1. List and explain the advantages of Internet protocol.

OR

Write a note on business case for IP. (VTU July 2019)

**Answer:** Key Advantages of Internet Protocol are

1. **Open and standards-based:** Operational technologies have often been delivered as turnkey features by vendors who may have optimized the communications through closed and proprietary networking solutions. The Internet of Things creates a new paradigm in which devices, applications, and users can leverage a large set of devices and functionalities while guaranteeing interchange ability and interoperability, security, and management.

2. **Versatile:** A large spectrum of access technologies is available to offer connectivity of "things" in the last mile. Additional protocols and technologies are also used to transport IoT data through backhaul links and in the data centre. IP architecture is well equipped to cope with any type of physical and data link layers.

3. **Ubiquitous:** All recent operating system releases, from general-purpose computers and servers to lightweight embedded systems have an integrated dual (IPv4 and IPv6) IP stack that gets enhanced over time. In addition, IoT application protocols in many industrial OT solutions have been updated in recent years to run over IP.

4. **Scalable:** As the common protocol of the Internet, IP has been massively deployed and tested for robust scalability. Millions of private and public IP infrastructure nodes have been operational for years, offering strong foundations for those not familiar with IP network management.

5. **Manageable and highly secure:** Communications infrastructure requires appropriate management and security capabilities for proper operations. Well-known network and security management tools are easily leveraged with an IP network layer.

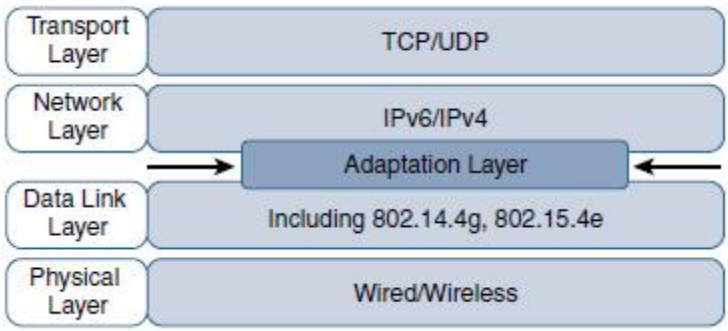
6. **Stable and resilient:** IP has a large and well-established knowledge base and, more importantly, it has been used for years in critical infrastructures, such as financial and defence networks. In addition, IP has been deployed for critical services, such as voice and video, which have already transitioned from closed environments to open IP standards. Finally, its stability and resiliency benefit from the large ecosystem of IT professionals who can help design, deploy, and operate IP-based solutions.

7. **Consumers' market adoption:** When developing IoT solutions and products targeting the consumer market, vendors know that consumers' access to applications and devices will occur predominantly over broadband and mobile wireless infrastructure.

8. **The innovation factor:** IP is the underlying protocol for applications ranging from file transfer and e-mail to the World Wide Web, e-commerce, social networking, and mobility. Innovations in IoT can also leverage an IP underpinning.

2. Illustrate with a neat block diagram, How to optimize IP for IOT using adaptation Layer.

**Answer:**



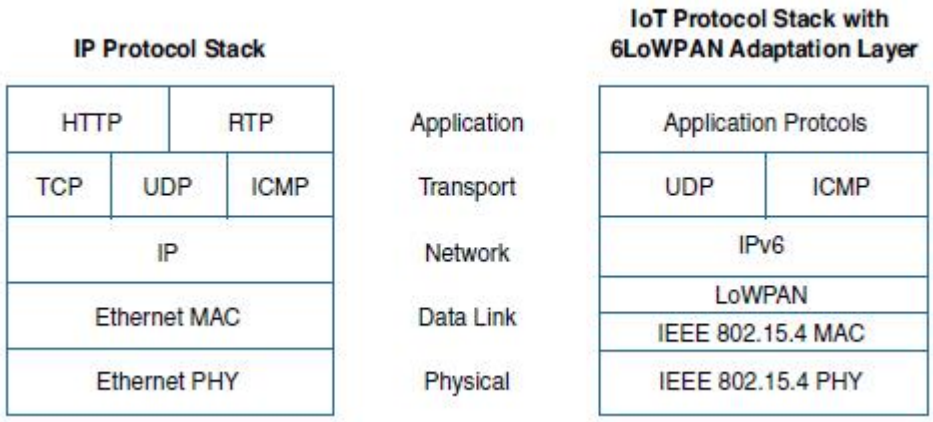

*Optimizing IP for IoT Using an Adaptation Layer*

1. In the IP architecture, the transport of IP packets over any given Layer 1 (PHY) and Layer 2 (MAC) protocol must be defined and documented. The model for packaging IP into lower-layer protocols is often referred to as an adaptation layer.

