Chinmay Nilkanth

MODULE 1

Introduction to Internet of Things

Syllabus

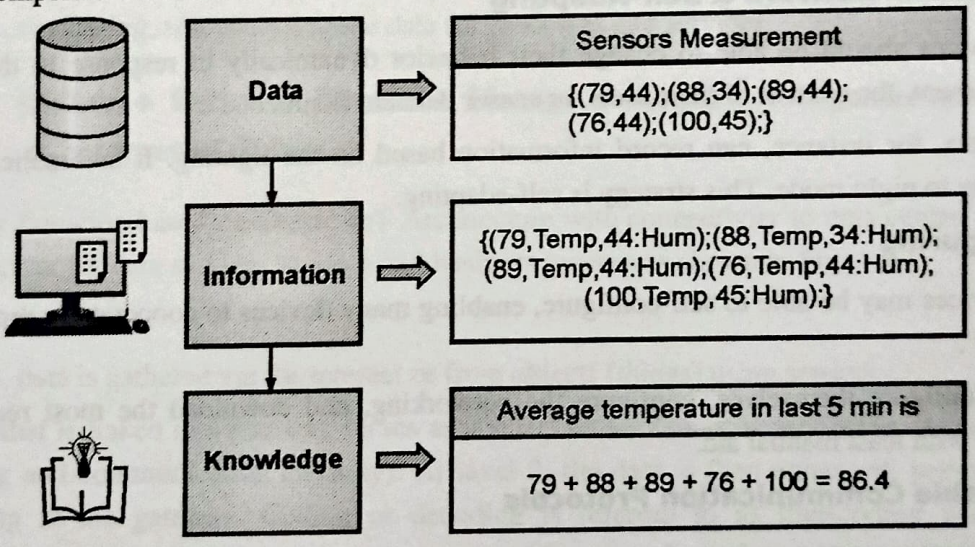
Introduction to IoT- Defining IoT, Characteristics of IoT, Conceptual Framework of IoT, Physical design of IoT, Logical design of IoT, Functional blocks of IoT, Brief review of applications of IoT, Smart Object – Definition, Characteristics and Trends

Self-learning Topics : Hardware and software development tools for - Arduino, NodeMCU, ESP32, Raspberry Pi, for implementing internet of things, Simulators-Circuit.io, Eagle, Tinkercad.

1.0 INTRODUCTION OF INTERNET OF THINGS (IOT)

Internet of Things (IoT) is fully networked and connected devices sending analytics data back to the cloud or data center. It is a network in which each object or thing has a unique identification number, and data is transmitted over a network without verbal confrontation. Innovations in sensor networks, mobile devices, wireless communications, networking, and cloud technologies are driving IoT. It has a much broader application than simply connecting devices to the internet; it also enables data processing, communication, and control through the use of applications.

Conceptualization of IoT can be made from Data-Information-Knowledge. Data doesn't have significance until this is contextualized, processed, and transformed into useful information and knowledge inference. Consider the raw sensor measurements produced by a weather monitoring station ((79,44) ; (88,34) ;...), for instance, Fig. 1.1.1. On its own, this information is meaningless. However, the data becomes meaningful when we add context (information), such as the fact that 44 is humidity and 79 is temperature. We can infer knowledge by observing the data/information and processing it. The average temperature over the previous five minutes can be deduced, for instance, by looking at the temperature of the last five tuples. Based on this knowledge, actions can be taken, such as setting off an alarm if the temperature rises above 80.



(1A1)**Fig. 1.1.1 : IoT conceptualization to Data, Information and Knowledge**

1.1 DEFINING IOT

Formal Definition

- A **dynamic global network** infrastructure with **self-configuring capabilities** based on standard and **interoperable communication protocols** where physical and virtual "things" **have identities,** physical attributes, and visual personalities and use intelligent interfaces and are seamlessly **integrated into information networks** that communicate data associated with users and their environments.

Informal Definition

- Informally, IoT is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.
- As with computers, tablets, and smartphones, the term "Internet of Things" refers to a network of physical objects that send, receive, or communicate information using the Internet or other communication technologies and networks, allowing for the monitoring, coordinating, or control of processes across the Internet or another data network
- According to another source, the phrase "IoT" means: A network of actual physical objects, or "things," that are equipped with electronics, software, sensors, and connectivity in order to enhance their functionality and value by exchanging data with other connected devices, their manufacturer, and/or their operator. Despite being able to communicate with one another within the current Internet infrastructure, each object has an embedded computing system that enables it to be uniquely identified.

1.2 CHARACTERISTICS OF IOT

Dynamic Global network & Self-Adapting

- IoT devices should be able to change their behavior dynamically in response to their operating environment, the context of their users, or sensor data environments.
- A camera, for instance, can record information based on the lighting. It automatically switches from day to night mode. This strategy is self-adapting.

Self Configuring

IoT devices may be able to self-configure, enabling many devices to cooperate to provide specific functionality .

Ex : Configure themselves, configure the networking, and download the most recent software updates with least manual aid.

Interoperable Communication Protocols

- IoT Devices support a variety of open standard communication protocols
- They Interact with both infrastructure and other devices
- For example: A smart Phone is able to control the smart TV of different manufacturers.

Unique Identity

- Every IoT device possesses a distinct identity.
- This unique identity can be an IP Address or Uniform Resource Indicator (URI).
- Its identification element is very **useful** if it needs **to access data from a specific device.**
- Along with the control, configuration, and management infrastructure, the IoT device also enables status monitoring, and querying for the user.

Integrated into Information Network

- This allows them to communicate and exchange data with other devices to perform certain analysis.
- This makes the IoT systems **"Smarter".**
- For example: a weaher monitoring node can describe its monitoring capabilities to another connected node, such that they can communicate and exchange data.

1.3 CONCEPTUALIZED FRAMEWORK OF IOT

An IoT Conceptualized framework can be given in terms of three expressions :

1.3.1 Physical Object + Controller, Sensor and Actuators + Internet = Internet of Things

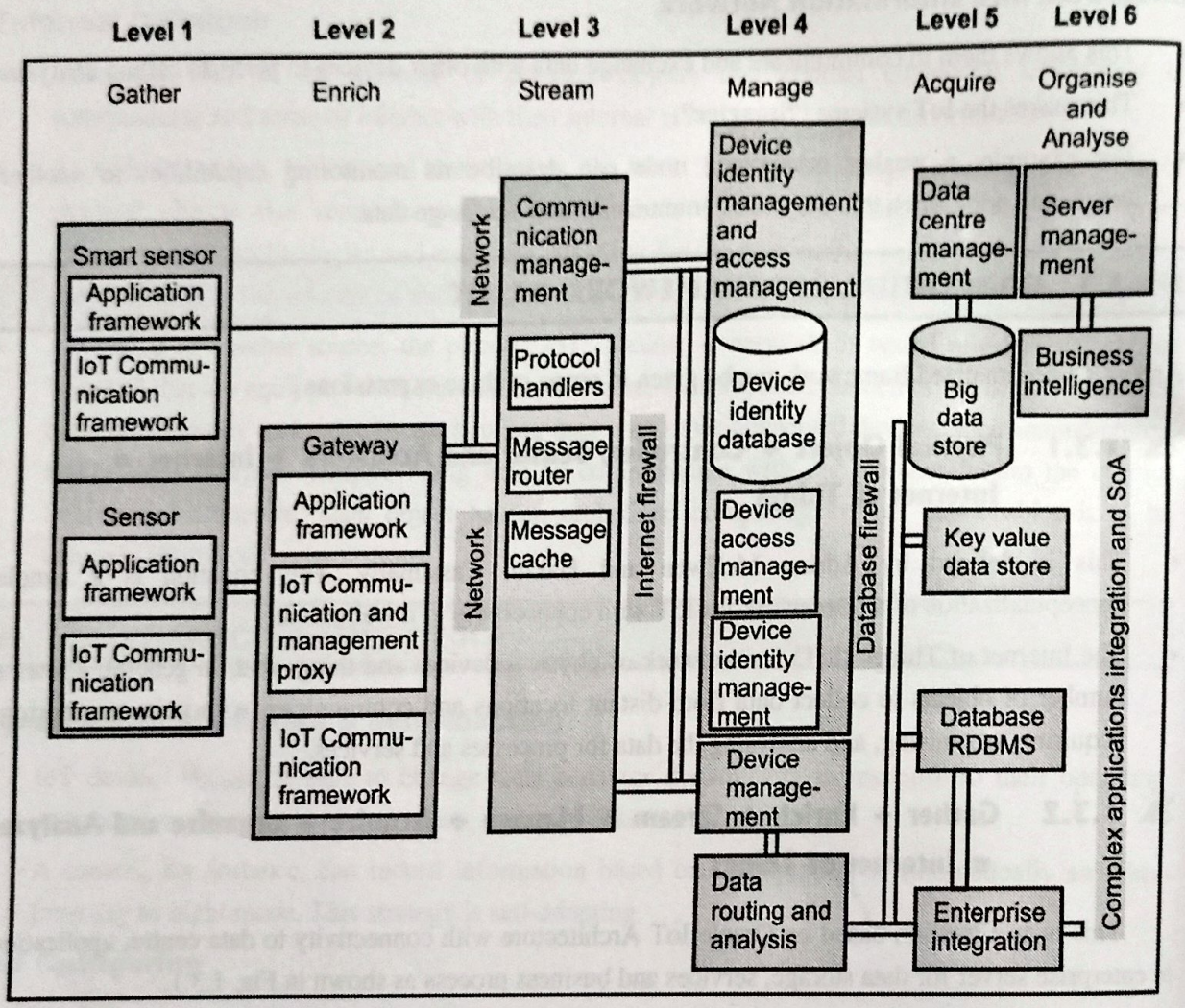
- This is defined by Adrian McEwen and Hakim Cassimally. This equation is a simple conceptualization of a framework for IoT with connectivity to a web service.
- The Internet of Things (IoT) is a network of physical devices and things that, in general, allows a number of objects to collect data from distant locations and communicate with units monitoring, acquiring, organising, and analysing the data for processes and services.

1.3.2 Gather + Enrich + Stream + Manage + Acquire + organize and Analyze = Internet of Things

This is an Equation based on Oracle IoT Architecture with connectivity to data centre, application or enterprise server for data storage, services and business process as shown in Fig. 1.3.1.

Following are the steps :

1. At level 1, data is gathered via the internet or from objects (things) using sensors.
2. A sensor that is linked to a gateway serves as a smart sensor (smart sensor refers to a sensor with computing and communication capacity). At level 2, the data is then enhanced, perhaps through transcoding at the gateway. Coding or decoding is referred to as transcoding when data is transferred between two entities.
3. At level 3, a communication management subsystem transmits or receives data streams.
4. At level 4, data from the device is received by the subsystems for device management, identity management, and access management.
5. At level 5, data is acquired by a data store or database.
6. At level 6, data from the devices and things is organized and analyzed. Data analysis is used, for instance, in corporate processes to gather business intelligence.

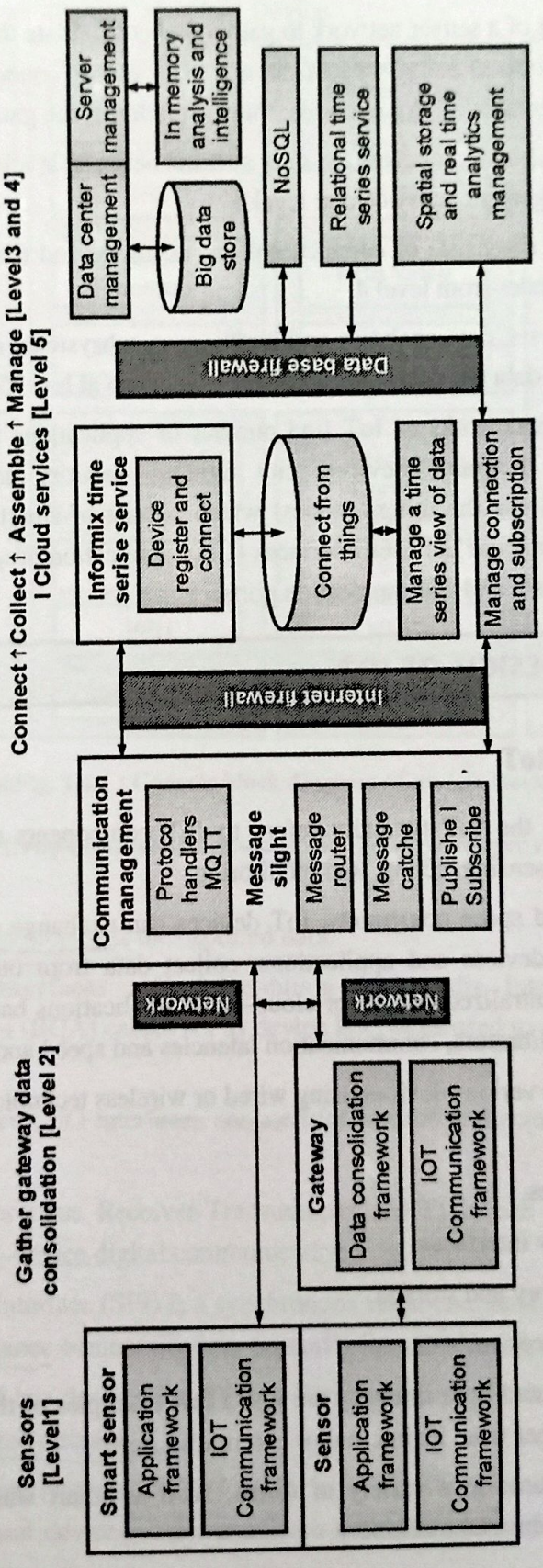


(1A3)**Fig. 1.3.1 : Oracle IoT Architecture**

1.3.3 Gather + Consolidate + Connect + Collect + Assemble + Manage and Analyse = Internet of Things

This is an Equation based on the IBM Framework. This concepts consists of a number of subsystems. The data is acquired at remote locations in a database or data store. The services and processes need data managing, acquiring, organising and analysing as shown in Fig. 1.3.2.

NOTES



(1A3)**Fig. 1.3.2 : IBM IoT Conceptual Framework**

The steps are as follows :

1. Levels 1 and 2 consist of a sensor network to gather and consolidate the data. First level gathers the data of the things (devices) using sensors circuits. The sensor connects to a gateway. Data then consolidates at the second level, for example, transformation at the gateway at level 2.
2. The gateway at level 2 communicates the data streams between levels 2 and 3. The system uses a communication-management subsystem at level 3.
3. An information service consists of connect, collect, assemble and manage subsystems at levels 3 and 4. The services render from level 4.
4. Real time series analysis, data analytics and intelligence subsystems are also at levels 4 and 5. A cloud infrastructure, a data store or database acquires the data at level 5.

Various conceptual frameworks of IoT find number of applications including the ones in M2M communication networks, wearable devices, city lighting, security and surveillance and home automation. Smart systems use the things (nodes) which consist of smart devices, smart objects and smart services. Smart systems use the user interfaces (UIs), application programming interfaces (APIs), identification data, sensor data and communication ports.

1.4 PHYSICAL DESIGN OF IOT

1.4.1 Things of IoT

- The term "Things" in the IoT typically refers to IoT components with distinct identities and capabilities for remote sensing, acting, and monitoring.
- Depending on time and space constraints, IoT devices can exchange data (directly or indirectly) with other connected devices and applications, collect data from other devices and process it locally, or send it to centralized servers or cloud-based applications back ends for processing (ie : Memory, processing calibrators, communication latencies and speed and deadlines).
- An IoT device connects various devices using wired or wireless technology.
- These incorporate :

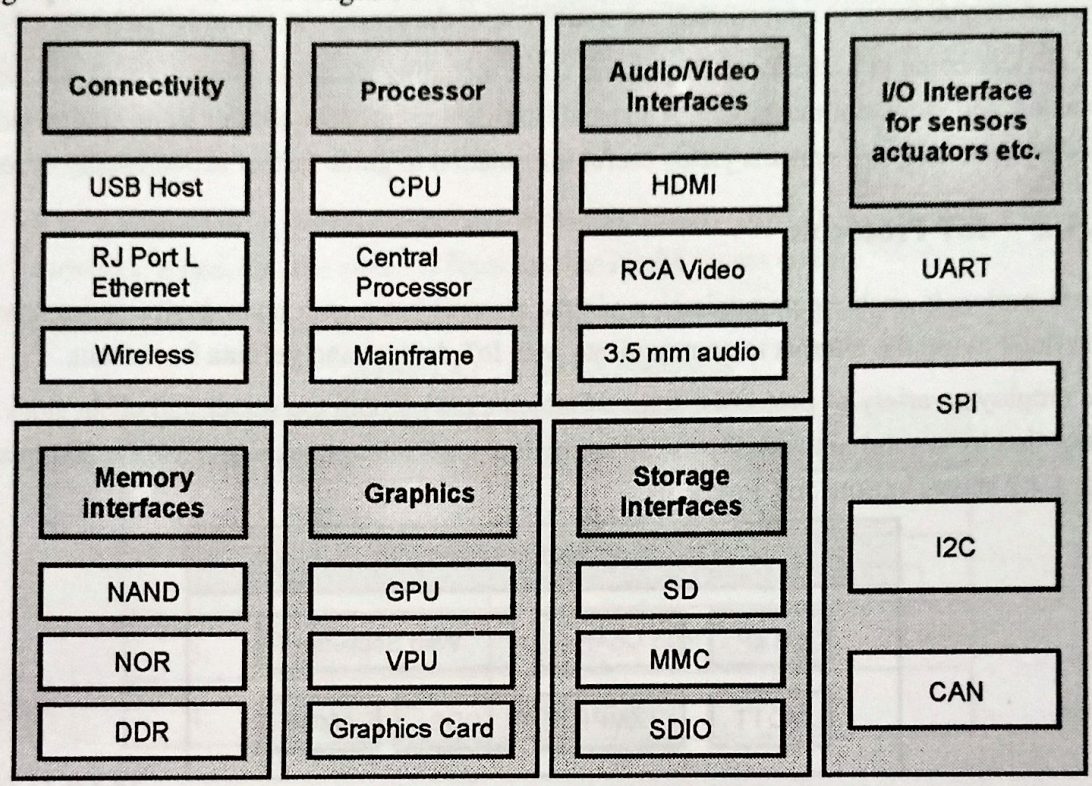
  I) Sensor IoT interfaces,

  II) Internet connectivity interfaces

  III) Interfaces for memory and storage

  IV) Audio video interfaces.
- Temperature, humidity, and light intensity are just a few examples of the several sorts of data that an IoT device might gather from its internal or external sensors.
- IoT devices can also come in a variety of forms, such as smart watches, LED light vehicles, wearable sensors, and industrial machinery.

- Fig. 1.4.1 shows the block diagram of an IoT Device



(1A4)**Fig. 1.4.1 : Generic block diagram of an IoT Devices**

(1) **Connectivity :** Connectivity between the devices and the server is provided through tools like USB hosts and ETHERNET.

(2) **Processor :** Processor is processes the captured data.

(3) **Audio and Video Interfaces :** High-Definition Multimedia Interface (HDMI) and Radio Corporation of America (RCA) cables act as device interfaces used to record audio and videos in a system.

