# Cray Networking on Product Line Systems

**Jay Blakeborough**, *Cray Inc. and Friends*

**ABSTRACT:** *This discussion will focus on the networking capabilities and plans for currently supported product line systems. The Cray Networking Subsystem (CNS) will be introduced along with current performance reports. Future directions and research plans for improving the scaling of networking on the Cray X1 will also be presented.*

## 1. Introduction

It has been two years since my last networking update to the Cray User Group (CUG). This paper reflects on the decisions and events of the past few years, discusses the present, and provides a glimpse of the future for networking on Cray Product Line systems.

## 2. A Bit of History

Early Cray PVP systems provided networking through remote stations via a proprietary front-end interface. TCP/IP functionality was first provided with the UNICOS operating system. With the Cray Model E I/O architectures, HyperChannel, FDDI, and HiPPI were available as well as ATM OC-3 via a bus-based gateway across HiPPI.

The protocol between the mainframe and the I/O subsystems (IOS) was message based with the drivers accessing main memory data through a high-speed channel.

In addition UNICOS on GigaRing I/O systems provided 10/100T Ethernet. The GigaRing I/O protocol was table based and utilized a single Unified Networking Driver on the Cray mainframe. Although somewhat different than the Model E-based I/O, the basic capabilities were unchanged.

These systems performed quite well when executing large I/O across networking interfaces with large Maximum Transmission Units (MTU). For example, on HiPPI with a 64K byte MTU, large transfers could achieve nearly 700 Megabits per second (Mb/s) on the 800-Mb/s media. On interfaces such as Ethernet which utilize only a 1500-byte MTU, Cray systems struggled to achieve 30 Mb/s on a 100 Mb/s media.

Cray machines were designed for large number crunching applications. Many of the trade-offs made to allow for this were and are not favourable to what we now term *Traditional Networking*.

Gigabit Ethernet (GigE) provided a unique challenge for these systems. The media is capable of 1000 Mb/s, but the standard MTU is only 1500 bytes. Even though 9000-byte Jumbo Frames have become a de facto standard, the traditional Cray machines were still not capable of utilizing even half of the bandwidth. To make GigE available, Cray worked with Essential Communications (now SBS Technologies) to create the Cray Layer-7 Router (L7R). The router provided a bridge from HiPPI to GigE along with specialized proxy software to coalesce small MTU traffic into large packets for the Cray machines. The router was first released in late 2001. It provided 90 Mb/s on 100T Ethernet and 350 Mb/s on GigE.

## 3. Cray X1 Beta Networking Plan

In parallel to some of the Cray L7R efforts other groups were working on the Cray X1 system. Plans for the system indicated that it could potentially perform much better than earlier machines did on smaller packet interfaces. The I/O plan contained a PCI-X bus directly attached to the mainframe through a System Port Channel (SPC). To keep the cost of development down (and thus lower the entry cost of the system), it was decided that network and disk access would be through the same HBA type – in this case Fibre Channel (FC). To provide other interfaces, networking would be bridged through the Sun I/O system.

Testing began in the fall of 2001 with a set of Sun servers. One Sun was designated as the Cray X1 mainframe, another the ION, and the third was used as a GigE testing endpoint. So, the network route was from the pseudo Cray X1 via Fibre Channel (FC) to the ION and finally to the other Sun via GigE.

Initial results were so poor that we assumed either configuration and/or hardware problems were to blame. Rates of 11-to-15 Mb/s were common. More confirmation and testing revealed that IP-over-FC on the Sun could produce only about 40 Megabytes per second (MB/s) on the 200 MB/s interface. Bridging to a 1500-bytem MTU network seemed to only make things worse.

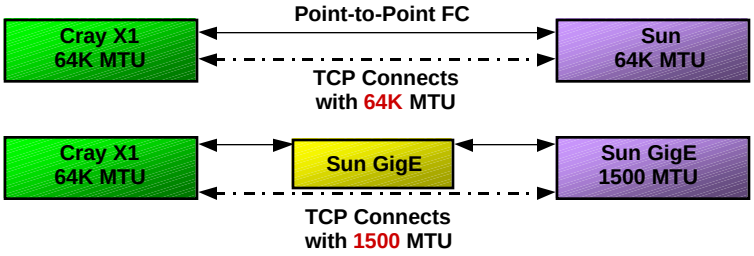
## 4. Cray X1 Networking

Cray X1 I/O was not yet available in early 2002. Many believed that the I/O capability of the Cray X1 would not have similar issues, but there was enough concern to warrant the investigation of other options. At least three options were considered. That is, the author can only recall these three.

