



High Performance Cluster Computing Architectures and Systems

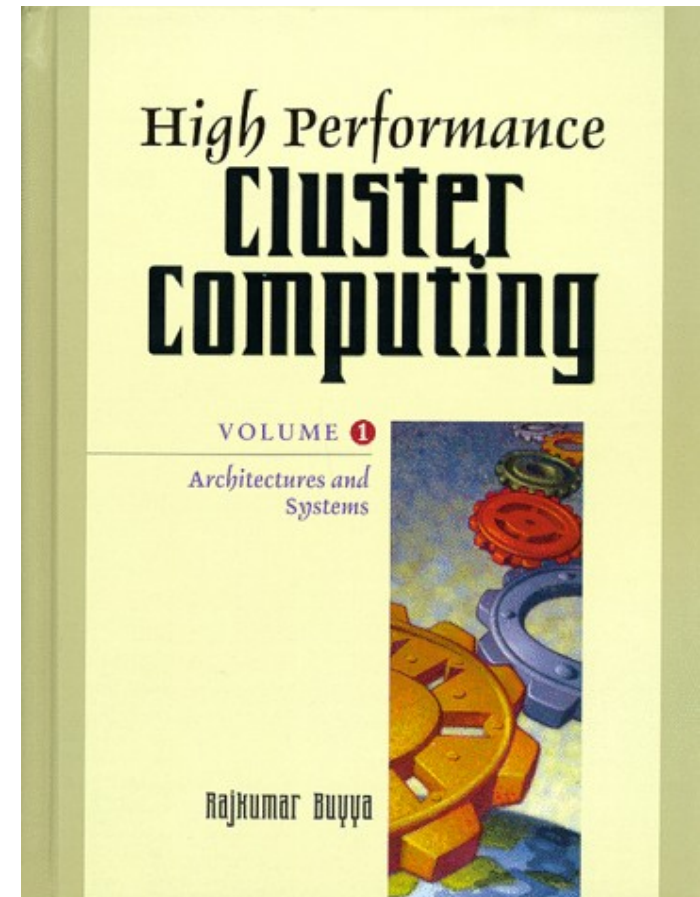
Hai Jin

Internet and Cluster Computing Center

Huazhong University of Science & Technology

High Speed Networks

- Introduction
- Design Issues
- Fast Ethernet
- High Performance Parallel Interface (HiPPI)
- Asynchronous Transfer Mode (ATM)
- Scalable Coherent Interface (SCI)
- ServerNet
- Myrinet
- Memory Channel
- Synfinity



Introduction

- Network is the most critical part of a cluster
- Its capabilities and performance directly influences the applicability of the whole system for HPC
- Starting from Local/Wide Area Networks (LAN/WAN) like Fast Ethernet and ATM, to System Area Networks(SAN) like Myrinet and Memory Channel

Choice of High Speed Networks (I)

■ Fast Ethernet

- 100 Mbps
- CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

■ HiPPI (High Performance Parallel Interface)

- copper-based, 800/1600 Mbps over 32/64 bit lines
- point-to-point channel

■ ATM (Asynchronous Transfer Mode)

- connection-oriented packet switching
- fixed length (53 bytes cell)
- suitable for WAN

■ SCI (Scalable Coherent Interface)

- IEEE standard 1596, hardware DSM support

Choice of High Speed Networks (II)

- **ServerNet**
 - 1 Gbps
 - originally, interconnection for high bandwidth I/O
- **Myrinet**
 - programmable microcontroller
 - 1.28 Gbps
- **Memory Channel**
 - 800 Mbps
 - virtual shared memory
 - strict message ordering
- **Synfinity**
 - 12.8 Gbps
 - hardware support for message passing, shared memory and synchronization

Evolution in Interconnect Trends

- Computational capacity vs. network bandwidth
- Computing power *was* the bottleneck in the past. But the communication *is* the bottleneck
- Simplicity, bus based -> complicated technology, switch based
- Ethernet
 - popular, easy to accommodate new technology
 - distance, speed, limited bandwidth
- High speed networking
 - Giga bps, distance (optical media) etc...

Design Issues

- Goals

- price/performance trade off

- General Architecture

- low design effort, free from processor

- Design Details

- simple, fast, pipelining
 - low start-up latencies, good overall throughput

Goals (I)

■ Price vs. Performance

- production volume, expensive physical layer, amount of storage
- Fast Ethernet(\$50-100) vs. Myrinet or ServerNet (\$1000 or more)

■ Scalability

- fixed topology vs. dynamic topology, shared media vs. private media
- traditionally fixed network topology (mesh, hypercube)
- **clusters are more dynamic**
- network can tolerate the increased load and deliver nearly the same bandwidth latency
- can afford arbitrary number of nodes

Goals(2)

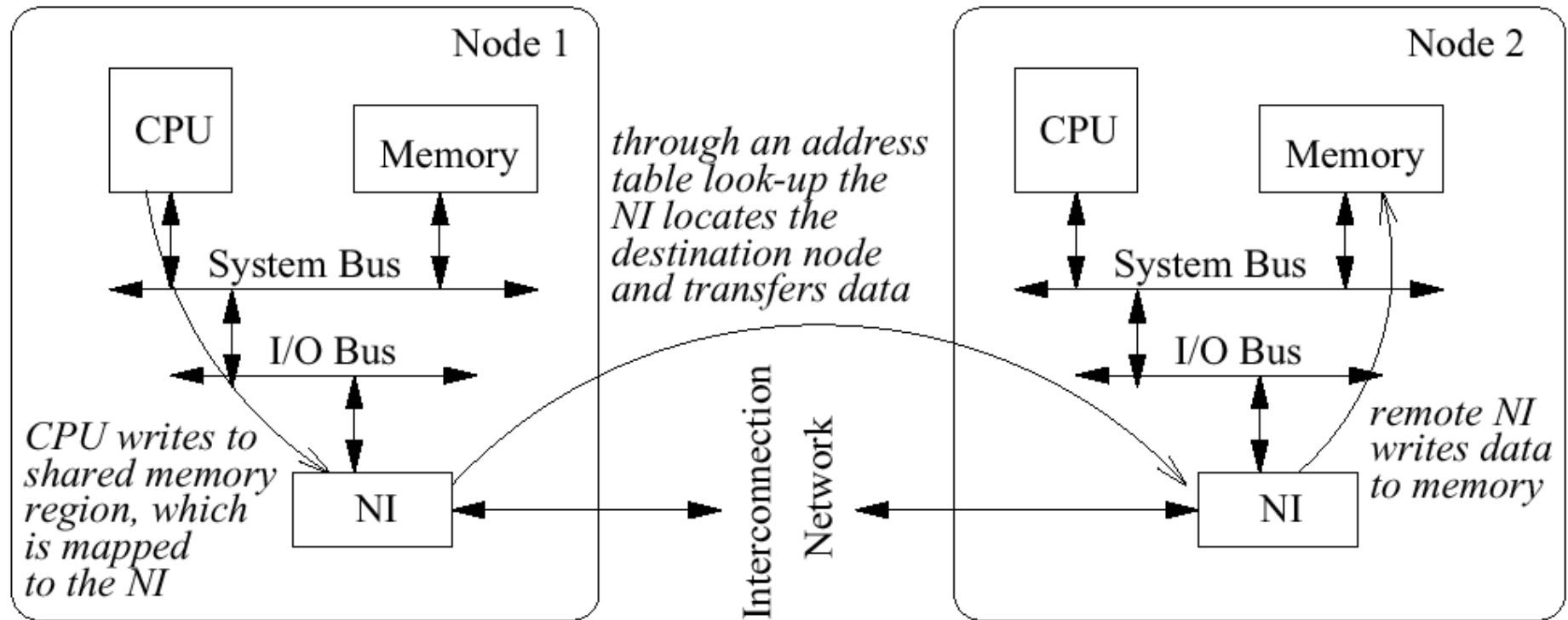
■ Reliability

- CRC check level/provider, buffering storage for retransmission, protocol complexity
- two classes of parallel computer
 - scientific and business computing
- can operate itself without software overhead
 - error freed physical layer
 - CRC can be computed by NI itself
 - error signaling (interrupt or status registers)
 - NI side buffer