2. IP adaptation layers are typically defined by an IETF working group and released as a Request for Comments (RFC).

3. Adaptation layer designed for IoT may include some optimizations to deal with constrained nodes and networks.

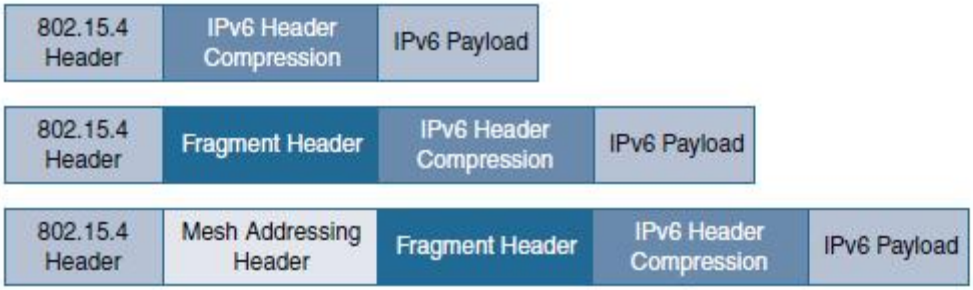
4. The main examples of adaptation layers optimized for constrained nodes or "things" are the ones under the 6LoWPAN working group and its successor, the 6Lo working group.




*Comparison of an IoT Protocol Stack Utilizing 6LoWPAN and an IP Protocol Stack*

## 3. Describe with a neat diagrams the header stacks of 6LoWPAN

**Answer:**




*6LoWPAN Header Stacks*

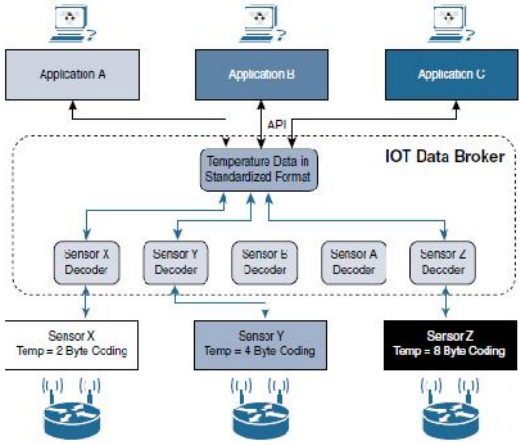
**Header Compression:** IPv6 header compression for 6LoWPAN was defined initially in RFC 4944 and subsequently updated by RFC 6282. This capability shrinks the size of IPv6's 40-byte headers and User Datagram Protocol's (UDP's) 8-byte headers down as low as 6 bytes combined in some cases

**Fragmentation:** The maximum transmission unit (MTU) for an IPv6 network must be at least 1280 bytes. The term *MTU* defines the size of the largest protocol data unit that can be passed.

**Mesh Addressing:** The purpose of the 6LoWPAN mesh addressing function is to forward packets over multiple hops. Three fields are defined for this header: Hop Limit, Source Address, and Destination Address. Analogous to the IPv6 hop limit field, the hop limit for mesh addressing also provides an upper limit on how many times the frame can be forwarded. Each hop decrements this value by 1 as it is forwarded. Once the value hits 0, it is dropped and no longer forwarded.

4. Demonstrate IOT Data broker with respect to application Layer protocol not present

Answer:




IoT Data Broker

a. Devices defined as class 0 send or receive only a few bytes of data.

b. For such as processing capability, power constraints, and cost, these devices do not implement a fully structured network protocol stack, such as IP, TCP, or UDP, or even an application layer protocol.

c. Class 0 devices are usually simple smart objects that are severely constrained. Implementing a robust protocol stack is usually not useful and sometimes not even possible with the limited available resources.

d. The solution to this problem is to use an IoT data broker.

e. Sensors X, Y, and Z are all temperature sensors, but their output is encoded differently. The IoT data broker understands the different formats in which the temperature is encoded and is therefore able to decode this data into a common, standardized format. Applications A, B, and C can access this temperature data without having to deal with decoding multiple temperature data formats.

