Module 3

IoT Network Layer

IP as the IoT Network Layer: The Business Case for IP, The need for Optimization, Optimizing IP for IoT, Profiles and Compliances, Application Protocols for IoT: The Transport Layer, IoT Application Transport Methods

IP as the IoT Network Layer, The Business Case for IP

IOT

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.

- 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. That is using **IP (Internet Protocol)**

The Business Case for IP

- The Key Advantages of Internet Protocol of Things:
- Open and standards-based
- Versatile
- Ubiquitous
- Scalable
- Manageable and highly secure
- Stable and resilient
- Consumers’ market adoption
- The innovation factor

The Business Case for IP

The Key Advantages of Internet Protocol

- 1. **Open and standards-based:**
- 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.
- 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 Business Case for IP

The Key Advantages of Internet Protocol

- **2. Versatile:**
- 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.
- 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.

The Business Case for IP

The Key Advantages of Internet Protocol

- **3.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.
- 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 when you look at what is supported across the various IoT solutions and industry verticals.

The Business Case for IP

The Key Advantages of Internet Protocol

- **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.
- Adding huge numbers of “things” to private and public infrastructures may require optimizations and design rules specific to the new devices.
- However, you should realize that this is not very different from the recent evolution of voice and video endpoints integrated over IP.
- IP has proven before that scalability is one of its strengths.

The Business Case for IP

The Key Advantages of Internet Protocol

- **5. Manageable and highly secure:**
- Communications infrastructure requires appropriate management and security capabilities for proper operations.
- 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.

The Business Case for IP

The Key Advantages of Internet Protocol

- **6. Stable and resilient:**
- IP has been around for 30 years, and it is clear that IP is a workable solution.
- 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.

The Business Case for IP

The Key Advantages of Internet Protocol

- **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.
- 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 Business Case for IP

The Key Advantages of Internet Protocol

- **8. 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.
- Innovations in IoT can also leverage an IP underpinning.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- How to implement IP in data center, cloud services, and operation centers hosting IoT applications may seem obvious, but the adoption of IP in the last mile is more complicated and often makes running IP end-to-end more difficult.
- The use of numerous network layer protocols in addition to IP is often a point of contention between computer networking experts.
- Typically, one of two models, adaptation or adoption, is proposed:
- **Adaptation** means application layered gateways (ALGs) must be implemented to ensure the translation between non-IP and IP layers.
- **Adoption** involves replacing all non-IP layers with their IP layer counterparts, simplifying the deployment model and operations.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

❖ In the industrial and manufacturing sector, Solutions and product lifecycles many protocols have been developed for serial communications.

- While IP and Ethernet support were not specified in the initial versions, more recent specifications for these serial communications protocols integrate Ethernet and IPv4.

❖ Supervisory control and data acquisition (SCADA) applications that operate both the IP adaptation model and the adoption model.

- Implementations that make use of IP adaptation have SCADA devices attached through serial interfaces to a gateway tunneling or translating the traffic.
- With the IP adoption model, SCADA devices are attached via Ethernet to switches and routers forwarding their IPv4 traffic.

❖ ZigBee that runs a non-IP stack between devices and a ZigBee gateway that forwards traffic to an application server.

- A ZigBee gateway often acts as a translator between the ZigBee and IP protocol stacks.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- Following factors when trying to determine which model is best suited for IP connectivity:
- Bidirectional versus unidirectional data flow
- Overhead for last-mile communications paths
- Data flow model
- Network diversity

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- **1. Bidirectional versus unidirectional data flow:** As defined in RFC 7228, may only infrequently need to report a few bytes of data to an application.
- These sorts of devices, particularly ones that communicate through LPWA technologies, include fire alarms sending alerts or daily test reports, electrical switches being pushed on or off, and water or gas meters sending weekly indexes.
- If there is only one-way communication to upload data to an application, then it is not possible to download new software or firmware to the devices. This makes integrating new features and bug and security fixes more difficult.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- **2. Overhead for last-mile communications paths:**
- IPv4 has 20 bytes of header at a minimum, and IPv6 has 40 bytes at the IP network layer.
- For the IP transport layer, UDP has 8 bytes of header overhead, while TCP has a minimum of 20 bytes.
- If the data to be forwarded by a device is infrequent and only a few bytes, you can potentially have more header overhead than device data—again, particularly in the case of LPWA technologies

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- **3. Data flow model:**
- Any node can easily exchange data with any other node in a network, although security, privacy, and other factors may put controls and limits on the “end-toend” concept.
- However, in many IoT solutions, a device’s data flow is limited to one or two applications.
- In this case, the adaptation model can work because translation of traffic needs to occur only between the end device and one or two application servers.
- Depending on the network topology and the data flow needed, both IP adaptation and adoption models have roles to play in last-mile connectivity.

The Business Case for IP

Adoption or Adaptation of the Internet Protocol

- **4.Network diversity:**
- One of the drawbacks of the adaptation model is a general dependency on single PHY and MAC layers.
- Integration and coexistence of new physical and MAC layers or new applications impact how deployment and operations have to be planned.
- This is not a relevant consideration for the adoption model.

The need for Optimizing IP for IoT

The Need for Optimization

- ❖ Internet of Things will largely be built on the Internet Protocol suite
- ❖ In coping with the integration of non-IP devices, may need to 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.
- ❖ The following concepts take a detailed look at why optimization is necessary for IP.
  - Constrained Nodes
  - Constrained Networks
  - IP Versions

The Need for Optimization

- ❖ **Constrained Nodes**
- ❖ IoT having different classes of devices coexist.
- ❖ Depending on its functions in a network, a “thing” architecture may or may not offer similar characteristics compared to a generic PC or server in an IT environment.
- ❖ Another limit is that this network protocol stack on an 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 is a key characteristic of constrained nodes.
  - Battery Enabled with the life span of Months to 10 years
  - battery-powered nodes impact communication intervals.

The Need for Optimization

- **Constrained Nodes**
- 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 (adaptation model).
- **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 behaviors must cope with the bandwidth constraints.

The Need for Optimization

- **Constrained Networks**
- Low-speed connections (like low-speed modems) demonstrated that IP could run over low-bandwidth 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.
- A constrained network can have high latency and a high potential for packet loss.
- Constrained networks are often referred to as low-power and lossy networks (LLNs).
- Constrained networks operate between a few kbps and a few hundred kbps and may utilize a star, mesh, or combined network topologies, ensuring proper operations.

The Need for Optimization

- **IP Versions**
- IETF has been working on transitioning the 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.
- Today, both versions of IP run over the Internet, but most traffic is still IPv4 based.
- Internet of Things has the Internet itself and support both IPv4 and IPv6 versions concurrently.
- Techniques such as tunneling and translation need to be employed in IoT solutions to ensure interoperability between IPv4 and IPv6.

The Need for Optimization

- **IP Versions**
- The following are some of the main factors applicable to IPv4 and IPv6 support in an IoT solution:
- **1. Application Protocol:** IoT devices versions of IP run over the Internet, but most 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.
- **2. 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.
- **3. Serial Communications:** Data is transferred using either proprietary or standards-based protocols, such as DNP3, Modbus, or IEC 60870-5-101. In the past, communicating this serial data over any sort of distance could be handled by an analog modem connection.

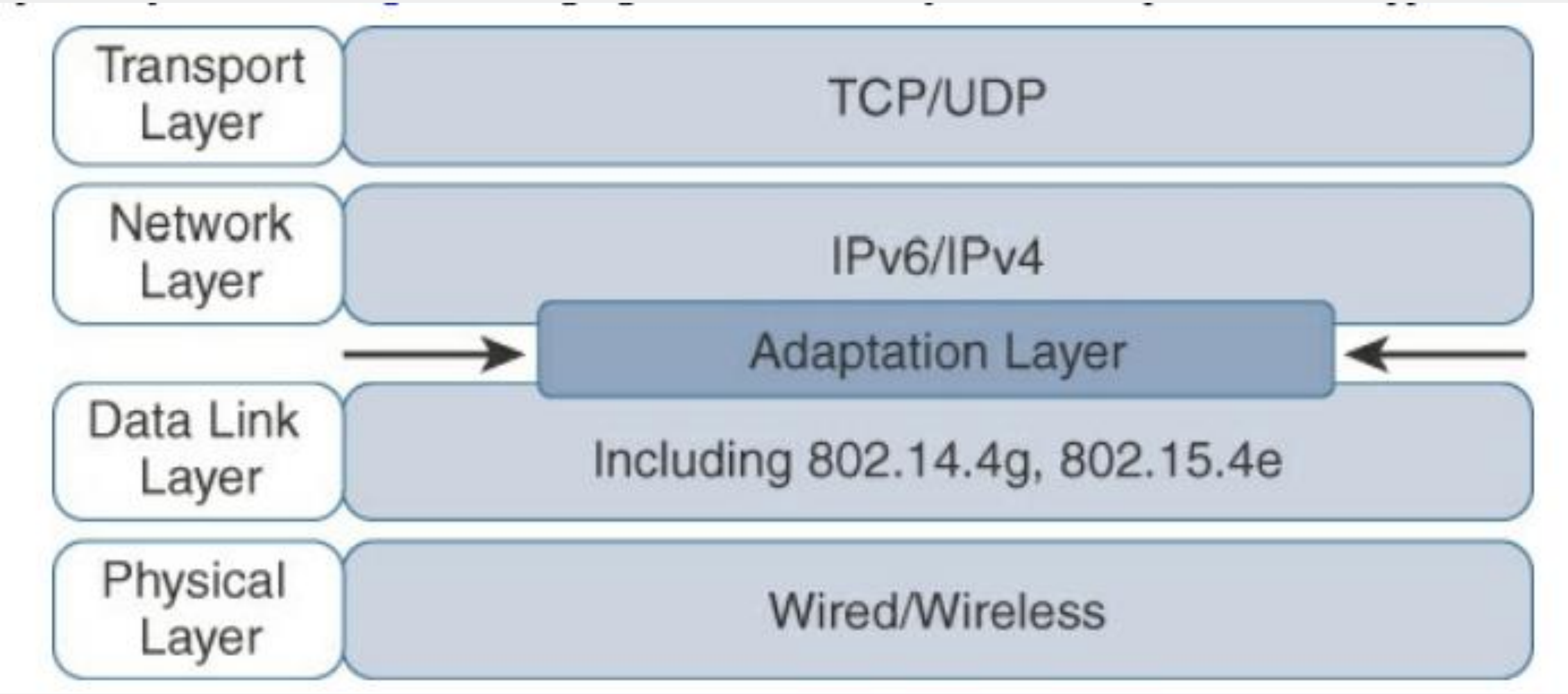
The Need for Optimization

- **IP Versions**
- **4. IPv6 Adaptation Layer:** The most common physical and data link layers (Ethernet, Wi-Fi, and so on) stipulate adaptation layers for both versions, newer technologies, such as IEEE 802.15.4 (Wireless Personal Area Network), IEEE 1901.2, and ITU G.9903 (Narrowband Power Line Communications) only have an IPv6 adaptation layer specified

Optimizing IP for IoT, Profiles, and Compliance

Optimizing IP for IoT

- While the Internet Protocol is key for a successful Internet of Things, constrained nodes and constrained networks mandate optimization at various layers and on multiple protocols of the IP architecture

