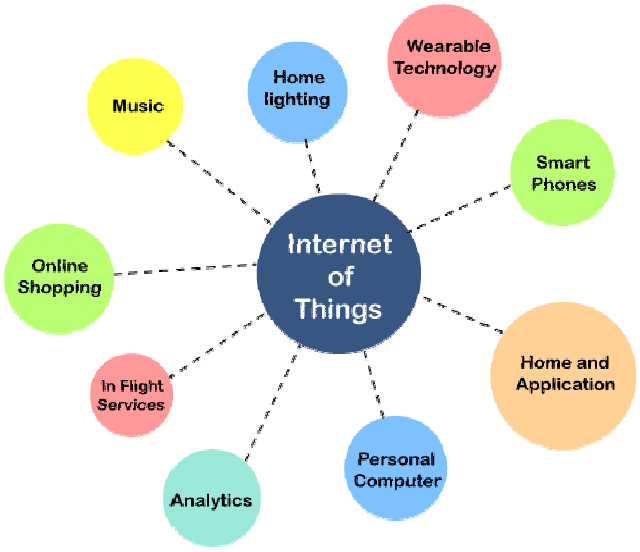
FUNDAMENTALS OF INTERNET OF THINGS

Introduction to Internet of Things

The internet of things(IoT) is a network of interrelated devices that connect and exchange data with other IoT devices and the cloud. IoT devices are embedded with technology such as sensors and software and can include mechanical and digital machines and consumer objects. With IoT, data is transferable over a network without requiring human-to-human or human-to-computer interactions.

Applications of IoT:-



- Smart Home Applications
- Health care
- Smart Cities
- Agriculture
- Healthcare

Characteristics of IoT:

Various characteristics of IoT are:

- Dynamic and self-adapting
- Self-configuring
- Interoperable Communication protocols
- Unique identity

Dynamic and self-adapting: The IoT devices can dynamically adapt with sensed environment, their operating conditions, and user's context and take actions accordingly. For ex: Surveillance System

Self-configuring: IoT devices can be able to upgrade the software with minimal interaction of user, whenever they are connected to the internet

Interoperable Communication: IoT allows different devices (different in architecture) to communicate with each other as well as with different network. For ex: MI Phone is able to control the smart AC and smart TV of different manufacturer.

Unique identities: The devices which are connected to the internet have unique identities i.e IP address through which they can be identified throughout the network

Physical Design of IoT

Physical Design of IoT refers to IoT Devices and IoT Protocols. Things(IoT Devices) are unique identities and can perform remote sensing, actuating and monitoring capabilities.

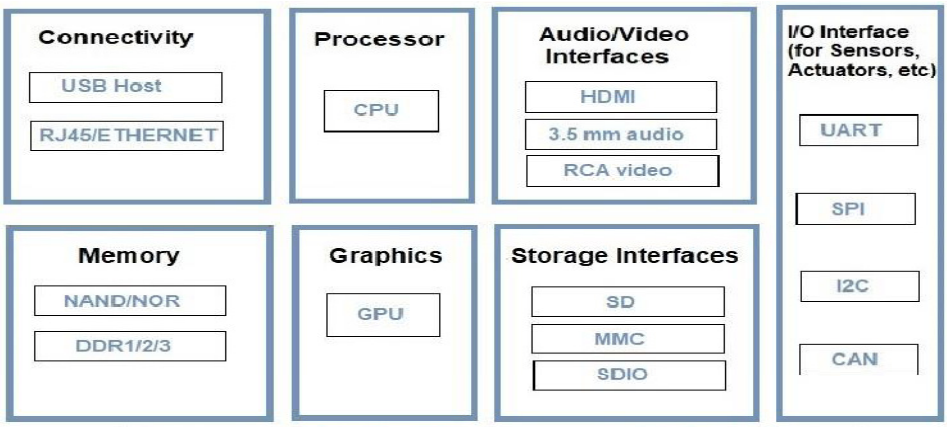
IoT devices can:

- Exchange data with other connected devices and applications (directly or indirectly).
- Collect data from other devices and process the data locally .
- Send the data to centralized servers or cloud-based application back-ends for processing the data

Block Diagram of IoT Devices

An IoT device may consist of several interfaces for connections to other devices, both wired and wireless.