## 5. Discuss tunnelling Legacy SCADA over IP network

Answer:

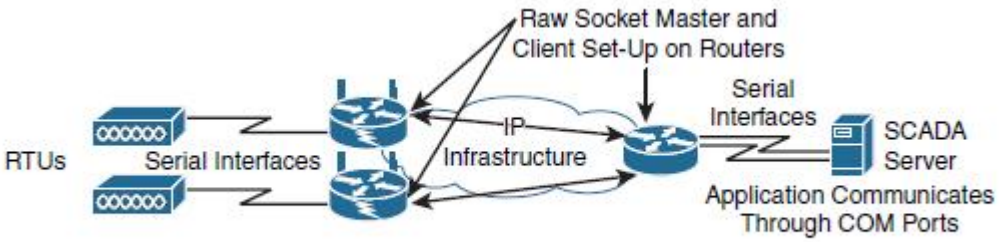
a. SCADA systems collect sensor data and telemetry from remote devices, while also providing the ability to control them.

b. SCADA systems allow global, real-time, data-driven decisions to be made about how to improve business processes.

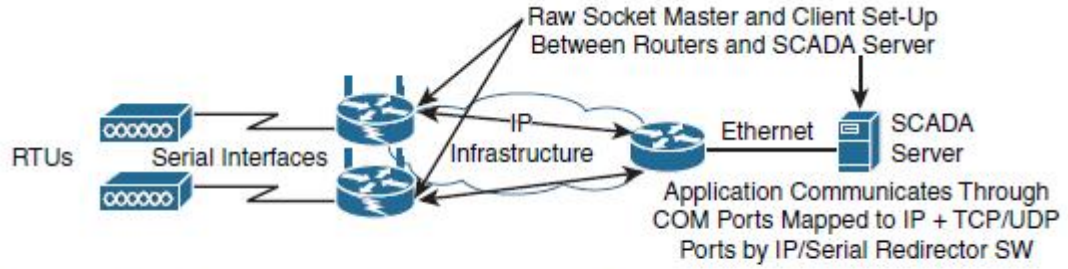
c. SCADA is mainly concentrated in the utilities and manufacturing/industrial verticals. Deployments of legacy industrial protocols like SCADA, in modern IP networks call for flexibility when integrating several generations of devices or operations that are tied to various releases and versions of application servers.

d. Transport of the original serial protocol over IP can be achieved either by **tunnelling** using raw sockets over TCP or UDP or by installing an intermediate device that performs protocol translation between the serial protocol version and its IP implementation.

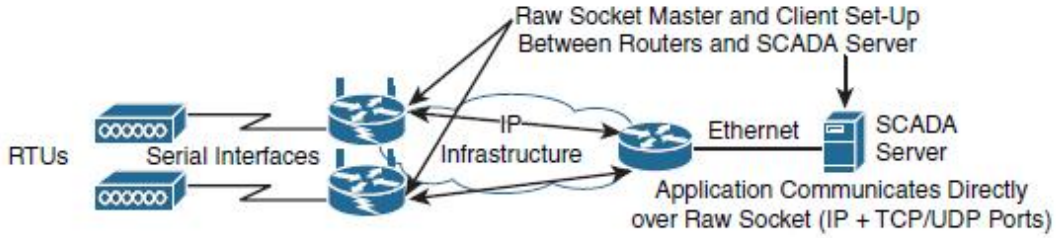
e. A raw socket connection simply denotes that the serial data is being packaged directly into a TCP or UDP transport.




**Scenario A: Raw Socket between Routers – no change on SCADA server**




**Scenario B: Raw Socket between Router and SCADA Server – no SCADA application change on server but IP/Serial Redirector software and Ethernet interface to be added**




**Scenario C: Raw Socket between Router and SCADA Server – SCADA application knows how to directly communicate over a Raw Socket and Ethernet interface**

f. In all the scenarios the routers connect via serial interfaces to the remote terminal units (RTUs), which are often associated with SCADA networks.

g. An RTU is a multipurpose device used to monitor and control various systems, applications, and devices managing automation.

h. Opposite the RTUs in each Figure 6-3 scenario is a SCADA server, or master, that varies its connection type.

6. Compare CoAP and MQTT with some factors.

Answer:

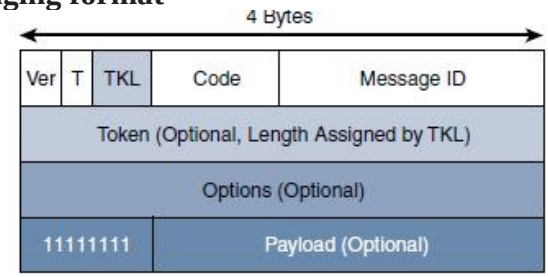
| Factor | CoAP | MQTT |
|---|---|---|
| Main transport protocol | UDP | TCP |
| Typical messaging | Request/response | Publish/subscribe |
| Effectiveness in LLNs | Excellent | Low/fair (Implementations pairing UDP with MQTT are better for LLNs.) |
| Security | DTLS | SSL/TLS |
| Communication model | One-to-one | many-to-many |
| Strengths | Lightweight and fast, with low overhead, and suitable for constrained networks; uses a RESTful model that is easy to code to; easy to parse and process for constrained devices; support for multicasting; asynchronous and synchronous messages | TCP and multiple QoS options provide robust communications; simple management and scalability using a broker architecture |
| Weaknesses | Not as reliable as TCP-based MQTT, so the application must ensure reliability. | Higher overhead for constrained devices and networks; TCP connections can drain low-power devices; no multicasting support |

