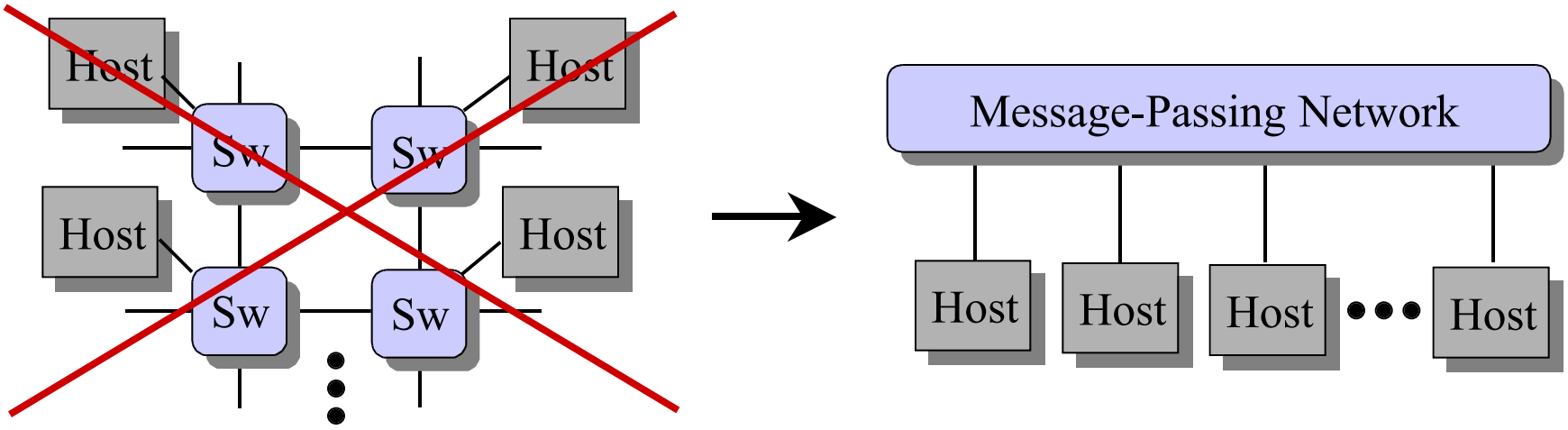
Recent Advances in Cluster Networks

Charles L. Seitz
CEO & CTO
Myricom, Inc.
chuck@myri.com

Newport Beach, California
11 October 2001 (10-11-01)

The Goal: Networks that are scalable and "flat"



- Message-passing networks that provide maximal throughput even when scaled to large numbers of hosts.
- Eliminate the topology considerations in process placement.
  - Let process placement be guided by load-balancing and cluster-management considerations, and without regard for the mapping of the communication patterns of the computation to the network topology.

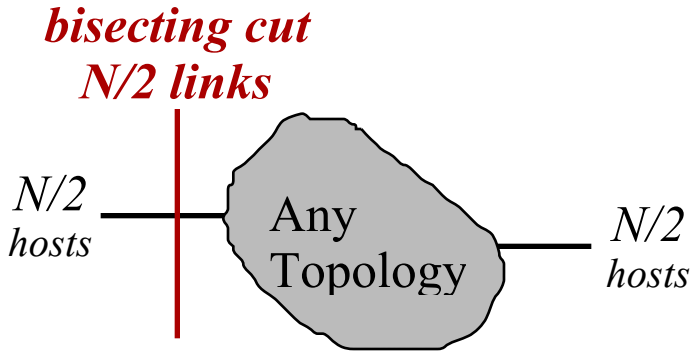
Historical footnote: For similar reasons, random-access (core, later semiconductor dRAM) primary memory displaced sequential-access (drum & disk) memory for similar reasons.

Part 1:
Preliminaries
(Topology Concepts)

Minimum Bisection

A message-passing network's *minimum bisection,* measured in links, is defined mathematically as the minimum number of links crossing any cut that bisects the hosts (half of the hosts on one side of the cut, the other half on the other side). The minimum bisection provides a metric for the minimum traffic-handling capacity of a network, no matter what the communication patterns between the hosts may be.

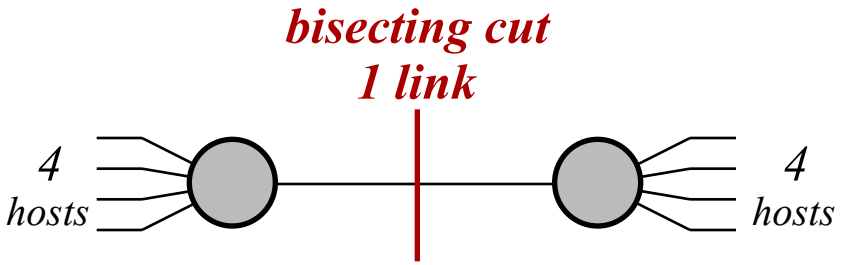
The upper bound on the minimum bisection of a network of *N* hosts is *N/2* links, because, no matter what the internal topology of the network, there will be bisecting cuts possible across half of the host links:



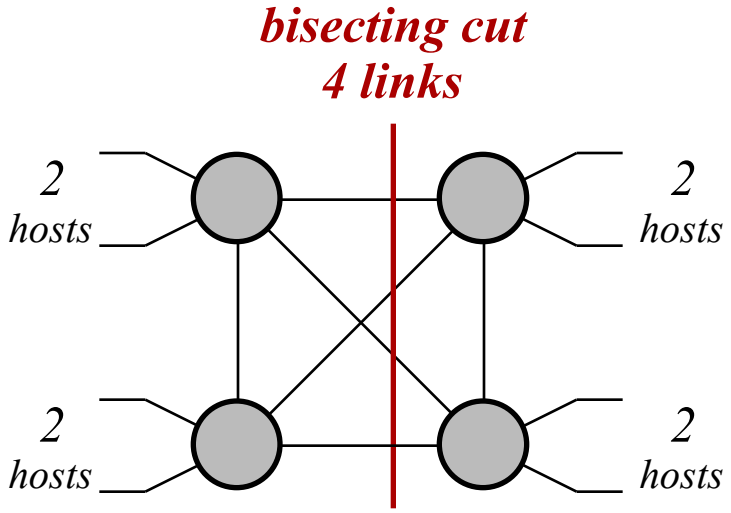
A network topology that achieves this upper bound is said to have *full bisection.*

Minimum Bisection -- Example

As an example of the relevance of the minimum bisection, suppose that we wished to connect 8 hosts, and the largest (crossbar) switches available had 5 ports. One way to connect the 8 hosts would be:



The bisecting cut that exhibits the minimum bisection is evident. If the 4 hosts on the left were sending messages to the 4 hosts on the right, and vice versa, the total traffic-handling capacity of the network would be throttled to that of a single link. The host links would be limited to an average of 1/4 of the link data rate, corresponding to the bisection being 1/4 full. Also shown is a full-bisection network to connect the 8 hosts with 5-port switches.



Rearrangeable Networks

A message-passing network is said to be *rearrangeable* if can route any permutation without blocking. Unlike a single crossbar switch, a special case of a rearrangeable network, the set of routes used from host to host is permitted in a rearrangeable network to be different for different permutations.

It should be evident that a network that is rearrangeable necessarily has full bisection. The converse is not true. A network may have full-bisection but not be rearrangeable.

For the distributed computations performed on clusters, it is generally not practical to coordinate the routing to take advantage of a network being rearrangeable. However, it is often easiest to demonstrate that a network has full bisection by showing that it is rearrangeable.

Is this a new problem?

