



Internet of Things

Lecture 4: Network Layer Protocols (IP as the IoT Network Layer)

Mehdi Rasti
Amirkabir University of Technology

Spring 2020



Lecture 4- Contents

- The Business Case for IP
- The Need for Optimization
- Optimizing IP for IoT

Mostly adopted from Chapters 4, 5, and 6 of **IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Thing**, Cisco press, 2017

The Business Case for IP

- Previous lecture focused on connectivity at Layer 1 (PHY) and Layer 2 (MAC).
- In this chapter, we move up the protocol stack and focus on network layer connectivity, which is commonly referred to as Layer 3.

The Business Case for IP

- The key advantages of the IP suite for the Internet of Things:
 - Open and standards-based
 - Versatile
 - Ubiquitous
 - Scalable
 - Manageable and highly secure
 - Stable and resilient
 - Consumers' market adoption
 - The innovation factor

* [IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Thing](#), Cisco press, 2017

Adoption or Adaptation of the Internet Protocol

- Typically, one of two models, adaptation or adoption, for IP in IoT 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.

Adoption or Adaptation of the Internet Protocol

- IP adaptation versus adoption model still requires investigation for particular last-mile technologies used by IoT

Adoption or Adaptation of the Internet Protocol

- You should consider the following factors when trying 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.
 - IoT devices, may only infrequently need to report a few bytes of data to an application.
 - For these cases, it is not necessarily worth implementing a full IP stack.
 - **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
 - It needs to decide whether the IP adoption model is necessary and, if it is, how it can be optimized.

Adoption or Adaptation of the Internet Protocol

- **Data flow model:**

- One benefit of the IP adoption model is that any node can easily exchange data with any other node in a network
- 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 Need for Optimization of IP for IoT

A network diagram in the top right corner showing a complex web of interconnected nodes. Some nodes are represented by solid circles, while others are concentric circles. They are connected by thin lines, some solid and some dashed, representing network links.

- Constrained Nodes
- Constrained Networks

The Need for Optimization of IP for IoT

- **Constrained Nodes**

- 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 of IP for IoT

- **Constrained Networks**

- Constrained networks are limited by
 - low-power,
 - low-bandwidth links (wireless and wired).
- 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.
- In contrast, highly stable and fast links are available for typical IP networks