# 7. Explain the following with respect to CoAP

i) Messaging format
ii) CoAP Communications in IoT Infrastructures
iii) CoAP Reliable Transmission

## Answer:

### i) CoAP Messaging format



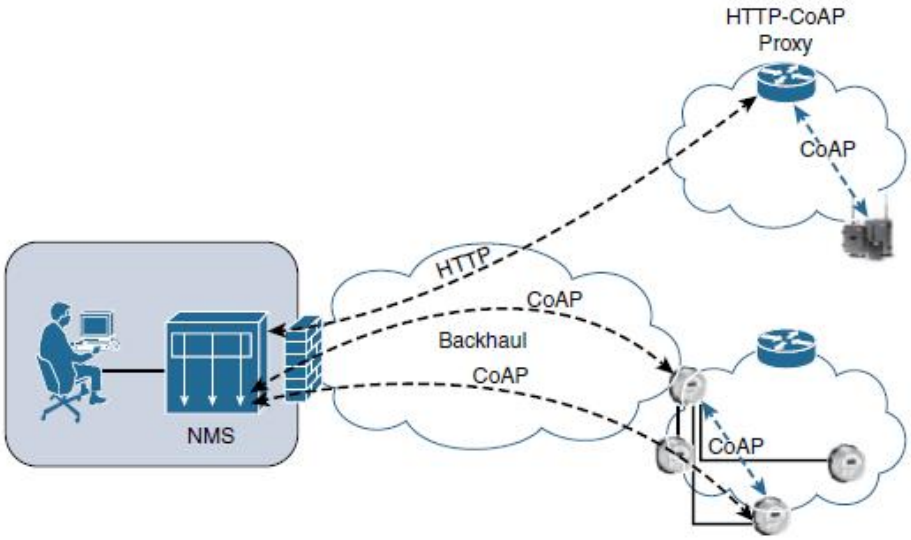

CoAP Message Format

a. The CoAP messaging model is primarily designed to facilitate the exchange of messages over UDP between endpoints, including the secure transport protocol Datagram Transport Layer Security (DTLS).

b. CoAP message is composed of a short fixed-length Header field (4 bytes), a variable-length but mandatory Token field (0–8 bytes), Options fields if necessary, and the Payload field. CoAP message format, delivers low overhead while decreasing parsing complexity.

| CoAP Message Field | Description |
|---|---|
| Ver (Version) | Identifies the CoAP version. |
| T (Type) | Defines one of the following four message types: Confirmable (CON), Non-confirmable (NON), Acknowledgement (ACK), or Reset (RST). |
| TKL (Token Length) | Specifies the size (0–8 Bytes) of the Token field. |
| Code | Indicates the request method for a request message and a response code for a response message. |
| Message ID | Detects message duplication and used to match ACK and RST message types to Con and NON message types. |
| Token | With a length specified by TKL, correlates requests and responses. |
| Options | Specifies option number, length, and option value. Capabilities provided by the Options field include specifying the target resource of a request and proxy functions. |
| Payload | Carries the CoAP application data. This field is optional. The purpose of this byte is to delineate the end of the Options field and the beginning of Payload. |

**ii) CoAP Communications in IoT Infrastructures:**




CoAP Communications in IoT Infrastructures

a. CoAP communications across an IoT infrastructure can take various paths.

b. Connections can be between devices located on the same or different constrained networks or between devices and generic Internet or cloud servers, all operating over IP.

c. Both HTTP and CoAP are IP-based protocols, the proxy function can be located practically anywhere in the network, not necessarily at the border between constrained and non-constrained networks.

d. CoAP is based on the REST architecture, but with a "thing" acting as both the client and the server. Through the exchange of asynchronous messages, a client requests an action via a method code on a server resource. A uniform resource identifier (URI) localized on the server identifies this resource. The server responds with a response code that may include a resource representation. The CoAP request/response semantics include the methods GET, POST, PUT, and DELETE.

