

Zigbee Protocol analysis

Zigbee

INDEX

Who am I ?

To explain who am I and what did I do?
What am I interested in?

Basic of zigbee

Zigbee communication is a kind of communication's architecture so let's talk about what we have to know for zigbee study

How to hack zigbee

This capture describes how to capture zigbee signal and how to make zigbee raw packet

Background

This chapter describes why zigbee.
And what is used for IoT communication.

Protocol analysis

Zigbee communication is little bit difficult So you have to know zigbee communications

QnA

Profile

Who am i?

Name : Lee Sang Sup

Age : 87.07 (< 30)

Group : Wizeguyz && 87 패밀리

History

2007 Hongik University Security Team

2008 Haceker school wiseguyz

2009 Airforce Operating command Computer Security & CERT

2010 Samsung SW membership

Current

Samsung electronic Co. SWC - Security Lab

Security Developer && SW & Device Hacking assessment

Study ?



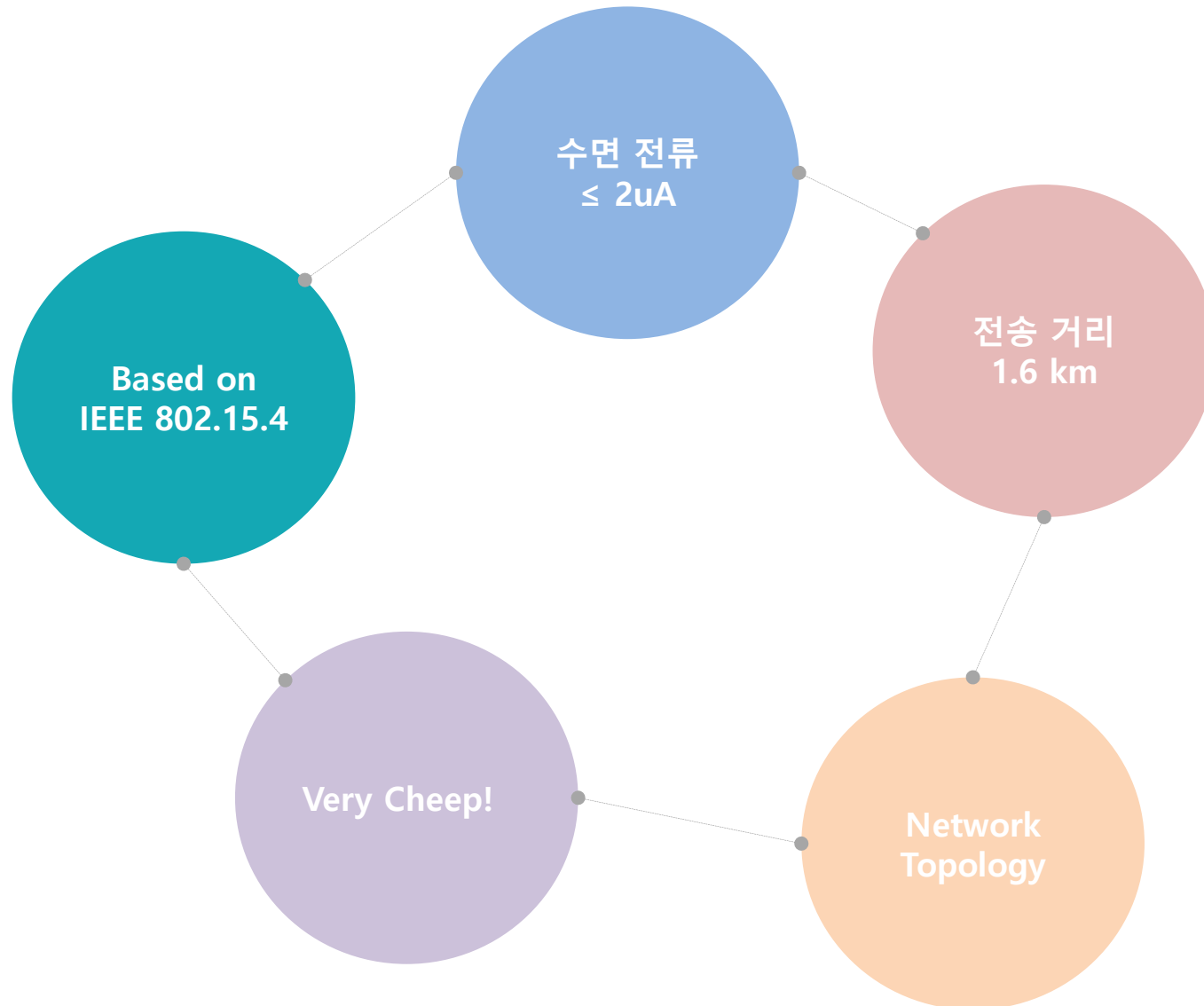
Why zigbee?

IoT generation is coming

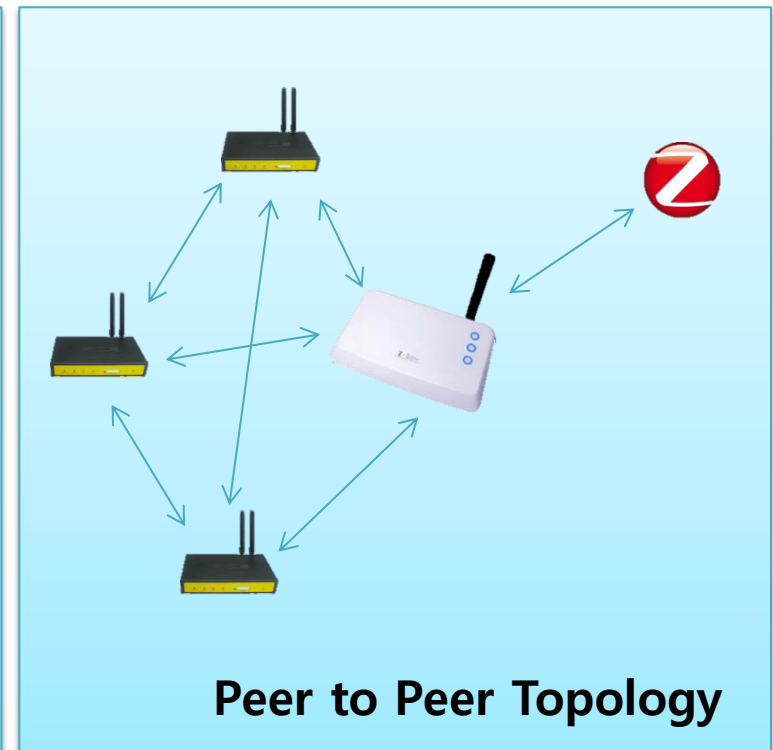
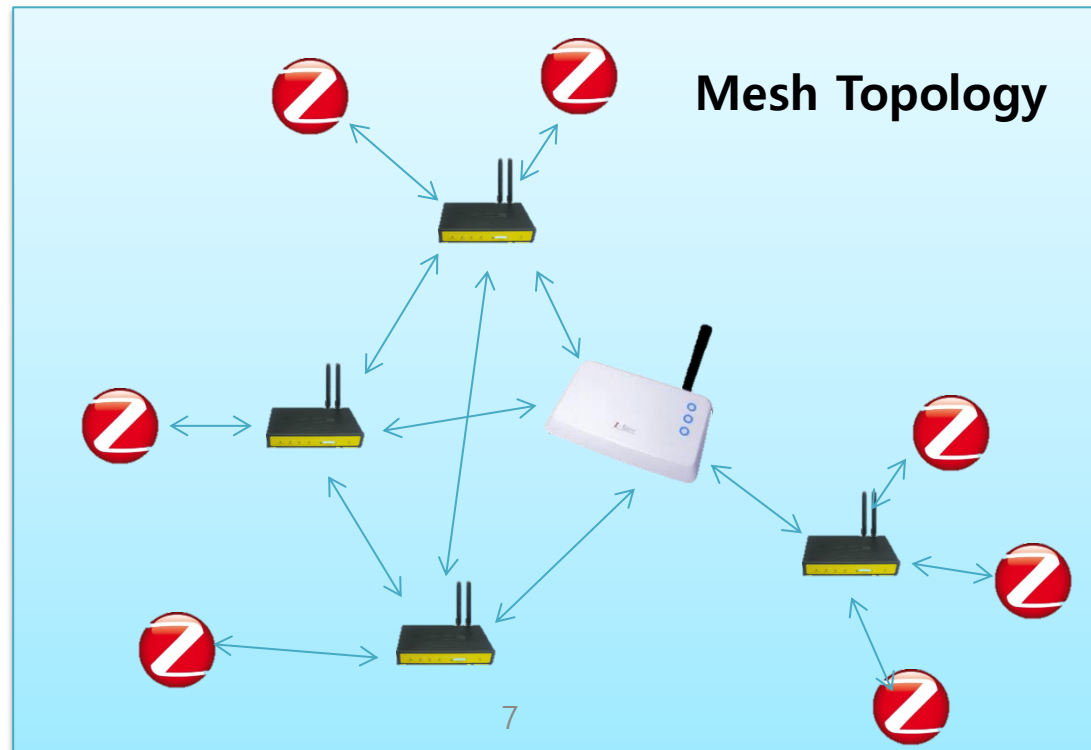
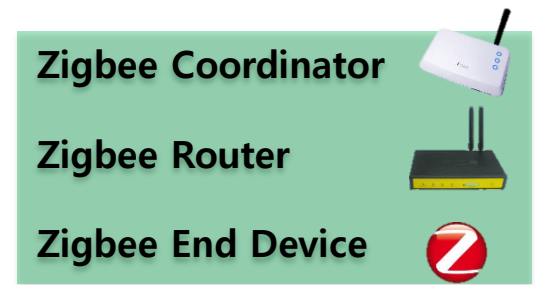
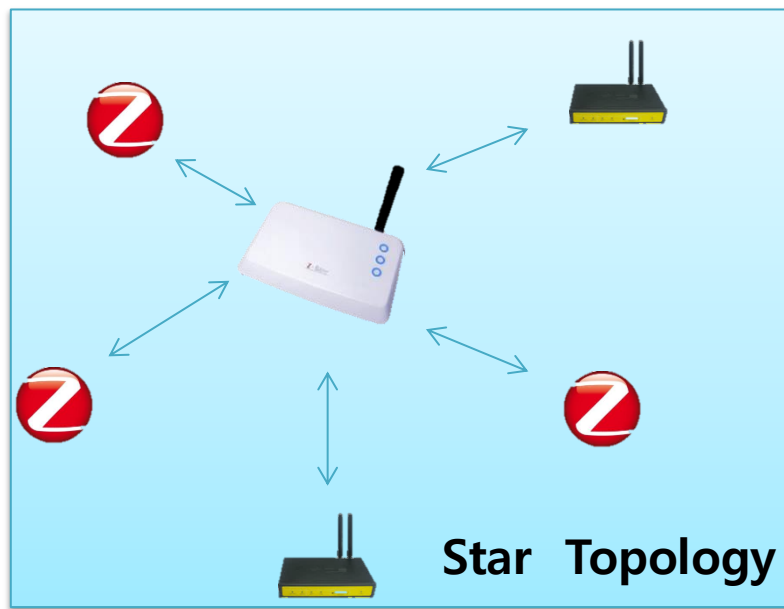


Why zigbee?

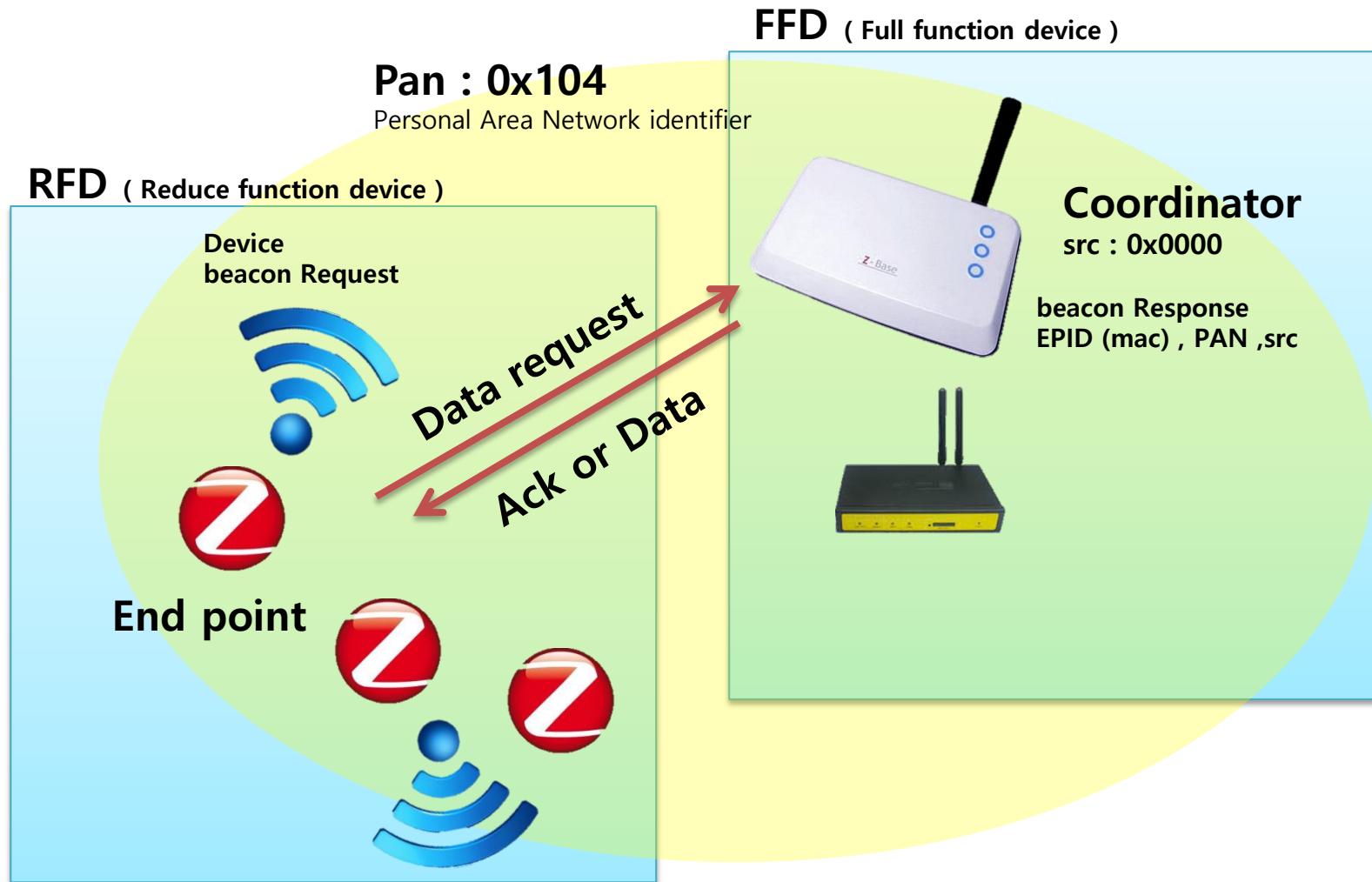
Various advantages



Simple Protocol Analysis



Channel



Broadcast

IEEE 802.15.4
ZigBee

10 Beacon Request

28 Beacon, Src: 0x0000, EPID: 8d:a5:b264:97:e09d:fa

0x0000

Zigbee protocol capture

How to capture zigbee communication?

