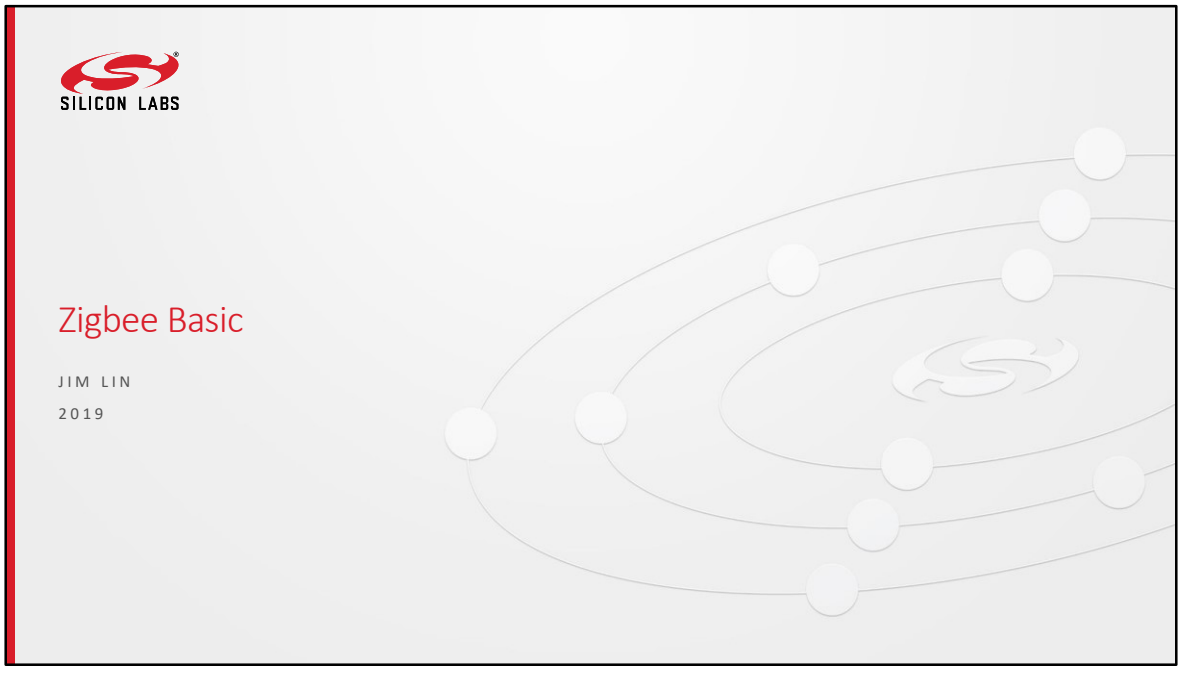


This is a basic introduction of Zigbee. From this lesson, we will learn what Zigbee is, how it works and the the basic concepts of Zigbee. This lesson will last for about 75 minutes.

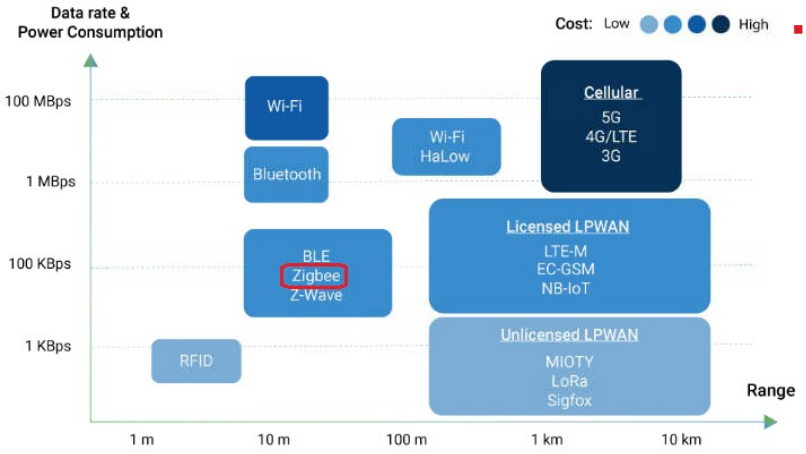
Agenda

- What's Zigbee?
- Protocol Overview
- Physical Layer
- MAC Layer
- Network Basic Concepts and Procedure
- Application Layer
- Security
- Q & A

2

This is the agenda of this training.
We start with the question "What's Zigbee". Together with that, the history of Zigbee and Zigbee Alliance will also be introduced.
Then, we give a basic introduction of the protocol overview and then introduce the protocol layers from the bottom to the top.
At last, we will introduce the basic joining procedure so that we can get a better understanding of how Zigbee works.

What is Zigbee?



- Zigbee
  - Short Range (10 ~ 100)
  - Low Data Rate (max=250kbps)
  - Low Power (can be less than 5uA)
  - Mesh Topology (max 65535 nodes)

3

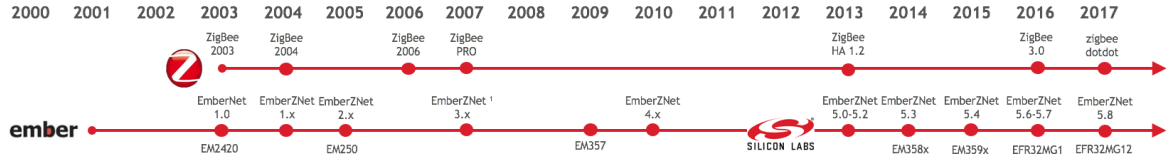
The left picture demonstrates the typical wireless technologies used in IoT industry.
In this picture, we can get a roughly understanding of Zigbee.

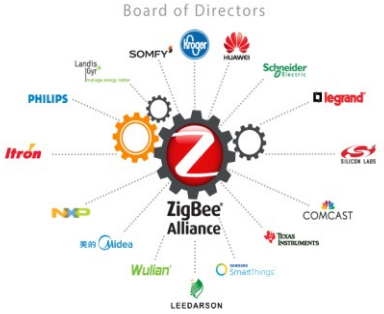
Some of the technologies are used in wide area network (WAN), like 2G/3G/4G, NB-IoT, LoRA and Sigfox. They work on either licensed spectrum or unlicensed spectrum, and the transmission distance can be longer than 1km.
The other technologies are used in personal area network (PAN), like Bluetooth, Zigbee, Z-Wave. The transmission distance is normally less than 100 meters.
Zigbee is one of the most popular wireless technologies used in personal area network (PAN).