(4) **Input/Output Interface :** I/O interfaces are used for sensors and actuators. There are various I/O interfaces like

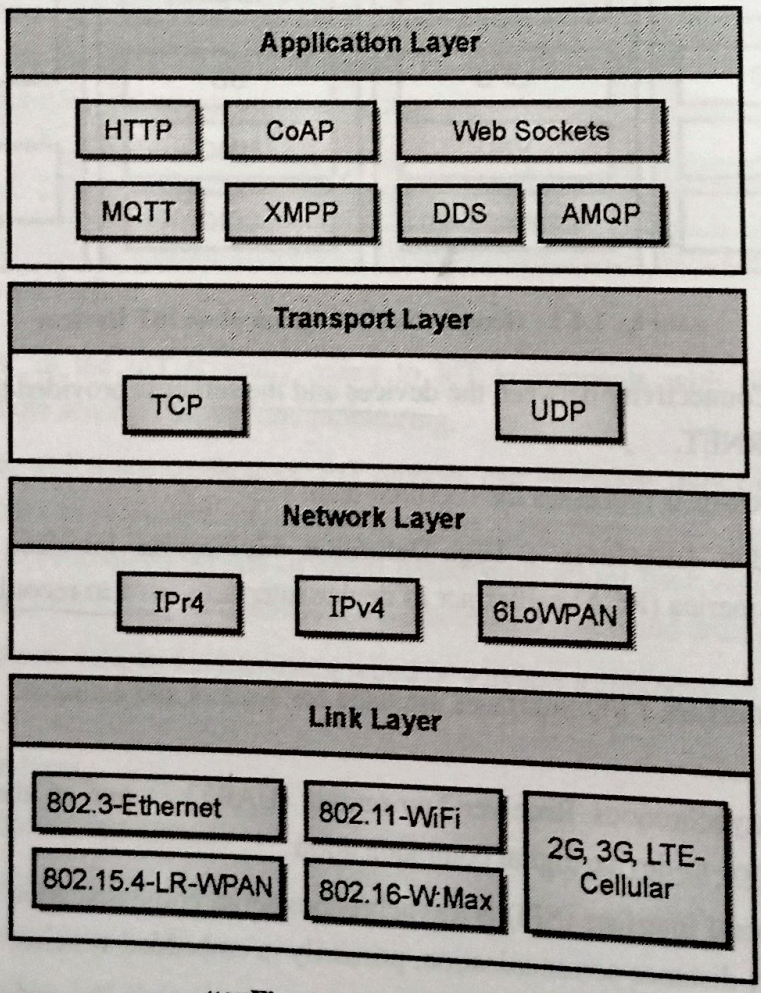
- Universal Asynchronous Receiver-Transmitter (UART) is one of the simplest and oldest forms of device-to-device digital communication
- Serial Peripheral Interface (SPI) is a synchronous serial communication interface specification used for short-distance communication, primarily in embedded systems.
- Inter-IC, a type of bus designed by Philips Semiconductors in the early 1980s, which is used to connect integrated circuits (ICs).
- A Controller Area Network (CAN bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer.

(5) **Storage Interfaces :** Memory devices like Secured Digital (SD), Multimedia Card (MMC), Secured Digital Input Output (SDIO) are used to store the sensed data from IoT sensors.

IoT devices come in a variety of shapes and sizes, including wearable sensors, smart watches, LED light vehicles, and industrial machinery. Almost all IoT devices produce data of some kind, which when processed by data analytics systems yields useful information to guide further action locally or remotely.

1.4.2 IoT Protocols

- These protocols enable communication via the internet between a node device and a server. It is beneficial to use the internet to communicate with IoT devices and get data from them.
- We employ a variety of protocols, some of which exist on both the server and client sides and are controlled by several network layers, including the application, transport, network, and link layers. Fig. 1.4.2 shows various IoT Protocols.



(1A5)**Fig. 1.4.2 : IoT Protocols**

- IoT protocols are divided in four layers namely Link Layer, Network Layer, Transport Layer and the Application Layer

1. Link Layers

Link Layer protocols control the physical transmission of data over the network's physical layer or media (example copper wire, electrical cable, or radio wave).

It also controls the devices' coding and signaling of the packets.

a) 802.3 Ethernet

- It is a collection of technologies and protocols mostly used in LANs. For wired Ethernet networks, it specifies the physical layer and the media access control.
- Various standards of ethernet and their sharing medium are mentioned in Table 1.4.1.

Table 1.4.1 : Ethernet Standards

Sr. No.	Standard	Shared Medium
1.	802.3	Coaxial Cable (10BASE5)
2.	802.3.i	Copper Twisted Pair (10BASE-T)
3.	802.3.j	Fiber Optic (10BASE-F)
4.	802.3.ae	Fiber (10gbps)

b) 802.11 Wi-Fi

- It is a set of LAN protocols that outlines the physical layer and media access control protocols needed to construct wireless local area networks.
- Various standards of ethernet are mentioned in Table 1.4.2.

Table 1.4.2 : Wi-fi Standards

Sr. No.	Standard	Operates in
1.	8011.a	5 Ghz Band
2.	802.11.b and 802.11g	2.4 Ghz Band
3.	802.11.n	2.4/5 Ghz Bands
4.	802.11.ac	5 Ghz Band
5.	802.11.ad	60 Ghz Band

c) 802.11 Wi-Max

It is a set of wireless broadband standards mentioned in Table 1.4.3. Wi-Max standards provide data rates from 1.5 Mbps to 1Gbps.

Table 1.4.3 : Wi-Max Standards

Sr. No.	Standard	Operates in
1.	802.16m	100 Mbps for Mobile Station and 1 Gbps for fixed stations

d) 802.15.4 LR-WPAN

- It is a collection of standards for low-rate wireless personal area networks (LR-WPAN).
- It is the basis for high-level protocols such as ZigBee.
- It supports data rates from 40-250 kbps. It provides low-cost and low-speed communication for power constrained devices.

e) 2G/3G/4G - Mobile Communication

- The many mobile communication standards, including second generation (2G), are listed here GSM and CDMA, third generation (3G), includes UMTS and CDMA2000 and Fourth generation (4G), includes LTE.
- Data rates of these standards range from 9.6 (2G) Kbps to 100 Mbps (4G).

2. Network Layer

This layer sends IP Datagrams from the source network to the destination network.

IPv4 and IPv6 protocols are used to identify hosts while sending data packets.

a) Internet Protocol Version 4 (IPv4):

- Each device connected to the network is given a protocol address, which is a specific numerical label.
- An IP address serves two key purposes: addressing hosts and locations.
- An IP address for IPv4 is 32 bits long that allows a total of 2^{32} addresses.

b) Internet Protocol Version 6 (IPv6)

It is the newest version and an IPv4 successor that has an IP address length of 128 bits.

c) IPv6 Low Power Wireless Personal Area Network (6LoWPAN)

- Low-power devices with little computing capacity can now use IPv6 through low-power wireless personal area networks.
- These networks operate in the 2.4 GHz band and have data transfer rates as low as 50 kb/s.

3. Transport Layer

End-to-end message transfer is made possible by the Transport layer protocols, independent of the underlying network. This layer is used to manage error control, flow control, segmentation and congestion control.

On connections, the message transfer functionality can be configured with (TCP) or without (UDP) handshake acknowledgements

a) Transmission Control Protocol (TCP)

- This is the most widely used protocol for web-browsers along with hypertext transfer protocols (HTTP), HTTPS Application layer protocols, email programs (SMTP) and file transfer protocol (FTP).
- While IP protocol deals with delivering packets, TCP enables consistent transmissions of packets in the correct order. TCP is a connection-oriented and stateful protocol.
- TCP also has the capacity to deduce errors, allowing duplicate packets to be rejected and low-quality packets to be retransmitted. Flow control maintains the data rates of senders and receivers.

b) User Datagram Protocol (UDP)

- UDP is a connectionless protocol and dont need any connection setup.
- This is transaction oriented and stateless protocol. UDP does not offer guaranteed delivery, message ordering, or duplicate removal.

4. Application Layer

The protocols at the application layer specify how an application interacts with those at lower layers to transmit data over a network.

The data is often stored in files, is encoded by the application layer protocol and encapsulated in the transport layer protocol. Process-to-process connections are made possible via application layer protocol employing ports.

a) Hypertext Transfer Protocol (HTTP)

- This protocol forms the foundation of the World Wide Web (WWW) and to transmit the media documents.
- This includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS etc.
- This follows a request-response model where a request is submits to a server and a client waits for a response.
- The server does not store any information between queries so it is also known as stateless protocol.
- The Universal Resource Indicator (URI) is used by HTTP Protocols to identify HTTP resources.

b) Constrained Application Protocol (CoAP)

- This protocol is used for Machine to Machine (M2M) applications and designed for limited environments with constrained devices and networks.
- CoAP is a web transmission protocol similar to http and has a request-response format; however, it runs on top of UDP rather than TCP.

- CoAP employs a client-server architecture in which clients and servers communicate via connectionless datagrams.
- CoAP supports methods such as GET,POST, PUT and DELETE.

c) WebSocket

- This Protocol allows full-duplex communication over a single socket.
- It is based on TCP.
- Here, clients can be a browser, IoT device or mobile application.

d) Message Queue Telemetry Transport (MQTT)

- It is a compact messaging protocol built on the public-subscribe model.
- In MQTT, clients, such as Internet of Things (IoT) devices, connect to the server, also known as the MQTT broker, and publish messages to topics on the server.
- The broker forwards the message to the clients subscribed to the topic.
- MQTT is well suited for constrained environments where devices have limited processing, low memory and n/w bandwith requirement

e) Extensible Messaging and Presence Protocol (XMPP)

- It is a protocol for XML data streaming and real-time communication between network elements.
- Various applications, including messaging, presence, data syndication, multiparty gaming chat, and voice/voice conversations, are powered by XMPP.
- With XMPP, you may instantly exchange short XML data snippets from one network object to another.
- Both client-server and server-client communication paths are supported by XMPP.
- It is a decentralized protocol.
- Example: Due to the XMPP protocol, in WhatsApp conversation, when a message is viewed by the recipient, a blue tick displayed.

f) Data Distribution Service (DDS)

- It is a data-centric middleware standard for device-to-device or machine-to-machine communication.
- DDS is a publish-subscribe model where publishers create topics to which subscribers can use.
- Provides Quality-of-service control and configurable reliability.

g) Advanced Message Queuing Protocols (AMQP)

- It is used for business messaging.
- It supports both point-to-point and publisher/subscriber models, routing and queuing.
- Here, brokers receives the messages from publishers and route them over connections to consumers through messaging queues.

Table 1.4.3 summaizes the IoT Protocols.

Table 1.4.3 : Wi-Max Standards

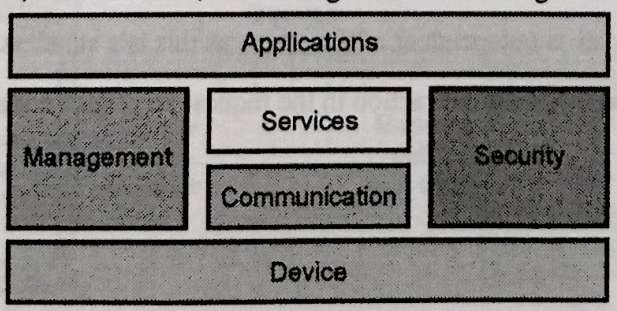
Parameters	HTTP	CoAP	XMPP	DDS	AMQP	MQTT
Protocols	TCP	UDP	TCP	TCP and UDP	TCP	TCP
Network Layer	IP	6LowPAN	IP	IP	IP	IP
Architecture	Client-Server	Client-Server and Publish-Subcribe	Client-Server and Publish-Subcribe	Publish-Subcribe	Client-Server	Publish-Subcribe
Synchronization	Needed	Not Needed	Needed	Sometimes Needed, Sometimes Not	Needed	Needed
Designed for	Internet	IoT/M2M	IoT/M2M	Realtime Systems	M2M	IoT/M2M
Application	World Wise Web (WWW)	Retrieving sensor data	WhatsApp, Gaming, Google Talk	Volkswagen, Smart Cars for video assistants	Google Cloud	Facebook Messanger

1.5 LOGICAL DESIGN OF IOT

- The logical design of an IoT system refers to an abstract representation of entities and processes without going into the low-level specifications of implementation.
- It uses Functional Blocks, Communication Models, and Communication APIs to implement a system.

1.5.1 IoT Functional Blocks

An IoT system is made up of several functional building elements that provide the system the ability to identify, sense, act, communicate, and manage as shown in Fig. 1.5.1.



(1A6)**Fig 1.5.1 : Functional Blocks of IoT**

These functional blocks are explained as follows :

(1) **Device :** These devices are used to provide sensing and monitoring control functions that gather information from the outside world.

(2) **Communication :** This block takes care of the communication of the IoT System.

(3) **Services :** It offers some services, like controlling and monitoring a device, publishing and erasing data, and system restoration.

(4) **Management :** It supports various functions to manage the IoT System.

(5) **Security :** This block is used to safeguard the IoT System by offering various functions such as authentication, authorization, message and content integrity, and data security.

(6) **Application :** It is an interface that offers a control system for the users to check the system status and analyze the processed data.

1.5.2 IoT Communication Model

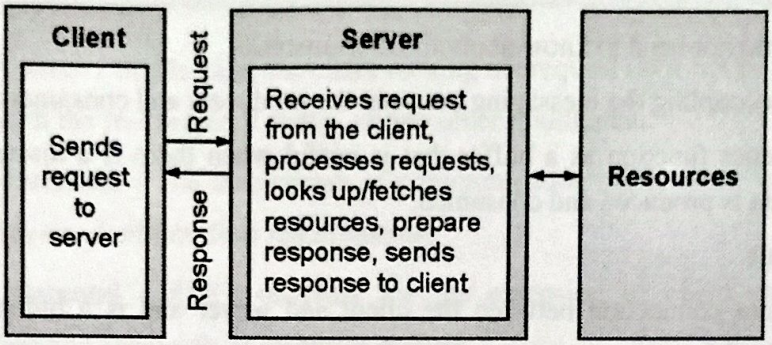
Communication Models determine the mechanism or the manner in which the data is exchanged or transfers between various devices in IoT Network.

There are four types of IoT Communication Models that are used for data exchange in the IoT Network as follows :

1.	Request-Response	2.	Publish-Subscribe
3.	Push-Pull	4.	Exclusive Pair

1. Request-Response

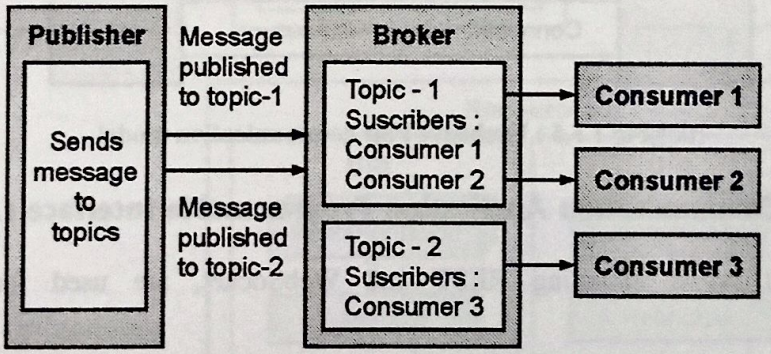
- This paradigm is a communication model in which a client requests data from a server, and the server provides the requested data.
- When a server receives a request, it fetches the requested information, gathers it, creates the response, and delivers it back to the client.
- Each request-response pair is independent of the others as this is a stateless communication model.
- Fig. 1.5.2 shows the Client-Server interaction in the request-response model.



(1A7)**Fig 1.5.2 : Request-response communication model**

2. Publish-Subscribe

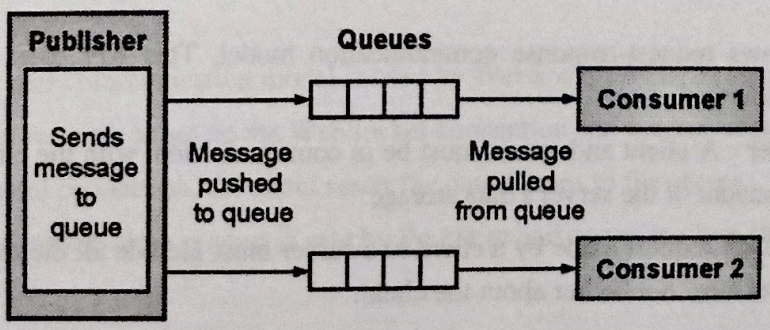
- Publish–Subscribe is a communication model that involves publishers, brokers and consumers.
- Publishers are the source of data. Publishers send the data to the topics which are managed by the broker. Publishers are not aware of the consumers.
- Consumers subscribe to the topics which are managed by the broker.
- When the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers.
- Fig. 1.5.3 shows the Publish-Subscribe communication model.



(1A8)**Fig 1.5.3 : Publish-Subscribe communication model**

3. Push-Pull

- In this model publishers push the data in the queue and this pushed data is pulled from the queue by consumers as shown in Fig. 1.5.4.

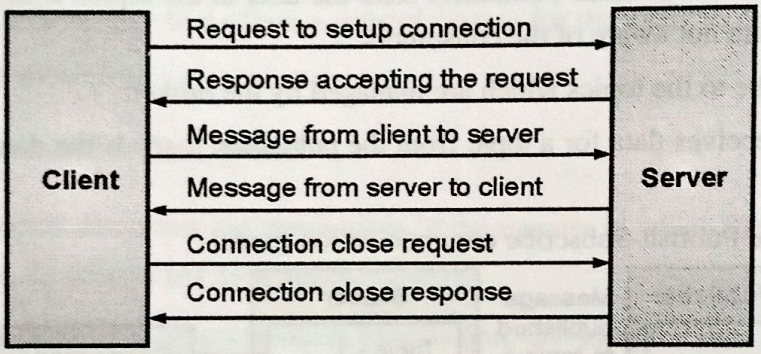


(1A9)**Fig 1.5.4 : Push-pull communication model**

- Here, publishers do not need to know about the consumers.
- Queues help in decoupling the messaging between the producers and consumers.
- Additionally, queues function as a buffer that is useful when there is a discrepancy between the rates at which data is produced and consumed.

4. Exclusive Pair

- It uses a persistent connection between the client and server and is a bidirectional full duplex communication model.
- Once the connection is set up it remains open until the client sends a request to close the connection.
- After connection setup, client and server can send the messages to each other as shown in Fig. 1.5.5.



(1A10)**Fig 1.5.5 : Exclusive Pair communication model**

1.5.3 IoT Communication Application Programmable Interface (API)

In IoT, several APIs, including REST and WebSocket, are used for server-to-system communication.

