Milind Kaushal 130108018 Final BTech(IT)

IOT Assignment

Q.1 IoT stands for Internet of Things. It refers to the rapidly growing network of connected objects that are able to collect & exchange data in real time using embedded sensors. Thermost-ats, cars, lights, refrigirators & more appliances can all be connected to the IoT.

Examples:- Smart Home - From the Amazon Echo to the Nest Thermostat, there are hundreds of products on the market that users can control w/ their voices to make their lives more connected.

Wearables - watches are no longer telling time. The Apple watch & other smart watches on the market have turned our wrists into smartphone holsters by enabling text messaging, phone calls & more.

Q.2 Internet of Things (IoT) describes the network of physical objects - "things" - that are embedded with sensors, software & other tech-nologies for the purpose of connecting & exchanging data w/ other devices & systems over the internet. These devices range from ordi--ny household objects to sophisticated industrial tools.

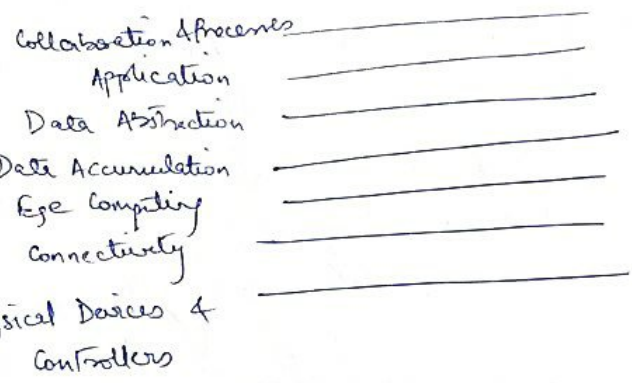
1) OneM2M

- It is about networking the machines & devices that pervade our every day lives.

OneM2M is the global standards initiative that covers require-ents, architecture, API specifications, security solutions & inter op

- ability for Machine-to-Machine & IoT technologies.
OneM2M specifications provide a framework to support applications & services such as the smart grid, connected car, home automation, public safety & health.

2) IoT World Forum Reference Model - There are seven layers of this World forum Reference Model.



→ Physical Devices & Controllers - The model calls this layer the "things" of the Internet of Things. "Things" are the sensors & devices that are directly managed by IoT architecture.

→ Connectivity - This layer spans from the "middle" of an Edge node device upto transport to the cloud.

→ Edge Computing - Required to some degree in any IoT system this layer interfaces the data & control planes to the higher layers of cloud, SaaS or enterprises of the layer.

→ Data Accumulation - Given the volume, velocity & variety that IoT systems can provide, it is important to provide data storage for processing, integration etc.

→ Edge Computing - Required to some degree in any IoT system this layer interfaces the data & control planes to the higher cloud, SaaS or enterprises o/s layer.

→ Data Accumulation - Given the volume, velocity & variety that systems can provide, it is important to provide storage for processing, integration etc.

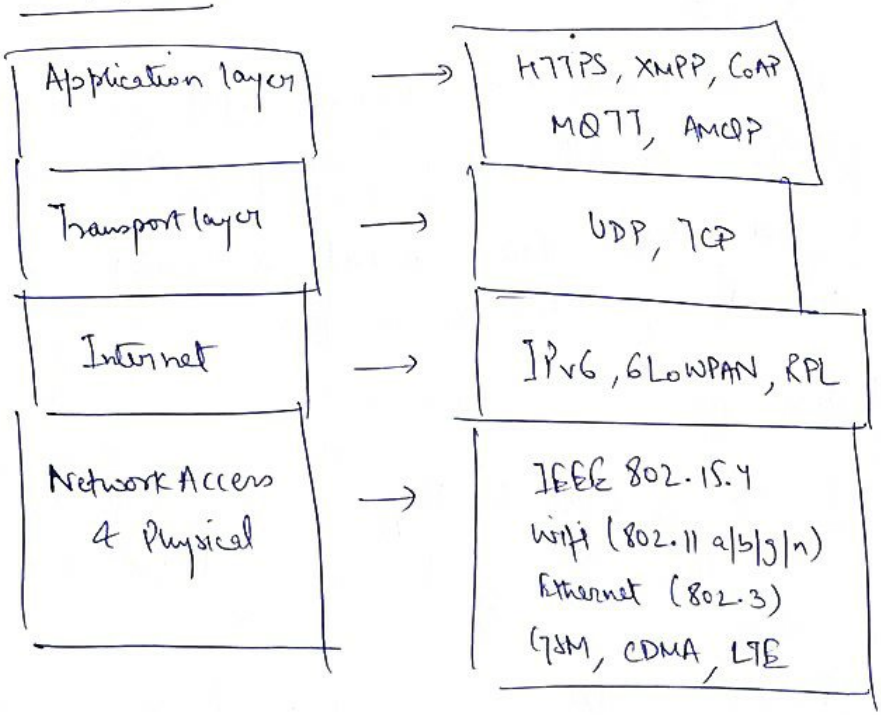
This layers saves the intermediate storage for incoming storage & outgoing traffic.

