THE HONG KONG POLYTECHNIC UNIVERSITY

Department of Computing

This is an open-book examination.

(COMP5311)

Internet Infrastructure and Protocols

13 December 2010 3.5 hours

[Answer at most 7 questions in section A and both questions in section B.]

Turn Over

Section A: Please answer AT MOST SEVEN questions in this section [8 marks each, making up a total of 56 marks out of 100]. You should always attach a succinct explanation to your answer.

1. A certain class B network `160.10.0.0/16` is subnetted according to the following structure. Note that variable length subnet mask is used and the subnetting is done with a certain pattern.

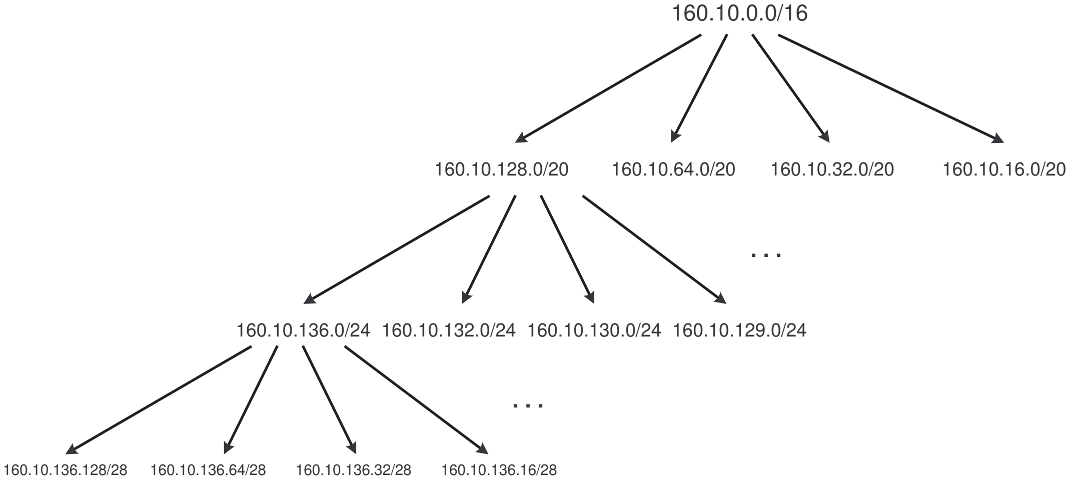


Figure 1: The subnet structure of `160.10.0.0/16`.

   (a) (3 marks) Which subnet corresponds to the subnet-directed broadcast address `160.10.47.255`?

   (b) (3 marks) Which subnet corresponds to the subnet-directed broadcast address `160.10.130.31`?

   (c) (2 marks) Is `160.10.127.255` a subnet-directed broadcast address in this class B network?

2. Consider that an IP packet of 1500 bytes (with destination address as D and a 20-byte IP header) entering into a router R, and the packet will be tunneled. The tunnel MTU is assumed to be 1480 bytes. Therefore, IP fragmentation is required for this packet. In this question, we explore a fragment-first-and-then-tunnel approach. Using this approach, the IP packet will be first fragmented in packets not exceeding the tunnel MTU, and then each fragment is tunneled by encapsulating the fragment with a 20-byte IP header.

   (a) (3 marks) What is the size of the second fragment?

   (b) (3 marks) If there is no further fragmentation, where will the fragments be reassembled? At the exit tunnel endpoint or D?

(c) (2 marks) If the fragments are further fragmented in the path between the two tunnel endpoints, where will these fragments of fragments be reassembled? Assume that there is no other IP tunnel in the path.

3. In a ping of death attack, an attacker sends out an IP datagram that exceeds the maximum size of an IP datagram (65,535 bytes). Since this attack packet will be fragmented, the victim would not know the actual size of the original packet until it attempts to reassemble the fragments. However, some systems do not know how to handle an IP datagram of more than the maximum size; as a result, the attack causes them to hang and to reboot.

