Q1 a) Explain Connectivity of COAP client object to COAP server

It is by three methods

1) Direct Access
2) Indirect Access by using mirror server
3) By using resource directory (look up table)
4) By using Proxies

1) Direct Access:

- Here the <u>COAP client directly accesses the COAP server objects.</u>
- The client requests a list of available resources on the server and then client receives the replay from server
- The value of this reply is send to application also.

2) Indirect Access by using mirror server

- In case of energy constrained devices like sensors have long sleeping period, disconnects the network often to save the energy. Therefore preventing them to acts as traditional web server.

- Hence mirror server which is a web server is used where these sleeping nodes can store and create their own resources.

3) By using resource directory (Or lookup of objects)

- Resource directory basically contains the description of resources held on other server

- The client can discover the resources in **resource directory** by using lookup table.

- The COAP server can register its resources with more then one **resource directory**.

4) By using Proxies

- Internet works on **http** protocol.
- COAP can work not only in the constrained environment between the constrained devices and server but also between server and devices across internet.
- Hence CORE working group ensures the mapping between HTTP and COAP so that protocols can work transparently.
- **For this gateways or proxies are used**.
- Proxies are used to format and translate the protocols as required.
- **Here two types of proxies are used**

  a) COAP-HTTP Proxy **b) HTTP-COAP Proxy**

COAP-HTTP Proxy:

IT accepts the request from COAP client using COAP protocol and sends the request to HTTP server using HTTP protocol.

HTTP-COAP Proxy:

IT accepts the request from HTTP client using HTTP protocol and sends the request to COAP server using COAP protocol.

Q1 a) Explain COAP and connectivity of CoaP client to CoAP server with necessary figures.

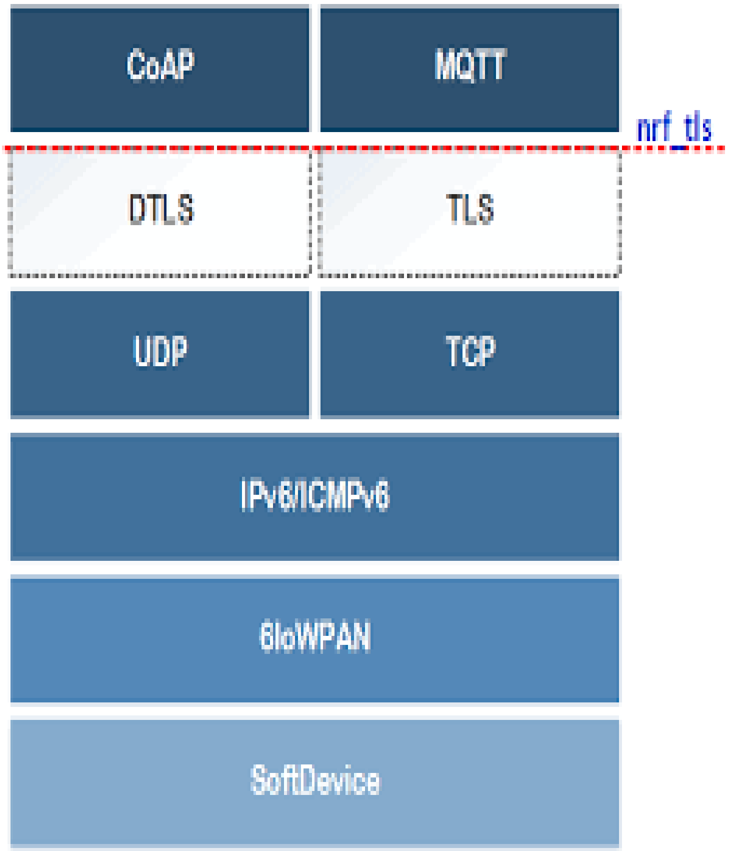
- It is lightweight application layer protocol and web transfer protocol
- Used for the constrained network i.e. it is designed for the transportation of small data between resource constrained nodes

DTLS: (Datagram Transport Layer Security)

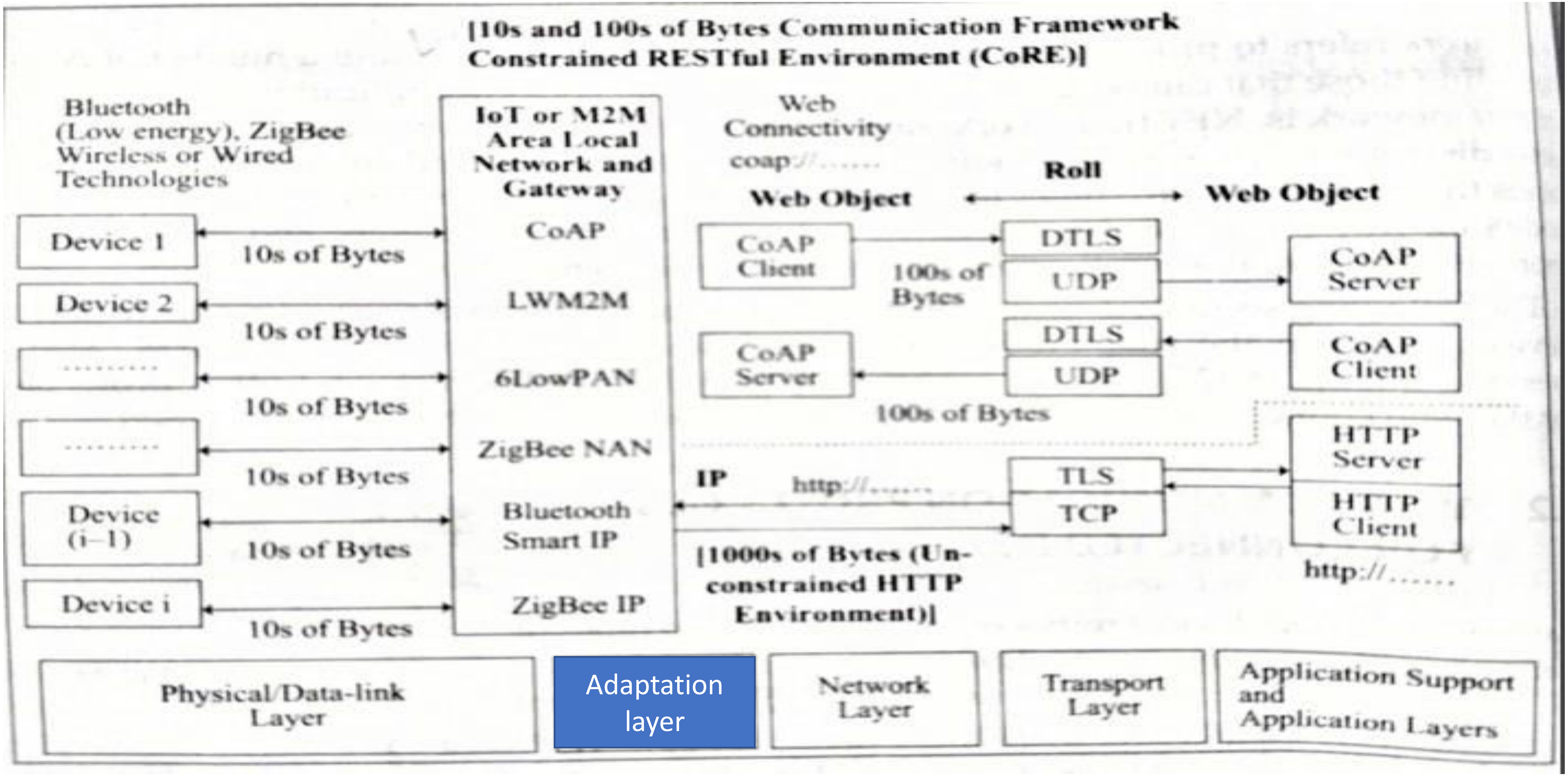
CoAP works with the UDP which is a unreliable protocol.

DTLS is application layer protocol which binds with UDP and provide

1) Security, integrity, authentication, confidentiality.
2) Packet retransmission
3) Assigning sequence number with handshake
4) Replay detection.



Web Communication Protocol for connected devices



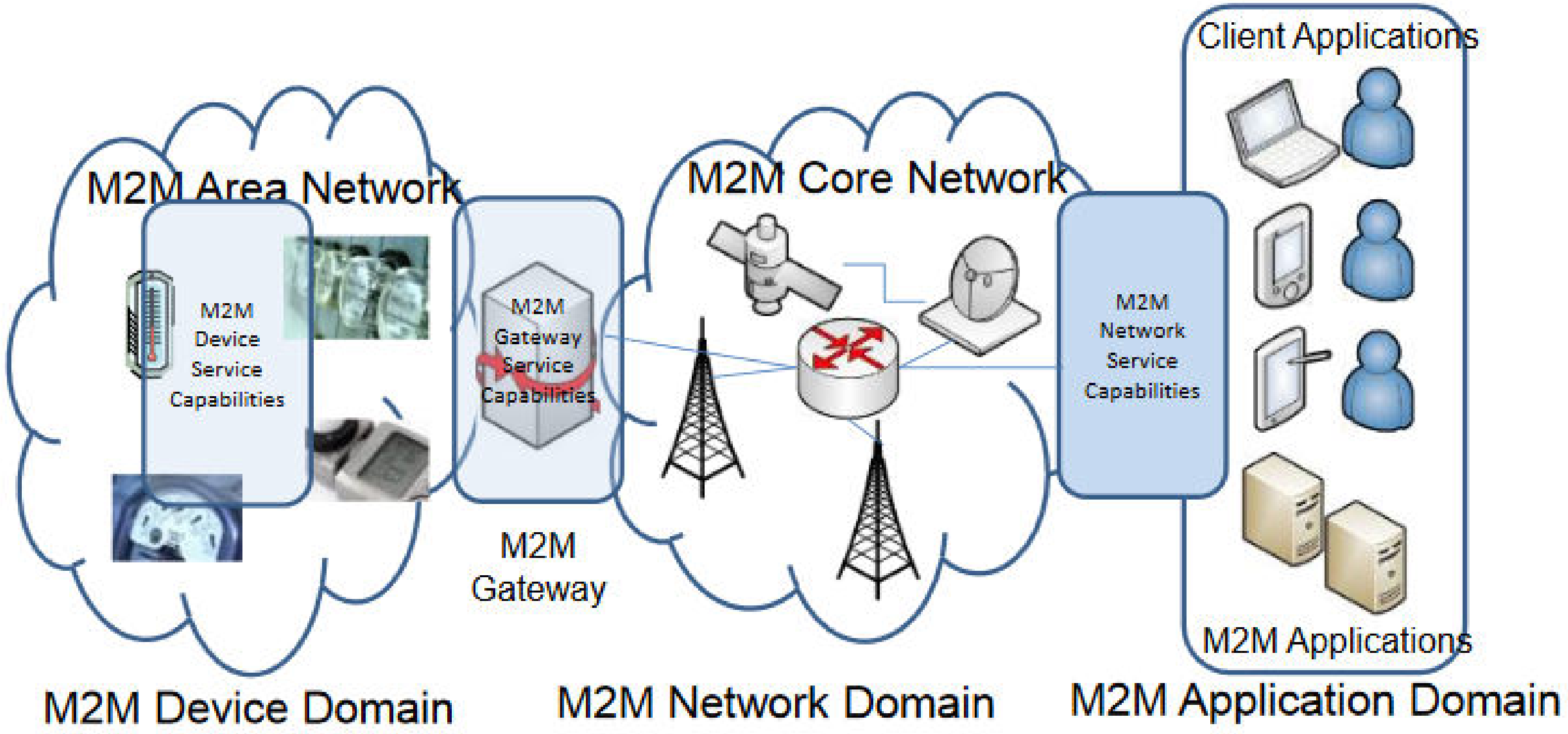
Q2 a) Explain three domains of M2M communication

- M2M refers to the process of **communication of the physical devices or machines or smart devices with the other machines of same type without intervention of humans**.