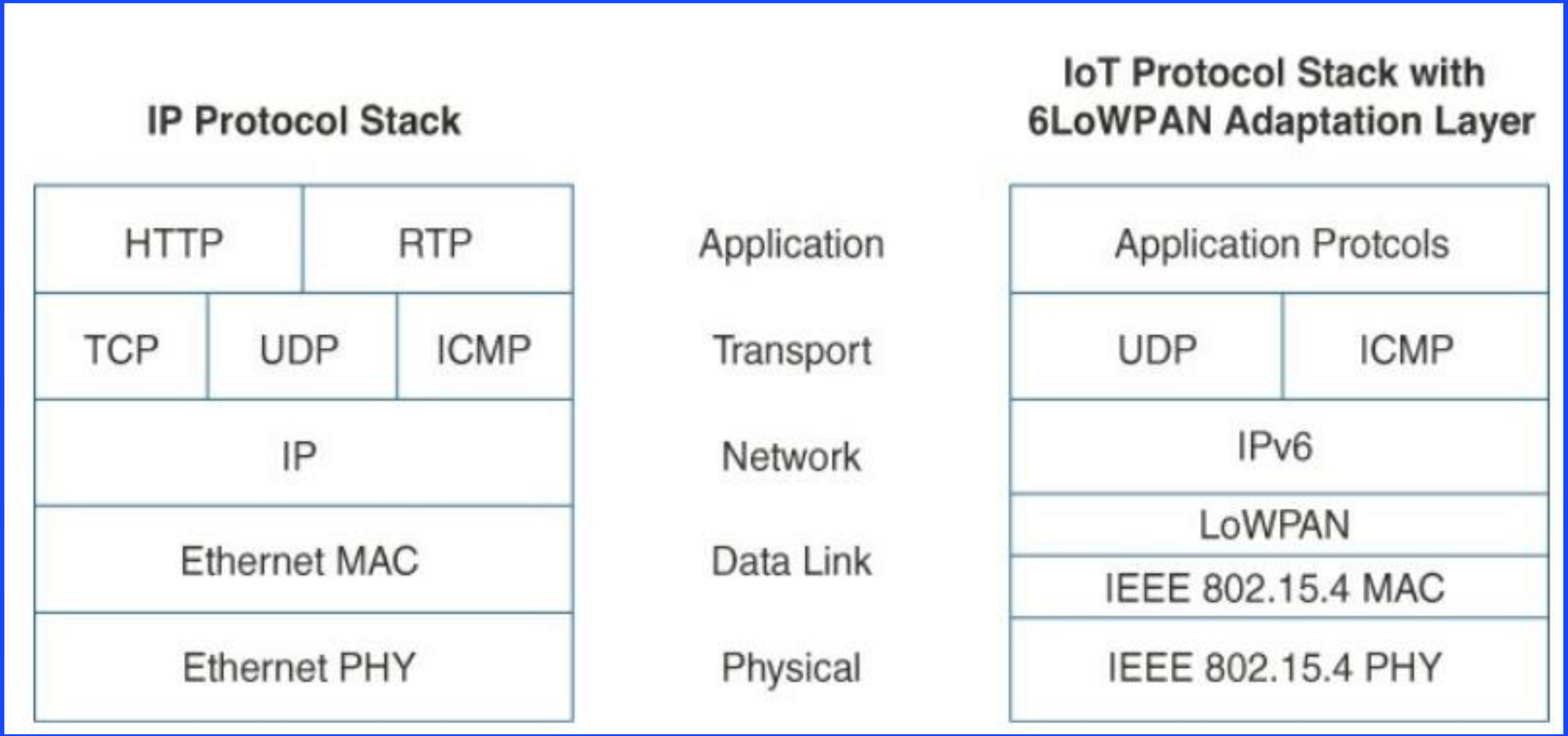


Optimizing IP for IoT

- ❖ The following optimizations technique of IP already available:
- ❖ **From 6LoWPAN to 6Lo**
  - Header Compression
  - Fragmentation
  - Mesh Addressing
  - Mesh-Under Versus Mesh-Over Routing
  - 6Lo Working Group
- ❖ 6TiSCH
- ❖ RPL
  - Objective Function (OF)
  - Rank
  - RPL Headers
  - Metrics
- ❖ Authentication and Encryption on Constrained Nodes
  - ACE
  - DICE

Optimizing IP for IoT

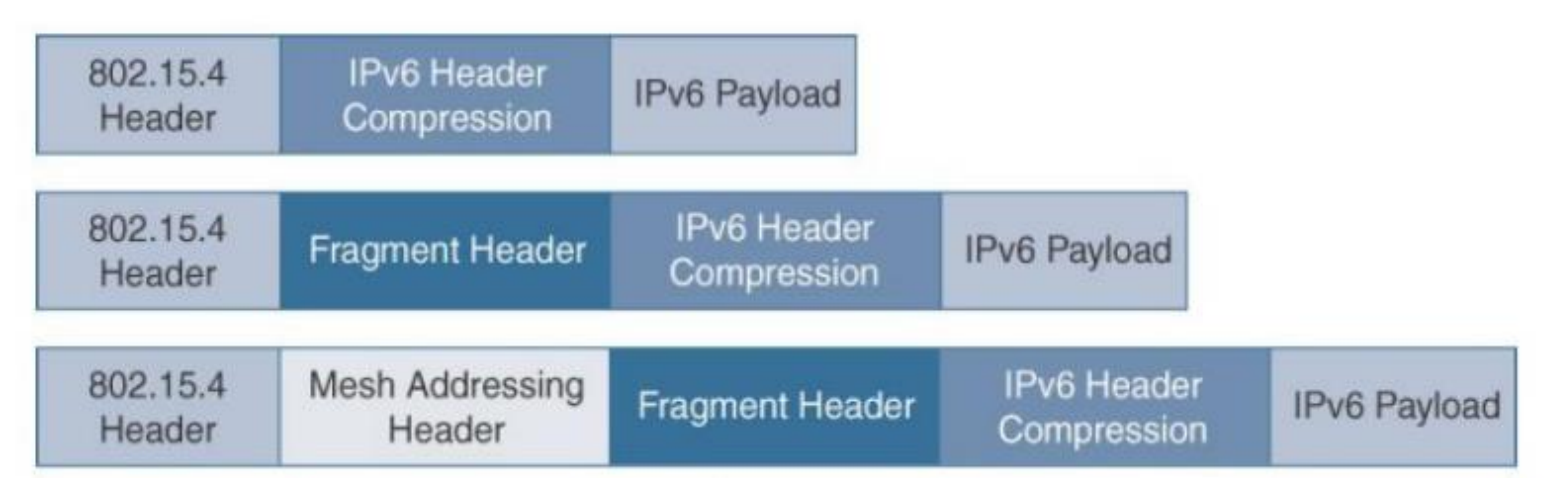
- **From 6LoWPAN to 6Lo**
- In the IP architecture, the transport of IP packets over any given Layer 1 (PHY) and Layer 2 (MAC) protocol must be defined.
- The model for packaging IP into lower-layer protocols is often referred to as an adaptation layer.
- The main examples of adaptation layers optimized 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.



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

Optimizing IP for IoT

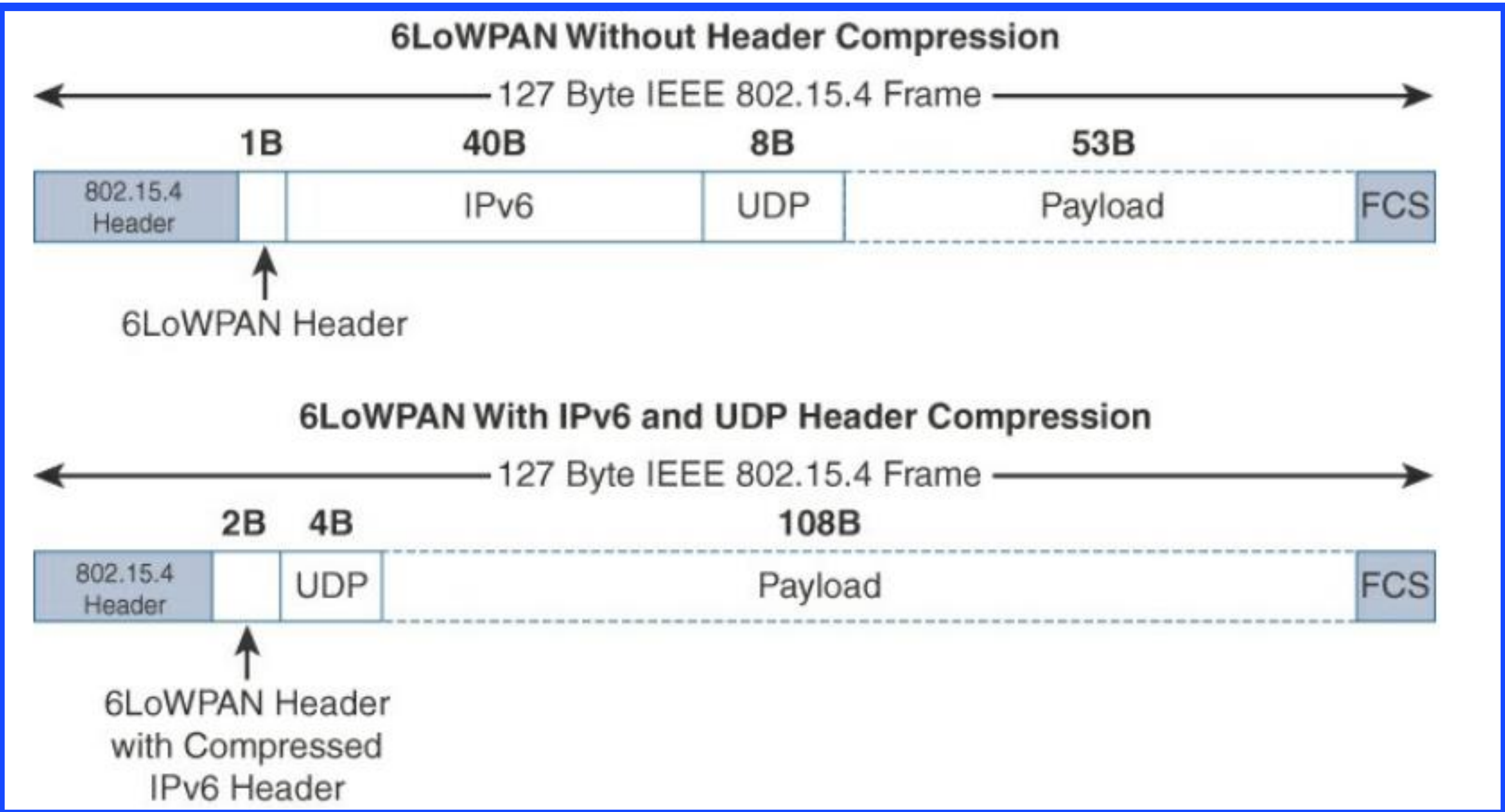
- **From 6LoWPAN to 6Lo**
- The 6LoWPAN working group published several RFCs (Request for Comments by IETF), but RFC defines frame headers for the capabilities of
- Header compression,
- Fragmentation,
- Mesh addressing.



