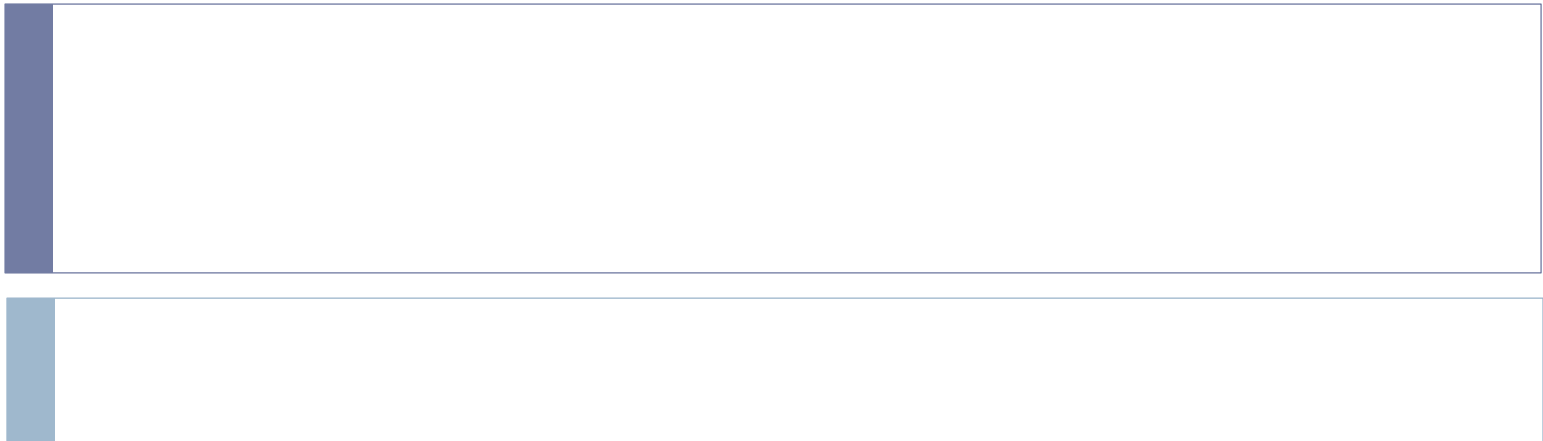


MODULE-3

IP as the IoT Network Layer



Outline

3.1 The Business Case for IP

3.1.1 The Key Advantages of Internet Protocol

3.1.2 Adoption or Adaptation of the Internet Protocol

3.2 The Need for Optimization

3.3 Profiles and Compliances

3.4 Application Protocols for IoT

3.5 The Transport Layer

3.6 IoT Application Transport Methods



3.1 The business case for IP

- ▶ Data flowing from or to “things” is consumed, controlled, or monitored by data center servers either in the cloud or in locations that may be distributed or centralized.
- ▶ Dedicated applications are then run over virtualized or traditional operating systems or on network edge platforms (for example, fog computing).
- ▶ Lightweight applications communicate with the data center servers. This is how and why the Internet Protocol (IP) suite started playing a key architectural role in the early 1990s.
- ▶ IP was not only preferred in the IT markets but also for the OT environment.



3.1.1 Key advantages of Internet Protocol

1. Open and standard based
2. Versatile
3. Ubiquitous
4. Scalability
5. Manageable and highly secure
6. Stable and resilient
7. Customer's market adoption
8. Innovative factors



I. Open and standards-based:

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



2. Versatile:


- A large spectrum of access technologies is available to offer connectivity of “things” in the last mile.
 - Additional protocols and technologies are also used to transport IoT data through backhaul links and in the data center.
 - Even if physical and data link layers such as Ethernet, Wi-Fi, and cellular are widely adopted, the history of data communications demonstrates that no given wired or wireless technology fits all deployment criteria.
 - Furthermore, communication technologies evolve at a pace faster than the expected 10- to 20-year lifetime of OT solutions.
-



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.
- ▶ In addition, IoT application protocols in many industrial OT solutions have been updated in recent years to run over IP.

4. Scalable:

- As the common protocol of the Internet, IP has been massively deployed and tested for robust scalability.
 - Millions of private and public IP infrastructure nodes have been operational for years, offering strong foundations for those not familiar with IP network management.
 - IP has proven before that scalability is one of its strengths.
- 

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

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..

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

8. The innovation factor: 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.



3.1.2 Adaptations of the Internet Protocol

- ▶ **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.



3.1.3 Factors that should consider when trying to determine which model is best suited for last-mile connectivity :

1. Bidirectional versus unidirectional data flow
2. Overhead for last-mile communications paths
3. Data flow model
4. Network diversity



i. Bidirectional versus unidirectional data flow

- ▶ While bidirectional communications are generally expected, some last-mile technologies offer optimization for unidirectional communication.



ii. Over head for last mile communications path

- ▶ With a per-packet overhead that varies depending on the IP version. 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. Consequently, you need to decide
- ▶ whether the IP adoption model is necessary and, if it is, how it can be optimized. This same consideration applies to control plane traffic that is run over IP for lowbandwidth, last-mile links. Routing protocol and other verbose network services may either not be required or call for optimization.



iii. 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.
 - Depending on the network topology and the data flow needed, both IP adaptation and adoption models have roles to play in last-mile connectivity.
-



iv. 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.
- ▶ Integration and coexistence of new physical and MAC layers or new applications impact how deployment and operations have to be planned.



3.2 The Need for Optimization

▶ 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.



3.2.1 Constrained Networks

- In the early years of the Internet, network bandwidth capacity was restrained due to technical limitations. Connections often depended on low-speed modems for transferring data.
 - However, these low-speed connections demonstrated that IP could run over lowbandwidth networks.
 - Fast forward to today, and the evolution of networking has seen the emergence of highspeed infrastructures.
 - However, high-speed connections are not usable by some IoT devices in the last mile.
 - A constrained network can have high latency and a high potential for packet loss.
-



3.2.2 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
- ▶ **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
- ▶ **Serial Communications** : Many legacy devices in certain industries, such as manufacturing and utilities, communicate through serial lines. Data is transferred using either proprietary or standards-based protocols, such as DNP3, Modbus, or IEC 60870-5-101.
- ▶ **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 ITU G.9903 (Narrowband Power Line Communications) only have an IPv6 adaptation layer specified



3.2.3 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.

