CS 118, Fall 2022
Homework 4
Due Nov. 18, 2022 at 11:59pm

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.

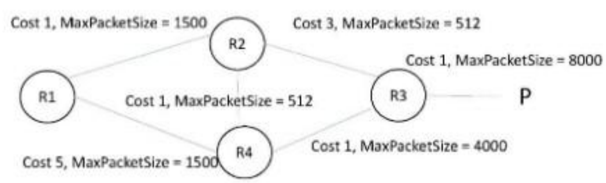


Figure 1: Router topology

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

Solution:

The shortest path is [$R1$, $R2$, $R4$, $R3$, P], which has a cost of 4. The largest packet size that is guaranteed to get through without fragmentation on this path is 512, enforced by the link between $R2$ and $R4$.

(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.

Solution:

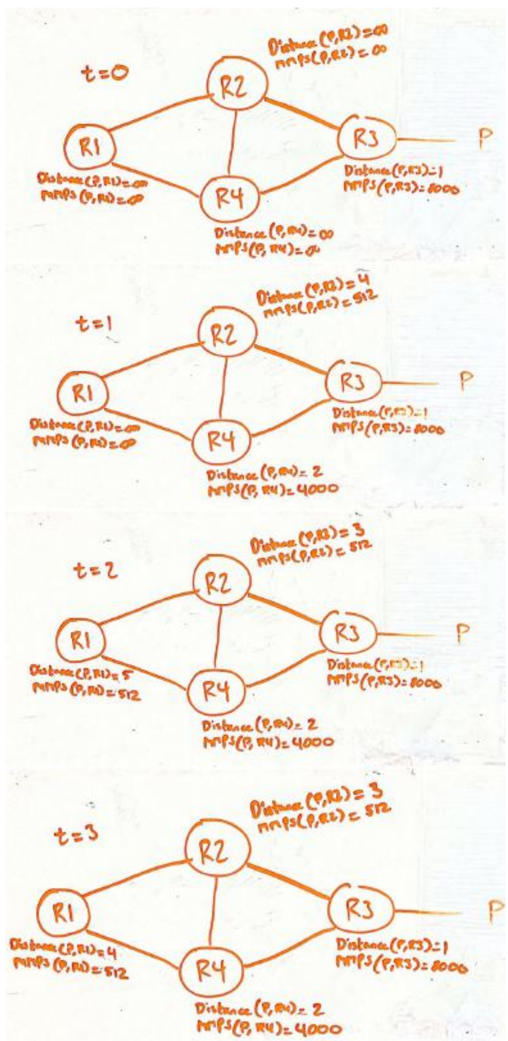
Note: $MMPS = MinMaxPacketSize$ and $MPS = MaxPacketSize$

$$Distance(P, R) = \text{Minimum across all neighbors } N \text{ of } Distance(P, N) + Distance(R, N)$$

$$MMPS(P, R) = \min(MMPS(P, N), MPS(R, N)) \text{ of the } N \text{ that minimizes } Distance(P, R)$$

(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)?

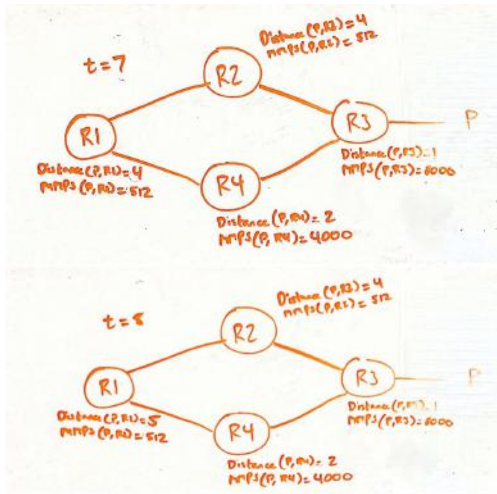
Solution:



The estimates converge at $t = 3$.

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

Solution:



Assuming $R2$ and $R4$ detect the failure and correct their routes at $t = 7$, the estimates will then reconverge at $t = 8$.

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.

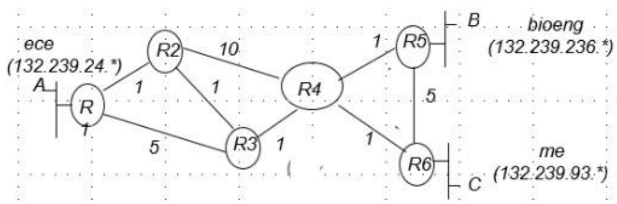


Figure 2: Router topology

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

Solution:

$R4$:

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

$R5$:

$R5$
LSP
$R4$, 1
$R6$, 5
`bioeng`, 1

$R6$:

$R6$
LSP
$R4$, 1
$R5$, 5
`me`, 1

(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).

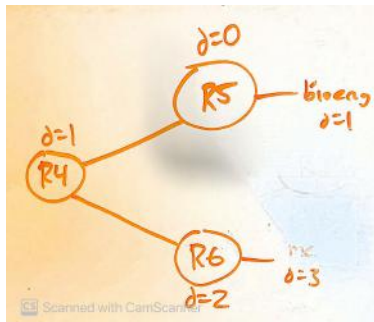
Solution:

$R4$
LSP
$R5$, 1
$R6$, 1

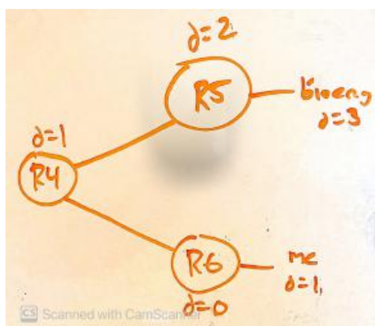
(c) Assume this new LSP from $R4$ gets to $R5$ and $R6$ in a few msec. Show the Dijkstra tree at $R5$ and $R6$ and explain why after $R5$ finishes its Dijkstra calculations (say a few msec), it will conclude that `ece` is unreachable even though it still has link state packets from $R1$, $R2$, and $R3$,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

Solution:

$R5$:



$R6$:



$R5$ and $R6$ will conclude that `ece` is unreachable because, when using Dijkstra's to compute a route, each router only looks at the LSPs of routers within the permanent set. Once $R4$ ends up in the permanent set, its new LSP will be used. This new LSP does not indicate that $R2$ or $R3$ are neighbors, so they will never enter the tentative set. Since they never enter the tentative set, they can never enter the permanent set. As a result, despite having out-of-date LSPs for $R1$, $R2$, and $R3$, they will never be considered by $R5$ or $R6$'s route computation algorithms, as they are never introduced into the tentative or permanent sets.