On the right, it's a list of the characteristics of Zigbee:

- Short range. – Normally the radio can cover from 10 to 100 meters;
- Low data rate – the maximum data rate is 250 Kbps;
- Low Power – a sleepy end device can use less than 5uA at sleep mode;
- It's a mesh technology – the network can be easily extended to very large. Theoretically maximum nodes number is 65535

Zigbee Alliance





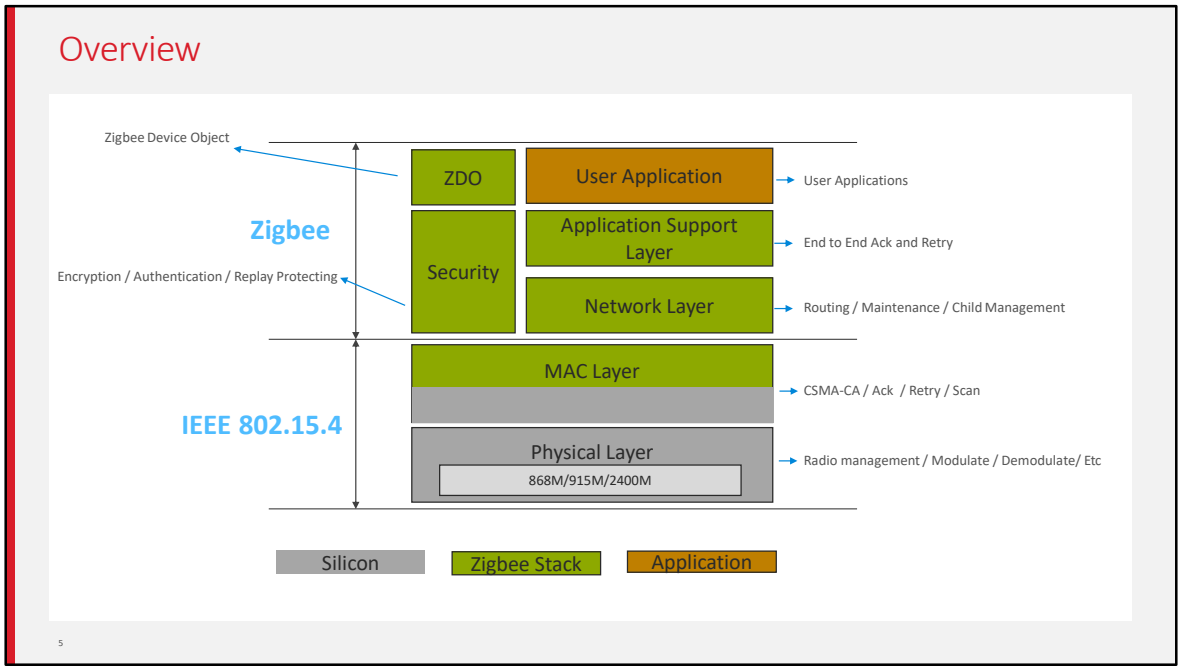
Develop, Certify, Promote

One profile that rules all

4

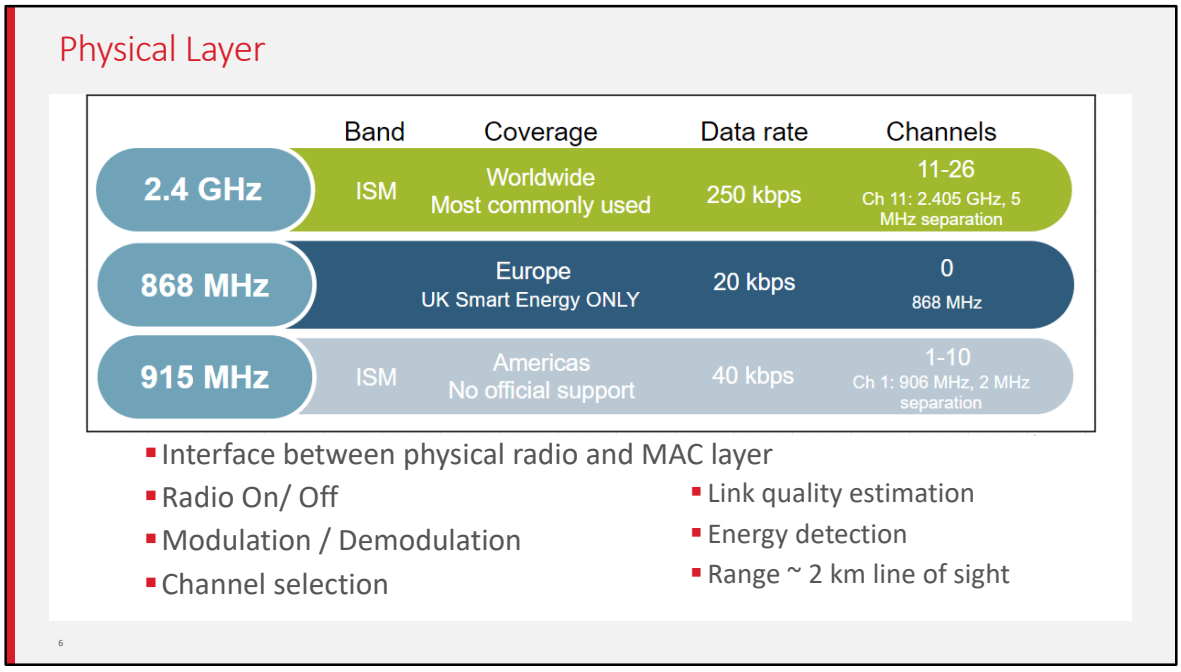
ZigBee Alliance is a group of companies that maintain and publish the ZigBee standard. The main job of the Alliance is to develop Zigbee technologies, to certify Zigbee products and to promote Zigbee industry.
Silicon Labs is member of the board of Zigbee Alliance. We provide Zigbee compliant platforms.

In 2016, Zigbee Alliance unified different Zigbee profiles and published the most popular standard Zibgee 3.0. Zigbee 3.0 also enhanced security features as it's getting more and more important.
Zigbee 3.0 can be compatible with the previous standard. Now Zigbee 3.0 is a mandated since 2017.



This is the protocol overview of Zigbee.

1. The physical layer and MAC layer are defined by IEEE 802.15.4. The rest part are defined by Zigbee specification.
2. The physical layer and part of the MAC layer are implemented by the SoC hardware. Normally the hardware is consist of a radio transceiver and a microprocessor. The blocks marked in green are implemented by Zigbee stack. User can develop their applications based on the stack, so that they can save a lot of time.



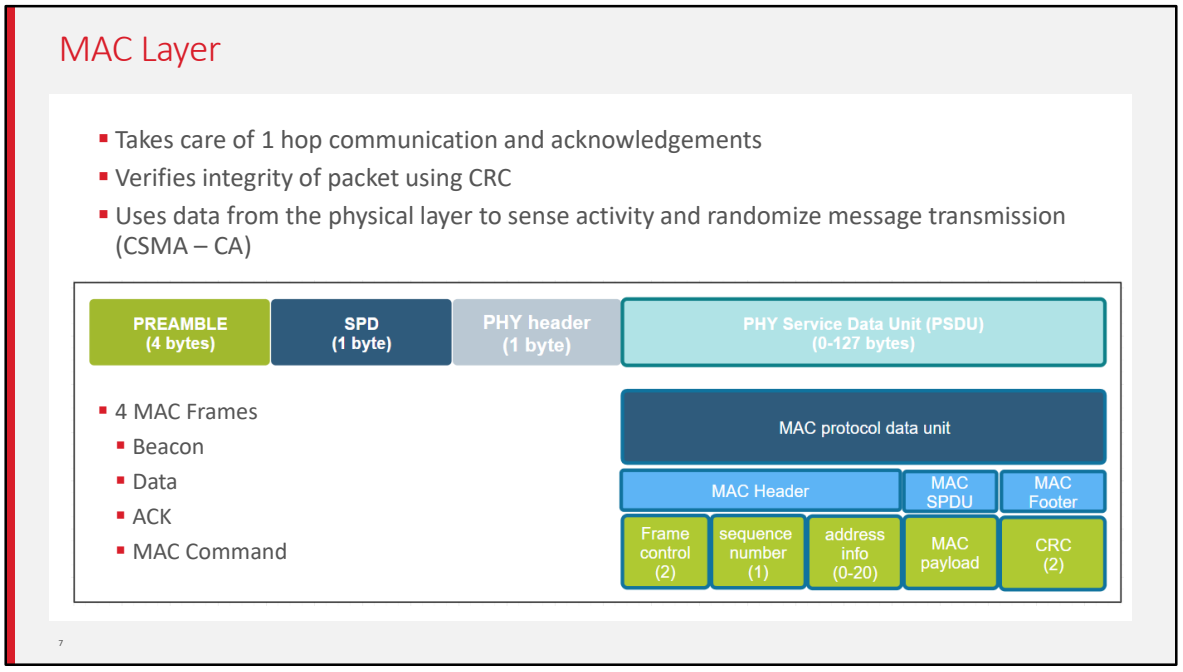
Physical layer handles the transmission and reception of raw bits of data.

