Internet Infrastructure and Protocols (COMP5311)

Assignment Three (due on 30 Nov. 2011)
Each question carries 8 marks, unless stated otherwise.

Rocky K. C. Chang

1) Figure 1 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.

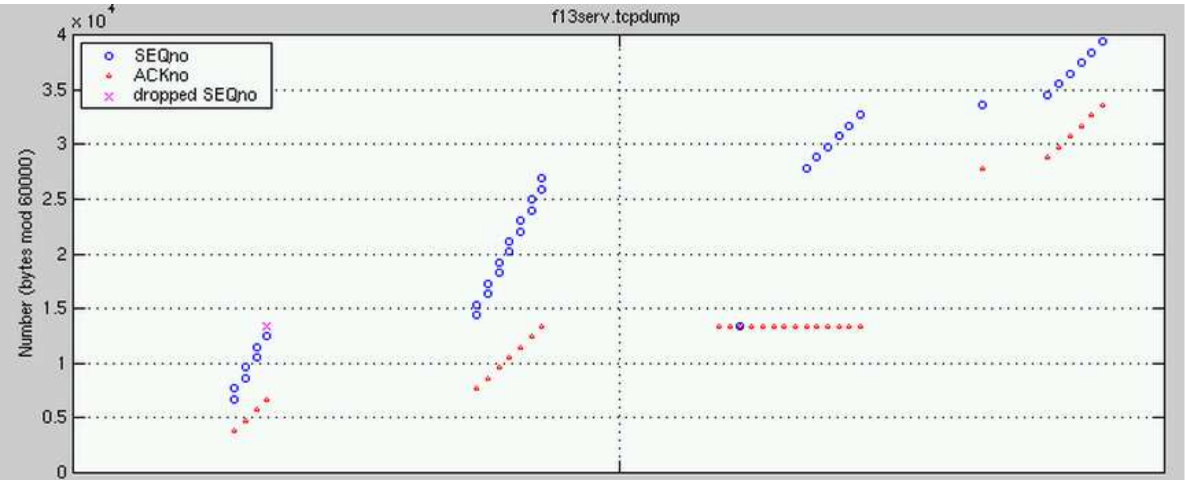


Fig. 1. TCP segment 14 is lost and is being fast retransmitted.

Answer the following questions concerning the trace in Figure 1 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?

Solutions:

a) (2 marks) By the time of transmitting data segment 14, the `cwnd` is eight segments, because there are a total of eight segments sent without receiving any ACK. Note that the first ACK acknowledged all the previous data.

b) (2 marks) The send window is full, and the sender waits for a new ACK for sending new data.

c) (2 marks) By the time of transmitting data segment 28, the `cwnd` is 15 segments, because 15 segments have been sent, including the lost segment 14.

d) (2 marks) Same reason as (b): the send window is full, and the sender waits for a new ACK for sending new data.

e) (2 marks) By the time of transmitting data segment 29 the `cwnd` is 16 segments, because 15 segments are still buffered.

f) (2 marks) There were no more duplicate ACKs to allow the sender to send new data.

g) (2 marks) By the time of transmitting data segment 35, the `cwnd` is 7, because six segments (29-34) are still buffered.

h) (2 marks) `snd_max` = `snd_nxt` = $n + 15m$, because the snd_wnd is 15 at that time.

i) (2 marks) Same as the last part, `snd_max` = $n + 15m$, but `snd_nxt` = n which is the number of the first byte in segment 14.

j) (4 marks) If the `rwnd` value is changed to 15 segments, segments 29-34 could not be sent during the fast recovery phase, because `snd_wnd` = $\min\{\texttt{cwnd}, \texttt{rwnd}\}$ = 15. When a new ACK arrives, `cwnd` is reduced to seven segments. Since all the outstanding segments have been acknowledged, the sender will send segments 29-35 at the same time.

2) Consider the IP network in Figure 2 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.

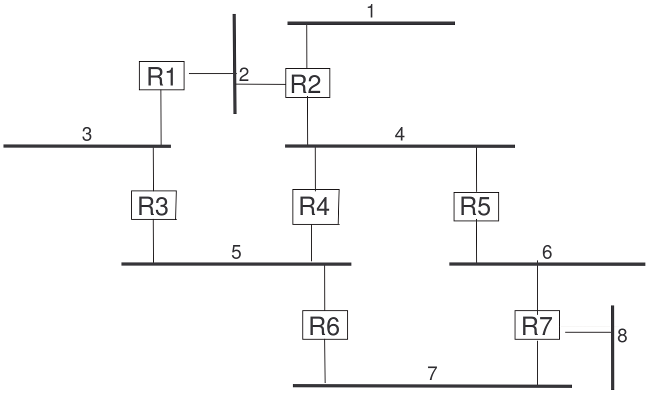


Fig. 2. An IP network running RIP-1.

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)

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?

Solutions: Subnets 1-4 will definitely be affected, because they have shorter distance to $R2$ than to $R7$. On the other hand, both subnets 6 and 7 still find $R7$'s route shorter because both $R5$ and $R6$ advertise routes to subnet 8 with a cost of at least 2. On subnet 5, the situation depends on whether $R4$ or $R6$ is the default router. If $R4$ is the default router, the packet will be sent to $R2$ (affected); otherwise, $R7$.

3) Consider case (a) in Figure 3. 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).

Solutions: With the additional “last router” information in the distance vectors, routers C and D maintain the same distance vector, as depicted in Figure 4. Therefore, when $A-B$ link breaks, B, who knows that the last-hop router from the distance vectors of C and D is itself, does not use C and D as the next hop, thus preventing routing loops from forming. The case (b) is similar. B always finds the last hop equal to its identity; therefore, it wouldn't use them.