General Architecture (I)

- Shared Memory vs. Distributed Memory
 - convenience vs. cost
 - shared memory model - transparency
 - DEC's Memory channel and SCI
 - virtual memory management
 - cache coherent (write and invalidate) -> overhead
 - small scale: manageable
 - distributed memory model
 - message passing
 - send / receive API (explicit)

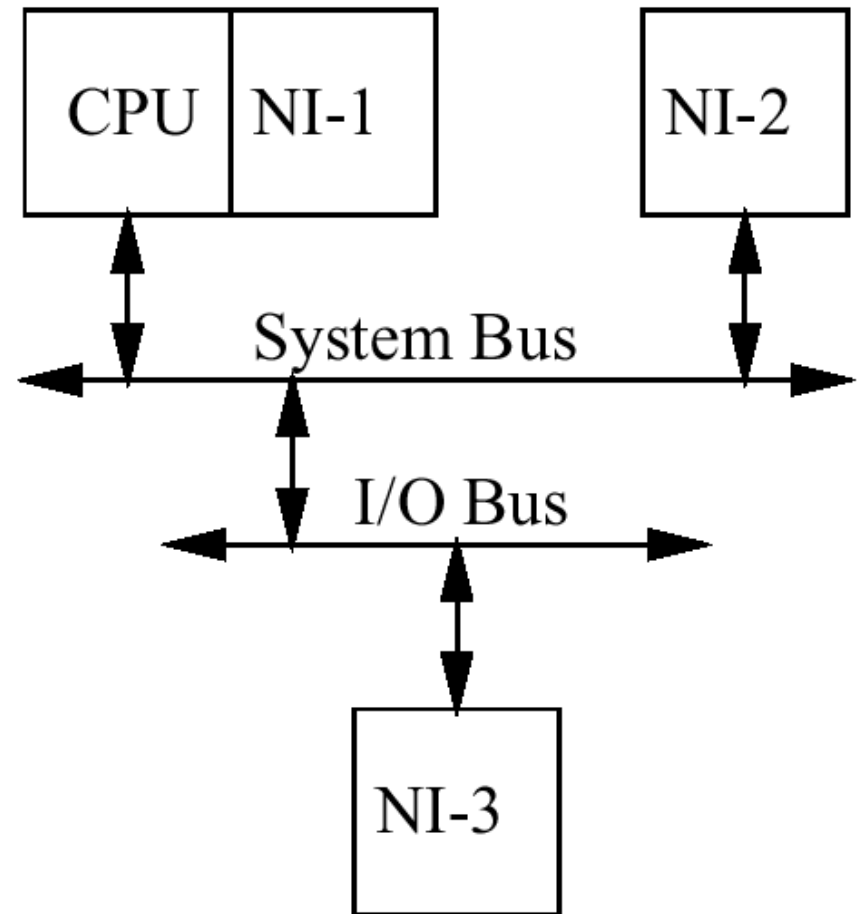
Write Operation to Remote Memory



General Architecture (II)

■ NI location

- Critical to performance and usability
- NI1
 - transputer, most implemented at the prototype phase
- NI2
 - best place for NI, but proprietary system buses
- NI3
 - most common today, no way to support cache coherence



General Architecture (III)

■ NI-1

- instruction set (special communication registers)
- Transputer from INMOS
- iWrap, related systolic architecture
- not successful (too small market)

■ NI-2

- ideal (case of high performance bus)
- system bus based NI
- poll on cache-coherent NI registers
- DMA can read/write from/to main memory using burst cycle
- NI implementation only

General Architecture (IV)

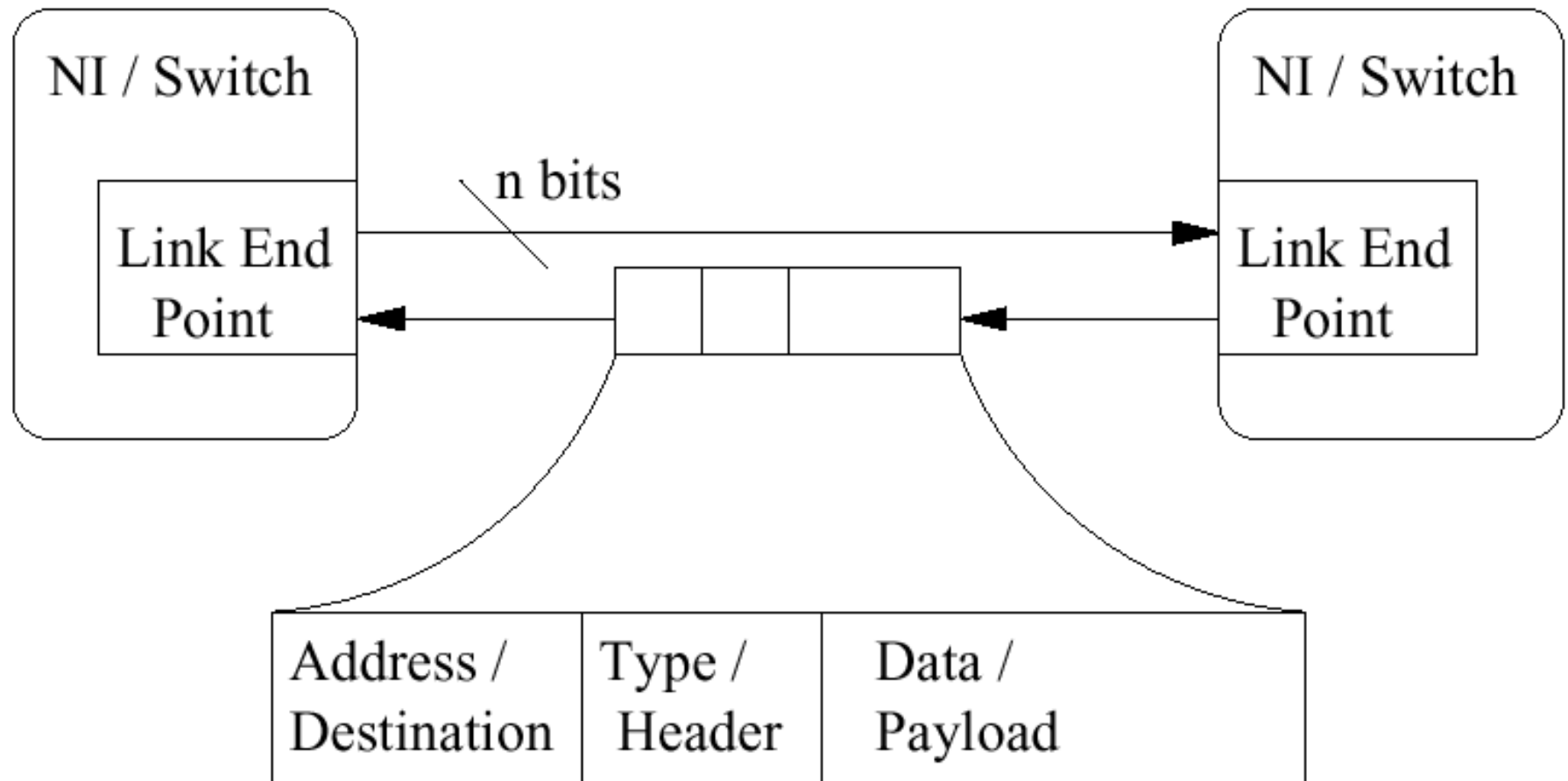
■ NI-3

- PCI-based NI
- at any system w/ PCI I/O bus
- current PCI 32bit/33 MHz - potential bottleneck
- future 64bit/ 66MHz
- disadvantage of the I/O bus location is the loss of some properties such as cache coherence

