CBCS SCHEME

USN | | | | | | | | | | |

18CS81

Eighth Semester B.E. Degree Examination, July/August 2022
Internet of Things

Time: 3 hrs. Max. Marks: 100

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

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

Module-1

1 a. What is Internet of Things (IoT)? Explain Genesis of IoT, with a neat diagram. **(08 Marks)**
b. Compare IT and OT Networks with different criterion. **(08 Marks)**
c. Explain different challenges faced by IoT. **(04 Marks)**

OR

2 a. Explain One M2M IoT Standardized Architecture, with a neat diagram. **(10 Marks)**
b. What is a Fog Node? Explain characteristics of Fog Computing Model. **(05 Marks)**
c. Illustrate various access technologies with respect to distances in core IoT functional stack. **(05 Marks)**

Module-2

3 a. Explain any 5 ways to group sensors into different categories. **(05 Marks)**
b. List any two advantages of Wireless based solution. Illustrate with a neat diagram, the interaction of Sensors and Actuators, with the Physical World. **(07 Marks)**
c. What is a Smart Object? Explain its characteristics. **(08 Marks)**

OR

4 a. Explain the following key factors involved in connecting smart objects to the network :
i) Range ii) Frequency bonds. **(10 Marks)**
b. Explain IEEE 802.15.4 IoT Access technology. **(10 Marks)**

Module-3

5 a. Explain the key advantages of IP suite for IoT. **(05 Marks)**
b. Explain 6 LOWPAN Protocol Header Compression and Fragmentation, with a neat diagram. **(08 Marks)**
c. Illustrate Routing Protocol for Low Power and Lossy Networks (RPL), with a neat diagram. **(07 Marks)**

OR

6 a. Explain the message format of the following protocols with a neat diagram :
i) Constrained Application Protocol (CoAP).
ii) Message Queuing Telemetry Transport (MQTT). **(10 Marks)**
b. Describe the Scheduling Management Mechanisms and forwarding Models and Supported by 6 TiSCH. **(10 Marks)**

Module-4

7 a. Explain different types of Data Analysis results with example. **(08 Marks)**
b. Distinguish between Supervised and Unsupervised Machine Learning. **(05 Marks)**
c. Explain Elements of Hadoop, with a neat diagram. **(07 Marks)**

OR

8 a. What is Apache Spark? Explain layers in Lambda Architecture, with a neat diagram. **(10 Marks)**

b. Explain OCTAVE Allegro steps and phases, with a neat diagram. **(10 Marks)**

Module-5

9 a. Explain the following with respect to Arduino Programming :
i) Structures ii) Functions iii) Variables iv) Flow control statements
v) Data type with example. **(10 Marks)**

b. Explain the steps to install operating system in the SD card of Raspberry Pi. Write a Python program to blink on LED. **(10 Marks)**

OR

10 a. Explain Key Verticals targeted in Smart Cities, with a neat diagram. **(10 Marks)**

b. Explain Smart City IoT Architecture, with a neat diagram. **(10 Marks)**

* * * * *

1 a. WHAT IS INTERNET OF THINGS(IoT)? EXPLAIN GENESIS OF IoT, WITH A NEAT DIAGRAM

- The basic premise and goal of IoT is to “connect the unconnected.”
- The world of IoT is broad and multifaceted.
- IoT is good to view it as an umbrella of various concepts, protocols, and technologies
- The age of IoT is often said to have started between the years 2008 and 2009.

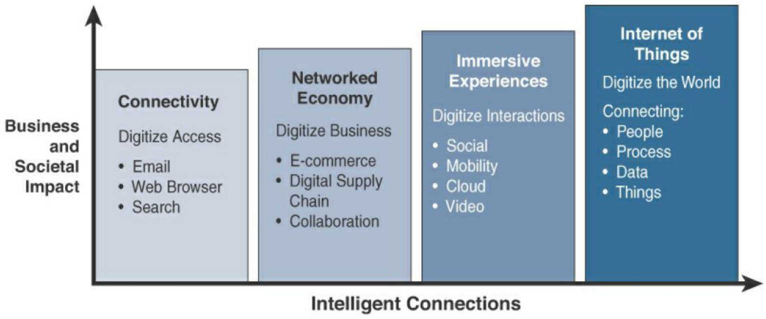


Figure 1-1 Evolutionary Phases of the Internet

1 b. COMPARE IT AND OT NETWORKS WITH DIFFERENT CRITERION

Criterion	Industrial OT Network	Enterprise IT Network
Operational focus	Keep the business operating 24x7	Manage the computers, data, and employee communication system in a secure way
Priorities	1. Availability 2. Integrity 3. Security	1. Security 2. Integrity 3. Availability
Types of data	Monitoring, control, and supervisory data	Voice, video, transactional, and bulk data
Security	Controlled physical access to devices	Devices and users authenticated to the network
Implication of failure	OT network disruption directly impacts business	Can be business impacting, depending on industry, but workarounds may be possible
Network upgrades (software or hardware)	Only during operational maintenance windows	Often requires an outage window when workers are not onsite; impact can be mitigated
Security vulnerability	Low: OT networks are isolated and often use proprietary protocols	High: continual patching of hosts is required, and the network is connected to Internet and requires vigilant protection

1 c. EXPLAIN DIFFERENT CHALLENGES FACED BY IoT

- Scale
- Security
- Privacy
- Big data & Data Analytics
- Interoperability

2 a. EXPLAIN ONE M2M IoT STANDARDIZED ARCHITECTURE, WITH A NEAT DIAGRAM

- It was created with a goal of accelerating Machine to machine applications & devices.
- It expanded to include IoT.
- One M2M was launched with a goal to promote efficient M2M communication systems and IoT.

- OneM2M Goal: create a common services layer.
- Its framework focuses on IoT services , applications, and platforms.
- It divides IoT functions into 3 major domains.
- It supports wide range of IoT technologies.

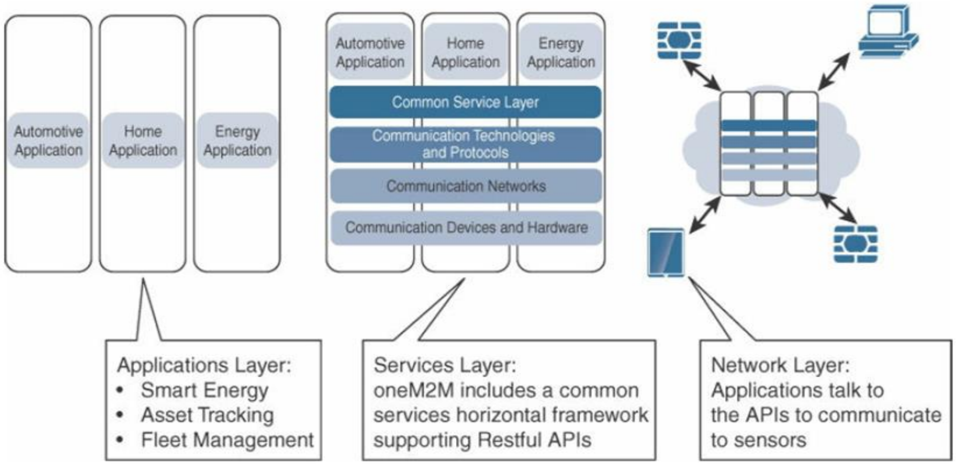


Figure 2-1 The Main Elements of the oneM2M IoT Architecture

2 b. WHAT IS FOG NODE? EXPLAIN CHARACTERISTICS OF FOG COMPUTING MODEL

- Any device with computing, storage, and network connectivity can be a ***fog node.***
- This introduces a new layer called "fog layer", to be placed in IoT Data Management and Compute Stack.