Divided into three domains

1) M2M device domain
2) M2M network domain
3) M2M application domain

ETSI M2M Network



M2M device domain

It consists of three subparts

a) **Physical devices and controllers**
b) **Communication interface**
c) **Gateway (BS)**

a) Physical devices and controllers:

- ➢ They are **sensors, physical devices, controllers, machines which are capable of transmitting data autonomously.**

a) Communication interface:

It is the port or processing unit that receives data from one interface and transmit it to other interface.

c) Gateway

Gateways and routers are the endpoints of the operator’s network in scenarios where sensors and M2M devices do not connect directly to the network

2) **M2M Network Domain (Communication Networks)**

It consists of ***M2M core and M2M service*** *capabilities.*

➢***M2M core*** covers the communications between the M2M Gateway(s) and M2M application(s), e.g. LTE, WiMAX, and WLAN.

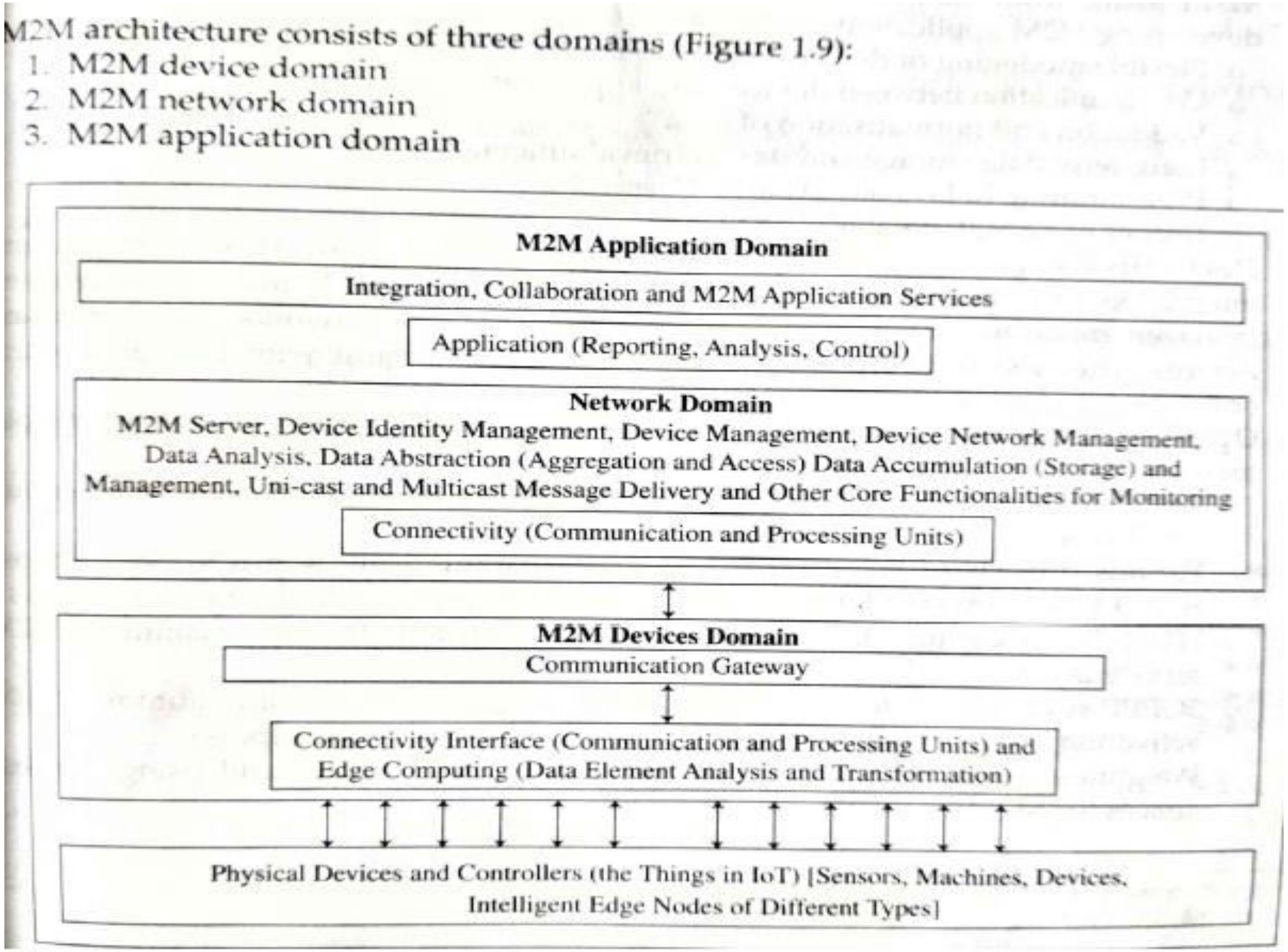
➢**M2M service** capabilities include network functions to support M2M applications. It also deals with management functions like device identity management, data storage, data collection, analysis, aggregation etc.

3) M2M application domain

Two types of applications -**M2M applications and client applications**

M2M applications: **These applications are located on the servers, interacts with M2M devices.**

Client applications: **These used to serve end-users; either receive services from M2M applications or directly from M2M devices**.



I2M architecture consists of three domains (Figure 1.9):

1. M2M device domain
2. M2M network domain
3. M2M application domain

M2M Application Domain

Integration, Collaboration and M2M Application Services

Application (Reporting, Analysis, Control)

Network Domain

M2M Server, Device Identity Management, Device Management, Device Network Management, Data Analysis, Data Abstraction (Aggregation and Access) Data Accumulation (Storage) and Management, Uni-cast and Multicast Message Delivery and Other Core Functionalities for Monitoring

Connectivity (Communication and Processing Units)

M2M Devices Domain

Communication Gateway

Connectivity Interface (Communication and Processing Units) and Edge Computing (Data Element Analysis and Transformation)

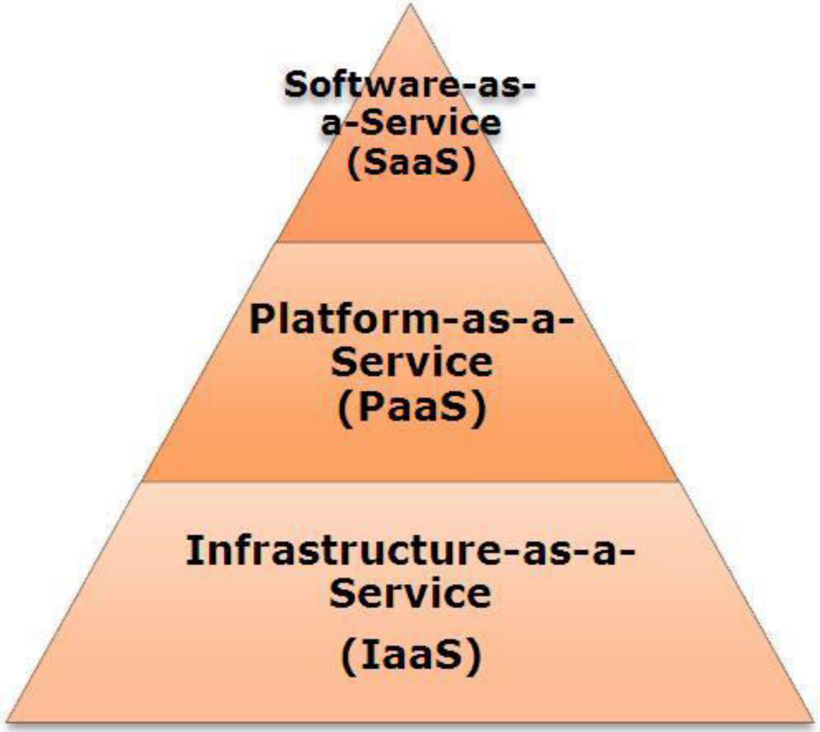
Physical Devices and Controllers (the Things in IoT) [Sensors, Machines, Devices, Intelligent Edge Nodes of Different Types]

Q 2 a). Explain Cloud computing and describe cloud service model with figure.

- Cloud computing is a paradigm that allow multiple clients to access the network and share computing resources on-demand.

- Cloud can provide services over network

Three basic service layers


Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service (IaaS)

Software as a Service (SaaS)

- **SaaS** is a method for delivering software applications over the Internet, on rent (i.e. on demand) to the end users typically on a subscription basis.

- There are several SaaS applications, some of them are listed below:

- Billing and Invoicing System
- Help Desk Applications

Platform as a Service (PaaS)

- Here instead of delivering software online, it supplies or rent a platform over some time for creating developing, testing, delivering and managing software applications like web or mobile apps, without worrying about setting up or managing the underlying infrastructure of servers, storage, network and databases needed for development

Google's App Engine, Force.com are examples of PaaS offering vendors.

Infrastructure as a Service (IaaS)

- It rent complete IT infrastructure to the user like—servers and virtual machines (VMs), storage, data center, networks, operating systems through IP-based connectivity as part of an on-demand service.

- Cloud paradigm also serves as a business model apart from technology.

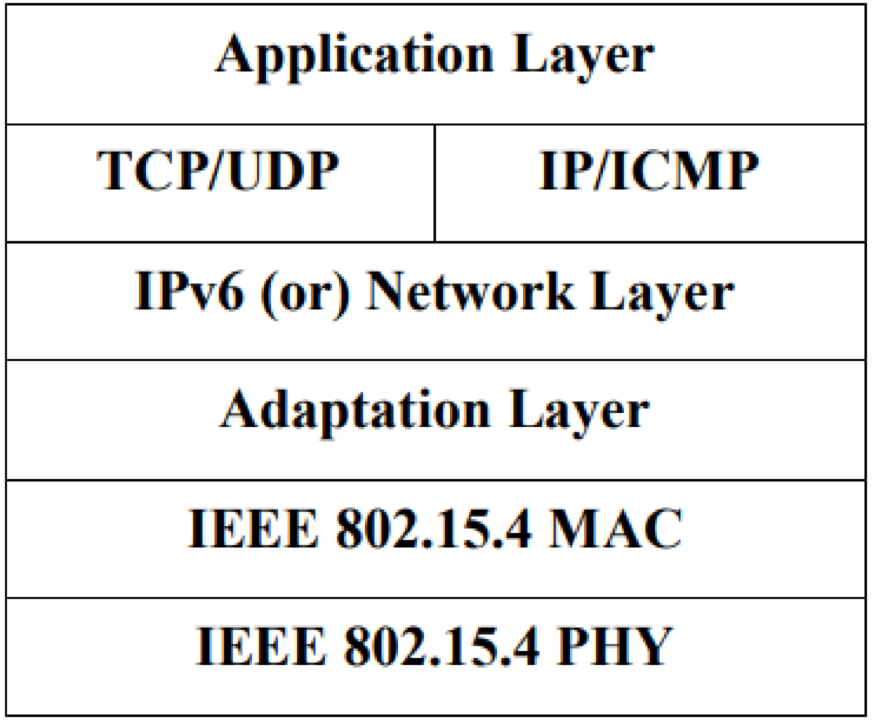
Infrastructure as a Service (IaaS)

- It rent complete IT infrastructure to the user like—servers and virtual machines (VMs), storage, data center, networks, operating systems through IP-based connectivity as part of an on-demand service.

- Cloud paradigm also serves as a business model apart from technology.

