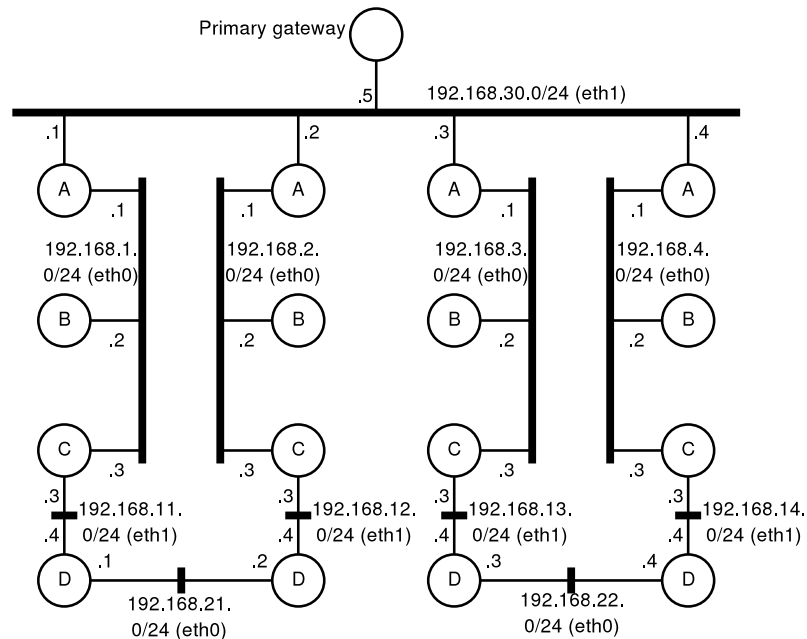


CSIS0234B Computer and Communication Networks (Class B)

Tutorial 11

The multicast routing daemon

We will use the programs we wrote during the last tutorial to examine how multicasting works over multicast routers. The programs are placed in the computer, under the home directory of the `root` user. The network is configured in the same way as the configuration during our OSPF tutorial. It is repeated here for ease of reference.



The OSPF configuration is already in the computer. We on purpose shuts down the link between the D's in the configuration, however. We use `mrouterd` on C so that B and D can exchange multicast packets.

1. Login as `root` in text-mode to shut down `eth0` using `ifdown`. Then go back to GUI mode, login as `root`, start the `zebra` and `ospfd` services, and stop `iptables`. Wait until the OSPF routing tables stabilize.
2. Compile the programs of the last tutorial. Try to send multicast packets from B, C and D to the group `239.255.0.x` (x is the group number) when all of them are receiving packets. Observe that D never receives a multicast packet sent by B and C, and vice versa. This is because C doesn't listen or send multicast packets to `eth1`.
3. Now run `mrouterd` at C, without any configuration. With `ps -x`, confirm that `mrouterd` is running. Try the above experiment again. Can host D sends multicasts to other hosts?
4. Repeat the experiment, this time capturing the packet using `ethereal` (with a filter `!proto ospf` if you don't want to see all the OSPF packets), and find the TTL set for the IP packet. Why in the last step you cannot forward a packet?
5. Modify the sender program so that it sets the TTL to 2 before sending the packet. Retry the experiment, and confirm that the packet is indeed received.
6. Try to use the multicast address `224.0.0.20` and `224.0.1.20` respectively and see what effect you can see.

7. Now send a SIGUSR1 signal to the mouted process (with `killall -USR1 mouted`). Look at the file `/var/tmp/mouted.dump` to see the current multicast routing table. Read the man page of `mouted`, in particular the EXAMPLES section at the end of the man page, to understand the information presented.
8. Build a tunnel with the group opposite to you. It should have threshold 1 and no boundary. Remember that apart from configuring `mouted` in `mroute.conf`, you must load the `ipip` module before restarting `mouted`. Then stop and restart `mouted`. Can multicast of your network can reach to the other side and vice versa? Try to change the TTL value to an even larger value (say 10) and try again. Remember to use a small file for this step, since large IP packet cannot work through the tunnel.
9. See what is the smallest TTL value that succeed in getting through. To see why this TTL is needed, use `ethereal` to check the two IP headers of the IP packet received.
10. Send a multicast datagram at an address that the other side is **not** listening, while capturing packets. See how `prune` and `graft` messages are used to prevent future reception of the messages¹.
11. Now send a SIGUSR2 signal to `mouted`, and look at the routing cache stored in `/var/tmp/mouted.cache`. Note that pruning are recorded in the routing cache. In a larger network with many groups, `mouted` must keep a lot of information about the groups.
12. Modify the tunnel so that it is a boundary for the range `239.255.0.0/16`. Restart `mouted`. Confirm that multicast messages within the range no longer work through the tunnel. Also check that the routing cache (sending a SIGUSR2 signal to `mouted`) contain such boundary information.
13. (Optional) Enable the connection between the D's of your group and your opposite group. Run `mouted` without any configuration in both D's. See the multicast routing table (by sending a SIGUSR1 signal to `mouted`). Understand the multicast routes at the bottom.

¹Pruning is done between multicast routers, so without any multicast router to communicate with, there won't be any pruning before connecting to another multicast router: there is no multicast router to prune messages from.