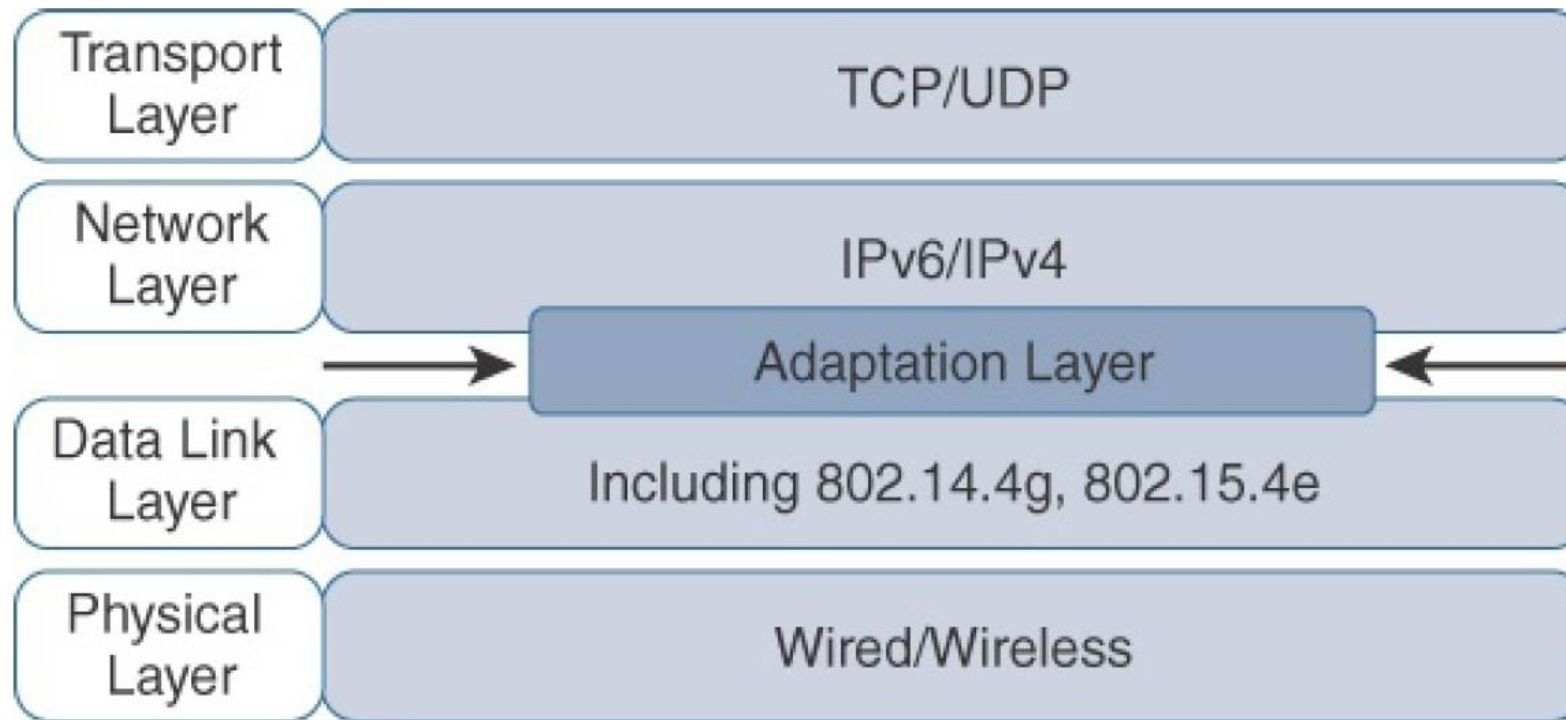
The Need for Optimization of IP for IoT

- **Constrained Networks**

- Packet delivery rate (PDR) oscillate between low and high percentages
- Large bursts of unpredictable errors and even loss of connectivity at times may occur
- Control plane traffic must also be kept at a minimum; otherwise, it consumes the bandwidth that is needed by the data traffic.
- Finally, you have to consider the power consumption in battery-powered nodes.
 - Any failure or verbose control plane protocol may reduce the lifetime of the batteries.

Optimizing IP for IoT

- constrained nodes and constrained networks mandate optimization at various layers and on multiple protocols of the IP architecture.

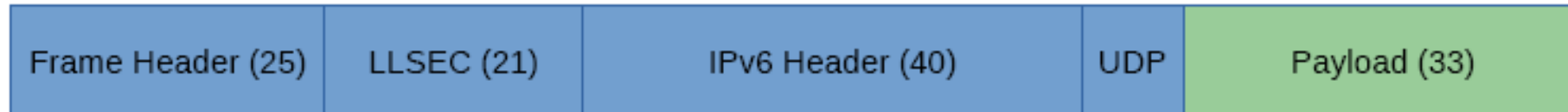


Optimizing IP for IoT

- 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*.
- An adaptation layer designed for IoT may include some optimizations to deal with constrained nodes and networks

The Header Size Problem

- Worst-case scenario calculations
 - Maximum frame size in IEEE 802.15.4: 127 byte
 - Reduced by the max. frame header (25 byte): 102 byte
 - Reduced by highest link-layer security (21 byte): 81 byte
 - Reduced by standard IPv6 header (40 byte): 41 byte
 - Reduced by standard UDP header (8 byte): 33 byte
 - This leaves only 33 byte for actual payload
 - The rest of the space is used by headers



Optimizing IP for IoT

- 6LoWPAN as IP Adaptation Layer for IoT

IP Protocol Stack

HTTP		RTP	
TCP	UDP	ICMP	
IP			
Ethernet MAC			
Ethernet PHY			

Application

Transport

Network

Data Link


Physical

IoT Protocol Stack with
6LoWPAN Adaptation Layer

Application Protocols	
UDP	ICMP
IPv6	
LoWPAN	
IEEE 802.15.4 MAC	
IEEE 802.15.4 PHY	

Movement Towards IP



- A IoT protocols are moving towards IP
 - TCP/IP is not one size fits all
 - Adaptations needed for MTU size
 - Reduce of header overhead
 - UDP instead of TCP to avoid latencies
- 

Optimizing IP for IoT

- The IPv6 over Low-Power Wireless Personal Area Networks (6LowPAN) working group focused on enabling IPv6 over IEEE 802.15.4 networks.
- The group started its work in 2005 and concluded in 2014 after working through the following goals:
 - Defining a fragmentation and reassembly layer to allow adaptation of IPv6 to IEEE 802.15.4 links
 - Introduce an IPv6 header compression mechanism to avoid excessive fragmentation and reassembly, since the IPv6 header alone is 40 bytes long, without optional headers.
 - Examining mesh routing protocol suitability to 802.15.4 networks, especially in light of the packet size constraints.