Q 3 a) Explain 6LoWPAN with necessary figure

- 6LoWPAN is an **adaptation-layer protocol** for the IEEE 802.15.4 network devices.
- The devices are the **WPAN** nodes having low power and low speed and forms a mesh network.
- Features of 6LoWPAN are **header compression, fragmentation and reassembly**.



Application Layer

TCP/UDP | IP/ICMP

IPv6 (or) Network Layer

Adaptation Layer

IEEE 802.15.4 MAC

IEEE 802.15.4 PHY

Protocol Stack of 6LoWPAN Architecture

Q 3 b) Explain in brief about XMPP.

- *XMPP uses XML technology for real time communication includes instant messaging, presence and collaboration*.
- The protocol is *used in constrained environment* for messaging.
- It is also used for publish-subscribe systems, signaling for VoIP, video, file transfer, gaming etc.
- *It enables multiuser chat using instant messaging.*

XMPP (Extensible Messaging and Presence Protocol)

X- Extensible:

XMPP is designed to be extensible, in has been designed to grow and accommodate changes.

M-Messaging:

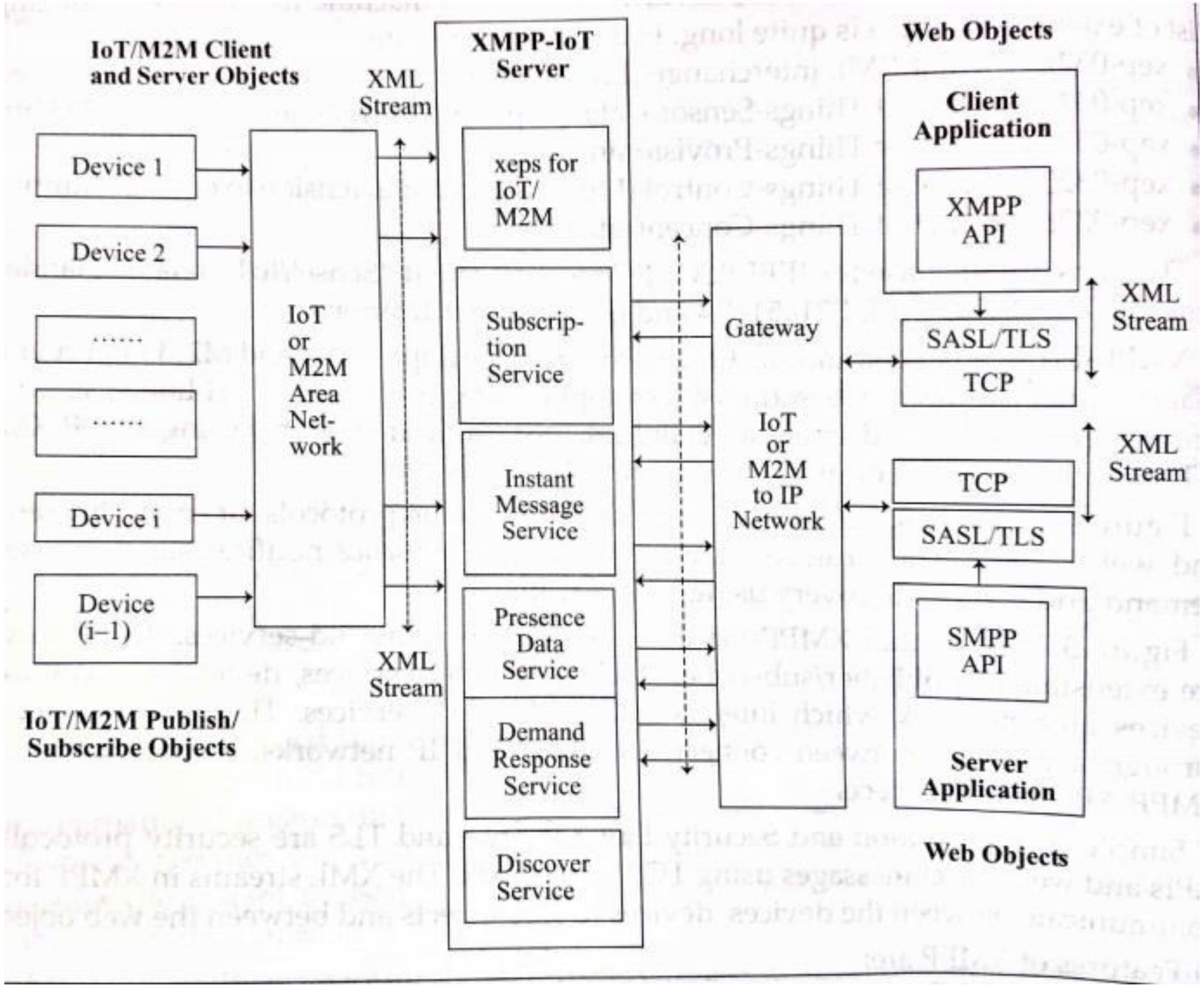
XMPP has been designed to send instant message.

P-Presence:

The presence indicator tells the server that you are online/offline/busy.

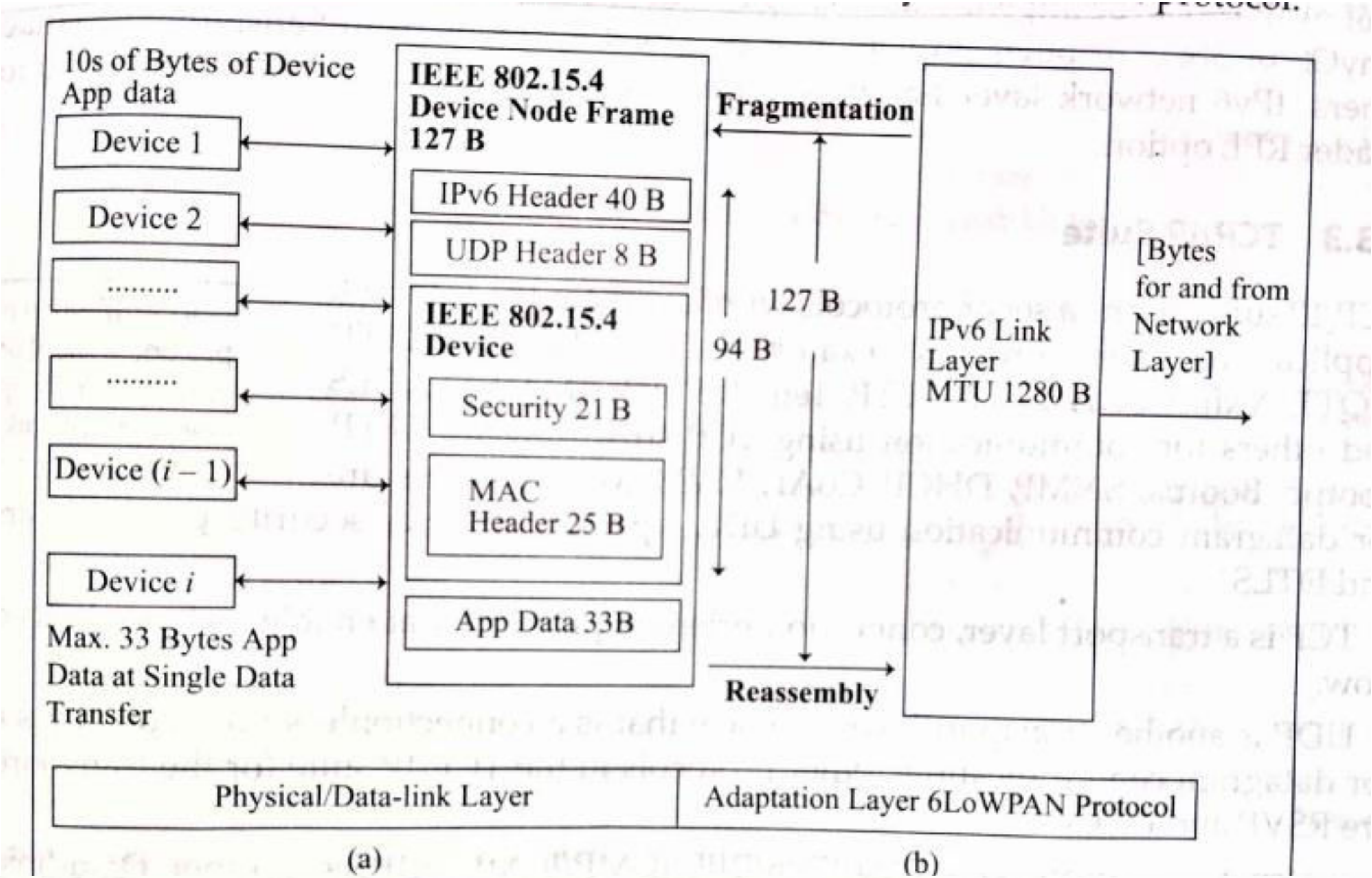
Protocol:

XMPP is a protocol; a set of standards to talk to each other. It is widely used across web but is unadvertised.



XMPP-IOT server/ XMPP M2M server is used for exchanging the messages between machines.

Figure ‘a’ shows the networked ‘i’ devices physical layer/Data link layer in IEEE 802.15.4 WPAN.
Figure ‘b’ shows adaptation layer 6LoPAN protocol



(a) Networked *i* devices at physical layer in IEEE 802.15.4 WPAN and (b) Adaptation layer 6LoWPAN protocol 127 B (maximum) fragmented frames reassembly into IPv6 maximum 1280 B or fragmentation of IPv6 MTU 1280 B into 127 B frames for transfer to a device.

Q3 a) Explain IOT? Explain conceptual frame work of IOT with necessary equations and reference model suggested by CISCO

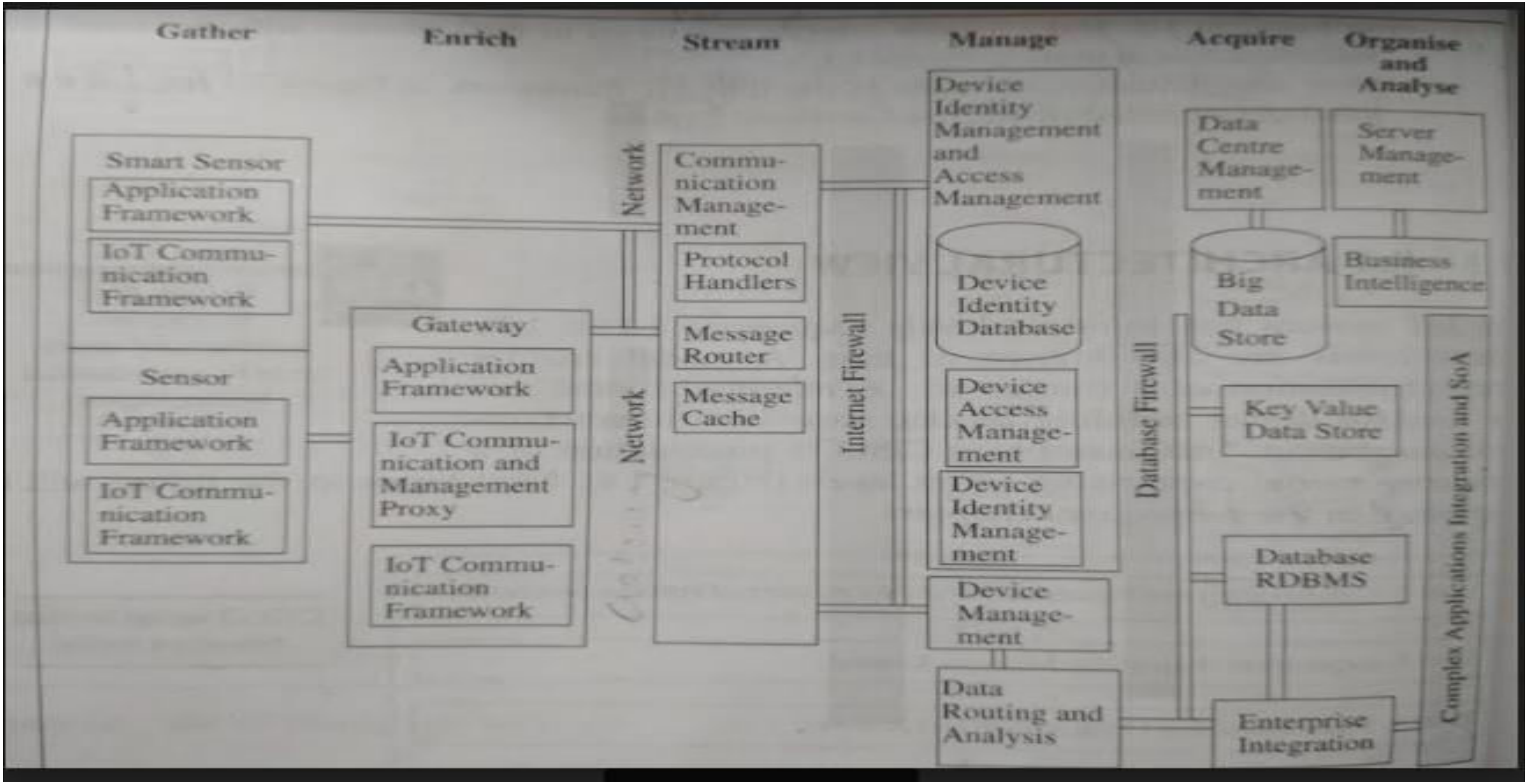
The **Internet of Things** (**IoT**) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and have connectivity which enables these things to connect and exchange data.

