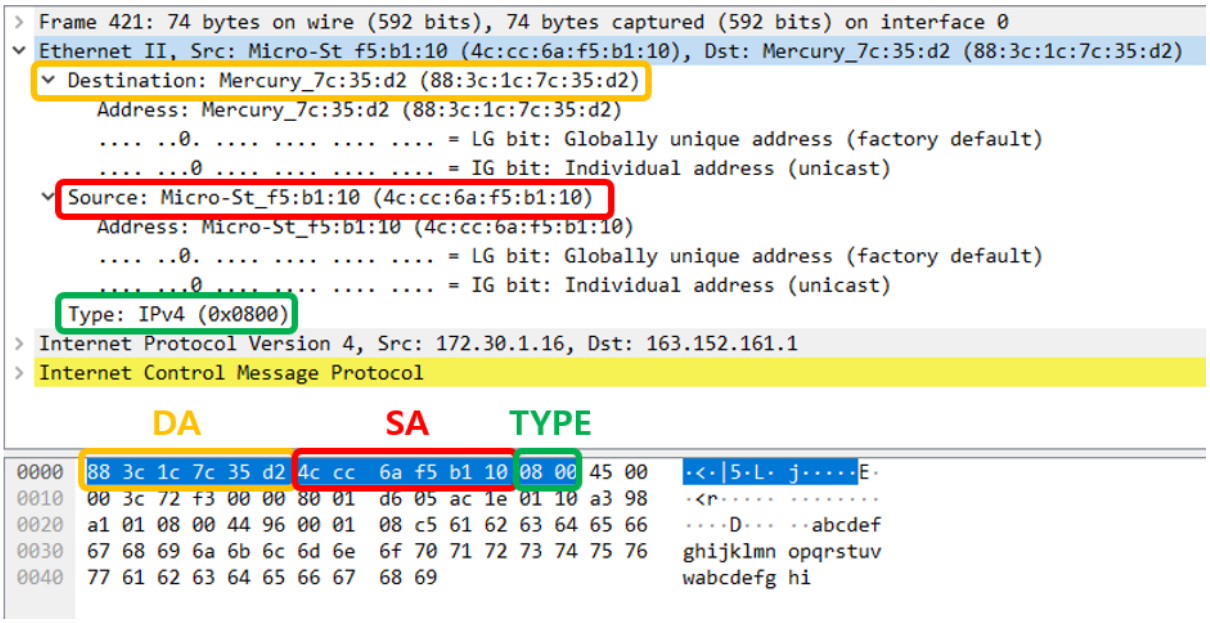
데이터통신 #1-1

패킷 캡쳐 화면과 Ethernet (802.3) Header 분석

데이터링크 계층

이더넷 프레임 헤더 구조



이더넷 프레임 헤더 분석

Destination MAC address [88 3c 1c 7c 35 d2]

목적지의 데이터링크 계층 주소를 6바이트로 나타낸다.

6 바이트 주소 첫번째 비트가 1이면 Multicast이고, 0이면 Boradcast를 나타낸다.

Source MAC address [4c cc 6a f5 b1 10]

출발지의 데이터링크 계층 주소를 6바이트로 나타낸다.

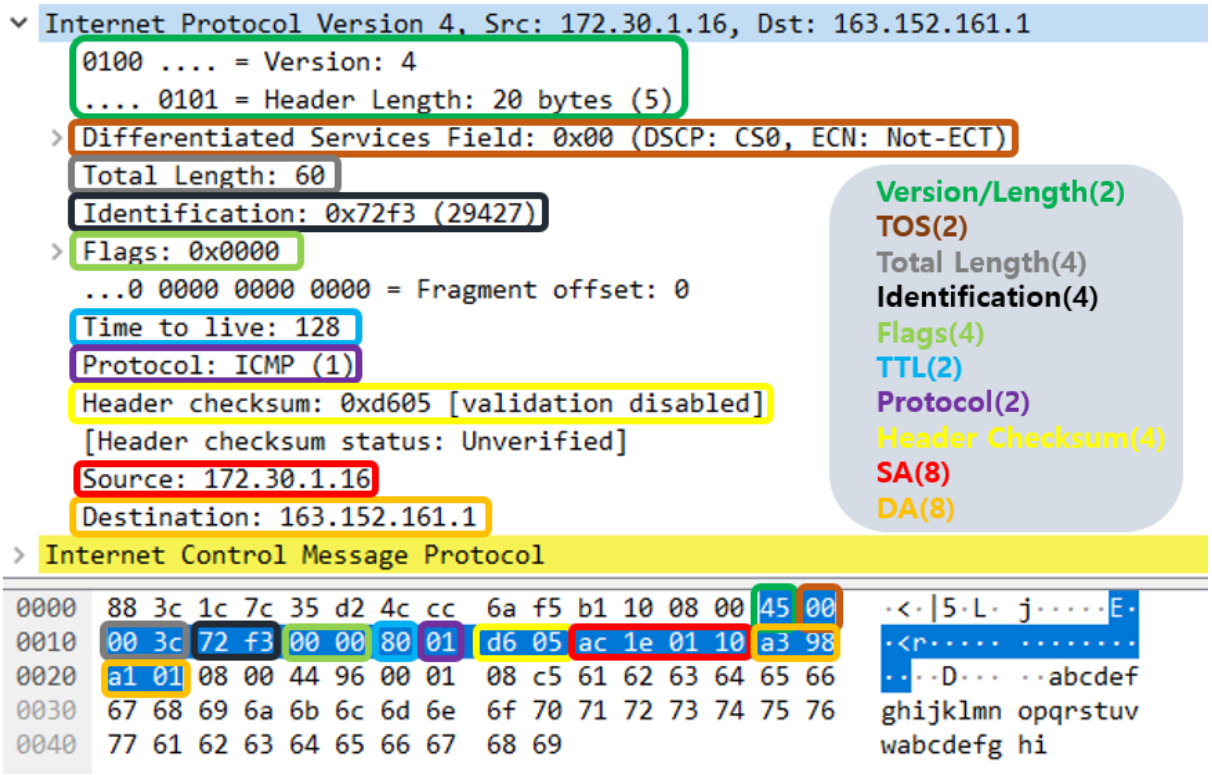
Type [08 00]

수신된 프레임의 값에 따라 IEEE 802.3 Length (0x600 미만), Ethernet II Type (0x0600 이상) 으로 판단한다.

여기서는 타입을 의미하며 0x0800은 IP Protocol을 의미한다.

네트워크 계층

패킷 헤더 구조



패킷 헤더 분석

Version [4]

패킷이 IPv4 인지 IPv6인지 나타낸다. 여기서는 IPv4를 사용한다.

Length [5]

IHL 이라고도 하며 IP 헤더의 길이를 4byte단위로 나타낸다. 여기서는 20 이다.

Type Of Service(TOS) [00]

음성, 영상, 문자와 같은 IP패킷의 우선순위를 정의한다.

TotalLength [00 3c]

헤더와 데이터를 포함한 전체 패킷의 길이를 바이트 단위로 나타낸다. 최소 46 바이트부터 최대 1500 바이트까지 표현한다.

Identification [72 f3]

일련번호로서 패킷의 식별을 담당한다.

Flag [00 00]

패킷이 단편화 되었는지 판단할 수 있다. 각 비트의 역할은 다음과 같다.

첫 비트는 예약됨 (0).

두번째 비트는 0 이면 분할, 1이면 분할되지 않음. (Don't' Fragment)

세번째 비트는 0 이면 이후 패킷 더 존재함, 1이면 마지막 패킷. (More Fragment)

TTL [80]

IP 패킷의 수명을 나타낸다. 수명이란 패킷이 경유할 수 있는 최대 홉 수를 의미한다. 최대값으로 255를 가질 수 있으며 라우터를 통과할 때 마다 1씩 감소한다. 이 값이 0이 되면 송신측으로 ICMP 메시지와 함께 패킷이 폐기된다.

Protocol [01]

상위 계층 프로토콜을 나타낸다. 대표적인 프로토콜은 다음과 같다.

ICMP : 0x01

TCP : 0x06

UDP : 0x11

Header Checksum [d6 05]

IP 패킷 헤더의 오류 발생 체크에 사용되는 비트이다. 와이어샤크에서는 이 값이 기본적으로 disabled 되어 있어서 [validation disabled] 메시지를 확인할 수 있다.