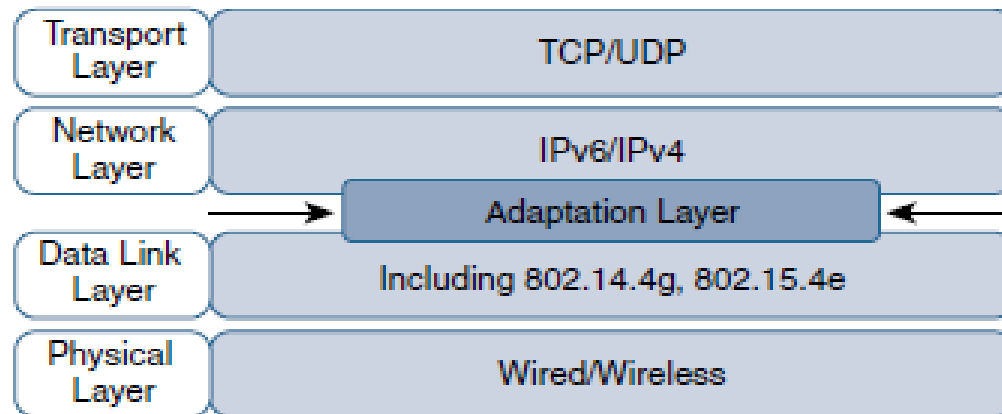
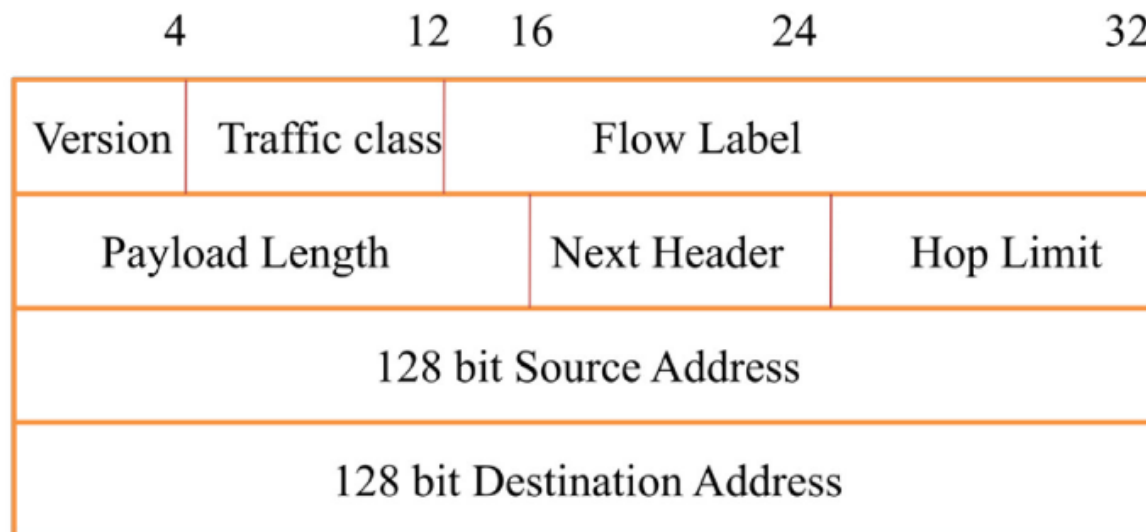


Figure 5-1 *Optimizing IP for IoT Using an Adaptation Layer*

0	3	4	7	8	15	16	31
Version	Length	DSCP			Total Length		
Identifier					Flags	Fragmented Offset	
TTL		Protocol			Header Checksum		
Source IP Address							
Destination IP Address							
Options and Padding							



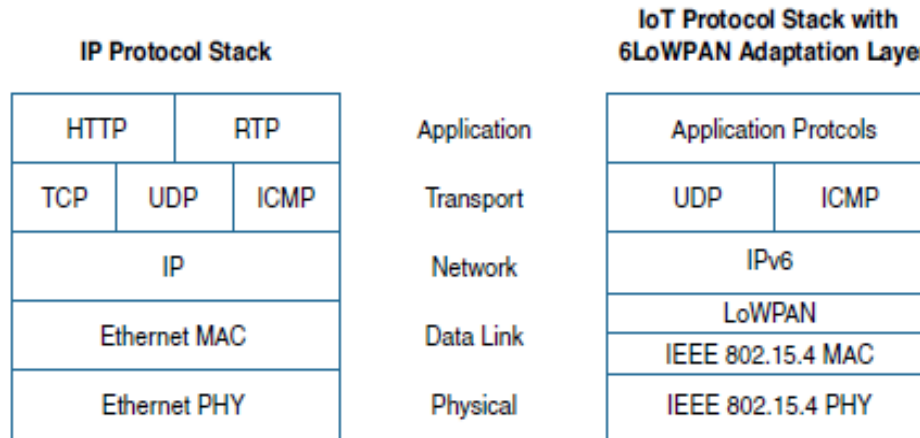


3.2.4 6LoWPAN to 6Lo

- ▶ **6LoWPAN** is an acronym of *IPv6 over Low -Power Wireless Personal Area Networks*. 6LoWPAN is the name of a concluded working group in the Internet area of the IETF.
 - ▶ The 6LoWPAN concept originated from the idea that "the Internet Protocol could and should be applied even to the smallest devices," and that low-power devices with limited processing capabilities should be able to participate in the Internet of Things.
 - ▶ The 6LoWPAN group has defined encapsulation and header compression mechanisms that allow IPv6 packets to be sent and received over IEEE 802.15.4 based networks.
 - ▶ IPv4 and IPv6 are the work horses for data delivery for local-area networks, metropolitan area networks, and wide-area networks such as the Internet. Likewise, IEEE 802.15.4 devices provide sensing communication-ability in the wireless domain. The inherent natures of the two networks though, are different.
-



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



6LoWAN Header Stacks :