Optimizing IP for IoT

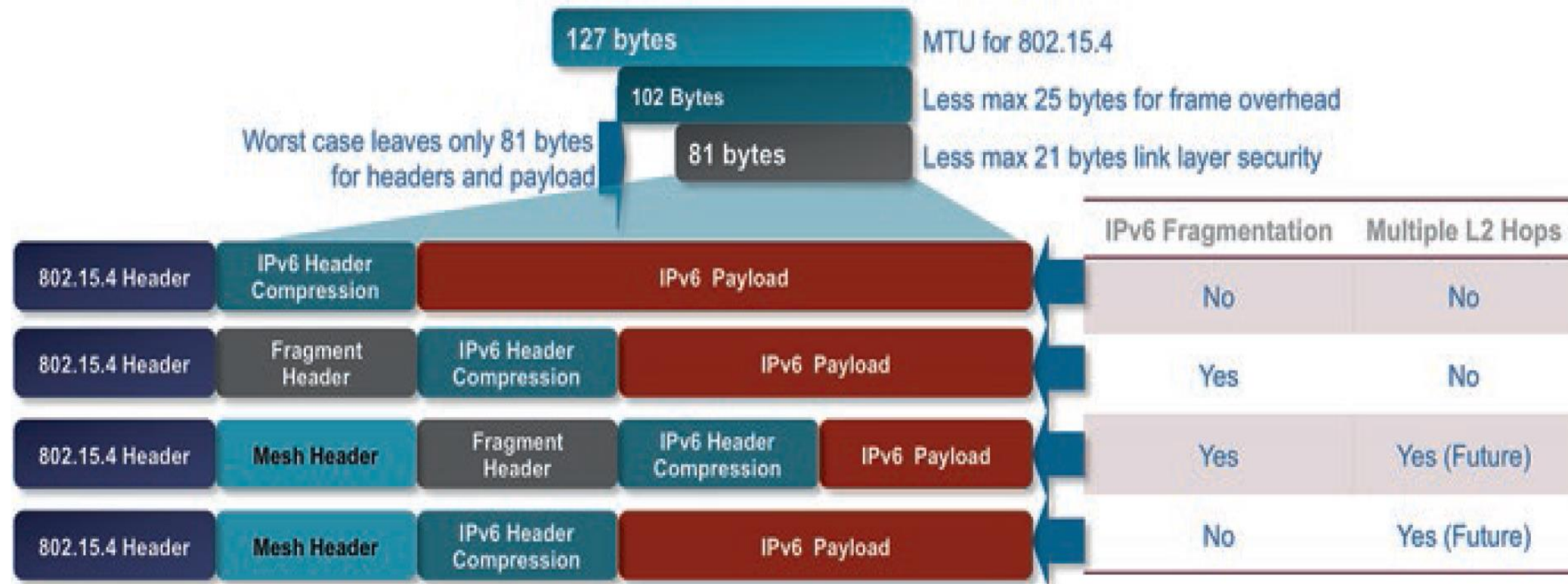


- 6LoWPAN as IP Adaptation Layer for IoT
 - Header compression
 - Fragmentation
 - Mesh addressing

