HW5 Solutions

December 2, 2016

In this solution, denote `Int1` with i_1 and `Int2` with i_2.

1 Counting Sheep

1.1 $D_v(u, t)$

Denote $1 + i_1 \mod 4$ with a, and $7 - i_2 \mod 3$ with b, $1 \leq a \leq 4$ and $5 \leq b \leq 7$

- **`Init` is completed:**

A	via A	via B	via C	via D
to A				
to B		2		
to C			b	
to D				

B	via A	via B	via C	via D
to A	2			
to B				
to C			a	
to D				3

C	via A	via B	via C	via D
to A	b			
to B		a		
to C				
to D				1

D	via A	via B	via C	via D
to A				
to B		3		
to C				
to D				1

- **Iterations** [Program] Run util.Main_Q1_1 and input i1 i2 in order, then the table will be generated.

1.2 Loop

Denote $10(i_1 + 5)$ with c. $c > 40$.

Assumption 1: When A and B send out update messages, C is still waiting. So C immediately updates its own table after receiving messages from A and B, and sends out its update message. However, A and B have finished their 1st iteration now, so they will receive these messages in iteration 2.

- Initial

A	via A	via B	via C
to A			
to B		5	41
to C		6	40

B	via A	via B	via C
to A	5		7
to B			
to C	11		1

A	via A	via B	via C
to A	40	6	
to B	45	1	
to C			

Table 1: Before change

- Iteration 1:
  A sends D(A, B) = 41, D(A, C) = 40 to B, C;
  B sends D(B, A) = 41 to A, C;

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		7
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	8	
to B	81	1	
to C			

Table 2: Iteration 1 - Assumption 1

- Iteration 2

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		9
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	10	
to B	81	1	
to C			

Table 3: Iteration 2 - Assumption 1

- Iteration 3

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		11
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	12	
to B	81	1	
to C			

Table 4: Iteration 3 - Assumption 1

- Iteration 4

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		13
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	14	
to B	81	1	
to C			

Table 5: Iteration 4 - Assumption 1

- Iteration 5

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		15
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	16	
to B	81	1	
to C			

Table 6: Iteration 5 - Assumption 1

Assumption 2: In an update of HW5 Q1, it was stated that if one node does not receive any message at the beginning of one iteration, this iteration is skipped (implemented using something like a global synchronizing clock). C doesn't receive or sense anything when A and B sense the change, so after A and B send out their messages, all three of them enter the next iteration.

- Initial: The same as under Assumption 1.
- Iteration 1

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		7
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	6	
to B	45	1	
to C			

Table 7: Iteration 1 - Assumption 2

- Iteration 2

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		7
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	8	
to B	81	1	
to C			

Table 8: Iteration 2 - Assumption 2

- Iteration 3

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		9
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	8	
to B	81	1	
to C			

Table 9: Iteration 3 - Assumption 2

- Iteration 4

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		9
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	10	
to B	81	1	
to C			

Table 10: Iteration 4 - Assumption 2

- Iteration 5

A	via A	via B	via C
to A			
to B		c	41
to C		$c+1$	40

B	via A	via B	via C
to A	c		11
to B			
to C	$c+6$		1

C	via A	via B	via C
to A	40	10	
to B	81	1	
to C			

Table 11: Iteration 5 - Assumption 2

2 Secret Intelligence Service

2.1 Event causing window-size decrease

- A Triple duplicate ACK
- C Triple duplicate ACK
- D Timeout

2.2 Calculate RTT

packets sent: $N = 2^{4+(i_1 \mod 3)} - 1 = \sum_{i=0}^{3+(i_1 \mod 3)} 2^i = 4 + (i_1 \mod 3)$ windows of size 2^i

Answer 1: Counting handshake packets

1 RTT for handshake, $4 + (i_1 \mod 3)$ RTT for data packets, so $5 + (i_1 \mod 3)$ RTT.

$$\text{RTT} = 1s/(5 + (i_1 \mod 3))$$

Since $i_1 \mod 3$ can only take on 3 different values, there are only 3 possible solutions:

$i_1 \mod 3$	N	RTT
0	15	1/5
1	31	1/6
2	63	1/7

Answer 2: Assuming handshake already happened, as stated on Piazza

$$\text{RTT} = 1s/(4 + (i_1 \mod 3))$$

Since $i_1 \mod 3$ can only take on 3 different values, there are only 3 possible solutions:

$i_1 \mod 3$	N	RTT
0	15	1/4
1	31	1/5
2	63	1/6

2.3 Window size when B happens

$$\text{cwnd}_B = \frac{1}{2} \cdot 2^{3+(i_1 \mod 3)} = 2^{2+(i_1 \mod 3)}$$

2.4 Window size when C happens

$$\begin{aligned}\text{cwnd}_C &= \text{cwnd}_B + \#\text{ RTT between B and C} \\ &= 2^{2+(i_1 \mod 3)} + \frac{120s}{\text{RTT}} \\ &= 2^{2+(i_1 \mod 3)} + 2\cdot(5+(i_1 \mod 3))\end{aligned}$$

2.5 How long to get back

Half of cwnd when D happens: $2^{3+(i_2 \mod 3)}$

Time before half: $3+(i_2 \mod 3)$

Time after half: $2^{3+(i_2 \mod 3)}$

Total time in terms of RTT: $3+(i_2 \mod 3)+2^{3+(i_2 \mod 3)}$

3 Attack from Root

3.1 Packets sent by sender

Window size grows to 4, so sender sends 4 packets.

Sequence numbers:

- $(i_1 \mod 5+1)\cdot 900+1$
- $(i_1 \mod 5+1)\cdot 1800+1$
- $(i_1 \mod 5+1)\cdot 2700+1$
- $(i_1 \mod 5+1)\cdot 3600+1$

Each of size $(i_1 \mod 5+1)\cdot 900$ bytes

3.2 Window size

$(1+M)^n$

3.3 Protcol enhancement

Sample answer: Compare received ACK number with sequence numbers of packets sent previously and index of next byte to send. If none of them equal the ACK number, ignore that ACK.

(Any reasonable answer which can limit the window size or check the sequence number before increasing cwnd is acceptable)