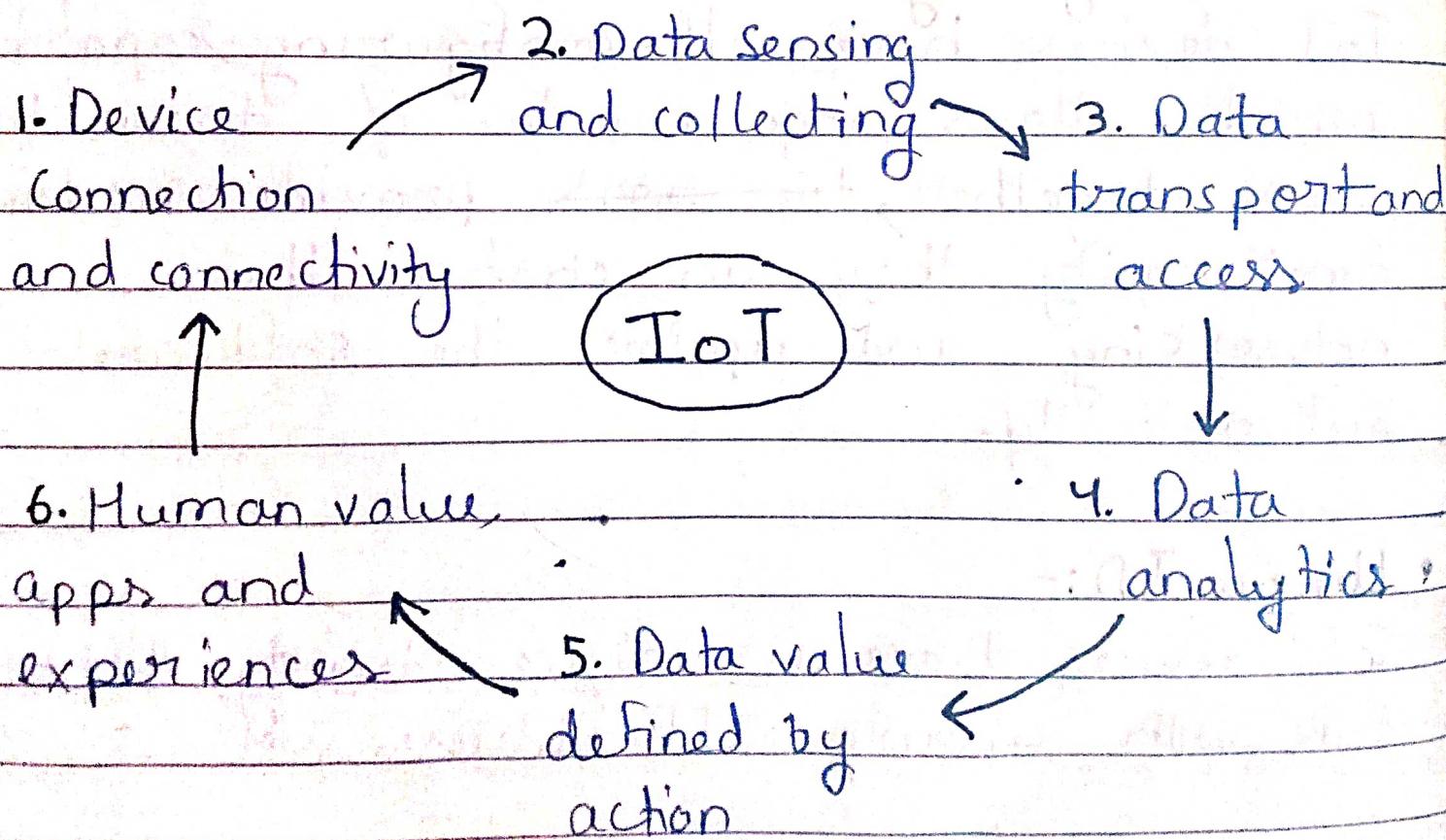


# IoT (Internet of Things)

Definition:- The IoT is the network of physical objects with unique identifiers that are connected with each other and are embedded with electronics, software and sensors which enables these objects to collect and exchange data.

How all this actually happens



# Characteristics of IoT

- **Dynamic and Self-Adapting :-**

IoT devices and system may have the capability to change dynamically depending upon the system and operating conditions or sensed environment.

Example - the surveillance cameras can change their modes based on day or night.

