Program : **B.E**

Subject Name: **Internet of Things**

Subject Code: **IT-8004**

Semester: **8th**

Department of Information Technology

Internet of Things (IT8004)

Unit I: Internet of Things (IoT)**:** Vision, Definition, Conceptual Framework, Architectural view, technology behind IoT, Sources of the IoT, M2M Communication, IoT Examples . Design Principles for Connected Devices: IoT/M2M systems layers and design standardization, communication technologies, data enrichment and consolidation, ease of designing and affordability

INTERNET OF THINGS (IoT):

IoT Vision: The vision behind IoT is to have plug-n-play smart objects that can be deployed in any environment with an interoperable interconnection backbone that allows them to blend with other smart objects around them. Standardization of frequency bands and protocols plays a pivotal role in accomplishing this goal.

IoT Definition: The Internet of Things refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems.

IOT Conceptual Framework

The main tasks of this framework are to analyze and determine the smart activities of these intelligent devices through maintaining a dynamic interconnection among those devices. The proposed framework will help to standardize IoT infrastructure so that it can receive e-services based on context information leaving the current infrastructure unchanged. The active collaboration of these heterogeneous devices and protocols can lead to future ambient computing where the maximum utilization of cloud computing will be ensured. This model is capable of logical division of physical devices placement, creation of virtual links among different domains, networks and collaborate among multiple application without any central coordination system. IaaS can afford standard functionalities to accommodate and provides access to cloud infrastructure. The service is generally offered by modern data centers maintained by giant companies and organization. It is categorized as virtualization of resources which permits a user to install and run application over virtualization layer and allows the system to be distributed, configurable and scalable.

Total infrastructure system can be categorized into 4 layers to receive context supported e-services out of raw data from the Internet of Things. These 4 layers establish a generic framework that does not alter the current network infrastructure but create an interfacing among services and entities through network virtualization.

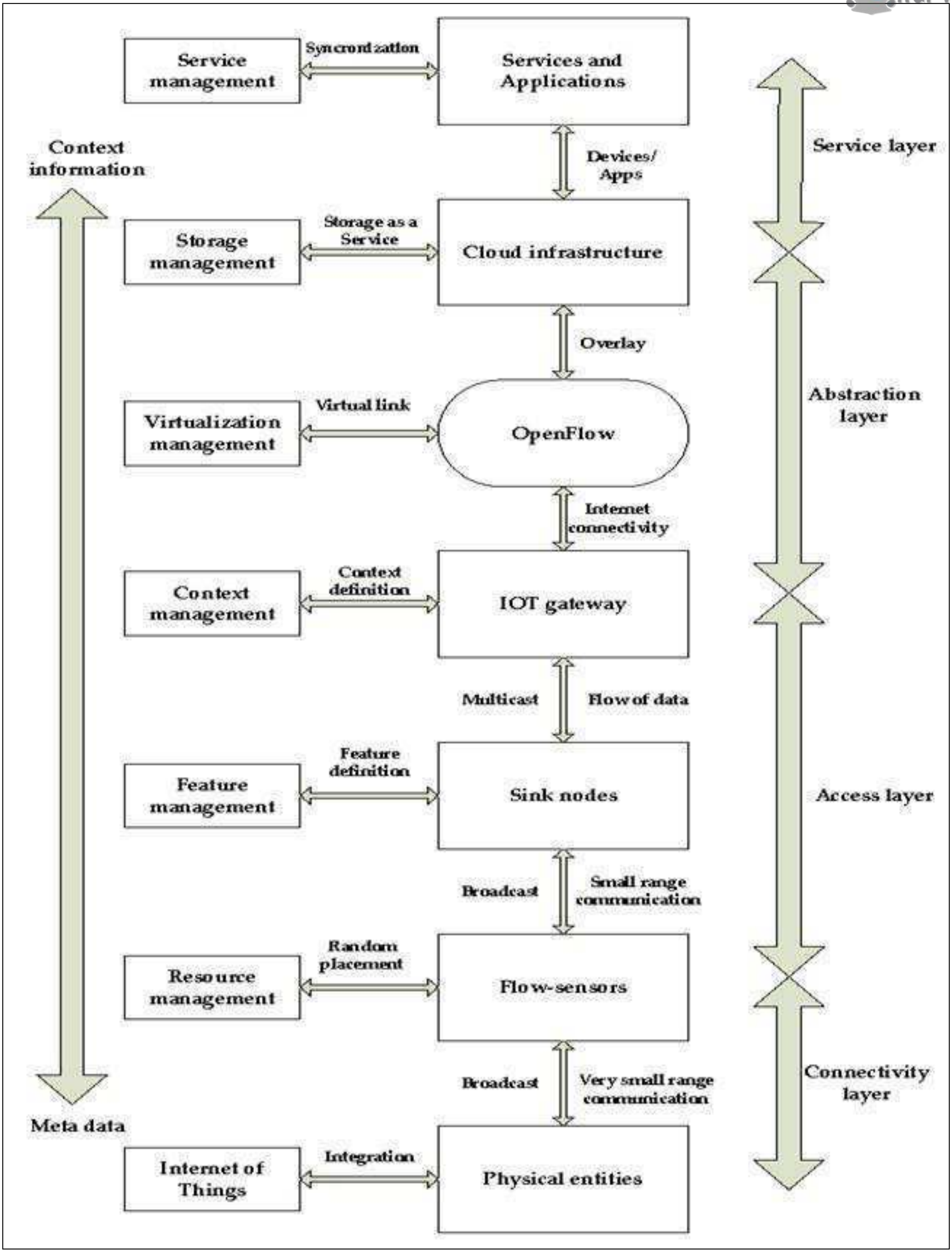


Figure 1.1 : IOT Conceptual View

1. Connectivity Layer

This layer includes all the physical devices involved in the framework and the interconnection among them. Future internet largely depends on the unification of these common objects found everywhere near us and these should be distinctly identifiable and controllable.

This layer also involves assigning of low range networking devices like sensors, actuators, RFID tags etc and resource management checks the availability of physical resources of all the devices and networks involved in the underlying infrastructure. These devices contain very limited resources and resource management ensures the maximum utilization with little overhead. It also allows sharing and distribution of information among multiple networks or single network divided into multiple domains.

2. Access Layer

Context Data will be reached to internet via IoT Gateway as captured by short range devices in form of raw data. Access layer comprises topology definition, network initiation, creation of domains etc. This layer also includes connection setup, intra-inter domain communication, scheduling, packet transmissions between flow-sensors and IoT gateway. The simulation was run later in this paper for different scenario based on this layer. Feature management contains a feature filter which accepts only acceptable context data and redundant data are rejected. Large number of sensor maintains lots of features but only a small subset of features is useful generate a context data.

Feature filter helps to reduce irrelevant data transmission, increases the data transfer rate of useful data and reduce energy and CPU consumption too. Number of features can be different based on the application requirements and context data types.

3. Abstraction Layer

One of the most important characteristics of OpenFlow is to add virtual layers with the preset layers, leaving the established infrastructure unchanged. A virtual link can be created among different networks and a common platform can be developed for various communication systems. The system is fully a centralized system from physical layer viewpoint but a distribution of service (flow visor could be utilized) could be maintained. One central system can monitor, control all sorts of traffics. It can help to achieve better band-width, reliability, robust routing, etc. which will lead to a better Quality of Services (QoS).

In a multi-hopping scenario packets are transferred via some adjacent nodes. So, nodes near to access points bears too much load in comparison to distant nodes in a downstream scenario and inactivity of these important nodes may cause the network to be collapsed. Virtual presence of sensor nodes can solve the problem where we can create a virtual link between two sensor networks through access point negotiation. So, we can design a three a three layer platform, where common platform and virtualization layer are newly added with established infrastructure. Sensors need not to be worried about reach-ability or their placement even in harsh areas. Packet could be sent to any nodes even if it is sited on different networks.

4. Service Layer

Storage management bears the idea about all sorts of unfamiliar and/or important technologies and information which can turn the system scalable and efficient. It is not only responsible for storing data but also to provide security along with it. It also allows accessing data effectively; integrating data to enhance service intelligence, analysis based on the services required and most importantly increases the storage efficiency. Storage and management layer involves data storage & system supervision, software services and business management & operations. Though they are included in one layer, the business support system resides slightly above of cloud computing service whereas Open-Flow is placed below of it as presented to include virtualizations and monitor management.

Service management combines the required services with organizational solutions and thus new generation user service becomes simplified. These forthcoming services are necessitated to be co interrelated and combined in order to meet the demand socio- economic factors such as environment analysis, safety measurement, climate management, agriculture modernization etc.

IOT architectural view-IOT architecture consists of different layers of technologies supporting IOT. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IOT deployments in different scenarios.

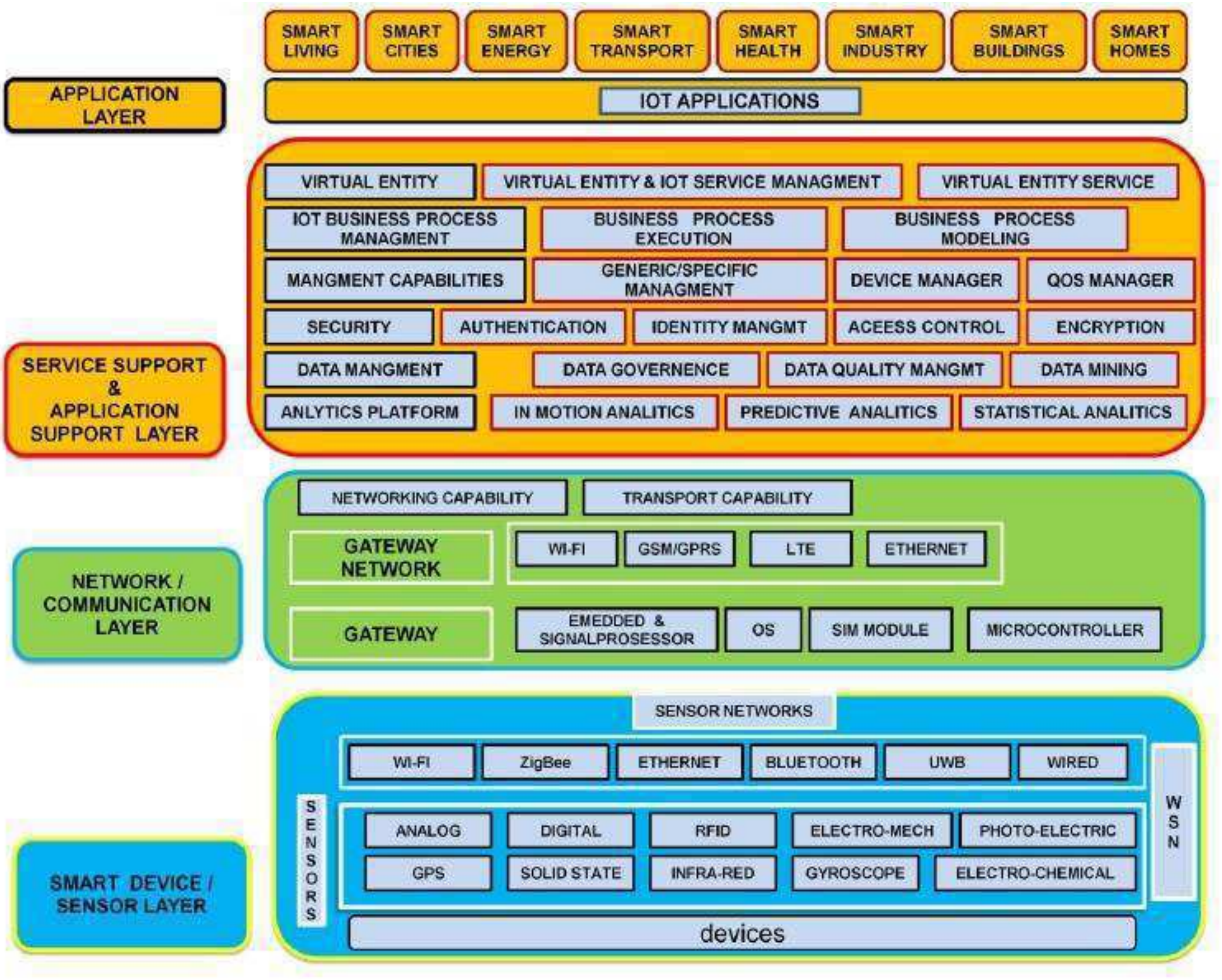


Figure 1.2: IOT architectural view

The functionality of each layer is described below:

- **Smart device / sensor layer:** The lowest layer is made up of smart objects integrated with sensors. The sensors enable the interconnection of the physical and digital worlds allowing real-time information to be collected and processed. There are various types of sensors for different purposes. The sensors have the capacity to take measurements such as temperature, air quality, speed, humidity, pressure, flow, movement and electricity etc. In some cases, they may also have a degree of memory, enabling them to record a certain number of measurements. A sensor can measure the physical property and convert it into signal that can be understood by an instrument. Sensors are grouped according to their unique purpose such as environmental sensors, body sensors, home appliance sensors and vehicle telemetric sensors, etc
- **Gateways and Networks-**Massive volume of data will be produced by these tiny sensors and this requires a robust and high performance wired or wireless network infrastructure as a transport medium. Current networks, often tied with very different protocols, have been used to support machine-to-machine (M2M) networks and their applications. With demand needed to serve a wider range of IOT services and applications such as high speed transactional services, context-aware applications, etc, multiple networks with various technologies and access protocols are needed to work with each other in a heterogeneous configuration. These networks can be in the form of a private, public or hybrid models and are built to support the communication requirements for latency, bandwidth or security. Various gateways (microcontroller, microprocessor) & gateway networks (WI-FI, GSM, GPRS).

