HW II CS 6390 Summer 2017 Solutions.

Question 1 (MOSPF)

===================

The message will go through routers R3, R2, R4, so the lans traversed will be f, h, e, c

Thus, the routers that actually *see* the message are

R3, R6, R2, R4, R5, R1, R7

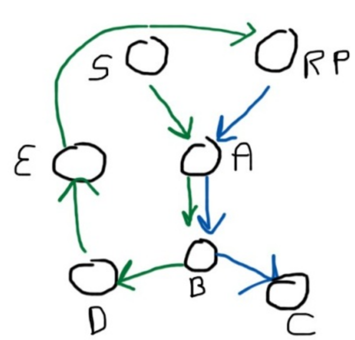
ALL of these routers will create an entry in their cache, even if they are not on the path. The reason being that if they receive the message again (and they will, assuming the source sends more multicast) it would be wasteful to recompute the tree again.

So, the cache entries are as follows.

R1: (S, G, e) [(e,infty),(g, infty)]
R2: (S, G, h) [(e,1),(h, infty)]
R3: (S, G, f) [(f, infty),(i, infty),(h,2)]
R4: (S, G, e) [(e, infty),(c,1)]
R5: (S, G, e) [(d, infty),(e, infty)]
R6 (S, G, f) [(d,infty),(f, nfty)]
R7: (S, G, c) [(b,infty),(c, infty)]

Question 2 (PIM)

==========



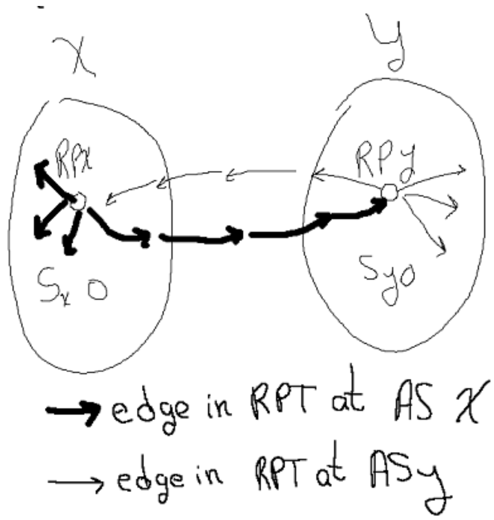
SPT edges are in green
RPT edges are in blue

Note that PIM does not make any assumptions about the underlying unicast routing, and hence, the routing towards the source will form a tree (that is the only requirement we have) and similarly the routing towards the RPT will be a tree.

Note that if A does not prune itself away from the RPT, then data will originate at s, go towards b, d, and e, then to the rp (who has joined the tree of s) and back to a.

Question 3 (PIM Inter-domain)

==========



if we have a source in each of AS X and AS Y, then the RP at each of these two ASms will join each other's tree. and multicast messages may loop around from one AS to another.

This however could be mitigated if border routers do not accept back into their ASm's multicast packets whose source is in its own AS.

Question 4 (OLSR)