   The following is a tcpdump trace of such an attack. The fragmentation is performed according to an MTU of 1500 bytes, and the IP headers do not contain options.

```
17:26:11.013622 cslwin95 > arkroyal: icmp: echo request (frag 6144:1480@0+)
17:26:11.015079 cslwin95 > arkroyal: (frag 6144:1480@1480+)
17:26:11.016637 cslwin95 > arkroyal: (frag 6144:1480@2960+)
17:26:11.017577 cslwin95 > arkroyal: (frag 6144:1480@4440+)
17:26:11.018833 cslwin95 > arkroyal: (frag 6144:1480@5920+)
17:26:11.020112 cslwin95 > arkroyal: (frag 6144:1480@7400+)
17:26:11.021346 cslwin95 > arkroyal: (frag 6144:1480@8880+)
17:26:11.022641 cslwin95 > arkroyal: (frag 6144:1480@10360+)
17:26:11.023869 cslwin95 > arkroyal: (frag 6144:1480@11840+)
17:26:11.025140 cslwin95 > arkroyal: (frag 6144:1480@13320+)
17:26:11.026604 cslwin95 > arkroyal: (frag 6144:1480@14800+)
17:26:11.027628 cslwin95 > arkroyal: (frag 6144:1480@16280+)
17:26:11.028871 cslwin95 > arkroyal: (frag 6144:1480@17760+)
17:26:11.030100 cslwin95 > arkroyal: (frag 6144:1480@19240+)
17:26:11.031307 cslwin95 > arkroyal: (frag 6144:1480@20720+)
17:26:11.032542 cslwin95 > arkroyal: (frag 6144:1480@22200+)
17:26:11.033774 cslwin95 > arkroyal: (frag 6144:1480@23680+)
17:26:11.035018 cslwin95 > arkroyal: (frag 6144:1480@25160+)
17:26:11.036576 cslwin95 > arkroyal: (frag 6144:1480@26640+)
17:26:11.037464 cslwin95 > arkroyal: (frag 6144:1480@28120+)
17:26:11.038696 cslwin95 > arkroyal: (frag 6144:1480@29600+)
17:26:11.039966 cslwin95 > arkroyal: (frag 6144:1480@31080+)
17:26:11.041218 cslwin95 > arkroyal: (frag 6144:1480@32560+)
17:26:11.042579 cslwin95 > arkroyal: (frag 6144:1480@34040+)
17:26:11.043807 cslwin95 > arkroyal: (frag 6144:1480@35520+)
17:26:11.046276 cslwin95 > arkroyal: (frag 6144:1480@37000+)
17:26:11.047236 cslwin95 > arkroyal: (frag 6144:1480@38480+)
17:26:11.048478 cslwin95 > arkroyal: (frag 6144:1480@39960+)
17:26:11.049698 cslwin95 > arkroyal: (frag 6144:1480@41440+)
17:26:11.050929 cslwin95 > arkroyal: (frag 6144:1480@42920+)
17:26:11.052164 cslwin95 > arkroyal: (frag 6144:1480@44400+)
17:26:11.053398 cslwin95 > arkroyal: (frag 6144:1480@45880+)
17:26:11.054685 cslwin95 > arkroyal: (frag 6144:1480@47360+)
17:26:11.056347 cslwin95 > arkroyal: (frag 6144:1480@48840+)
17:26:11.057313 cslwin95 > arkroyal: (frag 6144:1480@50320+)
17:26:11.058357 cslwin95 > arkroyal: (frag 6144:1480@51800+)
17:26:11.059588 cslwin95 > arkroyal: (frag 6144:1480@53280+)
17:26:11.060787 cslwin95 > arkroyal: (frag 6144:1480@54760+)
17:26:11.062023 cslwin95 > arkroyal: (frag 6144:1480@56240+)
17:26:11.063247 cslwin95 > arkroyal: (frag 6144:1480@57720+)
17:26:11.064479 cslwin95 > arkroyal: (frag 6144:1480@59200+)
17:26:11.066252 cslwin95 > arkroyal: (frag 6144:1480@60680+)
```

```
17:26:11.066957 cslwin95 > arkroyal: (frag 6144:1480@62160+)
17:26:11.068220 cslwin95 > arkroyal: (frag 6144:1480@63640+)
17:26:11.069107 cslwin95 > arkroyal: (frag 6144:398@65120)
```

Answer the following questions concerning the trace.

