

CSIS0234B Computer and Communication Networks (Class B)

Reading for Tutorial 11

Inter-network Multicasting and mouted

Last tutorial focus on a simple case of multicasting, where the multicast packet is received by the recipient because the packet is broadcasted on the physical broadcast medium, and the network card is configured to listen to the particular multicast address. But for multicast packet to work across networks, there must be a router which will forward multicast packets from one network to another.

Multicast packets can be routed using regular routing daemons, if they are supported. Unluckily, most routers do not support multicasting. Indeed, none of the Internet routing protocols (RIP, OSPF, BGP) we have learnt has any facility for multicasting; and commercial routers do not yet support multicasting. The reason is that multicasting is a much more complicated issue than unicast routing. Indeed, there are multiple protocols under development to perform multicasting, and they disagree in principles. We will try out an early protocol and its implementation to get concrete ideas about the concerns involved.

1. Concerns of Inter-network multicast routing protocols

Just like multicast in the datalink layer (i.e., using broadcast capability of network card directly), inter-network multicast works by some senders sending to a multicast IP address, or *multicast group*; and if at that time some receiver joined that group, the receiver will get the packet. The sender does not need to be a group member before it can send a multicast packet. And as we have briefly examined in the last tutorial, the range of the multicast may be limited by the TTL of the packet and the multicast address being used.

If a packet (with sufficiently large TTL) is sent to a group that most people joins, multicasting is the same as inter-network broadcasting. Therefore, multicast includes broadcast as a special case. In the converse, inter-network broadcast mechanisms can be used to implement multicasting: just send the packet to everybody. Those computers which joined the group gets the messages and those which didn't drops it. So multicasting and broadcasting is highly related. Broadcasting mechanisms like Reverse Path Forwarding (RPF) plays an important role in multicast routing.

However, to be a useful multicast mechanism, the router must try to minimize the number of occasions when a packet has to be sent and forwarded by a router even when actually there is no hosts in the group which is interested in the group. Otherwise it is just a broadcast mechanism, and it is quite clear that if a Internet-wide broadcasting mechanism becomes widely available, every router will be slashdotted¹. Such "noise reduction" is called pruning: a router finds that no reachable host or router is interested in the message, and thus drops the packet. To allow this, the hosts must tell the routers that it joins and leaves a group; and routers must have a way to exchange information about what groups they are interested in.

So any multicasting protocol has two basic decisions: how to obtain enough information to perform inter-network broadcasting (i.e., a routing table), and how to obtain enough information to perform pruning. Different protocols have vastly different strategies to perform the tasks.

¹Slashdot (<http://slashdot.org/>, the name purposely chosen to be short but very difficult to pronounce) is an Internet technical news site. Since it has a very good selection of news, it becomes very popular—so popular that when news items appear in Slashdot, a huge amount of people will visit the sites referred to by the news item. At many times the referred sites are not prepared for that amount of traffic, and becomes very slow or plain inaccessible. The readers of Slashdot would say that the referred site is "Slashdotted".

2. DVMRP

One of the earliest attempts to support inter-network multicasting is the Distance Vector Multicast Routing Protocol, DVMRP. In DVMRP, the routing table for performing RPF is found using the Distance Vector protocol, separated from any (unicast) routing protocol used in the underlying network.

DVMRP takes a very passive approach to pruning. DVMRP routers listens to connected hosts about their joining and leaving of a group, so each of them keeps accurate information about which groups its hosts are interested in. However, normally it does not forward such group membership information to neighbouring routers. As a result, when somebody wants to send a packet to a particular group, a full broadcast is performed using RPF. When a router receive such a packet and finds that it is a leaf (i.e., it cannot send messages to its peer), a *prune message* is sent to the originator of the message, asking the originator to stop sending messages to it. The originator might find that all its links has pruned a group, and thus further prunes towards the ultimate sender. If a router which prunes a group later finds that it wants the group again, a *graft message* is sent to undo the prune message.

3. Mrouted and Tunneling

The most popular implementation of DVMRP is a Unix program `mrouted` developed by Stanford. This is in spite of the fact that it is not completely free software, thus is not installed by most distribution and must be separately compiled and installed; and in spite of that the program is designed for BSD, and requires a patch to allow it to work on Linux.

