



SnIP Application Guide

SnIP Multicast Routing

Revision History

Rev 0.1 7-28-2010 Initial Release.

1.0 SnIP Multicast Routing Overview

The purpose of this document is to explain the procedures necessary to achieve IP multicasting on both Point to Point (PtP) and Point to Multipoint (PtMP) network configurations using a SnIP as the data interface.

A good reference of IP multicasting can be found on [this Wiki page](#). This document serves as a quick start guide to get you started with multicasting for your Datum modems equipped with SnIP interfaces.

Many of the procedures described below are done via a telnet or console session. If you have never used telnet before, please refer to our *SnIP Quick Start Guide* on our website. There are also many nice Linux command line tutorials online.

1.1 Easy Multicast Setup

All of the discussion and procedures below are intended to provide an understanding of what is necessary to use multicasting in common network link types. The SnIP filesystem versions 0.6.14 and later "configwiz" program, available from console or telnet sessions, is capable of setting up either of the multicast protocols and other associated settings. This is done by selecting "Multicast Router" in configwiz and choosing either the "mouted" or "smcroute" router and saving the settings.

Using configwiz also makes the multicasting settings persistent so that they are recovered on power-up.

You should however read the remainder of this document to learn when and where these routers are used. Case examples of systems are also shown with procedures for setting up the router and network parameters.

1.2 Multicasting Protocol Used

The routing protocol we have provided in our SnIP interface is Distant Vector Multicast Routing Protocol (DVMRP). In order to enable the SnIP interface to forward multicasting packets a multicast routing daemon has to be started.

- Mouted: This is a dynamic multicast routing daemon. A separate multicast routing table is constructed automatically. The daemon is ideal to use when the network is a point to point link.
- SMCRoute: Since Snip Ver. 6.12, the smcroute daemon was added to the File System. This is a “static” router which requires that users define each multicast route manually. This routing daemon is ideal for networks that are point to multipoint where unidirectional links are involved. Unidirectional links cannot use a dynamic router.

Instructions on how to use these two daemons are provide in the sections below.

1.3 Multicast Checklist

Before we get started, please make sure you have the following prerequisites for multicasting met.

For all the SnIP interfaces, you must have:

- If you desire to have static multicast routing with smcroute, SnIP Ver. 6.12 or later. The latest software can be found on our website.
- IP Forwarding is enabled. This can be done by type the following command in the command line:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

The default setting is set to have IP Forwarding enabled. Therefore, this step is unnecessary for most users. However, it is a good practice to make sure that you have all the right settings.
- Multicast should be turned on for both eth0 and hdlc0 interfaces. This can be checked with the “ifconfig” command in the command line. The highlight in **Figure 1** shows that both eth0 and hdlc0 are enabled for multicasting.

```
[Trans:1 /root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:19:78:00:03:3A
          inet addr:10.11.1.2  Bcast:10.11.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:645 errors:0 dropped:0 overruns:0 frame:0
          TX packets:302 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:60976 (59.5 KiB)  TX bytes:19302 (18.8 KiB)

hdlc0     Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.200.1  P-t-P:192.168.200.1  Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:85 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1200
          RX bytes:0 (0.0 B)  TX bytes:1870 (1.8 KiB)
          Base address:0x3f00
```

Figure 1: Both interfaces have multicast turned on

If for some reason, one or both of the interfaces are disabled for multicasting the user can enable it by typing:

```
iplink set <dev> multicast on
```

where <dev> should be replaced with either eth0 or hdlc0.

2.0 Point to Point Multicasting

Note: It is assumed that a point to point network has already been setup successfully. Also make sure you have all the prerequisites listed in the previous section.

It is extremely easy to enable multicasting on a point to point link. All you have to do is enable the mrouterd daemon in both of the SnIP interfaces by typing in:

```
mrouterd
```

At this point, you should be able to pass multicast packets across your satellite link.

You can make sure the daemon is enabled by requesting the current process status with the command:

```
ps
```

You should see “mrouterd” listed.

