

LoRa (Long Range)

What are LoRa® and LoRaWAN®?

- LoRa is an RF modulation technology for **low-power**, wide area networks (LPWANs).
- The name, **LoRa**, is a reference to the extremely long-range data links that this technology enables.
- **Created by Semtech to standardize LPWANs**, LoRa provides for long-range communications: **up to three miles (five kilometers) in urban areas, and up to 10 miles (15 kilometers) or more in rural areas (line of sight)**.
- A key characteristic of the LoRa-based solutions **is ultra-low power requirements**, which allows for the creation of battery-operated devices that can last for up to 10 years.
- **Deployed in a star topology**, a network based on the open LoRaWAN protocol is perfect for applications that require **long-range or deep in-building communication** among a large number of devices that have low power requirements and that collect small amounts of data.

- Consider the differences between LoRa and other network technologies that are typically used in IoT or traditional machine-to-machine (M2M) connectivity solutions:


<u>Traditional Cellular</u> Long Range High Data Rates Low Battery Life High Cost	<div><p>LPWAN (3-5B in 2022)</p><p>Long Range Low Data Rates Long Battery Life Low Cost High Capacity Potential</p></div>	<u>Cat-M1</u> Long Range High Data Rates Low Battery Life Medium Cost
<u>Local Area Network</u> (Wi-Fi) Short Range High Data Rates Low Battery Life Medium Cost	<u>Narrow-Band IoT</u> (NB-IoT) Stationary Devices Short Range (indoor coverage) Low Data Rates Good Battery Life Low Cost	<u>Personal Area Network</u> (Bluetooth®) Very Short Range Low data rates Good Battery Life Low Cost

Figure 1: IoT Technologies

highlights some important advantages of deploying a LoRaWAN network:

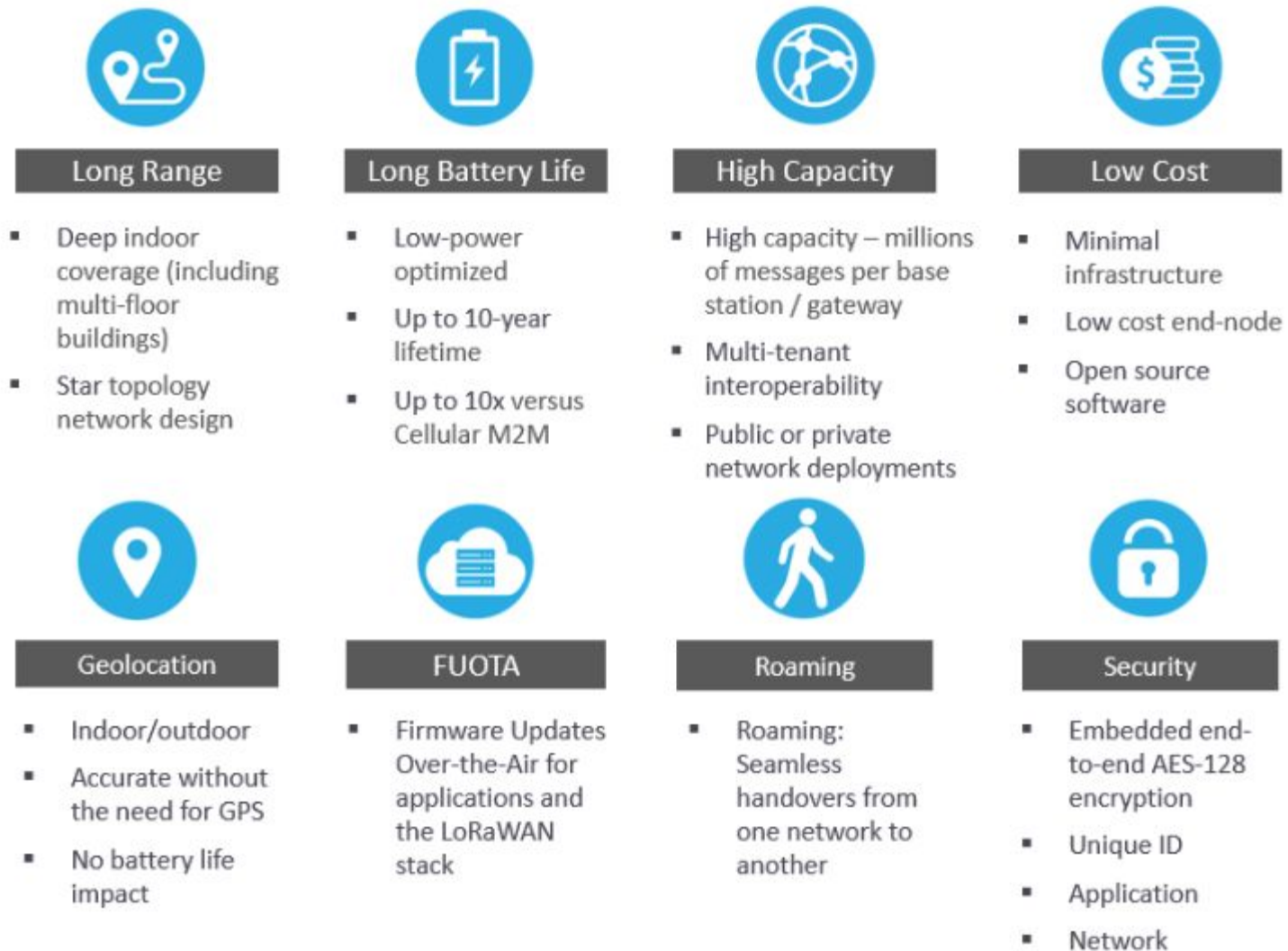
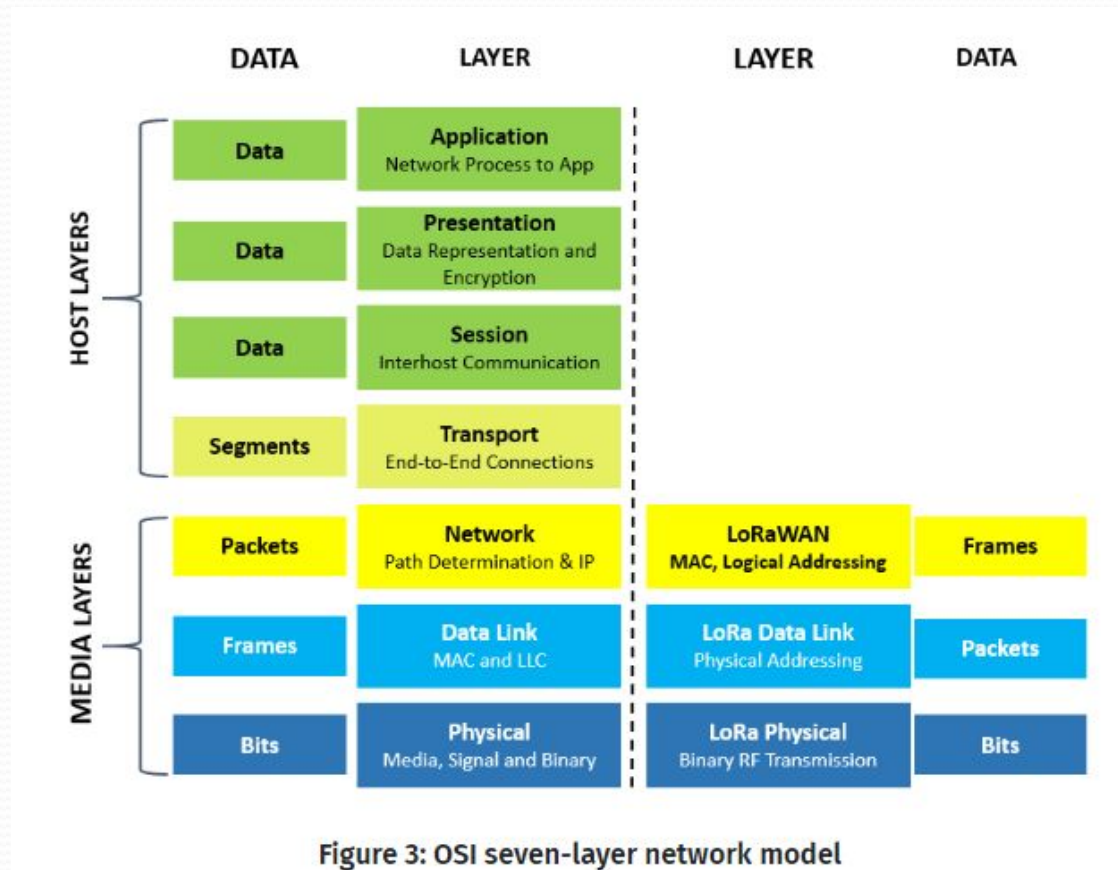


Figure 2: Advantages of deploying a LoRaWAN network

Radio Modulation and LoRa

- A proprietary spread-spectrum modulation technique derived from existing **Chirp Spread Spectrum (CSS) technology**, LoRa offers a **trade-off between sensitivity and data rate**, while operating in a fixed-bandwidth channel of either **125 KHz or 500 KHz (for uplink channels), and 500 KHz (for downlink channels)**.
- Additionally, LoRa uses orthogonal spreading factors.
- This allows the network to preserve the battery life of connected end nodes by making adaptive optimizations of an individual end node's power levels and data rates.
- For example, an end device located close to a gateway should transmit data at a low spreading factor, since very little link budget is needed. However, an end device located several miles from a gateway will need to transmit with a much higher spreading factor. This higher spreading factor provides increased processing gain, and higher reception sensitivity, although the data rate will, necessarily, be lower.

- **LoRa is purely a physical (PHY), or “bits” layer implementation,** as defined by the OSI seven-layer Network Model, depicted in Figure . Instead of cabling, the **air is used as a medium for transporting LoRa radio waves from an RF transmitter in an IoT device to an RF receiver in a gateway,** and vice versa.



LoRaWAN Network Fundamentals

- To fully understand LoRaWAN networks, we will start with a look at the technology stack.
- As shown in Figure 7, LoRa is the physical (PHY) layer, i.e., the wireless modulation used to create the long-range communication link. **LoRaWAN is an open networking protocol** that delivers secure bi-directional communication, mobility, and localization services standardized and maintained by the LoRa Alliance.