Equation:

Gather + Enrich + Stream + Manage + Acquire + Organize and Analyze

- *The equation represents the action and communication of the data through successive layers in IOT through interconnected devices and objects.*

ORACLE IOT ARCHITECTURE OR FRAME WORK



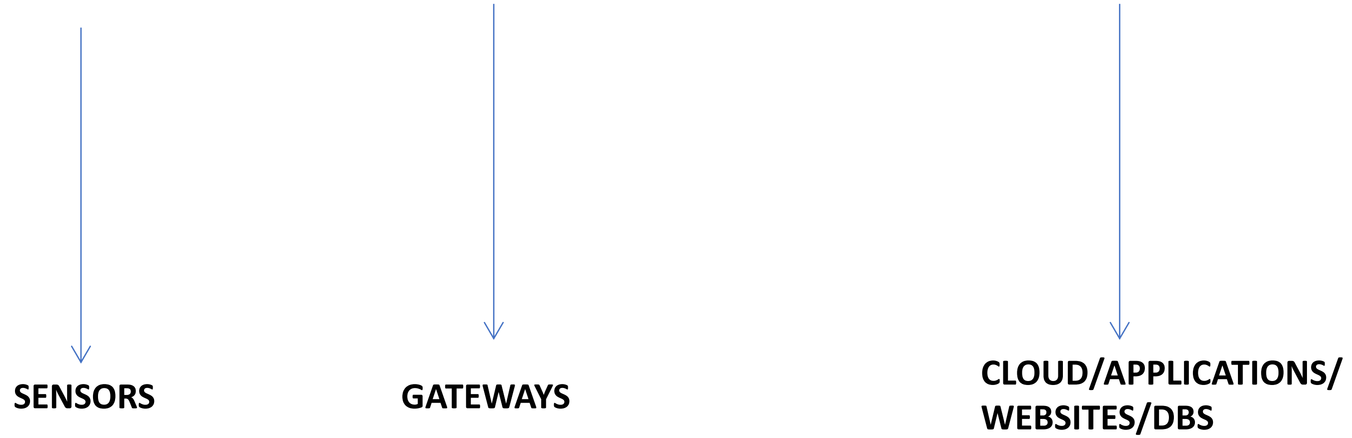
SOA: service oriented architecture

Working of Oracle IOT architecture or framework.

1) It serves as a reference in IoT applications in services and business process.
2) Smart sensors gathers or capture the data, do some preprocessing like transcoding, enrichment, data analysis, streaming also etc… and then connect to the cloud/data center through internet. But some sensors connect through the gateways where gateways connect to backend servers.
3) Backend server/Cloud/data center perform communication management, data management and device management.
4) Finally data is transferred from data center to application server or enterprise server or website which require the data

Another EQUATION given by IBM is:

Gather + Consolidate + Connect + Collect + Assemble + Manage and analyse= IOT



SENSORS

GATEWAYS

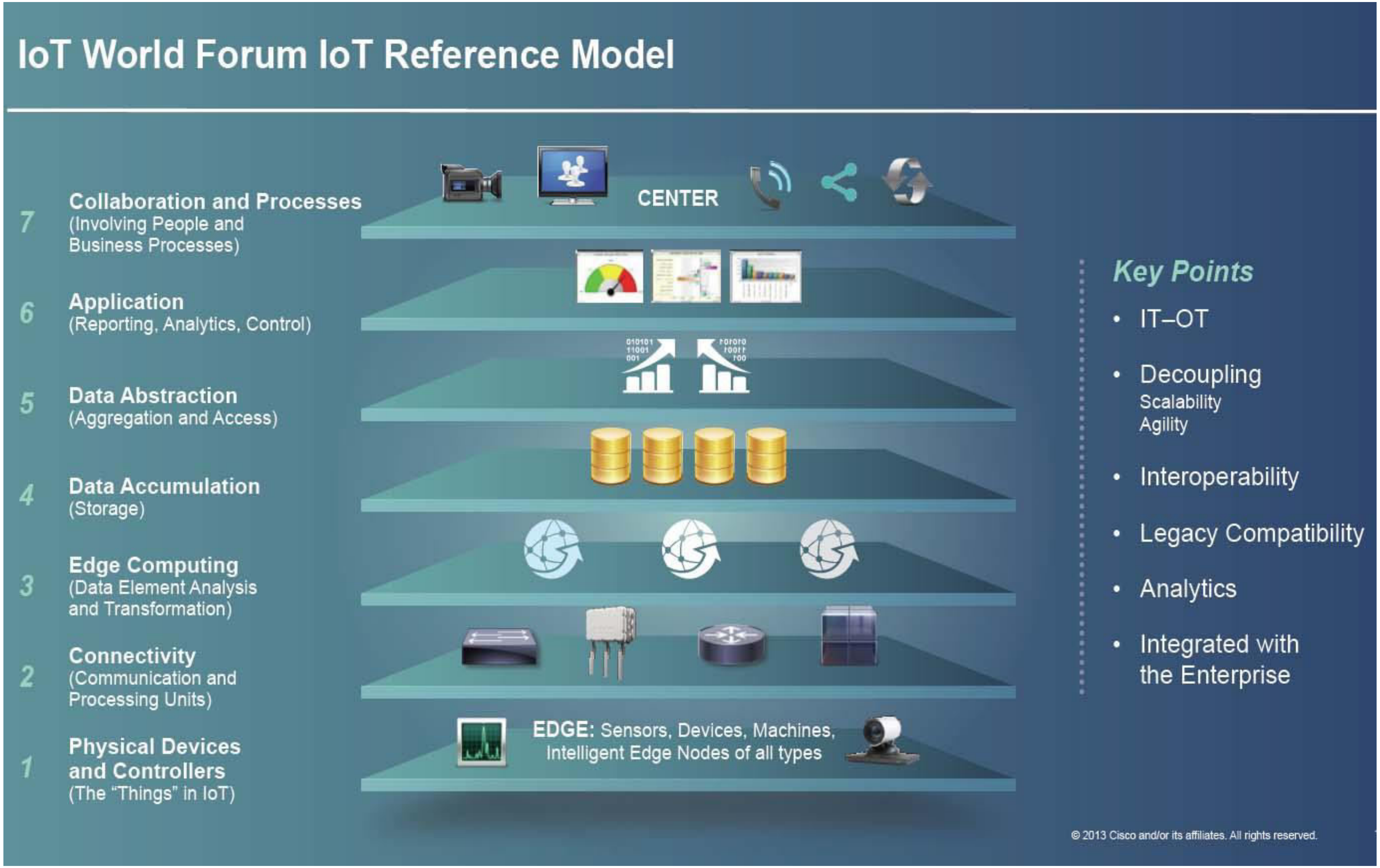
CLOUD/APPLICATIONS/
WEBSITES/DBS

This equation defines a general framework for IOT using cloud based services.

IOT Architectural view and IOT reference model suggested by CISCO

(Refer Ch01 Fig. 1.4 of the Book)

IOT reference model suggested by CISCO (Architectural view)



IoT World Forum IoT Reference Model

7 **Collaboration and Processes** (Involving People and Business Processes)

6 **Application** (Reporting, Analytics, Control)

5 **Data Abstraction** (Aggregation and Access)

4 **Data Accumulation** (Storage)

3 **Edge Computing** (Data Element Analysis and Transformation)

2 **Connectivity** (Communication and Processing Units)

1 **Physical Devices and Controllers** (The "Things" in IoT)

CENTER
EDGE: Sensors, Devices, Machines, Intelligent Edge Nodes of all types

Key Points

- IT–OT
- Decoupling
  Scalability
  Agility
- Interoperability
- Legacy Compatibility
- Analytics
- Integrated with the Enterprise

Gather + Enrich + Stream + Manage + Acquire + Organize and Analyze

IEEE suggested P2413 standard for Architecture of IoT

- A reference architecture of IoT
- The IOT reference model has 7 levels called "LAYERS" OR "TIERS".
- Each level is defined with some terminology.
- Each level perform some specific function.
- The model describes how the task at each layer should be handled to maintain simplicity and scalability.
- IN IOT the data flows in both directions i.e.

    From top to bottom (LAYER 7 to LAYER 1) – control pattern.

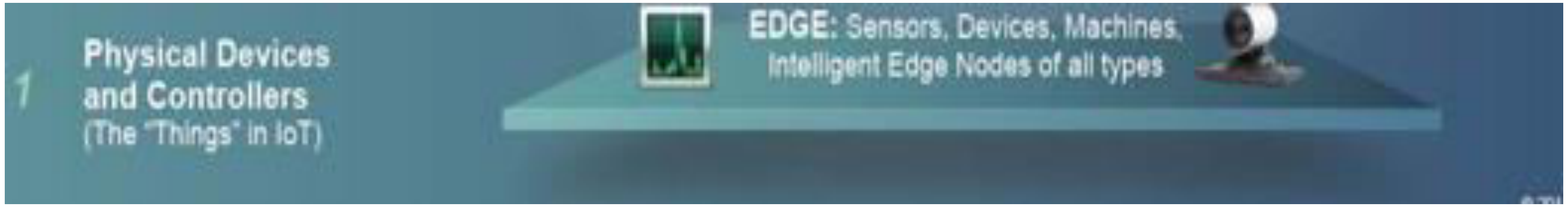
    From bottom to top (LAYER 1 to LAYER 7) – monitoring pattern

But Basically follows top-down approach (means consider top layer design first and then move to the lowest).

