Module-3
Chapter 5

IP AS THE IOT NETWORK LAYER

- **This chapter includes..**

- **The Business Case for IP**
- **The Need for Optimization**
- **Optimizing IP for IoT**
- **Profiles and Compliances**

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.

- 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.

- This is how and why the **Internet Protocol (IP)** suite started playing a key architectural role in the early 1990s.

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.

- The key advantages of the IP suite for the Internet of Things are as follows:
- **Open and standards-based**
- **Versatile**
- **Ubiquitous**
- **Scalable**
- **Manageable and highly secure**
- **Stable and Resilient**
- **Consumer's Market Adoption**
- **The innovation factor**

- **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 inter changeability and interoperability, security, and management.
  - This calls for implementation, validation, and deployment of open, standards-based solutions.

- **Versatile**
  - A large spectrum of access technologies is available to offer connectivity of “things” in the last mile.
  - Additional protocols and technologies are also used to transport IoT data through backhaul links and in the data center.

- **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.
  - In fact, IP is the most pervasive protocol which is supported across the various IoT solutions and industry verticals.

- **Scalable**
  - As the common protocol of the Internet, IP has been massively deployed and tested for robust scalability.
  - 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.
  - Well known network and security management tools are easily leveraged with an IP network layer.

- **Stable and Resilient**
  - IP has a large and well-established knowledge base and, more importantly, it has been used for years in critical infrastructures, such as financial and defense networks.
  - 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.

- 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.

- **The innovation factor**
  - The past two decades have largely established the adoption of IP as a factor for increased innovation.
  - IP is a standards-based protocol that is ubiquitous, scalable, versatile, and stable. Network services such as naming, time distribution, traffic prioritization, isolation, and so on are well known and developed techniques that can be leveraged with IP.

Adoption or Adaptation of the Internet Protocol

- The implementation of 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.
- Multiprotocol routers were needed to handle this proliferation of network layer protocols.
- 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:

  i. **Adaptation** : means application layered gateways (ALGs) must be implemented to ensure the translation between non-IP and IP layers.

  ii. **Adoption** : involves replacing all non-IP layers with their IP layer counterparts, simplifying the deployment model and operations.

IP Adaptation and Adoption applied to IoT last mile connectivity

- In the industrial and manufacturing sector, there has been a move toward IP adoption. Solutions and product lifecycles in this space are spread over 10+ years, and 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 are typical examples of vertical market deployments that operate both the IP adaptation model and the adoption model.

- With the IP adoption model, SCADA devices are attached via Ethernet to switches and routers forwarding their IPv4 traffic.

- Another example is a ZigBee solution that runs a non-IP stack between devices and a ZigBee gateway that forwards traffic to an application server.

- The following factors has to be taken into consideration to determine which model is best suited for last-mile connectivity:

  - **Bidirectional versus unidirectional data flow**
    - While bidirectional communications are generally expected, some last-mile technologies offer optimization for unidirectional communication.
    - For ex : different classes of IoT devices, as defined in RFC 7228.
    - 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

- **Overhead for last-mile communications paths**
  - IP adoption implies a layered architecture with a per-packet overhead that varies depending on the IP version.
  - If the data to be forwarded by a device is infrequent and only a few bytes, we can potentially have more header overhead than device data.
  - we need to decide whether the IP adoption model is necessary and, if it is, how it can be optimized.

Data flow model

- One benefit of the IP adoption model is the end-to-end nature of communications.
- 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-to-end” 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.

- **Network Diversity**
  - One of the drawbacks of the adaptation model is a general dependency on single PHY and MAC layers.
  - For example, ZigBee devices must only be deployed in ZigBee network islands. This same restriction holds for ITU G.9903 G3-PLC nodes.
  - Therefore, a deployment must consider which applications have to run on the gateway connecting these islands and the rest of the world.

The Need for Optimization

- There lot of challenges in building the IoT solutions based on IP.
- In addition to coping with the integration of non-IP devices, we may need to deal with the limits at the device and network levels that IoT often imposes.
- Therefore, optimizations are needed at various layers of the IP stack to handle the restrictions that are present in IoT networks.
- Let us see why optimization is necessary in IP based IoT solutions:

Constrained Nodes

- IoT is a platform, where 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.
- 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 costs of computing power, memory, storage resources, and power consumption are generally decreasing. At the same time, networking technologies continue to improve and offer more bandwidth and reliability.

Constrained Networks

- In the early years of the Internet, network bandwidth capacity was restrained due to technical limitations.
- A constrained network can have high latency and a high potential for packet loss.
- Constrained networks have unique characteristics and requirements. In contrast with typical IP networks, where highly stable and fast links are available, constrained networks are limited by low-power, low-bandwidth links.
- Finally, we have to consider the power consumption in battery-powered nodes. Any failure or verbose control plane protocol may reduce the lifetime of the batteries.

IP Versions

- The IETF has been working on transitioning the Internet from IP version 4 to IP version 6.
- Today, both versions of IP run over the Internet, but most traffic is still IPv4 based.
- Techniques such as tunnelling and translation need to be employed in IoT solutions to ensure interoperability between IPv4 and IPv6.

- Most often these factors include a legacy protocol or technology that supports only IPv4. Newer technologies and protocols almost always support both IP versions.

- The following are some of the main factors applicable to IPv4 and IPv6 support in an IoT solution

➢ Application Protocol

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

- For ex : SCADA protocols such as DNP3/IP (IEEE 1815), Modbus TCP, or the IEC 60870- 5-104 standards are specified only for IPv4.

- So, there are no known production implementations by vendors of these protocols over IPv6 today.

- For IoT devices with application protocols defined by the IETF, such as HTTP/HTTPS, CoAP, MQTT, and XMPP, both IP versions are supported.