Design Details (I)

- Link Protocol
 - used for layout of messages
 - detect start/end
 - signal event

A Bidirectional Link and the General Message Format



Design Details (II)

■ Physical Layer

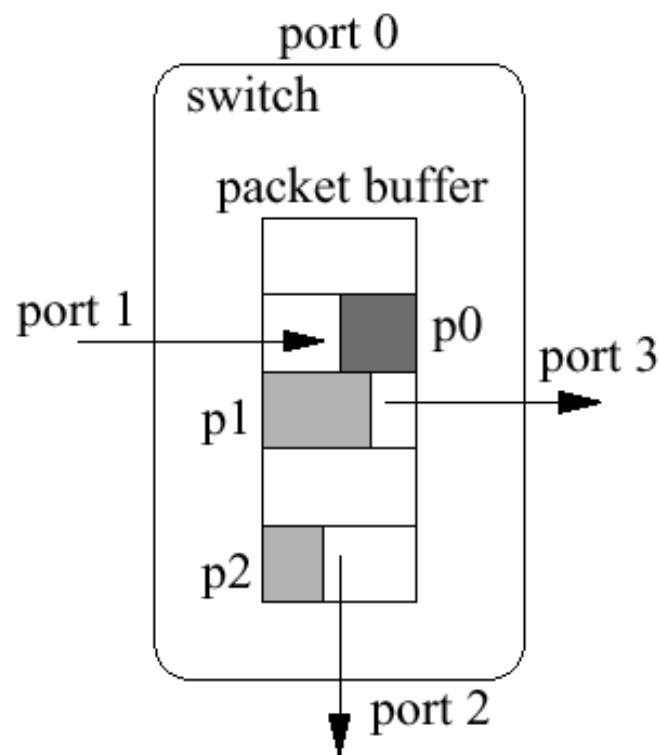
- choose physical medium (data rate, cost)
- serial medium - adequate standard link level transport layer (ex. Gbps Ethernet)
- 64 bit wide cable (ex. HiPPI)
 - high data rate
 - pin count is limitation for the implementation
 - 8 x 8 unidirectional switch with 32 bit signal lines
=> 1024 pins for the link (Too Much)
- byte-wide link (ex. Myrinet or ServerNet)
 - good compromise
 - moderate size of switches and moderate speed
- optical layers will replace copper-base layers

Design Details (III)

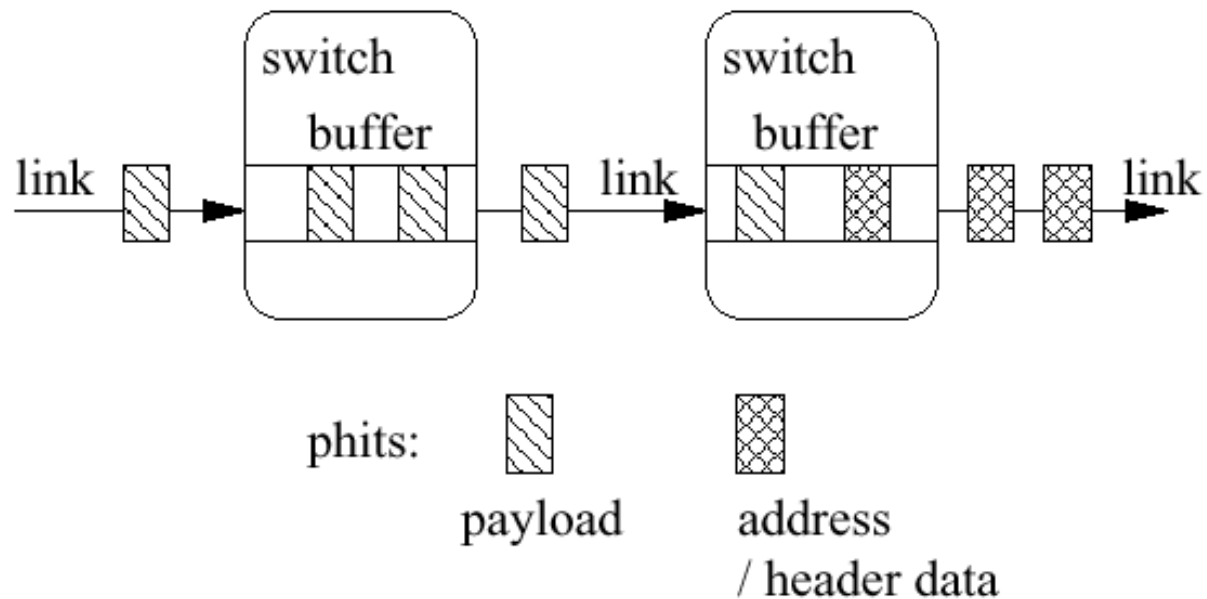
■ Switching

- two main technique
 - packet switching - traditional
 - store/forward operation => limit packet's characteristic
 - wormhole switching - Myrinet
 - forward if header is decoded
 - low latency and small buffer size
 - message size can be various
 - **but error correction is difficult**

Switching Techniques



packet switching



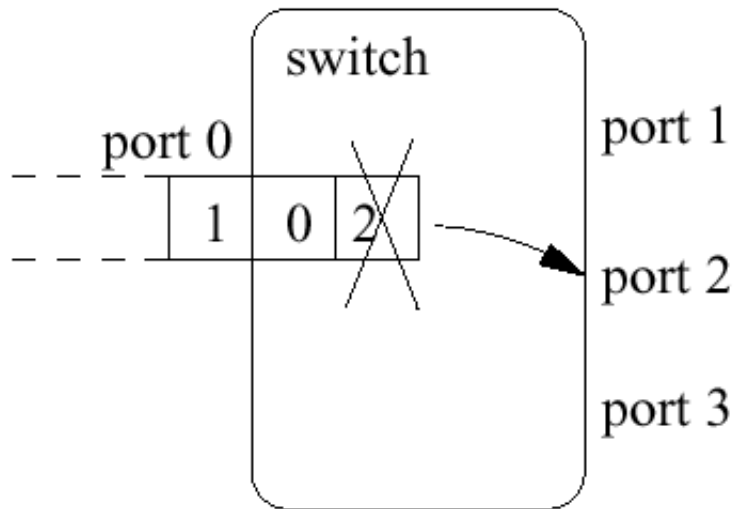
wormhole switching

Design Details (IV)

■ Routing

- outgoing w/ contained address in header
- deterministic and **adaptive routing schemes**
- source-path and table-based routing

