and 915 MHz frequency bands for communication. LoRa uses bi-directional communication links symmetrically and a spread spectrum with a 125 kHz wideband for operating. Applications such as electric grid monitoring are typically suited for utilizing LoRa for communications. Typical communication of LoRa devices ranges from 15 to 20 km, with support for millions of devices. Figure 7.21 shows the LoRa network architecture



NB-IoT

NB-IoT or narrowband IoT is an initiative by the Third Generation Partnership Project (3GPP) to develop a cellular standard, which can coexist with cellular systems (2G/3G/4G), be highly interoperable and that too using minimum power [15]. It is reported that a major portion of the NB-IoT applications can support a battery life of up to ten years. NB-IoT also boasts of significant improvements in reliability, spectrum efficiencies, and system capacities. NB-IoT uses orthogonal frequency division multiplexing (OFDM) modulation, which enhances the system capacity and increases spectrum efficiency (Figure 7.23). However, device complexities are quite high. NB-IoT also provides support for security features such as confidentiality, authentication, and integrity

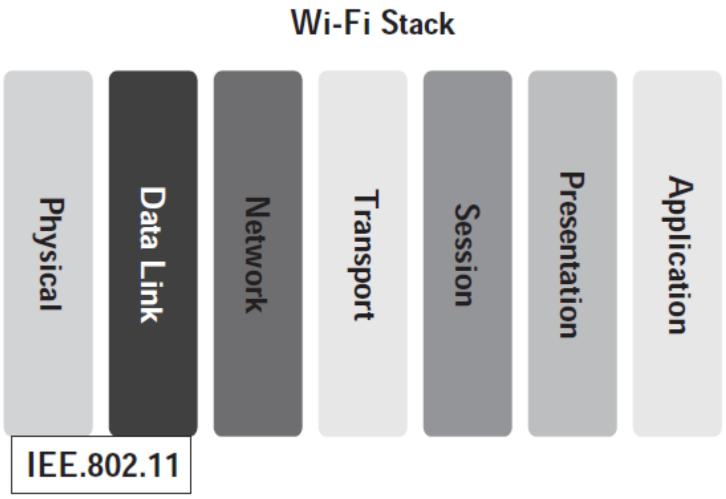


Wi-Fi

Wi-Fi or WiFi is technically referred to by its standard, IEEE 802.11, and is a wireless technology for wireless local area networking of nodes and devices built upon similar standards (Figure 7.25). Wi-Fi utilizes the 2.4 GHz ultra high frequency (UHF) band or the 5.8 GHz super high frequency (SHF) ISM radio bands for communication [16]. For operation, these bands in Wi-Fi are subdivided into multiple channels. The communication over each of these channels is achieved by multiple devices simultaneously using time-sharing based TDMA multiplexing. It uses CSMA/CA for channel access.

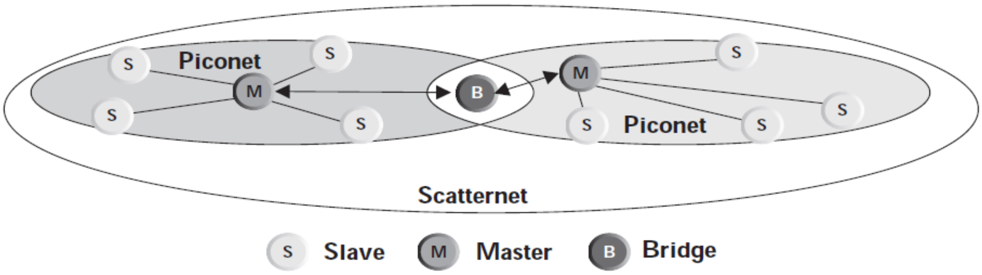
Various versions of IEEE 802.11 have been popularly adapted, such as a/b/g/n. The IEEE 802.11a achieves a data rate of 54 Mbps and works on the 5 GHz band using

OFDM for communication. IEEE 802.11b achieves a data rate of 11 Mbps and operates on the 2.4 GHz band. Similarly, IEEE 802.11g also works on the 2.4 GHz band but achieves higher data rates of 54 Mbps using OFDM. Finally, the newest version, IEEE 802.11n, can transmit data at a rate of 140 Mbps on the 5 GHz band.



