1. CIDR

Suppose P, Q, and R are network service providers, with respective CIDR address allocations C1.0.0.0/8, C2.0.0.0/8, and C3.0.0.0/8. Each provider's customers initially receive address allocations that are a subset of the provider's. P has the following customers:

PA, with allocation C1.A3.0.0/16, and
PB, with allocation C1.B0.0.0/12.

Q has the following customers:

QA, with allocation C2.0A.10.0/20, and
QB, with allocation C2.0B.0.0/16.

Assume there are no other providers or customers.

(1) Give routing tables for P, Q, and R assuming each provider connects to both of the others.

(2) Now assume P and R are no longer directly connected. Give tables for P and R.

(3) Suppose customer PA acquires a direct link to Q, and QA acquires a direct link to P, in addition to existing links. Give tables for P and Q.

2. BGP

Suppose a network N within a larger organization A acquires its own direct connection to an Internet Service Provider, in addition to an existing connection via A. Let R1 be the router connecting N to its own provider, and let R2 be the router connecting N to the rest of A.

(1) Assuming N remains a subnet of A, how should R1 and R2 be configured? What limitations would still exist with N's use of its separate connection? Would A be prevented from using N's connection? Specify your configuration in terms of what R1 and R2 should advertise, and with what paths. Assume a BGP-like mechanism is available.

(2) Now suppose N gets its own network number; how does this change your answer in (1)?

(3)Describe a router configuration that would allow A to use N's link when its own link is down.

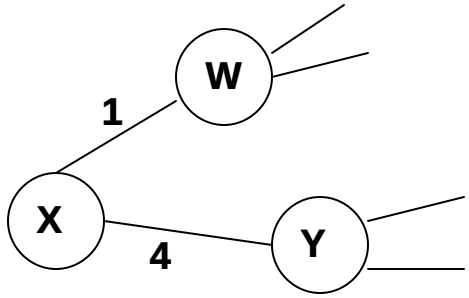
3. Distance Vector Routing

Consider the network fragment show below. X has only two attached neighbors, W and Y. W has a minimum-cost path to destination A of 5 and Y has a minimum-cost path to A of 6. The complete paths from W and Y to A (and between W and Y) are not shown. All link costs in the network have strictly positive integer values.

(1) Give X's distance table (row) entries for destinations X, Y and A.

(2) Give a link-cost change for either c(X, W) or c(X, Y) such that X will inform its neighbors of a new minimum-cost path to A as a result of executing lines 15 and 24 of the distance vector algorithm.

(3) Give a link-cost change for either c(X, W) or c(X, Y) such that X will not inform its neighbors of a new minimum-cost path to A as a result of executing lines 15 and 24 of the distance vector algorithm (the algorithm is shown below).