→ Data Abstraction - In this layer, we 'make sense' of the data, collected → 'link' information from multiple IoT sensors or measurements.

→ Application layer - This is where control plane & data plane logic is executed. Monitoring, process optimisation etc.

→ Collaboration & Processes - Application processing is presented to users & data processed at lower layers is integrated into business application

Q.3 IoT Protocols



required to some degree in any IoT system

interfaces the data & control planes to the higher

of cloud, SaaS or enterprises o/s layer.

→ Data Accumulation - Given the volume, velocity & variety that

systems can provide, it is important to provide a

storage for processing, integration etc.

Network Access & ~~Link layer~~ Physical layer

→ LPWAN - low Power Wide Area Network. It is a category of tech. design-ed for low-power, long range wireless comm. Ideal for large scale deployments of low-power IoT devices.

→ Wifi - It is a standard wireless networking based on IEEE 802.11 a/b/g/n specifications.

→ Ethernet - widely deployed for wired connectivity o/n local area n/ws. It implements the IEEE 802.3 std.

Internet layer

→ IPv6 - Every device at this layer is identified by their IP addresses. Typically used for IoT applications over legacy IPv4 addressing.

→ 6LoWPAN - Allows IPv6 to be used over 802.15.4 wireless n/ws. It is often used for wireless sensor n/ws.

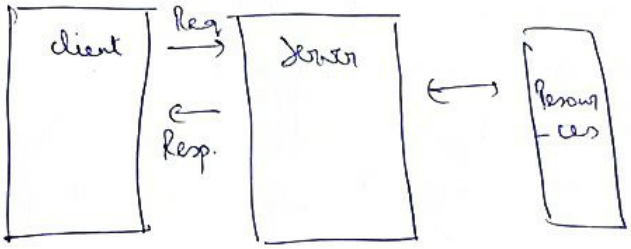
Application layer

→ MQTT - Message Queue Telemetry Transport is a publish/subscribe-based messaging protocol that was designed for use in low bandwidth situations.

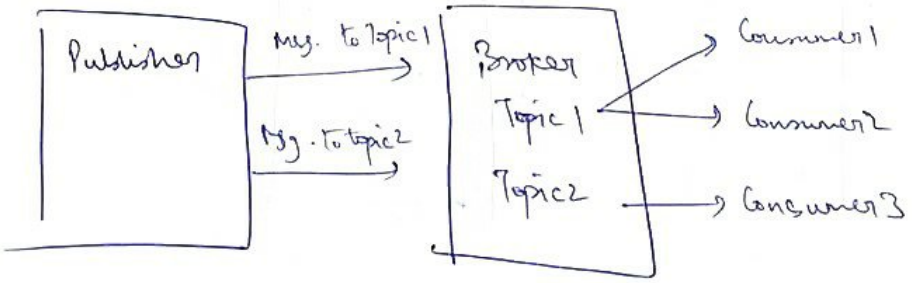
→ XMPP - Extensible Messaging & Presence Protocol was originally designed for real-time human-to-human comm. including instant messaging.

- Extensible Messaging & Presence Protocol was originally for real-time human-to-human comm. including instant

Q.4 1) Request-Response Model - The client is the IOT device that sends a request to the server. The request maybe for transfer of data or upload of data. The server maybe remote or local & can handle requests of multiple clients. This model is stateless & hence each request is independently handled.



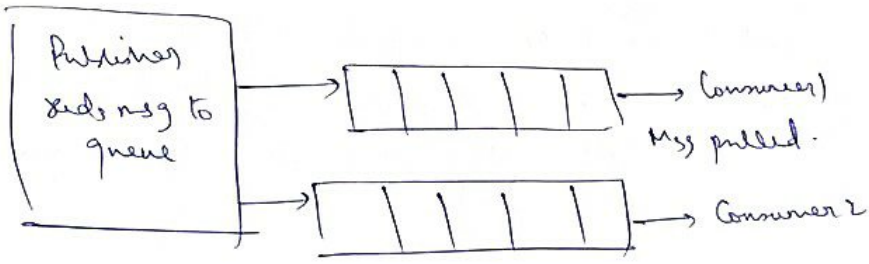
2) Publish-Subscribe Model - There are 3 entities publishers, brokers & consumers. Publishers send the data to the brokers on topics managed by the brokers. Consumers subscribe to topics & brokers send the data on topics to the consumers.