Note: Using the mrouterd command manually from the command line is not persistent. It will be lost on reboot of the SnIP. Use configwiz to set the router protocol persistently. The dynamic routing protocol automatically generates the routing table entries by talking with other dvmp routers in the network.

2.1 Mrouterd Routing Table

In order to view the populated routing table, you have to know the current pid number of your mrouterd daemon. This pid number is conveniently stored in the file /tmp/mrouterd.pid. Type in:

```
cat /tmp/mrouterd.pid
```

to get the pid number.

In some cases, it takes as long as a whole minute to start this daemon.

Like any other Linux programs, you can type “mrouterd -h” for help with this program.

To browse the routing tables, typing the following:

```
kill -USR1 <pid number> <ENTER>
cat /tmp/mrouted.dump
```

The following is a list of signals can be passed to mrouted along side with -USR1:

- HUP: restarts **mrouted**. The configuration file is reread every time this signal is evoked.
- INT: terminates execution gracefully (i.e., by sending good-bye messages to all neighboring routers).
- TERM: same as INT
- USR1: dumps the internal routing tables to /tmp/mrouted.dump.
- USR2: dumps the internal cache tables to /tmp/mrouted.cache.
- QUIT: dumps the internal routing tables to stderr (only if **mrouted** was invoked with a non-zero debug level).

3.0 Point to Multipoint Multicasting

Note: It is assumed that a point to multipoint network has already been setup successfully. Also make sure you have all the prerequisites listed in the Sec. 1.3.

Replace <pid number> with the number you get from the previous command.

Since many of the links on a point to multipoint network are unidirectional, the dynamic “mrouted” router is not capable of automatically generating the routing table. The static “smcroute” is required to enable multicast on this type of network so users must configure static multicast routes themselves. Once these routes are known they can be put into the “/etc/config/smc-routes” file for later loading.

We show below a classic example of multicasting in the next section to help you learn how to configure your SnIP interfaces on a point to multipoint link.

3.1 Case Study I: Multicasting in a Point to Multipoint Network

This example network is fully bi-directional when viewed from the outside, but internally it uses unidirectional links for return traffic from remotes and a common outbound channel which may be considered unidirectional. This type network is shown in the diagram below. The unidirectional links make use of the mrouted dynamic multicast router not possible.

