Homework Q & A

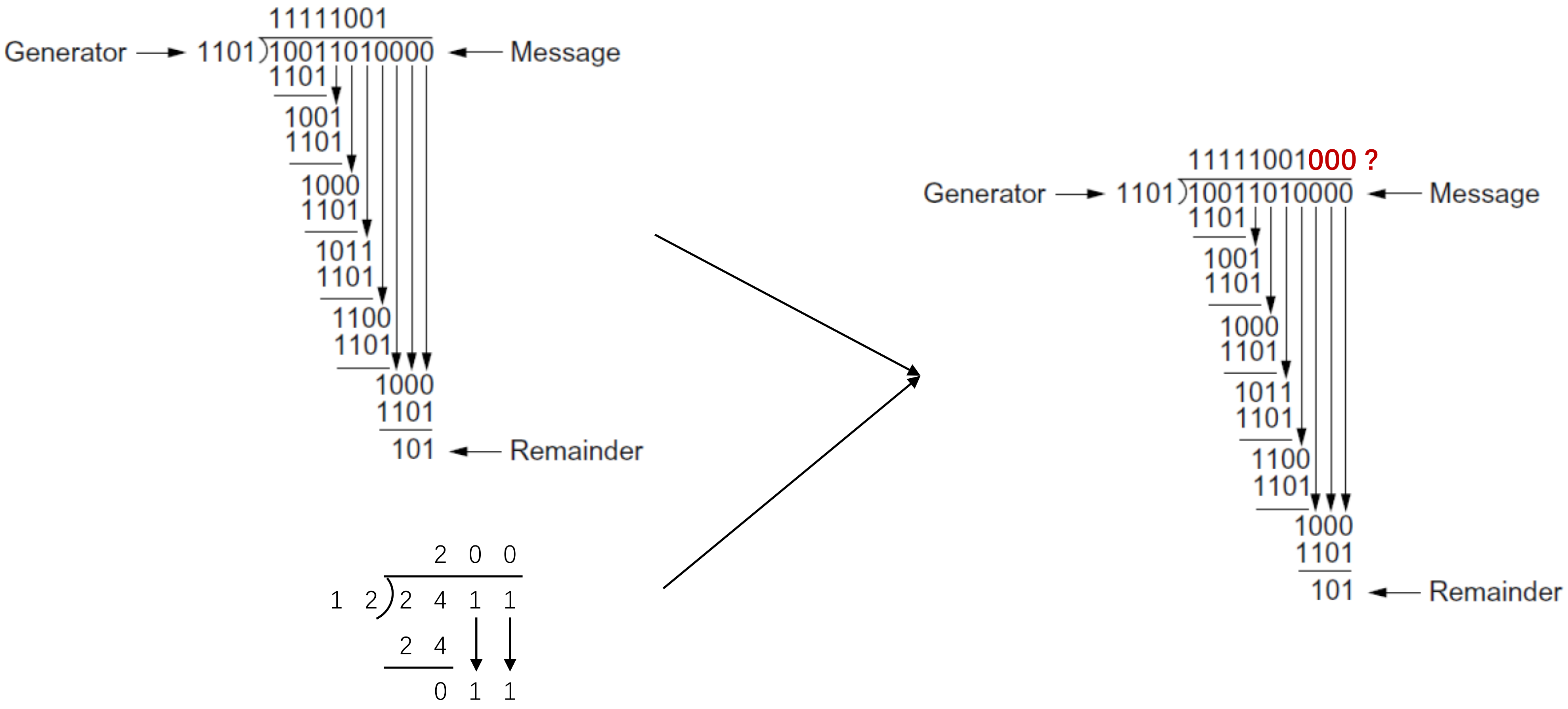
Yan Yihui
yhyan@shanghaitech.edu.cn

Wang Zhiwei
wangzw@shanghaitech.edu.cn

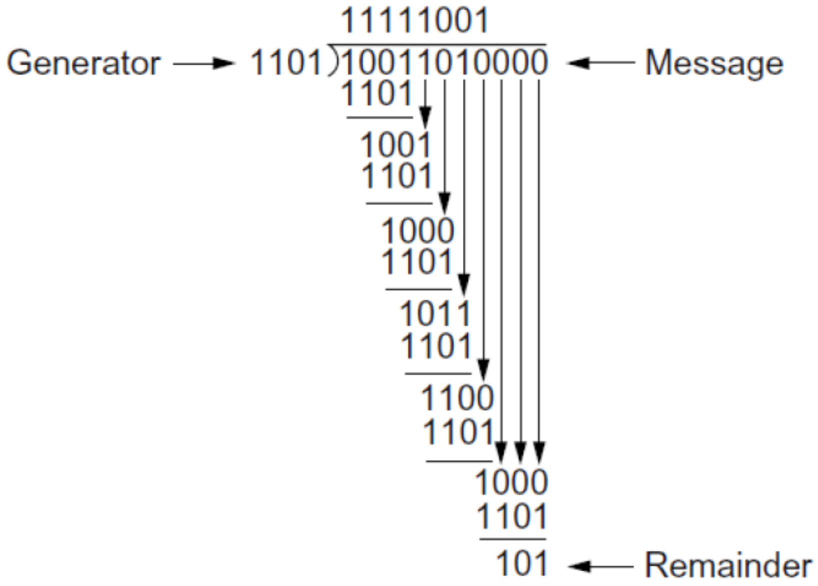
Solution:

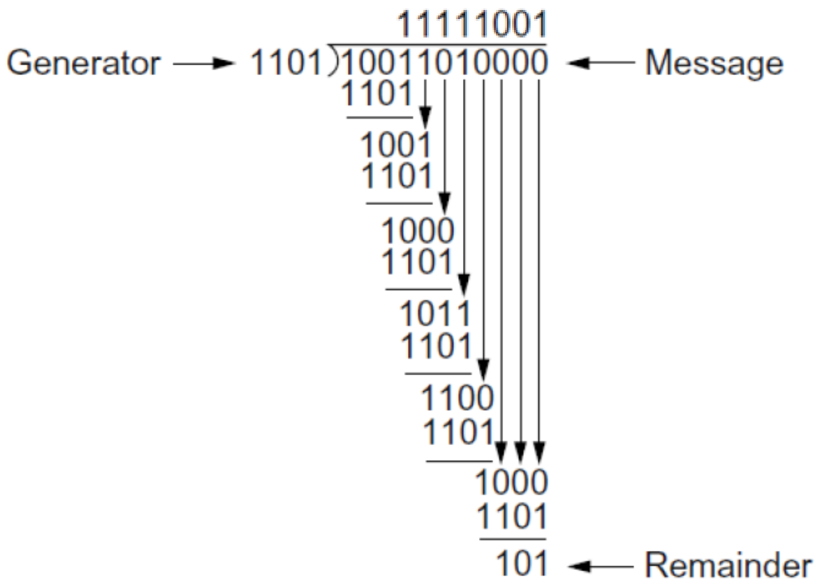
CS120 Homework Assignment #1

HW1 · Q4 · CRC


11111001
Generator
1101
10011010000
Message
1101
1001
1101
1000
1101
1011
1101
1100
1101
1000
1101
101
Remainder
2 0 0
1 2
2 4 1 1
2 4
0 1 1
11111001000 ?
Generator
1101
10011010000
Message
1101
1001
1101
1000
1101
1011
1101
1100
1101
1000
1101
101
Remainder

HW1 · Q4 · CRC


Generator
1101
Message
1001101000
11111001
Remainder
101


Generator
1101
Message
1001101000
11111001
Remainder
101

HW1 · Q4 · CRC

(a) Verify, for $p = 110$, that the quotients $p \frown 000 \div C$ and $p \frown 111 \div C$ are the same; that is, it doesn't matter what the trailing bits are. (5 points) $C = C(x)$ be $x^3 + x^2 + 1$, or 1101.

Solution 1

$p \frown 000 \div C$

$$
\begin{array}{ccccccccccc}
 & & & & & & & & 1 & 0 & 0 \\ \cline{6-11}
1 & 1 & 0 & 1 & \big) & 1 & 1 & 0 & 0 & 0 & 0 \\
 & & & & & 1 & 1 & 0 & 1 & & \\ \cline{6-9}
 & & & & & & & & 1 & 0 & 0
\end{array}
$$

$p \frown 111 \div C$

$$
\begin{array}{ccccccccccc}
 & & & & & & & & 1 & 0 & 0 \\ \cline{6-11}
1 & 1 & 0 & 1 & \big) & 1 & 1 & 0 & 1 & 1 & 1 \\
 & & & & & 1 & 1 & 0 & 1 & & \\ \cline{6-9}
 & & & & & & & & 0 & 1 & 1
\end{array}
$$

Solution 2

$p \frown \mathrm{XXX} \div C$

$$
\begin{array}{ccccccccccc}
 & & & & & & & & 1 & 0 & 0 \\ \cline{6-11}
1 & 1 & 0 & 1 & \big) & 1 & 1 & 0 & \mathrm{X} & \mathrm{X} & \mathrm{X} \\
 & & & & & 1 & 1 & 0 & 1 & & \\ \cline{6-9}
 & & & & & & & & \mathrm{Y} & \mathrm{X} & \mathrm{X}
\end{array}
$$

HW1 · Q4 · CRC

(b) Fill in the missing entries in the table. (10 points) $C = C(x)$ be $x^3 + x^2 + 1$, or 1101.

