Homework questions

lukas_borges

Chapter 4:

Review Questions R1,R2,R3,R4 (omit the parts of these review questions that refer to virtual circuits since we did not cover it in class), R8, R9, R12, R13, R16, R18, R21, R23, R24, R27

Problems: P7, P10, P24, P26, P30, P37, P42

Chapter 4:

Review questions:

R1.

Datagram.
Router's forwarding decisions are based on network-layer field value. A switch is a packet switcher whose forwarding decisions are based on the link-layer field value. Also, routers use IP addresses while link-layer switches use MAC addresses.

R2.

Forwarding and Routing.

R3.

Routing determines the end-to-end paths that packets should take from source to destination (uses routing algorithms).

Forwarding moves the packet that arrives at the input link to the correct output link (uses forwarding tables).

R4.

Yes they do.

For Datagram Networks:

- The forwarding tables consist of **header value** or **destination address** and **output link** fields.

Virtual Circuit Networks:

- Forwarding table consists of **incoming interface**, **incoming VC number**, **outgoing interface**, and **outgoing VC numbers** fields.

R8.

1. Switching via memory
2. Switching via bus
3. Switching via interconnection network

Memory:
Switching of input and output ports are directly controlled by CPU. Packets arrive from input port by fetching the interrupt signal in CPU. Routing processor catches the destination address from header, uses forwarding table to find appropriate output port and then copies the packet to the output port buffer.

Bus:
Packets are moved from input to output port directly over a shared bus. Because this bus is shared, only one packet at a time. When a packet arriving at the input port finds the bus busy, it is blocked and queued.

Interconnection network:
A crossbar switch is an example of such. A packet arriving at input port travels along the horizontal bus attached to the input port until it intersects with a vertical bus which leads to the appropriate output port. When the vertical bus is free, the packet can be transferred. If not, the packet is blocked and queued.

Multiple packets can be sent in parallel using switching via Interconnection network. This architecture allows packets from different input ports to each same output port.

R9.

Switching fabric could be slow (increasing queue size). Increased queue sizes cause router’s buffering space to be completely used.

When the queue size is too large, packet losses are more likely to happen.

To eliminate packet loss, increasing switching fabric speed at least by n times faster than the input line's speed. (n = number of input ports)

R12.

Yes. Every router has an IP address to address itself and each device connected.
Right now my computer is connected to my home network. My router's IP is `10.0.0.1`, and I can even use that address on my internet browser to access my router's settings page.
I can also tell that my computer's local IP address is `10.0.0.9` and laptop is `10.0.0.13`. These are local area network IPs used for intranet/router communication.

R13.

IP: `223.1.3.27`

This is IP version 4, meaning that each number can go from 0 to 255, because each of the numbers separated by a dot is composed of 8 bits.

We can convert each of these numbers into binary and fill remaining bits (if any) with 0s:

223: `1101 1111`

1: `0000 0001`

3: `0000 0011`

27: `0001 1011`

Patching everything together, we end up with the following 32-bit sequence:

`1101 1111 0000 0001 0000 0011 0001 1011`

R16.

Data	Measure
Chunk	40 bytes
Interval	0.020 seconds
TCP Header	20 bytes
IP Header	20 bytes

$Total\,Header\,Size = TCP\,Header + IP\,Header$

$= 20\,bytes + 20\,bytes$

$= 40\,bytes$

$Total\,Segment\,Size = Chunk + Total\,Header\,Size$

$= 40\,bytes + 40\,bytes$

$= 80\,bytes$

$80\,bytes = 100\%$

$40\,bytes = x\%$

$$x = \frac{40 \cdot 100}{80}$$

$$x = 50\%$$

$$50\%\,overhead\ 50\%\,data$$

R18.

The router will assign IP addresses to the 5 PCs automatically using Dynamic Host Control Protocol (DHCP). The default gateway, usually `192.168.0.1` or `10.0.0.1` is the router's IP. The beginning address for devices usually starts at `Default Gateway IP's + 1`.

So, if the default gateway is `10.0.0.1`, the router might do the following:

Device	IP
PC 1	`10.0.0.2`
PC 2	`10.0.0.3`
PC 3	`10.0.0.4`
PC 4	`10.0.0.5`
PC 5	`10.0.0.6`

These are local area network (LAN) IPs.
Whenever one of the PCs requests something from the public Internet, the router keeps the requesting IP and port in a table. Then it performs the request on the PC's behalf using the public IP. When it receives a packet, it forwards it to the requesting LAN IP accordingly. This is known as NAT, and it is necessary in this scenario because Internet service providers (ISP) cannot assign a specific public IP to each device connected. There are not enough public IPs (in IPv4) to serve each device connected.

R21.

Both are used to compute least-cost path between source and destination.

Differences:

Link-State:

- Network topology and all link costs are input.
- Computes least-cost path from source to destination with complete knowledge of the network.
- Uses Dijkstra's algorithm to calculate shortest path.
- Count-to-infinity problem can be averted.
- Creates routing table, neighbor table and topology table (more memory required).
- Updates are multicasted.

Distance-Vector:

- All associated costs with current node to all its neighbors is the input
- Computes least-cost path in iterative and distributed manner

- Uses Bellman-Ford algorithm to calculate shortest path.
- Count-to-infinity might be a problem.
- Creates a routing table (less memory space required).
- Updates are broadcasted.

R23.

No. In fact, it is better to have different routing algorithms because their trade-offs suite different cases more appropriately.
Each Autonomous System (AS) has administrative control for routing within it.

R24.

No. From the advertisement, D can reach z in 11 hops by using the path through A. D can already reach to z in 7 hops by using the path through B. As the value of D through path B is less than through path A, the table has no need to modify the entry of z.

R27.

BGP (Border Gateway Protocol) is used for Inter-AS Routing protocol.

RIP (Router Information Protocol) and **OSPF** (Open Shortest Path First) are Intra-AS protocols.

Inter-AS:

- BGP carries path attributes and provides controlled distribution of routing information. Its routing decisions are policy-based.
- Ability to scale and handle routing among large number of networks.
- Policy dominates quality and performance of routes.

Intra-AS:

- Policy is much less important when choosing routes (system goes for best choice).
- Ability to scale routing is more difficult. Might have to divide in smaller AS.
- Performance is focused on router because of single AS.

Problems:

P7.

a. No, there is no way to forward both packets through the switch at the same time. Shared bus means only one packet can be transferred at a single time over the bus.

b. Yes. forwarding two packets to two different output ports at the same time is possible using crossbar switch fabric.

c. No. Crossbar switch does not allow packets to be forwarded at the same time to the same output ports.