Devices for Zigbee Communication

RZUSBSTICK



Freakduino-Chibi

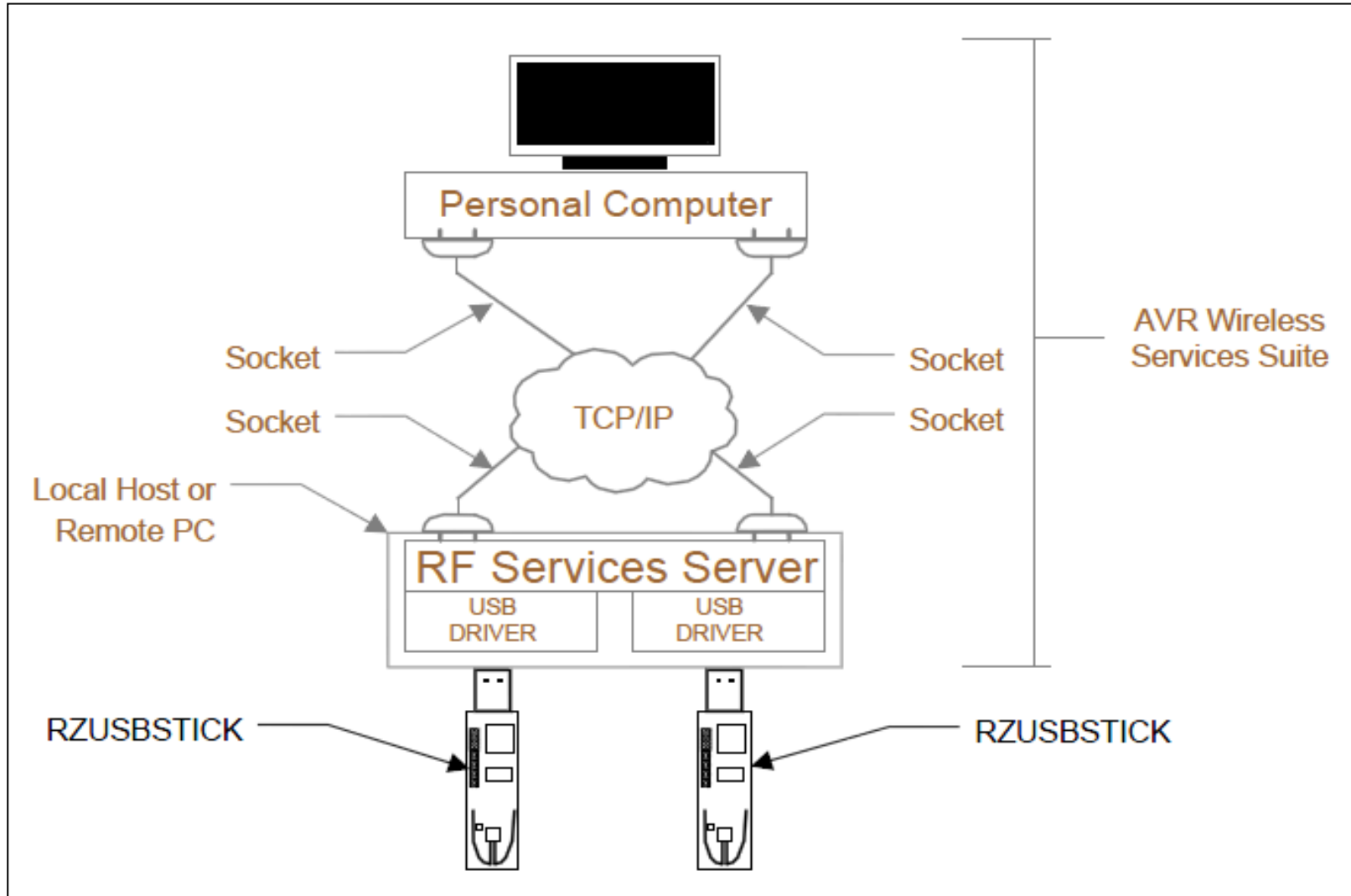


Tmote Sky



How to capture zigbee communication?

RZUSBSTICK for zigbee



How to capture zigbee communication?

Atmel wireshark

<https://gallery.atmel.com/?q=wireshark&orderBy=undefined&SelectedCategoryId=undefined>

With **Atmel Gallery**, getting the tool or software library you need has never been easier. Download and instantly extend the Atmel Studio environment.

Each contribution is licensed to you under a License Agreement by its owner, not Atmel. Atmel does not guarantee the contribution or purport to grant rights to it.


Browse
All extensions in the Gallery

Partner
Become a developer

Search the Gallery


wireshark

RECENTLY ADDED
Page: [All](#)




Wireshark Sniffer Interface Tool v3.0.0.10
Atmel (Atmel)
The Atmel Wireshark Sniffer Interface Tool is an easy to use tool for debugging of IEEE802.15.4 wireless applications...

FREE




Atmel Wireshark Interface
Atmel (Atmel)
Standalone WireShark interface for the ZigBit USB sticks, ATAVRRZUSBSTICK and the RF231 USB Stick in the AVRZ600 kit

FREE




Wireshark Sniffer Interface Tool v3.0.0.10
Atmel (Atmel)
The Atmel Wireshark Sniffer Interface Tool is an easy to use tool for debugging of IEEE802.15.4 wireless applications...

FREE




Atmel Wireshark Interface
Atmel (Atmel)
Standalone WireShark interface for the ZigBit USB sticks, ATAVRRZUSBSTICK and the RF231 USB Stick in the AVRZ600 kit

FREE



Wireshark Sniffer Interface Tool v3.0.0.10
Atmel (Atmel)
The Atmel Wireshark Sniffer Interface Tool is an easy to use tool for debugging of IEEE802.15.4 wireless applications...

FREE



Atmel Wireshark Interface
Atmel (Atmel)
Standalone WireShark interface for the ZigBit USB sticks, ATAVRRZUSBSTICK and the RF231 USB Stick in the AVRZ600 kit

FREE

FREE

AVRZ600 kit
RF231 USB stick in the
ATAVRRZUSBSTICK and the
for the ZigBit USB sticks

FREE

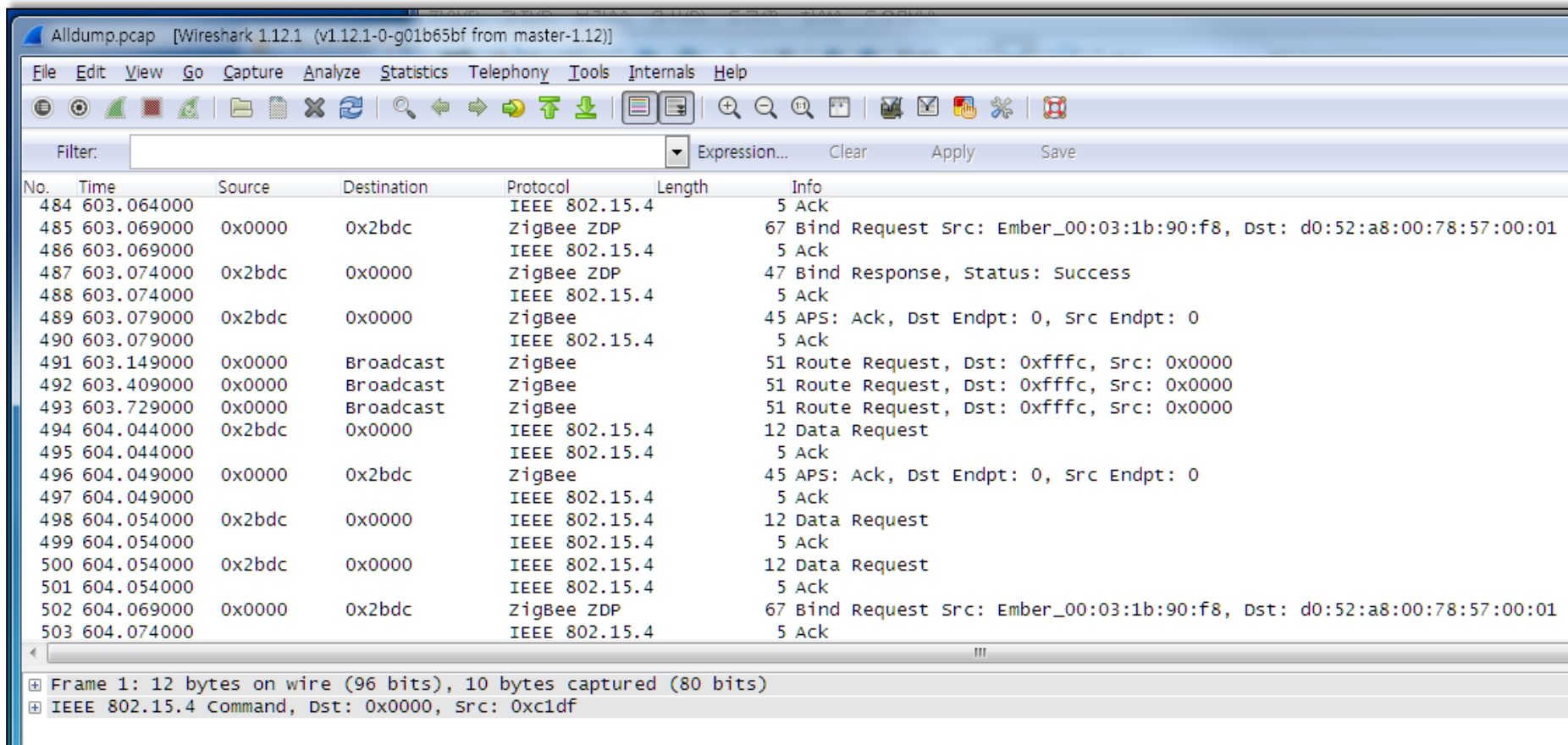
applications...
IEEE802.15.4 wireless
tool for debugging of
Interface Tool is an easy to use
The Atmel Wireshark Sniffer

FREE

AVRZ600 kit
RF231 USB stick in the
ATAVRRZUSBSTICK and the
for the ZigBit USB sticks

How to capture zigbee communication?

Atmel wireshark



The image shows a Wireshark capture of ZigBee communication. The main packet list displays various frames, including ACKs, Bind Requests, Bind Responses, APS Acks, and Route Requests. The packet details pane shows the structure of a ZigBee ZDP frame (Frame 1).

No.	Time	Source	Destination	Protocol	Length	Info
484	603.064000			IEEE 802.15.4	5	Ack
485	603.069000	0x0000	0x2bdc	ZigBee ZDP	67	Bind Request Src: Ember_00:03:1b:90:f8, Dst: d0:52:a8:00:78:57:00:01
486	603.069000			IEEE 802.15.4	5	Ack
487	603.074000	0x2bdc	0x0000	ZigBee ZDP	47	Bind Response, Status: Success
488	603.074000			IEEE 802.15.4	5	Ack
489	603.079000	0x2bdc	0x0000	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0
490	603.079000			IEEE 802.15.4	5	Ack
491	603.149000	0x0000	Broadcast	ZigBee	51	Route Request, Dst: 0xffffc, Src: 0x0000
492	603.409000	0x0000	Broadcast	ZigBee	51	Route Request, Dst: 0xffffc, Src: 0x0000
493	603.729000	0x0000	Broadcast	ZigBee	51	Route Request, Dst: 0xffffc, Src: 0x0000
494	604.044000	0x2bdc	0x0000	IEEE 802.15.4	12	Data Request
495	604.044000			IEEE 802.15.4	5	Ack
496	604.049000	0x0000	0x2bdc	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0
497	604.049000			IEEE 802.15.4	5	Ack
498	604.054000	0x2bdc	0x0000	IEEE 802.15.4	12	Data Request
499	604.054000			IEEE 802.15.4	5	Ack
500	604.054000	0x2bdc	0x0000	IEEE 802.15.4	12	Data Request
501	604.054000			IEEE 802.15.4	5	Ack
502	604.069000	0x0000	0x2bdc	ZigBee ZDP	67	Bind Request Src: Ember_00:03:1b:90:f8, Dst: d0:52:a8:00:78:57:00:01
503	604.074000			IEEE 802.15.4	5	Ack

Frame 1: 12 bytes on wire (96 bits), 10 bytes captured (80 bits)
IEEE 802.15.4 Command, Dst: 0x0000, Src: 0xc1df

IEEE 805.12.4 Command, Dst: 0x0000, Src: 0xc1df
Frame 1: 12 bytes on wire (96 bits), 10 bytes captured (80 bits)

203 004'014000 IEEE 805.12.4 2 ACK
205 004'000000 0x0000 0x3pqc Z1db66 ZDb 01 B1uq kedne2r Src: Ember_00:03:1b:90:f8, Dst: d0:52:a8:00:78:57:00:01
207 004'024000 IEEE 805.12.4 2 ACK
200 004'024000 0x3pqc 0x0000 IEEE 805.12.4 15 Data Request
400 004'024000 IEEE 805.12.4 2 ACK
408 004'024000 0x3pqc 0x0000 IEEE 805.12.4 15 Data Request

What is run via zigbee?

What is run via zigbee?