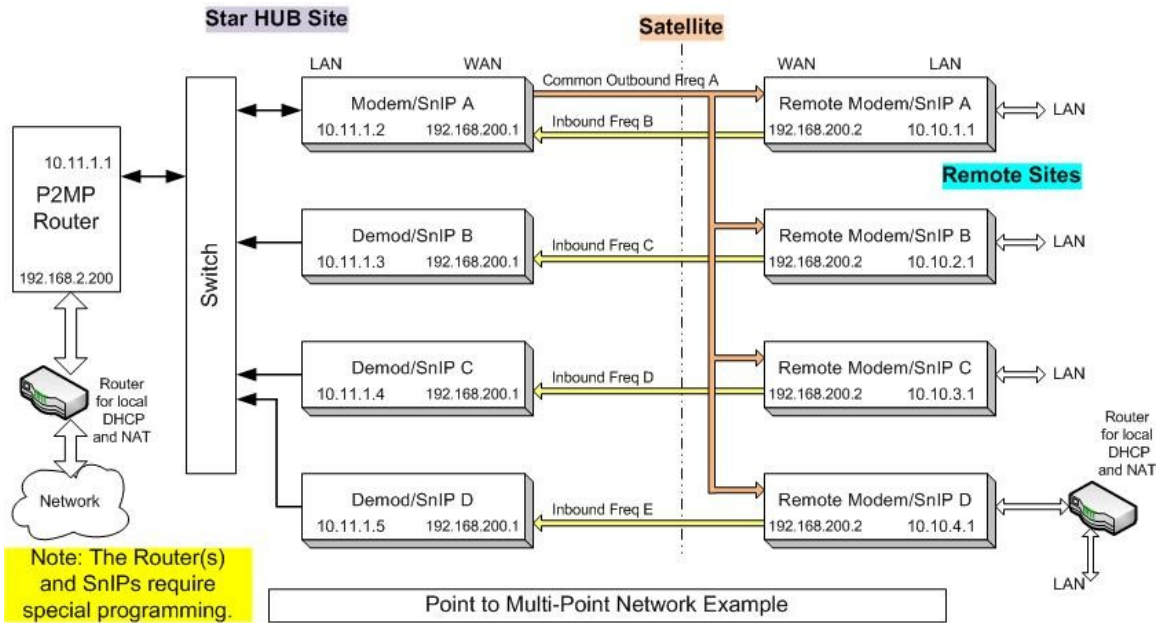


Figure 2: A Point to Multipoint Network

Figure 2 shows a case of point to multipoint network. We will assume there is a remote host with an IP address of 10.10.1.10 (not shown in the picture) on Remote 1 wants to send a multicast packet to a computer on the Near End of the link (the end with the routers) and to a computer that is on Remote 2. The multicast group IP address is 224.1.1.2.

3.1.1 The Two Variables

The two variables you need to know are the source computer's IP address and the multicast group IP address. In our case:

Source IP: 10.10.1.10
 Multicast Group IP: 224.1.1.2

3.1.2 Modems Involved

In this case the modems involved are:

- Remote1: Transmits the packet from the source to the Near End. (Transmit)
- Demod1: Receives the packet that is transmitted from Remote1. (Receive)
- Modulator1: Sends out the packet for the computer on Remote2 to receive. (Transmit)
- Demod2: Receives the packet from Modulator 1. (Receive)

3.1.3 Configuration

Since we have all the information we need, we will start enter those static routes in each of the modems involved.

For **Transmit** Modems, enter the following in its telnet session:

```
smcroute -d <ENTER>
smcroute -a eth0 <Source IP> <Group IP> hdlc0 <ENTER>
```

For **Receive** Modems, enter the following in its telnet session:

```
smcroute -d <ENTER>
smcroute -a hdlc0 <Source IP> <Group IP> eth0 <ENTER>
```

In all modems, the <Source IP> and <Group IP> should be replaces with the values we got from Sec. 3.1.1.

Note: Using the smcroute command manually from the command line is not persistent. It will be lost on reboot of the SnIP. Use configwiz to set the router protocol persistently. The routes themselves are then read automatically from the “/etc/config/smc-routes” file on power up.

3.1.4 Iptables

If you followed our guide on how to configure a point to multipoint network, your SnIP currently should be filtering out all IP packets except for the ones designated to the address within the subnet of the SnIP. You need to allow your SnIP to receive multicast packet too.

Multicast IP address range from 224.0.0.0 to 239.255.255.255. That is 224.0.0.0/4 in submask notation.

For all modems involved, type in:

```
iptables -I FORWARD 1 -s 224.0.0.0/4 -j ACCEPT <ENTER>
iptables -I FORWARD 1 -d 224.0.0.0/4 -j ACCEPT <ENTER>
service iptables save
```

3.2 Case Study II: Multicast on Unidirectional Link

This case study shows the capability of multicasting on our SnIP in a network with no return path.



Figure 3: Unidirectional Multicasting

The source modem in Figure 3 is a Mod only, and two Demod only modems are located at two separate remote sites.

Note: It is assumed that a point to multipoint network has already been setup successfully. Also make sure you have all the prerequisites listed in the Sec. 1.3.

3.2.1 The Two Variables

It is important to know that the source IP address is not the IP address of the SnIP, but is the IP address of the multicast traffic transmitter.

```
Source IP: 10.10.1.2
Multicast Group IP: 224.1.1.2
```

3.2.2 Configuration

For the Mod only modem:

```
smcroute -d <ENTER>
smcroute -a eth0 <Source IP> <Group IP> hdlc0 <ENTER>
```

For the Demod only modem:

```
smcroute -d <ENTER>
smcroute -a hdlc0 <Source IP> <Group IP> eth0 <ENTER>
```

HYZ
End of Document.