P10.

a)

Prefix Match	Link Interface
`11100000 00`	0
`11100000 01000000`	1
`1110000`	2
`11100001 1`	3
otherwise	3

b)

```
11001000 10010001 01010001 01010101
```

```
11100001 01000000 11000011 00111100
```

```
11000001 10000000 00010001 01110111
```

- First address prefix matches 5th entry - link interface 3
- Second address prefix matches 3rd entry - link interface 2
- Third address prefix matches 4th entry - link interface 3

P24.

Paths from $y \to u$ without repeating nodes:

1. $y \rightarrow x \rightarrow u$
2. $y \rightarrow w \rightarrow u$
3. $y \rightarrow x \rightarrow v \rightarrow u$
4. $y \rightarrow w \rightarrow v \rightarrow u$
5. $y \rightarrow w \rightarrow x \rightarrow u$
6. $y \rightarrow x \rightarrow w \rightarrow u$
7. $y \rightarrow z \rightarrow w \rightarrow u$
8. $y \rightarrow x \rightarrow w \rightarrow v \rightarrow u$
9. $y \rightarrow z \rightarrow w \rightarrow v \rightarrow u$
10. $y \rightarrow w \rightarrow v \rightarrow x \rightarrow u$
11. $y \rightarrow x \rightarrow v \rightarrow w \rightarrow u$
12. $y \rightarrow z \rightarrow w \rightarrow x \rightarrow u$
13. $y \rightarrow w \rightarrow x \rightarrow v \rightarrow u$
14. $y \rightarrow z \rightarrow w \rightarrow v \rightarrow x \rightarrow u$
15. $y \rightarrow z \rightarrow w \rightarrow x \rightarrow v \rightarrow u$

P26.

- Least-cost path from one node to all other nodes in the network.
- Iterative

$D(v)$ = least cost of path from source to destination (for node v)

$p(v)$ = previous node along the current path with least cost from source to v

N' = set of nodes

Step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	x	∞	∞	3,x	6,x	6,x	8,x
1	xv	7,v	6,v	3,x	6,x	6,x	8,x
2	xvu	7,v	6,v	3,x	6,x	6,x	8,x
3	xvuw	7,v	6,v	3,x	6,x	6,x	8,x
4	xvuwy	7,v	6,v	3,x	6,x	6,x	8,x
5	xvuwyt	7,v	6,v	3,x	6,x	6,x	8,x
6	xvuwytz	7,v	6,v	3,x	6,x	6,x	8,x

Shortest paths from x

node	path	cost
t	xvt	7
u	xvu	6
v	xv	3
w	xw	6
y	xy	6
z	xz	8

P30.

Least cost path from w to u = 5

Least cost path from y to u = 6

a)

$$D_x(w) = 2$$

$$D_x(y) = 4$$

$$D_x(u) = 7$$

b) If we drop $c(x, y)$ to a value smaller than 1 we will get a new least cost path from x to u. Node x will have to announce to its neighbors.

c) As long as $c(x, y) \geqslant 1$, least cost path from x to u is still going to be 7 (tied with the previous one), so it will not announce any changes.

P37.

a) **eBGP**

b) **iBGP**

c) **eBGP**

d) **iBGP**

P42.

W can receive from B only:

Tell B: $A \to W$.

Tell C: $C \to B \to A \to W$

V can receive from B or C:

Tell B: $A \to V$

Tell C: $A \to V$

C can go to V using $C \to A \to V$

C can go to W and V using $B \to A \to W$ and $B \to A \to V$

Wireshark Lab

Lab 6.01

1. nslookup

1.

```
✔ Luke  master  hw3
❯ nslookup
> kyoto-u.ac.jp
Server:		75.75.75.75
Address:	75.75.75.75#53

Non-authoritative answer:
Name:	kyoto-u.ac.jp
Address: 130.54.130.65
>
```

`130.54.130.65`

2.

```
✔ Luke  master±  hw3
❯ nslookup -type=NS cam.ac.uk
Server:		75.75.75.75
Address:	75.75.75.75#53

Non-authoritative answer:
cam.ac.uk	nameserver = authdns0.csx.cam.ac.uk.
cam.ac.uk	nameserver = sns-pb.isc.org.
cam.ac.uk	nameserver = ns2.ic.ac.uk.
cam.ac.uk	nameserver = dns0.cl.cam.ac.uk.
cam.ac.uk	nameserver = dns0.eng.cam.ac.uk.

Authoritative answers can be found from:
sns-pb.isc.org	internet address = 192.5.4.1
sns-pb.isc.org	has AAAA address 2001:500:2e::1
```

cambridge university:

authoritative servers:

`192.5.4.1` and `2001:500:2e::1`

3.

```
✔ Luke  master  hw3
❯ nslookup -type=NS cam.ac.uk
Server:         75.75.75.75
Address:        75.75.75.75#53

Non-authoritative answer:
cam.ac.uk       nameserver = sns-pb.isc.org.
cam.ac.uk       nameserver = dns0.eng.cam.ac.uk.
cam.ac.uk       nameserver = dns0.cl.cam.ac.uk.
cam.ac.uk       nameserver = ns2.ic.ac.uk.
cam.ac.uk       nameserver = authdns0.csx.cam.ac.uk.

Authoritative answers can be found from:
sns-pb.isc.org  internet address = 192.5.4.1
sns-pb.isc.org  has AAAA address 2001:500:2e::1

✔ Luke  master  hw3
❯ nslookup mail.yahoo.com sns-pb.isc.org
Server:         sns-pb.isc.org
Address:        2001:500:2e::1#53

** server can't find mail.yahoo.com: REFUSED

✔ Luke  master  hw3
❯ nslookup mail.yahoo.com authdns0.csx.cam.ac.uk
Server:         authdns0.csx.cam.ac.uk
Address:        2001:630:212:8::d:a0#53

** server can't find mail.yahoo.com: REFUSED
```

mail.yahoo.com was refused.

3. Tracing DNS with Wireshark

4.

