ServerNet SAN I/O Architecture

Robert W. Horst David Garcia

Tandem Computers Incorporated
19333 Vallco Parkway
Cupertino, CA 95014
horst_bob@tandem.com
daveg@ntos.tandem.com

Abstract

This paper presents several different ways of designing systems with ServerNet-connected I/O devices. The ServerNet System Area Network is a multistage, wormhole-routed network designed for reliable I/O and inter-processor communications. Designing a SAN to be used both for I/O and IPC has benefits in system cost and flexible expansion, but requires a network well-suited to the task. The SAN must support read operations, and must have other features for reliability, expandability, and fairness. The paper discusses ServerNet-based I/O architectures for fault-tolerant and clustered systems. The clusters can connect I/O subsystems with embedded SAN support, I/O controllers on remote PCI busses, or native peripherals.

1.0 Introduction

The movement from compute-centric to data-centric applications makes the I/O architecture a critical element in system performance. I/O should no longer be considered a "subsystem" with "peripheral" devices and long access times. Instead, data movement through the CPUs and I/O devices must be carefully designed to eliminate unnecessary data movement, to reduce overhead and latency, and to maximize I/O throughput. The increased emphasis on clustered computing also makes it necessary to consider the I/O connectivity in conjunction with the inter-processor communications (IPC). If the I/O and IPC share the same interconnect, the system can be designed with lower cost, improved reliability, and greater I/O scalability. The unification of the interconnects gives flexibility in supporting shared-peripheral or shared-nothing databases. Peripheral sharing allows load balancing and peripheral switching to support fault tolerance. A unified interconnect can support direct device-to-device transfers, eliminating data copies and freeing memory bandwidth in the host CPUs.

The ServerNet system area network (SAN) has been designed to unify the I/O and IPC connectivity. This paper presents several different ways of designing systems with I/O connected through ServerNet. After a brief review of the ServerNet architecture, specific I/O implementations are discussed.

2.0 ServerNet Overview

2.1 ServerNet I

The ServerNet SAN is a scalable network that supports read, write and interrupt semantics similar to previous generation I/O busses. The first generation of ServerNet (previously called TNet), has bi-directional point-to-point 50 MHz byte-serial links [1]. Variable length data packets include a header, up to 64 bytes of data, and a 32-bit CRC. The header includes 20-bit source and destination IDs to address end nodes, plus a 32-bit address offset to address memory within each node. The network can be expanded by cascading multiple levels of 6-port routers in a number of possible network topologies. Each router is a single ASIC holding input queues, a routing table, and crossbar switch. The network uses wormhole routing, and each additional stage of routing adds approximately 300 ns of latency.

ServerNet supports optional fault tolerance through a pair of independent "X" and "Y" fabrics. Dual-ported processor or I/O interfaces maintain connectivity to the rest of the system despite loss of one of the fabrics. Self-checking hardware in the interfaces and routers assure that faults are detected and isolated quickly. The network interfaces are carefully designed to prevent a slow or malfunctioning node from causing both fabrics to back up with traffic. This isolates problems to a single node and stops them from propagating to other nodes or causing network deadlocks that could take down the entire cluster.

The network hardware provides delivery and ordering guarantees to allow light-weight messaging and I/O protocols. The guarantees obviate the need for software to perform checksums or to reorder portions of a message that may have arrived out-of-order. For standard read and write transactions, every request packet has a hardware-generated response packet. If the time between request and response exceeds a specified value, a time-out counter generates an interrupt, and the software can take recovery action knowing how much has been transferred. Interrupts from devices are sent as standard write packets to special address ranges causing them to be queued in memory. In-order delivery assures that the I/O completion interrupts are never delivered before the data transfer is complete.

Several different ASICs have been designed to interface between the ServerNet fabrics and processor or peripheral busses. In Tandem fault-tolerant systems, processor-interface ASICs connect ServerNet directly to the memory bus. For PC server interfaces, other processor-interface ASICs connect ServerNet to host CPUs through the PCI bus. In I/O subsystems, bus-interface ASICs allow remote I/O busses to communicate with host memory through ServerNet.