6LoWPAN Header Stacks

Optimizing IP for IoT

- **From 6LoWPAN to 6Lo**
- **Header Compression:**
- 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.
- Header compression for 6LoWPAN is only defined for an IPv6 header and not for IPv4.
- However, a number of factors affect the amount of compression, such as implementation of RFC, whether UDP is included, and various IPv6 addressing scenarios.
- 6LoWPAN works by taking advantage of shared information known by all nodes from their participation in the local network.
- In addition, it omits some standard header fields by assuming commonly used values.



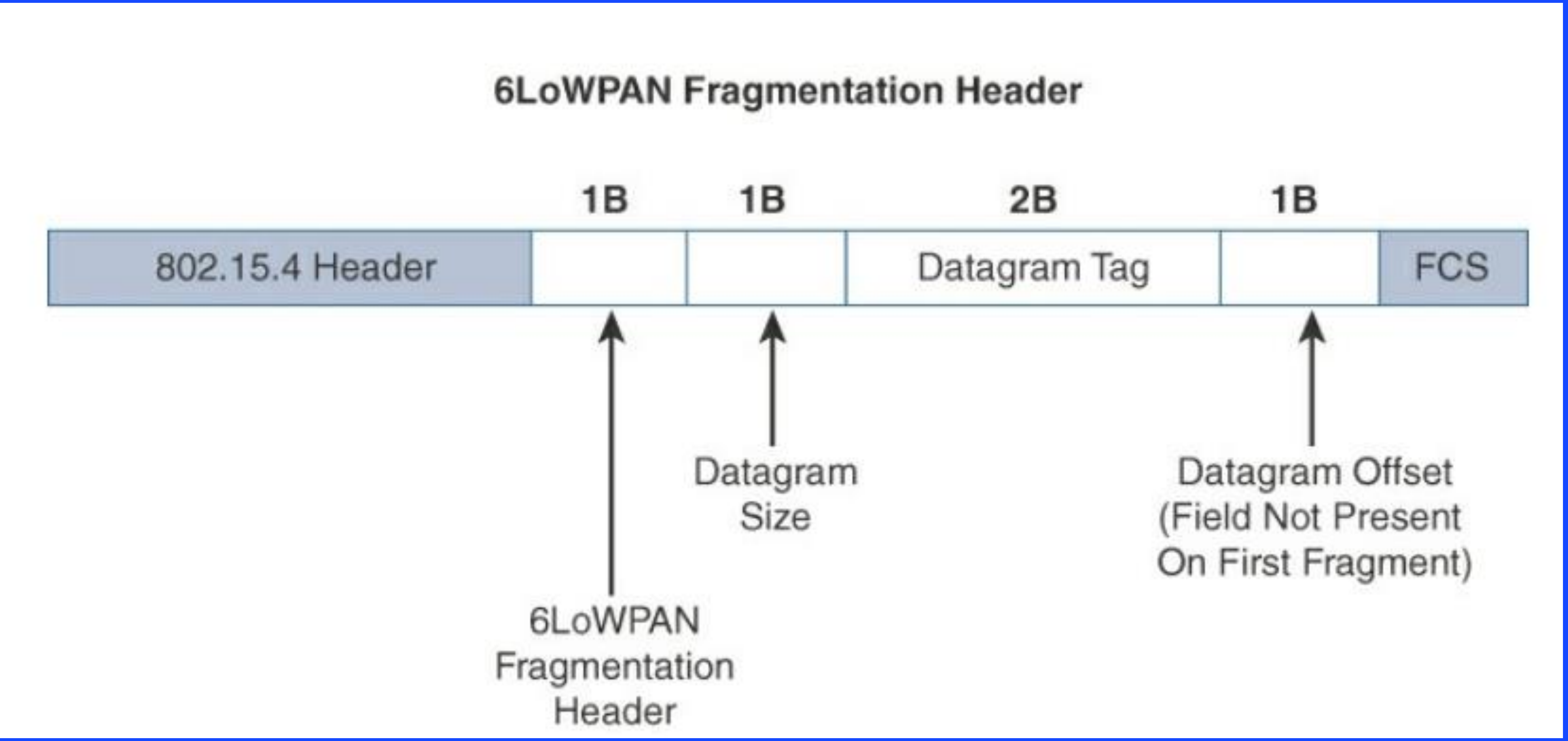
6LoWPAN Header Compression

Optimizing IP for IoT

- **From 6LoWPAN to 6Lo**
- **Header Compression:**
- At the top of, a 6LoWPAN frame without any header compression enabled:
  - The full 40-byte IPv6 header and 8-byte UDP header are visible.
  - The 6LoWPAN header is only a single byte in this case.
  - Uncompressed IPv6 and UDP headers leave only 53 bytes of data payload out of the 127-byte maximum frame size in the case of IEEE 802.15.4.
- The bottom half of shows a frame where header compression enabled:
  - The 6LoWPAN header increases to 2 bytes to accommodate the compressed IPv6 header, and UDP has been reduced in half, to 4 bytes from 8.
  - Most importantly, the header compression has allowed the payload to more than double, from 53 bytes to 108 bytes, which is obviously much more efficient.

Optimizing IP for IoT

- **From 6LoWPAN to 6Lo**
- **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.
- For IEEE 802.15.4, 127 bytes is the MTU.
- A problem because of IPv6, with a much larger MTU, is carried inside the 802.15.4 frame with a much smaller one.
- To remedy this situation, large IPv6 packets must be fragmented across multiple 802.15.4 frames at Layer 2.



6LoWPAN Fragmentation Header

Optimizing IP for IoT

- **From 6LoWPAN to 6Lo**
- The fragment header utilized by 6LoWPAN is composed of three primary fields:
  - • Datagram Size: The 1-byte field specifies the total size of the unfragmented payload
  - • Datagram Tag: identifies the set of fragments for a payload.
  - • Datagram Offset: field delineates how far into a payload a particular fragment occurs.
- • The 6LoWPAN fragmentation header field itself uses a unique bit value to identify that the subsequent fields behind it are fragment fields as opposed to another capability, such as header compression.
- • In the first fragment, the Datagram Offset field is not present because it would simply be set to 0.
- • This results in the first fragmentation header for an IPv6 payload being only 4 bytes long.

Optimizing IP for IoT

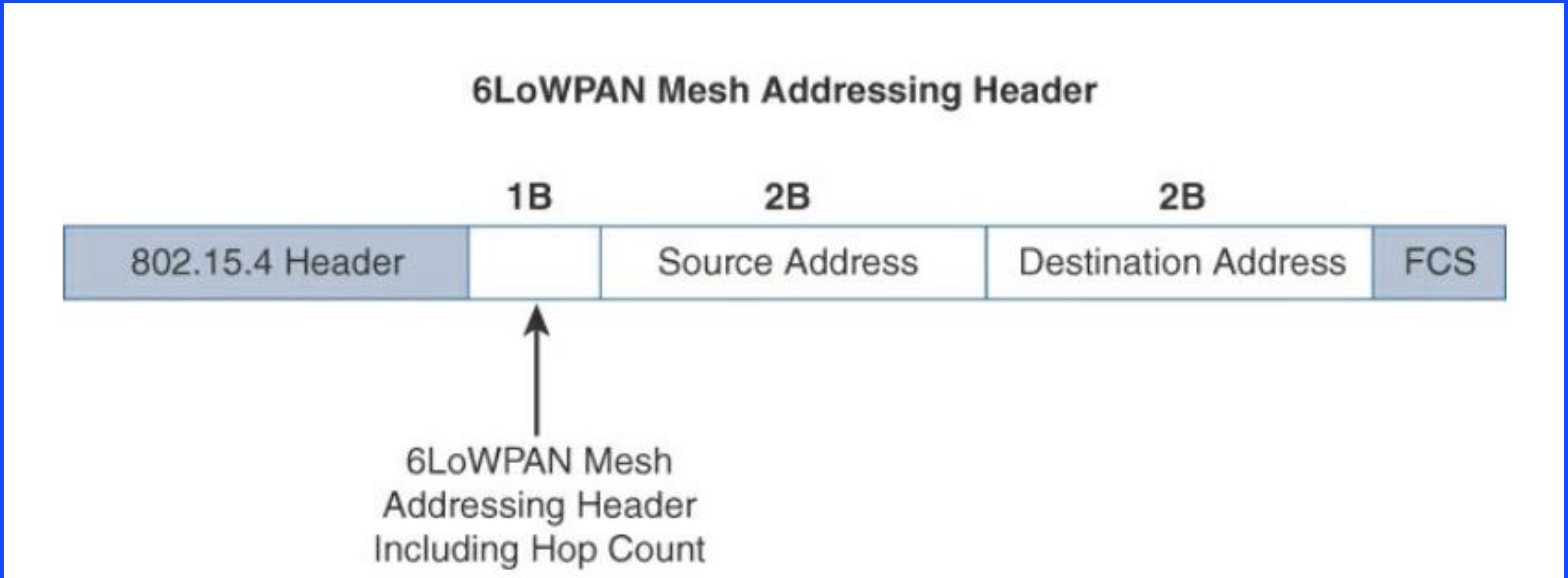
❖ **From 6LoWPAN to 6Lo**

❖ **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: 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.
- Source Address, and Destination Address: The Source Address and Destination Address fields for mesh addressing are IEEE 802.15.4 addresses indicating the endpoints of an IP hop.



6LoWPAN Mesh Addressing Header

Optimizing IP for IoT

- **From 6LoWPAN to 6Lo**
- **Mesh-Under Versus Mesh-Over Routing:**
- IEEE 802.15.4, IEEE 802.15.4g, and IEEE 1901.2a that support mesh topologies and operate at the physical and data link layers, two main options exist for establishing reachability and forwarding packets.
- **“Mesh-under”:** the routing of packets is handled at the 6LoWPAN adaptation layer.
- **“Mesh-over” or “route-over”:** utilizes IP routing for getting packets to their destination

Optimizing IP for IoT

- **From 6LoWPAN to 6Lo**
- Mesh-under routing,
- The routing of IP packets leverages the 6LoWPAN mesh addressing header to route and forward packets at the link layer.
- The term mesh-under is used because multiple link layer hops can be used to complete a single IP hop.
- Nodes have a Layer 2 forwarding table that they consult to route the packets to their final destination within the mesh.
- An edge gateway terminates the mesh-under domain.
- The edge gateway must also implement a mechanism to translate between the configured Layer 2 protocol and any IP routing mechanism implemented on other Layer 3 IP interfaces.

