Computer Networks Fall 2016
Problem Sheet #3

Tom Wiesing

October 12, 2016

1 Problem 1

1.1 1a)

We assign **R1** IP address 198.51.100.1 on interface en1.1 and 203.0.113.1 on en1.2. Furthermore we configure the following forwarding table:

Prefix	Next Hop	Interface
198.51.100.3/32	198.51.100.3	en1.1
203.0.113.4/32	203.0.113.4	en1.2

We assign **R2** IP address 198.51.100.2 on interface en2.1 and 203.0.113.2 on en2.2. Furthermore we configure the following forwarding table:

Prefix	Next Hop	Interface
198.51.100.3/32	198.51.100.3	en2.1
203.0.113.4/32	203.0.113.4	en2.2

1.2 1b)

In this scenario the following messages will be sent:

1. A sends a TCP SYN from A to B.
   - A makes an ARP query for the R1 ip adresss on en1.1 (198.51.100.1)
   - R1 responds with an ARP response mac($en2.1$)
   - R1 makes an ARP request to the B ip adress on enB.0 (203.0.113.4)
   - B responds with an ARP response mac($enB.0$)
2. B sends a TCP SYN-ACK from B to A.
   - B makes an ARP request for the R2 ip adresss on en2.2 (203.0.113.2)
   - R2 responds with an ARP response mac($en2.2$)
   - R2 makes an ARP request to the A ip adress on enA.0 (198.51.100.3)
   - A responds with an ARP response mac($enA.0$)
3. A now sends a TCP ACK from A to B.
4. This procedure can then repeat.

no	segments	eth-src	eth-dst	ip-src	ip-dest	description
1	A, B, ...	mac($ena.0$)	mac()	ip($ena.0$)	ip($en1.1$)	ARP request ip($en1.1$)
2	B, A, ...	mac($en1.1$)	mac($ena.0$)	ip($en1.1$)	ip($ena.0$)	ARP response ip($en1.1$)
3	A, B	mac($ena.0$)	mac($enb.0$)	ip(ena.0)	ip($en1.1$)	TCP SYN A $\rightarrow$ B (step 1)
4	D, F, ...	mac($en1.2$)	mac()	ip($en1.1$)	ip($enb.0$)	ARP request ip($enb.0$))
5	F, D, ...	mac($enb.0$)	mac($en1.2$)	ip($enb.0$)	ip($en1.1$)	ARP response ip($enb.0$))
6	D, F	mac($en1.2$)	mac($enb.0$)	ip($ena.0$)	ip($enb.0$)	TCP SYN A $\rightarrow$ B (step 2)
7	F, E, ...	mac($enb.0$)	mac()	ip($enb.0$)	ip($en2.2$)	ARP request ip($en2.2$)
8	E, F, ...	mac($en2.2$)	mac($enb.0$)	ip($en2.2$)	ip($enb.0$)	ARP response ip($en2.2$)
9	E, F	mac($enb.0$)	mac($en2.2$)	ip($enb.0$)	ip($ena.0$)	TCP SYN-ACK B $\rightarrow$ A (step 1)
10	C, A, ...	mac($en2.1$)	mac()	ip($en2.1$)	ip($ena.0$)	ARP request ip($ena.0$)
11	A, C, ...	mac($ena.0$)	mac($en2.1$)	ip($ena.0$)	ip($en2.1$)	ARP response ip($ena.0$)
12	C, A	mac($en2.1$)	mac($ena.0$)	ip($enb.0$)	ip($ena.0$)	TCP SYN-ACK B $\rightarrow$ A (step 2)
13	A, B	mac($ena.0$)	mac($en1.1$)	ip($ena.0$)	ip($enb.0$)	TCP ACK A $\rightarrow$ B (step 1)
14	D, F	mac($en1.2$)	mac($enb.0$)	ip($ena.0$)	ip($enb.0$)	TCP ACK A $\rightarrow$ B (step 2)

1.3 1c)

Benefits:

- it is easy to configure
- traffic between A and B and B and A goes via different routes $\rightarrow$ each segment has to handle less traffic

Problems:

- The routes can **only** reach A and B. If more nodes are connected, the network has to be reconfigured.

2 Problem 2

2.1 2a)

The longest prefix length in the tables is 2, so it is sufficient to check all prefixes of length 2 to check equivalence of F_1 and F_2. In F_1, we have

- $00* \rightarrow R_2$
- $01* \rightarrow R_1$
- $10* \rightarrow R_2$
- $11* \rightarrow R_3$

and in F_2 we have

- $00* \rightarrow R_2$
- $01* \rightarrow R_1$
- $10* \rightarrow R_2$
- $11* \rightarrow R_3$

Thus they are equivalent.

2.2 2b)

There is no equivalent forwarding forwarding table with less than four entries. As there are three different hosts involved in the routes, at least 3 entries are needed. Thus if there is a table with exactly 3 entires. But it is not possible to combine the two entries for R_2.