

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 $y \rightarrow u$ 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	xvuwx	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 $c(x, y) \geq 1$, 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 : $A \rightarrow W$.

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

V can receive from B or C :

Tell B : $A \rightarrow V$

Tell C : $A \rightarrow V$

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

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

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 00 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...S."

0030 00 00 00 00 00 00 04 69 65 74 66 03 6f 72 67 00i 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
0010	57	1e	73	08	00	45	00	x1..<.D2	.W.s..E.
0020	00	46	00	00	40	00	3b	11	9f
0030	04	b4	4b	4b	4b	0a	00	.F..@.;.	..KKKK..
0040	00	0d	00	35	f9	05	00	32	0a
0050	7f	a1	ee	81	80	00	01	...5...2
	00	01	00	00	00	00	04	69	65
	74	66	03	6f	72	67	00i	etf.org.
	00	01	00	01	c0	0c	00	01	00
	01	00	00	07	08	00	04
	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.

▼ Answers

- ▶ www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
- ▶ www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
- ▶ e9566.dscb.akamaiedge.net: type A, class IN, addr 23.41.11.175

```

0000 78 31 c1 ef 3c a0 44 32 c8 57 1e 73 08 00 45 00 x1.<.D2 .W.s..E.
0010 00 92 00 00 40 00 3b 11 9e b8 4b 4b 4b 4b 0a 00 ....@.;. ..KKKK..
0020 00 0d 00 35 d3 33 00 7e 8a ec 17 9e 81 80 00 01 ...5.3.~ .....
0030 00 03 00 00 00 00 03 77 77 77 03 6d 69 74 03 65 .....w ww.mit.e
0040 64 75 00 00 01 00 01 c0 0c 00 05 00 01 00 00 07 du.....
0050 08 00 19 03 77 77 77 03 6d 69 74 03 65 64 75 07 ....www. mit.edu.
0060 65 64 67 65 6b 65 79 03 6e 65 74 00 c0 29 00 05 edgekey. net..)..
0070 00 01 00 00 00 02 00 18 05 65 39 35 36 36 04 64 ..... e9566.d
0080 73 63 62 0a 61 6b 61 6d 61 69 65 64 67 65 c0 3d scb.akam aiedge.=
0090 c0 4e 00 01 00 01 00 00 00 14 00 04 17 29 0b af .N..... ..).

```

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

15.

No.	Time	Source	Destination	Protocol	Length	Info
7	01:54:47.955641	10.0.0.13	75.75.75.75	DNS	71	Standard query 0x179e A www.mit.edu
8	01:54:48.017913	75.75.75.75	10.0.0.13	DNS	160	Standard query response 0x179e A www.mit.edu CNAME www.mit.edu.edgekey.net

```

▶ Frame 7: 71 bytes on wire (568 bits), 71 bytes captured (568 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: 54067 (54067), Dst Port: 53 (53)
▼ Domain Name System (query)
  [Response In: 8]
  Transaction ID: 0x179e
  Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0

```

```

0000 44 32 c8 57 1e 73 78 31 c1 ef 3c a0 00 00 45 00 D2.W.sx1 ..<...E.
0010 00 39 ec 04 00 00 40 11 ee 0c 0a 00 00 0d 4b 4b .9....@. ....KK
0020 4b 4b d3 33 00 35 00 25 21 4f 17 9e 01 00 00 01 KK.3.5.% !0.....
0030 00 00 00 00 00 00 03 77 77 77 03 6d 69 74 03 65 .....w ww.mit.e
0040 64 75 00 00 01 00 01 00 00 14 00 04 17 29 0b af du.....

```

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.

ip.addr==10.0.0.13							Expression...	+
No.	Time	Source	Destination	Protocol	Length	Info		
3	02:11:08.130105	10.0.0.13	75.75.75.75	DNS	67	Standard query 0xeef3 NS mit.edu		
4	02:11:08.202442	75.75.75.75	10.0.0.13	DNS	358	Standard query response 0xeef3 NS mit.edu NS eur5.akam.net NS ns1-37...		