Routing Mechanisms



source-path routing

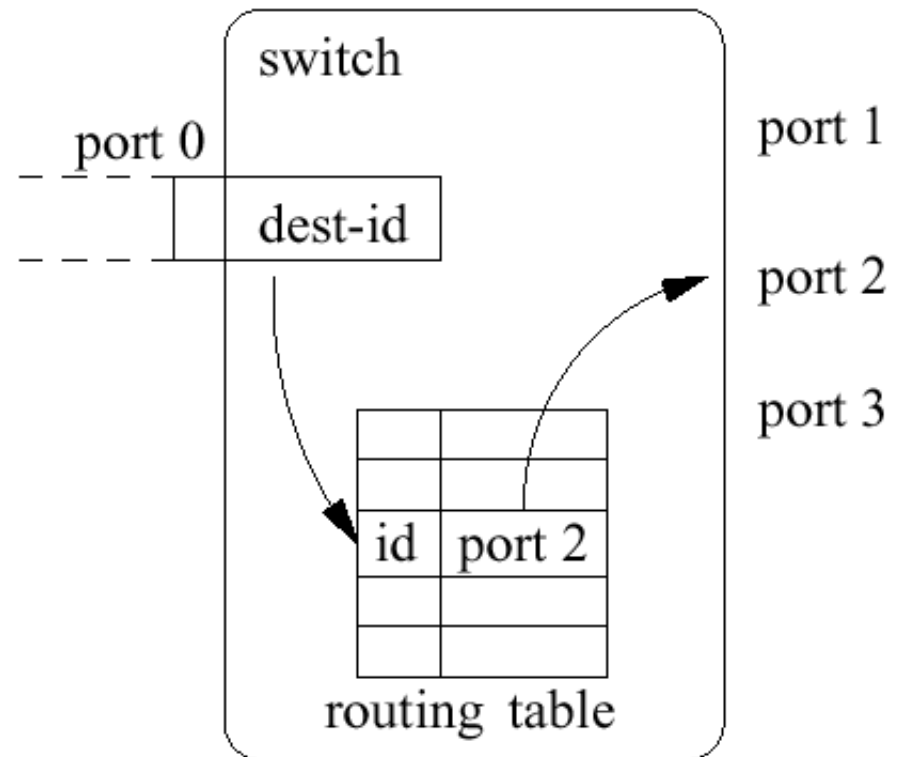


table-based routing

Design Details (V)

■ Flow Control

- avoid buffer overrun
- credit based schemes (get and consume)
 - in case of Myrinet STOP and GO control bytes
- flow control signal travels in the opposite direction relative to the data

■ Error detection and correction

- very low error rate in physical level
- CRC
- software -> hardware
- so need not s/w CRC check => NI can do it !!

Link Parameters

interconnect	unidirectional data rate	switching	routing
Fast Ethernet	100 Mbit/s	packet	table-based
Gigabit Ethernet	1 Gbit/s	packet	table-based
Myrinet	1.28 Gbit/s	wormhole	source-path
ServerNet II	125 Mbyte/s	wormhole	table-based
Memory Channel	100 Mbyte/s	packet	table-based
Synfinity	1.6 Gbyte/s	wormhole	source-path
SCI	400 Mbyte/s	packet	table-based
ATM(OC-12)	155(622) Mbit/s	packet	table-based
HiPPI	800 Mbit/s	packet	table-based

Design Details (VI)

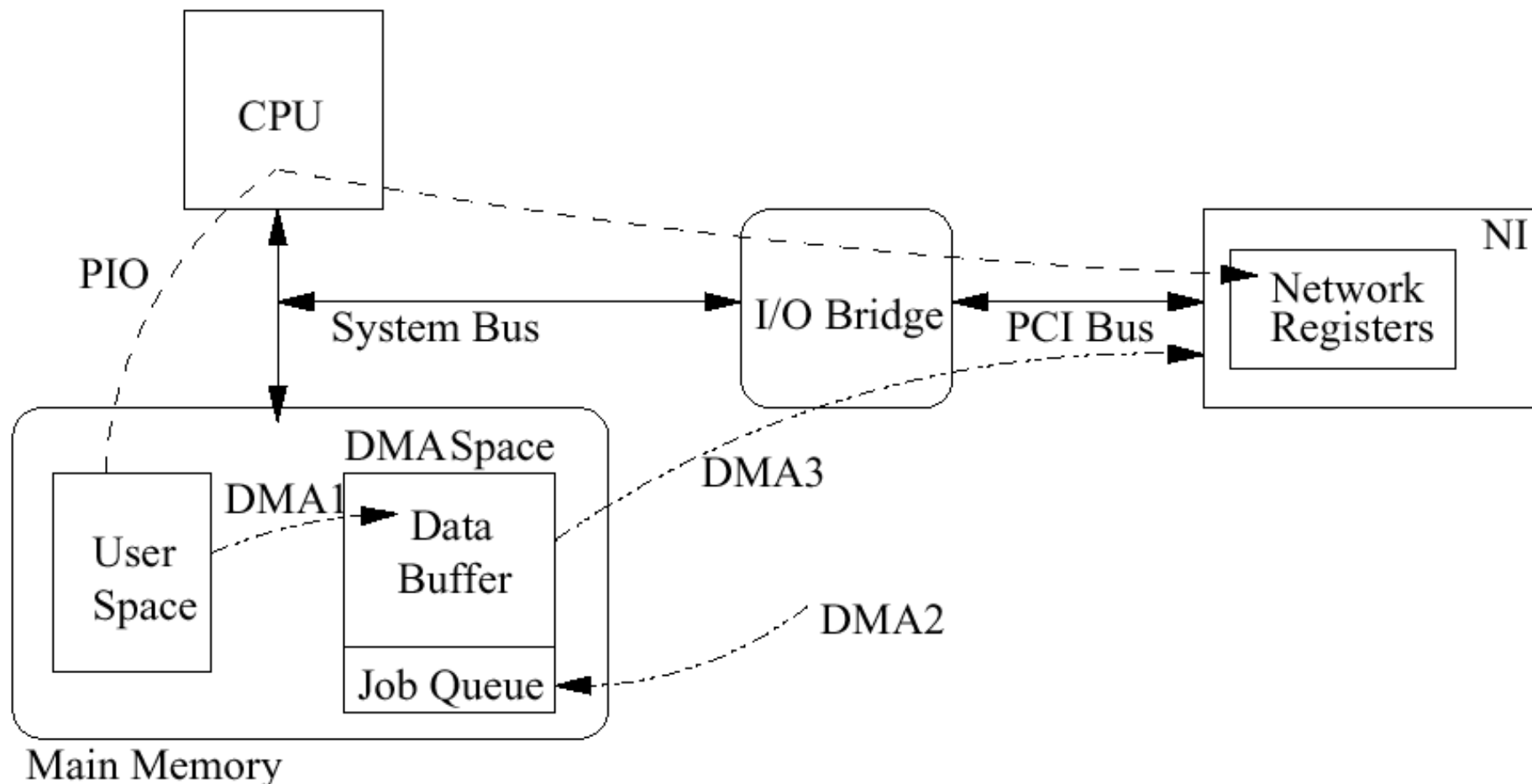
■ Data Transfer

- NI is critical
- user level operation to avoid the costs of OS call
- zero copy mechanism
 - data is transferred to main memory directly

■ Programmed I/O vs. Direct Memory Access

- PIO
 - processor copies data between memory and NI
 - low start up times but inefficient as message size grows
- DMA
 - network device itself initiate the transfer
 - need a bit more: can swap anytime ? Is it running?
 - DMA1 : data copy
 - DMA2 : insert queue
 - DMA3 : NI sets up a DMA transfer to read the msg data from memory

PPIO versus DMA Data Transfer



Design Details (VII)

■ Several factors

- PIO: write message sequentially into a single NI register
 - single bus cycles and poor bandwidth
 - operate burst !!
- writing on consecutive address
 - a special write buffer
 - issued as burst transaction
- an instruction set support for cache control
 - NI can read/write a large block of data
- PIO is superior to DMA for small message
 - because of copy overhead

Design Details (VIII)