Source Address [ac 1e 01 10]

송신측의 IP 주소를 나타낸다. 여기서는 172.30.1.16

Destination Address [a3 98 a1 01]

수신측의 IP 주소를 나타낸다. 여기서는 163.152.161.1

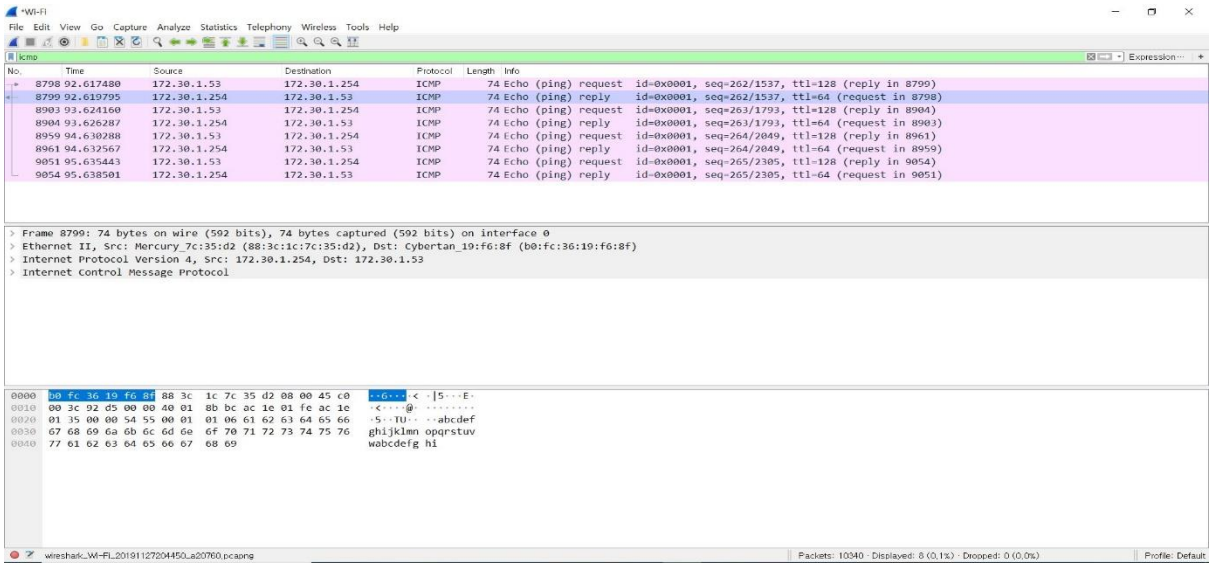
데이터통신 #1-2

wifi (802.11) 패킷 캡쳐 화면과 Layer2 Header 분석 by WireShark

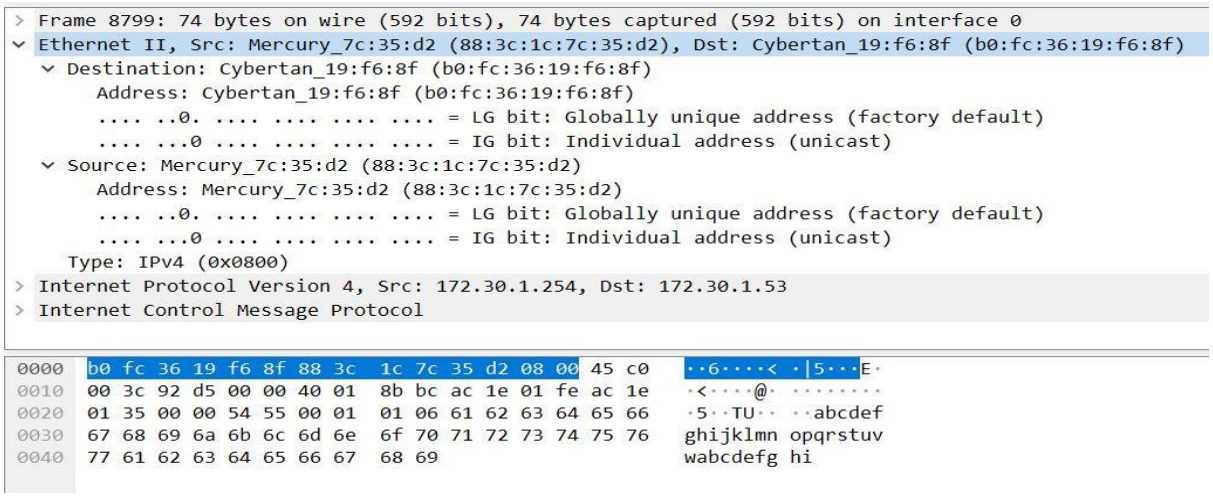
개요

무선환경으로 바뀌었지만 전반적으로 #1-1의 802.3 Ethernet의 내용과 유사하다. 따라서 #1-2 에서는 #1-1과 같은 방식으로 설명하되 중복된 부분은 생략한다.

전체 화면 캡쳐



이더넷 프레임 구조



Destination MAC address [88 3c 1c 7c 35 d2] : 목적지의 데이터링크 계층 주소

Source MAC address [88 3c 1c 7c 35 d2] : 출발지의 데이터링크 계층 주소

Type [08 00] : 여기서는 타입을 의미하며 0x0800은 IP Protocol을 의미한다.