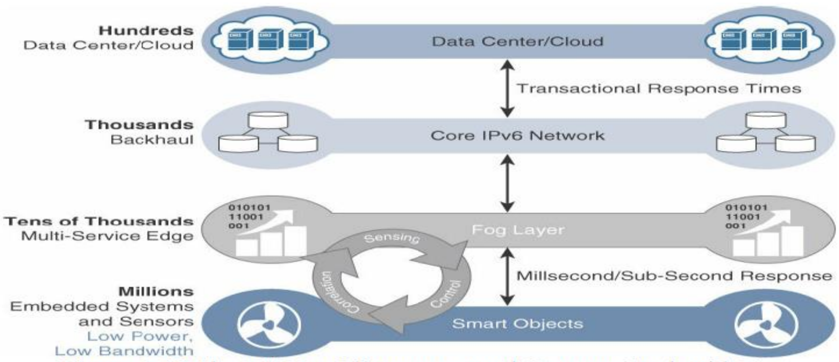


Figure 2-15 The IoT Data Management and Compute Stack with Fog Computing

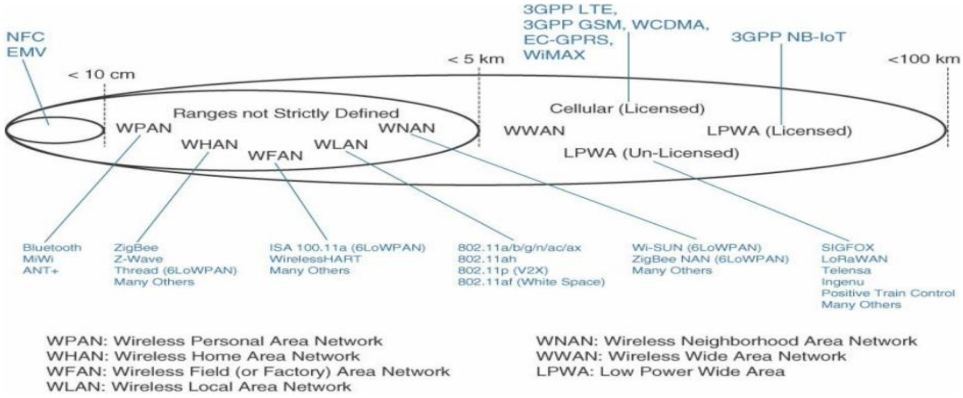
- ➢ Characteristic of fog computing are as follows:
- *Contextual location awareness and low latency:* The fog node delivers distributed computing.
- *Geographic distribution:* The services and applications targeted by the fog nodes demand widely distributed deployments.
- *Deployment near IoT endpoints:* Fog nodes are typically deployed in the presence of a large number of IoT endpoints.

- *Wireless communication between the fog and the IoT endpoint:* wireless access is the easiest way to achieve large scale communication.
- *Use for real-time interactions:* Important fog applications involve real-time interactions rather than batch processing.

2 c. ILLUSTRATE VARIOUS ACCESS TECHNOLOGIES WITH RESPECT TO DISTANCES IN CORE IoT FUNCTIONAL STACK

- ➢ PAN (personal area network): Scale of a few meters. This is the personal space around a person. A common wireless technology for this scale is Bluetooth.
- ➢ HAN (home area network): Scale of a few tens of meters. At this scale, common wireless technologies for IoT include ZigBee and Bluetooth Low Energy (BLE).

- NAN (neighborhood area network): Scale of a few hundreds of meters. The term NAN is often used to refer to a group of house units from which data is collected.
- FAN (field area network): Scale of several tens of meters to several hundred meters. FAN typically refers to an outdoor area larger than a single group of house units. The FAN is oftenseen as “open space” (and therefore not secured and not controlled). A FAN is sometimes viewed as a group of NANs, but some verticals see the FAN as a group of HANs or a groupof smaller outdoor cells.
- LAN (local area network): Scale of up to 100 m. This term is very common in networking, and it is therefore also commonly used in the IoT space when standard networking technologies (such as Ethernet or IEEE 802.11) are used.



3 a. EXPLAIN ANY 5 WAYS TO GROUP SENSORS INTO DIFFERENT CATEGORIES

There are a number of *ways to group and cluster sensors* into different categories, including the following:

■ **Active or passive**: Sensors can be categorized based on whether they produce an energy output and typically require an external power supply (active) or whether they simply receive energy and typically require no external power supply (passive).
■ **Invasive or non-invasive**: Sensors can be categorized based on whether a sensor is part of the environment it is measuring (invasive) or external to it (non-invasive).
■ **Contact or no-contact**: Sensors can be categorized based on whether they require physical contact with what they are measuring (contact) or not (no-contact).
■ **Absolute or relative:** Sensors can be categorized based on whether they measure on an absolute scale (absolute) or based on a difference with a fixed or variable reference value (relative).
■ **Area of application:** Sensors can be categorized based on the specific industry or vertical where they are being used.
■ **How sensors measure:** Sensors can be categorized based on the physical mechanism used to measure sensory input (for example, thermoelectric, electrochemical, piezoresistive, optic, electric, fluid mechanic, photoelastic).
■ **What sensors measure:** Sensors can be categorized based on their applications or what physical variables they measure.

3 b. LIST ANY TWO ADVANTAGES OF WIRELESS BASED SOLUTION. ILLUSTRATE WITH A NEAT DIAGRAM THE INTERACTION OF SENSORS AND ACTUATURS, WITH THE PHYSICAL WORLD.

The following are some advantages that a wireless-based solution offers:

Advantages:

- Greater deployment flexibility (especially in extreme environments or hard-to-reach places)
- Simpler scaling to a large number of nodes
- Lower implementation costs
- Easier long-term maintenance
- Effortless introduction of new sensor/actuator nodes

- Better equipped to handle dynamic/rapid topology changes

- ✓ Actuators are natural complements to sensors.
- ✓ Actuators receive some type of control signal (commonly an electric signal or digital command) that triggers a physical effect, usually some type of motion, force, and so on.

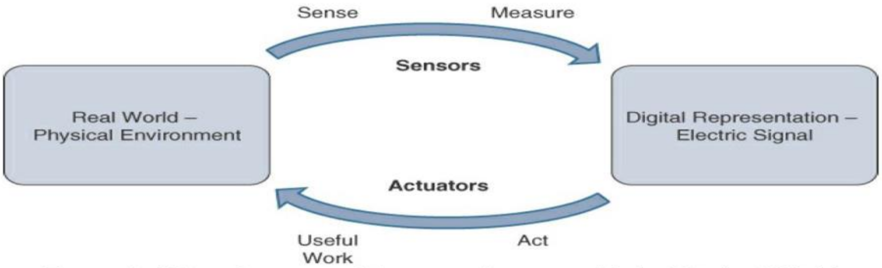


Figure 3-4 How Sensors and Actuators Interact with the Physical World

3 c. WHAT IS SMART OBJECT? EXPLAIN ITS CHARACTERISTICS.

- ✓ The term smart object, despite some semantic differences, is often used interchangeably with terms such as smart sensor, smart device, IoT device, intelligent device, thing, smart thing, intelligent node, intelligent thing and intelligent product.
- ✓ A smart object, is described as a device that has, at a minimum, one of the following four defining characteristics:
- ✓ **Processing unit:** A smart object has some type of processing unit for acquiring data, processing and analyzing sensing information received by the sensor(s), coordinating control signals to any actuators, and controlling a variety of functions on the smart object, including the communication and power systems.

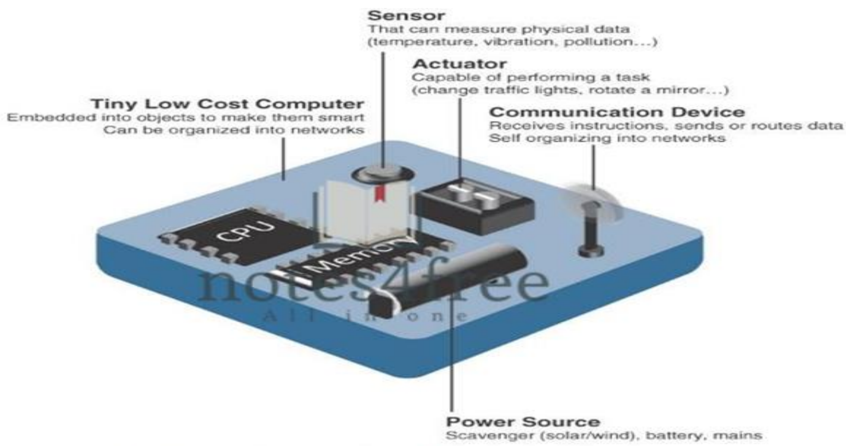


Figure 3-7 *Characteristics of a Smart Object*

- ✓ **Sensor(s) and/or actuator(s):** A smart object is capable of interacting with the physical world through sensors and actuators.
- ✓ **Communication device:** The communication unit is responsible for connecting a smartobject with other smart objects and the outside world (via the network).
- ✓ **Power source:** Smart objects have components that need to be powered. Interestingly, the most significant power consumption usually comes from the communication unit of a smart object.

