

IOT describes the network of physical objects (things) that are embedded with sensors, actuators, software and another technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet without human intervention. It is a simple concept of controlling different devices by establishing a connection and communication with the help of mobile apps and web browser.

## UNIT-I

### Introduction of IOT

IoT comprises things that have unique identities and are connected to internet. By 2020 there will be a total of 50 billion devices /things connected to internet. IoT is not limited to just connecting things to the internet but also allow things to communicate and exchange data.

#### Definition:

A dynamic global n/w infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual —things have identities, physical attributes and virtual personalities and use intelligent interfaces, and are seamlessly integrated into information n/w, often communicate data associated with users and their environments.

#### Characteristics:

- 1) **Dynamic & Self Adapting:** IoT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating conditions, user's context or sensed environment.  
**Eg:** the surveillance system is adapting itself based on context and changing conditions.
- 2) **Self-Configuring:** allowing a large number of devices to work together to provide certain functionality.
- 3) **Inter Operable Communication Protocols:** support a number of interoperable communication protocols and can communicate with other devices and also with infrastructure.
- 4) **Unique Identity:** Each IoT device has a unique identity and a unique identifier(IPaddress).
- 5) **Integrated into Information Network:** that allow them to communicate and exchangedata with other devices and systems.

#### Applications of IoT:

- 1) Home
- 2) Cities
- 3) Environment
- 4) Energy
- 5) Retail
- 6) Logistics
- 7) Agriculture
- 8) Industry
- 9) Health & LifeStyle

##### How does IOT works?

IOT works on web enabled smart devices that use embedded systems i.e. processor and sensor and communication hardware to collect and send the data from environment.

##### Advantages:

1. Minimize human effort- As IOT devices interact and communicate with each other, they can automate the tasks helping to improve the quality of services and reducing the need for human intervention.
2. Save time
3. Enhance data collection
4. Efficient resource utilization.

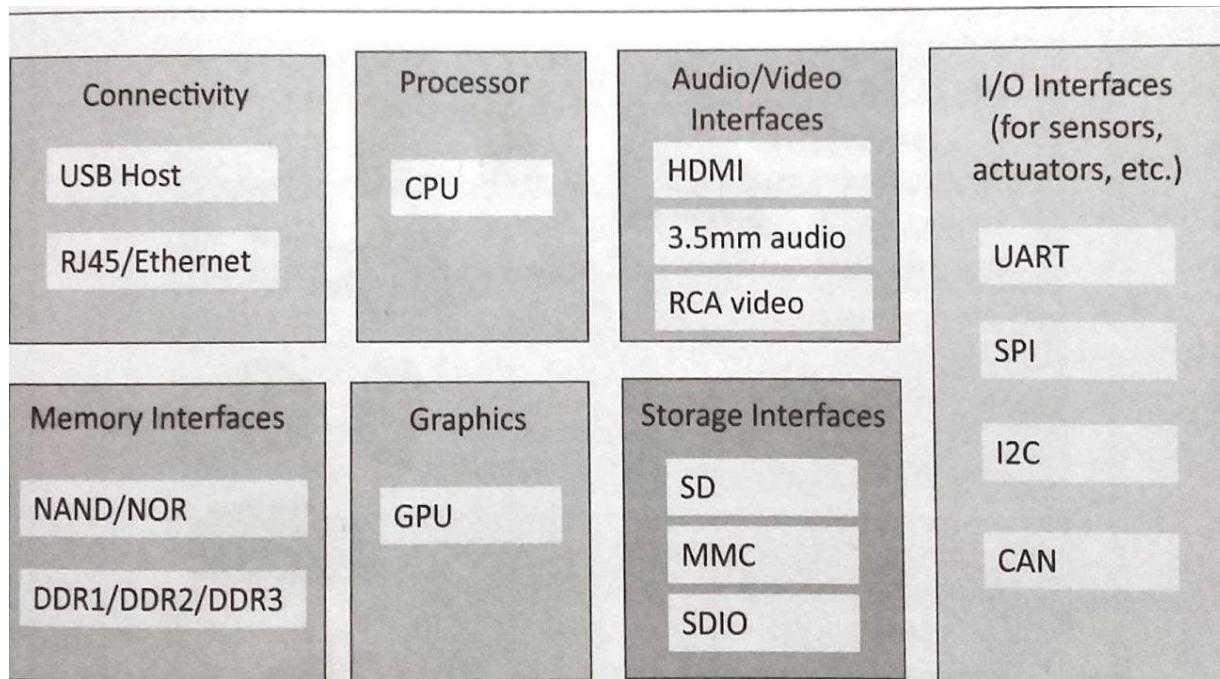
##### Issues/ Challenges:

1. Security - Cyber-attacks, Data theft
2. Privacy - Controlling access and ownership of data
3. Complexity
4. Interoperability - Integration Inflexibility
5. Legality and Rights - Data Protection laws be followed, data retention and destruction policies.
6. Economy and Development - Investment Incentives, Technical Skill Requirement

## Physical Design of IoT

is referred to the Things and Protocols used to build an IOT system.

### 1) Things in IoT: devices that are IOT enabled



The things in IoT refers to IoT devices which have unique identities and perform remote sensing, actuating and monitoring capabilities. IoT devices can exchange data with other connected devices applications. It collects data from other devices and process data either locally or remotely. An IoT device may consist of several interfaces for communication to other devices both wired and wireless.

These includes:

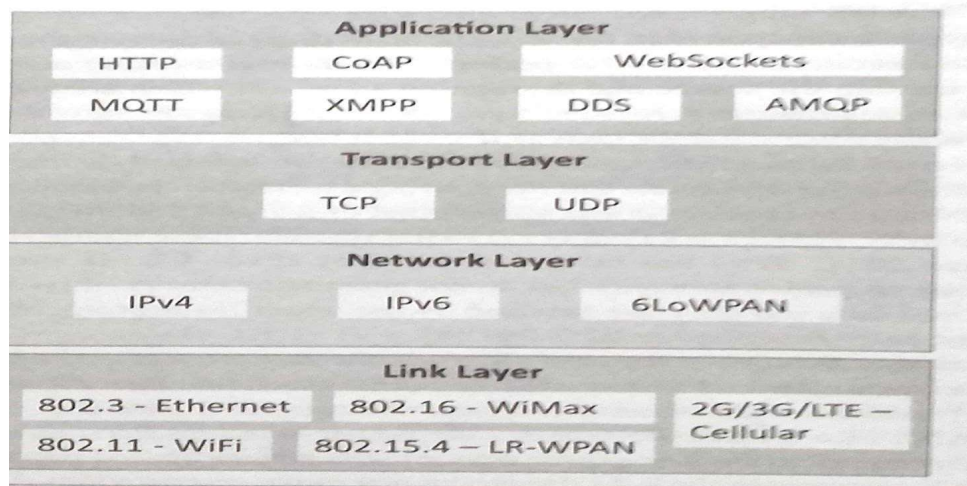
1. I/O interfaces for sensors
2. Interfaces for internet connectivity,
3. Memory and storage interfaces and,
4. Audio/video interfaces.

### 2) IoT Protocols:

**a) Link Layer:** Protocols determine how data is physically sent over the network's physical layer or medium. Local network connect to which host is attached. Hosts on the same link exchange data packets over the link layer using link layer protocols. Link layer determines how packets are coded and signaled by the h/w device over the medium to which the host is attached.

Protocols are used to establish communication between the IOT devices and servers, over the Internet. It helps to send commands to and receive data from an IOT device over the Internet.

These protocols are managed by application layer, transport layer, network layer and link layer.