Queries
Answers
<ul style="list-style-type: none"> mit.edu: type NS, class IN, ns eur5.akam.net mit.edu: type NS, class IN, ns ns1-37.akam.net mit.edu: type NS, class IN, ns ns1-173.akam.net mit.edu: type NS, class IN, ns use5.akam.net mit.edu: type NS, class IN, ns asia1.akam.net mit.edu: type NS, class IN, ns usw2.akam.net mit.edu: type NS, class IN, ns asia2.akam.net mit.edu: type NS, class IN, ns use2.akam.net
Additional records
<ul style="list-style-type: none"> use5.akam.net: type A, class IN, addr 2.16.40.64 use5.akam.net: type AAAA, class IN, addr 2600:1401:1::40 asia1.akam.net: type A, class IN, addr 95.100.175.64 usw2.akam.net: type A, class IN, addr 104.26.161.64 asia2.akam.net: type A, class IN, addr 95.101.36.64 use2.akam.net: type A, class IN, addr 96.7.49.64 eur5.akam.net: type A, class IN, addr 23.74.25.64

0040	02 00 01 c0 0c 00 02 00	01 00 00 06 cf 00 0f 04
0050	65 75 72 35 04 61 6b 61	6d 03 6e 65 74 00 c0 0c	eur5.aka m.net...
0060	00 02 00 01 00 00 06 cf	00 09 06 6e 73 31 2d 33ns1-3
0070	37 c0 2a c0 0c 00 02 00	01 00 00 06 cf 00 0a 07	7.*.....
0080	6e 73 31 2d 31 37 33 c0	2a c0 0c 00 02 00 01 00	ns1-173.
0090	00 06 cf 00 07 04 75 73	65 35 c0 2a c0 0c 00 02us e5.*.....
00a0	00 01 00 00 06 cf 00 08	05 61 73 69 61 31 c0 2aasia1.*
00b0	c0 0c 00 02 00 01 00 00	06 cf 00 07 04 75 73 77usw
00c0	32 c0 2a c0 0c 00 02 00	01 00 00 06 cf 00 08 05	2.*.....
00d0	61 73 69 61 32 c0 2a c0	0c 00 02 00 01 00 00 06	asia2.*.....
00e0	cf 00 07 04 75 73 65 32	c0 2a c0 6b 00 01 00 01use2.*.k....
00f0	00 00 5d 6f 00 04 02 10	28 40 c0 6b 00 1c 00 01	...jo... (@.k....
0100	00 01 5b bd 00 10 26 00	14 01 00 01 00 00 00 00	...[...6.....
0110	00 00 00 00 40 c0 7e	00 01 00 01 00 01 07 1d@.~.....

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.

```

> nslookup www.mit.edu 8.8.8.8
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
www.mit.edu canonical name = www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net canonical name = e9566.dscb.akamaiedge.net.
Name:   e9566.dscb.akamaiedge.net
Address: 23.41.11.175

```

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.

ip.addr==10.0.0.13							Expression...	+
No.	Time	Source	Destination	Protocol	Length	Info		
1	02:44:49.924701	10.0.0.13	8.8.8.8	DNS	71	Standard query 0xf0e1 A www.mit.edu		
2	02:44:50.004261	8.8.8.8	10.0.0.13	DNS	160	Standard query response 0xf0e1 A www.mit.edu CNAME www.mit.edu.edgek...		