Optimizing IP for IoT

- **From 6LoWPAN to 6Lo**
- Mesh-over or route-over scenarios,
- IP Layer 3 routing is utilized for computing reachability and then getting packets forwarded to their destination, either inside or outside the mesh domain.
- Each full-functioning node acts as an IP router, so each link layer hop is an IP hop.
- When a LoWPAN has been implemented using different link layer technologies, a mesh-over routing setup is useful.
- While traditional IP routing protocols can be used, a specialized routing protocol for smart objects, such as RPL

Optimizing IP for IoT

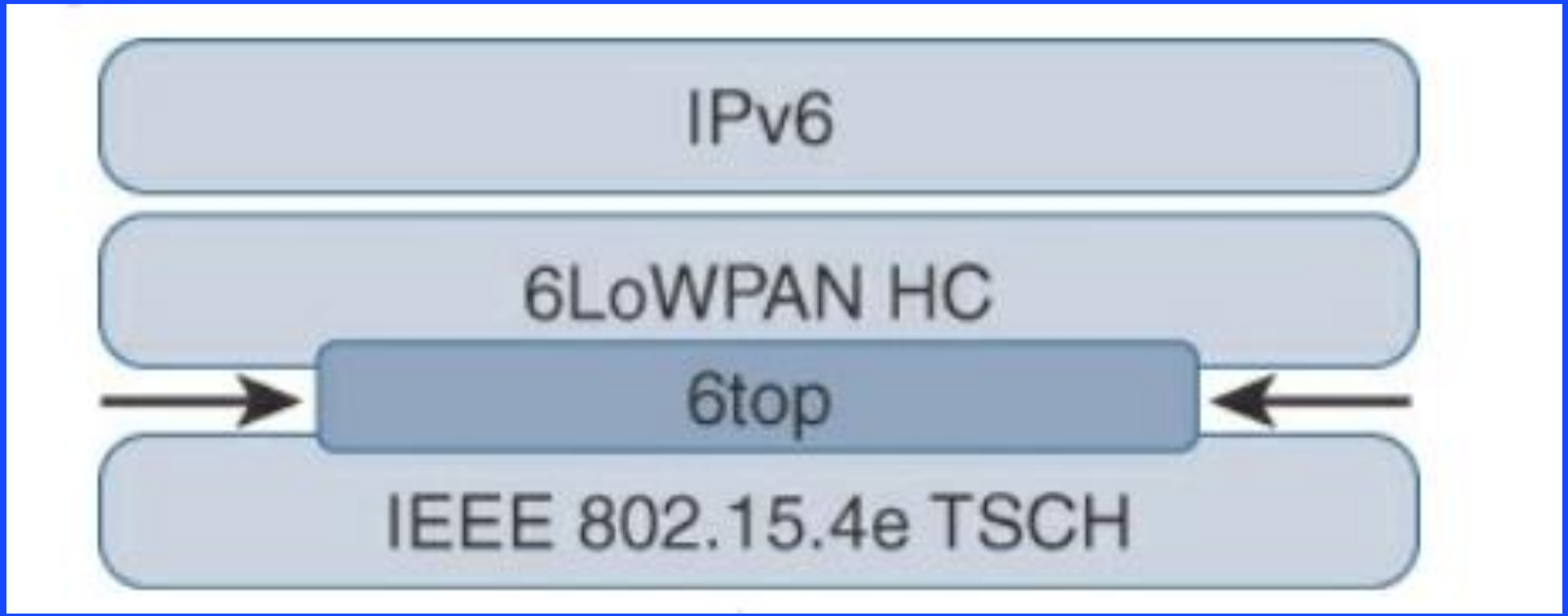
- **From 6LoWPAN to 6Lo**
- 6Lo Working Group:
- The 6Lo working group seeks to expand on this completed work with a focus on IPv6 connectivity over constraine dnode networks.
- While the 6LoWPAN working group initially focused its optimizations on IEEE 802.15.4 LLNs

Optimizing IP for IoT

- **From 6LoWPAN to 6Lo**
- 6Lo working group is focused on the following:
- IPv6-over-foo adaptation layer specifications using 6LoWPAN technologies (RFC4944, RFC6282, RFC6775) for link layer technologies:
- For example, this includes:
- IPv6 over Bluetooth Low Energy
- Transmission of IPv6 packets over near-field communication
- IPv6 over 802.11ah
- Transmission of IPv6 packets over DECT Ultra Low Energy
- Transmission of IPv6 packets on WIA-PA (Wireless Networks for Industrial Automation–Process Automation)
- Transmission of IPv6 over Master Slave/Token Passing (MS/TP)
- Information and data models such as MIB modules:
- One example is RFC 7388, "Definition of Managed Objects for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)."
- Optimizations that are applicable to more than one adaptation layer specification:
- For example, this includes RFC 7400, "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)."
- Informational and maintenance publications needed for the IETF specifications in this area

Optimizing IP for IoT

- 6TiSCH
- IEEE 802.15.4e, Time-Slotted Channel Hopping (TSCH), is an add-on to the Media Access Control (MAC) portion of the IEEE 802.15.4 standard,
- Devices implementing IEEE 802.15.4e TSCH communicate by following a Time Division Multiple Access (TDMA) schedule.
- An allocation of a unit of bandwidth or time slot is scheduled between neighbor nodes.
- This allows the programming of predictable transmissions and enables deterministic, industrial-type applications.
- Not like other IEEE 802.15.4
- To standardize IPv6 over the TSCH mode of IEEE 802.15.4e (known as 6TiSCH)
- The IEEE 802.15.4e standard defines a time slot structure, but it does not mandate a scheduling algorithm for how the time slots are utilized.
- This is left to higher-level protocols like 6TiSCH.
- Scheduling is critical because it can affect throughput, latency, and power consumption.



Location of 6TiSCH's 6top Sublayer

Optimizing IP for IoT

- 6TiSCH
- Schedules in 6TiSCH are broken down into cells.
- A cell is simply a single element in the TSCH schedule that can be allocated for unidirectional or bidirectional communications between specific nodes.
- Nodes only transmit when the schedule dictates that their cell is open for communication.
- The 6TiSCH architecture defines four schedule management mechanisms:
- Static scheduling
- Neighbor-to-neighbor scheduling
- Remote monitoring and scheduling management
- Hop-by-hop scheduling .

Optimizing IP for IoT

- 6TiSCH
- In addition to schedule management functions, the 6TiSCH architecture also defines three different forwarding models.
- Forwarding is the operation performed on each packet by a node that allows it to be delivered to a next hop or an upperlayer protocol.
- The forwarding decision is based on a pre existing state that was learned from a routing computation.
- There are three 6TiSCH forwarding models:
- Track Forwarding (TF)
- Fragment forwarding (FF)
- IPv6 Forwarding (6F)

Optimizing IP for IoT

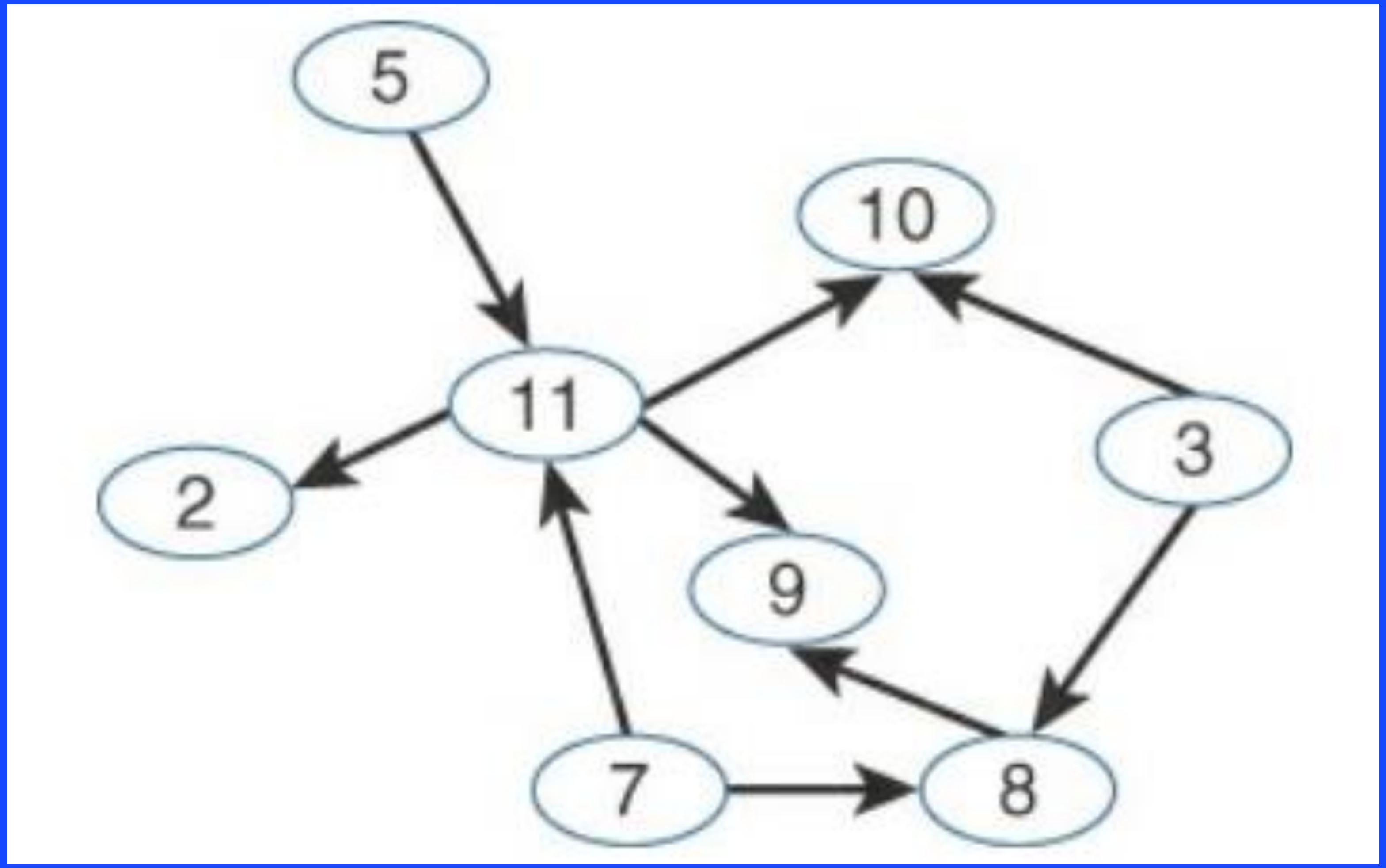
- RPL
- IETF chartered the RoLL (Routing over Low-Power and Lossy Networks) working group to evaluate all Layer 3 IP routing protocols and determine the needs and requirements for developing a routing solution for IP smart objects.
- The new routing protocol should be developed for use by IP smart objects is IPv6 Routing Protocol for Low Power and Lossy Networks (RPL).
- In an RPL network,
- each node acts as a router and becomes part of a mesh network.
- Routing is performed at the IP layer.
- Each node examines every received IPv6 packet and determines the nexthop destination based on the information contained in the IPv6 header