(a) (1 mark) What does the number immediately before ”@” represent?

(b) (1 mark) What does the number immediately after ”@” represent?

(c) (1 mark) What does the ”+” symbol represent?

(d) (1 mark) What is the number 6144 referred to?

(e) (4 marks) What is the size of the original IP packet before fragmentation?

4. Consider a class B network `158.132.0.0` which is subnetted with a subnet mask of `255.255.255.0`. Moreover, a host with IP address `158.132.1.1` is compromised in that an attack program was installed in that machine. Discuss the effect of the attack if the attack program in that machine sends out an ICMP echo request message (ping) with the following source and destination addresses.

   (a) (3 marks) Source address = `158.132.1.100` and destination address = `255.255.255.255`.

   (b) (3 marks) Source address = `158.132.2.100` and destination address = `255.255.255.255`.

   (c) (2 marks) Source address = `158.132.2.100` and destination address = `158.132.2.255`.

   Note that the source addresses in ping messages are spoofed (i.e., not equal to `158.132.1.1`). Assume the followings:

   - All routers inside the network turn off the support for subnet-directed and all-subnet-directed IP broadcasts, i.e., dropping those packets.
   - All nodes (hosts and routers) must reply with an ICMP echo reply message when receiving an ICMP echo request message.
   - When forwarding a packet, a router only examines the destination IP address, but not the source IP address.
   - The destination addresses of the ping messages belong to hosts, but not to routers.

5. An IP packet with a total length of 3000 bytes and a header length of 20 bytes is to be fragmented into 1500-byte IP packets.

   (a) (3 marks) Write down the total length, the offset values, and the M-bit value in each fragment.

   (b) (2 marks) If the total length is changed to 3100 bytes, how would your answers be different from (a)?

   (c) (3 marks) If the header length becomes 40 bytes (and the IP packet's total length remains 3000 bytes), how would your answers be different from (a)?

6. When a TCP node receives a valid TCP data segment from the other side of a TCP connection, it has to check, among others, whether the data sent in the segment has previously been received. If positive, it will discard the duplicate data. Let SEQ be the sequence number in the TCP data packet and LEN be the length of the TCP packet's payload.

   (a) (2 marks) Give the condition in terms of the state variables kept by the receiver (e.g., `rcv_nxt` and `rcv_wnd`) that at least some data in the TCP packet are duplicate.

   (b) (3 marks) Give the condition in terms of the state variables kept by the receiver (e.g., `rcv_nxt` and `rcv_wnd`) and LEN that the entire payload of the TCP packet is duplicate.

   (c) (3 marks) If the entire payload is a duplicate, it is essential for the receiver to send an ACK to the sender before dropping the data. Explain why this is necessary.

7. Figure 2 shows a Wireshark trace of a web session: from a client with port 65416 to a server with port 80. Assume that both sides use an initial sequence number (SN) of 0. Therefore, the SN and the acknowledgment number (AN) in the SYN-ACK packet are 0 and 1, respectively. The SN and AN in the third hand-shaking packet are both 1.

   Answer the following questions concerning Figure 2 with succinct explanation.

   (a) (2 marks) What is the server's `snd_nxt` just after sending the SYN-ACK packet at time 1.282?

   (b) (2 marks) What is the client's `snd_nxt` just after sending the data packet at time 1.283 (i.e., the fourth packet in Figure 2)?

