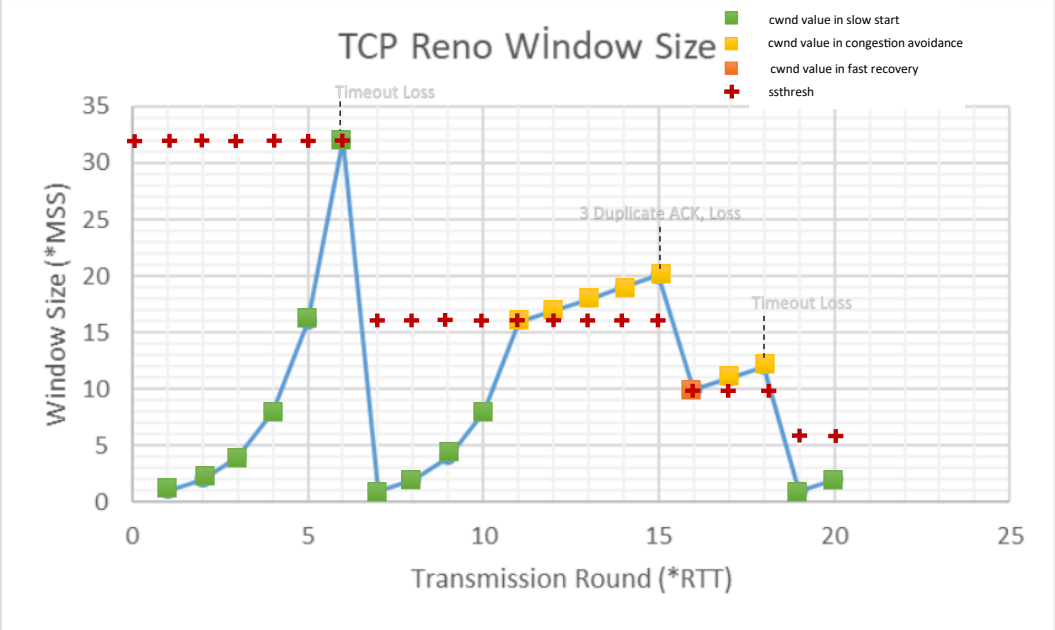
Esra Şekerci

2698215

IS 504 – Homework 3, due 2 Jan 2024

1. The complete solution is shown in the figure below:

- For intervals of time when TCP is in slow start, the plotted value of cwnd is shown as a green square
- For intervals of time when TCP is in congestion avoidance, the plotted value of cwnd is shown as a yellow square
- For intervals of time when TCP is in fast reccovery, the plotted value of cwnd is shown as an orange square
- The values for ssthresh are shown following a change as a red plus sign
- A flight of packets experiencing a loss has the loss type (which determines the next value of cwnd) labeled above



a. TCP slow start occurs when the window size is increasing exponentially. TCP slowstart is operating in the intervals [1, 6] [7, 10] and [19, 20]

b. TCP congestion avoidance occurs after slow start and is characterized by linear growth. TCP congestio avoidance is operating in the intervals [11, 15] and [16, 18]

c. The times where TCP has a loss by timeout are: 6, 18
The times where TCP has a loss by triple duplicate ACK are: 15

d. 16. At a cwnd size of 32, a timeout occurs. ssthreshold is set to half of 32 (16), and the cwnd size drops to 1.

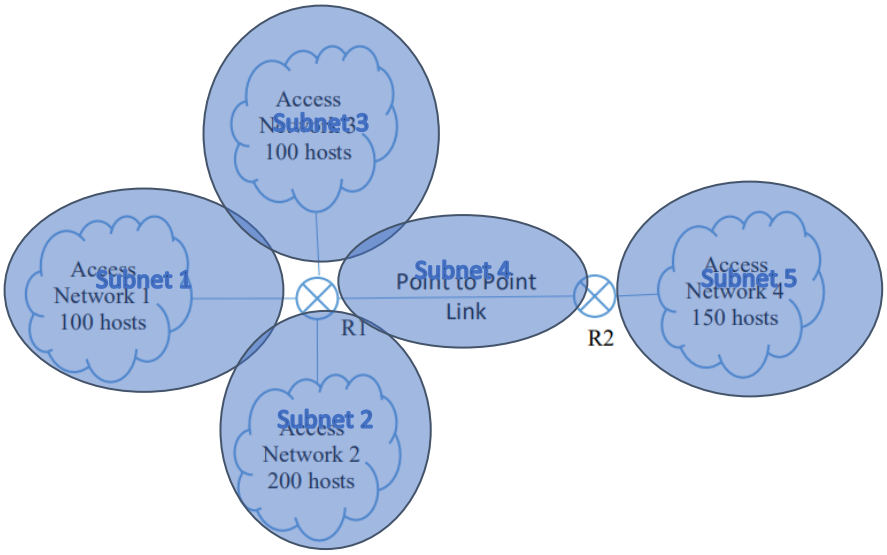
e. 10. At a cwnd size of 20, the cwnd is halved to 10 and the ssthreshold is set to 10.

f. 6. At a cwnd size of 12, the cwnd is halved to 6 and the ssthreshold is set to 6.

g. If slow start begins at round 19 and there's no loss in the 20th round, the window size in the 21st round would continue the slow start progression. Assuming a doubling pattern in slow start, the window size in the 21st round should be 4, as it doubles from the value in the 20th round.