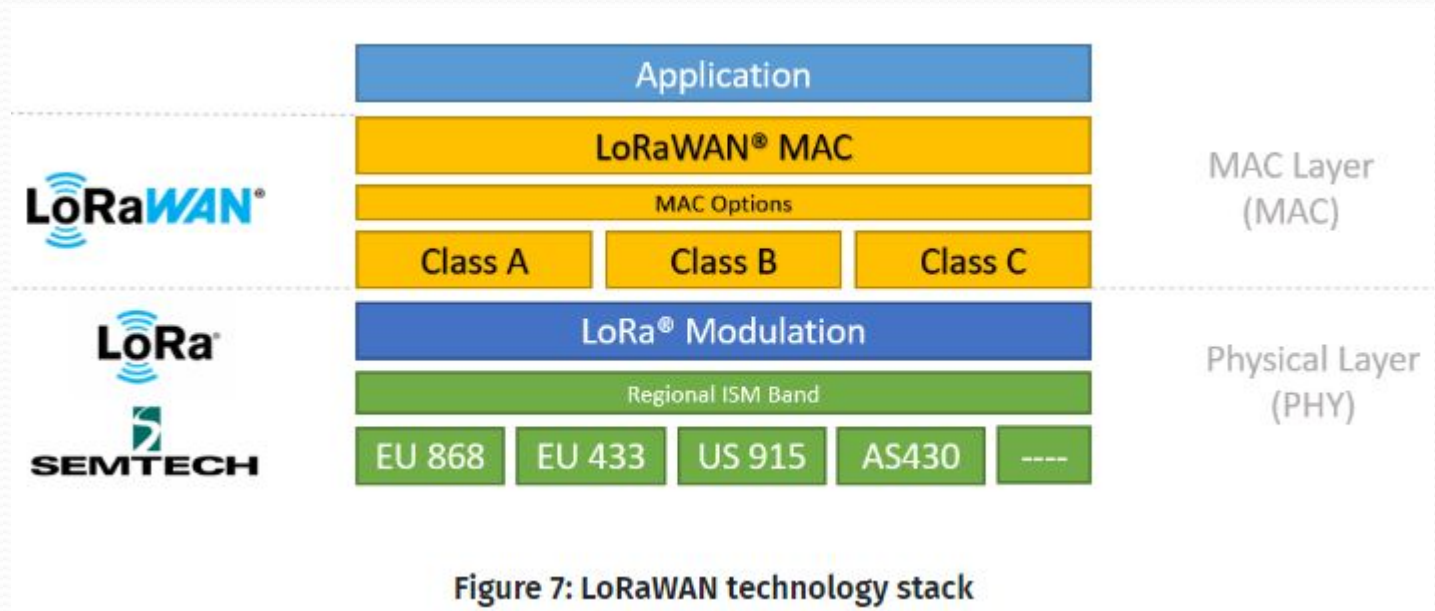
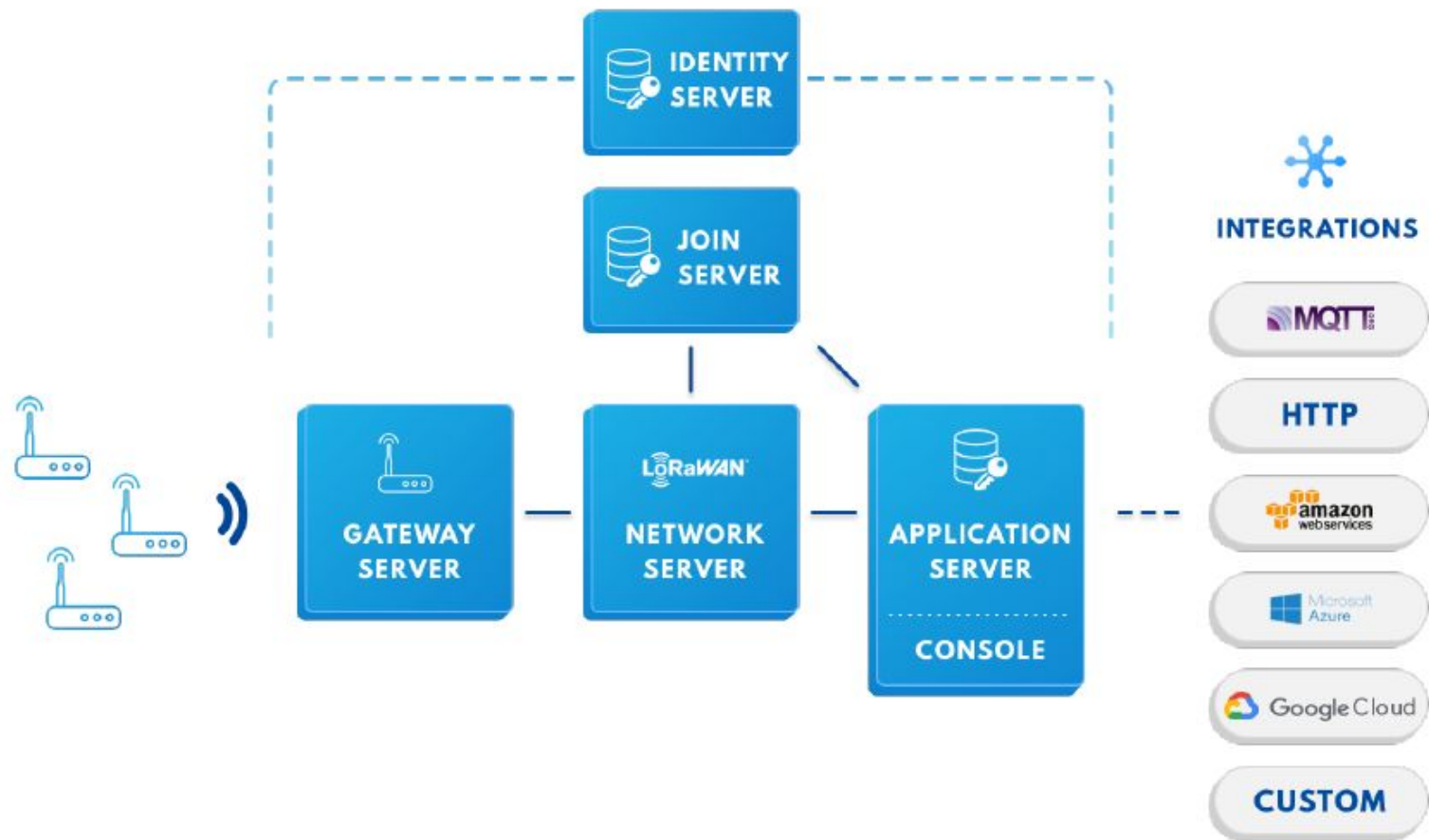
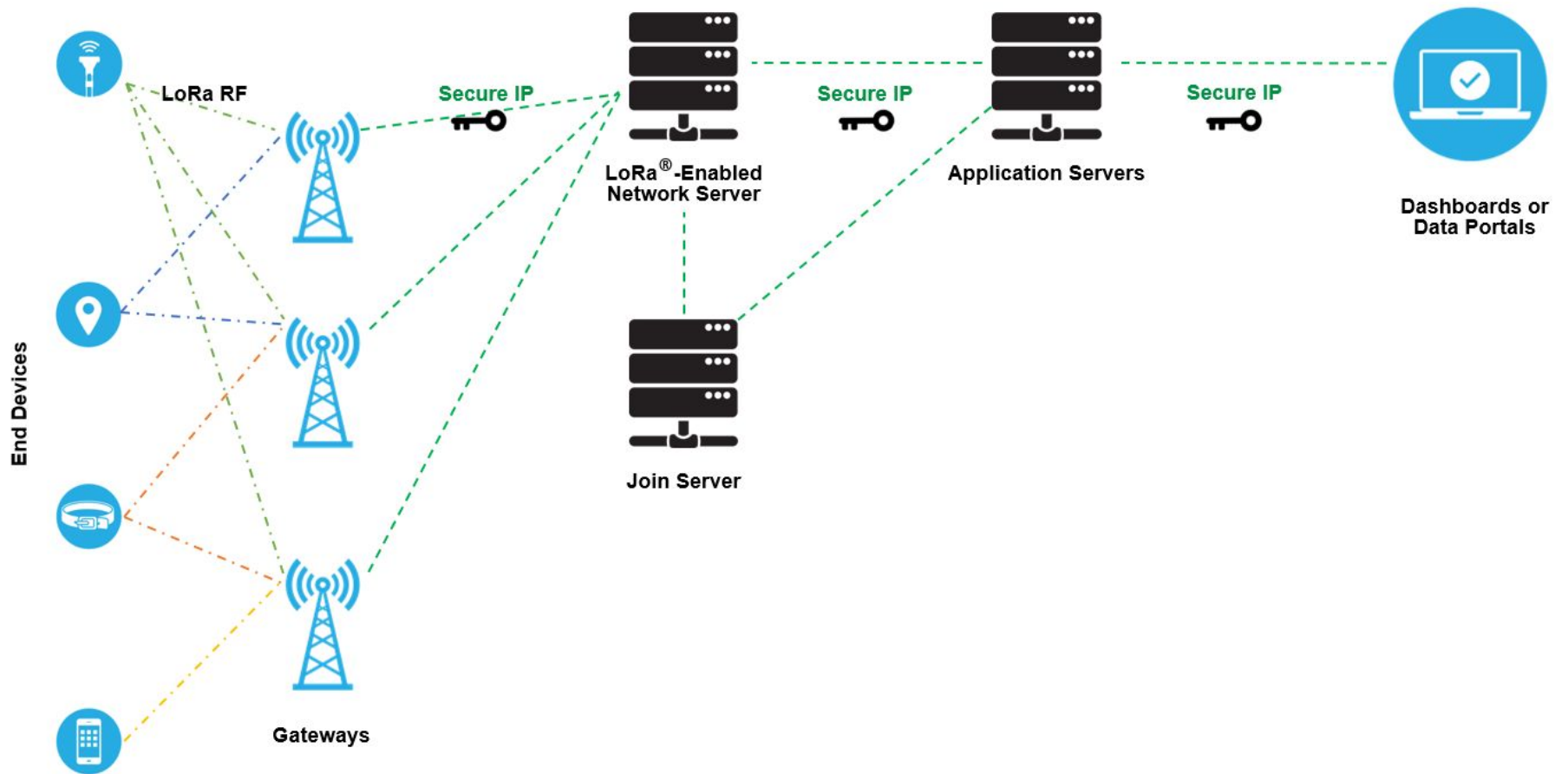


Figure 7: LoRaWAN technology stack

LoRaWAN Network Fundamentals

LoRaWAN Components






LoRa-based End Devices

- A LoRaWAN-enabled **end device** is a sensor or an actuator which is wirelessly connected to a LoRaWAN network through radio gateways using LoRa RF Modulation.
- In the majority of applications, an end device is an autonomous, often battery-operated sensor that digitizes physical conditions and environmental events. Typical use cases for an actuator include: street lighting, wireless locks, water valve shut off, leak prevention, among others.
- **When they are being manufactured, LoRa-based devices are assigned several unique identifiers.** These identifiers are used to securely activate and administer the device, to ensure the safe transport of packets over a private or public network and to deliver encrypted data to the Cloud.

LoRaWAN Gateways

- A LoRaWAN **gateway** receives LoRa modulated RF messages from any end device in hearing distance and forwards these data messages to the LoRaWAN network server (LNS), which is connected through an IP backbone. There is no fixed association between an end device and a specific gateway. Instead, the same sensor can be served by multiple gateways in the area. With LoRaWAN, each uplink packet sent by the end-device will be received by all gateways within reach, as illustrated in Figure 10. This arrangement significantly reduces packet error rate (since the chances that at least one gateway will receive the message are very high), significantly reduces battery overhead for mobile/nomadic sensors, and allows for low-cost geolocation (assuming the gateways in question are geolocation-capable).
- The IP traffic from a gateway to the network server can be backhauled via Wi-Fi, hardwired Ethernet or via a Cellular connection. LoRaWAN gateways operate entirely at the physical layer and, in essence, are nothing but LoRa radio message forwarders. They only check the data integrity of each incoming LoRa RF message. If the integrity is not intact, that is, if the CRC is incorrect, the message will be dropped. If correct the gateway will forward it to the LNS, together with some metadata that includes the receive RSSI level of the message as well as an optional timestamp. For LoRaWAN downlinks, a gateway executes transmission requests coming from the LNS without any interpretation of the payload. Since multiple gateways can receive the same LoRa RF message from a single end device, the LNS performs data de-duplication and deletes all copies. Based on the RSSI levels of the identical messages, the network server typically selects the gateway that received the message with the best RSSI when transmitting a downlink message because that gateway is the one closest to the end device in question.

- 
- Furthermore, LoRa allows for **scalable, cost-optimized gateway implementation, depending on deployment objectives**. For example, in North America, 8-, 16-, and 64-channel gateways are available.
 - **The 8-channel gateways are the least expensive**. The type of gateway needed will depend on the use case. Eight- and 16-channel gateways are available for both indoor and outdoor use. Sixty-four channel gateways are only available in a carrier-grade variant. **This type of gateway is intended for deployment in such places as cell towers, the rooftops of very tall buildings, etc.**

Network Server

- The LoRaWAN network server (LNS) manages the entire network, dynamically controls the network parameters to adapt the system to ever-changing conditions, and **establishes secure 128-bit AES connections** for the transport of both the end to end data (from LoRaWAN end device to the end users Application in the Cloud) as well as for the **control of traffic** that flows from the LoRaWAN end device to the LNS (and back). The network server ensures the authenticity of every sensor on the network and the integrity of every message. At the same time, the network server cannot see or access the application data.
- In general, all LoRaWAN network servers share the following features:
 - Device address checking
 - Frame authentication and frame counter management
 - Acknowledgements of received messages
 - Adapting data rates using the ADR protocol
 - Responding to all MAC layer requests coming from the device,
 - Forwarding uplink application payloads to the appropriate application servers
 - Queuing of downlink payloads coming from any Application Server to any device connected to the network
 - **Forwarding Join-request and Join-accept messages between the devices and the join server**

Application Servers

- Application servers are **responsible for securely handling, managing and interpreting sensor application data.** They also generate all the application-layer downlink payloads to the connected end devices.