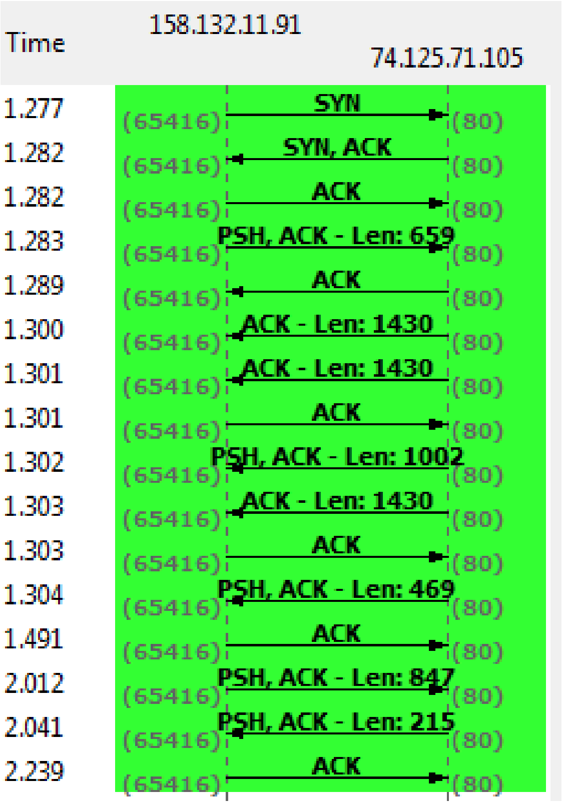
   (c) (2 marks) What is the client's `rcv_nxt` just after receiving the data packet from the server at time 1.301 (i.e., the seventh packet in Figure 2)?

   (d) (2 marks) What is the AN in the last packet in Figure 2?

8. Consider the IP network in Figure 3 that is subnetted with a fixed-length subnet mask. The numbers next to the LAN segments indicate their subnet numbers. The routers use RIP-I to share the routing information with split horizon and poisonous reverse. A hop count of 16 is used to represent infinity.

   A hacker gets hold of $R2$ and sends out a false route for subnet 8. Specifically, $R2$ sends out the following distance vector on subnet 4 and a similar distance vector on subnet 2.

Destination (subnet number)	Number of hops
1	1
2	1
3	2
4	(not included)
5	16
6	16
7	16
8	1 (a false value)



Time	158.132.11.91		74.125.71.105
1.277	(65416)	SYN →	(80)
1.282	(65416)	← SYN, ACK	(80)
1.282	(65416)	ACK →	(80)
1.283	(65416)	PSH, ACK - Len: 659 →	(80)
1.289	(65416)	← ACK	(80)
1.300	(65416)	← ACK - Len: 1430	(80)
1.301	(65416)	← ACK - Len: 1430	(80)
1.301	(65416)	ACK →	(80)
1.302	(65416)	← PSH, ACK - Len: 1002	(80)
1.303	(65416)	← ACK - Len: 1430	(80)
1.303	(65416)	ACK →	(80)
1.304	(65416)	← PSH, ACK - Len: 469	(80)
1.491	(65416)	ACK →	(80)
2.012	(65416)	PSH, ACK - Len: 847 →	(80)
2.041	(65416)	← PSH, ACK - Len: 215	(80)
2.239	(65416)	ACK →	(80)

Figure 2: TCP transmissions in a web session.

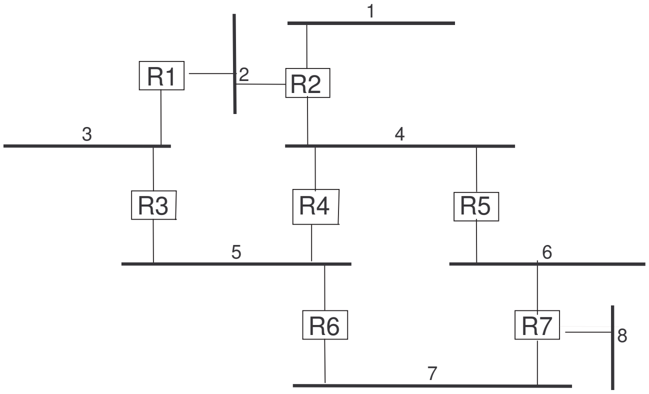


Figure 3: An IP network running RIP-1.

Which subnets (1-7) will be affected by this false route when the hosts on these subnets send packets to a destination on subnet 8, and why?

9. Consider case (a) in Figure 4. Routers A to D use a distance vector routing protocol with source tracing capability. You may use the hop count as the routing metric. Shortly after the routing protocol converges, the link $A - B$ breaks, describe how this routing protocol prevents routing loops from forming. Repeat it for case (b).