2.2 ServerNet II

The second generation of the ServerNet SAN takes advantage of the speed and density of new ASIC technology to add several new features [2]. The primary motivation to evolve the network was to increase the peak throughput from 50 MB/s to 125 MB/s per link. The size of end-node packet buffers has also been increased from the 64-bytes allowed in ServerNet I to 1024 bytes in ServerNet II. A larger packet size allows header overhead to be amortized over more data, and improves the efficiency when interfacing to high-overhead memory busses (especially PCI). Other key features of ServerNet II include the addition of 64-bit ServerNet addresses, the ability to configure “fat pipes” to increase bandwidth beyond the capability of a single 125 MB/s link, and support for the emerging Virtual Interface Architecture standard.

ServerNet II physical links are designed to directly drive standard 8b/10b serializer/deserializer components to interface to standard 1000BaseX cabling. The following table summarizes the physical interfaces available with ServerNet II components:

Table 1: ServerNet Physical Interface

Physical Interface	ServerNet	
	I	II
Direct from the ASIC 9b@50MB/s	✔	✔
Direct from the ASIC 8b/10b @ 125MB/s.		✔
Parallel cable with LVDS signaling.	✔	
Parallel cable with differential ECL signaling[a].	✔	
Copper Serial (1000BaseCX).	✔	✔
Short haul optical (850nm laser on multi mode fiber, 1000BaseSX).	✔	✔
Long haul optical (1300nm laser on single mode fiber, 1000BaseLX).	✔	✔

a. Only Tandem NSK and Integrity Unix products use parallel ECL ServerNet cables.

3.0 I/O requirements

3.1 Pull (read) capability

The requirements for I/O connectivity are somewhat different than IPC requirements, making it unlikely that a technology developed only for IPC will be a good match for I/O. In particular, most IPC networks, such as EtherNet and ATM, include support for pushing data through the network, but not for pulling data. Even Fibre Channel, a network designed for I/O, has omitted the pull capability and instead focussed on long one-way transfers. Without support for pulling data (read transactions), reads sent to a remote node require CPU cycles to be consumed at the remote node.

The pull capability is critical for I/O expansion because it allows outbound I/O traffic to be controlled and paced by the DMA controller on the device interface rather than by multiple DMA engines in the host. Without this capability, the I/O concurrency is limited by the number of host engines and the software’s ability to effectively schedule the use of the engines.

If the network interface does not provide pull capability, it could be provided by emulation with a processor at each I/O node. Although most new peripheral interface chips have some kind of embedded processor, they typically do not have sufficient extra capacity or memory to run a complete communications protocol. Adding another processor between the interface chip and the network may be reasonable in relatively expensive controllers, such as RAID controllers, but is too costly for simpler devices such as WAN controllers or individual disk drives.

The other problem with relying on an additional I/O processor to emulate reads is that it injects a relatively slow software execution time into a fast path that could otherwise be handled directly by the hardware. For instance, network-attached DRAM may be useful for increasing the amount of sub-microsecond memory in a cluster, but would have little utility if a processor is required for all read operations. As the speed of storage and network connections increase, this processor penalty will be increasingly unacceptable.

3.2 Reliability

For I/O attachment, it is important to keep the packet error rate very low. I/O device drivers typically assume a reliable connection between the host and the I/O device and do not have the ability to cope with the error frequency of less reliable media. Networks such as ATM and Ethernet are not usually considered for I/O because network congestion can cause packets to be dropped and retried at some indeterminate later time. Running TCP/IP to handle the communications of an unreliable network results in unacceptable processor overhead. A reliable SAN with a low overhead protocol is necessary for I/O attachment lest the CPU be heavily utilized by the communications protocols required for talking to the IO adapters.

3.3 Expandability

Large servers require I/O expansion and the ability to add devices on-line. Expansion is typically very limited in bus-based I/O systems due to loading restrictions that limit the number of controllers on a single segment. Bridges, such as the ones commonly found in PCI-based systems, have proved to be rather unsatisfactory in performance, cost, and amount of expansion. Using a network assures that expansion limits are not defined by arbitrary packaging constraints. It is also much easier to make a network hot-pluggable to allow on-line upgrades and repairs.