- I/O interfaces for sensors
- Interfaces for Internet connectivity
- Memory and storage interface
- Audio/Video interface



Generic Block Diagram of IoT Devices

- HDMI: High Definition Multimedia Interface.
- 3.5mm: Audio jack with headphone adapter
- RAC: Radio Corporation of America
- UART: Universal Asynchronous Receiver/Transmitter
- SPI: serial peripheral interface
- I2C : Inter-Integrated Circuit
- CAN: Controller area network
- SD: Secure Digital
- MMC: Multimedia card
- SDIO: Secure Digital Input Output
- GPU: Graphics processing unit
- NAND/NOT(**NOT AND/NOT OR):** which refers to the Boolean operator
- DDR: Double Data Rate

IoT Protocols

IoT protcols help to establish Communication between IoT Device (Node Device) and Cloud based Server over the Internet. It help to sent commands to IoT Device and received data from an IoT device over the Internet.

- **Application Layer**
- **Link Layer**
- **NetworkLayer**
- **Transport Layer**

- **Application Layer:-**

Application layer protocols define how the applications interface with the lower layer protocols to send over the network. Enables process-to-process communication using ports

Protocols:

- **HTTP :** *Hypertext Transfer Protocol (HTTP)* is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers. HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response.

- **Link Layer:-**

Link layer protocols determine how data is physically sent over the network's physical layer or medium (Coxial cable or radio wave). This Layer determines how the packets are coded and signaled by the hardware device over the medium to which the host is attached (eg. coxial cable).

Protocols:

- **802.3 – Ethernet :** Ethernet is a set of technologies and protocols that are used primarily in LANs. It was first standardized in 1980s by IEEE(Institute of Electrical and Electronics Engineers) 802.3 standard.

- **Network Layer:-**

Responsible for sending of IP data grams from the source network to the destination network. Network layer performs the host addressing and packet routing. Data gram contains source and destination address.

Protocols:

- IPv4: IPv4 is the most deployed internet protocol used to identify the device on a network using addressing scheme.

- **Transport Layer:-**

This layer provides end-to-end message transfer capability independent of the network. It set up on connection with ACK as in TCP and without ACK as in UDP(User Datagram Protocol).

Protocols:

- **TCP :** TCP (Transmission Control Protocol) defines how to establish and maintain a network conversation through which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other.

Logical Design of IoT

Logical design of IoT system refers to an abstract representation of the entities & processes without going into the low-level specifies of the implementation.

1) **IoT Functional Blocks**
2) **IoT Communication Models**
3) **IoT Communication APIs**

1) IoT Functional Blocks:-

An IoT system comprises of a number of functional blocks that provide the system the capabilities for identification, sensing, actuation, communication and management.

Functional blocks are:

Device: An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.

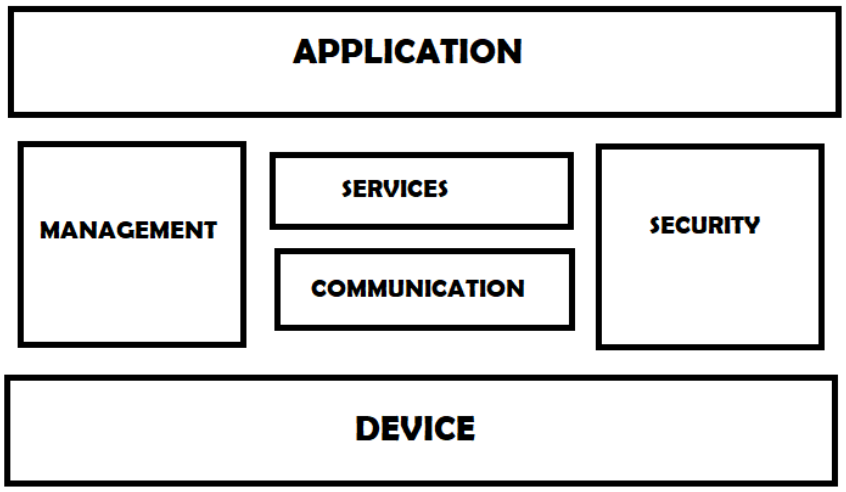
Communication: Handles the communication for the IoT system.