A Study of Non-Blocking Switching Networks

By CHARLES CLOS

(Manuscript received October 30, 1952)

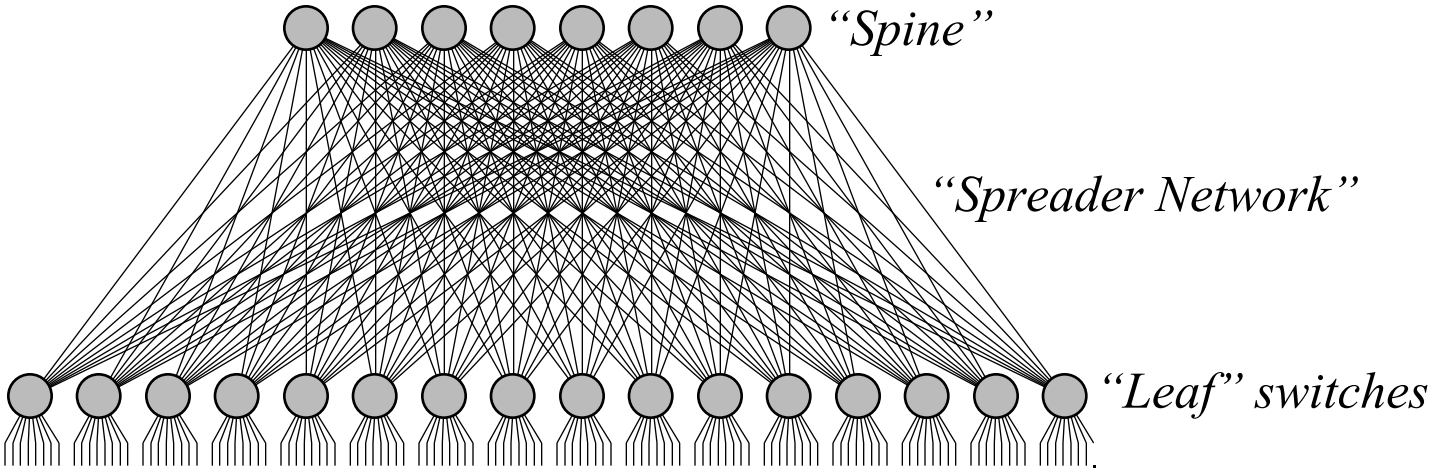
This paper describes a method of designing arrays of crosspoints for use in telephone switching systems in which it will always be possible to establish a connection from an idle inlet to an idle outlet regardless of the number of calls served by the system.

INTRODUCTION

The impact of recent discoveries and developments in the electronic art is being felt in the telephone switching field. This is evidenced by the fact that many laboratories here and abroad have research and development programs for arriving at economic electronic switching systems. In some of these systems, such as the ECASS System,* the role of the switching crossnet array becomes much more important than in present day commercial telephone systems. In that system the common control equipment is less expensive, whereas the crosspoints which

***Bell System Technical Journal* 32, 406-424 (March 1953).**

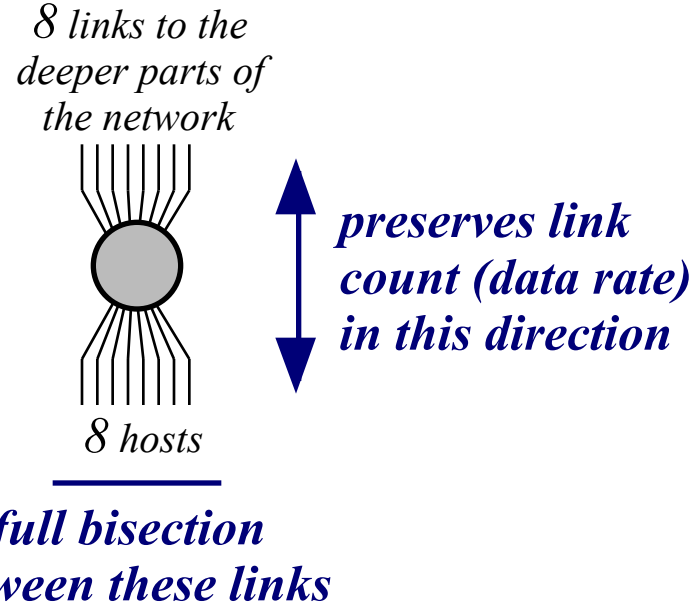
Clos Networks



Clos networks are named for Charles Clos, who introduced them in a paper titled "A Study of Non-Blocking Switching Networks," published in the *Bell System Technical Journal* in March 1953. Clos networks are rearrangeable: researchers at that time were seeking ways to route arbitrary permutations of telephone connections. Clos networks thus exhibit full bisection The Clos network topology has other excellent properties -- scaling to large sizes, modularity, and multiple-path redundancy -- that make it an ideal topology for cluster networks.

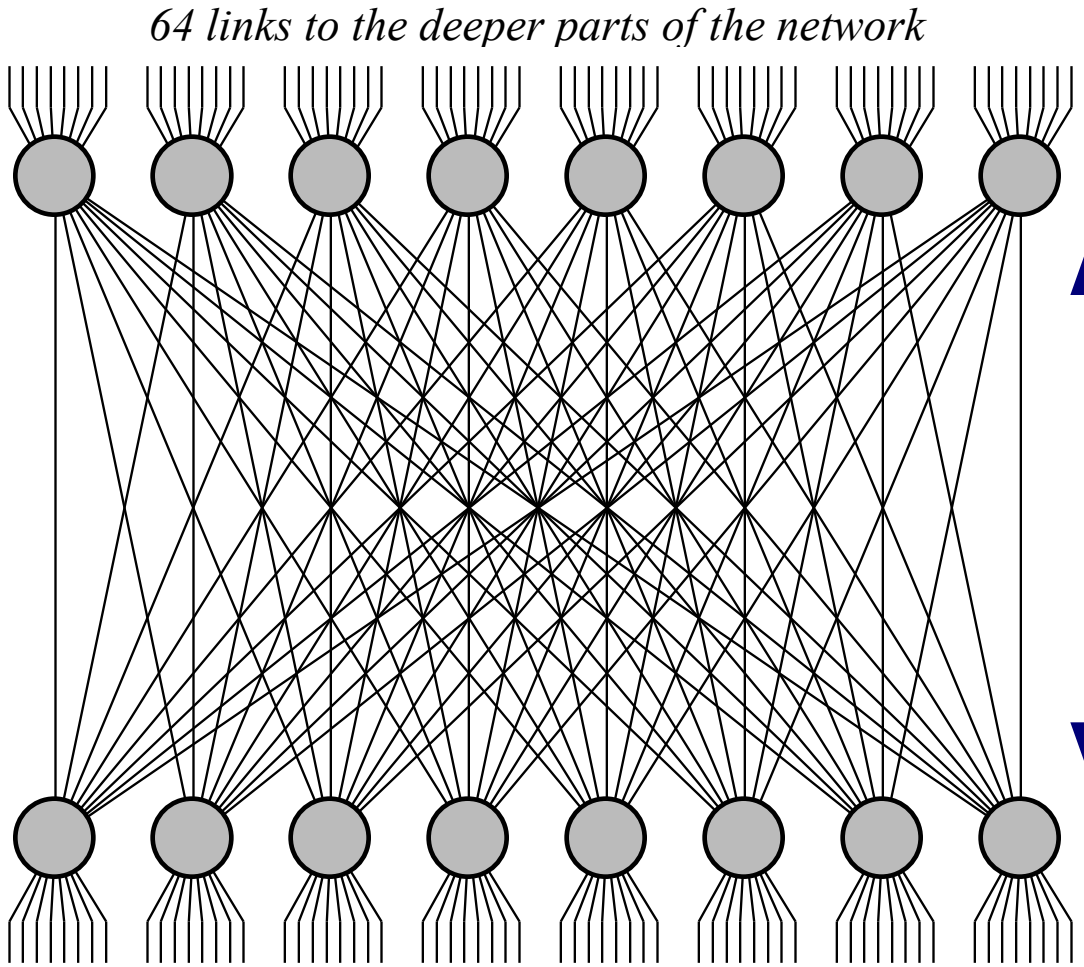
Why Does a Clos Network Have Full Bisection (intuitive)?

Observe that 8 ports of the leaf switches connect to hosts; the other 8 ports connect to the spine switches. It is possible that all of the packet traffic to and from the 8 hosts connected to a single leaf switch is to or from hosts connected to other leaf switches; thus, the leaf switches must have at least as many links to the spine switches as to the hosts.



The proof that the 128-host Clos network of 16-port switches is a rearrangeable network -- a stronger statement than that it exhibits full bisection -- proceeds by showing how a set of routes can be found for any permutation. Once the problem is cast into combinatorial terms, the proof is made by appealing to Hall's Theorem (1935) on systems of distinct representatives.

