Myrinet Installation and Troubleshooting Guide

Myricom, Inc.
Draft: 29 April 2003

The most recent version of this document can be downloaded from
http://www.myri.com/scs/doc/troubleshooting_guide.pdf

Table of Contents

I. Introduction 3
II. What Hardware Do I Need to Install? 3
III. Hardware Installation 3
IV. What Software Do I Need? 7
V. Software Installation 7
1. Configuring and compiling GM. 8
2. Installing the GM driver. 8
3. Run the GM mapper 10
4. Enabling IP over Myrinet (Ethernet emulation) (*OPTIONAL*) 12
5. Testing/Validation 12
a. Check the LEDs on each switch port and interface 13
b. Run gm_board_info on one host 13
c. Run gm_debug to test the PCI bandwidth 14
d. Run gm_allsize to test links in the network 15
e. Run gm_stress to test the network 16
f. Run Mute on the cluster to test for bad links 19
Appendix A: Determining if a Problem is Hardware or Software Related 21
Appendix B: Diagnosing the Cause of a Hardware Problem 22
B.1. How do I determine if a cable is defective? 22
B.2. How do I determine if a port on a switch line card is defective? 23
B.3. How do I determine if an interface is defective? 23
B.4. I still have a problem. What else could I try? 24

I. Introduction

This Myrinet Installation and Troubleshooting Guide describes the hardware and software installation procedures for a Myrinet cluster. Section II summarizes the required hardware, and Section III provides detailed installation instructions for each hardware component. Sections IV and V address the software installation of GM and the testing and validation of the Myrinet cluster.   Appendices A and B provide diagnostics for determining if a problem is hardware- or software-related, as well as procedures for isolating the source of a hardware failure.

II. What Hardware Do I Need to Install?

A Myrinet-2000 network consists of the following hardware components connected to your host computers.

Myrinet-2000 PCI/Host Interfaces

Myrinet-2000 Switch(es)

Myrinet-2000 Fiber cables

The basic requirements are:

Interfaces: one per host
Switches: at least one port per host (more required for clusters larger than 128 hosts).
Cables: one per interface (connecting to a port on a switch), plus any required between switches.

Detailed product specifications are available:

http://www.myri.com/myrinet/product_list.html

III. Hardware Installation

Upon receipt of the Myrinet hardware, read the following two documents:

"Guide to Switches and Switch Networks”
http://www.myri.com/myrinet/m3switch/guide/index.html

"Guide to Myrinet/PCI Host Interfaces”

http://www.myri.com/scs/doc/guide_to_interfaces.pdf

After reading these two guides, you should be aware of the following important information and precautions:

- *Myrinet-2000 PCI/Host interfaces function correctly in hosts with 32-bit or 64-bit PCI slots, with either a 33MHz or 66MHz PCI clock, and with either 3.3V or 5V signaling.*
- *Myrinet-2000 PCI/Host interfaces function correctly in PCI-X slots.*
- *If your Myrinet-2000 switch is equipped with a monitoring line card (located in the top-slot of the switch), this monitoring line card contains 10base-T dual ethernet ports and DHCP is required for its installation.*
- *A Myrinet-2000 switch does not require any configuration.*
- *Switch line cards (M3-SW16-8F, M3-SPINE-8F, M3-BLANK) are hot-swappable.*
- *A line card, a fan try, or an enclosure is a Field Replaceable Unit (FRU). The power supply is not an FRU; it is built into the enclosure.*
- *We recommend the use of dust plugs in unused ports on the switch, and require blank panels in unused slots (for cooling and EMI reasons).*
- *You must provide proper ventilation for the switch(es), otherwise shutdown due to overheating could occur.*
- *Fiber-cable ends and ports on line cards must be kept free of dust particles. Accumulation of dust can cause faults from the port-to-fiber connection.*
- *Myrinet-2000 fiber cables are 50/125 multimode fiber pairs with LC connectors. Myricom does not guarantee correct operation with other than 50/125 fiber cables.*
- *Myrinet-2000 cables are hot-pluggable.*
- *Cables should be disconnected by carefully depressing the connector tab, otherwise damage can result.*
- *Avoid crimping cables (bending at tight angles) as damage can result. The minimum bend radius for fiber cables is a "finger width" (or 1/4" radius).*
- *You should provide support restraints for cabling of large cluster configurations. E.g.,*
  - *http://www.phys.lsu.edu/faculty/tohline/capital/beowulf.html*
  - *http://helics.iwr.uni-heidelberg.de/gallery/index.html*

Installation of the Myrinet PCI/Host interfaces

Following the installation instructions in the "Guide to Myrinet PCI/Host Interfaces" document, you will perform the following steps:

1. Install the Myrinet interface(s) into your host(s).
2. Power on the host(s).
3. Detect the interface(s) in your host(s) using the Linux command **/sbin/lspci**. (If you are using an operating system other than Linux, appropriate detection commands are listed in the aforementioned document.)

We assume that the operating system has already been installed on the host(s), and riser cards are installed if the host is a rack-mountable unit.

