Homework 2, CS 6390 Spring 2012

Question 1

(a)

LAN	Parent
(a)	R8
(b)	R8
(c)	R4
(d)	R8
(e)	R2
(f)	R3
(g)	R1
(h)	R3

(b)

Leaf LANs: (a), (b), and (c)

Truncated: (b) and (c)

(c)

R4 and R1 send NMRs to R2, and R2 cannot send an NMR because there is a receiver on LAN (e)

(d)

The mcast message is sent over LANS (i), (f) (d) (a) (h) (e)

Question 2

I am going to assume (and hopefully you did too although I did not explicitly ask it in the question) that we don't have any knowledge of WHICH unicast routing protocol is being used. We only know that there is a unicast protocol, and it can give us the next hop to the source (this is the same that PIM assumes)

Note that in DVMRP, the only time we need distance vector is to:

a) Know if I am the parent of a LAN
b) Know if I have any children routers on the LAN

Since we don't have this information, we have to send extra messages to figure it out.

I would have each router periodically broadcast a "hello" message on each LAN. In this way, each router knows if there are other routers on the LANs

Upon receiving the first message of the source, we forward it along all LANs that have either receivers (IGMP) or routers (from the hello protocol), and we forward it only if it comes from OUR next hop to the source (basically reverse-path forwarding)

All routers in a LAN realize a source S is active, so they broadcast “assert” (call it whatever you want) messages on each LAN (except their next hop to S). In the assert message, they indicate their cost to S (hopefully also given to us by unicast algorithm). This message is used to break ties among possible parents. In the LAN that is the next hop to S, the router broadcasts a “child” message, indicating to other routers that the router uses this LAN to receive multicasts from S.

From these messages, a) and b) above can be determined. The rest is just as in DVMRP

Question 3

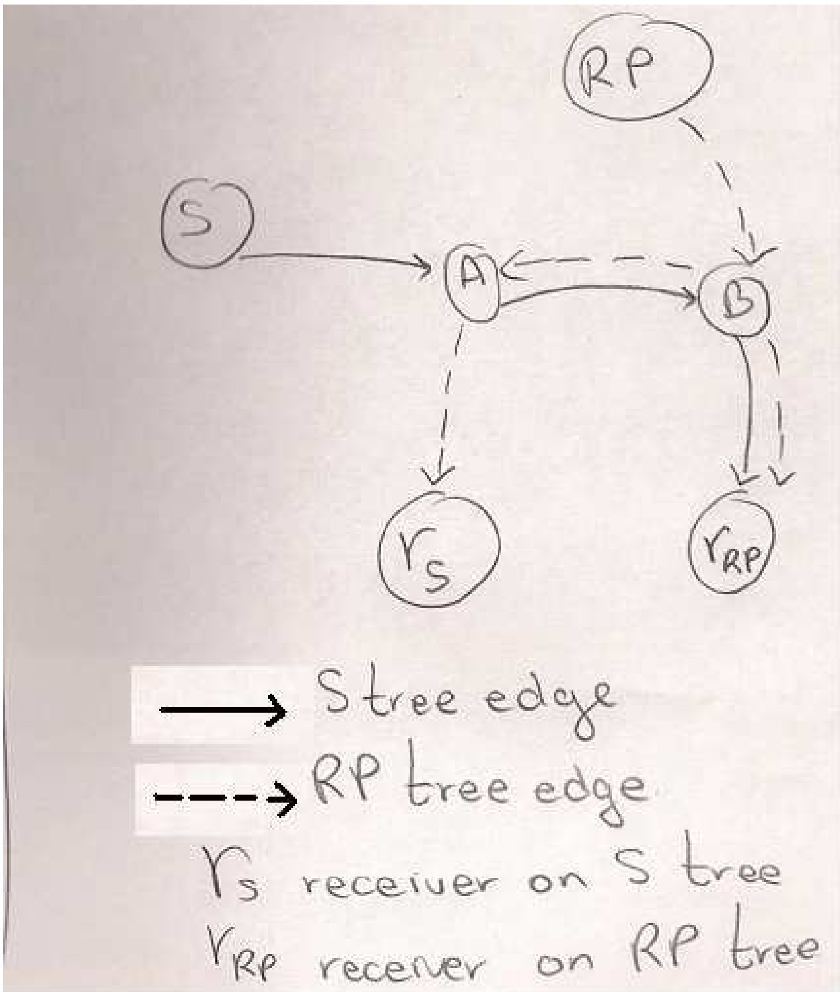
The mcast message is sent over LANS (i), (f) (d) (a) (h) (e), assuming we break ties the same way (in favor of smaller IP address)

The only router that does not see the multicast message is R7. Thus, all other routers must have a cache entry for the (S,G) pair

Depending on how you count, you may put a 0 instead of a 1 for a directly attached receiver, no problem.