### *4.1 Tune Networking Parameters on the Sun and the Pseudo Cray X1*

As we discovered with UNICOS at the 2001 CUG, there was simply no amount of tuning that could correct the problem we expected (and later verified) when Cray X1 I/O became available.

To allow transmission across dissimilar networks, TCP negotiates to the least common denominator (MTU) throughout the path to the connection endpoint. In the case of directly connected hosts using a media capable of 64K-byte MTUs, the connection will be built and maintained using 64K-byte packets. When another network (or host) is involved that does not allow 64K-byte packets, the connection is negotiated down to the smallest MTU (in this case 1500 bytes on GigE).



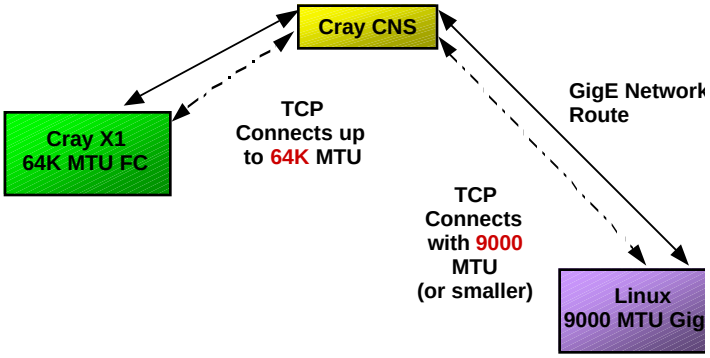

While there are ways around this, like forcing the first machine in the network path to fragment the packets, these practices are not considered good network neighbor behavior. You are then forcing other machines to perform work (assembly and reassembly). Also, there was/is no standard TCP capability to provide packet reassembly before the data gets to the Cray mainframe.

### *4.2 Utilize a GigE Off-load Network Interface Card (NIC) Directly Attached to the Cray X1*

This ended up to be a fairly short research project. There are many NICs that perform some level of TCP off-load for the host to which it is connected. *Connected* here is the operative word. The Cray X1 I/O provides PCI-X connectivity. It does ***not*** allow for PCI connectivity. At the time, there were no PCI-X-based GigE Off-load NICs available. As of the date of this paper, there are some, but they do not provide the level off-loading we believe is necessary. See a later discussion on off-loading for more details.

### *4.3 Utilize/Improve the Cray L7R Technology*

Some brief experiments with IP-over-FC on commodity systems showed significant promise over the Sun-based solution, so we chose to investigate further. Using the commodity-based hardware, and the TCP assist functions developed with the Cray L7R, we would be able to utilize well-tested, commodity NICs and provide much greater bandwidth to the customer network.




For those not familiar with the TCP assist functions of the Cray L7R, the drawing above highlights its methods. When a connection is made from the Cray X1, TCP actually connects via Fibre Channel at 64K MTUs to the Cray Network Subsystem (CNS), the remainder of the connection is made across GigE using 1500 to 9000 byte MTUs. While there are two connections. The user on the Cray X1 and the networking end point are typically not aware that this is happening. The Cray CNS disassembles and reassembles data across the connection to allow the Cray X1 to maintain a large-packet connection to a small-MTU network and thus maintain reasonable performance to machines connecting with GigE.

We have had generally good experiences with the Cray L7R product. The early prototype performance achieved in the lab was encouraging, so we chose to pursue this option while the original plan (Sun ION) was prepared for shipment with Early Production Machines.

Our in-house experience with ION-based networking on the Early Production Machines confirmed the suspicions that this planned method would not meet the needs of our customers. We decided to pursue the CNS development for inclusion with the first customer product shipments of the Cray X1.