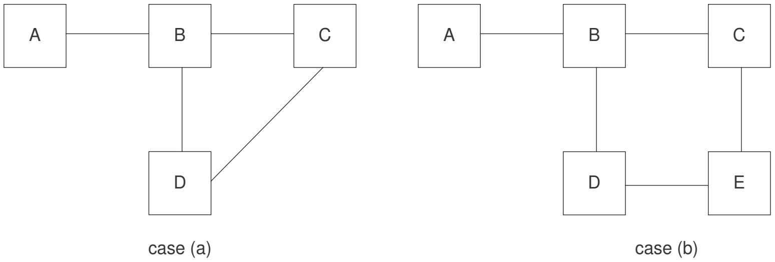


Figure 4: Two IP networks running a distance vector routing protocol with source tracing capability.

Section B: Please answer both questions in this section. Each question carries 22 marks.

10. To verify the answers to question 2 of assignment 2, we configured the hosts $H1$ (`Ibm_b3:f0:cf`) and $H2$ (`Ibm_16:8b:06`) with different subnet addresses, and the hosts communicated with each other through router R (`Tp-LinkT_c9:41:05`), as depicted in Figure 5. Their configurations and forwarding tables were:

- R (`eth1.0`): `123.123.1.10`; (`eth1.1`): `123.123.2.10`; MAC Addr: `00:21:27:C9:41:05`
- $H1$ (`eth0`): `123.123.2.1/24`; gateway `123.123.2.10`; MAC Addr: `00:11:25:B3:F0:CF`
- $H2$ (`eth0`): `123.123.1.1/24`; gateway `123.123.1.10`; MAC Addr: `00:11:25:16:8B:06`.

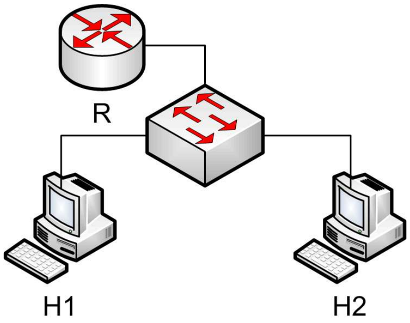


Figure 5: Router R connected to hosts $H1$ and $H2$ through a switch.

$H1$'s routing table:

Destination	Gateway	Net mask	Interface
123.123.2.0	123.123.2.1	255.255.255.0	eth0
0.0.0.0	123.123.2.10	0.0.0.0	eth0

$H2$'s routing table:

Destination	Gateway	Net mask	Interface
123.123.1.0	123.123.1.1	255.255.255.0	eth0
0.0.0.0	123.123.1.10	0.0.0.0	eth0

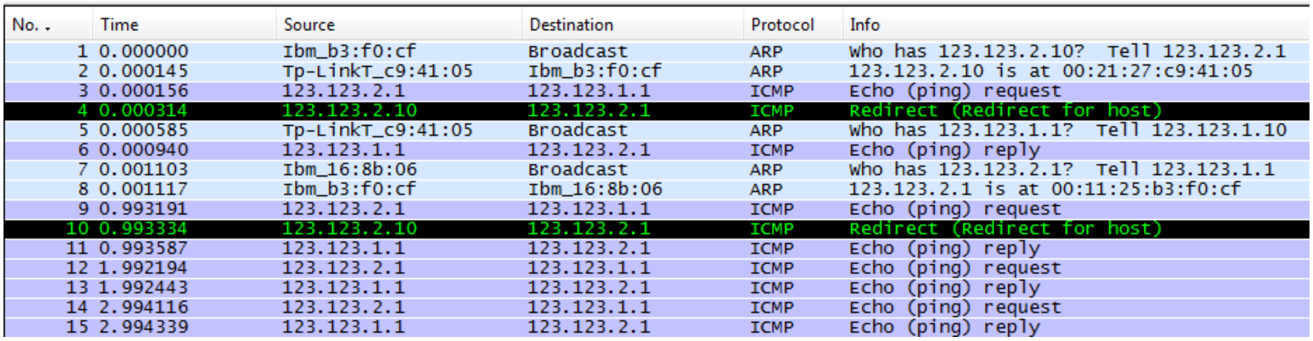
R's routing table:

Destination	Gateway	Net mask	Interface
123.123.2.0	123.123.2.10	255.255.255.0	eth1
123.123.1.0	123.123.1.10	255.255.255.0	eth1

Moreover, their ARP caches were initially empty. Then $H1$ sent ping requests (ICMP echo requests) to $H2$. The following list records what actually happened.