3.4 Fairness

By using a combined network for IPC and I/O, the same hardware and software components can be used to expand either the processing or I/O as needed. But in order to get the benefits of using the same network for both, the network must assure fairness to prevent nodes with long messages from starving other nodes that require frequent short messages. It is also important to have distributed network arbitration algorithms that can guarantee service to a node no matter where it may reside in the interconnection hierarchy.

4.0 ServerNet I/O in Tandem systems

The expansion limits of bus-based I/O systems and the limitations of existing networks were the impetus to develop ServerNet. This solved the I/O expansion problem, and at the same time, unified the I/O connectivity with IPC. ServerNet-based I/O is now incorporated throughout Tandem's product families. Following are descriptions of the I/O connectivity in Tandem's fault-tolerant products, as well as a discussion of the use of ServerNet in open NT clusters.

4.1 Integrity S4000

The first commercial system to use ServerNet for I/O is the Tandem Integrity S4000 that began shipping in early 1996 [3]. The S4000 is a Unix-based one to four-way SMP with all I/O devices connected through ServerNet. Hardware fault-tolerance is accomplished by implementing each processor with two pairs of microprocessors and two main memories. During normal operation, all four processors run the same instruction stream. Each pair of processors has a copy of main memory. The two memories are maintained with identical data using a special mode of the ServerNet router that replicates inbound I/O packets and delivers them to both memories at the same instruction time. If a microprocessor fails, it does not produce the same outputs as its mate, and that pair is taken offline. The remaining pair continues to operate with no interruption of service.

Figure 1 shows the configuration of routers in a two-way fault-tolerant configuration. I/O expansion routers are integrated into the Standard System Controller card that supplies ServerNet connections to the internal SCSI/LAN controller, as well as to external cables for I/O expansion. A second board holds the routers used for CPU expansion.

The initial I/O controllers for the S4000 use an ASIC called the Icepack (IP). The IP has packetizers and depacketizers to interface between a ServerNet port, and a standard 32-bit bus such as a 68000-based bus or VME bus. The IP accepts read or write transactions on the 32-bit bus, turns them into remote ServerNet transactions, and then either reads or writes remote memory. The 32-bit bus does not support split transactions, hence read transactions hold the bus until the remote data has been returned. This does not limit the maximum I/O, because each ServerNet end point has its own IP, hence the I/O devices are served by many independent 32-bit busses.

The IP supports a programming model very similar to that of bus-based systems. Drivers are similar to standard I/O drivers with some changes for switching paths on errors, and to optimize performance given the longer latencies. The simple programming model comes at some cost in performance. For outbound I/O, each read on the 32-bit bus must travel all the way to the host memory and back before the next read can be issued. This can limit outbound I/O throughput for IP-based I/O devices. Future I/O bus interface ASICs will use prefetching to reduce this performance penalty.

The I/O performance in the S4000 has shown significant improvement over previous generations of the Integrity product line. The I/O expansion limits of previ-