- **Management Service Layer-**The management service renders the processing of information possible through analytics, security controls, process modeling and management of devices. One of the important features of the management service layer is the business and process rule engines. IOT brings connection and interaction of objects and systems together providing information in the form of events or contextual data such as temperature of goods, current location and traffic data. Some of these events require filtering or routing to post-processing systems such as capturing of periodic sensory data, while others require response to the immediate situations such as reacting to emergencies on patient's health conditions. The rule engines support the formulation of decision logics and trigger interactive and automated processes to enable a more responsive IOT system.
- **Application Layer-**The IoT application covers "smart" environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

The Technology Behind IoT

Several technologies are involved in making the Internet of Things possible. First is identification. Since there will be billions of devices that will connect to the Internet, each one requires a unique identification. This is only possible if they're IPv6 enabled, as the current IPv4 network has depleted its universe of IP addresses.

An IoT device needs to sense, which is possible by putting sensors that measure various aspects of an object. The object needs to have the ability to communicate what it has measured with the outside world. This world could be the Internet, or other similar objects around it. A central server where data from all the objects will be collected for analysis. It could be an application or an appliance that can download all data, and allow the user to control, manage, and analyze it.

Identification Technology for IoT

Radio Frequency Identification (RFID): This technology uses radio waves to accomplish communication between the data from an electronic device for the objective of identifying and also to locate and sense the environment around.

Quick Response Code: Short for QR code, this is a machine readable visual label that contains information about things to which it is affixed. It uses 4 standardized encoding modes viz. numeric, alpha numeric, byte/binary to effectively store information about those things. A QR Code on any device/things consists of black segments organized in a rectangle which can be read by QR software. Nowadays smartphones act as QR code readers, which interpret the code and extract information from it. QR codes can also be used to track where the thing has been scanned and find its location.

Communication Protocols Used

Things must communicate with each other. Data then must be collected and sent to a remote server, indicating device information or the environment around it which if required is sent back to devices with other information to trigger various decisions or actions. For this purpose various protocols are used.

Message Queue Telemetry Transport (MQTT) is a protocol to gather device data and communicate it to the IT infrastructure and servers. Large networks of devices can be controlled or monitored with it. The protocol works on top of TCP providing a reliable stream of data flow.

Extensible Messaging and Presence Protocol (XMPP) is a protocol used for connecting devices to people. It's an alternative to the D2S protocol, as people are connected to servers. XMPP provides a great way for example, to connect your home heater to a Web server so that you can monitor it from your smartphone. This protocol is ideal for consumer-oriented IoT applications.

Data Distribution Service (DDS) is a device-to-device communication protocol. It shares device data with other devices over a network. DDS provides effective ways to filter and choose exactly which information goes where.

Advanced Message Queuing Protocol (AMQP) is a queuing protocol used to link web servers to each other. In IoT, AMPQ is suited for server based functions.

Constrained Application Protocol (CoAP) is a protocol developed to be used in electronics devices allowing them to communicate interactively over the Internet. It is primarily focused on low power sensors, switches, valves and similar components that need to be controlled or accessed remotely, through standard Internet networks. CoAP works on application layer to be used in resource-constrained internet devices, such as Wireless Sensor Networks and WSN nodes.

IPv6 and the Internet of Things

Without public IP addresses, IoT capability would be greatly reduced. IPv6 spreads the addressing space in order to support all growing Internet-enabled devices. IPv6 has been designed to provide secure communications to users and mobility for all devices attached to the user. It has been regarded as the most suitable technology for IoT as it provides scalability, end to end connectivity, extended address space, etc. IPv6 integrated with Internet of Things can bring the world to a whole new level of interoperable devices leading to smarter cities, intelligent transport systems, advanced healthcare, etc. When all things can be represented by IPv6, we can ensure that makes use of IP protocols such as MIPv6 for mobile mobility and IPSec for security.

Other Protocols and APIs Used

IoT uses the REST API architecture and JSON, Java programming languages to function. The REST API is a platform that defines a set of principles by which web services can be developed to focus on a system's resources, such as how the resource states are addressed and transferred over HTTP by a wide range of devices. REST is mostly used in smart-phone applications, and automated business processes.

Xively REST API: This is a Platform as a Service (PaaS) for the Internet of Things. Xively makes it easier for interconnecting devices, data, people and places, to create powerful alternative solutions that will transform how people experience their world.

Devices/Hardware Available to Build Internet Connected Hardware in IOT

Developing an internet connected thing requires both hardware and software. With devices like Spark Core, Smart Things, Nest, WeMo, etc, designing a hardware that communicates with software and software that communicates with hardware can be achieved.

SOURCES OF IOT

Arduino Uno

Arduino Uno, for sure, is one the most used development board. It is an open-source development board based on ATmega328P.

Arduino MKR1000

Arduino MKR1000 is one of the latest board. MKR1000 has been designed for IoT projects. It supports natively WIFI connection and it is very easy to use. It is based on the Atmel ATSAMW25.

BeagleBone black

This board is a low-cost board supported by the community made for developers and hobbyists. It uses Linux as OS

Raspberry PI 2 Model B

Raspberry, like Arduino, is one the most used development board. Raspberry Pi has various versions with various power capabilities. Moreover, Raspberry can be considered a small computer because it has all the features we can find in a common PC: keyboard, HDMI and so on. It is very small (more or less it is like a credit card) and is powered by Linux. Raspberry PI 2 model B has been replaced by the latest Raspberry PI 3.

UDOO

UDOO Neo is an interesting board that differs from others. It is a Arduino-Android/Linux device. It has a set of built-in features like WIFI, g-axis motion sensor and Bluetooth support.

Particle Photon

Particle Photon is a very small development board with a built-in WIFI module. This makes it ready for IoT project. It has a set of expansion kits that make the development process faster.

ESP8266

This board is a low-cost board with a built-in WIFI system that enables rapid IoT project prototyping. It comes with several variants having specific features like memory capacity or pins number.

Intel Edison

Intel Edison is a new IoT development board very powerful. It comes in several variants, like Intel Edison with Arduino breakout kit.

MACHINE-TO-MACHINE COMMUNICATION

M2M means two machines "communicating," or exchanging data, without human interfacing or interaction. This includes serial connection, power line connection (PLC), or wireless communications in the industrial Internet of Things (IoT). Switching over to wireless has made M2M communication much easier and enabled more applications to be connected.

M2M allows virtually any sensor to communicate, which opens up the possibility of systems monitoring themselves and automatically responding to changes in the environment, with a much reduced need for human involvement. M2M can refer to any two machines—wired or wireless—communicating with one another.

Traditionally, M2M focused on "industrial telematics," which is a fancy way of explaining data transfer for some commercial benefit. But many original uses of M2M still stand today, like smart meters. Wireless M2M has been dominated by cellular since it came out in the mid-2000's with 2G cell networks. Because of this, the cellular market has tried to brand M2M as an inherently cellular thing by offering M2M data plans. But cellular M2M is only one subsection of the market, and it shouldn't be thought of as a cellular-only area.

M2M Working

The machine-to-machine communication makes the Internet of Things possible. M2M technologies can connect millions of devices within a single network. The range of connected devices includes anything from vending machines to medical equipment to vehicles to buildings. Virtually anything that houses sensor or control technology can be connected to some sort of wireless network.

Figure 1.3: M2M Communication

M2M networks are very similar to LAN or WAN networks, but are exclusively used to allow machines, sensors, and controls, to communicate. These devices feed information they collect back to other devices in the network. This process allows a human (or an intelligent control unit) to assess what is going on across the whole network and issue appropriate instructions to member devices

IOT EXAMPLES

1. Smart Home

With IoT you could switch on air conditioning before reaching home or switch off lights even after you have left home. Or unlock the doors to friends for temporary Smart Home products are promised to save time, energy and money. With Smart home companies like Nest, Ecobee, Ring and August, to name a few, will become household brands and are planning to deliver a never seen before experience.

2. Wearables

Wearable devices are installed with sensors and softwares which collect data and information about the users. This data is later pre-processed to extract essential insights about user.These devices broadly cover fitness, health and entertainment requirements. The pre-requisite from internet of things technology for wearable applications is to be highly energy efficient or ultra-low power and small sized.

3. Connected Cars

A connected car is a vehicle which is able to optimize it's own operation, maintenance as well as comfort of passengers using onboard sensors and internet connectivity.

4. Industrial Internet

Industrial Internet is the new buzz in the industrial sector, also termed as Industrial Internet of Things (IIoT). It is empowering industrial engineering with sensors, software and big data analytics to create brilliant machines.

The philosophy behind IIoT is that, smart machines are more accurate and consistent than humans in communicating through data. And, this data can help companies pick inefficiencies and problems sooner.

IIoT holds great potential for quality control and sustainability. Applications for tracking goods, real time information exchange about inventory among suppliers and retailers and automated delivery will increase the supply chain efficiency.

5. Smart Cities

Smart city is another powerful application of IoT generating curiosity among world's population. Smart surveillance, automated transportation, smarter energy management systems, water distribution, urban security and environmental monitoring all are examples of internet of things applications for smart cities.

By installing sensors and using web applications, citizens can find free available parking slots across the city. Also, the sensors can detect meter tampering issues, general malfunctions and any installation issues in the electricity system.

6. IoT in agriculture

Farmers are using meaningful insights from the data to yield better return on investment. Sensing for soil moisture and nutrients, controlling water usage for plant growth and determining custom fertilizer are some simple uses of IoT.

7. Smart Retail

Smart phones will be the way for retailers to remain connected with their consumers even out of store. Interacting through Smart phones and using Beacon technology can help retailers serve their consumers better. They can also track consumer's path through a store and improve store layout and place premium products in high traffic areas.

8. Energy Engagement

The basic idea behind the smart grids is to collect data in an automated fashion and analyze the behavior or electricity consumers and suppliers for improving efficiency as well as economics of electricity use.

Smart Grids will also be able to detect sources of power outages more quickly and at individual household levels like nearby solar panel, making possible distributed energy system.

9. IOT in Healthcare

IoT in healthcare is aimed at empowering people to live healthier life by wearing connected devices.

The collected data will help in personalized analysis of an individual's health and provide tailor made strategies to combat illness.

10. IoT in Poultry and Farming

Livestock monitoring is about animal husbandry and cost saving. Using IoT applications to gather data about the health and well being of the cattle, ranchers knowing early about the sick animal can pull out and help prevent large number of sick cattle.

DESIGN PRINCIPLES FOR CONNECTED DEVICES

Interoperability

At the most fundamental level, a connected system requires sensors, machines, equipment, and sites, to communicate and exchange data. Interoperability is the underlying principle throughout all design processes.