Architecture of IoT

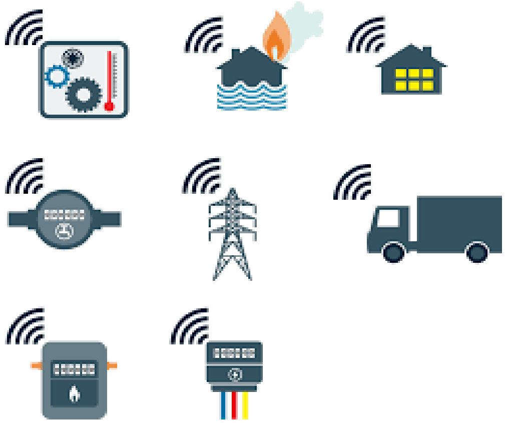
- It defines basic architectural building blocks and their integration capability into multi-tiered systems.
- The reference model defining relation-ships among various IoT verticals, for example, transportation and healthcare
- Gives a blueprint for data abstraction
- Recommends quality ‘quadruple’ trust
- “Protection, Security, Privacy, and Safety”
- Defines no new architecture and no reinvent but existing architectures congruent with it

LAYER 1. Physical devices and controllers.



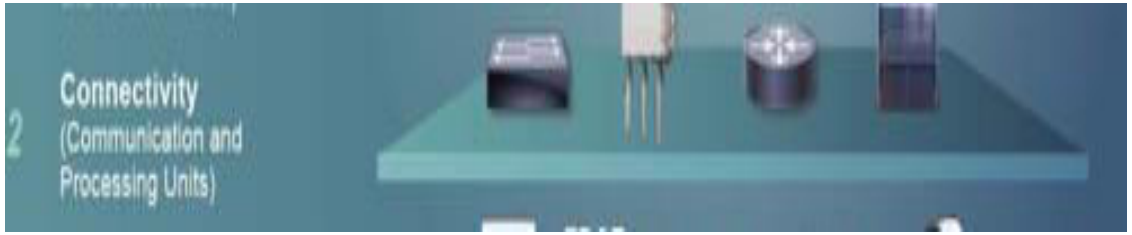
1 Physical Devices and Controllers (The "Things" in IoT)

EDGE: Sensors, Devices, Machines, Intelligent Edge Nodes of all types



- They are the **physical devices** , also called as "**THINGS**" in IOT .
- Basically they are **Embedded Devices**, Embedded hardware/software like Sensors/Actuators , RFID, Hardware (Arduino, Raspberry Pi, Intel Edison, Beagle Bone Black and Wireless SoC…).
- They are **ready to send and receive the information**.
- Devices are **unlimited, diverse and no rules about the size**, location etc…. For example..
- Devices are capable of **Analog to digital conversion** and vice versa.
- Devices are capable of **generating data and being queried**.

LAYER 2. Connectivity (Communication and processing units)



Processing Units:

- Contains **Routers and Gateways**
- Main task is to deliver the right information at right time and to right machine i.e. reliable transmission.

Commu-
nication
Manage-
ment
Protocol
Handlers
Message
Router
Message
Cache

Communication:

- Includes **protocol handlers, message routers, message cache**
- It can be between smart device and network/ internet directly
- It can be through gateways then to network
- **Therefore main task involves switching and routing, enriching, transcoding, translation between protocols, security and self learning etc.**
- **Communication occurs networks – EAST-WEST communication**

Popular Communication Protocols Used from sensors to gateways are:

- ZigBee, 6LOWPAN , Bluetooth, RFID
- WiFi, WiMax, 2G/3G/4G/5G

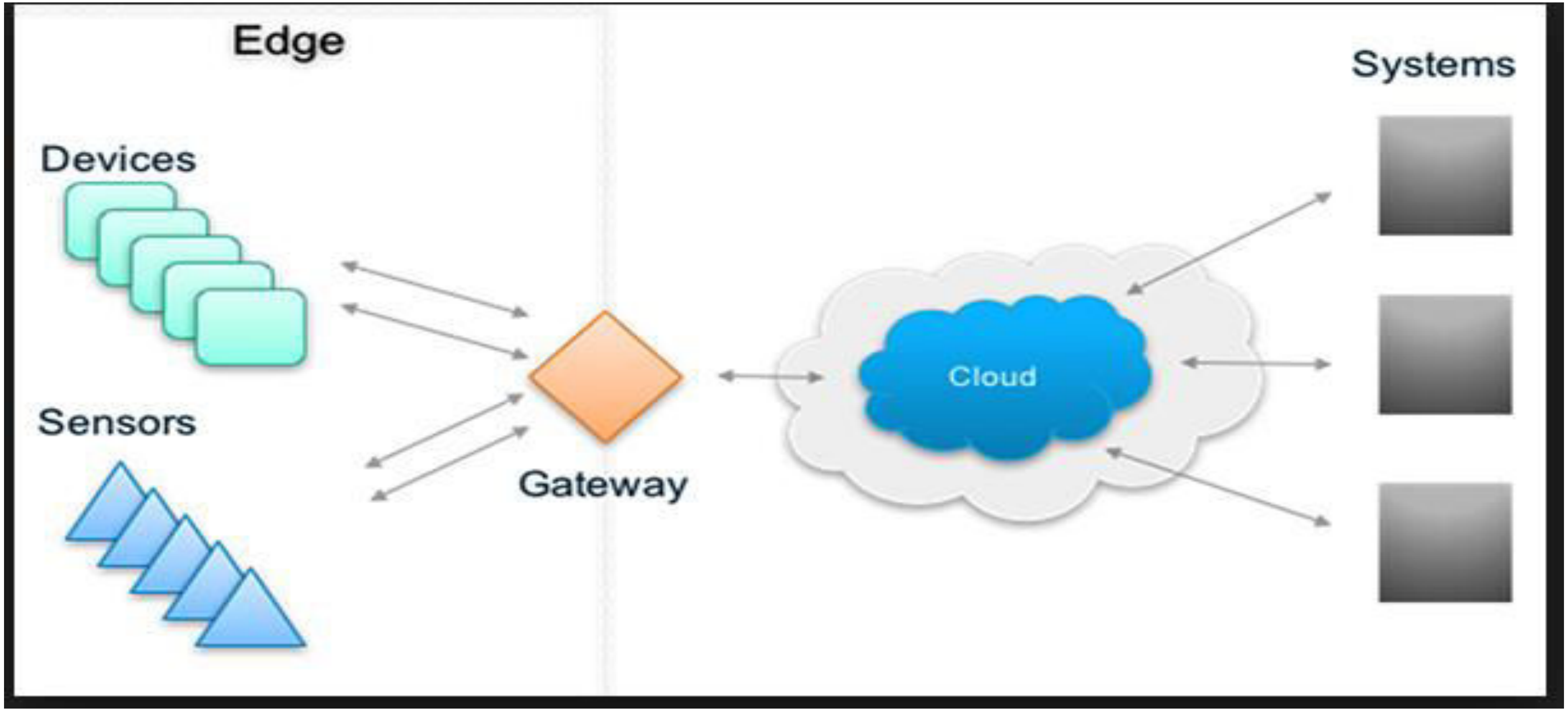
Layer 3 [Edge Computing Or Fog Computing]

- ➢Edge computing is an architecture that uses **edge devices / network edge** like **routers, gateways, switches**, multiplexers, integrated access devices to do some **preprocessing of data**.

- ➢**Preprocessing** includes data aggregation, storage, data filtering, cleanup, analysis, transformation (formatting, decoding, distillation) , Threshold(alert), event generation etc.

- ➢Finally the data is routed to web servers/cloud.

Then the data from gateways are send to higher layers i.e. to backend server/cloud or data base centres using some protocols like CoAP, RESTful HTTP, MQTT, XMPP (Extensible Messaging and Presence Protocol)



Layer 4 [Data Accumulation and storage]

Data management is done at backend server/cloud or data base centres

Main roles of layer 4 are:

- **Convert data in motion to data at rest.**
- **Convert format from network packets to database relational tables.**
- **Convert Event based data to query based data (it bridges the gap between real time networking and non real time)**
- **The concept of BIG DATA is used at layer 4.**

Layer 5 Data Abstraction

Data abstraction is done at backend server/cloud or data base centres

Abstraction **means providing the essential and relevant information** of the data by hiding the irrelevant one.

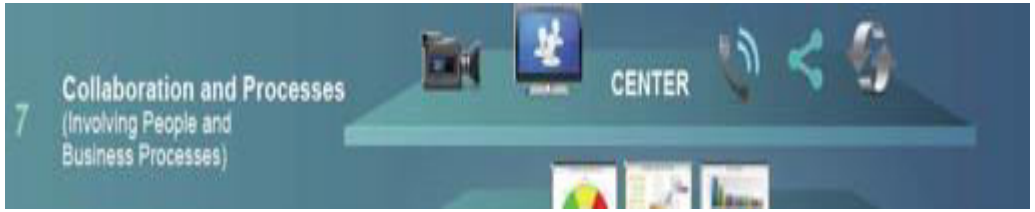
Main roles are:

1) **Provide multiple storage systems** to accommodate data from different IOT devices.
2) **Reconciling multiple data format** from different sources.
3) **Filtering, selecting, projecting and reformatting the data** to serve client application.
4) **Protecting the data** with appropriate authentication and authorization.

Layer 6 Application

- Layer 6 deals with reporting, analysis and control
- i.e. the data is analysed and then send to controlling device like actuator.
- And then the data is passed to specific application like mobile application or webpage or to the business enterprise which require that data.

Layer 7 Collaboration and processes.



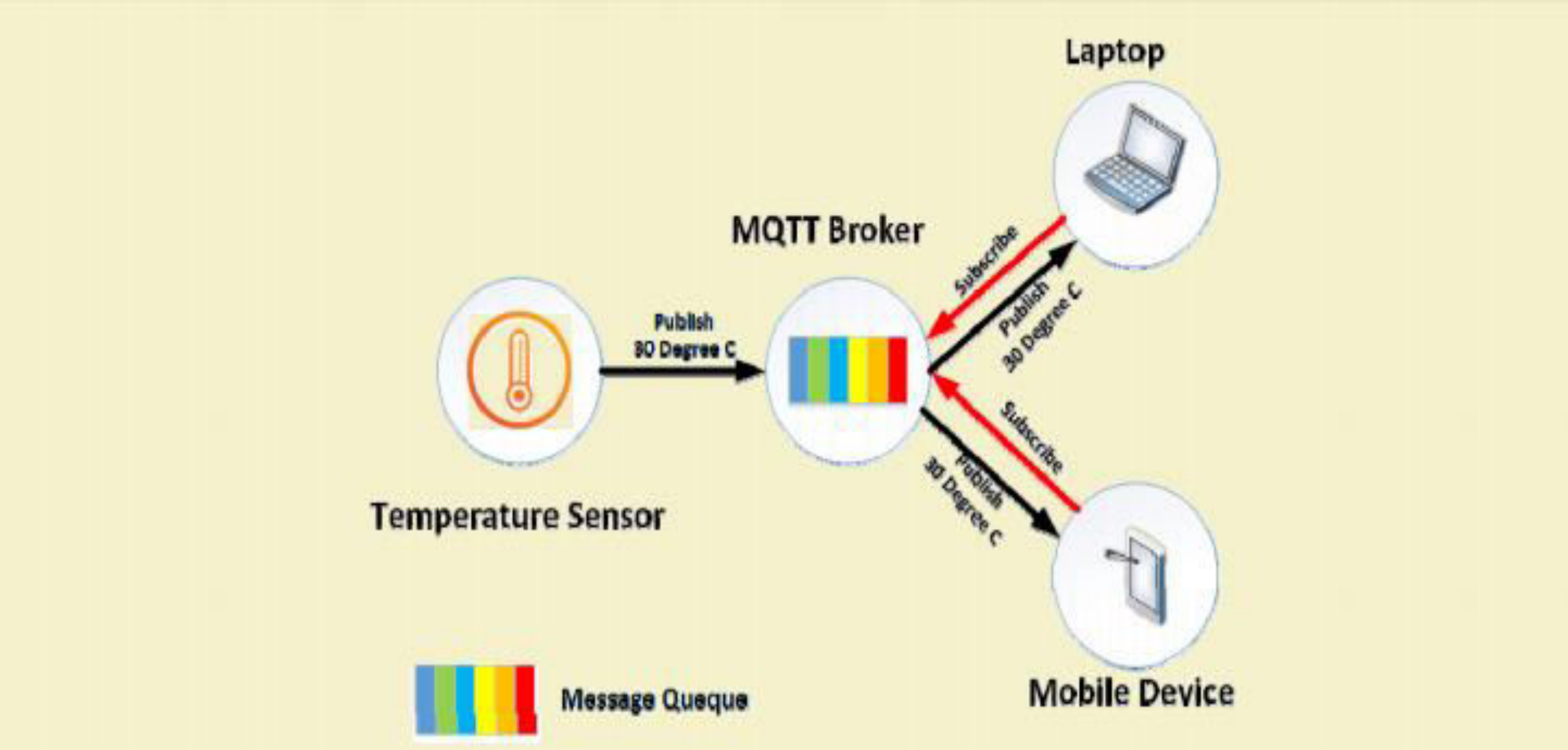
- It means involving people and business process.
- Basically multiple people are using same applications for a range of different purpose
- But in IOT the main objective is to empower people to do their work better, not the application.