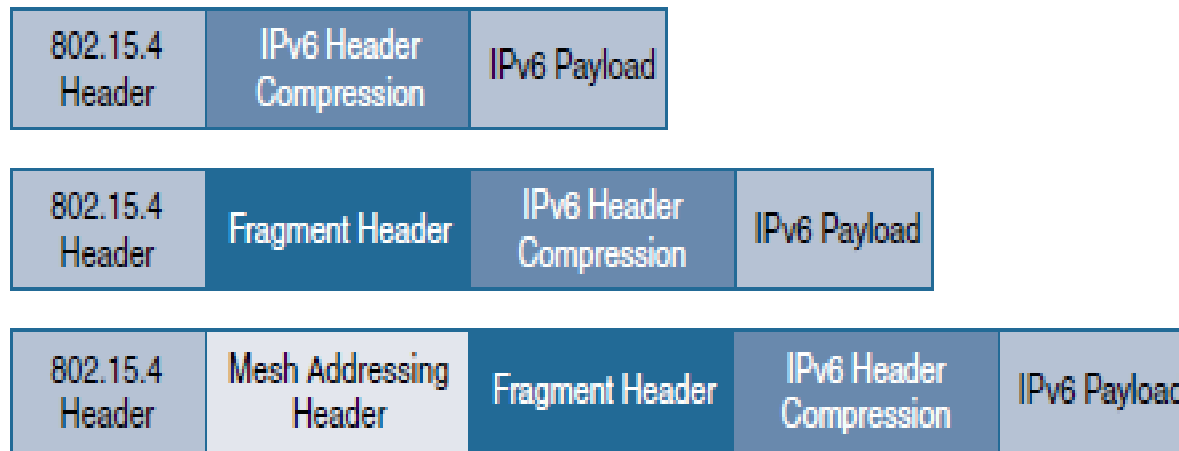
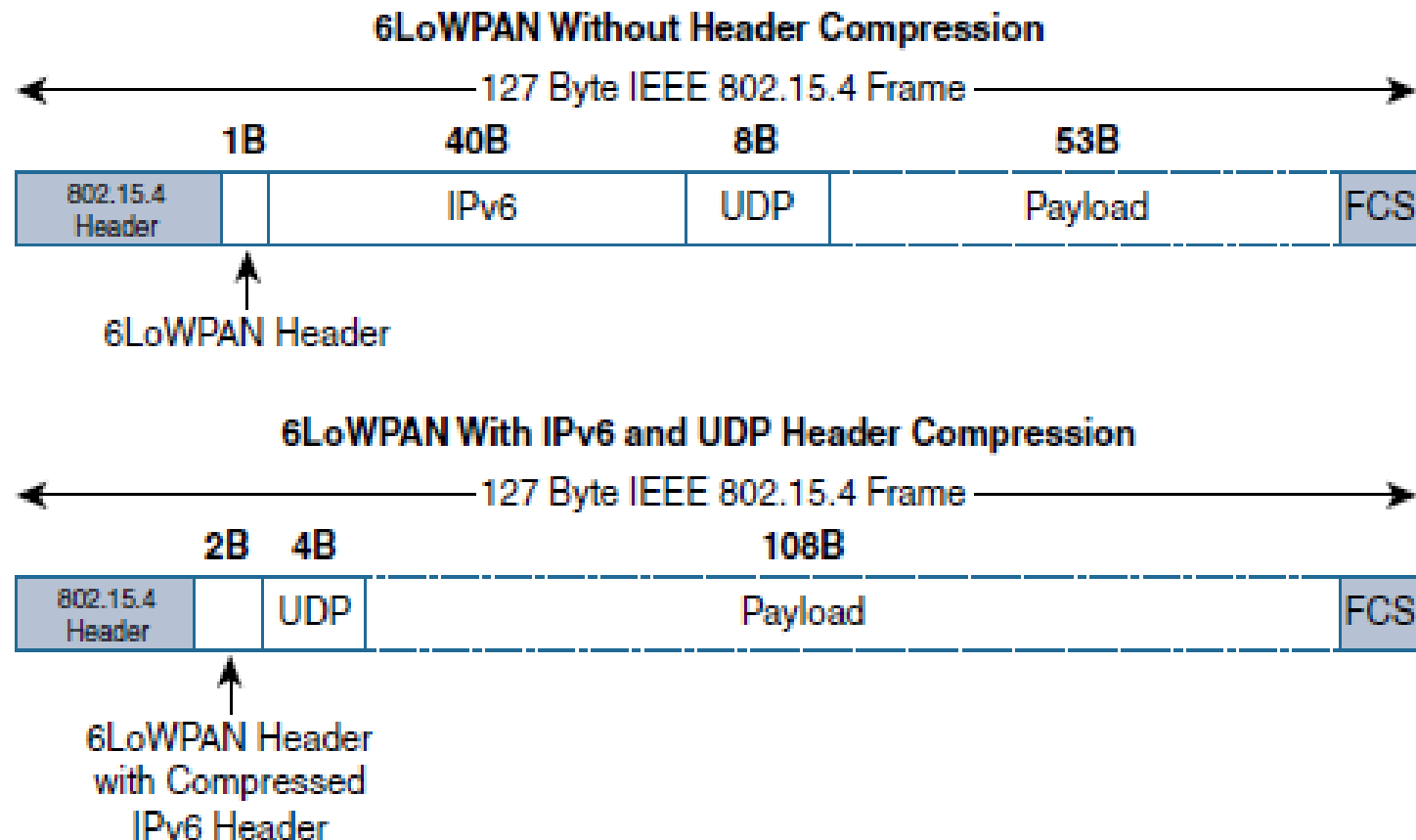


Figure 5-3 *6LoWPAN Header Stacks*

6LoPAN Header Compression



- IPv6 header compression for 6LoWPAN was defined initially in RFC 4944 and subsequently updated by RFC 6282. This capability shrinks the size of IPv6's 40-byte headers and User Datagram Protocol's (UDP's) 8-byte headers down as low as 6 bytes combined in some cases.
- The 6LoWPAN protocol does not support IPv4, and, in fact, there is no standardized
- IPv4 adaptation layer for IEEE 802.15.4.
- 6LoWPAN header compression is stateless, and conceptually it is not too complicated.
- At a high level, 6LoWPAN works by taking advantage of shared information known by all nodes from their participation in the local network.
- The full 40-byte IPv6 header and 8-byte UDP header are visible. The 6LoWPAN header is only a single byte in this case.



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.
- The fragment header utilized by 6LoWPAN is composed of three primary fields: **Datagram Size, Datagram Tag, and Datagram Offset.**
- The 1-byte Datagram Size field specifies the total size of the unfragmented payload.
- Datagram Tag identifies the set of fragments for a payload.

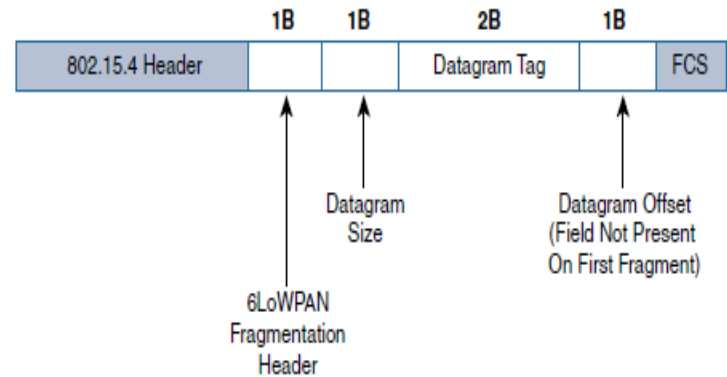


Figure 5-5 6LoWPAN Fragmentation Header

3.3 Profiles and Compliances

- ▶ 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.



3.3.1 Internet Protocol for Smart Objects (IPSO) Alliance

- Established in 2008, the Internet Protocol for Smart Objects (IPSO) Alliance has had its objective evolve over years.
- 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.
- 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.
- As the IPSO Alliance declares in its value and mission statement, it wants to ensure that “engineers and product builders will have access to the necessary tools for ‘how to build the IoT RIGHT.’”
- For more information on the IPSO Alliance, visit www.ipso-alliance.org.



3.3.2 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.
 - ▶ Currently, 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.
-



3.3.3 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.



3.3.4 IPv6 Ready Logo

- ▶ Initially, the IPv6 Forum ensured the promotion of IPv6 around the world.
 - ▶ Once IPv6 implementations became widely available, the need for interoperability and certification led to the creation of the IPv6 Ready Logo program.
 - ▶ The IPv6 Ready Logo program has established conformance and interoperability testing programs with the intent of increasing user confidence when implementing IPv6.
 - ▶ The IPv6 Core and specific IPv6 components, such as DHCP, IPsec, and customer edge router certifications, are in place.
 - ▶ These certifications have industry-wide recognition, and many products are already certified. An IPv6 certification effort specific to IoT is currently under definition for the program.
-



3.4 The Transport Layer

- ▶ 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.
- You can view it as an equivalent to a traditional telephone conversation, in which two phones must be connected and the communication link established before the parties can talk.
- ▶ TCP is the main protocol used at the transport layer. This is largely due to its inherent characteristics, such as its ability to transport large volumes of data into smaller sets of packets.
- ▶ It ensures reassembly in a correct sequence, flow control and window adjustment, and retransmission of lost packets.