Bluetooth

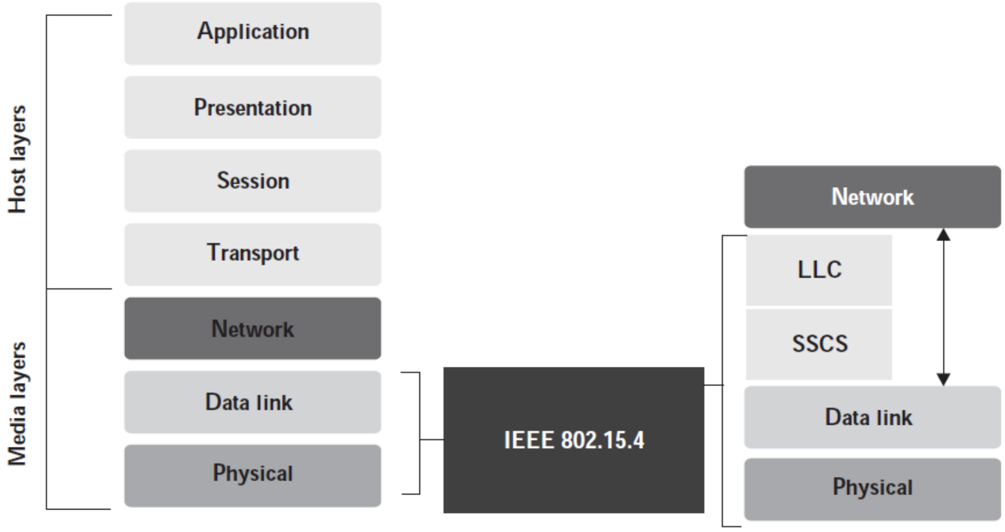
Bluetooth is defined by the IEEE 802.15.1 standard and is a short-range wireless communication technology operating at low power to enable communication among two or more Bluetooth-enabled devices [17]. It was initially developed as a cable replacement technology for data communication between two or more mobile devices such as smartphones and laptops. This standard allows the transmission of data as well as voice-over short distances. Bluetooth functions on the 2.4 GHz ISM band and has a range of approximately 10 m. The transmission of data is done through frequency hopping spread spectrum (FHSS), which also reduces the interference caused by other devices functioning in the 2.4 GHz band. The data is divided into packets before transmitting them by Bluetooth. The packets are transmitted over the 79 designated channels, each 1MHz wide in the 2.4 GHz band.



Q 8 b) Determine the requirements associated with any of IoT connectivity protocols in real world solutions

Answer: IEEE 802.15.4

The IEEE 802.15.4 standard represents the most popular standard for low data rate wireless personal area networks (WPAN) [1]. This standard was developed to enable monitoring and control applications with lower data rate and extend the operational life for uses with low-power consumption. This standard uses only the first two layers—physical and data link—for operation along with two new layers above it: 1) logical link control (LLC) and 2) service-specific convergence sublayer (SSCS). The additional layers help in the communication of the lower layers with the upper layers.

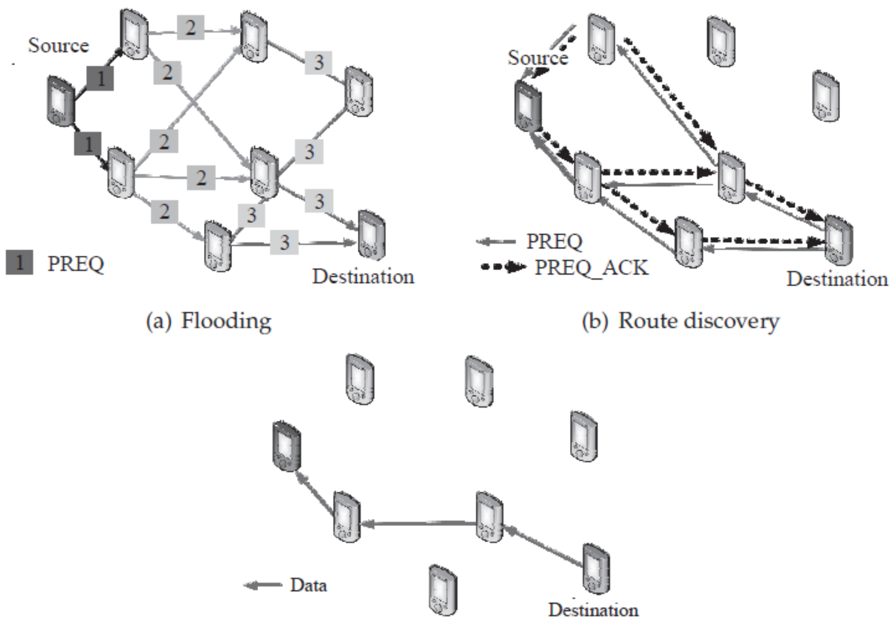


Q 9q) Describe in detail the various infrastructure protocols in IoT based communication technologies