Table 1 Table-Driven CRC Calculation		
p	$q = p \frown 000 \div C$	$C \times q$
000	000	000 000
001	001	001 101
010	011	010 __ ③
011	0__ ①	011 __ ④
100	111	100 011
101	110	101 110
110	100	110 __ ⑤
111	__ ②	111 __ ⑥

①

```
                  0 1 0
1 1 0 1 ) 0 1 1 0 0 0
            1 1 0 1
          ---------
                  0 1 0
```

②

```
                  1 0 1
1 1 0 1 ) 1 1 1 0 0 0
          1 1 0 1
          -------
            1 1 0 0
            1 1 0 1
            -------
                0 0 1
```

③

```
      1 1 0 1
        0 1 1
-------------
      1 1 0 1
    1 1 0 1
  0 0 0 0
-------------
  0 1 0 1 1 1
```

④

```
      1 1 0 1
        0 1 0
-------------
      0 0 0 0
    1 1 0 1
  0 0 0 0
-------------
  0 1 1 0 1 0
```

⑤

```
      1 1 0 1
        1 0 0
-------------
      0 0 0 0
    0 0 0 0
  1 1 0 1
-------------
  1 1 0 1 0 0
```

⑥

```
      1 1 0 1
        1 0 1
-------------
      1 1 0 1
    0 0 0 0
  1 1 0 1
-------------
  1 1 1 0 0 1
```

HW1 · Q4 · CRC

(c) Use the table to divide 101 001 011 001 100 by C. Hint: The first 3 bits of the dividend are $p = 101$, so from the table the corresponding first 3 bits of the quotient are $q = 110$. Write the 110 above the second 3 bits of the dividend, and subtract $C \times q = 101\ 110$, again from the table, from the first 6 bits of the dividend. Keep going in groups of 3 bits. There should be no remainder. (10 points)

Table 1 Table-Driven CRC Calculation

p	$q = p \frown 000 \div C$	$C \times q$
000	000	000 000
001	001	001 101
010	011	010 111
011	010	011 010
100	111	100 011
101	110	101 110
110	100	110 100
111	101	111 001

```
              q   110   101   011   100
C  1101  | 101 p 001   011   001   100
           101   110
                 111   011
                 111   001
                       010   001
                       010   111
                             110   100
                             110   100
                                     0
```

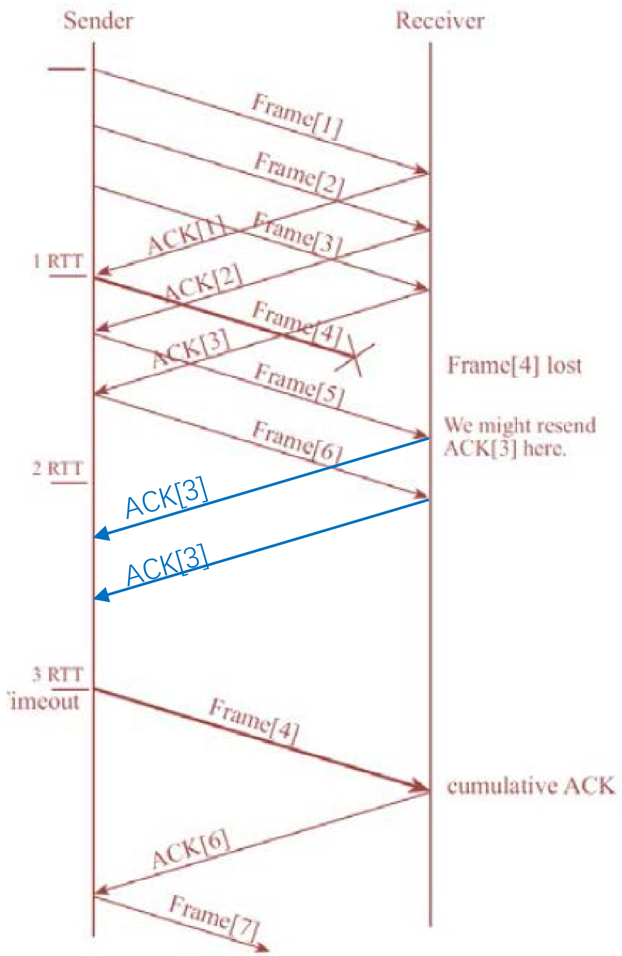
$p \frown XXX \div C$

```
               1 0 0
1 1 0 1 ) 1 1 0 X X X
          1 1 0 1
          -------
              Y X X
```

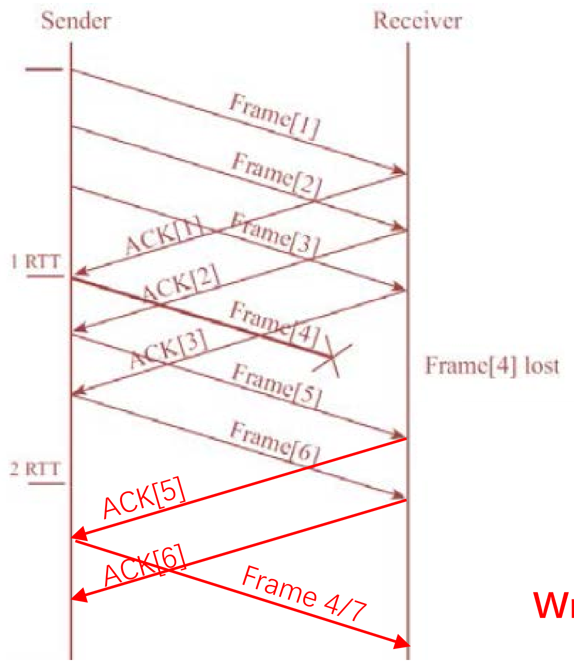
HW1 · Q5 · TCP sliding window

(20 points) Draw a time line diagram for the sliding window algorithm with SWS = RWS = 3 frames, for the following two situations. Use a timeout interval of about 2 × RTT.

(a) Frame 4 is lost. (10 points)



Right

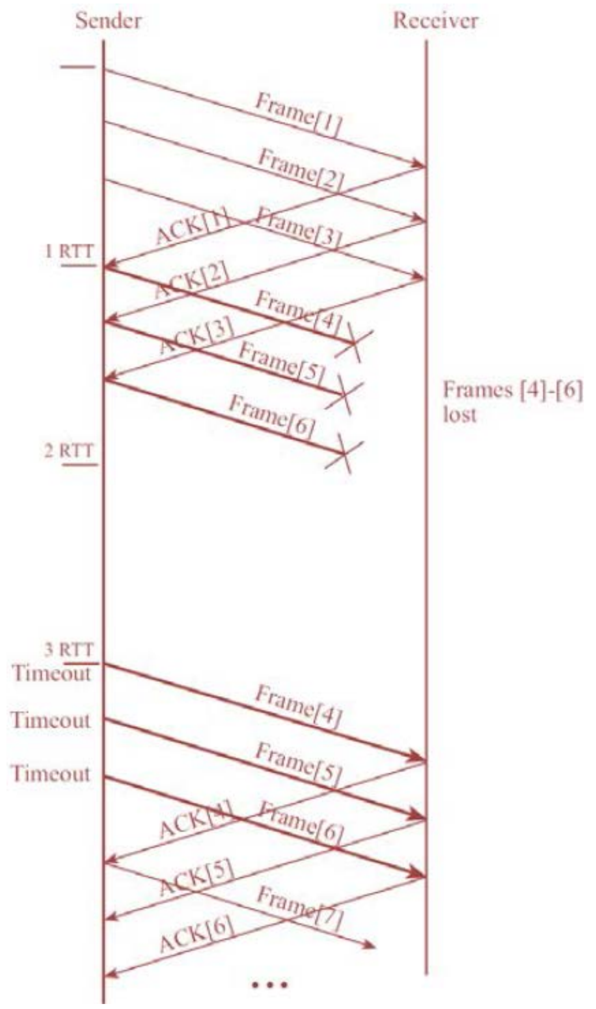


Wrong

HW1 · Q5 · TCP sliding window

(20 points) Draw a time line diagram for the sliding window algorithm with SWS = RWS = 3 frames, for the following two situations. Use a timeout interval of about 2 × RTT.

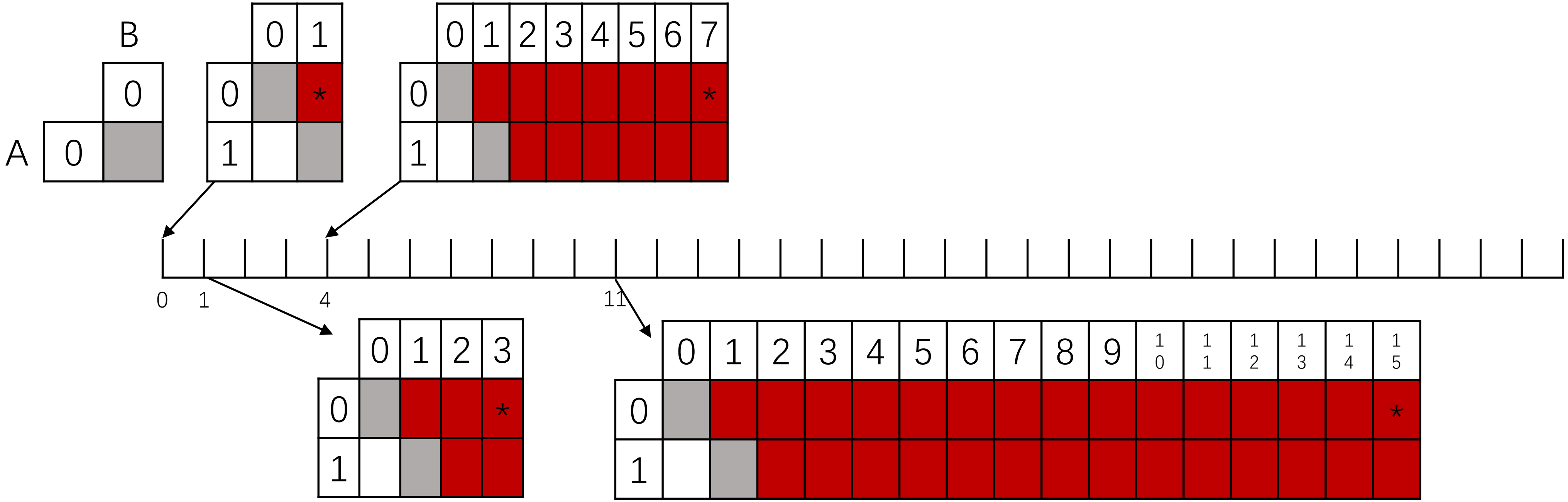
(b) Frames 4 to 6 are lost. (10 points)



HW1 · Q6 · Exponential Backoff

(20 points) Let A and B be two stations attempting to transmit on an Ethernet. Each has a steady queue of frames ready to send; A's frames will be numbered A_1, A_2, and so on, and B's similarly. Let $T = 51.2$ μs be the exponential backoff base unit.