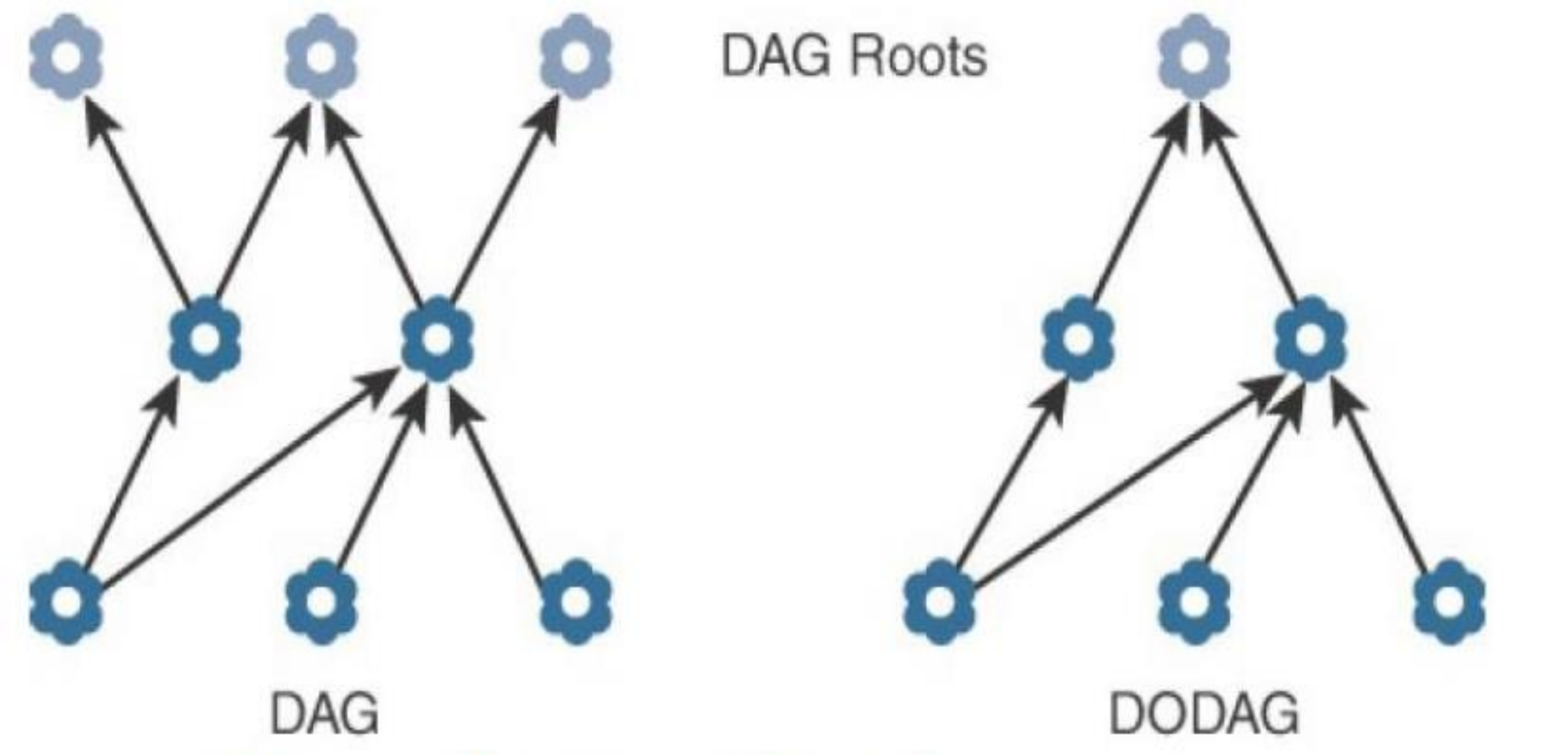
Optimizing IP for IoT

- **RPL**
- The constraints of computing and memory that are common characteristics of constrained nodes, the protocol defines two modes:
- **Storing mode:**
- All nodes contain the full routing table of the RPL domain. Every node knows how to directly reach every other node.
- **Non-storing mode:**
- Only the border router(s) of the RPL domain contain(s) the full routing table.
- All other nodes in the domain only maintain their list of parents and use this as a list of default routes toward the border router.
- This abbreviated routing table saves memory space and CPU.
- When communicating in non-storing mode, a node always forwards its packets to the border router, which knows how to ultimately reach the final destination.

Optimizing IP for IoT

- **RPL**
- 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, you 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.
- 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.
- Observe that that a DAG has multiple roots, whereas the DODAG has just one.





Example of a Directed Acyclic Graph (DAG)

DAG and DODAG Comparison

Optimizing IP for IoT

- Authentication and Encryption on Constrained Nodes
- The Authentication and Authorization for Constrained Environments (ACE) working group is tasked with evaluating the applicability of existing authentication and authorization protocols and documenting their suitability for certain constrained-environment use cases.
- ACE working group will focus its work on CoAP with the Datagram Transport Layer Security (DTLS) protocol.
- The ACE working group expects to produce a standardized solution for authentication and authorization that enables authorized access (Get, Put, Post, Delete) to resources identified by a URI and hosted on a resource server in constrained environments.
- An unconstrained authorization server performs mediation of the access.

Optimizing IP for IoT

- Authentication and Encryption on Constrained Nodes
- DICE:
- New generations of constrained nodes implementing an IP stack over constrained access networks are expected to run an optimized IP protocol stack.
- The DTLS in Constrained Environments (DICE) working group focuses on implementing the DTLS transport layer security protocol in these environments.
- The first task of the DICE working group is to define an optimized DTLS profile for constrained nodes.
- In addition, the DICE working group is considering the applicability of the DTLS record layer to secure multicast messages and investigating how the DTLS handshake in constrained environments can get optimized.

Profiles and Compliances

- Profile definitions, certifications, and promotion by alliances can help implementers develop solutions that guarantee interoperability and/or interchangeability of devices.
- Some of the main industry organizations working on profile definitions and certifications for IoT constrained nodes and networks.
- Internet Protocol for Smart Objects (IPSO) Alliance
- Wi-SUN Alliance
- Thread
- IPv6 Ready Logo

Profiles and Compliances

- **Internet Protocol for Smart Objects (IPSO) Alliance**
- The alliance initially focused on promoting IP as the premier solution for smart objects communications.
- Today, it is more focused on how to use IP, with the IPSO Alliance organizing interoperability tests between alliance members to validate that IP for smart objects can work together and properly implement industry standards.

Profiles and Compliances

- **Wi-SUN Alliance**
- 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 utilities industry is the main area of focus for the Wi-SUN Alliance.
- The Wi-SUN field area network (FAN) profile enables smart utility networks to provide **resilient, secure, and cost-effective** connectivity with extremely good coverage in a range of **topographic environments,** from dense urban neighborhoods to rural areas.

Profiles and Compliances

- **Thread**
- Thread 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.

Profiles and Compliances

- **IPv6 Ready Logo**
- The IPv6 Ready Logo program has established conformance and interoperability testing programs with the intent of increasing user confidence when implementing IPv6.
- The IPv6 Core and specific IPv6 components, such as DHCP, IPsec, and customer edge router certifications, are in place.

Application Protocols for IoT

Application Protocols for IoT

- ❖ **Introduction**
- ❖ Concepts covered about the higher-layer IoT protocols
- ❖ **The Transport Layer:**
  - IP-based networks use either TCP or UDP. However, the constrained nature of IoT networks requires a closer look at the use of these traditional transport mechanisms.
- ❖ **IoT Application Transport Methods:**
  - The various types of IoT application data and the ways this data can be carried across a network.

Application Protocols for IoT

- ❖ **The Transport Layer**
- ❖ The selection of a protocol for the transport layer as supported by the TCP/IP architecture in the context of IoT networks.
- ❖ With the TCP/IP protocol, two main protocols are specified for the transport layer:
- ❖ **Transmission Control Protocol (TCP):**
  - This connection-oriented protocol requires a session to get established between the source and destination before exchanging data.
- ❖ **User Datagram Protocol (UDP):**
  - With this connectionless protocol, data can be quickly sent between source and destination but with no guarantee of delivery.

Application Protocols for IoT

- ❖ IoT Application Transport Methods
- ❖ Concepts Covered:
- ❖ Application Layer Protocol Not Present
- ❖ SCADA
  - A Little Background on SCADA
  - Adapting SCADA for IP
- ❖ IoT Application Layer Protocols
- ❖ CoAP
- ❖ Message Queuing Telemetry Transport (MQTT) .

Application Protocols for IoT

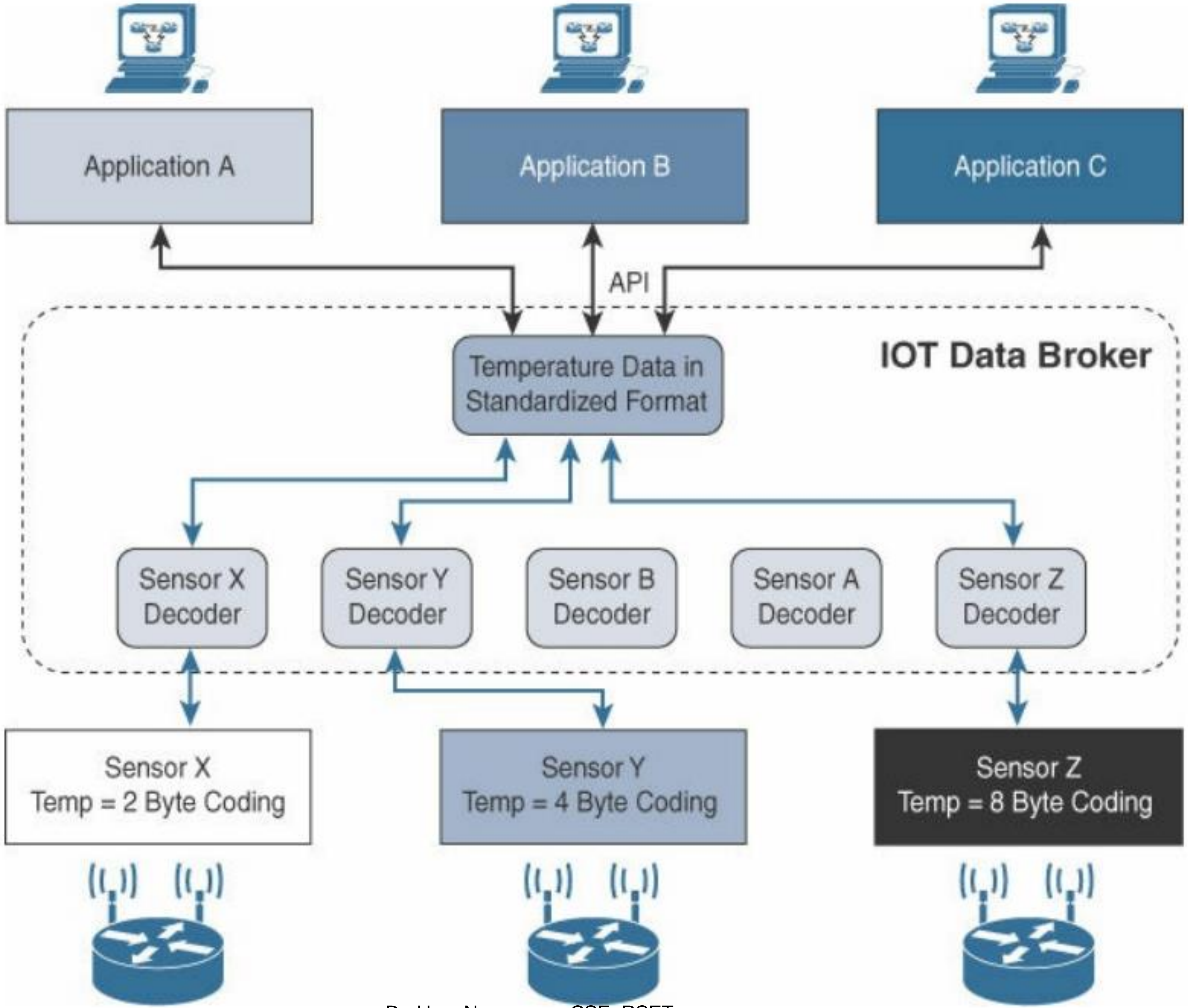
- **The Transport Layer**
- Because of constrained nodes and network need to use new IoT application protocol, such as Constrained Application Protocol (CoAP), almost always uses UDP and why implementations of industrial application layer protocols may call for the optimization and adoption of the UDP transport layer if run over LLNs.
  - Select TCP for cellular networks because these networks are typically more robust and can handle the overhead.
  - For LLNs, where both the devices and network itself are usually constrained, UDP is a better choice and often mandatory.
- TCP and UDP are the two main choices at the transport layer for the TCP/IP protocol. The performance and scalability of IoT constrained devices and networks is impacted by which one of these is selected.