➢ 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.
- For the first three generations of data services—GPRS, Edge, and 3G—IPv4 is the base protocol version.
- Consequently, if IPv6 is used with these generations, it must be tunneled over IPv4.
- On 4G/LTE networks, data services can use IPv4 or IPv6 as a base protocol, depending on the provider.

➢ Serial Communications

- Many legacy devices in certain industries, such as manufacturing and utilities, communicate through serial lines.
- In the past, communicating this serial data over any sort of distance could be handled by an analog modem connection.
- However, as service provider support for analog line services has declined, the solution for communicating with these legacy devices has been to use local connections.

- To make this work, you connect the serial port of the legacy device to a nearby serial port on a piece of communications equipment, typically a router.

- This local router then forwards the serial traffic over IP to the central server for processing.

IPv6 Adaptation Layer

- IPv6-only adaptation layers for some physical and data link layers for recently standardized IoT protocols support only IPv6.

- While 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 ITUG.9903 (Narrowband Power Line Communications only have an IPv6 adaptation layer specified.

- This means that any device implementing a technology that requires an IPv6 adaptation layer must communicate over an IPv6-only subnetwork.

- This is reinforced by the IETF routing protocol for LLNs, RPL, which is IPv6 only.

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.

- Figure 5.1 highlights the TCP/IP layers where optimization is applied.

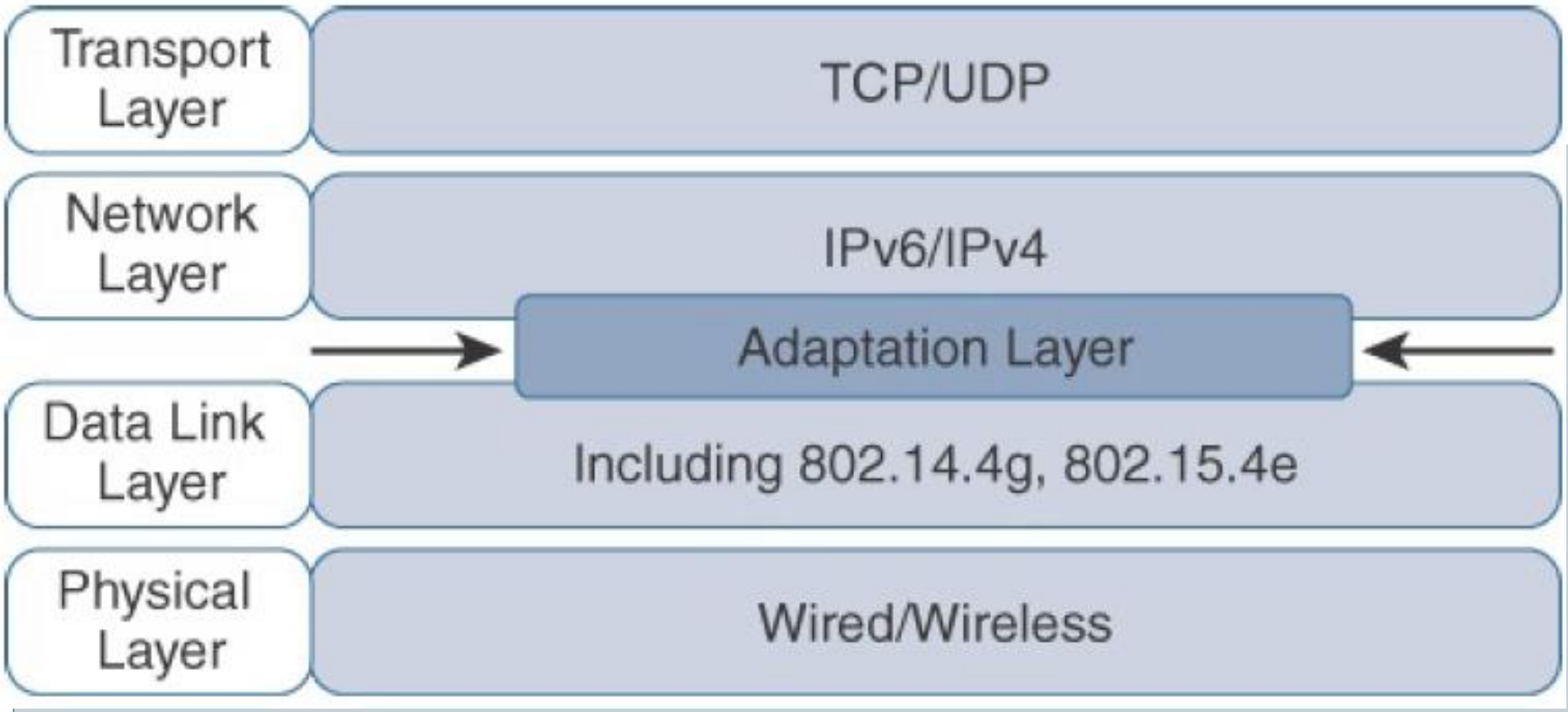

Figure 5.1 : Optimizing IP for IoT Using an Adaptation Layer

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***.

- 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.

- 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.

