

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
HAYSTACK OBSERVATORY
WESTFORD, MASSACHUSETTS 01886
 March 19, 2010

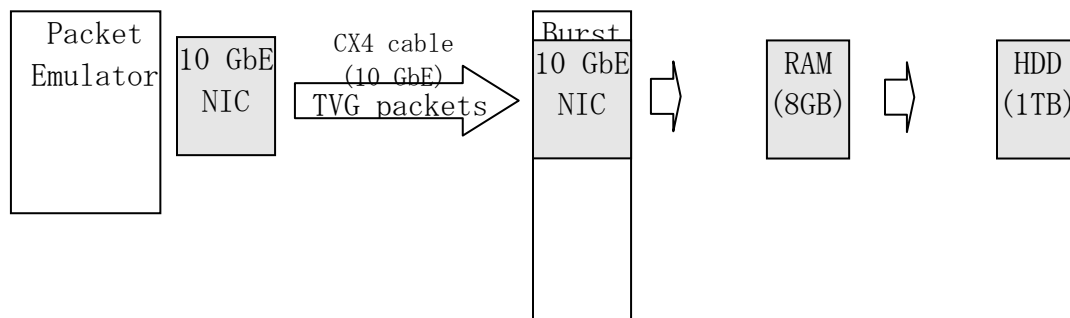
Telephone: 781-981-5400
Fax: 781-981-0590

To: UVLBI Group
 From: T. Cappallo
 Subject: Burst Mode Benchmarking

Preliminary testing is underway to determine throughput benchmarks for the Burst Mode System.

I. Components

- a) The Packet Emulator is a PC running Mark5CPacEmu to generate realistic data packets in TVG format. Currently, Mark5CPacEmu inserts a fixed character string in place of actual TVG-format data, but this will be changed as testing continues. Relevant specs for the current machine:
- 64-bit Intel Xeon X3330 (quad-core 2.66 Ghz, 6M L2 cache, 1333MHz FSB) running CentOS
 - Myri 10G-PCIE-8A-C 10 GbE NIC
 - 4 GB DDR2 800MHz RAM
- b) The Burst Mode System is a PC which will run the receiver software to buffer input in RAM and spool it to the HDD. Relevant specs for the current machine:
- 64-bit Intel Xeon E5462 (quad-core 2.80 GHz, 12M L2 cache, 1600 MHz FSB) running Debian
 - Supermicro AOC-STG-i2 2-port 10 GbE NIC
 - 8 GB DDR2 800MHz RAM
 - Barracuda 7200.11 400 GB HDD
- c) They are connected by one CX4 cable capable of transmitting up to 10 Gbits/sec. The incoming data, after being received by the NIC, is written to RAM either using standard system calls or by utilizing DMA routines, depending on the results of speed testing. The data are then spooled to the HDD at a lower rate.



II. Running the emulator

The Mark5CPacEmu program generates packets containing TVG data and transmits them over a UDP connection or raw socket. It provides a number of command line options for specifying the format and manner of delivery of these packets.

Typical usage for benchmarking over a raw socket:

```
m5L2 -d [dest MAC address] -i [eth interface #] -fb (Mark5B emulation) -t [duration in seconds] -r [data rate in Mbps]
```

III. Benchmarking

a) *Test setup.* Both NIC cards were set to 9000 MTU to support jumbo frames, with

```
/sbin/ifconfig eth2 mtu 9000
```

Mark5CPacEmu was run using 8800-byte packets with varying data rates, using tcpdump on the receiving computer to verify that the packets were being transmitted intact. Each run was timed by prefixing the Linux command time, to verify that the packets were being transmitted at roughly the correct data rate. On the receiving end, the packets were either written to disk or routed to /dev/null, neither of which appeared to impact performance.

b) *Results.*

-t (seconds to send)	-r (data rate in Mbps)	time (real execution time)	packets captured by BMS kernel	packets lost by BMS kernel
1	1024	3.095	14545	0
1	2048	3.095	29090	0
1	4096	3.068	58181	0
1	4608	3.060	65454	0
1	5120	3.017	72727	129
1	5632	3.041	74779	5221
1	6144	3.827	87272	6664
1	8192	6.987	116363	2479

c) *Conclusion.*

It appears that as currently configured, the Packet Emulator is capable of transmitting packets intact up to at least 5.5 Gbps, while the kernel on the Burst Mode System can receive packets intact up to at least 4.5 Gbps. This throughput is sufficient to begin developing the software RAM buffer for the Burst Mode System operating at 4Gbps.