4 a. EXPLAIN THE FOLLOWING KEY FACTORS INVOLVED IN CONNECTING SMART OBJECTS TO THE NETWORK :
i) RANGE ii) FREQUENCY BANDS

- ✓ Wireless communication is prevalent in the world of smart object connectivity because it allows smart objects to be mobile.

RANGE:

- ✓ **Short range:** Are alternative to a serial cable, supporting tens of meters of maximum distance between two devices. Ex- Bluetooth, Visible Light Communications.

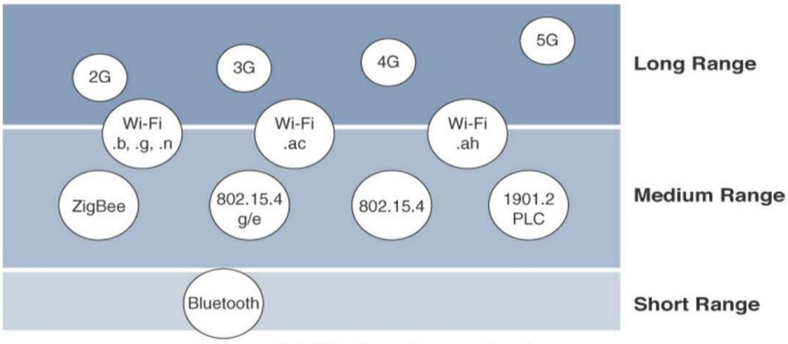


Figure 4-1 Wireless Access Landscape

- ✓ **Medium range:** This range is the main category of IoT access technologies.
- The maximum distance is generally less than 1 mile between two devices.
- Ex-Wi-Fi, Ethernet, Narrowband Power Line Communications.
- ✓ **Long range:** Distances greater than 1 mile between two devices require long-range technologies.
- Ex- Wi-Fi , Low-Power Wide-Area technologies.
- These technologies are ideal for battery-powered IoT sensors

FREQUENCY BAND:

- Telecommunication Union and the Federal Communications Commission, groups define the regulations and transmission requirements for various frequency bands.
- to utilize licensed spectrum users must subscribe to services when connecting their IoT devices.
- Most well-known ISM(Industrial scientific Medical) bands are:
- ✓ 2.4 GHz band as used by IEEE 802.11b/g/n Wi-Fi
- ✓ IEEE 802.15.1 Bluetooth
- ✓ IEEE 802.15.4 WPAN

4 b. EXPLAIN IEEE 802.15.4 IoT ACCESS TECHNOLOGY

- IEEE 802.15.4 is a wireless access technology for low-cost and low-data-rate devices that are powered or run on batteries.
- IEEE 802.15.4 is commonly found in the following types of deployments:
  - ✓ Home and building automation
  - ✓ Automotive networks
  - ✓ Industrial wireless sensor networks
  - ✓ Interactive toys and remote controls
- Criticisms of IEEE 802.15.4 often focus on its MAC reliability, unbounded latency, and susceptibility to interference and multipath fading

Standardization and Alliances:

- IEEE 802.15.4 or IEEE 802.15 Task Group 4 defines low-data-rate PHY and MAC layer specifications for wireless personal area networks (WPAN).
- The IEEE 802.15.4 PHY and MAC layers are the foundations for several networking protocol stacks.

Physical Layer:

- The 802.15.4 standard supports an extensive number of PHY options that range from 2.4 GHz to sub-GHz frequencies in ISM bands.
- original IEEE 802.15.4-2003 standard specified only 3 PHY options based on **Direct Sequence Spread Spectrum (DSSS)** modulation.

The original physical layer transmission options were as follows:

- 2.4 GHz, 16 channels, with a data rate of 250 kbps(operates worldwide)

- 915 MHz, 10 channels, with a data rate of 40 kbps
- 868 MHz, 1 channel, with a data rate of 20 kbps

MAC Layer: The 802.15.4 MAC layer performs the following tasks:

- Network beaconing for devices acting as coordinators (New devices use beacons to join an 802.15.4 network).
- PAN association and disassociation by a device.
- Device security.
- Reliable link communications between two peer MAC entities.

Topology:

- IEEE 802.15.4–based networks can be built as star, peer-to-peer, or mesh topologies.
- **Mesh networks tie together many nodes.**
- A minimum of 1 **FFD** acting as a PAN coordinator **is required** to deliver services that allow other devices to form a cell or PAN.

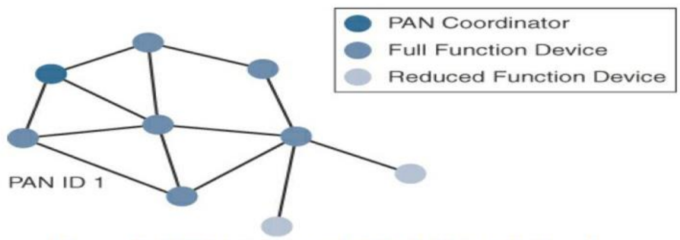


Figure 4-7 802.15.4 Sample Mesh Network Topology

Security:

- The IEEE 802.15.4 specification uses Advanced Encryption Standard (AES) with a 128-bit key length.
- In addition to encrypting the data, AES here, also validates the data.
- Enabling these security features for 802.15.4 changes the frame format slightly.

Competitive Technologies:

- A competitive radio technology that is different in its PHY and MAC layers is DASH7.
- DASH7 was used by US military forces for many years, mainly for logistics purposes.
- The current DASH7 technology offers low power consumption, a compact protocol stack, range up to 1 mile, and AES encryption.
- DASH7 is promoted by the DASH7 Alliance.

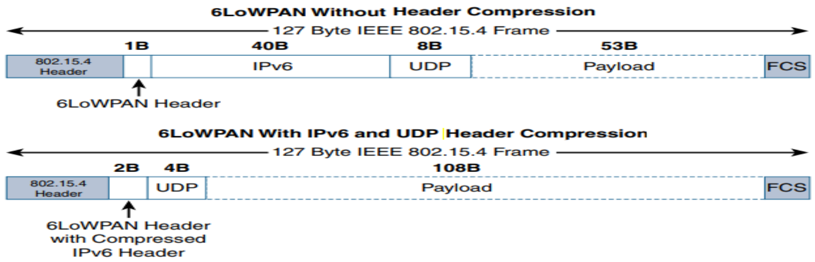
5 a. EXPLAIN KEY ADVANTAGES OF IP SUITE FOR IoT

- IP as the network layer transport for IoT is a foundational element.
- Key Advantages of IP:
  - ✓ Open and standards-based
  - ✓ Versatile
  - ✓ Ubiquitous
  - ✓ Scalable
  - ✓ Manageable and highly secure
  - ✓ Stable and resilient
  - ✓ Consumers' market adoption
  - ✓ The innovation factor

5 b. EXPLAIN 6LOWPAN PROTOCOL HEADER COMPRESSION AND FRAGMENTATION, WITH A NEAT DIAGRAM

Header Compression:

- It shrinks IPv6's 40-byte headers and UDP's 8- byte headers to 6 bytes in some cases.
- Figure highlights an example that shows the amount of reduction that is possible with 6LoWPAN header compression.