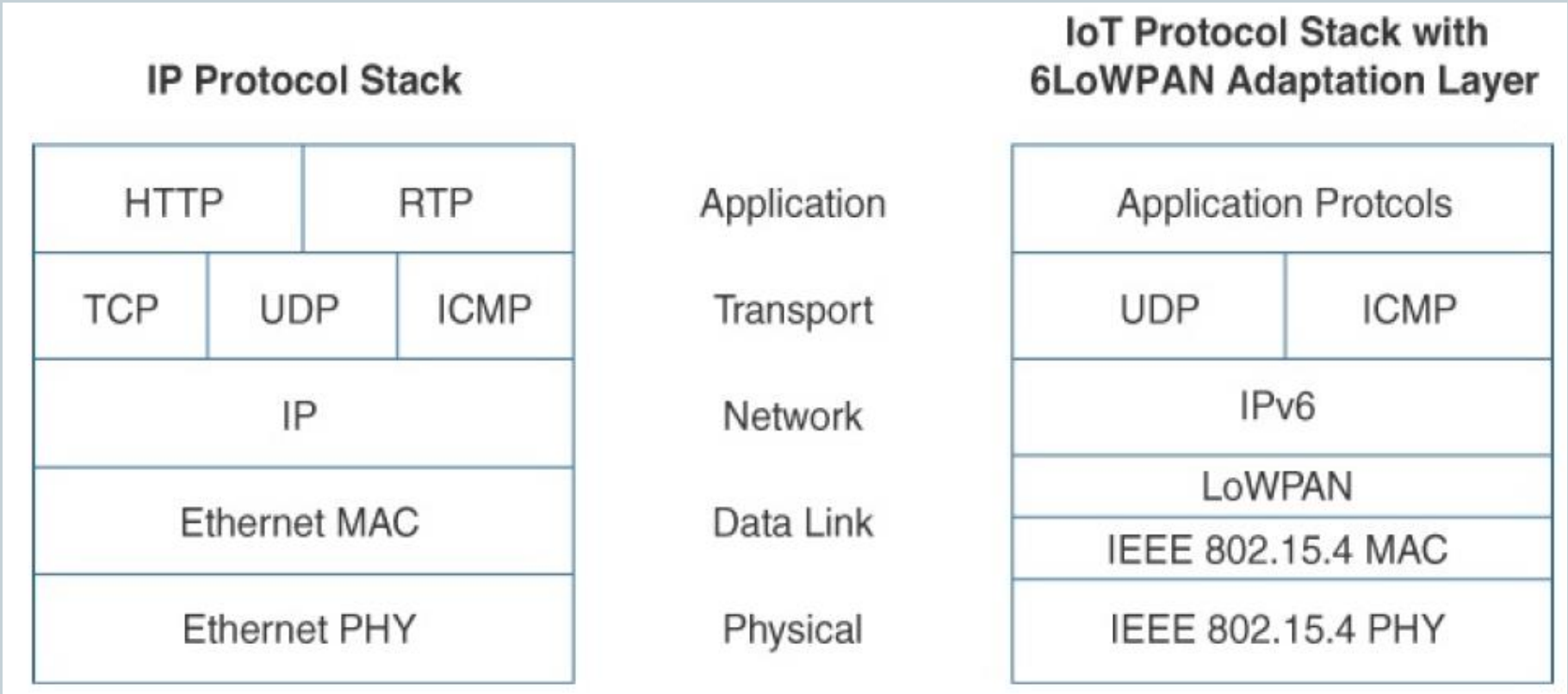

Figure 5.2: Comparison of an IoT Protocol Stack Utilizing 6LoWPAN and an IP Protocol Stack

- The 6LoWPAN working group 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

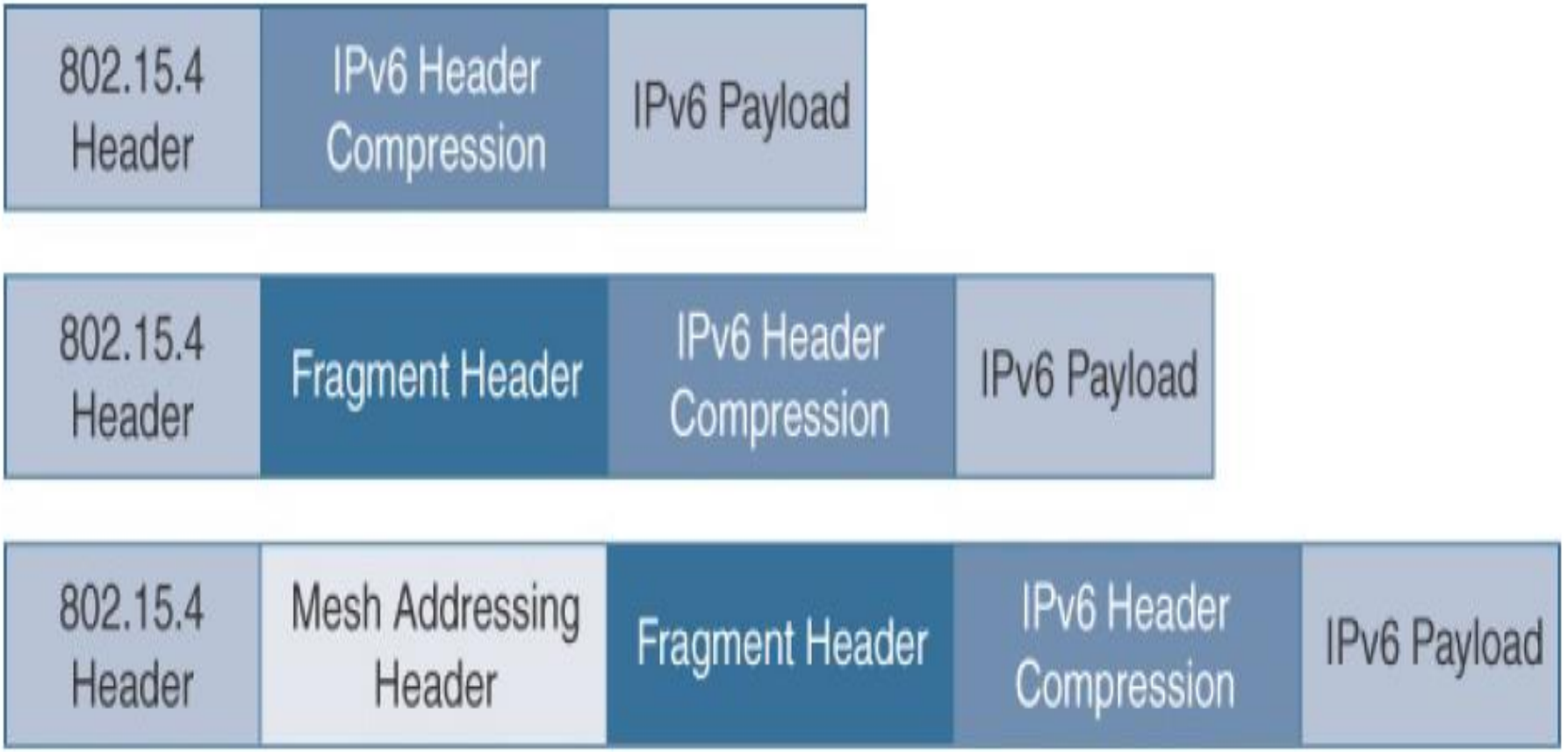

Figure 5.3: 6LoWPAN Header Stacks

- **Header Compression**

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

- 6LoWPAN header compression is stateless, and conceptually it is not too complicated.