IoT zigbee



+Sangsup



웹문서

이미지

뉴스

동영상

지도

더보기

검색 도구



세이프서치



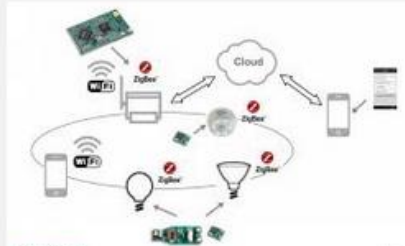
	ZigBee RF4CE	ZigBee PRO	ZigBee IP
Application Standard	ZigBee Remote Control ZigBee Input Device ZigBee Building Automation ZigBee Health Care ZigBee Home Automation ZigBee Meter Services ZigBee Smart Energy 1.0 ZigBee Transport Services		ZigBee Smart Energy 2.0
Network	ZigBee RF4CE	ZigBee PRO	ZigBee IP
MAC	IEEE 802.15.4 - MAC		IEEE 802.15.4 - MAC
PHY	IEEE 802.15.4 Sub-GHz (specified per region)	IEEE 802.15.4 - 2.4 GHz (worldwide)	IEEE 802.15.4 2009 - 2.4GHz or other



IoT의 주요 기술 Zigbee

Zigbee는 지그재그로 돌아다니는 꿀벌이라는 뜻이며, 꿀벌이 꽃과 꽃 사이에서 하는 단순한 행동을 보고 따온 말이다.

사물과 사물이 대화를 할 때 같은 언어로 말해야 서로 알아들을 수 있는데, Zigbee는 이런 사물 간 통신에서 사용되는 언어 중 하나라고 볼 수 있다.



ZigBee Alliance Platform	Application Profiles	ZigBee or OEM
	Application Framework	
	Network Layer	
	Medium Access Control Layer	IEEE 802.15.4
	Physical Layer	



BLACKHAT & DEFCON



Black Hat is the premiere information security conference in the world. The event takes place in Las Vegas, USA and is a highly technical information security conference that brings together thought leaders from all facets of the infosec world – from the corporate and government sectors to academic and even underground researchers. The environment is strictly vendor-neutral and focused on the sharing of practical insights and timely, actionable knowledge. Cognosec is incredibly excited to have been chosen to speak at this year's event – presenting their security research findings in the field of Internet of Things in **ZigBee Exploited: The Good, the Bad and the Ugly**.

ZigBee Exploited:

August 6th @ 12:10 – in South Seas ARF

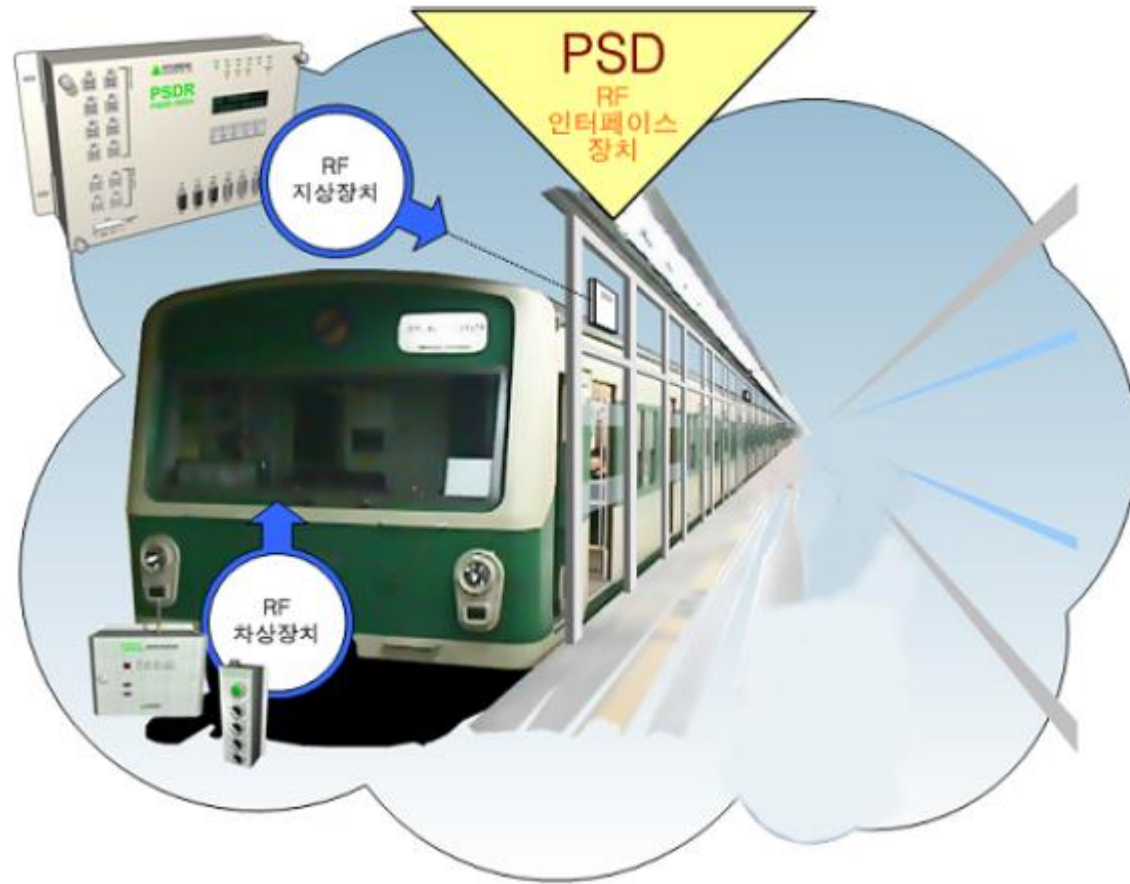
DEFCON is one of the world's largest hacker conferences, held every year in Las Vegas just as Black Hat begins to wrap up, and is a sacred haven of all hackers world over. At this year's event, we will be showcasing our new Internet of Things vulnerability exploitation tool, **SecBee**, as well as presenting on the "Security of Wireless Smart Homes" in DEFCON's **IoT Village**. The world of IoT is exploding, and Cognosec is making sure that proper security protocols are established before it is too late.

Security of Wireless Home Automation Systems – A World Beside TCP/IP:

August 7th @ 4:00pm – in IoT's Bronze Room

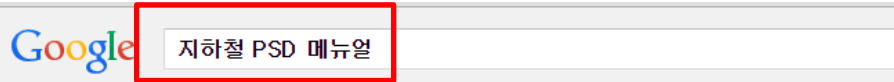
What is run via zigbee?

Platform screen door



What is run via zigbee?

Platform screen door



웹문서 이미지 동영상 뉴스 지도 더보기 ▼ 검색 도구

검색결과 약 349,000개 (0.69초)

이것을 찾으셨나요? 지하철 PSD 메뉴얼

PDF 응급조치 및 유지보수 메뉴얼(PSD)_최종.hwp

https://edu.smrt.co.kr/Common/.../Download.aspx?file_dir=...

PSD분야 응급조치/유지보수 메뉴얼. - 171 - I. PSD 시스템 현황 및 구성. 1. 일반현황.

1.1. PSD 설치 목적. 가. 승객안전시설. 1) 지하철의 선로와 승강장을 차단하여 ...

이 페이지를 3번 방문했습니다. 최근 방문 날짜: 15. 4. 16

지하철 2호선 Seoul Subway ソウルの地下鉄 신림역 스크린 ...



www.youtube.com/watch?v=ukAXI3uvWkY

2015. 3. 15. - 업로더: 김탐장

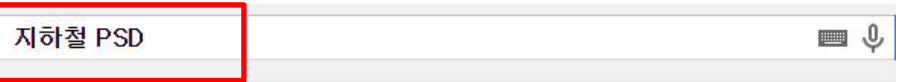
지하철 2호선 Seoul Subway ソウルの地下鉄 신림역 스크린도어 고장 _ ... Eventually opened the screen door in manual mode ...

스크린도어 - 위키백과, 우리 모두의 백과사전

ko.wikipedia.org/wiki/스크린도어

상트페테르부르크 지하철 2호선 모스크브스카야 역. 출입문을 ... 완전밀폐형(PSD)

PLATFORM SCREEN DOOR : 승강장에서 전장까지를 완전히 막는 형태이다.



웹문서 이미지 동영상 지도 뉴스 더보기 ▼ 검색 도구

검색결과 약 349,000개 (0.64초)

관련검색: 지하철 스크린 도어 지하철 스크린 도어 광고비
지하철 스크린 도어 광고 지하철 스크린 도어 시 인전 지하철 스크린 도어

지하철 PSD 관련 이미지

이미지 신고



지하철 PSD에 대한 이미지 더보기

PSD시스템 - 서울특별시도시철도공사

www.smrt.co.kr/main/publish/view.jsp?menuID=001007003009005

승강장안전문 시스템. 승강장안전문은 열차출입문과 연동하여 동작하며, 지하철의 선로와 승강장을 차단하여 안전사고를 방지하고, 승강장 공기 질을 개선하는 ...

스크린도어 - 위키백과, 우리 모두의 백과사전

ko.wikipedia.org/wiki/스크린도어

상트페테르부르크 지하철 2호선 모스크브스카야 역. 출입문을 ... 완전밀폐형(PSD)

PLATFORM SCREEN DOOR : 승강장에서 전장까지를 완전히 막는 형태이다.

PDF 전동차와 연동한 승강장스크린도어(PSD) 제어방식 소개

www.elevatorkorea.org/board/download.php?&bbs_id=1...

현 할 수 있는가에 대한 소개를 PSD 제어장치 시스템을 통해 설명 하고자한다. 1. 서론. 지하철 역사에 설치되는 승강장 스크린도어는 신설역사와 노후된 시설에 대한 ...

What is run via zigbee?

Platform screen door

• PSD 제어방식

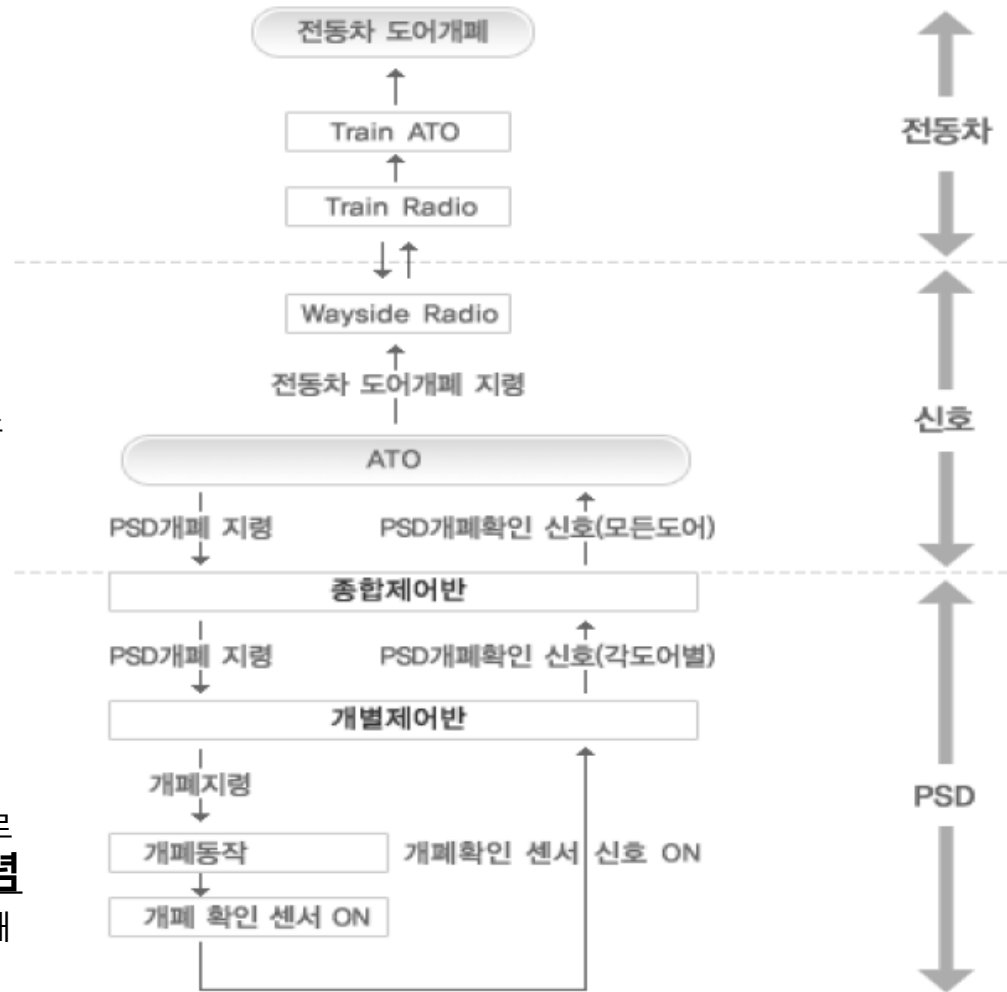
- ATO 장치에 의한 제어
- **무선(RF)통신에 의한 제어**
- 출입문 검출장치에 의한 제어

3.2) RF통신 제어방식

열차(지하철, 국철, 전철 등) 승, 하차 승객의 안전을 위해 역 구내에 설치되어 운용되고 있는 PSD를 제어하기 위한 장치로 **ATO 시스템이 없는** 지하철에 적용하여 시스템의 편의성, 안전성 제고를 위한 방식이다.

통신 운영방식은 승강장 **채널고정** 방식으로 운영되며, 열차승무원이 열차의 상태와 연동되는 PSD의 상태와 제어를 무선으로 할 수 있는 RF-차상장치와 이 RF-차상장치와 무선통신을 할 수 있도록 하는 무선중계장치인 RF-지상장치로 구성되어 있다. 차상장치는 열차에 설치되며 지상장치는 PSD가 있는 역사에 설치한다.

RF 통신방식과 도어검출 센서방식을 같이 사용하는 경우로서 **RF장치에 장애가 있을 경우 Back-Up 개념으로** 승강장 스크린도어를 도어 검출센서 장치에 의해 개폐 하도록 하는 PSD 제어방식이다.





Packet Analysis

What is run via zigbee?

