

Explain IEEE 802.11ah Sectorization

Analyze the key advantages of the IP suite for the Internet of Things

Explain the Lora WAN Layers with a neat diagram

Explain the DNP3 protocol over IP with a diagram.

Explain DAG and DODAG with its comparison

List and explain CoAP message fields

Explain LoRaWAN security with a neat diagram

Explain MQTT publish/subscribe framework

Explain general MAC Frame Format for IEEE 1901.2 with a neat diagram

Analyze the business case of the IP suite for the Internet of Things.

Compare the IOT protocol stack utilizing 6LoWPAN and IP protocol stack with a neat diagram.

List and explain any 5 RPL routing metrics.

Explain CoAP message format with a neat diagram

Explain the differences between MQTT and CoAP

Explain MQTT message format with a neat diagram.



You

Explain IEEE 802.11ah Sectorization in IoT



ChatGPT

IEEE 802.11ah is a standard for Wireless LAN in the sub-1 GHz band, primarily designed for Internet of Things (IoT) applications. Sectorization in IEEE 802.11ah refers to dividing the coverage area into sectors to improve network performance and efficiency. This concept is particularly relevant in IoT scenarios where there are many devices spread across a large area.

Here's how sectorization works in IEEE 802.11ah for IoT:

1. **Coverage Area Division:** The coverage area is divided into smaller sectors. Each sector typically covers a specific geographic area or zone.
2. **Access Points (APs):** Multiple APs are deployed within the coverage area, with each AP responsible for serving one or more sectors.
3. **Beamforming and Directionality:** IEEE 802.11ah supports beamforming techniques, allowing APs to focus their signals towards specific sectors. This helps in reducing interference and improving signal strength and reliability for devices within those sectors.
4. **Channel Allocation:** Different sectors may operate on different channels or frequencies to minimize interference between adjacent sectors.
5. **Traffic Management:** By segmenting the coverage area into sectors, network traffic can be managed more effectively. For example, critical IoT devices can be assigned to dedicated sectors with optimized settings for latency-sensitive applications.
6. **Capacity Planning:** Sectorization enables better capacity planning as network resources can be allocated based on the number of devices and their traffic patterns within each sector.

Topology

- While IEEE 802.11ah is deployed as a star topology, it includes a simple hops relay operation to extend its range. This relay option is not capped, but the IEEE 802.11ah task group worked on the assumption of two hops.
- It allows one 802.11ah device to act as an intermediary and relay data to another.
- In some ways, this is similar to a mesh, and it is important to note that the clients and not the access point handle the relay function.
- This relay operation can be combined with a higher transmission rate or **modulation and coding scheme (MCS)**.

124

10-03-2023

- This means that a higher transmit rate is used by relay devices talking directly to the access point.
- The transmit rate reduces as we move further from the access point via relay clients. This ensures an efficient system that limits transmission speeds at the edge of the relays so that communications close to the AP are not negatively affected.
- Sectorization is a technique that involves partitioning the coverage area into several sectors to get reduced contention within a certain sector.
- This technique is useful for limiting collisions in cells that have many clients. This technique is also often necessary when the coverage area of 802.11ah access points is large, and interference from neighboring access points is problematic.

125

10-03-2023

- Sectorization uses an antenna array and beam-forming techniques to partition the cell coverage area.
- Figure 4.14 shows an example of 802.11ah sectorization.

Security

- No additional security has been identified for IEEE 802.11ah compared to other IEEE 802.11 specifications.
- These protocols include IEEE 802.15.4, IEEE 802.15.4e, and IEEE 1901.2a, and the security information for them is also applicable to IEEE 802.11ah.

126

10-03-2023

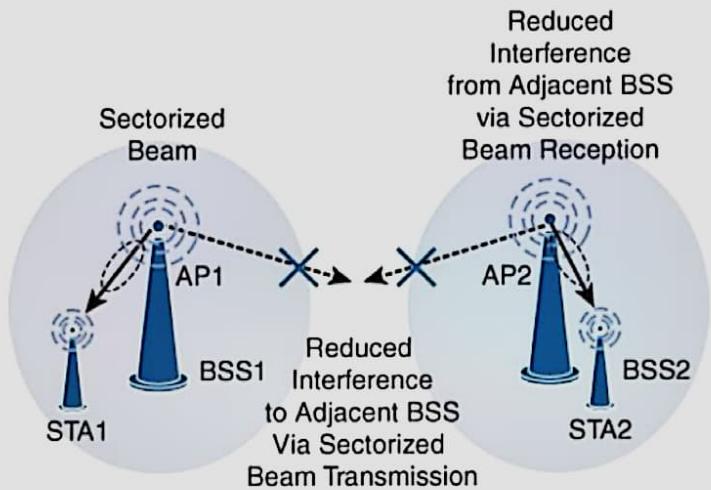


Figure 4.14: IEEE 802.11ah Sectorization

127

10-03-2023

Competitive Technologies

- Competitive technologies to IEEE 802.11ah are IEEE 802.15.4, IEEE 802.15.4e and 802.15.4g, DASH7 etc.

Conclusions

- The IEEE 802.11ah access technology is an ongoing effort of the IEEE 802.11 working group to define an “**industrial Wi-Fi**.”
- Currently, this standard is just at the beginning of its evolution, and it is not clear how the market will react to this new Wi-Fi standard.

128

10-03-2023

- This specification offers a longer range than traditional Wi-Fi technologies and provides good support for low-power devices that need to send smaller bursts of data at lower speeds.
- At the same time, it has the ability to scale to higher speeds as well.
- IEEE 802.11ah is quite different in terms of current products and the existing Wi-Fi technologies in the 2.4 GHz and 5 GHz frequency bands.
- To gain broad adoption and compete against similar technologies in this space, it will need an ecosystem of products and solutions that can be configured and deployed at a low cost.

129

10-03-2023



You

Analyze the key advantages of the IP suite for the Internet of Things



ChatGPT

The Internet Protocol (IP) suite, also known as the TCP/IP protocol suite, forms the backbone of modern networking, including the Internet of Things (IoT). Here are some key advantages of using the IP suite for IoT:

1. **Interoperability:** IP-based communication allows devices from different manufacturers and with different protocols to communicate seamlessly. This interoperability is crucial for the IoT, where numerous devices with varying capabilities need to work together in a cohesive ecosystem.
2. **Scalability:** The IP suite is highly scalable, capable of accommodating a vast number of devices and networks. This scalability is essential for IoT deployments, which often involve massive numbers of connected devices spread across wide geographic areas.
3. **Global Reach:** IP networks operate globally, providing connectivity across borders and regions. This global reach is advantageous for IoT applications that require connectivity and data exchange on a worldwide scale.
4. **Security:** The IP suite includes security protocols such as IPsec (IP Security) for encryption and authentication, helping to secure IoT communications and protect sensitive data transmitted between devices and servers.
5. **Routing and Addressing:** IP enables efficient routing of data packets across networks and supports unique IP addresses for each device. This capability facilitates data transmission and ensures that messages reach their intended destinations accurately.
6. **Flexibility:** IP is protocol-agnostic, meaning it can work with various higher-layer protocols, such as TCP, UDP, and ICMP. This flexibility allows for diverse communication patterns and application requirements within the IoT ecosystem.
7. **Management and Monitoring:** IP-based networks support management and monitoring functionalities, including tools like SNMP (Simple Network Management Protocol), which are essential for overseeing and controlling IoT devices and networks.
8. **Integration with Existing Infrastructure:** Many organizations already have IP-based networks in place for their IT infrastructure. Leveraging the IP suite for IoT integration allows for easier integration with existing systems, reducing deployment complexities and costs.
9. **Future-Proofing:** IP has evolved over decades and continues to evolve with advancements such as IPv6, which provides a vast address space to accommodate the growing number of IoT devices. This future-proofing ensures that IP-based IoT solutions can adapt to changing technological landscapes.

Overall, the IP suite offers a robust foundation for IoT deployments, providing interoperability, scalability, security, global connectivity, and flexibility to support diverse IoT applications and requirements.

The Key Advantages of Internet Protocol

- One of the main differences between traditional information technology (IT) and operational technology (OT) is the lifetime of the underlying technologies and products.
- One way to guarantee multi-year lifetimes is to define a layered architecture such as the 30-year-old IP architecture.
- IP has largely demonstrated its ability to integrate small and large evolutions.

3

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- The key advantages of the IP suite for the Internet of Things are as follows:

□ 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 interchangeability and interoperability, security, and management.
- This calls for implementation, validation, and deployment of open, standards-based solutions.

4

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- While many standards development organizations (SDOs) are working on Internet of Things definitions, frameworks, applications, and technologies, none are questioning the role of the Internet Engineering Task Force (IETF) as the foundation for specifying and optimizing the network and transport layers.
- The IETF is an open standards body that focuses on the development of the Internet Protocol suite and related Internet technologies and protocols.

□ Versatile

- A large spectrum of access technologies is available to offer connectivity of “things” in the last mile.

5

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- Additional protocols and technologies are also used to transport IoT data through backhaul links and in the data center.
- Even if physical and data link layers such as Ethernet, Wi-Fi, and cellular are widely adopted, the history of data communications demonstrates that no given wired or wireless technology fits all deployment criteria.
- Furthermore, communication technologies evolve at a pace faster than the expected 10- to 20-year lifetime of OT solutions.
- So, the layered IP architecture is well equipped to cope with any type of physical and data link layers.

6

Manoj T, Dept. of CSE, SMVITM

05-04-2019

□ Ubiquitous

- All recent operating system releases, from general-purpose computers and servers to lightweight embedded systems (TinyOS, Contiki, and so on), 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.
- While these updates have mostly consisted of IPv4 to this point, recent standardization efforts in several areas are adding IPv6.
- In fact, IP is the most pervasive protocol which is supported across the various IoT solutions and industry verticals.

7

Manoj T, Dept. of CSE, SMVITM

05-04-2019

□ 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.
- Of course, adding huge numbers of “things” to private and public infrastructures may require optimizations and design rules specific to the new devices.

□ Manageable and highly secure

- Communications infrastructure requires appropriate management and security capabilities for proper operations.

8

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- One of the benefits that comes from 30 years of operational IP networks is the well-understood network management and security protocols, mechanisms, and toolsets that are widely available.
- Adopting IP network management also brings an operational business application to OT.
- Well known network and security management tools are easily leveraged with an IP network layer.