Apply the Same Principle to Scale Beyond 128 Hosts

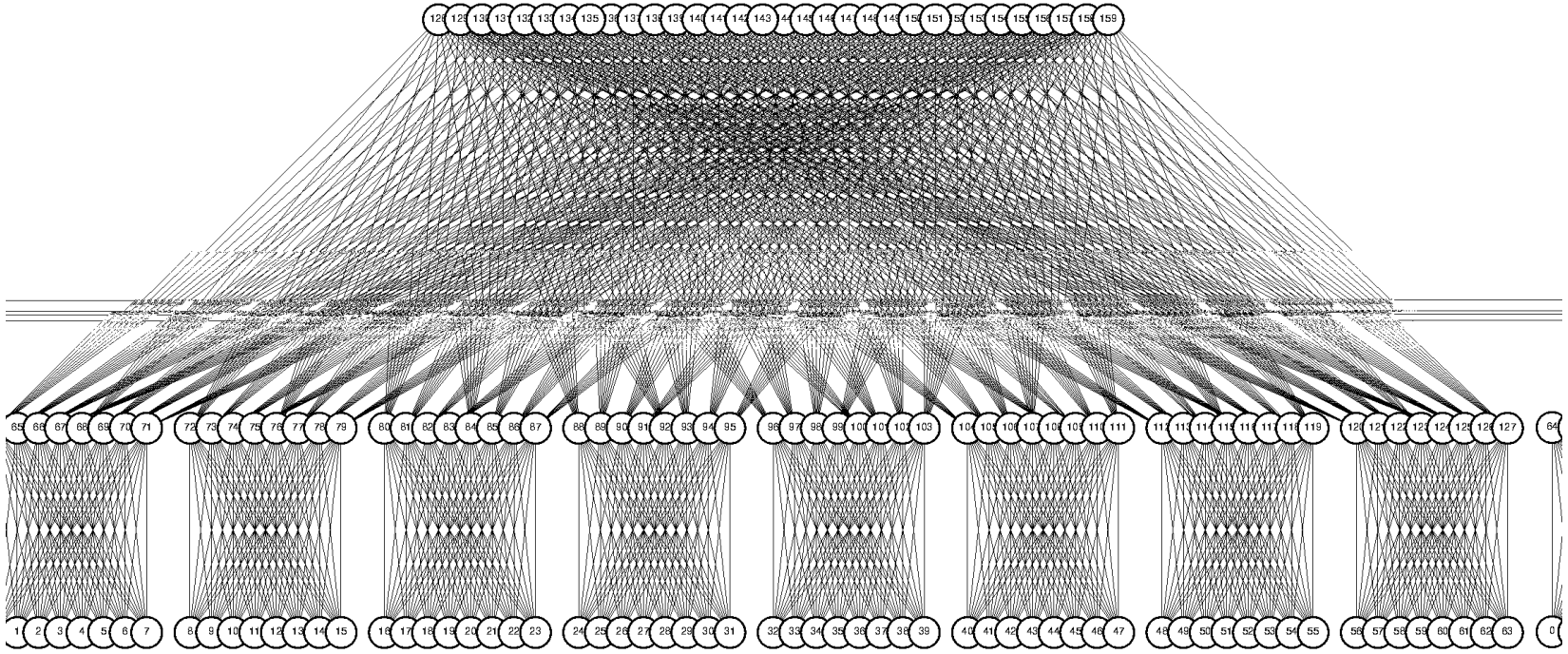


preserves link count (data rate) in this direction

"Clos64+64"

full bisection between these links

Example: Myrinet Clos Network for 512 Hosts



The 512 hosts connect to 8 ports on each of these 64 "leaf" switches

- 160 16-port switches (2,560 switch ports); 1,024 switch-to-switch links; diameter 5.
- The bisection data rate (total throughput) is 1.024 Terabits/s (128 GigaBytes/s).
- This network is routine today, and can scale at a similar cost/host to 8,192 hosts.

Part 2:
Technology

In the Beginning, …

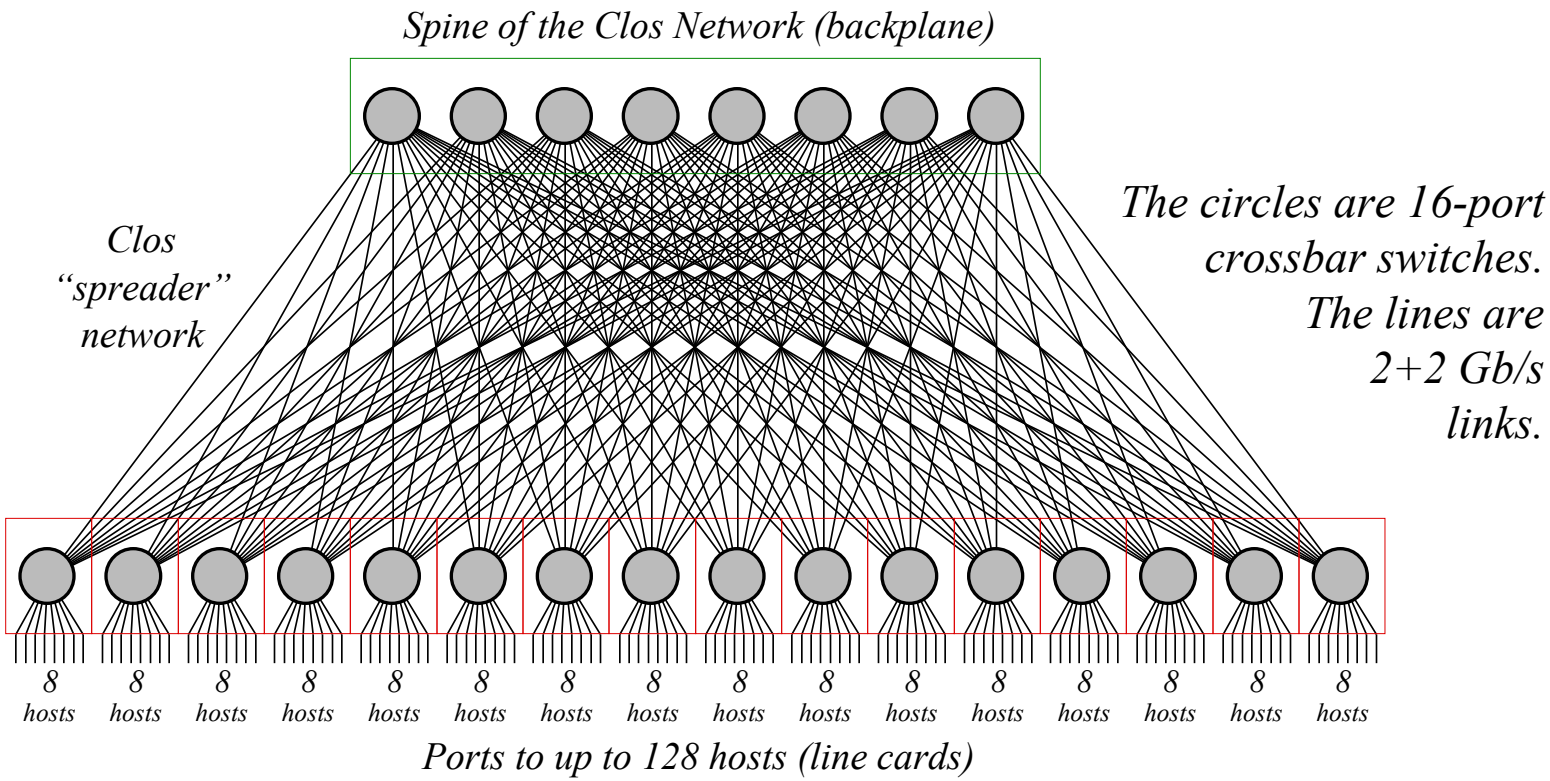
… Myricom packaged only single switches (8 ports, and later 16 ports), and encouraged people to use whatever topology appealed to them. The mapping (topology discovery) process and the route computation (up*/down* algorithm) assured that any host could send messages to any other host, and that the communication would be deadlock-free.