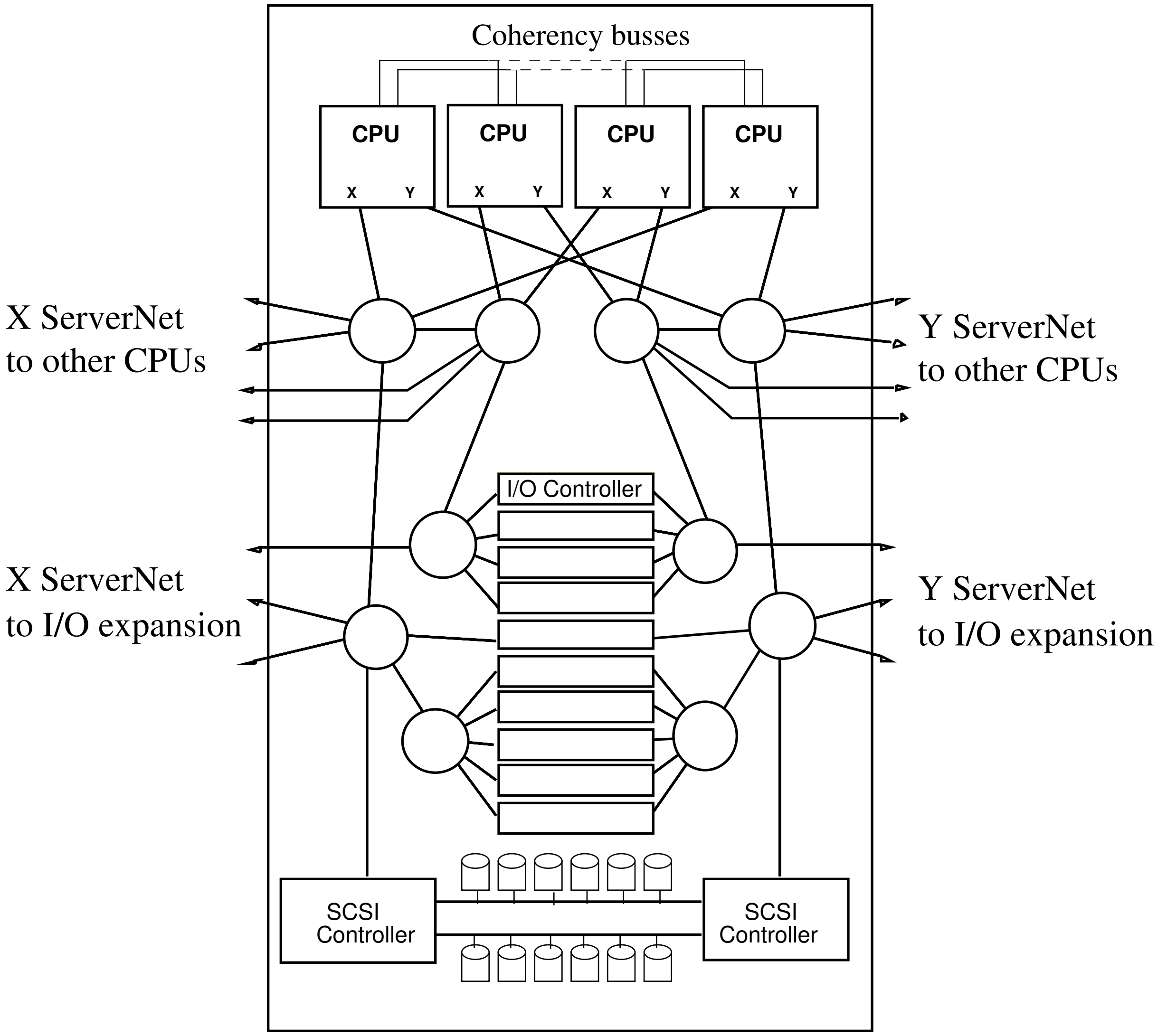

Figure 1. Integrity S4000 block diagram

ous generations have been eliminated by the ability to add devices through the expansion links.

4.2 Himalaya S7000

The Himalaya S7000, which began shipments in early 1997, is the first ServerNet-based machine in the Himalaya line of software fault-tolerant systems running the Tandem NonStop Kernel operating system. The message-based OS implements fault tolerance by sending checkpoint messages between process pairs, and by using transaction logging for the distributed NonStop SQL database. The hardware is fully self-checked for fail-fast operation. The power, cooling, and packaging have the required redundancy to allow fault-tolerant operation and on-line serviceFigure 2 shows a diagram of a 16-CPU S7000 system. The diagram shows only the "X" half of the ServerNet router fabric; the identical Y fabric is not shown. The center of the diagram shows a set of four fully-connected routers forming a tetrahedron. The tetrahedron has been shown to give good bisection bandwidth, low link contention and deadlock-free routing [4]. Connected to the tetrahedron is an unbalanced tree of routers, with topology dictated primarily by packaging constraints. By adding ServerNet expansion boards, the number of I/O or CPU nodes can be expanded. The true limits on expansion are not set by ServerNet's limit of 20-bit node IDs, but are set more by the ability to manage and afford massive configurations.