User Datagram Protocol (UDP):

- With this connectionless protocol, data can be quickly sent between source and destination—but with no guarantee of delivery.
- ▶ This is analogous to the traditional mail delivery system, in which a letter is mailed to a destination. Confirmation of the reception of this letter does not happen until another letter is sent in response.
- ▶ UDP is most often used in the context of network services, such as Domain Name System (DNS), Network Time Protocol (NTP), Simple Network Management Protocol (SNMP), and Dynamic Host Control Protocol (DHCP), or for real-time data traffic, including voice and video over IP

3.5 IoT Application Transport Methods

The following categories of IoT application protocols and their transport methods are explored in the following sections:

- **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.
- **Supervisory control and data acquisition (SCADA):** SCADA is one of the most common industrial protocols in the world, but it was developed long before the days of IP, and it has been adapted for IP networks.
- **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), covered later in this chapter, are two well known examples of IoT application layer protocols.



3.5.1 SCADA

- ▶ In the world of networking technologies and protocols, IoT is relatively new.
- ▶ Combined with the fact that IP is the de facto standard for computer networking in general, older protocols that connected sensors and actuators have evolved and adapted themselves to utilize IP.
- ▶ A prime example of this evolution is supervisory control and data acquisition (SCADA).
- ▶ Designed decades ago, SCADA is an automation control system that was initially implemented without IP over serial links, before being adapted to Ethernet and IPv4.



3.5.2 A Little Background on SCADA

- Vertical industries have developed communication protocols that fit their specific requirements.
- Many of them were defined and implemented when the most common networking technologies were serial link-based, such as RS-232 and RS-485.
- This led to SCADA networking protocols, which were well structured compared to the protocols described in the previous section, running directly over serial physical and data link layers.
- At a high level, SCADA systems collect sensor data and telemetry from remote devices, while also providing the ability to control them.
- Used in today's networks, SCADA systems allow global, real-time, data-driven decisions to be made about how to improve business processes.
- SCADA networks can be found across various industries, but you find SCADA mainly concentrated in the utilities and manufacturing/industrial verticals. Within these specific industries, SCADA commonly uses certain protocols for communications between devices and applications.



3.5.3 Adapting SCADA for IP

- ▶ In the 1990s, the rapid adoption of Ethernet networks in the industrial world drove the evolution of SCADA application layer protocols. For example, the IEC adopted the Open System Interconnection (OSI) layer model to define its protocol framework. Other protocol user groups also slightly modified their protocols to run over an IP infrastructure.

Specifications :

- DNP3 (adopted by IEEE 1815-2012) specifies the use of TCP or UDP on port 20000 for transporting DNP3 messages over IP.
- The Modbus messaging service utilizes TCP port 502. IEC 60870-5-104 is the evolution of IEC 60870-5-101 serial for running over
- Ethernet and IPv4 using port 2404.
- DLMS User Association specified a communication profile based on TCP/IP in the DLMS/COSEM Green Book (Edition 5 or higher), or in the IEC 62056-53 and IEC 62056-47 standards, allowing data exchange via IP and port 4059.



3.5.4 Protocol Stack for Transporting Serial DNP3 SCADA over IP

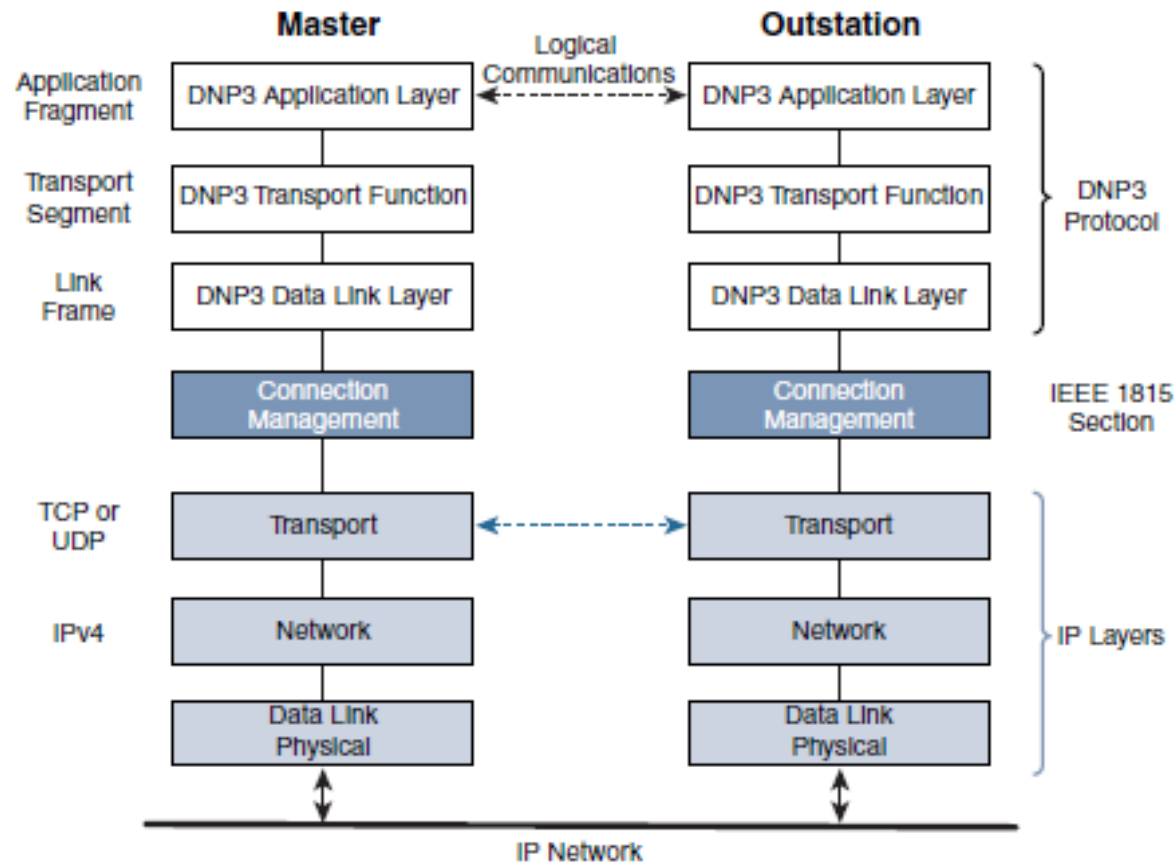


Figure 6-2 Protocol Stack for Transporting Serial DNP3 SCADA over IP

- ▶ 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*.
- ▶ Outstations monitor and collect data from devices that indicate their state, such as whether a circuit breaker is on or off, and take measurements, including voltage, current, temperature, and so on.



- ▶ The IEEE 1815-2012 specification describes how the DNP3 protocol implementation must be adapted to run either over TCP (recommended) or UDP.
- ▶ This specification defines connection management between the DNP3 protocol and the IP layers, as shown in Figure.
- ▶ Connection management links the DNP3 layers with the IP layers in addition to the configuration parameters and methods necessary for implementing the network connection.
- ▶ The IP layers appear transparent to the DNP3 layers as each piece of the protocol stack in one station logically communicates with the respective part in the other.
- ▶ This means that the DNP3 endpoints or devices are not aware of the underlying IP transport that is occurring.



3.6 IoT Application Layer Protocols

■ When considering constrained networks and/or a large-scale deployment of constrained nodes, verbose web-based and data model protocols, as discussed in the previous section, may be too heavy for IoT applications.

■ To address this problem, the IoT industry is working on new lightweight protocols that are better suited to large numbers of constrained nodes and networks.