No.	Time	Source	Destination	Protocol	Length	Info
15	01:32:17.972677	10.0.0.13	75.75.75.75	DNS	68	Standard query 0xa1ee A ietf.org
16	01:32:17.972807	10.0.0.13	75.75.75.75	DNS	68	Standard query 0xe20b AAAA ietf.org
17	01:32:18.062706	75.75.75.75	10.0.0.13	DNS	96	Standard query response 0xe20b AAAA ietf.org AAAA 2001:1900:3001:11::2c
18	01:32:18.073198	75.75.75.75	10.0.0.13	DNS	84	Standard query response 0xa1ee A ietf.org A 4.31.198.44
19	01:32:18.073963	10.0.0.13	4.31.198.44	TCP	78	49673 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=60891225 TSecr=0 S…
20	01:32:18.154506	4.31.198.44	10.0.0.13	TCP	74	80 → 49673 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1…
21	01:32:18.154604	10.0.0.13	4.31.198.44	TCP	66	49673 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=60891305 TSecr=179882109
22	01:32:18.154915	10.0.0.13	4.31.198.44	HTTP	358	GET / HTTP/1.1
23	01:32:18.235235	4.31.198.44	10.0.0.13	TCP	66	80 → 49673 [ACK] Seq=1 Ack=293 Win=30080 Len=0 TSval=179882130 TSecr=60891305
24	01:32:18.237584	4.31.198.44	10.0.0.13	HTTP	1514	HTTP/1.1 200 OK (text/html)
25	01:32:18.237941	4.31.198.44	10.0.0.13	TCP	1514	80 → 49673 [ACK] Seq=1449 Ack=293 Win=30080 Len=1448 TSval=179882130 TSecr=608…
26	01:32:18.238004	10.0.0.13	4.31.198.44	TCP	66	49673 → 80 [ACK] Seq=293 Ack=2897 Win=129600 Len=0 TSval=60891387 TSecr=179882…

```
▶ Frame 15: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▶ Ethernet II, Src: Apple_ef:3c:a0 (78:31:c1:ef:3c:a0), Dst: Technico_57:1e:73 (44:32:c8:57:1e:73)
▶ Internet Protocol Version 4, Src: 10.0.0.13, Dst: 75.75.75.75
▶ User Datagram Protocol, Src Port: 63749 (63749), Dst Port: 53 (53)
▶ Domain Name System (query)
```

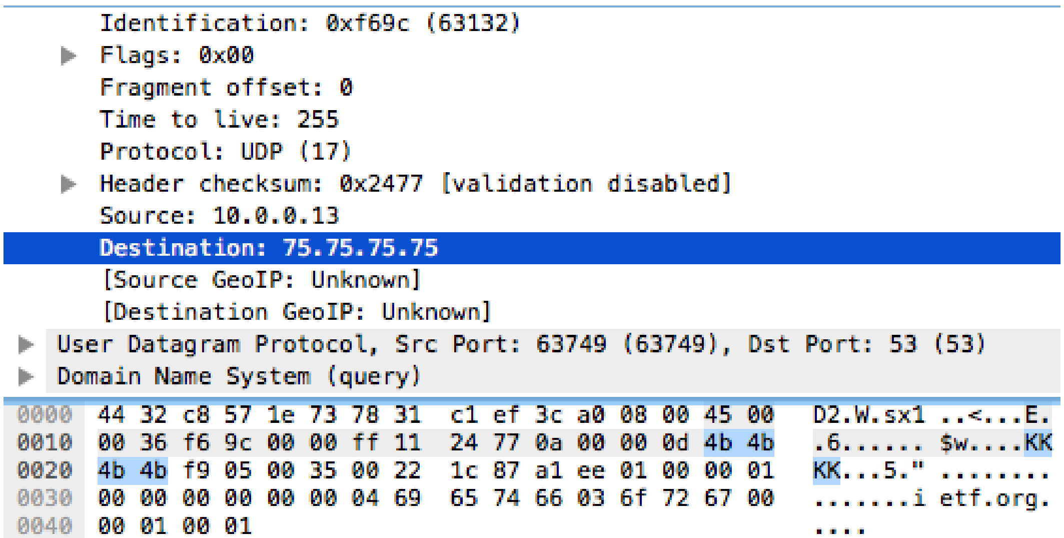
```
0000  44 32 c8 57 1e 73 78 31  c1 ef 3c a0 08 00 45 00   D2.W.sx1 ..<...E.
0010  00 36 f6 9c 00 00 ff 11  24 77 0a 00 00 0d 4b 4b   .6...... $w....KK
0020  4b 4b f9 05 00 35 00 22  1c 87 a1 ee 01 00 00 01   KK...5." ........
0030  00 00 00 00 00 00 04 69  65 74 66 03 6f 72 67 00   .......i etf.org.
0040  00 01 00 01                                        ....
```

Clearly, UDP.

5.

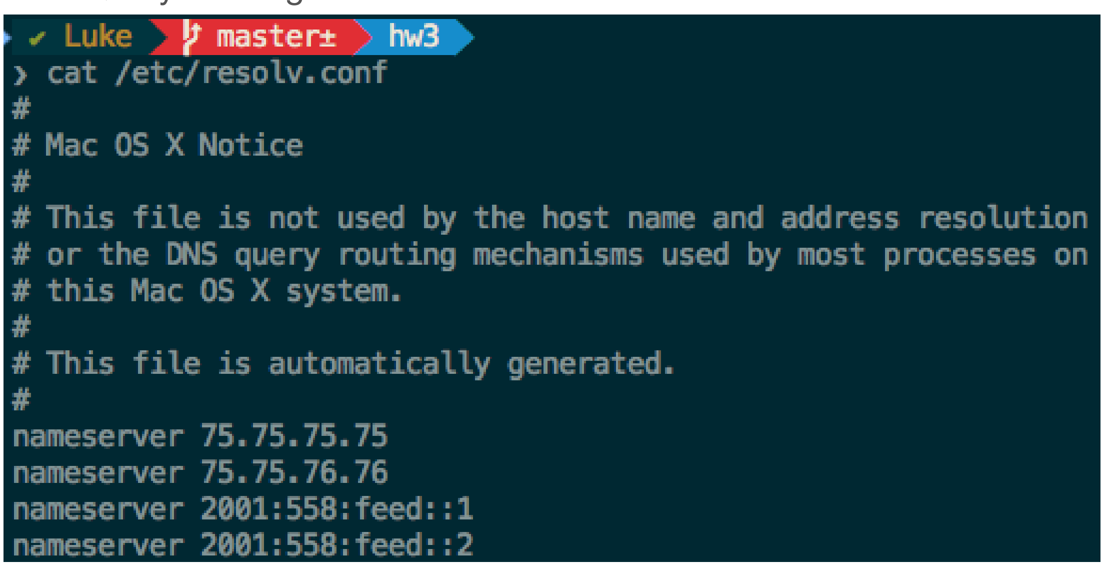
message	source	destination
query	63749	53
response	53	63749