### Protocols:

- **802.3-Ethernet:** IEEE802.3 is collection of wired Ethernet standards for the link layer. Eg: 802.3 uses co-axial cable; 802.3i uses copper twisted pair connection; 802.3j uses fiber optic connection; 802.3ae uses Ethernet over fiber.
- **802.11-WiFi:** IEEE802.11 is a collection of wireless LAN (WLAN) communication standards including extensive description of link layer. Eg: 802.11a operates in 5GHz band, 802.11b and 802.11g operates in 2.4GHz band, 802.11n operates in 2.4/5GHz band, 802.11ac operates in 5GHz band, 802.11ad operates in 60 GHz band.
- **802.16 - WiMax:** IEEE802.16 is a collection of wireless broadband standards including exclusive description of link layer. WiMax provide data rates from 1.5 Mb/s to 1 Gb/s.
- **802.15.4-LR-WPAN:** IEEE802.15.4 is a collection of standards for low rate wireless personal area network (LR-WPAN). Basis for high level communication protocols such as ZigBee provides data rate from 40kb/s to 250kb/s.
- **2G/3G/4G-Mobile Communication:** Data rates from 9.6kb/s (2G) to up to 100Mb/s (4G).

**b) Network/Internet Layer:** Responsible for sending IP datagrams from source n/w to destination n/w., performs the host addressing and packet routing. Datagrams contains source and destination address.

### Protocols:

- **Pv4:** Internet Protocol version 4 is used to identify the devices on a n/w using a hierarchical addressing scheme. 32 bit address. Allows total of  $2^{32}$  addresses.
- **IPv6:** Internet Protocol version 6 uses 128 bit address scheme and allows  $2^{128}$  addresses.
- **6LoWPAN:** (IPv6 over Low power Wireless Personal Area Network) operates in GHz frequency range and data transfer 250 kb/s.

- c) **Transport Layer:** Provides end-to-end message transfer capability independent of the underlying n/w. Set up on connection with ACK as in TCP and without ACK as in UDP. Provides functions such as error control, segmentation, flow control and congestion control.

**Protocols:**

- **TCP:** Transmission Control Protocol used by web browsers (along with HTTP and HTTPS), email (along with SMTP, FTP). Connection oriented and stateless protocol. IP Protocol deals with sending packets, TCP ensures reliable transmission of protocols in order to avoid n/w congestion and congestion collapse.
- **UDP:** User Datagram Protocol is connectionless protocol. Useful in time sensitive applications, very small data units to exchange. Transaction oriented and stateless protocol. Does not provide guaranteed delivery.

d) **Application Layer:** Defines how the applications interface with lower layer protocols to send data over the n/w. Enables process-to-process communication using ports.

**Protocols:**

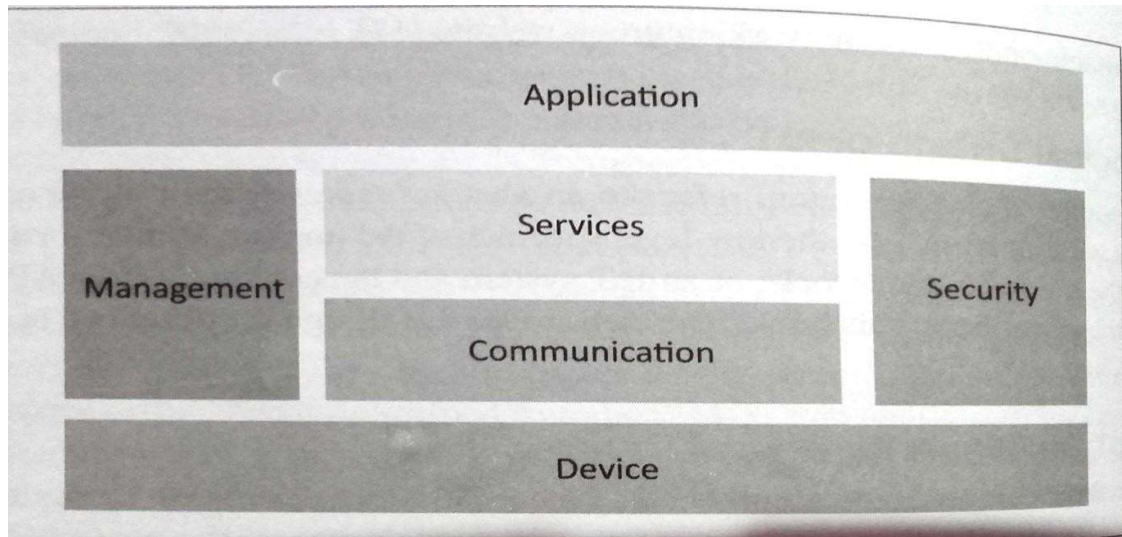
- **HTTP:** Hyper Text Transfer Protocol that forms foundation of WWW. Follow request-response model Stateless protocol.
- **CoAP:** Constrained Application Protocol for machine-to-machine (M2M) applications with constrained devices, constrained environment and constrained n/w. Uses client-server architecture.
- **WebSocket:** allows full duplex communication over a single socket connection.
- **MQTT:** Message Queue Telemetry Transport is light weight messaging protocol based on publish-subscribe model. Uses client server architecture. Well suited for constrained environment.
- **XMPP:** Extensible Message and Presence Protocol for real time communication and streaming XML data between network entities. Support client-server and server-server communication.
- **DDS:** Data Distribution Service is data centric middleware standards for device-to-device or machine-to-machine communication. Uses publish-subscribe model.
- **AMQP:** Advanced Message Queuing Protocol is open application layer protocol for business messaging. Supports both point-to-point and publish-subscribe model.