Information transparency

The rapid growth of connected devices means continuous bridging between the physical and digital worlds. In this context, information transparency means that physical processes should be recorded and stored virtually, creating a Digital Twin.

Technical assistance

A driving benefit of IoT, technical assistance refers to the ability of connected systems to provide and display data that helps people to make better operational decisions and solve issues faster. In addition, IoT-enabled things should assist people in laborious tasks to improve productivity and safety.

Decentralized decisions

The final principle is for the connected system to go beyond assisting and exchanging data, to be able to make decisions and execute requirements according to its defined logic.

IoT/M2M systems layers and design standardization

Internet Engineering Task Force(IETF) is responsible for creating the design and standardization of IOT abs M2M. IETF suggests some IOT specifications like:

Modified OSI for IoT and M2M

The modified OSI model consists of six layers

New applications and services are present in layer-6 . The Layer-5 is used for client-server interaction with help of the protocols like CoAP. The Application support layer also may include process for data managing, acquiring and organizing.

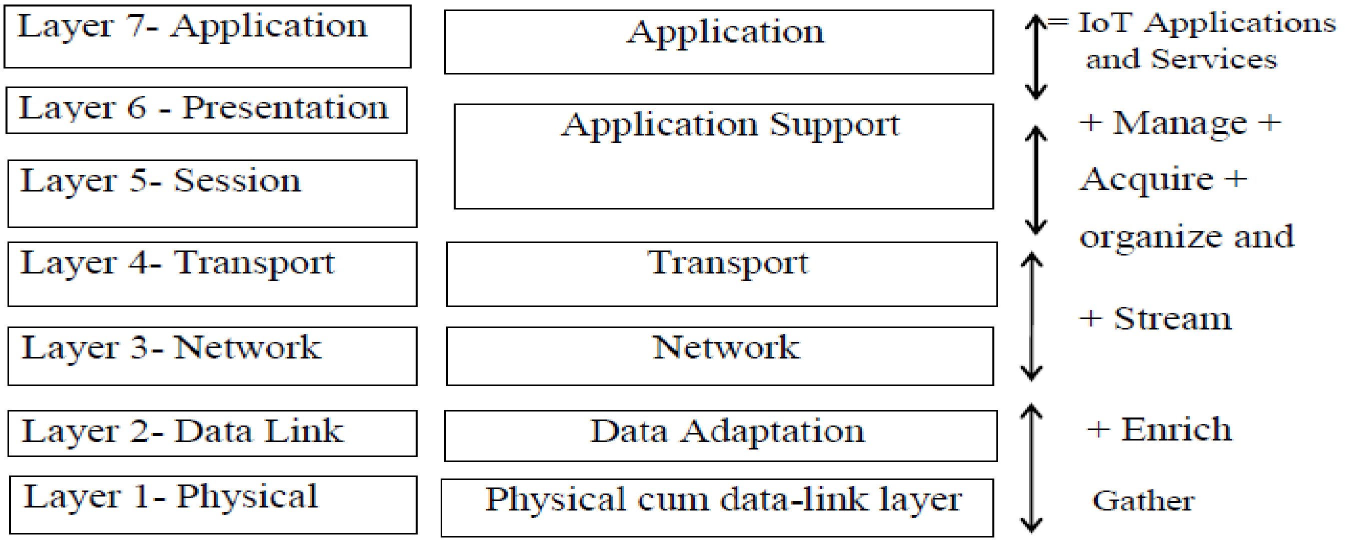


Figure 1.4: Modified 6 layer OSI

Another layer ie data adaptation layers contains a gateway that enable the communication between devices network and the web.

A physical IoT/M2 hardware may integrate a transceiver using a communication protocol as well as data link protocol for linking the data stacks

Wi-Fi:

Wi fi is a interface technology that uses IEEE 802.11 protocol and enables WLAN. WiFi connects the distributed WLANs network using internet.

RF Transceivers and RF Modules

These are the simplest RF circuits. It receives RF from one end and transmit RF from another end. IoT and M2M deploys RF Modules with transceivers

Communication Technologies

GPRS/GPS Cellular networks

Data enrichment and Consolidation

Data consolidation refers to the collection and integration of data from multiple sources into a single destination. During this process, different data sources are put together, or consolidated, into a single data store

Because data comes from a broad range of sources, consolidation allows organizations to more easily present data, while also facilitating effective data analysis. Data consolidation techniques reduce inefficiencies, like data duplication, costs related to reliance on multiple databases and multiple data management points.

Data enrichment is a general term that refers to processes used to enhance, refine or otherwise improve raw data. This idea and other similar concepts contribute to making data a valuable asset for almost any modern business or enterprise. It also shows the common imperative of proactively using this data in various ways.

Although data enrichment can work in many different ways, many of the tools used for this goal involve a refinement of data that might include small errors. A common data enrichment process could, for example, correct likely misspellings or typographical errors in a database through the use of precision algorithms. Following this logic, data enrichment tools could also add information to simple data tables.

Another way that data enrichment can work is in extrapolating data. Through methodologies such as fuzzy logic, engineers can produce more from a given raw data set. This and other projects can be described as data enrichment activities.

Ease of Designing

- Design for connected devices for IoT Applications, Services and business processes
- Designer considers the ease in designing the devices physical, data link, adaption layers and gateway
- Means availability of sensors, actuators, controllers and IoT devices
- Low in cost and hardware
- Use preferably open source software components and protocols
- Device hardware should embed minimum of components
- Use ready solutions for ease in designing local devices personal area network
- Ensure the secure connectivity with the Internet

Affordability of IoT devices

- The card: An embedded microcontroller, memory, OS, NFC peripheral interfaces, access point based device activation, RF module and transceiver and all that at low cost
- For example, Wireless sensors use Mote (mobile terminal)
- Mote: Low cost devices with open source OS (tiny OS) and software components
- Provides ease and affordance in the WSN networks

Program : **B.E**

Subject Name: **Internet of Things**

Subject Code: **IT-8004**

Semester: **8th**

Unit II: Hardware for IoT: Sensors, digital sensors, actuators, radio frequency identification (RFID) technology, wireless sensor networks, participatory sensing technology. Embedded Platforms for IoT: Embedded computing basics, Overview of IOT supported Hardware platforms such as Arduino, Raspberry pi, Beagle Bone, Intel Galileo.

HARDWARE FOR IOT:

Sensors: A sensor is any physical device that converts one form of energy into another. So, sensor converts some physical phenomenon into an electrical impulse that can then be interpreted to determine a reading. A microphone is a sensor that takes vibration energy (sound waves), and converts it to electrical energy in a useful way for other components in the system to correlate back to the original sound.

Digital Sensors: Electronic sensors or electrochemical sensors in which data conversion and data transmission takes place digitally are called as digital sensors. These digital sensors are replacing analog sensors as they are capable of overcoming the drawbacks of analog sensors. The digital sensor consists of majorly three components: senor, cable, and transmitter. In digital sensors, the signal measured is directly converted into digital signal output inside the digital sensor itself. And this digital signal is transmitted through cable digitally. There are different types of digital sensors that overcome disadvantages of analog sensors.

Actuator: An actuator operates in the reverse direction of a sensor. It takes an electrical input and turns it into physical action. For instance, an electric motor, a hydraulic system, and a pneumatic system are all different types of actuators.

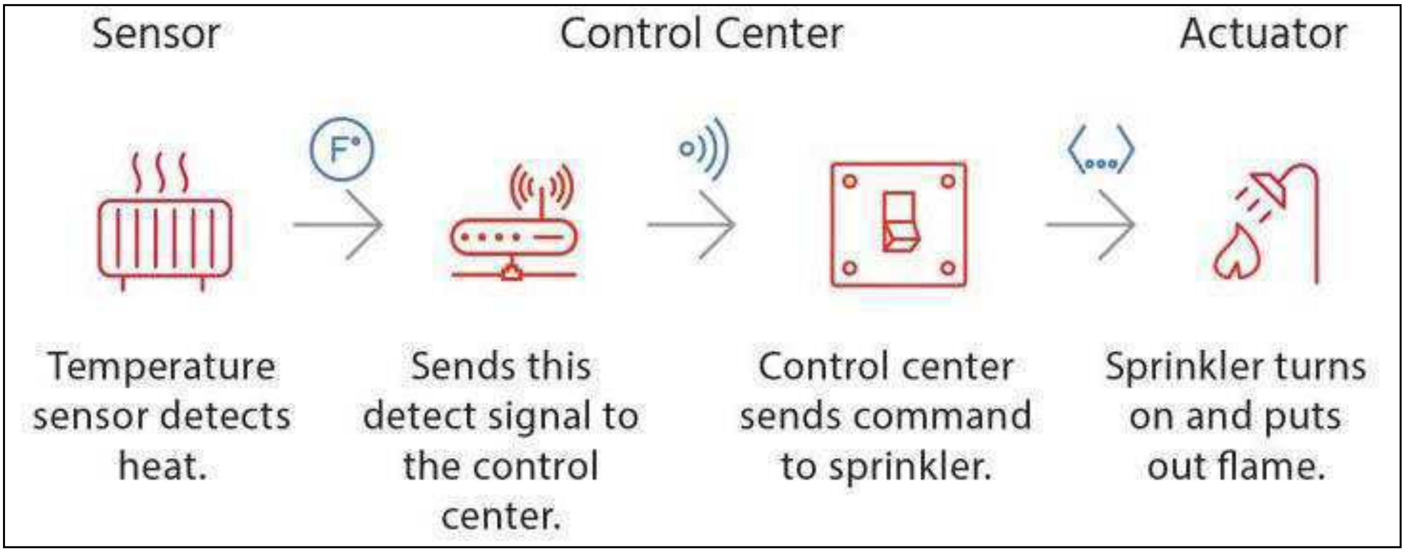

Sensor
Control Center
Actuator
F°
Temperature sensor detects heat.
Sends this detect signal to the control center.
Control center sends command to sprinkler.
Sprinkler turns on and puts out flame.

Figure 2.1: IOT System with sensor and actuator

There are many different types of sensors in an IoT system. Flow sensors, temperature sensors, voltage sensors, humidity sensors, and the list goes on. In addition, there are multiple ways to measure the same thing. For instance, airflow might be measured by using a small propeller like the one you would see on a weather station. Alternatively, as in a vehicle measuring the air through the engine, airflow is measured by heating a small element and measuring the rate at which the element is cooling.

RFID Technology: Radio-Frequency Identification (RFID) is the use of radio waves to read and capture information stored on a tag attached to an object. A tag can be read from up to several feet away and does not need to be within direct line-of-sight of the reader to be tracked.

A RFID system is made up of two parts: a tag or label and a reader. RFID tags or labels are embedded with a transmitter and a receiver. The RFID component on the tags has two parts: a microchip that stores and processes information, and an antenna to receive and transmit a signal. The tag contains the specific serial number for one specific object.

To read the information encoded on a tag, a two-way radio transmitter-receiver called an interrogator or reader emits a signal to the tag using an antenna. The tag responds with the information written in its memory bank. The interrogator will then transmit the read results to an RFID computer program.

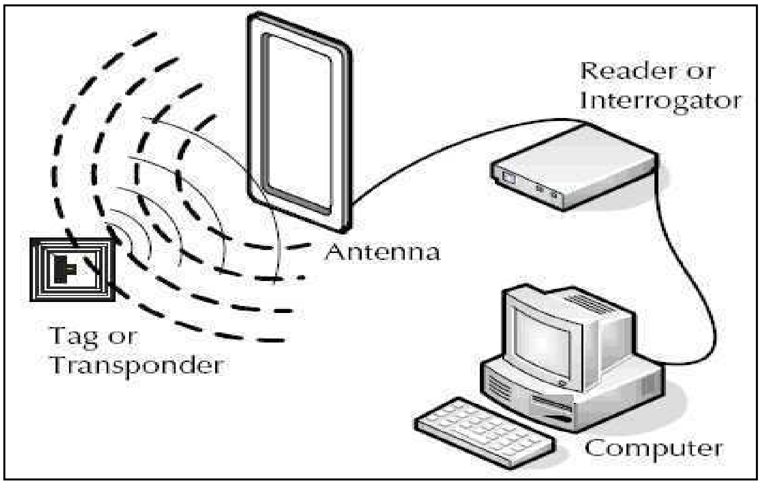


Figure 2.2: Working of RFID

There are two types of RFID tags: passive and battery powered. A passive RFID tag will use the interrogator's radio wave energy to relay its stored information back to the interrogator. A batter powered RFID tag is embedded with a small battery that powers the relay of information.