6.



```
Identification: 0xf69c (63132)
▶ Flags: 0x00
Fragment offset: 0
Time to live: 255
Protocol: UDP (17)
▶ Header checksum: 0x2477 [validation disabled]
Source: 10.0.0.13
Destination: 75.75.75.75
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
▶ User Datagram Protocol, Src Port: 63749 (63749), Dst Port: 53 (53)
▶ Domain Name System (query)
0000  44 32 c8 57 1e 73 78 31  c1 ef 3c a0 08 00 45 00   D2.W.sx1 ..<...E.
0010  00 36 f6 9c 00 00 ff 11  24 77 0a 00 00 0d 4b 4b   .6...... $w....KK
0020  4b 4b f9 05 00 35 00 22  1c 87 a1 ee 01 00 00 01   KK...5." ........
0030  00 00 00 00 00 00 04 69  65 74 66 03 6f 72 67 00   .......i etf.org.
0040  00 01 00 01                                        ....
```

DNS Query message was sent to `75.75.75.75`



```
✔ Luke  master±  hw3
❯ cat /etc/resolv.conf
#
# Mac OS X Notice
#
# This file is not used by the host name and address resolution
# or the DNS query routing mechanisms used by most processes on
# this Mac OS X system.
#
# This file is automatically generated.
#
nameserver 75.75.75.75
nameserver 75.75.76.76
nameserver 2001:558:feed::1
nameserver 2001:558:feed::2
```

Yes, they are the same.

P.S.: `cat /etc/resolv.conf` shows IP address of local DNS server in Unix systems.

7. The message is clearly “type A” (standard host address resource record). No

answers.

8. The DNS Type A query has only one response.

   This is the response:

```
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▶ Queries
  ▼ Answers
    ▼ ietf.org: type A, class IN, addr 4.31.198.44
          Name: ietf.org
          Type: A (Host Address) (1)
          Class: IN (0x0001)
          Time to live: 1800
          Data length: 4
          Address: 4.31.198.44
0000  78 31 c1 ef 3c a0 44 32  c8 57 1e 73 08 00 45 00   x1..<.D2 .W.s..E.
0010  00 46 00 00 40 00 3b 11  9f 04 4b 4b 4b 4b 0a 00   .F..@.;. ..KKKK..
0020  00 0d 00 35 f9 05 00 32  0a 7f a1 ee 81 80 00 01   ...5...2 ........
0030  00 01 00 00 00 00 04 69  65 74 66 03 6f 72 67 00   .......i etf.org.
0040  00 01 00 01 c0 0c 00 01  00 01 00 00 07 08 00 04   ........ ........
0050  04 1f c6 2c                                        ...,
```

It contains some information, but most importantly, the IP address.

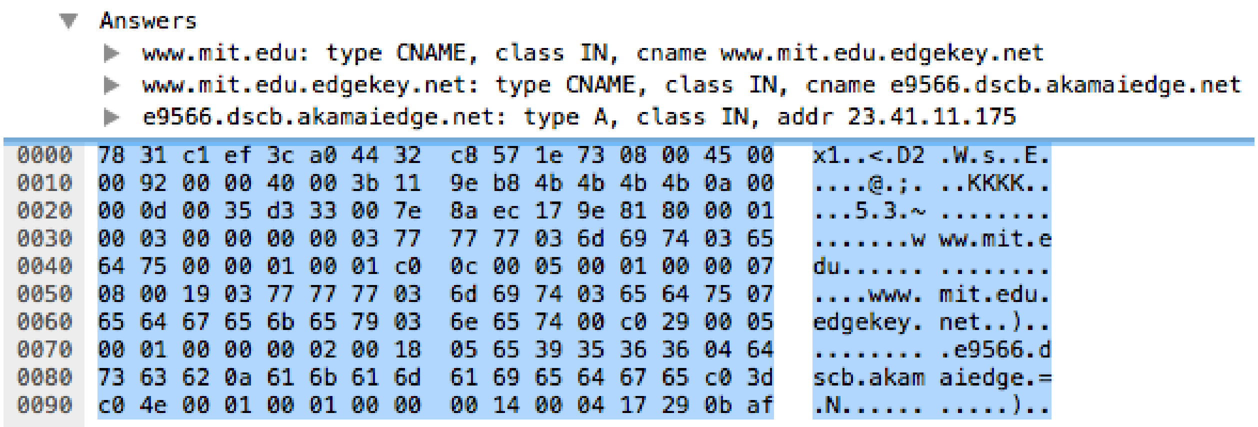
9.

15	01:32:17.972677	10.0.0.13	75.75.75.75	DNS	68	Standard query 0xa1ee A ietf.org
16	01:32:17.972807	10.0.0.13	75.75.75.75	DNS	68	Standard query 0xe20b AAAA ietf.org
17	01:32:18.062706	75.75.75.75	10.0.0.13	DNS	96	Standard query response 0xe20b AAAA ietf.org AAAA 2001:1900:3001:11::2c
18	01:32:18.073198	75.75.75.75	10.0.0.13	DNS	84	Standard query response 0xa1ee A ietf.org A 4.31.198.44
19	01:32:18.073963	10.0.0.13	4.31.198.44	TCP	78	49673 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=60891225 TSecr=0 S…
20	01:32:18.154506	4.31.198.44	10.0.0.13	TCP	74	80 → 49673 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1…
21	01:32:18.154604	10.0.0.13	4.31.198.44	TCP	66	49673 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=60891305 TSecr=179882109
22	01:32:18.154915	10.0.0.13	4.31.198.44	HTTP	358	GET / HTTP/1.1
23	01:32:18.235235	4.31.198.44	10.0.0.13	TCP	66	80 → 49673 [ACK] Seq=1 Ack=293 Win=30080 Len=0 TSval=179882130 TSecr=60891305
24	01:32:18.237584	4.31.198.44	10.0.0.13	HTTP	1514	HTTP/1.1 200 OK (text/html)
25	01:32:18.237941	4.31.198.44	10.0.0.13	TCP	1514	80 → 49673 [ACK] Seq=1449 Ack=293 Win=30080 Len=1448 TSval=179882130 TSecr=608…
26	01:32:18.238004	10.0.0.13	4.31.198.44	TCP	66	49673 → 80 [ACK] Seq=293 Ack=2897 Win=129600 Len=0 TSval=60891387 TSecr=179882…

   Yes, the destination address of the following TCP packet matches the IP from the last DNS response.