Services: services for device monitoring, device control service, data publishing services and services for device discovery.

Management: this blocks provides various functions to govern the IoT system.

Security: this block secures the IoT system and by providing functions such as authentication , authorization, message and content integrity, and data security.

Application: This is an interface that the users can use to control and monitor various aspects of the IoT system.

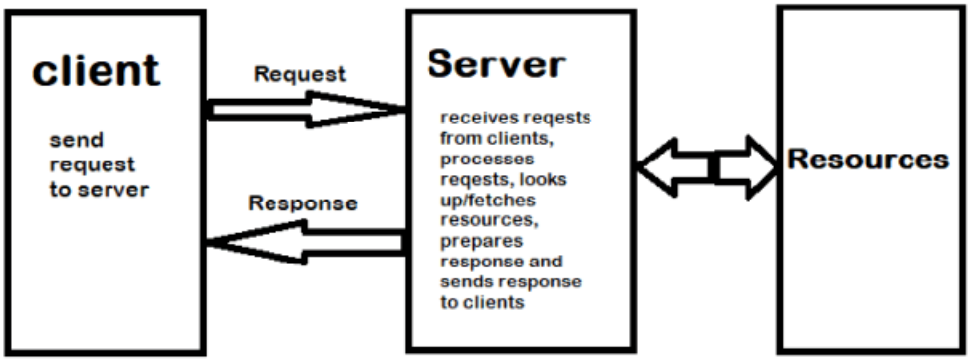


2) IoT Communication Models

a) Request-Response Model
b) Publish-Subscribe Model
c) Push-Pull Model
d) Exclusive Pair Model

a) Request-Response Model:-

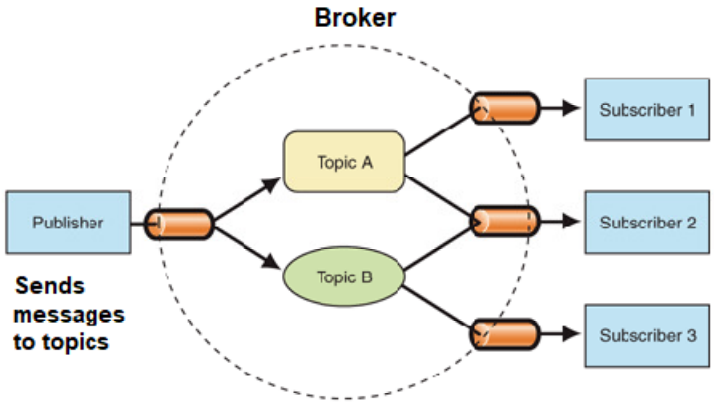
Request-response model is communication model in which the client sends requests to the server and the server responds to the requests. When the server receives a request, it decides how to respond, fetches the data, retrieves resource representation, prepares the response, and then sends the response to the client. HTTP works as a request-response protocol between a client and server.