The PHY layer uses binary phase shift keying (BPSK) in the 868/915 MHz bands and offset quadrature phase shift keying (O-QPSK) at 2.4 GHz. The information is coded onto the carrier with direct sequence spread spectrum (DSSS), an inherently robust method of improving multipath performance and receiver sensitivity through signal processing gain. Note that 2.4 GHz is the most commonly used frequency band for Zigbee communication worldwide. The only official sub GHz support is for UK smart energy.

Output power:
802.15.4 is designed for low power, low data rate networks with a low-cost objective in mind. These are generally referred to as PANs or Personal Area Networks. The idea here is that these would be low to moderate radio range application designs. But amplification is also possible. It is possible to get up to roughly +20 dBm output power in most countries. In Europe it is regulated a little bit lower to around +10 dBm. But, that's enough to get you anywhere from about one to three kilometers, depending on what your link budget is and what kind of amplification you have and/or what kind of antenna you have.

Data rate:
The raw bit rate is 250 kilobits per second using the 2.4 GHz direct sequence spread spectrum Phy or DSSS. In the real world you are going to see about a quarter or fifth of that. The expected throughout is comparable to a 56k baud modem. Around 52700 kilobits per second on a single hop link. Once you put in multi-hop effects, things will take a little bit longer to propagate.

Open field range
With 2.4 GHz PHY, the ranges we mentioned could be roughly two kilometers with line of sight. This is with a fair amount of amplification and still within legal limits in most areas. And because of all the channels, you have robust communications such that you can avoid interference by making sure to pick channels that are not terribly noisy. Now if you do pick a channel and it becomes noisy, ZigBee has a high level response with what they call "frequency agility." So that some network manager can move the network to a different channel. The other advantage to the 2.4 GHz spectrum is that it's available globally which means you have a wide range install base for your product.



The main function of the MAC layer is to ensure reliable one-hop message delivery by verifying the checksum and sending one-hop acknowledgements. The MAC PDU is shown in the picture. Here are some more details on these functions.

CSMA CA

802.15.4 allows for multiple networks to be on the same channel. Therefore there needs to be some way to avoid having packets from different networks collide over the air and cause errors in communication. MAC sub-layer controls access to the radio using CSMA-CA (Carrier sense multiple access with Collision avoidance). Collision avoidance is done by CCA (Clear Channel Assessment). Before transmitting, every node shall check to see if the airwaves are clear (RSSI below CCA threshold). If they are, the node shall go ahead and transmit after a small random backoff. If the CCA does not pass, then the node shall wait a number of back off periods before trying the process again. The random backoff allows multiple nodes to stagger transmissions so at some point they can find clear air to transmit. Although the bit rate is low, since packets are small (128 bytes), each node completes its transmission successfully even if the channel is fairly busy.

Acknowledgements:

MAC layer also provides a method for nodes to know that 1 hop unicast transmission have been successfully received by way of acknowledgements and that the integrity of the transmitted message has been preserved by verifying a CRC.

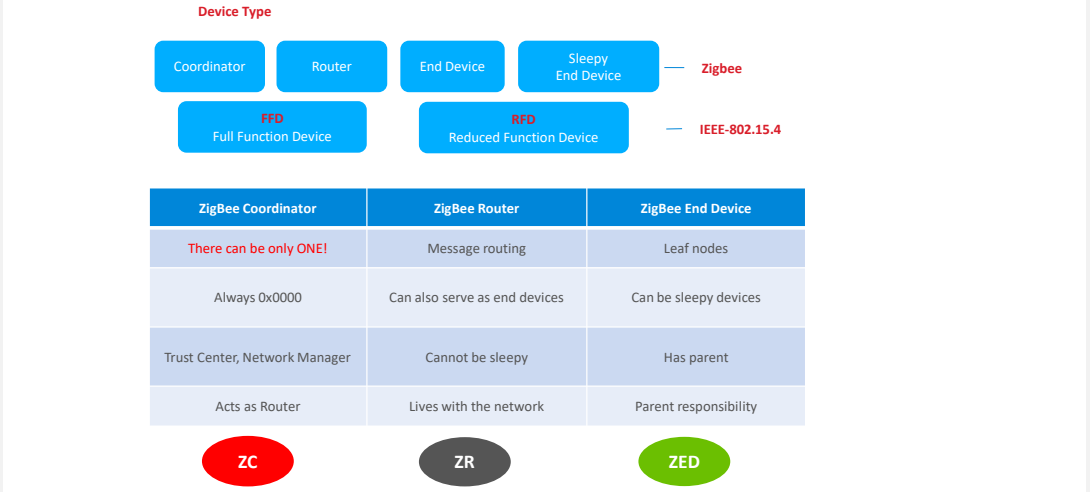
Multi hop transmissions shall be acknowledged on every hop. After the node performs the CCA check and transmits the message, it waits for a MAC acknowledgment. If it does not receive one, the node shall attempt to resend the message multiple times until it eventually succeeds, or the maximum retries have been exhausted. The Silabs Ember ZNet stack provides additional mac retries providing earlier corrective action for a failed message transmission instead of waiting until an end-end retry to kick in, which could take several seconds.

Network Layer Basic

- **Basic Concepts**
  - Device Type
  - Network Address
    - PAN ID
    - Extend PAN ID
  - Device Address
    - Node ID
    - Eui64

Network layer is defined by Zigbee specification. It's in charge of the network transmitting, receiving, routing, child maintenance. In this lesson, we only discuss some basic concepts here, including device types in Zigbee, and also addresses in Zigbee.

Device Type



ZigBee Coordinator	ZigBee Router	ZigBee End Device
There can be only ONE!	Message routing	Leaf nodes
Always 0x0000	Can also serve as end devices	Can be sleepy devices
Trust Center, Network Manager	Cannot be sleepy	Has parent
Acts as Router	Lives with the network	Parent responsibility

ZC ZR ZED

9

In the MAC layer, which is defined in IEEE802.15.4, we have two device types, full function device (FFD) and reduced function device (RFD). An FFD is capable of performing all the duties described in the IEEE 802.15.4 standard and can accept any role in the network. An RFD, on the other hand, has limited capabilities. The processing power and memory size of RFD devices are normally less than those of FFD devices.