Join Server

- The join server manages the **over-the-air activation process** for end devices to be added to the network.
- The join server contains the information required to process **uplink *join-request* frames** and generate the **downlink *join-accept* frames**.
- It signals to the network server which application server should be connected to the end-device, and **performs the network and application session encryption key derivations**.
- It communicates the **Network Session Key of the device to the network server, and the Application Session Key to the corresponding application server**


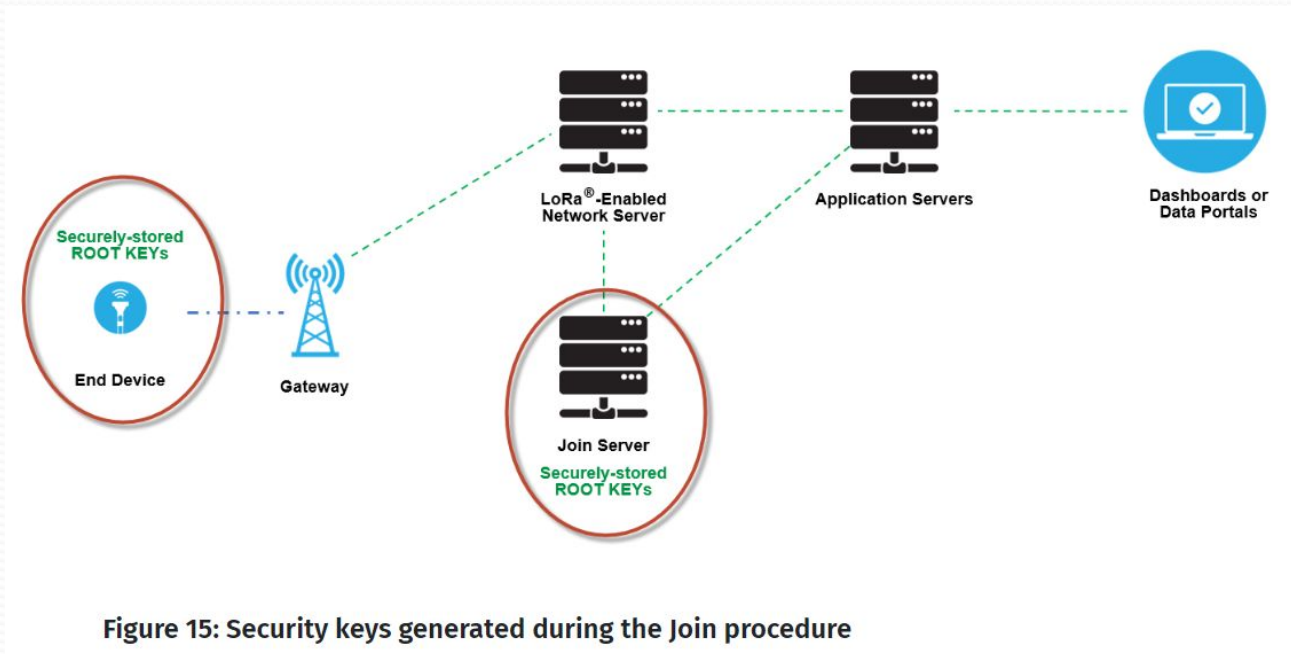
- 
- An end device can connect to a network with LoRaWAN in two ways:
 - **Over-the-air Activation (OTAA):** A device has to establish a network key and an application session key to connect with the network.
 - **Activation by Personalization (ABP):** A device is hardcoded with keys needed to communicate with the network, making for a less secure but easier connection.

Table 3: Activation Types

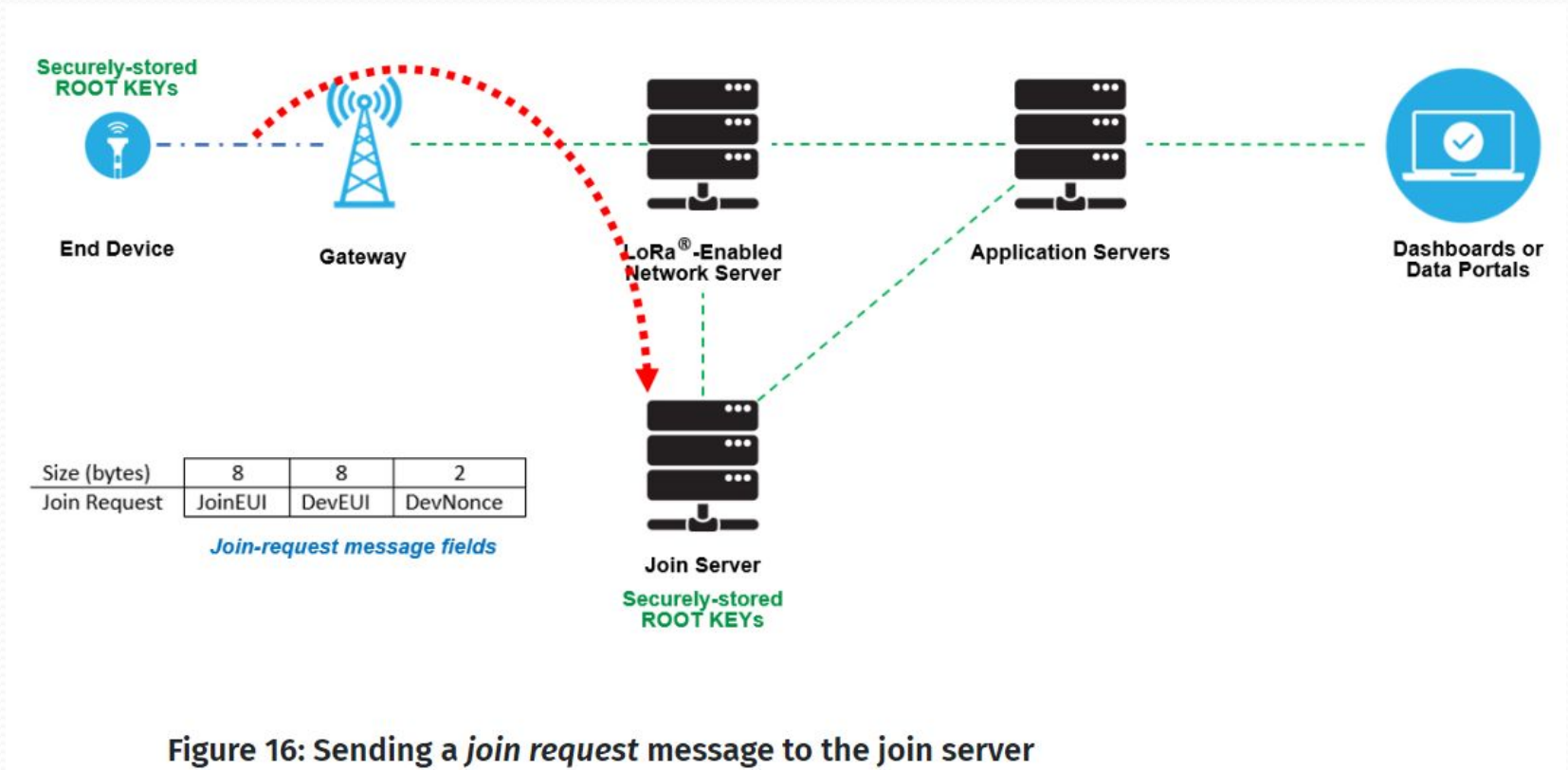
Over-the-Air Activation (OTAA)	Activation by Personalization (ABP)
<ul style="list-style-type: none">• Device manufacturers autonomously generate essential provisioning parameters• Secure keys (session-long and derived) can be renewed regularly• Devices can store multiple “identities” to dynamically and securely switch networks and operators during its lifetime• High-grade, tamper-proof security options are available	<ul style="list-style-type: none">• A simplified (less secure) commissioning process• IDs and Keys are personalized at fabrication• Devices become immediately functional upon powering up; the Join procedure is skipped• Devices are tied to a specific network/service; the NetID is a portion of the device network address

The Join Procedure

- We will begin with the security keys, as illustrated in Figure 15. Individual root keys are securely stored on the end devices, and matching keys are securely stored on the join server.



- The end device sends a *join request* message to the join server, as illustrated in Figure 16.



- After the join server authenticates the device requesting to join the network, it returns a *join accept message* to the device, as illustrated in Figure 17.

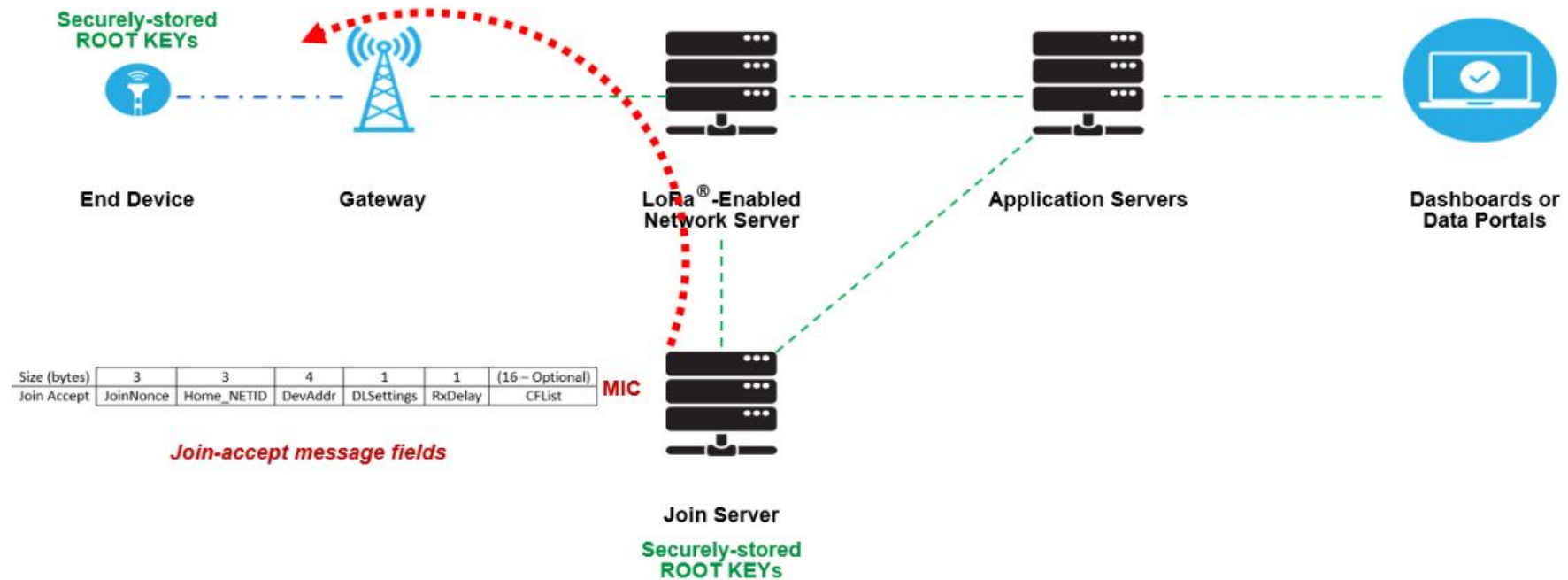


Figure 17: Sending a *join accept* message to an end device

- Next, the end device **derives session keys locally**, based on the DevEUI, Join EUI, DevNonce, root keys and fields in the join request and join accept messages.
- On its end, the join server also derives session keys from the serial IDs, root keys and fields in join requests and join accept messages.
- Finally, the **join server shares session keys with network and application servers**, as illustrated in Figure 18.

