CS342
Report
Assignment 3: Wireshark

Anil Kag
10010111
a.kag@iitg.ernet.in

March 20, 2013

1 Part 1
Basics

1. Protocols which appear in the protocol column in the unfiltered packet-listing

   (a) ARP
   (b) DHCiPv6
   (c) DNS
   (d) Ethernet
   (e) ICMP
   (f) HTTP
   (g) LLC
   (h) LLMNR
   (i) SSDP
   (j) STP
   (k) TCP
   (l) UDP

2. Packet short details are as follows

no	time	source	destination	proto	len	info
310	17:00:18.753711	172.16.27.59	202.141.80.22	HTTP	867	GET http://www.google.co.in/ HTTP/1.1
391	17:00:18.903545	202.141.80.22	172.16.27.59	HTTP	66	HTTP/1.0 200 OK (text/html)

   Time taken = 0.903545-.753711 sec = 0.149834 sec

3. IP of google cannot be determined by looking at the above two packets because the proxy server(202.141.80.22) handles the connection part between my host & google.com

Host ip ⇒ 172.16.27.59

2 Part 2
Ethernet

Reference packets used for solving this part

no	time	source	destination	proto	len	info
113	17:47:02.684944	172.16.27.59	202.141.80.22	HTTP	668	GET http://www.faqs.org/rfcs/rfc826.html HTTP/1.1
391	17:47:03.771542	202.141.80.22	172.16.27.59	HTTP	3935	HTTP/1.0 200 OK (text/html)

1. HTTP GET message's ethernet header
   Ethernet II, Src: Pegatron_b3:05:c4 (38:60:77:b3:05:c4), Dst: Cisco_9d:70:00 (00:24:f9:9d:70:00)

   Ethernet address of your computer 38:60:77:b3:05:c4

2. Destination Ethernet address 00:24:f9:9d:70:00

   It's not the ethernet address of the RFC website.
   It's actually the ethernet address of the next hop for reaching the destination in my computer's routing table.

   You can actually check the IP for the device having destination ethernet address as this by running 'arp -n' on your linux machine & check the IP corresponding to this ethernet address on my machine.

```
$ arp -n
Address                  HWtype  HWaddress           Flags Mask    Iface
172.16.27.68             ether   f0:4d:a2:4f:15:6d   C             eth0
172.16.24.254            ether   00:24:f9:9d:70:00   C             eth0
```

   ⇒ it's ethernet address of 172.16.24.254

   Incomplete

3. Type field value = 0x0800 ⇒ IP packet ** What about the flags?

4. Ethernet Header Contents *Incomplete*

```
0000   00 24 f9 9d 70 00 38 60   77 b3 05 c4 08 00 45 00    .$..p.8` w.....E.
0010   02 8e 56 bd 40 00 40 06   ff bd ac 10 1b 3b ca 8d    ..V.@.@. .....;..
0020   50 16 c9 6a 0c 38 94 40   d2 f1 66 42 57 69 80 18    P..j.8.@ ..fBWi..
0030   00 e5 28 7f 00 00 01 01   08 0a 00 9e 84 4b 15 b1    ..(..... .....K..
0040   50 ee 47 45 54 20 68 74   74 70 3a 2f 2f 77 77 77    P.GET ht tp://www
```

ASCII letter 'G' starts on line 5 with base $0x0040$ & offset $0x0003$
$\Rightarrow$ position (in bytes) from the start of Ethernet Frame = $4*16+3 = 67$ ($0x0010 = 16$ in decimal)

5. HTTP Response
   Ethernet II, Src: Cisco_9d:70:00 (00:24:f9:9d:70:00), Dst: Pegatron_b3:05:c4 (38:60:77:b3:05:c4)
   src = 00:24:f9:9d:70:00 $\Rightarrow$ hop just before my computer in the path from website to my computer

6. dst = 38:60:77:b3:05:c4 $\Rightarrow$ my computer (you can verify via ifconfig & look at the hwaddress for the eth0 interface)

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 38:60:77:b3:05:c4
          inet addr:172.16.27.59  Bcast:172.16.27.255  Mask:255.255.252.0
          inet6 addr: fe80::3a60:77ff:feb3:5c4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8008125 errors:0 dropped:4342 overruns:0 frame:0
          TX packets:173669 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1129843477 (1.1 GB)  TX bytes:20988447 (20.9 MB)
          Interrupt:43 Base address:0xe000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:21789 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21789 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1986523 (1.9 MB)  TX bytes:1986523 (1.9 MB)
```

Incomplete

7. Type field == 0x0800 $\Rightarrow$ IP
   * what about flags?

3 Part 3
IP

1. Internet Protocol Version 4, Src: 172.16.27.59 (172.16.27.59), Dst: 202.141.80.21 (202.141.80.21)
   IP my computer = 172.16.27.59

2. Protocol: UDP (17)

3. Internet Header Length = 20bytes
   (if only looking at packet, value given in IHLen is $5 \Rightarrow 5 * 32$ bits $= 5 * 4$bytes $= 20$ bytes)

   ** Is it correct?
   Total Length = 56bytes = Header Length + IP Payload Length
   $\Rightarrow$ IP Payload Length = 56bytes - 20bytes = 36bytes
   (as Header length = 20bytes)

4. Flags: 0x00
   0... .... = Reserved bit: Not set
   .0.. .... = Don't fragment: Not set
   ..0. .... = More fragments: Not set
   Fragment offset: 0
   Since fragment offset is 0 & no more fragments are going to come
   Rightarrow no fragmentation

5. Identification & Checksum always change while going from one packet to other
   * Will TTL come here
   For each three packet TTL will be fixed & after that it'll be incremented by 1

6. const fields $\Rightarrow$ Version, Header Length(?), Protocol = UDP, Src & Dest IP
   field may change *Rightarrow* TTL, More Fragments, Total Length, Fragment Offset
   Which fields must change & why?

   ** More robust answer

7. Identification & TTL values do not remain same
   * why?

4 Part 4
UDP

1. (a) source port
   (b) destination port
   (c) length
   (d) checksum

2. Each Field is 4bytes(16 bits) long

3. Select the DNS query portion & it expands over 46bytes which is equal to the length given in the UDP packet
   ⇒ length in UDP packet refers to the actual data length

4. Protocol Number = 17(decimal), 0x11(hexadecimal)

5. Request "39","23:43:44.573062","172.16.27.59","202.141.80.9","DNS","80","Yes","Standard query A jampui.iitg.ernet.in"
   Internet Protocol Version 4, Src: 172.16.27.59 (172.16.27.59), Dst: 202.141.80.9 (202.141.80.9)
   User Datagram Protocol, Src Port: 56060 (56060), Dst Port: domain (53)

   Response "42","23:43:44.573688","202.141.80.9","172.16.27.59","DNS","171","Yes","Standard query response A 202.141.80.21"
   Internet Protocol Version 4, Src: 202.141.80.9 (202.141.80.9), Dst: 172.16.27.59 (172.16.27.59)
   User Datagram Protocol, Src Port: domain (53), Dst Port: 56060 (56060)

   Source port in one becomes the destination in other & vice-versa