- $H1$ broadcasted an ARP request for the MAC addr of `123.123.2.10`.
- R replied the ARP request with its MAC addr (`00:21:27:C9:41:05`).
- $H1$ sent out a ping request to $H2$ (`123.123.1.1`).
- R received the ping request, sent an ICMP Redirect message to $H1$, and forwarded the ping request to $H2$.
- $H2$ replied the ping request by sending an ICMP echo reply to R. Similar to before, R sent an ICMP Redirect message to $H2$.
- $H2$ broadcasted an ARP request for the MAC address of `123.123.2.1`, and $H1$ replied.
- $H1$ sent a second ping request to $H2$ via R, and ICMP Redirect messages were again issued by R to $H1$ and $H2$.
- Starting from the third ping request and on, $H1$ and $H2$ were able to exchange ping requests and replies directly, without going through R.

Figure 6 shows the Wireshark capture at $H1$ with the promiscuous mode off (i.e., captures only packets destined to $H1$ and sent by $H1$). Note that since this trace was captured at $H1$, it does not contain the communication between R and $H2$.



No.	Time	Source	Destination	Protocol	Info
1	0.000000	Ibm_b3:f0:cf	Broadcast	ARP	Who has 123.123.2.10? Tell 123.123.2.1
2	0.000145	Tp-LinkT_c9:41:05	Ibm_b3:f0:cf	ARP	123.123.2.10 is at 00:21:27:c9:41:05
3	0.000156	123.123.2.1	123.123.1.1	ICMP	Echo (ping) request
4	0.000314	123.123.2.10	123.123.2.1	ICMP	Redirect (Redirect for host)
5	0.000585	Tp-LinkT_c9:41:05	Broadcast	ARP	Who has 123.123.1.1? Tell 123.123.1.10
6	0.000940	123.123.1.1	123.123.2.1	ICMP	Echo (ping) reply
7	0.001103	Ibm_16:8b:06	Broadcast	ARP	Who has 123.123.2.1? Tell 123.123.1.1
8	0.001117	Ibm_b3:f0:cf	Ibm_16:8b:06	ARP	123.123.2.1 is at 00:11:25:b3:f0:cf
9	0.993191	123.123.2.1	123.123.1.1	ICMP	Echo (ping) request
10	0.993334	123.123.2.10	123.123.2.1	ICMP	Redirect (Redirect for host)
11	0.993587	123.123.1.1	123.123.2.1	ICMP	Echo (ping) reply
12	1.992194	123.123.2.1	123.123.1.1	ICMP	Echo (ping) request
13	1.992443	123.123.1.1	123.123.2.1	ICMP	Echo (ping) reply
14	2.994116	123.123.2.1	123.123.1.1	ICMP	Echo (ping) request
15	2.994339	123.123.1.1	123.123.2.1	ICMP	Echo (ping) reply

Figure 6: A partial Wireshark capture of the packets when $H1$ pings $H2$ at $H1$ with the promiscuous mode off.

Given the information above, answer the questions below with succinct explanation.

(a) (2 marks) What is the destination MAC address in the frame of packet 3 in the Wireshark trace?

(b) (2 marks) What is the value in the “Gateway address” field in the ICMP Redirect message (i.e., packets 4 and 10 in the Wireshark trace)?

(c) (2 marks) How many ICMP header(s) is (are) included in the IP packet that carries the ICMP Redirect message?

(d) (2 marks) The Wireshark at $H1$ could capture R's ARP requests for $H2$'s MAC address but not the ARP reply. What is the reason for that?

(e) (2 marks) $H1$ could send ping requests directly to $H2$ without first sending out an ARP request for $H2$'s MAC address. What is the reason for that?

(f) (4 marks) If we examine the Wireshark trace captured at R with the promiscuous mode off, which packets in Figure 6 that will *not* show up in the trace?