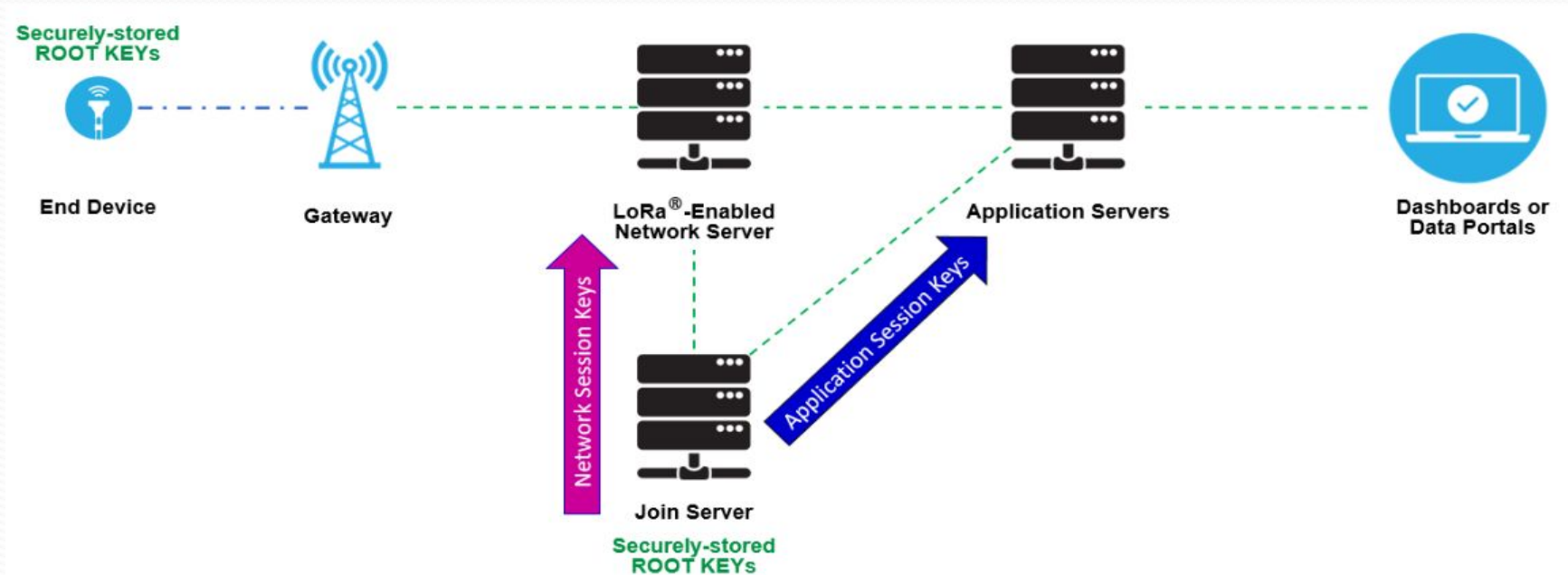
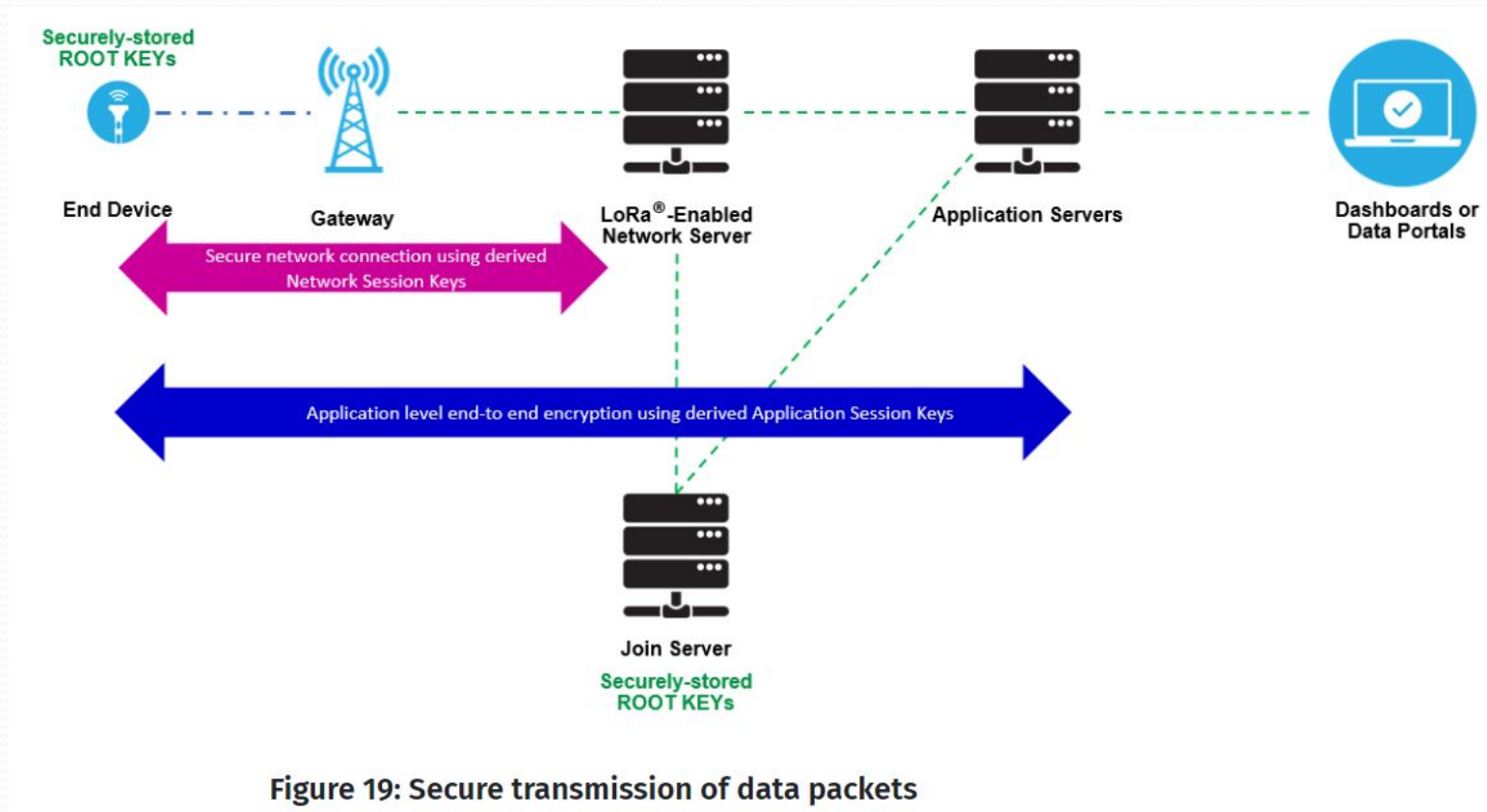
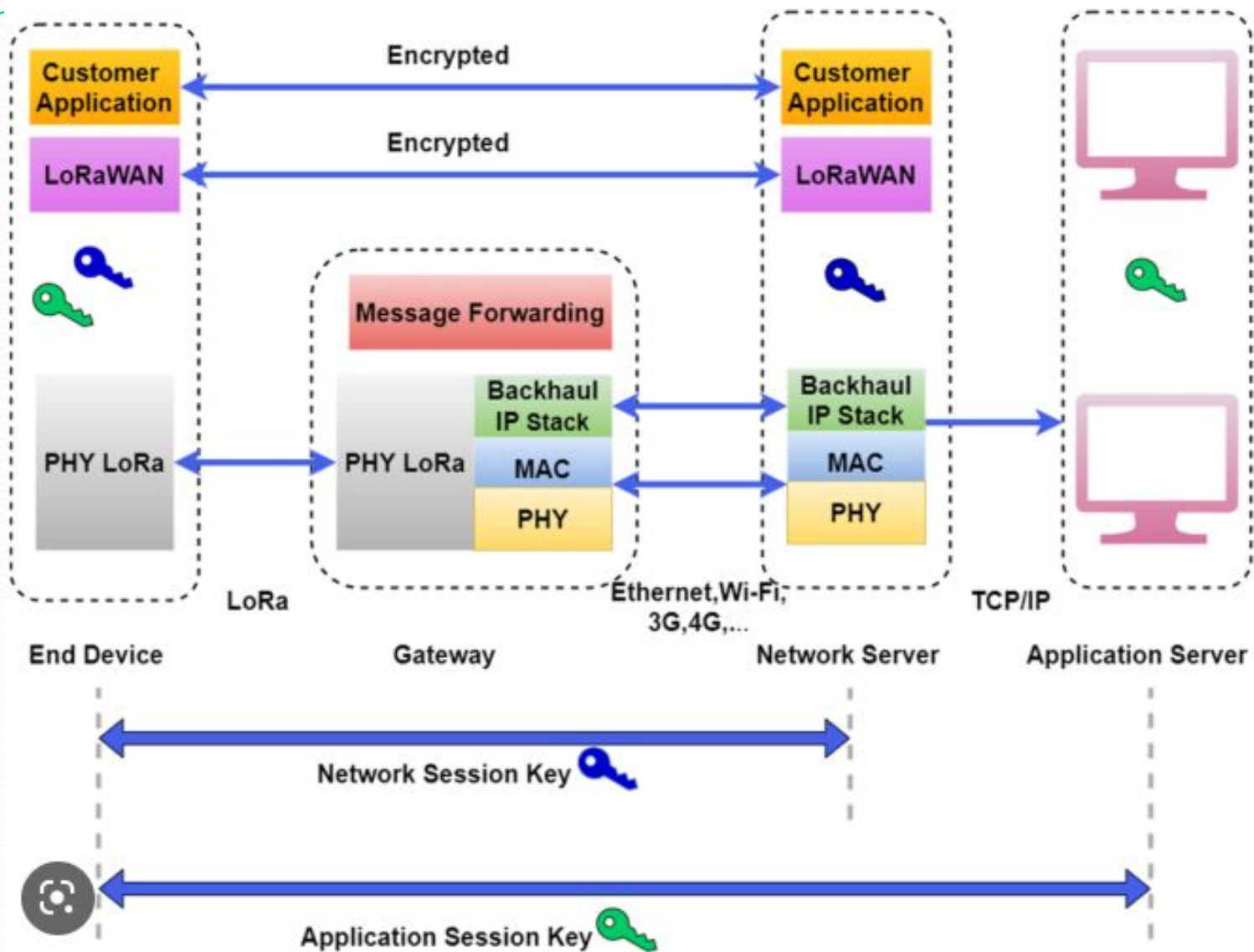


Figure 18: Session keys are shared with the network server and the application server

- Figure 19 illustrates the security of data packet transmissions. The control traffic between the end device and the network server is secured with a **128-bit AES Network Session Key (NwkSKey)**.
- The data traffic that travels between the end device and the application server, is secured with a **128-bit Application Session Key (AppSKey)**.
- **This method ensures that neither the gateway nor the network server can read the user data.**

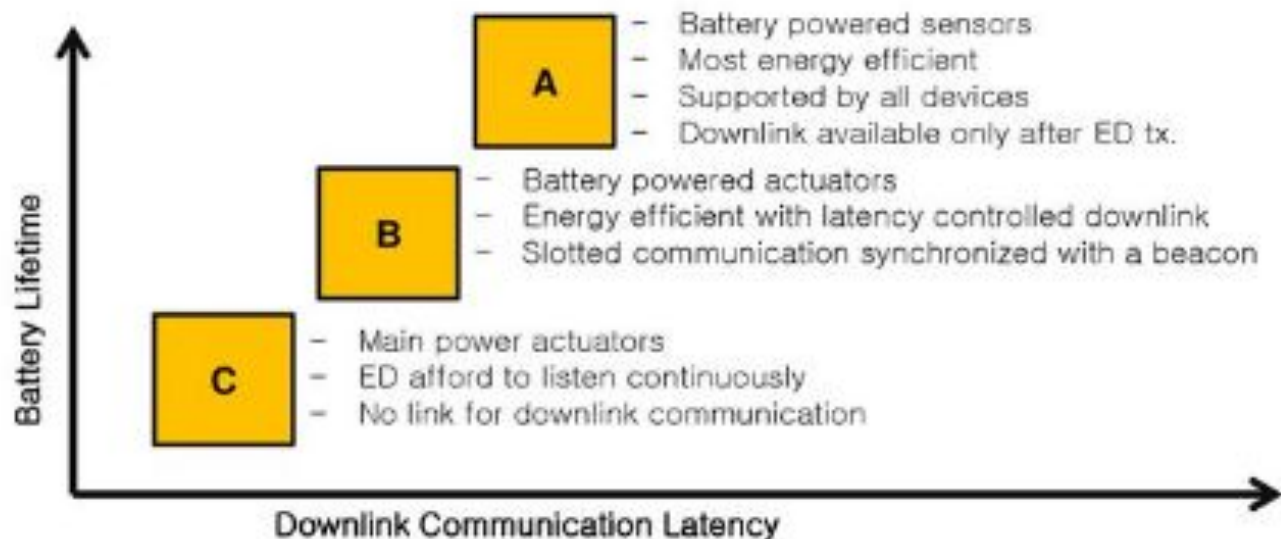




Device Classes: A, B and C

LoRa End Nodes

- Before an ED join a LoRaWAN, it must be activated:
 - Over the Air Activation (OTAA)
 - Activation By Personalization (ABP)



Device Classes: A, B and C

Class A devices

Class A devices are

- meant to have the longest battery life
- sleeping most of the time, and thus, do not listen regularly to the network
- can receive a message from the network (**downlink**) only as a response to a message they have just sent (**uplink**)
- Examples of Class A devices are fire alarms, flood detectors, intrusion detectors etc

Class B devices

Class B devices are

- meant to have average battery life as the power consumption will be higher than class A devices
- listen to the network periodically, meaning the network can initiate the communication and the device sends the data only when it is asked, with a few seconds of latency
- can initiate a **downlink** message from the network without waiting for an **uplink**
- Examples of Class B devices are metering like temperature, Humidity and moisture etc