As people started building larger and larger clusters, we started getting a lot of questions about which topology to use. "We've got this 32-host cluster that we want to expand to be a 50-host cluster. What topology should we use?"

We've known about Clos networks since before Myricom was founded, but the Clos and other full-bisection network topologies were expensive to cable together from individual switches.

Finally we realized that we could and should build networks and sub-networks "in a box." In this way we could achieve economies for our customers by sharing power and monitoring among a group of switches, and by connecting many or most of the inter-switch links internally rather than with external cables.

Myrinet-2000 128-Host "Network in a Box"

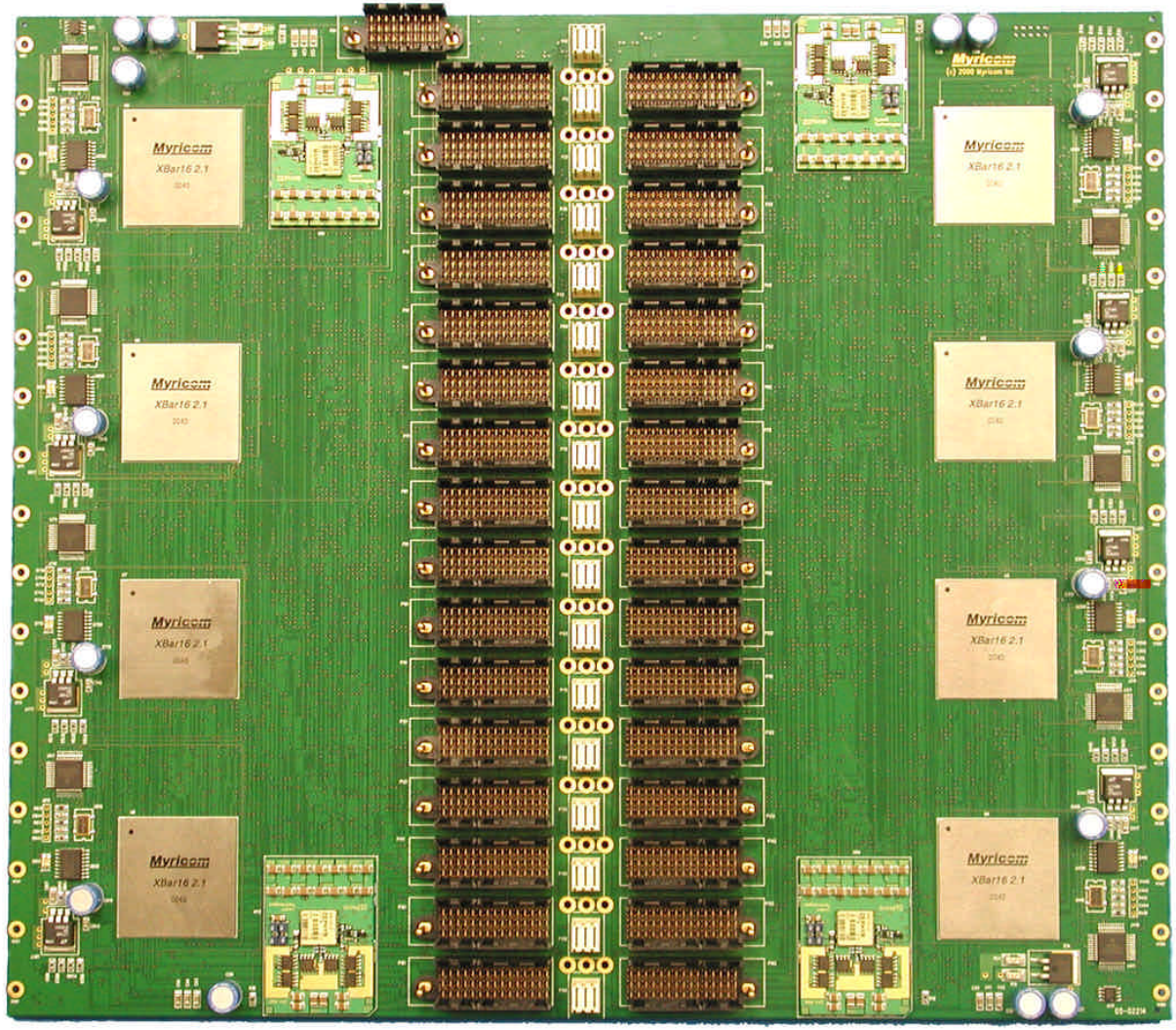


Different types of line cards have Fiber, Serial, SAN, or legacy LAN ports
The Fiber and Serial line cards are also made in a "spine" version without a switch

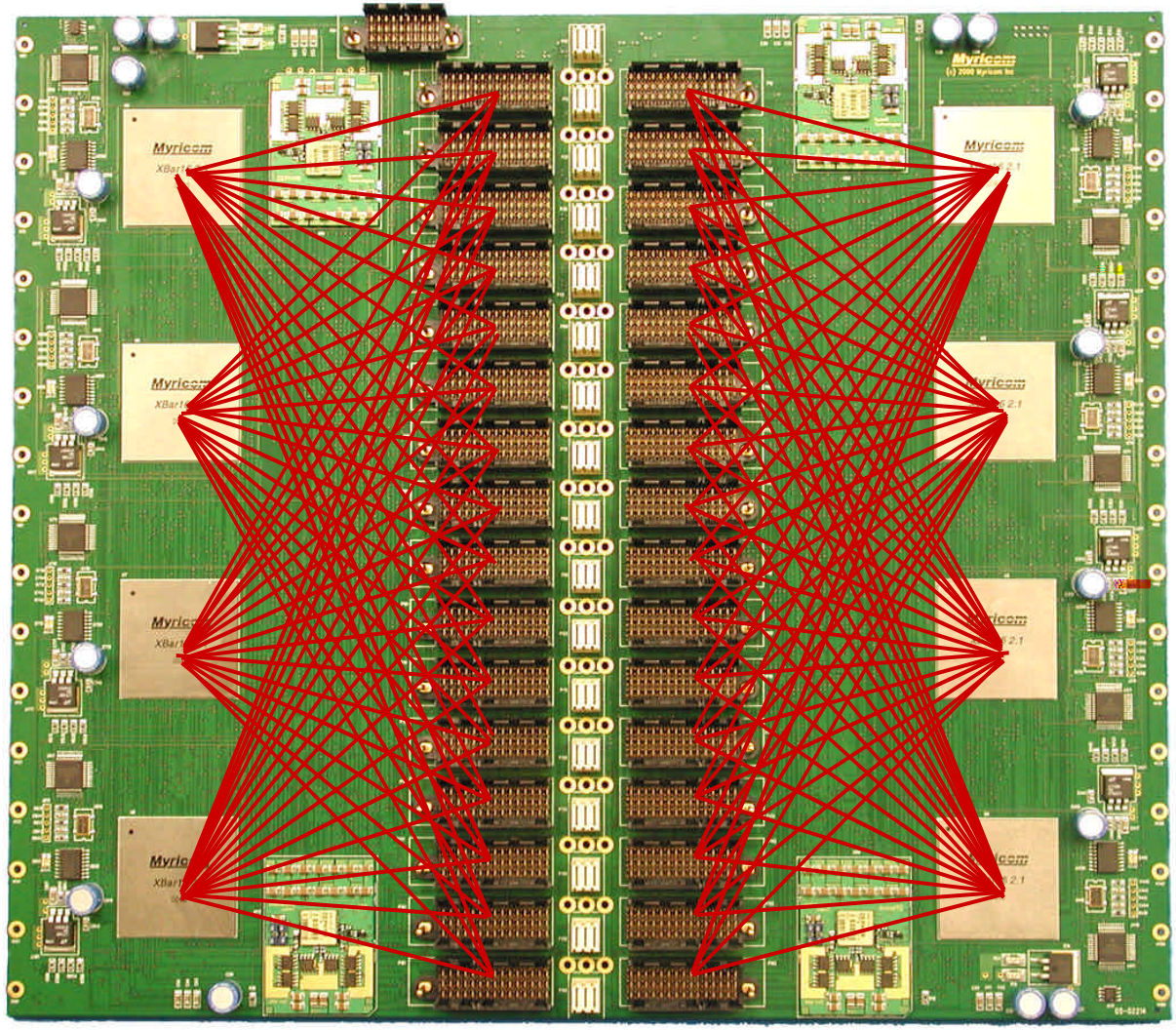
This family of products support hot-plugging of line cards, fans, and (soon) dual redundant power supplies. Microcomputer monitoring provides extensive diagnostic capabilities, and management features needed for large installations and for high-availability applications.