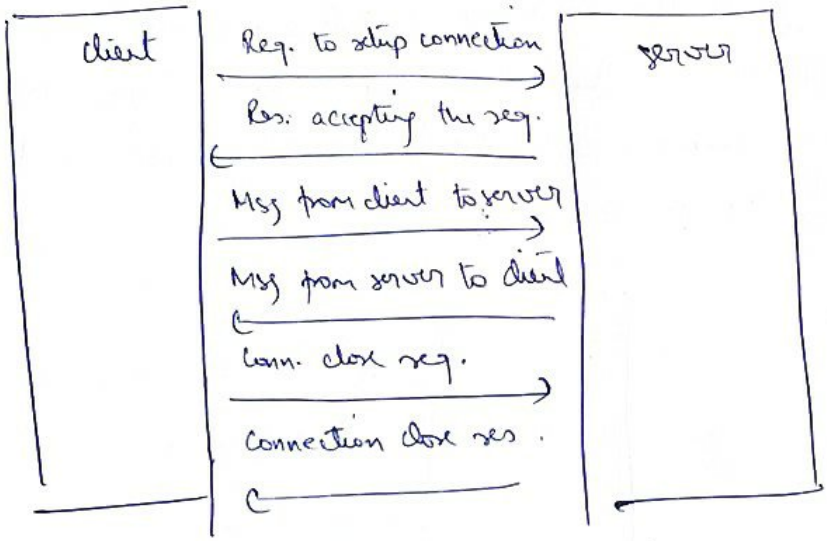
3) Push-Pull Model - Data producers push data to queues & consumers pull data from queues. Producers & Consumers are not aware of each other. Queue acts as buffers & are useful when producers produce data at

for real-time human-to-human comm. including

rate at which is faster than the rate at which consumers can download.



4) Exclusive pair - It is a bidirectional, full duplex communication model that uses a persistent connection b/w client & server. The connection is persistent & remains open till client sends a request to close the connection. It is a stateful connection model & server is aware of all open connections



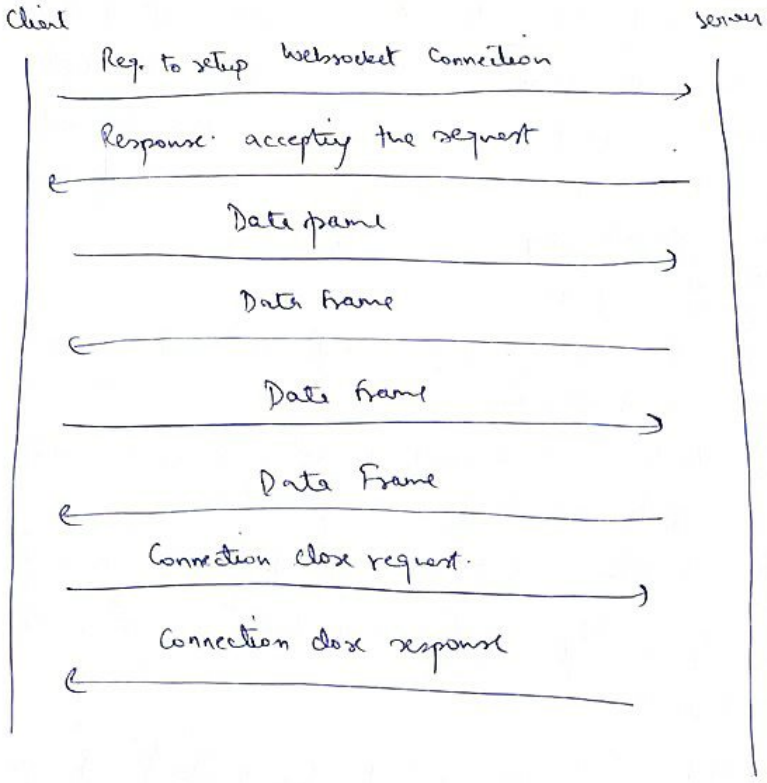
Q.5 (1) Rest Based Communication API - It is a set of architectural principles by which you can design Web services the Web APIs that focus on system's resources & how resource states are addressed & transported. Rest APIs that follow the request response comm. model ; the rest architectural constraint apply to the components.