Class C devices

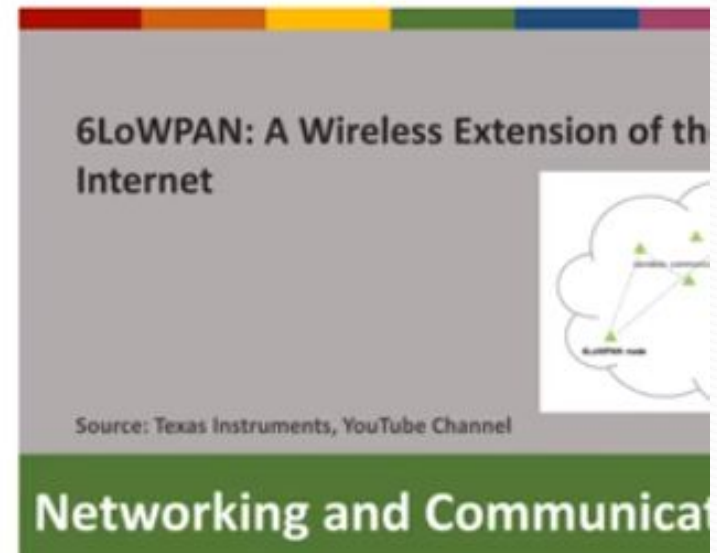
Class C devices are

- Highly power consuming
- continuously listening to the network except when they are sending **uplinks**
- No latency for **downlink** messages
- Examples of Class C devices are traffic monitoring and any near real time applications

6LoWPAN

Introduction to 6LoWPAN

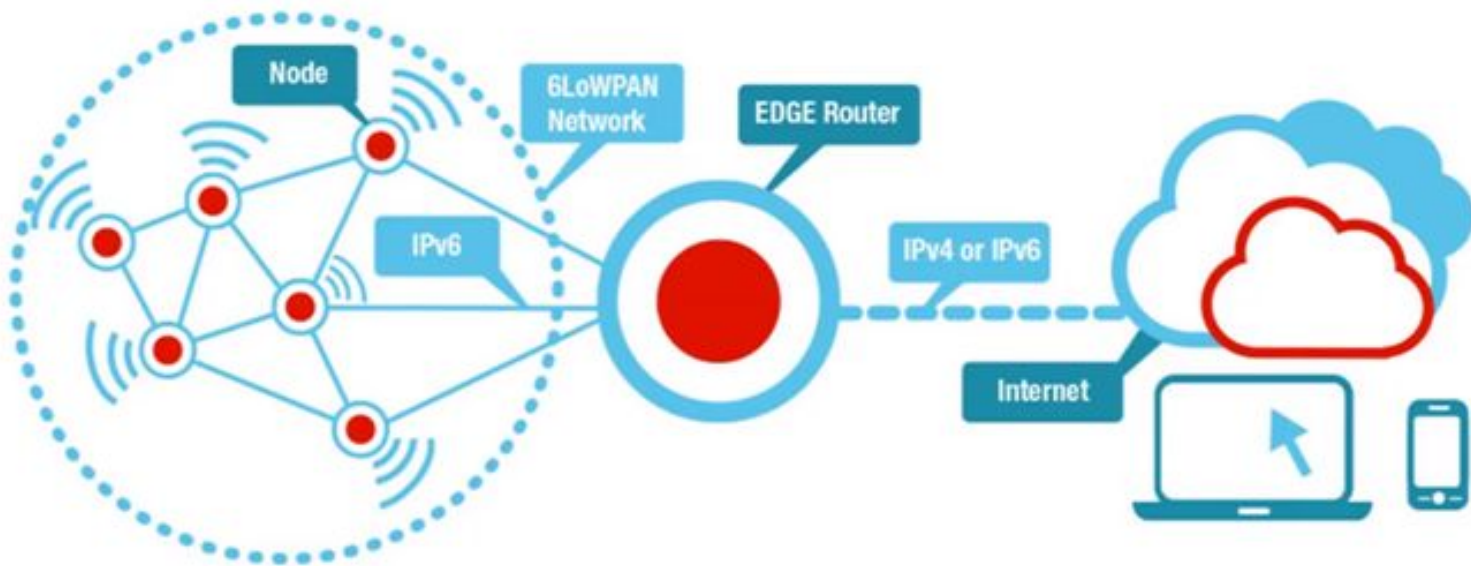
- Low-power Wireless Personal Area Networks over IPv6.
- 6LoWPAN can be seen as a combination of two protocols: Internet Protocol version 6 (IPv6) and Low-Power Wireless Personal Network (LoWPAN).
- Allows for the smallest devices with limited processing ability to transmit information wirelessly using an Internet protocol.
- Allows low-power devices to connect to the Internet.
- Created by the Internet Engineering Task Force (IETF) - RFC 5933 and RFC 4919.



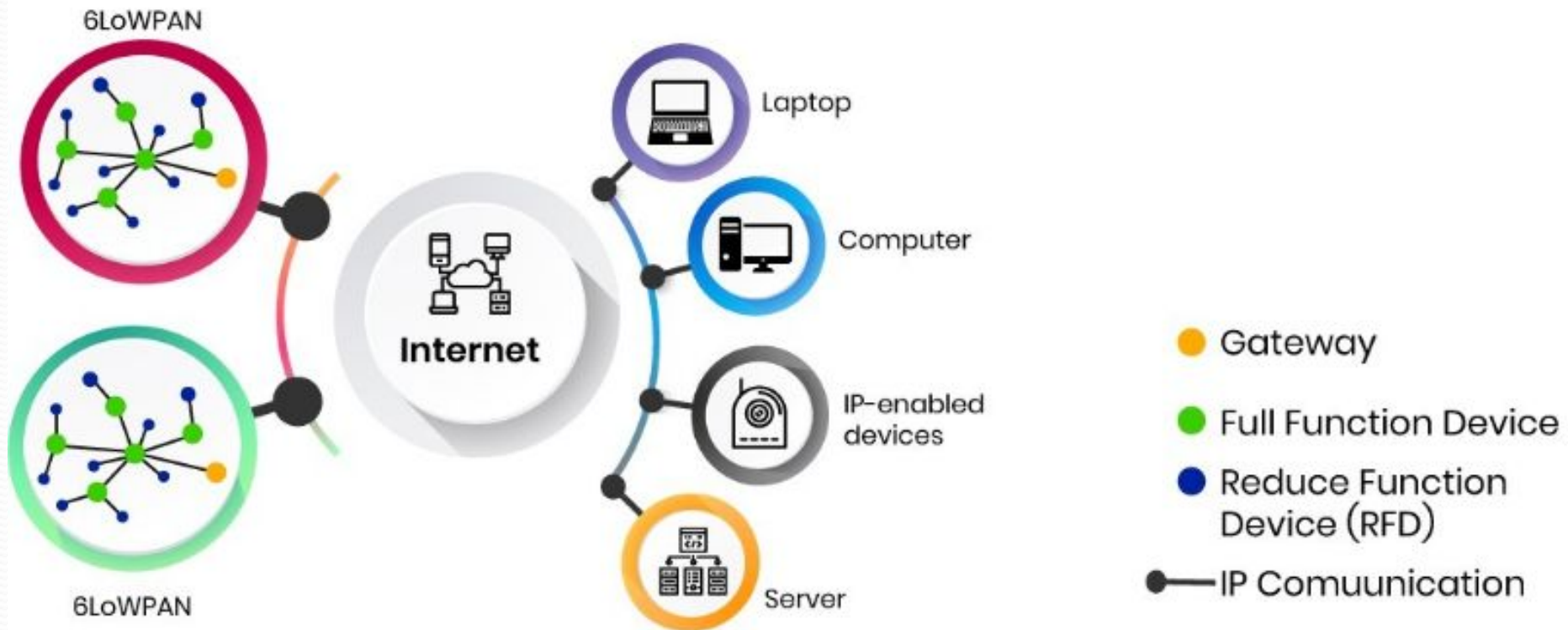
6LoWPAN

IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN)

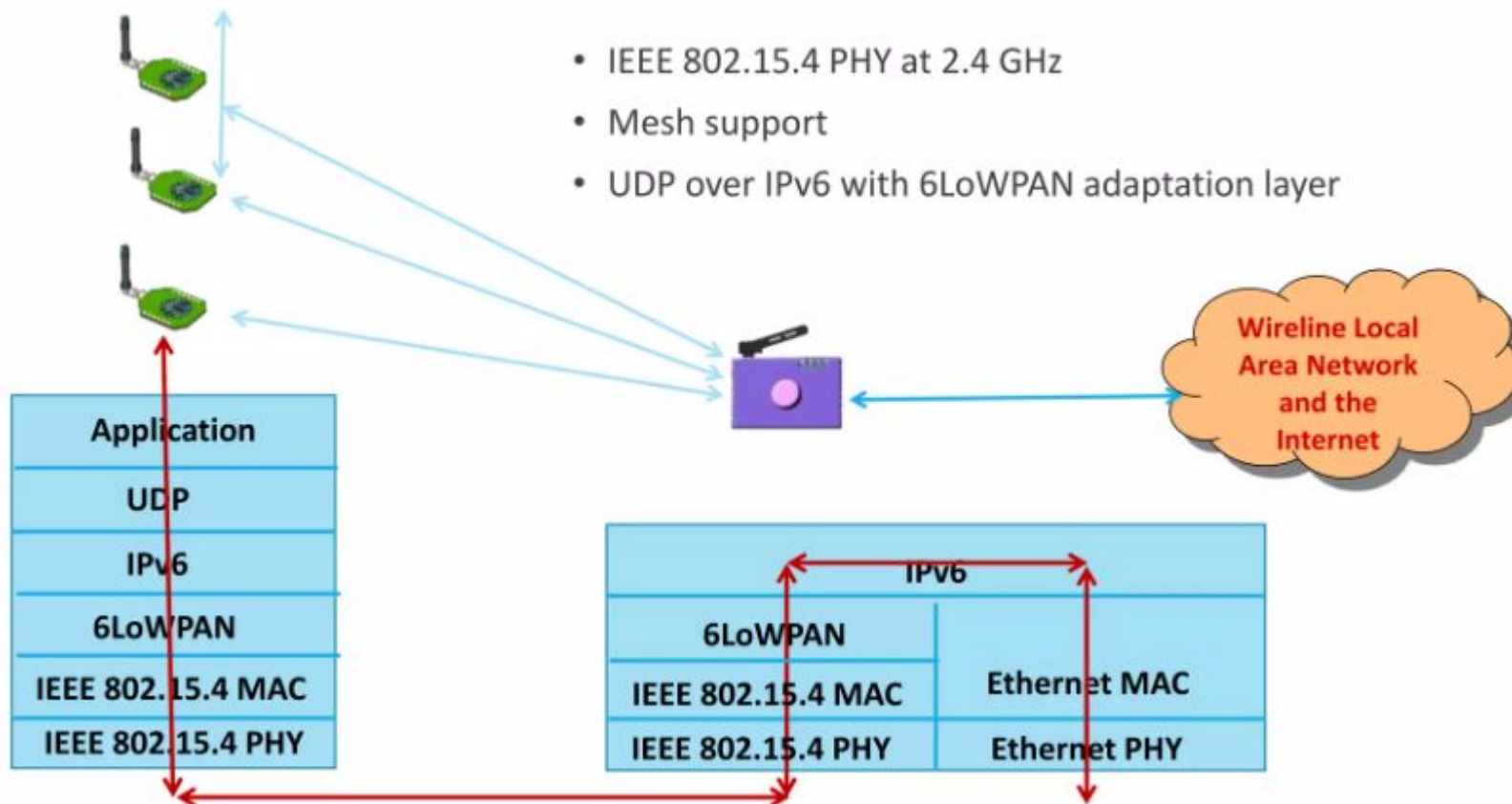
IPv6 network with a 6LoWPAN mesh network