Backplane of the M3-E128

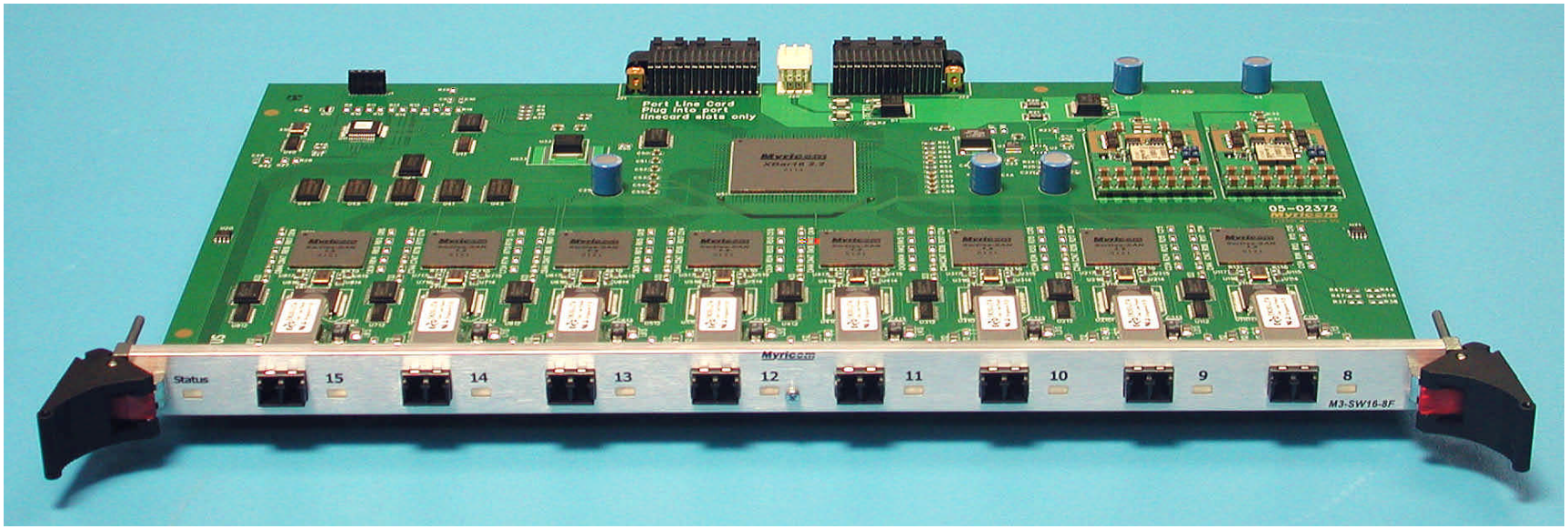
(Throughput = 1/4 Terabit/s)



Backplane of the M3-E128 (lots of PCB wires!)