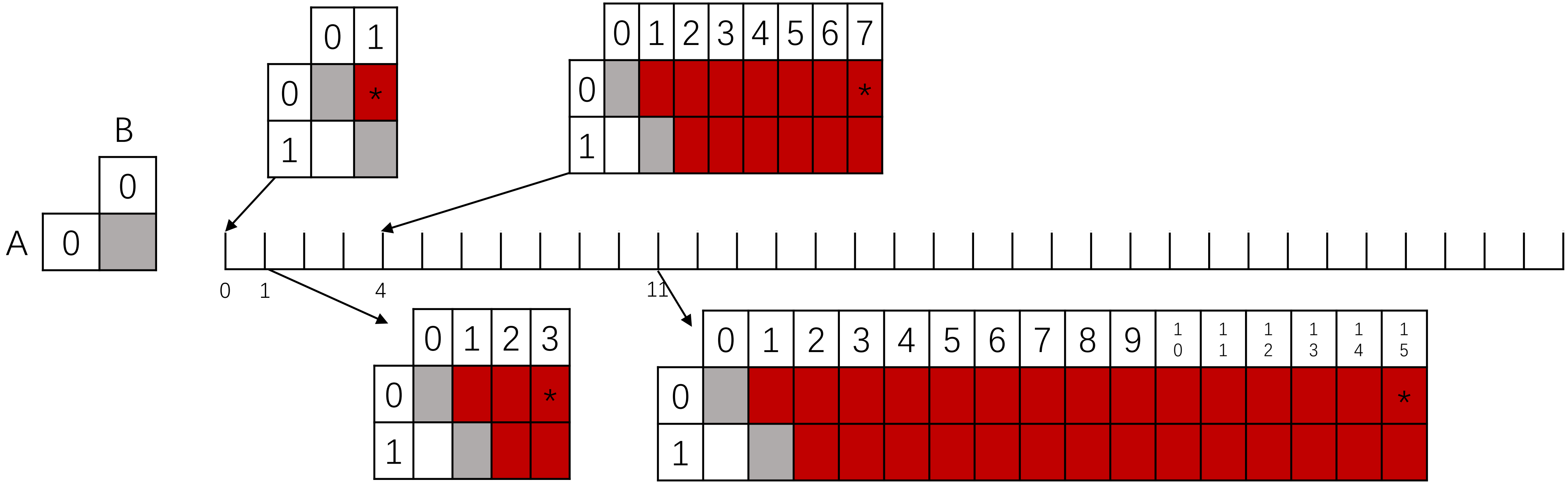
Suppose A and B simultaneously attempt to send frame 1, collide, and happen to choose backoff times of $0 \times T$ and $1 \times T$, respectively, meaning A wins the race and transmits A_1 while B waits. At the end of this transmission, B will attempt to retransmit B_1 while A will attempt to transmit A_2. These first attempts will collide, but now A backs off for either $0 \times T$ or $1 \times T$, while B backs off for time equal to one of $0 \times T, \cdots, 3 \times T$.



HW1 · Q6 · Exponential Backoff

1st: 1/4

2nd: 13/16



3rd: 5/8

4th: 29/32

$$\mathrm{i^{th}}: \frac{2^{i+1}-3}{2^{i+1}}$$

HW1 · Q6 · Exponential Backoff

4~n

(c) Give a reasonable lower bound for the probability that A wins all the remaining backoff races. (5 points)

$$\Pr\{A\ wins\ 4{\sim}n^{th}\ race\} = \prod_{i=4}^{n} \frac{2^{i+1}-3}{2^{i+1}} \qquad (\ \mathrm{i^{th}}:\ \frac{2^{i+1}-3}{2^{i+1}}\ )$$

$$\approx \begin{cases} 0.8239, n \to \infty \\ 0.8251, n = 10 \end{cases}$$

- Exponential Backoff
  - After m th collisions, chooses K at random from {0,1,2, ···, 2^m-1}
    - if m>11 **i = 10**
      - chooses K at random from {0,1,2, ···, 1024}
    - if m=16
      - done
      - go idle
  - Waits K*one time slot

$$\prod_{i=11}^{15} \frac{2^{10+1}-3}{2^{10+1}} \approx 0.9926$$

$$\Pr\{A\ wins\ all\ remaining\} = 0.8251 * 0.9926\ \approx 0.8189$$

HW1 · Q6 · Exponential Backoff

4~n

(c) Give a reasonable lower bound for the probability that A wins all the remaining backoff races. (5 points)

Another Solution

(c) P(winning race 1) = 5/8 > 1/2 and P(winning race 2) = 13/16>3/4; generalizing, we assume the odds of A winning the ith race exceed $(1 - 1/2^{i-1})$.

We now have that:

P(A wins every race given that it wins races 1-3) $\geqslant (1 - 1/8)(1 - 1/16)(1 - 1/32)(1 - 1/64)\ldots \approx 3/4$

(d) What then happens to the frame B_1? (5 points)

(d) B gives up on it, and starts over with B_2.

2. (20 points) Calculate the total time required to transfer a 1000-KB file in the following cases, assuming an RTT of 50 ms, a packet size of 1 KB data, and an initial 2 ×RTT of "handshaking" before data is sent:

(a) The bandwidth is 1.5 Mbps, and data packets can be sent continuously. (5 points)

(b) The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one. RTT before sending the next. (5 points)

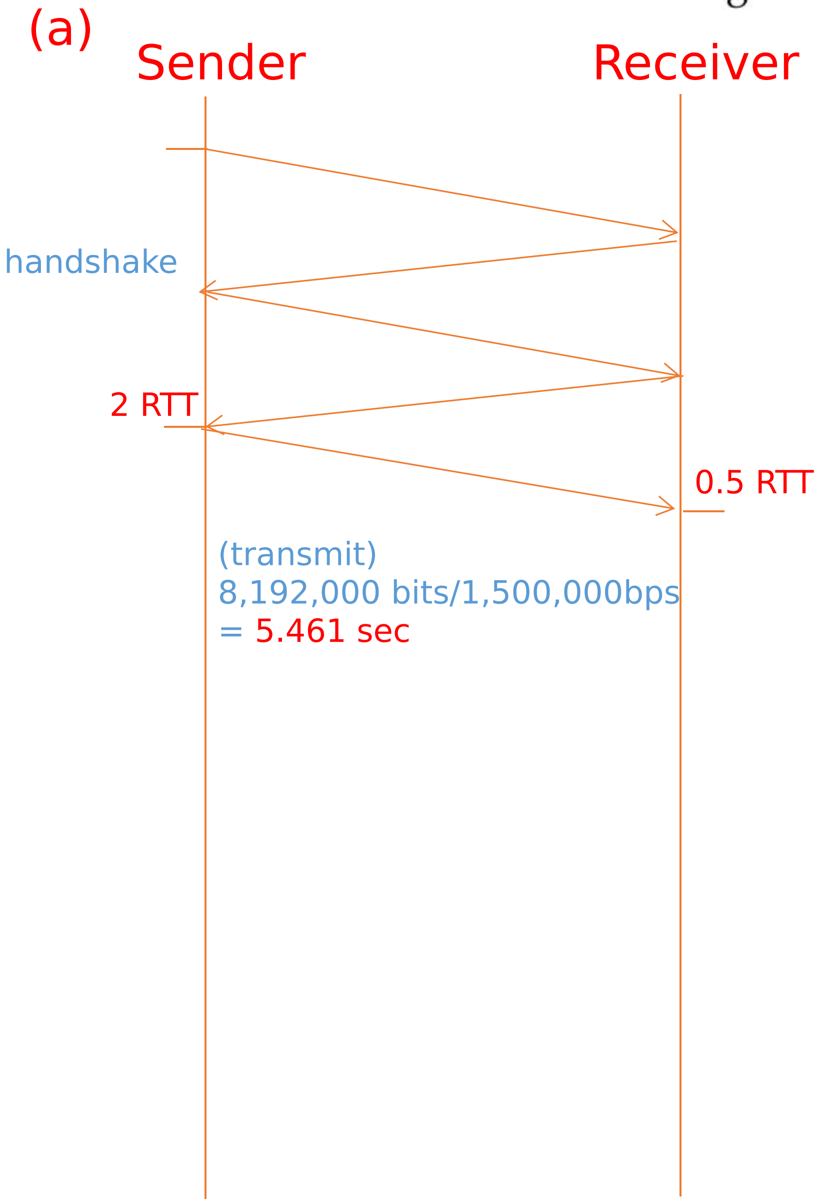
(c) The bandwidth is "infinite", meaning that we take transmit time to be zero, and up to 20. packets can be sent per RTT. (5 points)