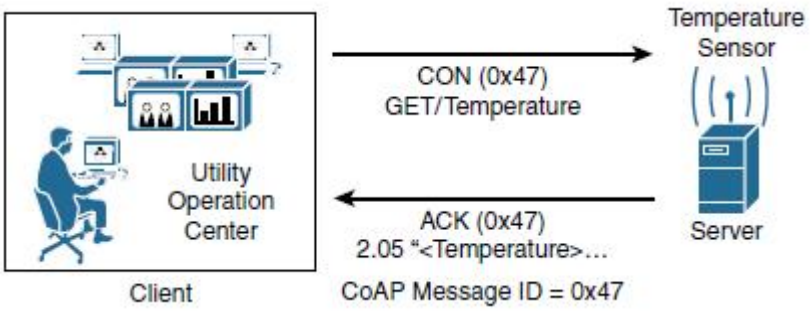
**iii) CoAP Reliable Transmission:**

a. While running over UDP, CoAP offers a reliable transmission of messages when a CoAP header is marked as "confirmable."

b. CoAP supports basic congestion control with a default time-out, simple stop and wait retransmission with exponential back-off mechanism, and detection of duplicate messages through a message ID.

c. If a request or response is tagged as confirmable, the recipient must explicitly either acknowledge or reject the message, using the same message ID.

d. If a recipient can't process a non-confirmable message, a reset message is sent.




CoAP Reliable Transmission Example

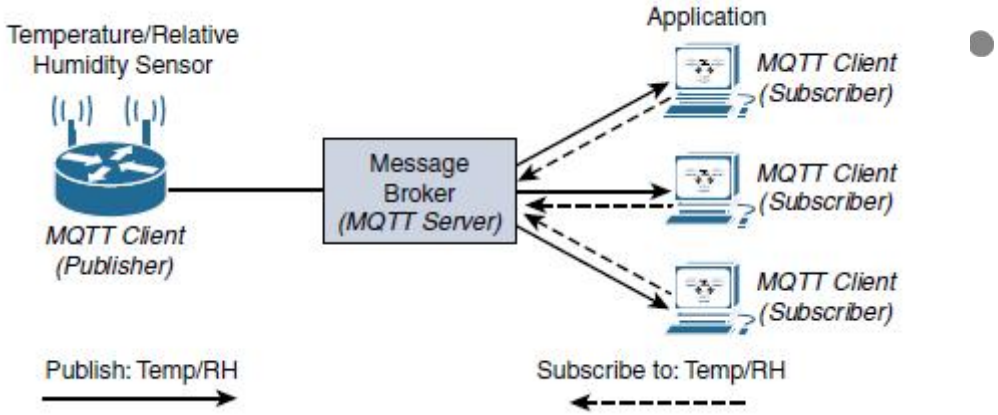
e. The example shows a utility operations center on the left, acting as the CoAP client, with the CoAP server being a temperature sensor on the right of the figure. The communication between the client and server uses a CoAP message ID of 0x47. The CoAP Message ID ensures reliability and is used to detect duplicate messages.

f. The client sends a GET message to get the temperature from the sensor. Notice that the 0x47 message ID is present for this GET message and that the message is also marked with CON. A CON, or confirmable, marking in a CoAP message means the message will be retransmitted until the recipient sends an acknowledgement (or ACK) with the same message ID.

g. The temperature sensor does reply with an ACK message referencing the correct message ID of 0x47. In addition, this ACK message piggybacks a successful response to the GET request itself. This is indicated by the 2.05 response code followed by the requested data.

8. Message Queuing Telemetry Transport (MQTT) with respect to following:
i) MQTT Publish/Subscribe Framework
ii) MQTT Message Format
iii) MQTT QoS
Answer:

**i) MQTT Publish/Subscribe Framework:**




MQTT Publish/Subscribe Framework

1. An MQTT client can act as a publisher to send data (or resource information) to an MQTT server acting as an MQTT message broker.

2. The MQTT server (or message broker) accepts the network connection along with application message from the publishers.

3. The MQTT server also handles the subscription and unsubscribes process and pushes the application data to MQTT clients acting as subscribers.

4. Clients can subscribe to all data (using a wildcard character) or specific data from the information tree of a publisher.

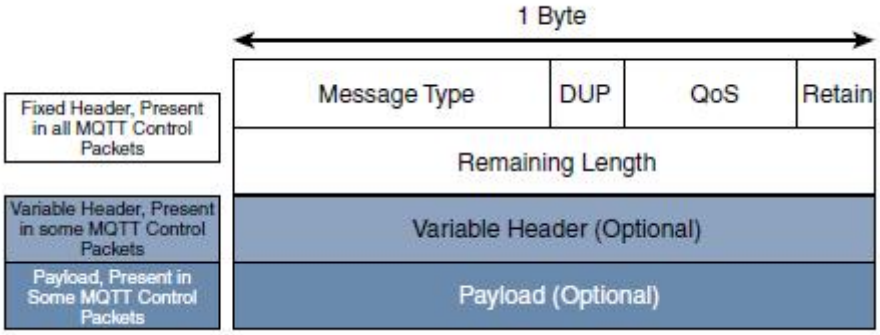
5. The presence of a message broker in MQTT decouples the data transmission between clients acting as publishers and subscribers.