■ Two of the most popular protocols are CoAP and MQTT.

CoAP	MQTT
UDP	TCP
IPv6	
6LoWPAN	
802.15.4 MAC	
802.15.4 PHY	

Figure 6-6 Example of a High-Level IoT Protocol Stack for CoAP and MQTT



3.6.1 CoAP

- ❑ Constrained Application Protocol (CoAP) resulted from the IETF Constrained RESTful Environments (CoRE) working group's efforts to develop a generic framework for resource-oriented applications targeting constrained nodes and networks.
 - ❑ The CoAP framework defines simple and flexible ways to manipulate sensors and actuators for data or device management. The IETF CoRE working group has published multiple standards-track specifications for CoAP, including the following:
 - ▶ RFC 6690: Constrained RESTful Environments (CoRE) Link Format
 - ▶ RFC 7252: The Constrained Application Protocol (CoAP)
 - ▶ RFC 7641: Observing Resources in the Constrained Application Protocol (CoAP)
 - ▶ RFC 7959: Block-Wise Transfers in the Constrained Application Protocol (CoAP)
 - ▶ RFC 8075: Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)
-



CoAP Message Format

Table 6-1 *CoAP Message Fields*

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 Communications in IoT Infrastructures

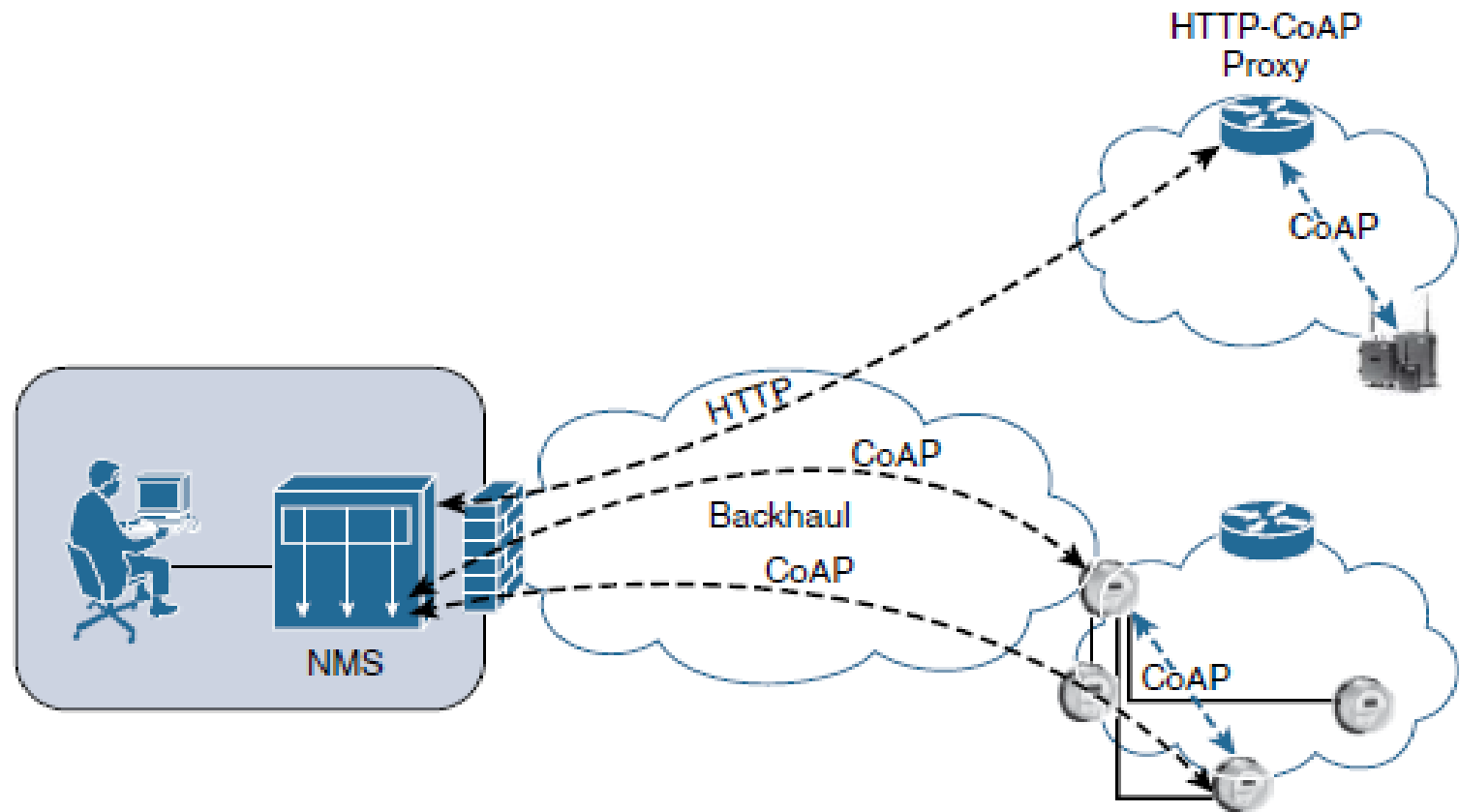


Figure 6-8 *CoAP Communications in IoT Infrastructures*

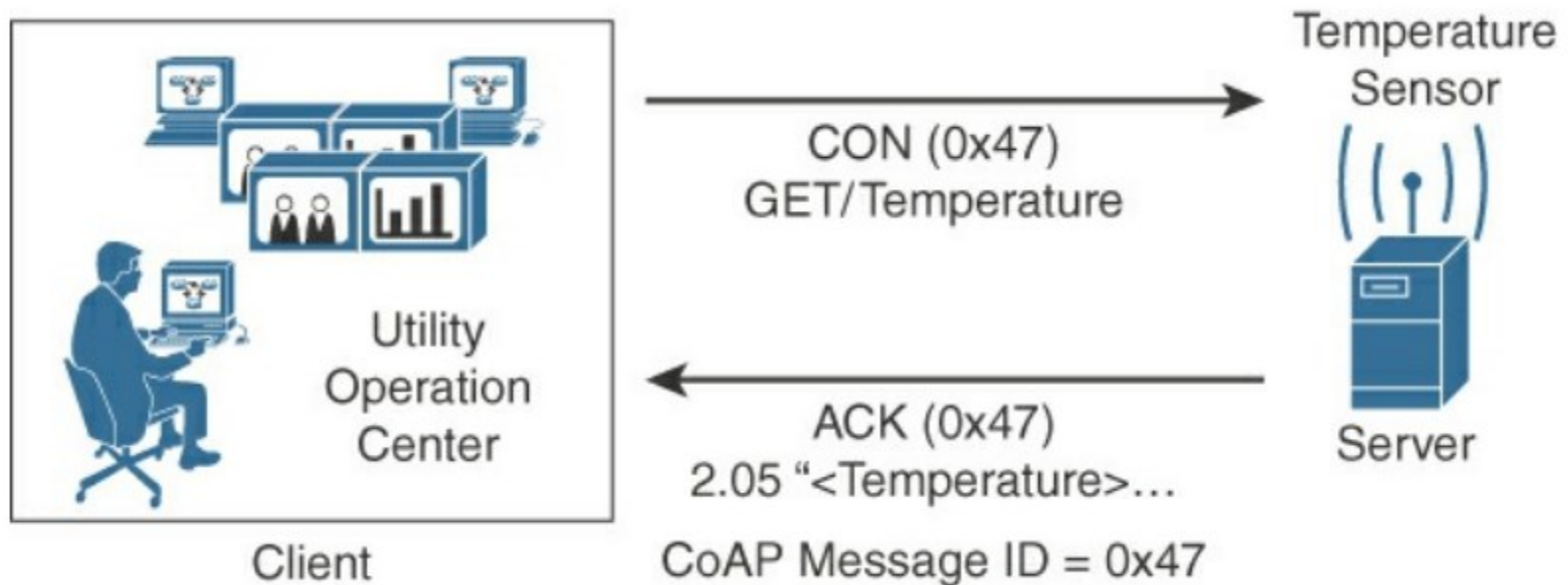