Wireless Sensor Networks: A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to monitor physical or environmental conditions. A WSN system incorporates a gateway that provides wireless connectivity back to the wired world and distributed nodes

Wireless Sensor Network Architecture

The most common WSN architecture follows the OSI architecture Model. The architecture of the WSN includes five layers and three cross layers. Mostly in sensor n/w we require five layers, namely application, transport, n/w, data link & physical layer. The three cross planes are namely power management, mobility management, and task management. These layers of the WSN are used to accomplish the n/w and make the sensors work together in order to raise the complete efficiency of the network.

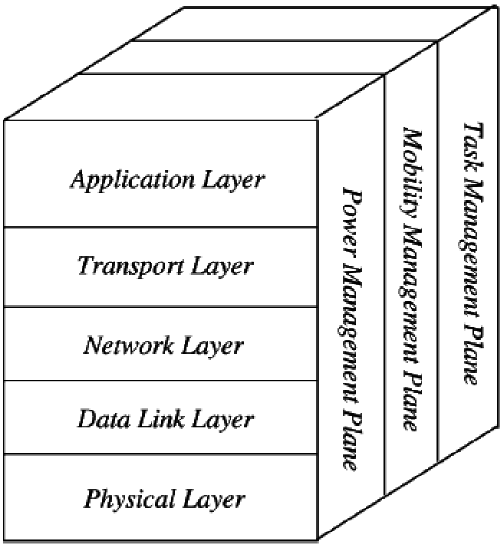


Figure 2.3: WSN Architecture

Application Layer: The application layer is liable for traffic management and offers software for numerous applications that convert the data in a clear form to find positive information. Sensor networks arranged in numerous applications in different fields such as agricultural, military, environment, medical, etc.

Transport Layer: The function of the transport layer is to deliver congestion avoidance and reliability where a lot of protocols intended to offer this function are either practical on the upstream. These protocols use dissimilar mechanisms for loss recognition and loss recovery. The transport layer is exactly needed when a system is planned to contact other networks.

Network Layer: The main function of the network layer is routing, it has a lot of tasks based on the application, but actually, the main tasks are in the power conserving, partial memory, buffers, and sensor don't have a universal ID and have to be self-organized.

Data Link Layer: The data link layer is liable for multiplexing data frame detection, data streams, MAC, & error control, confirm the reliability of point–point (or) point– multipoint.

Physical Layer: The physical layer provides an edge for transferring a stream of bits above physical medium. This layer is responsible for the selection of frequency, generation of a carrier frequency, signal detection, Modulation & data encryption. IEEE 802.15.4 is suggested as typical for low rate particular areas & wireless sensor network with low cost, power consumption, density, the range of communication to improve the battery life. CSMA/CA is used to support star & peer to peer topology. There are several versions of IEEE 802.15.4.V.

Participatory Sensing Technology: Participatory Sensing is an approach to data collection and interpretation in which individuals, acting alone or in groups, use their personal mobile devices and web services to systematically explore interesting aspects of their worlds ranging from health to culture.

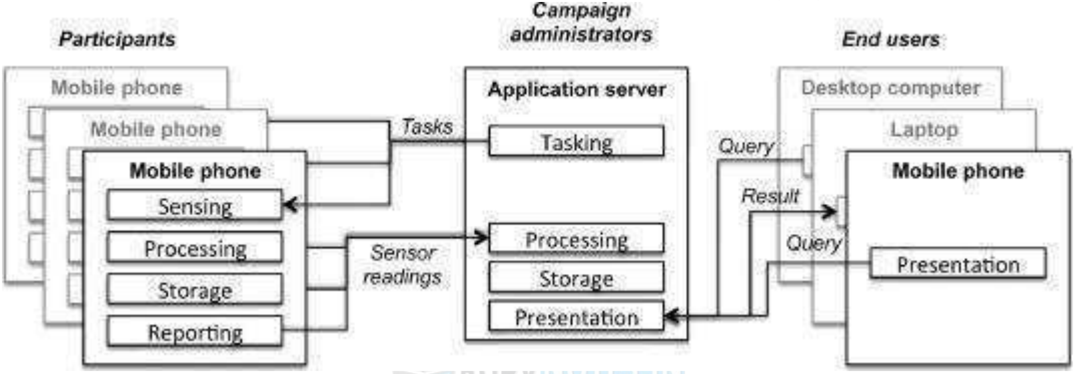


Figure 2.4: Participatory Sensing Technology:

Participatory sensing is the process whereby individuals and communities use ever more capable mobile phones and cloud services to collect and analyze systematic data for use in discovery. The convergence of technology and analytical innovation with a citizenry that is increasingly comfortable using mobile phones and online social networking sets the stage for this technology to dramatically impact many aspects of daily lives. Ubiquitous data capture, leveraged data processing, and personal data vault are the essential components for these emerging systems.

EMBEDDED PLATFORMS FOR IoT

Embedded system is defined as a way of working, performing or organizing one or many tasks according to a fixed set of rules (or) an arrangement in which all the units assemble and work together according to the program or plan. Examples of embedded systems are a watch and washing machine.

Embedded Systems Basics

The embedded systems basics include the components of embedded system hardware, embedded system types and several characteristics. An embedded system has three main components: Embedded system hardware, Embedded system software and Operating system.

Embedded System Hardware:

As with any electronic system, an embedded system requires a hardware platform on which it performs the operation. Embedded system hardware is built with a microprocessor or microcontroller. The embedded system hardware has elements like input output (I/O) interfaces, user interface, memory and the display. Usually, an embedded system consists of:

- Power Supply
- Processor
- Memory
- Timers

- Serial communication ports
- Output/Output circuits
- System application specific circuits

Embedded System Software:

The embedded system software is written to perform a specific function. It is typically written in a high level format and then compiled down to provide code that can be lodged within a non-volatile memory within the hardware. An embedded system software is designed to keep in view of the three limits:

1. Availability of system memory
2. Availability of processor's speed
3. When the system runs continuously, there is a need to limit power dissipation for events like stop, run and wake up.

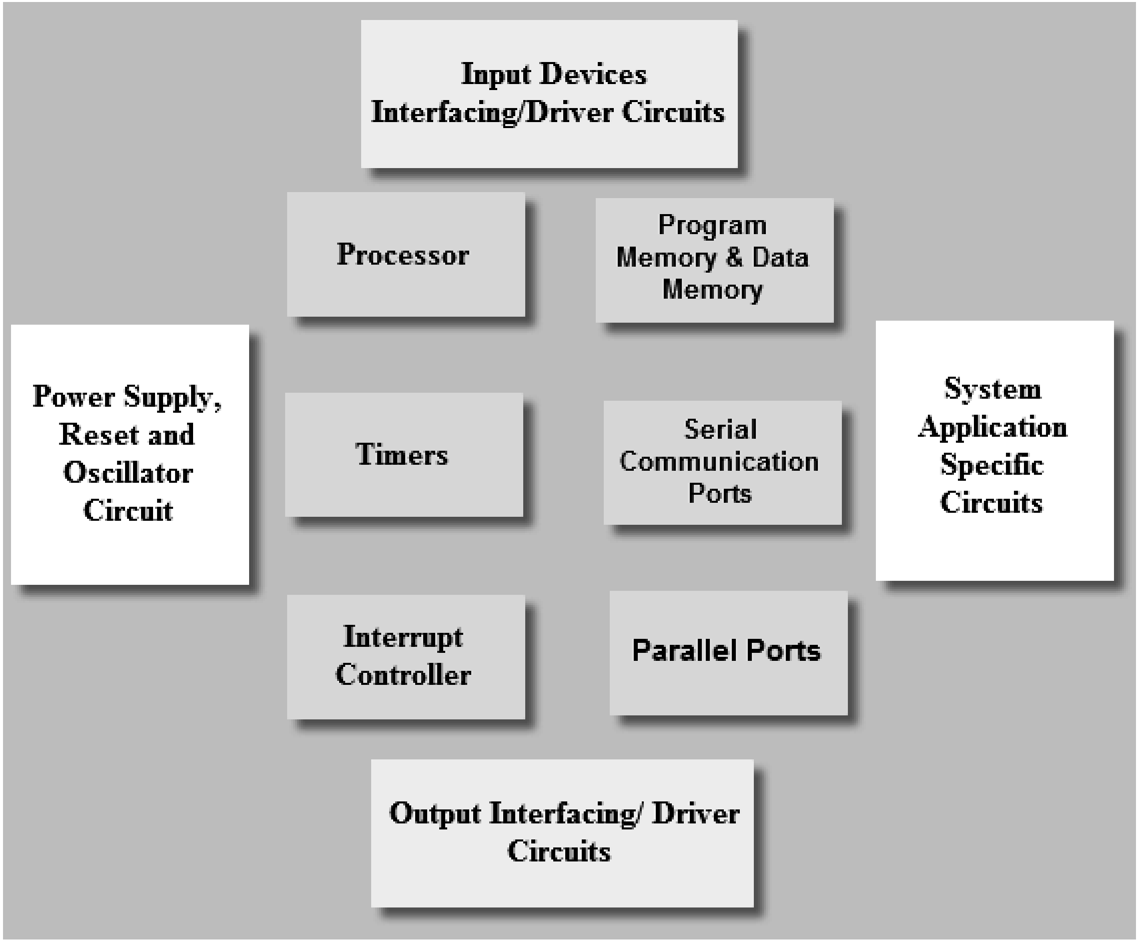

Input Devices Interfacing/Driver Circuits
Processor
Program Memory & Data Memory
Power Supply, Reset and Oscillator Circuit
Timers
Serial Communication Ports
System Application Specific Circuits
Interrupt Controller
Parallel Ports
Output Interfacing/ Driver Circuits

Figure 2.5: Embedded System Block Diagram

Real Time Operating System

A system is said to be real time, if it is essential to complete its work and deliver its service on time. Real time operating system manages the application software and affords a mechanism to let the processor run. The Real Time operating system is responsible for handling the hardware resources of a computer and host applications which run on the computer.

An RTOS is specially designed to run applications with very precise timing and a high amount of reliability. Especially, this can be important in measurement and industrial automation systems wherein downtime is costly or a program delay could cause a safety hazard.

Memory:

In an embedded system, there are different types of memories. The various forms of memories are presented in the below chart.

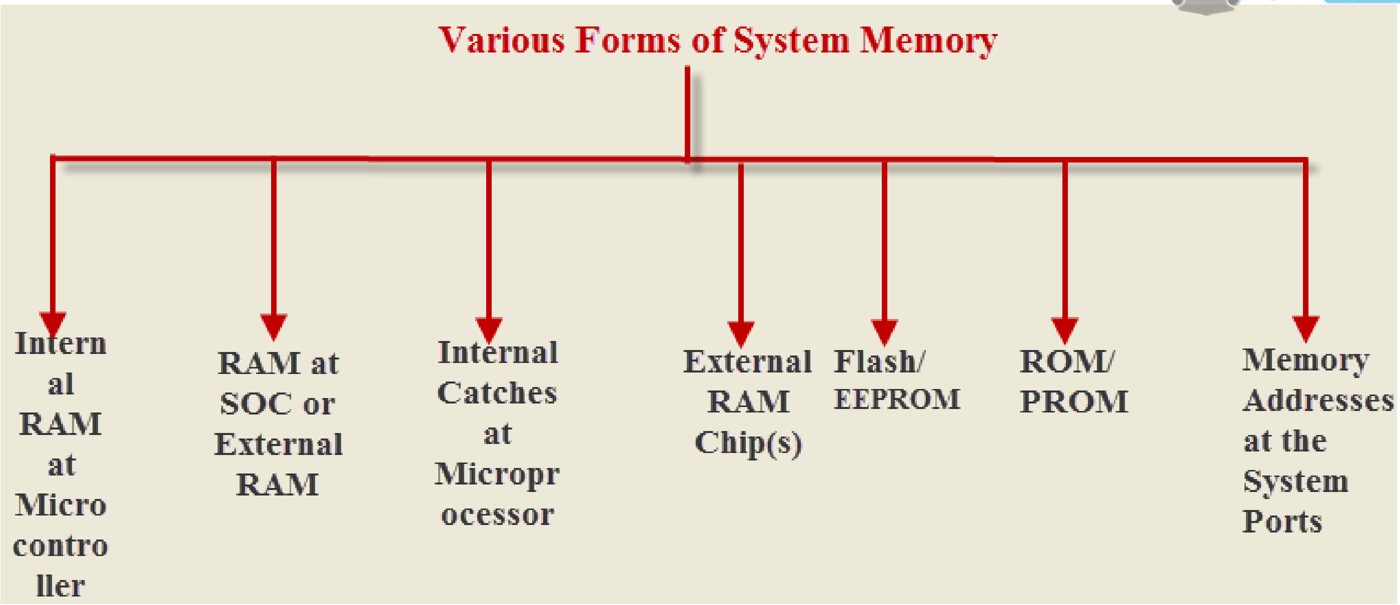

Various Forms of System Memory
Internal RAM at Microcontroller
RAM at SOC or External RAM
Internal Catches at Microprocessor
External RAM Chip(s)
Flash/ EEPROM
ROM/ PROM
Memory Addresses at the System Ports

Figure 2.5: various forms of memories

Processors:

Different processors used in embedded systems are microprocessor, (DSP) Digital Signal Processor, microcontroller, RISC processor, ASIP processor, Arm processor and ASSP processor.