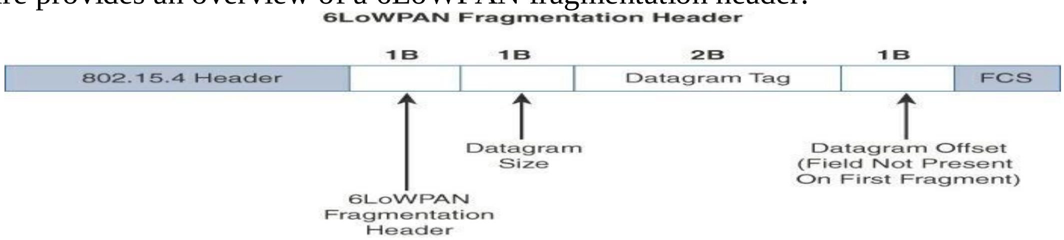


6LoWPAN Header Compression

Fragmentation:

- The maximum transmission unit (MTU) for an IPv6 network must be at least 1280 bytes.
- For IEEE 802.15.4, 127 bytes is the MTU.
- IPv6 with a much larger MTU, is carried inside the 802.15.4 frame with a much smaller one.
- To remedy this situation, large IPv6 packets must be fragmented across multiple 802.15.4 frames at Layer 2.
- The fragment header utilized by 6LoWPAN is composed of three primary fields:
- ➢ **Datagram Size,**
- ➢ **Datagram Tag,** and
- ➢ **Datagram Offset.**

- The 1-byte Datagram Size field specifies the total size of the unfragmented payload.
- Datagram Tag identifies the set of fragments for a payload.
- Datagram Offset field delineates how far into a payload a particular fragment occurs.
- Figure provides an overview of a 6LoWPAN fragmentation header.



5 c. ILLUSTRATE ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORKS(RPL), WITH A NEAT DIAGRAM

- In an RPL network, each node acts as a router and becomes part of a mesh network.
- Routing is performed at the IP layer
- Each node examines the received IPv6 packet and determines the next-hop destination based on the information contained in the IPv6 header.
- RPL is based on the concept of a directed acyclic graph (DAG).
- A DAG is a directed graph where no cycles exist.
- All of the edges are arranged in paths oriented toward and terminating at one or more root nodes.
- Figure shows a basic DAG

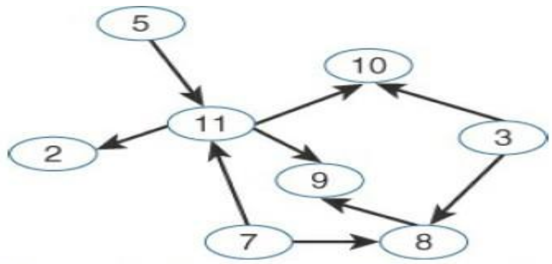


Figure 5-8 Example of a Directed Acyclic Graph (DAG)

- A basic RPL process involves building a destination-oriented directed acyclic graph (DODAG).
- A DODAG is a DAG rooted to 1 destination.
- In RPL, this destination occurs at a border router known as the DODAG root.

- Figure compares a DAG and a DODAG.
- From figure a DAG has multiple roots, whereas the DODAG has just one.

- In a DODAG, each node maintains up to three parents that provide a path to the root.
- Upward routes in RPL are discovered and configured using DAG Information Object (DIO) messages.
- Nodes establish downward routes by advertising their parent set toward the DODAG root using a Destination Advertisement Object (DAO) message.
- In the case of the *non-storing mode* of RPL, nodes sending DAO messages report their parent sets directly to the DODAG root (border router), and only the root stores the routing information.
- For *storing mode*, each node keeps track of the routing information that is advertised in the DAO messages.
- RPL messages, such as DIO and DAO, run on top of IPv6.
- As illustrated in Figure DAO and DIO messages move both up and down the DODAG, depending on the exact message type.

Objective Function (OF)

- It defines how metrics are used to select routes and establish a node's rank Whenever a node establishes its rank, it simply sets the rank to the current Minimum Expected Number of Transmissions (METX) among its parents.

Rank

- The rank is a rough approximation of how "close" a node is to the root.
- It helps avoid routing loops and the count-to-infinity problem.
- Nodes can only increase their rank when receiving a DIO message with a larger version number.

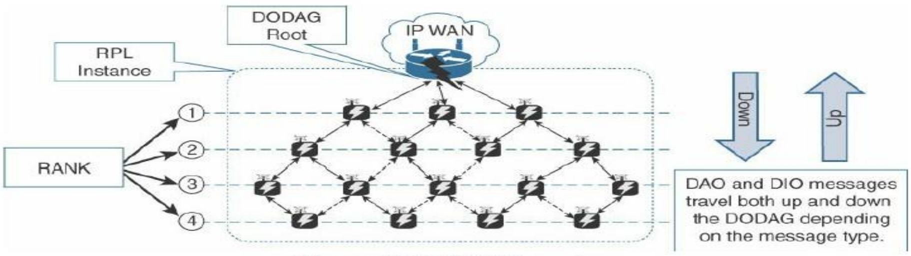


Figure 5-10 RPL Overview

RPL Headers

- RFC 6553 defines a new IPv6 option, known as the RPL option.
- The RPL option is carried in the IPv6 Hop- by-Hop header.
- The purpose of this header is to leverage data packets for loop detection in a RPL instance

RPL defines a large and flexible set of new metrics and constraints for routing in RFC 6551

1. Expected Transmission Count (ETX)
2. Hop Count
3. Latency
4. Link Quality Level
5. Link Color
6. Node State and Attribute
7. Node Energy
8. Throughput

6 a. EXPLAIN THE MESSAGE FORMAT FOR THE FOLLOWING PROTOCOLS WITH A NEAT DIAGRAM:

i) CONSTRAINED APPLICATION PROTOCOL

- ✓ Constrained Application Protocol (CoAP) resulted from the IETF Constrained RESTful Environments (CoRE) working group's efforts to develop a generic framework for resource-oriented applications targeting constrained nodes and networks.
- ✓ From a formatting perspective, a CoAP message is composed of a short fixed-length Header field (4 bytes), a variable-length but mandatory Token field (0–8 bytes), Options fields if necessary, and the Payload field. Figure 3.17 details the CoAP message format, which delivers low overhead while decreasing parsing complexity.

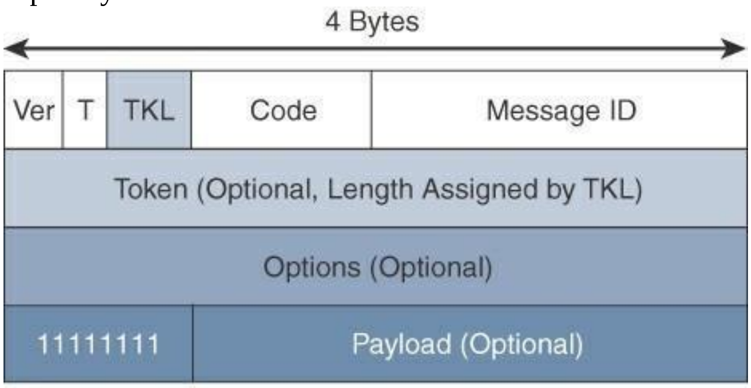
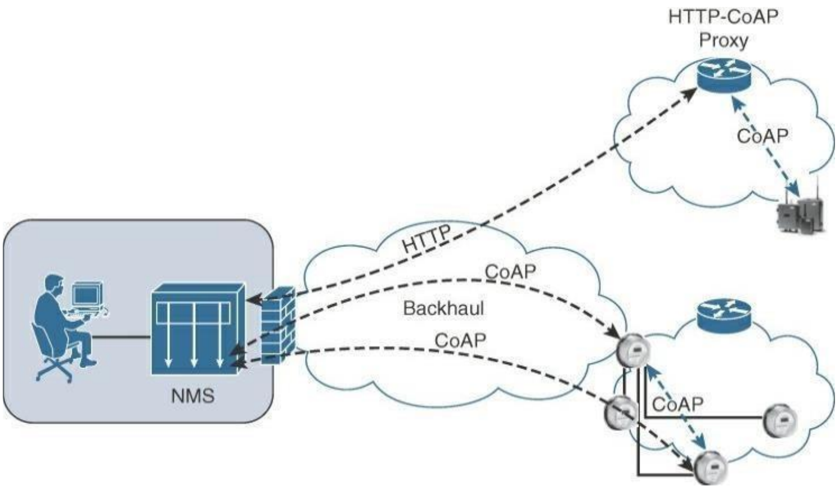


Figure *CoAP Message Format*

As illustrated in Figure 3.18, CoAP communications across an IoT infrastructure can take various paths.