## Logical Design of IoT

Refers to an abstract represent of entities and processes without going into the low level specifics of implementation.

1) IoT Functional Blocks 2) IoT Communication Models 3) IoT Comm. APIs

1) **IoT Functional Blocks:** Provide system the capabilities for identification, sensing, actuation, communication and management.



It also allows users to view the system status and view or analyze the processed data.

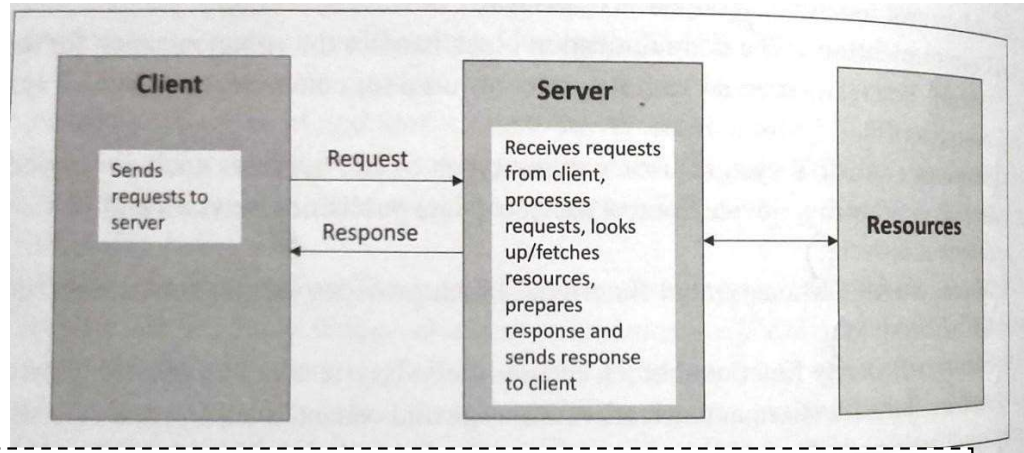
- **Device:** An IoT system comprises of devices that provide sensing, actuation, and monitoring and control functions.
- **Communication:** Handles the communication for IoT system.
- **Services:** For device monitoring, device control services, data publishing services and services for device discovery.
- **Management:** Provides various functions to govern the IoT system.
- **Security:** Secures IoT system and priority functions such as authentication, authorization, message and context integrity and data security.
- **Application:** IoT application provide an interface that the users can use to control and monitor various aspects of IoT system.



