CS3212 計算機網路概論 期末考 解答

1.

a. AIMD **(5pt)**

Additive increase: Increase CongWin by 1 maximum segment size (MSS) every RTT in the absence of loss events.

Multiplicative decrease: Cut CongWin in half after loss event.

b. Slow Start **(5pt)**

When connection begins, CongWin = 1 MSS, increase rate by doubling value of CongWin every RTT until first loss event

When the first lost event occurs, CongWin is cut in half and then increases linearly.

c. Reaction to Timeout **(5pt)**

When timeout occurs, Threshold set to CongWin/2 and CongWin is set to 1 MSS. window then grows exponentially to a threshold, then grows linearly. (enter slow start phase)

2. (5pt)

TCP Tahoe: After Timeout event or receiving 3 dup ACKs. Tahoe will reduce congestion window to 1 MSS, and reset to slow-start state.

TCP Reno: TCP Reno cancels the slow start phase of TCP Tahoe after a triple duplicate ACK. If three duplicate ACKs are received, Reno will halve the congestion window, perform a "fast recovery"

3. (10pt)

The minimum latency is 2RTT+O/R. The minimum W that achieves this latency is

$$W = \min\left\{ w : w \geq \frac{RTT + S/R}{S/R} \right\}$$
$$= \left\lceil \frac{RTT}{S/R} \right\rceil + 1 .$$

R	min latency	W
100 Kbps	8.2 sec	4
1 Mbps	1 sec	25

4. (8pt, each 2 pt)

a) True

b) True

c) False

d) False

5. (12pt)

(3pt)

In distance vector, each node talks only to its directly connected neighbors, but it tells them everything it has learned (i.e., distance to all nodes). The nodes in distance vector do not know the global view of the whole network.

In link state, each node talks to all other node, but it tells them only what it knows for sure (i.e., only the state of its directly connected links). The nodes in link state know global network information

Message complexity: (3pt)

LS: with n nodes, E links, O(nE) messages sent

DV: exchange between neighbors only

Speed of Convergence: (3pt)

LS: O(n^2) algorithm requires O(nE) messages

DV: convergence time varies (may be routing loops, count-to-infinity problem)

Robustness: (3pt)

LS: node can advertise incorrect *link* cost, each node computes only its *own* table.

DV: DV node can advertise incorrect *path* cost, each node's table used by others. Robustness is worse because of error propagation thru network

6. (10pt)

	z	u	v	x	y
z	0	4	5	2	3
u	4	0	1	2	3
v	5	1	0	3	4
x	2	2	3	0	1
y	3	3	4	1	0

7. (10pt)

a. There is no corresponding ip/port mapping entry between Alex and Benny in NAT's translation table.

b. (1) Alex sends a request to Cindy, which asks Benny to establish a direct connection to Alex.

(2) Cindy, who can communicate with Benny directly, relays this request to Benny.

(3) Benny receives this request and establishes a connection to Alex through NAT. Now, there is a new ip/port mapping entry of this connection in NAT's translation table.

(4) Then Alex can use this TCP connection to communicate with Benny directly.

8. (10pt)

a. Adaptor receives datagram from network layer and creates frame.

b. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits.

c. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame.

d. If adapter detects another transmission while transmitting, aborts and sends jam signal.

e. After aborting, adapter enters exponential backoff: after the n*th* collision, adapter chooses a K at random from {0,1,2,…,2**m**-1} where m = min(n, 10). Adapter waits K*512 bit times and returns to Step b.

9. (12pt)

	Hubs	Switches	Routers
Traffic Isolation	No	Yes	Yes
Plug and Play	Yes	Yes	No
Optimal Routing	No	No	Yes
Cut-through	Yes	Yes	No

10. (8pt)

PPP Data frame has delimiter flag <01111110>. In order to do "data transparency", we must add some "stuffs".

Sender:

a. Adds ("stuffs") extra <01111101> byte before each <01111110> data byte.

b. Adds ("stuffs") extra <01111101> byte before each <01111101> data byte.

Receiver:

a. single <01111101> byte: discasrd <01111101>.
b. Two <01111101> bytes in a row: discard first byte, continue data reception.
c. Singe <01111110> : flag byte.