- **Self-configuring :-**

IoT devices have self-configuring capability which allows large number of devices to work together, to work provide certain functionality they can change their networking and update the software automatically.

- **Unique ID :-**

IoT devices have a unique identity differentiated with unique IP address.

- **Interoperable Communication Protocol :-**

IoT devices can communicate with number of interoperable (communicate with other devices without special effort) communication protocols.

- Integrated into information Network :-  
IoT devices are integrated into the information network that allows them to communicate and exchange data with other devices and systems.

- Connectivity :-

Things in IoT should be connected to the infrastructure.

- Intelligence :-

Extraction of Knowledge from the generated data is important, sensor generate data and this data should be

- Scalability :-

IoT devices should be designed in such a way that they can be scaled up or down easily on demand.

## Design of IoT

→ Physical design of IoT

→ Logical design of IoT

# 1) Physical Design of IoT

The physical design of IoT consists of the  
→ things in IoT  
→ IoT protocols & layers

## • Things in IoT

The word 'things' refers to the IoT devices which have unique identifiers, identities and can perform remote sensing, actuating and monitoring capabilities.

These devices can exchange data and communicate with each other.

The IoT devices consists of several interfaces.

Connectivity	Processor	Audio/Video	I/O interface
[USB]			
RJ45/Ethernet	CPU	Video interface	
		HDMI/3.5mm	UART
		RCA video	SPI
			I2C
Memory interfaces	Graphics	Storage interfaces	CAN
NAND/NOR			
DDR1/DDR2	GPU	SD	
DDR3		MMC	
		SDIO	

## IoT Protocols

### 1) Link layer

This protocol determines how the data is physically sent over the network layer. It determines how the packet are coded and signaled by the hardware device over the medium to which the host is attached.

Application

Transport

Network / Internet  
Link

Example:- IEEE 802.3 - Ethernet (Wired -

802.11 - WiFi

802.16 - WiMax

2G / 3G / 4G - Mobile communication

### 2) Network / Internet Layer

The network layers are responsible for sending of IP datagram's from the source network to the destination network. It performs host addressing and network routing. The datagram's consists of source and destination addresses where host identifies using IP schemes as IPv4 and IPv6.

IPV4: It is used to identify the devices on a network using hierarchical addressing scheme. It uses 32 bit addresses that allows total  $2^{32}$  or 4 billion devices.

IPV6: It is the new version of internet protocol which uses 128 bits of addresses that allows  $2^{128}$  or  $3 \times 10^{28}$  addresses.

### 3) Transport layer

The transport layer protocols provide end to end message transfer capability independent of the underlying network. The message transfer capability can be set up on connections, either using handshakes (TCP) or without handshakes / acknowledgements (UDP). The transport layer provide functions such as error control, segmentation, flow control and congestion control.

#### TCP - (Transmission Control Protocol)

- |                                      |  |
|--------------------------------------|--|
| → Connection-oriented protocol       | → reliable as it guarantees delivery of data to destination monitor. |
| → provides extensive error checking. |  |

- Sequencing of data → Retransmission of lost packets.
- slower than UDP.

header - 20-80 bytes

### UDP (User Datagram Protocol)

- Connection less → does not guarantee delivery
- basic error checking → no sequencing of data
- no retransmission → header - 8 bytes

## Application Layer

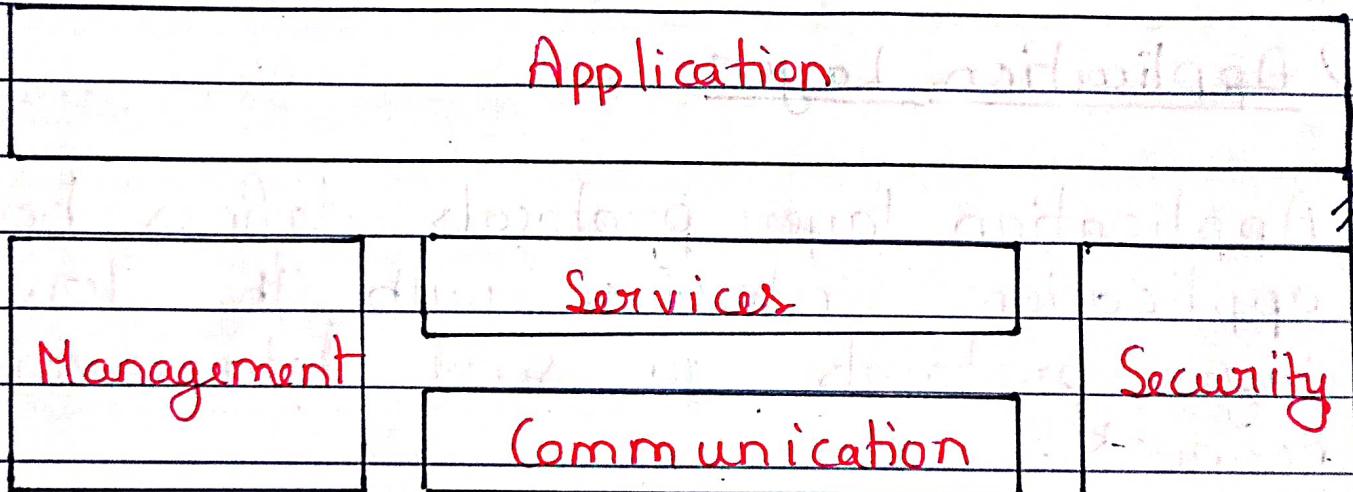
Application layer protocols defines how the applications interface with the lower layer protocols to send data over the network.