Application Protocols for IoT

- **IoT Application Transport Methods**
- Application layer protocol not present: In this case, the data payload is directly transported on top of the lower layers. No application layer protocol is used.
- 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.
- 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.
- 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), are two well-known examples of IoT application layer protocols

IoT Application Transport Methods

- **Application Layer Protocol Not Present**
- In Class 0 send or receive only a few bytes of data.
- For many reasons, 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.
- Implementing a robust protocol stack is usually not useful and sometimes not even possible with the limited available resources.
- While many constrained devices, such as sensors and actuators, have adopted deployments that have no application layer, this transportation method has not been standardized.
- This lack of standardization makes it difficult for generic implementations of this transport method to be successful from an interoperability perspective.

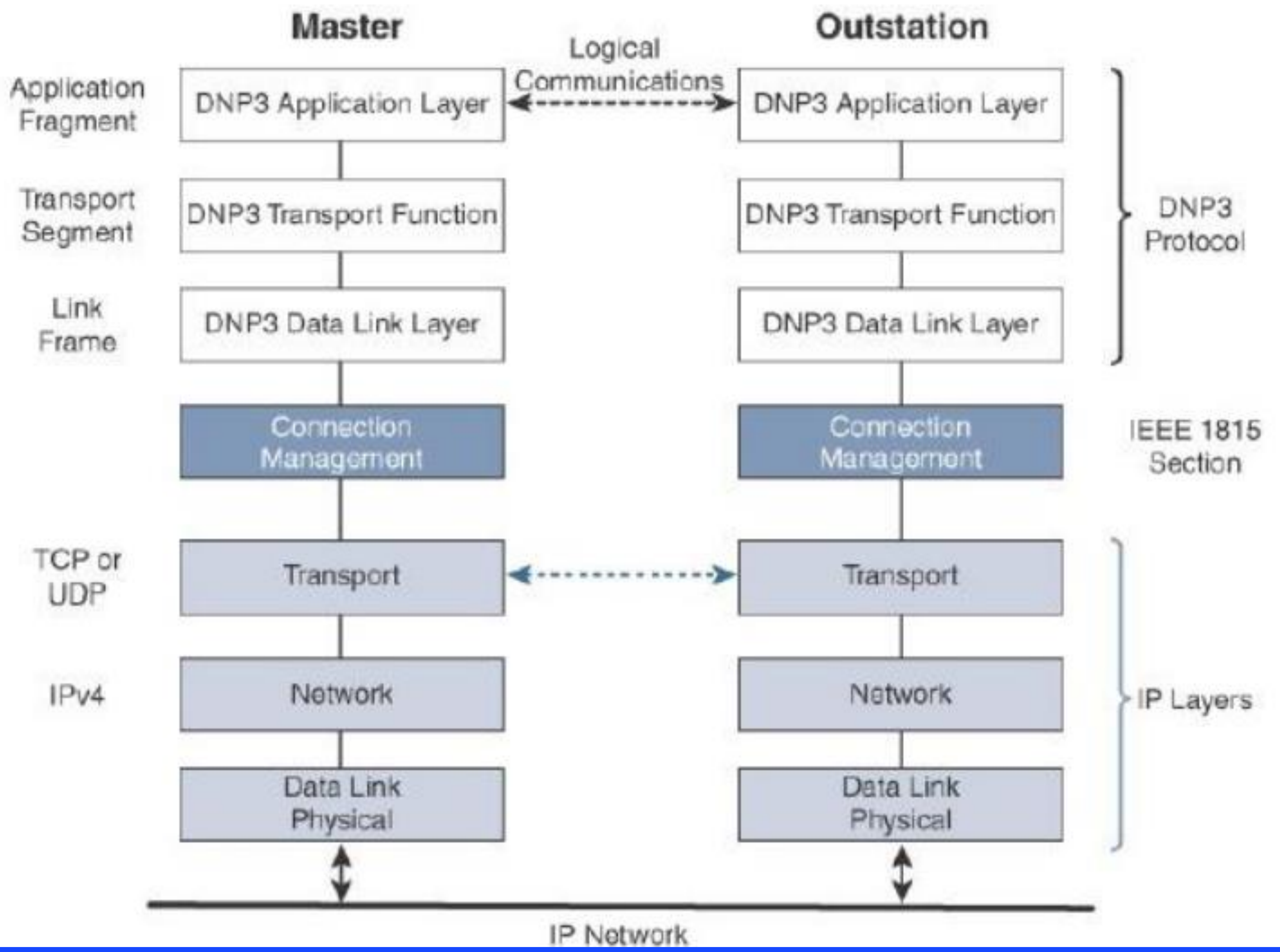

Application A
Application B
Application C
API
IOT Data Broker
Temperature Data in
Standardized Format
Sensor X
Decoder
Sensor Y
Decoder
Sensor B
Decoder
Sensor A
Decoder
Sensor Z
Decoder
Sensor X
Temp = 2 Byte Coding
Sensor Y
Temp = 4 Byte Coding
Sensor Z
Temp = 8 Byte Coding

IoT Application Transport Methods

- **Supervisory control and data acquisition (SCADA).**
- Designed decades ago, SCADA is an automation control system that was initially implemented without IP over serial links (such as RS-232 and RS-485), before being adapted to Ethernet and IPv4.
- SCADA - A Little Background on SCADA 91
- SCADA networking protocols, running directly over serial physical and data link layers.
- At a high level, SCADA systems collect sensor data and telemetry from remote devices, and to control them.
- SCADA systems allow global, real-time, data-driven decisions to be made about how to improve business processes

IoT Application Transport Methods

- **Supervisory control and data acquisition (SCADA).**
- SCADA - Adapting SCADA for IP
- Like many of the other SCADA protocols, 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, and a slave is a remote device with computing resources found in a location such as a substation.
- DNP3 refers to slaves specifically as outstations



Protocol Stack for Transporting Serial DNP3 SCADA over IP

IoT Application Transport Methods

- **Supervisory control and data acquisition (SCADA).**
- SCADA - Adapting SCADA for IP
- 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

IoT Application Transport Methods

- **Supervisory control and data acquisition (SCADA).**
- SCADA - Adapting SCADA for IP
- End-to-end native IP support is preferred, in the case of DNP3.
- Otherwise, transport of the original serial protocol over IP can be achieved either by tunneling 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.
- **A raw socket connection simply denotes that the serial data is being packaged directly into a TCP or UDP transport.**
- A socket is a standard application programming interface (API) composed of an IP address and a TCP or UDP port that is used to access network devices over an IP network.

IoT Application Transport Methods

- **Generic Web-Based Protocols**
- 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.
- 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
- In case of constrained networks the embedded web server software with advanced features are now implemented with very little memory (in the range of 10KB).

IoT Application Transport Methods

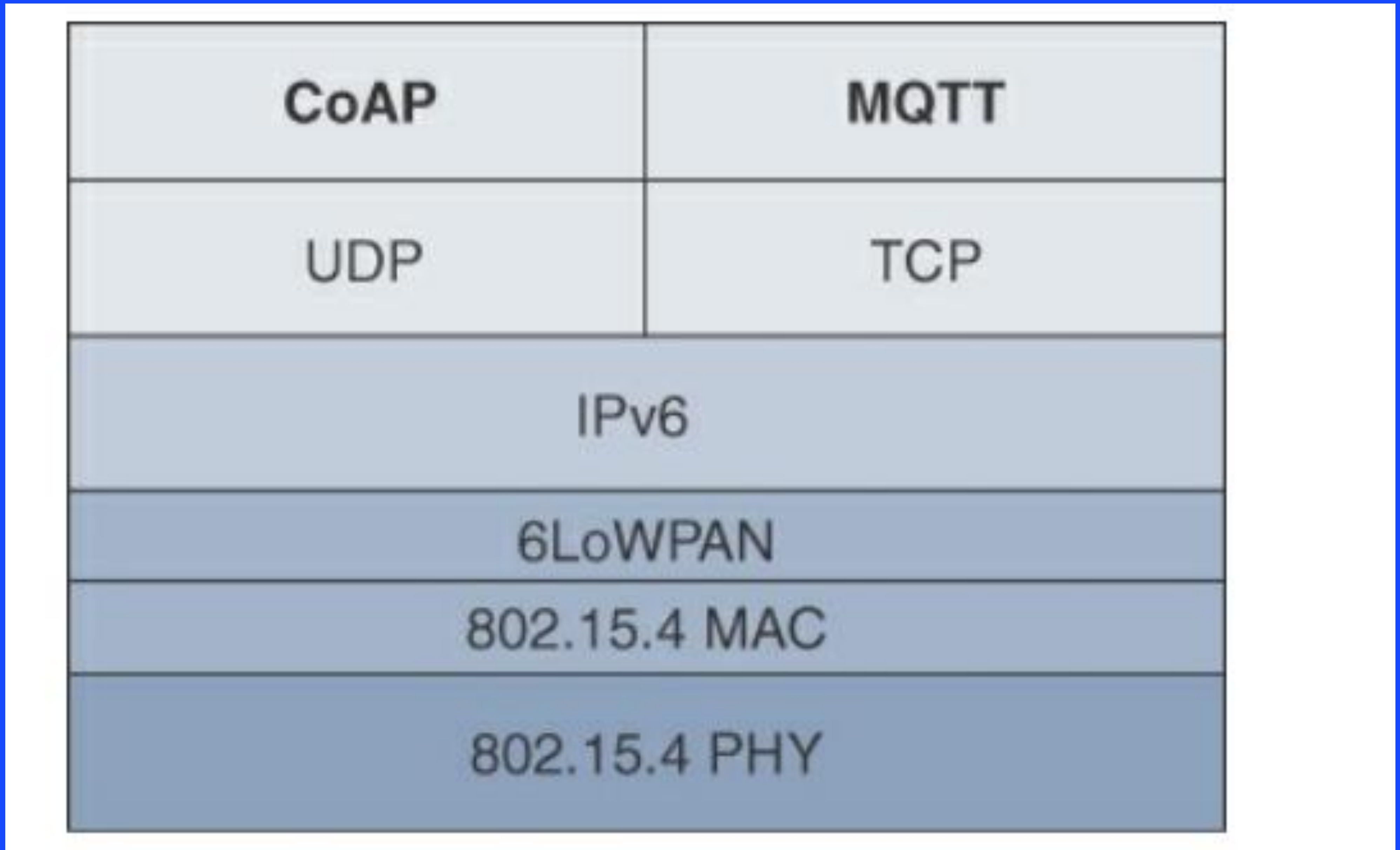
- **Generic Web-Based Protocols**
- IoT devices that only push data to an application may need to implement web services on the client side.
- For example,
- • an Ethernet- or Wi-Fi-based weather station reporting data to a weather map application.
- • The HTTP client side only initiates connections and does not accept incoming ones.
- • Some IoT devices, such as a video surveillance camera, may have web services implemented on the server side.
- • Interactions between real-time communication tools powering collaborative applications, such as voice and video, instant messaging, chat rooms, and IoT devices, are also emerging.