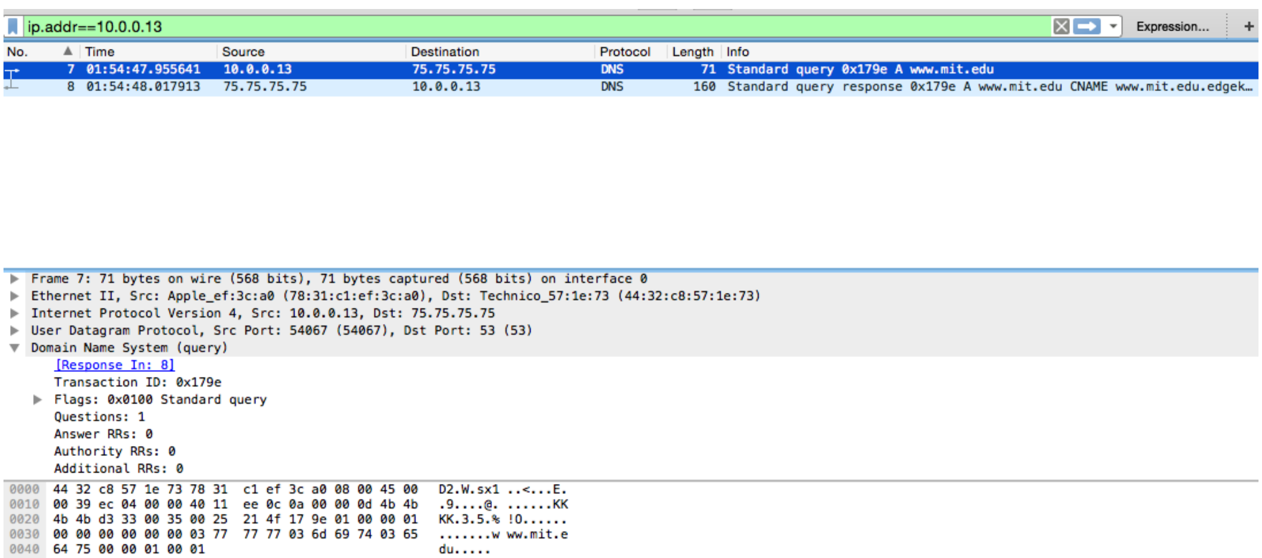
   `4.31.198.44`

10. No, no additional queries are necessary. All pictures should be accessible from `4.31.198.44` .
11. `53` for query message destination and `53` for query response source.
12. Destination IP: `75.75.75.75` , same as my default local DNS server.
13. DNS Query message is Type A (standard). Query message contains no answers.
14. Three answers provided.



These are non-authoritative server aliases (canonical names). The last one contains the IP `23.41.11.75` .

15.

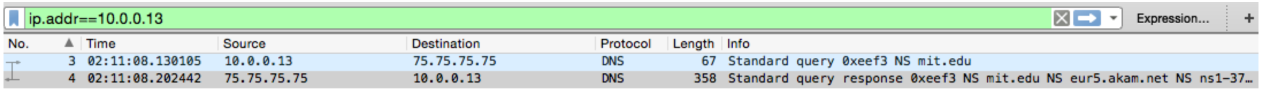


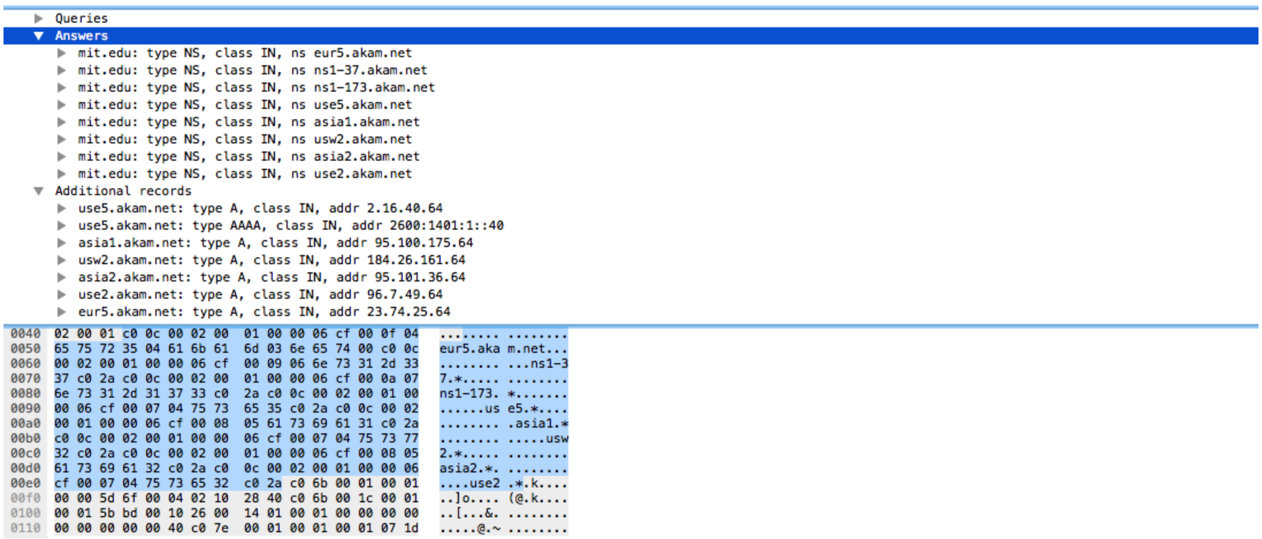
16. `75.75.75.75` . Yes it is.

17. Since we specified `-type=NS` , this is a `NS` (name server) query message. No answers.

18. ns1-37.akam.net
ns1-173.akam.net
use5.akam.net
asia1.akam.net
usw2.akam.net
asia2.akam.net
use2.akam.net

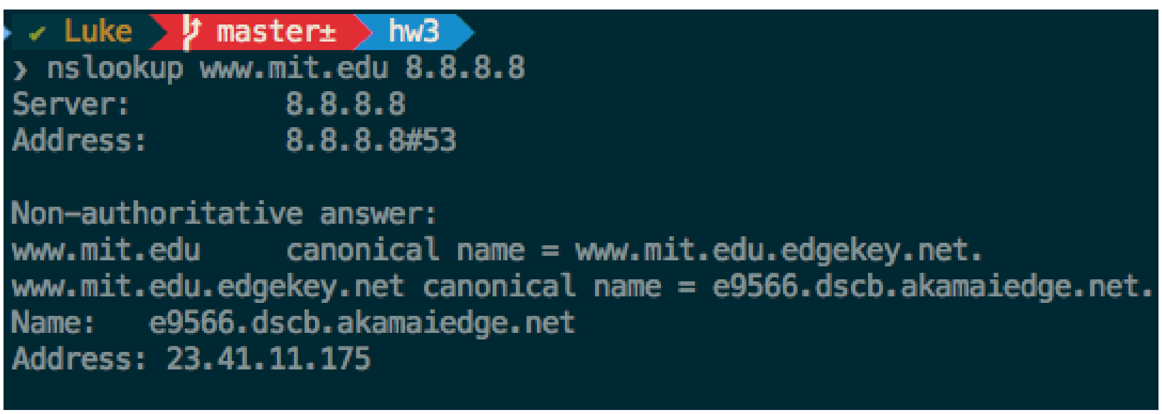
19.