Application Type	Application Layer Protocol
Electronic mail	Send: SMTP (Simple Mail Transfer) Receive: POP3 (Post Office P)
M2M	CoAP (Constrained App. P)
WWW	HTTP (Hyper Text Transfer P)
File transfer	FTP, TFTP (Trivial FTP)
Internet telephony	Proprietary

## 2) Logical Design of IoT

Logical design of an IoT describes about abstract representation of the entities and process without going to low level specifics of the implementation.

### → IoT functional blocks



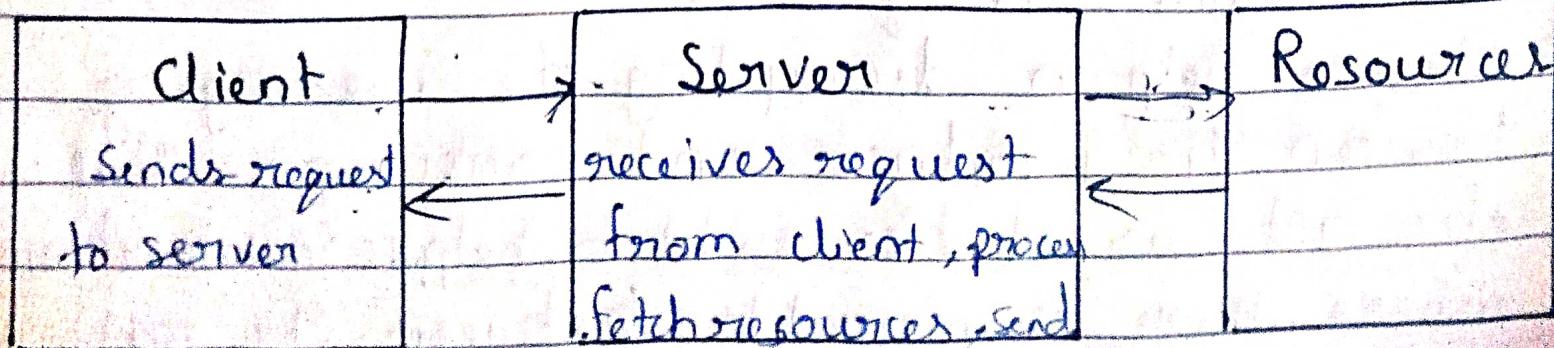
- Device - IoT devices which provide sensing, monitoring and control functions.
- Communication - Handles communication for IoT system
- Services - deals with services such as device monitoring, d. control services and d. discovery.

- Management - used to monitor the complete IoT system.
- Security - provide security by providing the functions such as authentication, authorization and data security.
- Application - IoT applications provide an interface that the users can use to control and monitor various aspects of an IoT system.