## 2) IoT Communication Models:

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

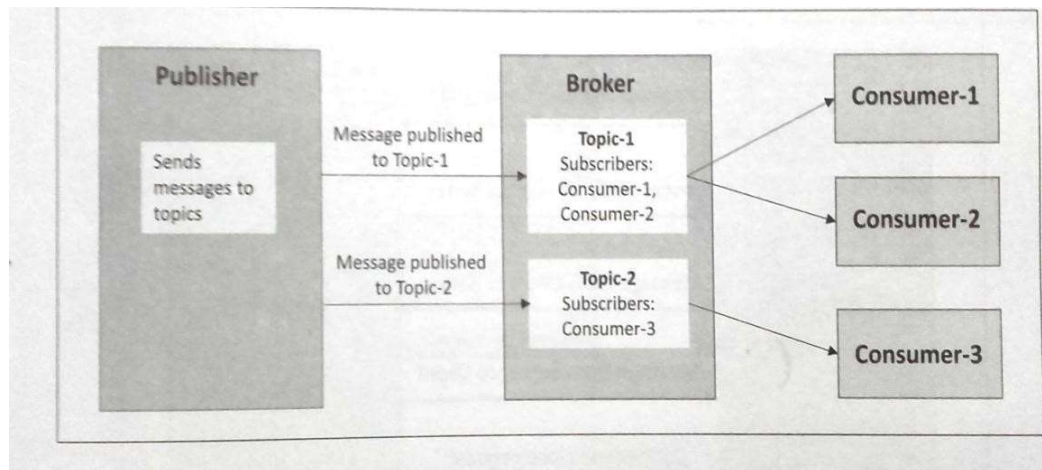
### 1) Request-Response Model:



Main diff b/w Request-Response and Publish-Subscribe model is that there is no middle man in request response model.

In which the client sends request to the server and the server replies to requests. It is a stateless communication model and each request-response pair is independent of others.

### 2) Publish-Subscribe Model:

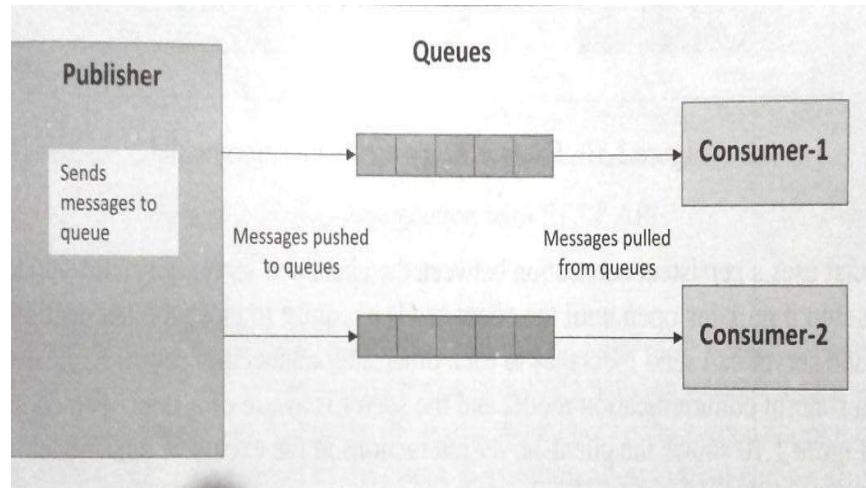


It is the responsibility of the broker to accept data from publisher and send it to respective consumers.

Involves publishers, brokers and consumers. **Publishers** are source of data. Publishers send 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.

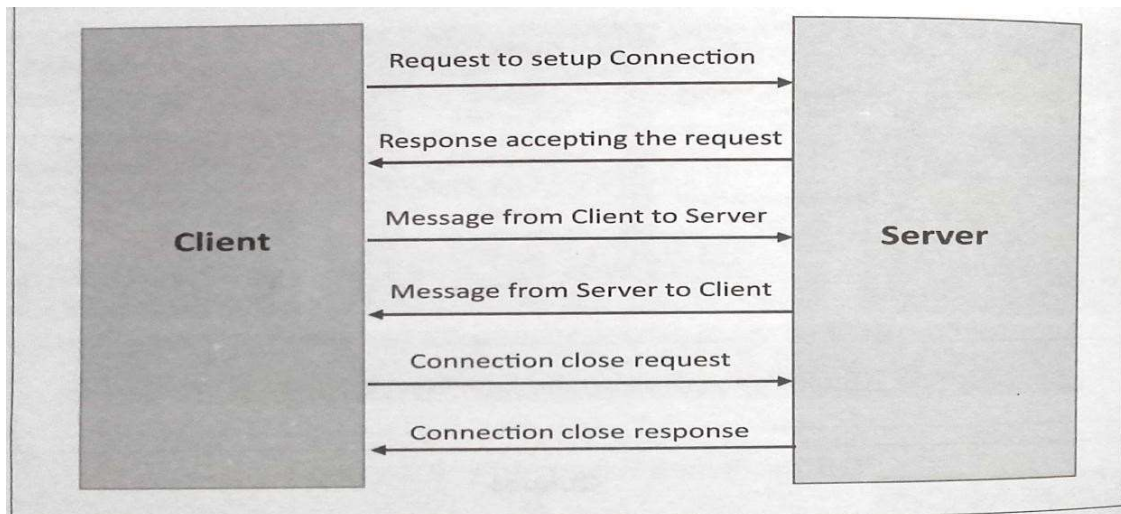
- 3) **Push-Pull Model:** In which data producers push data to queues and consumers pull data from the queues. Producers do not need to be aware of the consumers. Queues help in decoupling the message between the producers and consumers.