(g) (4 marks) If we examine the Wireshark trace captured at $H2$ with the promiscuous mode off, which packets in Figure 6 that will *not* show up in the trace?

(h) (2 marks) Are the ICMP headers of packets 3 and 12 in Figure 6 identical?

(i) (2 marks) If the router's two interfaces `eth1.0` and `eth1.1` are now configured with two separate physical interfaces each of which is connected directly to a host, will R still send the ICMP Redirect messages? Note that the routing tables remain the same.

11. Figure 7 shows a similar plot (which was generated using a different simulator) as problem 1 of assignment 3. In this trace, we label the lost data segment (the one with a cross) to be segment 14, and other data segments are labeled consecutively. The only main difference as compared with the plot in the assignment is that the ACK values here refer to the next expected segment. Recall that ack-every-segment strategy is used, and assume that the `rwnd` value is always equivalent to 30 data segments.

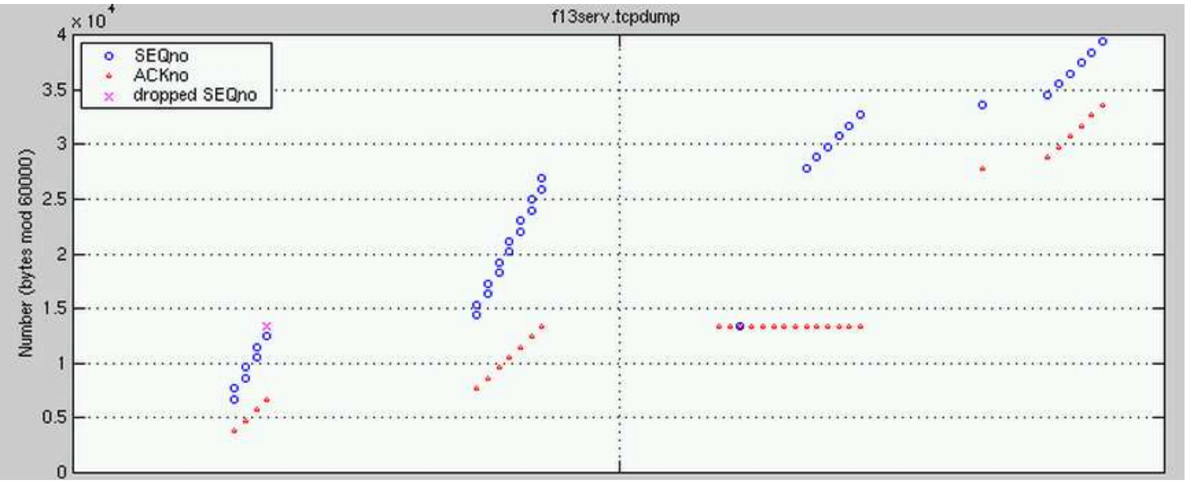


Figure 7: TCP segment 14 is lost and is being fast retransmitted.

Answer the following questions concerning the trace in Figure 7 with succinct explanation. You do not need to know the `ssthresh` values to answer these questions.

(a) (2 marks) What is the value of `cwnd` by the time of transmitting data segment 14?

(b) (2 marks) Why is there an idle period between sending segment 14 and segment 15?

(c) (2 marks) What is the value of `cwnd` by the time of transmitting data segment 28?

(d) (2 marks) Why is there an idle period between sending segment 28 and segment 29?

(e) (2 marks) What is the value of `cwnd` by the time of transmitting data segment 29? Explain your answer *without* using `ssthresh`.

(f) (2 marks) Why is there an idle period between sending segment 34 and segment 35?

(g) (2 marks) What is the value of `cwnd` by the time of transmitting data segment 35? Explain your answer *without* using `ssthresh`.

(h) (2 marks) Assume that just after sending data segment 28, `snd_una` $= n$, and the MSS is given by m bytes. What are the values of `snd_max` and `snd_nxt` in terms of m and n?

(i) (2 marks) By the time of retransmitting segment 14, what are the values of `snd_max` and `snd_nxt` in terms of m and n?

(j) (4 marks) If the `rwnd` value is changed to 15 segments, what would be different in the plot?

— End of the Examination Paper —