▶	Frame 2: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits) on interface 0
▶	Ethernet II, Src: Technico_57:1e:73 (44:32:c8:57:1e:73), Dst: Apple_ef:3c:a0 (78:31:c1:ef:3c:a0)
▶	Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.0.0.13
▶	User Datagram Protocol, Src Port: 53 (53), Dst Port: 55381 (55381)
▼	Domain Name System (response)
	[Request In: 1]
	[Time: 0.079560000 seconds]
	Transaction ID: 0xf0e1
▶	Flags: 0x8180 Standard query response, No error
	Questions: 1
	Answer RRs: 3
	Authority RRs: 0
	Additional RRs: 0
▶	Queries
▼	Answers
▶	www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
▶	www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
▶	e9566.dscb.akamaiedge.net: type A, class IN, addr 23.41.11.175

0000	78 31 c1 ef 3c a0 44 32 c8 57 1e 73 00 00 45 20	x1...D2 .W.s..E
0010	00 92 a8 2d 00 00 35 11 c2 f1 08 00 00 00 0a 00	...5.
0020	00 04 00 35 08 55 00 7e 3a d6 f0 e1 81 00 00 01	...5.U~
0030	00 03 00 00 00 00 03 77 77 77 03 6d 69 74 03 65w ww.mit.e
0040	64 75 00 00 01 00 01 c0 0c 00 05 00 01 00 00 06	du.....
0050	00 00 19 03 77 77 77 03 6d 69 74 03 65 64 75 07	...www. mit.edu..
0060	65 64 67 65 6b 65 79 03 6e 65 74 00 c0 29 00 05	edgekey. net..)...
0070	00 01 00 00 00 3b 00 18 05 65 39 35 36 36 04 64;...e9566.d
0080	73 63 62 0a 61 6b 61 6d 61 69 65 64 67 65 c0 3d	scb.akam aiedge.=
0090	c0 4e 00 01 00 01 00 00 00 13 00 04 17 29 0b af	.N.....).....

Lab 6.0

capture trace

tcp							Expression...	+
No.	Time	Source	Destination	Protocol	Length	Info		
1	11:51:00.093659	10.0.0.13	128.119.245.12	TCP	78	49262 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=481977645 TSecr=0 SACK_PERM=1		
2	11:51:00.209437	128.119.245.12	10.0.0.13	TCP	74	80 → 49262 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=117202896 TSecr...		
3	11:51:00.209530	10.0.0.13	128.119.245.12	TCP	66	49262 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=481977760 TSecr=117202896		
4	11:51:00.209816	10.0.0.13	128.119.245.12	HTTP	1514	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1		
5	11:51:00.209816	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=1449 Ack=1 Win=131744 Len=1448 TSval=481977760 TSecr=117202896		
6	11:51:00.209817	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=2897 Ack=1 Win=131744 Len=1448 TSval=481977760 TSecr=117202896		
7	11:51:00.264068	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=1449 Win=31872 Len=0 TSval=117203005 TSecr=481977760		
8	11:51:00.264169	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=4345 Ack=1 Win=131744 Len=1448 TSval=481977814 TSecr=117203005		
9	11:51:00.264170	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=5793 Ack=1 Win=131744 Len=1448 TSval=481977814 TSecr=117203005		
10	11:51:00.264444	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=2897 Win=34816 Len=0 TSval=117203006 TSecr=481977760		
11	11:51:00.264447	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=4345 Win=37768 Len=0 TSval=117203006 TSecr=481977760		
12	11:51:00.264521	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=7241 Ack=1 Win=131744 Len=1448 TSval=481977814 TSecr=117203006		
13	11:51:00.264523	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=8689 Ack=1 Win=131744 Len=1448 TSval=481977814 TSecr=117203006		
14	11:51:00.264536	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=10137 Ack=1 Win=131744 Len=1448 TSval=481977814 TSecr=117203006		
15	11:51:00.264537	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=11585 Ack=1 Win=131744 Len=1448 TSval=481977814 TSecr=117203006		
16	11:51:00.316031	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=5793 Win=40576 Len=0 TSval=117203058 TSecr=481977814		
17	11:51:00.316126	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=13033 Ack=1 Win=131744 Len=1448 TSval=481977865 TSecr=117203058		
18	11:51:00.316127	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=14481 Ack=1 Win=131744 Len=1448 TSval=481977865 TSecr=117203058		
19	11:51:00.318028	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=7241 Win=43520 Len=0 TSval=117203060 TSecr=481977814		
20	11:51:00.318128	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=15929 Ack=1 Win=131744 Len=1448 TSval=481977866 TSecr=117203060		
21	11:51:00.318129	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=17377 Ack=1 Win=131744 Len=1448 TSval=481977866 TSecr=117203060		
22	11:51:00.318513	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=8689 Win=46336 Len=0 TSval=117203061 TSecr=481977814		
23	11:51:00.318577	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=18825 Ack=1 Win=131744 Len=1448 TSval=481977866 TSecr=117203061		
24	11:51:00.318578	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=20273 Ack=1 Win=131744 Len=1448 TSval=481977866 TSecr=117203061		
25	11:51:00.319943	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=10137 Win=49280 Len=0 TSval=117203063 TSecr=481977814		
26	11:51:00.320036	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=21721 Ack=1 Win=131744 Len=1448 TSval=481977867 TSecr=117203063		
27	11:51:00.320037	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=23169 Ack=1 Win=131744 Len=1448 TSval=481977867 TSecr=117203063		
28	11:51:00.320795	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=11585 Win=52224 Len=0 TSval=117203063 TSecr=481977814		
29	11:51:00.320863	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=24617 Ack=1 Win=131744 Len=1448 TSval=481977867 TSecr=117203063		
30	11:51:00.320863	10.0.0.13	128.119.245.12	TCP	1514	49262 → 80 [ACK] Seq=26065 Ack=1 Win=131744 Len=1448 TSval=481977867 TSecr=117203063		
31	11:51:00.321468	128.119.245.12	10.0.0.13	TCP	66	80 → 49262 [ACK] Seq=1 Ack=13033 Win=55040 Len=0 TSval=117203063 TSecr=481977814		