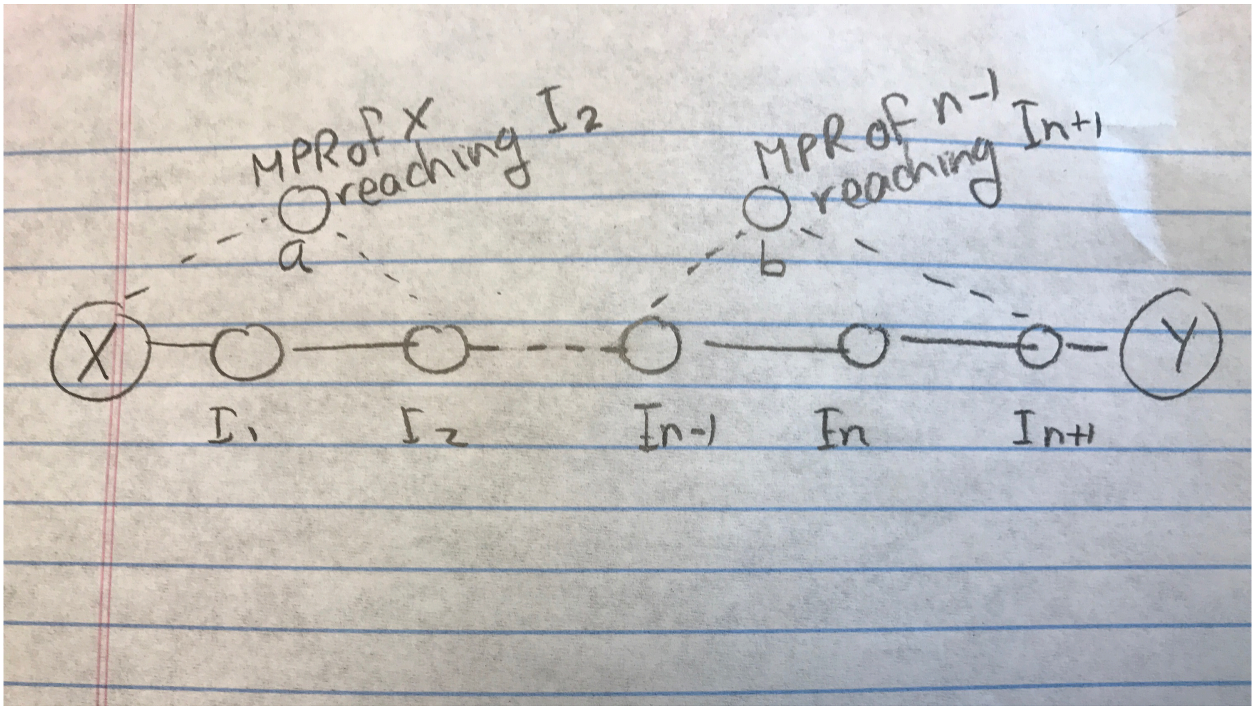
=========

a)

Assume X and Y are two nodes with path consisting only of bidirectional links between them.

Let h be the number of intermediate nodes between X and Y. I.e., if X and Y are neighbors, then h = 0, and if there are two hops from X to Y, then h = 1.

The case when X and Y are neighbors (h = 0) is trivial since Y will hear the broadcast from X.



Proof by induction:
Base case: h=1, i.e., X and Y are 2-hop away from each other, and there is only one intermediate node between them. Call this node I1. I.e., the path is X – I1 - Y.

From the definition of MPRs, there must be an MPR of X, call it “a”, that will forward the message from X and reach Y (recall Y is two hops away in the base case). Thus, the flood will reach Y.

Induction Step:
As shown in the figure, assume that the path from X to Y is

X – I1 – I2 – I3 – etc. – I(n-1) – In – I(n+1) - Y

Assume that the flooding mechanism works for the case when the number of intermediate nodes is n, n >= 1, and show that the flood works for n+1. I.e., we assume that the flood can reach all nodes I1 through I(n+1), and based on this we must show that it reaches Y.

Since the flood reaches I(n-1), and I(n+1) is two hops away from I(n-1), there must exist an MPR of I(n-1) that will forward messages from I(n-1) to I(n+1). Let this MPR be “b” (see figure).

Note that Y is two hops away from b. I.e., there is an edge (b, I(n+1)) and an edge (I(n+1),Y). Hence, from the definition of MPRs, there must be an MPR of b that will forward the message from b to Y. Hence, the message will also arrive to Y.