Optimizing IP for IoT



Optimizing IP for IoT



Optimizing IP for IoT-6LoWPAN Header compression

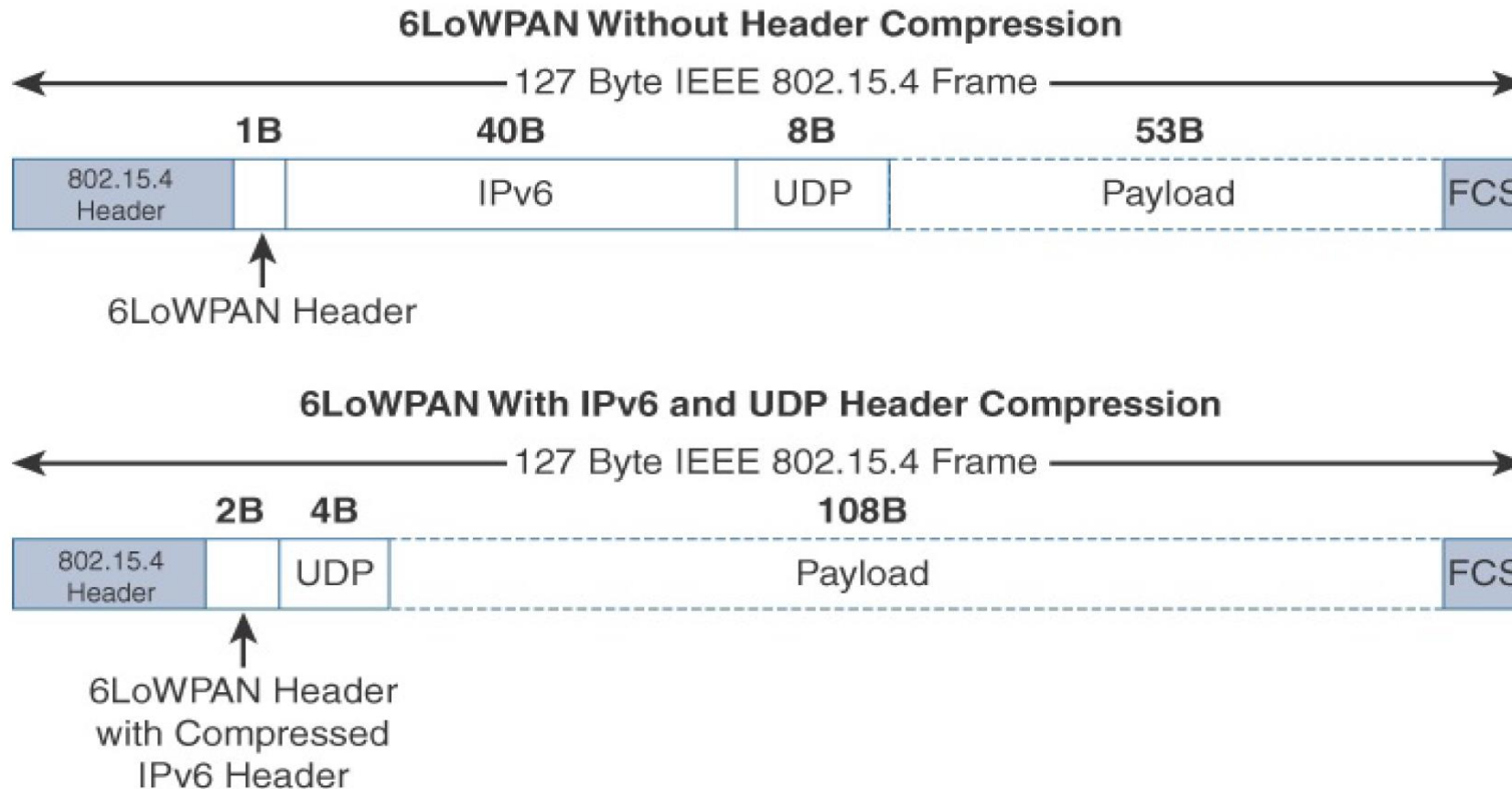
- At a high level, 6LoWPAN works by taking advantage of shared information known by all nodes from their participation in the local network.
- In addition, it omits some standard header fields by assuming commonly used values.

Optimizing IP for IoT-6LoWPAN Header compression

- A 6LoWPAN frame without any header compression enabled:
 - The full 40-byte IPv6 header and 8-byte UDP header
 - The 6LoWPAN header is only a single byte
 - 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.
- A 6LoWPAN frame with a 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- 6LoWPAN Header compression

- Header compression



Optimizing IP for IoT- 6LoWPAN Header compression

- Header compression
 - Note that the 2-byte header compression applies to intra-cell communications, while communications external to the cell may require some field of the header to not be compressed.