Platform Screen Door

The image shows a Wireshark packet capture of an IEEE 802.15.4 Data frame. The packet list on the left shows a packet of type IEEE 802.15.4 Data, Dst: 0x0c90, Src: 0x0a18. The packet details pane on the right shows the following fields:

- Frame Control Field: Data (0x8841)
 - 001 - Frame Type: Data (0x0001)
 - 0... = Security Enabled: False
 - 0... = Frame Pending: False
 - 0... = Acknowledge Request: False
 - .1... = Intra-PAN: True
 - 10... = Destination Addressing Mode: Short/16-bit (0x0002)
 - ..00... = Frame Version: 0
 - 10... = Source Addressing Mode: Short/16 bit (0x0002)
- Sequence Number: 10
- Destination PAN: 0x5934
- Destination: 0x0c90
- Source: 0x0a18
- Data (15 bytes)
 - Data: 81e190180215042919092428102718
 - [Length: 15]

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII. The first 15 bytes are highlighted in blue, and the last 15 bytes are highlighted in pink. The first 15 bytes are 81 e1 90 18 02 15 04 29 19 09 24 28 10 27 18. The last 15 bytes are 29 19 09 24 28 10 27 18 0a 81 e1 90 18 02 15 04. The ASCII representation shows the first 15 bytes as A..4i... and the last 15 bytes as)..\$(..

Channel

Pan : 0x104

Personal Area Network identifier

RFD (Reduce function device)

Device
beacon Request

End point

Data request
Ack or Data

FFD (Full function device)

Coordinator

src : 0x0000

beacon Response
EPID (mac) , PAN ,src

Broadcast

IEEE 802.15.4
ZigBee

10 Beacon Request

28 Beacon, Src: 0x0000, EPID: 8d:a5:b264:97:e09d:fa

0x0000

Channel 21

beacon

screen

screen

screen

screen

0x0a1a(모란)

0xa10a : 81 e0 ff 1a 02 95 4f (역 구분)



Pan : 0x6934

Personal Area Network identifier

Channel 21

beacon

screen

screen

screen

screen

0x0a1a(모란)

0xa10a : 81 e0 ff 1a 02 95 4f (역 구분)

자신의 도착신호를 알림

Metro - 0x0c03, 0x0c45, 0x0c90

Pan : 0x6934

Personal Area Network identifier

헤더정보 열차/역 좌/우 무결성

0x0c03 : DATA : 81 f0 03 1a 02 51 bf

Channel 21

beacon

screen

screen

screen

screen

0x0a1a(모란)

0xa10a : 81 e0 ff 1a 02 95 4f (역 구분)



자신의 도착신호를 알림

Metro - 0x0c03, 0x0c45, 0x0c90

헤더정보 열차/역 좌/우 무결성

0x0c03 : DATA : 81 f0 03 1a 02 51 bf

OPEN 헤더정보 열차/역 좌/우 연 월 일 시 분 초 문 상태 정보? 무결성

>0x0a1a : DATA : 81 e1 03 1a 02 15 04 26 15 22 35 2a 10 3c eb

차종별 고유번호

Pan : 0x6934

Personal Area Network identifier

지상에서 동기화 신호를 보냄

Channel 21

beacon

screen

screen

screen

screen

0x0a1a(모란)

0xa10a : 81 e0 ff 1a 02 95 4f (역 구분)



자신의 도착신호를 알림

Metro - 0x0c03, 0x0c45, 0x0c90

Pan : 0x6934

Personal Area Network identifier

헤더정보 열차/역 좌/우 무결성

0x0c03 : DATA : 81 f0 03 1a 02 51 bf

헤더정보 열차/역 좌/우 연 월 일 시 분 초 문 상태 정보? 무결성

>0x0a1a : DATA : 81 e1 03 1a 02 15 04 26 15 22 35 2a 10 3c eb

지상에서 동기화 신호를 보냄

차종별 고유번호

OPEN

헤더정보 열차/역 좌/우 무결성

0x0c03 : 81 f1 03 1a 02 cb 1f 40 82 5e 97 상태유지

>0x0a1a : DATA : 81 e1 03 1a 02 15 04 26 15 22 35 28 10 3c eb

Channel 21

beacon

screen

screen

screen

screen

0x0a1a(모란)

0xa10a : 81 e0 ff 1a 02 95 4f (역 구분)



자신의 도착신호를 알림

Metro - 0x0c03, 0x0c45, 0x0c90

Pan : 0x6934

Personal Area Network identifier

헤더정보 열차/역 좌/우 무결성

0x0c03 : DATA : 81 f0 03 1a 02 51 bf

헤더정보 열차/역 좌/우 연 월 일 시 분 초 문 상태 정보? 무결성

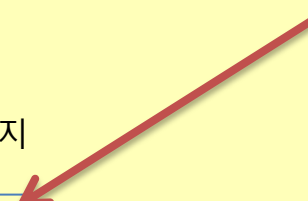
>0x0a1a : DATA : 81 e1 03 1a 02 15 04 26 15 22 35 2a 10 3c eb

지상에서 동기화 신호를 보냄

차종별 고유번호

헤더정보 열차/역 좌/우 무결성

0x0c03 : 81 f1 03 1a 02 cb 1f 40 82 5e 97 상태유지



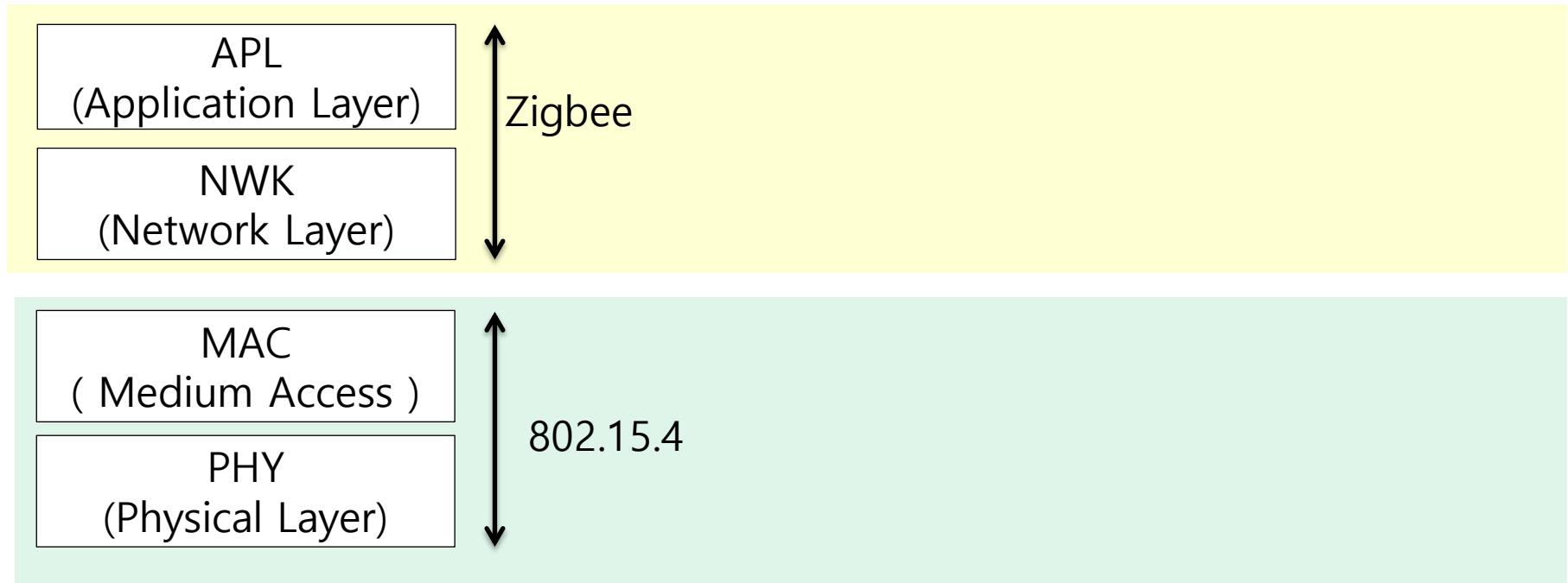
CLOSE

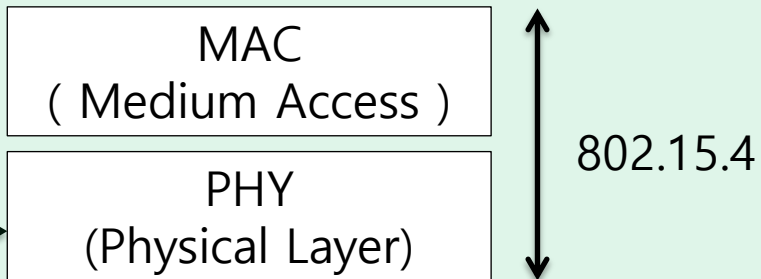
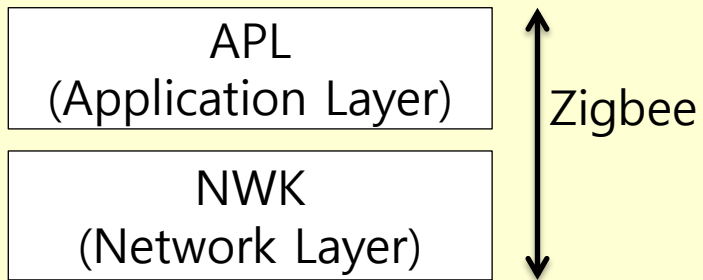
상태변경

0x0c03 : 81 f1 03 1a 02 1b 20 42 82 f4 32

>0x0a1a : DATA : 81 e1 03 1a 02 15 04 26 15 22 35 29 10 3c eb

Zigbee Protocol Structure





주파수를 맞추는 담당을 함

PHY	Frequency band	Data parameters			Spreading parameters	
		Bit rate (kb/s)	Symbol rate (kbaud)	Modulation	Chip rate (Mchips/s)	Modulation
868/915	868.0–868.6 MHz	20	20	BPSK	0.3	BPSK
MHz PHY	902.0–928.0 MHz	40	40	BPSK	0.6	BPSK
2.4 GHz PHY	2.4–2.4835 GHz	250	62.5	16-ary orthogonal	2.0	O-QPSK

APL
(Application Layer)

NWK
(Network Layer)

↑
Zigbee
↓

MAC
(Medium Access)

PHY
(Physical Layer)