• Extensible Messaging and Presence Protocol (XMPP).

IoT Application Transport Methods

- 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 new lightweight protocols that are better suited to large numbers of constrained nodes and networks.
- Two of the most popular protocols are
- CoAP
- MQTT

Application Protocols for IoT-CoAP



Protocol stack

IoT Application Transport Methods

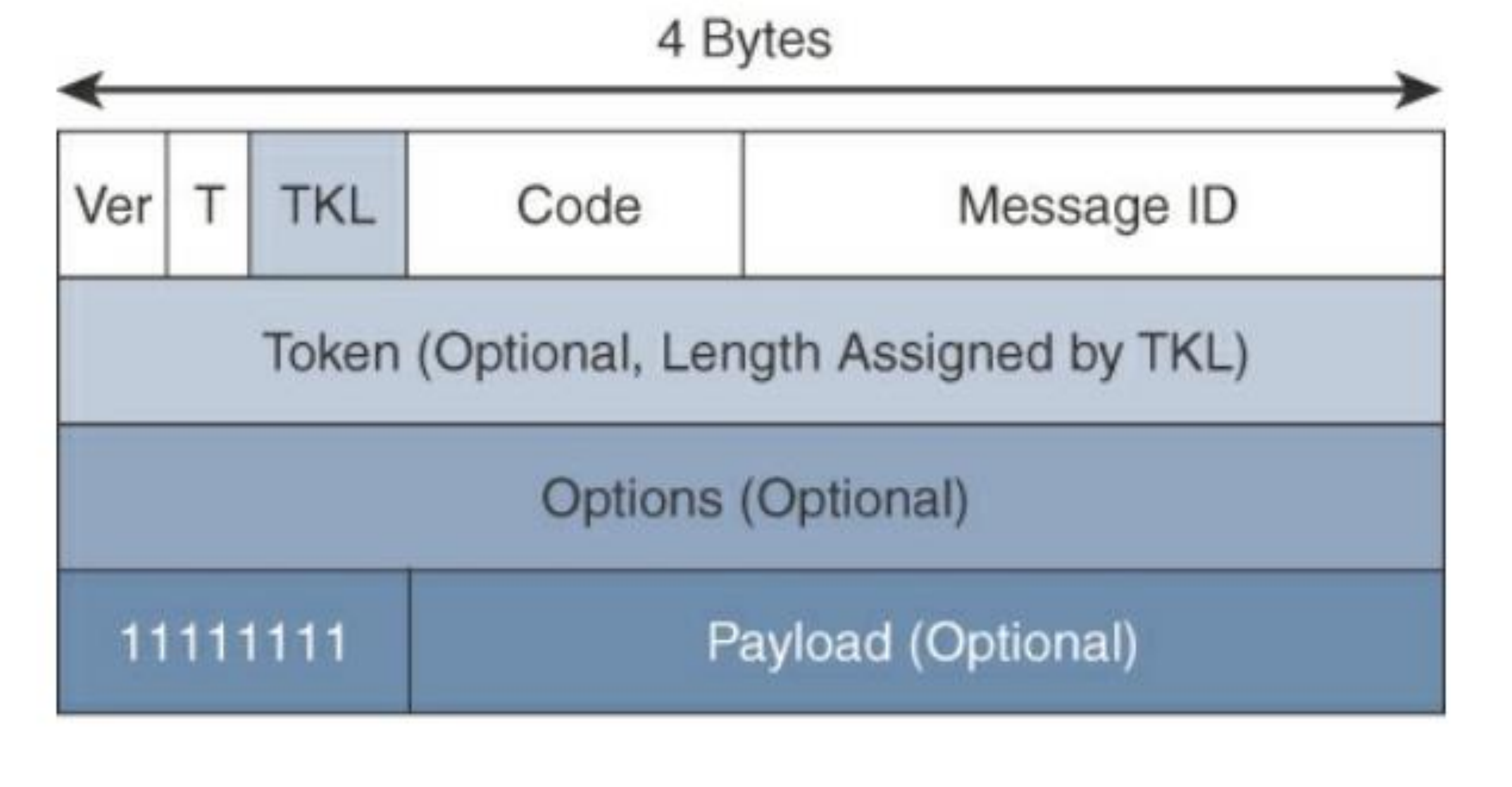
- IoT Application Layer Protocols
- CoAP and MQTT are at the top of this sample IoT stack, based on an IEEE 802.15.4 mesh network.
- CoAP deployed over UDP and MQTT running over TCP

CoAP

- Constrained Application Protocol (CoAP)
- 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 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).
- CoAP over Short Message Service (SMS) as defined in Open Mobile Alliance for Lightweight Machine-to-Machine (LWM2M) for IoT device management.

CoAP

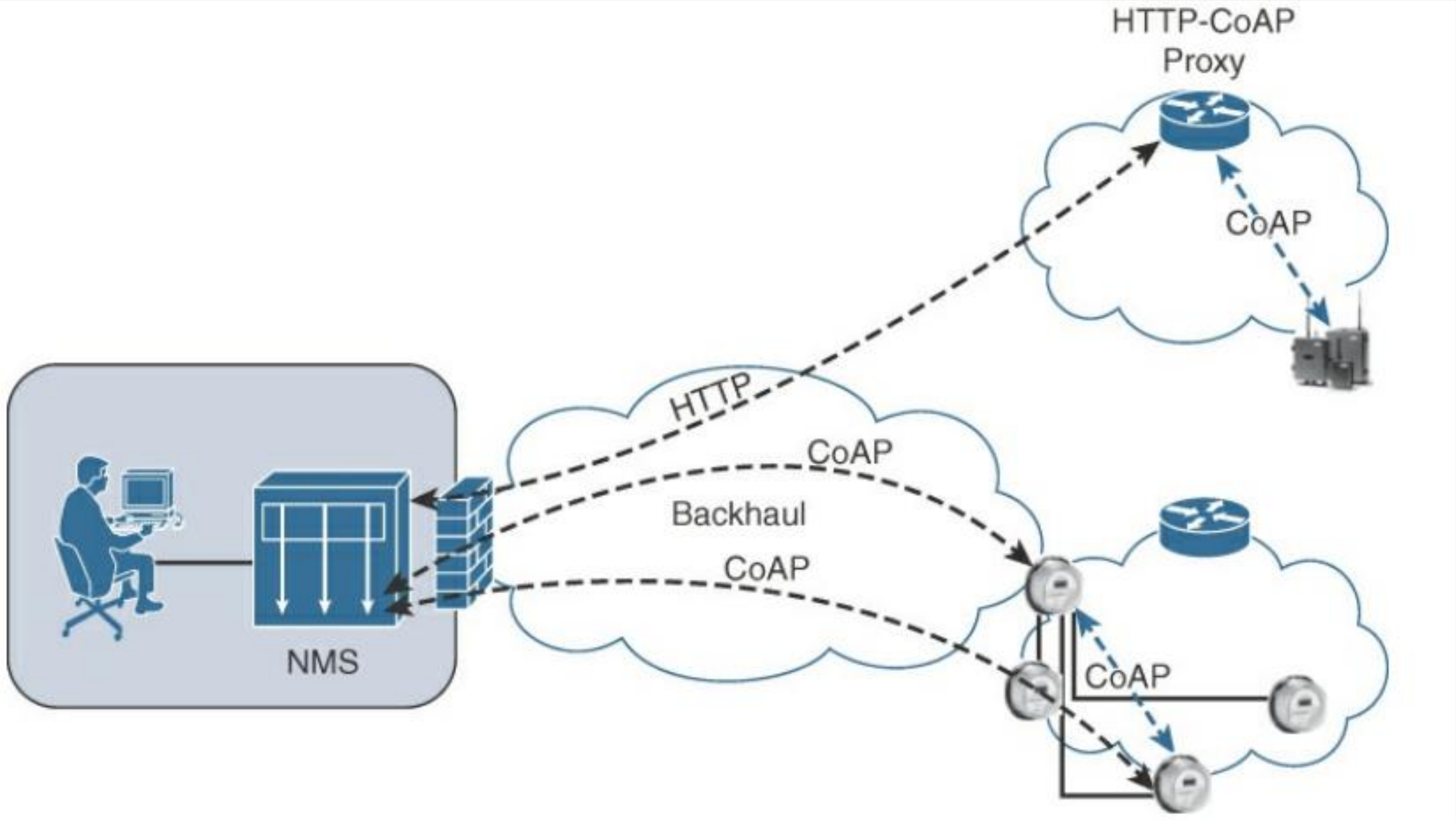
- A CoAP message is composed of
- Short fixed-length Header field (4 bytes),
- Variable-length but mandatory Token field (0–8 bytes),
- Options fields if necessary, and the Payload field.



CoAP

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.

CoAP



CoAP

- CoAP communications across an IoT infrastructure can take various paths.
- 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.
- As 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.