Representational State Transfer (REST)-based communication APIs

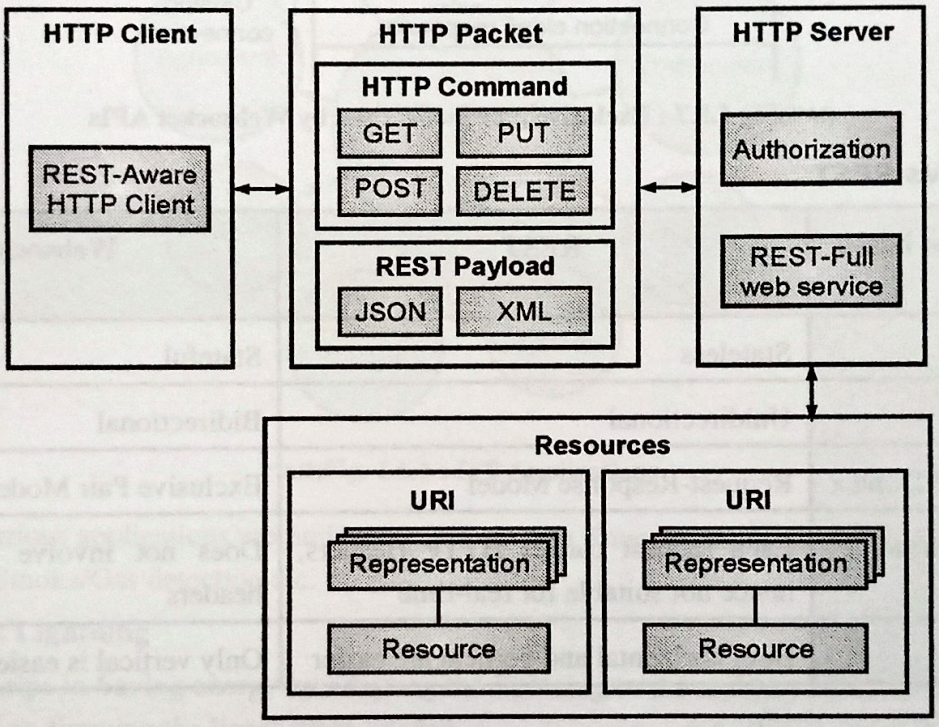
- REST API uses a set of architectural principles used to design web services and web APIs as shown in Fig. 1.5.6.
- These APIs focus on the systems' resources and how the resouce states are addressed and transferred.
- These APIs follows request-response communication model. This API uses some architectural constraints as follows :
  - (1) **Client-Server** : A client and server must be in communication, with the client's user interface being independent of the server's data storage.
  - (2) **Stateless :** Each request made by a client to a server must include all the required information and the server must not bother about the client.
  - (3) **Cache-able** : A response should be declared as cacheable or not. If set then another client cache is given the rights to reuse that response data for later.

(4) **Layered System** : In this case, the client making the request need not be aware of whether it is speaking with the real server, a proxy, or any other middleman.

(5) **Uniform Interface** : The mechanism of communication between the client and server must comply with standard interface requirements.

(6) **Code on demand** : REST provides for the extension of client capability through the downloading and executing of scripts or applets.

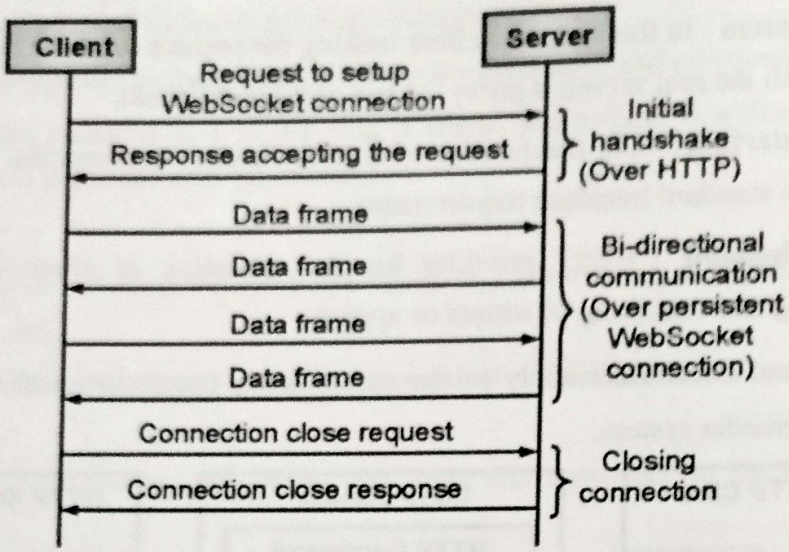
- REST architectural constraints apply to the components, connectors and data elements within a distributed hypermedia system.



(1A11)**Fig. 1.5.6 : Communication with REST APIs**

WebSocket based communication API

- The use of WebSocket APIs enables full-duplex, bi-directional communication between clients and servers.
- The exclusive pair communication model is used by WebSocket APIs, as seen in Fig. 1.5.7.
- Here, the client requests to set up the WebSocket connection and the server responds to the request.
- After a successful connection, the client sends the data frames to the server.
- Finally the connection close request is sent by the client and server to close the connection.



(1A12)**Fig 1.5.7 : Exclusive pair model used by Websocket APIs**

Websocket Vs REST

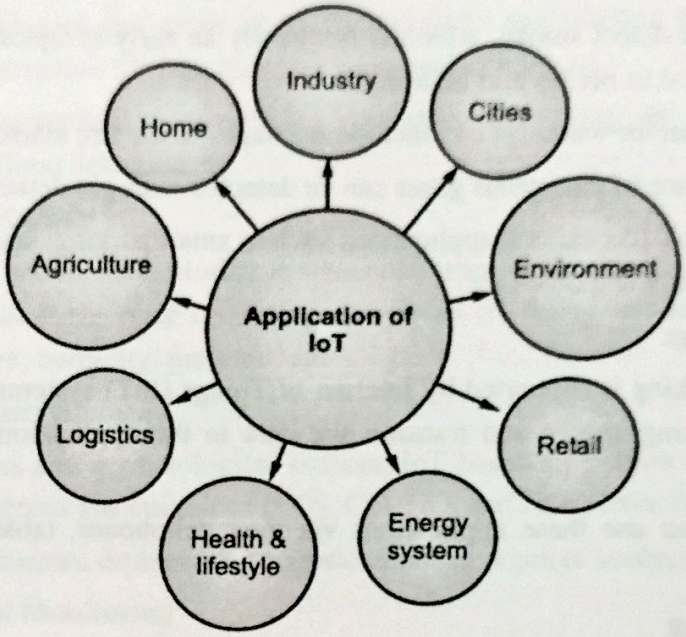
Comparison based on	REST	Websocket
State	Stateless	Stateful
Directional	Unidirectional	Bidirectional
Req-Res/Full Duplex	Request-Response Model	Exclusive Pair Model
Header Overhead	Each request carries HTTP Headers, hence not suitable for real-time	Does not involve overhead of headers
Scalability	Both horizontal and vertical are easier	Only vertical is easier

Physical Design Vs Logical Design

Physical Design	Logical Design
Physical design is highly detailed.	Logical design is a high-level design and doesn't provide any detail.
Physical design is more graphical than textual; however, it can comprise both.	Logical design can be textual, graphic, or both.
A physical design focuses on specific solutions explaining how they are assembled or configured.	A logical design focuses on satisfying the design factors, including risks, requirements, constraints, and assumptions.

1.6 BRIEF REVIEW OF APPLICATIONS OF IOT

Various applications of IoT including home automation, health and fitness tracking, environmental protection, smart cities, energy system, retail, logistics, agriculture and industrial settings as shown in Fig. 1.6.1.



(1A13)**Fig. 1.6.1 : IoT Applications**

(1) Home : Various applications are built using IoT such as Smart lighting, smart appliances, Intrusion detection, Smoke/Gas detection etc.

a) Smart Lightning

This helps in Saving energy by adapting the lightning to the ambient conditions and switching on/off or dimming the lights when needed.

b) Smart Appliances

- There are many appliances in modern homes, including TVs, refrigerators, audio systems, washers and dryers, etc.
- Each appliance has its own controls or remote controls, making it difficult to manage and regulate them.
- Smart appliances simplify management and give users remote access to status information.
- Examples of smart watches and dryers that may be operated remotely and send alerts when a cycle of washing or driving is completed. Or, when an item is running low on stock, a smart refrigerator may keep track of what is being stored and send the user an update.

c) **Home Intrusion**

- Security cameras, PIR sensors, and door sensors are utilized in home intrusion detection systems to detect intrusion and sound the alarm.
- A user may receive alerts in the form of an email or an SMS.

d) **Smoke Gas Detector**

- In order to detect smoke, which is frequently an early symptom of fire, smoke detectors are mounted in homes and buildings.
- Smoke detector warnings can include messages to the fire alarm system.
- The presence of dangerous gases can be detected with gas detectors.

(2) **Cities** : For cities, IoT has various applications such as smart parking, smart roads, structural health monitoring, emergency responses etc.

a) **Smart Parking:**

- Smart parking is supported by Internet of Things (IoT) systems that count the number of open parking spaces and transmit the data to the applications' back ends through the internet.
- Drivers can use these applications via their cellphones, tablets, and in-car navigation systems.

b) **Smart Lighting**

- Energy-saving smart lighting systems for buildings, parks, and roads are possible.
- Smart lighting enables dynamic lighting control and environment-specific lighting adaptation.
- Remote configuration of lighting schedules and intensity is possible with smart lights connected to the internet.

c) **Smart Road**

- Sensor-equipped "smart" roads can offer information on traffic conditions, travel time estimates, and alarms for dangerous road conditions, heavy traffic, and accidents.
- Such knowledge can improve road safety and lessen traffic congestion.

d) **Structural Health Monitoring**

- A network of sensors is used by structural health monitoring systems to track the vibration levels in buildings and bridges.
- It is possible to pinpoint the damage to a structure, find cracks and mechanical breakdown, as well as determine the structure's remaining life, by evaluating the data.
- By using such methods, a structure's impending failure can be forewarned of in advance.

e) **Surveillance**

- To ensure safety and security, surveillance of public transportation, infrastructure, and even cities is necessary.

- It is possible to build a city-wide surveillance system made up of numerous distributed, internet-connected video surveillance cameras.

f) Emergency Response

- Cities' vital infrastructure, including buildings, gas and water pipelines, public transportation, and power substation systems, can be monitored via IoT technologies.
- IoT technologies can assist in producing alerts and limiting their consequences on the vital infrastructure for fire, gas, and water leak directions.

(3) Environment : It has applications consisting of weather monitoring, air and noise pollution, forest fire detection, river flood detection etc.

a) Weather Monitoring

- IoT-based weather monitoring systems can transfer data to cloud-based applications and storage back-ends after collecting data from a variety of associated sensors, such as temperature, humidity, pressure, etc.

b) Air pollution surveillance

- Using gases and dermatological sensors, IoT-based air pollution monitoring systems can track dangerous gas emissions (CO_2, CO, NO, and NO_2) from facilities and autos.
- Making educated decisions on pollution control requires analysis of the collected data.

c) Noise Pollution Monitoring

- Stress and disturbed sleep are two health risks that can result from noise pollution in people.
- Monitoring noise pollution can be used to create noise maps for cities.
- Urban planners can use maps of urban noise to aid in regulating noise levels around residential areas, schools, and parks.
- The IoT-based smart metering system for noise pollution makes use of several noise monitoring stations that are placed around a metropolis.

d) Forest Fire Detection

- Various factors, such as lightning, human error, volcanic eruptions, and sparks from rock falls, can trigger forest fires.
- Forest fires that are detected early can have their impact reduced.
- An early warning system for forest fires gives possibility of forest fires a head start.

e) River Flood Detection

- A river flood monitoring system is described that detects river and weather conditions using wireless sensor nodes fitted with various sensors.
- Monitoring applications raise alert when a rapid increase in water level and flow rate is noticed.

- The systems contain a water level monitoring module and a data processing module that give raw data, forecasted data, and video feeds for flood information.

(4) **Energy** : Smart grid, renewable energy system, prognostics etc. included in this category.

a) **Smart Grid**

- The term "smart grid" refers to an electrical grid-integrated data communication network that gathers and analyses real-time data on electricity transmission, distribution, and consumption.
- Electricity generation (centralised or dispersed models) storage (or conversion of the energy into various forms), distributions, and equipment health data are all collected by smart grids.

b) **Smart Meters**

- Smart metres have the ability to remotely control electricity generation and consumption, as well as remotely turn off the supply when necessary.
- Smart metres can stop power thefts by evaluating data on power generation, transmission, and consumption.

c) **Prognostics**

- By assessing the degree of deviations from the system's typical operating patterns, IoT-based prognostic real-time health management systems may forecast the performance of energy or mechanical systems.

(5) **Retail** : Some of the applications comprises smart payments, inventory management, smart vending machines etc.

a) **Inventory Management**

- A radio frequency identification Internet of Things system RFID tags can be used to control inventory and keep the appropriate amounts of inventory.
- The products can be tracked in real time thanks to RFID tags that are attached to them, allowing inventory levels to be precisely calculated and low-stock items to be restocked.
- RFID scanners placed in the warehouse or on the shelves of retail establishments can be used for tracking.

b) **Smart Payments**

- Contact list payments using smart payment solutions are enabled by Bluetooth and NFC technologies.
- Smartphones and other gadgets can communicate with one another using a set of standards called near field communication by touching or bringing them close to one another.

c) Smart Vending Machines

- Smart vending machines with internet connections enable contactless payments, promotions, and remote monitoring of inventory levels.
- Users can record their preferences and preferred products in smart vending machines, which employ sensors to track their operation and send data to the cloud for preventative maintenance.
- Smart vending machines can share their inventory levels and connect with other vending machines nearby so that customers can be directed to the closest units in the event that a product runs out in one machine.

(6) Logistics : Applications in this category are route generation and scheduling, fleet tracking, shipment monitoring, remote vehicle tracking/diagnosting etc.

a) Route Generation and Scheduling

- The availability of vehicles allows the transportation system to offer new services like advanced route guidance and dynamic vehicle routing that anticipate customer demand for pickup and delivery issues.
- Another example is route generation and scheduling systems that are candidate end-to-end using combinations of road patterns and transportation smooth and feasible schedule.

b) Fleet Monitoring

- GPS technology is used by vehicle fleet monitoring systems to track the locations of vehicles in real time.
- When planned routes are diverted, alerts may be generated.
- The data on vehicle locations and routes can be combined and evaluated to find supply chain bottlenecks caused by things like traffic conditions, route of various supply chain parts, and alternate route generations.
- The system may examine messages provided from the cars to identify unexpected occurrences

c) Shipment Monitoring

- Transportation systems can monitor the conditions inside containers via shipment monitoring solutions.
- To avoid food deterioration, for instance, containers containing fresh vegetables might be watched.

d) Remote Vehicle Diagnostic

- Remote car diagnostic devices can identify vehicle defects and alert the driver to potential problems.
- These diagnostic systems integrate on-board diagnostic systems with IoT devices to collect data on vehicle operation, including speed, engine RPM, coolant temperature, fault code number, and status of the various vehicle subsystems.

(7) **Agriculture** : Several applications of agriculture are smart irrigation, greenhouse control etc.

a) **Smart Irrigation**

- Smart irrigation techniques can increase crop productivity.
- Smart irrigation systems use IoT devices with soil moisture sensors to
- Measure the soil's moisture content and only allow water to flow via irrigation pipes when the level of moisture falls below a predetermined threshold.
- Measurements of the moisture level are also collected by smart irrigation systems and stored on a computer or in the cloud, where the data is evaluated to determine when to water plants.

b) **Green House Control**

- To offer the greatest circumstances for plant growth, the climatological conditions inside a greenhouse can be monitored and managed.
- Sensors are used to measure the levels of temperature, humidity, soil moisture, light, and carbon dioxide, and actuation devices are used to automatically change these variables' climatological conditions.
- For better administration and maintenance of agricultural produce, the system uses a wireless sensor network to track and adjust agricultural characteristics including temperature and humidity in real-time.

(8) **Industry** : Numerous applications are included in this such as machine diagnosis and prognosis, indoor air quality monitoring etc.

a) **Machine Diagnosis and Prognosis**

- Machine prognosis is the process of estimating a machine's performance based on data analysis, present operating conditions, and the degree to which they deviate from ideal operating conditions.
- Finding the root causes of a machine fault is referred to as machine diagnostic.
- Machine sensors can keep an eye on things like operational temperatures and vibration levels.

b) **Indoor Air Quality Monitoring**

- Using a variety of gas sensors, an IoT-based gas monitoring system can assist in monitoring the interior air quality.
- For various locales, the quality of the indoor air can change.
- IoT-based wireless sensor networks can identify more dangerous areas so that remedial action can be taken to ensure adequate ventilation.

(9) **Health and Lifestyle** : Various applications such as wearable electronics, health and fitness monitoring etc.

a) Health and fitness monitoring

- Wearable Internet of Things (IoT) devices that are low-impact and continuously monitor physiological data can aid in continuous health and fitness monitoring.
- These accessories may be integrated into wristbands or take other shapes.

b) Wearable Electronics

- Wearable Electronics such as smart watches smart glasses wristband and fashion electronics (with electronic integrated in clothing and accessories, example Google glass for Moto 360 smart watches provide various functions and future to assist us in our daily activities and making us lead healthy Lifestyle.
- Smart watches allow users to search the internet, play audio and video files, make calls with or without paired mobile phones, play games, and use a variety of mobile applications.
- Smart watches enable users to take pictures and record videos, get directions, check flight status, and perform voice searches on the internet.

1.7 SMART OBJECT

1.7.1 Definition

Smart objects are any physical things with embedded technology that can communicate with one another or an outside agent, detect their surroundings, and/or interact with them in a meaningful way.

1.7.2 Characteristics

There are mainly four characteristics of smart objects described as follows :

1. Processing unit

- This unit acquires data, processes it and analyzes sensing information received by the sensor(s).
- Also sends coordinating control signals to any actuators, and controls a variety of functions on the smart object, including the communication and power systems.
- The most common processing unit is a microcontroller because of its small form factor, flexibility, programming simplicity, ubiquity, low power consumption, and low cost.

2. Sensor(s) and/or actuator(s)

A smart object is capable of interacting with the physical world through sensors and actuators.

3. Communication Unit

- This unit is responsible for connecting a smart object with other smart objects and the outside world (via the network).
- Communication devices for smart objects can be either wired or wireless.

MODULE

2

IoT Architecture

Syllabus

Drivers Behind New Network Architectures : Scale, Security, Constrained Devices and Networks, Data, Legacy Device Support

Architecture : The IoT World Forum (IoTWF) Standardized Architecture :Layer 1-7, IT and OT Responsibilities in the IoT Reference Model, Additional IoT Reference Models A Simplified IoT Architecture

The Core IoT Functional Stack : Layer 1-3 , Analytics Versus Control Applications , Data Versus Network Analytics Data Analytics Versus Business Benefits , Smart Services,

IoT Data Management and Compute Stack : Fog Computing , Edge Computing ,The Hierarchy of Edge, Fog, and Cloud.

Self-learning Topics : Brief review of applications of IoT: Connected Roadways , Connected Factory, Smart Connected Buildings , Smart Creatures etc.