- Number of factors affect the amount of compression, such as implementation of RFC 4944 versus RFC 6922, whether UDP is included, and various IPv6 addressing scenarios.

- This capability shrinks the size of IPv6's 40-byte headers and User Datagram Protocol's (UDP's) 8-byte headers down as low as 6 bytes combined in some cases.

- The header compression for 6LoWPAN is only defined for an IPv6 header and not IPv4. The 6LoWPAN protocol does not support IPv4, and, in fact, there is no standardized IPv4 adaptation layer for IEEE 802.15.4.

- Figure 5.4 highlights an example that shows the amount of reduction that is possible with 6LoWPAN header compression.

- At the top of Figure 5.4, we see 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.

  - Notice that 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.

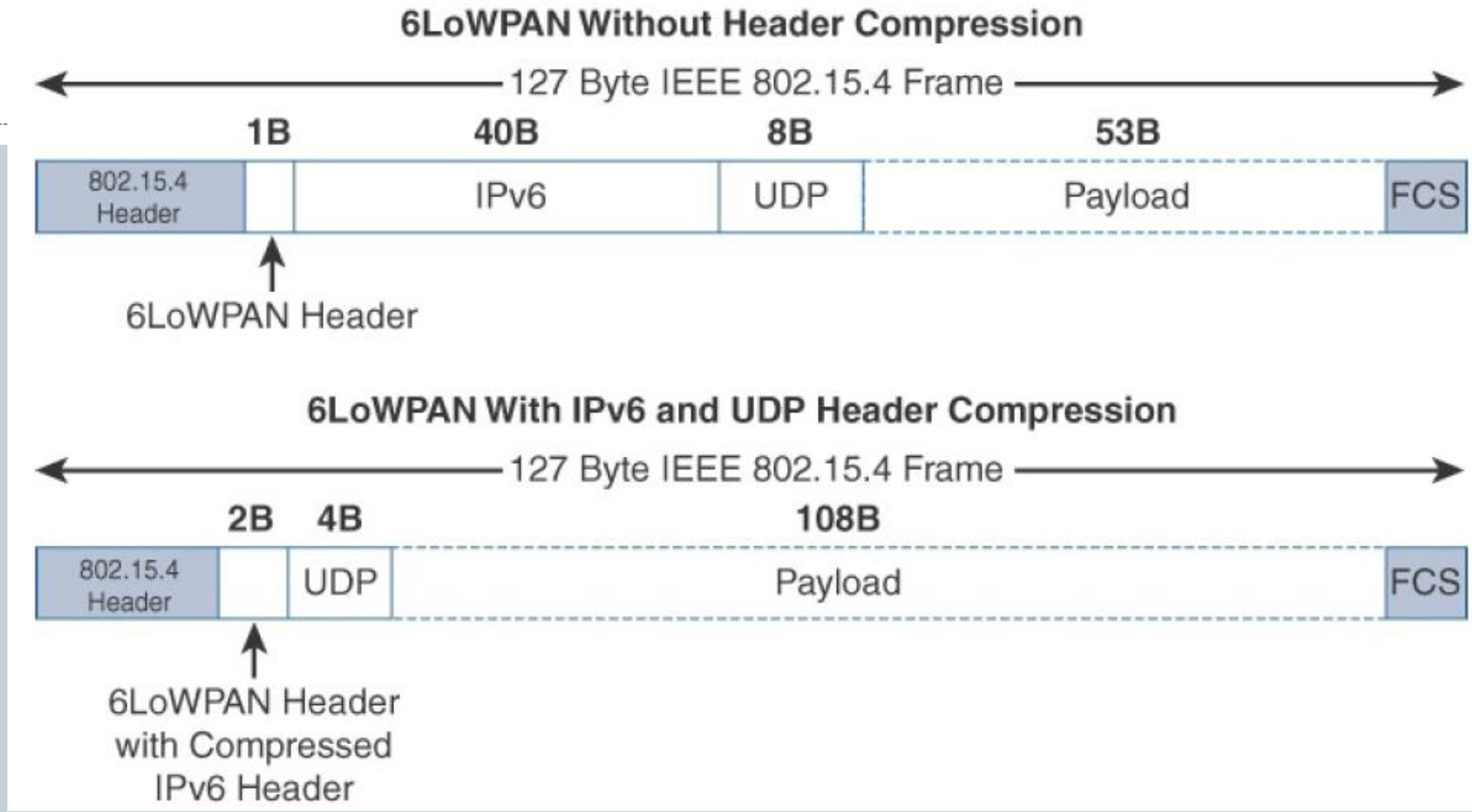

Figure 5.4: 6LoWPAN Header Compression

- The bottom half of Figure 5.4 shows a frame where header compression has been enabled for a best-case scenario.
- 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.

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.
- Large IPv6 packets must be fragmented across multiple 802.15.4 frames at Layer 2.
- The fragment header utilized by 6LoWPAN is composed of three primary fields:

i. Datagram Size

ii. Datagram Tag and

iii. Datagram offset

Figure 5.5 provides an overview of a **6LoWPAN fragmentation header.**

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.