## → IoT Communication Model

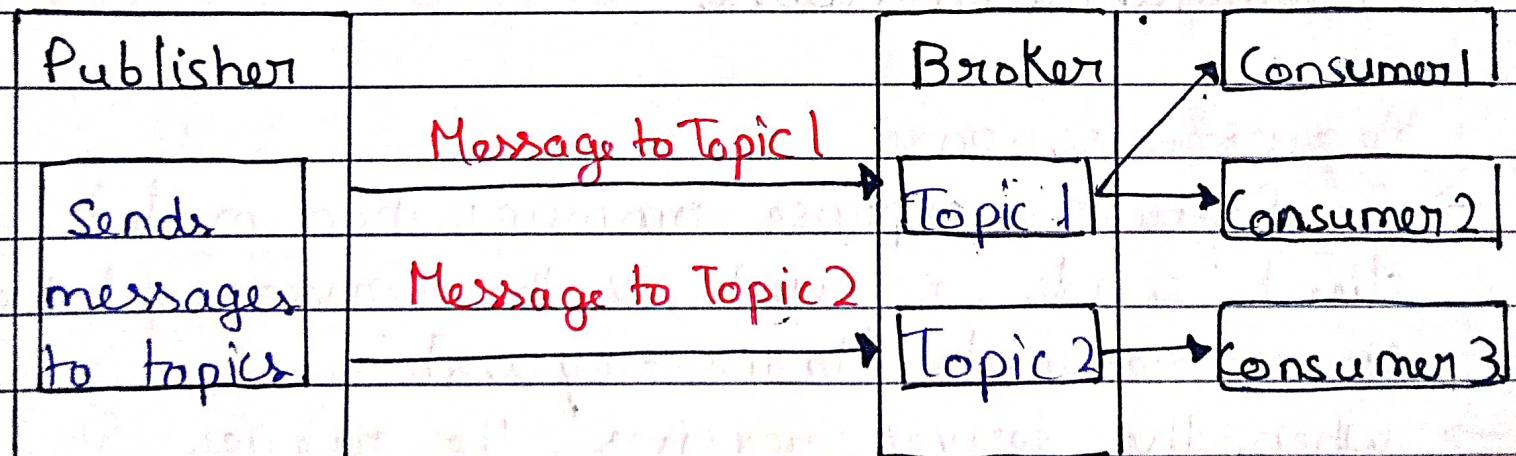
### • Request-Response

- In Request-response communication model client sends request to the server and the server responds to the request.
- When the server receives the request it decides how to respond, fetches the data, retrieves resources, and prepares the response and sends to the client.



## • Publish-Subscribe

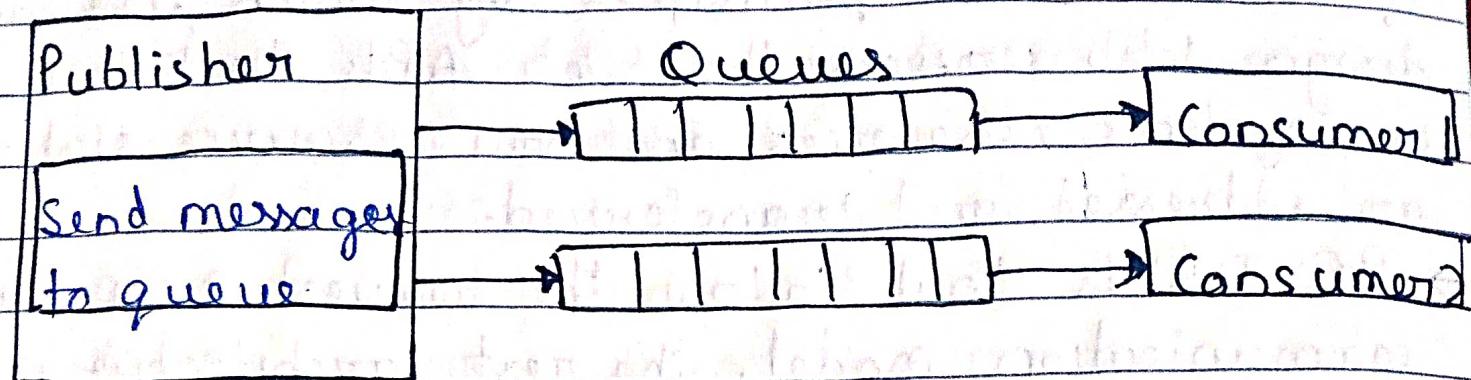
- This model involves publishers, brokers and consumers.
- Publishers are the source of data. It sends the data to the topics which are managed by the broker. They are not aware of consumers.
- Consumers subscribe to the topics which are managed by the broker.
- When broker receives the data from the publisher, it sends to all the consumers.