- Connections can be between devices located on the same or different constrained networks or between devices and generic Internet or cloud servers, all operating over IP.
- Proxy mechanisms are also defined, and RFC 7252 details a basic HTTP mapping for CoAP. As both HTTP and CoAP are IP-based protocols, the proxy function can be located practically anywhere in the network, not necessarily at the border between constrained and non-constrained networks



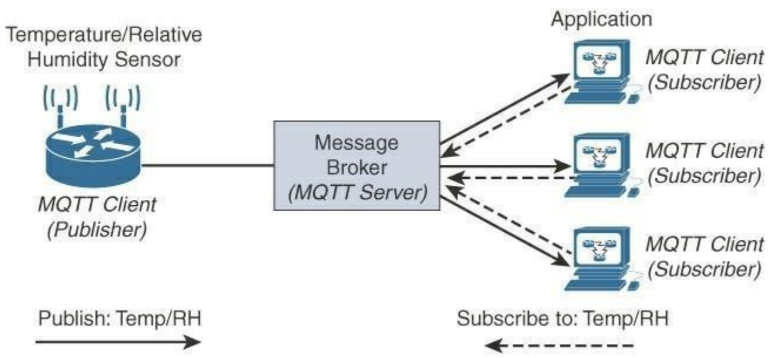
6 a. EXPLAIN THE MESSAGE FORMAT FOR THE FOLLOWING PROTOCOLS WITH A NEAT DIAGRAM:

ii) CONSTRAINED APPLICATION PROTOCOL

- At the end of the 1990s, engineers from IBM and Arcom (acquired in 2006 byEurotech) were looking for a reliable, lightweight, and cost-effective protocol to monitor and control a large number of sensors and their data from a central server location, as typically used by the oil and gas industries.

- Their research resulted in the development and implementation of the Message Queuing Telemetry Transport (MQTT) protocol that is now standardized by the Organization for the Advancement of Structured Information Standards (OASIS).
- Considering the harsh environments in the oil and gas industries, an extremely simple protocol with only a few options was designed, with considerations for constrained nodes, unreliable WAN backhaul communications, and bandwidth constraints with variable latencies.
- These were some of the rationales for the selection of a client/server and publish/subscribe framework based on the TCP/IP architecture, as shown in Figure.
- An MQTT client can act as a publisher to send data (or resource information) to an MQTT server acting as an MQTT message broker. In the example illustrated in Figure, the MQTT client on the left side is a temperature (Temp) and relative humidity (RH) sensor that publishes its Temp/RH data.
- The MQTT server (or message broker) accepts the network connection along with application messages, such as Temp/RH data, from the publishers. It also handles the subscription and unsubscription process and pushes the application data to MQTT clients acting as subscribers.
- The application on the right side of Figure 3.20 is an MQTT client that is a subscriber to the Temp/RH data being generated by the publisher or sensor on the left.
- This model, where subscribers express a desire to receive information from publishers, is well known. A great example is the collaboration and social networking application Twitter.

With MQTT, clients can subscribe to all data (using a wildcard character) or specific data from the information tree of a publisher. In addition, the presence of a message broker in MQTT decouples the data transmission between clients acting as publishers and subscribers. In fact, publishers and subscribers do not even know (or need to know) about each other. A benefit of having this decoupling is that the MQTT message broker ensures that information can be buffered and cached in case of network failures. This also means that publishers and subscribers do not have to be online at the same time.



MQTT Publish/Subscribe Framework

6 b. DESCRIBE THE SCHEDULING MANAGEMENT MECHANISMS FORWARDING MODELS AND SUPPORTED BY 6TiSCH.

The 6TiSCH architecture defines four schedule management mechanisms:

- **Static scheduling**: All nodes in the constrained network share a fixed schedule. Cells are shared, and nodes contend for slot access in a slotted aloha manner. Slotted aloha is a basic protocol for sending data using time slot boundaries when communicating over a shared medium.

- **Neighbor-to-Neighbor scheduling**:
  A schedule is established that correlates with the observed number of transmissions between nodes. Cells in this schedule can be added or deleted as traffic requirements and bandwidth needs change.

- **Remote monitoring and scheduling management:**
  Time slots and other resource allocation are handled by a management entity that can be multiple hops away. The scheduling mechanism leverages 6top and even CoAP in some scenarios.

- **Hop-by-hop scheduling**: A node reserves a path to a destination node multiple hops away by requesting the allocation of cells in a schedule at each intermediate node hop in the path.

There are three 6TiSCH forwarding models:

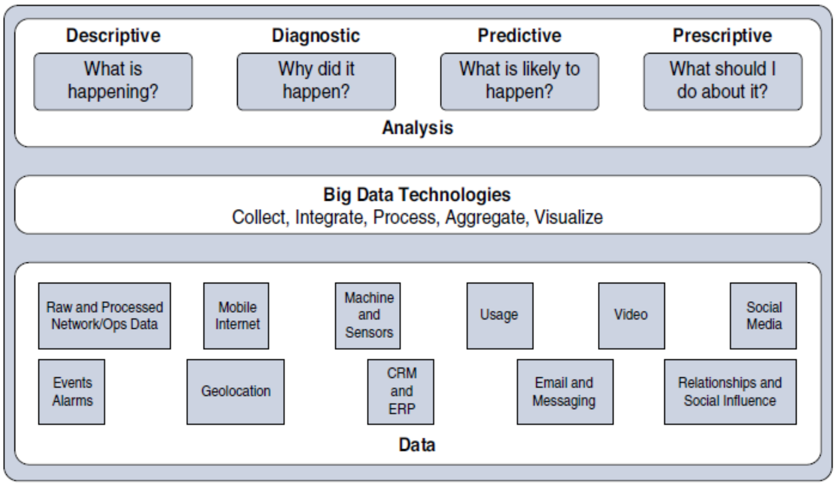
- **Track Forwarding (TF):** This is the simplest and fastest forwarding model. A "track" in this model is a unidirectional path between a source and a destination. This track is constructed by pairing bundles of receive cells in a schedule with a bundle of receive cells set to transmit. So, a Frame received within a particular cell or cell bundle is switched to another cell or cell bundle. This forwarding occurs regardless of the network layer protocol.

- **Fragment forwarding (FF):** This model takes advantage of 6LoWPAN fragmentation to build a Layer 2 forwarding table. Fragmentation within the 6LoWPAN protocol is covered earlier in this chapter, in the section "Fragmentation."

- **IPv6 Forwarding (6F):** This model forwards traffic based on its IPv6 routing table. Flows of packets should be prioritized by traditional QoS (quality of service) and RED (random early detection) operations. QoS is a classification scheme for flows based on their priority, and RED is a common congestion avoidance mechanism.

7 a. EXPLAIN DIFFERENT TYPES OF DATA ANALYSIS RESULTS WITH EXAMPLE

Data analysis is typically broken down by the types of results that are produced



Types of Data Analysis Results

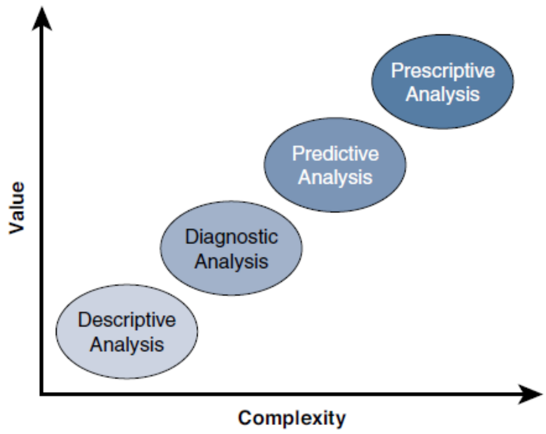
- **Descriptive:**
  ✓ Descriptive data analysis tells you what is happening, either now or in the past. For example, a thermometer in a truck engine reports temperature values every second.
  ✓ From a descriptive analysis perspective, you can pull this data at any moment to gain insight into the current operating condition of the truck engine. If the temperature value is too high, then there may be a cooling problem or the engine may be experiencing too much load.