↑
802.15.4
↓

Coordinator

Network Device

beacon

Data Req

Ack

Data

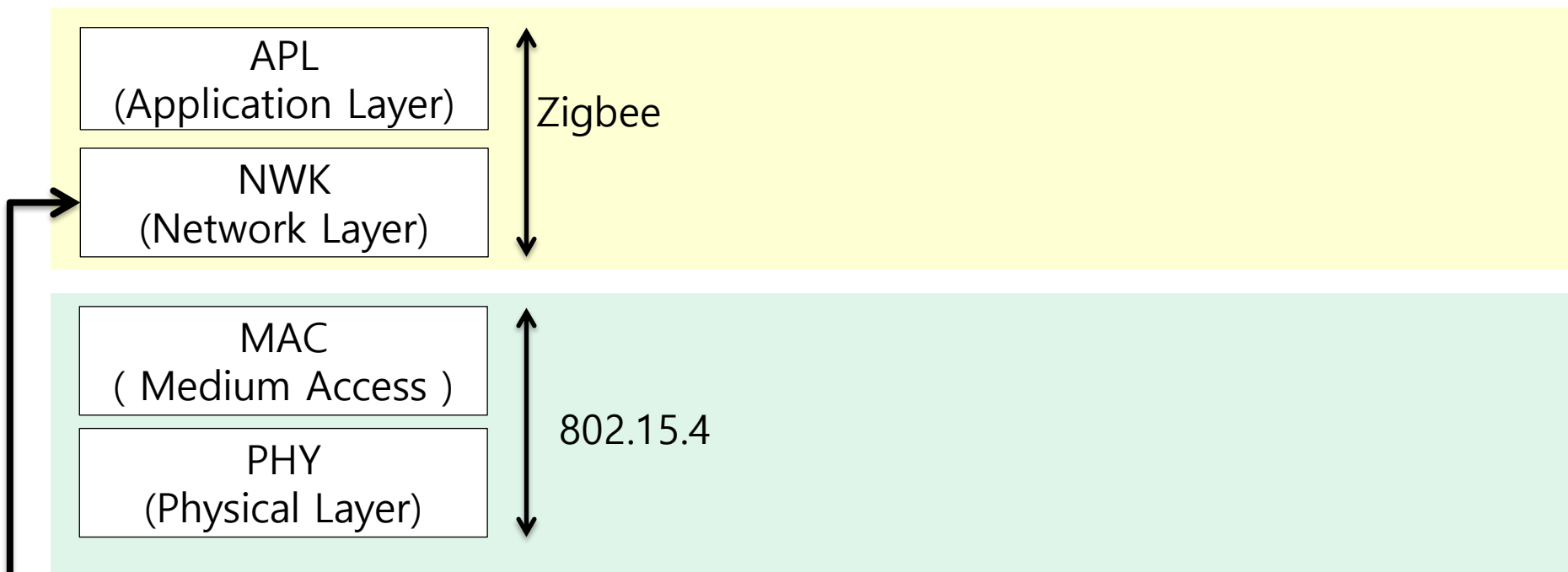
Coordinator

Network Device

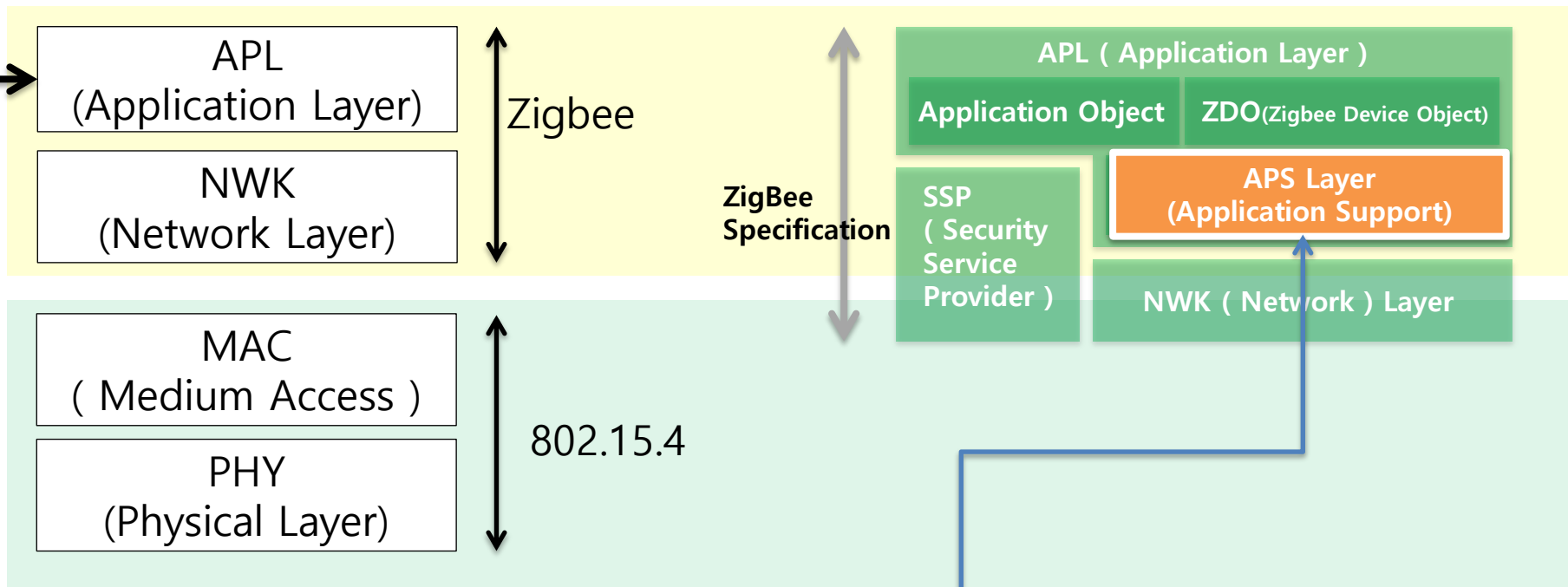
Data Req

Ack

Data



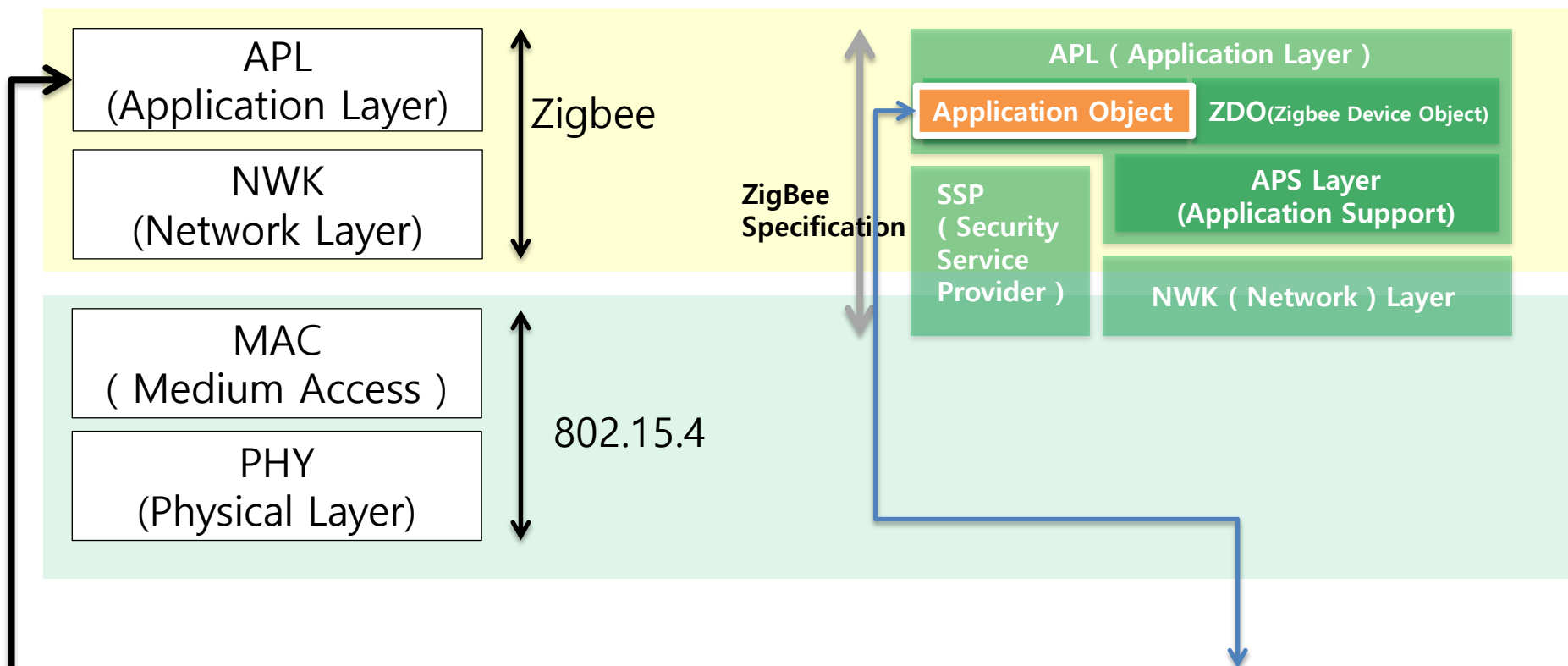
- 새로운 네트워크를 맺는 담당
- Friend node 의 join 과 leave를 담당
- Zigbee Coordinator 의 경우 join 한 device 에게 short address를 부여
- 패킷을 포워딩 을 담당(src, dst 정보를 가지고 있음)
- 이웃을 beacon 신호를 이용하여 검색



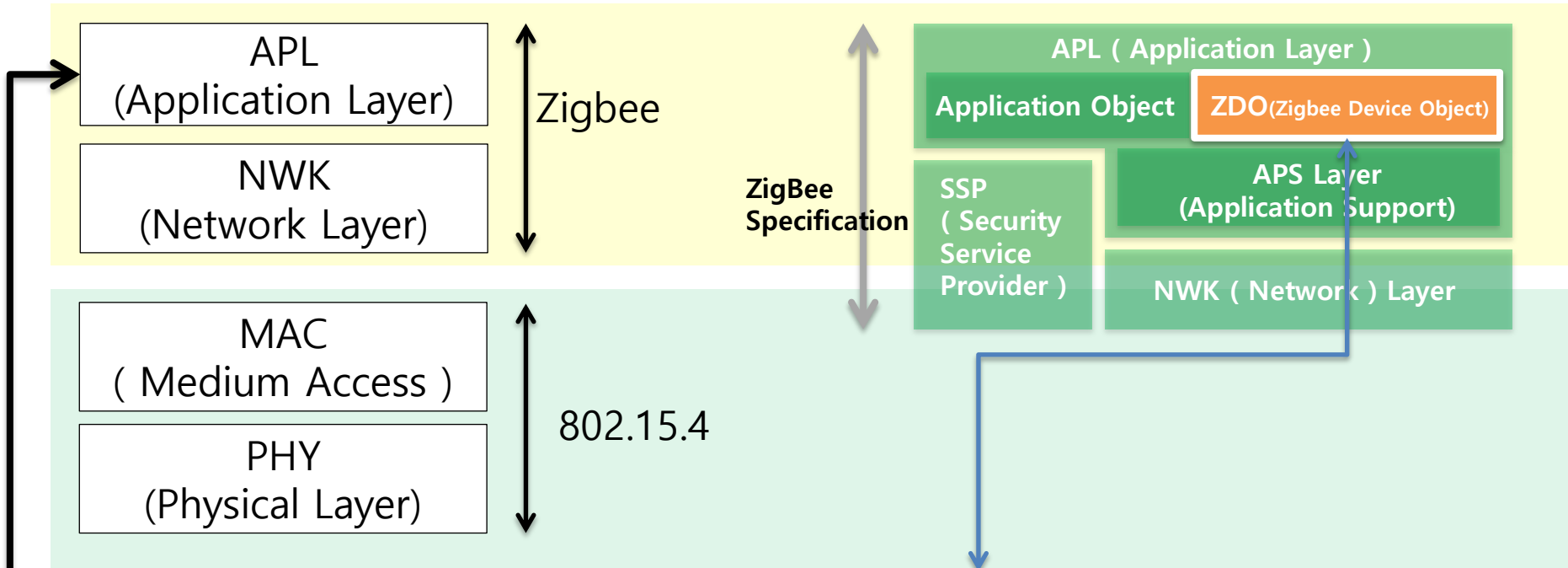
Application Object를 frame 으로 생성하고, 하위 Layer NWK에 연결시킴

일반적인 Data를 전달할 서비스 프로토콜을 정의 함.

Data fragmentation 과 reassembly 를 담당하고 신뢰기반 통신을 하도록 도와줌.



Application Object는 **Application layer**의 **input** 과 **output**에 대한 주체가 됨
 (일종의 APL을 사용하게 되는 구성 장치 라고 볼 수 있음 – 전구, 스위치, LED 등..)
 Application profile 은 application object 에서 동작함



ZigBee Device **Profile** (ZDP)

ZDP 는 ZCL 과 ZDO 들이 모여서 구성되는 최종 프로토콜

ex) Device , Service Discovery, Binding service, Management service.

ZigBee APL 통신이 되기 위해선

ZDO(The ZigBee Device Object) &&

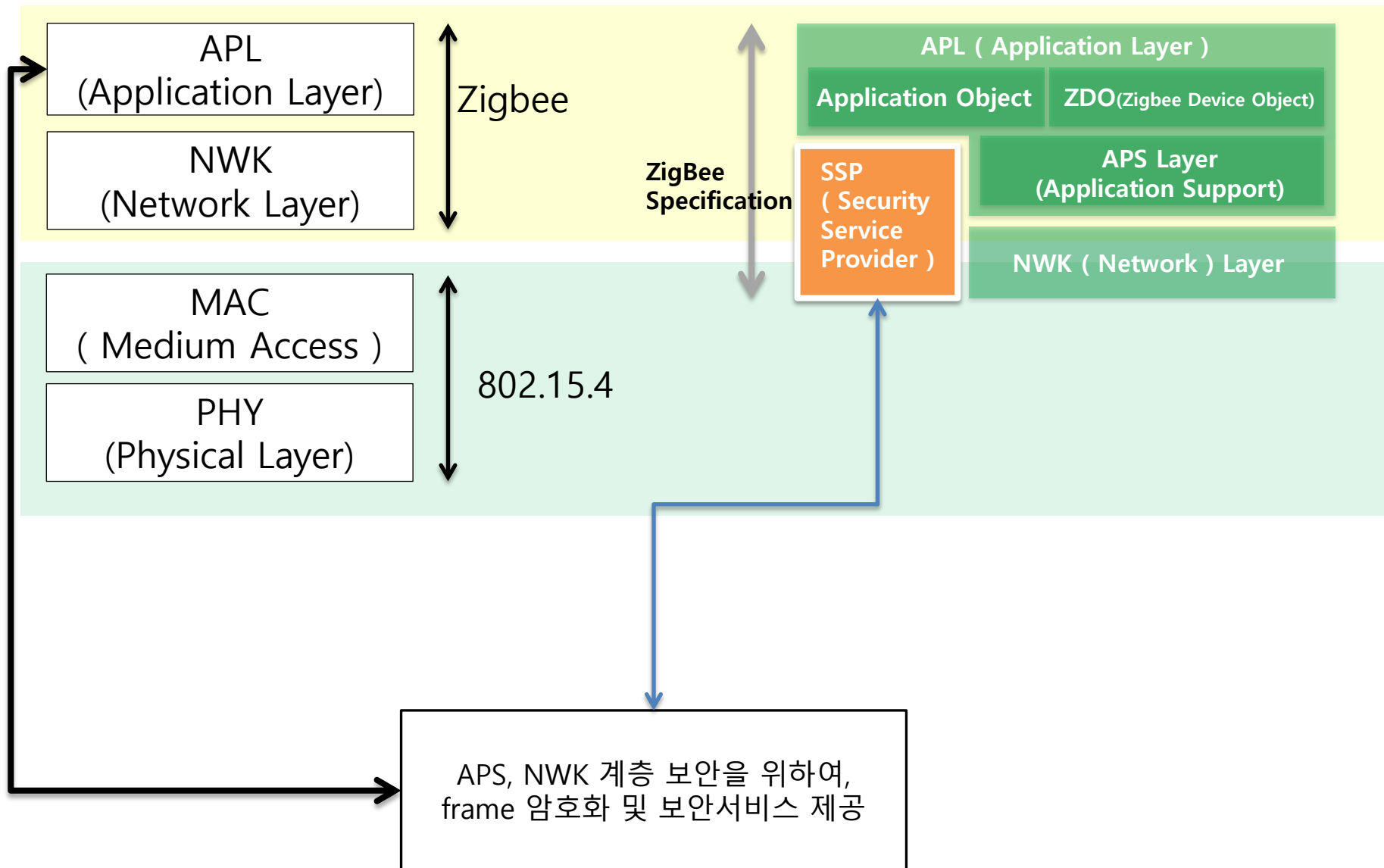
ZCL(The ZigBee Cluster Library) 로 프로토콜이 결정됨
(Cluster 란 APO 인터페이스의 한 종류)

Profile ID: 16-bits(0x0000~0xFFFF)

Application Pro

Stack Pro

Cluster ID: 8-bits(0x00~0xFF)



How express zigbee structure in wireshark?