Answer: There are eight popular IoTbased communication technologies: Internet Protocol Version 6 (IPv6), Lightweight On-demand Ad hoc Distance vector Routing Protocol–Next Generation (LOADng), Routing Protocol for Low-Power and Lossy Networks (RPL), IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), Quick UDP Internet Connection (QUIC), micro IP (uIP), nanoIP, and Content-Centric Networking (CCN).

LOADng

LOADng stands for Lightweight On-demand Ad hoc Distance vector Routing Protocol – Next Generation. This protocol is inspired by the AODV (Ad hoc On-Demand Distance Vector) routing protocol, which is primarily a distance vector routing scheme [6]. Figure 8.3 illustrates the LOADng operation. Unlike AODV, LOADng was developed as a reactive protocol by taking into consideration the challenges of Mobile Ad hoc Networks (MANETs). The LOADng process starts with the initiation of the action of route discovery by a LOADng router through the generation of route requests (RREQs), as illustrated in Figure 8.3(a). The router forwards packets to its nearest connected neighbors, each of which again forwards the packets to their one-hop neighbors. This process is continued until the intended destination is reached. Upon receiving the RREQ packet, the destination sends back a route reply (RREP) packet toward the RREQ originating router (Figure 8.3(b)). In continuation, route error (RERR) messages are generated and sent to the origin router if a route is found to be down between the origin and the destination.



(a) Flooding

(b) Route discovery

(c) Route establishment

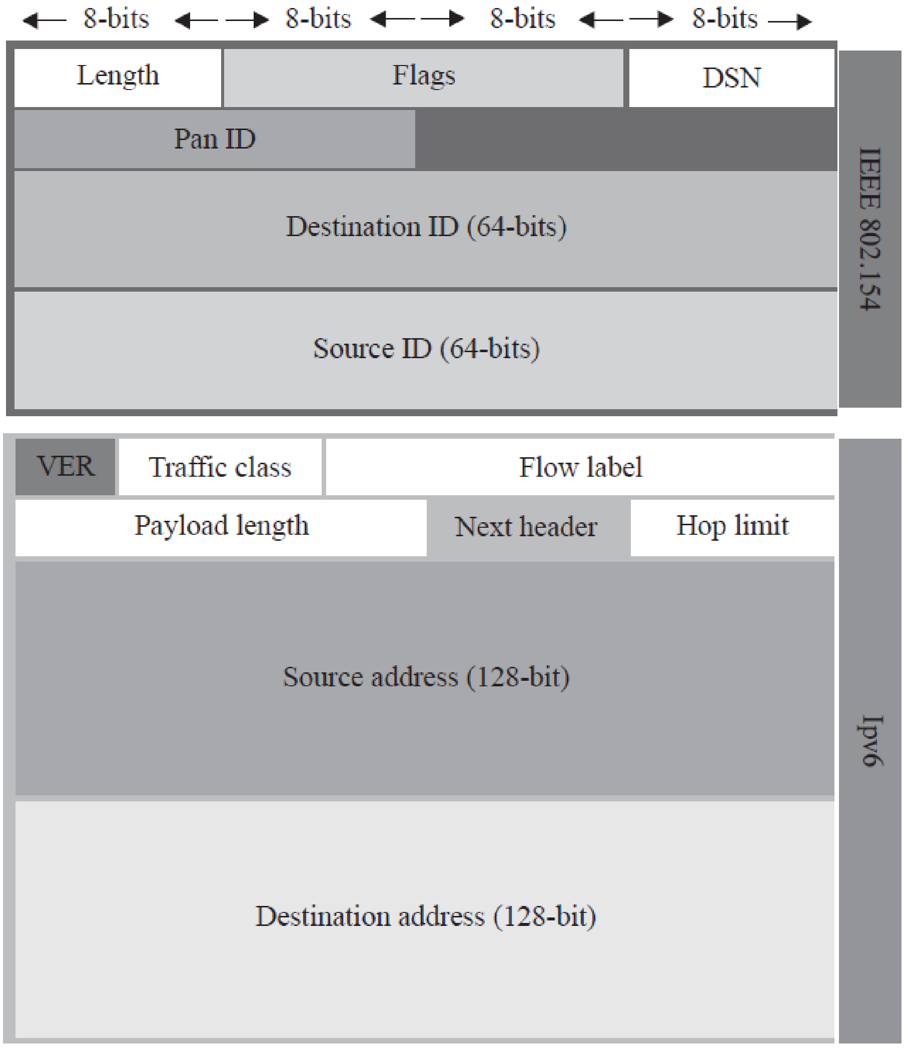
To summarize the operation of LOADng, a router performs the following tasks:
• Bi-directional network route discovery between a source and a destination.
• Route establishment and route maintenance between the source and the destination only when data is to be sent through the route.