Classification of Embedded Systems:

Embedded systems are classified into three:

1. Small Scale Embedded Systems
2. Medium Scale Embedded Systems
3. Sophisticated Embedded Systems

Small Scale Embedded Systems: Small scale embedded systems are designed with a single 8 or 16-bit microcontroller which may even be operated with a battery. For developing embedded software for these types of systems, an editor, assembler, (IDE) integrated development environment, and cross assembler are the main programming tools.

Medium Scale Embedded Systems: Medium scale embedded systems are designed with a single or few 16 or 32 bit microcontrollers, DSPs or RISCs. These systems have both hardware and software complexities. When developing embedded software for these types of systems, the following programming tools are available: C, C++, Visual C++, Java, and RTOS, source code engineering tool, debugger, simulator and integrated development environment.

Sophisticated Embedded Systems: Sophisticated embedded systems have huge hardware and software complexities and may need PLAs, IPs, ASIPs, scalable processors or configurable processors. They are used for cutting-edge applications that need hardware and software co-design & components which have to combine in the final system**.**

OVERVIEW OF IOT SUPPORTED HARDWARE PLATFORMS SUCH AS ARDUINO, RASPBERRY PI, BEAGLE BONE, INTEL GALILEO.

Arduino: Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. User can tell board what to do by sending a set of instructions to the microcontroller on the board. For these the Arduino programming language, and the Arduino Software (IDE) is used.

Advantages of Arduino:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50 (USA)

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it

Raspberry Pi: The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It’s capable of doing everything like browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.
Advantages of Raspberry Pi:

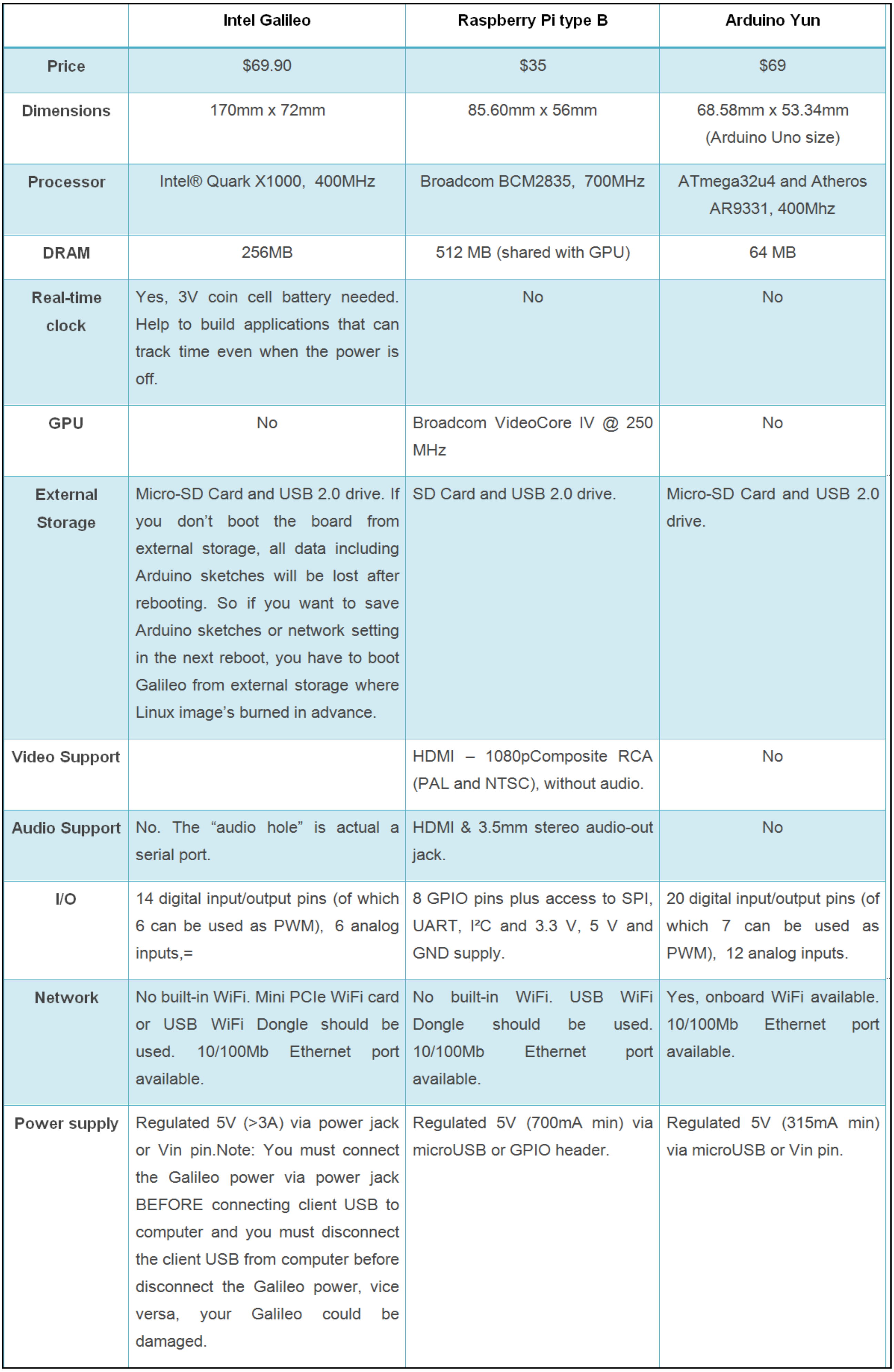
- Affordable cost
- Easy Availability in India
- Many addons are already available for RPi
- Small Size and portability
- Conitnuous improvement and ability to load normal Linux OS and there are many Linux Distributions exclusively for RPi and Windows 10 IoT Core is also possible to load in RPi

BeagleBone: BeagleBone Black is a low-cost, community-supported development platform for developers and hobbyists. Boot Linux in under 10 seconds and get started on development in less than 5 minutes with just a single USB cable.
Advantages of BeagleBone:

1. Setting up the BeagleBone Black is really simple
2. Provides huge number of supported interfaces
3. Tasks such as reading from external sensors, commanding actuators (such as motors or light systems), and networking are simpler and also more efficient.

Intel Galileo: Intel Galileo is the first in a line of Arduino-certified development boards based on Intel x86 architecture and is designed for the maker and education communities. Intel released two versions of Galileo, referred to as Gen 1 and Gen 2. These development boards are sometimes called "Breakout boards".
Intel Galileo combines Intel technology with support for Arduino ready-made hardware expansion cards (called "shields") and the Arduino software development environment and libraries The development board runs an open source Linux operating system with the Arduino software libraries, enabling re-use of existing software, called "sketches". The sketch runs every time the board is powered. Intel Galileo can be programmed through OS X, Microsoft Windows and Linux host operating software. The board is also designed to be hardware and software compatible with the Arduino shield ecosystem.

Comparision Intel Galileo, Raspberry Pi And Arduino



	Intel Galileo	Raspberry Pi type B	Arduino Yun
Price	$69.90	$35	$69
Dimensions	170mm x 72mm	85.60mm x 56mm	68.58mm x 53.34mm (Arduino Uno size)
Processor	Intel® Quark X1000, 400MHz	Broadcom BCM2835, 700MHz	ATmega32u4 and Atheros AR9331, 400Mhz
DRAM	256MB	512 MB (shared with GPU)	64 MB
Real-time clock	Yes, 3V coin cell battery needed. Help to build applications that can track time even when the power is off.	No	No
GPU	No	Broadcom VideoCore IV @ 250 MHz	No
External Storage	Micro-SD Card and USB 2.0 drive. If you don't boot the board from external storage, all data including Arduino sketches will be lost after rebooting. So if you want to save Arduino sketches or network setting in the next reboot, you have to boot Galileo from external storage where Linux image's burned in advance.	SD Card and USB 2.0 drive.	Micro-SD Card and USB 2.0 drive.
Video Support		HDMI – 1080pComposite RCA (PAL and NTSC), without audio.	No
Audio Support	No. The "audio hole" is actual a serial port.	HDMI & 3.5mm stereo audio-out jack.	No
I/O	14 digital input/output pins (of which 6 can be used as PWM), 6 analog inputs,=	8 GPIO pins plus access to SPI, UART, I²C and 3.3 V, 5 V and GND supply.	20 digital input/output pins (of which 7 can be used as PWM), 12 analog inputs.
Network	No built-in WiFi. Mini PCIe WiFi card or USB WiFi Dongle should be used. 10/100Mb Ethernet port available.	No built-in WiFi. USB WiFi Dongle should be used. 10/100Mb Ethernet port available.	Yes, onboard WiFi available. 10/100Mb Ethernet port available.
Power supply	Regulated 5V (>3A) via power jack or Vin pin.Note: You must connect the Galileo power via power jack BEFORE connecting client USB to computer and you must disconnect the client USB from computer before disconnect the Galileo power, vice versa, your Galileo could be damaged.	Regulated 5V (700mA min) via microUSB or GPIO header.	Regulated 5V (315mA min) via microUSB or Vin pin.

Linux distribution	Pre-installed very light distribution and it's booted from 8MB on-board flash memory. Nothing's saved after rebooting. To access WiFi, Python, Node.js, SSH, openCV… you need to install Linux Image for Galileo from Intel website.	Raspbian, Debian, Fedora, ARCH Linux ARM.	Pre-installed OpenWrt-Yun, based on OpenWrt.
How to find IP address to telnet/SSH in Terminal	It's little bit tricky to find IP address in Galileo because it doesn't support display interface. But there's hack for that, Telnet from Arduino sketch (using system() command in Arduino sketch) is a solution. I will show you how in the next post.The other workaround is to boot Galileo from micro-SD card, use an Arduino sketch to start the LAN connection and fix IP address in etc/networks/interfaces. Booting Galileo from micro SD card is able to keep saving your network configuration in the next boot.	Plugin HDMI monitor to RPi and use ifconfig in Terminal.Set hostname for RPi and use IP scanner software to get IP address.	It's quite easy for us. What we have to do is enter http://arduino.local in the address bar and get the IP address there.
Applications	When you need to combine popular language (Python, Node.js…) on Linux with Arduino shields. The project that's more hardware based is appropriate with Galileo than RPi.	Suite for graphics and media intensive applications (XBMC distro is recommended) and camera-related application (used with camera module)Since it's a full-fat Linux computer, it's also suitable for software heavy application than hardware based.Internet based application is also best match for using RPi because you can run a bunch of web server services on RPi.	Suite for Internet based appication as it has own built-in Wi-Fi module and connect with Internet services of Temboo.
IDE & language required	Familiar Arduino IDE. You still need some basic Python, Node.js or Linux system skills if you want to take full advantage of Linux power in Galileo.Arduino is on emulation mode within SoC Quark X1000. Talking with Linux through system() command. Basically, emulating MCU in software might slow down its own performance. But you have more memory for your application instead of running out of memory occasionally when compiling with Arduino Uno, Mega…	Many many choices. Imagining the Linux computer beside you.	Familiar Arduino IDE. Arduino is separate with Linux, there's a Bridge library to help them communicate.