## 5. Cray Network Subsystem (CNS)

The Cray Network Subsystem (CNS) was put together starting in June of 2002. We chose a new commodity platform (different than the Cray L7R). What is the platform, you ask? Well today, it is based on a specially configured Dell 2650. As most readers know, commodity hardware shipped today can be quite different from that shipped tomorrow. Vendors in this space (small servers, disks, and networking) update and end-of-life their products frequently and sometimes without much warning. Comparable products become available at lower prices the very day you make your purchase. This has been and will be a significant challenge for our purchasing, development and qualifications folks. Our goal is to provide Cray customers with the best value at the time their CNS is

shipped. While it may be possible, we have no plans to support a generic, customer-provided CNS platform.

Back to the CNS…we are utilizing IP-over-Fibre Channel to connect the CNS to the Cray X1. Given recent demand exceeding our supplies of the Cray L7R, we have also created a version of the CNS that uses HiPPI to connect to previous Cray mainframes. We are supporting two customer network GigE interface types – copper and fiber. We also support HiPPI as an external network connection when used on the Cray X1.

The first release of the CNS (1.0) was made available in December of 2002. We have sent out two updates since then and are planning a new release (1.1) in June of this year.

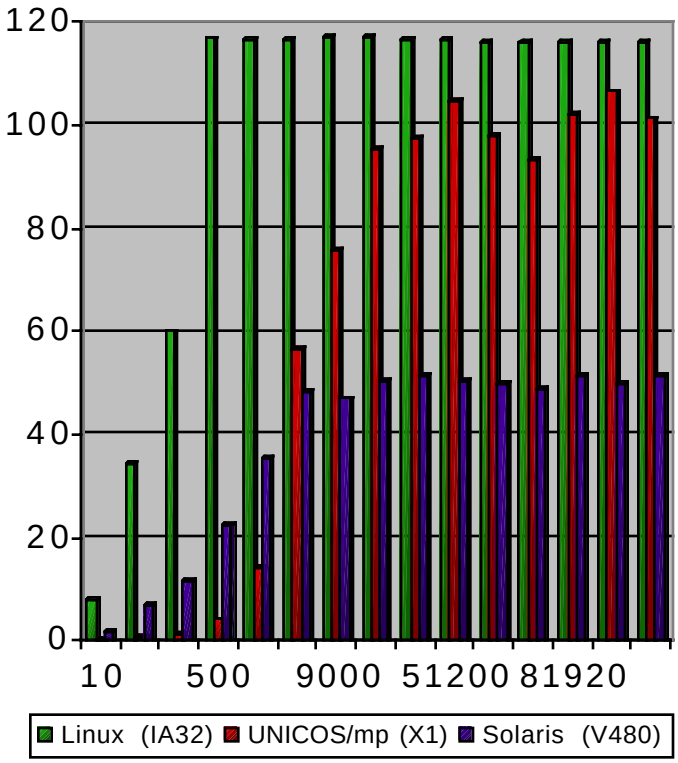
## 5. Cray X1 Networking Performance

Given our history of networking at Cray, I must say that I am very proud of all the folks that have helped to make the CNS a reality. Since its first shipment, performance has increased almost 50% as we continue to explore areas in the CNS and the UNICOS/mp operating system that can be improved.

For our performance testing, we utilized three platforms that were readily available to us:

- Linux 2.4.18 running on 2.4 Ghz, Dual-CPU Intel IA32
- UNICOS/mp running on the Cray X1
- Solaris running on a dual processor V480 server

The Linux platform was used as a consistent connection end-point system. All tests were run using a 1500-byte MTU GigE network.




This and following graphs are better viewed on a color printout or screen. A PowerPoint presentation with larger versions of the graphs is also available with the CUG 2003 proceedings.