□ Stable and Resilient

- IP has been around for 30 years, and it is clear that IP is a workable solution.

9

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- 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 defense 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.

□ Consumer's 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.

10

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- The main consumer devices range from smart phones to tablets and PCs. The common protocol that links IoT in the consumer space to these devices is IP.

□ The innovation factor

- The past two decades have largely established the adoption of IP as a factor for increased innovation.
- IP is the underlying protocol for applications ranging from file transfer and e-mail to the World Wide Web, e-commerce, social networking, mobility, and more.
- Even the recent computing evolution from PC to mobile and mainframes to cloud services are perfect demonstrations of the innovative ground enabled by IP.

11

Manoj T, Dept. of CSE, SMVITM

05-04-2019



You

Explain the LoRa WAN Layers in IoT



ChatGPT

LoRaWAN (Long Range Wide Area Network) is a wireless communication protocol specifically designed for low-power, wide-area networks (LPWANs) used in IoT applications. It operates on the sub-gigahertz bands and is known for its long-range capabilities and low power consumption. The LoRaWAN protocol stack consists of several layers, each serving specific functions in IoT communications. Here's an overview of the LoRaWAN layers:

1. **Physical Layer (PHY):**

- **RF Modulation:** LoRaWAN uses a proprietary spread spectrum modulation technique called Chirp Spread Spectrum (CSS). This modulation allows LoRaWAN to achieve long-range communication and robustness against interference.
- **Frequency Bands:** LoRaWAN operates in various frequency bands, including 433 MHz, 868 MHz, and 915 MHz, depending on regional regulations.

2. **Data Link Layer (DLC):**

- **LoRaWAN Protocol:** The LoRaWAN protocol defines how data is formatted, transmitted, and acknowledged between end-devices (sensors, actuators) and the network server.
- **MAC Layer:** The Medium Access Control (MAC) layer manages access to the shared radio channel, implements data rate adaptation, and handles acknowledgment and retransmission mechanisms.

3. **Network Layer:**

- **LoRaWAN Gateways:** Gateways receive LoRaWAN packets from end-devices and forward them to the network server over IP-based backhaul connections (Ethernet, Wi-Fi, cellular).
- **Addressing and Routing:** LoRaWAN uses unique 64-bit device addresses and supports adaptive data rate (ADR) to optimize communication parameters based on device proximity and network conditions.

4. **Application Layer:**

- **Device Management:** LoRaWAN supports over-the-air (OTA) updates, remote configuration, and device status monitoring through the application layer.
- **Payload Processing:** Application servers decode and process the payload data received from end-devices, performing application-specific logic and actions.

LoRaWAN

- In recent years, a new set of wireless technologies known as **Low-Power Wide-Area (LPWA)** has received a lot of attention from the industry and press.
- Particularly well adapted for long-range and battery-powered endpoints, LPWA technologies open new business opportunities to both services providers and enterprises considering IoT solutions.
- An example of an unlicensed-band LPWA technology, known as **LoRaWAN**

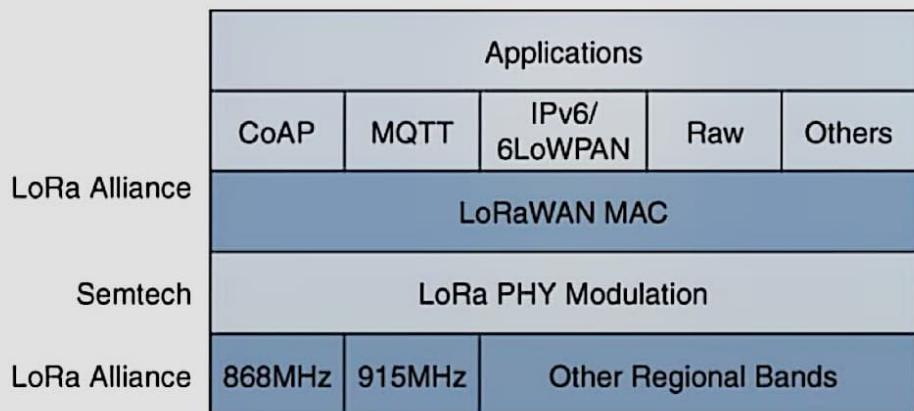


Figure 4.15: LoRa WAN Layers

133

10-03-2023

Physical Layer

- Semtech LoRa modulation is based on **chirp spread spectrum** modulation, which trades a lower data rate for receiver sensitivity to significantly increase the communication distance.
- In addition, it allows demodulation below the noise floor, offers robustness to noise and interference, and manages a single channel occupation by different spreading factors.
- This enables LoRa devices to receive on multiple channels in parallel.
- LoRaWAN 1.0.2 regional specifications describe the use of the main unlicensed sub-GHz frequency bands of 433 MHz, 779–787 MHz, 863–870 MHz, and 902–928 MHz,

134

10-03-2023

as well as regional profiles for a subset of the 902–928 MHz bandwidth.

- For ex : Australia utilizes 915–928 MHz frequency bands, while South Korea uses 920–923 MHz and Japan uses 920–928 MHz.
- The LoRa gateways is critical to understanding a LoRaWAN system.
- A LoRa gateway is deployed as the center hub of a star network architecture. It uses multiple transceivers and channels and can demodulate multiple channels at once or even demodulate multiple signals on the same channel simultaneously.
- The data rate in LoRaWAN varies depending on the frequency bands and **adaptive data rate (ADR)**.

135

10-03-2023

- ADR is an algorithm that manages the data rate and radio signal for each endpoint.
- The ADR algorithm ensures that packets are delivered at the best data rate possible and that network performance is both optimal and scalable.
- Endpoints close to the gateways with good signal values transmit with the highest data rate, which enables a shorter transmission time over the wireless network, and the lowest transmit power.
- An important feature of LoRa is its ability to handle various data rates via the spreading factor.

136

10-03-2023

- Devices with a **low spreading factor (SF)** achieve less distance in their communications but transmit at faster speeds, resulting in less airtime.
- A higher SF provides slower transmission rates but achieves a higher reliability at longer distances.
- Table 4.4 illustrates how LoRaWAN data rates can vary depending on the associated spreading factor for the two main frequency bands, 863–870 MHz and 902–928 MHz.
- We can notice in Table 4.4 the relationship between SF and data rate. For ex : at an SF value of 12 for 125 kHz of channel bandwidth, the data rate is 250 bps. However, when the SF is decreased to a value of 7, the data rate increases to 5470 bps.

137

10-03-2023

Configuration	863–870 MHz bps	902–928 MHz bps
LoRa: SF12/125 kHz	250	N/A
LoRa: SF11/125 kHz	440	N/A
LoRa: SF10/125 kHz	980	980
LoRa: SF9/125 kHz	1760	1760
LoRa: SF8/125 kHz	3125	3125
LoRa: SF7/125 kHz	5470	5470
LoRa: SF7/250 kHz	11,000	N/A
FSK: 50 kbps	50,000	N/A
LoRa: SF12/500 kHz	N/A	980
LoRa: SF11/500 kHz	N/A	1760
LoRa: SF10/500 kHz	N/A	3900
LoRa: SF9/500 kHz	N/A	7000
LoRa: SF8/500 kHz	N/A	12,500
LoRa: SF7/500 kHz	N/A	21,900

Table 4.4 : LoRaWAN Data Rate Example

138

10-03-2023



You

Explain the DNP3 protocol over IP with a diagram



ChatGPT

DNP3 (Distributed Network Protocol version 3) is a widely used communication protocol in the field of industrial automation and control systems (IACS). It is designed for reliable and efficient communication between various components in SCADA (Supervisory Control and Data Acquisition) systems. DNP3 can operate over different transport layers, including serial links and IP networks. When used over IP (Internet Protocol), it typically relies on TCP/IP for communication.

Here's an overview of DNP3 over IP along with a simplified diagram:

1. TCP/IP Stack:

- **Application Layer:** DNP3 Protocol
- **Transport Layer:** TCP (Transmission Control Protocol)
- **Internet Layer:** IP (Internet Protocol)
- **Network Access Layer:** Ethernet or other network medium

2. DNP3 Protocol:

- **Master Station:** Initiates communication and collects data from Remote Stations.
- **Remote Stations:** Field devices such as sensors, actuators, or other control equipment.

3. Communication Flow:

- **Initialization:** The Master Station establishes a TCP/IP connection with the Remote Station(s) over Ethernet or another IP-based network.
- **Request-Response:** The Master Station sends requests (polls) to Remote Stations to collect data or issue commands. Remote Stations respond with data or acknowledge commands.
- **Data Exchange:** Data such as sensor readings, status information, or control commands are exchanged between the Master Station and Remote Stations.
- **Error Handling:** TCP/IP provides reliable, error-checked communication, ensuring data integrity and robustness against network issues.

4. Security Considerations:

- DNP3 over IP can implement additional security measures such as encryption (e.g., SSL/TLS) and authentication to protect data and prevent unauthorized access.

Representative Use Case : Evolution of DNP3 Protocol using IP

- DNP3 is based on a master/slave relationship.
- The term **master** in this case refers to what is typically a powerful computer located in the control center of a utility.
- A **slave** is a remote device with computing resources found in a location such as a substation. DNP3 refers to slaves specifically as **outstations**.