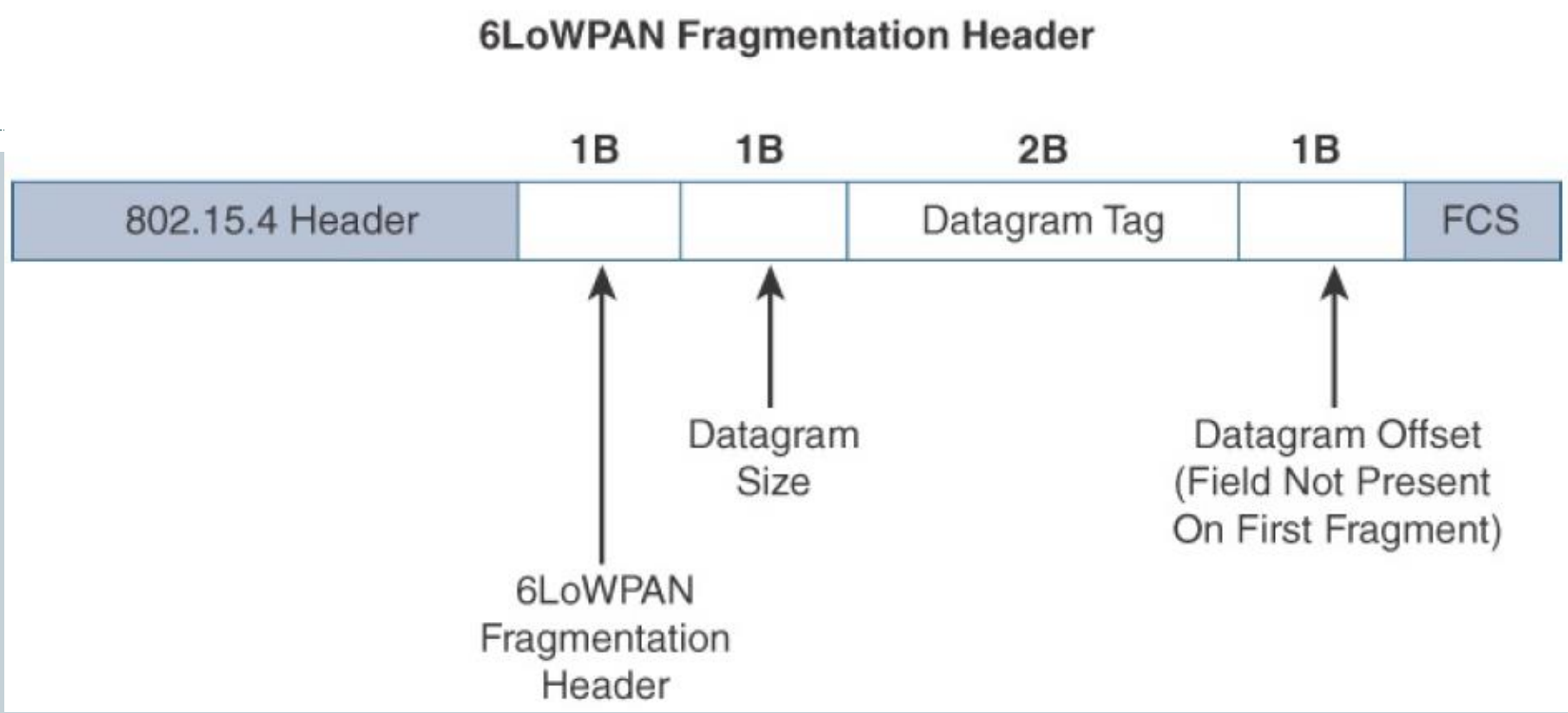

Figure 5.5: 6LoWPAN Fragmentation Header

- The 1-byte **Datagram Size** field specifies the total size of the unfragmented payload.
- **Datagram Tag** identifies the set of fragments for a payload.
  - the Datagram Offset field delineates how far into a payload a particular fragment occurs.

- Also, 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. The remainder of the fragments have a 5-byte header field so that the appropriate offset can be specified.

- **Mesh Addressing**

- The purpose of the **6LoWPAN mesh addressing** function is to forward packets over multiple hops.

- **Three fields** are defined for this header: Hop Limit, Source Address, and Destination Address.

- 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 **droppe**d and no longer forwarded.
- The **Source Address and Destination Address** fields for mesh addressing are IEEE 802.15.4 addresses indicating the **endpoints of an IP hop.**
- Figure 5.6 details the 6LoWPAN mesh addressing header fields.