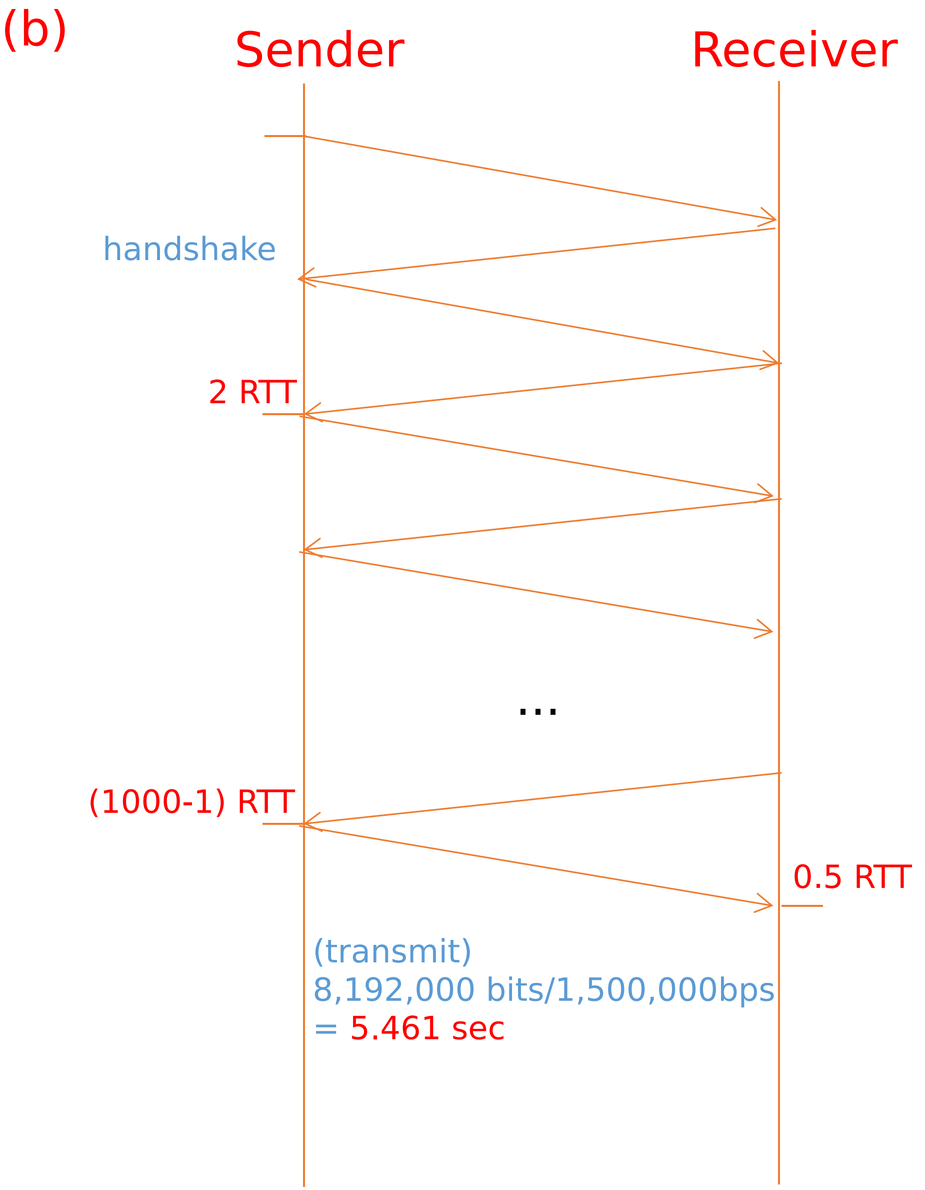
(d) The bandwidth is infinite, and during the first RTT we can send one packet (2^{1-1}), during the second RTT we can send two packets (2^{2-1}), during the third we can send four (2^{3-1}), and so on. (5 points)

* RTT/2 (propagation = 25ms)
** 1K = 1000 in bandwidth and 1K = 1024 in file

(a) The bandwidth is 1.5 Mbps, and data packets can be sent continuously. (5 points)

(b) The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one. RTT before sending the next. (5 points)

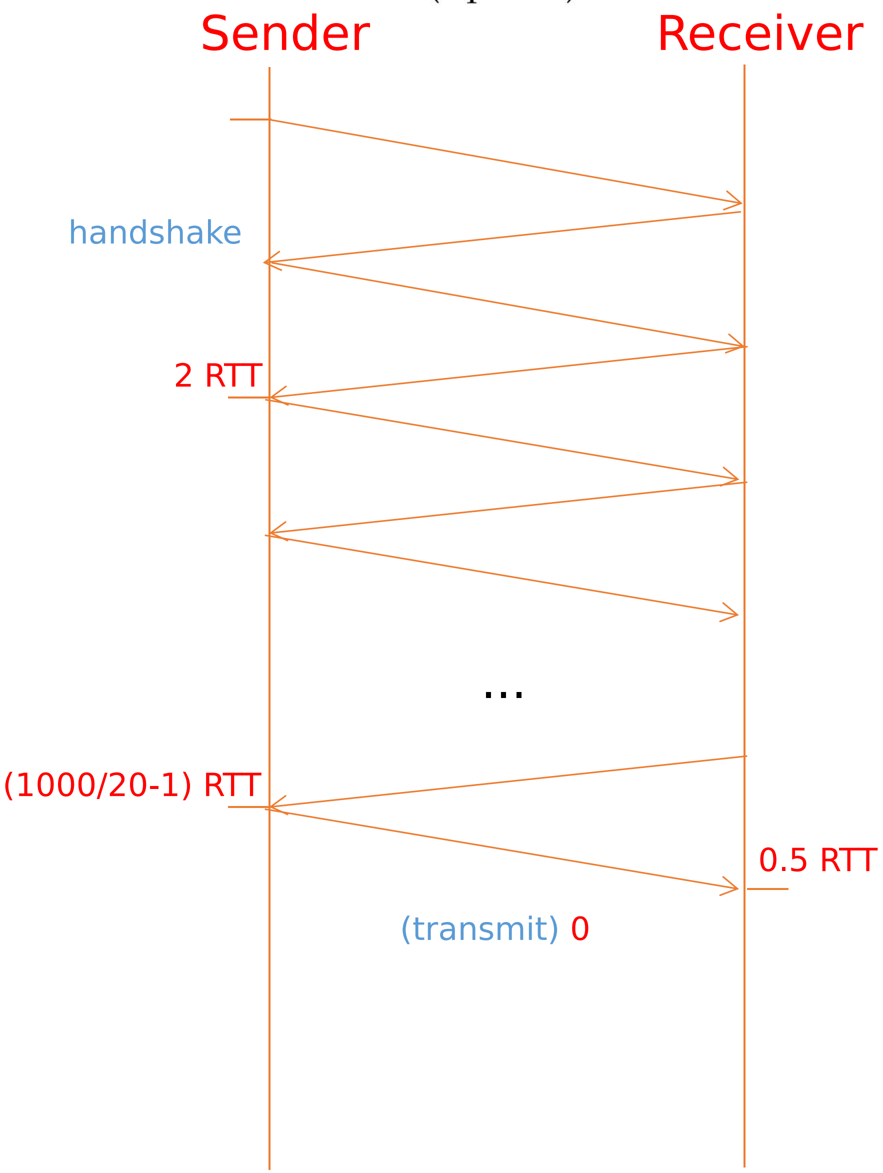




(c) The bandwidth is “infinite”, meaning that we take transmit time to be zero, and up to 20. packets can be sent per RTT. (5 points)

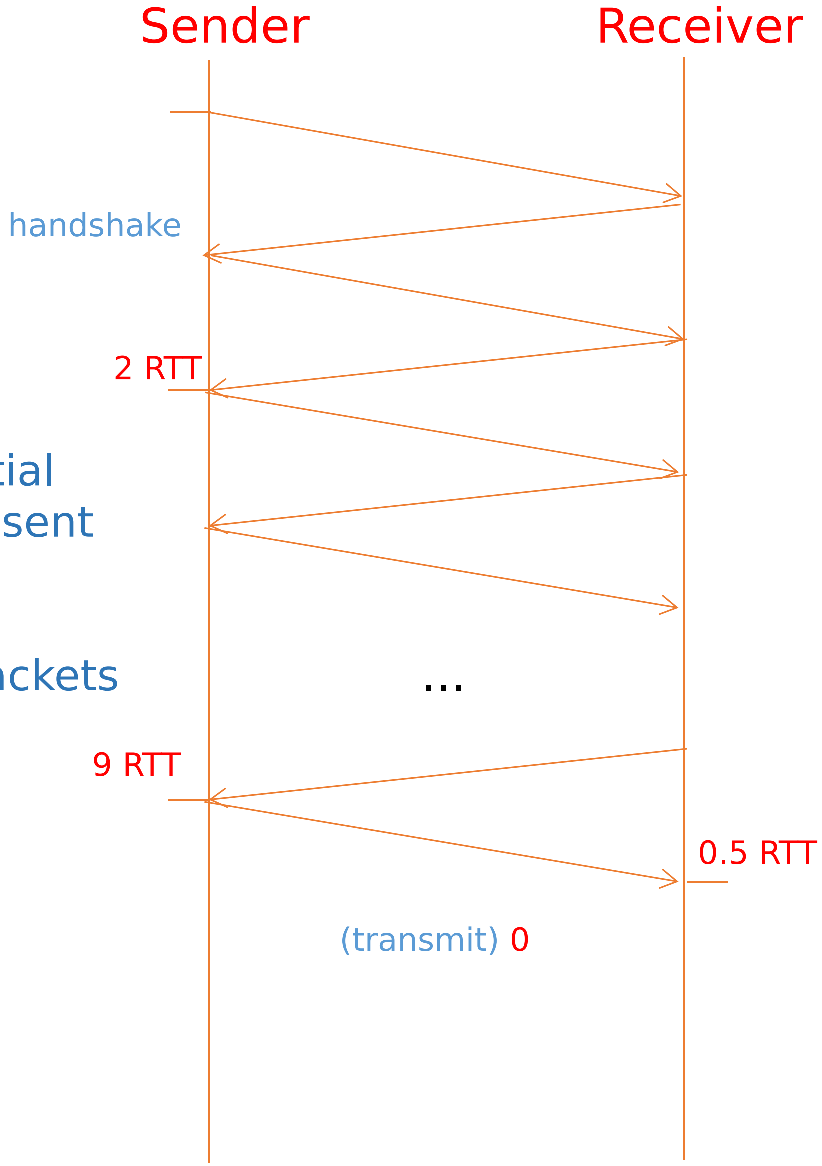
(d) The bandwidth is infinite, and during the first RTT we can send one packet (2^{1-1}), during the second RTT we can send two packets (2^{2-1}), during the third we can send four (2^{3-1}), and so on. (5 points)