The S7000 system uses ServerNet for all I/O connections as well as for the inter-processor communications. ServerNet-based I/O slots in the S7000 can use the same controllers and ServerNet expansion cards as the Integrity S4000. The system is partitioned with a backplane

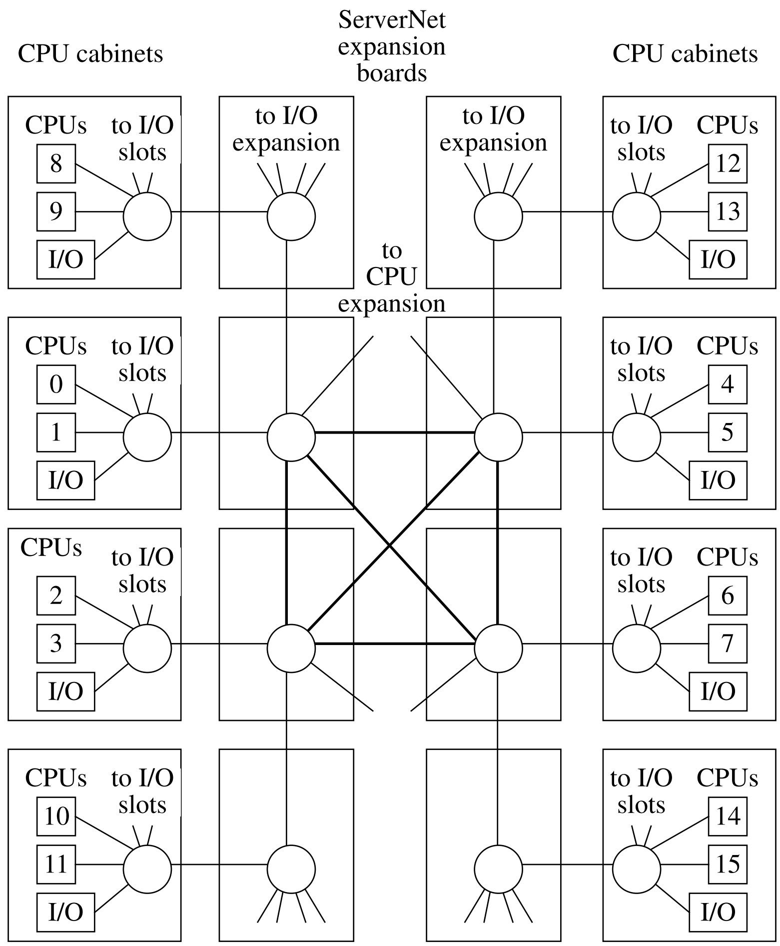

Figure 2. Himalaya S7000 block diagram. Only "X" fabric shown.

connecting two CPU modules, internal I/O devices, two dual-ported ServerNet I/O slots, and two (X and Y) ServerNet expansion slots. The internal I/O has support for two SCSI strings of disk and an Ethernet port.

I/O controllers for the S7000 use either the IP chip, or another ASIC, called Tioga. Tioga is used on intelligent controllers for ATM and Ethernet, allowing them to multiplex LAN traffic and deliver it directly to the CPU that is consuming or producing the traffic. This allows a small number of LAN controllers to service the entire cluster, rather than requiring one per CPU. The number of controllers required is governed by the traffic requirements instead of the number of CPUs in the cluster.

I/O in the Tioga-based LAN controllers is managed by an I/O processor on the board. The LAN chip transfers directly into a buffer memory on the controller, and the I/O processor interprets the LAN headers and programs a DMA engine in Tioga to carry out the transfer between the buffer and main memory.

Early experience with S7000 systems shows significant improvement in both I/O and IPC performance compared with earlier Himalaya systems. The architecture also has encouraged several new software development projects that will improve future performance by taking advantage of the richer connectivity available in the S7000.

5.0 I/O in NT clusters

Many of the same techniques and components used to support I/O in Tandem's fault-tolerant systems are being moved to clusters of servers interconnected by ServerNet and running the Microsoft NT operating system. The following sections give three different techniques for connecting I/O devices to these clusters using embedded ServerNet subsystems, remote PCI, and native interfaces.

5.1 Overview of NT clustering

ServerNet technology can be exploited in a number of ways for building expandable, reliable NT clusters. Systems can be configured as two-node failover clusters running Microsoft's Wolfpack clustering software [5], or as multinode systems running a distributed database. ServerNet PCI adapter cards provide dual ServerNet links between nodes for fast, reliable message passing. Beyond two nodes, systems can be expanded through 6-port ServerNet switches.*

The PCI adapter cards are based on the SAIL ASIC that was designed specifically for interfacing PC servers to ServerNet. SAIL includes a pair of X and Y ServerNet interfaces, a Block Transfer Engine (BTE) for programming DMA transfers, Address Validation and Translation logic (AVT) for translating incoming ServerNet addresses into the endnode's physical addresses, and interrupt hardware for queuing interrupt packets in main memory. SAIL is carefully designed with selfchecking logic and backpressure timers to assure that a misbehaving node cannot cause both X and Y fabrics to back up with traffic. This ensures that no single point of failure will disrupt the cluster's communication over ServerNet.