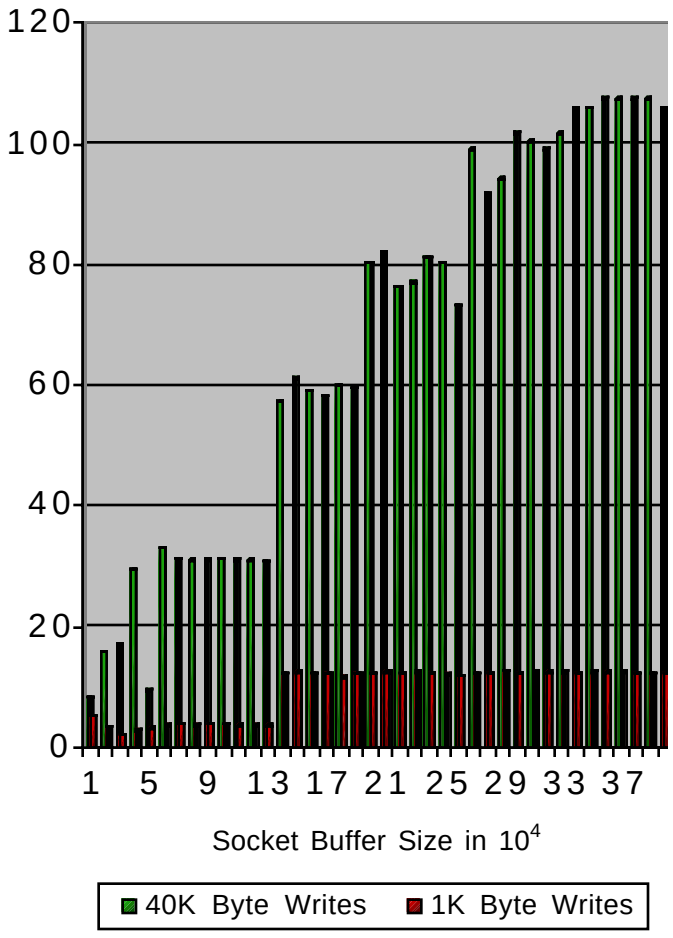
This graph shows the relationship between write size (X-axis) and performance in MB/s (Y-axis). On our test platform, Linux networking performance ramps up rather quickly to achieve near wire speed rates at write sizes of only 500 bytes!

Solaris on the V480 ramps to its maximum rate of approximately 50 MB/s at around 5000 bytes.

UNICOS/mp on the Cray X1 ramps more slowly than Solaris, but over takes it at around 3500-byte writes. It also exceeds the top performance of Solaris on this platform by a factor of 2. The graph also shows some anomalies as write sizes increase. These unexpected drops are believed to be a function of pages sizes on the Cray X1, but more investigation is needed.

As we focus on smaller write sizes, it is clear that we have things to work on. I do not anticipate that we will be able to rival Linux on the IA32 in this space, but we will work to improve it.

The next graph shows the affect on performance given a small (1K byte) verses a large (40K byte) write size over increasing socket buffer sizes. For small write sizes, the



performance increases substantially using a socket buffer size of at least 135K bytes, but further increases in buffer size for these writes are not appropriate. Given larger write sizes, bandwidth continues to improve until the socket buffer reaches almost 330K bytes, On UNICOS/mp 2.1, the default socket buffer size was set to only 61440 bytes – hardly sufficient given these tests. To allow for maximum throughput by covering some network system call latencies the socket buffer size has been increased to 458752 bytes for the up coming UNICOS/mp 2.2 release.

## 6. CNS Plans

The next major release of the CNS (1.1) is planned in June of this year. It will contain:

- GigE and Fibre Channel driver fixes and updates
- Improved installation and configuration
- Functions to help make upgrades easier

Future releases will support multiple Fibre Channel links to the Cray X1 mainframe. The primary goal of these additional links is to provide path resiliency between the mainframe and the CNS. As secondary, though not unimportant goal, is to allow for increased performance to the CNS in hopes of providing multiple GigE connections to the customer networks.

## 7. What's the Buzz?

I have had many opportunities to review several presentations and communications regarding networking technologies that could provide some benefit to the Cray X1. Many of the ideas are intriguing on the surface but quickly lose their shine as the details emerge.

### *7.1 TCP Off-Load*

One of these technologies is referred to as a TCP Off-Load Engine or (TOE). The concept has been around since the early days of GigE. The hope is that processor-consuming functions can be off-loaded to the NIC. The first round of off-load consisted of checksum and interrupt hold-off. The packet checksums are generated by the NIC and the NIC holds off interrupting the host until a configurable number of packets have arrived. Without the latter, a system could get interrupted as much as 80,000 times per second. One of the first optimisations on the CrrayX1 was to utilize our Vector hardware to checksum the packets. The vectorized checksum routines showed performance similar to not calling the scalar checksum routines at all.