Zigbee Packet Structure

```
+ Frame 1: 54 bytes on wire (432 bits), 52 bytes captured (416 bits)
+ IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x2bdc
```

```
+ ZigBee Network Layer Data, Dst: 0x0000, Src: 0x2bdc
```

```
- ZigBee Application Support Layer Data, Dst Endpt: 1, Src Endpt: 1
```

```
- Frame Control Field: Data (0x40)
```

```
.... ..00 = Frame Type: Data (0x00)
```

```
.... 00.. = Delivery Mode: Unicast (0x00)
```

```
..0. .... = Security: False
```

```
.1.. .... = Acknowledgement Request: True
```

```
0... .... = Extended Header: False
```

```
Destination Endpoint: 1
```

```
Cluster: Intruder Alarm System Zone (0x0500)
```

```
Profile: Home Automation (0x0104)
```

```
Source Endpoint: 1
```

```
Counter: 66
```

```
- ZigBee Cluster Library Frame, Cluster-specific Command: 0x00, Seq: 8
```

```
- Frame Control Field: cluster-specific (0x09)
```

```
.... ..01 = Frame Type: Cluster-specific (0x01)
```

```
.... .0.. = Manufacturer Specific: False
```

```
.... 1... = Direction: To Client
```

```
....0 .... = Disable Default Response: False
```

```
Sequence Number: 8
```

```
Command: Zone Status Change Notification (0x00)
```

```
- Zone Status: 0x0024
```

```
.... ....0 = Alarm 1: Closed or not alarmed
```

```
.... ....0. = Alarm 2: Closed or not alarmed
```

```
.... ....1.. = Tamper: Tampered
```

```
.... ....0... = Battery: Battery OK
```

```
.... ....0 .... = Supervision Reports: Does not report
```

```
.... ....1. .... = Restore Reports: Reports restore
```

```
.... ....0... = Trouble: OK
```

```
.... ....0... = AC (mains): AC/Mains OK
```

```
Extended Status: 0x00
```

```
Zone ID: 0x00
```

APS support Layer

APS Layer

Profile 번호에 따라
protocol이 결정됨

Zigbee singnal generate

What did I choooce?

Killerbee framework



The screenshot shows the Google Code project page for Killerbee. The browser's address bar displays the URL `https://code.google.com/p/killerbee/source/checkout`. The page header features the Killerbee logo (a yellow bee with the Chinese character '道' on its back) and the project name 'killerbee' in a large, bold font. Below the name is the tagline 'Framework and tools for exploiting ZigBee and IEEE 802.15.4 networks'. A navigation bar contains links for 'Project Home', 'Downloads', 'Wiki', 'Issues', and 'Source', with the 'Source' link highlighted by a red rectangular box. Below the navigation bar, there are links for 'Checkout', 'Browse', and 'Changes'. A light blue box contains the text: 'How-to: Explore this project's source code by clicking the "Browse" and "Changes" links above.' Under the heading 'Command-line access', there is a paragraph stating: 'Use this command to anonymously check out the latest project source code:'. Below this, a light blue box contains the following text: '# Non-members may check out a read-only working copy anonymously over HTTP. svn checkout **http://killerbee.googlecode.com/svn/trunk/** killerbee-read-only', which is also enclosed in a red rectangular box.

← → ↻ <https://code.google.com/p/killerbee/source/checkout>

 **killerbee**
Framework and tools for exploiting ZigBee and IEEE 802.15.4 networks

[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) **Source**

[Checkout](#) [Browse](#) [Changes](#)

How-to: Explore this project's source code by clicking the "Browse" and "Changes" links above.

Command-line access

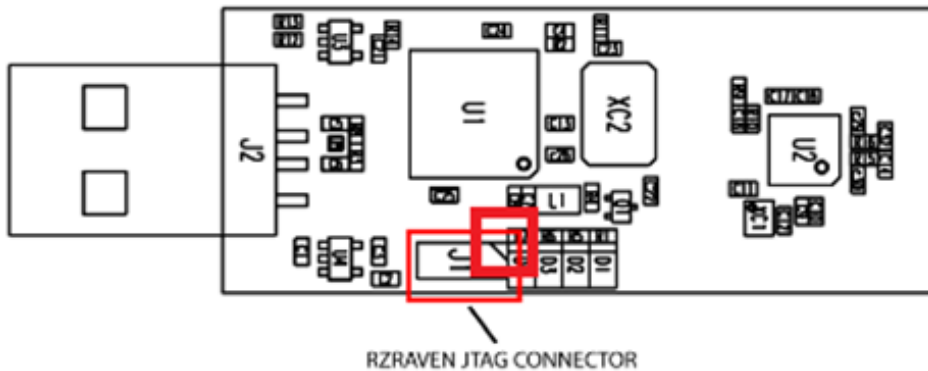
Use this command to anonymously check out the latest project source code:

Non-members may check out a read-only working copy anonymously over HTTP.
svn checkout **http://killerbee.googlecode.com/svn/trunk/** killerbee-read-only

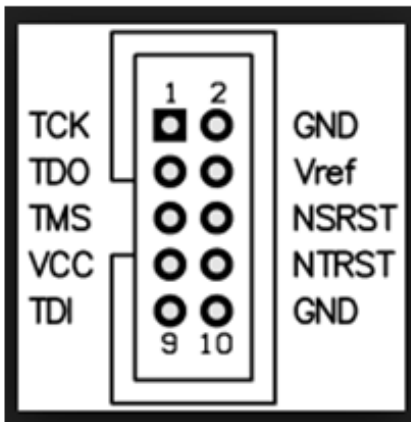
What did I choooce?

Killerbee framework

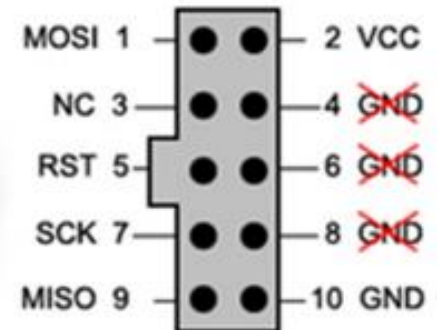
Figure 8-1 Placement of RZRAVEN JTAG Connector



1번 포트 위치 표시

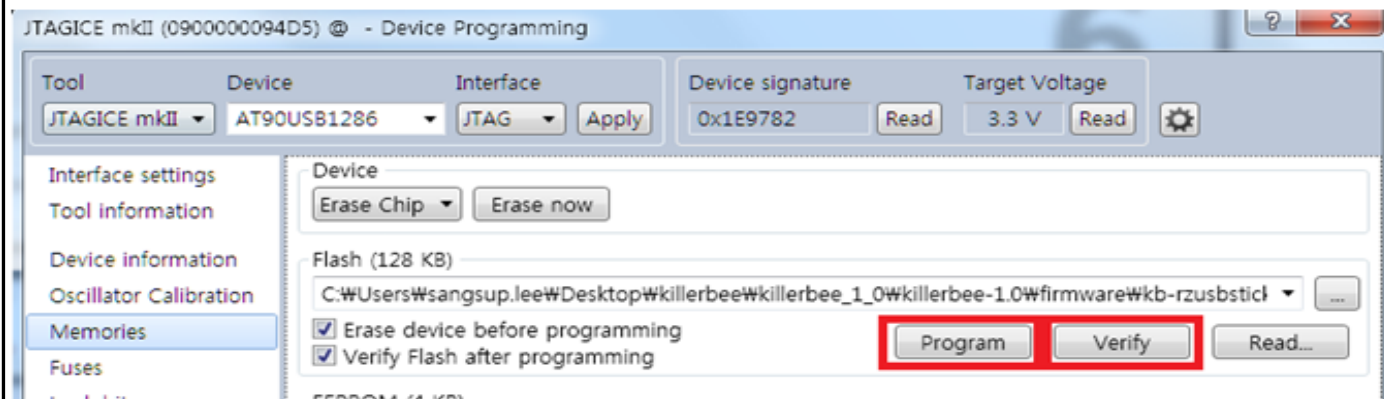
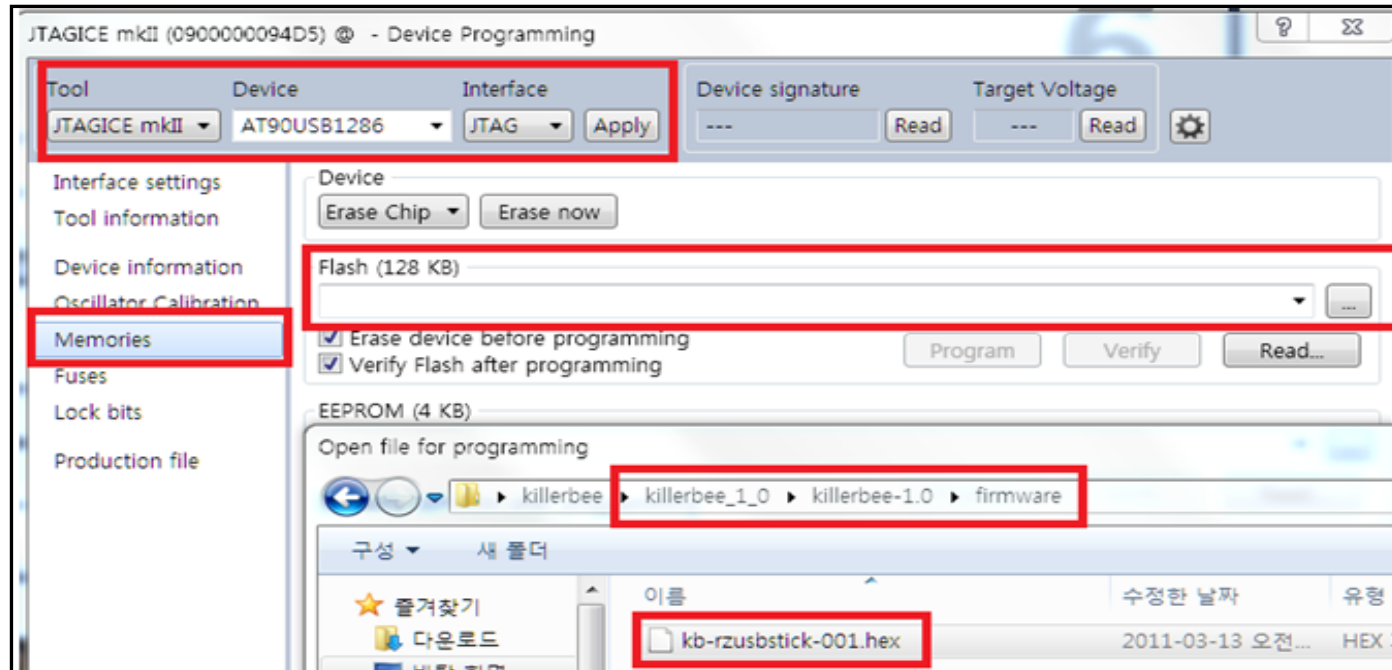


(jtag pin)



What did I choooce?

Killerbee framework



메모리 플래싱 완료 되면 초록색 불빛 되신 호박색 불빛이 뜬

What did I choooce?

Killerbee framework

- **zbid** : List available devices supported
- **zbdump** : "tcpdump-w" clone (libpcapor commercial DaintreeSNA
- **zbconvert** : convert capture file formats
- **zbreplay** : **Replay attack**
- **zdsniff** : OTA crypto key sniffer
- **zbfind** : GUI for ZigBee location tracking
- **zbgoodfind** : Search memory dump for key

killerBee API

- 채널 선택 및 패킷 인젝트, 스니핑을 위한 인터페이스 제공
- MAC NWK 와 APS frame 의 디코딩
- 암호화 방법과 패킷 캡처 기능 제공
- zbdsniff 같은 패킷 스니핑 기능을 api 로 제공
- API documentation 은 ToorCon killerBee CD 에 있음

[cybertools](#) / [scapy-radio](#) / [source](#) / — [Bitbucket](#)

[bitbucket.cassidiancybersecurity.com/scapy-radio](#) ▾ [이 페이지 번역하기](#)

This tool is a modified version of [scapy](#) that aims at providing an quick and ...

Bluetooth LE (advertising only); 802.15.4 (used by [Zigbee](#), [Xbee](#), [6LoWPAN](#)) ...

[이 페이지를 2번 방문했습니다. 최근 방문 날짜: 15. 3. 21](#)

cybertools / scapy-radio / source / — Bitbucket

bitbucket.cassidiancybersecurity.com/scapy-radio ▾ [이 페이지 번역하기](#)

This tool is a modified version of `scapy` that aims at providing an quick and ...

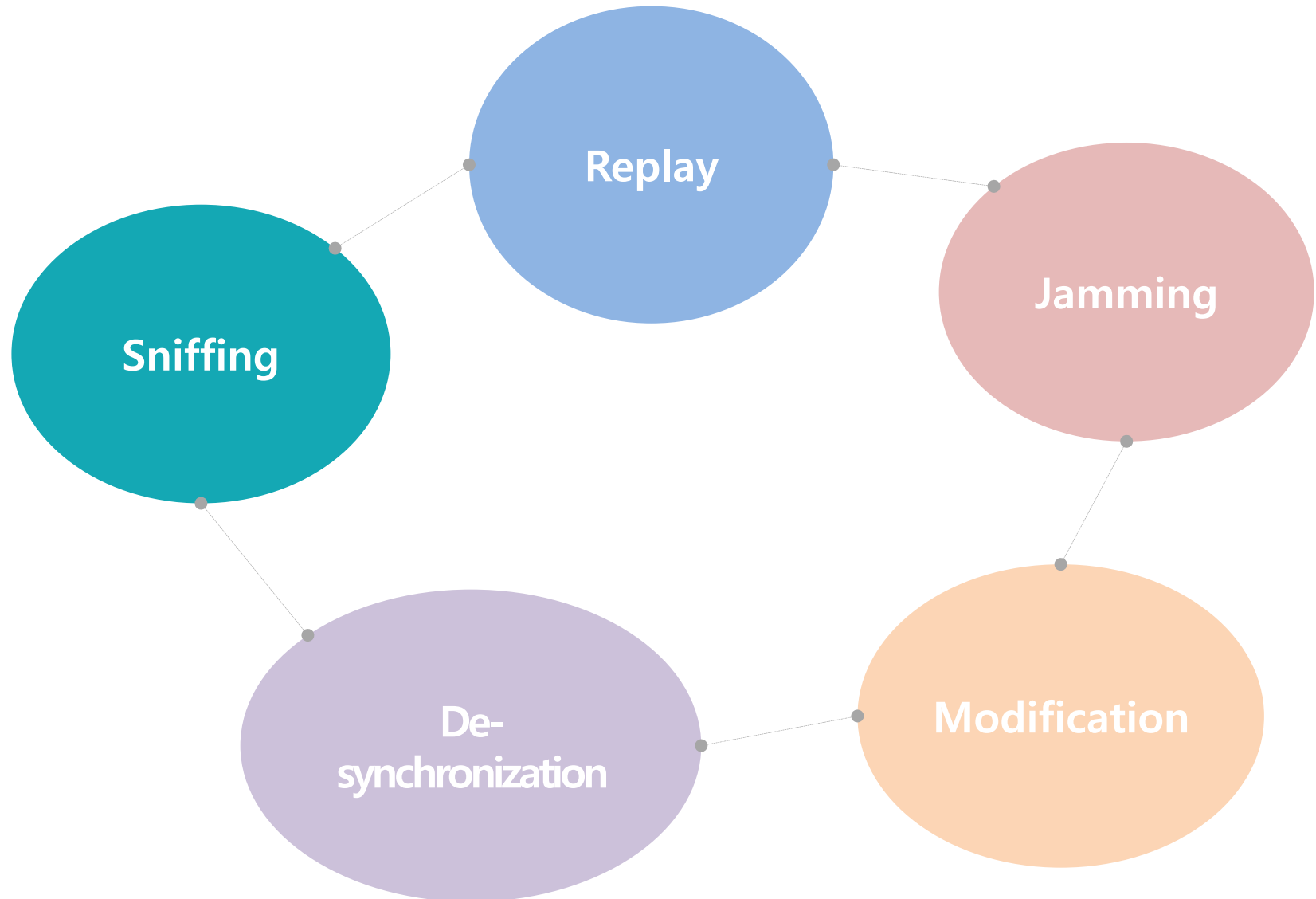
Bluetooth LE (advertising only); 802.15.4 (used by Zigbee, Xbee, 6LoWPAN) ...