Generation of control and signaling traffic in the network only when data is to

be transferred or a route to the destination is down.

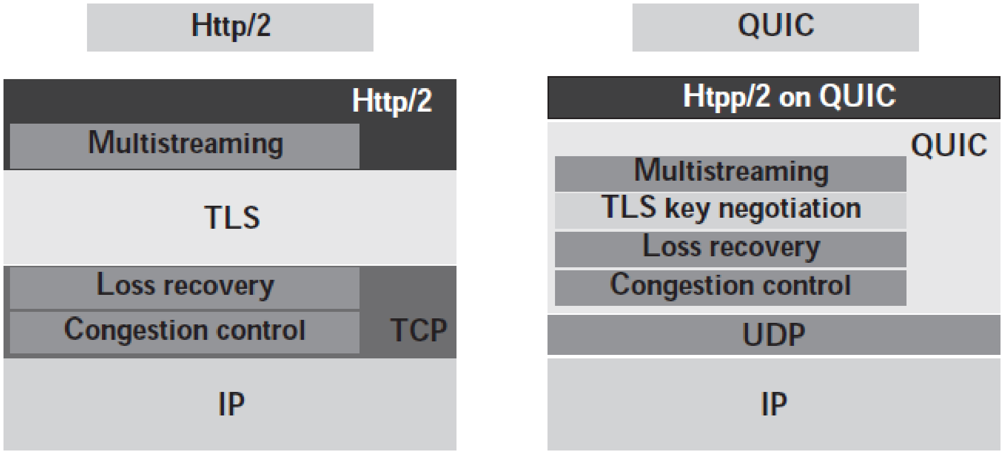
6LoWPAN

6LoWPAN allows low power and constrained devices/nodes to connect to the Internet. 6LoWPAN stands for IPv6 over low power wireless personal area networks. As the name suggests, it enables IPv6 support for WPANs, which are limited concerning power, communication range, memory, and throughput [8]. 6LoWPAN is designed to be operational and straightforward over low-cost systems, and extend IPv6 networking capabilities to IEEE 802.15.4-based networks. Popular uses of this protocol are associated with smart grids, M2M applications, and IoT. 6LoWPAN allows constrained IEEE 802.15.4 devices to accommodate 128-bit long IPv6 addresses. This is achieved through header compression, which allows the protocol to compress and retro-fit IPv6 packets to the IEEE 802.15.4 packet format.
6LoWPANnetworks can consist of both limited capability (concerning throughput, processing, memory, range) devices—called reduced function devices (RFD)—and devices with significantly better capabilities, called full function devices (FFD). The RFDs are so constrained that for accessing IP-based networks, they have to forward their data to FFDs in their personal area network (PAN). The FFDs yet again forward the forwarded data from the RFD to a 6LoWPAN gateway in a multi-hop manner. The gateway connects the packet to the IPv6 domain in the communication network. From here on, the packet is forwarded to the destination IP-enabled node/device using regular IPv6-based networking.



QUIC

Quick UDP internet connections (QUIC) was developed (and still undergoing developments) to work as a low-latency and independent TCP connection [9]. The aim behind the development of this protocol is to achieve a highly reduced latency (almost zero round-trip-time) communication scheme with stream and multiplexing support like the SPDY protocol developed by Google. Figure 8.8 illustrates the differences between the positions of the various functionalities in QUIC and regular HTTP protocols.