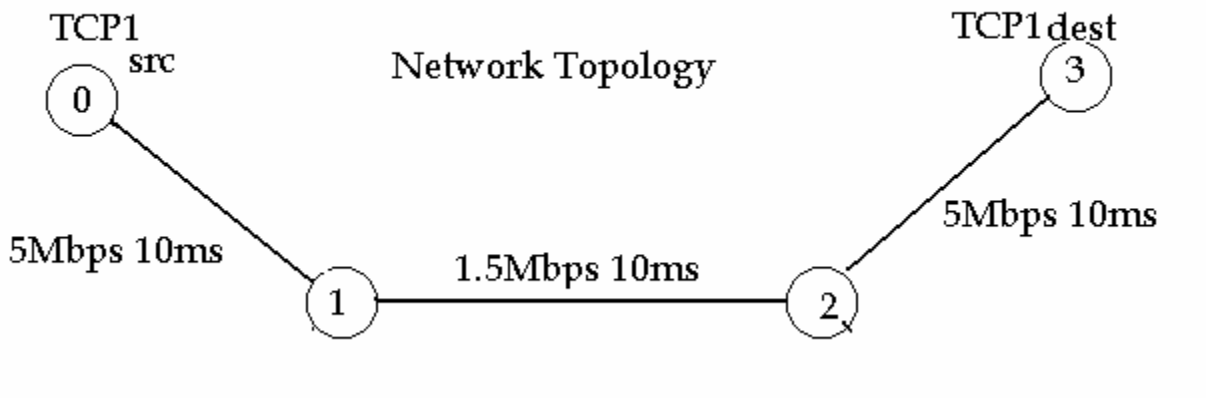


Distance Vector Algorithm
At each node , X:

```
1  Initialization
2     For all adjacent nodes v:
3        D^X(*, v) = ∞         /* the * operator means "for all rows" */
4        D^X(v, v) = c(X, v)
5     For all destinations, y
6        Send min_w D(y, w) to each neighbor   /* w over all X's neighbors */
7
8  Loop
9     Wait (until I see a link cost change to neighbor V
10          or until I receive an update from neighbor V)
11
12    If (c(X, V) changes by d)
13       /* change cost to all dest's via neighbor v by d */
14       /* note: d could be positive or negative */
15       for all destinations y: D^X(y, v) = D^X(y, v) + d
16
17    else if (update received from V w.r.t. destination y)
18       /* shortest path from V to some Y has changed */
19       /* V has sent a new value for its min_w D^v(Y,w) */
20       /* call this received new value "newval" */
21       for the single destination y: D^X(Y, V) = c(X, V) + newval
22
23    if we have a new min_w D^X(Y, w) for any destination Y
24       send new value of min_w D^X(Y, w) to all neighbors
25
26 forever
```

4 Network Simulation
Part A

(i) Setup a network topology as shown below.



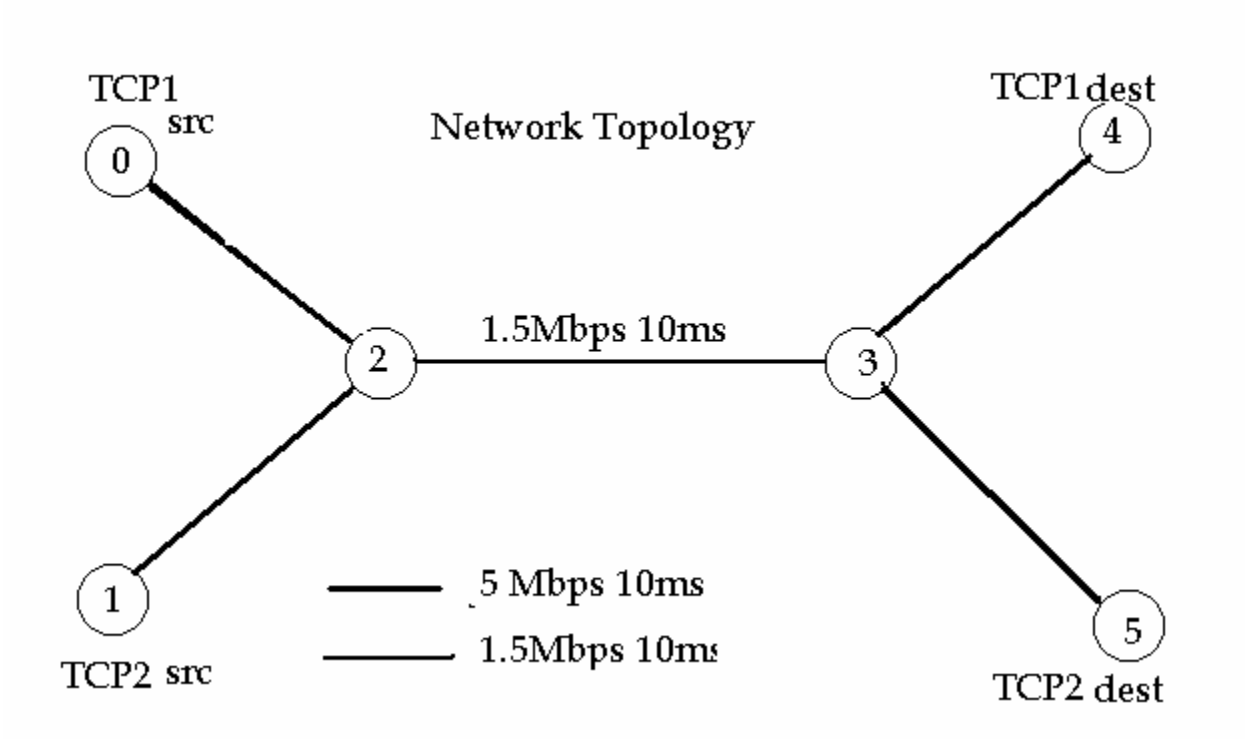
Traffic:

1 TCP connection with source at node 0, sink at node 3 and with an FTP source agent. The FTP flow starts at 0.5 seconds

Terminate the simulation at 12 seconds.

(a) Explain the TCP slow start based on your observation. Also explain the TCP behavior after packet loss

(ii) Construct the following topology.



Traffic:

1 TCP connection with source at node 0, sink at node 4, and with an FTP source agent. The FTP flow starts at 0.5 seconds.

1 TCP connection with source at node 1, sink at node 5, and with an FTP source agent. The FTP flow starts at 0.65 seconds.

Terminate the simulation at 12.0 seconds.

Write an ns script to construct the above network with the specified events.

- The script should use the nam tool to graphically display the scenario.
- Use drop tail queue for all the links and monitor the queue of the bottleneck link between 2 and 3.
- Set TCP window to 200.
- Mark the flows for better visualization (see template).
- Use **FullTcp** agent.
- Print the TCP throughput of the two TCP flows onto the console, at the end of the simulation.

  TCP throughput = (total bytes received by the TCP sink)/(total duration of TCP flow).

  To obtain total number of bytes received by TCP sink. Use

  ```
  set totalbytes [$tcpsink set bytes_]
  ```

  To read current time into 'currentTime' variable. Use

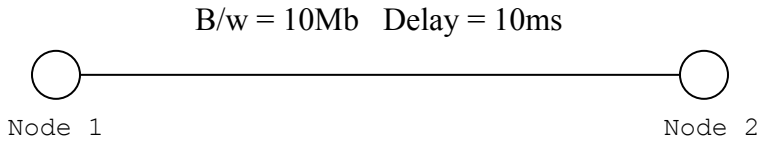
  ```
  set currentTime [$ns now]
  ```

The expression to calculate the TCP throughput should be according to Tcl syntax. Refer to the examples in Marc Greis tutorial or any Tcl programming document.

(b) Describe the observed behavior during and after slow start. Explain how TCP self-clocking results to this behavior.
(c) Consider the scenario #2 where the delay of the duplex link node 3 to 5 is 100ms. Explain how this is different from the previous scenario.

Part B

(iii) Construct the following Topology



Traffic

Create two FullTcp agents and setup according the instructions in template file.

Execute the tcl script and answer the following questions

(a) Submit trace file
(b) Indicate the cwnd_ value when each ack is received till t= 0.35
(c) Indicate values of cwnd_ and ssthresh when the third dupack is received
(d) Indicate the cwnd_ after the retransmitted packet has been acked
(e) Write a short note on how TCP behaves after the packet drop.

(iv) Change the number of drops in the window

Make the following change in the part (iii) tcl file:

Replace line $err droplist { 10 }
by line $err droplist { 9 10 }

Execute the trace file and answer the following questions

(a) Submit trace file
(b) Indicate the cwnd_ value when each ack is received till t= 1.5
(c) Explain the behavior seen.

Submit:

Submit a write up (text format) explaining the above questions
Submit the ns scripts you have generated along with their trace files.

NOTE:

Name the write up as writeup.txt, ns scripts as ns_a.tcl, ns_b.tcl, ns_c.tcl ns_d.tcl,ns_e.tcl and the ns trace files as outa.tr, outb.tr, outc.tr ,outd.tr and outf.tr