IP 패킷 구조

```
> Frame 8799: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: Mercury_7c:35:d2 (88:3c:1c:7c:35:d2), Dst: Cybertan_19:f6:8f (b0:fc:36:19:f6:8f)
v Internet Protocol Version 4, Src: 172.30.1.254, Dst: 172.30.1.53
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  v Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
      1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 60
    Identification: 0x92d5 (37589)
  v Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x8bbc [validation disabled]
    [Header checksum status: Unverified]
    Source: 172.30.1.254
    Destination: 172.30.1.53
> Internet Control Message Protocol
```

```
0000  b0 fc 36 19 f6 8f 88 3c  1c 7c 35 d2 08 00 45 c0   ··6····< ·|5···E·
0010  00 3c 92 d5 00 00 40 01  8b bc ac 1e 01 fe ac 1e   ·<····@· ········
0020  01 35 00 00 54 55 00 01  01 06 61 62 63 64 65 66   ·5··TU·· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

패킷 분석

Version [4]

패킷이 IPv4 인지 IPv6인지 나타낸다. 여기서는 IPv4를 사용한다.

Length [5]

IP 헤더의 길이를 4byte단위로 나타낸다. 여기서는 20 이다.

Type Of Service(TOS) [00]

음성, 영상, 문자와 같은 IP패킷의 우선순위를 정의한다.

TotalLength [00 3c]

헤더와 데이터를 포함한 전체 패킷의 길이를 바이트 단위로 나타낸다.

Identification [92 d5]

일련번호로서 패킷의 식별을 담당한다.

Flag [00 00]

패킷이 단편화 되었는지 판단할 수 있다. 단편화 되지 않았다.

TTL [80]

IP 패킷의 수명을 나타낸다 80만큼의 홉을 경유 할 수 있다.

Protocol [01]

상위 계층 프로토콜을 나타낸다. 여기서는 ICMP를 나타낸다.

Header Checksum [8b bc]

IP 패킷 헤더의 오류 발생 체크에 사용되는 비트.

Source Address [ac 1e 01 10]

송신측의 IP 주소를 나타낸다. 여기서는 172.30.1.254

Destination Address [a3 98 a1 01]

수신측의 IP 주소를 나타낸다. 여기서는 172.30.1.53

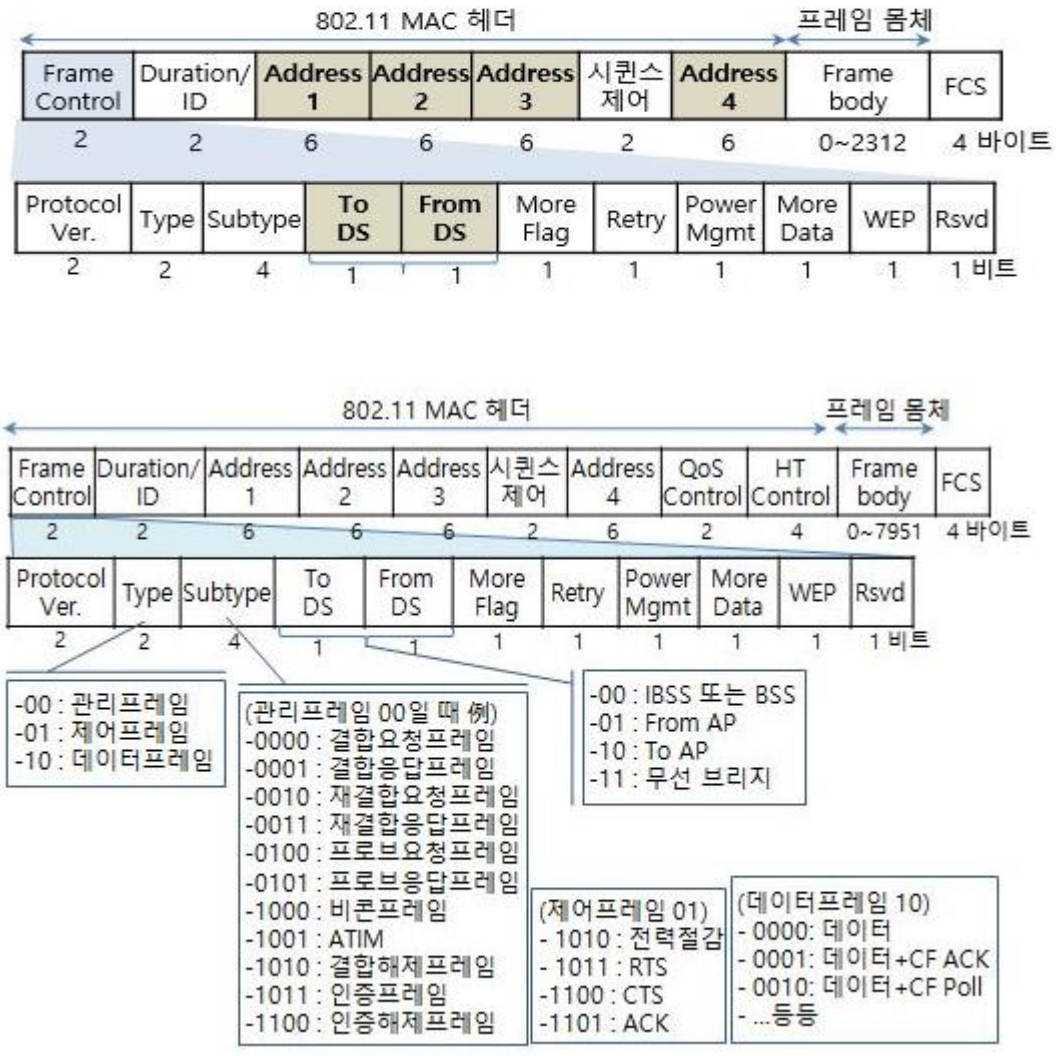
데이터통신 #1-3

wifi (802.11) 패킷 캡쳐 화면과 Layer2 Header 분석 by Network Monitor

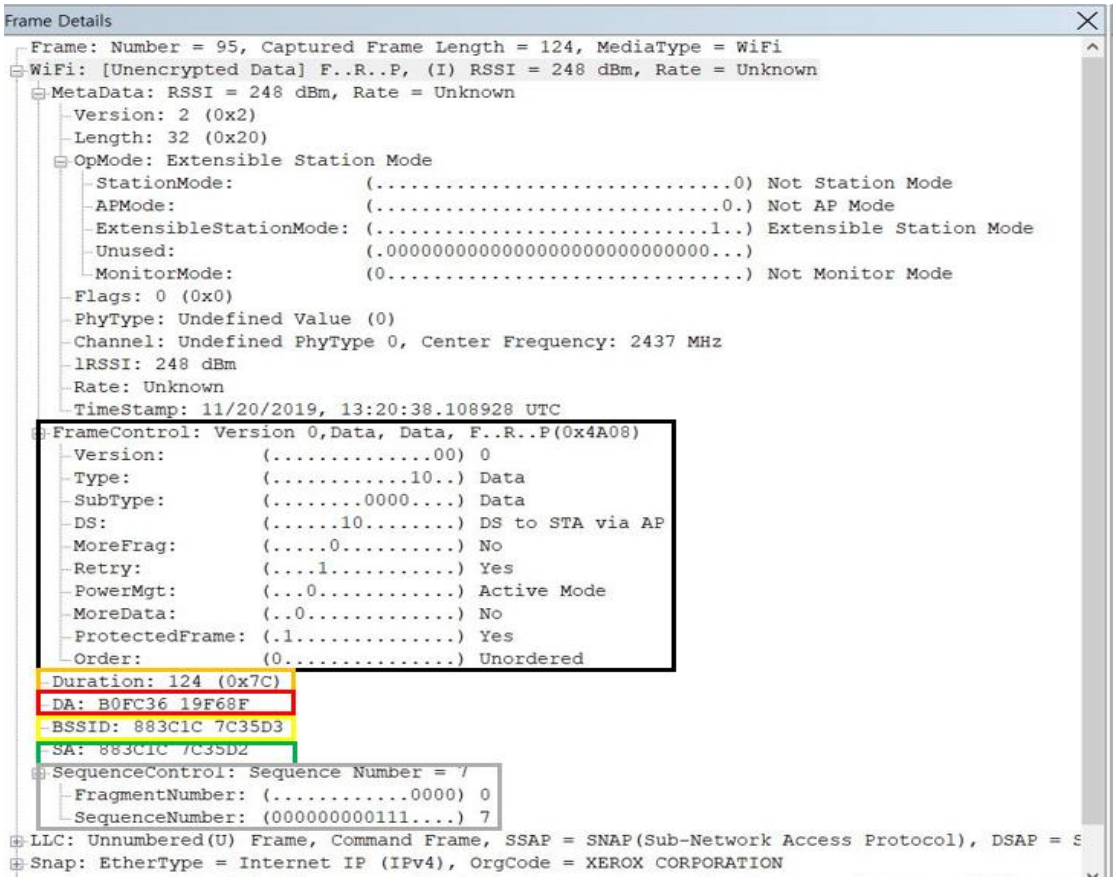
개요

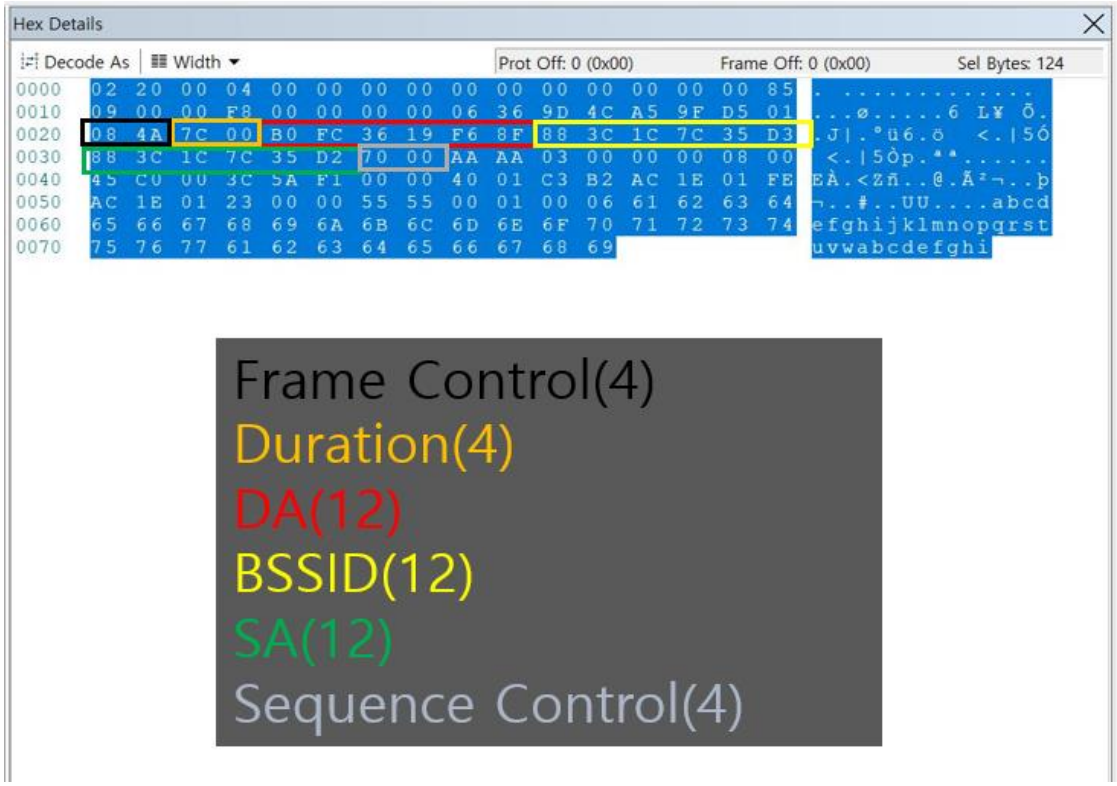
Layer2(Data Link Layer)는 크게 LLC 부계층과 MAC 부계층으로 이루어져 있다. 상부에 해당하는 LLC 부계층은 두 지점 간에 신뢰성 있는 패킷 링크로 전환을 담당하고 MAC 부계층은 신뢰성에 대한 보장없이 패킷 전송서비스를 제공한다. 하지만 MAC 계층은 상이한 장치(device)들의 프로토콜을 사용할 수 있게 하여 망의 토폴로지와 관계없이 통신이 가능하게 한다.

IEEE 802.11 MAC 프레임 구조 [정보통신기술용어해설 참조]



MAC 프레임 구조


Frame Details
Frame: Number = 95, Captured Frame Length = 124, MediaType = WiFi
WiFi: [Unencrypted Data] F..R..P, (I) RSSI = 248 dBm, Rate = Unknown
MetaData: RSSI = 248 dBm, Rate = Unknown
Version: 2 (0x2)
Length: 32 (0x20)
OpMode: Extensible Station Mode
StationMode: (.....0) Not Station Mode
APMode: (.....0.) Not AP Mode
ExtensibleStationMode: (.....1..) Extensible Station Mode
Unused: (.....)
MonitorMode: (0.....) Not Monitor Mode
Flags: 0 (0x0)
PhyType: Undefined Value (0)
Channel: Undefined PhyType 0, Center Frequency: 2437 MHz
lRSSI: 248 dBm
Rate: Unknown
TimeStamp: 11/20/2019, 13:20:38.108928 UTC
FrameControl: Version 0,Data, Data, F..R..P(0x4A08)
Version: (.....00) 0
Type: (.....10..) Data
SubType: (....0000....) Data
DS: (.....10.....) DS to STA via AP
MoreFrag: (.....0.....) No
Retry: (....1.....) Yes
PowerMgt: (...0.....) Active Mode
MoreData: (..0.....) No
ProtectedFrame: (.1.....) Yes
Order: (0.....) Unordered
Duration: 124 (0x7C)
DA: B0FC36 19F68F
BSSID: 883C1C 7C35D3
SA: 883C1C 7C35D2
SequenceControl: Sequence Number = 7
FragmentNumber: (.....0000) 0
SequenceNumber: (000000000111....) 7
LLC: Unnumbered(U) Frame, Command Frame, SSAP = SNAP(Sub-Network Access Protocol), DSAP =
Snap: EtherType = Internet IP (IPv4), OrgCode = XEROX CORPORATION


Hex Details
Decode As
Width
Prot Off: 0 (0x00)
Frame Off: 0 (0x00)
Sel Bytes: 124
0000 02 20 00 04 00 00 00 00 00 00 00 00 00 00 00 85
0010 09 00 00 F8 00 00 00 00 06 36 9D 4C A5 9F D5 01
0020 08 4A 7C 00 B0 FC 36 19 F6 8F 88 3C 1C 7C 35 D3
0030 88 3C 1C 7C 35 D2 70 00 AA AA 03 00 00 00 08 00
0040 45 C0 00 3C 5A F1 00 00 40 01 C3 B2 AC 1E 01 FE
0050 AC 1E 01 23 00 00 55 55 00 01 00 06 61 62 63 64
0060 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74
0070 75 76 77 61 62 63 64 65 66 67 68 69
Frame Control(4)
Duration(4)
DA(12)
BSSID(12)
SA(12)
Sequence Control(4)

MAC 프레임 필드 분석

Frame Control [08 4A]

802.11 MAC frame 관련 제어 정보(프레임 종류)를 담고 있다. 각 프레임의 맨 처음 시작부분 2바이트 필드로서 무슨 노드와 노드 간에 전송되는 프레임의 제어에 대한 정보를 담고 있다. 이에 속하는 정보들은 다음과 같다.

1. Version (00) – 2bit => 0

Protocol version 이다. 여기서는 00 으로 되어있고 현재 1가지 버전만 존재한다.

2. Type (10) – 2bit => Data frame

프레임의 유형을 나타낸다. 프레임별 종류는 다음과 같다.

00 = 관리프레임

01 = 제어프레임

10 – 데이터프레임

3. Subtype (0000) – 4bit

타입과 함께 각 프레임의 세부 종류를 나낸다. 일반적으로 10종이상의 subtype을 지닌다.

4. DS (10) - 2bit => DS to STA via AP

Distribution System으로 여러 AP를 연결하여 무선망을 확장시키는 무선LAN용 백본 네트워크를 의미한다. DS로 전송되는 프레임의 방향을 ToDS, 반대방향을 FromDS라고한다. 둘의 값을 비트로 표현한다.

5. More Frag (0) – 1bit => No