> **Caution:** *Riser cards can be a significant source of problems in a hardware configuration. Although PCI riser cards are commonly used, they will generally violate PCI specifications for the length of signal traces. A riser card may also introduce impedance discontinuities and signal degradation between the mother board, riser card, and interface card. If you observe PCI-communication errors when using a riser card, refer to the diagnostic procedure below. A higher quality riser card – one whose traces match the impedance of signals on the mother board and interface – may also solve the problem.*
>
> *It is important that a 64-bit riser card is used in a 64-bit PCI slot, and likewise a 32-bit riser card is used in a 32-bit PCI slot. If you are using a riser card with multiple slots, the Myrinet interface should be placed in the slot closest to the PCI connector on the mother board to minimize the distance between the PCI connector on the mother board and the PCI connector on the riser card. Otherwise, the Myrinet PCI interface may not be correctly detected and/or serious performance irregularities will result.*

If a Myrinet PCI/Host interface is not detected using **/sbin/lspci**, then

- Are you using a riser card?
- Is the interface properly seated in the PCI slot or riser card?
- Have you tried inserting the interface into a different slot on the riser card?
- Have you tried inserting the interface directly into the PCI slot?
- Have you tried using a different PCI slot?
- Have you tried using a different riser card and/or a different brand of riser card?
- Have you tried using a newer BIOS for this mother board?

Installation of the Myrinet switch and cables

Once the Myrinet PCI/Host interface(s) have been installed and correctly detected in your host(s), you can now proceed to installing the switch(es) and connecting the cables.

The installation of the switch involves the following steps:

1. Plug in the power cord of the switch and make sure that all of the switch line cards are properly seated. If your switch contains a monitoring line card, do not yet seat the monitoring line card or connect ethernet to the monitoring line card. Installation of the monitoring line card will be performed after all of the fiber cables have been connected.
2. A cable should then be connected between the fiber port on each interface and a port on a switch line card. (*Note: If more than one Myrinet switch is being used in your configuration, refer to the provided network diagram for cabling details.)*
3. You should next install the monitoring line card (located in the top-slot) in your switch. The presence of a monitoring line card in the switch is optional. If your switch does not contain a monitoring line card, you can skip this step of the hardware installation. To install a monitoring line card in your Myrinet-2000 switch, do the following:

    Step 1. Read the MAC address from the faceplate of the monitoring line card, and register this MAC address with a static IP address in the reservation table of the DHCP server on the local network. The DHCP server will then serve this static IP address to the monitoring line card when it boots and asks for it. On Linux, this file is **/etc/dhcpd.conf**.

    Note: The MAC address is a group of 6 hexadecimal numbers separated by colons, and should begin with 00:60:dd:7f:??:??.

    Step 2. Before sliding the monitoring line card into the top slot of the Myrinet-2000 switch, connect at least the first ethernet port to the LAN. For high availability, the second ethernet port can also be connected.

    Note: The monitoring line card can ONLY be installed in the top slot of your Myrinet-2000 switch.

    Note: The ethernet ports on the monitoring line card are 10base-T.

    Step 3. When the monitoring line card is locked in position, a green LED for the line card and the LEDs for the connected ethernet port(s) will illuminate. The monitoring line card will immediately start to broadcast DHCP requests. When the monitoring line card has received its IP address, it is reachable. You can *ping* the card or open a web browser to it.

Note: To determine the IP address that was assigned to the monitoring line card, look at the file **/var/state/dhcp/dhcpd.leases** on your DHCP server.

Each time a monitoring line card is powered on, it will ask for its IP address (and netmask) via DHCP. You can specify a gateway with the DHCP "routers" option. The lease time is 1 day.

To test that your monitoring line card is properly installed, you can *ping* its IP address or open a web browser to its IP address. We suggest that you familiarize yourself with the features of the HTTP interface to the monitoring line card, as many of these features can be very useful diagnostic tools. A description of these features can be found in the "Switch Information" section of the Myrinet FAQ (http://www.myri.com/scs/FAQ/).

If you have difficulties installing the monitoring line card, refer to the Myrinet FAQ entry "*How do I install the monitoring line card in my Myrinet-2000 switch?*" and feel free to contact help@myri.com for assistance.

IV. What Software Do I Need?

Myricom supplies and supports Myrinet software (low-level firmware and a choice of middleware) for a variety of operating systems and processors. At present, the following middlewares are available: MPICH-GM, VI-GM, Sockets-GM, and PVM-GM.

Performance graphs for Myricom-supported software are available:
http://www.myri.com/myrinet/performance/

The first Myrinet software package you must install is GM, the low-level message-passing system for Myrinet. GM includes a driver, Myrinet-interface control program, a network mapping program, and the GM API, library, and header files.

For the purposes of this document, we shall only discuss a software installation on the Linux operating system. Similar installation instructions exist for all of the other supported operating systems and can be found on their respective OS-specific download page (accessible via http://www.myri.com/scs/).