6LoWPAN Architecture



The Network Protocol Stack



6LoWPAN Architecture

Common topologies include star, mesh, and combinations of star and mesh

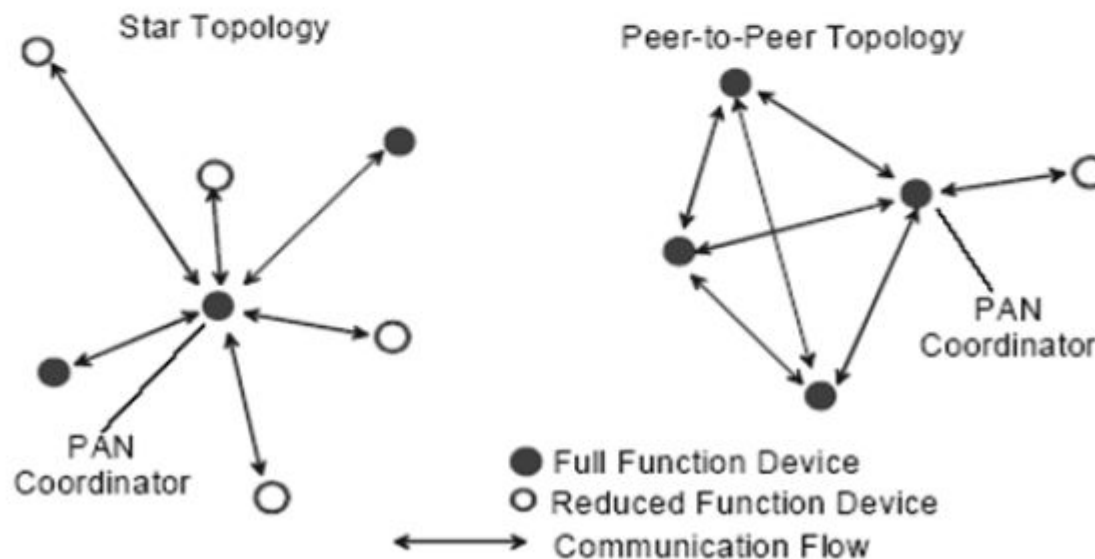
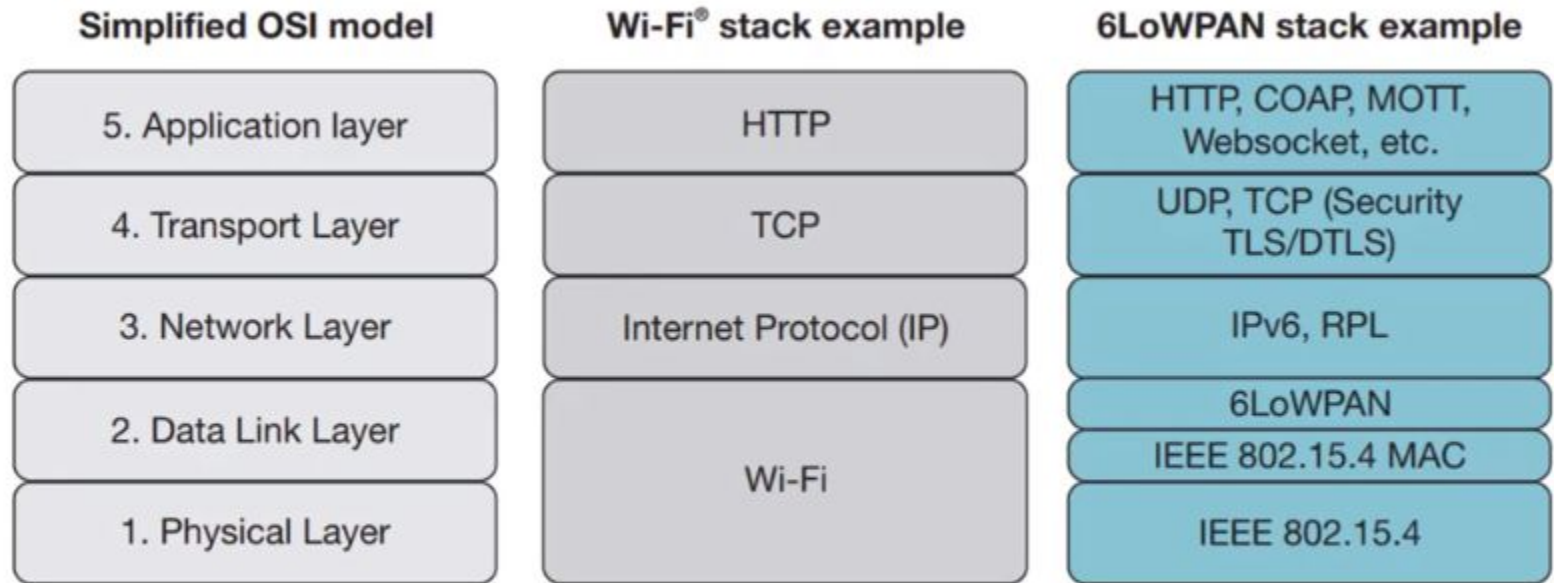


Figure 1—Star and peer-to-peer topology examples

IPv6 network with a 6LoWPAN mesh network



In 6LoWPAN, the Adaptation Layer is responsible for packaging and transporting the IP packets from the Internet layer to the Physical layer (Datalink layer and Network layer in OSI model), so the end-to-end connection is addressable, and a router can be used for routing tasks.

6LoWPAN Protocol Stack

TCP/IP Protocol Stack

HTTP		RTP	
TCP	UDP	ICMP	
IP			
Ethernet MAC			
Ethernet PHY			

Application

Transport

Network

Data Link

Physical

6LoWPAN Protocol Stack

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

6LoWPAN Security Perspective

- It is anticipated that the Internet of Things, IoT will offer hackers a huge opportunity to take control of poorly secured devices and also use them to help attack other networks and devices.
- Accordingly security is a major issue for any standard like 6LoWPAN, and it uses AES-128 link layer security which is defined in IEEE 802.15.4. This provides link authentication and encryption.
- Further security is provided by the transport layer security mechanisms that are also included. This is defined in RFC 5246 and runs over TCP.
- For systems where UDP is used the transport layer protocol defined under RFC 6347 can be used, although this may require some specific hardware requirements.

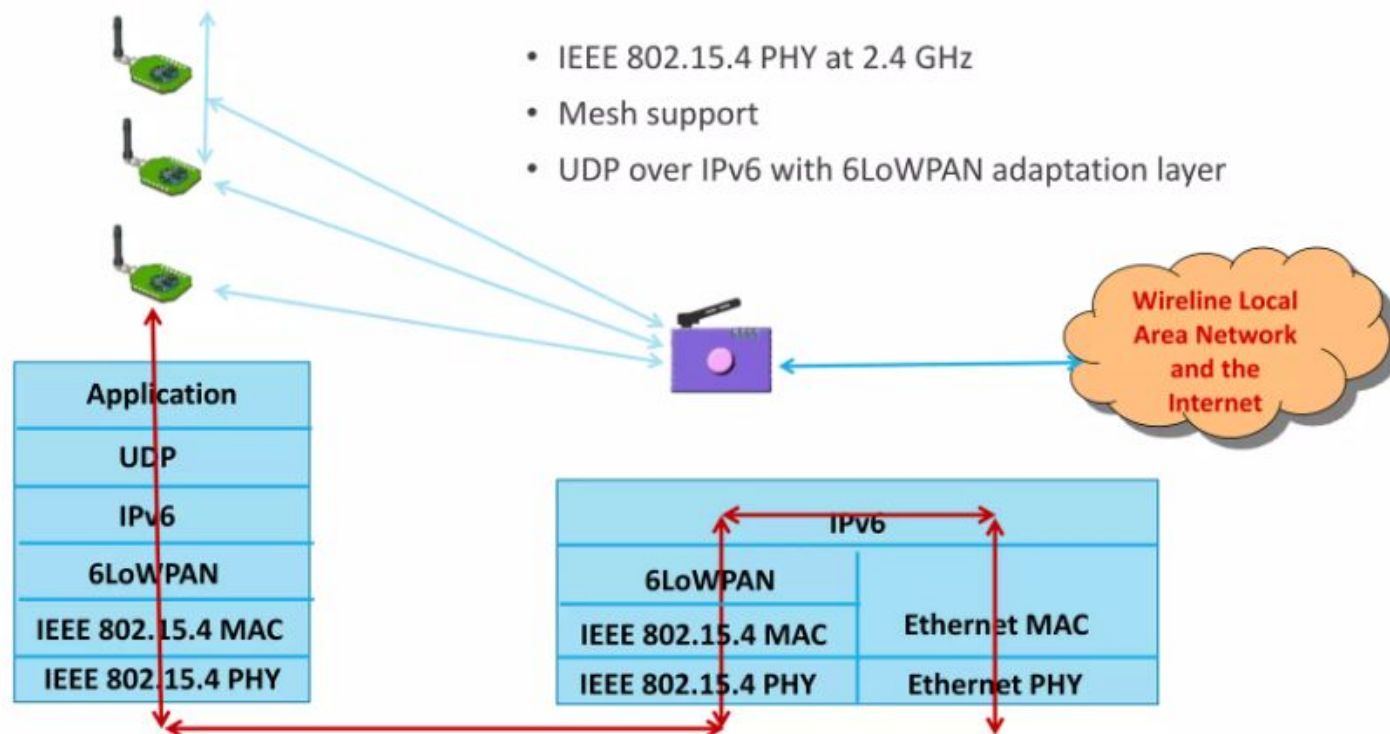
6LoWPAN Interoperability Perspective

- One key issue with IoT device is interoperability. It is vital that equipment from different manufacturers operates together.
- When testing for interoperability, it is necessary to ensure that all layers of the OSI stack are compatible. To ensure that this can be achieved there several different specifications that are applicable.
- 6LoWPAN is a wireless / IoT style standard that has quietly gained significant ground. Although initially aimed at usage with IEEE 802.15.4, it is equally able to operate with other wireless standards making it an ideal choice for many applications.

6LoWPAN Gateway

- LoWPAN is an acronym for **Low power Wireless Personal Area Networks**.
- The 6LoWPAN IoT gateway functions as a border router in a 6LoWPAN network, connecting a wireless IPv6 network to the Internet.

The Network Protocol Stack



Introduction

- Low Power and Lossy Networks (LLN) are resource constrained
- Routers are usually limited in terms of processing power, battery and memory, and their interconnects are characterised by unstable links with high loss rates, low data rates and low packet delivery rates
- The traffic patterns could be P2P or P2MP or MP2P
- Lossy means the packet drop rate will be high.

CoAP	Application layer
UDP	Transport layer
IPv6, RPL	Network layer / Routing
6LoWPAN	Adaptation layer
MAC 802.15.4	Datalink layer
PHY 802.15.4	Physical layer

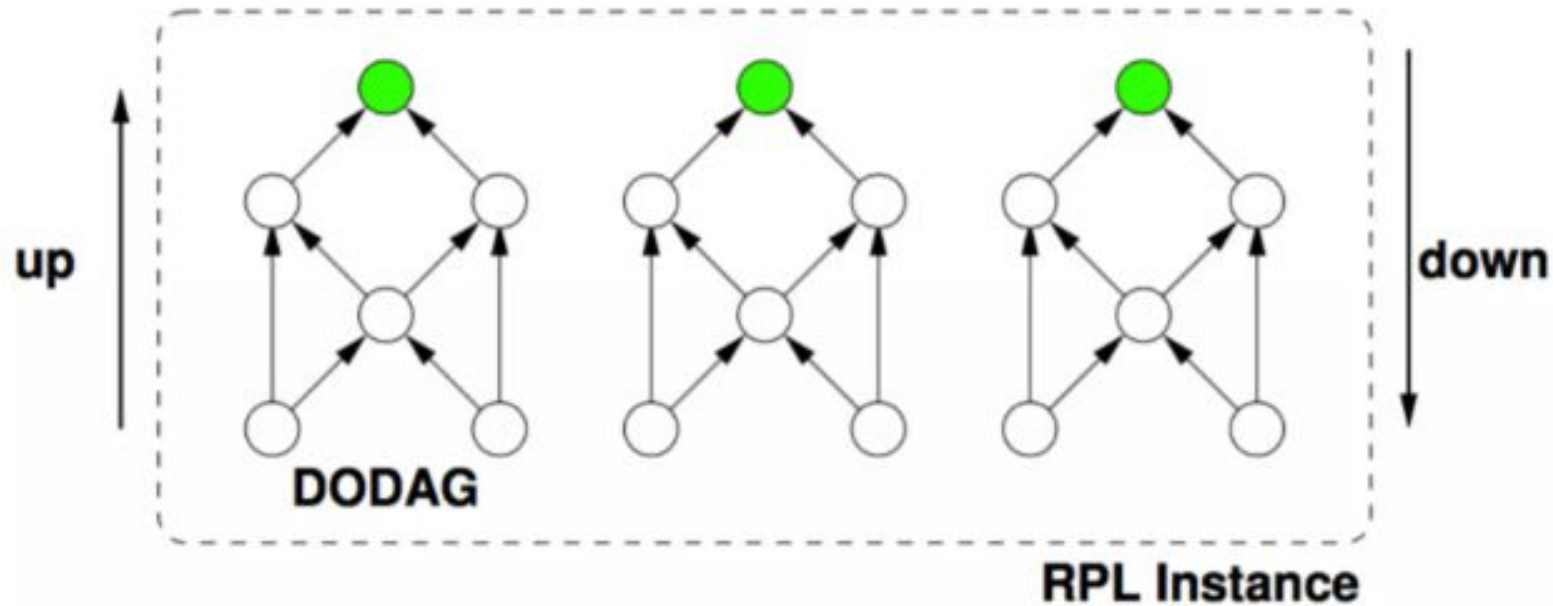
Introduction

- RPL is a distance vector routing protocol
- RPL mainly targets collection-based networks, where nodes periodically send measurements to a collection point.
- The protocol was designed to be highly adaptive to network conditions and to provide alternate routes, whenever default routes are inaccessible.
- RPL provides a mechanism to disseminate information over the dynamically formed network topology
- Contains thousands of nodes...