Figure 3 illustrates several features of multinode PC clusters. Each PC server has a ServerNet PCI adapter card with a SAIL ASIC providing two ServerNet ports to each server. Separate X and Y switches allow for CPU expansion, I/O device expansion or connections to other switches for expanding the ServerNet fabric. Both X and Y fabrics are used for added bandwidth in normal operation, but one fabric is sufficient to provide connectivity while the other fabric is serviced.

The figure also illustrates the three types of I/O connections to the ServerNet fabric. I/O subsystems can be designed with a) embedded ServerNet support, b) I/O controllers on remote PCI busses, or c) native ServerNet I/O devices. The pros and cons of the different approaches are discussed in the following sections.

5.2 I/O subsystems with embedded ServerNet ports

As NT clusters begin to run large enterprise applications, there will be a need to interface large I/O subsystems to the cluster to give direct access to all the CPUs. These subsystems, such as RAID storage arrays and communications front ends, can incorporate ServerNet ASICs in place of the SCSI or other connections previously used for interfacing to individual hosts.

Figure 3a illustrates a subsystem designed with an internal ServerNet connection. One characteristic of this type of subsystem is that there is usually an embedded microprocessor and buffer storage managed by the processor. In such an application, the best approach for ServerNet attachment is to use an ASIC, such as SAIL, with a Block Transfer Engine (BTE) that can be programmed by the remote processor to set up DMA transfers between the host memory and the remote buffer memory. In some sense, this model treats the subsystem as another CPU in the cluster. The hardware and messaging software to communicate with the remote processor may be similar to that used to communicate to any other CPU in the cluster.

* In this paper the term "router" is used to describe the 6-port ServerNet ASIC with built-in routing table, while "switch" refers to the assembly that packages one or more routers along with power, cable drivers and initialization hardware.

This type of interface has the following characteristics:

- It works best in applications where the cost of a remote processor and memory has already been justified and need not be added just to program the BTE.
- The application requires or can tolerate a store-and-forward step between main memory and the peripheral. (For instance, RAID parity calculation or communication header interpretation.)
- It is acceptable to modify the subsystem code with changes to program the BTE.
- The remote microprocessor bus is accessible or can be redesigned to incorporate the ServerNet ASIC.

5.3 I/O through remote PCI busses

A second alternative for ServerNet I/O is to use the bridging capabilities of a ServerNet bus interface ASIC, such as the IP, to drive a remote PCI bus (Figure 3b). This removes host restrictions on the number of PCI devices, and on the maximum length of a PCI bus. ServerNet I/O expansion can be used to support many remote PCI busses at very long distances (30 meters today, kilometers soon). Standard PCI controllers can be used because the bus interface ASICs translate the PCI read and write transactions into ServerNet read and write transactions that are satisfied remotely at host memory. There is no store-and-forward step, and transfers proceed with only a small amount of added latency for each transactions. Preliminary measurements of one SCSI controller operating in this fashion shows less than 15% throughput degradation for a controller connected through ServerNet versus one connected directly on the host PCI.

Software drivers for the remote devices normally require small modifications in the way resources are registered and in the memory maps. Modifying the code for future controllers can be fairly mechanical and based on the drivers that have already been modified. More extensive modifications are required to convert a single-host device driver to allow the devices to be accessible from multiple hosts simultaneously. In some cases, this also involves modification of the firmware of the device itself. The first multi-host adapters are now being devel-