27

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- Outstations monitor and collect data from devices that indicate their state, such as whether a circuit breaker is on or off, and take measurements, including voltage, current, temperature, and so on.
- This data is then transmitted to the master when it is requested, or events and alarms can be sent in an asynchronous manner.
- The master also issues control commands, such as to start a motor or reset a circuit breaker, and logs the incoming data.
- The IEEE 1815-2012 specification describes how the DNP3 protocol implementation must be adapted to run either over TCP (recommended) or UDP.
- This specification defines connection management between the DNP3 protocol and the IP layers, as shown in Figure 6.2

28

Manoj T, Dept. of CSE, SMVITM

17-04-2019

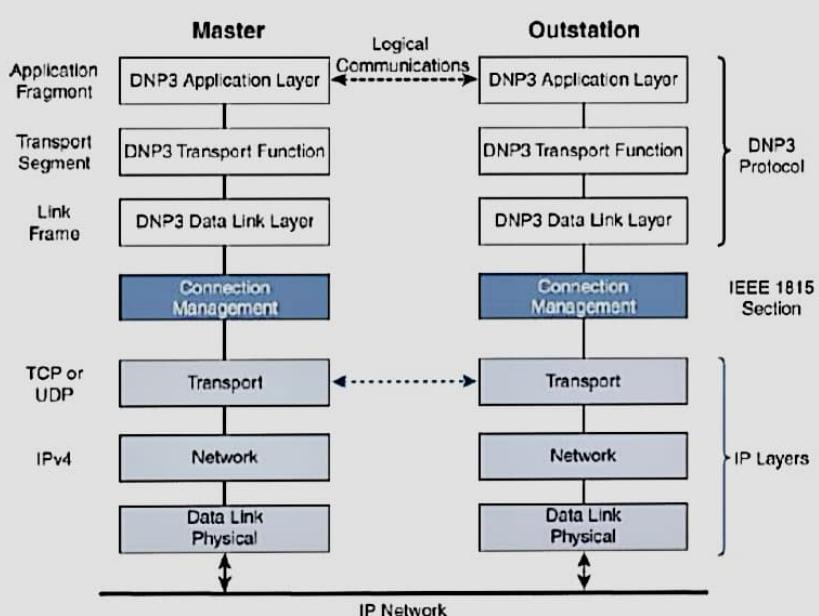


Figure 6.2 : Protocol Stack for Transporting Serial DNP3 SCADA over IP

29

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- Connection management links the DNP3 layers with the IP layers in addition to the configuration parameters and methods necessary for implementing the network connection.
- The IP layers appear transparent to the DNP3 layers as each piece of the protocol stack in one station logically communicates with the respective part in the other.
- This means that the DNP3 endpoints or devices are not aware of the underlying IP transport that is occurring.
- In Figure 6.2 the master side initiates connections by performing a TCP active open. The outstation listens for a connection request by performing a TCP passive open.

- Dual endpoint is defined as a process that can both listen for connection requests and perform an active open on the channel if required.
- Master stations may parse multiple DNP3 data link layer frames from a single UDP datagram, while DNP3 data link layer frames cannot span multiple UDP datagrams.
- Single or multiple connections to the master may get established while a TCP keepalive timer monitors the status of the connection.
- Keepalive messages are implemented as DNP3 data link layer status requests.
- If a response is not received to a keepalive message, the connection is deemed broken, and the appropriate action is taken.



You

Explain DAG and DODAG with its comparison



ChatGPT

DAG (Directed Acyclic Graph) and DODAG (Destination-Oriented Directed Acyclic Graph) are concepts used in the context of routing protocols for Low-Power and Lossy Networks (LLNs), which are common in IoT and sensor network environments. These concepts are particularly associated with protocols like RPL (Routing Protocol for Low-Power and Lossy Networks), which is designed for efficient routing in such constrained network scenarios.

1. **DAG (Directed Acyclic Graph):**

- **Definition:** A DAG is a directed graph with nodes connected by edges in a way that no cycles exist. In the context of RPL, a DAG represents the network topology where nodes are devices (e.g., IoT devices, sensors) and edges represent communication links between them.
- **Structure:** Nodes in a DAG can have different roles such as root, parent, and child nodes. The root node typically serves as the central coordinator or gateway for the network, while parent nodes are responsible for forwarding data from child nodes to the root.
- **Advantages:** DAG structures are efficient for routing in LLNs as they allow for optimized path selection, multipath routing, and adaptability to dynamic network conditions.

2. **DODAG (Destination-Oriented Directed Acyclic Graph):**

- **Definition:** A DODAG is a specific type of DAG used in RPL-based routing protocols. It is oriented towards a specific destination or root node, with edges directed towards that destination.
- **Structure:** In a DODAG, nodes maintain directed edges towards the root or a preferred parent node. This orientation helps in establishing efficient routes towards the root, optimizing energy consumption and latency.
- **Advantages:** DODAGs provide a structured and hierarchical routing framework suitable for LLNs. They enable effective route discovery, parent selection, and path optimization for data transmission towards a central coordinator or gateway.

Comparison:

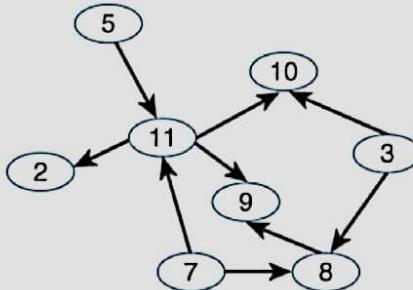
- **Purpose:**
 - DAG: Represents the overall network topology without a specific focus on a destination.
 - DODAG: Focuses on routing paths towards a specific destination (root node or gateway).
- **Orientation:**
 - DAG: Does not have a specific orientation as it represents the general connectivity of nodes in the network.
 - DODAG: Directed towards a root node or preferred parent node, facilitating efficient routing towards the destination.
- **Routing:**
 - DAG: Can be used for various routing purposes, including source-based routing or multipath routing.
 - DODAG: Optimizes routing paths towards the root or gateway, considering factors like link quality and energy efficiency.
- **Implementation:**
 - DAG: Can be implemented in various routing protocols and network structures.
 - DODAG: Specifically used in RPL-based routing protocols for LLNs, providing hierarchical and destination-oriented routing.

- RPL is based on the concept of a directed acyclic graph (DAG). A DAG is a directed graph where no cycles exist.
- This means that from any vertex or point in the graph, we cannot follow an edge or a line back to this same point.
- All of the edges are arranged in paths oriented toward and terminating at one or more root nodes.
- Figure 5.8 shows a basic DAG
- A basic RPL process involves building a destination-oriented directed acyclic graph (DODAG). A DODAG is a DAG rooted to one destination.
- In RPL, this destination occurs at a border router known as the DODAG root.

75

Manoj T, Dept. of CSE, SMVITM

05-04-2019

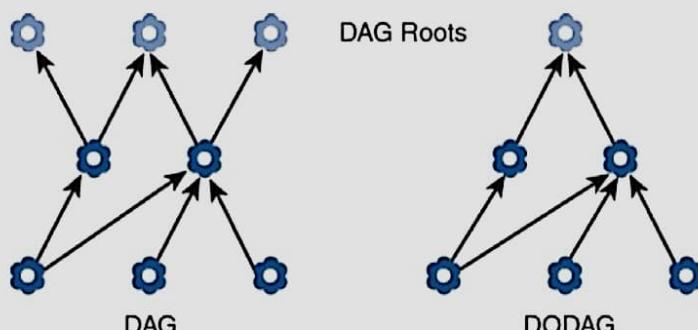
**Figure 5.8** : Example of a Directed Acyclic Graph(DAG)

- Figure 5.9 compares a DAG and a DODAG. You can see that that a DAG has multiple roots, whereas the DODAG has just one.
- In a DODAG, each node maintains up to three parents that provide a path to the root.

76

Manoj T, Dept. of CSE, SMVITM

05-04-2019

**Figure 5.9** : DAG and DODAG comparison

- In a DODAG, each node maintains up to three parents that provide a path to the root.
- Typically, one of these parents is the preferred parent, which means it is the preferred next hop for upward routes toward the root.

77

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- The routing graph created by the set of DODAG parents across all nodes defines the full set of upward routes.
- RPL protocol implementation should ensure that routes are loop free by disallowing nodes from selected DODAG parents that are positioned further away from the border router.
- Upward routes in RPL are discovered and configured using DAG Information Object (DIO) messages.
- Nodes listen to DIOs to handle changes in the topology that can affect routing. The information in DIO messages determines parents and the best path to the DODAG root.
- Nodes establish downward routes by advertising their parent set toward the DODAG root using a Destination Advertisement Object (DAO) message

78

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- DAO messages allow nodes to inform their parents of their presence and reachability to descendants.
- In the case of the non-storing mode of RPL, nodes sending DAO messages report their parent sets directly to the DODAG root (border router), and only the root stores the routing information.
- The root uses the information to then determine source routes needed for delivering IPv6 datagrams to individual nodes downstream in the mesh.
- For storing mode, each node keeps track of the routing information that is advertised in the DAO messages.
- While this is more power- and CPU-intensive for each node, the benefit is that packets can take shorter paths between destinations in the mesh.

79

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- RPL messages, such as DIO and DAO, run on top of IPv6. These messages exchange and advertise downstream and upstream routing information between a border router and the nodes under it.
- Figure 5.10 illustrates that the DAO and DIO messages move both up and down the DODAG, depending on the exact message type.



You

List and explain CoAP message fields



ChatGPT

CoAP (Constrained Application Protocol) is a lightweight communication protocol designed for constrained devices and networks, often used in IoT applications. CoAP messages consist of several fields that facilitate efficient and reliable communication between CoAP endpoints. Here are the main CoAP message fields along with explanations for each:

1. **Version (Ver):**

- Size: 2 bits
- Explanation: Indicates the CoAP protocol version being used. The current version is 1.

2. **Type (T):**

- Size: 2 bits
- Explanation: Defines the type of CoAP message, which can be one of the following:
 - Confirmable (0): Message requires acknowledgment.
 - Non-confirmable (1): Message does not require acknowledgment.
 - Acknowledgment (2): Acknowledgment for a confirmable message.
 - Reset (3): Indicates that a message is invalid and should be ignored.

3. **Token Length (TKL):**

- Size: 4 bits
- Explanation: Specifies the length of the Token field in bytes. Tokens are used for message identification and correlation in CoAP.

4. **Code:**

- Size: 8 bits
- Explanation: Represents the CoAP method or response code. For requests, the code indicates the desired operation (GET, POST, PUT, DELETE, etc.). For responses, the code indicates the status of the operation (Success, Client Error, Server Error, etc.).

5. **Message ID:**

- Size: 16 bits
- Explanation: A unique identifier for each CoAP message. Used for message correlation and acknowledgment.

6. **Token:**

- Size: Variable (based on TKL)
- Explanation: A variable-length field used for message identification and token-based filtering. Tokens help match requests with corresponding responses and can be used for request deduplication.

7. **Options:**

- Size: Variable
- Explanation: Contains optional parameters or metadata associated with the CoAP message. Options include things like content format, URI path, URI query, observe option for observing resource changes, etc.

8. **Payload:**

- Size: Variable
- Explanation: Carries the actual data or payload of the CoAP message. The payload can be used to transfer application-specific information between CoAP endpoints.