For links to software download and abbreviated installation instructions on Linux, please refer to:

http://www.myri.com/scs/linux.html

V. Software Installation

GM installation is performed in 5 easy steps:

1. Configuring and compiling GM.

2. Installing the GM driver.
3. Running the GM mapper.
4. Enabling IP over Myrinet (ethernet emulation) (OPTIONAL)
5. Testing/Validation

1. Configuring and compiling GM.

Download GM
`ftp://comrade@ftp.myri.com/pub/GM/gm-1.6.4_Linux_and_AIX.tar.gz`

```
gunzip -c gm-1.6.4_Linux_and_AIX.tar.gz | tar xvf -
cd gm-1.6.4_Linux_and_AIX
./configure --with-linux=<linux-source-dir>
```

where <*linux-source-dir*> specifies the directory for the Linux kernel source.

```
make
```

2. Installing the GM driver.

Select an installation directory path <*install_path*>. It is usually best for <*install_path*> to be the path to an NFS directory available on all machines that are to share this GM installation. The directory must be accessible using <*install_path*> on all machines that are to share the installation. <*install_path*> must be an absolute path; it must start with /. However, <*install_path*> may contain symbolic links.

***Note**: The <install_path> installation directory must be created prior to invoking the GM_INSTALL script.*

```
cd binary
./GM_INSTALL <install_path>
```

If you omit the <*install_path*>*,* the driver will be installed in the default directory, **/opt/gm/.**

Next, you must run

```
su root
<install_path>/sbin/gm_install_drivers
/etc/init.d/gm start
```

on each machine to install/copy the driver on that machine.

The **gm_install_drivers** script performs the following operations:

- Shuts down existing IP over Myrinet
- Unloads existing GM module (if it exists)
- Creates the devices (/dev/gm* and /dev/gmp*), one device per interface

- Loads the GM module (insmod)

Important note: The **gm_install_drivers** script does not configure the IP device. If you wish to run IP over GM/Myrinet (ethernet emulation), you must configure the device. (Refer to Step 4 of the installation process.)

If you wish the driver to auto-load at boot, you must create appropriate links in the **/etc/rcN** directories to the **/etc/init.d/gm** and /etc/init.d/myri scripts. Alternatively, you may start and stop the driver manually using

```
    su root
    /etc/init.d/gm start
    /etc/init.d/gm stop
```

or

```
    su root
    /etc/init.d/gm restart
```

Green “link” LEDs should illuminate on interfaces and ports on the line cards when the hardware is connected through a cable to an operating component and the GM firmware has been loaded. If you do not see a green “link” LED illuminated on a component (port on an interface or port on a switch line card), refer to the following diagnostics:

- If you do not see any green “link” LEDs illuminated, is the switch powered on?
- If you do not see green “link” LEDs illuminated on only a specific line card, is the line card properly seated in the enclosure? (Refer to the “Guide to Switches and Switch Networks” for the proper procedure to insert/remove a line card.)
- If you do not see a green “link” LED on a specific port on an interface or port on a line card, is the port connected by a cable to another component?
- If an interface port does not have a green link LED illuminated, is its host powered on?
- Have you tried disconnecting and reconnecting the cable at both ends (at the interface port and the port on the line card?
- Have you tried a different cable or a different port on the line card?

If you have tried all of these procedures and you cannot resolve the problem, contact help@myri.com for assistance. You cannot continue with the software installation until this issue is resolved.

Once the **gm_install_drivers** script has been run on each host, the yellow/amber “LANai” LED should illuminate on the faceplate of the Myrinet PCI/Host interface. The yellow "LANai" LED is controlled by the LANai processor, and will pulse like a heartbeat while the GM MCP/firmware is running, and will pulse faster when there is more packet-sending activity (including sending acknowledge packets in reply to packets received.) If the yellow LED is not pulsing, the GM MCP is not loaded or is not running.

Note:

1. GM is not in the critical performance path so it does not need to be built with specialized compilers and flags. GM should be built with Gnu **gcc** and only built with -O level of optimization.
2. GM should be installed in an NFS-mounted area.
3. **gm_install_drivers** needs to be run on all nodes in the cluster!
4. The kernel header files MUST match the running kernel exactly: not only should they both be from the same version, but they should also contain the same kernel configuration options. (Be careful with RedHat kernel packages.)
5. By default, we also assume that you have PCI64, PCI64A, PCI64B, or PCI64C interfaces. (PCI32 interfaces are not supported in gm-1.6.3 and later.)
6. If a host is rebooted, you must reload the GM driver (and rerun the GM mapper).

The most common **gm_install_drivers** failures are:

- APIC IRQ conflicts (encountered on several Tyan and AMD mother boards)
- Running kernel / source kernel mismatch (commonly encountered with RedHat kernel packages)
- AGP (nVidia, ATI) conflicts
- Defective or inadequate riser cards

The solutions for these problems are summarized in the FAQ entry *"GM_INSTALL or gm_install_drivers fails. What does this error message mean?*".

Undoubtedly, if you encounter an issue on a specific mother board or version of Linux, someone else has too, and it will be documented on the Myricom web site. If not, contact

us at help@myri.com.

3. Run the GM mapper

```
cd <install_path>/sbin/
su root
./mapper ../etc/gm/map_once.args
```

Important points to note:

- The GM mapper is ONLY run on one node in the cluster. You should choose one node in the cluster to be the *mapper node*, and any subsequent invocations of the mapper should be done on this node only.
- The GM mapper must be run before any communication over Myrinet can occur.
- If a host is rebooted, you must reload the GM driver and rerun the GM mapper.