2.1 INTRODUCTION

- Internet of Things (IoT) is a system of interrelated, internet-connected objects which are able to collect and transfer data over a wireless network without human intervention. For example, smart fitness bands or watches, driverless cars or drones, smart homes that can be unlocked through smartphones and smart cars, etc.
- Due to the ever-evolving nature of IoT devices, and the wide diversity of sensors, there is no one-size-fits-all architecture for IoT projects. However, some of the building blocks will be similar from project to project.
- First, you will need to build with scalability in mind. The amount of data that you will collect over time will take on enormous proportions and you will need a platform that can accommodate this in the long run.
- You will also need to ensure that you have high availability at any given time. Having system failures could make you lose some business in the best case, or could have fatal consequences in the worst cases.
- Finally, you will need a system that is flexible enough to accommodate quick and frequent changes. As your architecture evolves, or your business needs change, you will need to iterate quickly without breaking the existing architecture.
- Administrators use IoT architecture to manage and support IoT devices. IoT devices can be anything from an internet-connected light bulb to pressure safety sensors in a chemical plant.
- These devices use small sensors to collect data about their environment and send that data to a server for processing. Servers process this data to create information and insights for businesses. Many times this information is used to automate tasks that improve uptime and efficiency across multiple business systems.
- IoT architecture makes this all possible by ensuring data gets where it needs to and is processed correctly. Without proper IoT architecture, networks would become unreliable, defeating the entire purpose of investing in IoT in the first place.

2.2 DRIVERS BEHIND NEW NETWORK ARCHITECTURES

This begins by comparing how using an architectural blueprint to construct a house is similar to the approach we take when designing a network. Take a closer look at some of the differences between IT and IoT networks, with a focus on the IoT requirements that are driving new network architectures, and considers what adjustments are needed. Following are the IoT Architecture drivers

1. Scale,
2. Security,
3. Constrained Devices and Networks
4. Data,
5. Legacy Device Support

(1) Scale

- Generally an IT network is on the scale of order of several thousand devices—typically printers, mobile wireless devices, laptops, servers, and so on.
- The traditional three-layer campus networking model, supporting access, distribution, and core (with sub architectures for WAN, Wi-Fi, data center, etc.), is well understood.
- IoT introduces a model where an average-sized utility, factory, transportation system, or city could easily be asked to support a network of thousand end point to few million end points.
- Based on scale requirements of this order, IPv6 is the natural foundation for the IoT network layer.

(2) Security

- Already today machine and networks are targeted with malicious attacks using vulnerabilities in networked machines,
- Brute force attack , ransomware attack and many such which may effect the systems network badly. Even if there is world III then it amu be due to security in CyberSpace.
- The frequency and impact of cyber attacks in recent years has increased highly. This is now need of time and Responsibility of IT department for Protecting corporate data from intrusion and theft.
- IT departments also to protect servers, applications, and the network, setting up defense-in-depth models with layers of security designed to protect the cyber crown jewels of the corporation.
- Even if all the efforts taken to protect networks and data, hackers are smart enough to penetrate trusted networks.
- In IT networks, the circumference of defense is often the perimeter firewall. It is very critical to place any endpoint outside the firewall.
- But if the IoT endpoints are located in wireless sensor networks and if they are using unlicensed spectrum and are not only visible to the world and also accessible and widely distributed in the field.
- Traditional models of IT security are can be easily attacked by IoT System. IoT systems require consistent mechanisms of authentication, encryption, and intrusion prevention techniques that understand the behavior of industrial protocols and can respond to attacks on critical infrastructure.
- For optimum security, IoT systems must:
    - Be able to identify and authenticate all entities involved in the IoT service (that is, gateways, endpoint devices, home networks, roaming networks, service platforms)
    - Ensure that all user data shared between the endpoint device and back-end applications is encrypted

- Comply with local data protection legislation so that all data is protected and stored correctly
- Utilize an IoT connectivity management platform and establish rules-based security policies so immediate action can be taken if anomalous behavior is detected from connected devices
- Take a holistic, network-level approach to security

(3) Constrained Devices and Networks

- IoT sensors are designed for a single task, and this sensor are small and inexpensive. They have limited power, CPU, and memory, and they transmit only when there is something important.
- Because of the massive scale of these devices and the large, uncontrolled environments where they are usually deployed, the networks that provide connectivity also tend to be very lossy and support very low data rates.
- On the other and IT networks, which are speeds(Many gigabytes) and endpoints with powerful CPUs. In case if performance constraints, then network can be simply upgraded to a faster network.
- If too many devices are on one VLAN and are impacting performance, simply a new VLAN can be designed and continue to scale as much needed.
- However, this approach cannot meet the constrained nature of IoT systems. IoT requires a new technology of connectivity that meet both the scale and constraint limitations.

(4) Data

- IoT devices/Sensor generate a huge amount of data. In general, most IT shops don't really care much about the unstructured chatty data generated by devices on the network.
- The data generated by IT network is not that important as compared to data generated by the IoT network. The IoT the data can used to enhance the businesses to deliver new IoT services that enhance the customer experience, reduce cost, and deliver new revenue opportunities.
- The IoT-generated data is mostly unstructured, these data is to be processes through analytics, now its needed to create new business models to process this Data.
- Lets take example of smart city with a few hundred thousand smart streetlights , traffic signal and other sensors, all connected through an IoT network. All this information communicated between the network modules and the control centers, data pattern can help us in predicting when lights , signals and sensors need to be replaced or whether they can be turned on or off at certain times, which will save operational expense.
- But when all this data is combined, its very difficult to manage and analyze effectively. IoT systems are designed to move data consumption throughout the architecture, both to filter and reduce unnecessary data going upstream and to provide the fastest possible response to devices when necessary.

(5) Legacy Device Support

- Upgrading system in IT network is easy like OS version, Protocols support can be upgraded either automatically or manually.
- In IoT systems, end devices are likely to be on the network for a very long time sometimes decades. As IoT networks are deployed, they need to support the older devices already present on the network, as well as devices with new capabilities. In many cases, legacy devices are so old that they don't even support IP.
- For example, a factory may replace machines only once every 20 years or perhaps even longer! It does not want to upgrade multi-million-dollar machines just so it can connect them to a network for better visibility and control.
- However, many of these legacy machines might support older protocols, such as serial interfaces, and use RS-232. In this case, the IoT network must either be capable of some type of protocol translation or use a gateway device to connect these legacy endpoints to the IoT network.

2.2.1 IoT Architectural Drivers

Table 2.2.1 : IoT Architectural Drivers

Challenge	Description	IoT Architectural Change Required
Scale	The massive scale of IoT end-points (sensors) is far beyond that of typical IT networks	The IPv4 address space has reached exhaustion and is unable to meet IoT's scalability requirements. Scale can be met only by using IPv6. IT networks continue to use IPv4 through features like Network Address Translation (NAT).
Security	IoT devices, especially those on wireless sensor networks (WSNs), are often physically exposed to the world	Security is required at every level of the IoT network. Every IoT endpoint node on the network must be part of the overall security strategy and must support device-level authentication and link encryption. It must also be easy to deploy with some type of a zero-touch deployment model.
Devices and networks constrained By power, CPU, Memory and Link speed	Due to the massive scale and longer distances, the networks are often constrained, lossy, and capable of supporting only minimal data rates (tens of bps to hundreds of Kbps).	New last-mile wireless technologies are needed to support constrained IoT devices over long distances. The network is also constrained, meaning modifications need to be made to traditional network-layer transport mechanisms.
The massive volume of data generated	The sensors generate a massive amount of data on a daily basis, causing network bottlenecks and slow analytics in the cloud.	Data analytics capabilities need to be distributed throughout the IoT network, from the edge to the cloud. In traditional IT networks, analytics and applications typically run only in the cloud.

Challenge	Description	IoT Architectural Change Required
Support for legacy devices	An IoT network often comprises a collection of modern, IP-capable endpoints as well as legacy, non-IP devices that rely on serial or proprietary protocols.	Digital transformation is a long process that may take many years, and IoT networks need to support protocol translation and/or tunneling mechanisms to support legacy protocols over standards-based protocols, such as Ethernet and IP.
The need for data to be analyzed in real time	Whereas traditional IT networks perform scheduled batch processing of data, IoT data needs to be analyzed and responded to in real-time	Analytics software needs to be positioned closer to the edge and should support real-time streaming analytics. Traditional IT analytics software (such as relational databases or even Hadoop), are better suited to batch-level analytics that occur after the fact.

2.3 COMPARING IOT ARCHITECTURES

- The challenges and requirements of IoT systems have driven a whole new discipline of network architecture. In the past several years, architectural standards and frameworks have emerged to address the challenge of designing massive-scale IoT networks.
- In effort to standardize the rapidly growing field of machine to machine (M2M) communications, the European telecommunications Standards Institute (ETSI) created the M2M Technical committee in 2008.
- The goal of this committee was to create a common architecture that would help accelerate the adoption of M2M applications and devices. Over the time, the scope has expanded to include the Internet of Things. Recognizing this need, in 2012 ETSI and 13 other founding members launched one M2M as a global initiative designed to promote efficient M2M communication systems and IoT.

2.3.1 The oneM2M IoT Standardized Architecture

- The one M2M architecture divides IoT functions into three major domains as shown in Fig. 2.3.1.
  - the application layer,
  - the services layer and
  - the network layer.
- While this architecture is very rich and promotes interoperability through IT-friendly APIs and supports a wide range of IoT technologies. Let's see each of these domains in turn:

1. Applications Layer

- The one M2M architecture gives major attention to connectivity between devices and their applications.
- This domain includes the application layer protocols and attempts to standardize northbound API definitions for interaction with Business Intelligence (BI) Systems.
- Applications tend to be industry specific and have their own sets of data models and thus they are shown as vertical entities.

2. Services Layer

- This layer is shown as a horizontal framework across the vertical industry applications.

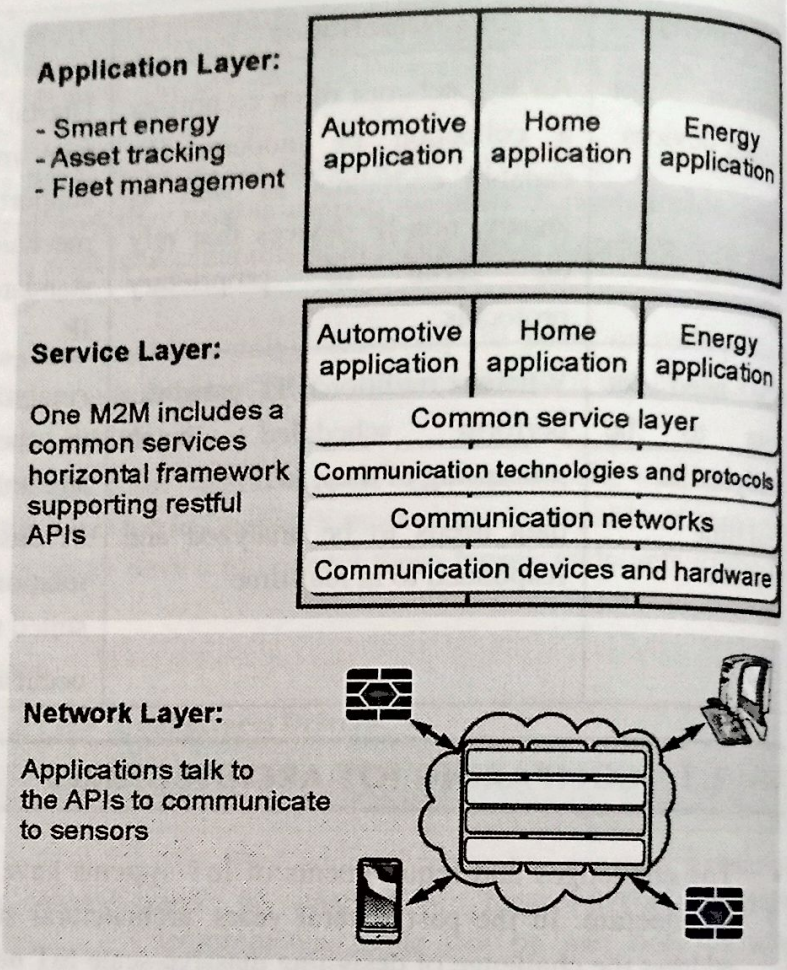


Fig. 2.3.1 : Main Element of the one M2M IoT architecture

- At this layer, horizontal modules include the physical network that the IoT applications run on, the underlying management protocols, and the hardware.
- Examples include backhaul communications via cellular, MPLS (Multiprotocol label switching) networks, VPNs and so on. Riding on top is the common services layer.
- This conceptual layer adds APIs and middleware supporting third party services and applications.

3. Network Layer

- This is the communication domain for the IoT devices and endpoints.
- It includes the devices themselves and the communication network that links them. Embodiments of this communication infrastructure includes wireless mesh technologies such as IEEE 802.15.4 and wireless point to multipoint systems IEEE 801.11ah.
- In many cases, the smart (and sometimes not so smart) devices communicate with each other. In other cases, machine-to-machine communication is not necessary, and the devices communicate through a field area network (FAN) to use case specific apps in the IoT application domain.

- Therefore, the device domain also includes the gateway device, which provides communications up into the core network and acts as a demarcation point between the device and network domains.

2.3.2 The IoT World Forum (IoTWF) Standardized Architecture

- In 2014 the IoTWF architectural committee (led by Cisco, IBM, Rockwell Automation, and others)published a seven-layer IoT architectural reference model shown in Fig.2.3.2.
- There are various IoT reference models but the one proposed by the IoT World Forum offers a clean, simplified perspective on IoT and includes edge computing, data storage, and access.
- It provides an expressed way of visualizing IoT from a technical perspective briefly and clearly.
- Each of the seven layers is broken down into specific functions, and security encompasses the entire model. Fig. 2.3.2. shows details the IoT Reference Model published by the IoTWF.

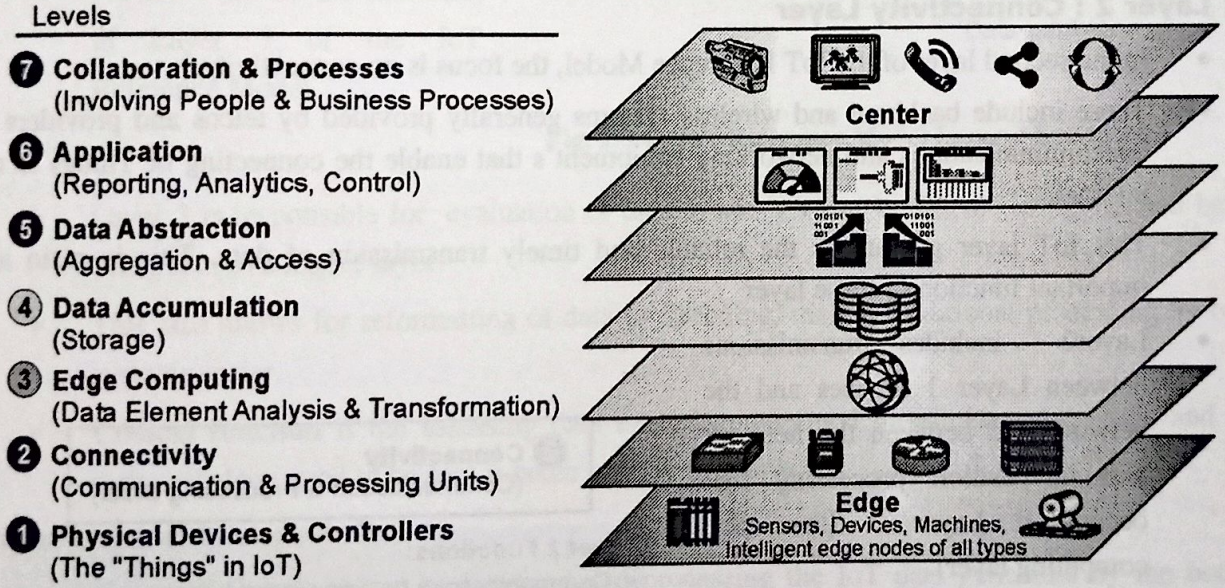


Fig. 2.3.2 : IoT Reference Model Published by the IoT World Forum

- As shown in Fig. 2.3.2, the IoT Reference Model defines a set of levels with control flowing from the center (this could be either a cloud service or a dedicated data center), to the edge, which includes sensors, devices, machines, and other types of intelligent end nodes.
- In general, data travels up the stack, originating from the edge, and goes northbound to the center.
- Using this reference model, we are able to achieve the following:
  - Decompose the IoT problem into smaller parts
  - Identify different technologies at each layer and how they relate to one another
  - Define a system in which different parts can be provided by different vendors
  - Have a process of defining interfaces that leads to interoperability
  - Define a tiered security model that is enforced at the transition points between levels

The following sections look more closely at each of the seven layers of the IoT Reference Model.

(a) Layer 1 : Physical Devices and Controllers Layer

- The first layer of the IoT Reference Model is the physical devices and controllers layer.
- In this layer connected devices, sensors and machines at the edge with which other devices, sensors and machines or people interact.
- This layer includes the various endpoint devices and sensors that send and receive information and its like home to the "things" in the IoT,.
- There are various size of "things" which ranges from miniature(microscopic) Sensors to big(giant) machines used in factory.
- Generation of data is primary function and also they are capable of being queried and/or controlled over a network.

(b) Layer 2 : Connectivity Layer

- In the second layer of the IoT Reference Model, the focus is on connectivity.
- These include backhaul and wireless systems generally provided by telcos and providers of telecommunications and networking equipment's that enable the connecting of Things to the internet.
- This IoT layer guarantees the reliable and timely transmission of data. This is main and important function of these layer
- Layer2 includes transmissions between Layer 1 devices and the network and between the network and information processing that occurs at Layer 3 (the edge computing layer).
- The connectivity layer hold within all networking elements of IoT and doesn't really distinguish between the last-mile network, gateway, and backhaul networks.
- Fig. 2.3.3 shows in details functions of the connectivity layer

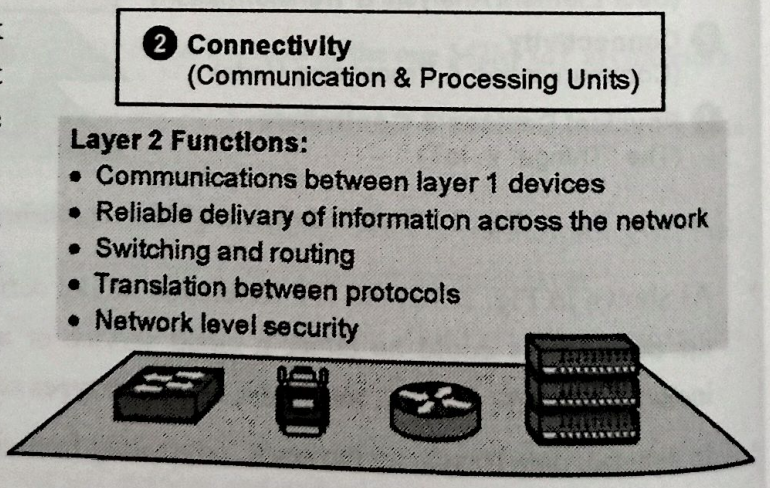


Fig. 2.3.3 : IoT Reference Model Connectivity Layer Functions

(c) Layer 3: Edge Computing Layer

- This layer allows for a highly scalable low latency architecture by enabling computing close to where it is needed.
- Edge computing is often referred to as the "fog" layer.