*This model is Time saving but inefficient because of less storage capacity.*



*Queues act as a buffer and are useful when there is a mismatch b/w the rate at which the producers push data and the data rate at which the consumers pull data.*

- 4) **Exclusive Pair:** It is bi-directional, fully duplex communication model that uses a persistent connection between the client and server. Once connection is set up it remains open until the client sends a request to close the connection. It is a state full communication model and server is aware of all the open connections.

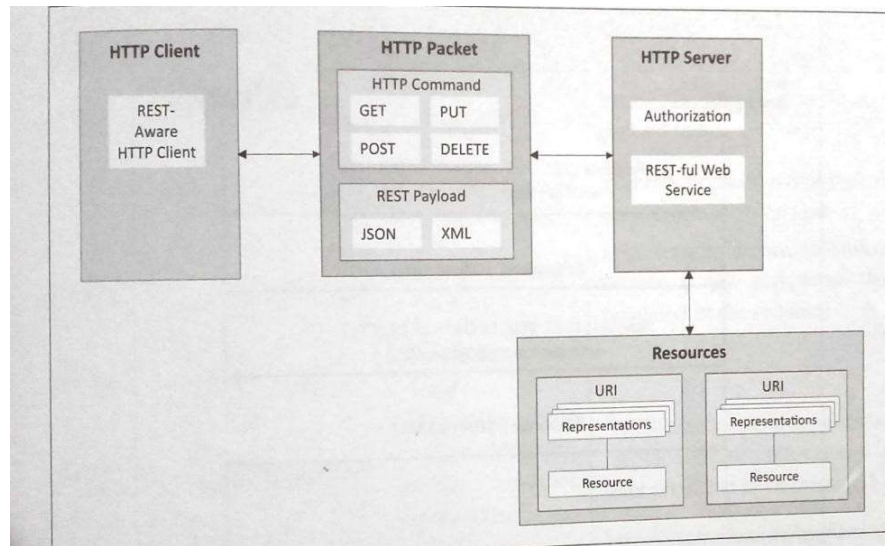


### 3)IoT CommunicationAPIs:

- a) REST based communication APIs (Request-Response Based Model)
- b) Web Socket based Communication APIs(Exclusive Pair Based Model)

a) **REST based communication APIs:** Representational State Transfer(REST) is a set of architectural principles by which one can design web services and web APIs that focus on a system's resources and explain how resource states are addressed and transferred.

**The REST architectural constraints:** Fig. shows communication between client servers with REST APIs.



**Client-Server:** The principle behind client-server constraint is the separation of concerns. Separation allows client and server to be independently developed and updated.

**Stateless:** Each request from client to server must contain all the info. Necessary to understand the request, and cannot take advantage of any stored context on the server.

**Cache-able:** Cache constraint requires that the data within a response to a request be implicitly or explicitly labeled as cache-able or non-cacheable. If a response is cache-able, then a client cache is given the right to reuse that response data for later, equivalent requests.

**Layered System:** Constrains the behavior of components such that each component cannot see beyond the immediate layer with which they are interacting.

**User Interface:** This constraint requires that the method of communication between a client and a server must be uniform.

**Code on Demand:** Servers can provide executable code or scripts for clients to execute in their context. This constraint is the only one that is optional.