■ Polling vs. Interrupts

■ case of DMA

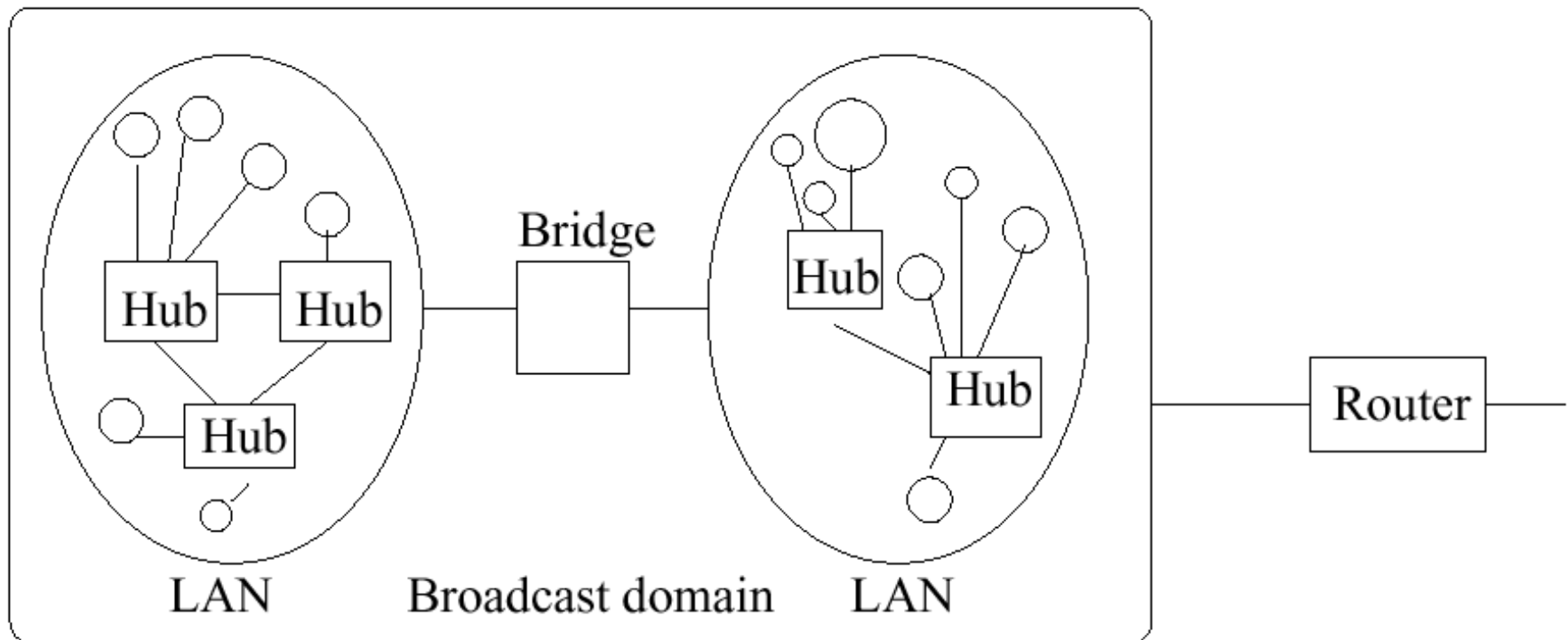
- I/O bus NI - polling wastes a lot of bandwidth
- mirror NI status into main memory (cache-coherent memory)
- Interrupt - context switching overhead
- hybrid solution - programmable watchdog timer

Design Details (IX)

- Collective operation
 - sometimes need collective communication
 - only few have direct communication
 - broadcasting, multicasting, barrier synchronization
 - cluster networks leave collective communication to software: tree-based algorithm
 - case of shared bus system, easy to broadcast
 - Myrinet or ServerNet is more complicate

Fast Ethernet (I)

- 100 Mbps over UTP or fiber-optic cable
- MAC protocol: CSMA/CD (Carrier Sense Multiple Access with Collision Detection)



Fast Ethernet (II)

- Interconnection devices
 - Repeater
 - restore data and collision signal
 - amplify and regenerate signals
 - Hub
 - central point
 - repeat and copy: All can see it
 - Bridge
 - link adjacent LANs: datalink layer
 - filtering
 - forward to other segment
 - Router
 - link adjacent LANs: network layer
 - shortest path

Fast Ethernet Migration

- Replacement of hubs with Ethernet switches
- Fast Ethernet repeater is the perfect complement to a LAN growth strategy

High Performance Parallel Interface (HiPPI) (I)

- Designed to facilitate high speed communication between very high speed computers, & thereby to attempt to meet their I/O requirements
- Designed to be a rapid mover of data, as well as a very implementable standard
- An efficient simplex point-to-point link using copper twisted pair cable for distance of up to 25m
- Standard capable of transferring data: 800Mbit/sec over 32 parallel lines or 1.6Gbit/sec 64 parallel lines

High Performance Parallel Interface (HiPPI) (II)

■ HiPPI standard

- HiPPI-PH
 - the mechanical, electrical, & signaling of HiPPI physical layer
 - support only a single point-to-point connection
- HiPPI-FP
 - packet format and content (including header)
- HiPPI-SC
 - allow a switching mechanism to be built which could allow multiple simultaneous point-to-point connections to occur

■ HiPPI drawbacks

- does not provide a complete, general purpose solution
- a collection of multiprocessors and management systems are needed without sacrificing HiPPIs data transfer speed or efficiency
- max 25 m distance between nodes -> serial HiPPI, SONET extension -> storage problem
- # of connection is restricted due to the SC complexity

HiPPI (III)

■ HiPPI-SC (Switch Control)

- HiPPI-PH: only single point-to-point
- alleviate the number of connections (but not final solution)
- switch grows $O(n^2)$
- linearly connect up to 50-pair within 25 m

■ Serial HiPPI

- overcome 25m distance (HiPPI-PH, standard 50-pair)
- Gigabit dark fiber optics and copper coaxial
- increase the turn around time
- latencies are hidden (only show connected or not)

HiPPI (IV)

- High Speed SONET Extensions
 - HiPPI extender using SONET's STS-12s
 - create a set of terminal devices
 - place STS-12c payload for transmission
 - convert it back to HiPPI
 - lay out PLCP (Physical Layer Convergence Protocol) which maps HiPPI bursts into STS-12c rows
 - Rely on large RAM in the down-link extender
 - $2 \times \text{bandwidth} \times \text{delaybytes}$
 - Features
 - HiPPI to SONET interface implemented completely in hardware
 - 4 Mbyte of RAM of data buffering (on the down-link side)
 - 64 bit CRC transmitted with each STS-12c frame
 - An i960 RISC-based microprocessor is included for supervisory functions and monitoring

HiPPI (V)