(c)



(d)

At **n RTTs** past the initial handshaking we have sent
$1 + 2 + 4 + \cdots + 2^n$
$= 2^{n+1}-1$ packets
n=9, send all 1,000 packets



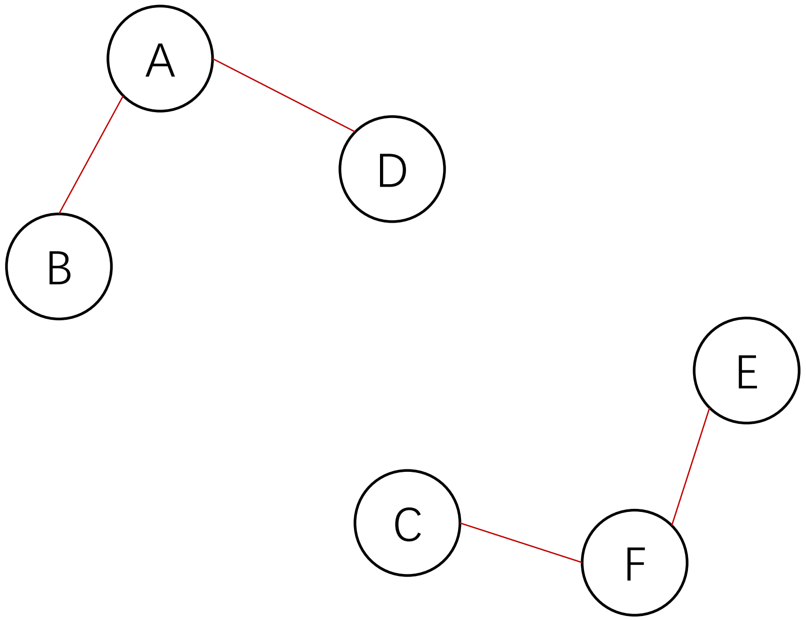
Solution:

CS120 Homework Assignment #2

HW2 · Q1 · Forwarding Table

(10 points) Suppose we have the forwarding tables shown in Table 1 for nodes A and F, in a network where all links have cost 1. Give a diagram of the smallest network consistent with these tables.

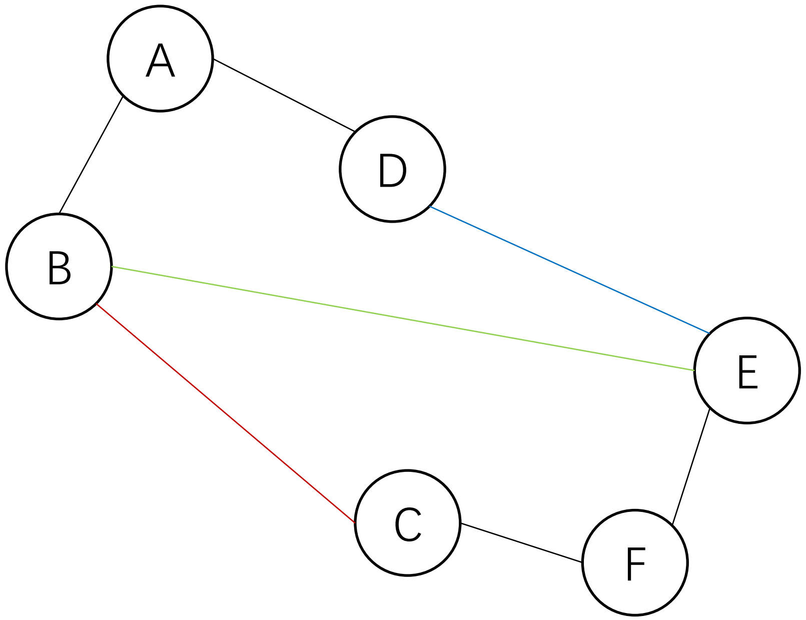
Table 1: Forwarding Tables for Exercise 1		
A		
Node	Cost	Next hop
B	1	B
C	2	B
D	1	D
E	2	B
F	3	D
F		
Node	Cost	Next hop
A	3	E
B	2	C
C	1	C
D	2	E
E	1	E



HW2 · Q1 · Forwarding Table

(10 points) Suppose we have the forwarding tables shown in Table 1 for nodes A and F, in a network where all links have cost 1. Give a diagram of the smallest network consistent with these tables.

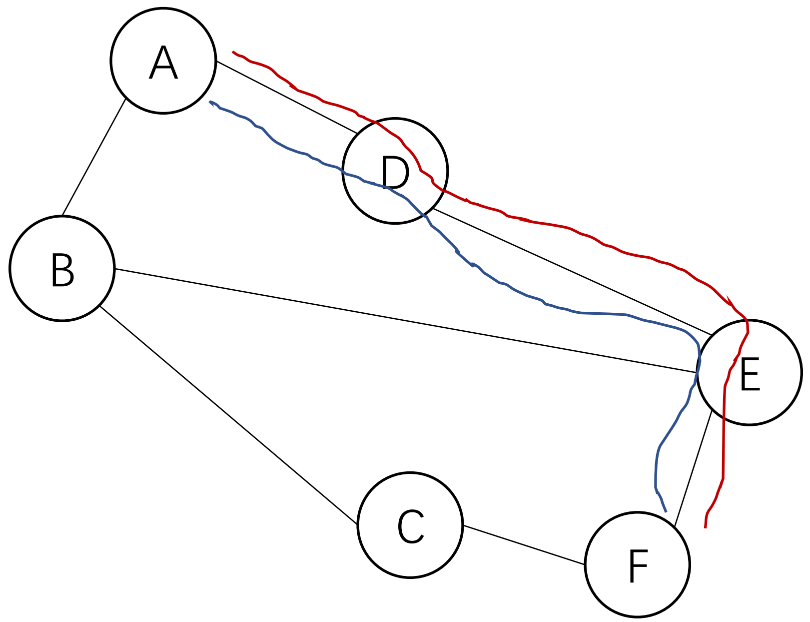
Table 1: Forwarding Tables for Exercise 1		
A		
Node	Cost	Next hop
C	2	B
E	2	B
F	3	D
F		
Node	Cost	Next hop
A	3	E
B	2	C
D	2	E

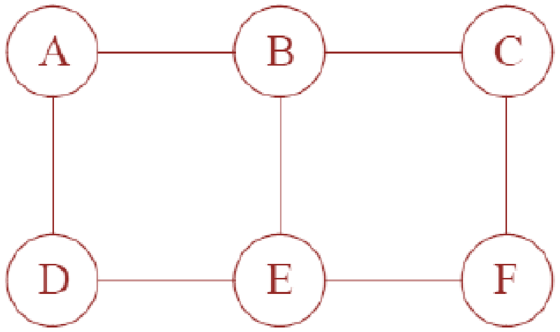


HW2 · Q1 · Forwarding Table

(10 points) Suppose we have the forwarding tables shown in Table 1 for nodes A and F, in a network where all links have cost 1. Give a diagram of the smallest network consistent with these tables.

Table 1: Forwarding Tables for Exercise 1		
A		
Node	Cost	Next hop
F	3	D
F		
Node	Cost	Next hop
A	3	E

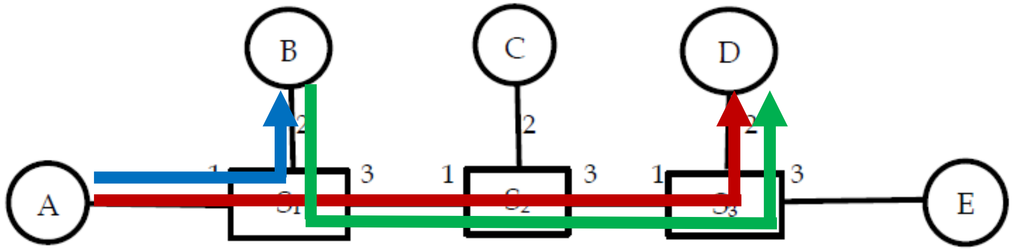




HW2 · Q2 · Virtual Circuit Switching

(10 points) Consider the virtual circuit switches in Figure 1. Table 2 lists, for each switch, what <port, VCI> (or <VCI, interface>) pairs are connected to what other. Connections are bidirectional. List all endpoint-to-endpoint connections.