ZigBee has three types of basic node types. They are: the ZigBee coordinator or the ZC, the ZigBee router which we abbreviate as ZR, and the ZigBee end device - the ZED. The differences among these types of nodes or devices mainly come down to how they interact with other nodes in the network.

The coordinator is the most important part of the Centralized Zigbee Network. Only one coordinator is allowed per Network. It is the node that form the network. The shortID of the coordinator is always 0x0000.

Senondly we have Routers, they can relay messages from other nodes. It can not fall asleep. Usually it’s powered by a main supply. These devices should be planned to be powered as long as we want the network to function.

End devices are devices that don’t participate in any routing. The only concept of routing that they have is to send things to their parent or get things from their parent. When I say parent I mean there is some router node, potentially the coordinator, that is responsible for that end device, so it bears the responsibility of forwarding messages out and proxying messages in for that end device. An end device relies on its parent for communication to the network. If that communication is lost, the end device then has to go out and find a new parent, and re-attach itself to the network through this new parent.

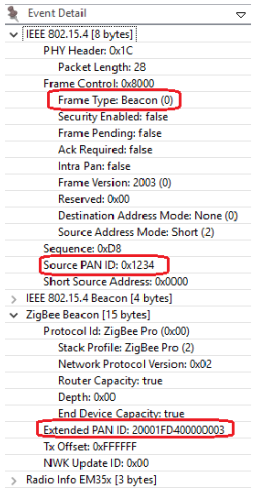
Addressing in Zigbee: Network Address -- PAN ID and Extended PAN ID

PAN ID – Personal Area Network Identifier

- Identifies the network
- MAC layer filters messages meant for the network
- Coordinator picks a random value
- Should be unique but conflicts can happen
- Stack resolves conflicts -> **xPAN ID**

Extended PAN ID – 64bit Unique ID

- Randomly generated by Coordinator at time of network formation
- Only sent over-the-air in response to active scan.
- Enhances network selection
- Short PAN ID changed? Still recognize the network
- Conflict results in unusable network



10

PAN ID:
The PAN, or Personal Area Network, is separated from other networks through its PAN ID. This is a 16-bit identifier that all nodes in the same PAN will share. So it's something akin to a subnet mask in the Ethernet world in that you generally would only be communicating with devices within your local network, which is the PAN in this case. This identifier is placed into the low-level MAC-layer header in every out-going packet, and it allows devices that receive the packet to filter out the messages that don't pertain to their network. They can compare it against their own PAN ID, and decide if this is a message from someone in their own network, or if it's from someone in a different network that just happens to be on this channel so there's no need to try to decode or decrypt it.

The PAN ID is chosen by the coordinator upon network formation. Because the PAN ID is the distinguishing factor between one network and another, it should be random to unsure its uniqueness. It's recommended that you select a random 16-bit value for your PAN ID that keeps your network from coinciding with any other network that happens to exist in the area.

Now, what if you happened to pick a PAN ID that's already used by another network?

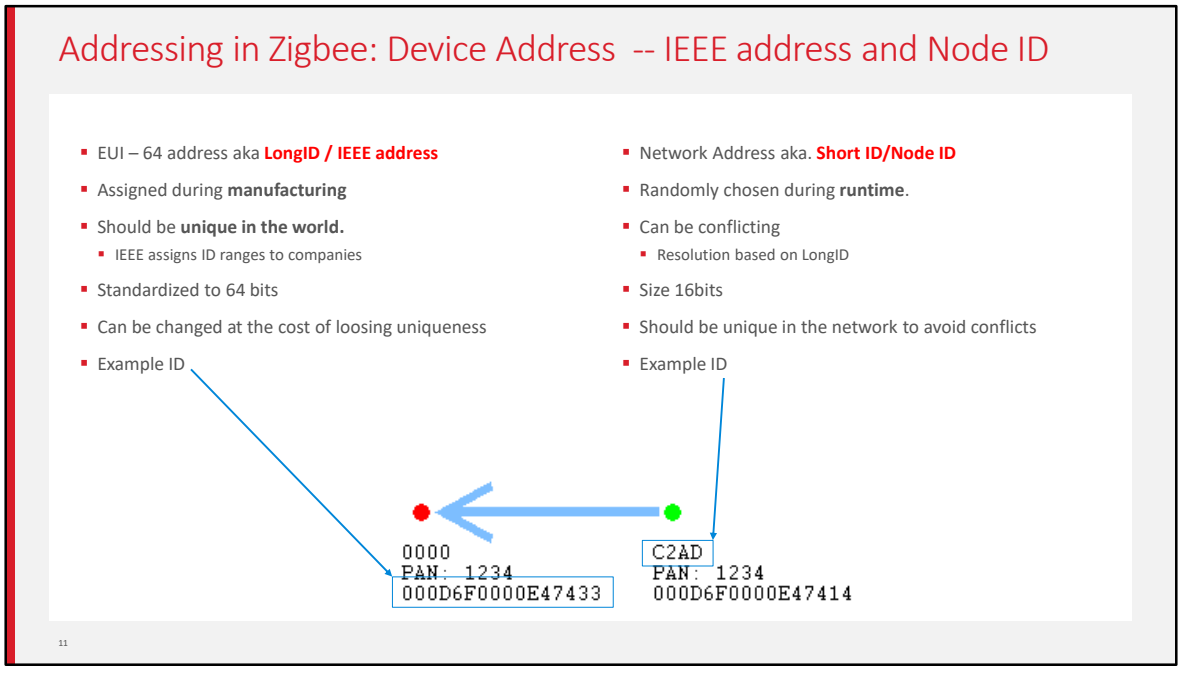
Or what if you did pick a random PAN ID that wasn't in conflict with any other network, but later another network grew to overlap with yours? If the PAN ID conflict ever happens, the stack can in fact detect such a conflict and can update its PAN ID automatically and inform all the nodes in its network to move to the new PAN ID, so that each node can continue communicating with nodes in its original network and exclude anyone on the conflicting network. You may be wondering how the stack does this.

Extended PAN ID:
Well, it is done through the use of the extended PAN ID, which is another network identifier known by all nodes in the PAN.

While the normal short 16-bit PAN ID is transmitted over the air in all the packets because it's short and simple, the 64-bit extended PAN ID is rarely transmitted over the air. The extended PAN ID is also unique for every PAN, and it's basically used as a backup criteria when the 16-bit PAN ID is not enough to always distinguish one network from another. For instance, when a PAN ID conflict occurs and you want to notify all devices in your network to move, the way that you distinguish your network from the conflicting network is, those devices in your network all share the same **extended** PAN ID. The extended PAN ID is highly unlikely to ever conflict because it has 64 bits compared to the 16 bits in the short PAN ID.

The extended PAN ID is also chosen by the coordinator during network formation. It's only sent over the air in response to an Active Scan when nodes are soliciting the network, or when a PAN ID update is occurring.