이 페이지를 2번 방문했습니다. 최근 방문 날짜: 15. 3. 21

```
1 |#!/usr/bin/env python
2 |import socket
3 |import time
4 |import sys
5 |
6 |from scapy.config import conf
7 |from scapy.layers import dot15d4, zigbee
8 |
9 |# GNU radio packet receiver server
10 |UDP_IP = "127.0.0.1"
11 |UDP_PORT = 52001
12 |
13 |setattr(conf, "zigbee_force_HA_profile", True)
14 |setattr(conf, "zigbee_network_key", "7b2f7c9dd5719184094bf72dbbd0e150".decode('hex'))
15 |setattr(conf, "debug_dissector", True)
16 |
17 |
```

What kind of attack do we have?

List of wireless attack



Zigbee security (encryption)

Security Services

▪ Frame protection

- AES-CCM* 기반 암호화 -> 기밀성 및 무결성 제공.

NWK, APS layer services

▪ Key establishment

- 두 노드 간의 인증 후, 공유키를 생성.

APS layer services

▪ Key transport

- 하나의 노드에서 다른 노드로 무선 통신을 통해 키를 전달 (주로 ZC 또는 ZR로부터).

▪ Device management

- 새로운 노드의 네트워크 참여 및 관리.

Security procedures 참고

Using encryption key

Master Key (MK) - optional

두 노드 (TC-node 또는 node-node) 사이의 Pre-shared secret.

APL 계층의 보안키(LK) 생성을 위한 SKKE (Symmetric Key-Key Establishment) 에서 사용됨.

Key-transport 또는 pre-installation을 통하여 생성.

Pre-Installed Key 라고도 불림

Link Key (LK) - optional

두 노드 (TC-node 또는 node-node) 사이의 Shared Key (128-bit).

APL 계층에서 Unicast 메시지를 보호하기 위하여 사용됨.

Network Key (NK)

네트워크 내의 모든 노드에게 공유되는 128 bits 키. (Broad Cast , 디바이스들사이에서 모두 공유)
NWK, APL 계층 보안을 위해 사용될 수 있음.

Key-transport 또는 pre-installation을 통하여 생성.

Security modes* of Network Key

SSM (Standard Security Mode)

HSM (High Security Mode): ZigBee PRO에서 지원, 반드시 암호화된 NK 전송.

* NK 분배 방법과 Network frame counter 초기화 방법 등에 따라 분류한 것으로,
전송 메시지 보안과 관련 없음.

Using encryption key

⊕ Frame 1: 54 bytes on wire (432 bits), 52 bytes captured (416 bits)

⊕ IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x2bdc

⊖ ZigBee Network Layer Data, Dst: 0x0000, Src: 0x2bdc

⊖ Frame Control Field: Data (0x0248)

.... ..00 = Frame Type: Data (0x0000)

.... ..00 10.. = Protocol Version: 2

.... ..01.. = Discover Route: Enable (0x0001)

.... ..0 = Multicast: False

.... ..1. = Security: True

.... ..0.. = Source Route: False

.... 0... = Destination: False

.... 0 = Extended Source: False

Destination: 0x0000

Source: 0x2bdc

Radius: 30

Sequence Number: 92

⊖ ZigBee Security Header

⊖ Security Control Field

...0 1... = Key Id: Network Key (0x01)

..1. = Extended Nonce: True

Frame Counter: 28706

Extended Source: Ember_00:03:1b:90:f8 (00:0d:6f:00:03:1b:90:f8)

Key Sequence Number: 0

Message Integrity Code: 737c057f

[Decryption Key:]

Security Level : CCM* 사용 여부

Key Identifier : 사용된 Key 종류

Extended Nonce : Aux Header 내 source IP 포함여부

Frame Counter : replay 공격 방지용 count Number

Key sequence number : Key ID가 NW key 일 때 사용

어떤 key를 사용하여 secure 하게 보호하는지..
(여기서는 network key)

Zigbee 통신의 위변조를 막기 위한 MIC 무결성 체크 코드

⊕ ZigBee Application Support Layer Data, Dst Endpt: 1, Src Endpt: 1

⊕ ZigBee Cluster Library Frame, Cluster-specific Command: 0x00, Seq: 8

Encryption & verify

AES-CCM* 운영모드

모드	적용 방식
0x00. No security	보안 적용 안함
0x01. AES-CBC-MAC32 (MIC-32)	32bit 메시지 인증
0x02. AES-CBC-MA64 (MIC-64)	64bit 메시지 인증
0x03. AES-CBC-MAC128 (MIC-128)	128bit 메시지 인증
0x04. AES-CTR (ENC)	암호화만 적용
0x05. AES-CCM-32	32bit 암호화 및 메시지 인증
0x06. AES-CCM-64	64bit 암호화 및 메시지 인증
0x07. AES-CCM-128	128bit 암호화 및 메시지 인증

보안 모드 시 **Auxiliary header** 가 추가 되고
암호화 방식에 대한 종류가 저장됨

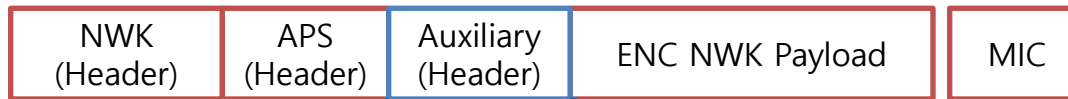


보안 모듈을 사용하지 않는 일반 패킷

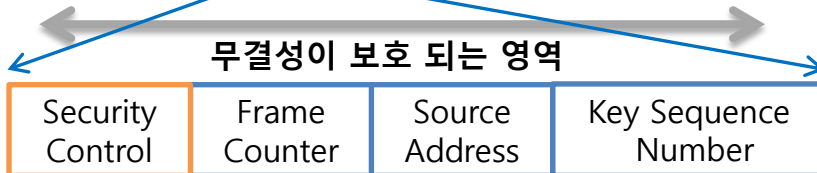


NWK frame에 대한 무결성 기밀성 제공

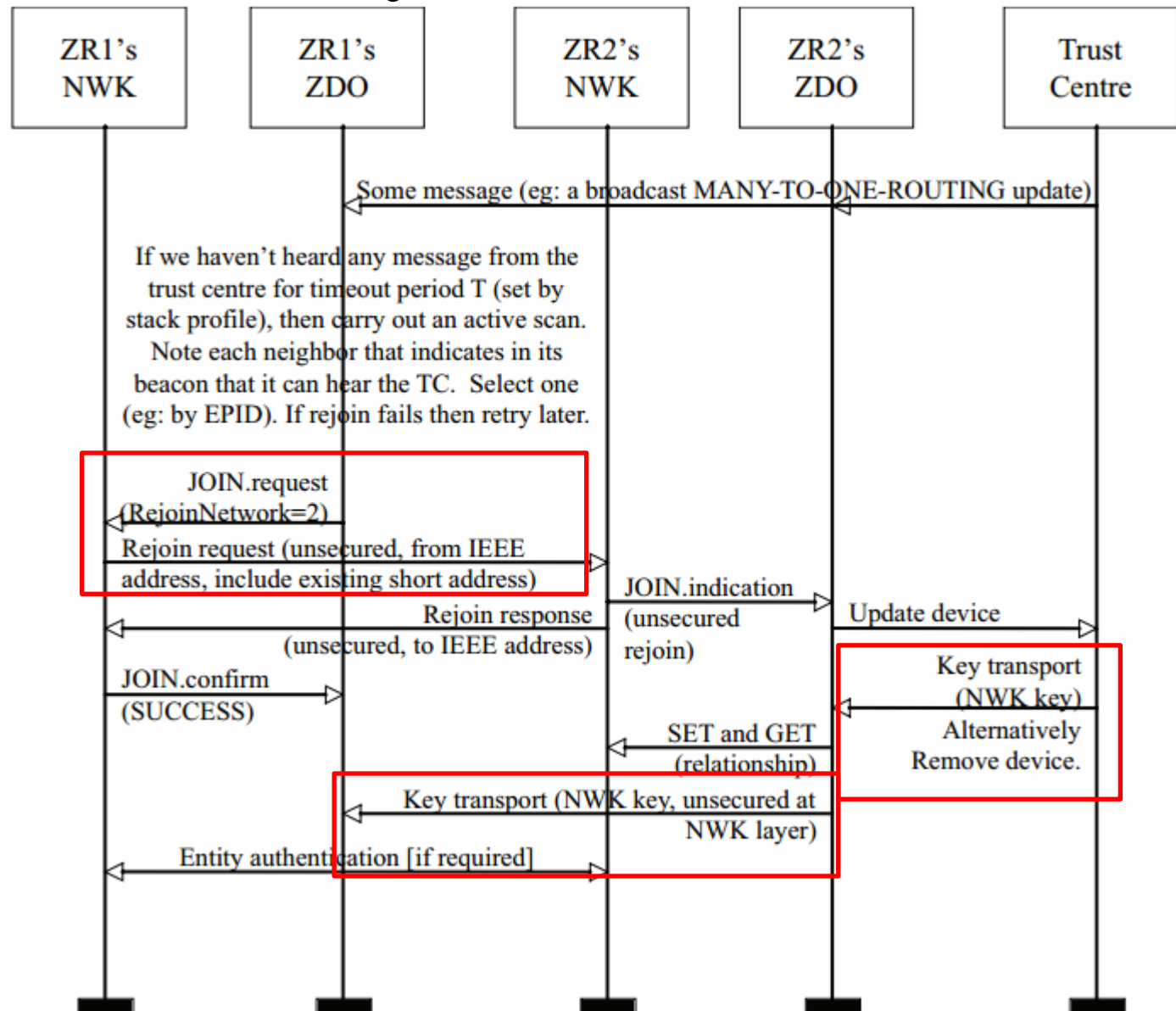
← 무결성이 보호 되는 영역 →



APL frame 에 대한 무결성 기밀성 제공



Encryption & verify



- VCDP
- VITA 49
- VLAN
- VNC
- VP8
- VRRP
- VSS-Monitoring
- Vuze-DHT
- WBXML
- WebSocket
- WiMax (wmx)
- WIMAX ASN CP
- WIMAX MAC-PHY
- WINS-Replication
- WOW
- X.25
- X11
- X2AP
- XMCP
- XML
- XMPP
- XOT
- YAMI
- YMSG
- ZEP
- ZigBee Green Power
- ZigBee NWK**
- + Statistics

Security Level: **AES-128 Encryption, No Integrity Protection**

Pre-configured Keys: Edit...

Pre-configured Keys - Profile: Default

Key	Byte Order	Label
-----	------------	-------

Up

Down

New

Edit...

Copy

Delete

Refresh

Clear

Key:

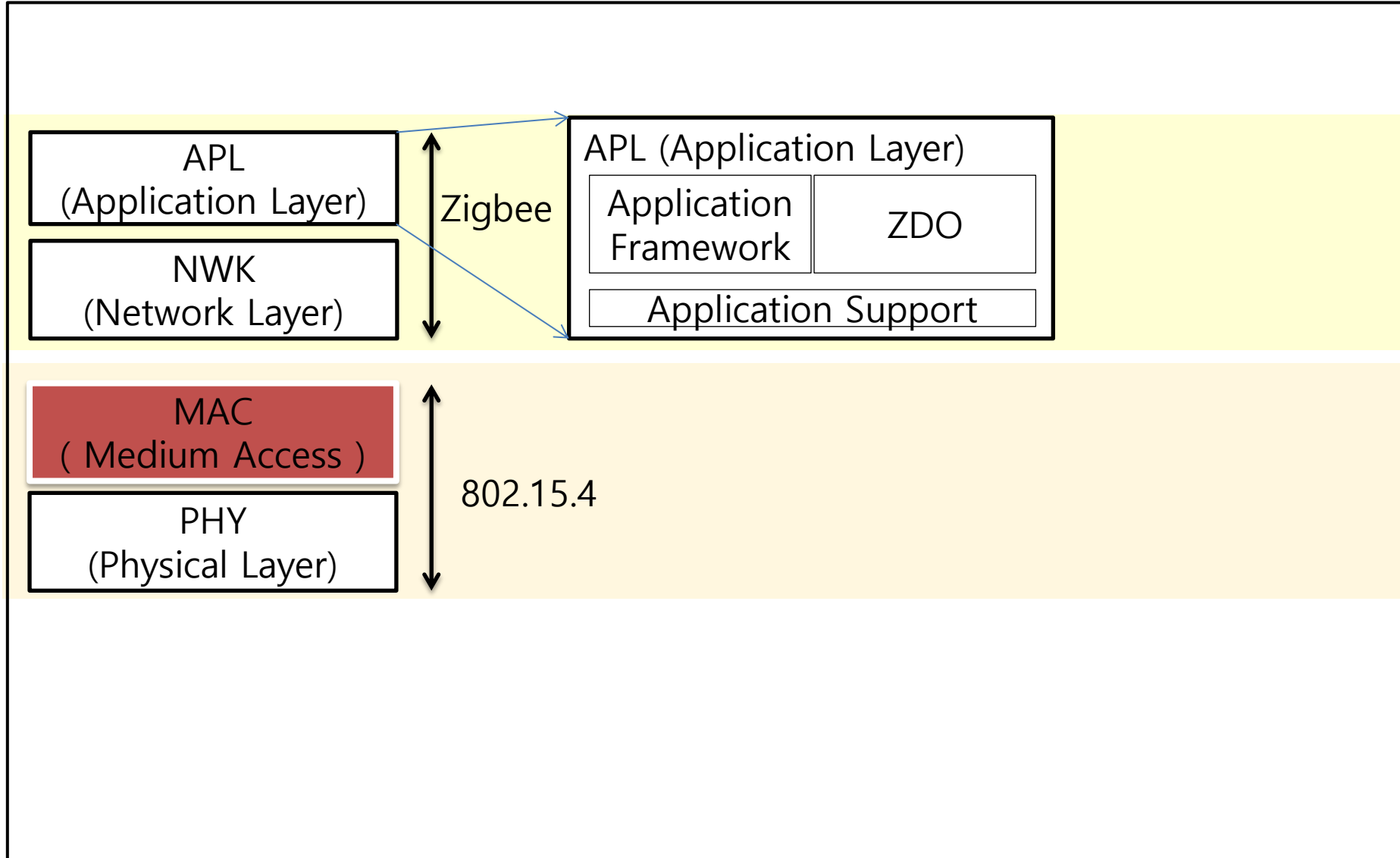
Byte Order: Normal

Label:

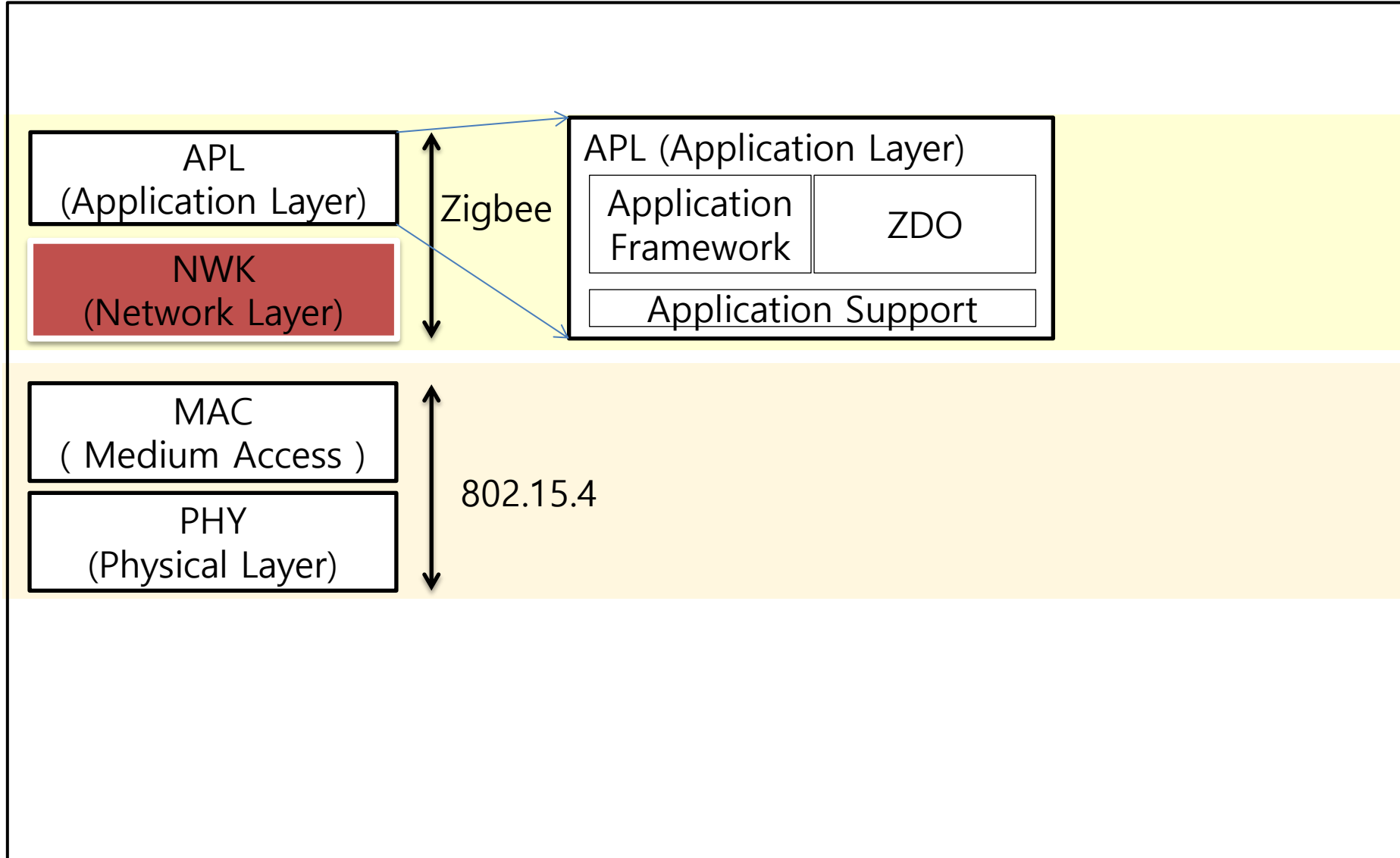
OK Cancel

OK Apply Cancel

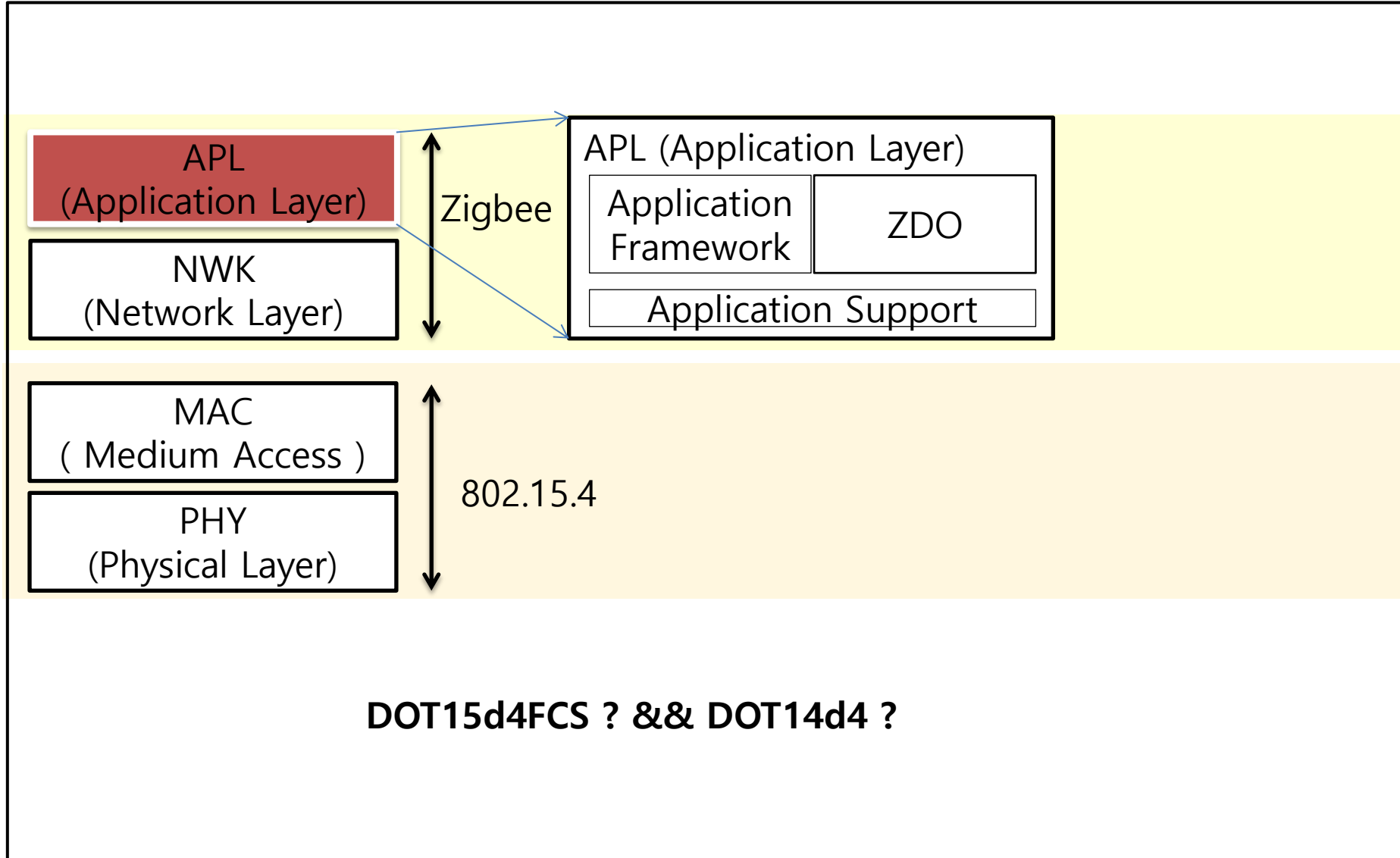
Sequence number to prevent repeat attacks



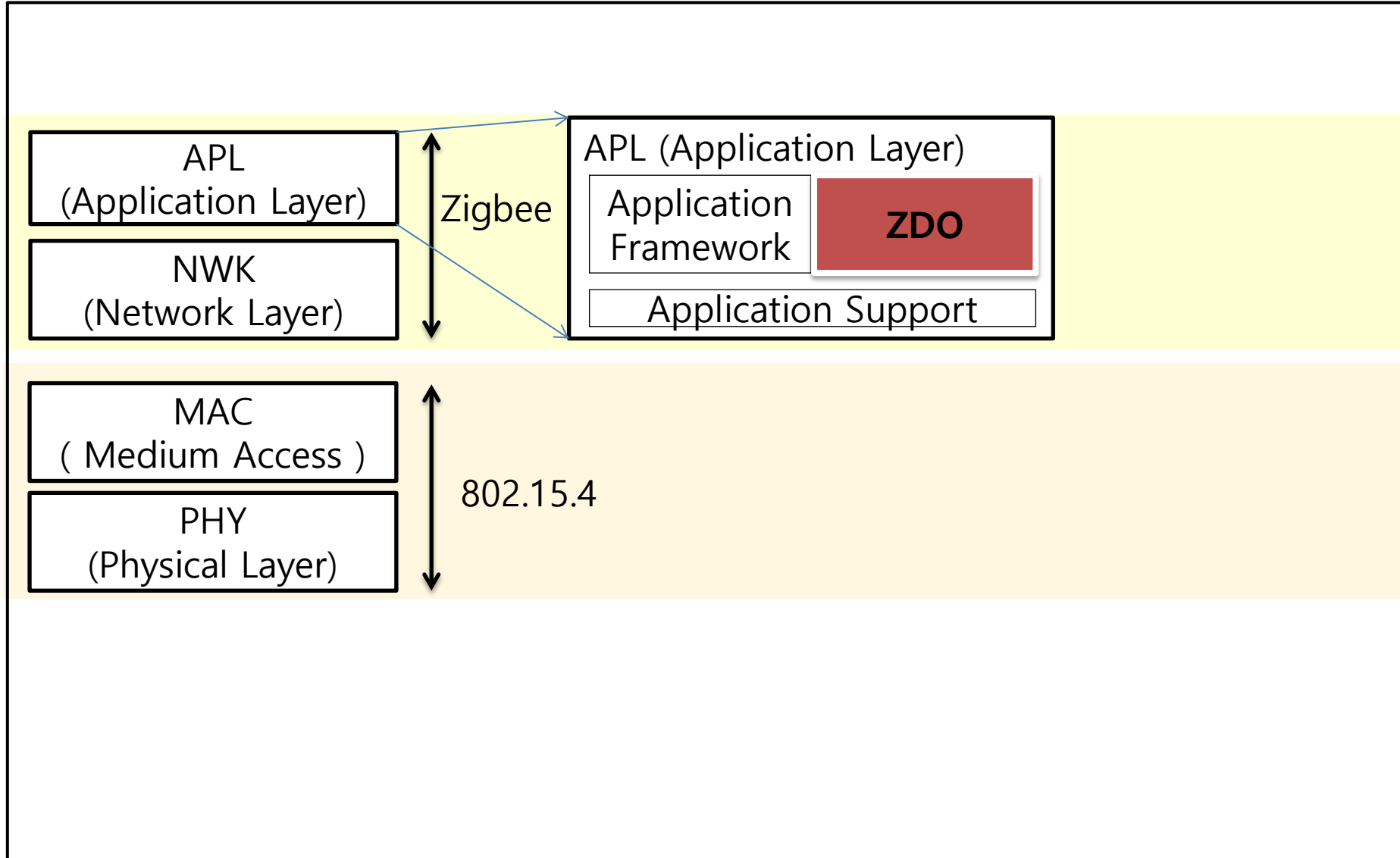
Sequence number to prevent repeat attacks



Sequence number to prevent repeat attacks



Sequence number to prevent repeat attacks



Encryption & verify

```
⊕ Frame 258: 65 bytes on wire (520 bits), 63 bytes captured (504 bits)
⊕ IEEE 802.15.4 Data, Dst: 0x4bd9, Src: 0x0000
⊕ ZigBee Network Layer Data, Dst: 0x4bd9, Src: 0x0000
⊕ ZigBee Application Support Layer Command
  ⊕ Frame Control Field: Command (0x21)
    Counter: 213
  ⊕ ZigBee Security Header
  ⊕ Command Frame: Transport Key
    Command Identifier: Transport Key (0x05)
    Key Type: Standard Network Key (0x01)
    Key: b4d71ce22a7583d30c6247128d0ed24b
    Sequence Number: 0
    Extended Destination: Ember_00:03:1b:90:f8 (00:0d:6f:00:03:1b:90:f8)
    Extended Source: Physical_00:78:57:00:01 (d0:52:a8:00:78:57:00:01)
```

4 Byte

8 Byte

16 Byte

32 Byte

Network key

Encryption

key

Initial Vector

MIC byte

Plain data

APS header

IV =

src addr

Frame Counter

Sec level

Get the key !!

END