- **Emphasis**: On data reduction and converting network data flows into information that is ready for storage and processing by higher layers.
- **Basic principles** of this reference model is that information processing is initiated as early and as close to the edge of the network as possible.
- Fig 2.3.4 shows the functions of Layer 3 of the IoT Reference Model.

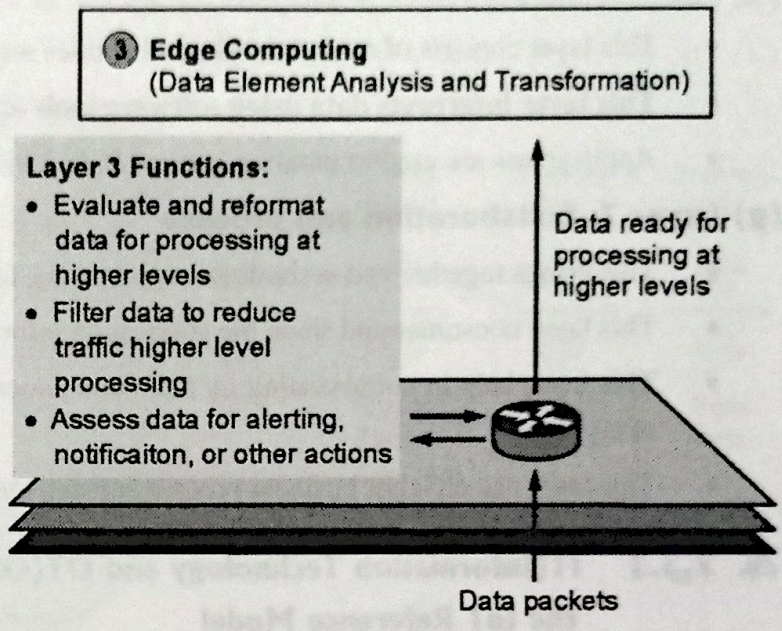


Fig. 2.3.4 : IoT Reference Model Layer 3 Functions

- Layer 3 is responsible for evaluation of data to see if it can be filtered or aggregated before being sent to a higher layer.
- This also allows for reformatting of data or decoding, making additional processing by other systems easier.
- **Critical function** if the assessing data exceeds predefined thresholds are crossed and any action or alerts need to be sent is taken care by this layer.

Upper Layers: Layers 4–7

- The upper layers deal with handling and processing the IoT data generated by the bottom layer.

(d) Layer 4: Data Accumulation

- This layer allows the storage of captured application data in databases, files, or preferences, in internal or removable storage which can used when needed.
- This layer also converts event based to query based processing

(e) Layer 5: Data Abstraction

- This layer allows the aggregation of data from distributed nodes at the edge and in the cloud
- This layer restores multiple data formats and ensure consistent semantics from various source.
- This layer confirms that the data set is complete and consolidates data into one place or multiple data stores using Virtualization

(f) Layer 6 : Application Layer

- This layer consists of essential tasks and entities required to achieve a business goal Page
- **This layer interprets data using software tools and applications.**
- Applications are used to monitor ,control and provide reports based on the analysis of data.

(g) Layer 7: Collaboration and process

- This brings together and orchestrates applications to achieve a business process
- This layer consumes and share the application information.
- This layer help in collaborating on and communicating IoT information which often requires many steps.
- This layer can changes business process and deliver the benefits of IoT

2.3.3 IT(Information Technology and OT(Open Technology) Responsibilities in the IoT Reference Model

(a) Operational Technology (OT) Network

- OT or Operational technology is a category of a computing system which process operational data such as telecommunication, technical components, computers and it is used to monitor devices, various industrial process and some of the industry events and accordingly make adjustments if required in an industry or enterprise.
- In other words, OT uses the combination of software and hardware which can be used to perform real-time operations to detect if there is any change occurred during the whole process which can be done by directly controlling the industrial equipment and some of the enterprise's events which make them reliable and increase their rate of availability and reliability.
- OT systems ensure the safety of industrial operations by continually monitoring them and also helps to support infrastructure such as manufacturing, defense utilities, etc. OT network works at the industrial level to process the operational data of any organization.

(b) Information Technology (IT) Network

- IT or Information Technology deals with the systems mainly computers and telecommunication for performing various operations like for giving input, for storing, recovering, transmitting, manipulating and protecting data or information so that data can be exchanged among different organizations.
- IT Network encompasses hardware (computers, physical servers, and network equipment), software (operating systems, applications), and peripheral equipment. Instead of performing a static set of functions, IT can be adjusted and re-programmed in so many different ways to fulfill the evolving and changing applications, business requirements and user needs.

- IT network in any industry is used to manage the computer systems and companies data in a more secure way.

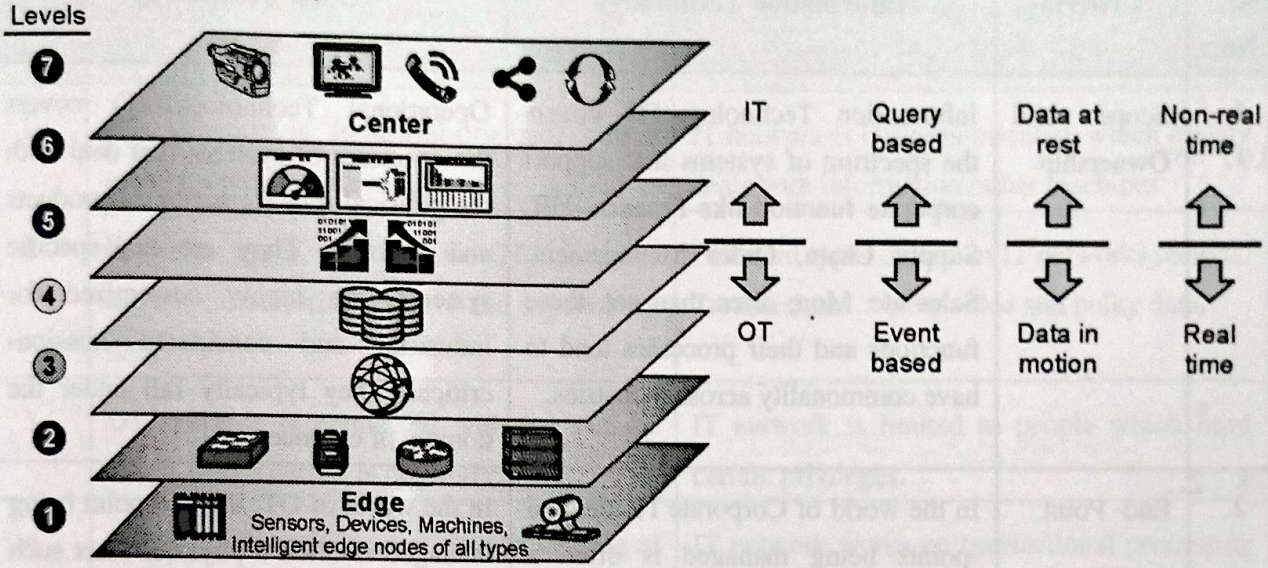


Fig. 2.3.5 : A differentiation point between IT and OT in the IoT Reference Model framework

- As Shown in Fig. 2.3.5 IoT systems have to cross many boundaries beyond the functional layers.
- The bottom of the stack is domain of OT. For an industry like oil, and gas, this includes sensors and devices connected to pipelines, oil rigs, refinery machinery. Also industries were manufacturing is needing controlled atmosphere , Precision Pharming were sensors are used to track different parameters.
- The top of the stack is in the IT area and includes things like the servers, databases, and applications, all of which run on a part of the network controlled by IT.
- Long back OT and IT have been independent and very little interaction between both. Since IoT is evolving the whole paradigm is now changing.
- In the bottom OT layers the devices/Sensors generate real-time data at their own rate—also huge amount of data is generated on daily basis. This huge amount of data is transmitted in the IoT network, the volume of data decides applications at the top layer will be able accepting data at the rate required.
- In order to synchronize between and data generated by sensors and accepted by application, data has to be buffered or stored at certain points within the IoT stack. Data management in this way throughout the stack helps the upper four layers to handle data at their own speed.
- As a result, the real-time "data in motion" close to the edge has to be organized and stored so that it becomes "data at rest" for the applications in the IT tiers. The IT and OT organizations need to work together for overall data management.

(c) Comparing OT and IT networks

Sr. No.	Criteria	Information Technology	Open Technology
1.	Scope and Ownership	Information Technology(IT) covers the spectrum of systems that support corporate function like Finance, HR, Supply Chain, Order Management, Sales etc. More often than not, these functions and their processes tend to have commonality across industries.	Operational Technology(OT) covers the spectrum of systems that deal with the physical transformation of products and services. They are task-specific systems, are highly customized for industries and considered mission-critical. They typically fall under the domain of engineering
2.	End- Point	In the world of Corporate IT, the end -points being managed is often a human(whose job tends to be information-intensive) using a computing device(that has been relatively homogenous until the recent and growing BYOD [bring your own device] trend)	In the world of OT, the end-point being managed is often a physical asset such as pumps, motors, conveyors, valves, for kifts etc. where these things come in all shapes ,sizes, level of complexity version and vintage
3.	Focus	The software applications that make up IT portfolio are people centric in the sense that they help people "make money" by managing and coordinating the higher-level processes and transactions of the business	In contrast, most of the software application in OT's portfolio are "thing-centric" in the sense that they help "make product" by controlling the physical equipment with a great deal of precision (and safety), where the humans role is supervisory (as automation increases)
4.	Architecture	Besides being pervasive in our personal lives, IT is a relatively standardized world and that is far more homogenous than OT) IT also tends to adapt far more quickly to multiple computing trends, from PCs to internet to mobility, all of which have broadly shaped todays corporate IT strategy.	In contrast, OT is filled with silos of proprietary architectures because of its tasks-specific nature. For example, a refinery is designed so it can run continuously for 5+ years before it is shut down for maintenance. In Other words, reliability can often trump innovation, open architecture, interoperability etc.

(d) Difference between OT Network and IT Network

Sr. No.	OT Network	IT Network
1.	OT network is industrial-oriented, which mainly interacts with machines.	IT network is business-oriented, which mainly deals with information rather machines.
2.	Different types of data in OT networks include : monitoring, control and supervisory data.	Different types of data in IT networks include: Transactional, voice, video and bulky data.
3.	OT network is connected with the outside world whose access is not limited.	IT network is limited to people which have certain privileges.
4.	OT network works on real-time processing of data.	IT network works on transactional processing of data.
5.	OT network may have risk regarding the information.	IT network may have automation risk.
6.	OT network failure can result in end-of life.	IT network failure can result in loss of data.
7.	OT has less changing environment as the requirements are not frequently changing.	IT has frequently changing environment.
8.	OT network requires network upgrades only during operational maintenance windows.	IT network often requires network upgrades.
9.	If there is any disturbance in OT network, it will directly impact the overall business.	IT network failure can be business impacting, and it depends on industry.
10.	OT network controls physical access to any device.	IT network ensures security by authenticating the devices and users to the network

2.3.4 Additional IoT Reference Models

- There are several other reference models exist apart the two model discussed.
- These models are endorsed by various organizations and standards bodies and are often specific to certain industries or IoT applications.
- Table 2-3 gives details of these additional IoT reference models.

IoT reference Model	Description
Purdue Model for Control Hierarchy	• The Purdue Model for Control Hierarchy (see www.cisco.com/c/en/us/td/docs/solutions/Verticals/EttF/EttFDIG/ch2_EttF.pdf) is a common and well-understood model that segments devices and equipment into hierarchical levels and functions. • It is used as the basis for ISA-95 for control hierarchy, and in turn for the IEC-62443 (formerly ISA-99) cyber security standard. • It has been used as a base for many IoT-related models and standards across industry.
Industrial Internet Reference Architecture (IIRA) by Industrial Internet Consortium (IIC)	• The IIRA is a standards-based open architecture for Industrial Internet Systems (IISs). • To maximize its value, the IIRA has broad industry applicability to drive interoperability, to map applicable technologies, and to guide technology and standard development. • The description and representation of the architecture are generic and at a high level of abstraction to support the requisite broad industry applicability. • The IIRA distills and abstracts common characteristics, features and patterns from use cases well understood at this time, predominantly those that have been defined in the IIC.
Internet of Things–Architecture (IoT-A)	• IoT-A created an IoT architectural reference model and defined an initial set of key building blocks that are foundational in fostering the emerging Internet of Things. • Using an experimental paradigm, IoT-A combined top-down reasoning about architectural principles and design guidelines with simulation and prototyping in exploring the technical consequences of architectural design choices.

2.4 A SIMPLIFIED IOT MODEL

- An IoT framework discussed here highlights the fundamental building blocks that are common to most IoT systems and which is intended to help you in designing an IoT network.
- This framework is presented as two parallel stacks shown in Fig. 2.4.1.
  - The IoT Data Management and Compute Stack and
  - the Core IoT Functional Stack.

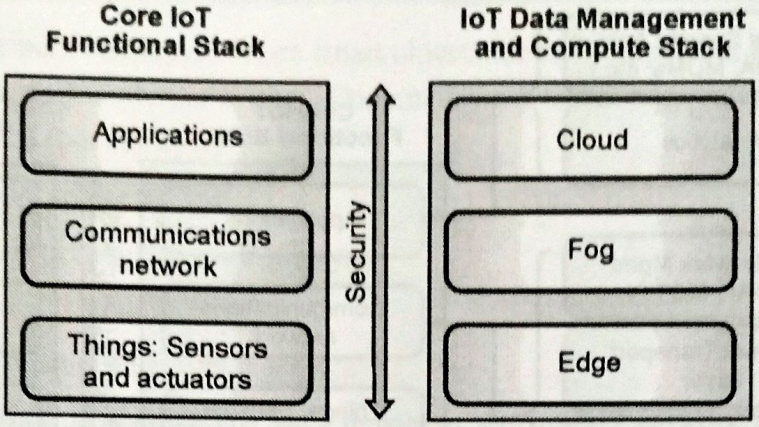


Fig. 2.4.1 : Simplified IoT Architecture

- Almost all IoT model includes core layers including
  - "things"
  - a communications network, and
  - applications.
- The framework separates the core IoT and data management into parallel and aligned stacks, allowing you to carefully examine the functions of both the network and the applications at each stage of a complex IoT system. This separation gives you better visibility into the functions of each layer.
- The **presentation** of the Core IoT Functional Stack in three layers is to simplify understanding of the IoT architecture into its most foundational building blocks.
- The **network communications** layer of the IoT stack itself involves a significant amount of detail and incorporates a vast array of technologies.
- The different types of IoT sensors and there are many different ways to connect them to a network. The network communications layer needs to add all these together.
- It also offer gateway and backhaul technologies, and ultimately bring the data back to a central location for analysis and processing.
- As compared to most IT networks, the applications and analytics layer of IoT doesn't necessarily exist only in the data center or in the cloud.
- This is unique challenges and requirements of IoT, it is often necessary to deploy applications and data management throughout the architecture in a tiered approach, allowing data collection, analytics, and intelligent controls at multiple points in the IoT system.
- The data management is aligned with each of the three layers of the Core IoT Functional Stack.
- The three data management layers shown in Fig. 2.4.2. are
  - the edge layer (data management within the sensors themselves),
  - the fog layer (data management in the gateways and transit network), and
  - the cloud layer (data management in the cloud or central data center).

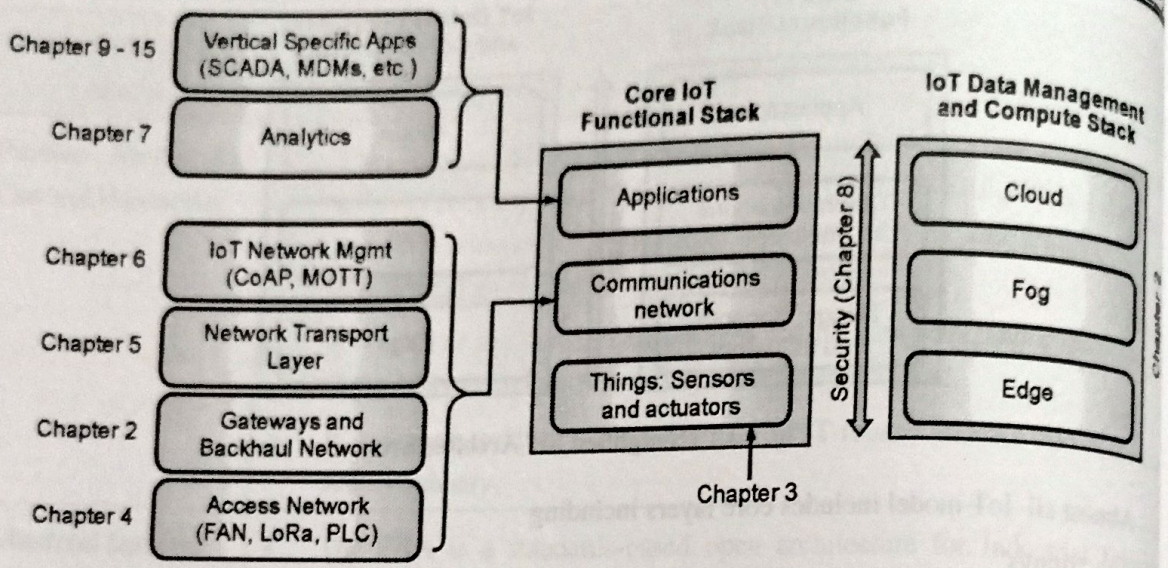


Fig. 2.4.2 : Detailed View of the simplified IoT Architecture

- The communications layer is broken down into four separate sub-layers, those are
  - the access network,
  - gateways and backhaul,
  - IP transport, and
  - operations and management sub-layers.
- The applications layer of IoT networks is different from the application layer of a typical enterprise network.
- Instead of simply using business applications, IoT often involves a strong big data analytics component.
- IoT is not just about the control of IoT devices but, rather, the useful insights gained from the data generated by those devices.
- Thus, the applications layer typically has both analytics and industry-specific IoT control system components.

2.4.1 The Core IoT Functional Stack

IoT networks are built around the concept of "things," or smart objects performing functions and delivering new connected services. These objects are "smart" because they use a combination of contextual information and configured goals to perform actions.

Looking at the architecture there are several components working together to make an IoT network operational

A. "Things" layer : At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.

B. **Communications network layer:** When smart objects are not complete or independent, they need to communicate with an external system. Communication technology used is wireless is most of cases. This layer has four sublayers:

1. Access network sublayer
2. Gateways and backhaul network sublayer
3. Network transport sublayer
4. IoT network management sublayer

1. **Access network sublayer:** The very important part of IoT network is the access network. Uses wireless technologies such as 802.11ah, 802.15.4g, and LoRa. The sensors connected to the access network can be wired.
2. **Gateways and backhaul network sublayer:** For the multiple smart object in a specific area a common gate way and communication network is defined. The gateways function is to forward the collected information through a longer-range medium (called the backhaul) to a central station where the information is processed. This information exchange is a Layer 7 (application) function, due to this object is also called as a gateway. On IP networks, this gateway also forwards packets from one IP network to another, and it therefore acts as a router.
3. **Network transport sublayer:** For communication to be successful, network and transport layer protocols such as IP and UDP must be implemented to support the variety of devices to connect and media to use.
4. **IoT network management sublayer:** Additional protocols must be in place to allow the headend applications to exchange data with the sensors. Examples include CoAP(Constrained Application Protocol) and MQTT(Message Queuing Telemetry Transport).