6. A benefit of having this decoupling is that the MQTT message broker ensures that information can be buffered and cached in case of network failures. Publishers and subscribers do not have to be online at the same time.

7. MQTT control packets run over a TCP transport using port 1883. TCP ensures an ordered, lossless stream of bytes between the MQTT client and the MQTT server.

8. MQTT is a lightweight protocol because each control packet consists of a 2-byte fixed header with optional variable header fields and optional payload. Control packet can contain a payload up to 256 MB.

### ii) MQTT Message Format:




MQTT Message Format

1. MQTT contains a smaller header of 2 bytes compared to 4 bytes for CoAP. The first MQTT field in the header is Message Type, which identifies the kind of MQTT packet within a message. Fourteen different types of control packets are specified in MQTT .Each of them has a unique value that is coded into the Message Type field. The values 0 and 15 are reserved.

| Message Type | Value | Flow | Description |
|---|---|---|---|
| CONNECT | 1 | Client to server | Request to connect |
| CONNACK | 2 | Server to client | Connect acknowledgement |
| PUBLISH | 3 | Client to server<br>Server to client | Publish message |
| PUBACK | 4 | Client to server<br>Server to client | Publish acknowledgement |
| PUBREC | 5 | Client to server<br>Server to client | Publish received |
| PUBREL | 6 | Client to server<br>Server to client | Publish release |
| PUBCOMP | 7 | Client to server<br>Server to client | Publish complete |
| SUBSCRIBE | 8 | Client to server | Subscribe request |
| SUBACK | 9 | Server to client | Subscribe acknowledgement |
| UNSUBSCRIBE | 10 | Client to server | Unsubscribe request |
| UNSUBACK | 11 | Server to client | Unsubscribe acknowledgement |
| PINGREQ | 12 | Client to server | Ping request |
| PINGRESP | 13 | Server to client | Ping response |
| DISCONNECT | 14 | Client to server | Client disconnecting |

2. The next field in the MQTT header is DUP (Duplication Flag). This flag, when set, allows the client to notate that the packet has been sent previously, but an acknowledgement was not received.

3. The QoS header field allows for the selection of three different QoS levels.

4. Retain flag notifies the server to hold onto the message data. This allows new subscribers to instantly receive the last known value without having to wait for the next update from the publisher.

5. Remaining Length field specifies the number of bytes in the MQTT packet following this field.

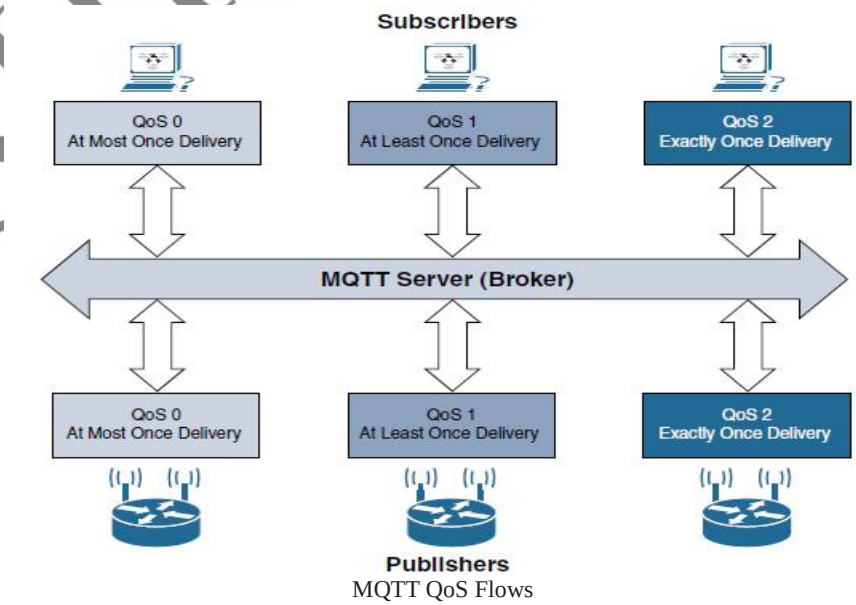
## iii) MQTT QoS

There are the three levels of MQTT QoS:

**QoS 0:** This is a best-effort and unacknowledged data service referred to as "at most once" delivery. The publisher sends its message one time to a server, which transmits it once to the subscribers. No response is sent by the receiver, and no retry is performed by the sender. The message arrives at the receiver either once or not at all.