Table 2.1: Comparision Intel Galileo, Raspberry Pi And Arduino

Program : **B.E**

Subject Name: **Internet of Things**

Subject Code: **IT-8004**

Semester: **8th**

Unit III: IoT PROTOCOLS: IoT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN – Network Layer: IP versions, Constrained Nodes and Constrained Networks, Zigbee – Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks – Application Transport Methods: Supervisory Control and Data Acquisition – Application Layer Protocols: CoAP and MQTT

IoT PROTOCOLS:

IoT Access Technologies: IoT Access Technology is spread across licensed and unlicensed spectrum and there are several number of Radio technologies. At high this access can be classified in two categories:

1. Non –Cellular Technologies
2. Cellular Technologies

Each of the technologies available for IoT connectivity has its own advantages and disadvantages. However, the range of IoT connectivity requirements – both technical and commercial – means cellular technologies can provide clear benefits across a wide variety of applications. While choosing technology following requirement needs to be considered

- Global Reach
- Matured Ecosystem
- Diverse and Secure
- Scalable and QoS support
- Low Total Cost of Ownership (TOC)

In terms of global reach, cellular networks already cover 90 percent of the world's population. WCDMA and LTE are catching up, but GSM will offer superior coverage across the Globe. The cellular mobile industry represents a huge and mature ecosystem, incorporating chipset, device and network equipment vendors, operators, application providers and others.

The global cellular ecosystem is governed by the 3GPP standardization forum, which guarantees broad industry support for future development.

When it comes to scalability, cellular networks are built to handle massive volumes of mobile broadband traffic; the traffic from most IoT applications will be relatively small and easily absorbed. Cellular connectivity offers the diversity to serve a wide range of applications with varying requirements within a network.

QoS mechanisms is essential for many IoT applications and cellular systems have mature QoS functionality, and this enables critical IoT applications to be handled together with traffic from sensors, voice and mobile-broadband traffic on the same carrier.

Traditionally, the security mechanisms of cellular networks have been based on a physical SIM attached to the device, referred to as a Universal Integrated Circuit Card (UICC). This has also enabled roaming between operators, which has been one of the main factors behind the huge success of mobile networks. The SIM will also be essential in future IoT applications, with SIM functionality embedded in the chipset (eUICC) or handled as a soft-SIM solution running in a trusted run-time environment of the module. One network connecting the whole diversifying IoT market will guarantee the lowest possible TOC as well as fast time to market.

Physical and MAC layers

The Institute of Electrical and Electronics Engineers (IEEE) committee 802 defines physical and data link technologies. The IEEE decomposes the OSI link layer into two sublayers:

1. The media-access control (MAC) layer, sits immediately on top of the physical layer (PHY), and implements the methods used to access the network, typically the carriersense multiple access with collision detection (CSMA/CD) used by Ethernet and the carrier-sense multiple access with collision avoidance (CSMA/CA) used by IEEE wireless protocols.
2. The logical link control layer (LLC), which formats the data frames sent over the communication channel through the MAC and PHY layers. IEEE 802.2 defines a frame format that is independent of the underlying MAC and PHY layers, and presents a uniform interface to the upper layers.

The physical layer: The physical layer is the initial layer in the reference model. The physical layer (PHY) ultimately provides the data transmission service, as well as the interface to the physical layer management entity. For IoT applications, the main characteristics of the **physical layer** that need to be considered are modulation, data rate, transmission mode, and channel encoding.

- **Modulation.** The nature of IoT applications, some involve infrequent data transmission that need low-cost low-complexity devices, preclude the use of high-order modulation or advanced channel coding like trellis-coded modulation. Unless mandatory, due to a harsh radio environment with narrowband interferers or regulatory constraints, spread spectrum, e.g., Direct Sequence Spread Spectrum (DSSS), is to be avoided as it increases the channel bandwidth, requiring a more costly and power-consuming RF frontend, with no data rate improvement.
- **Data rate**. IoT applications need to mix very low data rate requirements, e.g., a sensor or an actuator with limited data size either uplink or downlink, with more demanding requirements, e.g., a 6-inch 3-color ePaper display in a home that updates the daily weather forecast or the shopping list, easily amounting to more than 196 kB worth of data. Yet even for small data amounts, a carefully chosen higher data rate actually improves power-consumption thanks to shorter transmission time and reduced probability of collision. On the higher end, the transceiver complexity and power increase do not improve the actual useable throughput, as the overhead of packet acknowledgement and packet processing time become the bottleneck.
- **Transmission mode.** Full duplex communication is challenging, as it requires good isolation and does not allow for resource sharing between transmit and receive. Full duplex also typically involves different frequencies for downlink and uplink. Since the radio resource is a scarce resource, half-duplex is therefore selected, preferably on the same radio channel.
- **Channel coding.** There is the potential for improving link quality and performance with a limited complexity increase by using (adaptive) channel coding together with Automatic Repeat-Request (ARQ) retry mechanism. As of today, this is considered optional due to complexity-cost-performance trade-offs achieved with current technologies.

The MAC layer: The medium access control (MAC) enables the transmission of MAC frames through the use of the physical channel. Besides the data service, it offers a management interface and itself manages access to the physical channel. For IoT applications, the main characteristics of the media access layer control (MAC) that need to be considered are multiple access, synchronization, and network topology.

- **Multiple Access**. Looking back at decades of successful cellular system deployment, one can safely conclude that TDMA is a good fit for the IoT. TDMA is suited for low-power operation with a decent number of devices, as it allows for optimal scheduling of inactive periods. Hence, TDMA is selected for multiple access in the MAC layer.
- **Synchronization**. In IoT applications, there are potentially a very large number of power-sensitive devices with moderate throughput requirements. In such a configuration, it is essential to maintain a reasonably consistent time base across the entire network and potentially across different networks. Given that throughput is not the most critical requirement, it is suitable to follow a beacon-enabled approach, with a flexible beacon period to accommodate different types of services.
- **Network topology**. Mobile networks using a cellular topology have efficiently been servicing a large number of devices with a high level of security and reliability, e.g., 5,000+ per base station for LTE in urban areas. This typology is based on a star topology in each cell, while the cells are connected in a hierarchical tree in the network backhaul.

IEEE 802.15.4 network topologies

There are two main forms of network topology that can be used within IEEE 802.15.4. These network topologies may be used for different applications and offer different advantages.

The two IEEE 802.15.4 network topologies are:

Star topology: As the name implies the start format for an IEEE 802.15.4 network topology has one central node called the PAN coordinator with which all other nodes communicate.

Peer to Peer network topology: In this form of network topology, there is still what is termed a PAN coordinator, but communications may also take place between different nodes and not necessarily via the coordinator.

SECURITY OF IEEE 802.15.4

The key security requirements in IoT are:

Data Confidentiality: Data confidentiality is considered to be the most important issue. It is required to protect the data from disclosure.

Data Integrity: Keeping the data confidential does not protect it from external modifications. An adversary can always alter the data by adding some fragments or by manipulating the data within a packet. This packet can later be forwarded to the coordinator. Lack of data integrity mechanism is sometimes very dangerous

Data Authentication: It confirms the identity of the original source node. Apart from modifying the data packets, the intruder can also change a packet stream by integrating fabricated packets. The system must have the capability to verify the original source of data.

Data Freshness: The intrudar may sometimes capture data in transit and replay them later using the old key. Data freshness implies that the data is fresh and that no one can replay old messages. There are two types of data freshness: weak freshness, which guarantees partial data frames ordering but does not guarantee delay, and strong freshness, which guarantees data frames ordering as well as delay.

Secure Localization: Some IoT applications require accurate location. Lack of smart tracking mechanisms allow an attacker to send incorrect reports about the location either by reporting false signal strengths or by using replaying signals.

Availability: Availability implies efficient availability of information to the system. The attacker may target the availability of data by capturing or disabling a particular node or device. One of the best ways is to switch the operation of a node that has been attacked to another node in the network.

Secure Management: Secure management is required at the coordinator to provide key distribution to the nodes for encryption and decryption operation. In case of association and disassociation, the coordinator adds or removes the nodes in a secure manner.

Attacks on various layers of IoT Devices:

Layers	Attacks	Defenses
Physical	Jamming	Spread-spectrum, priority messages, region mapping, mode change
	Tampering	Tamper-proof, hiding
Link	Collision	Error correcting code
	Unfairness	Small frames
	Exhaustion	Rate limitation
Network	Neglect and greed	Redundancy, probing
	Homing	Encryption
	Misdirection	Egress filtering, authorization monitoring
	Black holes	Authorization, monitoring, redundancy
Transport	Flooding	Client Puzzles
	De-synchronization	Authentication

Table 3.1: Attacks on various layers of IoT Devices:

IEEE 802.15.4g and IEEE 802.15.4e

IEEE 802.15.4g and IEEE 802.15.4e are amendments of IEEE 802.15.4, respectively, dealing with physical layer and MAC layer requirements for applications such as SUN. IEEE 802.15.4g targeted at usage scenario of neighborhood area network (NAN), where utility meters are deployed outdoor and form mesh/ad hoc networks. Comparing with the baseline standard, such usage scenarios present more technical challenges due to harsher environment.

IEEE 802.15.4e intends to support a wide range of industrial and commercial applications that require lower latency, higher robustness, and deterministic behaviors. To achieve this, 802.15.4e has specified a number of mechanisms targeting different application domains with different features. On the other hand, IEEE 802.15.4e has also developed additional MAC functions to enhance the general capabilities, like low energy (LE), information element (IE), enhanced beacons (EB) and enhanced beacon requests (EBR), and so forth.

IEEE 1901.2a

This standard offers the flexibility to run any upper layer protocol. So, implementations of IPv6 6 LoWPAN and RPL IPv6 protocols are supported. These protocols enable the use of network layer routing to create mesh networks.

The IEEE 19.01.2a encryption and authentication are performed by AES. IEEE 1901.2a supports IEEE 802.15.9 key management protocol.

IEEE 802.11ah

Wi-Fi lacks sub-Ghz support for better signal penetration, low power for a battery powered nodes and the ability to support a large number of services. For these reasons, the IEEE 802.11 working group launched a tast group named IEEE 802.11ah to specify a sub-Ghz version of Wi Fi.

LoRaWAN

Low Power Wide Area Networks are well adapted for long range and battery powered endpoints. LPWA technologies open new business opportunities to both services providers and enterprises considering IOT solutions. LoRaWAN is a media access control (MAC) protocol for wide area networks. It is designed to allow low-powered devices to communicate with Internet-connected applications over long range wireless connections.

LoRaWAN can be mapped to the second and third layer of the OSI model. It is implemented on top of LoRa or FSK modulation in industrial, scientific and medical (ISM) radio bands.

The LoRaWAN specification defines three device types. All LoRaWAN devices must implement Class A, whereas Class B and Class C are extensions to the specification of Class A devices.

- Class A devices support bi-directional communication between a device and a gateway. Uplink messages (from the device to the server) can be sent at any time (randomly). The device then opens two receive windows at specified times after an uplink transmission. If the server does not respond in either of these receive windows, the next opportunity will be after the next uplink transmission from the device. The server can respond either in the first receive window, or in the second receive window, but should not use both windows.
- Class B devices extend Class A by adding scheduled receive windows for downlink messages from the server. Using time-synchronized beacons transmitted by the gateway, the devices periodically open receive windows.
- Class C devices extend Class A by keeping the receive windows open unless they are transmitting, as shown in the figure below. This allows for low-latency communication but is many times more energy consuming than Class A devices.

LoRaWN NETWORK LAYER

A LoRa network consists of several elements:

End points: The endpoints are the elements of the LoRa network where the sensing or control is undertaken. They are normally remotely located.