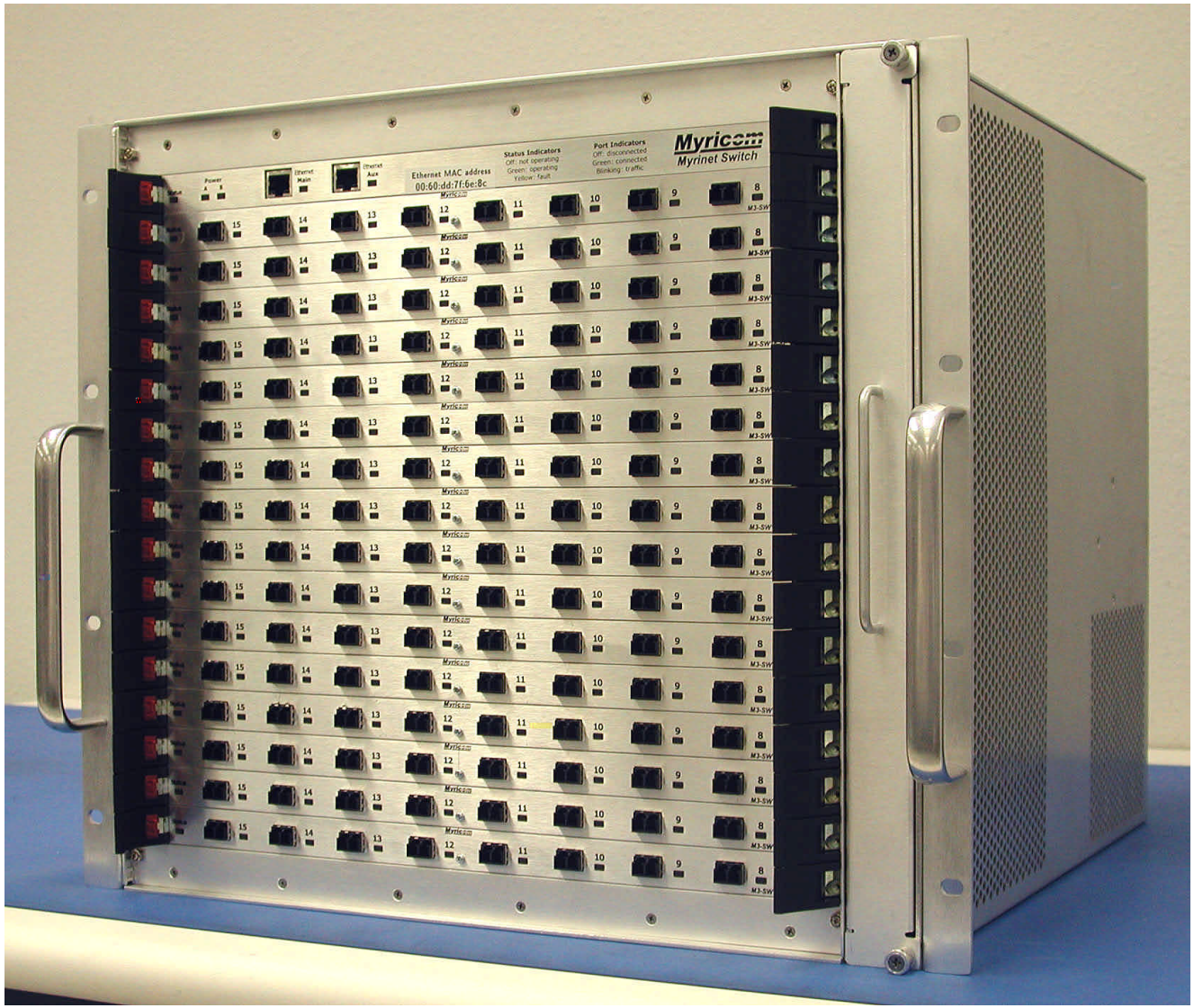


M3-SW16-8F (Fiber) Line Card

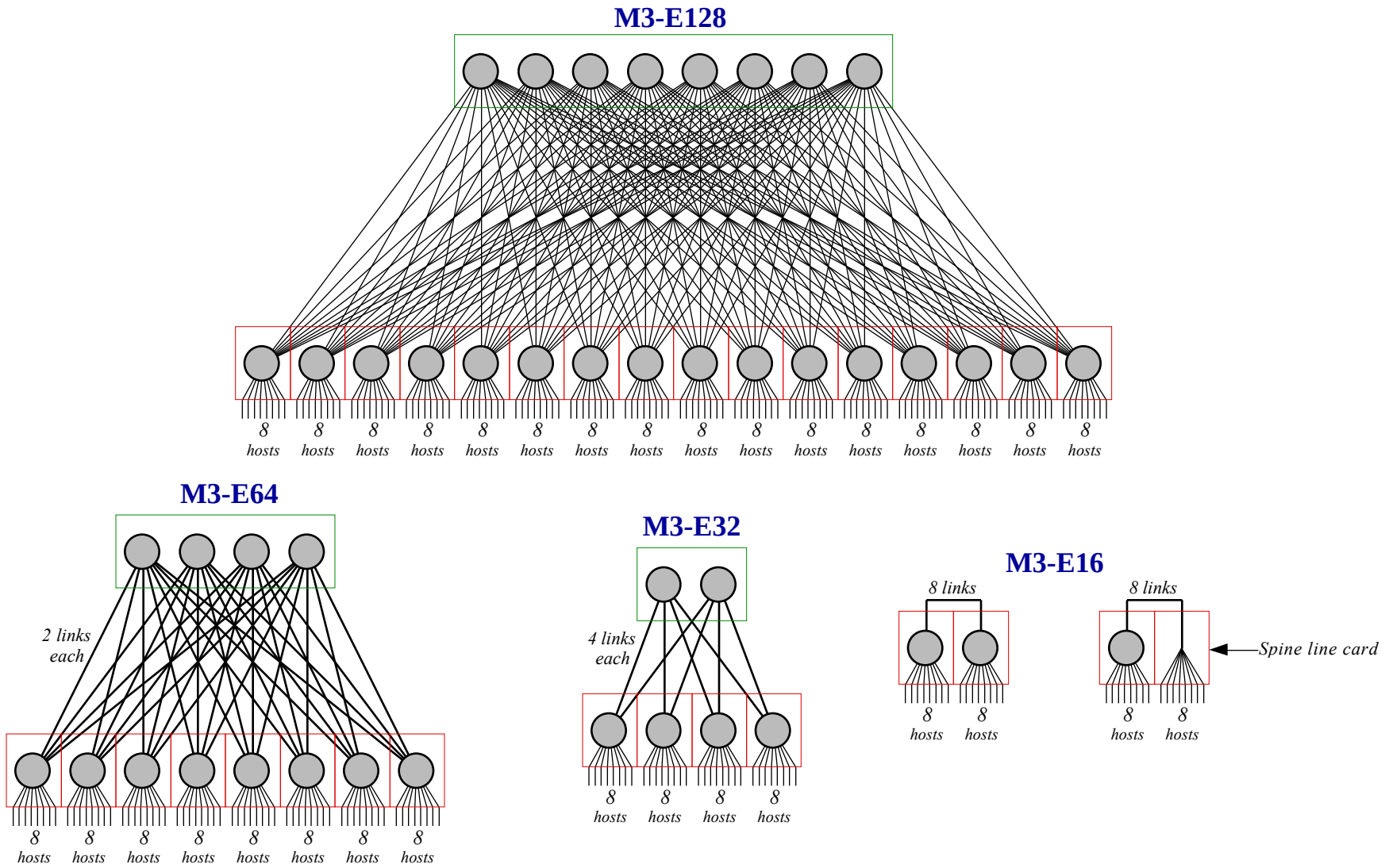


8 Fiber ports through the front panel, and 8 SAN ports to the backplane. Dual-redundant 12V input, with high-reliability Lucent switching regulators (upper right) to generate 3.3V and 2.5V. The small microcontroller (upper left) monitors the XBar16 scan path, fiber-conversion circuits, voltages, and temperatures. In addition to being able to report this information to the main microcontroller, the small microcontroller can shut down individual ports, or, if necessary, the 3.3V and 2.5V power.

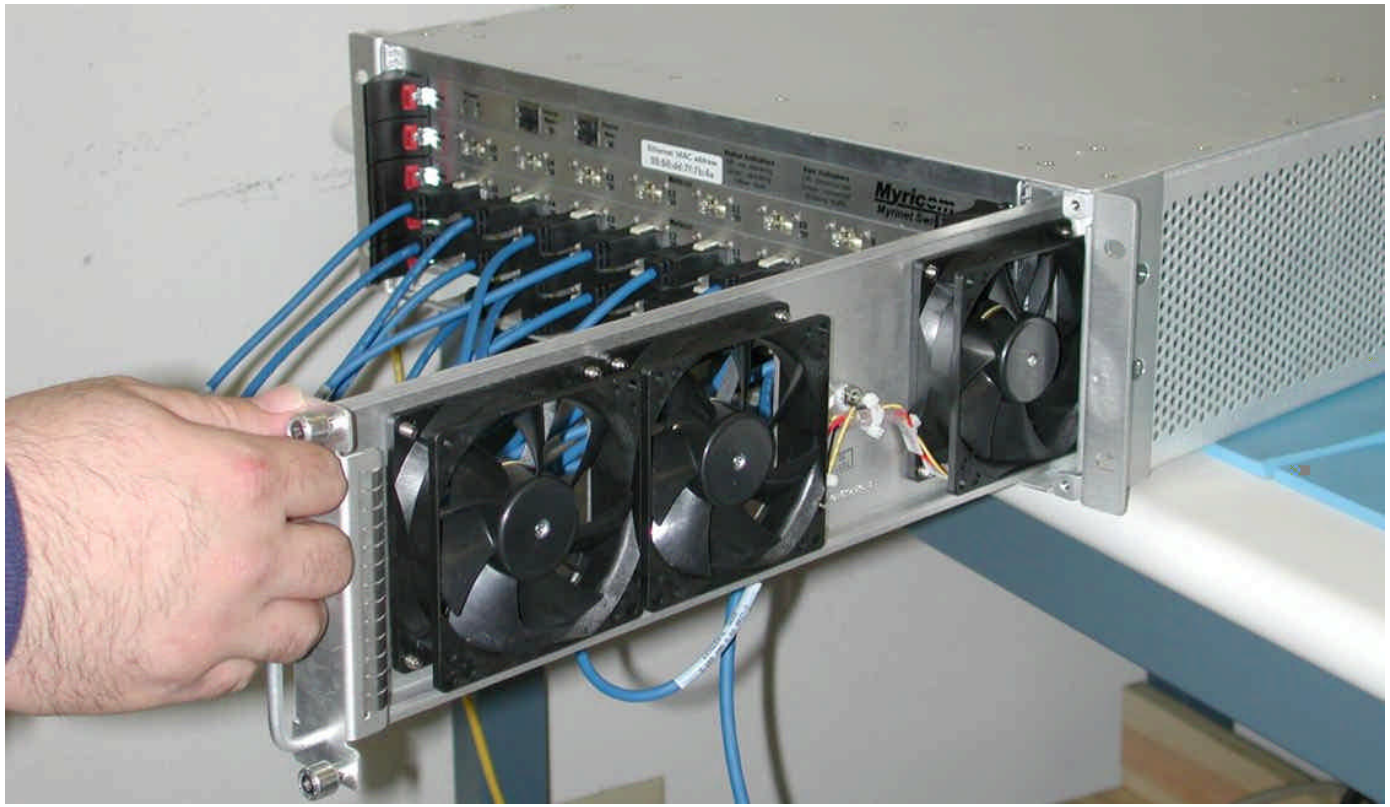
128-Host Clos “Network in a Box”