(b)
Yes, the routing will still work properly. This is because the information given out by the MPR set and the MS set are complementary and we can obtain one from the other. Consider the following example:

A —— X —— Y —— Z

Assume we are given the MS sets for the above nodes:

MS(A) = X
MS(X) = A, Y
MS(Y) = X, Z
MS(Z) = Y

Given this information, the MPR set can be derived from the MS set and would look like this:

MPR(A) = X
MPR(X) = A, Y
MPR(Y) = X, Z
MPR(Z) = Y

Even though both sets look alike in this case, even in the general case, no information is being lost or hidden when we replace the MS set with the MPR set in the TC message of each node.

Question 5 (DSR)
=============

Assume S is sending data to D along the path S->A->B->C ->D

Assume another path exists: S->Z->X->Y ->B->C ->D

Also, assume X has in its cache the path X->Y ->B->C->.....->D Suppose link B->C dies.

B sends router error message to S.

If S does not do piggybacking of Route Error on the new Route Request message then X will return to S an early route reply with the path S->Z->X->Y ->B->C ->D, which is invalid because link B->C is dead.

Question 6: Virtual Circuits
===================

a) Because the VID is used as an index into the VC table of B (not of A), and hence, B is the one that has to determine which entries in its own table are free and thus available for the new VC.

b) Because we use them to distinguish between the VC's (flows) on the same link, and these cannot be many (well, relatively not many, say, a few thousand). I.e., a router (or switch) is only involved in a relatively few flows at a time.

c)

(i) Build a circuit from A to D, and give it VID 10. For each of the other four individual circuits, we need to do a stack of VID (similar to a stack of MPLS labels). E.g., for circuit #5 from p, as it arrives to A, it Pushes a new VID on top of the old one (PUSH, NOT replace). It is then sent along circuit 10. When it arrives to D, the top VID is popped, restoring the VID #5. This is then used at D to determine the link and VID (i.e., 21) for the packet.

(ii)

I did not label the links, so I will only use the VIDs.

A:

Input VID	Output VID	PushNew VID?
5	10	Y
11	10	Y
20	10	Y
7	10	Y
19	10	Y

The PushNew VID indicates that the VID 10 has to be pushed (stacked) on top of the previous one.

D:

if packets come in with VID 10, their label is popped and the new top label examined, and the VC table is used again using the new top label.

Input VID	Pop?	Output VID
10	Y	none
5	N	21
11	N	15
20	N	3
7	N	2
19	N	4

Question 7: MPLS
=============

A has a view of the whole network (due to OSPF), so consider a neighbor W. A could find a path to W (other than directly to its link), and construct an MPLS path by source-routing an RSVP message towards W (along the new path). If the link between A and W fails, then data is sent to W by adding an MPLS label and send it on its way. At W the label is popped and normal IP routing resumes.

Perhaps an even better way is to locate all your two-hop neighbors (all nodes two-hops away from A) and establish MPLS tunnels with them. This is because if the link to W fails, you don't know if the link failed or the node itself failed, so you can bypass W altogether by establishing an MPLS path to the neighbors of W.

E.g., Assume we have A ---- W ---- P

Establish an MPLS path from A to P. You would then use this MPLS path for data whose shortest path to the destination goes through P (or at least it was the shortest path before A -- W died). Again, we know that this is the case from OSPF who gives us the whole topology of the network.

Question 8: Fair Queuing
==================

a) In every round, a "bit" is forwarded from every flow in the queue of the fake server. So, regardless of how many flows are queued, one bit from each flow is forwarded per round (the only thing that changes is the duration of the round)

Thus, once F(p1) is calculated, it indicates the round in which it will exit. The arrival or termination of other flows only changes the duration of the round, so F(p1) remains constant for p1.

This, by the way, is what makes fair queuing possible, we don't need to recalculate the labels of packets once they are assigned.

b)
V(t1) = X

from t1 to t2, the fake server forwarded (t2-t1)*C bits.

In each round, it forwards B bits

Hence, (t2-t1)*C/B is the number of rounds that occurred from t1 to t2.

Thus V(t2) = V(t1) + (t2-t1)*C/B = X + (t2-t1)*C/B