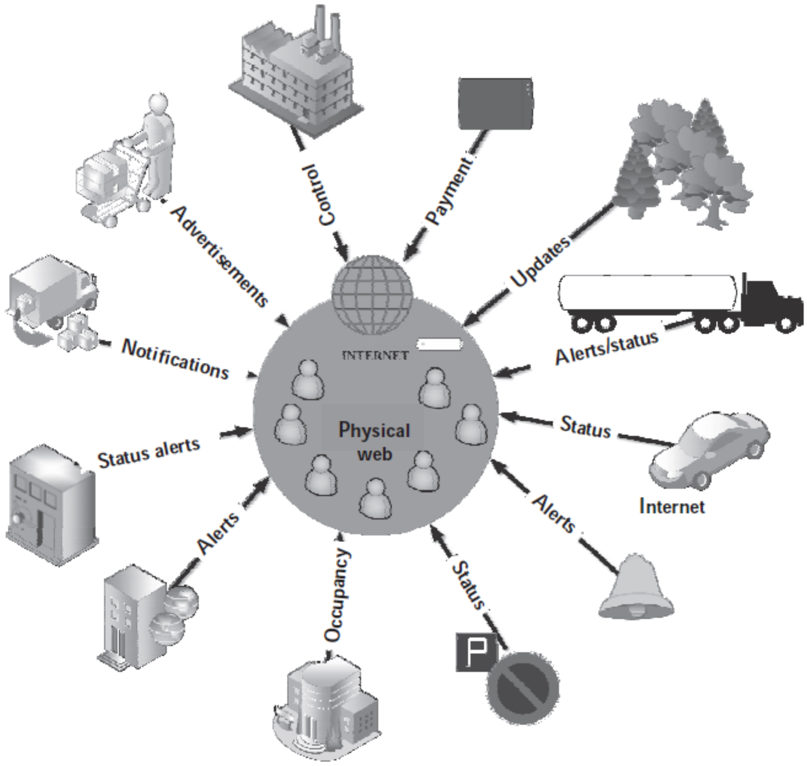


Q 9 b) Explain the following discovery protocols

i) Physical Web
ii) mDNS
iii) Universal Plug and play

Physical web

The physical web was designed to provide its users with the ability to interact with physical objects and locations seamlessly. The information to the users can be in the form of regular web pages or even dynamic web applications [13].
Some examples in the context of the physical web include user-friendly buses, which can alert its users about various route-related information, smart home appliances that can teach new users how to use them, self-diagnostic robots and machinery in industries, smart pet tags which can provide information about the pet's owner and its home location, and many more. Figure 8.13 shows the outline of a physical web model. The main takeaway of this concept is the seamless integration of several standalone smart systems through the web to provide information on demand to its users.



The physical web broadcasts a list of URLs within a short radius around it so that anyone in range can see the available URLs and interact with them. This paradigm is primarily built upon Bluetooth low energy (BLE), which is used to broadcast the content as beacons. The primary requirement of any supporting beacons to function in the physical web and broadcast URLs is their ability to support the Eddystone protocol specification. BLE was primarily chosen for the physical web due to its ubiquity, efficiency, and extended battery life of several years.

URLs are one of the core principles of the web and can be either flexible or decentralized. These URLs allow any application to have a presence on the web and enables the digital presence of an object or thing. As of now, physical web deployments have been undertaken in public spaces, and any device with a physical web scanner can detect these URLs. The use of URLs extends the benefits of the web security model to the physical web. Features such as secured login, secure communication over HTTPS (HTTP over secure socket layer), domain obfuscation, and others can be easily integrated with the physical web.

Multicast DNS (mDNS)

The multicast domain name system or mDNS is explicitly designed for small networks and is analogous to regular DNS (domain name system), which is tasked with the resolution of IP addresses [14]. Interestingly, this system is free from any local name server from an operational point of view. However, it can work with regular DNS systems as it is a zero-configuration service. It uses multicast UDP for resolving host names. An mDNS client initiates a multicast query on the IP network, which asks a remote host to identify itself. The mDNS cache in the associated network subnet is updated with the multicast response received from the target. A node can give up its claim to a domain name by setting the time-to-live (TTL) field to zero in its response packet to an mDNS query. Some popular implementations of mDNS include the Apple Bonjour service and the networked printer discovery service in Windows 10 operating system from Microsoft. The main drawback of mDNS is its host name resolution reach to a top-level domain only.