CoAP

- Constrained Application Protocol (CoAP) resulted from the IETF Constrained RESTful Environments (CoRE) working group's efforts to develop a generic framework for resource-oriented applications targeting constrained nodes and networks.
- The CoAP framework defines simple and flexible ways to manipulate sensors and actuators for data or device management.
- The IETF CoRE working group has published multiple standards-track specifications for CoAP, including the following:

52

Manoj T, Dept. of CSE, SMVITM

17-04-2019

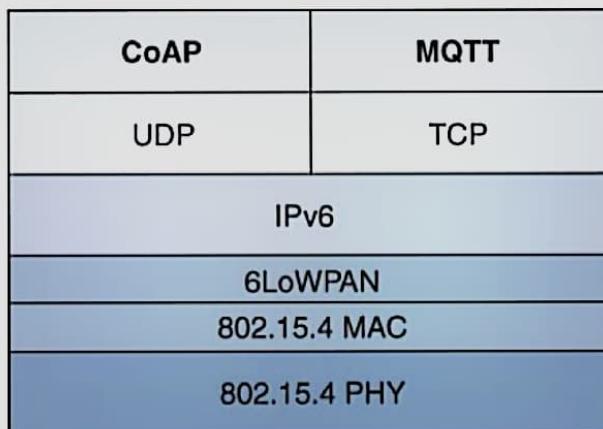


Figure 6.6: Example of a High-Level IoT Protocol Stack for CoAP and MQTT

53

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- **RFC 6690:** Constrained RESTful Environments (CoRE) Link Format
- **RFC 7252:** The Constrained Application Protocol (CoAP)
- **RFC 7641:** Observing Resources in the Constrained Application Protocol (CoAP)
- **RFC 7959:** Block-Wise Transfers in the Constrained Application Protocol (CoAP)
- **RFC 8075:** Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)

54

Manoj T, Dept. of CSE, SMVITM

17-04-2019

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). CON and ACK are highlighted in more detail in Figure 6-9.
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. For example, in Figure 6-9, GET is the request method, and 2.05 is the response code. For a complete list of values for this field, refer to RFC 7252.
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, but when it is present, a single byte of all 1s (0xFF) precedes the payload. The purpose of this byte is to delineate the end of the Options field and the beginning of Payload.



You

Explain LoRaWAN security with a neat diagram



ChatGPT

Certainly! Let's delve into LoRaWAN security and how it's structured. While I can't provide a diagram directly here, I'll describe the key components and relationships in a way that you can represent visually using tools like Lucidchart, draw.io, or any other diagramming tool of your choice.

1. End-Device Security:

- **Unique Device EUI:** Each end-device (such as sensors or IoT devices) is assigned a globally unique identifier known as the Device EUI.
- **Application Session Key (AppSKey):** A symmetric key shared between the end-device and the application server. It is used for encrypting and decrypting application payloads.
- **Network Session Key (NwkSKey):** Another symmetric key shared between the end-device and the network server. It is used for securing network-layer communication.

2. Network Server:

- **Handles Key Distribution:** The network server is responsible for distributing session keys (AppSKey, NwkSKey) securely to end-devices during the join procedure.
- **Decryption and Encryption:** It decrypts messages from end-devices using the NwkSKey and forwards decrypted payloads to the application server. It also encrypts downlink messages for end-devices using the NwkSKey.

3. Application Server:

- **Payload Processing:** Receives decrypted payloads from the network server and processes them for application-specific logic.
- **Encrypted Payloads:** Sends encrypted payloads (downlink messages) to the network server for transmission to end-devices.

4. Join Procedure:

- **OTAA (Over-The-Air Activation):** During device activation, the end-device and network server exchange messages to establish secure communication and derive session keys.
- **Join Request and Acceptance:** The end-device sends a join request containing its Device EUI, and the network server responds with a join acceptance message containing the necessary session keys.

5. Security Measures:

- **Message Integrity:** Payloads are secured using message integrity codes (MICs) to detect tampering or message alterations.
- **Frame Counters:** Both end-devices and network servers maintain frame counters to prevent replay attacks and ensure message freshness.
- **Adaptive Data Rate (ADR):** ADR mechanisms help optimize data rates based on signal quality and network conditions while maintaining security.

Security

- Security in a LoRaWAN deployment applies to different components of the architecture as shown in the Figure 4.18
- LoRaWAN endpoints must implement two layers of security, protecting communications and data privacy across the network.
- The first layer, called “**network security**” but applied at the MAC layer, guarantees the authentication of the endpoints by the LoRaWAN network server. Also, it protects LoRaWAN packets by performing encryption based on AES.
- Each endpoint implements a **network session key** (NwkSKey), used by both itself and the LoRaWAN network server

149

10-03-2023

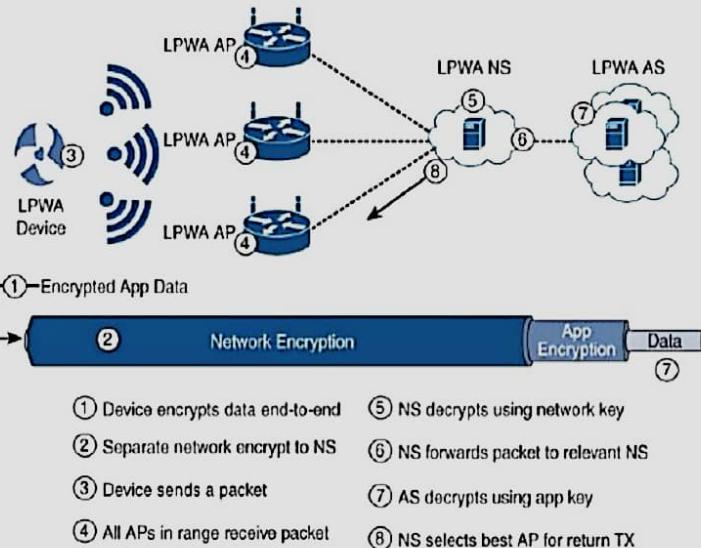


Figure 4.18: LoRaWAN Security

150

10-03-2023

- The NwkSKey ensures data integrity through computing and checking the MIC of every data message as well as encrypting and decrypting MAC-only data message payloads.
- The second layer is an **application session key** (AppSKey), which performs encryption and decryption functions between the endpoint and its application server. Furthermore, it computes and checks the application-level MIC, if included.
- Endpoints receive their AES-128 application key (AppKey) from the application owner. This key is most likely derived from an application-specific root key exclusively known to and under the control of the application provider.
- LoRaWAN endpoints attached to a LoRaWAN network must get registered and authenticated. This can be achieved through one of the two join mechanisms:

151

10-03-2023

□ Activation by Personalization(ABP)

- Endpoints don't need to run a join procedure as their individual details, including DevAddr and the NwkSKey and AppSKey session keys, are preconfigured and stored in the end device.
- This same information is registered in the LoRaWAN network server.

□ Over-the-air activation(OTAA)

- Endpoints are allowed to dynamically join a particular LoRaWAN network after successfully going through a join procedure.
- The join procedure must be done every time a session context is renewed.

- During the join process, which involves the sending and receiving of MAC layer join request and join accept messages, the node establishes its credentials with a LoRaWAN network server, exchanging its globally unique DevEUI, AppEUI, and AppKey.
- The AppKey is then used to derive the session NwkSKey and AppSKey keys.



You

Explain MQTT publish/subscribe framework



ChatGPT

MQTT (Message Queuing Telemetry Transport) is a lightweight and widely used messaging protocol designed for IoT and real-time communication scenarios. It follows a publish/subscribe messaging pattern, which is a messaging paradigm where senders (publishers) and receivers (subscribers) are decoupled, allowing for flexible and scalable communication. Here's an explanation of the MQTT publish/subscribe framework:

1. Publishers:

- **Message Source:** Publishers are entities (devices, applications, sensors) that generate messages or data to be shared with other components.
- **Topics:** Messages published by publishers are associated with specific topics. Topics are hierarchical in nature, allowing for structured message routing and filtering.
- **MQTT Publish:** Publishers use the MQTT 'PUBLISH' message to send data to the MQTT broker. The 'PUBLISH' message contains the topic and the payload (message data).

2. MQTT Broker:

- **Message Broker:** The MQTT broker is a central server or intermediary responsible for receiving, routing, and delivering messages between publishers and subscribers.
- **Topic-Based Routing:** The broker uses the topic information in the 'PUBLISH' messages to route messages to interested subscribers.
- **Persistence:** MQTT brokers can optionally store messages for subscribers that are offline or unavailable at the time of message publication.
- **Quality of Service (QoS):** Brokers manage different levels of QoS (0, 1, 2) to ensure message delivery reliability and consistency.

3. Subscribers:

- **Message Consumers:** Subscribers are entities (applications, services, devices) interested in receiving messages on specific topics.
- **Subscription:** Subscribers use the MQTT 'SUBSCRIBE' message to subscribe to one or more topics of interest. They specify the desired QoS level for message delivery.
- **MQTT Subscribe:** Subscribers receive messages published on topics they are subscribed to through the MQTT broker.
- **Unsubscription:** Subscribers can also unsubscribe from topics using the 'UNSUBSCRIBE' message when they no longer wish to receive messages.

4. Message Flow:

- **Publishing:** A publisher creates a message and sends it to the MQTT broker using the 'PUBLISH' message with a specified topic.
- **Routing:** The broker receives the message and routes it to all subscribers who have subscribed to the corresponding topic.
- **Delivery:** Subscribers receive the message based on their subscriptions and desired QoS level (0, 1, 2).
- **Acknowledgment:** Depending on the QoS level, acknowledgments (PUBACK, PUBREC, PUBCOMP) are sent back to the publisher to confirm message delivery.

Message Queuing Telemetry Transport

- At the end of the 1990s, engineers from IBM and Arcom (acquired in 2006 by Eurotech) were looking for a reliable, lightweight, and cost-effective protocol.
- They wanted to monitor and control a large number of sensors and their data from a central server location, as typically used by the oil and gas industries.
- Their research resulted in the development and implementation of the Message Queuing Telemetry Transport (MQTT) protocol that is now standardized by the Organization for the Advancement of Structured Information Standards (OASIS).

70

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- Considering the harsh environments in the oil and gas industries, an extremely simple protocol with only a few options was designed, with considerations for constrained nodes, unreliable WAN backhaul communications, and bandwidth constraints with variable latencies.
- These were some of the rationales for the selection of a client/server and publish/subscribe framework based on the TCP/IP architecture, as shown in Figure 6.10.
- An MQTT client can act as a publisher to send data (or resource information) to an MQTT server acting as an MQTT message broker.
- In the example illustrated in Figure 6.10, the MQTT client on the left side is a temperature (Temp) and relative humidity (RH) sensor that publishes its Temp/RH data.

