Python을 이용한 Packet analyzer

7.15 2019

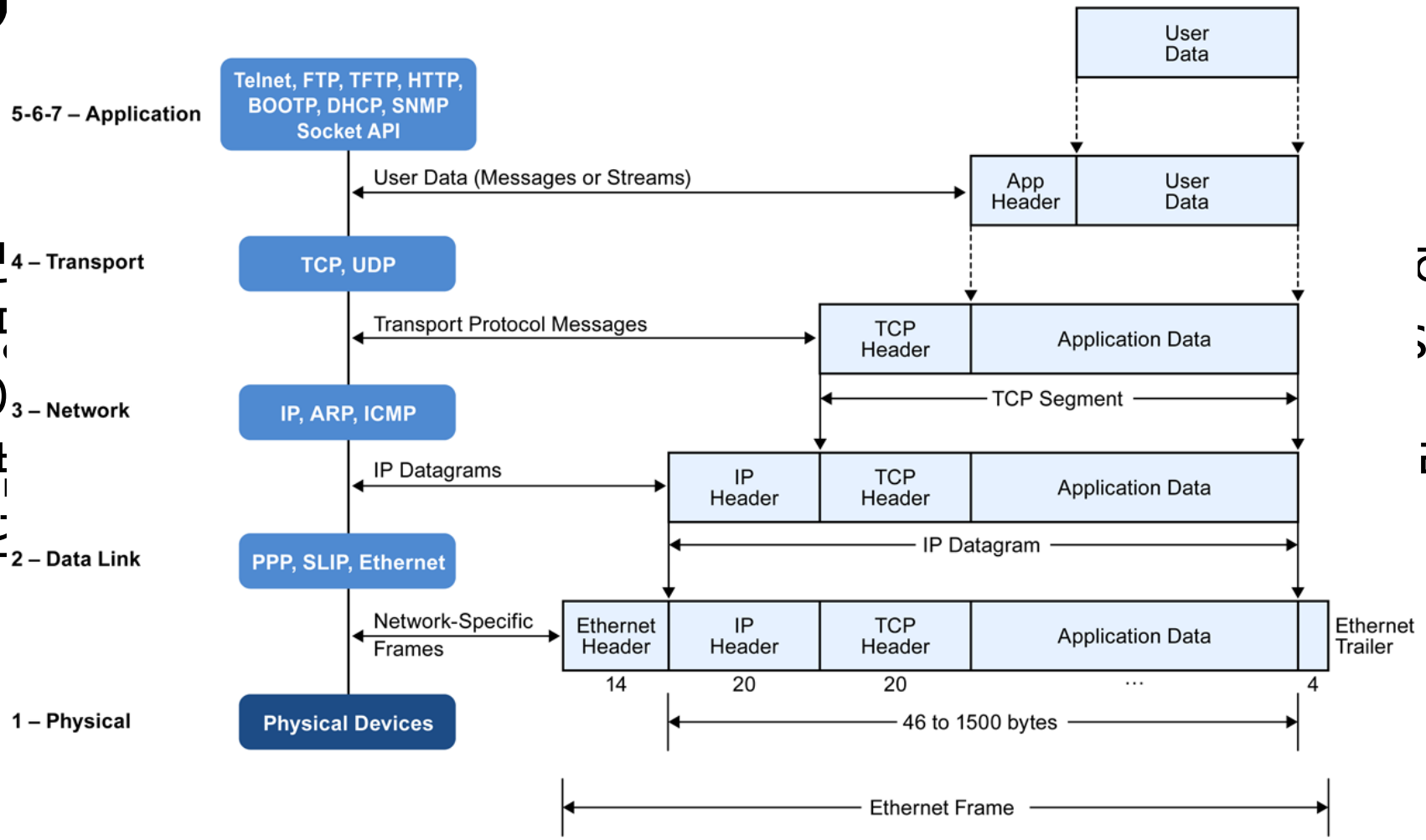
Jaehoon

- Pcap vs Socket
- Struct
- 예제
- Socket
- PF AF
- PF_INET PF_PACKET
- DEMO

목차

Pcap vs Socket

- L2
- 메모리
창 할
패킷이
터를
버퍼

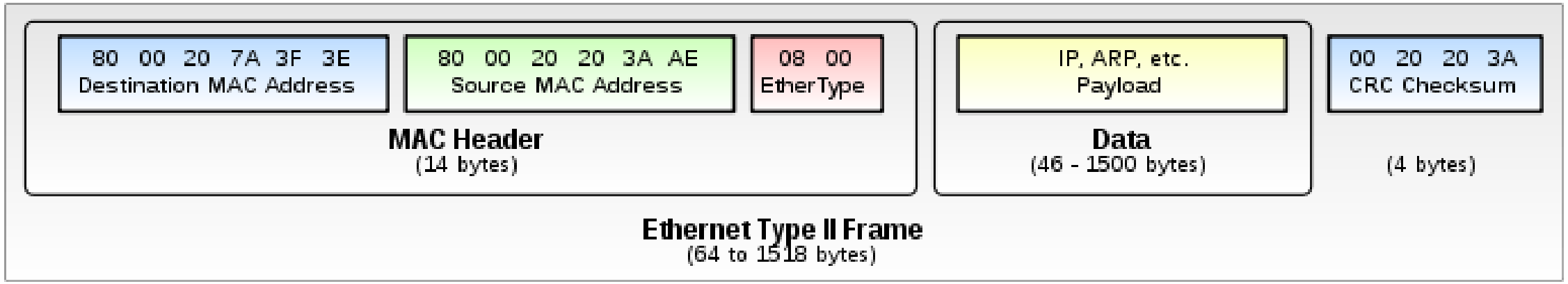


해 사용
socket
어나면
런 듯

struct

Format	C Type	Python Type	Standard size
x	pad byte	no value	
c	char	string of length 1	1
b	signed char	integer	1
B	unsigned char	integer	1
?	_Bool	bool	1
h	short	integer	2
H	unsigned short	integer	2
i	int	integer	4
I(대문자 i)	unsigned int	integer	4
l	long	integer	4
L	unsigned long	integer	4
q	long long	integer	8
Q	unsigned long long	integer	8
f	float	float	4
d	double	float	8
s	char[]	string	
p	char[]	string	
P	void *	integer	

Character	Byte Order	Size
@	시스템에 따름	시스템에 따름
=	시스템에 따름	표준
<	리틀 엔디안	표준
>	빅 엔디안	표준
!	네트워크(빅 엔디안)	표준




```python
def ethernet_frame(data):
    dst_mac, src_mac, eth_typ = struct.unpack('! 6s 6s H', data[:14])
    return get_mac_addr(dst_mac), get_mac_addr(src_mac), eth_typ, data[14:]
```

예제

```python
import socket
import struct

def main():
    conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(3))

    while True:
        raw_data, addr = conn.recvfrom(65536)
        dst_mac, src_mac, eth_typ, data = ethernet_frame(raw_data)
        l3 = "undefined ether type"
        if eth_typ == 0x0800:
            l3 = "IP"
        elif eth_typ == 0x0806:
            l3 = "ARP"
        print('\nEthernet Frame:')
        print('Destination: {}\nSource: {}\nEther Type: {}'.format(dst_mac, src_mac, l3))

def ethernet_frame(data):
    dst_mac, src_mac, eth_typ = struct.unpack('! 6s 6s H', data[:14])
    return get_mac_addr(dst_mac), get_mac_addr(src_mac), eth_typ, data[14:]

def get_mac_addr(bytes_addr):
    bytes_str = map('{:02X}'.format, bytes_addr)
    return ':'.join(bytes_str)

main()
```

예제

```
def main():
    conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(3))
```

첫번째 인자(family; 체계)

AF == Address Family

PF == Protocol Family

프로토콜 체계 (Protocol Family)	정의
_INET	IPv4인터넷 프로토콜