**QoS 1:** This QoS level ensures that the message delivery between the publisher and server and then between the server and subscribers occurs at least once. In PUBLISH and PUBACK packets, a packet identifier is included in the variable header. If the message is not acknowledged by a PUBACK packet, it is sent again. This level guarantees "at least once" delivery.

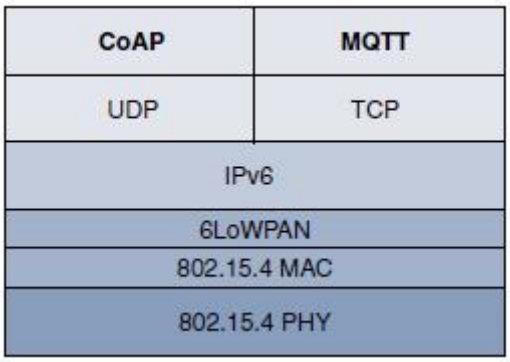
**QoS 2:** This is the highest QoS level, used when neither loss nor duplication of messages is acceptable. There is an increased overhead associated with this QoS level because each packet contains an optional variable header with a packet identifier. Confirming the receipt of a PUBLISH message requires a two-step acknowledgement
process. The first step is done through the PUBLISH/PUBREC packet pair, and the second is achieved with the PUBREL/PUBCOMP packet pair. This level provides a "guaranteed service" known as "exactly once" delivery, with no consideration for the number of retries as long as the message is delivered once.



MQTT QoS Flows

9. Discuss Application Protocols for IoT.

Answer:




High-Level IoT Protocol Stack for CoAP and MQTT

1. When considering constrained networks and/or a large-scale deployment of constrained nodes, verbose web-based and data model protocols may be too heavy for IoT applications.

2. IoT industry is working on new lightweight protocols that are better suited to large numbers of constrained nodes and networks. Two of the most popular protocols are CoAP and MQTT.

**Continue with the explanation of CoAP and MQTT briefly with the contents of question number 7 and 8**

10. List and explain the main industry organizations working on profile definitions and certifications for IoT constrained nodes and networks.

Answer:

1. **Internet Protocol for Smart Objects (IPSO) Alliance:** The alliance initially focused on promoting IP as the premier solution for smart objects communications. IPSO Alliance organises interoperability tests between alliance members to validate that IP for smart objects can work together and properly implement industry standards. IPSO documents the use of IP-based technologies for various IoT use cases and participates in educating the industry.

2. **Wi-SUN Alliance:** Defines a communication profile that applies to specific physical and data link layer protocols. Wi-SUN's main focus is on the IEEE 802.15.4g protocol and its support for multiservice and secure IPv6 communications with applications running over the UDP transport layer. The Wi-SUN field area network (FAN) profile enables smart utility networks to provide resilient, secure, and cost-effective connectivity.

3. **Thread:** This group has defined an IPv6-based wireless profile that provides the best way to connect more than 250 devices into a low-power, wireless mesh network. The wireless technology used by Thread is IEEE 802.15.4

4. **IPv6 Ready Logo:** Once IPv6 implementations became widely available, the need for interoperability and certification led to the creation of the IPv6 Ready Logo program. The IPv6 Ready Logo program has established conformance and interoperability testing programs with the intent of increasing user confidence when implementing IPv6.

11. Discuss the various methods used in IOT application transport.

Answer:

Categories of IoT application protocols and their transport methods are

1. **Application layer protocol not present:** The data payload is directly transported on top of the lower layers. No application layer protocol is used.

2. **Supervisory control and data acquisition (SCADA):** SCADA is one of the most common industrial protocols in the world, but it was developed long before the days of IP, and it has been adapted for IP networks.

3. **Generic web-based protocols:** Generic protocols, such as Ethernet, Wi-Fi, and 4G/ LTE, are found on many consumer- and enterprise-class IoT devices that communicate over non-constrained networks.