R1
(S, G) iif: (e) links: (e, infinity) (g infinity)
R2
(S, G) iif: (h) links: (e, 1) (h, infinity)
R3
(S, G) iif: (i) links: (h, 2) (i, infinity) (f, 3)
R4
(S, G) iif: (e) links: (c, infinity) (e, infinity)
R5
(S, G) iif: (e) links: (d, infinity) (e, infinity)
R6
(S, G) iif: (f) links: (f, infinity) (d, 2)
R8
(S, G) iif: (d) links: (a, 1) (b, infinity)
R9
(S, G) iif: (d) links: (d, infinity) (a. infinity)

Question 4



Note that if we use min-hop routing, and the links consist of only those indicated by the edges, then min-hop routing would create the above trees.

To prevent the multicast message from going around from one tree to the next, in PIM B would prune itself (selectively for S) from the RP tree (since it has a parent in S), similarly, A would selectively prune itself from the RP tree (i.e. from B).

Question 5

(a) When a node S wants to send multicast data, it will send message to notify the RP1 in its domain. This RP1 via MSDP will broadcast SA(source active) message to the peer RPs. When a node N wants to receive data from S, it will send join message to its own RP in the domain, say RP2. RP2 will decide to join the shortest path tree (SPT) by sending join message along the path to DR of S, this causes a path to be built in the SPT of S. Then node N will join the SPT of S, and S sends data to N along the SPT.

(b)

(i) The router should join the RPT of the group X, and hence, forward the join to the next hop along the unicast path to the RPT of X.

(ii) Drop it, since it makes no sense.

(iii) If the router is attached to S, encapsulate it and send it to the RPT of X. If not, then forward to its children on the SPT (S,X) AND also to its children on the RPT of X.

(iv) If the packet is received along from its parent on the SPT (S,Y), then forward to its children on this tree. DO NOT forward to children on the RPT of Y (there isn't any since this is a source specific tree)

Question 6

Consider the following network path:

A → B → C → D

The arrow indicates the next hop to the destination, which is node D. Assume D's seq no at A is less than that at B which is less than that at C which is less than that at D.

Assume C timeouts first, and forgets about D (removing it from its routing table). We thus have

A → B → C (C has no next hop to reach D, not even information that D exists, since D's entry in C's routing table is removed)

Assume that some node X sends a RREQ looking for D. This RREQ passes via C (which lets it go by), and assume C and A are neighbors (C moved closer to A or whatever) and the RREQ arrives at A. A, since it has information about D, sends an early RREPLY back to X (and hence along C). C then learns that D is in the direction of A, and points to A

A → B → C → A

a loop is thus formed.

The standard has a way to avoid this (don't recall it at the moment), but the above is possible given what we have covered about the protocol in class.

Question 7

Let's assume a source S is looking for destination D, and node X is the previous node toward destination D. Assume there are two paths toward X where one is longer than the other. If RREQ arrives at node X from the longer path first then node X will be forwarding this RREQ that has longer path. Node X will filter the RREQ that comes from the shorter path since it has already transmitted a RREQ with the same broadcast ID. Hence, shortest path to D will not be discovered.

Question 8

a. Non-label merging must be used. If we use label merging, given that IP packets are broken into multiple ATM cells, it is possible that cells from different packets (arriving over different links) could be merged together before being forwarded (note, they have the same VC ID due to label merging). The destination then would not be able to separate the cells into those that belong to different IP packets.

b. Under that reassembly method, what a switch can do is, on each input VC, as cells arrive, buffer all cells until a 1 is received, and then transfer all the cells at once into the output queue of the output VC (without merging them with cells of another input VC). In this way, cells from different IP packets are not merged together. Inefficient, but effective.