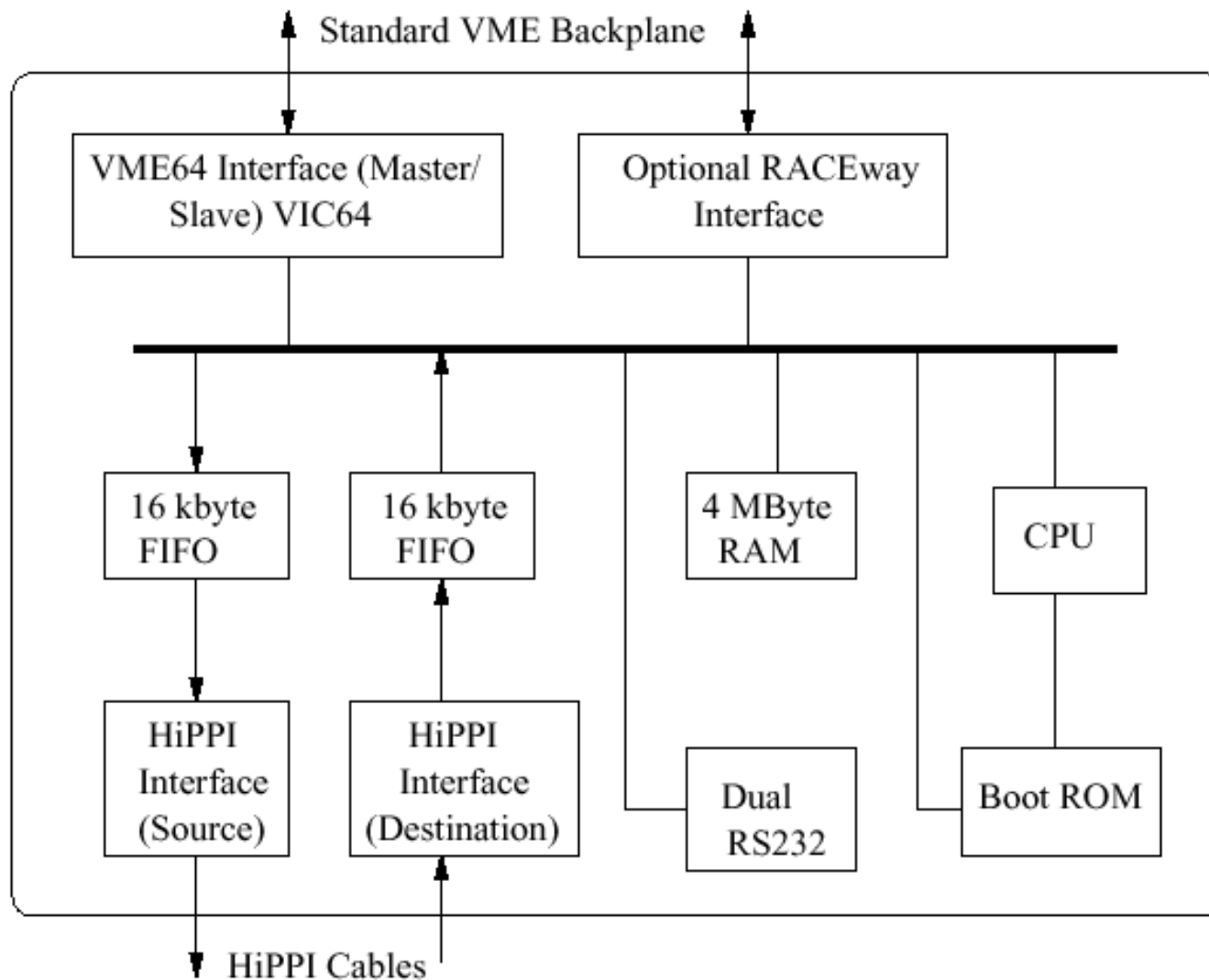
■ HiPPI Connection Management

- Centralized
- Broadcast: policies upon shared knowledge
- Distributed: no explicit knowledge (random wait and retrying methods)

■ HiPPI Interface

- Used primarily as a high speed networked data channel between computer systems & supercomputers
- Full duplex and direct connection with another HiPPI interface or conjunction of HiPPI
- Various HiPPI interfaces available
 - VME-HiPPI, SBUS, PC and workstation standard, special interfaces used in CRAY computers

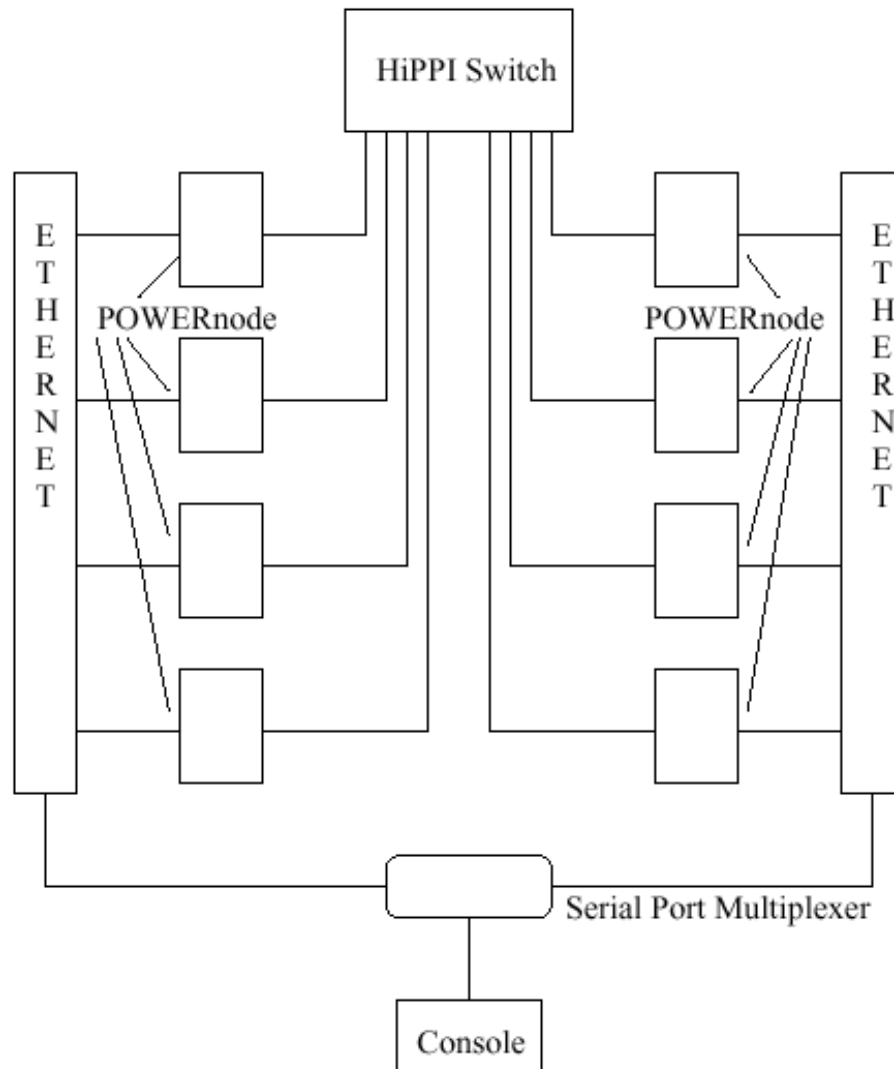
VME64 HiPPI Interface Block Diagram



HiPPI (VI)

- Array System: The HiPPI Interconnect
 - distributed memory multiprocessor
 - 100 or more MIPS processors in as many as 8 POWERnode
 - peak aggregate computing capacity in excess of 50 GFLOPS
 - HiPPI switch is nonblocking with sub-microsecond connection delays

Array System Schematic



Asynchronous Transfer Mode (ATM) (I)

- Offer significantly greater bandwidth, flexibility, & QoS service support
- Consist of ATM switches, ATM routers, & LAN switches
- Connection oriented packet switching
- Highly suitable for wide area LAN and WAN
- ATM routers and LAN switches are more effective
- Huge bandwidth
 - cost effective to transfer large quantities of data
- Not so good for cluster computer interconnection
 - hardware cost
 - not good performance in LAN
 - effective in supporting clusters over WAN

ATM (II)

■ Concepts

- VCI (Virtual Circuit Identifier)
 - information to be carrier over the network is broken up into blocks (cells) with an identifying label called VCI
 - VCI is attached to each block
 - VPI (virtual path identifier): group of VCIs
- Multicast Virtual Circuit
 - used to transfer info from a single source to several recipients
 - replication and forwarding (switching system)
- Switched virtual circuit (SVC) vs. Permanent virtual circuit (PVC)
 - SVC: automatically set up by ATM signaling & flexible a fast connection setup time of 10 ms
 - PVC: manually setup for leased lines applications

ATM (III)

■ Concepts

■ ATM Adaptation Layer (AAL)