▶	Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 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: 128.119.245.12
▶	Transmission Control Protocol, Src Port: 49262 (49262), Dst Port: 80 (80), Seq: 0, Len: 0

0000	44 32 c8 57 1e 73 78 31 c1 ef 3c a0 00 00 45 00	D2.W.sx1
0010	00 40 ee 59 40 00 00 06 cc cd 0a 00 00 0d 80 77	..@Y@.w
0020	15 0c c0 6e 00 50 2c 8f 00 60 00 00 00 b0 02	...n.P, .f.....
0030	ff ff 47 d1 00 02 04 05 b4 01 03 03 05 01 01	..G.....
0040	00 0a 1c ba 65 2d 00 00 00 00 04 02 00 00e---

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 = $0.875 * 0.054252 + 0.125 * 0.054628 = 0.115755$

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

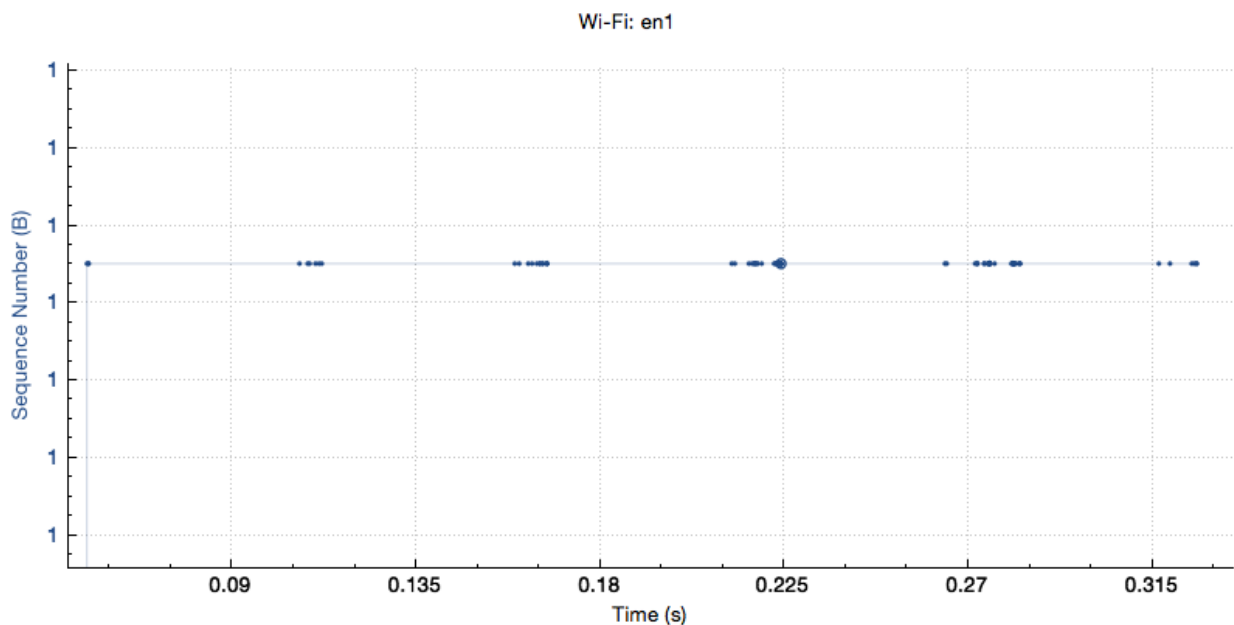
8. They're all 1448 bytes in length (1514 for data).