동일한 Mac Service Data Unit (layer 3~7의 실제 정보를 갖는 데이터) 에서 프래그먼트 되었는지에 대한 표시이다.

6. Retry (1) – 1bit => Yes

중복 수신을 방지하기위한 재전송 비트일 경우 세팅하는 값이다.

7. PowerMgt (0) – 1bit => Active Mode

전력절감에 대한 여부를 나타낸다. 0이면 활성화모드 1이면 절약모드이다.

8. MoreData (0) – 1bit => No

관리 및 데이터 프레임에서 사용하며 추가데이터 여부에 대한 상태이다.

9. ProtectedFrame (1) – 1bit => Yes

Frame body가 암호화 되어있는지에 대한 여부이다.

10. Order (0) – bit => Unordered

HT(High Throughput) filed 의 존재에 대한 여부이다.

Duration [7c]

보내진 프레임들을 무선 노드가 읽어서 목적지 어드레스가 자신의 MAC 어드레스와 다른 경우에 해당 프레임의 duration field를 읽어서 예약된 시간만큼의 대기를 하면 된다.

Address field 1,2,3,(4)

특수한 경우를 제외하고는 3가지 주소 필드를 사용한다. 현재 프레임에 사용되는 주소는 다음과 같다.

1. Destination Address (DA) [b0 fc 36 19 f6 8f]

2. Basic Service Set Identifier (BSSID) [88 8c 1c 7c 35 d3]

BSS를 식별하는 네트워크 ID

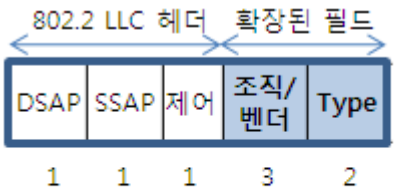
3. Source Address (SA) [88 3c 1c 7c 35 d2]

Sequence Control [70 00]

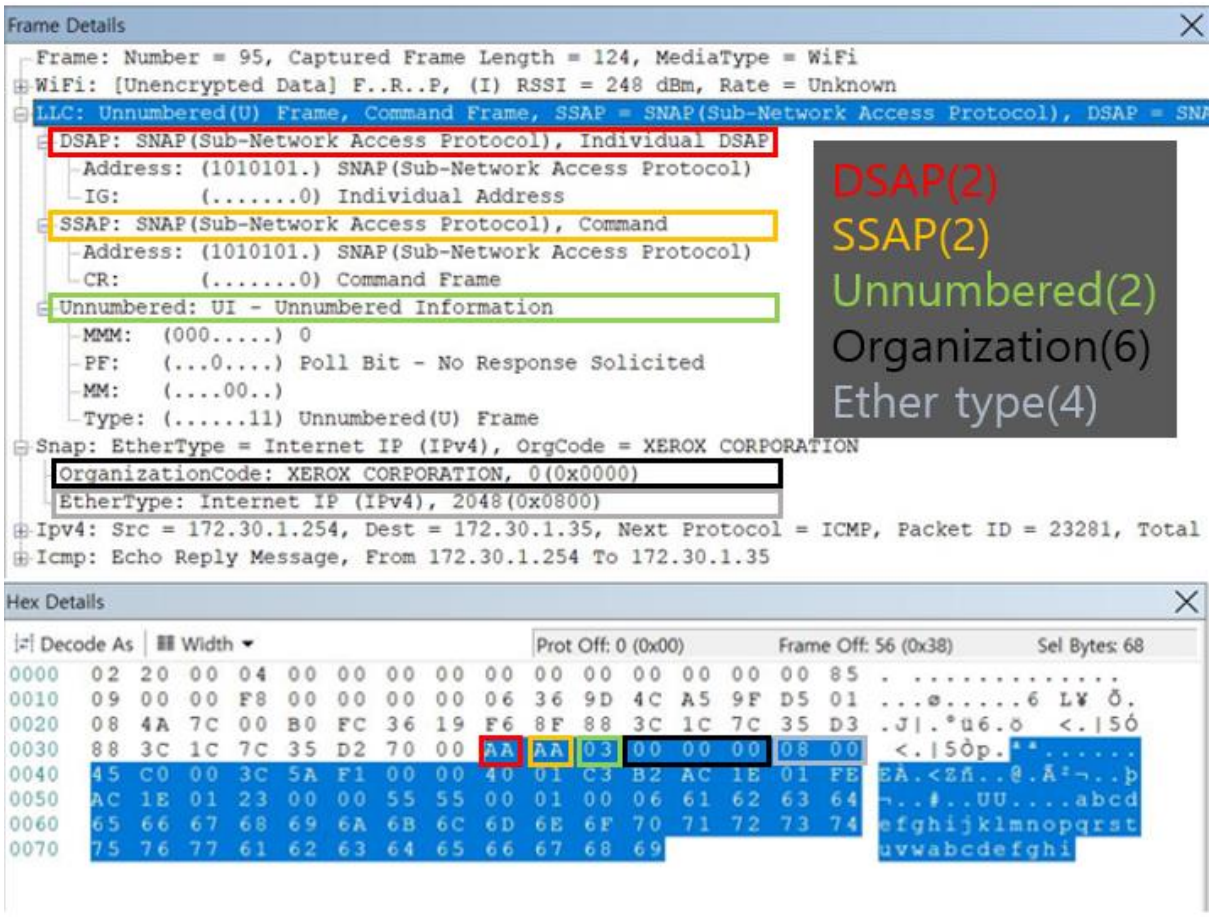
순서가 역전되거나 중복 패킷의 발생을 방지하기 위하여 존재하는 필드이다. 802.11 에서는 매 MSDU 마다 순서번호를 붙이는 Sequence number(12bit)와 조각된 프레임마다 0부터 순서대로 붙이는 Fragment number(4bit)로 구분되어 있다.

LLC 및 SNAP 프레임 분석

802.11 표준은 아니지만 조각화 및 암호화를 포함하여 MAC 계층 용도의 데이터처럼 취급되는데 실제로는 두 개의 헤더가 연결되어 있다. 802.2 LLC 헤더와 0 OUI 및 2 바이트 이더넷을 나타내는 5 바이트 SNAP 헤더 프로토콜 유형 필드가 결합되었다.



- 조직/벤더 : OUI
- Type 필드 : Ethertype과 같음



LLC & SNAP: 확장 LLC 및 SNAP의 경우 거의 고정된 값을 가진다.

Destination Service Access Point (DSAP) [aa] : aa가 SNAP 의미

Source Service Access Point (SSAP) [aa] : aa가 SNAP 의미

Unnumbered [03] : 제어필드

Organization Code [00 00 00]

조직/벤더를 나타내는데 제조사가 고유 번호를 부여하지만 실제로는 거의 null 값이 세팅된다.

Type [08 00]

Ethertype과 동일한 설정이며 여기서는 IP 를 가르킨다.

wireshark로 잡았을 때의 패킷과, network monitor로 잡았을 때의 패킷이 왜 다른지

모니터 모드는 무선 네트워크 인터페이스 컨트롤러(NIC)가 있는 장치가 무선 네트워크에서 수신 된 모든 트래픽을 모니터링 할 수 있도록 한다. 모니터 모드를 사용하면 액세스 포인트와 연결하거나 연결하지 않고도 패킷을 캡처 할 수 있다. 또한 무선 네트워크에서만 작동하지만 무차별 모드는 유선 및 무선 네트워크 모두에 적용 할 수 있다. 간단히 말해, 무선 카드는 필터링없이 수신 된 패킷을 "모니터링"할 수 있으니, 모니터 모드는 본질적으로 무선에 대한 "무차별 모드(promiscuous mode)"이다. 일부 무선 드라이버를 사용하는 경우에서 802.11 프레임을 보낼 수 있다. 그래서 여러 무선 도구를 사용하려면 어댑터가 작동하기 위해 모니터 모드에 있어야한다.

그런데 wireshark에서는 기본적으로 다른시스템간에 네트워크 트래픽을 캡쳐할때에는 802.11의 관리 또는 제어 패킷은 무시하고 데이터에만 관심을 가진다. 와이어샤크에서는 어댑터가 모든 트래픽을 캡처할 수 있는 모니터모드와 같은 특별한 설정을 해주지 않았으므로 제대로된 패킷을 분석할 수 없었던 것이다. 이와 같은 설정을 하지 않은 경우에 WLAN을 캡처하면 데이터 패킷만 캡처할 수 있기 때문에 관리 제어 패킷은 가짜 패킷이 되는 것이다.

데이터통신 #1-4

ppp configuration 과정을 pppd connection debug 기반으로 설명

개요

Point-to-Point (PPP)는 전화 접속 모뎀, DSL 연결 및 기타 여러 유형의 지점 간 링크를 통해 인터넷 링크를 설정하는 데 사용되는 프로토콜 이다 pppd deamon은 커널 PPP 드라이버와 협력하여 다른 시스템(피어)과 PPP 링크를 설정하고 유지하며 링크의 각 끝에 대해 IP(인터넷 프로토콜) 주소를 협상(negotiate)한다. pppd는 피어를 인증하거나 인증 정보를 제공한다.

이번 #1-4 에서는 이러한 pppd를 이용하여 두개의 리눅스 가상환경에서 서로 링크를 연결하여 패킷을 전달해 볼 것이다.

pppd 실행

다음과 같은 옵션을 주어서 pppd 실행을 한다.

'pppd -detach crtscts lock debug record /root/test02 noccp 10.02.2.15:10.0.2.15 /dev/ttyS0 38400'

Crtscts : pppd가 RS-232 인터페이스의 RTS 및 CTS 신호를 사용하여 하드웨어 흐름 제어를 사용하도록 직렬 포트를 설정해야 함을 지정한다.

Debug : 송수신되는 모든 제어패킷의 내용을 판독 가능한 형태로 기록하기 위함.

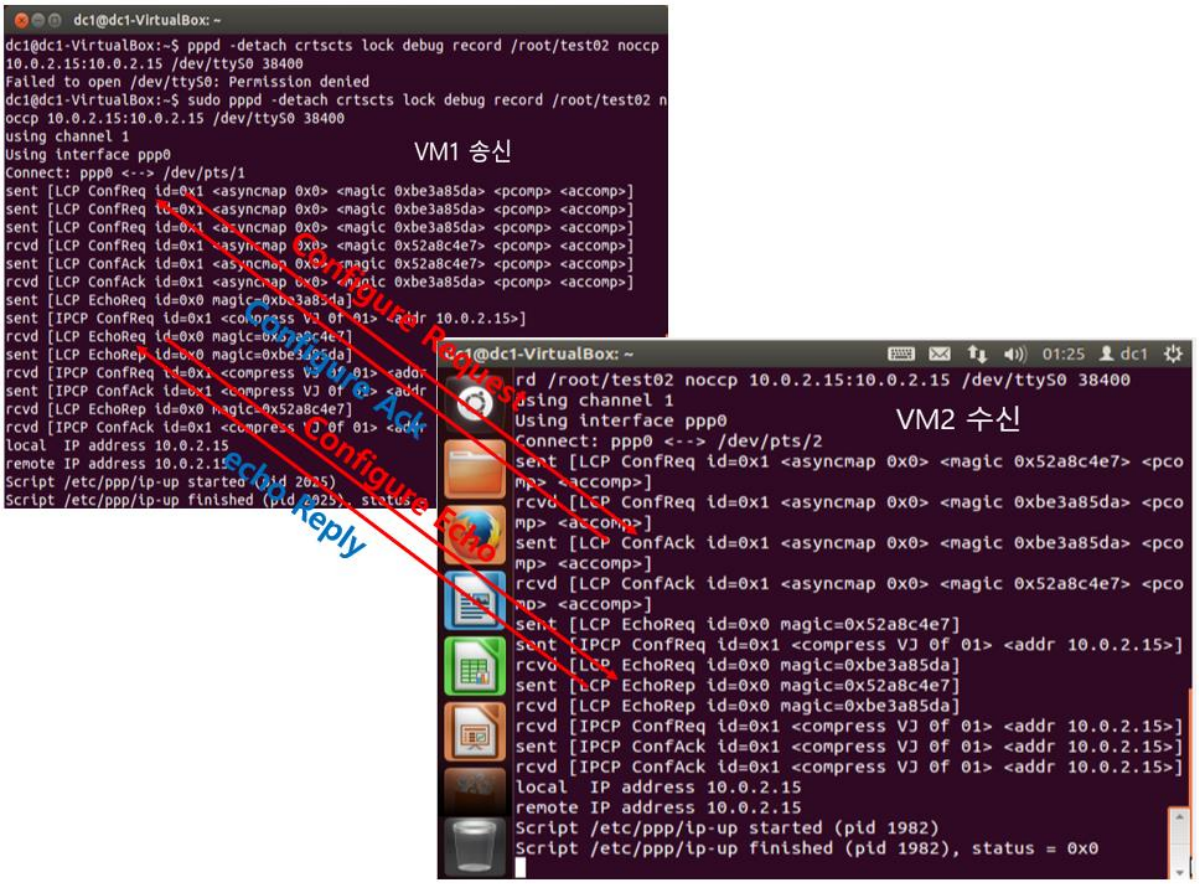
Record <Path> : dump 파일을 저장할 위치이다. /root/test02 경로에 저장하였다.

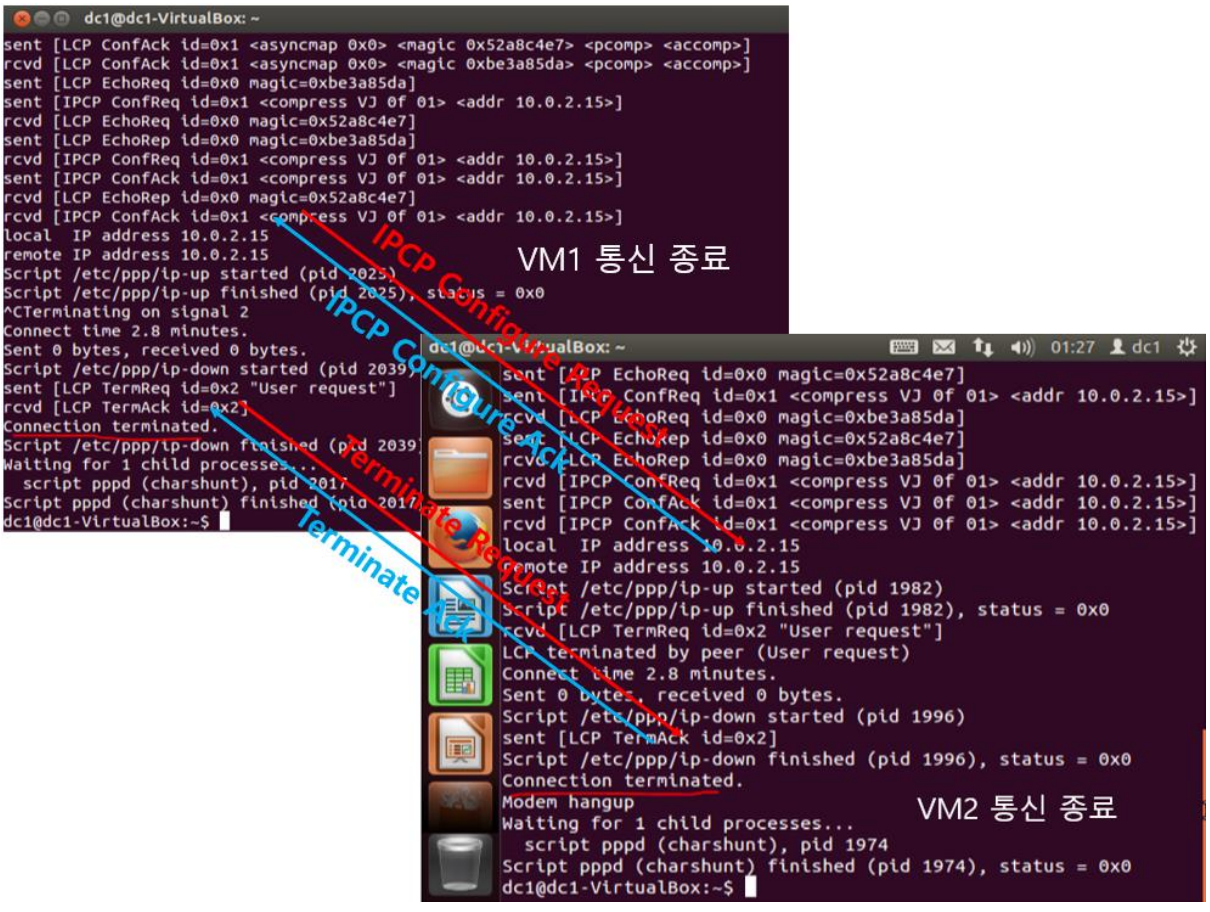
<Local IP> : <Remote IP > : ppp 를 위한 두 가상머신의 ip 주소이다. Ifconfig 를 통하여 확인하였다.

Ttypname : 피어와 통신하려면 ttyname이라는 직렬 포트를 사용한다.

Speed : tty 장치의 전송 속도이다.

더 자세한 옵션은 pppd(8) 문서에 상세히 나와있다.


dc1@dc1-VirtualBox: ~
dc1@dc1-VirtualBox:~$ pppd -detach crtscts lock debug record /root/test02 noccp
10.0.2.15:10.0.2.15 /dev/ttyS0 38400
Failed to open /dev/ttyS0: Permission denied
dc1@dc1-VirtualBox:~$ sudo pppd -detach crtscts lock debug record /root/test02 n
occp 10.0.2.15:10.0.2.15 /dev/ttyS0 38400
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/pts/1
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xbe3a85da> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xbe3a85da> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xbe3a85da> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x52a8c4e7> <pcomp> <accomp>]
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x52a8c4e7> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xbe3a85da> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0xbe3a85da]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
rcvd [LCP EchoReq id=0x0 magic=0x52a8c4e7]
sent [LCP EchoRep id=0x0 magic=0xbe3a85da]
rcvd [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr
sent [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr
rcvd [LCP EchoRep id=0x0 magic=0x52a8c4e7]
rcvd [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr
local IP address 10.0.2.15
remote IP address 10.0.2.15