Request-Response Communication Model

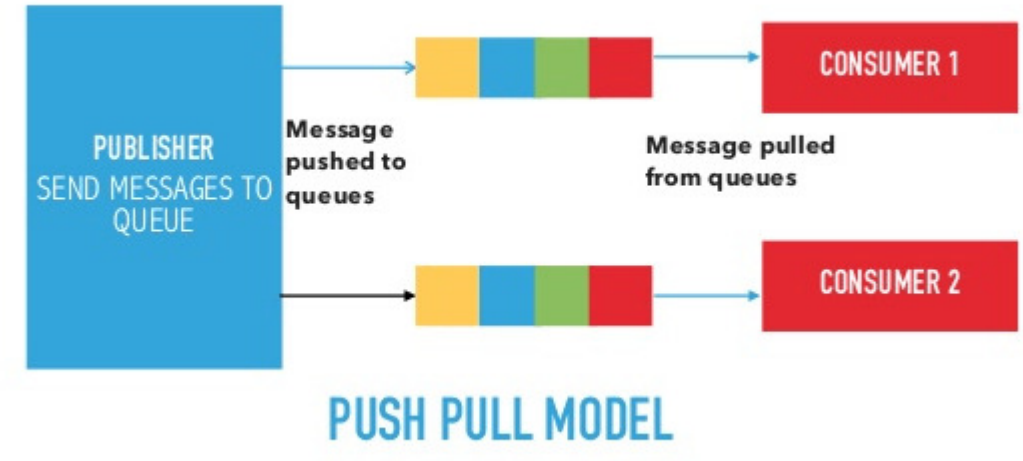
b) Publish-Subscribe Model:-

- 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 receive data for a topic from the publisher, it sends the data to all the subscribed consumers.



c) Push-Pull Model:-

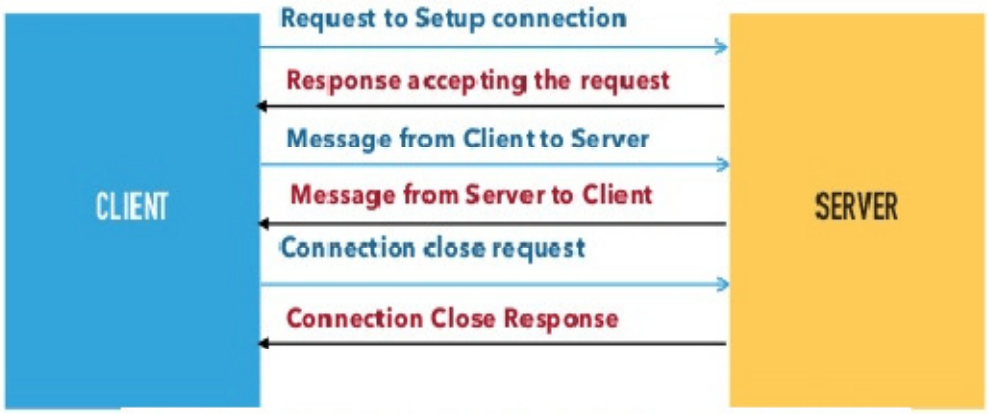
- Push-Pull is a communication model in which the data producers push the data to queues and the consumers Pull the data from the Queues. Producers do not need to be aware of the consumers.
- Queues help in decoupling the messaging between the Producers and Consumers.
- Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate rate at which the consumer pull data.



PUSH PULL MODEL

d) Exclusive Pair Model:-

- Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server.
- Connection is setup it remains open until the client sends a request to close the connection. Client and server can send messages to each other after connection setup.



EXCLUSIVE PAIR COMMUNICATION MODEL

3) IoT Communication APIs:-

a) REST-based Communication APIs (request response communication model)

b) WebSocket-based Communication APIs

a) REST-based Communication APIs:-

Representational state transfer (REST) is a set of principles by which we can design Web services the Web APIs that focus on system's resources and how resource states are addressed and transferred. REST APIs that follow the request response communication model

b) WebSocket-based Communication APIs:-

Websocket APIs allow bi-directional, full duplex communication between clients and servers. Websocket APIs follow the exclusive pair communication model. Unlike request-response model such as REST, the WebSocket APIs allow full duplex communication and do not require new connection to be setup for each message to be sent.