2.

a.



b. IP address block 4.1.0.0/16 = (binary) 0000 0100.0000 0001 | 0000 0000.0000 0000

Therefore addresses will have the prefix: 0000 0100.0000 0001 | **** ****.**** ****

Subnet	# of addresses needed	# of bits needed for host part	Mask	Prefix
Subnet 1	100 hosts + 1 router + 2 reserved = 103	7 $2^7>103>2^6$	/25	0000 0100.0000 0001 | 1110 0000.0*** **** 4.1.224.0/25
Subnet 2	200 hosts + 1 router + 2 reserved = 203	8 $2^8>203>2^7$	/24	0000 0100.0000 0001 | 0000 0000.**** **** 4.1.0.0/24
Subnet 3	100 hosts + 1 router + 2 reserved = 103	7 $2^7>103>2^6$	/25	0000 0100.0000 0001 | 1100 0000.0*** **** 4.1.192.0/25
Subnet 4	2 router + 2 reserved = 4	2 $2^2= 4$	/30	0000 0100.0000 0001 | 1111 0000.0000 00** 4.1.240.0/30
Subnet 5	150 hosts + 1 router + 2 reserved = 153	8 $2^8>153>2^7$	/24	0000 0100.0000 0001 | 1000 0000.**** **** 4.1.128.0/24

3.

a.

• D(b): cost of the least-cost path from the source node to destination b as of this iteration of the algorithm.

• p(b): previous node (neighbor of b) along the current least-cost path from the source to b.

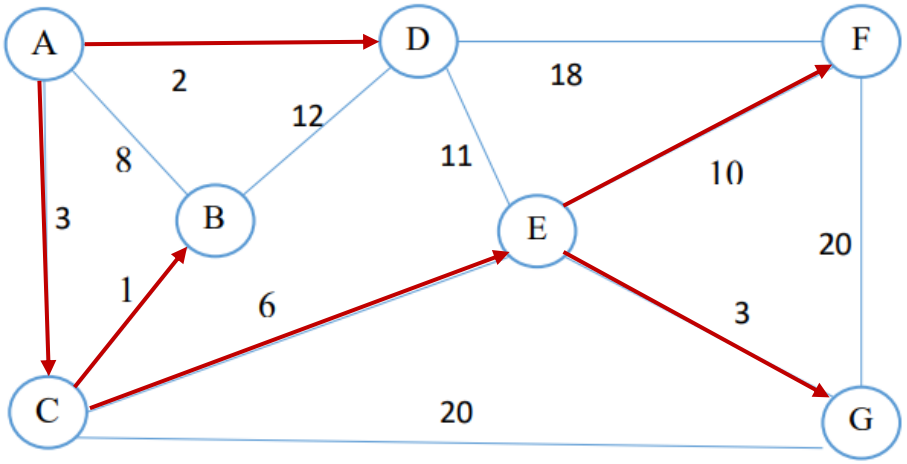
• N′: subset of nodes; b is in N′ if the least-cost path from the source to b is definitively known.

The centralized routing algorithm comprises an initialization step followed by a loop, which iterates a number of times equal to the number of nodes in the network. Upon completion, the algorithm will have computed the shortest paths from the source node 'a' to every other node in the network.

The table below illustrates the execution of the link-state algorithm on the provided network.

Step	*N'*	*D(b),p(b)*	*D(c),p(c)*	*D(d),p(d)*	*D(e),p(e)*	*D(f),p(f)*	*D(g),p(g)*
0	a	8,a	3,a	2,a	∞	∞	∞
1	ad	8,a	3,a		13,d	20,d	∞
2	adc	4,c			9,c	20,d	23,c
3	adcb				9,c	20,d	23,c
4	adcbe					19,e	12,e
5	adcbeg					19,e	
6	adcbegf						

The termination of the Link-State (LS) algorithm yields valuable information for each node, specifically its predecessor along the least-cost path originating from the source node. Extending this information, we can derive the predecessor for each of these predecessors, allowing the construction of the complete path from the source to all destinations.



b. Leveraging the acquired information, the forwarding table in a given node, such as node 'a', can be effectively assembled. This involves storing, for each destination, the next-hop node along the least-cost path from 'a' to the respective destination.

Below is the forwarding table for node 'a':

destination	next hop	cost
b	c	4
c	c	3
d	d	2
e	c	9
f	c	19
g	c	12