Script /etc/ppp/ip-up started (pid 2025)
Script /etc/ppp/ip-up finished (pid 2025), status
VM1 송신
Configure Request
Configure Ack
Configure Echo
Echo Reply
dc1@dc1-VirtualBox: ~
01:25 dc1
rd /root/test02 noccp 10.0.2.15:10.0.2.15 /dev/ttyS0 38400
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/pts/2
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x52a8c4e7> <pco
mp> <accomp>]
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xbe3a85da> <pco
mp> <accomp>]
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xbe3a85da> <pco
mp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x52a8c4e7> <pco
mp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0x52a8c4e7]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
rcvd [LCP EchoReq id=0x0 magic=0xbe3a85da]
sent [LCP EchoRep id=0x0 magic=0x52a8c4e7]
rcvd [LCP EchoRep id=0x0 magic=0xbe3a85da]
rcvd [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
sent [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
rcvd [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
local IP address 10.0.2.15
remote IP address 10.0.2.15
Script /etc/ppp/ip-up started (pid 1982)
Script /etc/ppp/ip-up finished (pid 1982), status = 0x0
VM2 수신


dc1@dc1-VirtualBox: ~
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x52a8c4e7> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xbe3a85da> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0xbe3a85da]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
rcvd [LCP EchoReq id=0x0 magic=0x52a8c4e7]
sent [LCP EchoRep id=0x0 magic=0xbe3a85da]
rcvd [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
sent [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
rcvd [LCP EchoRep id=0x0 magic=0x52a8c4e7]
rcvd [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
local IP address 10.0.2.15
remote IP address 10.0.2.15
Script /etc/ppp/ip-up started (pid 2025)
Script /etc/ppp/ip-up finished (pid 2025), status = 0x0
^CTerminating on signal 2
Connect time 2.8 minutes.
Sent 0 bytes, received 0 bytes.
Script /etc/ppp/ip-down started (pid 2039)
sent [LCP TermReq id=0x2 "User request"]
rcvd [LCP TermAck id=0x2]
Connection terminated.
Script /etc/ppp/ip-down finished (pid 2039)
Waiting for 1 child processes...
script pppd (charshunt), pid 2017
Script pppd (charshunt) finished (pid 2017)
dc1@dc1-VirtualBox:~$
VM1 통신 종료
IPCP Configure Request
IPCP Configure Ack
Terminate Request
Terminate Ack
dc1@dc1-VirtualBox: ~
01:27 dc1
sent [LCP EchoReq id=0x0 magic=0x52a8c4e7]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
rcvd [LCP EchoReq id=0x0 magic=0xbe3a85da]
sent [LCP EchoRep id=0x0 magic=0x52a8c4e7]
rcvd [LCP EchoRep id=0x0 magic=0xbe3a85da]
rcvd [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
sent [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
rcvd [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr 10.0.2.15>]
local IP address 10.0.2.15
remote IP address 10.0.2.15
Script /etc/ppp/ip-up started (pid 1982)
Script /etc/ppp/ip-up finished (pid 1982), status = 0x0
rcvd [LCP TermReq id=0x2 "User request"]
LCP terminated by peer (User request)
Connect time 2.8 minutes.
Sent 0 bytes, received 0 bytes.
Script /etc/ppp/ip-down started (pid 1996)
sent [LCP TermAck id=0x2]
Script /etc/ppp/ip-down finished (pid 1996), status = 0x0
Connection terminated.
Modem hangup
Waiting for 1 child processes...
script pppd (charshunt), pid 1974
Script pppd (charshunt) finished (pid 1974), status = 0x0
dc1@dc1-VirtualBox:~$
VM2 통신 종료

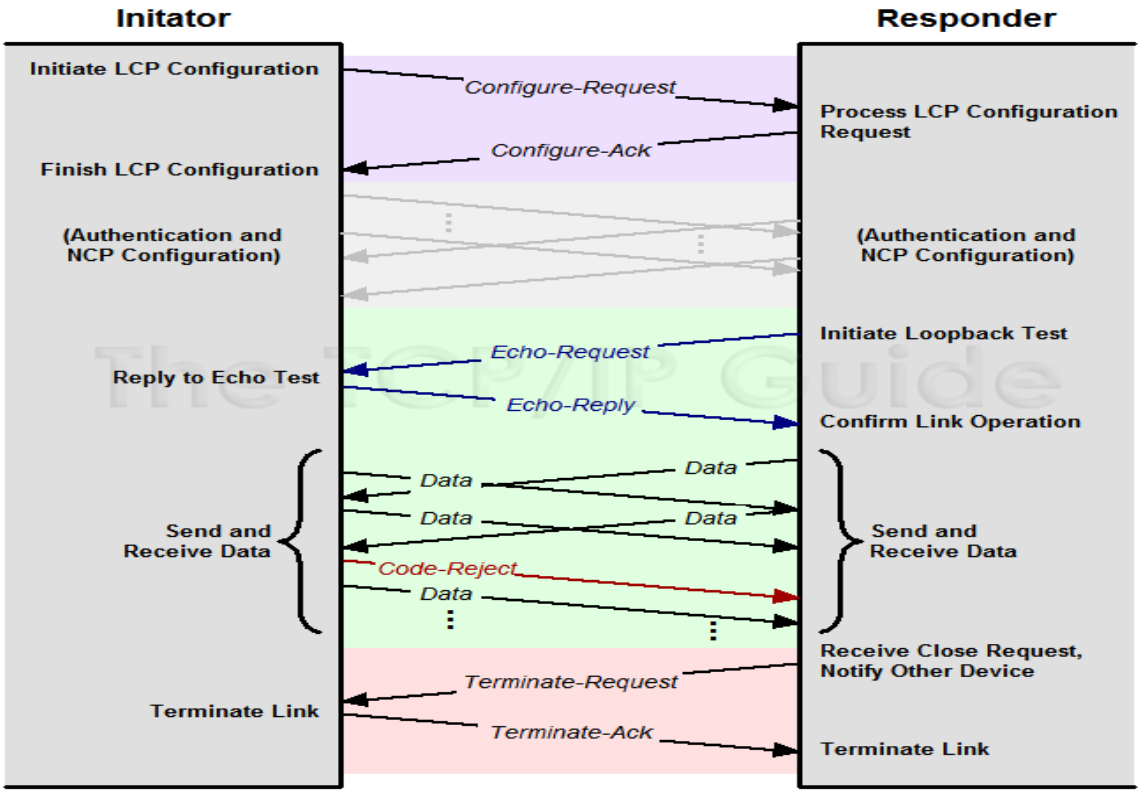
우선 처음 LCP(Link Control Protocol)는 연결을 PPP 연결의 데이터 링크 계층을 동적으로 구성한다. 다음과 같은 협상 이후 교환에 성공하면 Configure-Ack 패킷을 인지하여 LCP Opened 상태가 되어 패킷을 송수신 할 수 있다. 다음은 인증 단계를 거치게 된다.

암호 인증 프로토콜은 PAP 또는 CHAP 가 담당한다. 기본적으로 인증을 구성하지 않으면 지점 간 프로토콜 구성이 다소 간단하다. PPP 인증은 선택 사항이며 인증없이 링크를 설정할 수 있기 때문이다. 실제로 이전의 HDLC 구성과 비교할 때 여기에서 유일한 변경 사항 은 인터페이스 구성 모드에서 encapsulation ppp 명령 정도이다.

IPCP (또는 다른 네트워크 제어 프로토콜)를 시작하기 전에 인증이 만족스럽게 완료 되어야 한다. 피어가 인증을 받아야 하는데 인증에 실패하면 pppd는 LCP를 닫아 링크를 종료한다. IPCP가 원격 호스트에 허용되지 않는 IP 주소를 협상하면 IPCP가 닫힙니다. IP 패킷은 IPCP가 열려 있을 동안만 보내거나 받을 수 있다. 인증이 이루어 지면 네트워크 계층으로 넘어갈 수 있다. IPCP (Internet Protocol Control Protocol) 는 PPP 프레임 내부에 캡슐화 되어있는데, 기능은 PC에 IP 주소를 할당하는데 사용되어 압축 사용여부를 결정하고 패킷 구성을 설정한다.

마지막으로 어느 한쪽에서 종료를 원할 경우에 Terminate-Request 를 보내면 Ack 신호와 함께 양쪽에서 종료가 된다. 대략적인 흐름은 다음 그림과 같다.

.출처 [PPP-certification]



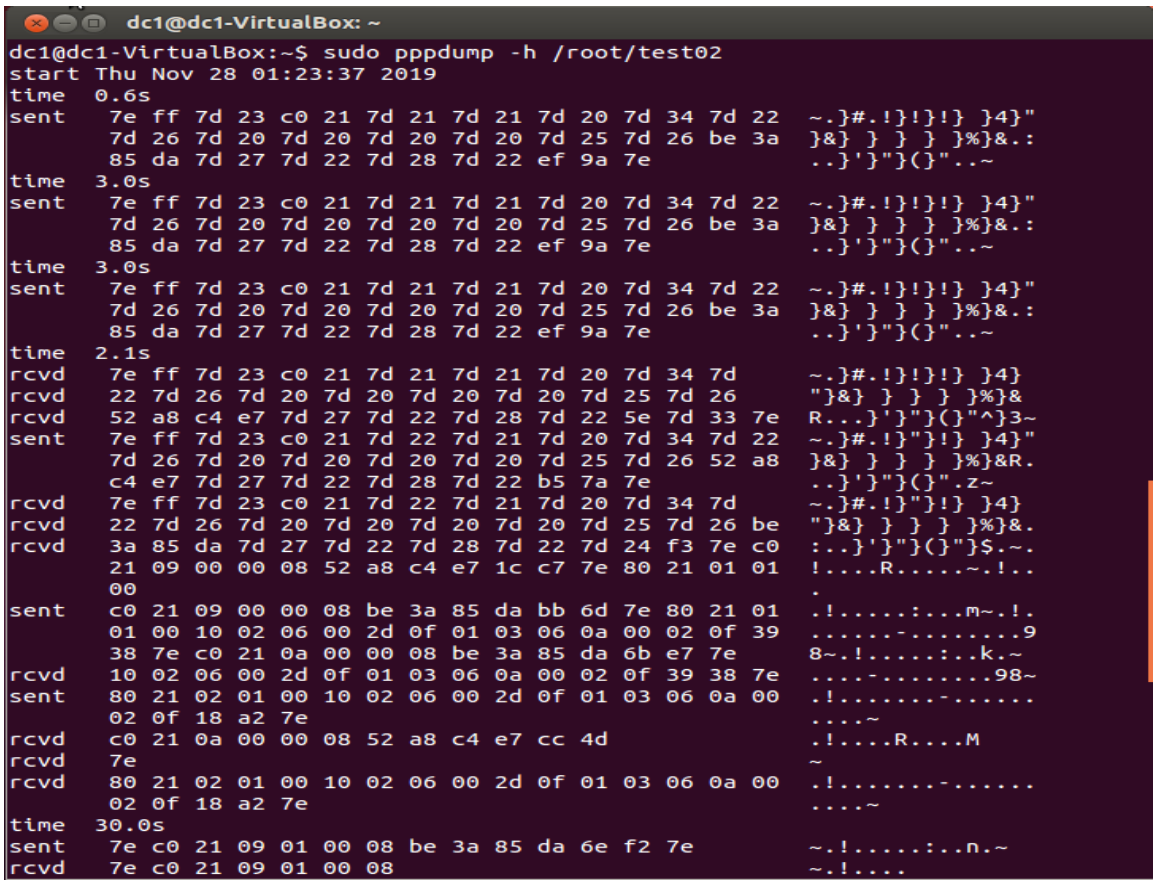
데이터통신 #1-5

ppp 실행 후, pppdump 패킷 분석

개요

#1-4 에서는 두개의 리눅스 가상머신 환경하에서 pppd 명령어를 실행시켰다. 지정해둔 경로에 있는 pppdump 파일을 확인하여 LCP, IPCP, IP Frame을 중심으로 패킷을 분석해보자.

전체 패킷 캡쳐