- **Diagnostic:**
  ✓ When you are interested in the "why," diagnostic data analysis can provide the answer. Continuing with the example of the temperature sensor in the truck engine, you might wonder why the truck engine failed.

- ✓ Diagnostic analysis might show that the temperature of the engine was too high, and the engine overheated. Applying diagnostic analysis across the data generated by a wide range of smart objects can provide a clear picture of why a problem or an event occurred.

- **Predictive:**
- ✓ Predictive analysis aims to foretell problems or issues before they occur. For example, with historical values of temperatures for the truck engine, predictive analysis could provide an estimate on the remaining life of certain components in the engine.
- ✓ These components could then be proactively replaced before failure occurs. Or perhaps if temperature values of the truck engine start to rise slowly over time, this could indicate the need for an oil change or some other sort of engine cooling maintenance.

- **Prescriptive:**
- ✓ Prescriptive analysis goes a step beyond predictive and recommends solutions for upcoming problems. A prescriptive analysis of the temperature data from a truck engine might calculate various alternatives to cost-effectively maintain our truck.
- ✓ These calculations could range from the cost necessary for more frequent oil changes and cooling maintenance to installing new cooling equipment on the engine or upgrading to a lease on a model with a more powerful engine.
- ✓ Prescriptive analysis looks at a variety of factors and makes the appropriate recommendation.



- ✓ Both predictive and prescriptive analyses are more resource intensive and increase complexity, but the value they provide is much greater than the value from descriptive and diagnostic analysis.

7 b. DISTINGUISH BETWEEN SUPERVISED AND UNSUPERVISED MACHINE LEARNING.

SUPERVISED MACHINE LEARNING:

- In supervised learning, the machine is trained with input for which there is a known correct answer. For example, suppose that you are training a system to recognize when there is a human in a mine tunnel.
- A sensor equipped with a basic camera can capture shapes and return them to a computing system that is responsible for determining whether the shape is a human or something else.
- With supervised learning techniques, hundreds or thousands of images are fed into the machine, and each image is labeled. This is called the training set.
- An algorithm is used to determine common parameters and common differences between the images. The comparison is usually done at the scale of the entire image, or pixel by pixel. Images are resized to have the same characteristics, and each point is analyzed.
- Human images have certain types of shapes and pixels in certain locations.

- Each new image is compared to the set of known "good images," and a deviation is calculated to determine how different the new image is from the average human image and, therefore, the probability that what is shown is a human figure. This process is called classification.
- After training, the machine should be able to recognize human shapes. Before real field deployments, the machine is usually tested with unlabeled pictures—this is called the validation or the test set, depending on the ML system used—to verify that the recognition level is at acceptable thresholds.
- If the machine does not reach the level of success expected, more training is needed.

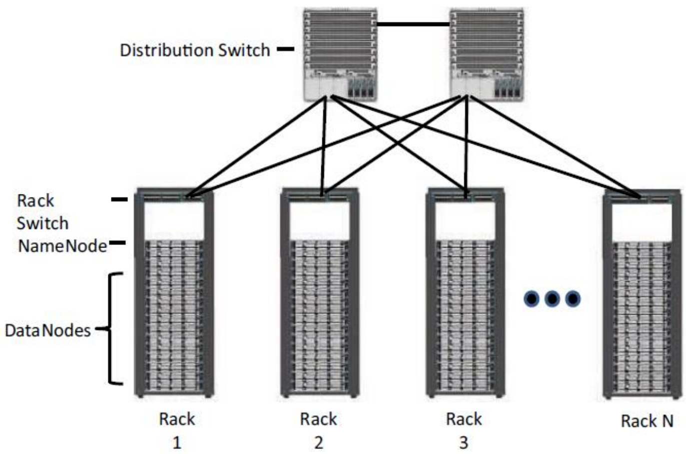
UNSUPERVISED MACHINE LEARNING:

- Supervised learning is not the best method for a machine to help with a human decision. Suppose that you are processing IoT data from a factory manufacturing small engine.
- About 0.1% of the produced engines on average need adjustments to prevent later defects, and your task is to identify them before they get mounted into machines and shipped away from the factory.
- With hundreds of parts, it may be very difficult to detect the potential defects, and it is almost impossible to train a machine to recognize issues that may not be visible. However, you can test each engine and record multiple parameters, such as sound, pressure, temperature of key parts, and so on.
- Once data is recorded, you can graph these elements in relation to one another (for example, temperature as a function of pressure, sound versus rotating speed over time). You can then input this data into a computer and use mathematical functions to find groups.
- For example, you may decide to group the engines by the sound they make at a given temperature. A standard function to operate this grouping, K-means clustering, finds the mean values for a group of engines. Grouping the engines this way can quickly reveal several types of engines that all belong to the same category.
- All engines of the same type produce sounds and temperatures in the same range as the other members of the same group. There will occasionally be an engine in the group that displays unusual characteristics .
- This is the engine that you send for manual evaluation. The computing process associated with this determination is called unsupervised learning. This type of learning is unsupervised because there is not a "good" or "bad" answer known in advance.
- It is the variation from a group behaviour that allows the computer to learn that something is different

7 c. EXPLAIN ELEMENTS OF HADOOP, WITH A NEAT DIAGRAM.

- Hadoop is the most recent entrant into the data management market, but it is arguably the most popular choice as a data repository and processing engine.
- Hadoop was originally developed as a result of projects at Google and Yahoo!, and the original intent for Hadoop was to index millions of websites and quickly return search results for open source search engines. Initially, the project had two key elements:

■ Hadoop Distributed File System (HDFS): A system for storing data across multiple nodes
■ MapReduce: A distributed processing engine that splits a large task into smaller ones that can be run in parallel



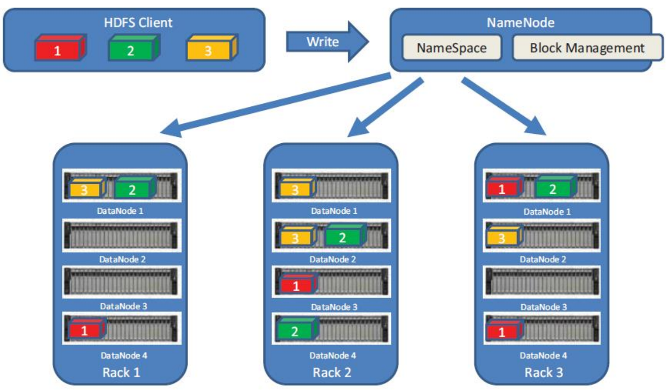
Distributed Hadoop Cluster

- ✓ For HDFS, this capability is handled by specialized nodes in the cluster, including NameNodes and DataNodes

- ✓ NameNodes: These are a critical piece in data adds, moves, deletes, and reads on HDFS. They coordinate where the data is stored, and maintain a map of where each block of data is stored and where it is replicated.

- ✓ DataNodes: These are the servers where the data is stored at the direction of the NameNode. It is common to have many DataNodes in a Hadoop cluster to store the data. Data blocks are distributed across several nodes and often are replicated three, four, or more times across nodes for redundancy.

Figure shows the relationship between NameNodes and DataNodes and how data blocks are distributed across the cluster.

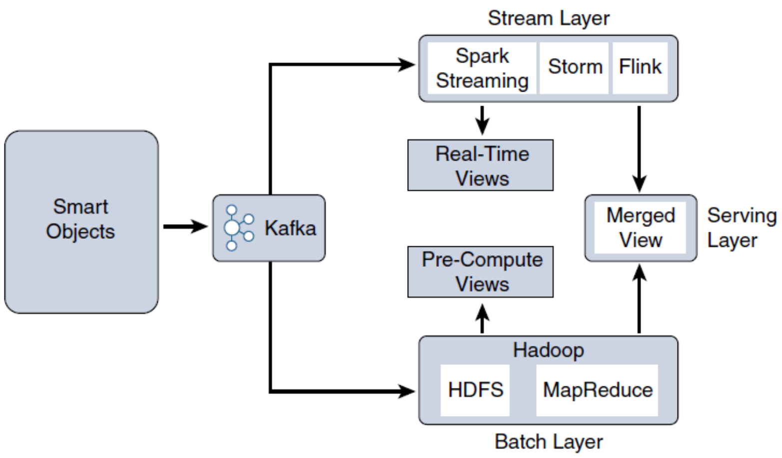


Writing a File to HDFS

8 a. WHAT IS APACHE SPARK? EXPLAIN LAYERS IN LAMBDA ARCHITECTURE, WITH A NEAT DIAGRAM.