Figure 6-9 *CoAP Reliable Transmission Example*

Benefits of CoAP

- ▶ Web Protocol Used in M2M With Constrained Requirements
- ▶ Asynchronous Message Exchange
- ▶ Low Overhead
- ▶ Very Simple To Perform Syntactic Analysis
- ▶ (URI) Uniform Resource Identifier
- ▶ Proxy and Caching Capabilities
- ▶



-
- 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.
 - Proxy mechanisms are also defined, and RFC 7252 details a basic HTTP mapping for CoAP. 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.



3.6.2 Message Queuing Telemetry Transport (MQTT)

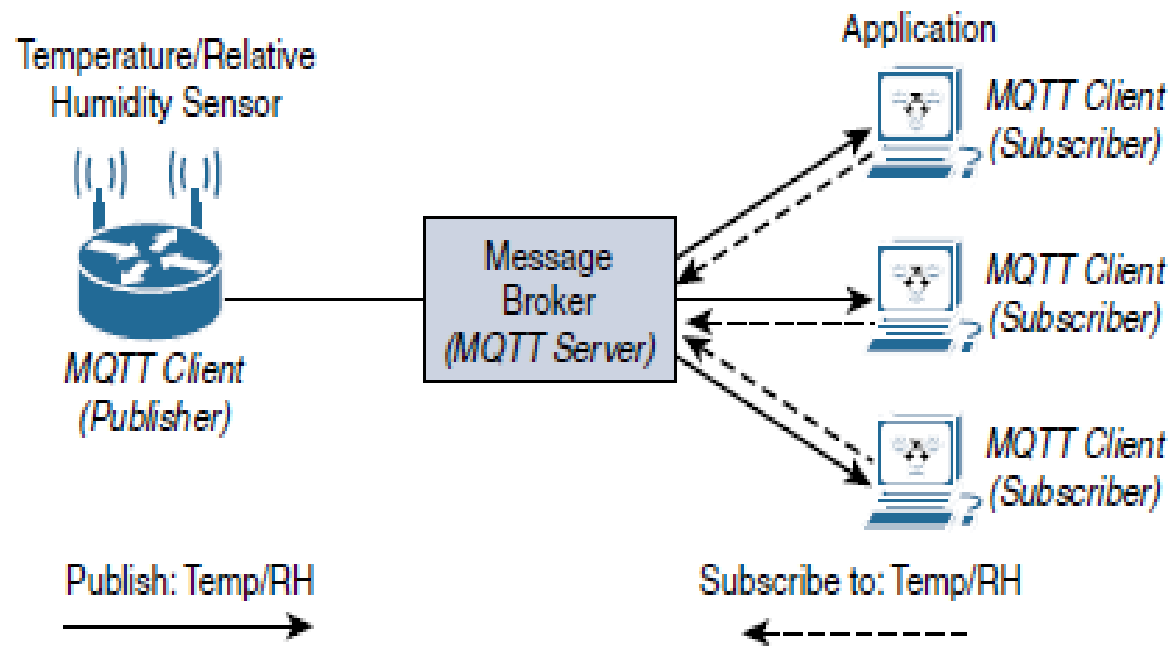


Figure 6-10 MQTT Publish/Subscribe Framework

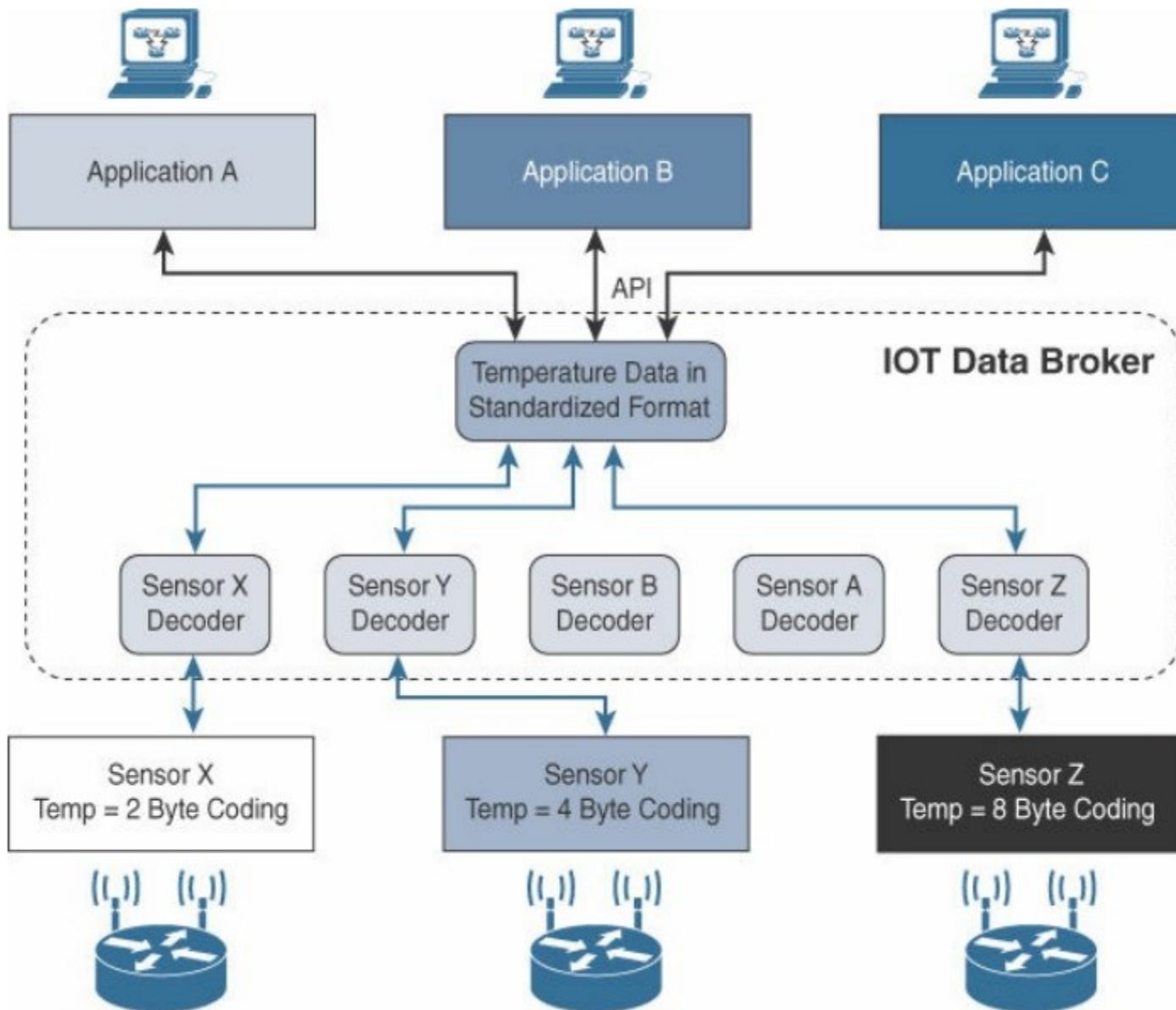


Figure 6-1 *IoT Data Broker*

- ▶ An MQTT client can act as a publisher to send data (or resource information) to an MQTT server acting as an MQTT message broker.
- ▶ 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.
- ▶ With MQTT, clients can subscribe to all data (using a wildcard character) or specific data from the information tree of a publisher.
- ▶ MQTT control packets run over a TCP transport using port 1883.
- ▶ MQTT is a lightweight protocol because each control packet consists of a 2-byte fixed header with optional variable header fields and optional payload.
- ▶ You should note that a control packet can contain a payload up to 256 MB



