IGMP snooping on ethernet switches

Let's see what happens if a program joins an IP multicast.

First start `tcpdump -v -n -i eth0 igmp`. Then start `vlc udp://@239.192.1.1:1234`. **tcpdump** will show something like this:

```
13:33:52.031769 IP (tos 0xc0, ttl 1, id 0, offset 0, flags [DF], proto
IGMP (2), length 40, options (RA)) xxx.xxx.xxx.xxx > 224.0.0.22: igmp v3
report, 1 group record(s) [gaddr 239.192.1.1 to_ex, 0 source(s)]
13:33:56.391736 IP (tos 0xc0, ttl 1, id 0, offset 0, flags [DF], proto
IGMP (2), length 40, options (RA)) xxx.xxx.xxx.xxx > 224.0.0.22: igmp v3
report, 1 group record(s) [gaddr 239.192.1.1 to_ex, 0 source(s)]
```

This means that your machine wants to join the multicast group `239.192.1.1`. Such an `IGMP JOIN` packet has an impact on two devices of your network:

- A (multicast capable) IP router checks if it receives multicast traffic for this group on any of its (IP) interfaces and forwards this traffic. Depending on the multicast routing configuration, this might also involve other routers as IP multicast can span several subnets. It might also involve a multicast routing protocol such as PIM. All this is layer 3 (IP) stuff and of no interest in small networks with only one IP subnet.
- A (IGMP snooping capable) ethernet switch will also examine this IP packet (yes, it examines layer 3 information although it's actually a layer 2 device, but this is usually only done for IGMP JOIN packets). As soon as it receives such a packet on a specific port, it will forward traffic for the specified multicast group to this port.

You might wonder how the switch can recognize the multicast traffic as it actually is a layer 2 device and the destination IP address is layer 3 information. The answer is that multicast traffic also has a special destination MAC address. Look at this tcpdump output:

```
13:54:35.686406 00:1b:fc:20:88:b9 > 01:00:5e:40:01:01, ethertype IPv4
(0x0800), length 1358: xxx.xxx.xxx.xxx.33424 > 239.192.1.1.1234: UDP,
length 1316
```

01:00:5e:xx:xx:xx are multicast MAC addresses. Use Google for more information on these addresses (it isn't a 1:1 mapping from IP multicast addresses!).

So, this is what «IGMP snooping» is about: An ethernet switch implementing this feature will recognize IGMP JOIN packets, compute the MAC addresses belonging to the subscribed multicast IP addresses and put these MAC addresses into its MAC table. A switch which doesn't do IGMP snooping will broadcast ethernet frames with destination address «01:00:5e:xx:xx:xx», thus flooding the network.

Why an IGMP querier is required

However, this isn't the whole story. For several reasons, an IGMP snooping capable switch will remove the multicast MAC addresses from its MAC table after a certain amount of time. Furthermore, if you start tcpdump and vlc as shown above, you can leave them running for several hours and you'll notice that no IGMP reports are sent after the initial two. This has the effect that a machine subscribed to a multicast group and connected to an IGMP snooping capable switch will receive the multicast traffic

for some minutes but then the reception suddenly stops (as the switch has expired the MAC address from its table). To prevent this effect, you'll require an IGMP querier on the network. It will periodically send out IGMP QUERY packets and the multicast subscribers will respond with an IGMP JOIN packet. This packet will again be recognized by the snooping switch which will refresh the MAC table entry and the timeout problem is gone.

If your existing router doesn't support multicast querying, you can put such a service onto the machine which is running the dvbyell service. You have two choices for this:

- xorp (eXtensible Open Router Platform): At first sight this might seem to be an overkill, as xorp supports BGP, OSPF, RIP, PIM etc. However, it will only do the protocols which are explicitely configured, and IGMP querying can be used without enabling all the other cruft, so it's just fine for our purpose. Get it from http://www.xorp.org. Compiling it takes some time but worked without problems for me. For your convinience the dvbyell tarball includes a configuration file which only enables IGMP querying (xorp/config.boot)
- mrouted (DVMRP multicast router): This is a rather ancient piece of software (last version released in 1999). You can get it at ftp://ftp.research.att.com/pub/fenner/mrouted/ but you'll need to apply the patch coming with the dvbyell tarball (patches/mrouted-3.9beta3+IOS12+linux.diff) to compile it on a Linux system. One of the problems with mrouted is that it expects more than one network interface or it will refuse to start. You can disable the if block for this check in vif.c (line 114) and recompile it to work around this. If you start it then, it seems to work at first. However, it somehow killed the network connectivity on my machine several times, so I can't really recommend using it. Also note that it apparently is not possible to turn of the DVMRP routing machinery and that mrouted is not open source software.