_INET6	IPv6인터넷 프로토콜
_LOCAL	LOCAL 통신을 위한 UNIX 프로토콜
_PACKET	Low level socket을 위한 인터페이스
_IPX	IPX 노벨 프로토콜

socket()함수에 프로토콜 패밀리에 AF_INET를 넣어도 되지만 PF_INET를 넣는게 바람직하고(내가 바람직하지 않게 짠거..)

struct sockaddr_in 구조체에 주소 체계를 넣을 때에도 PF_INET 를 넣어도 되지만 AF_INET를 넣는게 바람직하다.

출처 https://blog.naver.com/l18400/60109296392

예제

```
4  def main():
5      conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(3))
```

두번째 인자(Type; 소켓 유형)

socket.**SOCK_STREAM** → TCP 통신할 때

socket.**SOCK_DGRAM** → UDP 통신할 때

socket.**SOCK_RAW** → L2 조작 가능

socket.**SOCK_RDM**

socket.**SOCK_SEQPACKET**

L3

->python socket
원문에는 가장 많이 사용한다고 나와있지만 RAW도 많이 씀

예제

```
4  def main():
5      conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(3))
```

세번째 인자(proto; 프로토콜)

Linux의 경우 /etc/protocols에 나와있음

```
# Internet (IP) protocols
#
# Updated from http://www.iana.org/assignments/protocol-numbers and other
# sources.
# New protocols will be added on request if they have been officially
# assigned by IANA and are not historical.
# If you need a huge list of used numbers please install the nmap package.

ip          0       IP              # internet protocol, pseudo protocol
number
hopopt      0       HOPOPT          # IPv6 Hop-by-Hop Option [RFC1883]
icmp        1       ICMP            # internet control message protocol
igmp        2       IGMP            # Internet Group Management
ggp         3       GGP             # gateway-gateway protocol
ipencap     4       IP-ENCAP        # IP encapsulated in IP (officially ``IP'')
st          5       ST              # ST datagram mode
tcp         6       TCP             # transmission control protocol
egp         8       EGP             # exterior gateway protocol
igp         9       IGP             # any private interior gateway (Cisco)
pup         12      PUP             # PARC universal packet protocol
udp         17      UDP             # user datagram protocol
hmp         20      HMP             # host monitoring protocol
xns-idp     22      XNS-IDP         # Xerox NS IDP
(이하 생략)
```

DEMO