C. **Application and analytics layer :** The upper most layer is an application process the collected data, It also control the smart objects when necessary, but to make intelligent decision based on the information collected and, in turn, instruct the "things" or other systems to adapt to the analyzed conditions and change their behaviors or parameters.

2.5 BUILD YOUR OWN IOT COMMUNICATION NETWORK

In these section all elements are discussed with which one can architect its own IoT communication network.

2.5.1 Layer 1: Things: Sensors and Actuators Layer

To architect these layer one should have knowledge and classification of sensor and actuators ie things like

- Is the thing Battery-powered or power-connected
- Is the thing Mobile or static

- Is the thing working at Low or high reporting frequency
- Data generated is Simple or rich data
- What is the Report range
- How objects per cell i.e. Object density per cell

a) Battery-powered or power-connected

- This classification is based on whether the object carries its own energy supply or receives continuous power from an external power source.
- Battery-powered things can be moved more easily than line-powered objects.
- However, batteries limit the lifetime and amount of energy that the object is allowed to consume, thus driving transmission range and frequency.

b) Mobile or static

- This classification is based on whether the "thing" should move or always stay at the same location. A sensor may be mobile because it is moved from one object to another (for example, a viscosity sensor moved from batch to batch in a chemical plant) or because it is attached to a moving object (for example, a location sensor on moving goods in a warehouse or factory floor).
- The frequency of the movement may also vary, from occasional to permanent. The range of mobility (from a few inches to miles away) often drives the possible power source.

c) Low or high reporting frequency

- This classification is based on how often the object should report monitored parameters.
- A rust sensor may report values once a month. A motion sensor may report acceleration several hundred times per second.
- Higher frequencies drive higher energy consumption, which may create constraints on the possible power source (and therefore the object mobility) and the transmission range.

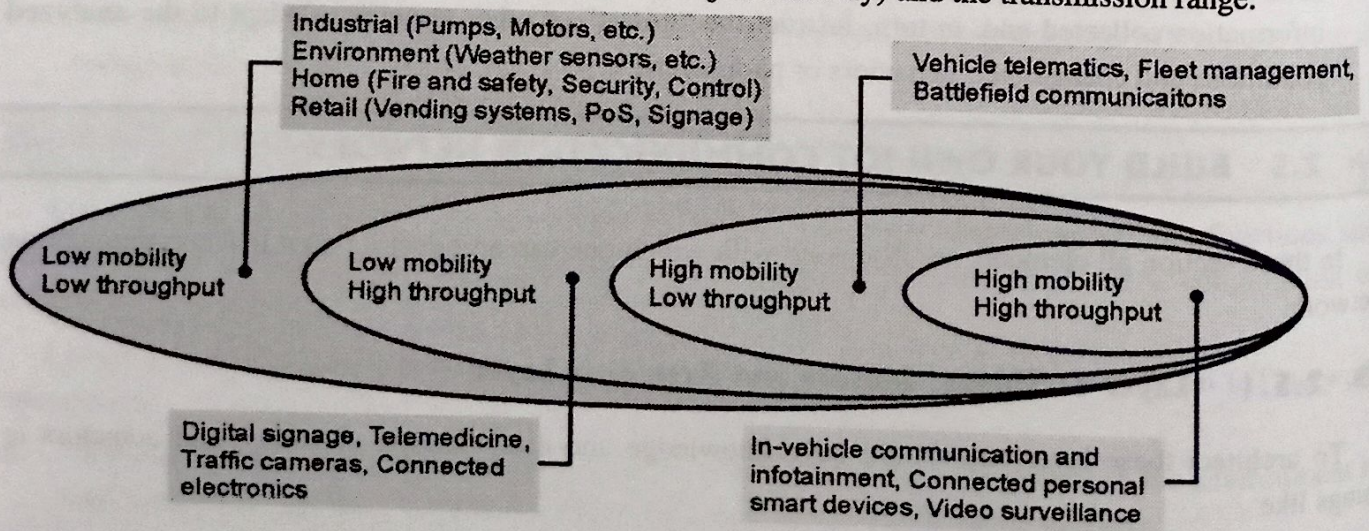


Fig. 2.5.1 : Sensor Application based on Mobility and Throughput

d) Simple or rich data

- This classification is based on the quantity of data exchanged at each report cycle. A humidity sensor in a field may report a simple daily index value (on a binary scale from 0 to 255), while an engine sensor may report hundreds of parameters, from temperature to pressure, gas velocity, compression speed, carbon index, and many others.
- Richer data typically drives higher power consumption. This classification is often combined with the previous to determine the object data throughput (low throughput to high throughput). You may want to keep in mind that throughput is a combined metric.
- A medium-throughput object may send simple data at rather high frequency (in which case the flow structure looks continuous), or may send rich data at rather low frequency (in which case the flow structure looks bursty).

e) Report range

- This classification is based on the distance at which the gateway is located. For example, for your fitness band to communicate with your phone, it needs to be located a few meters away at most.
- The assumption is that your phone needs to be at visual distance for you to consult the reported data on the phone screen.
- If the phone is far away, you typically do not use it, and reporting data from the band to the phone is not necessary.
- By contrast, a moisture sensor in the asphalt of a road may need to communicate with its reader several hundred meters or even kilometers away

f) Object density per cell

- This classification is based on the number of smart objects (with a similar need to communicate) over a given area, connected to the same gateway.
- An oil pipeline may utilize a single sensor at key locations every few miles. By contrast, telescopes like the SETI (Search for Extraterrestrial Intelligence.) Colossus telescope at the Whipple Observatory deploy hundreds, and sometimes thousands, of mirrors over a small area, each with multiple gyroscopes, gravity, and vibration sensors.
- From a network architectural standpoint, your initial task is to determine which technology should be used to allow smart objects to communicate.

2.5.2 Layer 2 : Communications Network Layer

- After knowing the smart object form factor(Size)m its transmission capabilities (transmission range, data volume and frequency, sensor density and mobility), now smart object is ready to connect and communicate.
- Computer and network assets used in IoT are very different from those in IT environments because of physical form factors between devices used by IT and OT.

- The operational differences must be understood in order to apply the correct handling to secure the target assets. Temperature variances are an easily understood metric.
- The cause for the variance is easily attributed to external weather forces and internal operating conditions.
- Remote external locations, such as those associated with mineral extraction or pipeline equipment can span from the heat to the cold .
- Humidity fluctuations can impact the long-term success of a system as well.
- Shock and vibration needs vary based on the deployment scenario.
- Solid particulates can also impact the gear. Most IT environments must contend with dust build-up that can become highly concentrated due to the effect of cooling fans.
- Hazardous location design may also cause corrosive impact to the equipment. Caustic materials can impact connections over which power or communications travel.
- Furthermore, they can result in reduced thermal efficiency by potentially coating the heat transfer surfaces.
- In some scenarios, the concern is not how the environment can impact the equipment but how the equipment can impact the environment.
- Power supplies in OT systems are also frequently different from those commonly seen on standard IT equipment. A wider range of power variations are common attributes of industrial compute components

2.5.2(a) Access Network Sublayer

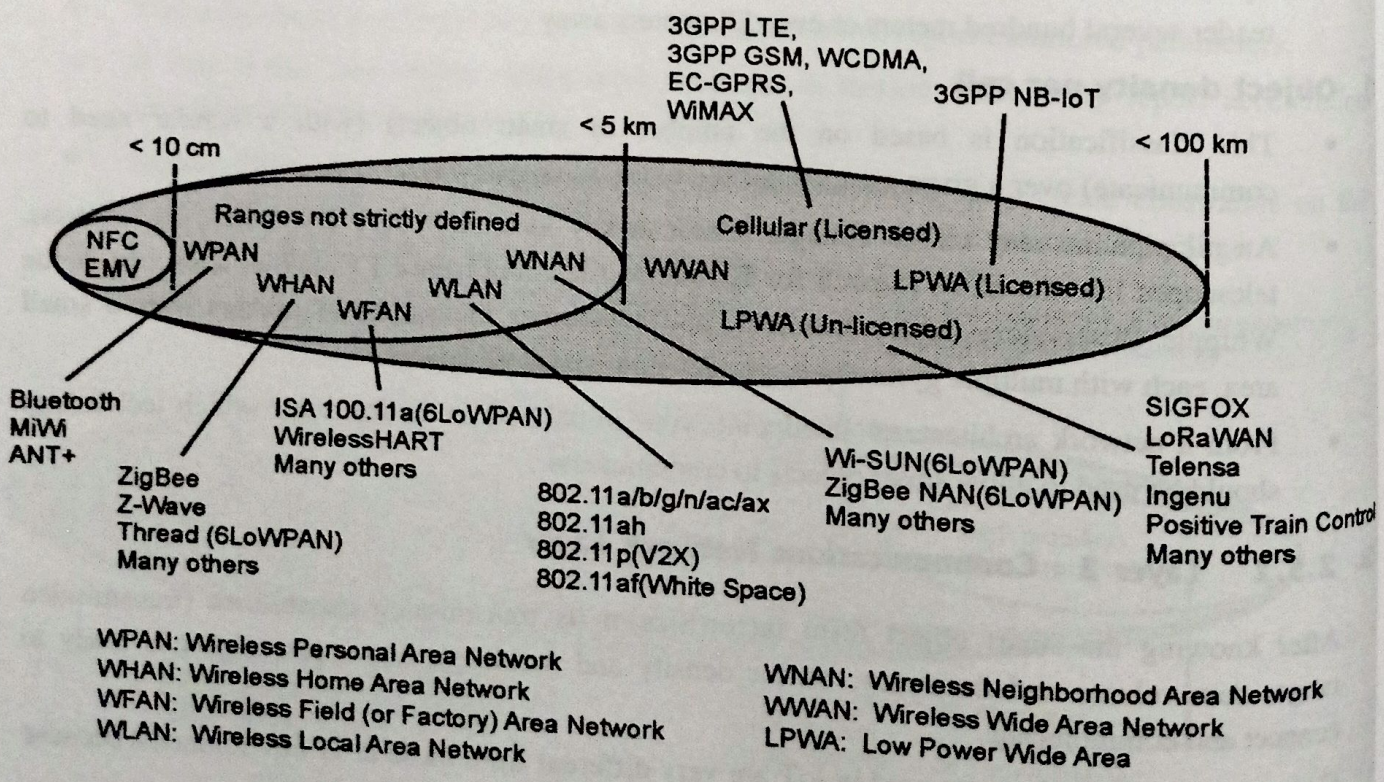


Fig. 2.5.2 : Access Technologies and Distance

- One key parameter determining the choice of access technology is the range between the smart object and the information collector
- Range estimates are grouped by category names that illustrate the environment or the vertical where data collection over that range is expected.
- Common groups are as follows:
  - **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 often seen 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 group of 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.

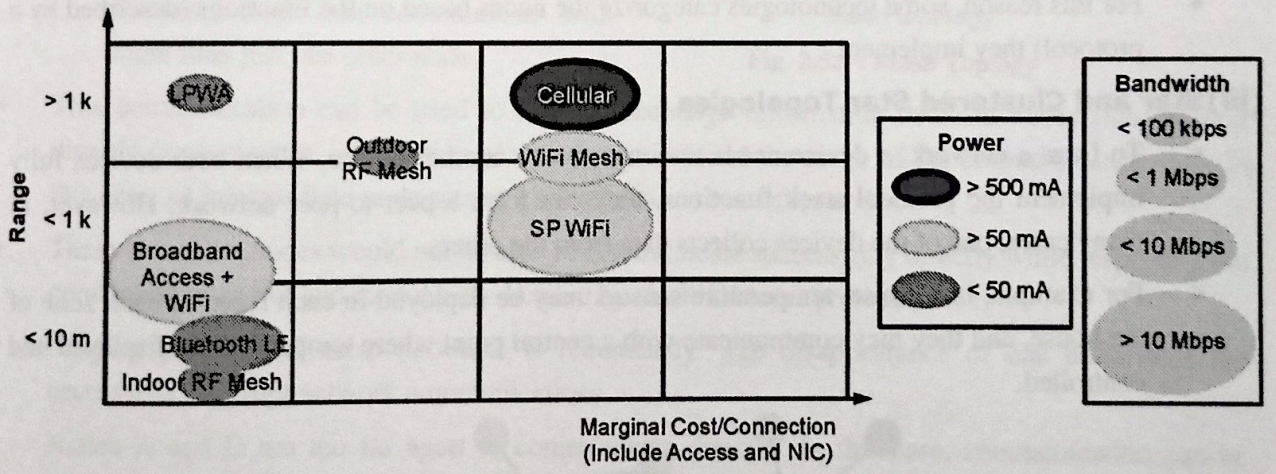


Fig. 2.5.3 : Combines cost, range, power consumption, and typical available bandwidth for common IoT access technologies

- The amount of data to carry over a given time period along with correlated power consumption
- (driving possible limitations in mobility and range) determines the wireless cell size and structure.
- Similar ranges also do not mean similar topologies.
- Some technologies offer flexible connectivity structure to extend communication possibilities:
  - Point-to-point topologies
  - Point-to-multipoint topologies

(i) Point-to-point topologies

- These topologies allow one point to communicate with another point. This topology in its strictest sense is uncommon for IoT access, as it would imply that a single object can communicate only with a single gateway.
- However, several technologies are referred to as "point-to-point" when each object establishes an individual session with the gateway.
- The "point-to-point" concept, in that case, often refers to the communication structure more than the physical topology.

(ii) Point-to-multipoint topologies

- These topologies allow one point to communicate with more than one other point. Most IoT technologies where one or more than one gateways communicate with multiple smart objects are in this category.
- However, depending on the features available on each communicating mode, several subtypes need to be considered.
- A particularity of IoT networks is that some nodes (for example, sensors) support both data collection and forwarding functions, while some other nodes (for example, some gateways) collect the smart object data, sometimes instruct the sensor to perform specific operations, and also interface with other networks or possibly other gateways.
- For this reason, some technologies categorize the nodes based on the functions (described by a protocol) they implement.

(iii) Star and Clustered Star Topologies

- To form a network, a device needs to connect with another device. When both devices fully implement the protocol stack functions, they can form a peer-to-peer network. However, in many cases, one of the devices collects data from the others.
- For example, in a house, temperature sensors may be deployed in each room or each zone of the house, and they may communicate with a central point where temperature is displayed and controlled.

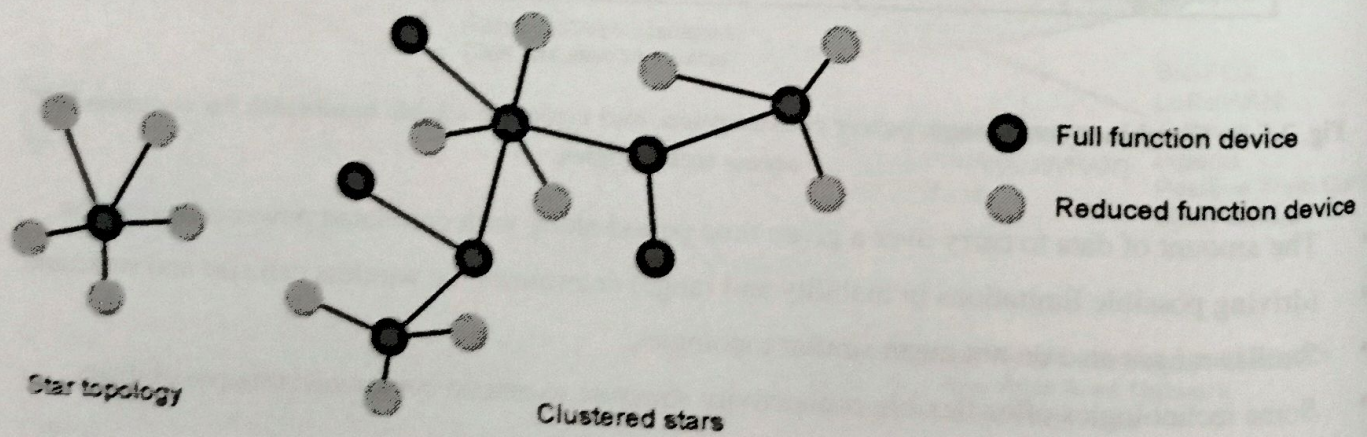
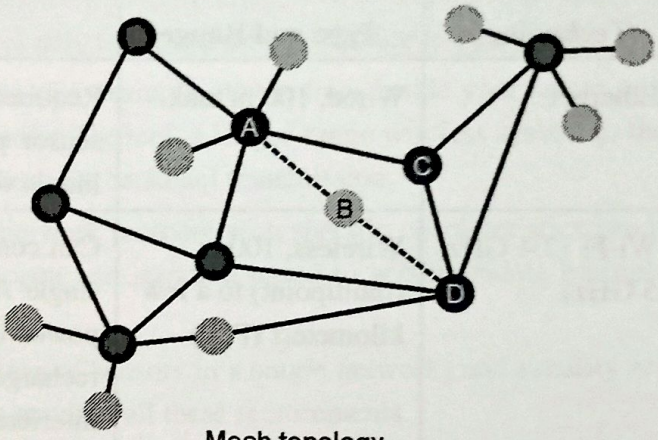


Fig. 2.5.4 : Star and Clustered Star Topologies

- The sensor can implement a subset of protocol functions to perform just a specialized part (communication with the coordinator). Such a device is called a reduced-function device (RFD).
- As shown in Fig. 2.5.4., An RFD cannot be a coordinator. An RFD also cannot implement direct communications to another RFD.
- The coordinator that implements the full network functions is called, by contrast, a full-function device (FFD). An FFD can communicate directly with another FFD or with more than one FFD, forming multiple peer-to-peer connections.
- Topologies where each FFD has a unique path to another FFD are called cluster tree topologies. FFDs in the cluster tree may have RFDs, resulting in a cluster star topology.

(iv) Mesh Topology

- Point-to-multipoint shown in Fig. 2.5.5. Technologies allow a node to have more than one path to another node, forming a mesh topology. This redundancy means that each node can communicate with more than just one other node.



Mesh topology

Fig. 2.5.5 : Mesh Toplogy

- This communication can be used to directly exchange information between nodes (the receiver directly consumes the information received) or to extend the range of the communication link. In this case, an intermediate node acts as a relay between two other nodes.
- These two other nodes would not be able to communicate successfully directly while respecting the constraints of power and modulation dictated by the PHY layer protocol.
- Another property of mesh networks is redundancy. The disappearance of one node does not necessarily interrupt network communications.
- Nodes A and D are too far apart to communicate directly. In this case, communication can be relayed through nodes B or C. Node B may be used as the primary relay.
- However, the loss of node B does not prevent the communication between nodes A and D. Here, communication is rerouted through another node, node C.

2.5.2(b) Gateways and Backhaul Sublayer

- Data collected from a smart object may need to be forwarded to a central station where data is processed. As this station is often in a different location from the smart object, data directly received from the sensor through an access technology needs to be forwarded to another medium (the backhaul) and transported to the central station. The gateway is in charge of this inter-medium communication.

