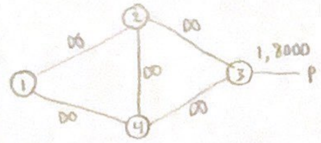
Ethan Wong
305 319 001

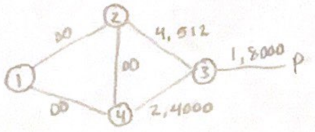
1. A) The shortest distance between R1 and P is: R1 → R2 → R4 → R3 → P. The smallest packet size to get through without fragmentation is 512.

B) Distance(P,R) = min (Distance(R,N) + Distance(P,N))
To find the minimum distance, calculate all possible paths from R to P through all neighbors N. Then select the smallest distance.
MinMaxPacketSize(P,R) = min(minmaxpacketsize(R,N), minmaxpacketsize(P,N))
We want to find the smallest packet size between neighbors and choose the smallest one.

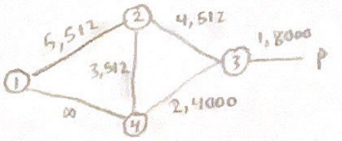
c) t=0



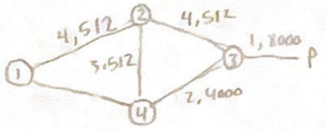
t=1



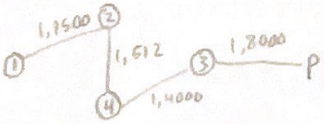
t=2



t=3

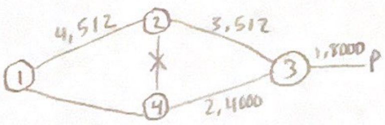


Final:

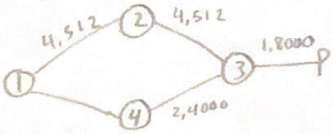


It takes 3 units of time for all the estimates to converge at t=3.

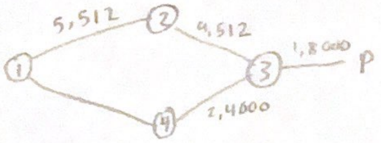
d) t=7



t=8



t=9



It takes 2 units of time for estimates to converge at t=9.

2. a)

R4
LSP
R2,10
R3,1
R5,1
R6,1

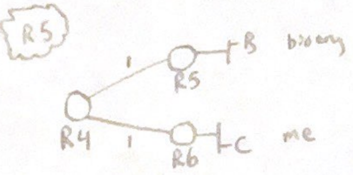
R5
LSP
R4,1
R6,5

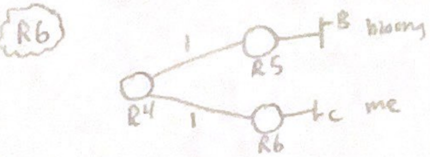
R6
LSP
R4,1
R5,5

b)

R4
LSP
R5,1
R6,1

c) Once R5 receives the updated LSP from R4, it will notice that this LSP no longer has the R2→R4 connection or the R3→R4 connection. Since this is the most recently received LSP, R5 will conclude that these links have crashed, despite having old (incorrect) LSPs from R2 and R3. These old LSPs will just be ignored.


R5
1
R5
B
bloomy
R4
1
R6
C
me


R6
1
R5
B
bloomy
R4
1
R6
C
me