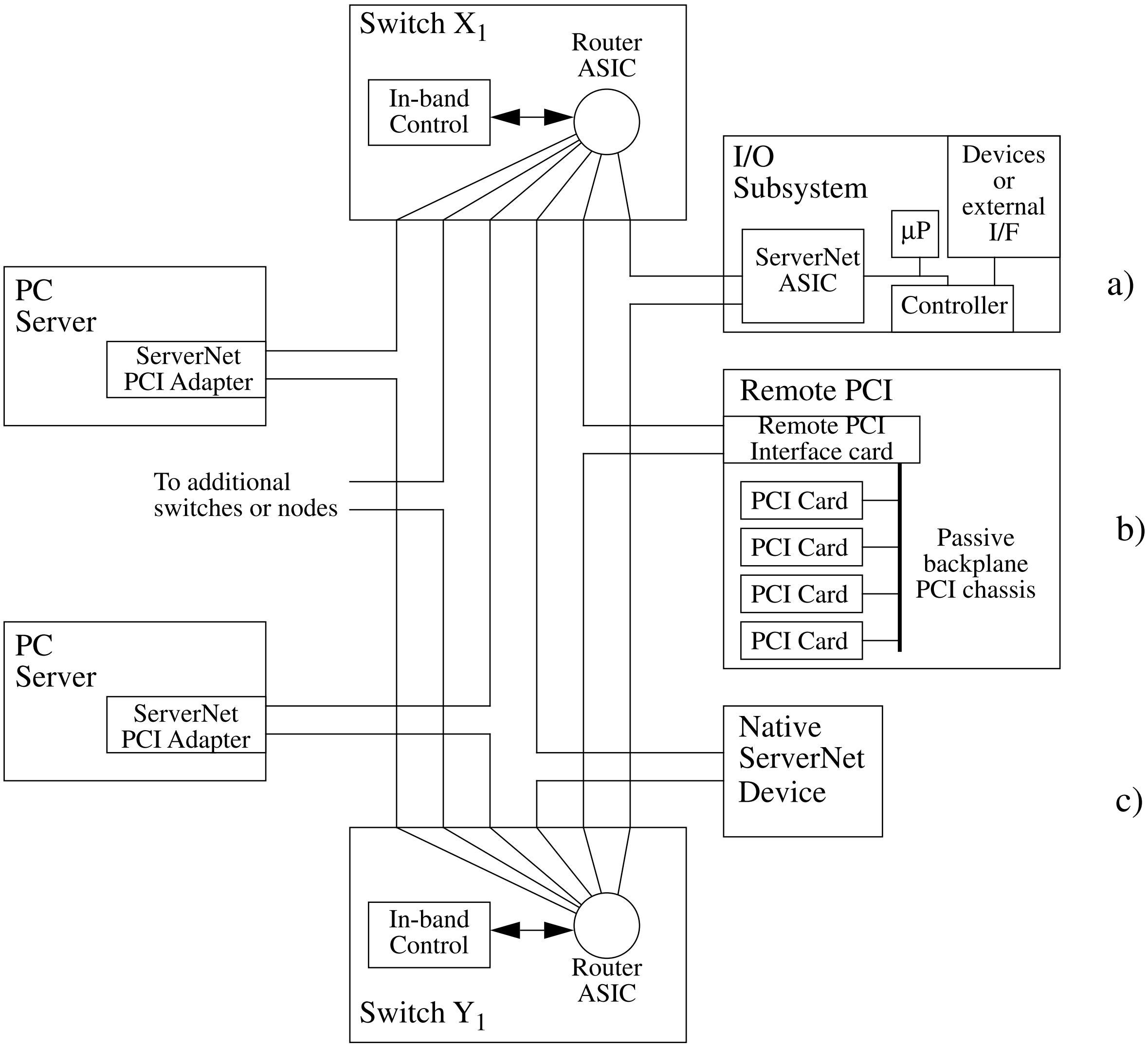

Figure 3. Cluster of PC servers with three different types of ServerNet-connected I/O devices.

oped for the remote PCI prototypes. Various different types of controllers may be supported through remote PCI, and may include SCSI, RAID, fibre channel, communications controllers, and data encryption devices.

The packaging of the remote PCI can be done in many different ways, but the first implementation places the ServerNet to PCI interface on a card in a standard passive backplane PCI cabinet. This interface card is inserted in the slot normally used for the CPU card in a standard cabinet. Many vendors make backplanes and cabinets that conform to this standard, assuring the availability of many different types of configurations. Fault tolerance can be provided by connecting both X and Y links to the remote cabinet to tolerate faults in one of the fabrics, and by providing redundant peripherals in separate cabinets to tolerate faults in the controllers or power to the I/O cabinet.

