CSC458 - Problem Set 2

Minh Le Hoang - 999 01 9930

November 12, 2015

Part I

Chapter 4

- 5)
  - a)

    P:

Destination	Netmask	NextHop
2.0.0.0/8	255.0.0.0	Q
3.0.0.0/8	255.0.0.0	R
1.A3.0.0/16	255.255.0.0	PA
1.B0.0.0/12	255.15.0.0	PB

    Q:

Destination	Netmask	NextHop
1.0.0.0/8	255.0.0.0	P
3.0.0.0/8	255.0.0.0	R
2.0A.10.0/20	255.255.15.0	QA
2.0B.0.0/16	255.255.0.0	QB

    R:

Destination	Netmask	NextHop
1.0.0.0/8	255.0.0.0	P
2.0.0.0/8	255.0.0.0	Q

  - b)

    P:

Destination	Netmask	NextHop
2.0.0.0/8	255.0.0.0	Q
3.0.0.0/8	255.0.0.0	Q
1.A3.0.0/16	255.255.0.0	PA
1.B0.0.0/12	255.15.0.0	PB

    R:

Destination	Netmask	NextHop
1.0.0.0/8	255.0.0.0	Q
2.0.0.0/8	255.0.0.0	Q

  - c)

    P:

Destination	Netmask	NextHop
2.0.0.0/8	255.0.0.0	Q
2.0A.10.0/20	255.255.15.0	QA
3.0.0.0/8	255.0.0.0	Q
1.A3.0.0/16	255.255.0.0	PA
1.B0.0.0/12	255.15.0.0	PB

Q:

Destination	Netmask	NextHop
1.0.0.0/8	255.0.0.0	P
1.A3.0.0/16	255.255.0.0	PA
3.0.0.0/8	255.0.0.0	R
2.0A.10.0/20	255.255.15.0	QA
2.0B.0.0/16	255.255.0.0	QB

- 25)

  Assuming that there is a communication between host A and host B. When host B is being from one network to another, the mobile host B will accquire a new IP. However, host A cannot know immediately that host B have moved to a new IP address, then host A will keep sending data to the old address of host B, not the new one. This can cause connection issue when using Voice over IP telephone call, and host B switch from 802.11 to 3G wireless. Hence if DHCP is used alone, when the mobile host changes it network, other hosts do not know about new loctation of the mobile host.

- 27)

  How might such a mechanism be used to steal traffic?

  – An adversary can pretend to be a home agent for a mobile node. Then the adversary can tell a correspondent node that the care of address of the mobile node is the adversary's node. All the trafic going for the mobile node from correspondent node now will go to the adversary. Hence, the adversary can steal all the trafic from the correspondent node for the mobile node. The adversary also have an option to relay the message back to the home agent of the mobile host, so that the adversary can pretend to be the correspondent node.

  How could it be used to launch a flood of attack traffic at another node?

  – An adversary can be or pretend to be home agents for mutliple mobile nodes. The adversary can tell multiple correspondent nodes that a target address is the the care of address of mobile nodes. After that, all the correspondent nodes will send packets to the target address. This bascially is launching a flood of attack traffic at the target address

Part II
Chapter 5

- 8)

  Because the sequence number is picked randomly from machine clock (the sequence number is clock-generated), the sequence number does not guarantee to start at 0. Because the sequence number can start at a number larger than 0 then it can wrap around to 0

- 15)

  $x =< y \iff (y - x) \leq 0$

  Note that, even though we have x and y unsigned, we still let $(y - x)$ be signed.

- 20)

- a)

Time	Come in	Queue	In flight	Sending	ACK
0	-	-	-	-	-
1	a	-	-	a	-
2	b	-	a	-	-
3	c	b	a	-	-
4	d	bc	a	-	-
5	e	bcd	a	-	-
5.1	-	bcde	-	bcde	a
6	f	-	bcde	-	-
7	g	f	bcde	-	-
8	h	fg	bcde	-	-
9	i	fgh	bcde	-	-
9.2	-	fghi	-	fghi	bcde
13.3	-	-	-	-	fghi

- b)

  User will see the character "a" at time 5.1 (after "d" got typed), then "bcde" at time 9.2(after i got typed) and "fghi" at 13.3

- c)

  * With Nagle algorithm
    The mouse will stop moving for a little bit (4.1 seconds) then jump to other position then back to stop moving
  * Without Nagle algorithm
    Every move made will have a lag in time (4.1 seconds) but the movement of the mouse is smooth and continuous.