- In the DSRC case, the entire "sensor field" is moving along with the gateway, but the general principles of IoT networking remain the same.
- The range at which DSRC can communicate is limited. Similarly, for all other IoT architectures, the choice of a backhaul technology depends on the communication distance and also on the amount of data that needs to be forwarded.
- When the smart object's operation is controlled from a local site, and when the environment is stable (for example, factory or oil and gas field), Ethernet can be used as a backhaul. Mesh is a common topology to allow communication flexibility in this type of dynamic environment.

Table 2.5.1 : Compares the main solutions from an architectural angle

Technology	Type and Range	Architectural Characteristics
Ethernet	Wired, 100 m max	Requires a cable. per sensor/sensor group; adapted to static sensor position in a stable environment; range is limited; link is very reliable
Wi-Fi (2.4 GHz, 5 GHz)	Wireless, 100 m (multipoint) to a few kilometers (P2P)	Can connect multiple clients (typically fewer than 200) to a single AP: range is Limited: adapted to cases where client power is not an issue (continuous power or client battery recharged easily); large bandwidth available, but interference from other systems likely; AP needs a cable.
802.11ah (Halo W, Wi-Fi in sub-1 GHz	Wireless, 1.5 km (multipoint), 10 km (P2P)	Can connect a Large number of clients (up to 6000 per AP); longer range than traditional Wi-Fi; power efficient; limited bandwidth; low adoption; and cost may be an issue
WiMAX (802.16)	Wireless,, several kilometers (last mile). up to 50 km (backhaul)	Can connect a Large number of clients:. Large bandwidth available in licensed spectrum (fee-based); reduced bandwidth in license-free spectrum (interferences from other systems likely); adoption varies on location
Cellular (for example, LTE)	Wireless, several kilometers	Can connect a Large number of clients; large bandwidth available' licensed spectrum (interference-free; license-based)

2.5.2(c) Network Transport Sub-layer

- Distribution automation (DA) allows your meter to communicate with neighboring meters or other devices in the electrical distribution grid. With such communication, consumption load balancing may be optimized.
- For example, your air conditioning pulses fresh air at regular intervals. With DA, your neighbor's AC starts pulsing when your system pauses; in this way, the air in both houses is kept fresh, but the energy consumed from the network is stable instead of spiking up and down with uncoordinated start and stop points.

- Similarly, smart meter may communicate with your house appliances to evaluate their type and energy demand. With this scheme, washing machine can be turned on in times of lower consumption from other systems, such as at night, while powering home theater system will never be deprived, always turning on when you need it.
- Once the system learns consumption pattern, charging of electric car can start and stop at intervals to achieve the same overnight charge without creating spikes in energy demand. Data may flow locally, or it may have to be orchestrated by a central application to coordinate the power budget between houses.
- This communication structure thus may involve peer-to-peer, point-to-point, point-to-multipoint, unicast and multicast communications. In a multitenant environment (for example, electricity and gas consumption management), different systems may use the same communication pathways.
- This communication occurs over multiple media (for example, power lines inside your house or a shortrange wireless system like indoor Wi-Fi and/or ZigBee), a longer-range wireless system to the gateway, and yet another wireless or wired medium for backhaul transmission.
- To allow for such communication structure, a network protocol with specific characteristics needs to be implemented. The protocol needs to be open and standard-based to accommodate multiple industries and multiple media.
- Scalability (to accommodate thousands or millions of sensors in a single network) and security are also common requirements. IP is a protocol that matches all these requirements.
- The flexibility of IP allows this protocol to be embedded in objects of very different natures, exchanging information over very different media, including low-power, lossy, and low-bandwidth networks.
- For example, RFC 2464 describes how an IPv6 packet gets encapsulated over an Ethernet frame and is also used for IEEE 802.11 Wi-Fi.
- Similarly, the IETF 6LoWPAN working group specifies how IPv6 packets are carried efficiently over lossy networks, forming an "adaption layer" for IPv6, primarily for IoT networks.
- Finally, the transport layer protocols built above IP (UDP and TCP) can easily be leveraged to decide whether the network should control the data packet delivery (with TCP) or whether the control task should be left to the application (UDP). UDP is a much lighter and faster protocol than TCP.
- However, it does not guarantee packet delivery. Both TCP and UDP can be secured with TLS/SSL (TCP) or DTLS (UDP).

2.5.2(d) IoT Network Management Sub-layer

- IP, TCP, and UDP bring connectivity to IoT networks. Upper-layer protocols need to take care of data transmission between the smart objects and other systems. Multiple protocols have been leveraged or created to solve IoT data communication problems.

- Some networks rely on a push model (that is, a sensor reports at a regular interval or based on a local trigger), whereas others rely on a pull model (that is, an application queries the sensor over the network), and multiple hybrid approaches are also possible.
- Following the IP logic, some IoT implementers have suggested HTTP for the data transfer phase. After all, HTTP has a client and server component. The sensor could use the client part to establish a connection to the IoT central application (the server), and then data can be exchanged.
- To find HTTP in some IoT applications, but HTTP is something of a fat protocol and was not designed to operate in constrained environments with low memory, low power, low bandwidth, and a high rate of packet failure. Despite these limitations, other web-derived protocols have been suggested for the IoT space.
- One example is WebSocket. WebSocket is part of the HTML5 specification, and provides a simple bidirectional connection over a single connection. Some IoT solutions use WebSocket to manage the connection between the smart object and an external application. WebSocket is often combined with other protocols, such as MQTT (described shortly) to handle the IoT-specific part of the communication.
- To respond to the limits of web-based protocols, another protocol was created by the IETF Constrained Restful Environments (CoRE) working group: Constrained Application Protocol (CoAP).
- CoAP uses some methods similar to those of HTTP (such as Get, Post, Put, and Delete) but implements a shorter list, thus limiting the size of the header. CoAP also runs on UDP (whereas HTTP typically uses TCP). CoAP also adds a feature that is lacking in HTTP and very useful for IoT: observation.
- Observation allows the streaming of state changes as they occur, without requiring the receiver to query for these changes.

2.5.3 Layer 3 : Applications and Analytics Layer

- Once connected to a network, your smart objects exchange information with other systems.
- Once IoT network spans more than a few sensors, the power of the Internet of Things appears in the applications that make use of the information exchanged with the smart objects.

2.6 ANALYTICS VERSUS CONTROL APPLICATIONS

2.6.1 Analytics Application

- Collects data from multiple smart objects
- Processes the collected data
- Displays information resulting from the data that was processed
- Application processes the data to convey a view of the network that cannot be obtained from solely looking at the information displayed by a single smart object.

2.6.2 Control Application

- Controls the behavior of the smart object or the behavior of an object related to the smart object.
- Used for controlling complex aspects of an IoT network with a logic that cannot be programmed inside a single IoT object
- An example of control system architecture is SCADA. SCADA was developed as a universal method to access remote systems and send instructions. One example where SCADA is widely used is in the control and monitoring of remote terminal units (RTUs) on the electrical distribution grid.
- Advanced IoT applications include both analytics and control modules. Data is collected from the smart objects and processed in the analytics module in many cases.
- The result of this processing may be used to modify the behavior of smart objects or systems related to the smart objects.
- The control module is used to convey the instructions for behavioral changes. When evaluating an IoT data and analytics application, you need to determine the relative depth of the control part needed for your use case and match it against the type of analytics provided.

2.7 DATA VERSUS NETWORK ANALYTICS

2.7.1 Data Analytics

- Processes the data collected by smart objects and combines it to provide an intelligent view related to the IoT system.
- A simple dashboard can display an alarm when a weight sensor detects that a shelf is empty in a store. A complex case, temperature, pressure, wind, humidity, and light levels collected from thousands of sensors may be combined and then processed to determine the chances of a storm and its possible path
- Data processing can be very complex and may combine multiple changing values over complex algorithms.
- Data analytics can also monitor the IoT system itself. For example, a machine or robot in a factory can report data about its own movements. This data can be used by an analytics application to report degradation in the movement speeds, which may be indicative of a need to service the robot before a part breaks.

2.7.2 Network Analytics

- Since the system is built with smart objects connected to the network. A loss or degradation in connectivity is likely to affect the efficiency of the system.
- For example, open mines use wireless networks to automatically pilot dump trucks. A loss of connectivity may result in an accident or degradation of operations efficiency (automated dump trucks typically stop upon connectivity loss).

- Also loss of connectivity means that data stops being fed to your data analytics platform, and the system stops making intelligent analyses of the IoT system.
- Most analytics applications employ both
  - Data Analytics and
  - Network Analytics modules.

(i) Network Analytics Modules

- Network analytics is necessary for connected systems. However, the depth of analysis depends on your use cases.
- A basic connectivity view may be enough if the smart objects report occasional status, without expectation for immediate action based on this report.
- Detailed analysis and trending about network performance are needed if the central application is expected to pilot in near-real-time connected systems.

(ii) Data Analytics Modules

- Data analytics is a wider space with a larger gray area (in terms of needs) than network analytics. Basic systems analytics can provide views of the system state and state trend analysis.
- More advanced systems can refine the type of data collected and display additional information about the system. The type of collected data and processing varies widely with the use case.

2.8 DATA ANALYTICS VERSUS BUSINESS BENEFITS

- A smarter architectural choice may be to allow for an open system where the network is engineered to be flexible enough that other sensors may be added in the future, and where both upstream and downstream operations are allowed.
- This flexibility allows for additional processing of the existing sensors and also deeper and more efficient interaction with the connected objects. This enhanced data processing can result in new added value for businesses that are not envisioned at the time when the system is initially deployed.
- An example of a flexible analytics and control application is Cisco Jasper, which provides a turnkey cloud-based platform for IoT management and monetization. Consider the case of vending machines deployed throughout a city. At a basic level, these machines can be connected, and sensors can be deployed to report when a machine is in an error state.
- A repair person can be sent to address the issue when such a state is identified. This type of alert is a time saver and avoids the need for the repair team to tour all the machines in turn when only one may be malfunctioning.
- This alert system may also avoid delay between the time when a machine goes into the error state and the time when a repair team visits the machine location. With a static platform, this use case is limited to this type of alert. With a flexible platform like Cisco Jasper, new applications may be imagined and developed over time.

- For example, the machine sensors can be improved to also report when an item is sold. The central application can then be enhanced to process this information and analyze what item is most sold, in what location, at what times.
- This new view of the machines may allow for an optimization of the items to sell in machines in a given area. Systems may be implemented to adapt the goods to time, season, or location—or many other parameters that may have been analyzed. In short, architecting open systems opens the possibility for new applications.

2.8.1 Data Analytics and Business Analytics Comparison Table

Following is the list of points that show the comparisons between Data Analytics and Business Analytics :

Basis For Comparison	Business Analytics	Data Analytics
Focus	A business analyst would be responsible for making the reports, KPI (Key Performance Index) matrix, trends in the data which would help the organization	A data analyst would just play with the data to find patterns, correlations and even build models to see how the data responds to his/her models.
Process	A business analyst would do a static and comparative study of the data.	A data analyst would do an explanatory analysis and then will try to experiment with data mining processes so as to give a good visual representation of the data.
Data Sources	A business analysts would pre-plan his/her sources of data as to what all are necessary and which should be excluded which is a slow process.	A data analyst finds a correlation on some data which is not a part of his earlier dataset then he/she would add the data source on the fly as needed.
Transform	A business analyst would transform the data upfront which is carefully planned.	All the transformations are done in-database and whenever there is a demand to enrich data it is done on the fly.
Data Quality	A business analyst would always present the data as a single version of truth	A business analyst would go by the phrase "Good enough" or theoretically with the probabilities
Data Model	A business analyst would go with schema on load data model	A data analyst would go with schema on query data model.
Analysis	Retrospective, descriptive	Predictive, prescriptive
Field	A subset of computer science and management where the study of data is done by using different methods and technologies	Covers entire technological field which is a superset of Data Science

2.8.2 Key Differences Between Data Analytics and Business Analytics

Below are the lists of points, describe the key Differences Between Data Analytics and Business Analytics :

- The key tasks of a business analyst will be checking the requirement assessing it with a point of operations and functions whereas a data analyst will only analyze the data in terms of collecting manipulating and analyzing the data.
- The business analyst goes through all the requirements by scoping and de-scoping the requirements and then assign the tasks to the developers to develop the code whereas a data analyst would be preparing dashboards charts or various visualizations which would help the higher management to take calls on what should be done next.
- The business analyst would research and try to gain valuable insights from the data, finding the optimal model for the business also lies with the business analyst whereas a data analyst would concentrate on developing new algorithms or to optimize the already developed algorithms.
- An example and try to differentiate between the two:
  1. We have a study where a telecom company needs to segregate their customers in order find the unwanted customers or let's just say the churn rate. A business analyst would ask the developers to build models by giving them all the data they require and then try to evaluate which model describes the best.
  2. Whereas a data analyst would be taking care of cleaning the data, transforming the data so that it could fit good enough for the model, tweaking the model for better results, building visual outputs so as to make the model easily understandable.

2.9 SMART SERVICES

- Due to the availability of large amounts of data and the possibility to use it for specific purposes, such as the analysis or improvement of certain processes, digital (smart) services can be offered via the Internet. Such digital services are known from online shops, for example, when personalized recommendations or individualized advertising banners are displayed based on the available data.
- The German National Academy of Science and Engineering acatech defines Smart Services as 'packages of products, services and features individually configured via the Internet, tailored to the preferences of private and commercial users in a needs-based and situation-specific 'as a service' manner.
- Digital platforms play a central role: This is where products and services are mapped virtually, combined, enhanced with additional digital services and offered as Smart Services'.
- Transferred to industrial applications, these services extend or improve existing functions for product services. They can thus be used as a basis for decisions on optimising, controlling or adapting individual processes (at customers).

2.9.1 Smart Services Use IoT and Aim for Efficiency

- For example, sensors can be installed on equipment to ensure ongoing conformance with regulations or safety requirements. This angle of efficiency can take multiple forms, from presence sensors in hazardous areas to weight threshold violation detectors on trucks.
- Smart services can also be used to measure the efficiency of machines by detecting machine output, speed, or other forms of usage evaluation. Entire operations can be optimized with IoT.

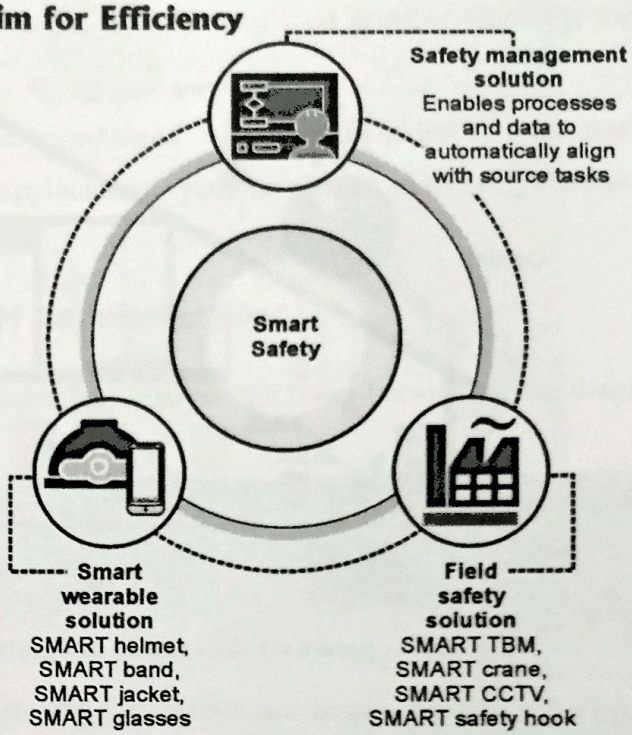


Fig. 2.9.1 : Smart Services in Safety and aim for efficiency

2.9.2 Smart Services in Hospitality

- For example, presence and motion sensors can evaluate the number of guests in a lobby and redirect personnel accordingly. The same type of action can be taken in a store where a customer is detected as staying longer than the typical amount of time in front of a shelf.
- Personnel can be deployed to provide assistance. Movement of people and objects on factory floors can be analyzed to optimize the production flow.

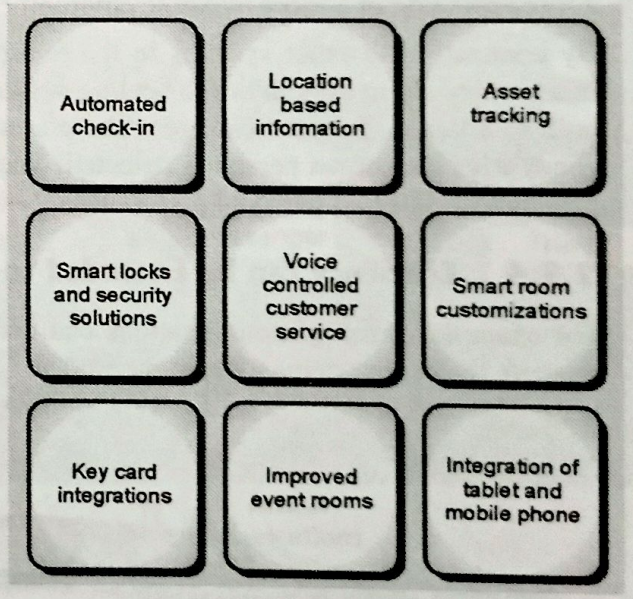


Fig. 2.9.2 : Smart Services in hospitality

2.9.3 Smart Services can be Integrated into an IoT System like Smart Home

- For example, sensors can be integrated in a light bulb. A sensor can turn a light on or off based on the presence of a human in the room.
- An even smarter system can communicate with other systems in the house, learn the human movement pattern, and anticipate the presence of a human, turning on the light just before the person enters the room.