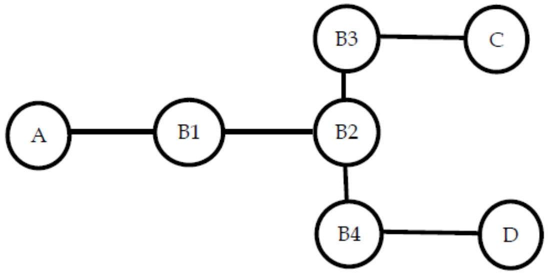
Table 2: VCI Tables for Switches in Figure 1			
Switch S_1			
Port	VCI	Port	VCI
1	2	3	1
1	1	2	3
2	1	3	2
Switch S_2			
Port	VCI	Port	VCI
1	1	3	3
1	2	3	2
Switch S_3			
Port	VCI	Port	VCI
1	3	2	1
1	2	2	2

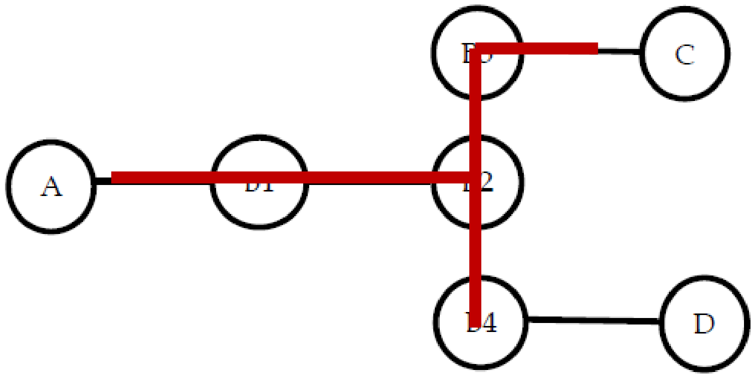


HW2 · Q3 · Bridge

(10 points) Consider the arrangement of learning bridges shown in Figure 2. Assuming all are initially empty, give the forwarding tables for each of the bridges B1 to B4 after the following transmissions:

- A sends to C
- C sends to A
- D sends to C



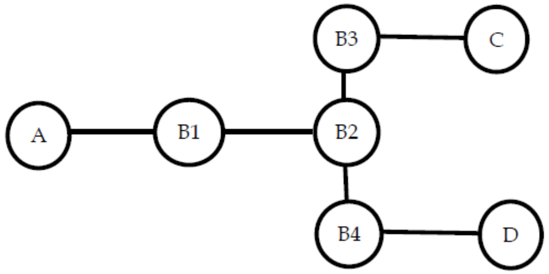


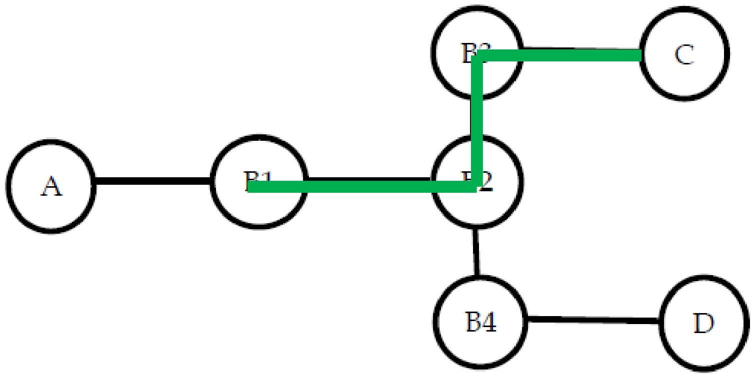
B1	B2	B3	B4
A, B1_A	A, B2_B1	A, B3_B2	A, B4_B2

HW2 · Q3 · Bridge

(10 points) Consider the arrangement of learning bridges shown in Figure 2. Assuming all are initially empty, give the forwarding tables for each of the bridges B1 to B4 after the following transmissions:

- A sends to C↵
- C sends to A↵
- D sends to C↵



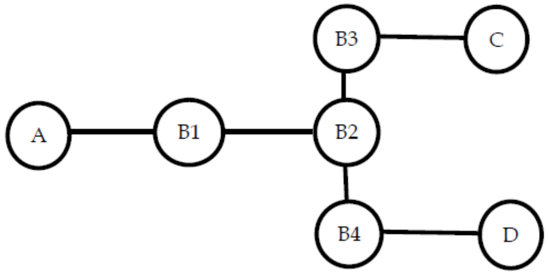


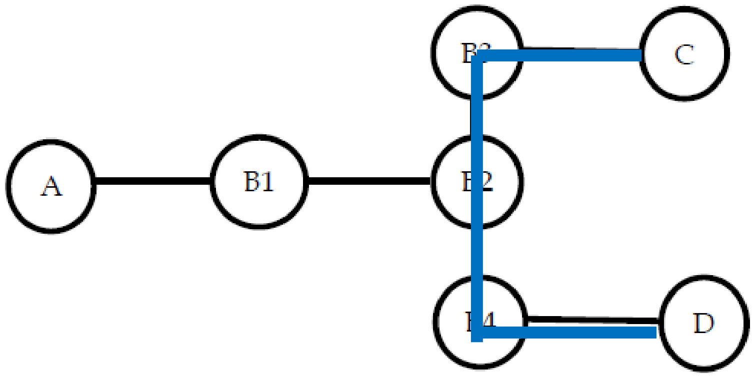
B1	B2	B3	B4
A, B1_A	A, B2_B1	A, B3_B2	A, B4_B2
C, B1_B2	C, B2_B3	C, B3_C	

HW2 · Q3 · Bridge

(10 points) Consider the arrangement of learning bridges shown in Figure 2. Assuming all are initially empty, give the forwarding tables for each of the bridges B1 to B4 after the following transmissions:

- A sends to C
- C sends to A
- D sends to C





B1	B2	B3	B4
A, B1_A	A, B2_B1	A, B3_B2	A, B4_B2
C, B1_B2	C, B2_B3	C, B3_C	
	D, B2_B4	D, B3_B2	D, B4_D

5. (5 points) Table 3 is a routing table using CIDR. Address bytes are in hexadecimal. The notation "/12" in C4.50.0.0/12 denotes a netmask with 12 leading 1 bits: FF.F0.0.0. Note that the last three entries cover every address and thus serve in lieu of a default route. State to what next hop the following will be delivered

a) C4.5E.13.87

b) C4.5E.22.09

c) C3.41.80.02

d) 5E.43.91.12

e) C4.6D.31.2E

Table 3: Routing Table for Exercise 5

Net/Mask Length	Next hop
C4.50.0.0/12	A
C4.5E.10.0/20	B
C4.60.0.0/12	C
C4.68.0.0/14	D
80.0.0.0/1	E
40.0.0.0/2	F
00.0.0.0/2	G

The rule in this case is based on the principle of "**longest match**". Using longest prefix matching

b) C4.5E.22.09

A - C4.5.../12

B - C4.5E.1.../20

6. (20 points) An organization has been assigned the prefix 212.1.1/24 (class C) and wants to form subnets for four departments, with hosts as follows:

A	75 hosts
B	35 hosts
C	20 hosts
D	18 hosts

There are 148 hosts in all.

(a) Give a possible arrangement of subnet masks to make this possible. (10 points)

(b) Suggest what the organization might do if department D grows to 32 hosts (Hint: Give A two subnets). (10 points)

(a) rounding up to the nearest power of 2
A: 2^7(.80), B: 2^6(.E0), C: 2^5(.C0), D: 2^5(.C0)

(b) A: 2^6+2^5, B: 2^6, C: 2^5, D: 2^6

* **all-0 and all-1** host values are reserved
for network address and broadcast address

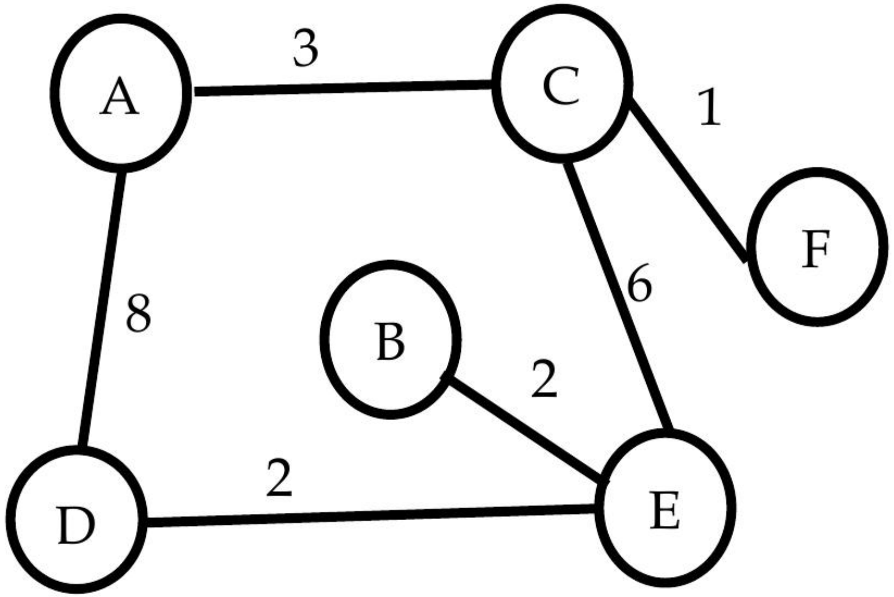
Subnet: 212.1.1.0

Mask: 255.255.255.80

or 212.1.1.0/25

8. (10 points) For the network given in Figure 3, show how the link-state algorithm builds the routing table for node D.