4) To verify the answers to question 2 of assignment 2, we configured the hosts $H1$ (`Ibm_b3:f0:cf`)

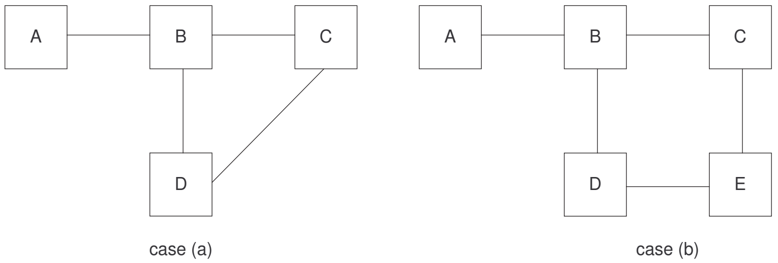


Fig. 3. Two IP networks running a distance vector routing protocol with source tracing capability.

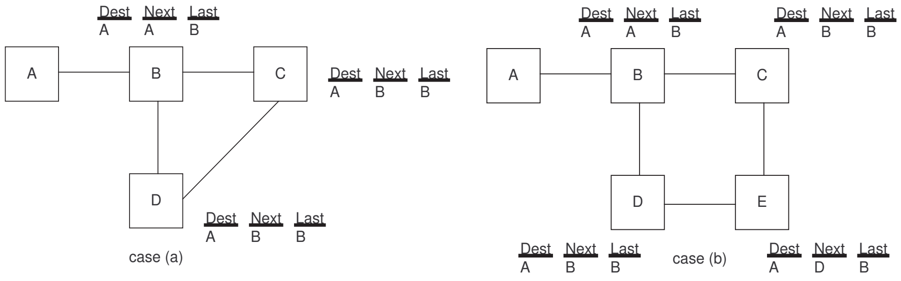


Fig. 4. Two IP networks running a distance vector routing protocol with source tracing capability.

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`.

$H1$'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.

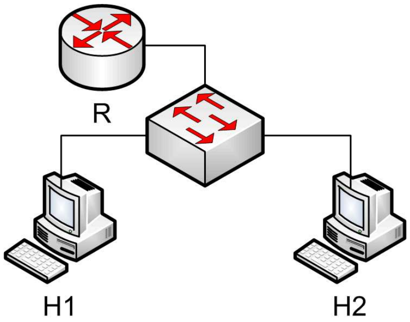


Fig. 5. Router R connected to hosts $H1$ and $H2$ through a switch.

- $H1$ 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 $H2$.

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?

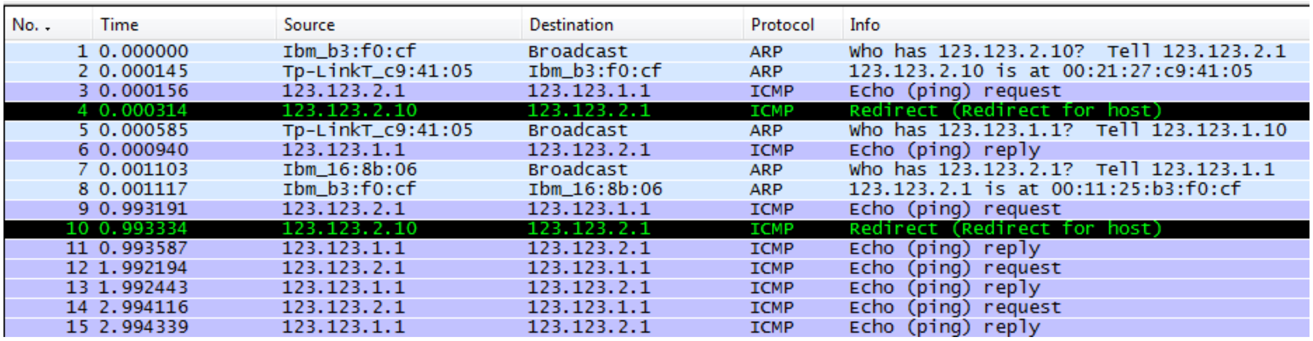
b) (2 marks) What is the value in the “Gateway address” field in the ICMP Redirect message (i.e., packets 4 and 10 in the Wireshark trace)?

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



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

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

packets in Figure 6 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 will *not* show up in the trace?

h) (2 marks) Are the ICMP headers in 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.

Solutions:

a) (2 marks) The destination MAC address is R's address (`00:21:27:C9:41:05`), because $H1$ forwards the IP packet to R.

b) (2 marks) The "Gateway address" field in the ICMP Redirect message is $H2$'s IP address (123.123.1.1), because R discovers that $H1$ could forward the packet directly to $H2$.

c) (2 marks) Two: one for the ICMP Redirect message and the other for the ICMP echo message which is included in the error message.

d) (2 marks) The Wireshark at $H1$ could capture R's ARP requests for $H2$'s MAC, because it was sent in data-link broadcast. But it did not receive the ARP reply, because it was sent to R in unicast.

e) (2 marks) $H1$ already learned $H2$'s MAC address from $H2$'s ARP request for $H1$'s MAC address in the seventh frame.

f) (4 marks) Frames 8, 11-15 will not show up in R's Wireshark capture, because they are sent directly between $H1$ and $H2$.

g) (4 marks) Frames 2-4, 6, 9-10 will not show up in $H2$'s Wireshark capture, because they are sent directly between $H1$ and R.

h) (2 marks) The two ICMP headers are not identical, because they include different identifiers to distinguish different echo replies.

i) (2 marks) No, because different physical interfaces are used for receiving and forwarding the same packet.