- If any topological change occurs in the cluster, the GM mapper must be rerun.
- Never run the GM mapper on multiple nodes at the same time, as serious routing confusion will result.

The aforementioned mapping procedure uses the most common form of mapping: "Map Once" Mapping. Depending upon your needs, there are three ways to run the GM mapper:

- Map Once Mapping
- Static or "File" Mapping
- High Availability (HA) Mapping

"Map Once" Mapping is by far the most common way of running the GM mapper. In this method, the mapper is run on one host in the network (any of the hosts). It is rerun if a host (re)boots or a hostname is changed or after a change of Myrinet topology (swapping of ports on a switch). The command for this method of running the GM mapper is:

```
cd <install_path>/sbin/
su root
./mapper ../etc/gm/map_once.args
```

"Static" Mapping is another way in which the GM mapper may be used. In this method, an active mapper is run once when ALL of the hosts are up and running the GM driver.

- This initial active mapper will generate a map file and a host file.
- These files are then shared by NFS, or copied to all of the hosts in the network.
- An entry in the boot scripts will allow each host to read the map file and the host file and update the routing table on its local Myrinet interface(s).

The command for this method of running the GM mapper is:

```
cd <install_path>/sbin/
su root
./mapper ../etc/gm/static.args
```

If the GM tree is not mounted by NFS, copy the 3 files created by this command (**static.map**, **static.routes**, and **static.hosts**) to each <install_path>/sbin/ directory on each host.

For auto-mapping at boot time, add the following command to the boot scripts of the host (scripts in /etc/init.d or /etc/rc.d/init.d).

```
cd <install_path>/sbin/
su root
./file_mapper ../etc/gm/file.args
```

"High Availability" Mapping is the third way in which the GM mapper may be used. This method is for users who have a need for High Availability (HA) in an aggressive computing environment. The command for this method of running the GM mapper is:

```
cd <install_path>/sbin/
su root
./mapper ../etc/gm/active.args &
```

"High Availability" Mapping will continuously run the GM mapper in the background to detect and add any new hosts or remove any non-responding hosts, to detect any change of topology (change of slots in the switch, change of innerswitch topology), and to periodically update the routing tables of the Myrinet cards (by default, every 30 seconds).

You should note that this HA mapping method is slightly intrusive. Since the GM mapper uses unreliable messages that may be dropped in case of heavy contention, this method of mapping can lead to hosts involved in a long computation being marked as "non-responding" and removed from the routing tables because they are unreachable.

For the majority of users, the "map_once" GM mapping method is sufficient. For the users with more production-level constraints, the "static mapping" is the recommended method. For fault-tolerant GM applications, the third method provides the best alternative.

4. Enabling IP over Myrinet (Ethernet emulation) (*OPTIONAL*)

If you wish to run IP over Myrinet (ethernet emulation), the Linux command to enable IP over GM is as follows:

```
/sbin/ifconfig myri0 <ip_address> up
```

where you must replace *myri0* with the appropriate name (*myri1*, *myri2*, etc.) if you have more than one Myrinet interface per host.

To obtain good IP performance over Myrinet:

- use Linux 2.4.x (Linux 2.4.20 is now available)
- configure GM with --enable-new-features to get a larger 9000byte MTU for IP-over-Myrinet.

5. Testing/Validation

Once the GM software is properly installed on all hosts/nodes in your cluster, you are ready to "validate" your Myrinet installation by performing the following sequence of tests.

- Check the LEDs on each switch port and interface
- Run gm_board_info on one host
- Run gm_debug to test the PCI bandwidth
- Run gm_allsize to test the links in the network
- Run gm_stress to test the network
- Run Mute on the cluster to check for bad links

These steps are detailed below and are also described in the “Troubleshooting” section of the FAQ (http://www.myri.com/scs/FAQ/). Once you have performed these tests, you will have a solid Myrinet installation.

a. Check the LEDs on each switch port and interface

After the hardware installation and GM software installation have been completed, there will be a green LED and flashing yellow/amber LED illuminated on each interface, and a green LED illuminated on each connected switch port.

If any of these LEDs are not illuminated, then refer to Section V (page 9).

b. Run gm_board_info on one host

If all of the LEDs are illuminated, you can now test that the GM mapper has correctly detected all of the hosts in your Myrinet network by typing the following command on one host:

```
cd <install_path>/bin/
./gm_board_info
```

Assuming that the gm module has been loaded and the GM mapper has been run, **gm_board_info** displays information about each Myrinet interface (board) installed on this host, as well as a list (routing table) of the Myrinet nodes with which it is able to communicate.

- If you see ***** No routes found ***** at the bottom of the display for each board, this is an indication that the GM mapper has not been run on the Myrinet network since this host was powered on or rebooted. You must run the GM mapper (as previously described on pages 10—11) before you can proceed.

- If the list of nodes (routing table) is not complete (that is, it does not contain all of the nodes in the cluster) you will need to determine why some nodes are not listed. Refer to the following guidelines.

    1. Was GM properly loaded on all hosts in the Myrinet network?
        - If so, you should see the gm module loaded (**/sbin/lsmod**) and the devices (**/dev/gm* and /dev/gmp***) created. Proceed to (2.) below.