It's also a useful factor in allowing you to select the network. If you are trying to come into a network rather than form one, you might wonder how to tell which networks are available. The way the networks are distinguishable from one another is not only in the PAN ID but also in the extended PAN ID. You might want to do something special where you decide you are only going to use a certain subset of extended PAN IDs so that you can distinguish your networks from other networks, but just don't limit yourself too much, because the more you limit this the more likely that you have a conflict, and if your extended PAN ID ever conflicts there's really nothing you can do to fix that. It's a little like a WiFi SSID, except that those can be the same between networks and this one can't.



Besides their network-wide criteria, one node is distinguished from another by its individual node addresses.

A node has a short address and a long address. The long address is the IEEE-assigned MAC address, or EUI-64. It is a 64-bit address that is globally unique, meaning no two IEEE-based radios in the world should ever have the same EUI-64. This is generally assigned at manufacturing time. They are assigned when the chips come out of our manufacturing facility before they arrive to you, and they will never change. That's how you tell one radio from another. But because 64 bits are a lot of data, this long address is not sent over the air very often.

Most of the time the much shorter, 16-bit address is used over the air. This is known as the node ID and unique within a network, similar to an IP address in Ethernet world. It is assigned as the node enters the network, and it's supposed to be unique within that network. There may be two networks each of which has a node with the same node ID, but because they in different PANs, it doesn't matter.

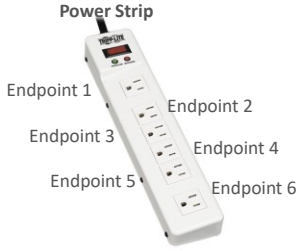
Note that it's possible for two nodes to have chosen the same random node ID when they enter the network. If that happens, much like the PAN ID scheme, there is a

method for conflict resolution. When the nodes notice the conflict, based on the EUI-64 information as a fallback, they can agree upon new addresses. So the nodes can change addresses at run-time if required, based on a conflict.

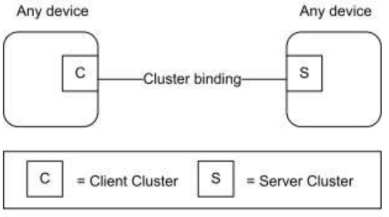
In addition to addresses **of** the node, there are also concepts of addresses **within** the node these are Endpoints and Clusters and they are explained in the “Clusters Endpoints Devices” training module.

Overview of Application Layer

- Endpoint: logical device
  - Endpoints 1-239 are available for user applications
  - Endpoints 0, 240-255 are reserved for special functions
    - Endpoint 0: Zigbee Device Object (ZDO); used for network config/admin
    - Endpoint 255: Used for Broadcasting a message for all endpoints



- Cluster: communication model
  - Client/Server model
  - Defined in Zigbee Cluster Library (ZCL)
  - Cluster ID
  - Commands
  - Attributes



12

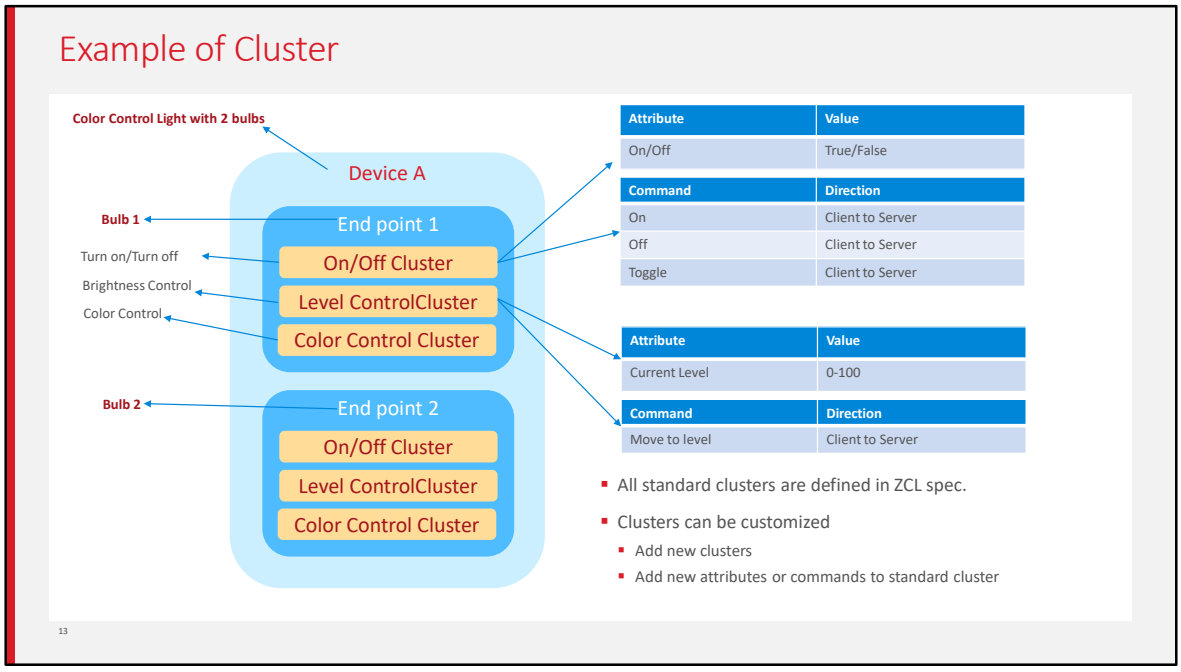
In application layer, a physical device can be split to several logic devices by implementing multiple endpoints.

Each endpoint represents a logic device. For example, if we have a smart outlet adapter with 6 outlets on it. We can implement it with 6 endpoints so that we can switch on/off each outlet respectively.

The endpoint ID is a 8bit value, ranging from 0 to 255.
Endpoint 0 is reserved for Zigbee Device Object, mainly used for management purpose.
Endpoint 240 to 254 are reserved for special applications. Like Zigbee Green Power use dedicate endpoint 242.
Endpoint 255 is used for broadcasting.
Endpoint 1 to 239 can be used by user applications.

In each endpoint, we can configure several clusters. Zigbee cluster is actually a communication model.
It's based on client/server mode and used to describe the application protocol between the two devices.
Each cluster has a cluster ID which is defined in Zigbee Cluster Library (ZCL).

A cluster may define several attributes and commands.



Here is an example which helps to understand the endpoint and clusters.

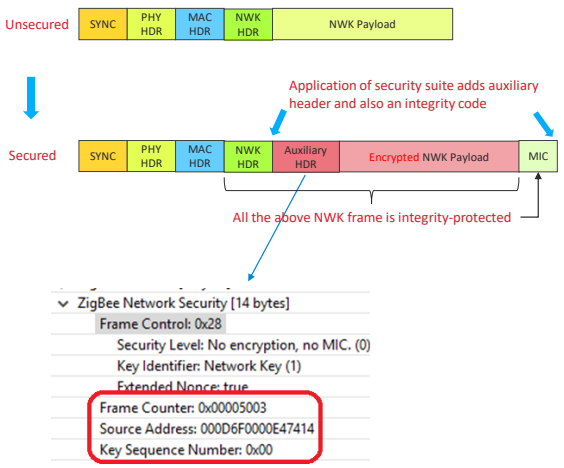
For example we need to implement a Light with two bulbs.
We can define two endpoints in it. Each endpoint represents a bulb.