Step	Confirmed	Tentative
1	(D,0,-)	
2	(D,0,-)	(A,8,A), **(E,2,E)**
3	(D,0,-), (E,2,E)	(A,8,A), **(B,4,E)**, (C,8,E)
4	(D,0,-), (E,2,E), (B,4,E)	**(A,8,A)**, (C,8,E)
5	(D,0,-), (E,2,E), (B,4,E), (A,8,A)	**(C,8,E)**
6	(D,0,-), (E,2,E), (B,4,E), (A,8,A), (C,8,E)	**(F,9,E)**
7	(D,0,-), (E,2,E), (B,4,E), (A,8,A), (C,8,E), (F,9,E)	



Solution:

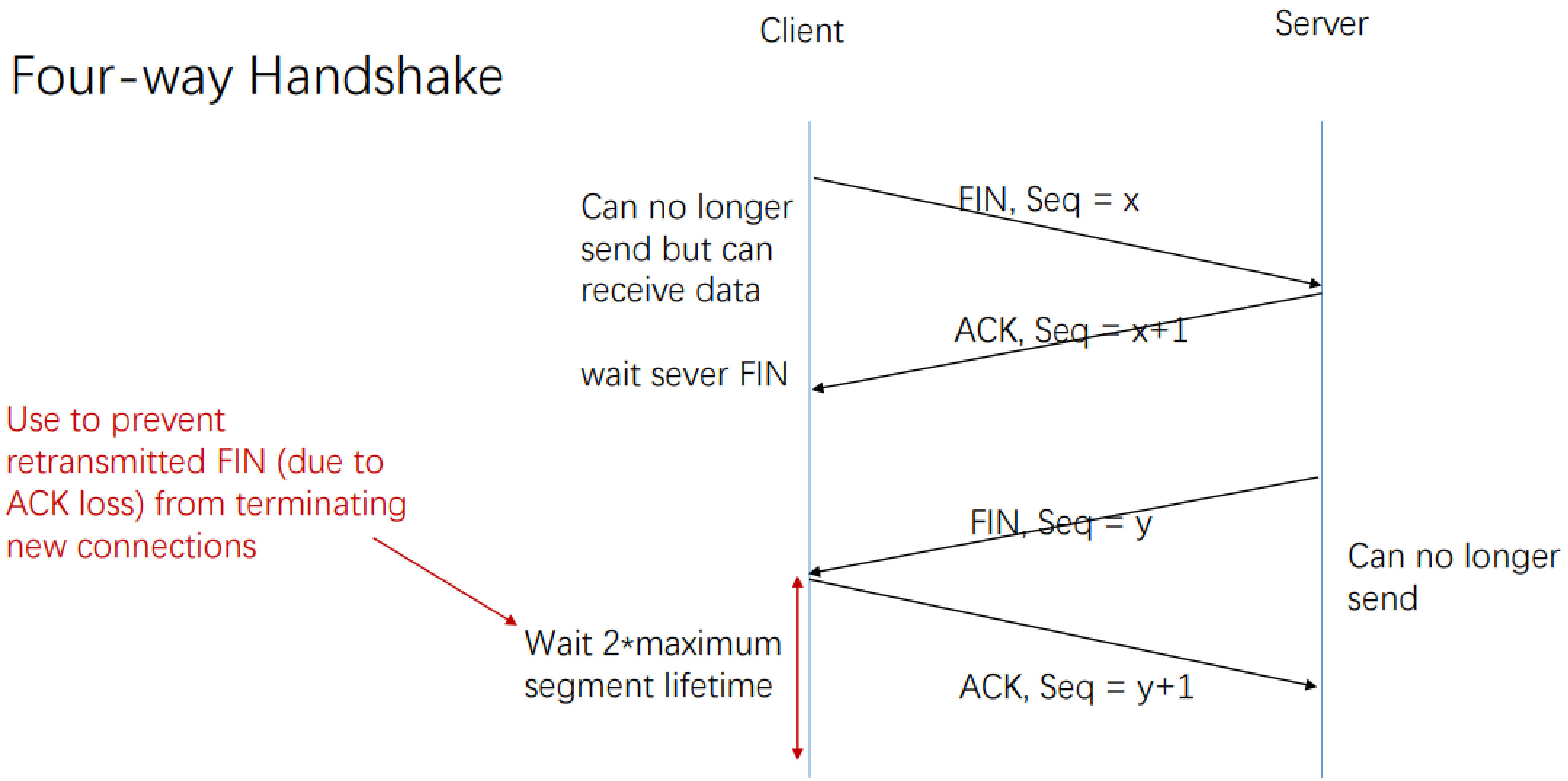
CS120 Homework Assignment #3

HW3 · Q1 · TIME_WAIT

(5 points) When closing a TCP connection, why is the two-segment-lifetime timeout not necessary on the transition from LAST_ACK to CLOSED?

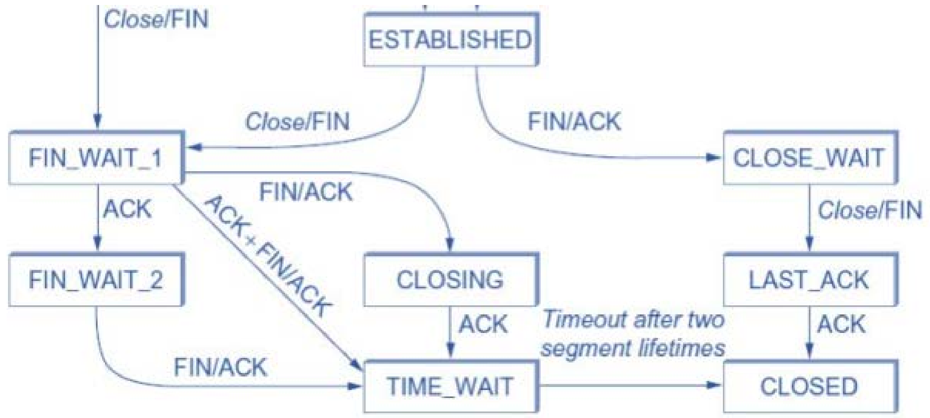
Connection Termination

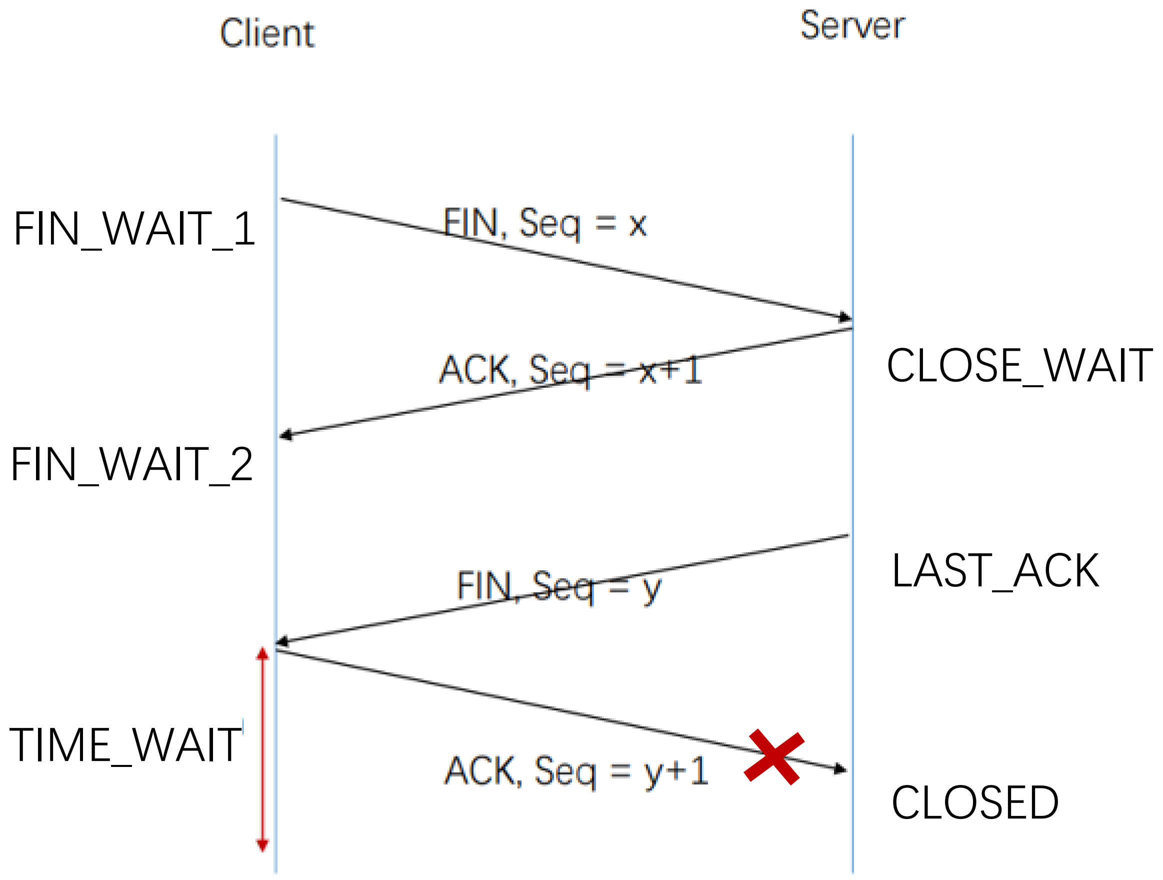
- Four-way Handshake



HW3 · Q1 · TIME_WAIT

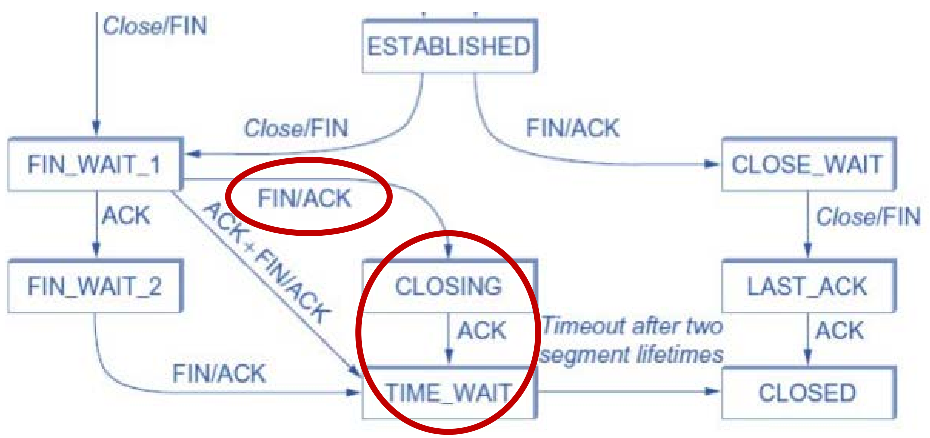
(5 points) When closing a TCP connection, why is the two-segment-lifetime timeout not necessary on the transition from LAST_ACK to CLOSED?