MQTT message

Table 6-2 *MQTT Message Types*

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

Message Type	Value	Flow	Description
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

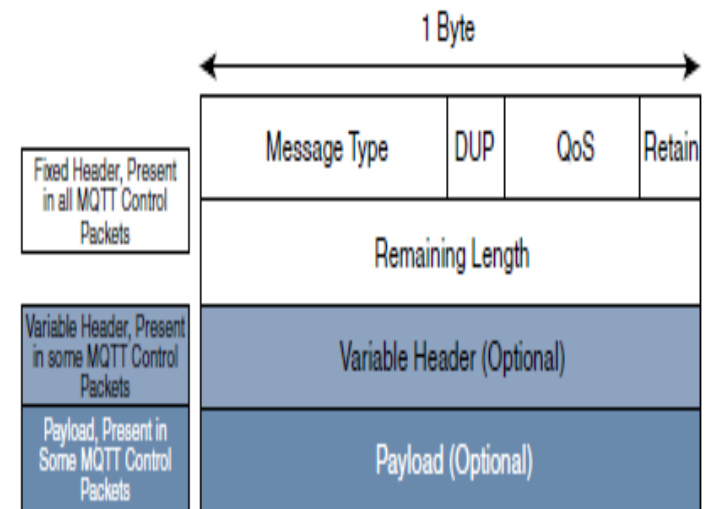


Figure 6-11 *MQTT Message Format*

Comparison between MQTT & CoAP

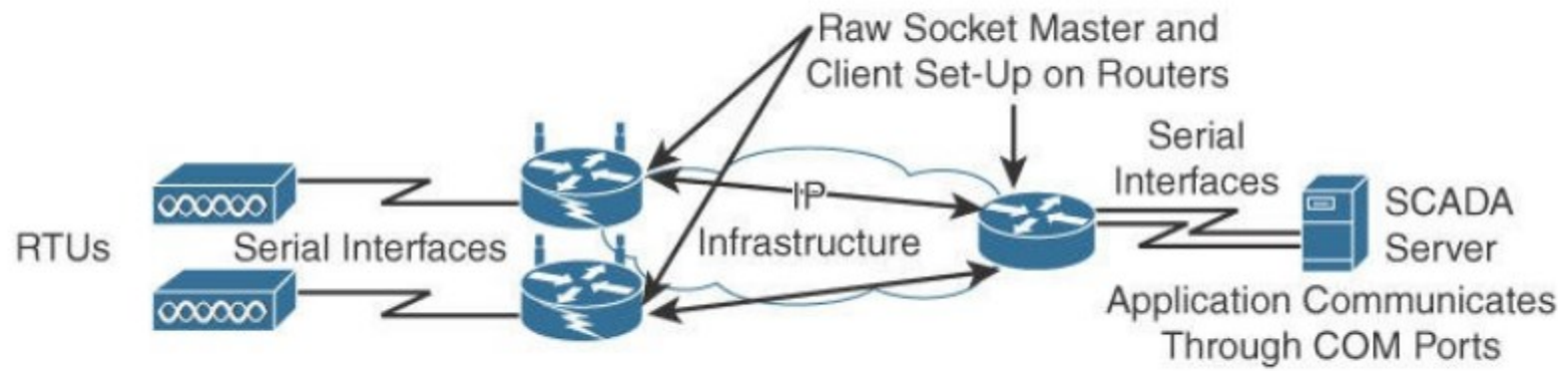
Table 6-3 *Comparison Between CoAP and MQTT*