4. **IoT application layer protocols:** IoT application layer protocols are devised to run on constrained nodes with a small compute footprint and are well adapted to the network bandwidth constraints on cellular or satellite links or constrained 6LoWPAN networks. Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP),

**Continue with brief explanation of each category using the contents of question 4, 5, 6, 7,8,13**

## 12. Discuss the need for optimization of IP in IOT (VTU July 2019)

## Answer:

Designers should deal with the limits at the device and network levels that IoT often imposes. Optimizations are needed at various layers of the IP stack to handle the restrictions that are present in IoT networks. Reason for optimization of IP is as follows:

1. **Constrained Nodes:** different classes of devices coexist. Depending on its functions in a network,"thing" architecture may or may not offer similar characteristics. IoT node may be required to communicate through an unreliable path. Even if a full IP stack is available on the node, this causes problems such as limited or unpredictable throughput and low convergence when a topology change occurs. Power consumption requirements on battery-powered nodes impact communication intervals. To help extend battery life, you could enable a "low-power" mode instead of one that is "always on." Another option is "always off," which means communications are enabled only when needed to send data. IoT constrained nodes can be classified as follows:

   - **Devices that are very constrained in resources, may communicate infrequently to transmit a few bytes, and may have limited security and management capabilities:** This drives the need for the IP adaptation model, where nodes communicate through gateways and proxies.

   - **Devices with enough power and capacities to implement a stripped-down IP stack or non-IP stack:** In this case, you may implement either an optimized IP stack and directly communicate with application servers (adoption model) or go for an IP or non-IP stack and communicate through gateways and proxies.

   - **Devices that are similar to generic PCs in terms of computing and power resources but have constrained networking capacities, such as bandwidth:** These nodes usually implement a full IP stack (adoption model), but network design and application behaviours must cope with the bandwidth constraints.

2. **Constrained Networks:** high-speed connections are not usable by some IoT devices in the last mile. The reasons include the implementation of technologies with low bandwidth, limited distance and bandwidth due to regulated transmit power, and lack of or limited network services. Constrained networks are limited by low-power, low-bandwidth links (wireless and wired). They operate between a few kbps and a few hundred kbps and may utilize a star, mesh, or combined network topologies, ensuring proper operations. Packet delivery rate (PDR) may oscillate between low and high percentages. Large bursts of unpredictable errors and even loss of connectivity at times may occur. These behaviours can be observed on both wireless and narrowband power-line communication links, where packet delivery variation may fluctuate greatly during the course of a day. Link layer environments create other challenges in terms of latency and control plane reactivity.

3. **IP Versions:** There is a transitioning of Internet from IP version 4 to IP version 6. The main driving force has been the lack of address space in IPv4 as the Internet has

grown. IPv6 has a much larger range of addresses that should not be exhausted for the foreseeable future. The main factors applicable to IPv4 and IPv6 support in an IoT solution are:

- **Application Protocol:** IoT devices implementing Ethernet or Wi-Fi interfaces can communicate over both IPv4 and IPv6, but the application protocol may dictate the choice of the IP version.

- **Cellular Provider and Technology:** IoT devices with cellular modems are dependent on the generation of the cellular technology as well as the data services offered by the provider.

- **Serial Communications:** Many legacy devices in certain industries, such as manufacturing and utilities, communicate through serial lines. Encapsulation of serial protocols over IP leverages mechanisms such as raw socket TCP or UDP. While raw socket sessions can run over both IPv4 and IPv6, current implementations are mostly available for IPv4 only.

- **IPv6 Adaptation Layer:** IPv6-only adaptation layers for some physical and data link layers for recently standardized IoT protocols support only IPv6. Device implementing a technology that requires an IPv6 adaptation layer must communicate over an IPv6-only subnet work.

13. Explain generic web based protocols.

Answer:

1. The level of familiarity with generic web-based protocols is high. Therefore, programmers with basic web programming skills can work on IoT applications, and this may lead to innovative ways to deliver and handle real-time IoT data.

2. The definition of constrained nodes and networks must be analyzed to select the most appropriate protocol. On non-constrained networks, such as Ethernet, Wi-Fi, or 3G/4G cellular, where bandwidth is not perceived as a potential issue, data payloads based on a verbose data model representation, including XML or JavaScript Object Notation (JSON), can be transported over HTTP/HTTPS or WebSocket. This allows implementers to develop their IoT applications in contexts similar to web applications.

3. Recent evolutions of embedded web server software with advanced features are now implemented with very little memory. This enables the use of embedded web services software on some constrained devices.

4. When considering web services implementation on an IoT device, the choice between supporting the client or server side of the connection must be carefully weighed. IoT devices that only push data to an application may need to implement web services on the client side. The HTTP client side only initiates connections and does not accept incoming ones.

5. IoT devices, such as a video surveillance camera, may have web services implemented on the server side. Because these devices often have limited resources, the number of incoming connections must be kept low.

6. Interactions between real-time communication tools powering collaborative applications, such as voice and video, instant messaging, chat rooms, and IoT devices, are also emerging. This is driving the need for simpler communication systems between people and IoT devices. One protocol that addresses this need is Extensible Messaging and Presence Protocol (XMPP).