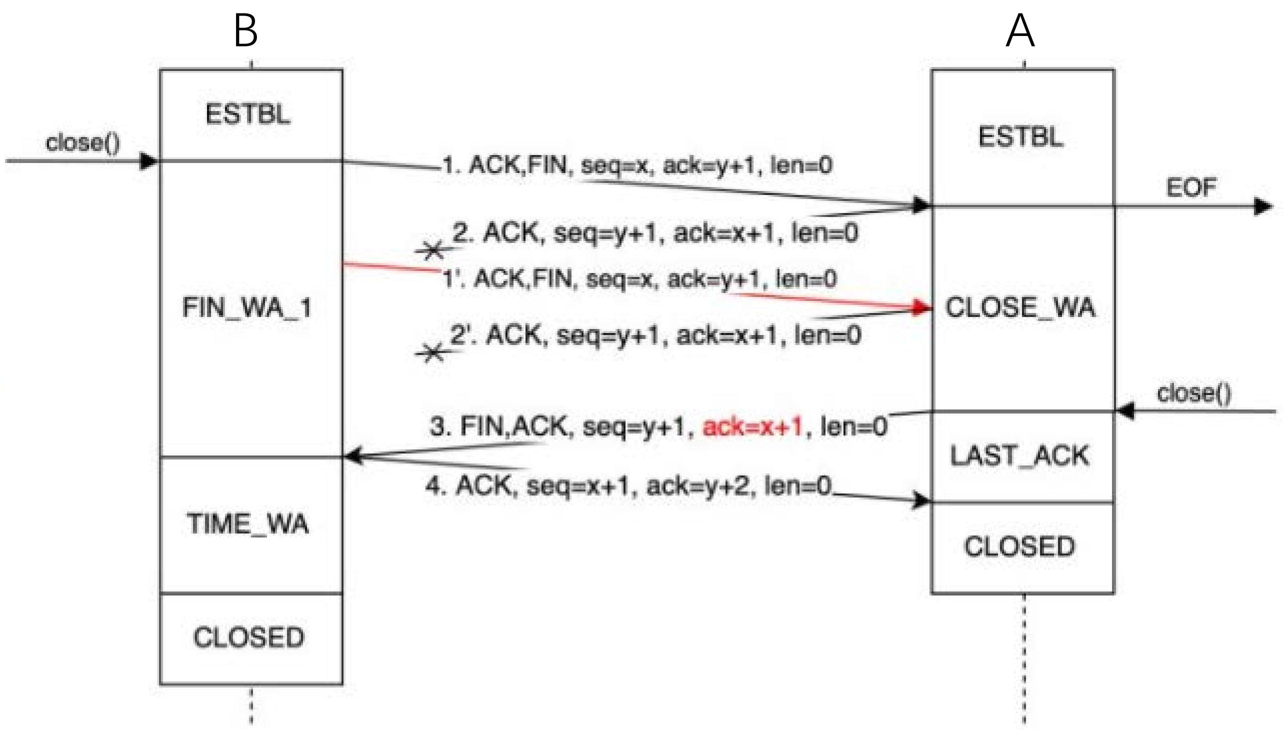




HW3 · Q1 · TIME_WAIT

(5 points) When closing a TCP connection, why is the two-segment-lifetime timeout not necessary on the transition from LAST_ACK to CLOSED?

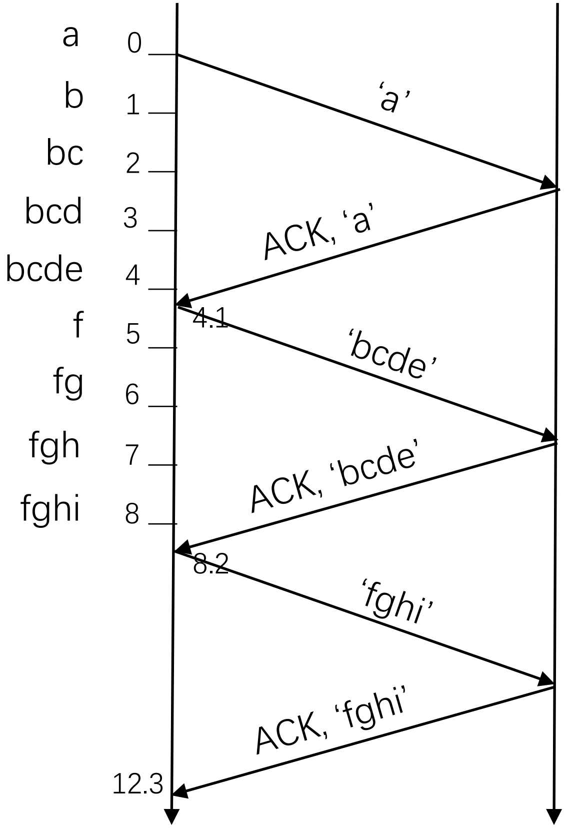




HW3 · Q3 · Nagle

(10 points) The Nagle algorithm, built into most TCP implementations, requires the sender to hold a partial segment's worth of data (even if PUSHed) until either a full segment accumulates or the most recent outstanding ACK arrives.

(a) Suppose the letters **abcdefghi** are sent, one per second, over a TCP connection with an RTT of 4.1 seconds. Draw a timeline indicating when each packet is sent and what it contains. (5 points)

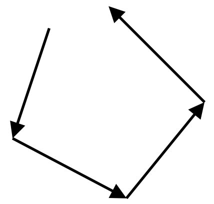


HW3 · Q3 · Nagle

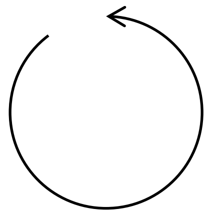
(b) Suppose that mouse position changes are being sent over the connection. Assuming that multiple position changes are sent each RTT, how would a user perceive the mouse motion with and without the Nagle algorithm? (5 points)

(b) With the Nagle algorithm, the mouse would appear to skip from one spot to another. Without the Nagle algorithm the mouse cursor would move smoothly, but it would display some inertia: it would keep moving for one RTT after the physical mouse were stopped.

With Nagle



Without Nagle



2. (10 points) You are hired to design a reliable byte-stream protocol that uses a sliding window (like TCP). This protocol will run over a 1-Gbps network. The RTT of the network is 100 ms, and the maximum segment lifetime is 30 seconds. How many bits would you include in the **AdvertisedWindow** and **SequenceNum** fields of your protocol header?

i. The advertised window should be large enough to keep the pipe full.

Required Window Size = Delay(RTT) * Bandwidth
here is 100ms × 1Gbps = 100Mb = 12.5 MB

This requires **24 bits** (2^24 ≈= 16million)

ii. The SequenceNum field must not wrap around in the maximum segment lifetime.

In 30 seconds, 30 Gb = 3.75 GB, requires **32 bits** (about 4GB)

* If the maximum segment lifetime were **not an issue**
SequenceNum > **2 *** the maximum window size

4. (10 points) Suppose, in TCP's adaptive retransmission mechanism, that EstimatedRTT is 4.0 seconds at some point and subsequent measured RTT's all are 1.0 second. How long does it take before the TimeOut value, as calculated by the Jacobson/Karels algorithm, falls below 4.0 seconds? Use $\delta = 1/8$. Use initial deviation value of 1.0 (Hint: You can use Excel to calculate).

Difference = SampleRTT − EstimatedRTT

EstimatedRTT = EstimatedRTT + (δ*Difference)

Deviation = Deviation + δ*(|Difference| − Deviation)

- δ is typically set to 1/8

TimeOut = μ * EstimatedRTT + φ * Deviation

- μ is typically set to 1 and φ is set to 4

Time/s	TimeOut/s	EstimatedRTT/s	SampleRTT/s	Deviation/s	Difference/s	δ	μ	φ
0.0	8.00	4.000	1.0	1.000	~~3.000~~	0.125	1	4
1.0	8.63	3.625	1.0	1.250	-2.625	0.125	1	4
2.0	8.98	3.297	1.0	1.422	-2.297	0.125	1	4
3.0	9.13	3.010	1.0	1.531	-2.010	0.125	1	4

EstimatedRTT = a * EstimatedRTT + (1 - a)* SampleRTT

8. (20 points) Assume that TCP implements an extension that allows window sizes much larger than 64 KB. Suppose that you are using this extended TCP over a 1-Gbps link with a latency of 50 ms to transfer a 10-MB file, and the TCP receive window is 1 MB. If TCP sends 1-KB packets (assuming no congestion and no lost packets):

(a) How many RTTs does it take until slow start opens the send window to 1 MB? (5 points)

(b) How many RTTs does it take to send the file? (5 points)

(c) If the time to send the file is given by the number of required RTTs multiplied by the link latency, what is the effective throughput for the transfer? What percentage of the link bandwidth is utilized? (10 points)

(a) the size of the window doubles every RTT
take **10 RTTs** before the send window has reached 2^10 KB = 1MB.

(b) the sending window is **limited** by the receive window 1 MB
After 10 RTTs, 1023KB = 1MB - 1KB has been transferred
So, it takes **another 10 RTTs** to send the file (9MB+1KB)

(c) It takes 1 second (20 RTTs) to send the file. (**20*50ms**)
The effective throughput is **(10MB / 1s) = 10MBps = 80Mbps**.
This is **8%** of the available link bandwidth.