- If not, run **gm_install_drivers** on the missing hosts, and then rerun the GM mapper. If **gm_install_drivers** fails, refer to the FAQ entry "*GM_INSTALL or gm_install_drivers fails. What does this error message mean*?".

2. If the gm module is loaded and the devices are created but you still do not see a specific host in the routing table, check the basic connectivity on the missing host by running the following **loopback-mode gm_allsize test** on this particular host:

   ```
   gm_simpleroute --loopback [--board=n]
   gm_allsize --geometric -exit-on-error [--board=n]
   ```

   If **gm_allsize** fails, then the host is disconnected; check the cable from the interface to the switch to make sure that it is well-connected and that a green "link" LED is illuminated on the interface and on the port on the switch to which the cable is connected. You should also check the kernel log messages (with dmesg) on this host for any GM errors. If there are abnormal messages in the kernel log, then check the FAQ or contact help@myri.com. If there are not any abnormal messages in the kernel log, the problem may be hardware-related. You should try a different cable and then try a different port on a switch line card to see if this interface can see the Myrinet network.

   Note: After running tests with **gm_simpleroute** the mapper must be run again so that this host will have the correct routes to other hosts.

Note that **gm_simpleroute** is an alternative to the GM mapper that you may prefer for early testing. (Loopback and point-to-point (back-to-back) network topologies require that **gm_simpleroute** must be run instead of the GM mapper.) **gm_simpleroute** may be used to set up a one-node "network", thus enabling sanity tests like **gm_board_info** and loopback-mode **gm_allsize**.

c. Run gm_debug to test the PCI bandwidth

We recommend the following test to verify your GM performance.

```
cd <install_path>/bin
./gm_debug –L
```

This **gm_debug** test displays the results of the hardware benchmark test of the PCI bus with the DMA engine of the Myrinet interface. The output of this command indicates the maximum sustained bandwidth that can be obtained from the PCI bus, and thus provides an upper bound on GM performance. A detailed description of this benchmark can be found in the FAQ entry "*Can you describe in detail the "hardware benchmark of the PCI bus" that is returned by gm_debug*?".

The output of this command also tells you if the Myrinet interface was correctly detected as 64-bit / 66 MHz, for example. If the interface was not correctly detected by the BIOS, you should suspect a riser card problem or a PCI slot problem.

Performance graphs (http://www.myri.com/myrinet/performance/) for GM are available. The performance measurements were obtained by running **gm_allsize** tests for latency and bandwidth as described on the FAQ entry "*What are the run-time options to gm_allsize?*".

Refer to the section entitled "GM Performance" in the <GM_source_path>/README for complete details on expected GM performance.

d. Run gm_allsize to test links in the network

If **gm_board_info** shows all of the hosts in the cluster, you are ready to proceed to running **gm_allsize**. As previously mentioned, the test program **gm_allsize** can be used as a basic connectivity test (loopback-mode); however, **gm_allsize** can also be used to measure latency and bandwidth. One-way latency and bandwidth tests between two hosts in the Myrinet network can be performed with **gm_allsize** by running in slave mode on one node and as master on the GM node to be tested, specifying the target host as the slave host. This is done by adding the **gm_allsize** run-time option **--slave** on the command line of the slave machine and the **gm_allsize** run-time option **--remote-host=<host1>** on the command line of the master node, where **<host1>** is the name of the machine running in slave mode.

(The name of each Myrinet host can be found by running **gm_board_info** on one of the hosts.)

Adding the **--verify** flag to any **gm_allsize** command will augment the test with verification of the contents of all messages, at the cost of significantly degraded performance. For a list of all options to **gm_allsize**, type `gm_allsize -help` or refer to the FAQ. For sample output of **gm_allsize**, refer to the FAQ entry "*What are the run-time options to gm_allsize?*".

- **Latency**

  To test the **latency** between two hosts (***host1*** and ***host2***), type the following on ***host1*:**

  ```
  gm_allsize -slave
  ```

  and on ***host2*** type:

  ```
  gm_allsize --remote-host=host1 --geometric
  ```

The output from this command will consist of two columns of data: the first column lists the message size (in bytes), and the second column lists the latency (in microseconds).

- **Unidirectional Bandwidth**

To test the **unidirectional bandwidth** between two hosts (***host1*** and ***host2***), type the following on ***host1*:**

```
gm_allsize --slave --size=15
```

and on ***host2*** type:

```
gm_allsize --unidirectional --bandwidth \
           --remote-host=host1 --size=15 --geometric
```

Unidirectional bandwidth is a PingPing test, measuring the startup and throughput of a single message sent between two processes, where messages are obstructed by oncoming messages.

The output from this command will consist of two columns of data: the first column lists the message size (in bytes) and the second column lists the bandwidth (in MB/s).

- **Bidirectional (Summed) Bandwidth**

To test the **bidirectional (summed) bandwidth** between two hosts (***host1*** and ***host2***), type the following on ***host1***:

```
gm_allsize --slave --size=15
```

and on ***host2*** type:

```
gm_allsize --both-ways --bandwidth \
            --remote-host=host1 --size=15 –geometric
```

This test has GM streaming packets in both directions (both nodes are always sending) and it causes GM to report the sum of the send and receive bandwidths.