Optimizing IP for IoT-6LoWPAN Fragmentation

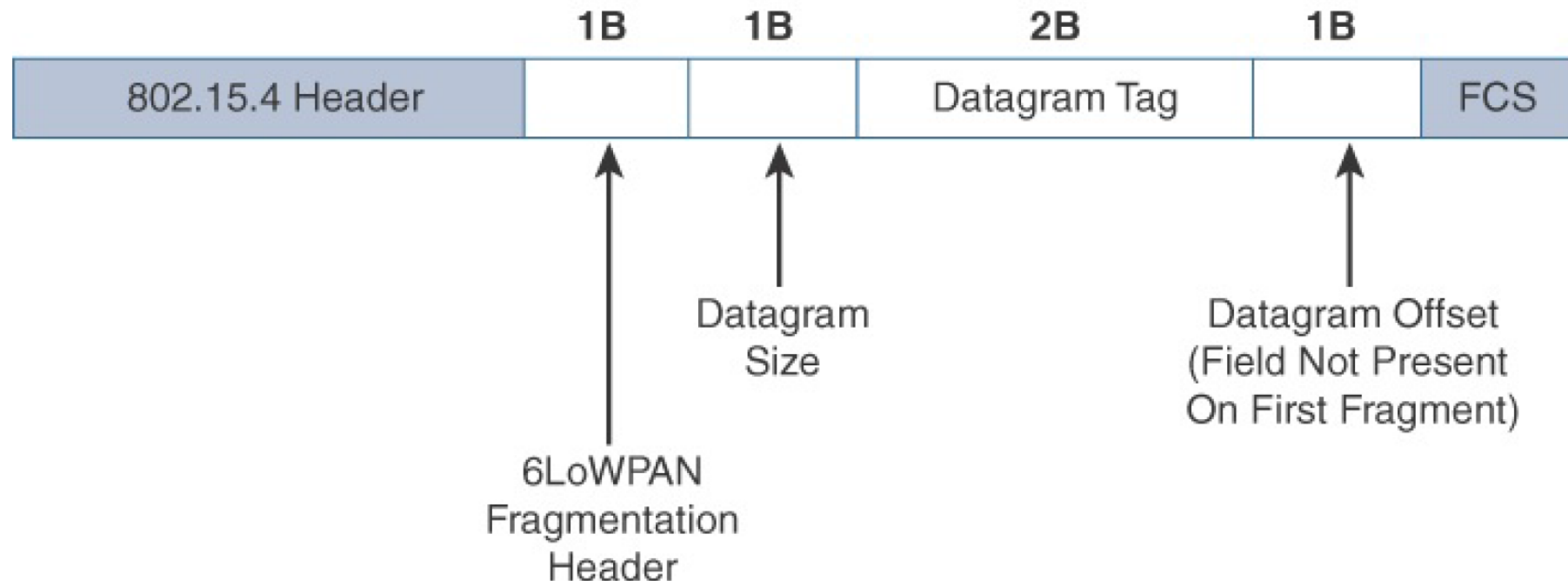
- The maximum transmission unit (MTU) for an IPv6 network is 1280 bytes.
 - The term *MTU* defines the size of the largest protocol data unit that can be passed.
- For IEEE 802.15.4, 127 bytes is the MTU. IPv6, with a much larger MTU, is carried inside the 802.15.4 frame with a much smaller one.
- To remedy this situation, large IPv6 packets must be fragmented across multiple 802.15.4 frames at Layer 2.

Optimizing IP for IoT-6LoWPAN Fragmentation

- The fragment header utilized by 6LoWPAN is composed of three primary fields:
 - Datagram Size: specifies the total size of the unfragmented payload.
 - Datagram Tag: identifies the set of fragments for a payload
 - Datagram Offset: delineates how far into a payload a particular fragment occur