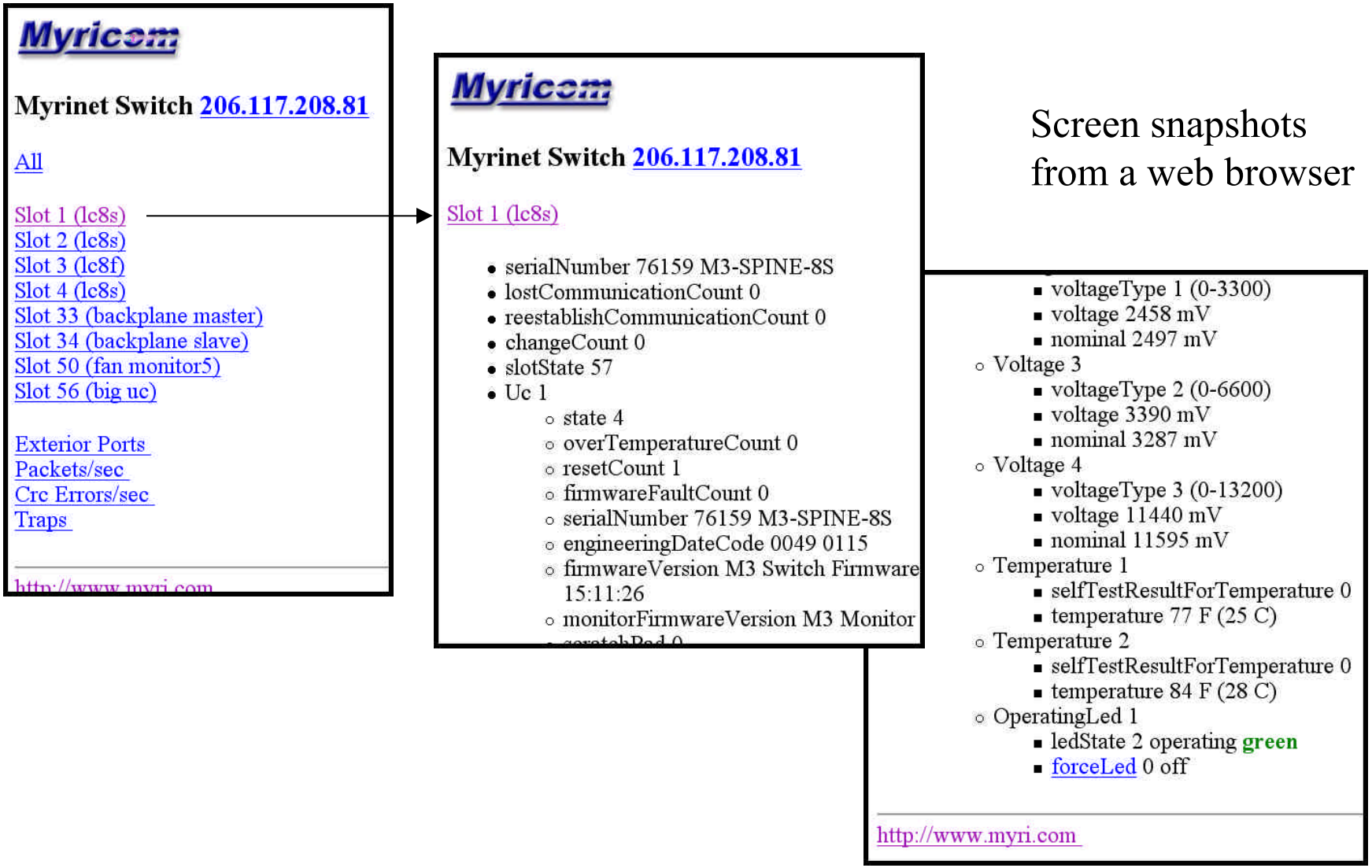
The Family of Myrinet-2000 Switch Products



Fan-Tray Replacement in a 32-Host “Network in a Box”



Examples of the Monitoring Information



Screen snapshots from a web browser

Part 3:
Dispersive Routing

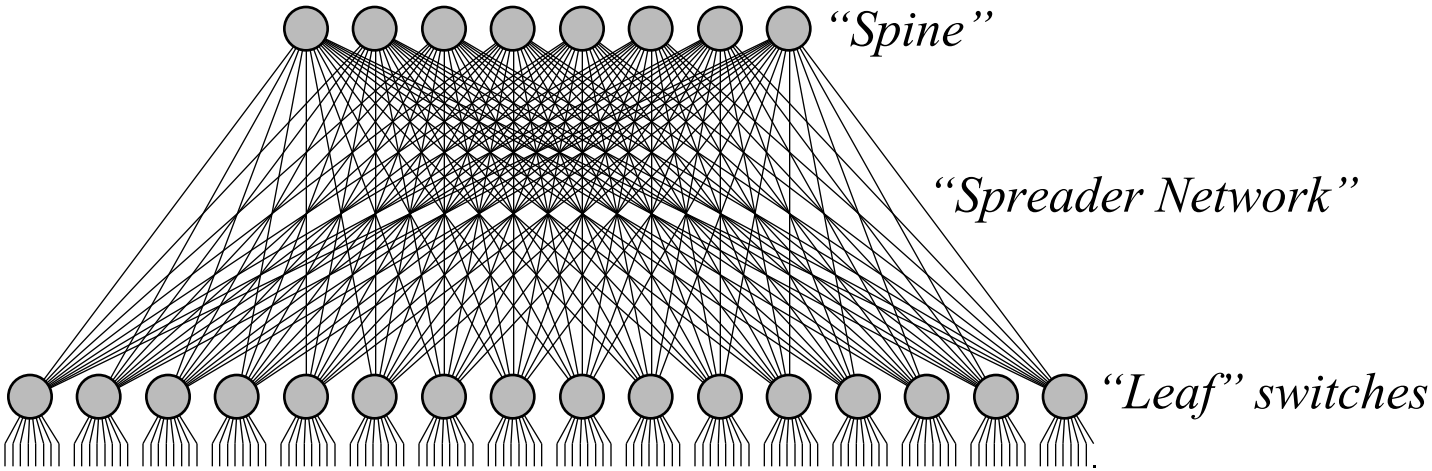
Dispersive Routing is Added with Software

There is nothing deficient about the Myrinet Clos networks used in large clusters. However, from the time when we first started building Clos "Network in a Box" products, it has been part of the plan to add features to the network mapping and routing software to take better advantage of the properties of Clos topology.

These software features will be available within a few months (in GM 2). We expect that they will be used to good advantage, particularly in large clusters. This part is meant to explain:

- Dispersive routing, its rationale, and how it works.
- Other gains that "fall out" from using multi-path routing.

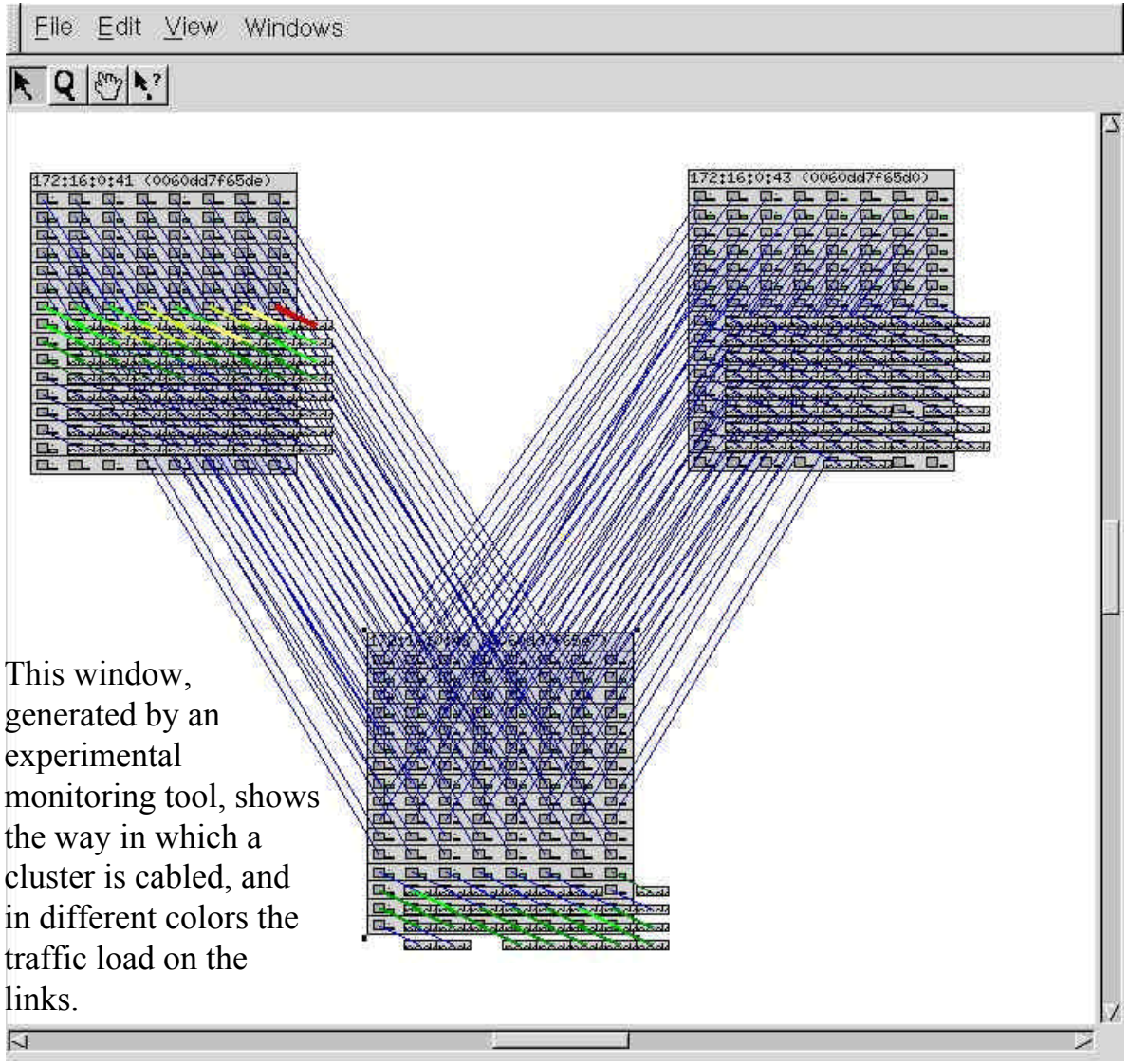
Use Two Additional Properties of the Clos Network