Q 4 a) Explain MQTT with pub/sub model with figure

- An **open source protocol for machine-to-machine (M2M**)/"Internet of Things" connectivity

- Designed to **provide connectivity** (mostly embedded) **between applications** and middle-wares **(M2M/IOT objects)** on one side **and networks and communications (WEB Objects)** on the other side.

Communication



Communication

1) **Publisher:**

- These **clients first make connections to the Broker and then publish a message to the broker.**
- **The message include the topic**. The topic is the routing information for the broker.

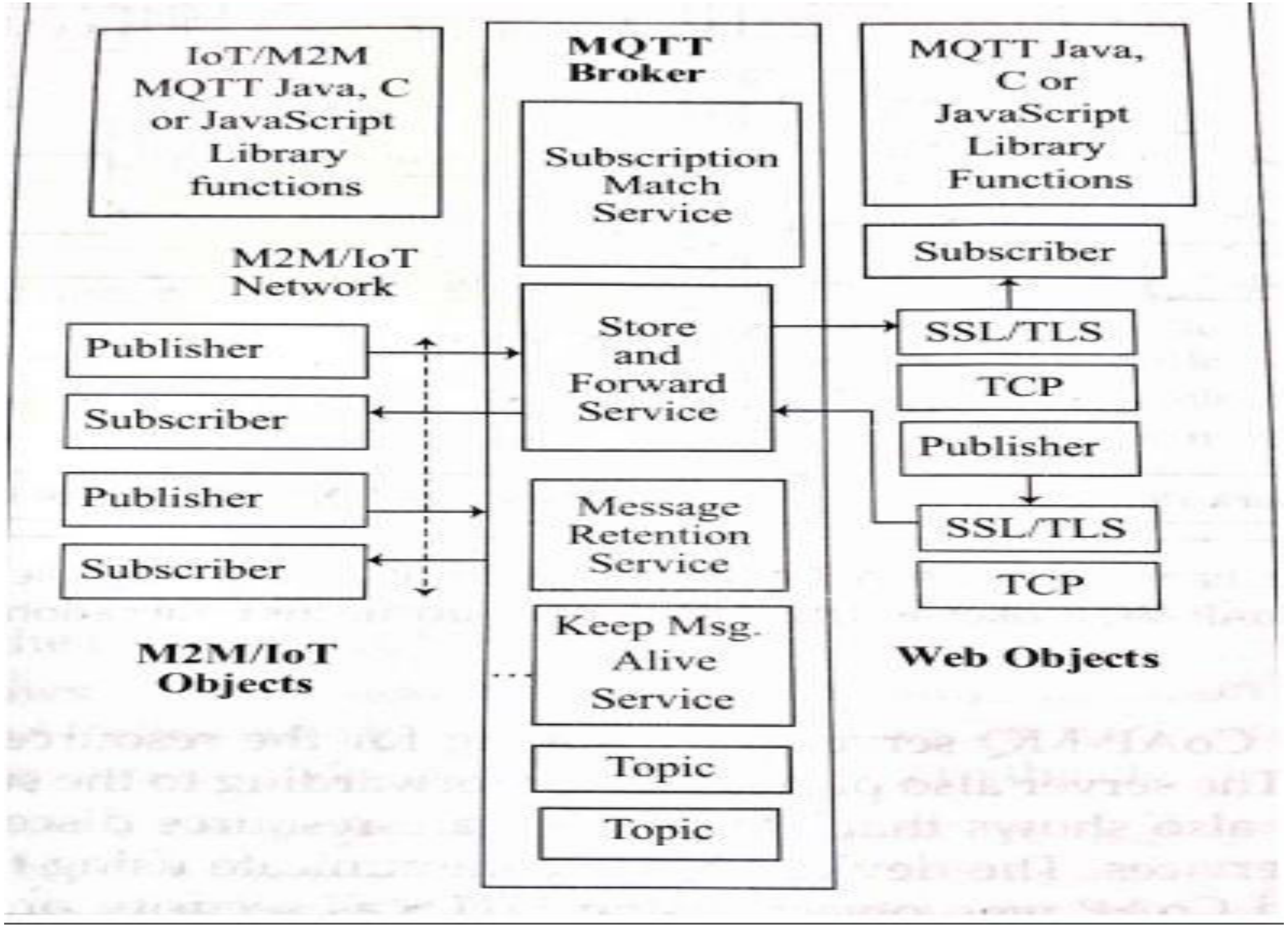
2) **Broker:**

- Perform **store and forward** operation
- **Receives the topics** from **publishers**
- **Each client** that wants to receive messages **first subscribes to a certain topic** and then **the broker delivers all messages with the matching topic to the client**.

3) **Subscribers:**

- They are the clients that require the information from publishers

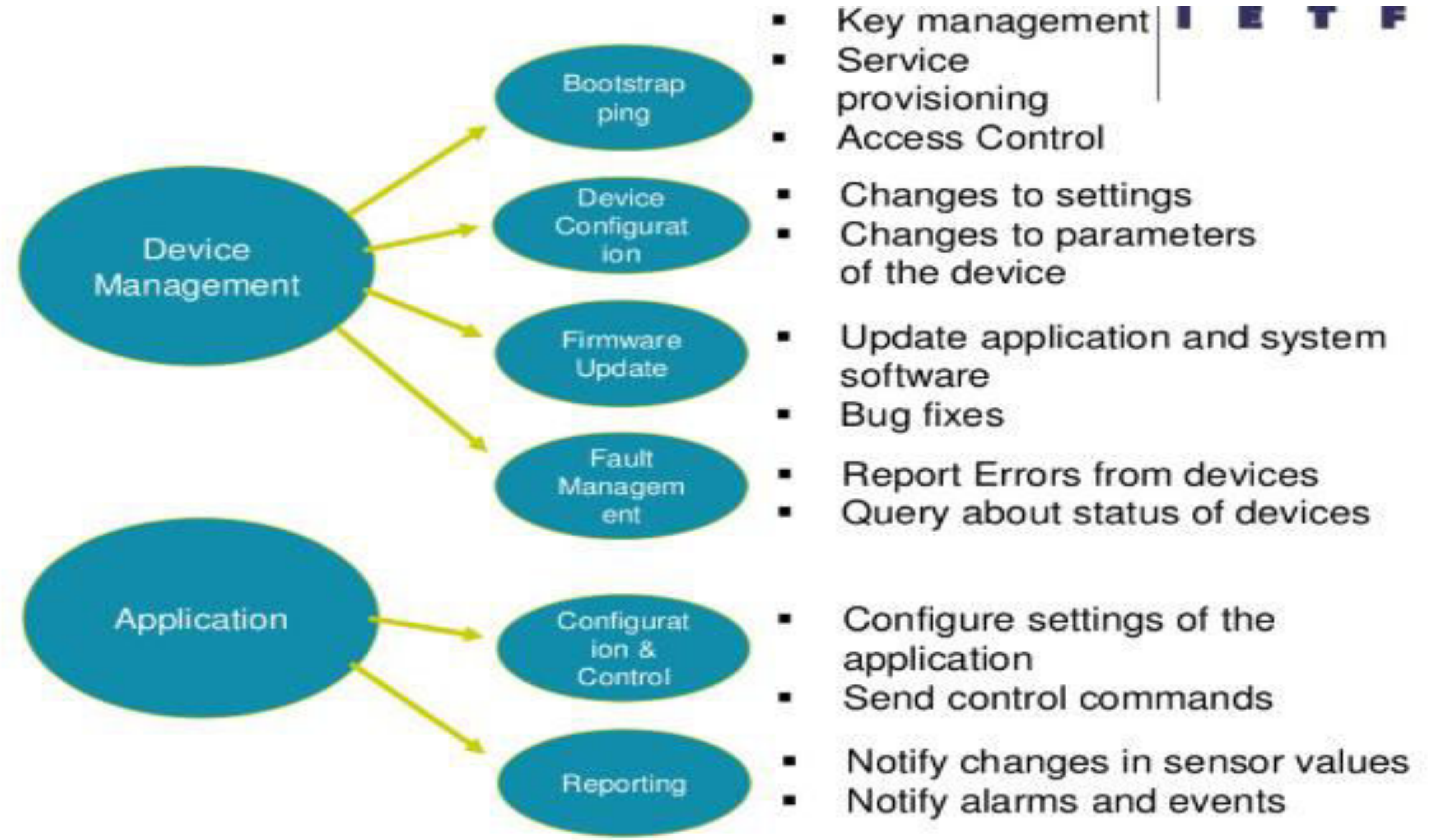
MQTT (Message Queuing Telemetry Transport)


IoT/M2M MQTT Java, C or JavaScript Library functions
M2M/IoT Network
Publisher
Subscriber
Publisher
Subscriber
M2M/IoT Objects
MQTT Broker
Subscription Match Service
Store and Forward Service
Message Retention Service
Keep Msg. Alive Service
Topic
Topic
MQTT Java, C or JavaScript Library Functions
Subscriber
SSL/TLS
TCP
Publisher
SSL/TLS
TCP
Web Objects

Q 4 a) Explain the web communication protocol LIGHT WEIGHT MACHINE TO MACHINE COMMUNICATION with figure.

- Called as light weight
- ✓ **As it can transfer upto 100s of bytes** unlike webpages of 1000s of bytes.
- ✓ The **format of data transfer between client and server is TLV** (Tag Length value) **or JSON** (Java Script Object Notation)
- ✓ The protocol is compact and have **small header.**

FEATURES



- Key management
- Service provisioning
- Access Control

- Changes to settings
- Changes to parameters of the device

- Update application and system software
- Bug fixes

- Report Errors from devices
- Query about status of devices

- Configure settings of the application
- Send control commands

- Notify changes in sensor values
- Notify alarms and events

Device management:

Bootstrapping

- It is the procedure for the device to get the secret keys and URL for reaching the servers.
- It's also useful for re-keying, upgrading security scheme or redirecting your device to another server.

Device configuration

- Changes to settings and parameter of the device.
- Make sure that device is working properly on the network.

Firmware Update

- Updating of system software, application, bug fixing etc**.**

Fault management

- Report about the error from devices
- Query about status of devices**.**

Objects:

They are the resources (information like read, write or execute) created by device application like plain text, or batches of resources like TLV, JSON, or Binary format

LWM2M client:

- It is the software that runs on device (like library around 5k Bytes)
- The client can manage a number of objects (resources) created by device application.

Interface:

- The objects are made accessible between client and LWM2M server (on the backend) using interface.

LWM2M server

- This server is used by multiple applications like for device management, network management, application data.