The remote PCI option has the following characteristics:

- Gives the fastest time to market through the use of standard PCI controller hardware and minimal changes to drivers.
- Provides good performance by allowing direct device to memory access without extra data copies.
- Can provide a very large number of PCI slots for I/O expansion without changes to the server PC.
- Can be used to extend the distance to I/O devices far beyond the usual PCI limits.

5.4 I/O to native ServerNet I/O devices

The ultimate way to attach I/O is to build the ServerNet ASICs or interface libraries directly into the peripheral devices, as shown in Figure 3c. For instance, if a disk is connected directly to ServerNet, the ServerNet logic replaces the SCSI or other interface logic at approximately the same cost. At the host end, if the cluster is already ServerNet-connected, the SCSI interface cards are eliminated without requiring any additional ServerNet cards, for a net cost savings. Performance can also be improved by eliminating some of the software in the SCSI driver, and by the added bandwidth that can be provided. The ServerNet-attached device gains the ability to be shared by all of the hosts, and ability to tolerate fabric failures due to the dual X and Y connections.

This architecture also gives many new possibilities for future systems to make use of the peripheral-peripheral capabilities allowed by ServerNet. There are clearly many technical benefits to this approach if the industry can be persuaded to build SAN-attached devices.

By connecting the storage devices directly to the SAN, it is possible to move major portions of the files system all the way to the disks. This architecture takes advantage of the processing power and buffering that are currently under-utilized on the drives. If the disks understand the types of data objects being stored, they can optimize the placement of data on the disks in ways not possible with current block-mode access. Although many open questions remain regarding the programming and security models [6], this is a promising path for improving the performance and lowering costs of distributed file systems.

In summary, native ServerNet attachment gives the following attributes:

- Gives the lowest cost and highest performance peripheral attachment through ServerNet.
- Requires high enough volumes in the cluster market to justify a new interfaces standard.
- Allows the most innovation in over all system architecture by eliminating intermediate hardware and protocols and by allowing device-to-device transfers.

6.0 Conclusions:

This paper presents several different ways of using the ServerNet System Area Network for attaching I/O devices. The production S4000 and S7000 system have proven the viability of unifying the interprocessor and I/O attachment through a multistage switching fabric. Three alternatives have been discussed for bringing this capability to PC clusters.

The unification of IPC and I/O will give new capabilities for scaling, distributing, and sharing I/O among many host processors. The difficulty in the past has been that no network had the right characteristics to serve this need. LAN networks such as EtherNet and ATM lack the reliability, and low latency needed for low-level I/O transfers. Fibre channel has been optimized for long transfers and has unacceptably long latencies for short IPC messages. Fibre channel and LANs also lack the ability to pull data with read operations, making them impossible to use for efficient remote PCI bridging. Other interconnect networks do not have the scalability or reliability required for the I/O systems of enterprise servers.

System Area Networks that are designed from the outset to serve both I/O and IPC can avoid the shortcomings of networks optimized for one function. The unified architecture presents new opportunities for innovation in moving functionality from host processors into the I/O network.

7.0 References

[1] R. W. Horst, “TNet: A Reliable System Area Network,” *IEEE Micro*, vol. 15, no. 1, pp. 37-45, February 1994.

[2] D. Garcia, and W. Watson, “ServerNet II,” *Proc. Parallel Computer Routing & Communication Workshop*, June 1997.

[3] W. E. Baker, et al., “A Flexible ServerNet-based Fault-Tolerant Architecture,” in *Proc. 25th Int. Symp. Fault-Tolerant Computing*, Pasadena, CA, June 27-30 1995.

[4] R. W. Horst, “ServerNet Deadlock Avoidance and Fractahedral Topologies,” in *Proc. 10th Int'l Parallel Processing Symposium*, Honolulu, Hawaii, pp. 274-280, 1995.

[5] R. Short, et al., “Windows NT Clusters for Availability and Scalability,” in *Proc. Compcon*, February, 1997.

[6] G. Gibson, et al., “A Case for Network-Attached Secure Disks,” *CMU-CS-96-142*, Carnegie Mellon University, June 17, 1996.