The output from this command will consist of two columns of data: the first column lists the message size (in bytes) and the second column lists the bandwidth (in MB/s).

e. Run gm_stress to test the network

gm_stress.c is an all-to-all test program for GM. It is available in GM 1.5.1 and higher in the bin/ subdirectory. The most up-to-date version, which may be newer than in any GM release, is located on the Myrinet FAQ entry "*How do I run gm_stress.c to validate my GM installation?*".

Compile the program like you would any GM program. For example:

```
     gcc -g -o gm_stress gm_stress.c \
     -L <install_path>/lib/ -lgm -I <install_path>/include
```

Run **gm_stress** on every host in a cluster to validate your installation of GM. If you decide to use run-time options, they must be the same on all hosts. You can use **rsh** to launch **gm_stress** on remote nodes. When **gm_stress** terminates on the local node, it has terminated on all the other nodes too, even if **rsh** does not clean up nicely on them. If the test failed on any node, then **gm_stress** on the local node will report an error. You can also send datagrams with this test, but they do not count towards terminating the program. Thus, if you send 100% datagrams, **gm_stress** will run forever. By default **gm_stress** runs all-to-all on the entire cluster. Use the **-f** run-time option to test on a subset.

The options to **gm_stress** can be obtained by invoking the following:

```
xxx@xxx:~/gm/binary/bin$ ./gm_stress -h
options:
-u, --unit=NUM            [0],      myrinet board number.
-p, --port=NUM            [2],      GM port number.
-n, --num-sends=NUM       [1024],   sends per host.
-m, --max-size=NUM        [19],     max send "size" (in log2 units).
-f, --file=FILENAME       [none],   file of hostnames, one per line
-c, --checksum            [off],    compute checksums.
-r, --min-receives=NUM    [4],      minimum number of receive buffers
per size.
-b, --barrier-seconds=NUM [300],    timeout for barrier.
-x, --max-resends=NUM     [30],     max resends per host destination
for send timeouts.
-d, --datagram=NUM        [0],      percent datagrams.
```

The format of the file specified by FILENAME is a list of hostnames (one per line) as reported by the routes in **gm_board_info**.

Example Usage on a subset of nodes in a cluster

You can use an **rsh** script like the following to launch **gm_stress** on the cluster.

```
xxx% /home/xxx/gm/tests# cat rsh_stress
#!/bin/bash

for i in `cat $1`
do
ssh $i $2 $3 $4 $5 $6 $7 $8 $9 ${10} ${11} ${12} ${13} ${14} ${15}
${16} ${17} &
done
```

```
xxx% /home/xxx/gm/tests# cat hosts
racksaver1
racksaver2
racksaver3
racksaver4
racksaver5
racksaver6
racksaver7

racksaver1% /home/xxx/gm/tests# ./rsh_stress hosts \
        /home/xxx/gm/tests/gm_stress -f /home/xxx/gm/tests/hosts
racksaver1% /home/xxx/gm/tests#
7 hosts.
7 hosts.
7 hosts.
7 hosts.
7 hosts.
7 hosts.
7 hosts.

sent / received 1024 messages per host
in lengths up to 524240 bytes.
at a rate of 93 / 92 MB/s
sent / received 1024 messages per host
in lengths up to 524240 bytes.
at a rate of 93 / 90 MB/s
sent / received 1024 messages per host
in lengths up to 524240 bytes.
at a rate of 93 / 93 MB/s
sent / received 1024 messages per host
in lengths up to 524240 bytes.
at a rate of 88 / 89 MB/s
sent / received 1024 messages per host
in lengths up to 524240 bytes.
at a rate of 88 / 91 MB/s
sent / received 1024 messages per host
in lengths up to 524240 bytes.
at a rate of 88 / 86 MB/s
sent / received 1024 messages per host
in lengths up to 524240 bytes.
at a rate of 93 / 94 MB/s
```

How does gm_stress work?

Every host sends a given number of random sized packets out of a big chunk of **gm_dma_malloc** memory to each host randomly and terminates when there is an error or when it has received the given number of packets from each host, (including itself). It uses **gm_zone.c**, which is part of **libgm**, for **malloc** and **free** from the send heap. If any host terminates without error then it means the test has succeeded globally. Otherwise the host that detected the error will terminate first, and then the others after sends fail. The test prints out a host count before the test starts. If it doesn't print anything out, then the test is hanging on the initial barrier, which indicates a user error: for example, the command was not started on all the hosts or GM is not installed properly on a host.

The **gm_stress** test sends messages to itself and will work in loopback but not in a point-to-point topology, where there is no route-to-self.

The datagrams option adds some percentage of uncounted datagrams to the stream. 100 percent is legal but will never terminate.

What happens when gm_stress fails?

The **gm_stress** program is meant to *stress* all of the connections in the Myrinet network. There are three ways in which **gm_stress** can fail.

