Homework 4
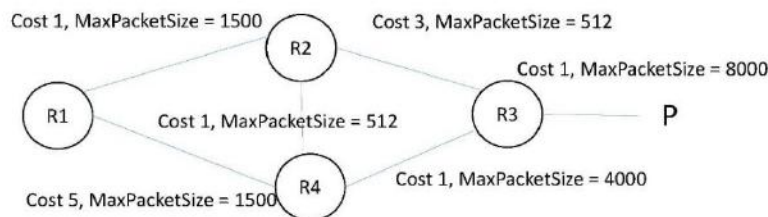
Directions: There are only 2 questions and only Question 2 will be graded for points while the other will be for completion credit only. There will be some final questions based on these problems. While I know you are busy with the project and other Finals preparation, please do try these problems by yourself as a form of exam preparation. We will post solutions before the Finals.
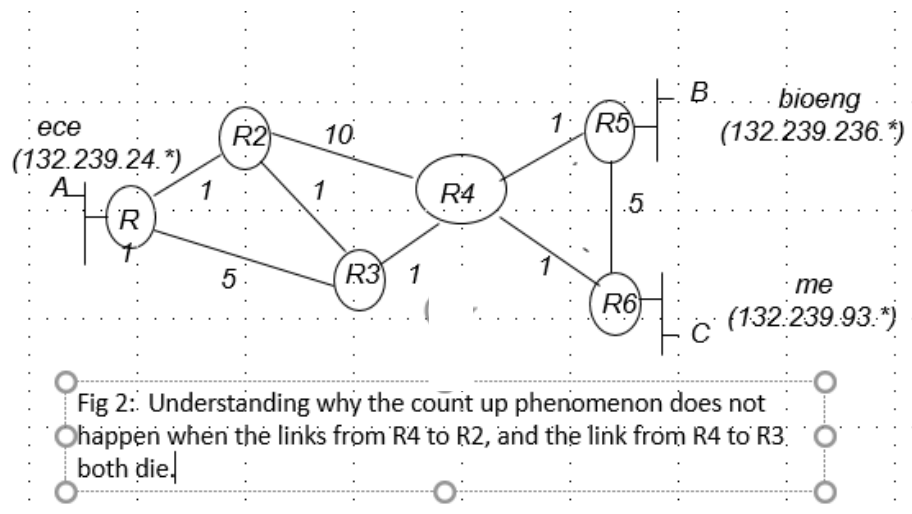
1. **Modifying Routing to Avoid Fragmentation:** We learned in class that in order to avoid fragmentation, IP tries various packet sizes till it finds one that works. Alternately, we could modify the routing protocol to compute the minimum of the maximum packet sizes of all links on the best route to each destination. In distance vector routing, a router R computes its own distance, *Distance* (P, R) to a destination prefix P using the distances sent by its neighbors as follows: *Distance*(P, R) = Minimum across all neighbors N of *Distance*(P, N) + *Distance*(R, N). We want to see how to modify this protocol to **also** compute the minimum max packet size on the shortest distance route to P?

   a) In Figure 1 what is the shortest path between R1 and P? What is the smallest packet size that is guaranteed to get through without fragmentation on this path?

   b). Assume each router R has received the variables *Distance(P, N)* and *MinMaxPacketSize (P, N)* from each neighbor N. Write an equation to compute *Distance(P, R)* and *MinMaxPacketSize(P, R)* from the corresponding variables obtained from all of R's neighbors

   c) Assume we are calculating these estimates only for distances to P. Assume that at t=0, router R3 has *Distance (P, R3) = 1 and MinMaxPacketSize (P, R3) = 8000* and all other routers have the distance and min packet size to P set to a default of infinity. Assume that at t = 0 each router sends an update to each neighbor. Note that t=1 means each router has sent one update to each of its neighbors (i.e. one transmission takes one unit of time). Draw several pictures of the same topology with the changing estimates of each router for its two variables based on the equation you wrote down until all estimates stop changing. After how much time do all the estimates converge (i.e., do not change any more).

   d) Now assume the link to R2 to R4 crashes say at time t = 7. Draw similar pictures for the time it takes to converge after the crash.

2. **Link State Routing and Crashes:** Consider the topology shown in Figure 2. This is the same topology we used in the slides to show the count up phenomenon for distance vector when both the links from R4 to R2, and the link from R4 to R3 went down *using distance vector.* We want to see why the count up phenomenon *does not happen with*

link state:

a) Draw the link state packets sent by R4, R5 and R6 before any link crashes

b) Draw the link state packet sent by R4 after the links from R4 to R2, and from R4 to R3 crash (R5 and R6 stay the same)

c) Assume this new LSP from R4 gets to R5 and R6 in a few msec. Show the Dijkstra tree at R5 and R6 and explain why after R5 finishes its Dijkstra calculations (say a few msec) it will conclude that *ece* is unreachable even though it still has link state packets from R1, R2, and R3, and R2 and R3's old link state packets are wrong!

Cost 1, MaxPacketSize = 1500   Cost 3, MaxPacketSize = 512

R2

Cost 1, MaxPacketSize = 8000

R1   Cost 1, MaxPacketSize = 512   R3   P

Cost 1, MaxPacketSize = 512

R4   Cost 1, MaxPacketSize = 4000

Cost 5, MaxPacketSize = 1500

Figure 1: Computing the Min-Max Packet Size on the shortest path by modifying routing

ece
(132.239.24.*)
A

R2   10   1 R5   B   bioeng
                              (132.239.236.*)

1   1   R4

R   5

5   R3 1   1

                    R6   me
                    C   (132.239.93.*)

Fig 2: Understanding why the count up phenomenon does not happen when the links from R4 to R2, and the link from R4 to R3 both die.