Universal plug and play (UPnP)

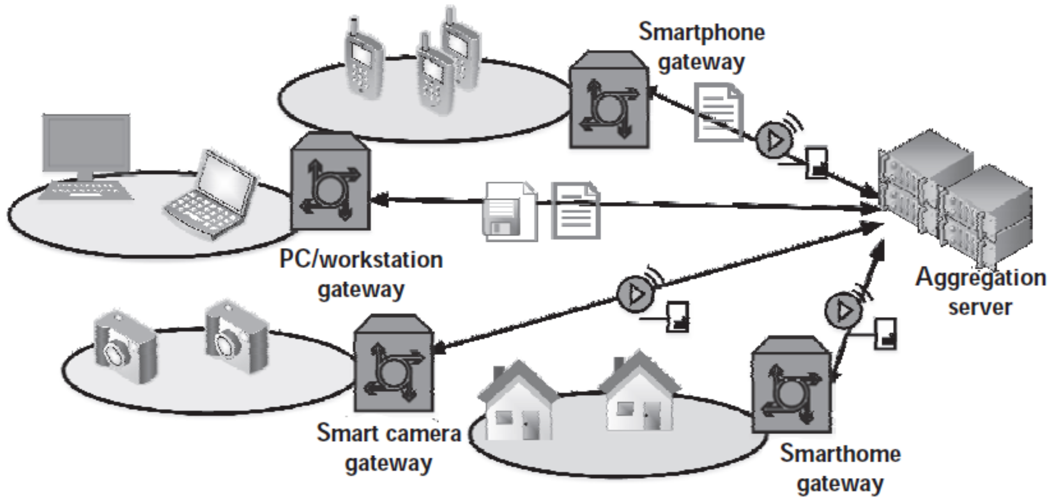
Universal plug and play or UPnP encompasses a set of networking protocols aimed at service discovery and the establishment of functional network-based data sharing and communication services [15]. In brief, it is mainly used for enabling dynamic connections of devices to computing platforms. This paradigm is termed plug and play as the devices attaching to a computer network can configure themselves and update their hosts about their working configurations over a network. The UPnP is backed by a forum of many consumer electronics vendors and industries and is managed by the Open Connectivity Foundation. As UPnP is primarily designed for non-enterprise devices, its scope includes the discovery and intercommunication between networked devices such as mobiles, printers, access points, gateways, televisions, and other regular commercial systems enabled with IP capabilities. Figure 8.14 outlines the underlying UPnP stack and the relative location of the various functionalities in the stack.

The present-day UPnP has been designed to run on IP enabled networks, and makes use of the networking services of HTTP, XML, and SOAP for data transfer, device descriptions, and event generation and monitoring. UPnP enables UDP-based HTTP device search requests and advertisements using multicasting. The responses to device requests are necessarily unicast. UPnP advertisements use UDP port 1900 for multicasting. The unnecessary overheads and traffic generated by UPnP systems and their multicast behavior make them unsuitable for enterprise systems.

Q 10a) With neat diagram, discuss the illustration of the various facets of the interoperability in IoT

Answer: The urgency in the requirement for interoperability and interoperable solutions in IoT arose mainly due to the following reasons:

Large-scale Cooperation: There is a need for cooperation and coordination among the huge number of IoT devices, systems, standards, and platforms; this is a long-standing problem. Proprietary solutions are seldom reusable and economical in the long run, which is yet another reason for the demand for interoperability.
(ii) **Global Heterogeneity:** The network of devices within and outside the purview of gateways and their subnets are quite large considering the spread of IoT and the applications it is being adapted to daily. Device heterogeneity spans the globe when connected through the Internet. A common syntax, platform, or standard is required for unifying these heterogeneous devices.
(iii) **Unknown IoT Device Configuration:** Device heterogeneity is often accompanied by further heterogeneity in device configurations. Especially considering the global-scale network of devices, the vast combinations of device configurations such as data rate, frequencies, protocols, language, syntax, and others, which are often unknown beforehand, further raise the requirement of interoperable solutions.
(iv) **Semantic Conflicts:** The variations in processing logic and the way data is handled by the numerous sensors and devices making up a typical IoT implementation, makes it impossible for rapid and robust deployment. Additionally, the variations in the end applications and their supported platform configurations further add to the challenges.