## • Push-Pull

- In this model the publisher push the data in queues and the consumers pull the data from the queues.
- Queues help in decoupling the messaging between the producer and consumers. Queues also act as buffer which helps in situation where there is mismatch between the rate at

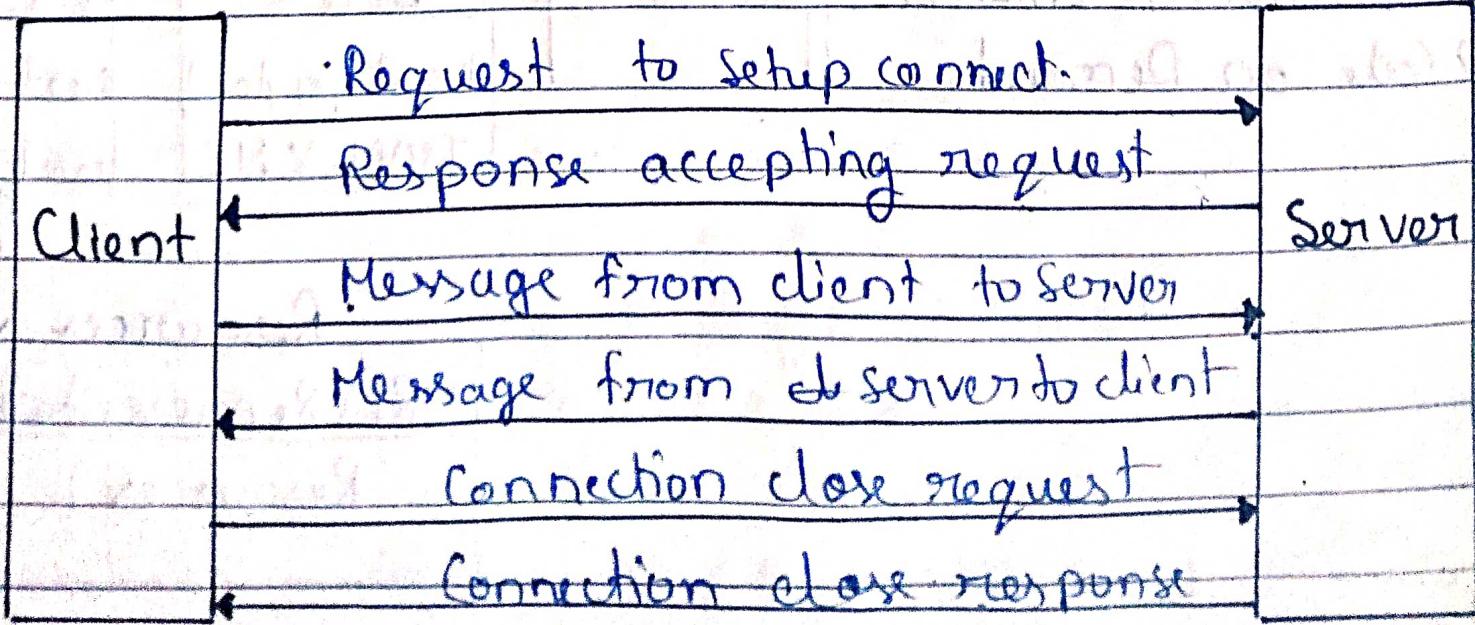
which the producers push the data and consumers pull the data.



Message pushed to queues      Message pulled from queues.

### • Exclusive Pair

- It is a bi-directional, fully duplex communication model that uses a persistent connection between the client and server, once the connection is established it remains open until the client sends a request.
- Both can send message to each other.



## IoT Communication API

### • REST Based G.A.C.A.

→ Representational state transfer (REST) is a set of architectural principles by which you 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, the rest architectural constraint apply to the components, connector and data elements, within a distributed hypermedia system.

→ The rest architectural constraint are as follows:

1) Client-Server

; HTTP

HTTP

HTTP

2) Stateless

Client

Packet

Server

3) Cache-able

4) Layered System

: Rest aware

Get, Put, Post

Authorisa

5) Uniform Interface

: HTTP client

Delete

tion,

6) Code on Demand.

: Rest payload

Restful

web service

: JSON, XML

Restful

web service

Resources

URI (Representations,  
Resources).

- WebSocket based Communication API

- It allows bi-directional, full duplex communication between clients and servers.
  - Websocket APIs follow the exclusive pair communication model.
  - W.C. begins with a connection setup request sent by the client or to server.
  - If the server supports websocket protocol the server responds to the websocket handshake response.
  - After the websocket API reduces the network traffic and latency as there is no overhead for connection setup and termination requests requirements.