1. **Connectivity problem.** Eventually a send will fail, usually with the "destination unreachable" error. On the host where the send failed, **gm_stress** will terminate. Subsequent sends from the other hosts to this first host will fail. On these hosts, **gm_stress** will terminate too, with "remote port closed" errors. Eventually, **gm_stress** will terminate globally in this manner. It will not happen immediately. If you are impatient, you can kill the lingering gm_stress tests. If you do this, however, you may also need to wait for the GM long timeout (around 30 seconds) before those ports will be usable again because GM may still be retransmitting some packets. The error messages will be a combination of send/recv errors.
2. **User error.** For instance, inconsistent host files as inputs, not running **gm_stress** on all hosts specified in the host file, or using mismatched size parameters. These may result in assertion failures on one host, followed by the kind of distributed shutdown described above. Or the test may seem to hang. You may have to kill the gm_stress tests by hand and wait for the 30-second GM long timeout for ports to close.
3. **Congestion.** If gm_stress fails with the "exceeded --max-resends" error on any node, this is just a problem with the test program, and doesn't indicate anything wrong with the cluster. Due to congestion at the receiver, send timeouts can occur. You will need to tune the parameters a bit. Try incrementally increasing --min-receives and decreasing --max-size, starting from their defaults (4 and 19). Or try increasing --max-resends to something large, like 300.

f. Run Mute on the cluster to test for bad links

Mute is a graphical diagnostic monitoring tool for Myrinet-2000 networks (switch(es), cables, and hosts). It exercises the monitoring capabilities of the Myrinet-2000 switches (M3 Switch Tools, m3-dist.tar.gz), as well as the Mapper Tools (located in the /mt subdirectory of the GM distribution). Mute builds an image of the Myrinet-2000 network, and can be used to non-intrusively monitor the Myrinet-2000 network in real time, analyzing the network traffic, and diagnosing/validating the integrity of the hardware components. Its graphical interface uses GNOME/GTK, which is currently only available for Linux. Refer to the **Mute** webpage (http://www.myri.com/scs/mute/) for an overview of **Mute**, installation instructions, and usage scenarios.

Appendix A: Determining if a Problem is Hardware or Software Related

Diagnosing a problem as hardware- or software-related can be difficult. The first goal is to isolate where the problem resides:

- Host computer hardware (e.g., a bad PCI slot, defective or inadequate riser card, buggy BIOS, etc)
- Myrinet hardware (interface, switch, or cable)
- Host computer software (e.g., OS not configured properly)
- Myrinet software (GM driver, GM mapper, MPICH-GM, etc)

If there are host computer hardware problems, these problems will most likely be encountered during the Myrinet hardware installation phase. Refer to Section III of this document for detailed hardware installation diagnostics.

Assuming you have a 64-bit / 66 MHz PCI bus, use **gm_debug –L** to see if the Myrinet PCI/Host interface was correctly detected as 64-bit / 66 MHz. If not, suspect a riser card problem or a PCI slot problem, as the card is not being corrected detected by the BIOS.

If you suspect a Myrinet hardware problem, have you installed **Mute** on your cluster?

Have you witnessed a high **badcrc_cnt** in the output of **gm_counters**? If the answer to this question is yes, then it is likely that a damaged hardware component (interface, cable, port on a switch line card) exists. **Mute** is the easiest way to verify and isolate a Myrinet hardware abnormality. Instructions for using **Mute** can be found at http://www.myri.com/scs/mute/.

If you are not able to install **Mute** on your cluster, then you need to follow the diagnostic procedures described in **Appendix B** to isolate the malfunctioning hardware component.

If you suspect a Myrinet software problem, please check the Myrinet Software and Customer Support webpage (http://www.myri.com/scs/) to see if there is a newer release, or the Myrinet FAQ (http://www.myri.com/scs/FAQ/) for any reports of known problems.

Appendix B: Diagnosing the Cause of a Hardware Problem

A high **badcrc_cnt** count (reported in **gm_counters**) usually signals a hardware problem. This hardware problem could be in an interface, a cable, a port on a switch, or within a switch. If the defect is in a combination of (interface, cable from the interface to the switch, or port on a switch) it is possible to diagnose this situation quite easily using a **gm_allsize loopback test** as described below. However, if the defect lies within a switch, or on a cable connecting two switches, the following procedure will not detect this kind of failure. A diagnostic tool called **Mute** is needed to detect this type of failure.

Our guarantee is an *average* of less than 1 packet-data error (**badcrc**) per hour on a link operating at full data rate.

Badcrcs will be generated if the switch line card or the interfaces are set to different speeds. Some products have a mechanical switch on the circuit board to allow the default data rate to be switched between SAN-2000 (2.0+2.0 Gb/s) and SAN-1280 (1.28+1.28 Gb/s). Please refer to the Myrinet FAQ entry "*I have Myrinet-2000 interfaces and Myrinet-1280 switches and my cards and switches aren't able to talk to each other. What do I do?*" for more details on checking and setting the speed.

You can isolate which combination of interface, cable, or port on a switch, is causing the **badcrc_cnt** using the following test. This test limits all communication to a specific interface, cable, and port on a switch. Do the following:

4. Reload the GM module on all nodes (resets the counters to zero)
5. Rerun the GM mapper
6. On each node, run:

```
 gm_counters
gm_allsize --min-size=10 --max-size=20 --count-per-length=10
gm_counters
```

If the **badcrc_cnt** (reported in **gm_counters**) increased significantly after the test, then you have identified a possible hardware trouble spot in your cluster and you must now isolate if the **badcrc_cnt** is coming from the interface, the cable, or the port on the switch. In most cases, the **badcrc_cnt** is coming from a damaged cable. If you have some extra cables, we suggest that you first try replacing the suspect cable, and then rerunning the above test to see if the **badcrc_cnt** has disappeared. If this does not eliminate the **badcrc_cnt** then it is possible that the defect is in an interface or a port on a switch.