```
dc1@dc1-VirtualBox: ~
rcvd    7e c0 21 09 01 00 08                                    ~.!....
rcvd    52 a8 c4 e7 c9 58 7e c0 21 0a 01 00 08 52 a8 c4  R....X~.!....R..
        e7 19                                                   ..
sent    c0 21 0a 01 00 08 be 3a 85 da be 78 7e            .!.....:...x~
rcvd    d2 7e                                                   .~
time   30.1s
rcvd    7e c0 21 09 02 00 08 52 a8 c4 e7 a7 f0            ~.!....R.....
rcvd    7e                                                      ~
sent    7e c0 21 09 02 00 08 be 3a 85 da 00 5a 7e c0 21  ~.!.....:...Z~.!
        0a 02 00 08 be 3a 85 da d0 d0 7e                  .....:....~
rcvd    c0 21 0a 02 00 08 52 a8 c4 e7 77 7a 7e            .!....R...wz~
time   30.0s
rcvd    7e c0 21 09 03 00 08 52 a8 c4 e7 72               ~.!....R...r
rcvd    6f 7e                                                   o~
sent    7e c0 21 09 03 00 08 be 3a 85 da d5 c5 7e c0 21  ~.!.....:....~.!
        0a 03 00 08 be 3a 85 da 05 4f 7e                  .....:...O~
rcvd    c0 21 0a 03 00 08 52 a8 c4 e7 a2 e5 7e            .!....R.....~
time   30.0s
rcvd    7e c0 21 09 04 00 08 52 a8 c4 e7 6a a8 7e         ~.!....R...j.~
sent    7e c0 21 09 04 00 08 be 3a 85 da cd 02 7e c0 21  ~.!.....:....~.!
        0a 04 00 08 be 3a 85 da 1d 88 7e                  .....:....~
rcvd    c0 21 0a 04 00 08 52 a8 c4 e7 ba 22 7e            .!....R...."~
time   30.0s
sent    7e c0 21 09 05 00 08 be 3a 85 da 18 9d 7e         ~.!.....:....~
rcvd    7e c0 21 09 05 00 08 52 a8 c4 e7 bf 37            ~.!....R....7
rcvd    7e c0 21 0a 05 00 08 52 a8 c4 e7 6f bd            ~.!....R...o.
sent    c0 21 0a 05 00 08 be 3a 85 da c8 17 7e            .!.....:....~
time   0.1s
rcvd    7e                                                      ~
time   14.2s
sent    7e ff 7d 23 c0 21 7d 25 7d 22 7d 20 7d 30 55 73  ~.}#.!}%}"} }0Us
        65 72 20 72 65 71 75 65 73 74 53 33 7e           er requestS3~
rcvd    7e ff 7d 23 c0 21 7d 26 7d                        ~.}#.!}&}
rcvd    22 7d                                                   "}
time   0.1s
rcvd    20 7d 24 94 7d 2d 7e                                     }$.}-~
time   1.0s
end send
dc1@dc1-VirtualBox:~$
```

PPP 일반 프레임