Interface:

It provides features like

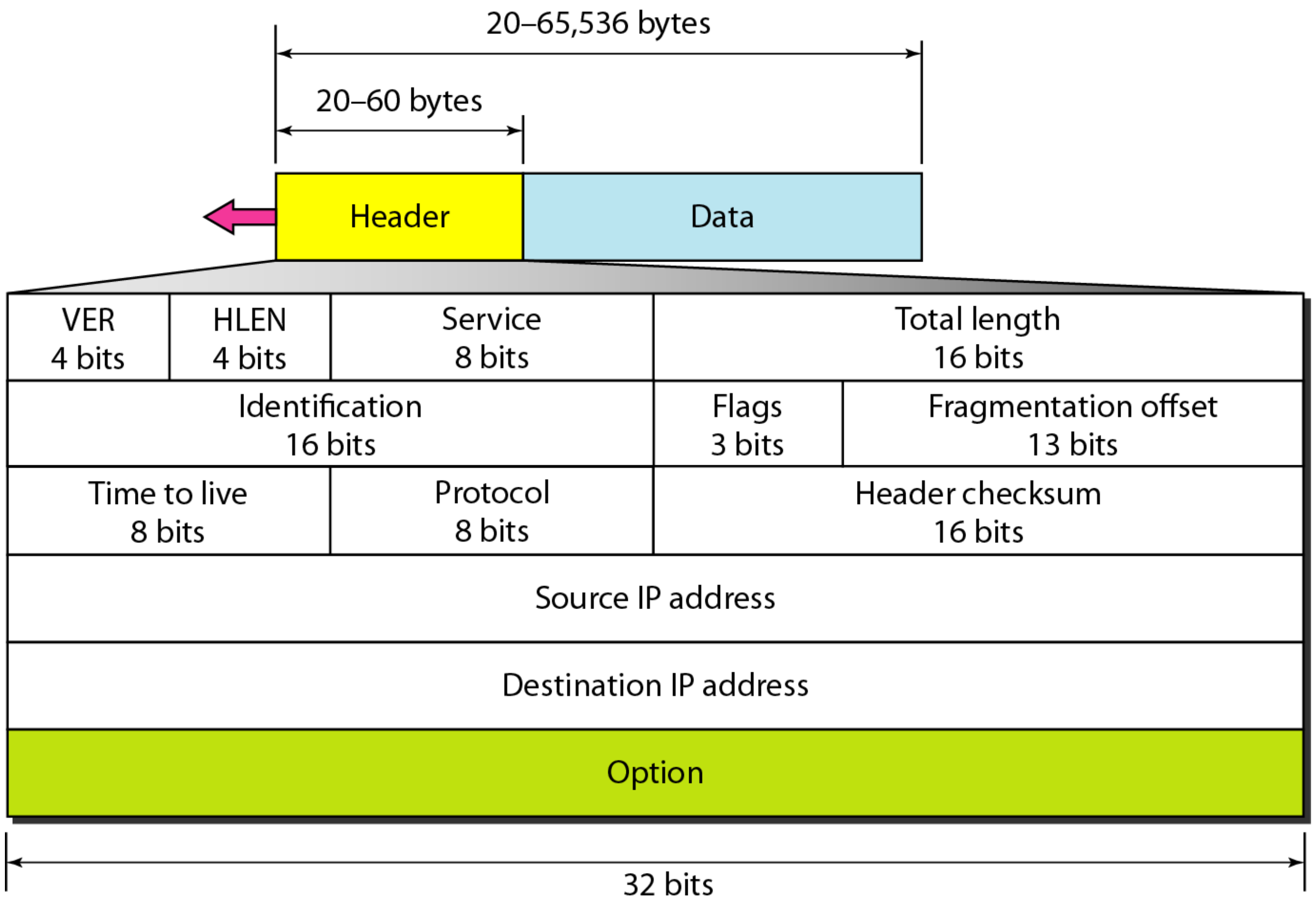
a) **Bootstrapping** :managing the keys, access control and configuration of device

b) **Registration :** LWM2M client register to LWM2M sever to let server know about its existence.

c) **Object and resource access:** Once server knows about the object, it can send command to the resources of objects, it can read the values to read the values of object.

d) **Report:** Allows the client to report resource information and periodically update and event.

Q 5 a) Explain IPv4 data stack.



<table>
<tr><td>VER
4 bits</td><td>HLEN
4 bits</td><td>Service
8 bits</td><td colspan="2">Total length
16 bits</td></tr>
<tr><td colspan="3">Identification
16 bits</td><td>Flags
3 bits</td><td>Fragmentation offset
13 bits</td></tr>
<tr><td colspan="2">Time to live
8 bits</td><td>Protocol
8 bits</td><td colspan="2">Header checksum
16 bits</td></tr>
<tr><td colspan="5">Source IP address</td></tr>
<tr><td colspan="5">Destination IP address</td></tr>
<tr><td colspan="5">Option</td></tr>
</table>

Version (VER). This 4-bit field defines the version of the IPv4 protocol.
Currently the version is 4.

Header length (HLEN). This 4-bit field defines the total length of the
Datagram header in 4-byte words.

I.e minimum 20 bytes and maximum 60 bytes

Services. This is 8-bit field,

Previously it is called as type of service and now is called differentiated services.

It assigns **the priority to each IP packet**

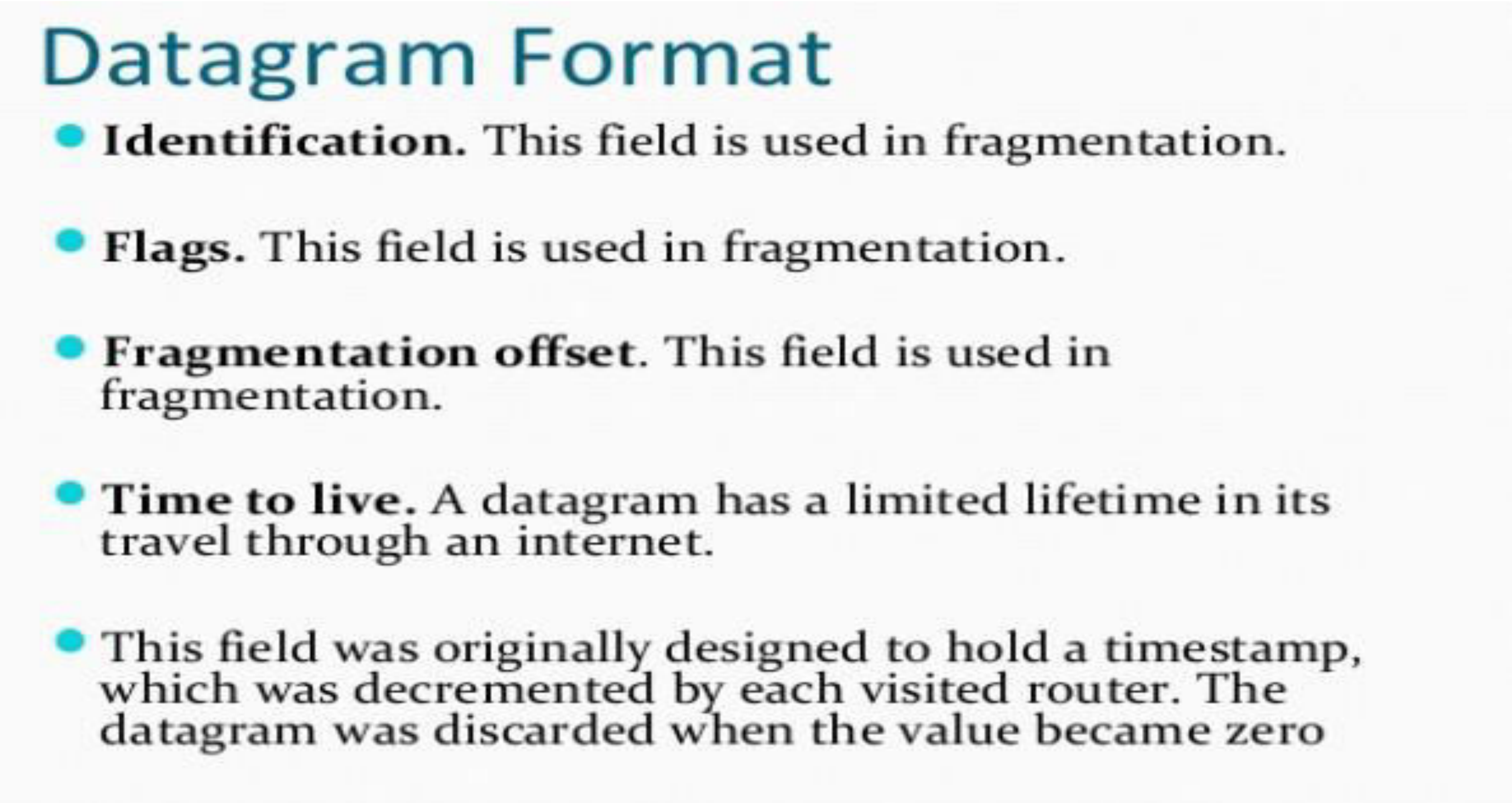
Types of service field

1	2	3	4	5	6	7
Precedence			Type of service			

Precedence : it is a **3 bit field which treats high priority packets** as more important than other packets.

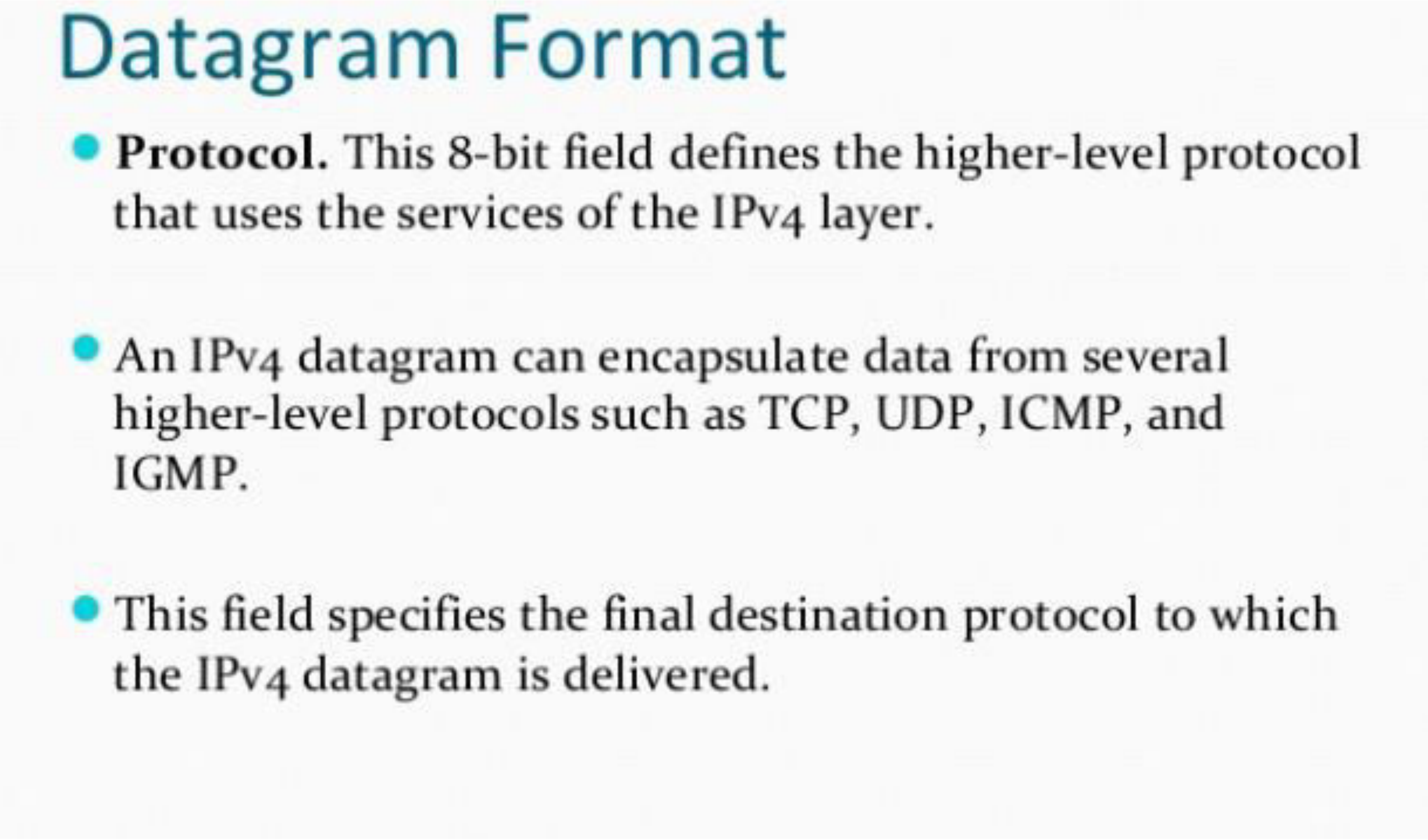
Type of Service: The last bit of Type of Service (bit 7) was defined as "Must Be Zero".