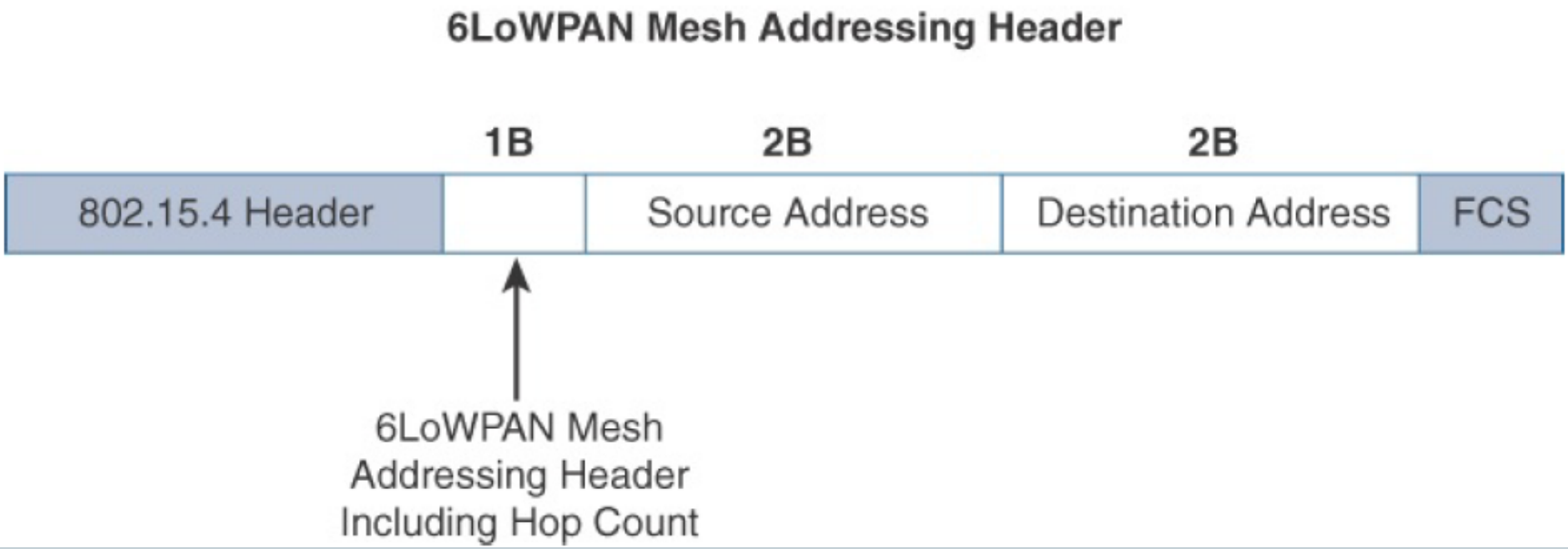

Figure 5.6 : 6LoWPAN Mesh Addressing Header

Mesh-Under Versus Mesh-Over Routing

- For network technologies such as **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.

- With the **first option**, mesh-under, the routing of packets is handled at the 6LoWPAN adaptation layer.

- The other option, known as **"mesh-over"** or **"route-over,"** utilizes IP routing for getting packets to their destination.

- With **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.

- In mesh-over or route-over scenarios, IP Layer 33 routing is utilized for computing reachability and then getting packets forwarded to their destination, either inside or outside the mesh domain.

6Lo Working Group

- With the work of the 6LoWPAN working group completed, the 6Lo working group seeks to expand on this completed work with a focus on IPv6 connectivity over constrained-node networks.

- While the 6LoWPAN working group initially focused its optimizations on IEEE 802.15.4 LLNs, standardizing IPv6 over other link layer technologies is still needed.

- Therefore, the charter of the 6Lo working group, now called the IPv6 over Networks of Resource-Constrained Nodes, is to facilitate the IPv6 connectivity over constrained-node networks.

- In particular, this 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**
- **Optimizations that are applicable to more than one adaptation layer specification**
- **Informational and maintenance publications needed for the IETF specifications in this area**

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, with direct inheritance from other standards, such as WirelessHART and ISA100.11a.

- Devices implementing IEEE 802.15.4e TSCH communicate by following a Time Division Multiple Access (TDMA) schedule.

- To standardize IPv6 over the TSCH mode of IEEE 802.15.4e (known as 6TiSCH), the IETF formed the 6TiSCH working group.

- Figure 5.7 shows where 6top resides in relation to IEEE 802.15.4e, 6LoWPAN HC, and IPv6.

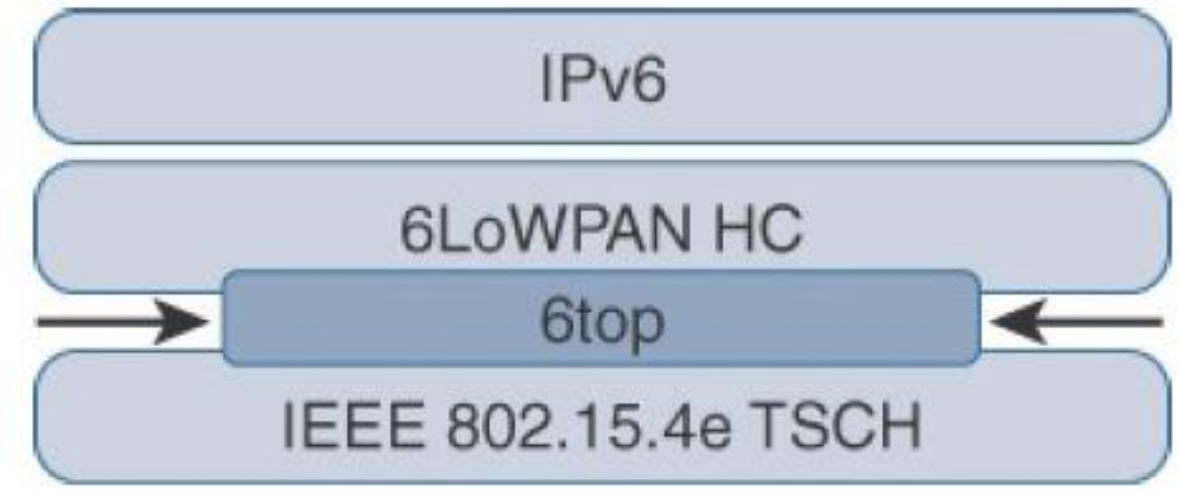

Figure 5.7 : Location of 6TiSCH's 6top Sublayer

- **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

- All nodes in the constrained network share a **fixed schedule**. Cells are shared, and nodes contend for slot access in a **slotted aloha manner.**
- **Slotted aloha** is a basic protocol for sending data using time slot boundaries when communicating over a shared medium.
- Static scheduling is a simple scheduling mechanism that can be used upon initial implementation or as a fallback in the case of network malfunction.
- The **drawback** with static scheduling is that nodes may expect a packet at any cell in the schedule. Therefore, energy is wasted idly listening across all cells.

- **Neighbor-to-neighbor scheduling**
  - A schedule is established that correlates with the observed number of transmissions between nodes.
  - Cells in this schedule can be added or deleted as traffic requirements and bandwidth needs change.