- **There are multiple shortest routes** (8 routes in the example above) **between hosts**.
  - One may or may not route all packets through the spine.
- **All progressive (minimal, shortest) routes are deadlock-free.**
  - Simplify the route computation! (This feature is available today.)
  - Why use the same route for every packet from host A to host B? Why not use different routes for successive packets? (This is dispersive routing.)

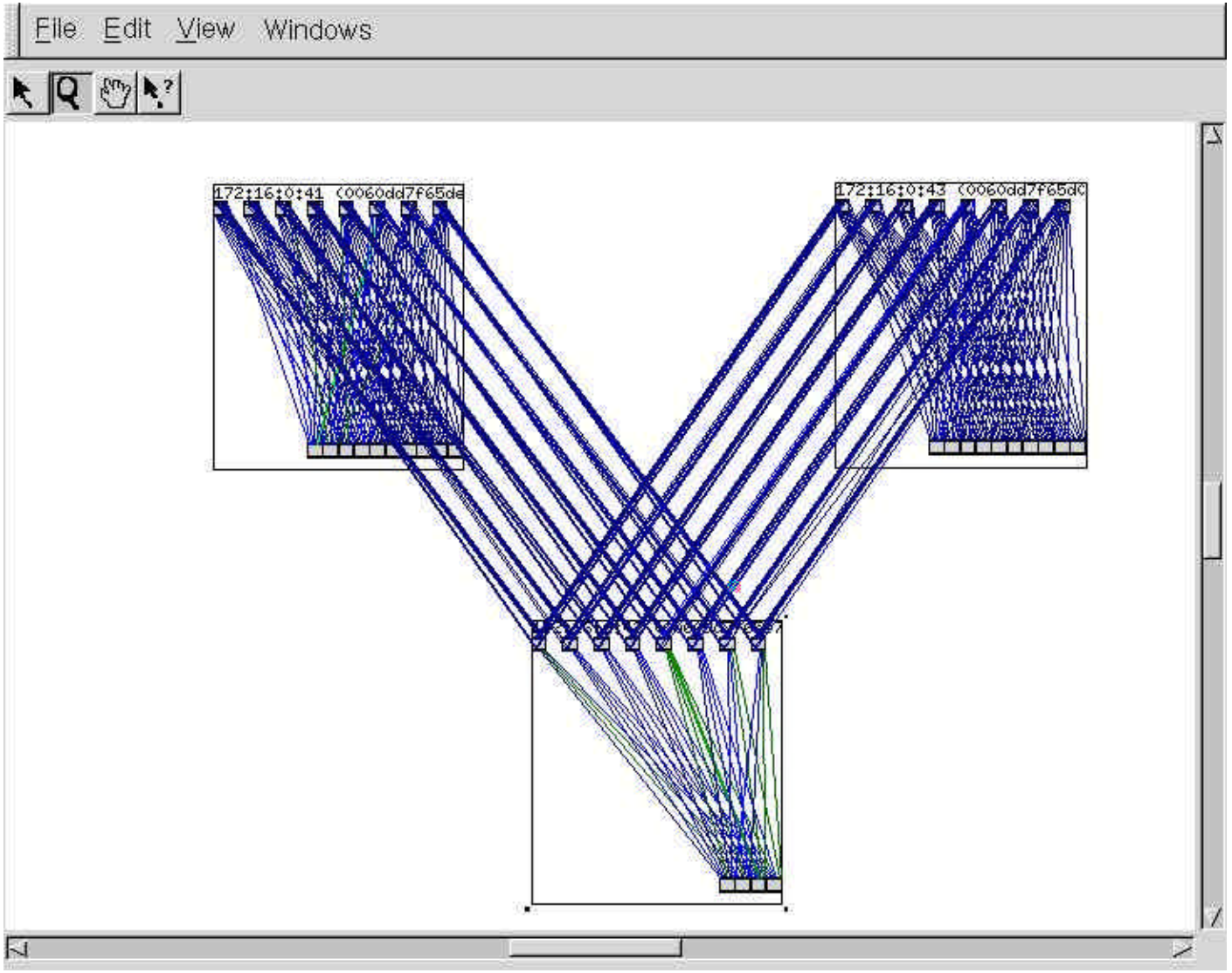
The possibility of network “hot spots”

The way in which the route computation works today is that there is a single route from each host to each other host. The route computation balances these “static” routes so that internal network links carry similar numbers of routes. This approach does not assure that the dynamic traffic loads are balanced.



This window, generated by an experimental monitoring tool, shows the way in which a cluster is cabled, and in different colors the traffic load on the links.

A different view, with just switch-to-switch links

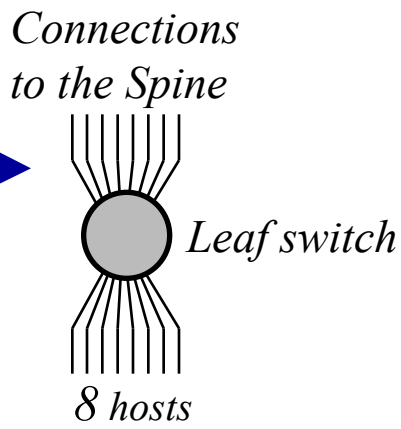


A Constructed "Hot Spot" Example

Suppose that you were doing a classical 3D grid-point computation, in which each of $N = 128$ processes (hosts) communicates with its 6 neighbors on each time step.

There are 8 x 120 = 960 routes distributed evenly across these 8 links

However, only 48 of these routes, an average of 6 per link, are used in this particular computation. What is the variation about this average?



There is a more than even chance for this single switch that some link will carry as many as 8 active routes. Across all 16 such switches, it's likely that some link will carry as many as 10 active routes. The progress of the computation depends on the slowest communication.

The point: It's easy to construct examples of such "hot spots."

How and Why Dispersive Routing Works

- **How it works**: With Myrinet, the mapping from destination host to a (source-determined) route is computed by the control program (*e.g.*, GM) that executes in the Myrinet interface. Suppose that the interface stored multiple routes (*e.g.*, 8 routes, or all routes) for each possible destination host. The control program could then, for each packet, use any route.
  - At random. (There are lots of theoretical papers about this approach.)
  - Cycle between them. (This approach can lead to complex modes of blocking.)
  - Adaptively, according to how quickly reply (ACK) packets have been received on this route. (There may be a Ph.D. thesis on this topic.)
- **Why it works:** The goal is to make the cluster network as "flat" as possible, so that it exhibits no particular locality. By dispersing or randomizing the routing, you give all traffic patterns the statistical characteristics of random traffic, or somewhat better.

Open Questions, and Opportunities

- Suppose that the GM control program fails to receive a reply packet for a packet sent on a particular route. It would be rather dumb for the control program to re-try on this same route.
  - It appears that by using "adaptive" dispersive routing, the Myrinet control program can achieve fault tolerance (preferential use of routes that work well) on the time scale of the reply packets (~1 millisecond) rather than on the time scale of remapping (~1 second).
- It has been shown that adaptive routing in switches has a lot of odd problems due to the limited, local information available to a single switch. Is route adaptivity better?
- Future Myrinet interfaces will have multiple ports. A control

Thanks, Tom! And questions.