The TOS field specifies datagram's priority and request a route for low-delay, high-throughput, or highly-reliable service



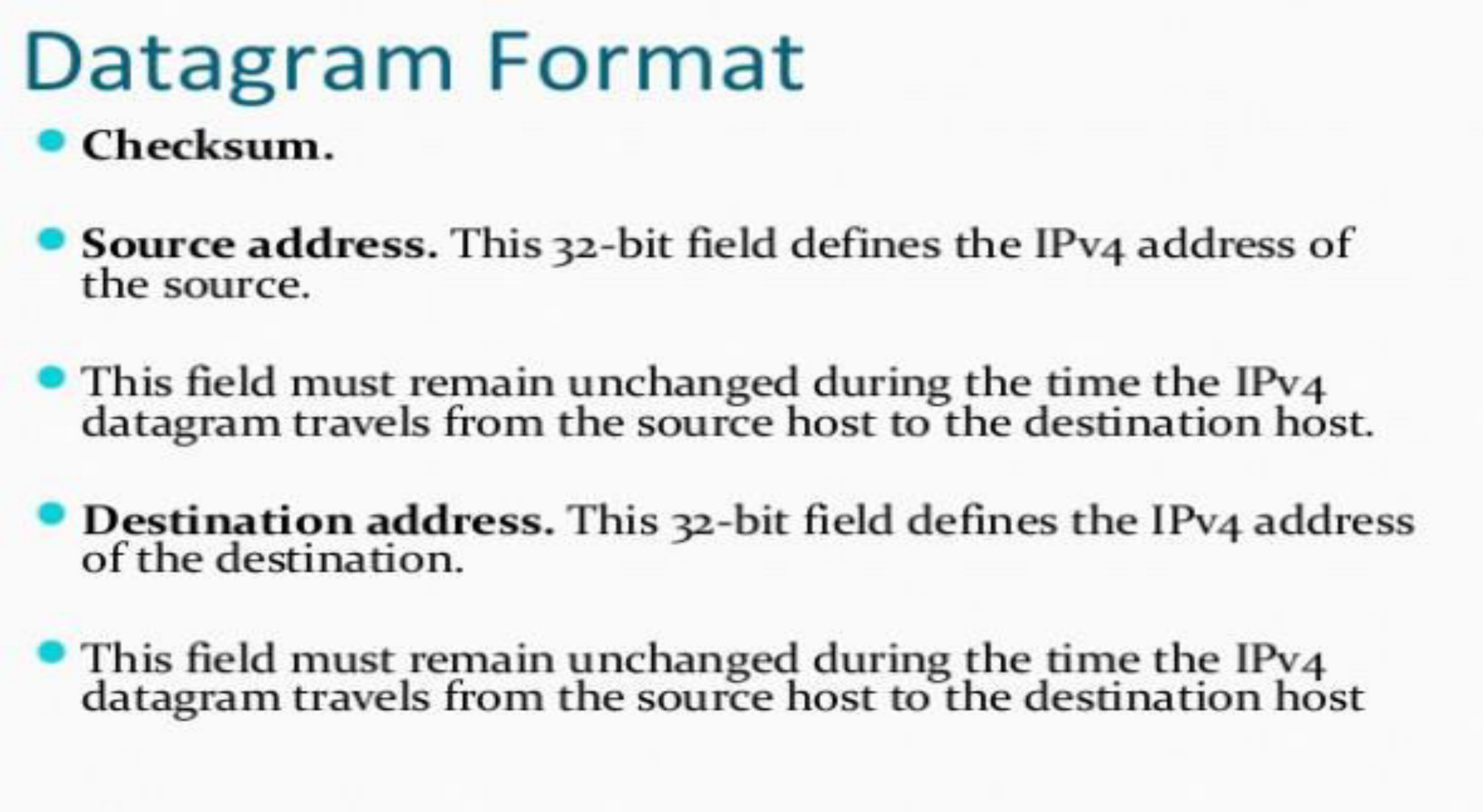
Datagram Format

- **Identification.** This field is used in fragmentation.
- **Flags.** This field is used in fragmentation.
- **Fragmentation offset**. This field is used in fragmentation.
- **Time to live.** A datagram has a limited lifetime in its travel through an internet.
- This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zero



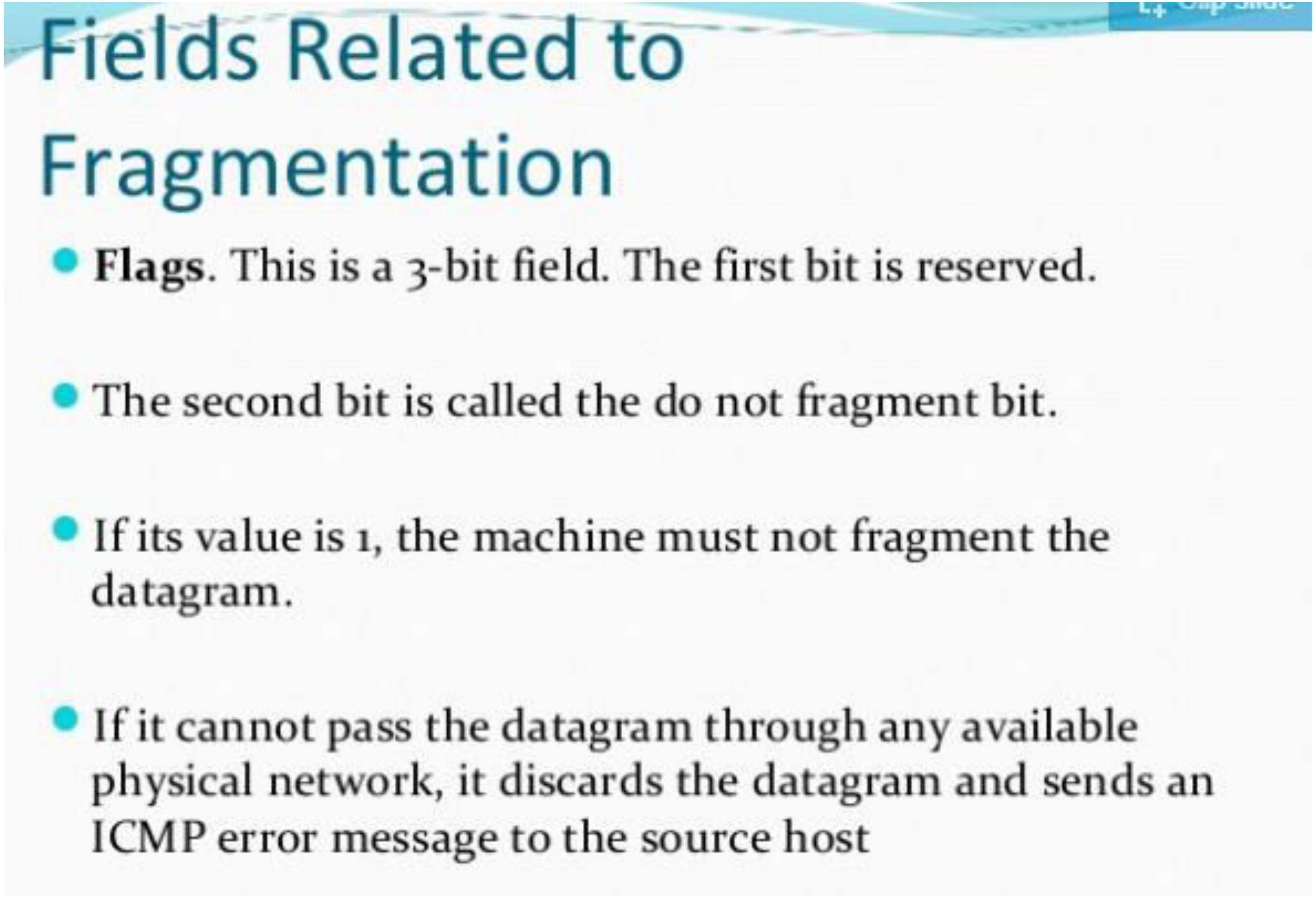
Datagram Format

- **Protocol.** This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer.
- An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP.
- This field specifies the final destination protocol to which the IPv4 datagram is delivered.



Datagram Format

- **Checksum.**
- **Source address.** This 32-bit field defines the IPv4 address of the source.
- This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
- **Destination address.** This 32-bit field defines the IPv4 address of the destination.
- This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host



Fields Related to Fragmentation

- **Flags**. This is a 3-bit field. The first bit is reserved.

- The second bit is called the do not fragment bit.

- If its value is 1, the machine must not fragment the datagram.

- If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host

Q 5 b) Explain in brief about IPv6

- Large address space.
- No more need of NAT
- Simple header
- Doesn't include a checksum in the header.
- No more broadcast
- Auto-configuration
- Fast forwarding/Routing
- IPSec
- Smooth transition.
- Enhanced priority support
- Anycast support
- Mobility
- And lot more.

Q 5 a) Applications of IOT

a) Smart Home / Smart Office

- Sensors controlling appliances and electrical devices in the house.
- Better lighting and heating in office buildings.
- **Like motion sensor,** Passive **Infrared** (PIR.

b) Biomedical / Medical

- Health Monitors like Glucose,Heart rate, Cancer detection
- Chronic Diseases like Artificial retina, Cochlear implants, Hospital Sensors
  - Monitor vital signs ,Record anomalies
  - Main sensors used are Photo Optic Sensors. Piezo Film Sensors. Pressure Sensors. Position Sensors. *Temperature* Sensors.

c) Military

Remote deployment of sensors for tactical monitoring of enemy troop movements.

Main sensors used are **active sensors**, **smart sensors**, intelligent sensors, **camera** sensors, IR sensors,

d) Industrial & Commercial

- Numerous industrial and commercial applications:
  - Agricultural Crop Conditions , Inventory Tracking, In-Process Parts Tracking, Automated Problem ,Reporting, RFID – Theft Deterrent and Customer Tracing, Plant Equipment Maintenance Monitoring
  - Main sensors used are *Ultrasonic sensors*, *Position sensors*, *Photoelectric sensors*,

e) Wearables

Wearables have experienced a explosive demand in markets all over the world. Companies like Google, Samsung have invested heavily in building such devices. Wearable devices are installed with sensors and softwares which collect data and information about the users.