The next versions of TOE implement transmit segmentation. The driver has some special hooks into the hosts TCP stack to allow larger chunks of data to be transmitted to the NIC where it is carved up into MTU-sized packets. The NIC also helps by generating the TCP and IP headers. The key word here is *transmission*. This level of TOE provides virtually no assistance on the receive side of the connection beyond checksum and interrupt hold-off. While transmission is the most expensive side of the connection, it is not the only side. Because the TCP protocol requires acknowledgements be seen from the receiving side before it will send more data, the ability to receive and process acknowledgement quickly is essential.

Finally, the full off-load which is sometimes referred to as *fast path* off-load is still in its early stages. We will be evaluating some of these NICs as they become available, but currently have no plans to place them directly into the Cray X1.

I should also point out that most of the current TOE implementations are available only on PCI-based NICs, not PCI-X. This does not rule them out as candidates for the CNS, but does not allow us to use them directly attached to the Cray X1.

### *7.2 Trunking/Bonding*

Channel bonding was devised in Linux as a method to multiplex multiple modem connections to achieve increased networking throughput. It has received some work since then to take advantage of multiple 100T Ethernet connections and to interface with Cisco's implementation of IEEE 802.3 Link Aggregation which they call Etherchannel.

With Etherchannel and GigE, the channels are assigned to a single IP address. This implementation can provide increased network bandwidth and resiliency to a system, but does not provide increased single-stream bandwidth. For example, using 4 GigE connections as an Etherchannel allows 4 connections to 4 other machines to run at the potential peak single-stream rates. It does **not** allow a single network connection (e.g., an `ftp`) to obtain rates greater than the capability of a single GigE connection (1 Gb/s).

As we investigate these technologies we will look to utilize them when possible for our mainframe-to-CNS connections as well as providing customers access to Etherchannel through the CNS.

## 8.0 What's the Story on 10 GigE

First of all, it must be noted that at this time, Cray Inc. has no current plans to provide a 10 GigE connection on the Cray X1 Series mainframes. If promising technologies become available we will certainly investigate them, but our plans will be based on our findings and our ability to take advantage of the products in our I/O structure.

Other Cray project teams are investigating early 10 GigE products. Some of the offerings that were on network vendor road maps with planned availability in late 2003 have slipped as much as a year already. One can only speculate, but a number of factors could be contributing to this delay – including the fact that NICs and switches are going to be very expensive for early adopters, most current copper-based infrastructures will not be capable of supporting 10 GigE, and a true *fast path* off-load will likely be required for most systems to take advantage of the bandwidth. The biggest issue with the *fast path* off-load is that there are several vendors pursuing non-standard, proprietary protocols that require changes to the operating system in areas that were previously unnecessary. Like everyone else, we would like to purse the methods that become at least a de facto standard. Unfortunately it may take some time for the standard to *coagulate*.

## 9.0 Looking Ahead…

Comments made throughout this paper and especially in the previous section are based on a Cray Product Line Networking Vision that our team developed last year;

*We will utilize current mature networking technologies to provide industry-standard, single-stream networking performance to our customers. We will design and implement methods to provide system aggregate network bandwidth of at least 8 times the single-stream performance.*

As we move forward in the coming years, we will purse this vision on Product Line systems. Some will consider the first part of the vision to be a bit lack-luster in that we are not hoping to lead the industry in performance. Statements like that were made at Cray Research, Inc. during a time when almost anything that performed well was running on or attached to a Cray system. The reality of this situation has changed and networking, while exceedingly important, is no longer an area in which choose to make the investment necessary to lead. Realistically, most of the other systems that our customers are and will run will have average networking capabilities. Our challenge is to strike a balance between reality and hype, between research and production, and between cost and value to our customers.

Keep those cards and letters coming. We really do appreciate your feedback in this area.

## About the Author

Jay Blakeborough is a software I/O manager for Cray Inc. He began working for Cray Research, Inc. in 1985 and has held a variety of technical and leadership positions within the software divisions of CRI, SGI, and Cray Inc. He can be contacted via email at jb@cray.com. His office is at the Cray Inc. facility in Mendota Heights, MN. Further contact information and directions are available on the Cray Inc. web site: http://www.cray.com.

The *friends* listed with the author are the many capable I/O engineers, testers, and support personnel who have provided information for this work either directly or indirectly through their efforts to improve networking capability and performance on Cray Inc. Product Line systems.