- ✓ The key elements of a data infrastructure to support many IoT use cases involves the collection, processing, and storage of data using multiple technologies. Querying both data in motion (streaming) and data at rest (batch processing) requires a combination of the Hadoop ecosystem projects discussed.
- ✓ One architecture that is currently being leveraged for this functionality is the Lambda Architecture. Lambda is a data management system that consists of two layers for ingesting data (Batch and Stream) and one layer for providing the combined data (Serving).
- ✓ These layers allow for the packages discussed previously, like Spark and MapReduce, to operate on the data independently, focusing on the key attributes for which they are designed and optimized.

- ✓ Data is taken from a message broker, commonly Kafka, and processed by each layer in parallel, and the resulting data is delivered to a data store where additional processing or queries can be run. Figure shows this parallel data flow through the Lambda Architecture.

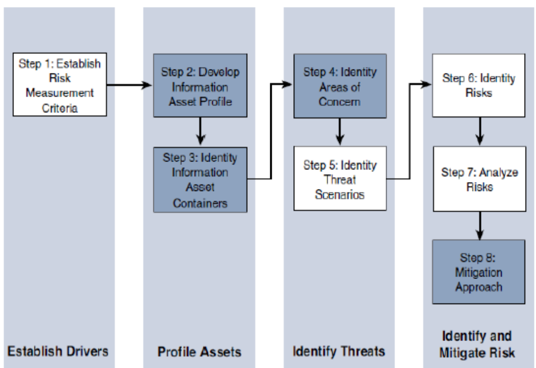


Lambda Architecture

- Stream layer: This layer is responsible for near-real-time processing of events. Technologies such as Spark Streaming, Storm, or Flink are used to quickly ingest, process, and analyze data on this layer.
- Batch layer: The Batch layer consists of a batch-processing engine and data store. If an organization is using other parts of the Hadoop ecosystem for the other layers, MapReduce and HDFS can easily fit the bill.
- Serving layer: The Serving layer is a data store and mediator that decides which of the ingest layers to query based on the expected result or view into the data. If an aggregate or historical view is requested, it may invoke the Batch layer.
- The Lambda Architecture can provide a robust system for collecting and processing massive amounts of data and the flexibility of being able to analyze that data at different rates.

8 b. EXPLAIN OCTAVE ALLEGRO STEPS AND PHASES, WITH A NEAT DIAGRAM

- The first step of the OCTAVE Allegro methodology is to establish a risk measurement criterion. OCTAVE provides a fairly simple means of doing this with an emphasis on impact, value, and measurement. The point of having a risk measurement criterion is that at any point in the later stages, prioritization can take place against the reference model



- The second step is to develop an information asset profile.
- The third step is to identify information asset containers.
- The fourth step is to identify areas of concern.
- In fifth step threat scenarios are identified.

- At the sixth step risks are identified.
- The seventh step is risk analysis.
- Finally, mitigation is applied at the eighth step.

9 b. EXPLAIN THE FOLLOWING WITH RESPECT TO THE ARDUINO PROGRAMMING:

i).STRUCTURES

✓ A struct is simply a collection of different types of variable

```
struct structName{
  item1_type item1_name;
  item2_type item2_name;
  .
  .
  .
  itemN_type itemN_name;
}
```

ii).FUNCTIONS

✓ Functions allow structuring the programs in segments of code to perform individual tasks.

Function Declaration

```
#include <stdio.h>

nt sum_func (int x, int y) // function declaration {
  int z = 0;
  z = x+y ;
  return z; // return the value
}

void setup () {
  Statements // group of statements
}

Void loop () {
  int result = 0 ;
  result = Sum_func (5,6) ; // function call
}
```

iii).VARIABLES

✓ The variables are defined as the place to store the data and values. It consists of a name, value, and type.

✓ The variables can belong to any data type such as int, float, char, etc. Consider the url - Arduino data types for detailed information.

Example:

```
int LEDpin = 8;
pinMode(LEDpin, OUTPUT);
pinMode( 8, OUTPUT);
```

v).DATA TYPE WITH EXAMPLE

- ✓ The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

void	Boolean	char	Unsigned char	byte	int	Unsigned int	word
long	Unsigned long	short	float	double	array	String-char array	String-object

9 b. EXPLAIN THE STEPS TO INSTALL OPERATING SYSTEM IN THE SD CARD OF RASPBERRY Pi. WRITE A PYTHON PROGRAM TO BLINK ON LED

- ✓ Install an OS With Raspberry Pi Imager
- ✓ Raspberry Pi Imager is available for Windows, macOS, and Ubuntu. Once installed on your computer, the process is simple:

1. Under Operating System click Choose OS
2. Browse the list for your preferred OSs and select the one you want
3. Click Ctrl+Shift+X to preconfigure advanced options (see below)
4. Next, click Choose Storage to select the SD card
5. Click Write

- ✓ Wait while the data is written and verified. When done, click Continue, then close the imager tool.
- ✓ To install the Python library open a terminal and execute the following

```
$ sudo apt-get install python-rpi.gpio python3-rpi.gpio
```

- ✓ To initialize the GPIO ports on the Raspberry Pi we need to first import the Python library, the initialize the library and setup pin 8 as an output pin.

```
import RPi.GPIO as GPIO    # Import Raspberry Pi GPIO library
from time import sleep     # Import the sleep function from the time module
GPIO.setwarnings(False)    # Ignore warning for now
GPIO.setmode(GPIO.BOARD)   # Use physical pin numbering
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW)   # Set pin 8 to be an output pin and set initial value to low (off)
```

- ✓ Next we need to turn the LED on and off in 1 second intervals by setting the output pin to either high (on) or low (off). We do this inside a infinite loop so our program keep executing until we manually stop it.

```
while True: # Run forever
GPIO.output(8, GPIO.HIGH) # Turn on
sleep(1)                  # Sleep for 1 second
GPIO.output(8, GPIO.LOW)  # Turn off
sleep(1)                  # Sleep for 1 second
```

- ✓ Combining the initialization and the blink code should give you the following full Python program:

```
import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library
from time import sleep # Import the sleep function from the time module
```

```
GPIO.setwarnings(False) # Ignore warning for now
GPIO.setmode(GPIO.BOARD) # Use physical pin numbering
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) # Set pin 8 to be an output pin and set initial value to low (off)
while True: # Run forever
GPIO.output(8, GPIO.HIGH) # Turn on
sleep(1) # Sleep for 1 second
GPIO.output(8, GPIO.LOW) # Turn off
sleep(1) # Sleep for 1 second
```

- ✓ With our program finished, save it as blinking_led.py and run it either inside your IDE or in the console with:

  $ python blinking_led.py

10 a. EXPLAIN KEY VERTICALS TARGETED IN SMART CITIES, WITH A NEAT DIAGRAM

- ✓ There are many differing approaches and solutions for city management. All these solutions typically start at the street level, with sensors that capture data on everything from parking space availability to water purity.
- ✓ Data analytics is also used extensively—for example, to reduce crime or improve traffic flows.
- ✓ Citizens can use tools to leverage their smart mobile devices, such as to report problems and make recommendations for improving urban life or locate available parking spaces.
- ✓ When enabled through connectivity, these smart solutions can have a transformative impact on quality of life.
- ✓ A recent Cisco study, as illustrated in Figure below, expects IoT to have the following economic impact over a 10-year period:

1) Smart buildings: Smart buildings have the potential to save $100 billion by lowering operating costs by reducing energy consumption through the efficient integration of heating, ventilation, and air-conditioning (HVAC) and other building infrastructure systems.

- Note that the financial gain applies to city budgets only when a building is city owned.
- However, the reduced emissions benefit the city regardless of who owns the buildings.