- **Remote monitoring and scheduling management**
  - Time slots and other resource allocation are handled by a management entity that can be multiple hops away.
  - The scheduling mechanism leverages 6top and even CoAP in some scenarios.
  - This **scheduling** mechanism provides quite a bit of **flexibility and control** in allocating cells for communication between nodes.

- **Hop-by-hop Scheduling**
  - A node reserves a path to a destination node multiple hops away by requesting the allocation of cells in a schedule at each intermediate node hop in the path.
  - The protocol that is used by a node to trigger this scheduling mechanism is not defined at this point.

- 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 upper-layer protocol.

i. Track Forwarding(TF)

- This is the **simplest and fastest forwarding model**. A "track" in this model is a unidirectional path between a source and a destination.
- This track is constructed by **pairing bundles of receive cells** in a schedule with a bundle of receive cells set to transmit.
- So, a frame received within a particular cell or cell bundle is switched to another cell or cell bundle.

ii. Fragment Forwarding(FF)

- This model takes advantage of **6LoWPAN fragmentation to build a Layer 2 forwarding table.**
- The **IPv6 packets can get fragmented at the 6LoWPAN sublayer** to handle the differences between IEEE 802.15.4 payload size and IPv6 MTU.
- Additional headers for RPL source route information can further contribute to the need for fragmentation.
- However, with FF, a mechanism is defined where the first fragment is routed based on the IPv6 header present.

iii. IPv6 Forwarding(6F)

- This model forwards traffic based on its **IPv6 routing table.**
- Flows of packets should be prioritized by **traditional QoS** (quality of service) and **RED (random early detection) operations.**
- QoS is a classification scheme for flows based on their priority, and **RED is a common congestion avoidance mechanism.**

RPL

- The 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.

- This new distance-vector routing protocol was named the IPv6 **Routing Protocol for Low Power and Lossy Networks** (RPL).

- The RPL specification was published as RFC 6550 by the RoLL working group.

- 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 next-hop destination based on the information contained in the IPv6 header.
- No information from the MAC-layer header is needed to perform next-hop determination.

- To cope with 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.

- 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.

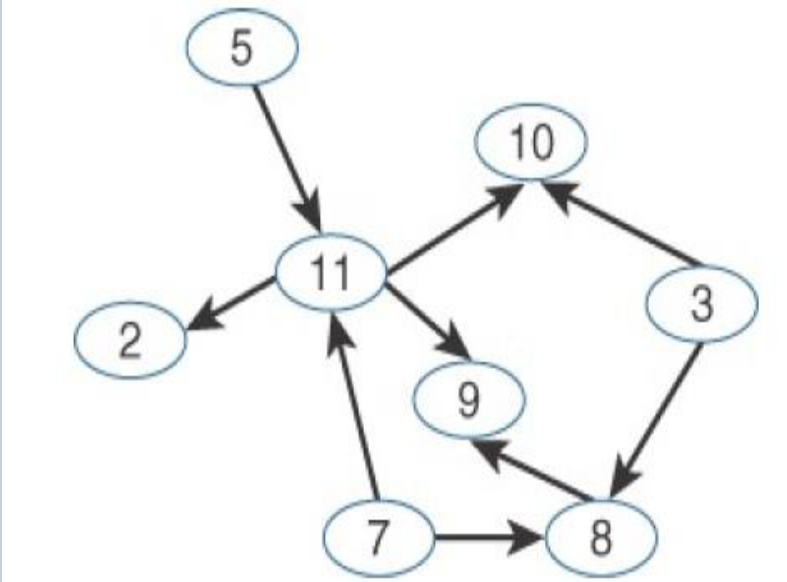

Figure 5.8 : Example of a Directed Acyclic Graph(DAG)

- 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.

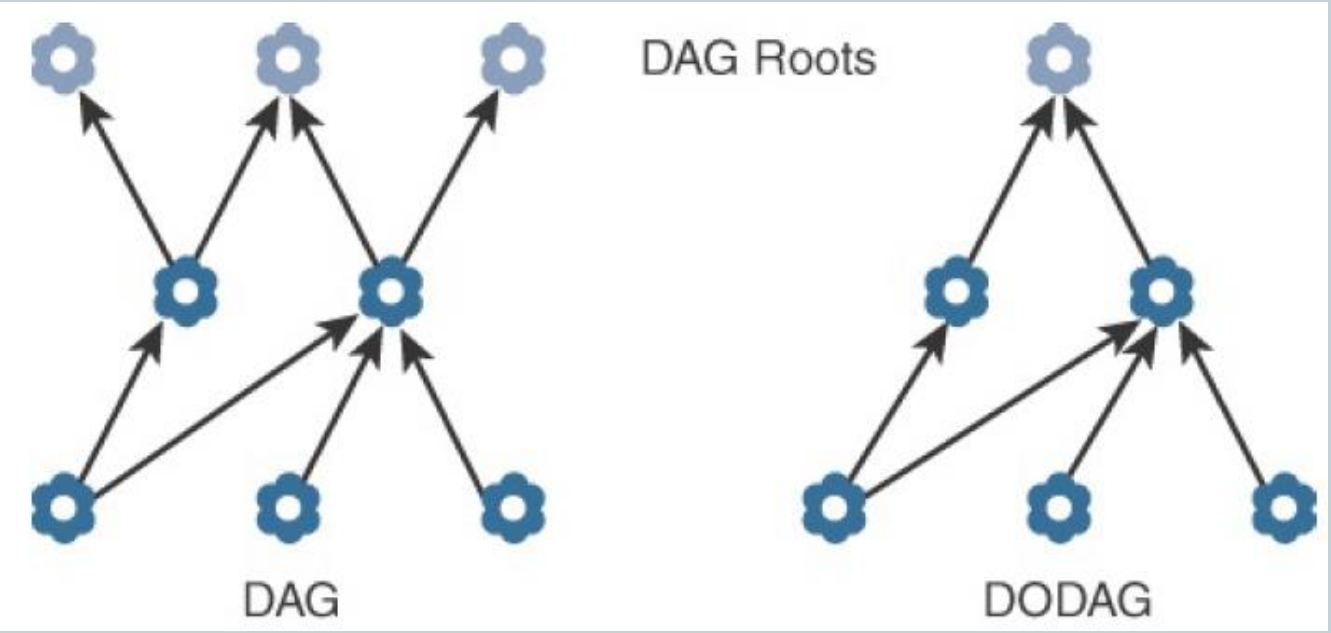

Figure 5.9 : DAG and DODAG comparison

- 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.
- 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.