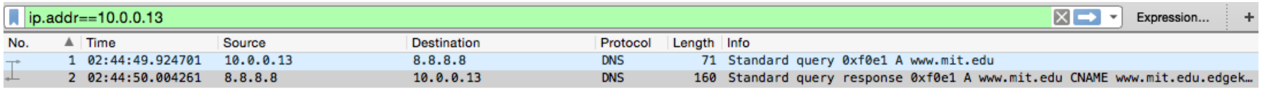
For this part, links are too old and acting weird.

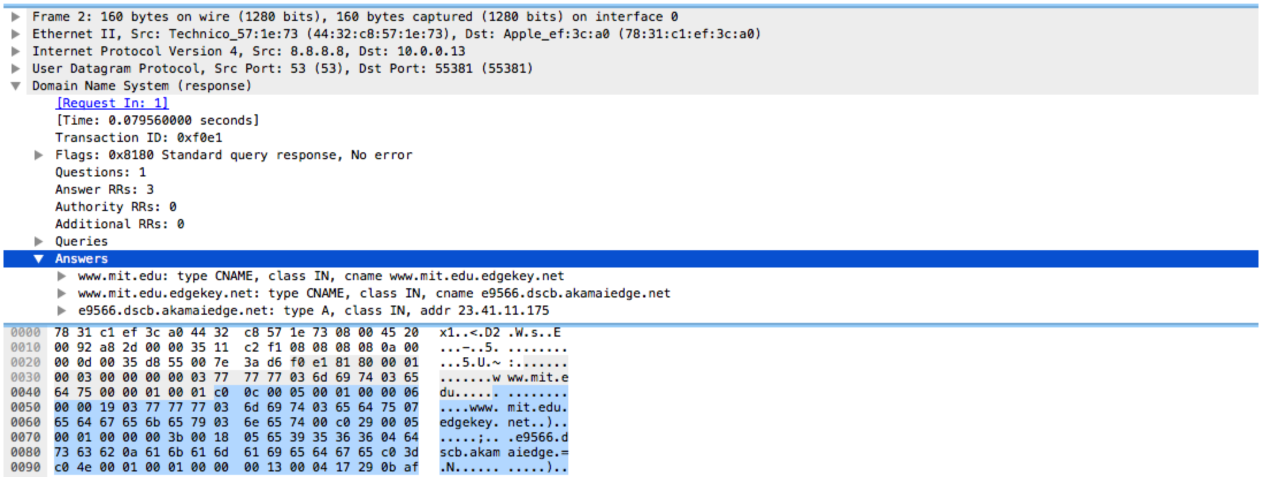
I used `nslookup www.mit.edu 8.8.8.8` which is google's public DNS address.



20. `8.8.8.8`. This is not my local DNS server's address, it corresponds to Google's public DNS.
21. Type A, no answers.
22. Three answers with the canonical names, and one host IP.

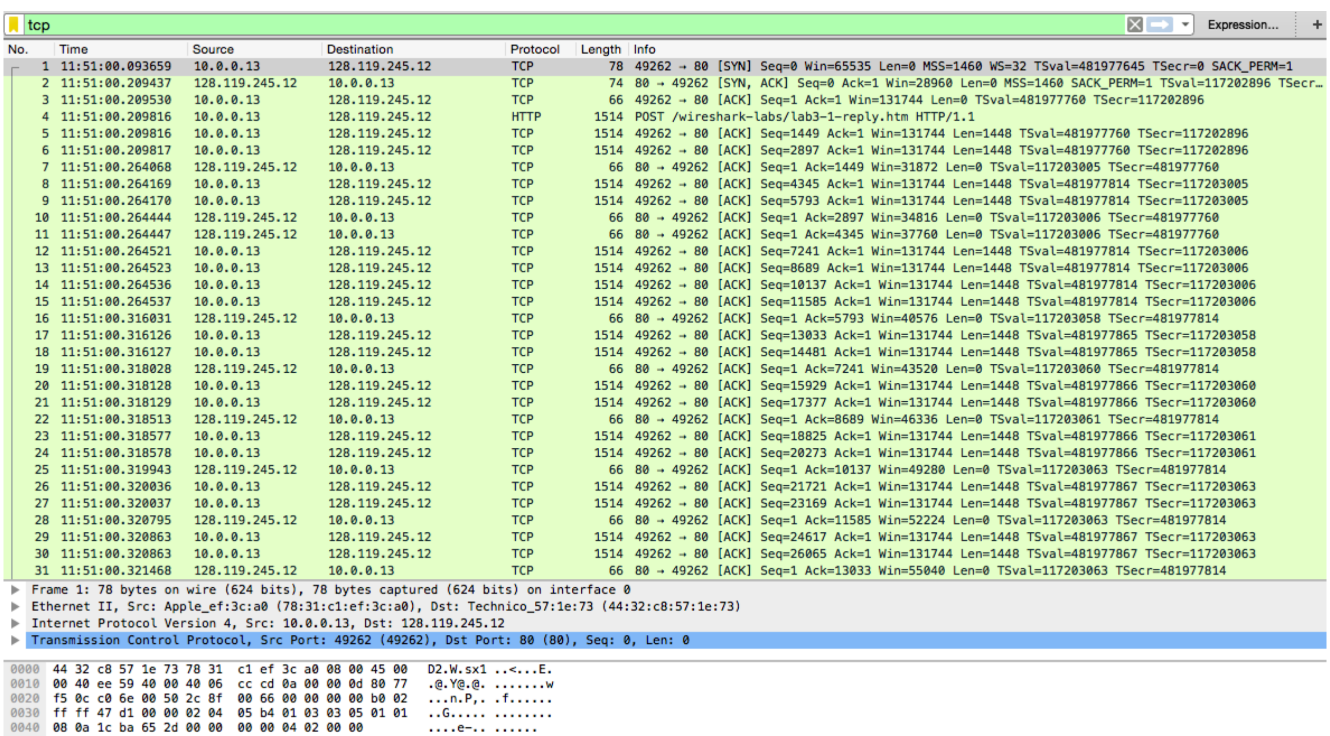
23.