For the basic function, like turning on/off, we can use the on/off cluster. The light is the server side, and the switch is the client side.
There is an attribute “on/off” defined in the server side, indicating if the light is on or off.
There is also commands like “turn on”, “turn off”, “toggle” defined and should be sent from the client side to server side.

If we want more functionalities, like we need to support level control. We can use the level-control cluster.
In this cluster, there is an attribute “level” defined in the server side, indicating the brightness of the bulb.
There is also commands like “move to level” defined and should be sent from the client side to server side.

And if we need even more functionalities, for example we need to support color control, we can use the color control cluster.

Network Layer Security



- Symmetric Encryption/Decryption
  - All devices use the same key
- Message Integrity Check
- Replay Attack Protect
- NWK Key Management

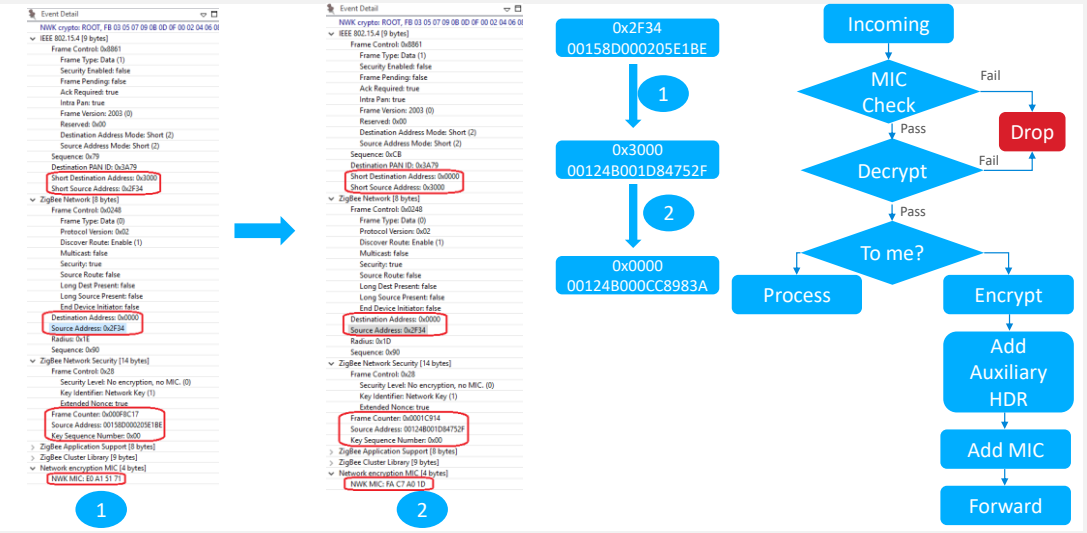
As Zigbee is a wireless technology, security is very important because hackers can sniffer those packets over the air. Imaging that you have a smart door lock.
If the hacker captured the packets of unlocking your door, then he could replay that to open your door. That would be dangerous.

To prevent that from happening, Zigbee defined many security features. Let talk about it.
First we will talk about network layer security. Look at the picture at upper left. It shows how an unsecured network frame is secured in network layer.
First, the network payload will be encrypted. After that, an security header will be added before the encrypted payload. Then calculate a hash value from the network header, security header and the encrypted payload. Finally appended the 32-bit hash value to the end of the frame. If any byte of the network header, security header and encrypted payload is changed, the hash value will be different. We call this value MIC, short for message integrity check.

The network encryption uses a symmetric encrypting algorithm (AES128), which means the same key is used for encryption and decryption. This is key is called network key.
For this reason, all devices in the same Zigbee network will use the same network key.

In the network security header, a field named “frame counter” and the source Eui64 of the node who encrypt the message are added to protect replay attack.
A key sequence number is added to support network key updating.

Network Layer Security – Hop-by-Hop Security

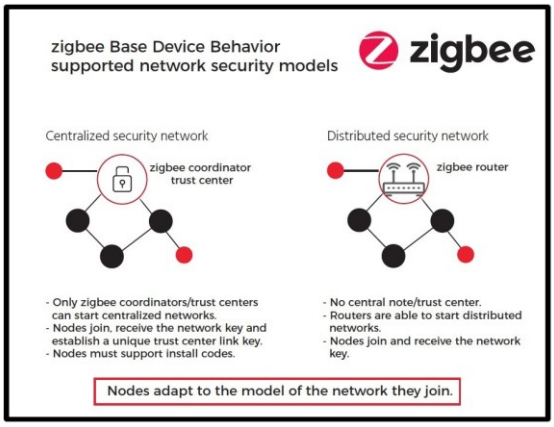


Network layer security is a hop-by-hop security.

The router node needs to decrypt the message, then encrypt it and replace the info in security header, after that send it out.
If the decrypting fails, the message will be dropped immediately.

The benefits of this is to drop the attacking messages as soon as we can.

Network Layer Security – NWK Key



- NWK key is a 16-byte octets
- Randomly generated when network formed
- Trust Center:
  - a network role who distribute NWK key to new devices
- Centralized Security Model:
  - Only one Trust Center (Coordinator)
- Distributed Security Model:
  - Every router can be Trust Center
- NWK key transportation:
  - Secured in application layer

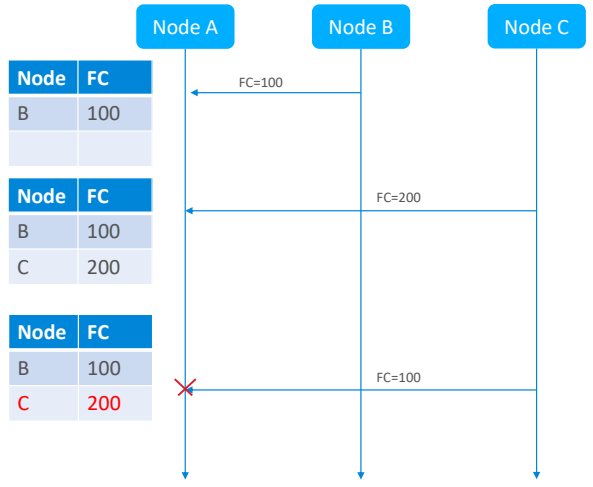
As we talked in earlier, all devices are using the same network key. Network key is a 16 bytes octets.
Normally it's randomly generated by coordinator when the network is formed.
When new devices join network, they must get a copy of the network key.

In Zigbee network, the role who distribute network key to new devices is called trust center. There are two typical security models, centralized security network and distributed security network.

In a centralized security network, there is only one trust center, normally it's the coordinator. All new devices will get network key from coordinator.
In a distributed security network, every router is a trust center. New devices can get network key from every router.

As the network key needs to be transported from one device to another, the key value needs to be encrypted during transporting.
This encryption is done in application layer. We will talk about it later.

Network Layer Security – Frame Counter



Node	FC
B	100

Node	FC
B	100
C	200

Node	FC
B	100
C	200

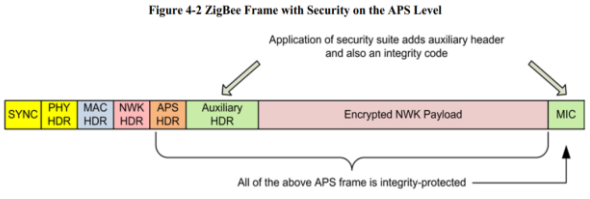
- Transmission side
  - Outgoing FC needs to increase
  - Outgoing FC needs to save in non-volatile memory and restore after reset
- Reception side:
  - Need to save all frame counter of neighbors
- FC must be <= 0x80000000
- Will wrap after a year if keep transmitting at a rate of 135pps.
- NWK key MUST be updated before it wraps.