RPL topology

- DODAG (Destination Oriented Directed Acyclic Graphs)
 - A DODAG is a DAG rooted at a single destination. The DODAG root has no outgoing edges. A DODAG is uniquely identified by a combination of RPL Instance ID and DODAG ID.
- Rank
 - A nodes Rank defines the nodes individual position relative to other nodes with respect to a DODAG root. Rank strictly increases in the Down direction and strictly decreases in the Up direction.
- DODAG Root
 - The DODAG root is the DAG root of the DODAG. The DODAG root may act as a **border router** for the DODAG, and aggregate routes in the DODAG and may redistribute DODAG routes into other routing protocols

RPL topology



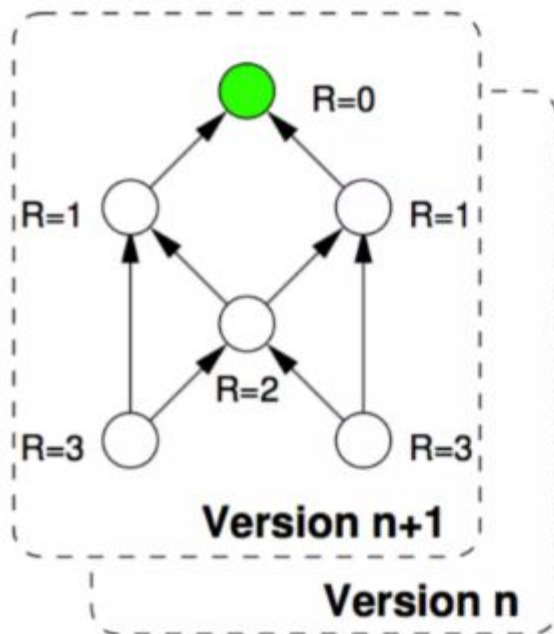
RPL topology

- Upward path is so common (mp2p)
- Downward path is optional mainly for p2p and p2mp
- An RPL Instance consists of multiple Destination Oriented Directed Acyclic Graphs (DODAGs). Traffic moves either up towards the DODAG root or down towards the DODAG leafs

RPL instance

- DODAGS are disjoint (no shared nodes)
- Link properties: (reliability, latency, . . .) \ Node properties: (powered or not, . . .)
- RPL Instance has an optimization objective
- Multiple RPL Instances with different optimization objectives can coexist

RPL Rank



- A node's Rank defines the node's individual position relative to other nodes with respect to a DODAG root. The scope of Rank is a DODAG Version.

Forwarding and routing

- Up routes towards nodes of decreasing rank (parents),
Down routes towards nodes of increasing rank
- Nodes inform parents of their presence and reachability to descendants.
- All routes go upwards and/or downwards along a DODAG
- When going up, always forward to lower rank when possible, may forward to sibling if no lower rank exists
- When going down, forward based on down routes

RPL control Messages

- **DIO** - DODAG Information Object
- **DIS** - DODAG information solicitation
- **DAO** - Destination advertisement object (propagate destination information upwards)
- **DAO-ACK** - DAO Acknowledgement (unicast packet by a DAO recipient)
- **CC** - Consistency Check (Checking for consistency in the messages)

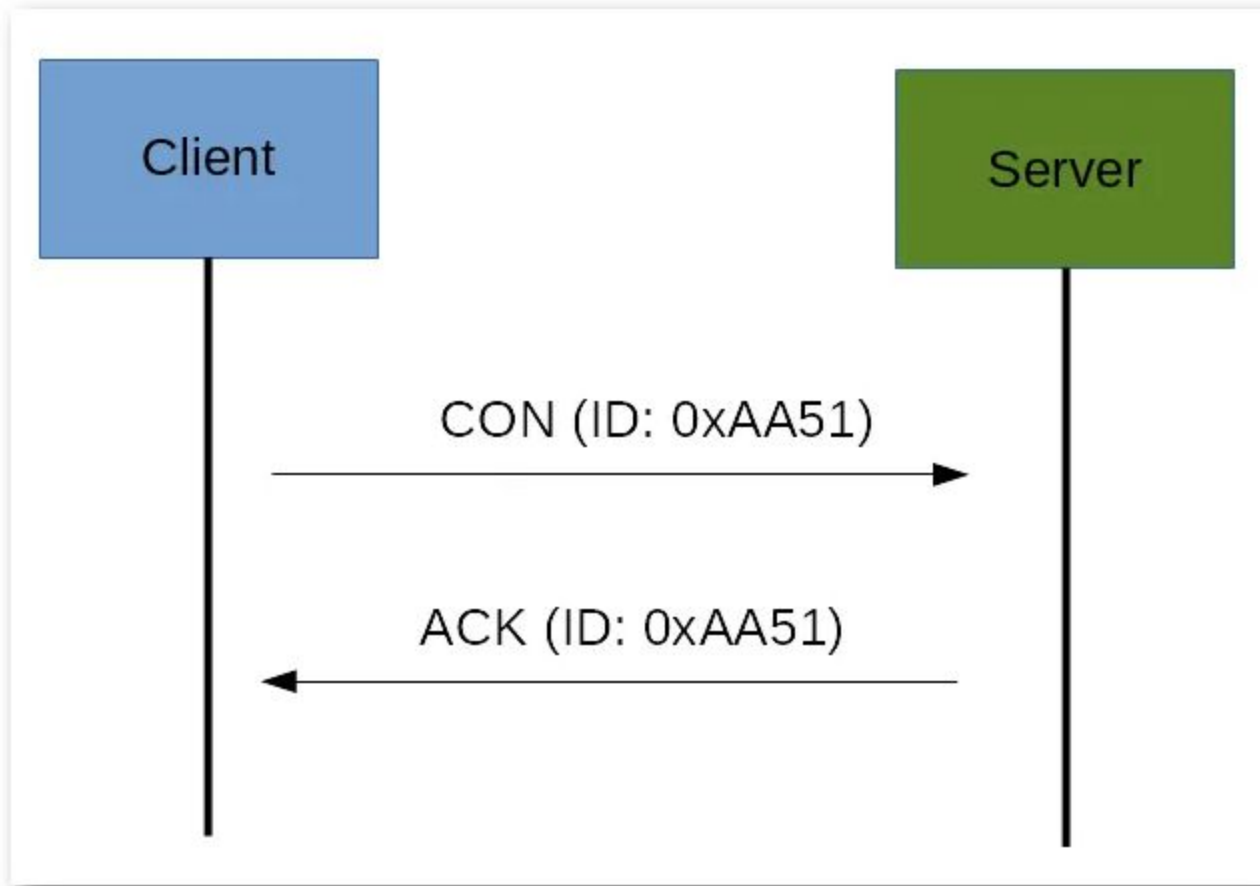


CoAP Message types

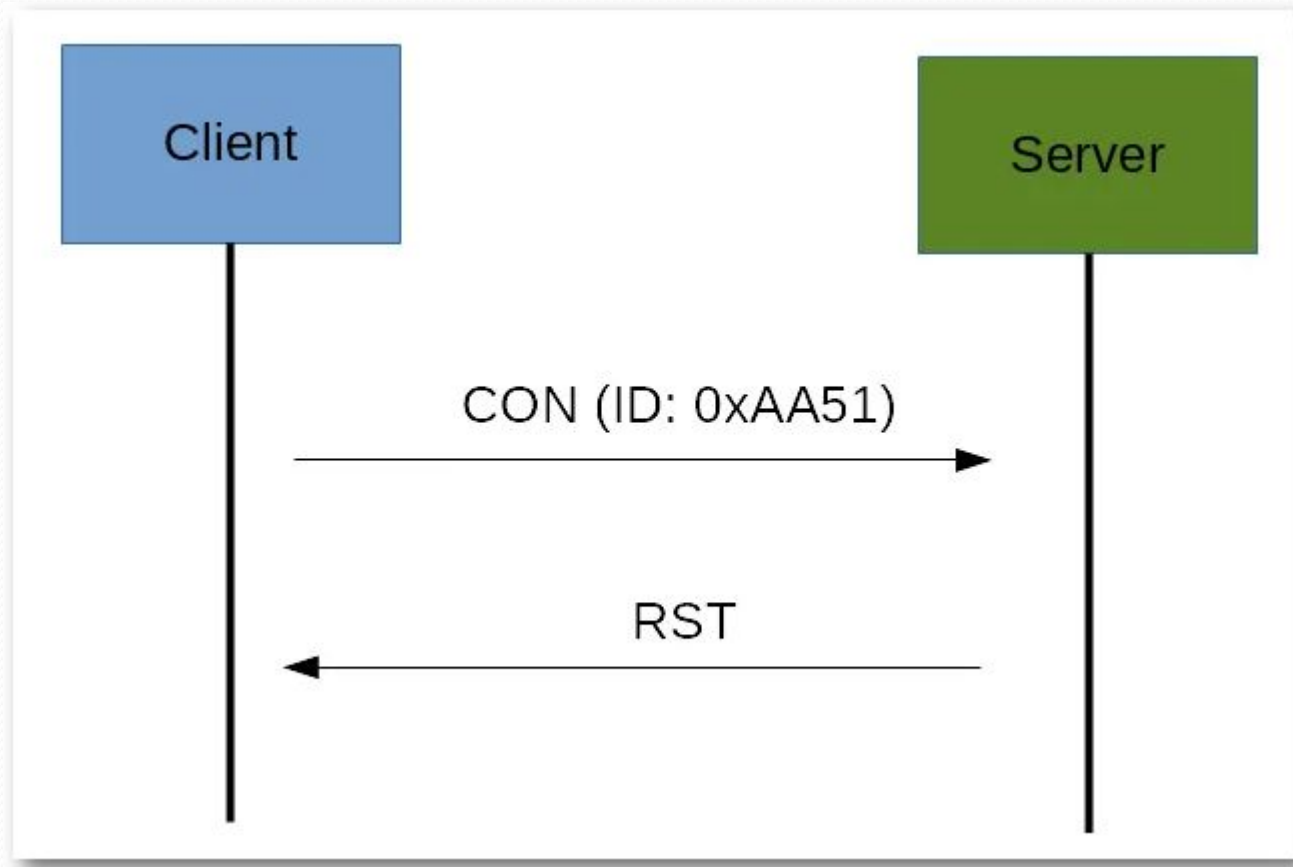
- CoAP supports four different message types:
- Confirmable
- Non-confirmable
- Acknowledgment
- Reset

CoAP protocol

- As said before, the CoAP protocol uses two kinds of messages:
- Confirmable message
- Non-confirmable message
- A CoAP confirmable message is a **reliable message**. When exchanging messages between two endpoints, these messages can be reliable. In the CoAP protocol, a reliable message is obtained using a Confirmable message (CON). As a result, using this kind of message, the client can be sure that the message will arrive at the server. **A CoAP Confirmable message is sent again and again until the other party sends an acknowledge message (ACK).** The ACK message contains the same ID of the confirmable message (CON).
- The picture below shows the CoAP message exchange process:



- If the server has troubles managing the incoming request it can **send back a Rest message (RST) instead of the Acknowledge message (ACK):**



Non-confirmable (NON) messages

- The other message category is the Non-confirmable (NON) messages. **These are messages that don't require an Acknowledge by the server.** Consequently, they are unreliable messages. In other words, these are messages that do not contain critical information to deliver to the server. For example, to this category belongs messages that contain values read from sensors.
- Even if these messages are unreliable, they have a unique ID.


```
graph LR; Client[Client] -- "NON (ID: 0xAA51)" --> Server[Server];
```