CoAP

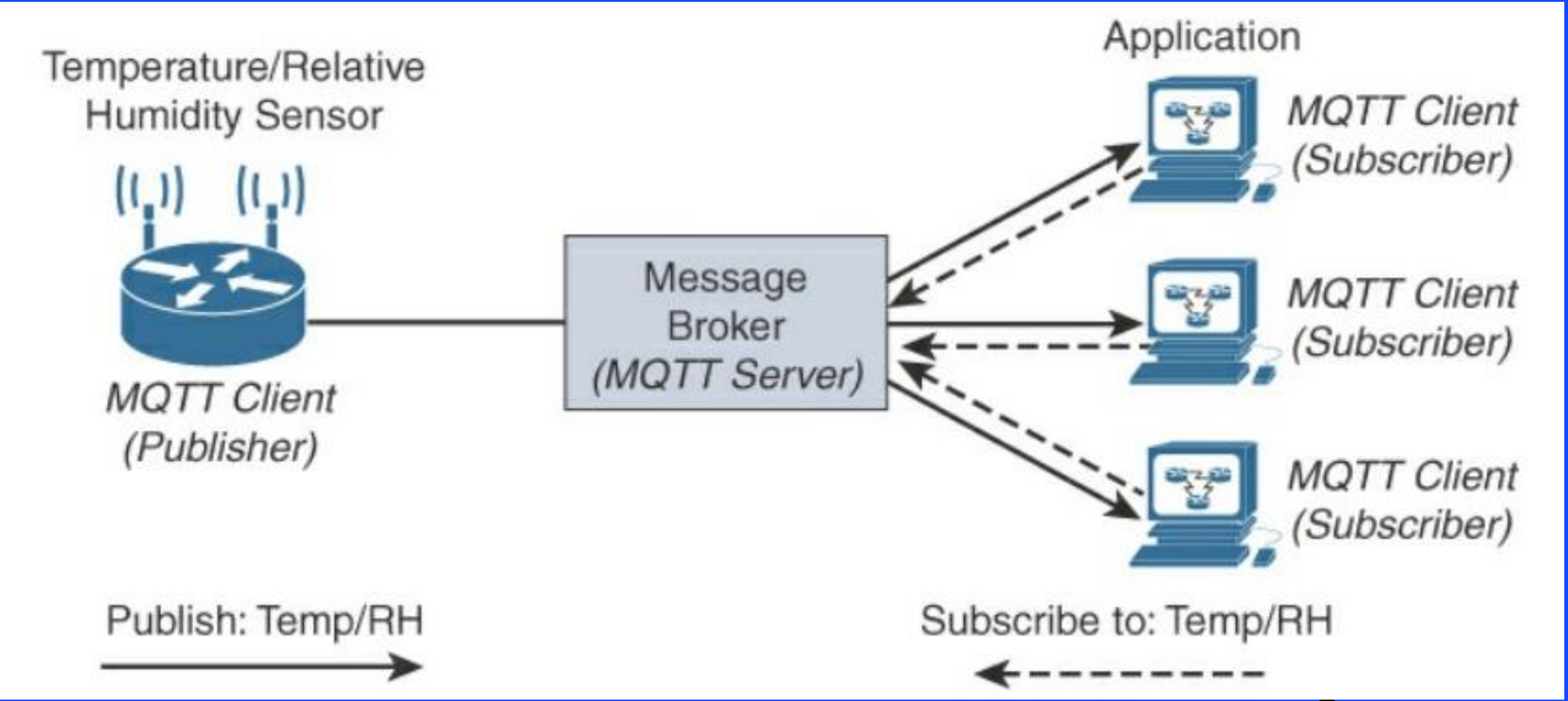
- Just like HTTP, CoAP is based on the REST architecture, but with a “thing” acting as both the client and the server.
- Through the exchange of asynchronous messages, a client requests an action via a method code on a server resource.
- A uniform resource identifier (URI) localized on the server identifies 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.

Application Protocols for IoT - MQTT

MQTT

- **Message Queuing Telemetry Transport (MQTT) :**
- 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,



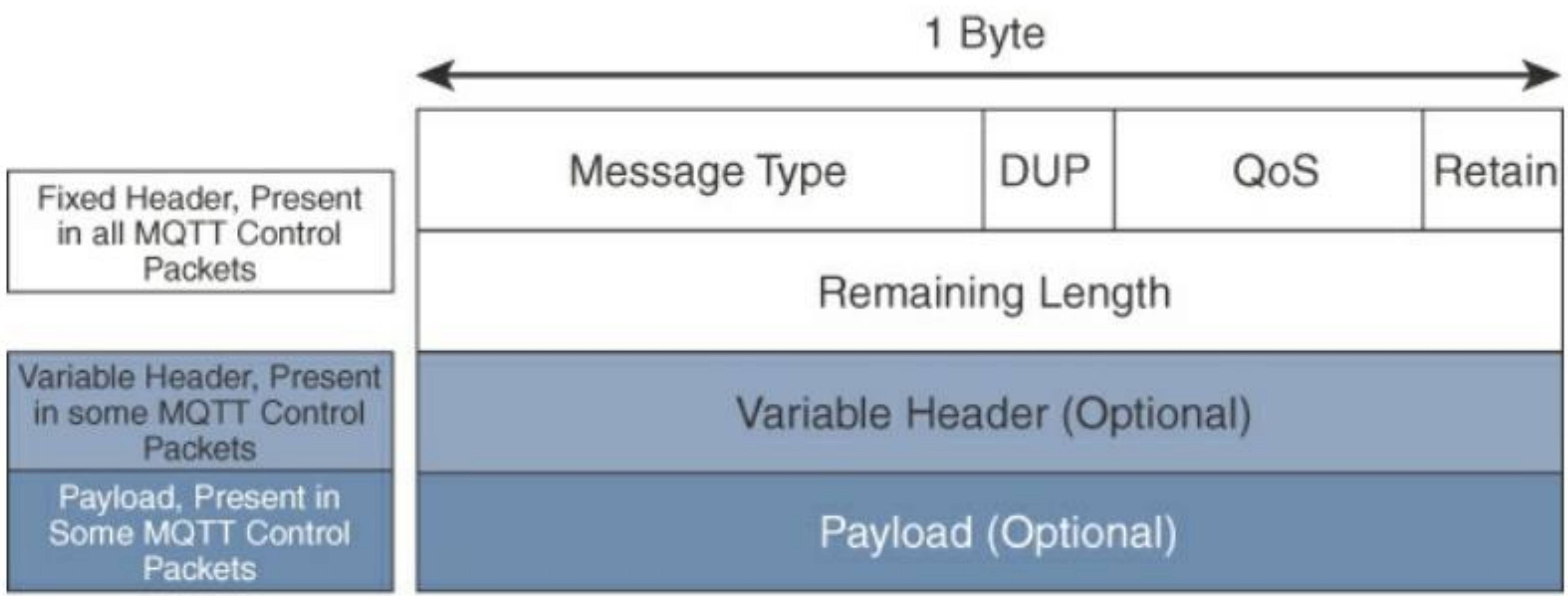
MQTT Publish/Subscribe Framework

MQTT

- 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 Figure the MQTT client on the left side is a temperature (Temp) and relative humidity (RH) sensor that publishes its Temp/RH data.
  - 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 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.

MQTT

- ❖ 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.
- ❖ Compared to the CoAP message, MQTT contains a smaller header of 2 bytes compared to 4 bytes for CoAP



MQTT Message Format

MQTT

- MQTT is a lightweight protocol because each control packet consists of a 2-byte fixed header with optional variable header fields and optional payload.
- 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

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

MQTT

- 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.
- The QoS header field allows for the selection of three different QoS levels.
- The next field is the Retain flag.
  - Only found in a PUBLISH message, the 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.
- The last mandatory field in the MQTT message header is Remaining Length.
  - This field specifies the number of bytes in the MQTT packet following this field.

IOT

MQTT

- MQTT sessions between each client and server consist of four phases: **session establishment, authentication, data exchange, and session termination**.
- Each client connecting to a server has a unique client ID, which allows the identification of the MQTT session between both parties.
- When the server is delivering an application message to more than one client, each client is treated independently.
- Subscriptions to resources generate SUBSCRIBE/SUBACK control packets, while unsubscription is performed through the exchange of UNSUBSCRIBE/UNSUBACK control packets.
- Graceful termination of a connection is done through a DISCONNECT control packet, which also offers the capability for a client to reconnect by re-sending its client ID to resume the operations

MQTT

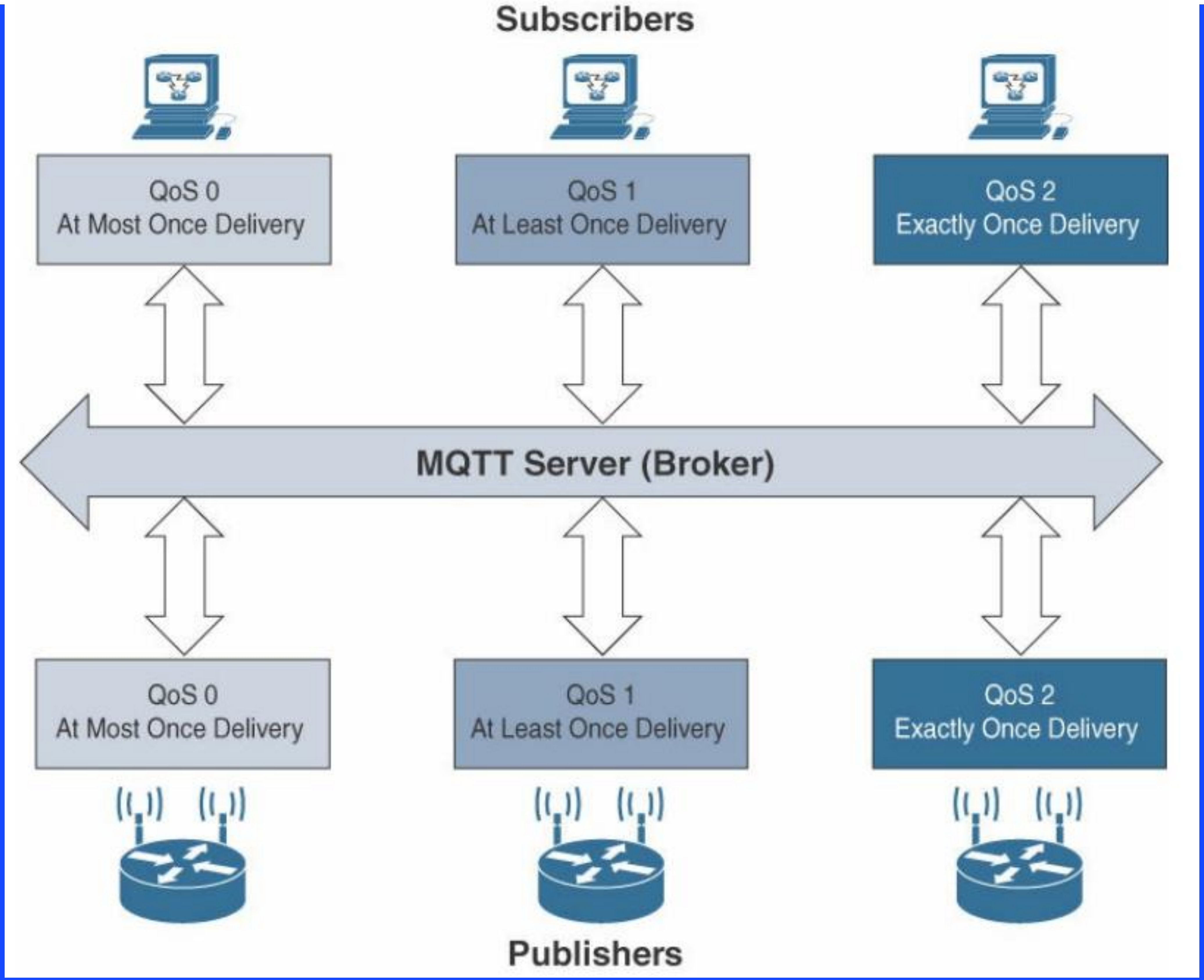
- PINGREQ/PINGRESP control packets are used to validate the connections between the client and server.
- Similar to ICMP pings that are part of IP, they are a sort of keepalive that helps to maintain and check the TCP session.
- Securing MQTT connections through TLS is considered optional because it calls for more resources on constrained nodes.
- When TLS is not used, the client sends a clear-text username and password during the connection initiation.
- The MQTT protocol offers three levels of quality of service (QoS).
- QoS for MQTT is implemented when exchanging application messages with publishers or subscribers, and it is different from the IP QoS that most people are familiar with.
- The delivery protocol is symmetric.

MQTT

- These 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.

MQTT

- **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

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

Comparison Between CoAP and MQTT

THANK YOU!