First, a node will record the frame counter of the received frame and the Eui64 of the node.
The frame counter of the next message from the same node must be bigger than the recorded.
Otherwise, the message will be considered to be a replay and will be dropped.

To achieve this, on the transmitting side, every node will save it's outgoing frame counter.
On the reception side, nodes need to save frame counters of all neighbors.

As frame counter is a 32-bit value, it could wrap if the device keep running for a very long time. Apparently there could be a problem if the frame counter wraps.
To prevent this from happening, the network key must be updated before it wraps.
If the network key is updated, the frame counter could start from zero again.

APS Layer Security



- End to end security
- Link key:
  - Trust center link key
  - Application link key – (Used in Smart Energy)
- Centralized Model
  - Default global link key: ZigbeeAlliance09
  - Install code
- Distributed Model
  - Global link key: will be offered by the Allience
  - Touch link key: will be offered by the Allience

18

Alayer security is quite similar to network security.
Also symmetric encrypting algorithm (AES128) is used. The key is called link key.
For most case, only the transporting network key message needs to be encrypted in application layer, and this only happens between trust center and the new device. So in this case, we also call it as

trust center link key.

APS layer security is an end-to-end security because only the two peers which participate the communication know the link key.

Devices in the network can use the same link key or different link key.
If the devices use the same link key, this key is a global link key.

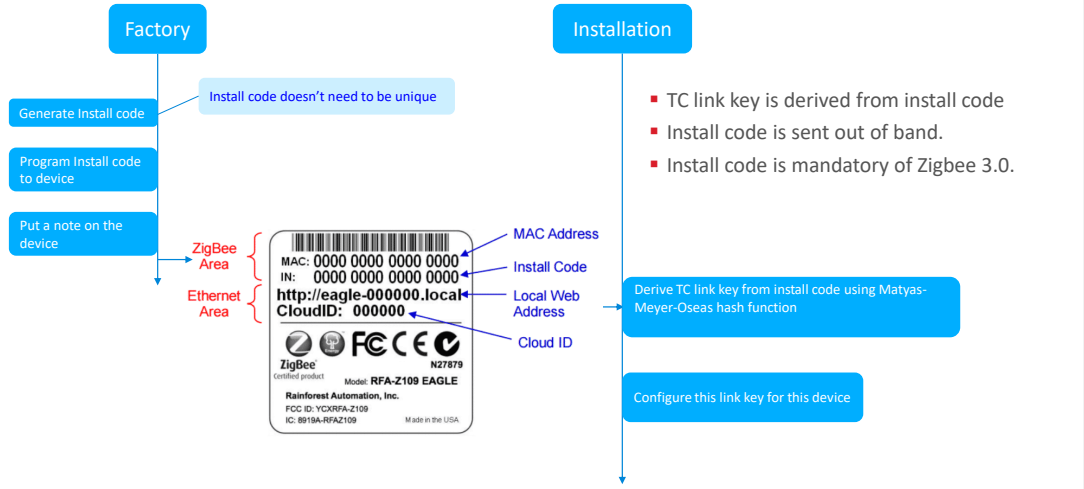
In distributed model, as every router could be a trust center, global link key will be used.
In centralized model, there is also a special global link key being used, which is the well-known link key. It’s the string “ZigbeeAlliance09”.
This is used in the standards before Zigbee 3.0 and is kept to keep backward

compatibility.

The trust center link key must be preconfigured on the devices so that they can join and work.
It would be easy if they are using the well-know link key.
But what if they need to use different link key?

Zigbee defines an approach to configure the link key out of band. It’s the install code.

APS Layer Security – Install Code



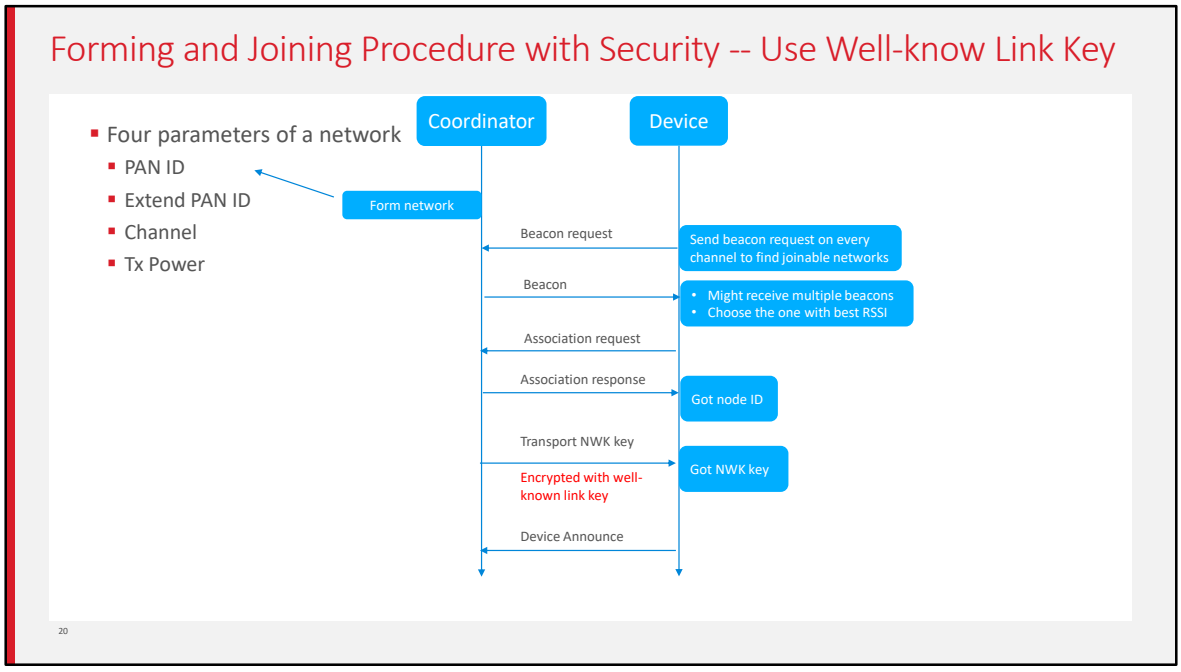
Install code is a 16 bytes polynomial + 2 bytes CRC

When device is manufactured, in the factory, an install code will be programmed into the device.

After that, on the label of the device, the install code and the Eui64 of the device will also be recorded.

When device is going to be installed, users get the install code and Eui64 from the label.
Then configure them to coordinator. The coordinator them derive a link key from the install code and set a table to use the link key for this specific device.

After that, this link key will be used to encrypt the message in application layer. On the device side, it reads the install code from the flash and then derive a link key with the same algorithm. This link key should be the same with the derived link key on the coordinator side. So that they can communicate in application layer even if the message is encrypted.