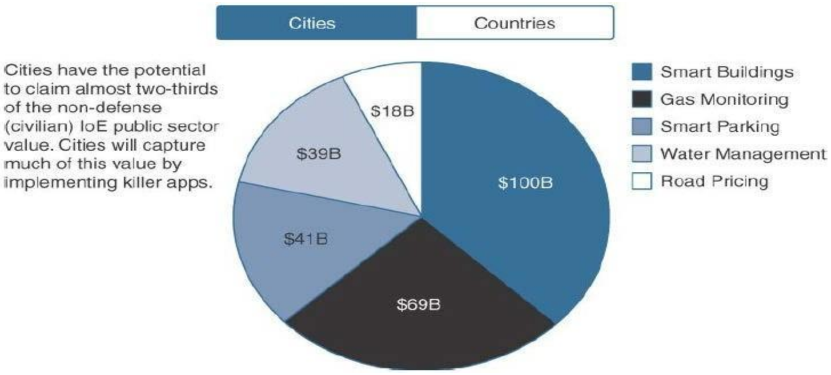


Figure Key Use Cases for Smart Cities

2) Gas monitoring: Monitoring gas could save $69 billion by reducing meter-reading costs and increasing the accuracy of readings for citizens and municipal utility agencies.

- The financial benefit is obvious for users and utility companies when the utility is managed by the city.
- There are also very important advantages in terms of safety, regardless of who operates the utility.
- In cases of sudden consumption increase, a timely alert could lead to emergency response teams being dispatched sooner, thus increasing the safety of the urban environment.

3) Smart parking: Smart parking could create $41 billion by providing realtime visibility into parking space availability across a city.

- Residents can identify and reserve the closest available space, traffic wardens can identify noncompliant usage, and municipalities can introduce demand based pricing.

4) Water management: Smart water management could save $39 billion connecting household water meters over an IP network to provide remote usage and status information.

- The benefit is obvious, with features such as real-time consumption visibility and leak detection.
- A gate or a pump can be opened and closed remotely and automatically in real time, based on a variety of flow input and output analytics data.
- Real-time traffic condition data is very valuable and actionable information that can also be used to proactively reroute public transportation services or private users.

Global vs. Siloed Strategies

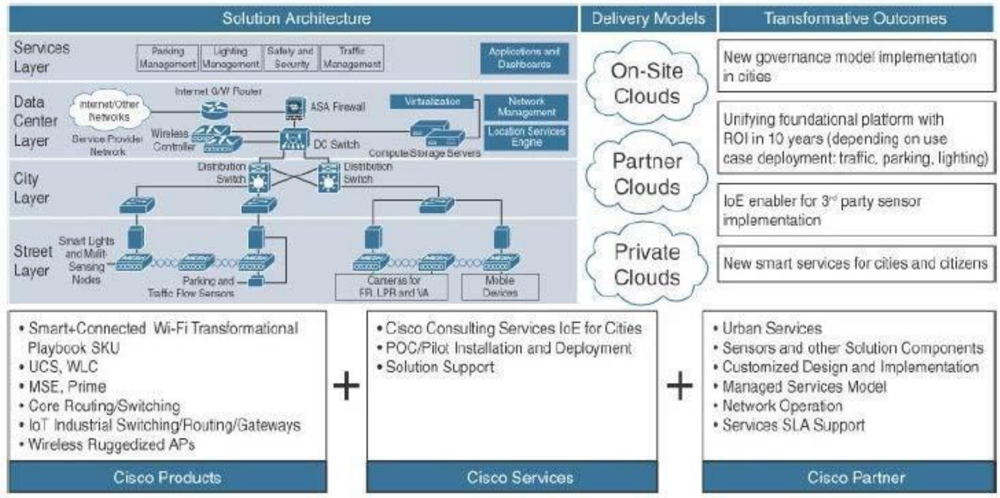
- The main obstacle in implementing smart solutions in today's traditional infrastructure is the complexity of how cities are operated, financed, regulated and planned.
- Cities attempting to upgrade their infrastructure to match the growing needs of the citizen population often invest in one problem at a time, and they do it independently.
- Even cities using IoT technology break up city assets and service management into silos that are typically unable to communicate or rely on each other.

The independent investment model results in the following problems:

- Isolation of infrastructure and IT resources
- No sharing of intelligence and information, such as video feeds and data from sensors.
- Waste and duplication in investment and effort
- Difficulty scaling infrastructure management

10 b. EXPLAIN SMART CITY IOT ARCHITECTURE, WITH A NEAT DIAGRAM

- A smart city IoT infrastructure is a four-layered architecture, as shown in Figure below.
- Data flows from devices at the street layer to the city network layer and connect to the data center layer, where the data is aggregated, normalized, and virtualized.
- The data center layer provides information to the services layer, which consists of the applications that provide services to the city.

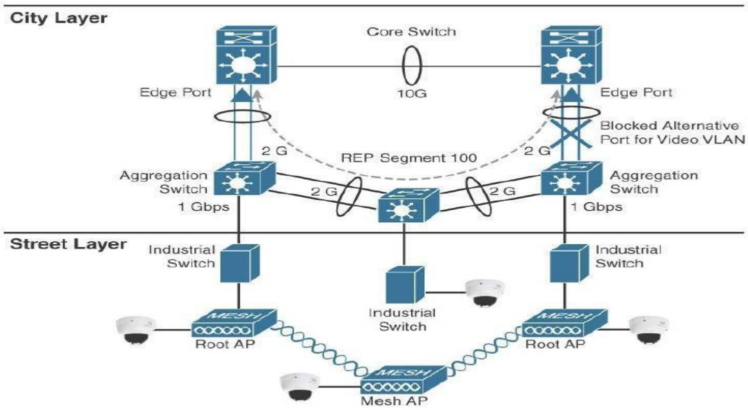


Street Layer

- The street layer is composed of devices and sensors that collect data and take action based on instructions from the overall solution, as well as the networking components needed to aggregate and collect data.

City Layer

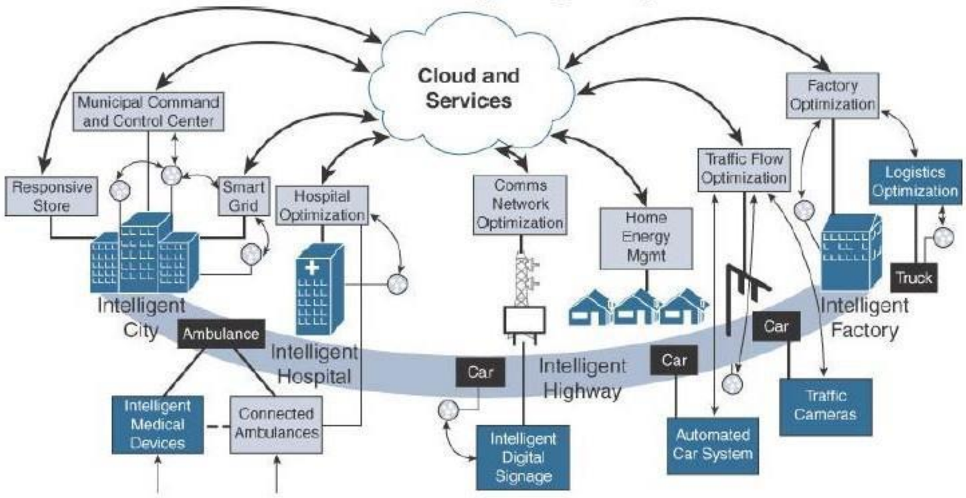
- At the city layer, which is above the street layer, network routers and switches must be deployed to match the size of city data that needs to be transported.
- This layer aggregates all data collected by sensors and the end-node network into a single transport network.



Data Center Layer

- data collected from the sensors is sent to a data center, where it can be processed and correlated.

- Based on this processing of data, meaningful information and trends can be derived, and information can be provided back



Services Layer

- The true value of ICT connectivity comes from the services that the measured data can provide to different users operating within a city.
- Smart city applications can provide value to and visibility for a variety of user types, including city operators, citizens, and law enforcement.