Constraints are as follows :-

1. Client-server - The principles behind the client-server constraint is the separation of concerns.
2. Stateless - Each request from client to server must contain all the info. necessary to understand the request.
3. Cache-able - It requires that the data w/n a response to a request be implicitly or explicitly labled as cacheable or non-cacheable.
4. layered system
5. Uniform Interface - It requires that the method of comm. b/w client & server must be uniform.
6. 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.

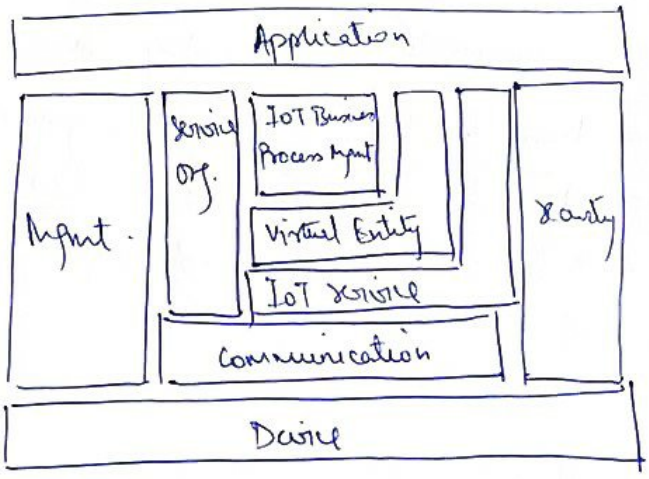
(2) WebSocket Based Communication API - They allow bi-directional, full duplex communication b/w clients & servers. They follow the exclusive pair comm. model. Unlike request-response model such as REST, these allow full duplex comm. & do not require a new connection setup request sent by the client to the server. The request is sent over HTTP & the server interprets it as an upgrade request.

These reduce the network traffic & latency.



Q.7 IOT Functional Model

Aims at describing mainly the Functional Groups (FG) & their interaction of the ARM.



→ Device functional Group - It contains all the possible functionality hosted by the physical devices used for interact the physical entities. This device functionality includes sensing, actuation, processing, storage etc.

→ Communication functional Group - Abstracts all the possible communication mechanisms used by the relevant devices in an actual system.

→ IoT Service functional Group - Corresponds mainly to the service class from the IoT domain model, & contains single IoT services exposed by Resources hosted on devices.

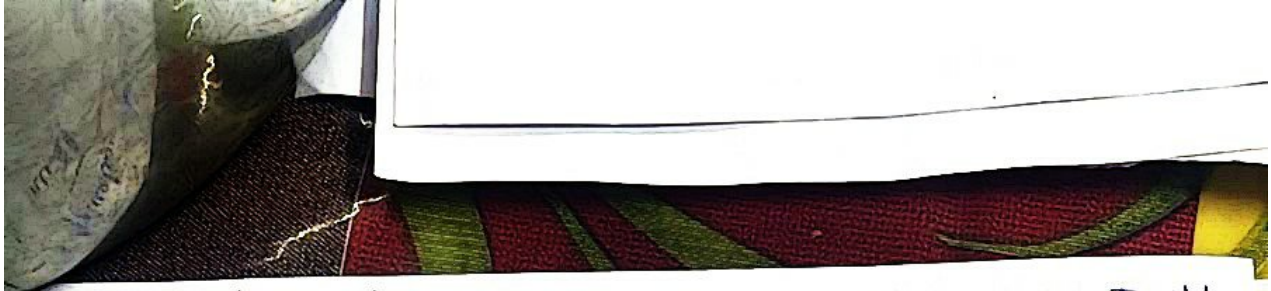
→ Virtual Entity functional Group - Corresponds to the virtual entity class in the IoT Domain model, & contains all the necessary functionality to manage associations of virtual entities with themselves.

→ IoT Service Organization functional Group - Its purpose is to host all functional components that support the composition & orchestration of IoT & Virtual Entity services.

→ IoT Process Management functional Group - It is a collection of functionalities that allows smooth integration of IoT-related services.

→ Management functional Group - Includes the necessary functions for enabling fault & performance monitoring of the system, configuration for making the system to be flexible to changing User demands.

→ Security functional Group - Contains the functional components that ensure the secure operation of the system as well as the management of privacy.



→ Application Functional Group - It is just a placeholder that represents all the needed logic for creating an IoT application.