First, the coordinator forms a network. To form a Zigbee network, you have to prepare 4 parameters:

- PAN ID
- Extend PAN ID
- The working channel
- Transmit power

You need to specify these four parameters.
If you don't, the coordinator will randomly choose a PAN ID and an extended PAN ID.
If you don't specify a channel, the coordinator will scan and pick a relatively quiet channel to work on.

After the network formed, new devices can start to join.
First, the new device will start to find the joinable network. In this phase, the new device will send beacon requests on each channel. Routers and coordinators will respond a beacon with the network info carried in the beacon frame. These info includes the PAN ID, the extend PAN ID and also some other properties of the router or coordinator, like if the device permit join, if the device has the capacity to let the new device join.

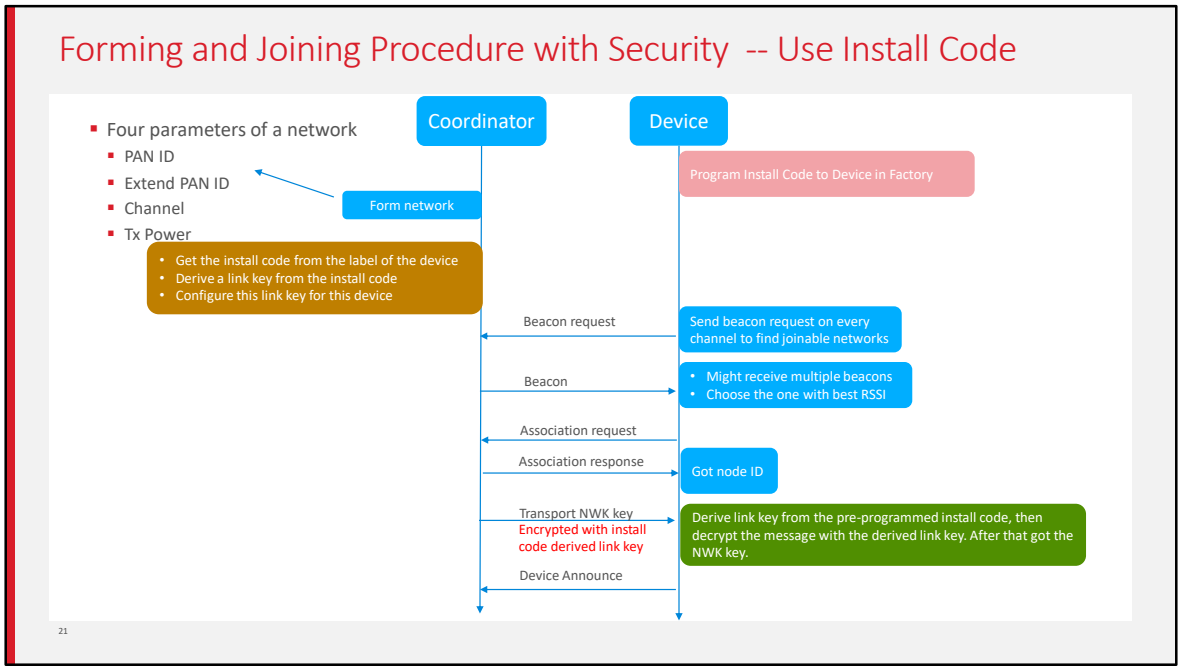
At this stage, the new device may receive multiple beacons from different devices. It will pick up one with the best signal quality and start to send association request. In this associaton request, the PAN ID is set to the chosen PAN and the destination node id is set to the node ID of the chosen device. In this frame, the capability of the new device will be carried on.

When the router or coordinator received this association request, it would pick up a node ID for the new device and respond with an association response.

Then the new device got its node ID and but it can't communicate with other nodes as it hasn't gotten the security key.

After the device got it's node ID, the coordinator will transport the current NWK key to the new device.
This transporting message is encrypted in application layer with the well-known link key.

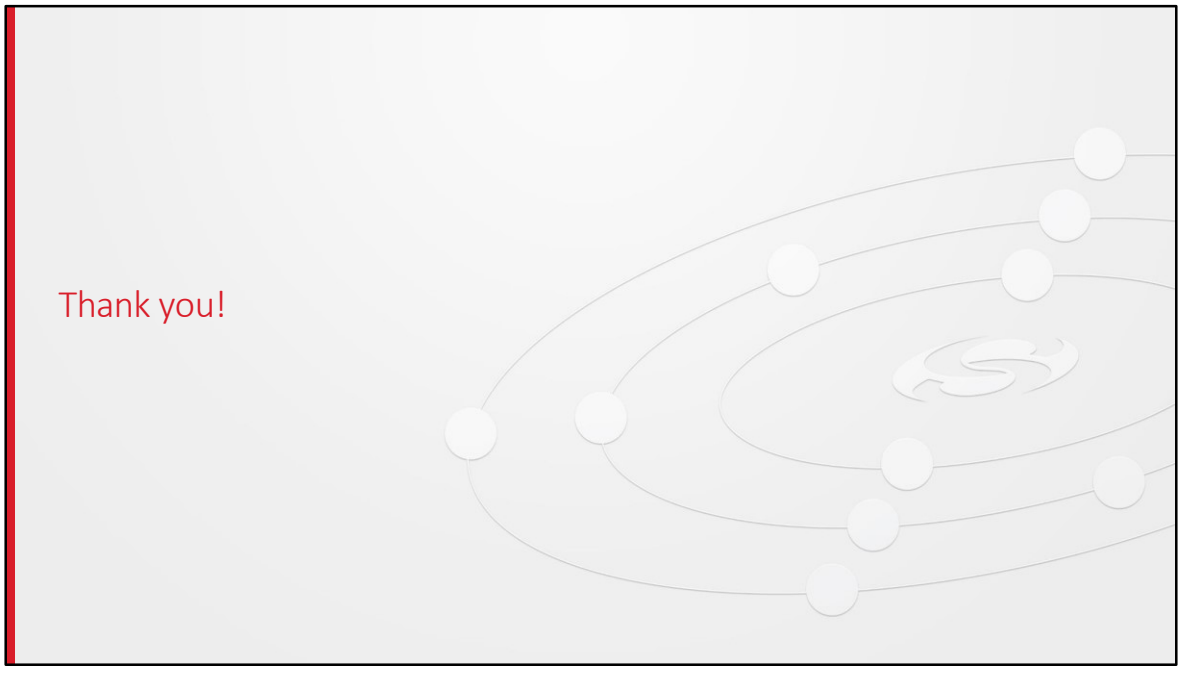
When the new device receives this message, it uses the well-known link key to decrypt the message and gets the network key.
After that, the device is really joined the network and is able to communicate with all other nodes in the network.
The device will send an announce message to notify the other nodes of the network to inform them that I'm joined.



The new device should have been programmed with the install code.
Before it joins, users need to get the install code and Eui64 of the new device, and then configure them on the coordinator.
The coordinator then derive a link key from the install code and set the coordinator to use this link key to encrypt the transport NWK key message for this new device.

When coordinator starts to transport network to the new device, it encrypt the message and transport it to the new device.
When the new device receives this message, it reads the install code from flash and derive a link key from it, then use this key to decrypt the message and get the network key.

Q&A



Thanks