71

Manoj T, Dept. of CSE, SMVITM

17-04-2019

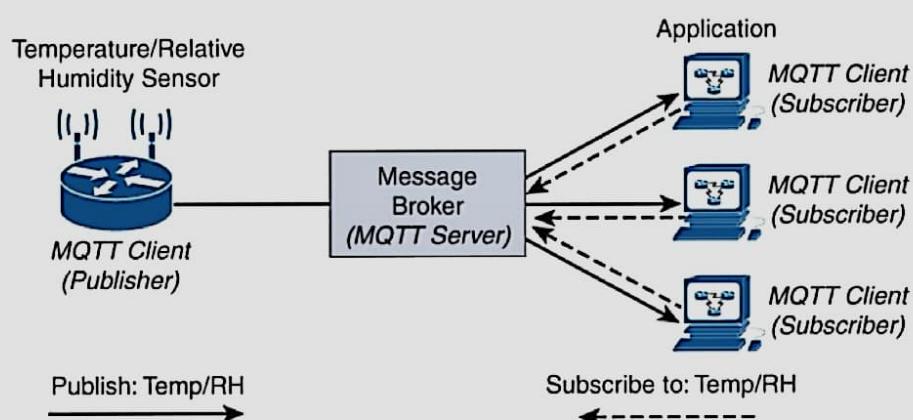


Figure 6.10: MQTT Publish/Subscribe Framework

72

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- The MQTT server (or message broker) accepts the network connection along with application messages, such as Temp/RH data, from the publishers.
- It also handles the subscription and unsubscription process and pushes the application data to MQTT clients acting as subscribers.
- The application on the right side of Figure 6-10 is an MQTT client that is a subscriber to the Temp/RH data being generated by the publisher or sensor on the left.
- This model, where subscribers express a desire to receive information from publishers, is well known. A great example is the collaboration and social networking application Twitter.

73

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- With MQTT, clients can subscribe to all data (using a wildcard character) or specific data from the information tree of a publisher.
- In addition, the presence of a message broker in MQTT decouples the data transmission between clients acting as publishers and subscribers.
- In fact, publishers and subscribers do not even know (or need to know) about each other. 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.
- This also means that publishers and subscribers do not have to be online at the same time.

74

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- 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.
- Optionally, MQTT can be secured using TLS on port 8883, and WebSocket (defined in RFC 6455) can also be used.
- MQTT is a lightweight protocol because each control packet consists of a 2-byte fixed header with optional variable header fields and optional payload.
- We should note that a control packet can contain a payload up to 256 MB. Figure 6.11 provides an overview of the MQTT message format.

75

Manoj T, Dept. of CSE, SMVITM

17-04-2019



You

Explain MQTT message format with a neat diagram.



ChatGPT

Certainly! The MQTT (Message Queuing Telemetry Transport) protocol uses a simple and efficient message format for communication between MQTT clients and brokers. Here's an explanation of the MQTT message format along with a diagram:

MQTT Message Format:

1. Fixed Header (1 Byte):

- **Message Type (4 bits)**: Indicates the type of MQTT message (CONNECT, PUBLISH, SUBSCRIBE, UNSUBSCRIBE, etc.).
- **Flags (4 bits)**: Includes control flags specific to each message type (e.g., QoS level, retain flag, clean session flag).

2. Remaining Length (Variable):

- Variable-length field representing the remaining length of the MQTT message, encoded using a variable-length encoding scheme (1-4 bytes).

3. Variable Header (Variable):

- **Connect Flags (1 Byte)**: Flags used during the CONNECT message for session management, clean session, will flag, username flag, password flag.
- **Keep Alive (2 Bytes)**: Duration in seconds for which the client and broker maintain the connection without activity.
- **Topic Name (Variable)**: Topic name or identifier for PUBLISH, SUBSCRIBE, and UNSUBSCRIBE messages.
- **Packet Identifier (2 Bytes)**: Identifier used for tracking QoS levels and message acknowledgment (for PUBLISH, PUBACK, PUBREC, PUBREL, PUBCOMP, SUBSCRIBE, UNSUBSCRIBE).

4. Payload (Variable):

- The actual data payload of the MQTT message, such as message content, subscription topics, or acknowledgment codes.
- For PUBLISH messages, the payload can include the message content, topic, QoS level, and retain flag.

Diagram of MQTT Message Format:

diff Copy code

Fixed Header	Remaining Length	Variable Header	Payload
Message Type			
Flags			

In the diagram:

- The Fixed Header contains the message type and control flags.
- The Remaining Length field specifies the length of the Variable Header and Payload.
- The Variable Header includes additional information specific to each message type, such as connect flags, keep alive duration, topic names, and packet identifiers.
- The Payload contains the actual data being transmitted, which varies based on the message type (e.g., message content, acknowledgment codes, subscription topics).

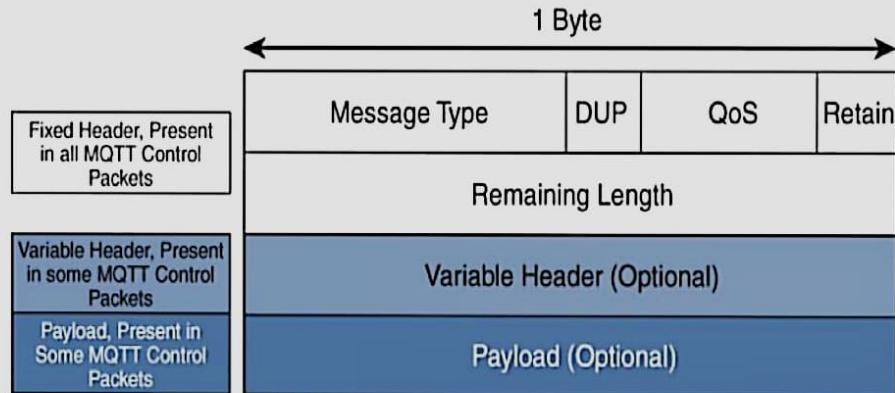


Figure 6.11: MQTT Message Format

76

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- Compared to the CoAP message format in Figure 6.7, we can see that 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 version 3.1.1.
- Each of them has a unique value that is coded into the Message Type field. Note that values 0 and 15 are reserved.
- MQTT message types are summarized in Table 6.2.

77

Manoj T, Dept. of CSE, SMVITM

17-04-2019

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 Server to client	Publish message
PUBACK	4	Client to server Server to client	Publish acknowledgement
PUBREC	5	Client to server Server to client	Publish received
PUBREL	6	Client to server Server to client	Publish release
PUBCOMP	7	Client to server 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

Table 6.2 : MQTT Message Types

78

Manoj T, Dept. of CSE, SMVITM

17-04-2019



You

Explain general MAC Frame Format for IEEE 1901.2 with a neat diagram



ChatGPT

IEEE 1901.2 is a standard for narrowband power line communication (PLC) systems, commonly used for networking over existing power line infrastructure. The MAC (Media Access Control) frame format in IEEE 1901.2 is designed to facilitate communication between devices connected via power line networks. Here's an explanation of the general MAC frame format along with a textual representation:

1. Preamble:

- Size: Variable (typically 3 bytes)
- Explanation: The preamble is used for synchronization and timing recovery between transmitting and receiving devices. It consists of a predefined bit pattern.

2. SOF (Start of Frame) Delimiter:

- Size: 1 byte
- Explanation: Marks the start of the MAC frame. It indicates the beginning of the frame and helps receivers identify the frame boundary.

3. Frame Control Field:

- Size: 1 byte
- Fields:
 - Frame Type (2 bits): Specifies the type of MAC frame (e.g., data frame, acknowledgment frame, control frame).
 - Security Enabled (1 bit): Indicates whether security mechanisms are enabled for the frame.
 - Frame Version (2 bits): Specifies the version of the MAC frame format being used.
 - Reserved (3 bits): Reserved for future use.

4. Destination Address Field:

- Size: Variable (typically 6 bytes for MAC address)
- Explanation: Contains the MAC address of the destination device or broadcast/multicast address.

5. Source Address Field:

- Size: Variable (typically 6 bytes for MAC address)
- Explanation: Contains the MAC address of the source device.

6. Payload Length Field:

- Size: 2 bytes
- Explanation: Indicates the length of the payload data in the MAC frame.

7. Payload Field:

- Size: Variable (based on Payload Length)
- Explanation: Contains the actual data being transmitted, such as network layer packets or higher-layer protocol data.

8. Frame Check Sequence (FCS):

- Size: 4 bytes
- Explanation: Contains a checksum or CRC (Cyclic Redundancy Check) value calculated over the entire MAC frame (excluding the preamble and SOF delimiter). It is used for error detection and frame integrity verification.

MAC Layer

- The MAC frame format of IEEE 1901.2a is based on the IEEE 802.15.4 MAC frame but integrates the latest IEEE 802.15.4e-2012 amendment, which enables key features to be supported.
- One of the key components brought from 802.15.4e to IEEE 1901.2a is **information elements**.

104

10-03-2023

- With IE support, additional capabilities, such as IEEE 802.15.9 Key Management Protocol and SSID, are supported.
- Figure 4.12 provides an overview of the general MAC frame format for IEEE 1901.2.

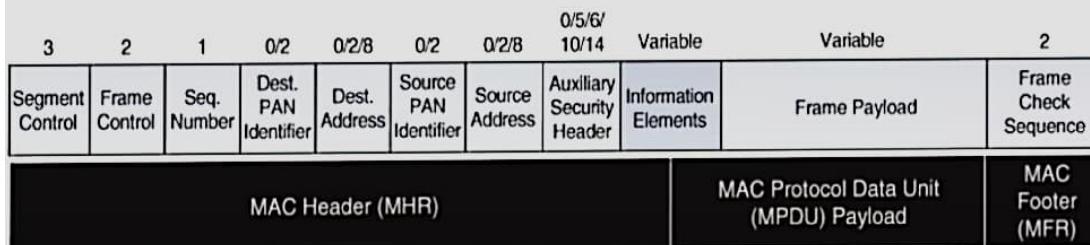


Figure 4.12: General MAC Frame Format for IEEE 1901.2

105

10-03-2023

- IEEE 1901.2 has a Segment Control field. This is a new field that was not present in the MAC frame for 802.15.4 and 802.15.4e.
- This field handles the segmentation or fragmentation of upper-layer packets with sizes larger than what can be carried in the MAC protocol data unit (MPDU).