```
dc1@dc1-VirtualBox:~$ sudo pppdump -h /root/test02
start Thu Nov 28 01:23:37 2019                          Basic PPP
time   0.6s
sent    7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 34 7d 22  ~.}#.!}!}!} }4}"
        7d 26 7d 20 7d 20 7d 20 7d 20 7d 25 7d 26 be 3a  }&} } } } }%}&.:
        85 da 7d 27 7d 22 7d 28 7d 22 ef 9a 7e           ..}'}"}(}"..~
time   3.0s
sent    7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 34 7d 22  ~.}#.!}!}!} }4}"
        7d 26 7d 20 7d 20 7d 20 7d 20 7d 25 7d 26 be 3a  }&} } } } }%}&.:
        85 da 7d 27 7d 22 7d 28 7d 22 ef 9a 7e           ..}'}"}(}"..~
time   3.0s
sent    7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 34 7d 22  ~.}#.!}!}!} }4}"
        7d 26 7d 20 7d 20 7d 20 7d 20 7d 25 7d 26 be 3a  }&} } } } }%}&.:
        85 da 7d 27 7d 22 7d 28 7d 22 ef 9a 7e           ..}'}"}(}"..~
time   2.1s
rcvd    7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 34 7d     ~.}#.!}!}!} }4}
rcvd    22 7d 26 7d 20 7d 20 7d 20 7d 20 7d 25 7d 26     "}&} } } } }%}&
rcvd    52 a8 c4 e7 7d 27 7d 22 7d 28 7d 22 5e 7d 33 7e  R...}'}"}(}"^}3~
```

Flag	Address	Control	Protocol	Data (Padding)	FCS	flag
7E	FF	7D 23	C0 21		EF 9A	7E

Flag [7e]

항상 01111110 (0x7E,126) 로 설정 되어 프레임의 시작과 끝을 나타낸다.

Address [ff]

HDLC에서 프레임의 목적지 주소역할 이였다. 그러나 PPP에서 우리는 두 장치 사이의 직접적인 연결을 다루고 있기 때문에 이 필드는 실제적인 의미가 없다. 따라서 항상 "11111111"(0xFF 또는 255)로 설정된다. (모든 스테이션)

Control [7d 23]

HDLD에서는 다양한 제어목적으로 사용되지만, PPP 에서는 "000000011" (Unnumbered information) 로 설정된다.

Protocol [c0 21]

HDLC에서는 없었던 필드로 프레임의 정보 필드에 캡슐화된 데이터그램의 프로토콜을 확인한다. 프로토콜 필드에 대한 자세한 내용은 IANA의 공식 PPP numbers에서 확인할 수 있다. 자주 사용되는 프로토콜은 네트워크 제어용 데이터 (IPCP : 0x 0821), 링크 제어용 데이터 (LCP : 0x C021), . IP 데이터그램(0x 0021) 등이 있다.

Table 34: Common Protocols Carried in PPP Frames and *Protocol* Field Values

Protocol Type	*Protocol* Field Value (hex)	Protocol
Encapsulated Network Layer Datagrams	0021	Internet Protocol version 4 (IPv4)
	0023	OSI Network Layer
	0029	Appletalk
	002B	Novell Internetworking Packet Exchange (IPX)
	003D	PPP Multilink Protocol (MP) Fragment
	003F	NetBIOS Frames (NBF/NetBEUI)
	004D	IBM Systems Network Architecture (SNA)
	0053	Encrypted Data (using ECP and a PPP encryption algorithm)
	0055	Individual-Link Encrypted Data under PPP Multilink
	0057	Internet Protocol version 6 (IPv6)
	00FB	Individual-Link Compressed Data under PPP Multilink
	00FD	Compressed Data (using CCP and a PPP compression algorithm)
Low-Volume Encapsulated Protocols	4003	CDPD Mobile Network Registration Protocol
	4025	Fibre Channel
Network Control Protocol (NCP) Control Frames	8021	PPP Internet Protocol Control Protocol
	8023	PPP OSI Network Layer Control Protocol
	8029	PPP Appletalk Control Protocol
	802B	PPP IPX Control Protocol
	803F	PPP NetBIOS Frames Control Protocol
	804D	PPP SNA Control Protocol
	8057	PPP IPv6 Control Protocol
LCP and Other Control Frames	C021	PPP Link Control Protocol (LCP)
	C023	PPP Password Authentication Protocol (PAP)
	C025	PPP Link Quality Report (LQR)
	C02B	PPP Bandwidth Allocation Control Protocol (BACP)
	C02D	PPP Bandwidth Allocation Protocol (BAP)
	C223	PPP Challenge Handshake Authentication Protocol (CHAP)

http://www.tcpipguide.com/

Data & Padding

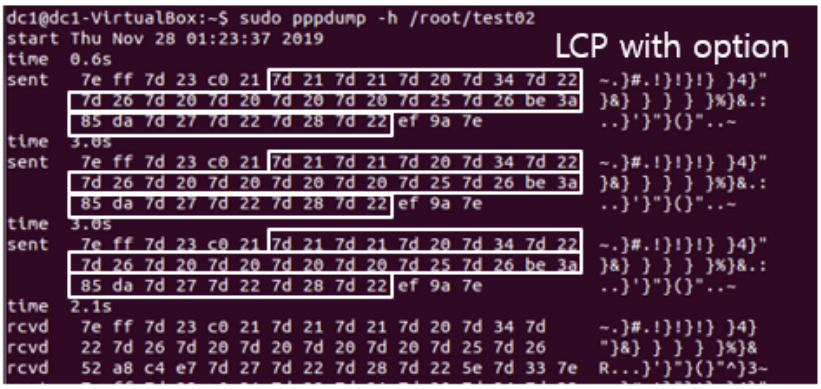
실제로 보내지는 데이터이다. 어떤 경우에는 PPP 프레임의 크기를 늘리기(패딩) 위해 더미 바이트를 추가할 수 있다.

FCS [ef 9a]

전송 오류에 대한 기본적인 보호를 제공하기 위해 프레임 위에 계산된 체크섬이다. FCS는 주소, 제어, 프로토콜, 정보 및 패딩 필드를 통해 계산된다. 이더넷에 사용되는 것과 같은 다른 layer2 프로토콜 오류 보호 체계에 사용되는 것과 유사한 CRC 코드다. 크기는 16비트 또는 32비트일 수 있다(기본값은 16비트).

LCP

PPP 링크의 회선관리용 프로토콜 이다. 포인트간 데이터 링크를 제어하는 역할을 한다. LCP는 PPP 데이타 링크를 개설,유지,종료하고, 시험하여 직렬 회선 제어 관리한다. 또한 링크의 정상 동작 및 고장을 확인 하고 기타 옵션을 통하여 인증, 제한, 매직넘버 옵션 등을 줄 수 있다.