9. First ACK packet (SYN ACK) advertises `Win = 1460`.

This number grows up to `Win = 46336`. The sender never throttles.

10.

Sequence Numbers (Stevens) for 128.119.245.12:80 → 10.0.0.13:49262



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:

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

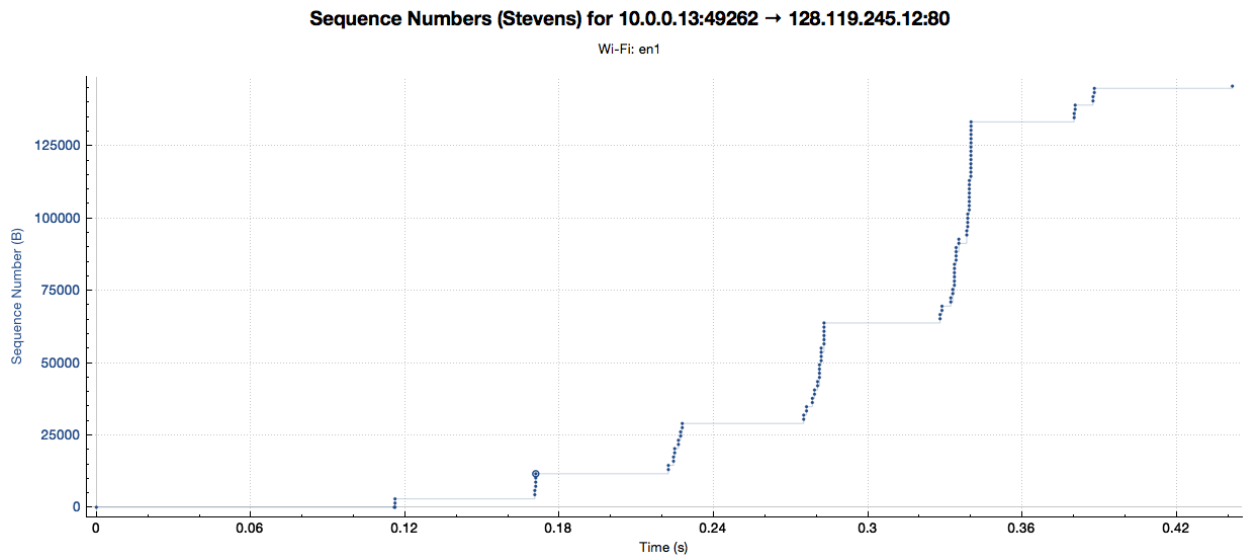
$447151.8744 = 447.15$ Kbytes/seconds

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

$65535 - 8 \text{ bytes (for header)} = 65527$

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.