Optimizing IP for IoT-6LoWPAN Fragmentation

- **Fragmentation:** *6LoWPAN Fragmentation Header*



Optimizing IP for IoT-6LoWPAN Fragmentation

- **Fragmentation**

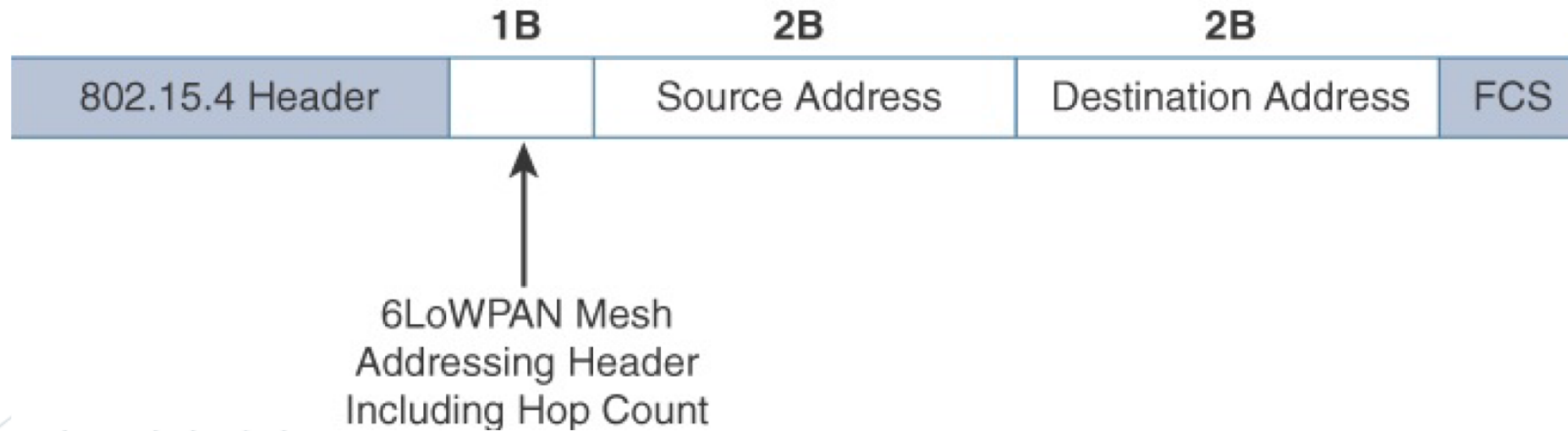
- 6LoWPAN fragmentation header field itself uses a unique bit value to identify that the subsequent fields behind it are fragment fields as opposed to another capability, such as header compression.
- Also, in the first fragment, the Datagram Offset field is not present because it would simply be set 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.

* *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Thing*, Cisco press, 2017

6LoWPAN: 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,
 - Destination Address
- Analogous to the IPv6 hop limit field, the hop limit for mesh addressing also provides an upper limit on how many times the frame can be forwarded.
 - Each hop decrements this value by 1 as it is forwarded.
 - Once the value hits 0, it is dropped and no longer forwarded.

6LoWPAN: Mesh Addressing



Mesh-Under Versus Mesh-Over Routing

- Two main options exist for establishing reachability and forwarding packets.
 - Mesh-under:
 - Routing of packets is handled at the 6LoWPAN adaptation layer.
 - Mesh-over” or “route-over,”
 - Utilizes IP routing for getting packets to their destination

Mesh-Under Routing

- With mesh-under routing, the routing of IP packets leverages the 6LoWPAN mesh addressing header discussed in the previous section to route and forward packets at the link layer.
- 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.
 - 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.

Mesh-Over Routing

- In 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 IPv6 Routing Protocol for Low Power and Lossy Networks (RPL), is recommended.
 - RPL is discussed in more detail later in this chapter.

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**

6Lo Working Group

- 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 **using 6LoWPAN technologies (RFC4944, RFC6282, RFC6775) for link layer technologies:**
 - IPv6 over Bluetooth Low Energy
 - Transmission of IPv6 packets over near-field communication
 - IPv6 over 802.11ah
 - Transmission of IPv6 packets on WIA-PA (Wireless Networks for Industrial Automation–Process Automation)
 - Transmission of IPv6 packets over DECT Ultra Low Energy

Optimizing IP for IoT

- In fact, based on the work of the 6LoWPAN working group and now the 6Lo working group:
 - the 6LoWPAN adaptation layer is becoming the de factor standard for connecting constrained nodes in IoT networks.

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
- A new routing protocol should be developed for use by IP smart objects, given the characteristics and requirements of constrained networks.
- This new distance-vector routing protocol was named the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL).