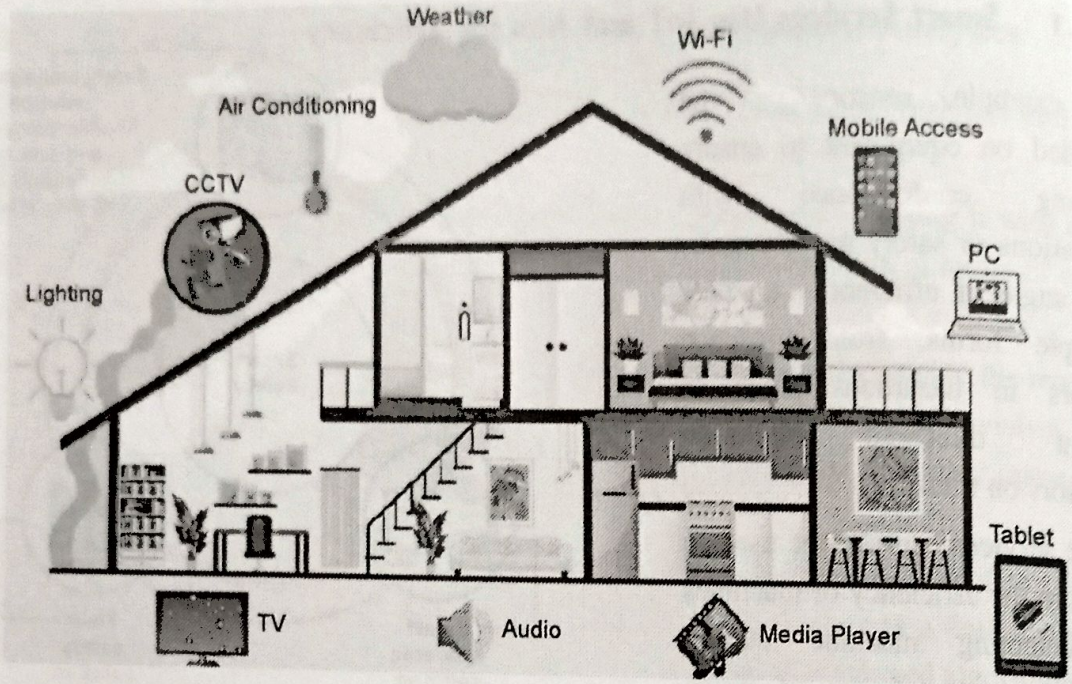


Fig. 2.9.3 : Smart services can be integrated into an IoT system like Smart Home

- An even smarter system can use smarter sensors that analyze multiple parameters to detect human mood and modify accordingly the light color to adapt to the learned preferences, or to convey either a more relaxing or a more dynamic environment.
- By connecting to other systems in the house, efficiencies can be coordinated. For example, the house entry alarm system or the heating system can coordinate with the presence detector in a light bulb to adapt to detected changes. The alarm system can disable volumetric movement alarms in zones where a known person is detected. The heating system can adapt the temperature to human presence or detected personal preferences.

2.9.4 Efficiency can be Extended to Larger Systems Like Smart Grid

- For example, smart grid applications can coordinate the energy consumption between houses to regulate the energy demand from the grid.

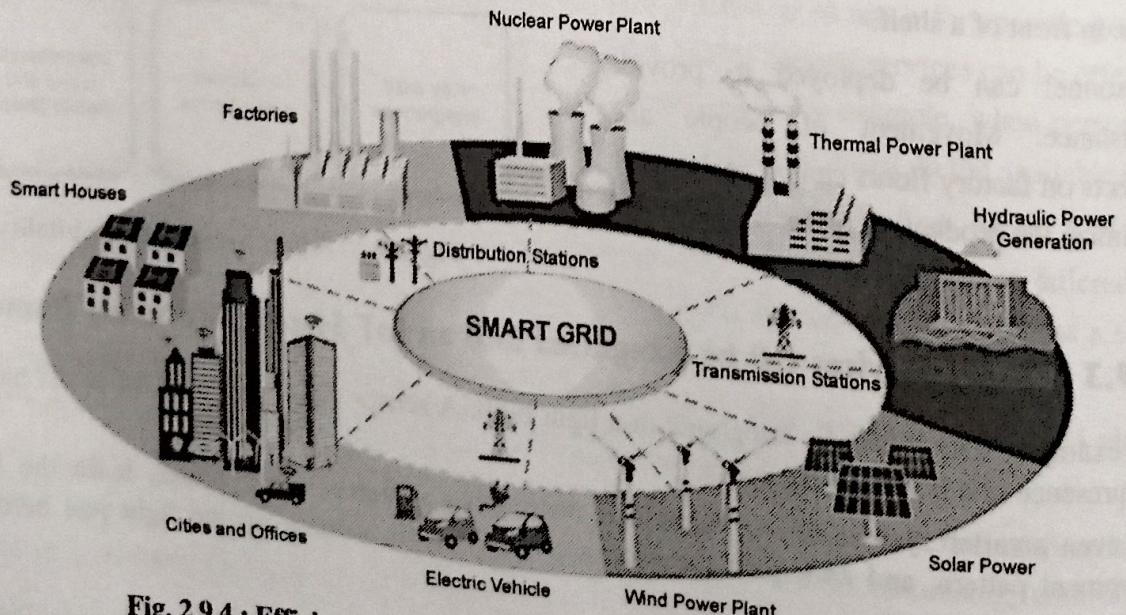


Fig. 2.9.4 : Efficiency can be extended to larger systems Like Smart Grid

- It is already mentioned that your washing machine may be turned on at night when the energy demand for heating and cooling is lower.
- Just as your air conditioning pulses can be coordinated with your neighbor's, your washing machine cycles can be coordinated with the appliances in your house and in the neighborhood to smooth the energy demand spikes on the grid.

2.9.5 Efficiency also applies to M2M Communications

- In mining environments, vehicles can communicate to regulate the flows between drills, draglines, bulldozers, and dump trucks.

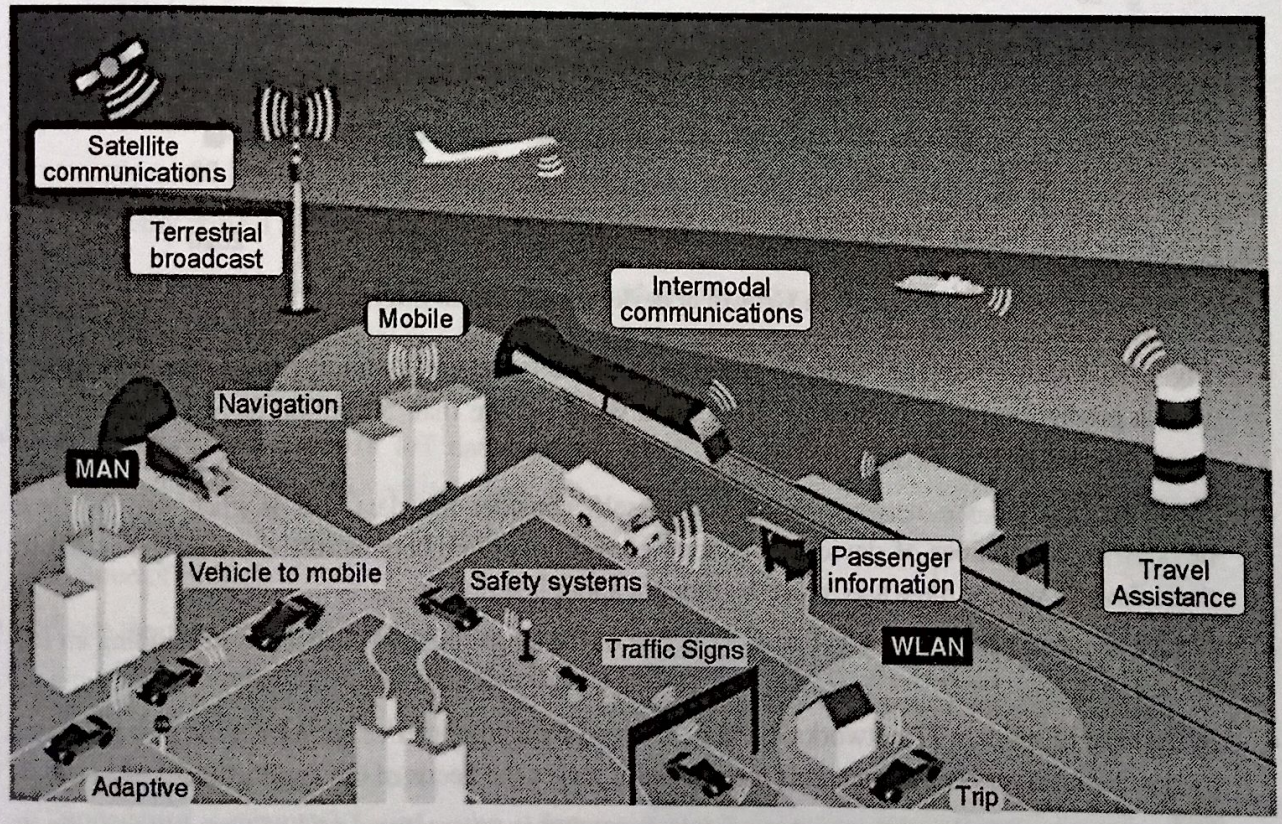


Fig. 2.9.5 : Efficiency also applies to M2M communications

- For example, making sure that a dump truck is always available when a bulldozer needs it. In smart cities, vehicles communicate.
- A traffic jam is detected and anticipated automatically by public transportation, and the system can temporarily reroute buses or regulate the number of buses servicing a specific line based on traffic and customer quantity, instantaneous or learned over trending.

2.10 IOT DATA MANAGEMENT AND COMPUTE STACK

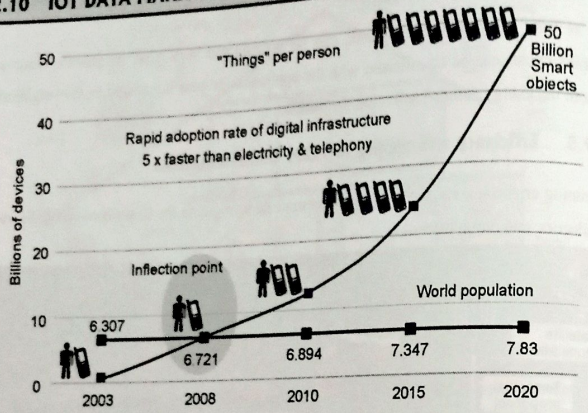


Fig. 2.10.1 : Growth of devices connected to Internet

- Fig. 2.10.1 shows how the "things" connected to the Internet are continuing to grow exponentially, with a prediction by Cisco that by 2020 there will be more than 50 billion devices connected to some form of an IP network.
- However, beyond the network architecture itself, consider the data that is generated by these devices. If the number of devices is beyond conventional numbers, surely the data generated by these devices will be huge and must be thought about it seriously.
- In fact, the data generated by IoT sensors is one of the single biggest challenges in building an IoT system.
- In the case of modern IT networks, the data sourced by a computer or server is typically generated by the client/server communications model, and it serves the needs of the application.
- In sensor networks, the vast majority of data generated is unstructured and of very little use on its own.
- For example, the majority of data generated by a smart meter is nothing more than polling data; the communications system simply determines whether a network connection to the meter is still active.
- This data on its own is of very little value. The real value of a smart meter is the metering data read by the meter management system (MMS). However, if you look at the raw polling data from a different perspective, the information can be very useful.
- For example, a utility may have millions of meters covering its entire service area. If whole sections of the smart grid start to show an interruption of connectivity to the meters, this data can be analyzed and combined with other sources of data, such as weather reports and electrical demand in the grid, to provide a complete picture of what is happening.

- This information can help determine whether the loss of connection to the meters is truly a loss of power or whether some other problem has developed in the grid. Moreover, analytics of this data can help the utility quickly determine the extent of the service outage and repair the disruption in a timely fashion.
- In most cases, the processing location is outside the smart object. A natural location for this processing activity is the cloud. Smart objects need to connect to the cloud, and data processing is centralized.
- However, this model also has limitations. As data volume, the variety of objects connecting to the network, and the need for more efficiency increase.
- These new requirements include the following:
  - **Minimizing latency :** Analyzing data close to the device that collected the data can make a difference between averting disaster and a cascading system failure.
  - **Conserving network bandwidth :** It is not practical to transport vast amounts of data from thousands or hundreds of thousands of edge devices to the cloud. Nor is it necessary because many critical analyses do not require cloud-scale processing and storage.
  - **Increasing local efficiency:** Collecting and securing data across a wide geographic area with different environmental conditions may not be useful.
- The volume of data also introduces questions about bandwidth management.
- As the massive amount of IoT data begins to funnel into the data center, does the network have the capacity to sustain this volume of traffic?
- Does the application server have the ability to ingest, store, and analyze the vast quantity of data that is coming in?
- This is sometimes referred to as the "impedance mismatch" of the data generated by the IoT system and the management application's ability to deal with that data.

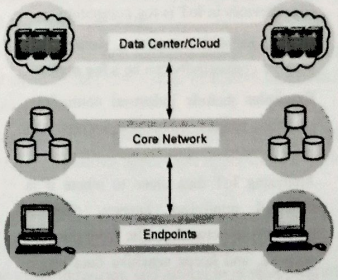


Fig. 2.10.2 : Traditional IT cloud Computing Model

- As shown in Fig. 2.10.2 data management in traditional IT systems is very simple.
- The endpoints (laptops, printers, IP phones, and so on) communicate over an IP core network to servers in the data center or cloud.
- Data is generally stored in the data center, and the physical links from access to core are typically high bandwidth, meaning access to IT data is quick.

2.10.1 Several Data-Related Problems Need to be Addressed

- **Bandwidth** in last-mile IoT networks is very limited. When dealing with thousands/millions of devices, available bandwidth may be on order of tens of Kbps per device or even less.
- **Latency** can be very high. Instead of dealing with latency in the milliseconds range, large IoT networks often introduce latency of hundreds to thousands of milliseconds.
- **Network** backhaul from the gateway can be unreliable and often depends on 3G/LTE or even satellite links. Backhaul links can also be expensive if a per-byte data usage model is necessary.
- The volume of data transmitted over the backhaul can be high, and much of the data may not really be that interesting (such as simple polling messages).
- **Big data** is getting bigger. The concept of storing and analyzing all sensor data in the cloud is impractical. The sheer volume of data generated makes real-time analysis and response to the data almost impossible.

2.10.2 Fog Computing

- The solution to the challenges mentioned in the previous section is to distribute data management throughout the IoT system, as close to the edge of the IP network as possible. The best-known embodiment of edge services in IoT is fog computing.
- Any device with computing, storage, and network connectivity can be a **fog node.**
- **Examples** include industrial controllers, switches, routers, embedded servers, and IoT gateways.
- Analyzing IoT data close to where it is collected minimizes latency, offloads gigabytes of network traffic from the core network, and keeps sensitive data inside the local network.

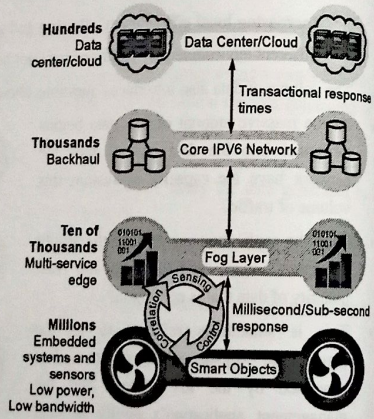


Fig. 2.10.3 : The Iot Data Management and Compute stack with Fog Computing

- Fog services are typically accomplished very close to the edge device, sitting as close to the IoT endpoints as possible.
- The fog node has contextual awareness of the sensors it is managing because of its geographic proximity to those sensors.

- The fog node is able to analyze information from all the sensors and can provide contextual analysis of the messages it is receiving and may decide to send back only the relevant information over the backhaul network to the cloud.
- IoT fog computing enables data to be preprocessed and correlated with other inputs to produce relevant information. This data can then be used as real-time, actionable knowledge by IoT-enabled applications. Longer term, this data can be used to gain a deeper understanding of network behavior and systems for the purpose of developing proactive policies, processes, and responses.
- Fog applications are as diverse as the Internet of Things itself. What they have in common is data reduction monitoring or analyzing real-time data from network-connected things and then initiating an action, such as locking a door, changing equipment settings, applying the brakes on a train, zooming a video camera, opening a valve in response to a pressure reading, creating a bar chart, or sending an alert to a technician to make a preventive repair.

2.10.2(a) Characteristic of Fog Computing

- Contextual location awareness and low latency: The fog node sits as close to the IoT endpoint as possible to deliver distributed computing.
- Geographic distribution: In sharp contrast to the more centralized cloud, 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. For example, typical metering deployments often see 3000 to 4000 nodes per gateway router, which also functions as the fog computing node.
- Wireless communication between the fog and the IoT endpoint: Although it is possible to connect wired nodes, the advantages of fog are greatest when dealing with a large number of endpoints, and wireless access is the easiest way to achieve such scale.
- Use for real-time interactions: Important fog applications involve real-time interactions rather than batch processing. Preprocessing of data in the fog nodes allows upper-layer applications to perform batch processing on a subset of the data.

2.10.3 Edge Computing

- Fog computing solutions are being adopted by many industries, and efforts to develop distributed applications and analytics tools are being introduced at an accelerating pace. The natural place for a fog node is in the network device that sits closest to the IoT endpoints, and these nodes are typically spread throughout an IoT network.
- IoT devices and sensors often have constrained resources, however, as compute capabilities increase. Some new classes of IoT endpoints have enough compute capabilities to perform at least low-level analytics and filtering to make basic decisions. For example, consider a water sensor on a fire hydrant.

- While a fog node sitting on an electrical pole in the distribution network may have an excellent view of all the fire hydrants in a local neighborhood, a node on each hydrant would have clear view of a water pressure drop on its own line and would be able to quickly generate an alert of a localized problem.
- The fog node would have a wider view and would be able to ascertain whether the problem was more than just localized but was affecting the entire area. Another example is in the use of smart meters.
- Edge compute–capable meters are able to communicate with each other to share information on small subsets of the electrical distribution grid to monitor localized power quality and consumption, and they can inform a fog node of events that may pertain to only tiny sections of the grid. Models such as these help ensure the highest quality of power delivery to customers.

2.10.3(a) The Hierarchy of Edge, Fog and Cloud

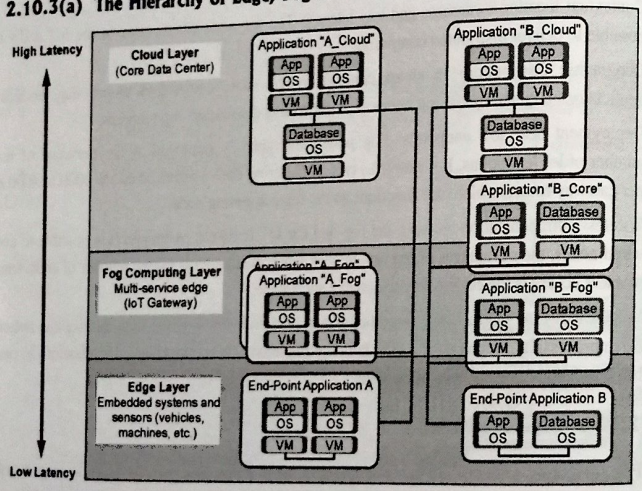


Fig. 2.10.4 : Distributed Computer and Data Management across an IoT System

- This model suggests a hierarchical organization of network, compute, and data storage resources. At each stage, data is collected, analyzed, and responded to when necessary, according to the capabilities of the resources at each layer. As data needs to be sent to the cloud, the latency becomes higher.

- Edge and fog thus require an abstraction layer that allows applications to communicate with one another.
- The abstraction layer :
  - exposes a common set of APIs for monitoring, provisioning, and controlling the physical resources in a standardized way.
  - requires a mechanism to support virtualization, with the ability to run multiple operating systems or service containers on physical devices to support multitenancy and application consistency across the IoT system.

From an architectural standpoint, fog nodes closest to the network edge receive the data from IoT devices. The fog IoT application then directs different types of data to the optimal place for analysis:

- The most time-sensitive data is analyzed on the edge or fog node closest to the things generating the data.
- Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action.
- Data that is less time sensitive is sent to the cloud for historical analysis, big data analytics, and long-term storage. For example, each of thousands or hundreds of thousands of fog nodes might send periodic summaries of data to the cloud for historical analysis and storage.

Chapter Ends...

□□□