Topology

- Use cases and deployment topologies for IEEE 1901.2a are tied to the physical power lines.
- As with wireless technologies, signal propagation is limited by factors such as noise, interference, distortion, and attenuation.

106

10-03-2023

- These factors become more prevalent with distance, so most NB-PLC deployments use some sort of mesh topology.
- Mesh networks offer the advantage of devices relaying the traffic of other devices so longer distances can be segmented.
- Figure 4.13 highlights a network scenario in which a PLC mesh network is applied to a neighborhood.
- The IEEE 1901.2a standard offers the flexibility to run any upper-layer protocol. So, implementations of IPv6 6LoWPAN and RPL IPv6 protocols are supported.
- These protocols enable the use of network layer routing to create mesh networks over PLC.

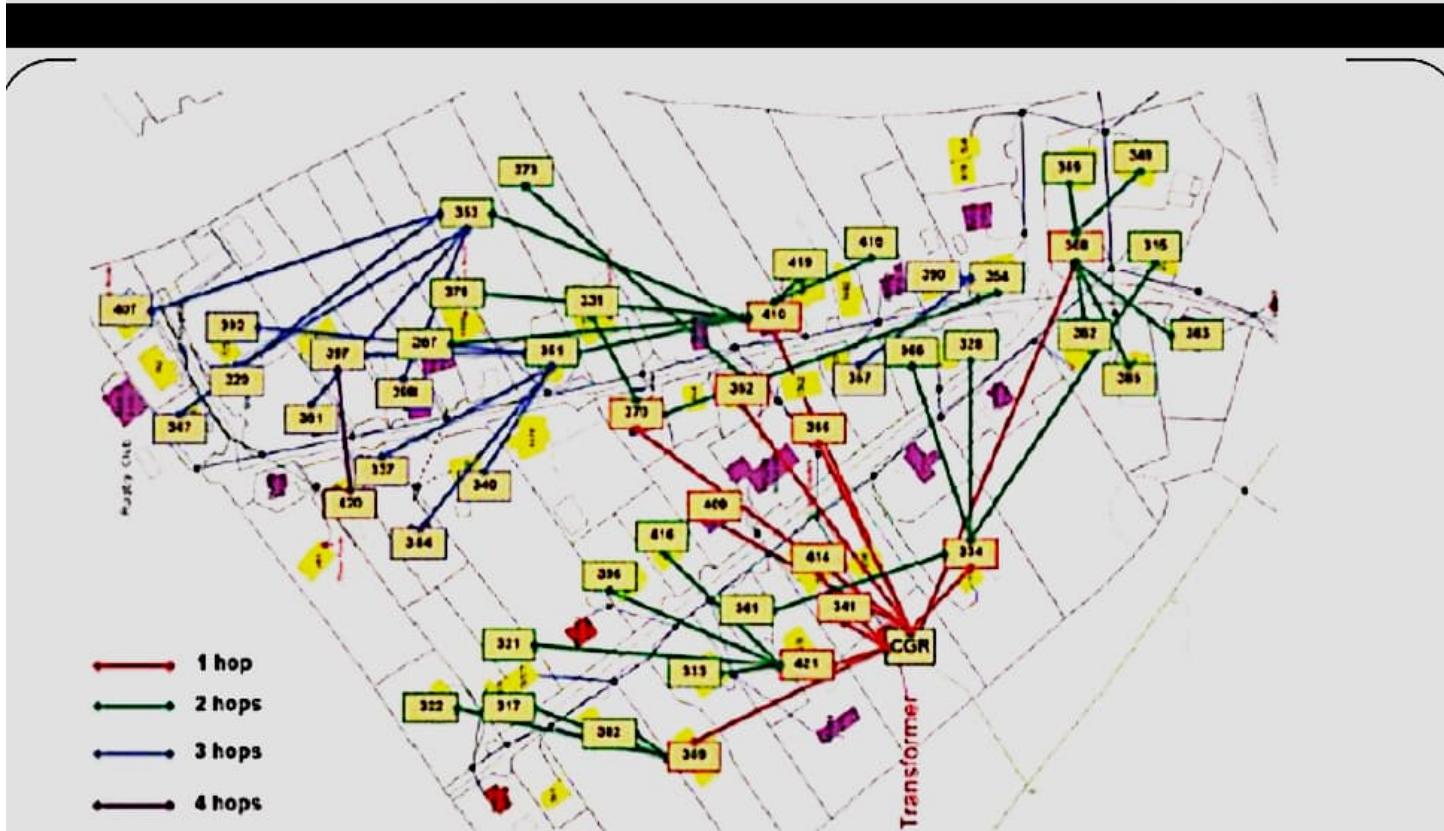


Figure 4.13: IPv6 Mesh in NB-PLC



You

Analyze the business case of the IP suite for the Internet of Things.



ChatGPT

The Internet Protocol (IP) suite plays a crucial role in the success of the Internet of Things (IoT) by providing a robust and scalable framework for communication and data exchange. Let's analyze the business case of the IP suite for IoT:

1. **Interoperability and Standardization:**

- **Business Benefit:** IP-based IoT solutions promote interoperability and standardization across diverse devices, platforms, and protocols.
- **Cost Savings:** Standardization reduces integration complexities and costs associated with proprietary protocols or vendor lock-in.
- **Market Expansion:** Interoperability fosters a larger ecosystem of compatible devices and services, attracting more customers and driving market growth.

2. **Scalability and Flexibility:**

- **Scalability:** IP networks can scale to accommodate millions of IoT devices, supporting large-scale deployments and future expansion.
- **Flexibility:** IP allows for flexible deployment models, including cloud-based, edge computing, and hybrid architectures, catering to varied business needs and use cases.
- **Agility:** Businesses can quickly adapt and scale their IoT solutions without major infrastructure overhauls, enabling rapid response to market changes and demands.

3. **Global Connectivity:**

- **Global Reach:** IP-based IoT solutions operate worldwide, facilitating global connectivity and data exchange without geographical limitations.
- **Market Penetration:** Businesses can target international markets and reach customers globally, leading to increased market penetration and revenue opportunities.

4. **Security and Privacy:**

- **Data Security:** IP supports secure communication protocols such as TLS/SSL, IPsec, and VPNs, ensuring data confidentiality, integrity, and authentication.
- **Regulatory Compliance:** Strong security measures help businesses comply with data protection regulations (e.g., GDPR, CCPA) and build trust with customers regarding data privacy.
- **Risk Mitigation:** Robust security reduces the risk of data breaches, cyberattacks, and unauthorized access, protecting sensitive IoT data and intellectual property.

5. **Integration with Existing Infrastructure:**

- **Legacy Systems:** IP integration allows businesses to leverage existing IT infrastructure, legacy systems, and networking technologies, maximizing ROI and minimizing migration efforts.
- **Hybrid Environments:** Businesses can create hybrid IoT environments by integrating IP-based solutions with legacy systems, on-premises infrastructure, and third-party services, optimizing resource utilization and performance.

6. **Management and Analytics:**

- **Remote Management:** IP-based IoT platforms enable remote device management, monitoring, and troubleshooting, improving operational efficiency and reducing maintenance costs.
- **Data Analytics:** IP networks facilitate data collection, aggregation, and analysis, empowering businesses with valuable insights for informed decision-making, predictive maintenance, and optimization of processes and services.

The Business Case for IP

- Data flowing from or to “things” is consumed, controlled, or monitored by data center servers either in the cloud or in locations that may be distributed or centralized.
- Dedicated applications are then run over virtualized or traditional operating systems or on network edge platforms (for example, fog computing). These lightweight applications communicate with the data center servers.
- Therefore, the system solutions combining various physical and data link layers call for an architectural approach with a common layer(s) independent from the lower (connectivity) and/or upper (application) layers

2

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- This is how and why the Internet Protocol (IP) suite started playing a key architectural role in the early 1990s.
- IP was not only preferred in the IT markets but also for the OT environment.



You

Compare the IoT protocol stack utilizing 6LoWPAN and IP protocol stack with a neat diagram.



ChatGPT

Let's compare the IoT protocol stack utilizing 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) and the IP protocol stack. Below is a comparison diagram followed by an explanation of each layer in both stacks:

IoT Protocol Stack Comparison:

1. Physical Layer (PHY):

- **6LoWPAN Stack:** Utilizes low-power wireless technologies like IEEE 802.15.4 for short-range communication.
- **IP Stack:** Can operate over various physical layers including wired (Ethernet), wireless (Wi-Fi, cellular), and low-power wireless technologies.

2. Link Layer:

- **6LoWPAN Stack:** Implements IEEE 802.15.4 MAC layer and provides link layer functionalities specific to low-power and lossy networks.
- **IP Stack:** Uses standard link layer protocols such as Ethernet, Wi-Fi (802.11), Bluetooth, etc., depending on the physical medium.

3. Network Layer:

- **6LoWPAN Stack:** Includes 6LoWPAN adaptation layer for compressing IPv6 packets to fit into low-power and constrained networks.
- **IP Stack:** Utilizes IPv6 or IPv4 for network addressing, routing, and communication across interconnected devices and networks.

4. Transport Layer:

- **6LoWPAN Stack:** Often uses UDP (User Datagram Protocol) for lightweight and connectionless communication.
- **IP Stack:** Supports both UDP and TCP (Transmission Control Protocol) for reliable and connection-oriented communication, depending on the application requirements.

5. Application Layer:

- **6LoWPAN Stack:** Typically includes CoAP (Constrained Application Protocol) or MQTT-SN (MQTT for Sensor Networks) for IoT-specific application messaging.
- **IP Stack:** Provides a wide range of application layer protocols such as HTTP, MQTT, CoAP, AMQP, etc., enabling diverse IoT applications and services.

Explanation of Layers:

1. Physical and Link Layers:

- Both stacks use appropriate physical and link layer technologies suitable for IoT devices, but 6LoWPAN focuses on low-power and low-data-rate communication.

2. Network Layer:

- 6LoWPAN includes an adaptation layer to handle IPv6 packets efficiently over constrained networks, while the IP stack directly uses IPv6 or IPv4 for network communication.

3. Transport Layer:

- Both stacks support lightweight protocols like UDP, but the IP stack offers the additional option of TCP for reliable and ordered data delivery.