Lab 6.0

capture trace



1. IP address: `10.0.0.13` , port: `49262`
2. IP address: `128.119.245.12` (gaia.cs.umass.edu) port: `80`

3. Source: IP address: `10.0.0.13` port: `49262`
   sends the .txt file to: IP address: `128.119.245.12` port: `80`

tcp basics

4. `Seq = 0`; `[SYN]` flag set to 1
5. `Seq = 0` and `Ack = 1`
   gaia.cs.umass.edu adds 1 to the initial sequence number of `[SYN]`, which was 0. 0 + 1 = 1. `[SYN]` and `[ACK]` flag are set to 1, signaling this is a `[SYNACK]`
6. `Seq = 1`

7.

seqs:

```
  4. Seq = 1
  5. Seq = 1449
  6. Seq = 2897
  8. Seq = 4345
  9. Seq = 5793
  12.Seq = 7241
```

acks:

```
7. Ack = 1449
10.Ack = 2897
11.Ack = 4345
16.Ack = 5793
19.Ack = 7241
22.Ack = 8689
```

$$EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT$$

segment	sent time	received time	RTT
4	00.209816	00.264068	0.054252
5	00.209816	00.264444	0.054628
6	00.209817	00.264447	0.054653
8	00.264169	00.316031	0.051862
9	00.264170	00.318028	0.053858
12	00.264521	00.318513	0.053992

$EstimatedRTT$ seg 4 = 0.054252

$EstimatedRTT$ seg 5 = $0.875 * 0.054252 + 0.125 * 0.54628 = 0.115755$

$EstimatedRTT$ seg 6 = $0.875 * 0.115755 + 0.125 * 0.054653 = 0.108117$

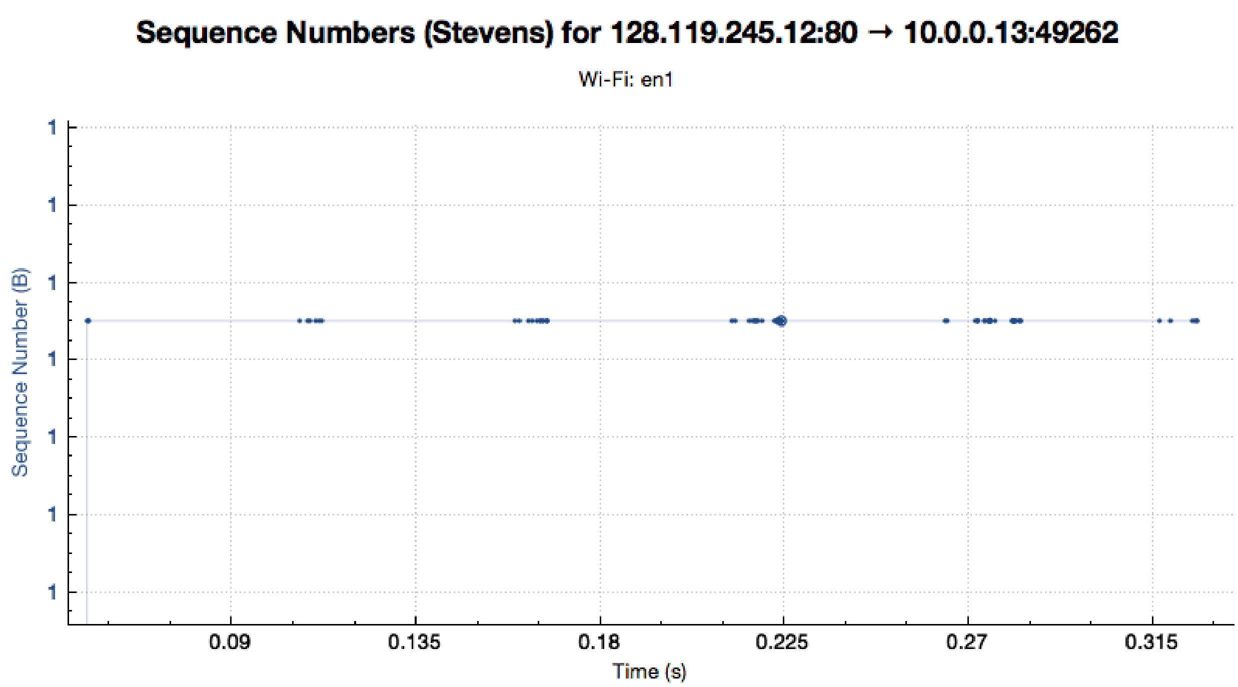
$EstimatedRTT$ seg 8 = $0.875 * 0.108117 + 0.125 * 0.051862 = 0.101085$

$EstimatedRTT$ seg 9 = $0.875 * 0.101085 + 0.125 * 0.054858 = 0.157021$

$EstimatedRTT$ seg 12 = $0.875 * 0.157021 + 0.125 * 0.053992 = 0.144142$

8. They’re all 1448 bytes in length (1514 for data).
9. First ACK packet (SYN ACK) advertises `Win = 1460`.
   This number grows up to `Win = 46336`. The sender never throttles.

10.



As we can see from the time sequence number graph, all sequence numbers acknowledged increase monotonically in relation to time.
No packets were retransmitted.

11.

ACK	acknowledged seqNum	acknowledged data
7	1449	1448
10	2897	1448
11	4345	1448
16	5793	1448
19	7241	1448
22	8689	1448

Acknowledged data was steadily 1448 bytes. (acknowledged sequence number increases by 1448 every for every new segment).

12. We can compute the total amount of data by calculating the difference between sequence number of the first TCP segment (1 byte for segment 4) and the acknowledged sequence number of the last ACK (segment 192, ACK 145578 bytes)

$145578 - 1 = 145577$ bytes.
So we calculate what time was the first TCP send and what time was the last ACK received:

segment 4 = sent at 00.209816
segment 192 = sent at 00.535381

$00.535381 - 00.209816 = 0.325565$ seconds

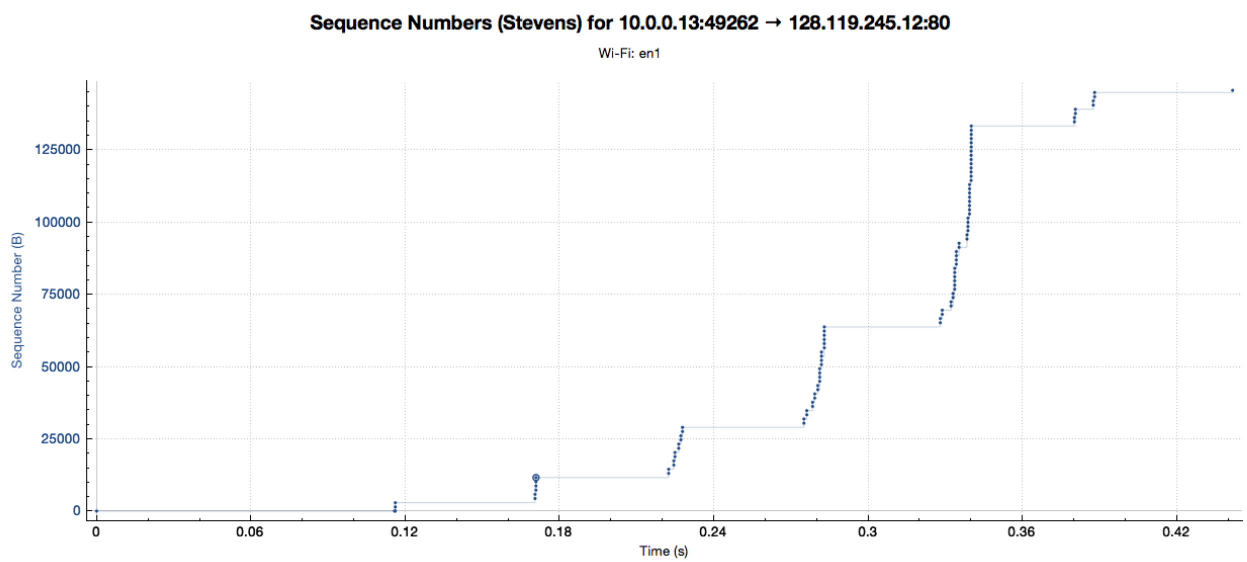
Now we'll do:

$$\frac{bytes}{seconds} = \frac{145577\,bytes}{0.325565\,seconds}$$

$447151.8744 = 447.15$ Kbytes/seconds

TCP Congestion control

13. According to the graph, the slow start phase begins at 0 and ends a little before 0.12 seconds. Then congestion takes over.
    During this small interval, the data transferred is only a small fraction of the window size instead of the ideal 1/3.
14. Screenshot:



Lab 6.1

UDP packet trace

1.

```
▶ Ethernet II, Src: Technico_57:1e:73 (44:32:c8:57:1e:73), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
▶ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 239.255.255.250
▼ User Datagram Protocol, Src Port: 1900 (1900), Dst Port: 1900 (1900)
     Source Port: 1900
     Destination Port: 1900
     Length: 345
  ▶ Checksum: 0x61c0 [validation disabled]
     [Stream index: 44]
▶ Data (337 bytes)
```

Four fields:

- Source port
- Destination port
- Length
- Checksum

2.

Header	Hex Value
Source Port	07 6c
Destination Port	07 6c
Length	01 59
Checksum	61 c0

each hexadecimal digit = 4 bits
16 bits for each header value (or 2 bytes).

3. $Header + Data = Length$

   4 header fields, each with 2 bytes = 8 bytes total

   8 bytes + 337 bytes of data = 345 bytes for length.

4. Max IP packet size $= 2^{16} - 1 = 65535$

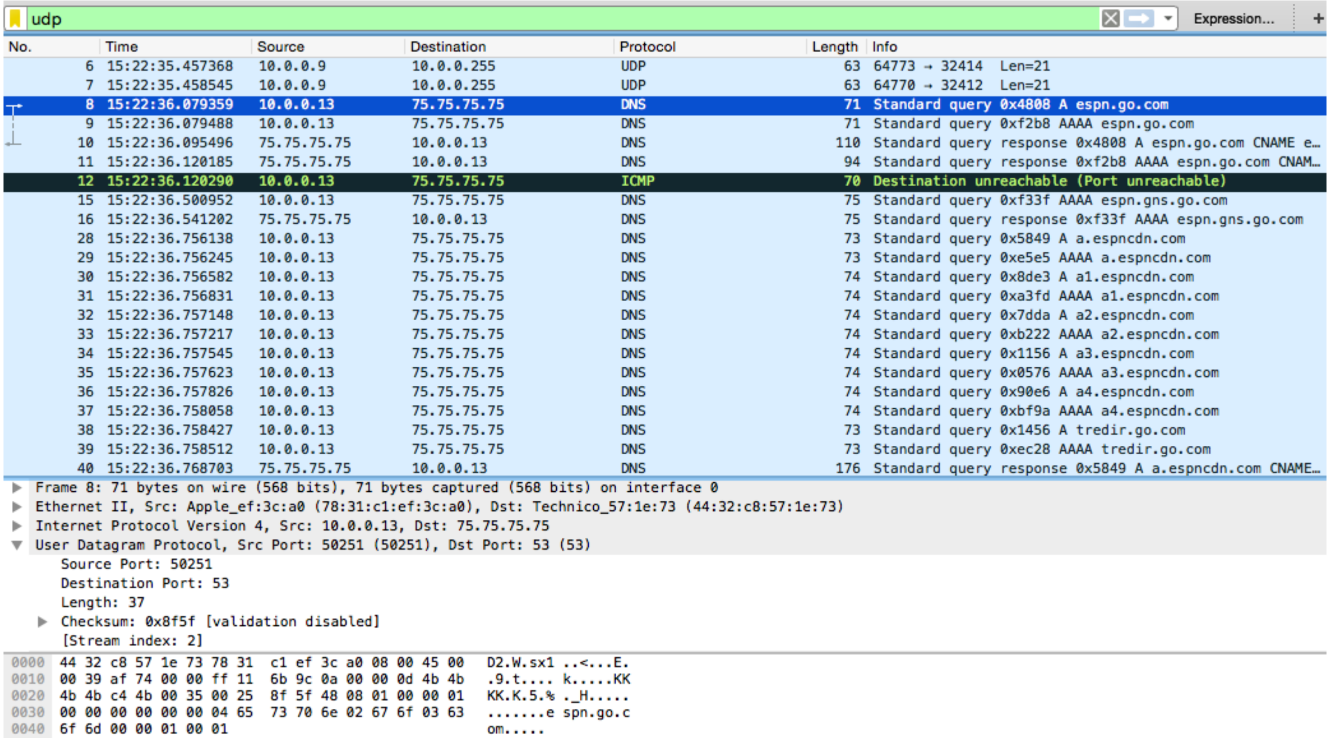
   $65535 - 8\, bytes$ (for header) $= 65527$

5. $2^{16} - 1 = 65535$

6. $0x11$ or $17d$

```
▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 365
  Identification: 0x714c (29004)
▶ Flags: 0x00
  Fragment offset: 0
  Time to live: 4
  Protocol: UDP (17)
▶ Header checksum: 0x4a39 [validation disabled]
  Source: 10.0.0.1
0010  01 6d 71 4c 00 00 04 11  4a 39 0a 00 00 01 ef ff   .mqL.... J9......
```

7.



Packets 8 and 10 are DNS queries made over UDP protocol.
my host (local IP 10.0.0.13) sends a query to 75.75.75.75 (my local DNS server) from port 50251 to port 53.

When this query is replied, source 75.75.75.75, port 53 sends a packet back to 10.0.0.13, port 50251

So, the source and destination port of the first packet are destination and source port of the second packet respectively.