- **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

- **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.

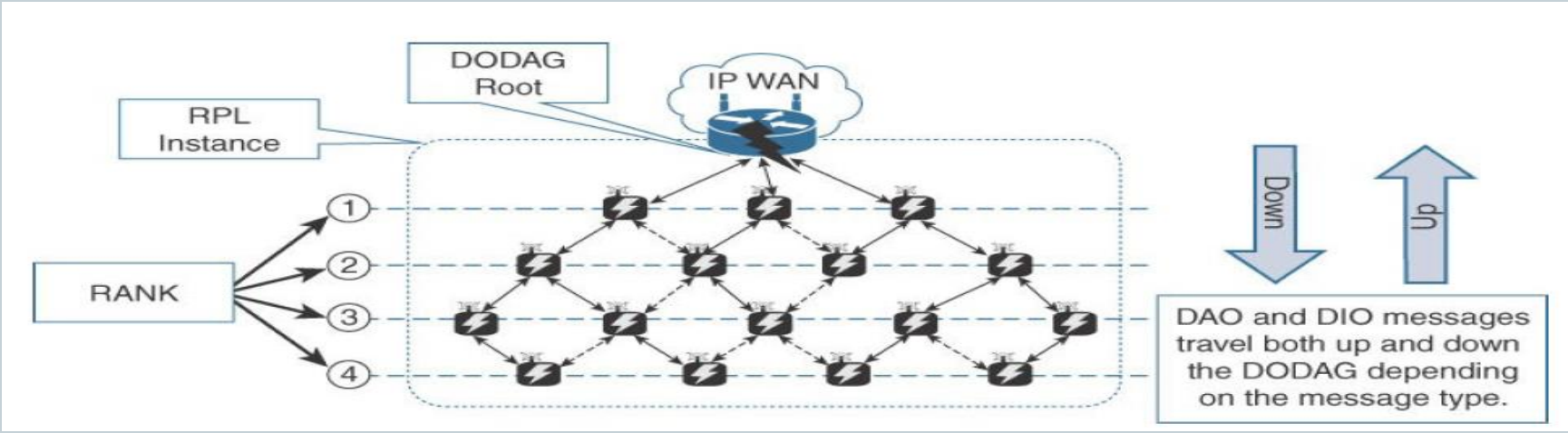

Figure 5.10 : RPL Overview

- **Objective Function(OF)**

- An **objective function (OF)** defines **how metrics** are used to select routes and establish a node's rank.

- Whenever a **node establishes its rank**, it simply sets the rank to the current **minimum METX** among its parents.

- **Rank**

- The rank is a rough approximation of how "close" a node is to the root and helps avoid routing loops and the count-to-infinity problem.

- Nodes can only increase their rank when receiving a DIO message with a larger version number.

- However, nodes may **decrease** their rank whenever they have established lower-cost routes.

- **RPL Headers**
- Specific network layer headers are defined for **datagrams** being forwarded within an RPL domain.
- The RPL option is carried in the **IPv6 Hop-by-Hop header**.
- The purpose of this header is to leverage **data-plane packets for loop detection in a RPL instance.**
- A border router or DODAG root inserts the SRH when specifying a source route to deliver datagrams to nodes downstream in the mesh network.

Metrics

- RPL defines a large and flexible set of new metrics and constraints for routing

- 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 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.

- **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.
  - 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.

- **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.

Authentication and Encryption on Constrained Nodes

- IoT security is a complex topic that often spawns discussions and debates across the industry.

- So it is worth mentioning here the IETF working groups that are focused on their security:

- **ACE**
- **DICE.**

.

ACE

- Much like the RoLL working group, the Authentication and Authorization for Constrained Environments (ACE) working group is tasked with evaluating the applicability of existing authentication and authorization protocols.

- The ACE working group may investigate other security protocols later, with a particular focus on adapting whatever solution is chosen to HTTP and TLS.

- 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. Aligned with the initial focus, access to resources at a resource server by a client device occurs using CoAP and is protected by DTLS.

DICE

- In constrained environments secured by DTLS, CoAP can be used to control resources on a device.

- 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

- The leveraging the Internet Protocol suite for smart objects involves a collection of protocols and options that must work in coordination with lower and upper layers.
- Therefore, profile definitions, certifications, and promotion by alliances can help implementers develop solutions that guarantee interoperability and/or interchangeability of devices.
- Let us see some of the main industry organizations working on profile definitions and certifications for IoT constrained nodes and networks.

Internet Protocol for Smart Object(IPSO) Alliance

- Established in 2008, the Internet Protocol for Smart Objects (IPSO) Alliance has had its objective evolve over years.

- The IPSO Alliance does not define technologies, as that is the role of the IETF and other standard organizations, but it documents the use of IP-based technologies for various IoT use cases and participates in educating the industry

Wi-SUN Alliance

- The Wi-SUN Alliance is an example of efforts from the industry to define a communication profile that applies to specific physical and data link layer protocols.
- 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

Thread

- A group of companies involved with smart object solutions for consumers created the Thread Group.
- This group has defined an IPv6-based wireless profile that provides the best way to connect more than 250 devices into a low-power, wireless mesh network.
- The wireless technology used by Thread is IEEE 802.15.4, which is different from Wi-SUN's IEEE 802.15.4g.

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.