LoRa gateway : The gateway receives the communications from the LoRa endpoints and then transfers them onto the backhaul system. This part of the LoRa network can be Ethernet, cellular or any other telecommunications link wired or wireless. The gateways are connected to the network server using standard IP connections. On this way the data uses a standard protocol, but can be connected to any telecommunications network, whether public or private. In view of the similarity of a LoRa network to that of a cellular one, LoRaWAN gateways may often be co-located with a cellular base station. In this way they are able to use spare capacity on the backhaul network.

LoRa Network Server: The LoRa network server manages the network and as part of its function it acts to eliminate duplicate packets, schedules acknowledgement, and adapts data rates. In view of the way in which it can be deployed and connected, makes it very easy to deploy a LoRa network.

Remote computer: a remote computer can then control the actions of the endpoints or collect data from them - the LoRa network being almost transparent.

In terms of the actual architecture for the LoRa network, the nodes are typically in a star-of-stars topology with gateways forming a transparent bridge. These relay messages between end-devices and a central network server in the backend.

Program : **B.E**

Subject Name: **Internet of Things**

Subject Code: **IT-8004**

Semester: **8th**

Unit IV: Security **:** Understanding the risks, Modes of attack - Denial of Service Guessing the credentials , Getting access to stored credentials, Man in the middle , Sniffing network communication , Port scanning and web crawling ,Search features and wildcards ,Breaking ciphers , Tools for achieving security - Virtual Private Networks , X.509 certificates and encryption , Authentication of identities , Usernames and passwords , Using message brokers and provisioning servers ,Centralization versus decentralization.

UNDERSTANDING THE RISKS

There are many solutions and products marketed today under the label IoT that lack basic security architectures. It is very easy for a knowledgeable person to take control of devices for malicious purposes. Not only devices at home are at risk, but cars, trains, airports, stores, ships, logistics applications, building automation, utility metering applications, industrial automation applications, health services, and so on, are also at risk because of the lack of security measures in their underlying architecture

MODES OF ATTACK

There are different modes of attack that can be expected some of the most common forms of attack is provided here for IoT applications. Some of them are listed below:

DENIAL OF SERVICE

A Denial of Service (DoS) or Distributed Denial of Service (DDoS) attack is to make a service on the Internet crash or become unresponsive. The attack consists in making repetitive requests to a server until its resources gets exhausted. In a distributed version, the requests are made by many clients at the same time, which obviously increases the load on the target. The attack gets more effective and difficult to defend against when the attack is distributed and the target centralized, the attack gets less effective if the solution itself is distributed. To guard against this form of attack, you need to build decentralized solutions where possible. In decentralized solutions, each target's worth is less, making it less interesting to attack.

GUESSING THE CREDENTIALS

One way to get access to a system is to impersonate a client in the system by trying to guess the client's credentials. To make this type of attack less effective, each client and each device must have a long and unique, randomly generated, set of credentials. Never use default user credentials that are the same for many clients or devices or factory default credentials that are easy to reset. Furthermore, a limit must be set to the number of authentication attempts per time unit permitted by the system; also, an event must be logged whenever this limit is reached and from where to which credentials were used. This makes operators to detect systematic attempts to enter the system.

GETTING ACCESS TO STORED CREDENTIALS

One common way to enter a system is when user credentials are found somewhere else and reused. Often, people reuse credentials in different systems. There are various ways to avoid this risk from happening. One is to make sure that credentials are not reused in different devices or across different services and applications. Another is to randomize credentials, avoiding the desire to reuse memorized credentials. A third way is to never store actual credentials centrally, even encrypted if possible, and instead store hashed values of these credentials. This is often possible since authentication methods use hash values of credentials in their computations. Furthermore, these hashes should be unique to the current installation. Even though some hashing functions are vulnerable in such a way that a new string can be found that generates the same hash value, the probability that this string is equal to the original credentials is very less. And if the hash is computed uniquely for each installation, the probability that this string can be reused somewhere else is even more remote.

MAN IN THE MIDDLE

Another way to gain access to a system is to try and impersonate a server component in a system instead of a client. This is often referred to as a Man in the middle (MITM) attack. The reason for the middle part is that the attacker often does not know how to act in the server and simply forwards the messages between the real client and the server. In this process, the attacker gains access to confidential information within the messages, such as client credentials, even if the communication is encrypted. The attacker might even try to modify messages for their own purposes.

To avoid this type of attack, it's important for all clients to always validate the identity of the server it connects to. If it is a high-value entity, it is often identified using a certificate. This certificate can both be used to verify the domain of the server and encrypt the communication. Make sure this validation is performed correctly, and do not accept a connection that is invalid or where the certificate has been revoked, is self-signed, or has expired. Another thing to remember is to never use an unsecure authentication method when the client authenticates itself with the server. If a server has been compromised, it might try to fool clients into using a less secure authentication method when they connect. By doing so, they can extract the client credentials and reuse them somewhere else. By using a secure authentication method, the server, even if compromised, will not be able to replay the authentication again or use it somewhere else. The communication is valid only once.

SNIFFING NETWORK COMMUNICATION

If communication is not encrypted, everybody with access to the communication stream can read the messages using simple sniffing applications, such as Wireshark. If the communication is point-to-point, this means the communication can be heard by any application on the sending machine, the receiving machine, or any of the bridges or routers in between. If a simple hub is used instead of a switch somewhere, everybody on that network will also be able to eavesdrop. If the communication is performed using multicast messaging service, as can be done in UPnP and CoAP, anybody within the range of the Time to live parameter (maximum number of router hops) can eavesdrop.

Remember to always use encryption if sensitive data is communicated. If data is private, encryption should still be used, even if the data might not be sensitive at first glance. A burglar can know if user is at home by simply monitoring temperature sensors, water flow meters, electricity meters, or light switches at home. Small variations in temperature alert to the presence of human beings. Change in the consumption of electrical energy shows whether somebody is cooking food or watching television. The flow of water shows whether somebody is drinking water, flushing a toilet, or taking a shower. No flow of water or a relatively regular consumption of electrical energy tells the burglar that nobody is at home. Light switches can also be used to detect presence, even though there are applications today that simulate somebody being home by switching the lights on and off.

PORT SCANNING AND WEB CRAWLING

Port scanning is a method where user systematically tests a range of ports across a range of IP addresses to see which ports are open and serviced by applications. This method can be combined with different tests to see the applications that might be behind these ports. If HTTP servers are found, standard page names and web-crawling techniques can be used to try to figure out which web resources lie behind each HTTP server. CoAP is even simpler since devices often publish well-known resources. Using such simple brute-force methods, it is relatively easy to find and exploit anything available on the Internet that is not secured.

To avoid any private resources being published unknowingly must be close all the incoming ports in any firewalls used. User should not use protocols that require incoming connections. Instead, use protocols that create the connections from inside the firewall. Any resources published on the Internet should be authenticated so that any automatic attempt to get access to them fails. It must be remembered that information that might seem trivial to an individual might be very interesting if collected en masse. This information might be coveted not only by teenage pranksters but by public relations and marketing agencies, burglars, and government agencies.

SEARCH FEATURES AND WILDCARDS

Now a day it is not difficult to find the identities of devices published on the Internet. For devices that use multicast communication, such as those using UPnP and CoAP, anybody can listen in and see who sends the messages. For devices that use single-cast communication, such as those using HTTP or CoAP, port-scanning techniques can be used. For devices that are protected by firewalls and use message brokers to protect against incoming attacks, such as those that use XMPP and MQTT, search features or wildcards can be used to find the identities of devices managed by the broker, and in the case of MQTT, even what they communicate.

User should always assume that the identity of all devices can be found, and that there's an interest in exploiting the device. For this reason, it's very important that each device authenticates any requests made to it if possible. Some protocols help more with this than others, while others make such authentication impossible.

XMPP only permits messages from accepted friends. The only thing the device needs to worry about is which friend requests to accept. This can be either configured by somebody else with access to the account or by using a provisioning server if the device cannot make such decisions by itself. The device does not need to worry about client authentication, as this is done by the brokers themselves, and the XMPP brokers always propagate the authenticated identities of everybody who send them messages. MQTT, on the other hand, resides in the other side of the spectrum. Here, devices cannot make any decision about who sees the published data or who makes a request since identities are stripped away by the protocol. The only way to control who gets access to the data is by building a proprietary end-to-end encryption layer on top of the MQTT protocol, thereby limiting interoperability. In between the two resides protocols such as HTTP and CoAP that support some level of local client authentication but lacks a good distributed identity and authentication mechanism. This is vital for IoT even though this problem can be partially solved in local intranets.

BREAKING CIPHERS

Many believe that by using encryption, data is secure. This is not the case, since the encryption is often only done between connected parties and not between end users of data. At most, such encryption safeguards from eavesdropping to some extent. But even such encryption can be broken, partially or wholly, with some effort.

Ciphers can be broken using known vulnerabilities in code where attackers exploit program implementations rather than the underlying algorithm of the cipher. This has been the method used in the latest spectacular breaches in code based on the OpenSSL library. To protect users from such attacks, user need to be able to update code in devices remotely, this is not always possible.

Other methods use irregularities in how the cipher works to figure out, partly or wholly, what is being communicated over the encrypted channel. This sometimes requires a considerable amount of effort. To safeguard against such attacks, it's important to realize that an attacker does not spend more effort into an attack than what is expected to be gained by the attack. By storing massive amounts of sensitive data centrally or controlling massive amounts of devices from one point, you increase the value of the target, increasing the interest of attacking it. On the other hand, by decentralizing storage and control logic, the interest in attacking a single target decreases since the value of each entity is comparatively lower. Decentralized architecture is an important tool to both mitigate the effects of attacks and decrease the interest in attacking a target. However, by increasing the number of participants, the number of actual attacks can increase, but the effort that can be invested behind each attack when there are many targets also decreases, making it easier to defend each one of the attacks using standard techniques.

TOOLS FOR ACHIEVING SECURITY

There are a number of tools that architects and developers can use to protect against malicious use of the system. Some of techniques are mentioned below.

VIRTUAL PRIVATE NETWORKS

A method that is often used to protect unsecured solutions on the Internet is to protect user using Virtual Private Networks (VPNs). Often, traditional M2M solutions working well in local intranets need to expand across the Internet. One way to achieve this is to create such VPNs that allow the devices to believe they are in a local intranet, even though communication is transported across the Internet. Even though transport is done over the Internet, it's difficult to see this as a true IoT application. It's rather a M2M solution using the Internet as the mode of transport. Because telephone operators use the Internet to transport long distance calls, it doesn't make it Voice over IP (VoIP). Using VPNs might protect the solution, but it completely eliminates the possibility to interoperate with others on the Internet, something that is seen as the biggest advantage of using the IoT technology.

X.509 CERTIFICATES AND ENCRYPTION

We've mentioned the use of certificates to validate the identity of high-value entities on the Internet. Certificates allow you to validate not only the identity, but also to check whether the certificate has been revoked or any of the issuers of the certificate have had their certificates revoked, which might be the case if a certificate has been compromised. Certificates also provide a Public Key Infrastructure (PKI) architecture that handles encryption. Each certificate has a public and private part.

The public part of the certificate can be freely distributed and is used to encrypt data, whereas only the holder of the private part of the certificate can decrypt the data. Using certificates incurs a cost in the production or installation of a device or item. They also have a limited life span, so they need to be given either a long lifespan or updated remotely during the life span of the device. Certificates also require a scalable infrastructure for validating them. For these reasons, it's difficult to see that certificates will be used by other than high-value entities that are easy to administer in a network. It's difficult to see a cost-effective, yet secure and meaningful, implementation of validating certificates in low-value devices such as lamps, temperature sensors, and so on, even though it's theoretically possible to do so.

AUTHENTICATION OF IDENTITIES

Authentication is the process of validating whether the identity provided is actually correct or not. Authenticating a server might be as simple as validating a domain certificate provided by the server, making sure it has not been revoked and that it corresponds to the domain name used to connect to the server. Authenticating a client might be more involved, as it has to authenticate the credentials provided by the client. Normally, this can be done in many different ways. It is vital for developers and architects to understand the available authentication methods and how they work to be able to assess the level of security used by the systems they develop.