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

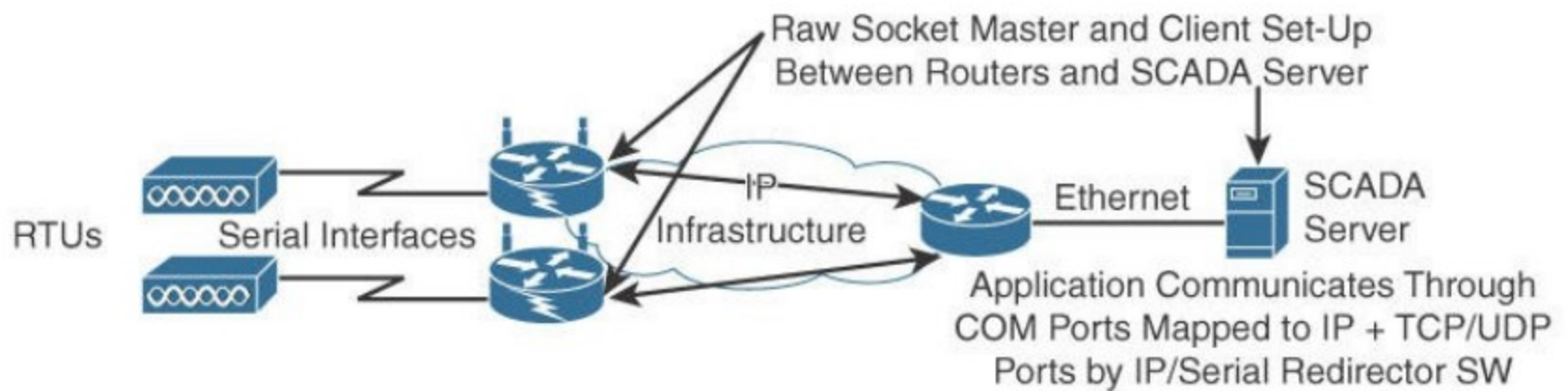
Tunneling Legacy SCADA over IP Networks

- ▶ SCADA servers
- ▶ Remote terminal units

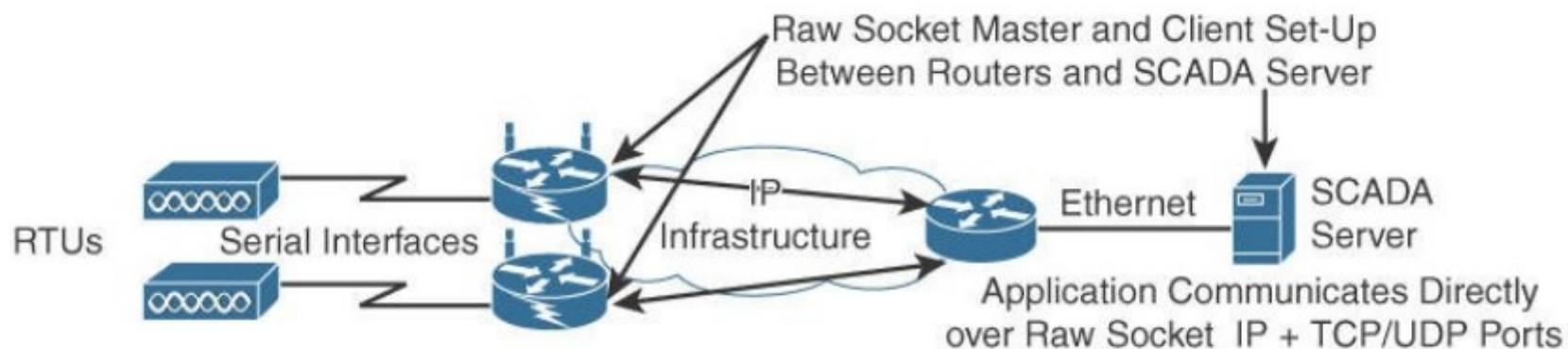




Scenario A: Raw Socket between Routers – no change on SCADA server



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



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

SCADA Protocol Translation

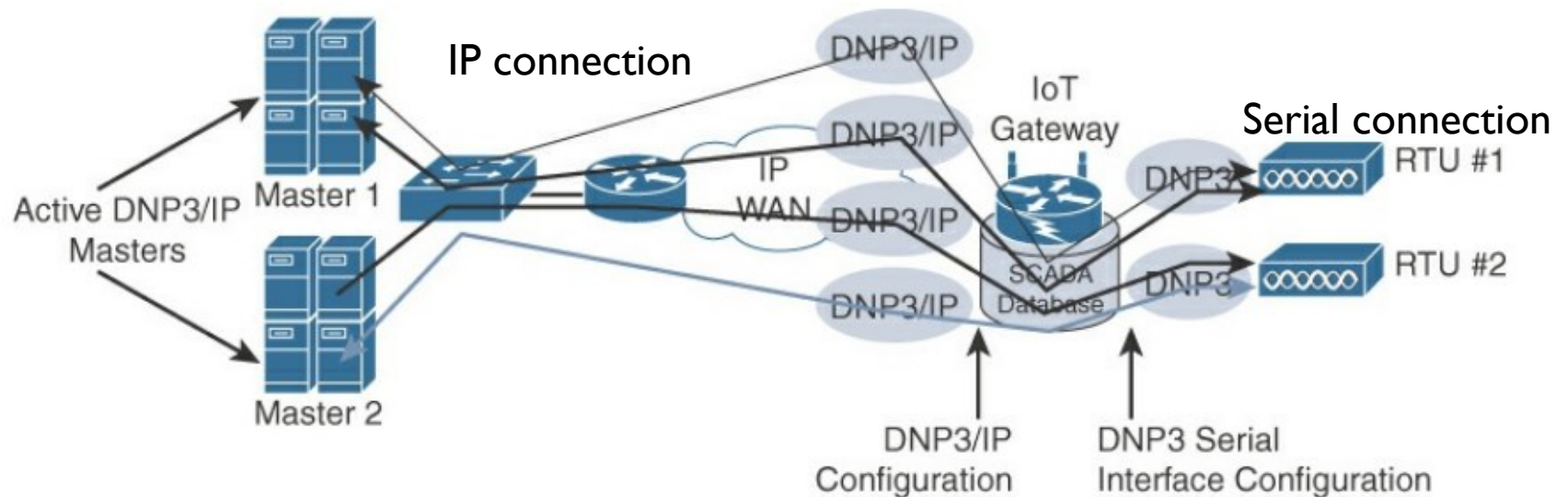


Figure 6-4 *DNP3 Protocol Translation*

SCADA Transport over LLNs with MAP-T

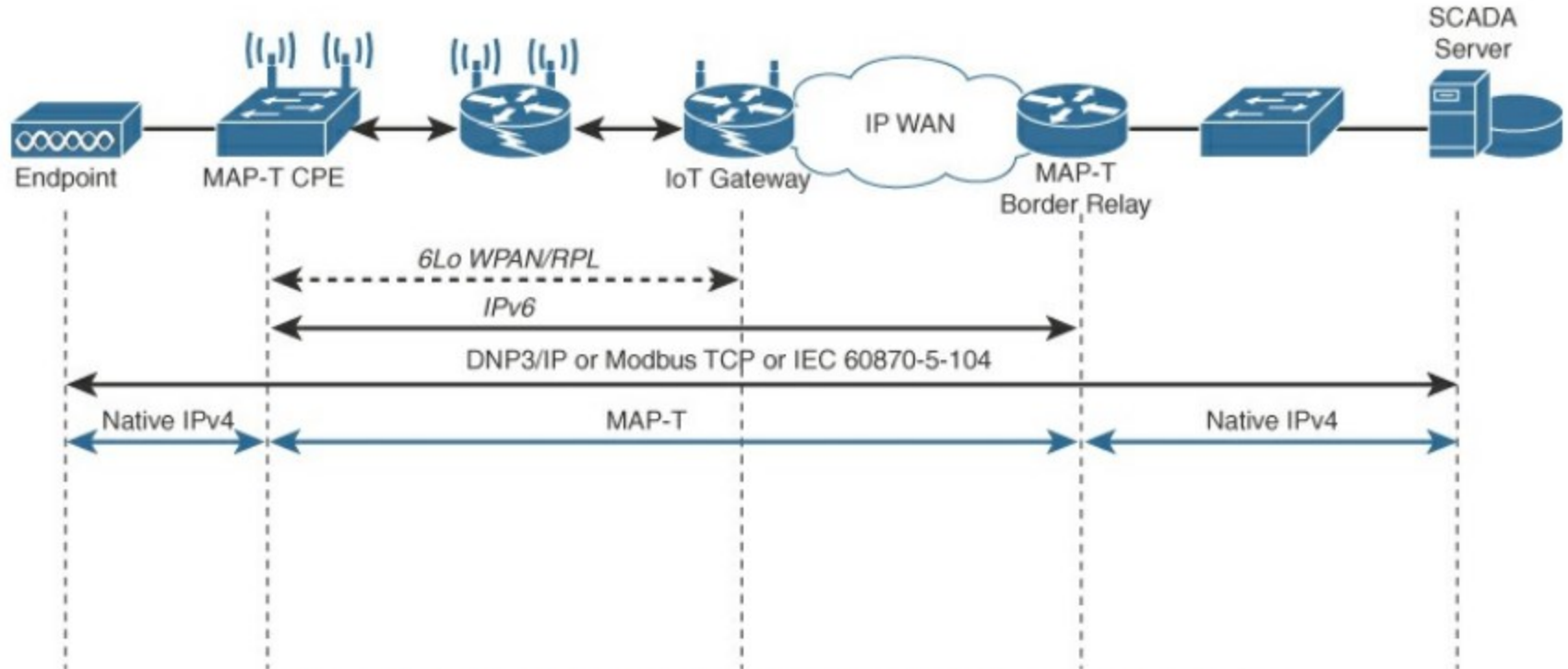


Figure 6-5 DNP3 Protocol over 6LoWPAN Networks with MAP-T