As said at the beginning, most Internet routers are not multicast-ready. So a multicast routing protocol that relies on having every participant to be connected by multicast routers would have little practical use. The issue is solved by allowing the use of **tunnels** to connect parts (or islands) of the Internet which have multicast capability. When an IP multicast packet arrives at one side of the tunnel, it is completely encapsulated in an IP unicast packet and send to the other side of the tunnel. The other side will then extract the IP multicast packet and send it through its own network. In this way, multicast packets can work through parts of the Internet that does not support multicasting. The DVMRP distance vector protocol will consider the tunnel as a simple link.

In Linux, the encapsulation required for a tunnel is done by a kernel module `ipip`. This module must be loaded before a tunnel can be used by `mrouted`.

4. Mrouted configuration and operation

One starts `mrouted` by simply running the program, and stops it by killing it. By default, `mrouted` listens to all broadcast network interfaces, and forwards messages from each interface to all other. No tunnel is created. If there is less then 2 interfaces, then `mrouted` cannot start (because it has no interface to forward packets onto). The `/etc/mrouted.conf` file can be used to modify these defaults. A `phyint` (physical interface) section determines which interfaces to turn on and its operational parameters. The operational parameters includes which ranges of multicast addresses not to forward to that interface (default none), a metric parameter which allows you to slightly affect the routing (default 1), and a threshold parameter which tells that if a packet has TTL less than the threshold it should not be forwarded (default 1). E.g.,

```
phyint eth0 boundary 239.255.0.0/16 threshold 16 metric 2
```

says that for a packet to be forwarded to the `eth0` interface, it must not be an address in the range from 239.255.0.0 to 239.255.255.255; and it must have $TTL \geq 16$. It has a metric of 2, meaning that it should be considered to have double cost when the distance vector algorithm is executed. Tunnels are similar, but apart from an interface, one has to specify the address of the other end of the tunnel. A tunnel can operate only if both sides configures the tunnel correctly. E.g., the following specifies that the other end is 192.168.6.3.

```
tunnel eth0 192.168.6.3 boundary 239.255.0.0/16 threshold 16
```

The file can also contain other options, which are described in more details in the manpage. The sample config file is very readable, though, so you might not really need to consult the man page to understand what each option means.

The interface of `mrouted` for inspection of its internal state is quite crude. In particular, you should send it a signal `SIGUSR1` or `SIGUSR2`. The former puts its current routing table to the file `/var/tmp/mrouted.dump`; the latter puts its current pruning information to the file `/var/tmp/mrouted.cache`. See the man page of `mrouted` if you want to see an example.

5. Other methodologies

As an early experimental system for a task as complicated as inter-network multicasting, `DVMRP` and `mrouted` have many deficiencies, heavily reducing the scalability of the system. One of the problems is the use of distance vector with small infinity value (infinity is 16), which makes it impossible to go through more than 16 hops (and less if some of them set a larger metric), and also have the well-known convergence problem. More recent protocols would use link state routing instead. In fact, many of them will re-use the sink trees obtained from an existing routing protocol, e.g., `OSPF`, to support multicasting.

But more importantly, more recent protocols would use hierarchical routing: divide the Internet into regions, each of these regions would elect a “core” router to do the top-level routing. Each router in the region would then use the core to send and receive multicast. In this way the costly multicast routing can be limited to a small number of core routers.

The passive approach to pruning is another serious drawback of `DVMRP`. Whenever a new group address is used to send a packet, it is sent to every participating routers, spending large amount of bandwidth. The prune message then propagates slowly to reduce the traffic. Furthermore, after a prune message is sent, the router must remember where the prune message is sent to, so that graft message can be sent to the correct routers if the group is later needed. This pose large memory requirements on the routers. Recent protocols that employs core routers would allow part of this process to be reversed: normally multicasts packets are not forwarded. When a host joins a group, the router forwards *join messages* towards the core router, until a router which is already listening to that group. That router than start forwarding the packets towards the interested host.