Some protocols, such as HTTP and XMPP, use the standardized Simple Authentication and Security Layer (SASL) to publish an extensible set of authentication methods that the client can choose from. This is good since it allows for new authentication methods to be added. But it also provides a weakness: clients can be tricked into choosing an unsecure authentication mechanism, thus unwittingly revealing their user credentials to an impostor. Make sure clients do not use unsecured or obsolete methods, such as PLAIN, BASIC, MD5-CRAM, MD5-DIGEST, and so on, even if they are the only options available. Instead, use secure methods such as SCRAM-SHA-1 or SCRAM-SHA-1-PLUS, or if client certificates are used, EXTERNAL or no method at all. If you're using an unsecured method anyway, make sure to log it to the event log as a warning, making it possible to detect impostors or at least warn operators that unsecure methods are being used.

Other protocols do not use secure authentication at all. MQTT, for instance, sends user credentials in clear text (corresponding to PLAIN), making it a requirement to use encryption to hide user credentials from eavesdroppers or client-side certificates or pre-shared keys for authentication. Other protocols do not have a standardized way of performing authentication. In CoAP, for instance, such authentication is built on top of the protocol as security options. The lack of such options in the standard affects interoperability negatively.

USING MESSAGE BROKERS AND PROVISIONING SERVERS

Using message brokers can greatly enhance security in an IoT application and lower the complexity of implementation when it comes to authentication, as long as message brokers provide authenticated identity information in messages it forwards. In XMPP, all the federated XMPP servers authenticate clients connected to them as well as the federated servers themselves when they intercommunicate to transport messages between domains. This relieves clients from the burden of having to authenticate each entity in trying to communicate with it since they all have been securely authenticated. It's sufficient to manage security on an identity level.

Unfortunately, not all protocols using message brokers provide this added security since they do not provide information about the sender of packets. MQTT is an example of such a protocol.

CENTRALIZATION VERSUS DECENTRALIZATION

The effect of a breach of security is much smaller in the decentralized case. Even though there are more baskets, which might increase the risk of an attack, the expected gain of an attack is much smaller. This limits the motivation of performing a costly attack, which in turn makes it simpler to protect it against. When designing IoT architecture, consider the following points:

- Avoid storing data in a central position if possible. Only store the data centrally that is actually needed to bind things together.
- Distribute logic, data, and workload. Perform work as far out in the network as possible. This makes the solution more scalable, and it utilizes existing resources better.
- Use linked data to spread data across the Internet, and use standardized grid computation technologies to assemble distributed data to avoid the need to store and replicate data centrally.
- Use a federated set of small local brokers instead of trying to get all the devices on the same broker. Not all brokered protocols support federation, for example, XMPP supports it but MQTT does not.
- Let devices talk directly to each other instead of having a centralized proprietary API to store data or interpret communication between the two.
- Contemplate the use of cheap small and energy-efficient microcomputers such as the Raspberry Pi in local installations as an alternative to centralized operation and management from a datacenter.

Program : **B.E**

Subject Name: **Internet of Things**

Subject Code: **IT-8004**

Semester: **8th**

Unit V: IoT Applications :Home Automation- Smart Appliances , Smoke/ Gas Detection, Cities – Smart Parking ,Smart Lighting , Smart Road , Health and Lifestyle- Health and fitness monitoring, Retail- Smart Payments.
Case Studies: Smart city streetlights: - control and monitoring

IOT APPLICATIONS:

HOME AUTOMATION- SMART APPLIANCES

Modern hoes have a number of appliances such as TVs, refrigerators, music systems, washer, etc. Managing and controlling these appliances can be cumbersome, with each appliance having its own controls or remote controls. Smart appliances make the management easier and also provides status information to the users remotely, for example smart washers that can be controlled remotely and notify when the washing cycle is complete. Smart thermostats allow controlling the temperature remotely and can learn the user preferences. Smart refrigerators can keep track of the items stored and send updates to the users when an item is low on stock.

Smart TVs allow users to search and stream videos and movies from the internet on a local storage drive, search TV channel schedules and fetch news, weather updates and other content from the internet. OpenRemote is an open source automation platform for homes and buildings. Open remote is platform agnostic and works with standard hardware. With OpenRemote, users can control various appliances using mobile or web applications. Open Remote comprises of three components - a Controller that manages scheduling and runtime integration between devices, a Designer that allows you to create both configurations for the controller and create user interface designs and control panels that allows interacting with the devices and contrail them. An IoT based appliances system uses a smart central controller to set up a wireless sensor and actuator network and control modules for appliances.

SMOKE/ GAS DETECTION

Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire. Smoke detectors use optical detection, ionization or air sampling techniques to detect smoke. Alerts raised by smoke detectors can e in the form of signals to a fire alarm system. Gas detectors can detect the pressure of harmful gases such as Carbon Monoxide, Liquid Petroleum Gas, etc. A smart smoke/gas detector can raise alerts in human voice describing where the problem is, send or an SMS or email to the user or the local fire safety department and provide visual feedback on its status (healthy, battery low, etc.)

CITIES – SMART PARKING

Smart Parking Finding a parking space during rush hours in crowded cities can be time consuming and frustrating. Furthermore, drivers blindly searching for parking spaces create additional traffic congestion. Smart parking makes the search for parking space easier and convenient for drivers. Smart parking are powered by IoT systems that detect the number of empty parking slots and send the information over the Internet to smart parking application back-ends. These applications can be accessed by the drivers from smart-phones, tablets and in-car navigation systems. In smart parking, sensors are used for each parking slot, to detect whether the slot is empty or occupied. This information is aggregated by a local controller and then sent over the Internet to the database.

SMART LIGHTING

Smart lighting systems for roads, parks and buildings can help in saving energy. According to an IEA report, lighting is responsible for 19% of global electricity use and around 6% of global greenhouse gas emissions. Smart lighting allows lighting to be dynamically controlled and also adaptive to the ambient conditions. Smart lights connected to the Internet can be controlled remotely to configure lighting schedules and lighting intensity. Custom lighting configurations can be set for different situations such as a foggy day, a festival, etc. Smart lights equipped with sensors can communicate with other lights and exchange information on the sensed ambient conditions to adapt the lighting.

SMART ROADS

Smart roads equipped with sensors can provide information on driving conditions, travel time estimates and alerts in case of poor driving conditions, traffic congestions and accidents. Such information can help in making the roads safer and help in reducing traffic jams. Information sensed from the roads can be communicated via Internet to cloud-based applications and social media and disseminated to the drivers who subscribe to such applications. A distributed and autonomous system of sensor network nodes for improving driving safety on public roads in proposed. The system can provide the drivers and passengers with a consistent view of the road situation a few hundred meters ahead of them or a few dozen miles away, so that they can react to potential dangers early enough.

HEALTH AND LIFESTYLE- HEALTH AND FITNESS MONITORING

Health monitoring systems use a network of sensors to monitor the vibration levels in the structures such as bridges and buildings. The data collected from these sensors is analyzed to assess the health of the structures. By analyzing the data it is possible to detect cracks and mechanical breakdowns, locate the damages to a structure and also calculate the remaining life of the structure. Using such systems, advance warnings can be given in the case of imminent failure of the structure. Since health monitoring systems use large number of wireless sensor nodes which are powered by traditional batteries, researchers are exploring energy harvesting technologies to harvesting ambient energy, such as mechanical vibrations, sunlight, and wind.

RETAIL- SMART PAYMENTS

Smart payment solutions such as contact-less payments powered by technologies such as Near field communication (NFC) and Bluetooth. Near field communication (NFC) is a set of standards for smart-phones and other devices to communicate with each other by bringing them into proximity or by touching them. Customers can store the credit card information in their NFC-enabled smart-phones and make payments by bringing the smart-phones near the point of sale terminals. NFC maybe used in combination with Bluetooth, where NFC (which offers low speeds) initiates initial pairing of devices to establish a Bluetooth connection while the actual data transfer takes place over Bluetooth.

CASE STUDIES:

SMART CITY STREETLIGHTS: - CONTROL AND MONITORING

Reference for the case study recommended for students:
https://enterpriseiotinsights.com/20170725/channels/fundamentals/20170725channelsfundamentals5-smart-lighting-case-studies-tag23-tag99

Cities and companies around the world are taking advantage of the implementation of smart lighting solutions. These innovative lighting solutions allows firms and city governments to reduce energy costs as well as to implement a wide range of new solutions.

Chicago

U.S. company Ameresco has recently announced it has contracted with the city of Chicago as part of the city's smart street lighting project to modernize its infrastructure. Ameresco said that the project is believed to be the largest city-led wireless smart street light program in the U.S., and will connect more than 250,000 street light fixtures across Chicago.

The four-year modernization project is expected to transform Chicago's street light system by replacing approximately 85% of the city's existing street lights with smart LEDs.

The new smart LED street lights will be owned and operated by the city of Chicago, supported by Silver Spring Networks' managed services and its streetlight. Vision control and management system (CMS) software.

The new LED street lights are expected to consume between 50 and 75% less electricity than the city's existing lighting infrastructure. Silver Spring's IPv6 platform will enable Chicago to remotely dim or brighten street lights as needed, as well as to remotely monitor street lights for proactive maintenance and faster repairs if failures do occur. The smart street light infrastructure will also be integrated into Chicago's 311 system.

Paris

A clear example of the benefits of the implementation of smart street lights can be seen in Paris. In order to reduce public lighting energy consumption, the city government had selected Silver Spring to implement project pilot including integrated smart street lighting, traffic signal controls, and an IPv6-based multi-application network to achieve immediate savings, strengthen the communications fabric and reduce risk. For this specific project, the U.S Company had expanded the functionality of its smart infrastructure platform to support smart city solutions such as intelligent street lighting, traffic signal control, and electric vehicle charging, among others.

Georgia

AT&T is also working in a number of smart street lighting initiatives across the country. The telco uses GE's IoT sensors, which are placed in the street lights and provide key information about traffic, crowds, crime, and air quality.

In January 2017, AT&T teamed with Current, powered by GE and Georgia Power, to test intelligent lighting solutions in Atlanta. The companies are using the AT&T smart cities framework as the foundation to add intelligent lighting solutions throughout the city.

The City of Atlanta and Georgia Power will be piloting Current's new IoT sensor platform for cities and installing 1,000 wirelessly controlled LED lights.

Smart poles

Dutch firm Philips and Swedish vendor Ericsson are cooperating on telco-integrated street lighting infrastructure. With co-created Philips lightpole site, the Dutch firm provides mobile broadband connectivity through smart street lighting. The pilots, which take place in Los Angeles and San Jose, California, have taken a major step in creating a connected smart city.

"Smart poles not only serve as an important connected light source which can be remotely managed, they house technology to improve mobile network performance across the city," Philips said.

Additionally, smart poles will enable the densification of mobile wireless operator's networks, offering providers new possibilities to find the right site location. Because street light poles are ubiquitous in urban landscapes, mobile broadband infrastructure can be scaled beyond traditional sites, a key enabler for evolving heterogeneous networks. As a result, operators can improve data coverage and capacity for residents, visitors, and businesses, enabling an enhanced Mobile Broadband user experience.

American Eagle

American Eagle turned to GE for a large-scale smart lighting solution for its distribution center in Hazleton, Pennsylvania, in order to obtain energy and maintenance savings.

Before choosing GE, American Eagle was faced with a 14,000 fluorescent lamp installation for the factory's pick-module alone. Because fluorescent lamps have shorter shelf life than light emitting diodes, AEO would have required a full-time maintenance employee to replace tubes. Identifying GE's LED solutions as a lighting choice took them down the path to Leadership in Energy and Environmental Design silver certification, while improving light output and cost savings.

After determining LED smart lighting was best for the pick-module areas, the facility management team reviewed other areas – inside the offices and warehouse space, as well as the exterior of the facility – to identify ways LEDs could help them achieve LEED certification.

The distribution center opened with 7,200 GE Albeo ALC4-Series fixtures mounted and arranged to illuminate conveyor belts and pick-module areas. These were designed to save energy by using an "on/off" dimming schedule when associates are not in factory aisles. GE's Albeo ABV1 LED high bay fixtures were installed in a 200,000-square-foot section as part of a second phase of construction. GE provided lighting in the warehouse with T5 fluorescent tubes, Lumination ET Series recessed LED troffers and LED downlights in the building's office space as well as some Evolve LED lights and wall packs for the parking lot and outdoors.