- support many kinds of services

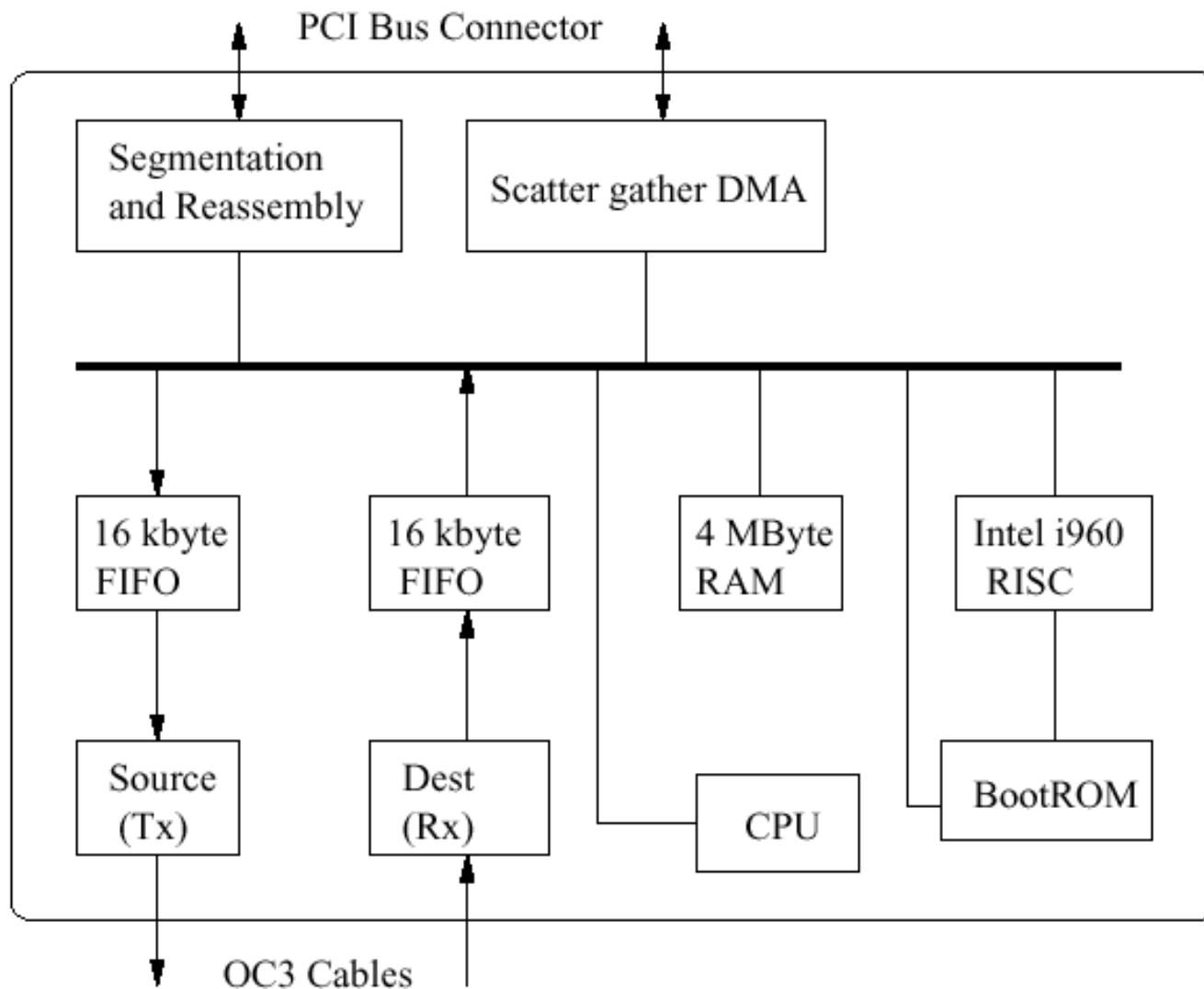
- Four types of AAL

- AAL1: support connection-oriented services that require constant bit rates services and have specific timing and delay requirements
- AAL2: support connection-oriented services that **do not** require constant bit rates, but service variable bit rate applications
- AAL3/4: for both connectionless and connection-oriented variable bit rate services
- AAL5: support connection-oriented variable bit rate services

ATM (IV)

- ATM Adapter
 - FORE system
 - embedded Intel i960 RISC
 - AAL5 and 3/4 Segmentation and Reassembly (SAR)
 - scatter-gather DMA

Block Diagram of ATM Adapter



ATM (V)

■ ATM API Basics

- SAP(Service access point)
 - ATM Address, ATM selector, Broadband low layer information (BLLI), and Broadband high layer information (BHLLI)
 - [ATM address, ATM Selector, BLLI id2, BLLI id3, BHLLI id]
 - BLLI ids: layer 2 protocol
 - BHLLI id: application layers
 - SAP vector element (SVE): tag, length, and value field

ATM (VI)

- Performance Evaluation of ATM
 - Hardware
 - 4 Sparc Sun workstations with 10 Mbit/s Ethernet adapter and a Fore system ATM adapter connected to a Fore system ASX-200 ATM switch
 - PDE (parallel differential equations)
 - parallel matrix multiplication
 - promising over local network
 - acceleration may be considered

Execution Time (sec) of Partial Differential Equation

Protocol hierarchy/ Network	Mesh size					
	16x16		64x64		256x256	
Accuracy	10^{-6}	10^{-12}	10^{-6}	10^{-12}	10^{-6}	10^{-12}
Sequential	0.07	0.17	5.15	10.28	330.71	661.45
PVM						
ATM	0.30	0.58	3.09	6.13	137.28	273.83
Ethernet(Silent)	0.33	0.65	3.27	6.50	138.39	276.78
Ethernet(30% loaded)	0.35	0.68	3.41	6.70	140.24	279.18
BSD Socket						
ATM	0.11	0.26	2.47	4.91	133.69	266.84
Ethernet(Silent)	0.14	0.28	2.65	5.19	134.79	268.79
Ethernet(30% loaded)	0.19	0.37	2.69	5.44	135.96	271.75
FORE's API						
ATM	0.12	0.22	2.45	2.83	133.25	266.07

ATM (VII)

- Issues in Distributed Networks for ATM Networks
 - Resource management
 - data rate can adapt to data traffic and available network availability
 - peak rate leads to significant inefficiency
 - Multicast routing
 - a greedy algorithm
 - add new endpoints using a shortest path from the endpoint to the connection
 - delete endpoints by pruning the branch needed only by the endpoint being dropped

Scalable Coherent Interface (SCI) (I)

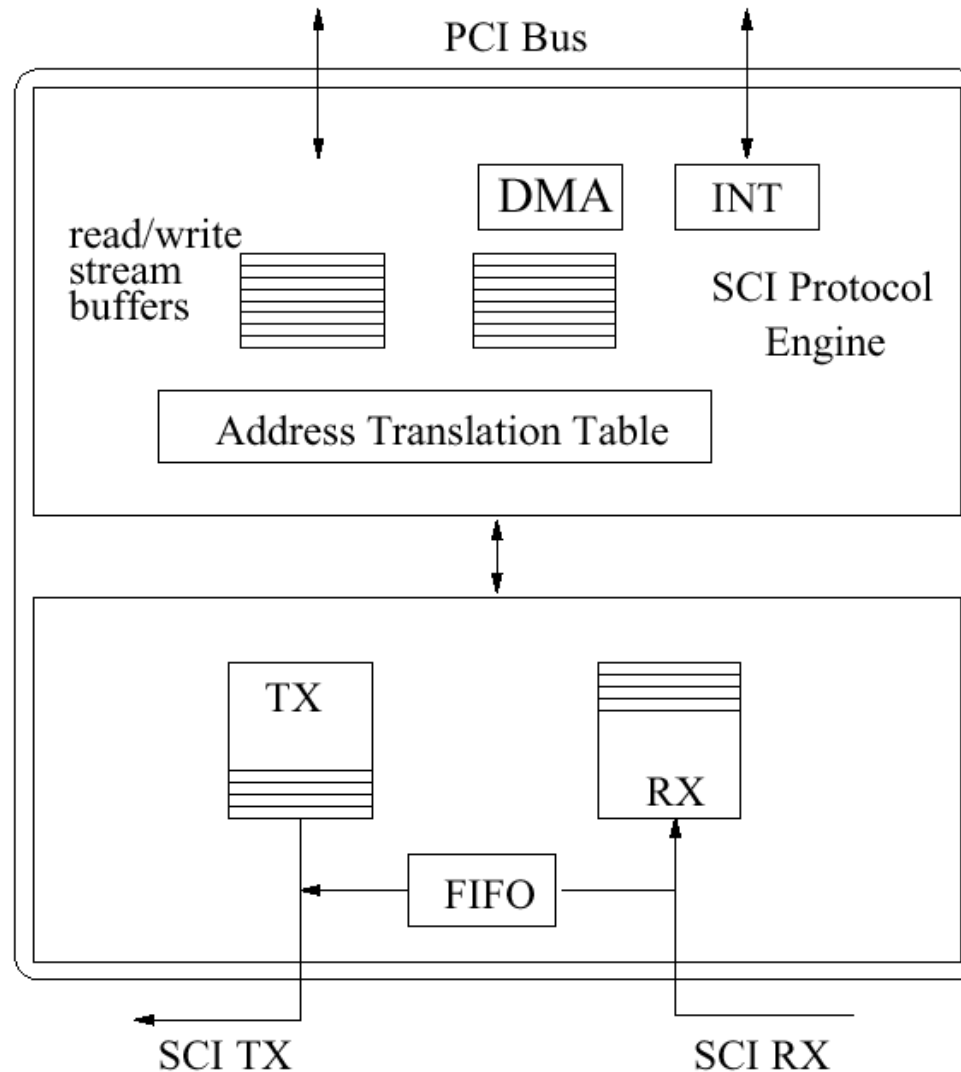
- A recent communication standard for cluster interconnects
- Effectively a processor memory & I/O bus, high performance switch, local area network, & optical network
- An info sharing & info communication system that provides distributed directory-based cache coherency for a global shared memory model & uses electrical or fiber optic point-to-point unidirectional cables of various widths
- A single address space is used to specify data as well as its source & destination when transported
- 200 Mbyte/s (CMOS) to 1000 Mbyte/s (BiCMOS) over distances of tens of meters for electrical cable & kilometers for serial fibers