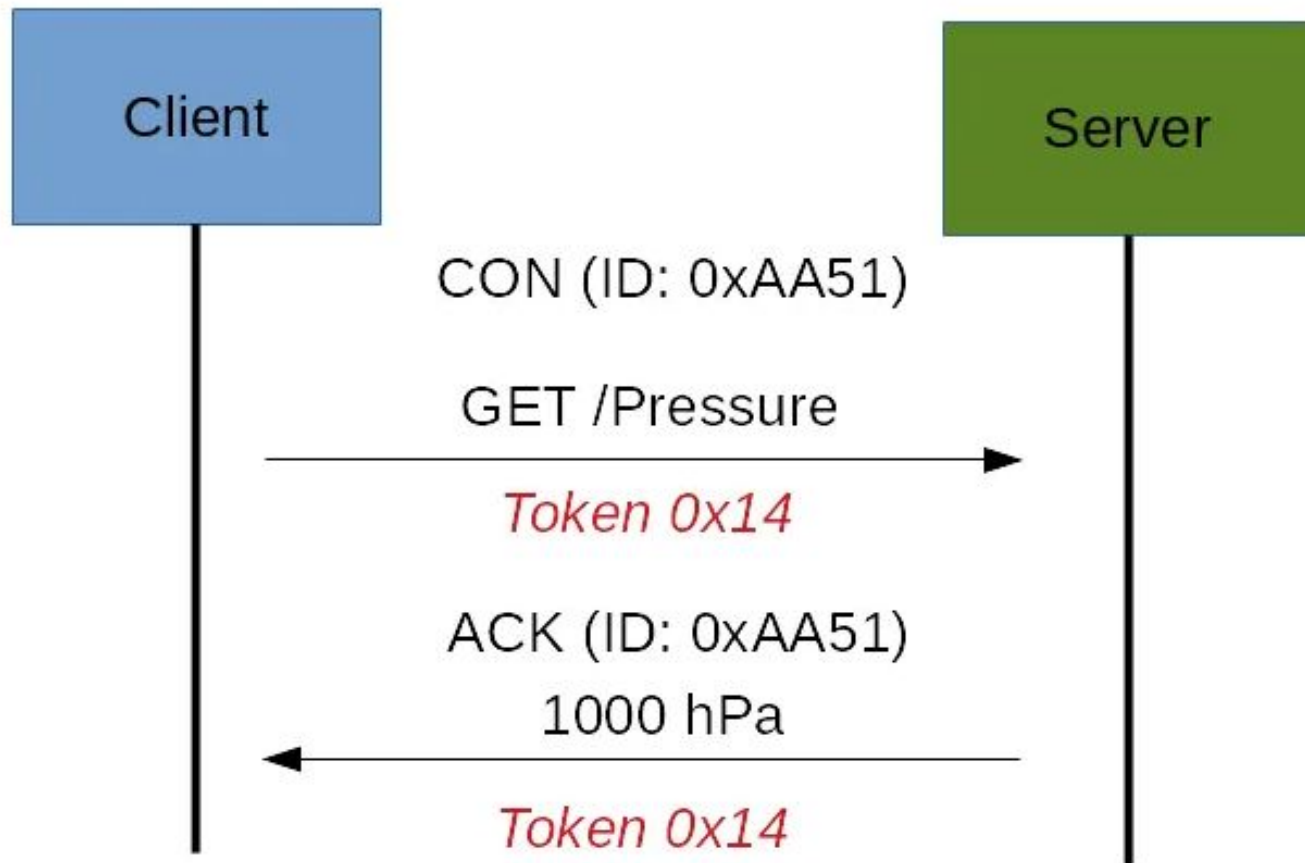
Client

Server


NON (ID: 0xAA51)

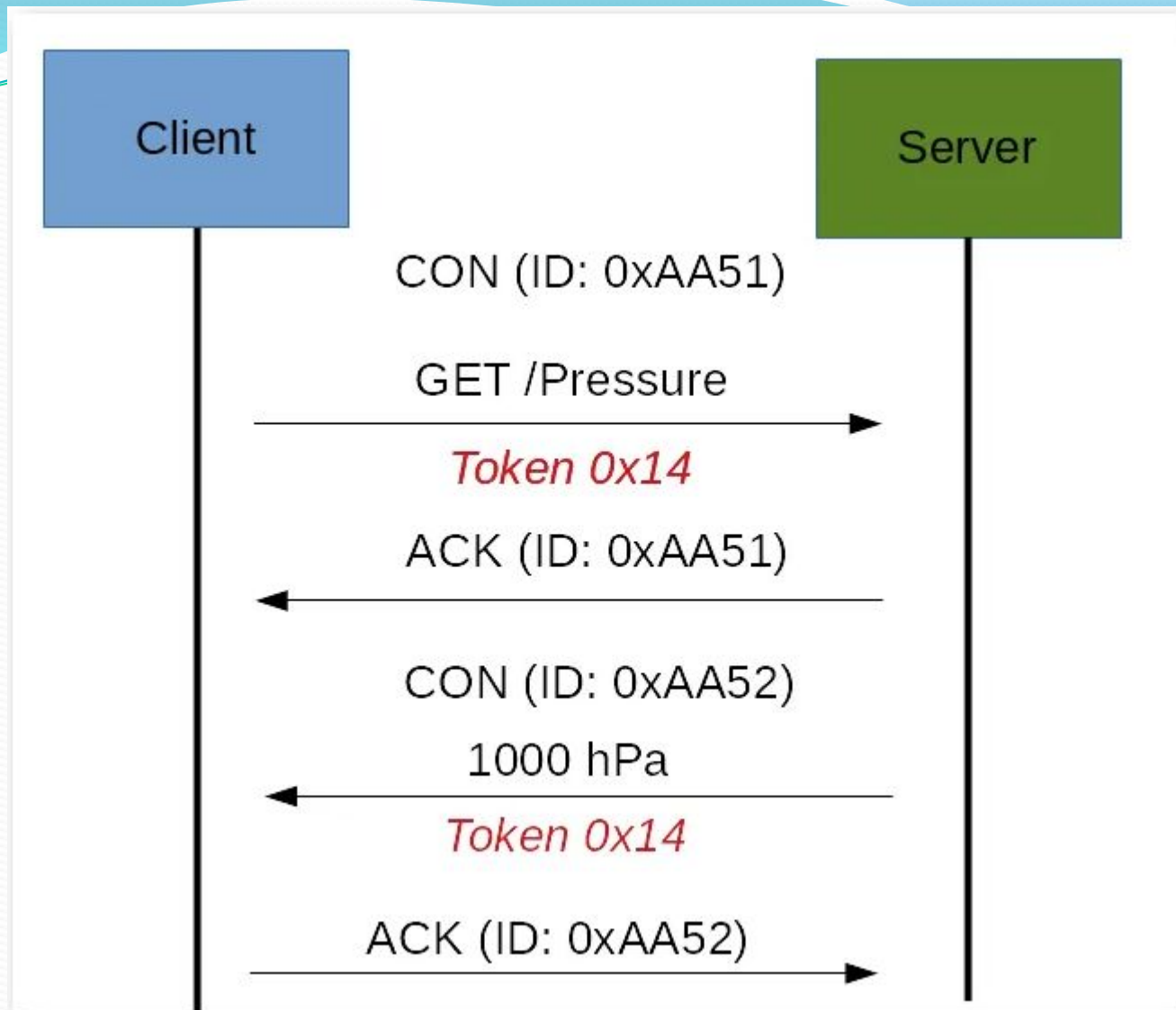
Request/Response Model

- The CoAP Request/Response is the second layer in the abstraction layer. The CoAP protocol sends **the request using a Confirmable (CON) or Non-Confirmable (NON) message**. There are several scenarios depending on if the server can answer immediately to the client request or the answer if not available:
 - If the server can answer immediately to the client request then if the request is carried using a Confirmable message (CON) then the server sends back to the client an Acknowledge message containing the response or the error code:



- As you can notice in the CoAP message there is a Token. The Token is different from the Message ID and **it is used to match the request and the response.**

- 
- If the server can't answer to the request coming from the client immediately, then it sends an Acknowledge message with an empty response. As soon as the response is available then the server sends a new Confirmable message to the client containing the response. At this point the client sends back an Acknowledge message:

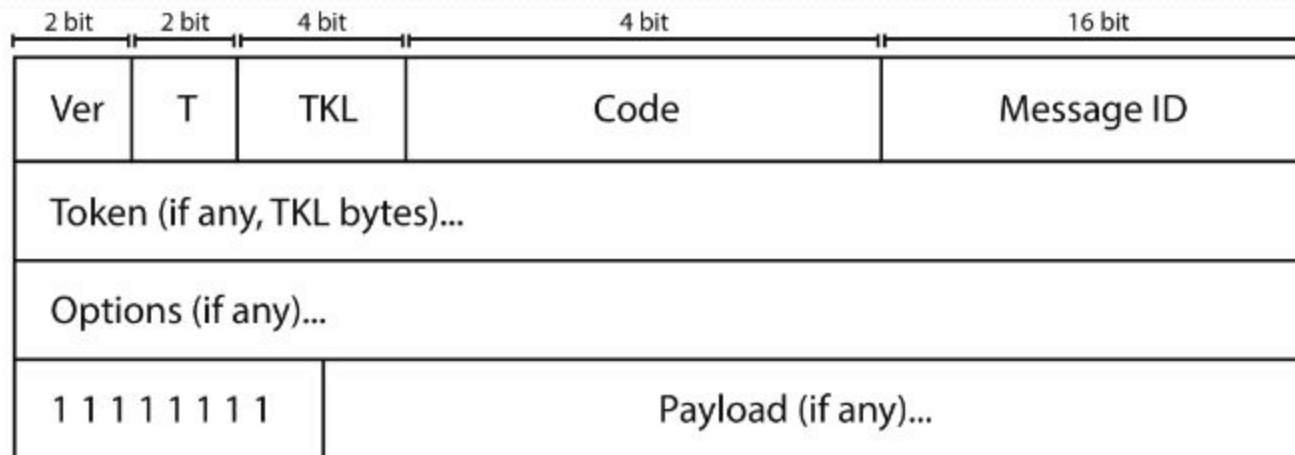


- If the request coming from the client is carried using a NON-confirmable message then the server answer using a NON-confirmable message.

Header CoAP



- **Ver**: It is a 2 bit unsigned integer indicating the version
- **T**: it is a 2 bit unsigned integer indicating the message type: 0 confirmable, 1 non-confirmable
- **TKL**: Token Length is the token 4 bit length
- **Code**: It is the code response (8 bit length)
- **Message ID**: It is the message ID expressed with 16 bit and so on.



Ver - Version (1)

T - Message Type (Confirmable, Non-Confirmable, Acknowledgement, Reset)

TKL- Token Length, if any, the number of Token bytes after this header

Code - Request Method (1-10) or Response Code (40-255)

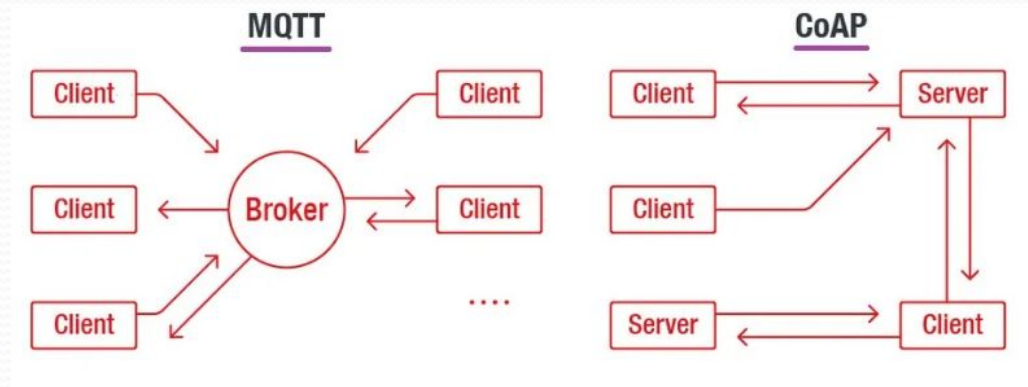
Message ID - 16-bit identifier for matching responses

Token - Optional response matching token

CoAP vs MQTT

- An important aspect to cover is the main differences between CoAP protocol and MQTT.
- As you may know, **MQTT** is another protocol widely used in IoT. There are several differences between these two protocols.
- The first aspect is the different paradigm used. MQTT uses a publisher-subscriber while CoAP uses a request-response paradigm.
- MQTT uses a central broker to dispatch messages coming from the publisher to the clients.
- CoAP is essentially a one-to-one protocol very similar to the HTTP protocol.
- Moreover, MQTT is an event-oriented protocol while CoAP is more suitable for state transfer

	CoAP	MQTT
Communication kind	Request-response model	Publish-subscribe model
Protocol type	P2P/one-to-one communication protocol	Many-to-many protocol
Messaging mode	Asynchronous and Synchronous	Only Asynchronous
Transport layer protocol	UDP	TCP/IP
RESTful-based	✓	✗
Message labeling	✓	✗



Reference:

- LoRaWAN
- <https://www.youtube.com/watch?v=Qd7kMGaQ5vI>
- <https://www.youtube.com/watch?v=Qd7kMGaQ5vI&t=328s>
- <https://www.youtube.com/watch?v=5CCrpqPZBwY> (Example of LoRaWAN)
- <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>
- MQTT
- <https://www.youtube.com/watch?v=NXyf7tVsi10&t=10s>
- <https://www.youtube.com/watch?v=Elxdz-2rhLs>
- CoAP
- https://www.youtube.com/watch?v=-mG_HjAPji0
- <https://dzone.com/articles/coap-protocol-step-by-step-guide>