4. Application Layer:

- 6LoWPAN and IP stacks offer various application layer protocols, but 6LoWPAN is often tailored for IoT-specific protocols like CoAP or MQTT-SN due to resource constraints.

From 6LoWPAN to 6Lo

- 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*.
- Unless the technology is proprietary, IP adaptation layers are typically defined by an IETF working group and released as a Request for Comments (RFC).
- An RFC is a publication from the IETF that officially documents Internet standards, specifications, protocols, procedures, and events.

40

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- For ex : RFC 864 describes how an IPv4 packet gets encapsulated over an Ethernet frame and RFC 2464 describes how the same function is performed for an IPv6 packet.
- IoT-related protocols follow a similar process. The main difference is that an adaptation layer designed for IoT may include some optimizations to deal with constrained nodes and networks.
- 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.
- The initial focus of the 6LoWPAN working group was to optimize the transmission of IPv6 packets over constrained networks such as IEEE 802.15.4.

41

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- Figure 5.2 shows an example of an IoT protocol stack using the 6LoWPAN adaptation layer beside the well-known IP protocol stack for reference.

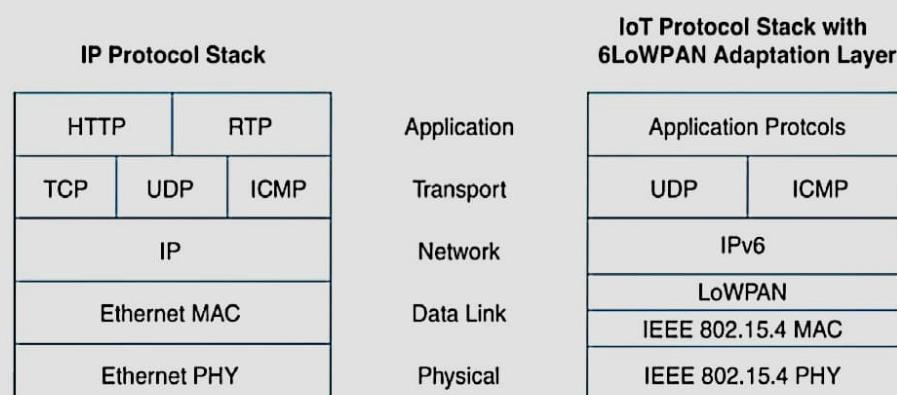


Figure 5.2: Comparison of an IoT Protocol Stack Utilizing 6LoWPAN and an IP Protocol Stack

42

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- The 6LoWPAN working group published several RFCs, but **RFC 4994** is foundational because it defines frame headers for the capabilities of header compression, fragmentation, and mesh addressing.
- These headers can be stacked in the adaptation layer to keep these concepts separate while enforcing a structured method for expressing each capability.
- Depending on the implementation, all, none, or any combination of these capabilities and their corresponding headers can be enabled.
- Figure 5.3 shows examples of typical 6LoWPAN header stacks

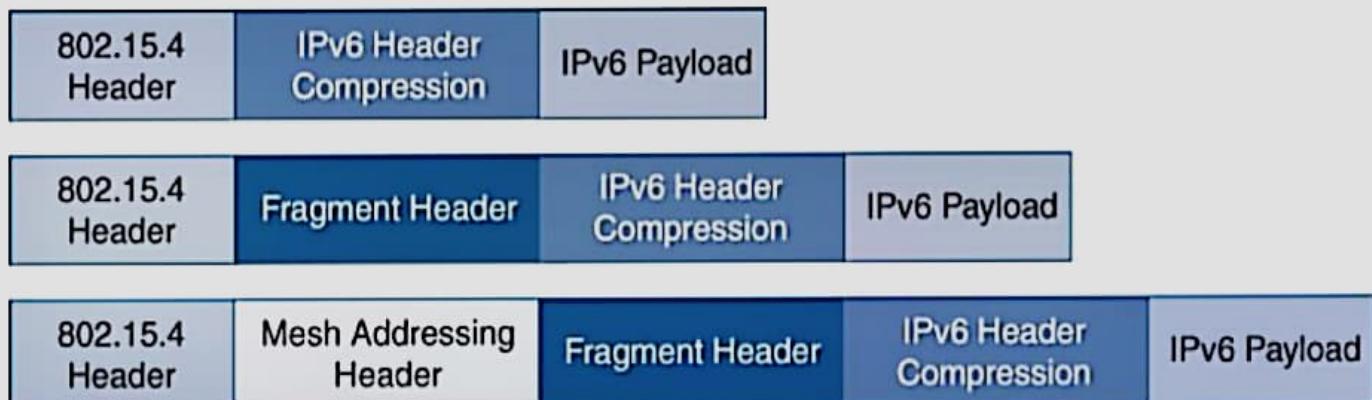


Figure 5.3: 6LoWPAN Header Stacks

Header Compression

- IPv6 header compression for 6LoWPAN was defined initially in RFC 4944 and subsequently updated by RFC 6282.



You

List and explain any 5 RPL routing metrics.



ChatGPT

RPL (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol designed for constrained IoT environments. It uses routing metrics to determine optimal paths for data transmission. Here are five commonly used RPL routing metrics along with explanations for each:

1. **ETX (Expected Transmission Count):**

- **Definition:** ETX represents the expected number of transmissions required for a packet to reach its destination without errors.
- **Calculation:** ETX is calculated based on link quality indicators such as packet delivery ratio (PDR) and link quality estimation (LQE).
- **Advantages:** ETX is effective in selecting routes with high reliability and low packet loss, making it suitable for applications requiring reliable data delivery.

2. **Hop Count:**

- **Definition:** Hop count simply measures the number of hops (intermediate nodes) between the source and destination nodes in a network.
- **Calculation:** It is a straightforward metric where each intermediate node increments the hop count as the packet passes through.
- **Advantages:** Hop count is easy to calculate and implement, making it a simple metric for basic route selection. However, it may not consider link quality or energy efficiency.

3. **MRHOF (Minimum Rank with Hysteresis Objective Function):**

- **Definition:** MRHOF is an objective function used in RPL that combines ETX and hop count to calculate the routing rank of nodes.
- **Calculation:** MRHOF considers both ETX for reliability and hop count for path length, balancing between reliability and energy efficiency.
- **Advantages:** MRHOF provides a balanced approach by considering both reliability and energy consumption, making it suitable for networks with constrained resources.

4. **Energy Consumption:**

- **Definition:** Energy consumption metrics estimate the energy required for data transmission over a specific route.
- **Calculation:** Energy consumption metrics consider factors such as transmission power, data packet size, and radio duty cycle to estimate energy usage.
- **Advantages:** Energy consumption metrics help in selecting energy-efficient routes, prolonging the battery life of constrained devices in IoT networks.

5. **Link Quality Estimation (LQE):**

- **Definition:** LQE measures the quality of wireless links based on parameters like signal strength, packet error rate, and signal-to-noise ratio (SNR).
- **Calculation:** LQE algorithms assess link quality using statistical models or empirical data collected from network measurements.
- **Advantages:** LQE helps in dynamically adapting routing decisions based on real-time link conditions, optimizing data transmission and network performance.

Metrics

- RPL defines a large and flexible set of new metrics and constraints for routing in RFC 6551.
- Developed to support powered and battery-powered nodes, RPL offers a far more complete set than any other routing protocol.
- Some of the RPL routing metrics and constraints defined in RFC 6551 include the following:

□ Expected Transmission Count (ETX)

- Assigns a discrete value to the number of transmissions a node expects to make to deliver a packet.

85

Manoj T, Dept. of CSE, SMVITM

05-04-2019

□ Hop Count

- Tracks the number of nodes traversed in a path. Typically, a path with a lower hop count is chosen over a path with a higher hop count.

□ Latency

- Varies depending on power conservation. Paths with a lower latency are preferred.

□ Link Quality Level

- Measures the reliability of a link by taking into account packet error rates caused by factors such as signal attenuation and interference.

□ Link Color

- Allows manual influence of routing by administratively setting values to make a link more or less desirable.

86

Manoj T, Dept. of CSE, SMVITM

05-04-2019

- These values can be either statically or dynamically adjusted for specific traffic types.

□ Node State and Attribute

- Identifies nodes that function as traffic aggregators and nodes that are being impacted by high workloads.
- High workloads could be indicative of nodes that have incurred high CPU or low memory states.
- Naturally, nodes that are aggregators are preferred over nodes experiencing high workloads.

□ Node Energy

- Avoids nodes with low power, so a battery-powered node that is running out of energy can be avoided and the life of that node and the network can be prolonged.

87

Manoj T, Dept. of CSE, SMVITM

05-04-2019

□ Throughput

- Provides the amount of throughput for a node link. Often, nodes conserving power use lower throughput.
- This metric allows the prioritization of paths with higher throughput.
- In addition to the metrics and constraints listed in RFC 6551, others can also be implemented.
- For ex : let's look at a scenario in which two constraints are used as a filter for pruning links that do not satisfy the specified conditions.
- One of the constraints is ETX. The other constraint, Relative Signal Strength Indicator (RSSI), specifies the power present in a received radio signal.

88

Manoj T, Dept. of CSE, SMVITM

05-04-2019



You

Explain CoAP message format with a neat diagram



ChatGPT

Certainly! The Constrained Application Protocol (CoAP) is a lightweight messaging protocol designed for IoT and constrained environments. CoAP messages follow a simple and efficient format for communication between clients and servers. Here's an explanation of the CoAP message format along with a diagram:

CoAP Message Format:

1. Fixed-Length Header (4 Bytes):

- **Version (Ver) (2 bits)**: Specifies the CoAP version (e.g., 1 for CoAP version 1).
- **Type (T) (2 bits)**: Indicates the message type (Confirmable, Non-confirmable, Acknowledgment, Reset).
- **Token Length (TKL) (4 bits)**: Specifies the length of the Token field in bytes (0-8 bytes).
- **Code (8 bits)**: Represents the CoAP method or response code (e.g., GET, POST, PUT, DELETE, Success, Client Error, Server Error).

2. Message ID (Mid) (2 Bytes):

- A unique identifier for each CoAP message, used for message correlation and acknowledgment.

3. Token (Variable Length):

- A variable-length field (0-8 bytes) used for message identification and correlation in CoAP. Tokens help match requests with corresponding responses and can be used for request deduplication.