```
dc1@dc1-VirtualBox:~$ sudo pppdump -h /root/test02
start Thu Nov 28 01:23:37 2019
time  0.6s
sent   7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 34 7d 22  ~.}#.!}!}!} }4}"
       7d 26 7d 20 7d 20 7d 20 7d 20 7d 25 7d 26 be 3a  }&} } } } }%}&.:
       85 da 7d 27 7d 22 7d 28 7d 22 ef 9a 7e           ..}'}"}(}"..~
time  3.0s
sent   7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 34 7d 22  ~.}#.!}!}!} }4}"
       7d 26 7d 20 7d 20 7d 20 7d 20 7d 25 7d 26 be 3a  }&} } } } }%}&.:
       85 da 7d 27 7d 22 7d 28 7d 22 ef 9a 7e           ..}'}"}(}"..~
time  3.0s
sent   7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 34 7d 22  ~.}#.!}!}!} }4}"
       7d 26 7d 20 7d 20 7d 20 7d 20 7d 25 7d 26 be 3a  }&} } } } }%}&.:
       85 da 7d 27 7d 22 7d 28 7d 22 ef 9a 7e           ..}'}"}(}"..~
time  2.1s
rcvd   7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 34 7d     ~.}#.!}!}!} }4}
rcvd   22 7d 26 7d 20 7d 20 7d 20 7d 20 7d 25 7d 26     "}&} } } } }%}&
rcvd   52 a8 c4 e7 7d 27 7d 22 7d 28 7d 22 5e 7d 33 7e  R...}'}"}(}"^}3~
```

Code	ID	length	비고
7D 21	7D 21	7D 20 7D 34	Config-Req

Type	Length	Option Data	비고
7D 22	7D 27	7D 20 7D 20 7D 20 7D 20	ACCM

Type	Length	Magic Number	비고
7D 25	7D 26	BE 3A 85 DA	Magic Data

Type	Length	Type	Length	비고
7D 27	7D 22	7D 28	7D 22	Protocol Field Compression

Code [7d 21] => Config. Req

제어 프레임에 있는 제어 메시지 유형을 나타내는 단일 바이트 값이다. 특정 PPP 표준에서는 그 대신에 "format"이라고 부르기도 한다. 코드의 숫자에 따라 어떤 상태인지 d라 수 있다. 각 숫자는 포맷을 의미하는데 다음과 같다.

Code Value	Control Message	Meaning
1	Configure Request	연결을 설정할 때 사용
2	Configure Ack	Config req를 받아 해석 하면 보냄
3	Configure NACK	Config req를 받아 해석 못하면 보냄
4	Configure Reject	Config req를 받아 옵션 적용 못하면 보냄
5	Terminate Ack	연결 종료할 때 사용
6	Code Reject	LCP 패킷 코드 에러
7	Protocol Reject	프로토콜 필드 값 에러
8	Echo Request	연결 테스트 요청
9	Echo Reply	연결 테스트 응답
11	Discard Request	성능 측정을 위하여 상대방 프레임을 폐기
12	Identification	기기 정보를 보낼 때 사용
13	Time Remaining	연결이후 얼마나 시간이 지났는지 나타냄
14	Reset Request	초기화를 위한 제어
15	Reset NACK	초기화 응답을 받아 적용 하지 못하면 보냄

Identifier(ID) [7d 21]

이 필드는 요청을 응답과 일치시키는 데 사용되는 레이블 필드다. 요청이 전송되면 새 식별자가 생성되고 회신이 생성되면 회신을 프롬프트한 요청의 Identifier 필드의 값이 회신의 Identifier 필드에 사용된다.

Length [7d 20 7d 34]

데이터 필드의 길이가 가변적이므로 제어 프레임의 길이를 이곳에 지정한다. 길이 필드는 바이트로 지정되며 코드, 식별자, 길이 및 데이터 필드를 포함한 제어 프레임의 모든 필드를 포함한다.

LCP Option

각 옵션들은 첫 비트에 옵션의 타입과 길이를 나타낸다. 사용되는 옵션들은 다음과 같다.

Code	Option Type	Meaning
0	Reserved	예약됨
1	Max. Rec. Unit	뒤의 패킷이 매우 크거나 작을 때
2	ACCM	비동기 동기 변환 시
3	Auth. Protocol	특정 인증 프로토콜을 협상할 때
4	Quality Protocol	링크 품질을 모니터링 할 때
5	Magic Number	상대 측 기기 loop back 모드일 때 되돌아 오는 프레임이 누구의 프레임인지 확인할 때
7	Prot. Field Compression	프로토콜의 값을 압축할 때
8	Address and Ctrl. Field Compression	주소와 제어필드의 값을 압축 할 때
F	Compound Frame	

IPCP option

IP를 나타내는 IPCP 옵션은 다음과 같다. 현재 1번과 4번 값은 거의 사용되지 않는다.

Code	Option Type	Meaning
1	IP Addresses	
2	IP Compression Protocol	압축 프로토콜의 방법에 대하여 협상 할 때
3	IP Address	특정 종단점의 IP 주소를 협상 할 때
4	Mobile IP	

마무리

PPP 의 send, recv 패킷들은 위와 같이 PPP 의 기본 프레임을 확인하고 프로토콜과 옵션들의 타입에 맞게 데이터를 읽을 수 있다. #1-4에 설명했듯이 가상환경의 양 종단에서 요청 및 응답 신호와 함께 LCP와 IPCP 패킷이 서로 오가는 것을 확인 할 수 있다.