udp							Expression...	
No.	Time	Source	Destination	Protocol	Length	Info		
6	15:22:35.457368	10.0.0.9	10.0.0.255	UDP	63	64773 → 32414 Len=21		
7	15:22:35.458545	10.0.0.9	10.0.0.255	UDP	63	64770 → 32412 Len=21		
8	15:22:36.079359	10.0.0.13	75.75.75.75	DNS	71	Standard query 0x4808 A espn.go.com		
9	15:22:36.079488	10.0.0.13	75.75.75.75	DNS	71	Standard query 0xf2b8 AAAA espn.go.com		
10	15:22:36.095496	75.75.75.75	10.0.0.13	DNS	110	Standard query response 0x4808 A espn.go.com CNAME e...		
11	15:22:36.120185	75.75.75.75	10.0.0.13	DNS	94	Standard query response 0xf2b8 AAAA espn.go.com CNAM...		
12	15:22:36.120290	10.0.0.13	75.75.75.75	ICMP	70	Destination unreachable (Port unreachable)		
15	15:22:36.500952	10.0.0.13	75.75.75.75	DNS	75	Standard query 0xf33f AAAA espn.gns.go.com		
16	15:22:36.541202	75.75.75.75	10.0.0.13	DNS	75	Standard query response 0xf33f AAAA espn.gns.go.com		
28	15:22:36.756138	10.0.0.13	75.75.75.75	DNS	73	Standard query 0x5849 A a.espncdn.com		
29	15:22:36.756245	10.0.0.13	75.75.75.75	DNS	73	Standard query 0xe5e5 AAAA a.espncdn.com		
30	15:22:36.756582	10.0.0.13	75.75.75.75	DNS	74	Standard query 0x8de3 A a1.espncdn.com		
31	15:22:36.756831	10.0.0.13	75.75.75.75	DNS	74	Standard query 0xa3fd AAAA a1.espncdn.com		
32	15:22:36.757148	10.0.0.13	75.75.75.75	DNS	74	Standard query 0x7dda A a2.espncdn.com		
33	15:22:36.757217	10.0.0.13	75.75.75.75	DNS	74	Standard query 0xb222 AAAA a2.espncdn.com		
34	15:22:36.757545	10.0.0.13	75.75.75.75	DNS	74	Standard query 0x1156 A a3.espncdn.com		
35	15:22:36.757623	10.0.0.13	75.75.75.75	DNS	74	Standard query 0x0576 AAAA a3.espncdn.com		
36	15:22:36.757826	10.0.0.13	75.75.75.75	DNS	74	Standard query 0x90e6 A a4.espncdn.com		
37	15:22:36.758058	10.0.0.13	75.75.75.75	DNS	74	Standard query 0xbf9a AAAA a4.espncdn.com		
38	15:22:36.758427	10.0.0.13	75.75.75.75	DNS	73	Standard query 0x1456 A tredir.go.com		
39	15:22:36.758512	10.0.0.13	75.75.75.75	DNS	73	Standard query 0xec28 AAAA tredir.go.com		
40	15:22:36.768703	75.75.75.75	10.0.0.13	DNS	176	Standard query response 0x5849 A a.espncdn.com CNAME...		
▶ Frame 8: 71 bytes on wire (568 bits), 71 bytes captured (568 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: 50251 (50251), Dst Port: 53 (53)								
Source Port: 50251								
Destination Port: 53								
Length: 37								
▶ Checksum: 0x8f5f [validation disabled]								
[Stream index: 2]								
0000	44 32 c8 57 1e 73 78 31	c1 ef 3c a0 00 00 45 00	D2.W.sx1 ..<...E.					
0010	00 39 af 74 00 00 ff 11	6b 9c 0a 00 00 0d 4b 4b	.9.t.... k.....KK					
0020	4b 4b c4 4b 00 35 00 25	8f 5f 48 08 01 00 00 01	KK.K.5.% ..H....					
0030	00 00 00 00 00 04 65	73 70 6e 02 67 6f 03 63e spn.go.c					
0040	6f 6d 00 00 01 00 01		om.....					

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.