RPL

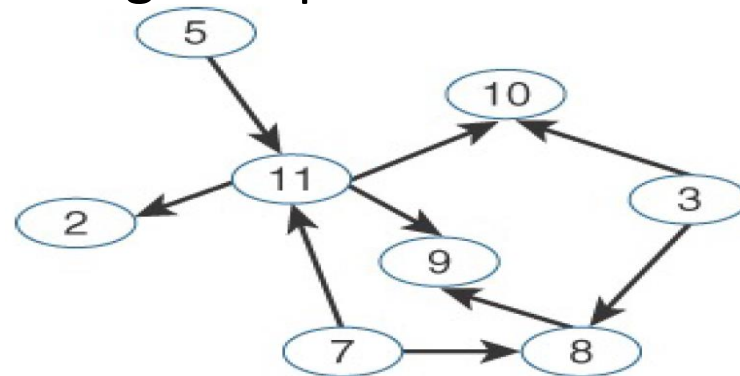
- In an RPL network, each node acts as a router and becomes part of a mesh network.
- Routing is performed at the IP layer.
- Each node examines every received IPv6 packet and determines the next-hop destination based on the information contained in the IPv6 header.
- No information from the MAC layer header is needed to perform next-hop determination.
 - this is referred to as mesh-over routing

RPL

- 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.
 - 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

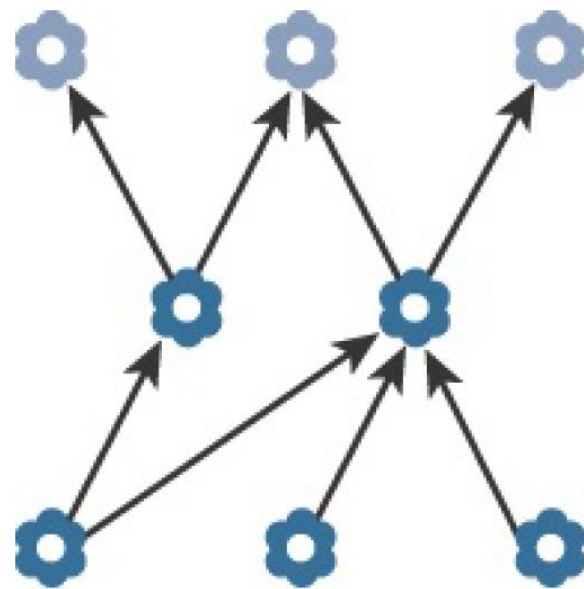
- RPL is based on the concept of a directed acyclic graph (DAG).
- A DAG is a directed graph where no cycles exist.
 - This means that from any vertex or point in the graph, you cannot follow an edge or a line back to this same point.
- All of the edges are arranged in paths oriented toward and terminating at one or more root nodes



RPL

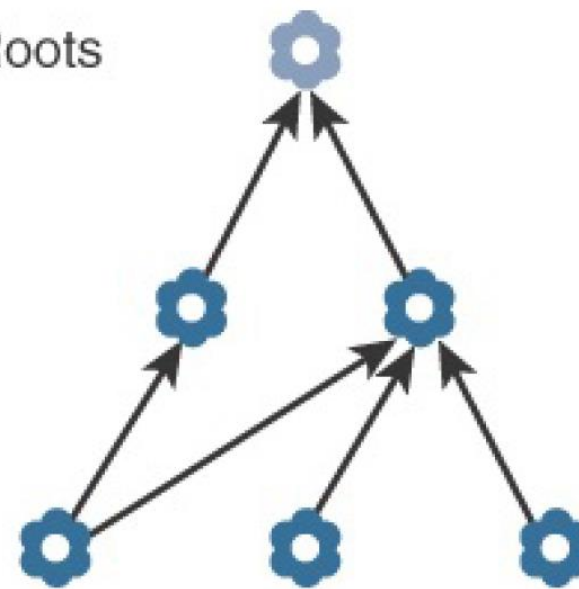
- A basic RPL process involves building a destination-oriented directed acyclic graph (DODAG).
- A DODAG is a DAG rooted to one destination.
- In RPL, this destination occurs at a border router known as the DODAG root.
- A DAG has multiple roots (a set of DODAG) , whereas the DODAG has just one.

RPL



DAG

DAG Roots



DODAG

RPL



- 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

- DAG Information Object (DIO) messages:
 - Upward routes in RPL are discovered and configured using DIO.
 - 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.
- Destination Advertisement Object (DAO) message
 - Nodes establish downward routes by advertising their parent set toward the DODAG root using a DAO message.
 - DAO messages allow nodes to inform their parents of their presence and reachability to descendants.