And now we need to isolate if the hardware failure is caused by an interface, a cable, or a port on a switch line card.

B.1. How do I determine if a cable is defective?

To determine if a cable is defective, try replacing the cable with a known good cable and rerun the **gm_allsize loopback test**. If the **badcrc_cnt** disappears, then you have isolated the damaged hardware component.

If a cable is believed to be defective, contact help@myri.com for instructions on how to return it for a replacement. You will be assigned a "Return Material Authorization" (RMA) number.

B.2. How do I determine if a port on a switch line card is defective?

To determine if a port on a switch is defective, do the following:

With a known good cable, try connecting the interface card to a different port on the switch. Re-run the GM mapper since the topology has changed, and run the **gm_allsize test** again. If the **badcrc** count no longer increases, then the old switch port is bad. The diagnostic tool called **Mute** can be used to verify that this port is bad, and the switch monitoring tools can be used to check for bad events on each port of the line card. Please note that if a cable is moved from one switch port to another switch port (or from one interface to another interface), the mapper must be re-run before any interface-to-interface communication can occur. Each GM process has a relative address to each other process (something like “go to the first switch, jump 3 ports, go to the next switch, jump -2 ports”), and if the cabling of the network has changed, then the mapper must be run again so that these relative addresses can be updated.

If a port on a switch line card is believed defective, contact help@myri.com for instructions on how to return it for a replacement. You will be assigned a "Return Material Authorization" (RMA) number.

B.3. How do I determine if an interface is defective?

If exchanging the cable and the port on the switch line card do not eliminate the errors, then the interface may be the problem. Here are some suggestions for determining whether an interface is defective.

First, try using the interface in isolation. Disconnect the standard Myrinet cable from the interface and attach a **loopback cable**, and then use **gm_simpleroute** and **gm_allsize** as follows:

```
gm_counters [--board=n]
gm_simpleroute --loopback [--board=n]
gm_allsize --geometric --exit-on-error [--board=n]
gm_counters [--board=n]
```

The **--board** flag is only necessary if the board number is other than 0.

If **gm_allsize** completed successfully, and the value for **badcrc_cnt** reported by **gm_counters** did not increase significantly, then the interface is OK. The problem may reside in the cable or the switch.

Note that after running **gm_simpleroute**, the GM mapper will have to be run again to restore the routes to other nodes in the system.

If the foregoing procedure is not feasible, you can try installing the problem interface in another slot or in another host. If you have more than one kind of host (e.g., both Solaris and Linux) running GM, an excellent test would be to try the interface on the alternate host.

If the questionable interface fails in a slot which is successful with another Myrinet interface- especially another interface of the same class - then this interface is probably defective. On the other hand, if GM installs and performs cleanly, then the interface is OK, and there is a platform-dependent hardware or software problem.

If an interface is believed to be defective, contact help@myri.com for instructions on how to return it for a replacement. You will be assigned a "Return Material Authorization" (RMA) number.

B.4. I still have a problem. What else could I try?

Here are some suggestions to use when the interface appears to be good, but some tests are failing anyway.

On Unix-like systems, check the **/dev/gm*** device special files with a command like **ls -l /dev/gm*.** There should be an entry corresponding to each board on this host. The permissions should be read-write (**rw**) to the world. Also check the **/dev/gmp*** device special files. The permissions should be read-write by root only. On some operating systems, the **/dev/gm*** file names are soft links to other file names; in this case, it's the files that are linked to which must have read-write permissions to the world.

Next, run **gm_board_info**:

```
cd <install_path>/bin
./gm_board_info
```

- If **gm_board_info** reports ***** No routes found ***** for some interface, but you believe the Myrinet network has been mapped, try running the GM mapper *on the problem node*, with **-verbose** turned on in the **map_once.args** file.
- If **gm_board_info** *still* reports ***** No routes found *****, try using **gm_simpleroute** to set a route for this specific board.

This procedure should always eliminate the ***** No routes found ***** message.

Now run the command

```
cd <install_path>/bin
gm_allsize --verify -geometric
```

- If this test completes successfully, it appears that there is a problem mapping the network using the original mapper. This can happen if part of the network is disconnected or not powered, or if there is any problem with the mapper software, such as invoking it incorrectly.
- If **gm_allsize** did *not* complete, then run **gm_counters** for the offending interface. Inspect the **bad*xxx*** fields. In particular, look at the **badcrc_cnt** entry, which tallies CRC errors. This number should be very small as compared to **netrecv_cnt**. If this number is large, you may have a damaged cable or port on a switch.

You can try swapping the cable or port on a switch line card with known good components to see whether the problem follows the cable or the port on the switch line card.

If the foregoing tests do not solve your problem, then collect all the relevant output and contact help@myri.com.