4. Options (Variable Length):

- Optional parameters or metadata associated with the CoAP message. Options include things like content format, URI path, URI query, observe option for observing resource changes, etc.
- Options are represented using a delta encoding format to optimize message size.

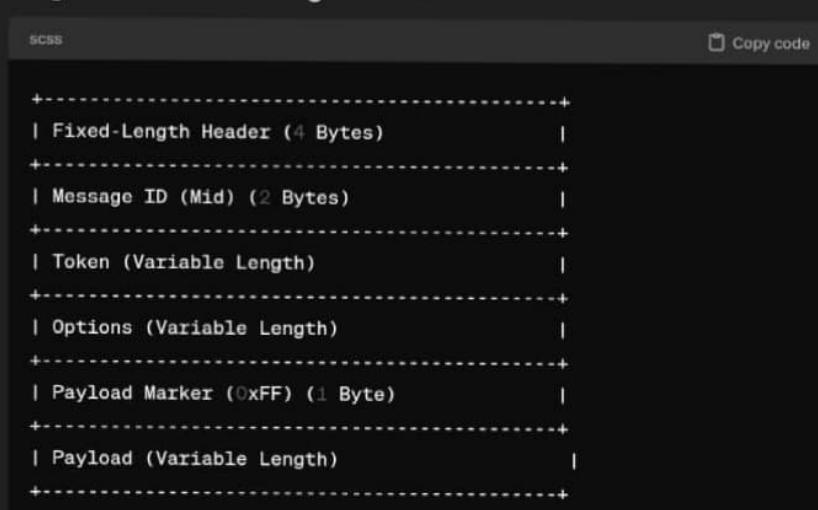
5. Payload Marker (0xFF) (1 Byte):

- Marks the beginning of the payload field. The payload contains the actual data being transferred (application-specific data).

6. Payload (Variable Length):

- The actual data or message payload being transferred between CoAP endpoints.

Diagram of CoAP Message Format:



In the diagram:

- The Fixed-Length Header includes Version, Type, Token Length, and Code fields.
- The Message ID (Mid) uniquely identifies each CoAP message.
- The Token field is variable in length and helps in message identification.
- Options and Payload are also variable in length and contain additional metadata and actual data, respectively.

IoT Application Layer Protocols

- 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.
- To address this problem, the 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. Figure 6.6 highlights their position in a common IoT protocol stack.
- In Figure 6.6, CoAP and MQTT are naturally at the top of this sample IoT stack, based on an IEEE 802.15.4 mesh network.

51

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- We will almost always find CoAP deployed over UDP and MQTT running over TCP.

CoAP

- Constrained Application Protocol (CoAP) resulted from the IETF Constrained RESTful Environments (CoRE) working group's efforts to develop a generic framework for resource-oriented applications targeting constrained nodes and networks.
- The CoAP framework defines simple and flexible ways to manipulate sensors and actuators for data or device management.
- The IETF CoRE working group has published multiple standards-track specifications for CoAP, including the following:

52

Manoj T, Dept. of CSE, SMVITM

17-04-2019

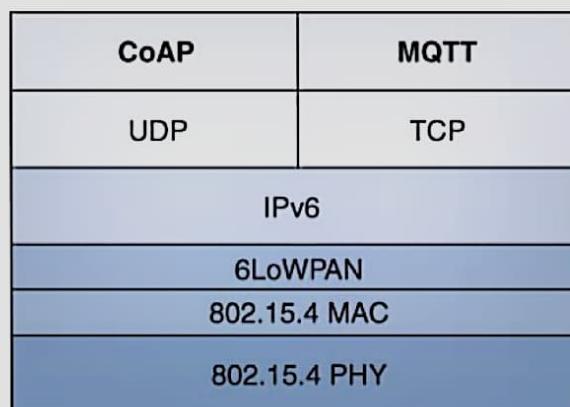


Figure 6.6: Example of a High-Level IoT Protocol Stack for CoAP and MQTT

53

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- **RFC 6690:** Constrained RESTful Environments (CoRE) Link Format
- **RFC 7252:** The Constrained Application Protocol (CoAP)
- **RFC 7641:** Observing Resources in the Constrained Application Protocol (CoAP)
- **RFC 7959:** Block-Wise Transfers in the Constrained Application Protocol (CoAP)
- **RFC 8075:** Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)

54

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- 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).
- The IETF CoRE working group is studying alternate transport mechanisms, including TCP, secure TLS, and WebSocket.
- CoAP over Short Message Service (SMS) as defined in Open Mobile Alliance for Lightweight Machine-to-Machine (LWM2M) for IoT device management is also being considered.
- RFC 7252 provides more details on securing CoAP with DTLS.
- It specifies how a CoAP endpoint is provisioned with keys and a filtering list.

55

Manoj T, Dept. of CSE, SMVITM

17-04-2019

55-56/89

- Four security modes are defined: NoSec, PreSharedKey, RawPublicKey, and Certificate. The NoSec and RawPublicKey implementations are mandatory.
- From a formatting perspective, a CoAP message is composed of a short fixedlength Header field (4 bytes), a variable-length but mandatory Token field (0–8 bytes), Options fields if necessary, and the Payload field.
- Figure 6.7 details the CoAP message format, which delivers low overhead while decreasing parsing complexity.
- The CoAP message format is relatively simple and flexible. It allows CoAP to deliver low overhead, which is critical for constrained networks, while also being easy to parse and process for constrained devices.

56

Manoj T, Dept. of CSE, SMVITM

17-04-2019



You

Explain the differences between MQTT and CoAP



ChatGPT

MQTT (Message Queuing Telemetry Transport) and CoAP (Constrained Application Protocol) are both lightweight communication protocols designed for IoT and machine-to-machine (M2M) applications. However, they have distinct differences in their design, use cases, and protocol features. Here are the key differences between MQTT and CoAP:

1. Protocol Design:

- **MQTT:** MQTT is a publish/subscribe-based messaging protocol. It uses a broker-based architecture where clients (publishers and subscribers) communicate through a central broker. Messages are published to topics, and subscribers receive messages based on their subscriptions to specific topics.
- **CoAP:** CoAP is a request/response-based protocol designed for constrained devices and low-power networks. It operates over UDP or DTLS (Datagram Transport Layer Security) and follows a client/server model. CoAP clients send requests to CoAP servers, which respond with corresponding responses.

2. Messaging Patterns:

- **MQTT:** Supports publish/subscribe and message queuing patterns. Clients can publish messages to topics, and multiple subscribers can receive messages from subscribed topics asynchronously.
- **CoAP:** Primarily uses request/response messaging. Clients send CoAP requests (GET, POST, PUT, DELETE) to servers to retrieve or modify resources. CoAP also supports observe notifications for resource observation and event notification.

3. Transport Protocol:

- **MQTT:** Typically operates over TCP/IP, providing reliable, connection-oriented communication. It also supports MQTT over WebSocket for communication over web-based applications.
- **CoAP:** Operates over UDP for lightweight, connectionless communication suitable for constrained networks. CoAP can also use DTLS for secure communication over UDP.

4. Message Size and Overhead:

- **MQTT:** Offers more flexibility in message size and supports larger payloads compared to CoAP. MQTT messages have a fixed-length header and optional variable-length payload.
- **CoAP:** Optimized for constrained environments with smaller packet sizes. CoAP messages are typically smaller in size compared to MQTT due to efficient header compression and minimal overhead.

5. Quality of Service (QoS):

- **MQTT:** Provides three levels of QoS (0, 1, 2) for message delivery reliability. QoS levels ensure message acknowledgment, retransmission, and ordering based on the desired level of reliability.
- **CoAP:** Offers different reliability modes including non-confirmable (no acknowledgment), confirmable (acknowledgment), and ack-piggyback (piggybacks ACKs with responses) for reliable communication over unreliable networks.

6. Usage and Application:

- **MQTT:** Commonly used for scenarios requiring asynchronous messaging, real-time data streams, and event-driven communication. Suitable for applications like telemetry, monitoring, and remote control.
- **CoAP:** Ideal for IoT applications on constrained devices, constrained networks, and resource-constrained environments. Used for resource discovery, sensor data retrieval, and machine-to-machine communication in IoT deployments.

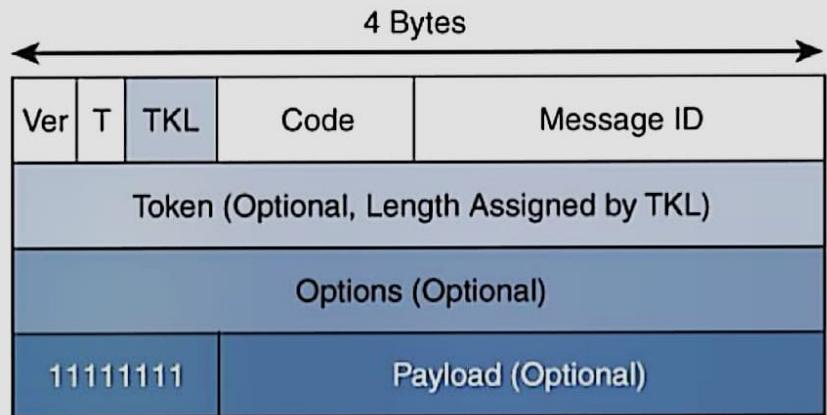


Figure 6.7: CoAP Message Format

57

Manoj T, Dept. of CSE, SMVITM

17-04-2019

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). CON and ACK are highlighted in more detail in Figure 6-9.
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. For example, in Figure 6-9, GET is the request method, and 2.05 is the response code. For a complete list of values for this field, refer to RFC 7252.
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, but when it is present, a single byte of all 1s (0xFF) precedes the payload. The purpose of this byte is to delineate the end of the Options field and the beginning of Payload.

58

Manoj T, Dept. of CSE, SMVITM

17-04-2019

- CoAP can run over IPv4 or IPv6. However, it is recommended that the message fit within a single IP packet and UDP payload to avoid fragmentation.
- For IPv6, with the default MTU size being 1280 bytes and allowing for no fragmentation across nodes, the maximum CoAP message size could be up to 1152 bytes, including 1024 bytes for the payload.
- In the case of IPv4, as IP fragmentation may exist across the network, implementations should limit themselves to more conservative values and set the IPv4 Don't Fragment (DF) bit.
- While most sensor and actuator traffic utilizes small-packet payloads, some use cases, such as firmware upgrades, require the capability to send larger payloads.

59

Manoj T, Dept. of CSE, SMVITM

17-04-2019