RPL

- In the case of the non-storing mode of RPL, nodes sending DAO message 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.
 - The nodes can make their own routing decisions; in non-storing mode, on the other hand, all packets must go up to the root to get a route for moving downstream.

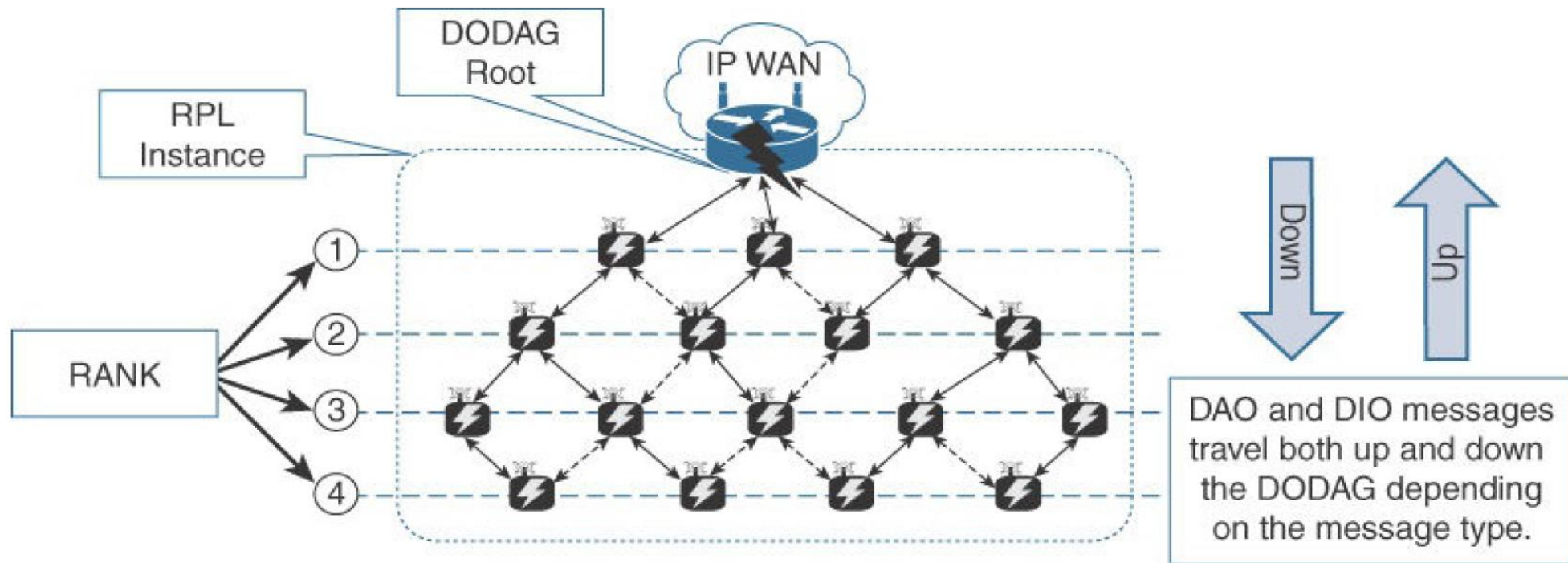
RPL



- 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.

RPL

- As illustrated in Figure 5-10, DAO and DIO messages move both up and down the DODAG, depending on the exact message type.



- **Objective Function (OF)**

- An objective function (OF) defines how metrics are used to select routes and establish a node's rank.
- Standards such as RFC 6552 and 6719 have been published to document OFs specific to certain use cases and node types.
- For example, nodes implementing an OF based on RFC 6719's Minimum Expected Number of Transmissions (METX) advertise the METX among their parents in DIO messages.
- Whenever a node establishes its rank, it simply sets the rank to the current minimum METX among its parents.

RPL

- **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.
- While the rank and routing metrics are closely related, the rank differs from routing metrics in that it is used as a constraint to prevent routing loops.

RPL

- **Metrics**

- Developed to support powered and battery-powered nodes, RPL offers a far more complete set than any other routing protocol, include the following:

- Expected Transmission Count (ETX):
 - Hop Count
 - Latency
 - Link Quality Level
 - Link Color
 - Node State and Attribute
 - Node Energy
 - Throughput

RPL