Q 10b) Discuss any four IoT interoperability standards

Answer: Standards

Toward enabling IoT interoperability, various technologies have been standardized and are recognized globally for incorporating consistent interoperability efforts worldwide across various industries, domains, and technologies. We list seven of the popular ones in this chapter.

EnOcean

EnOcean is a wireless technology designed for building automation systems, primarily based on the principle of energy harvesting [6]. Due to the robustness and popularity of EnOcean, it is being used in domains such as industries, transportation, logistics, and homes. As of 2012, EnOcean was adopted as a wireless standard under ISO/IEC 14543-3-10, providing detailed coverage of the physical, data link, and networking layers. EnOceanbased devices are batteryless. They use ultra-low power consuming electronics along with micro energy converters to enable wireless communication among themselves; the devices include networking components such as wireless sensors, switches, controllers, and gateways. The energy harvesting modules in EnOcean use micro-level variations and differences in electric, electromagnetic, solar, or other forms of energy to transform the energy into usable energy through highly efficient energy converters. The wireless signals from the batteryless EnOcean sensors and switches, which are designed to be maintenance-free, can operate up to 30 meters in buildings and homes and up to 300 meters in the open. EnOcean wireless sensor modules wirelessly transmit their data to EnOcean system modules, as shown in Figure 9.2.

EnOcean is typically characterized by low data rates (of about 125 kbit/s) for wireless packets that are 14 bytes long. This reduces the energy consumption of the EnOcean devices. Additional features such as the transmission of RF (radio frequency) energy only during transmission of 1s in the binary encoded message further reduce the energy consumption of these devices. Frequencies of 902 MHz, 928.35 MHz, 868.3 MHz, and 315 MHz are employed for transmission of messages in this technology.

DLNA

The Digital Living Network Alliance (DLNA), previously known as the Digital Home Working Group (DHWG), was proposed by a consortium of consumer electronics companies in 2003 to incorporate interoperability guidelines for digital media sharing among multimedia devices such as smartphones, smart TVs, tablets, multimedia servers, and storage servers. Primarily designed for home networking, this standard relies majorly on WLAN for communicating with other devices in its domain and can easily incorporate cable, satellite, and telecom service providers to ensure data transfer link protection at either end. The inclusion of a digital rights management layer allows for multimedia data sharing among users while avoiding piracy of data. The consumers in DLNA, which may consist of a variety of devices such as TVs, phones, tablets, media players, PCs, and others, can view subscribable content without any additional add-ons or devices through VidiPath. Figure 9.3 shows the steps involved in a typical DLNA-based multimedia streaming application. As of 2019, DLNA has over a billion devices following its guidelines globally

Konnex

Konnex or KNX is a royalty-free open Home Automation Network (HAN) based wired standard for domestic building and home applications. It relies on wired communication for achieving automation [8]. Wired configurations such as a star, tree, or line topologies can be achieved by using a variety of physical communication technologies involving twisted pair, power line, RF (KNX-RF), or IP-based (KNXnet/IP) ones. KNX evolved from three previous standards: 1) BatiBUS, 2) European Home Systems Protocol (EHS), and 3) European Installation Bus (EIB or Instabus). It has a broad scope of applications in building automation, which involve tasks such as controlling lighting, doors, windows, high-voltage AC (HVAC) systems, security systems, audio/video systems, and energy management. Figure 9.4 represents a typical Konnex-based building network. The KNX facilitates automation through distributed applications and their interaction using standard data types, objects, logical devices, and channels, which form an interworking model. The technology is robust enough to be supported by a wide range of hardware platforms, starting from a simple microcontroller to a sophisticated computer. The requirements of building automation often dictate the hardware requirements.

UPnP

The Universal Plug and Play (UPnP) was designed primarily for home networks as a set of protocols for networking devices such as PCs, printers, mobile devices, gateways, and wireless access points. UPnP can discover the presence of other UPnP devices on the network, as well as establish networks amongst them for communication and data sharing [9]. Whenever they are connected to a network, UPnP devices can establish working configurations with other devices. As of 2016, UPnP is managed by the Open Connectivity Forum (OCF). The underlying assumption of UPnP is the presence of an IP network over which it uses HTTP to share events, data, actions, and service/device descriptions through a device-todevice networking arrangement. Device search and advertisements are multicast through HTTP over UDP (HTTPMU) over port 1900.