- 26)

  $\delta = \frac{1}{8}; \mu = 1; \phi = 4$

  Assuming initially $Deviation = 2$ and $\delta = \frac{1}{8}; \mu = 1; \phi = 4$

Time(in second)	Difference	EstimatedRTT	SampleRTT	Deviation	TimeOut
0	n/a	4	1	2	12
1	-3	3.625	1	2.125	12.125
2	-2.625	3.296875	1	2.1875	12.046875
3	-2.296875	3.009765625	1	2.2011719	11.81445313
4	-2.0097656	2.758544922	1	2.1772461	11.4675293
5	-1.7585449	2.758544922	1	2.1249084	11.0383606
6	-1.5387268	2.346385956	1	2.0516357	10.55292892
7	-1.346386	2.178087711	1	1.9634795	10.03200579
8	-1.1780877	2.030826747	1	1.8653055	9.492048919
9	-1.0308267	1.901973404	1	1.7609957	8.945956178
10	-0.9019734	1.789226728	1	1.6536179	8.403698358
11	-0.7892267	1.690573387	1	1.545569	7.872849427
12	-0.6905734	1.604251714	1	1.4386946	7.359029943
13	-0.6042517	1.52872025	1	1.3343892	6.866277057
14	-0.5287202	1.462630219	1	1.2336806	6.39735255
15	-0.4626302	1.404801441	1	1.1372993	5.95399859
16	-0.4048014	1.354201261	1	1.0457371	5.537149487
17	-0.3542013	1.309926103	1	0.9592951	5.147106432
18	-0.3099261	1.271185341	1	0.878124	4.783681179
19	-0.2711853	1.237287173	1	0.8022566	4.446313702
20	-0.2372872	1.207626276	1	0.7316354	4.134168076
21	-0.2076263	1.181672992	1	0.6661343	3.846210205

With assumption that the initial $Deviation = 2$, it takes 21 seconds for time out to falls below 4.0 seconds.

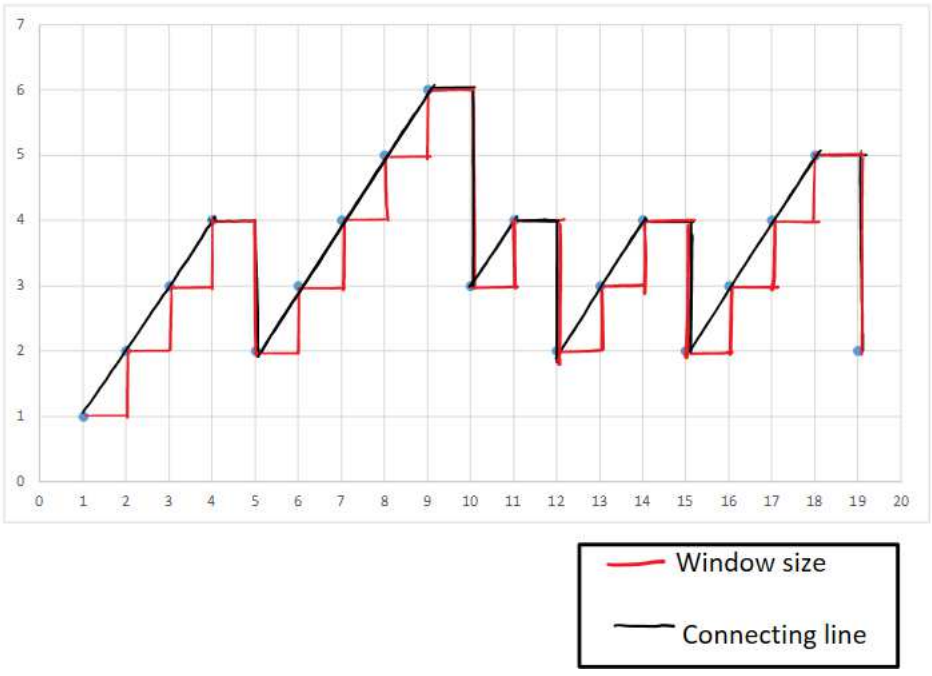
After a few trials(trying with $Deviation \in \{0, 1, 2, 3, 4, 5, 6\}$), it seems that the answer is still around 20 seconds(plus or minus)

Part III

Chapter 6

- 17)

Time	Window size	Number packets	Note
1	1	1	
2	2	3	
3	3	6	
4	4	**9**	9 dropped here
5	**2**	11	
6	3	14	
7	4	18	
8	5	23	
9	6	**25**	25 dropped here
10	**3**	28	
11	4	**30**	30 dropped here
12	**2**	32	
13	3	35	
14	4	**38**	38 dropped here
15	**2**	40	
16	3	43	
17	4	47	
18	5	**50**	50 dropped here
19	**2**	52	



- 26)

  The issue is that sender's window slide will keep increasing. Hence the sender will flood the network

  The sender can have a maxium window size, so that the number of packets in flight is capped at certain number. Another solution is that the ACK packet must contain a nonce sent from the sender to be valid.

- 27)

$T = 0$ to $T = 0.5$: slow start on startup

$T = 0.5$ to $T = 1.9$: sliding window is blocking TCP, no packet is sent

$T = 1.9$ to $T = 5.5$: linear-increase

$T = 5.5$ to $T = 5.7$: slow start after time out

$T > 5.7$: linear-increase

From $T = 0.5$ to $T = 1.9$, the sliding window is blocking TCP, for initial timeout at $T = 1.9$ after the slow start

The trace from Figure 6.28 and that of Figure 6.13 do not have fast recovery since everytime a packet loss happen, the TCP windows drop to 1 packet.

Compared to Figure 6.11, the trace from Figure 6.28 contains fast transmission but Figure 6.11 does not