SCI (II)

■ Data transfer via SCI

- can interface with common buses such as PCI, VME, etc, & to I/O connections such as ATM or Fiber Channel
- 8000 Mbps
- Usual approach of moving data
 - When the data arrive at the destination, hardware stores them in a memory buffer and alerts the processor by an interrupt when a packet is complete or the buffers are full
 - Software then moves the data to a waiting user buffer
 - User application examines the packet to find the desired data
- Cache coherence scheme is comprehensive & robust
 - independent of the interconnect type or configuration
 - can be handled entirely in HW
 - provide distributed shared memory with transparent caching that improves performance by hiding the cost of remote data access
 - eliminate the need for costly SW cache management

Block Diagram of the Dolphin PCI-SCI Bridge



SCI (III)

■ Advantages of SCI

- reduce the delay of interprocessor comm by an enormous factor
 - SCI eliminates the need for runtime layers of SW protocol-paradigm translation
- most useful for clustering over local area distance or less
 - least suitable over long distance
- remote communication is opcode
- remote address cache miss => get from data
- performing all the network protocol as a fraction of one instruction
- distributed cache-coherent mechanism
- simple - efficient with large blocks of data
 - but lots of handshaking and heavy traffic
- each interface chip can handle active packets concurrently (flight awaiting)

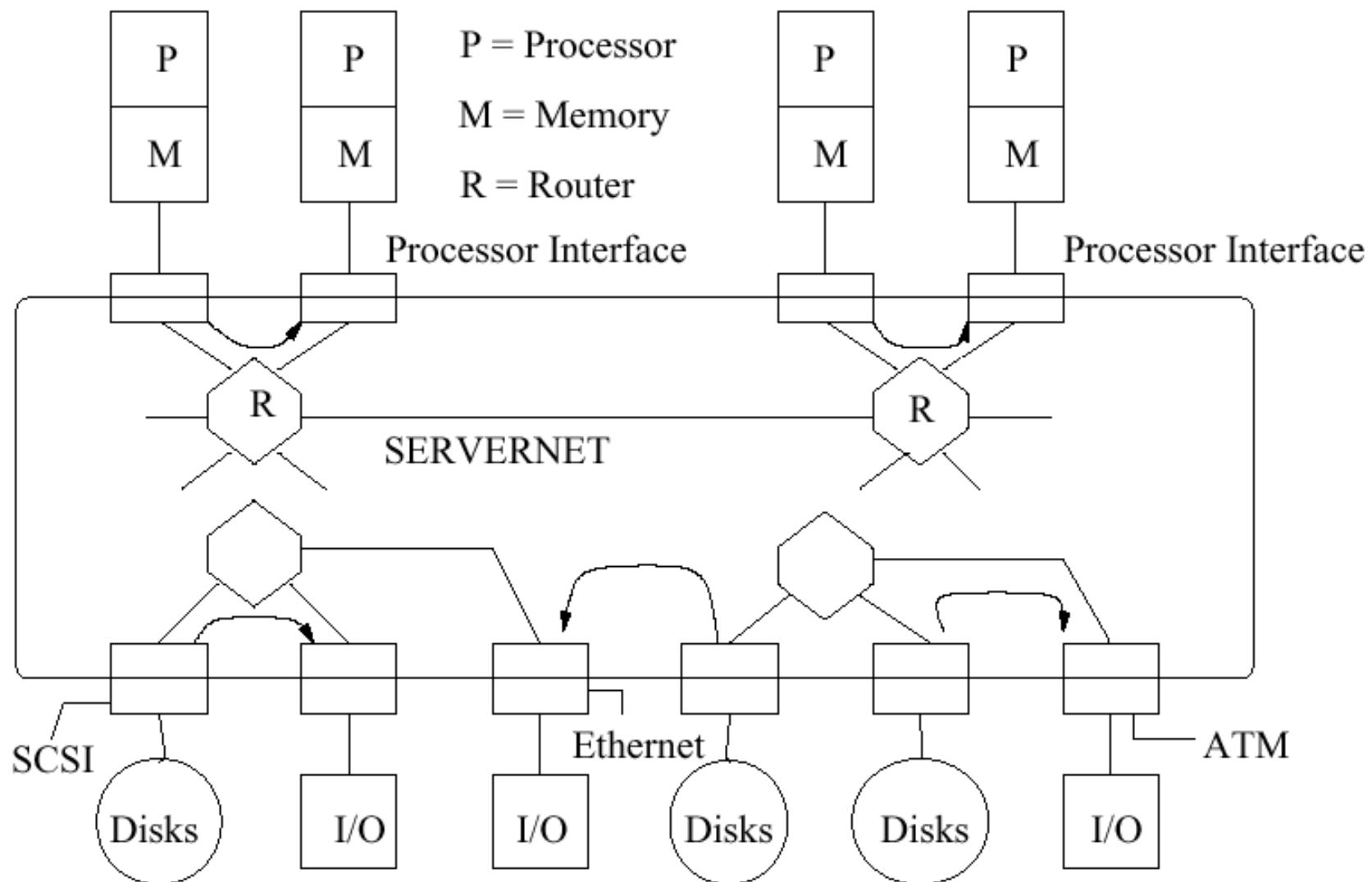
ServerNet (I)

- The first commercially available implementation of a SAN in 1995 by Tandem, now supported by Compaq
- ServerNet II in 1998
 - raise the bandwidth & add new features
- Intend to provide high bandwidth, scalable, and reliable interconnect between processors and I/O devices required by HPC applications, but turned quickly into a general purpose SAN
- 125Mbyte/sec bandwidth between two nodes in a clustered system

ServerNet (II)

- Scalability and reliability as main goal
 - consist of endnodes & routers
 - endnodes with interface to the system bus or various I/O interfaces
 - routers to connect all endnodes to one clustered system
- Ability to transfer data directly between 2 I/O devices, thus relieving processors of plain data copy jobs (zero copy)

A Sample ServerNet Configuration



ServerNet (III)

■ ServerNet Links

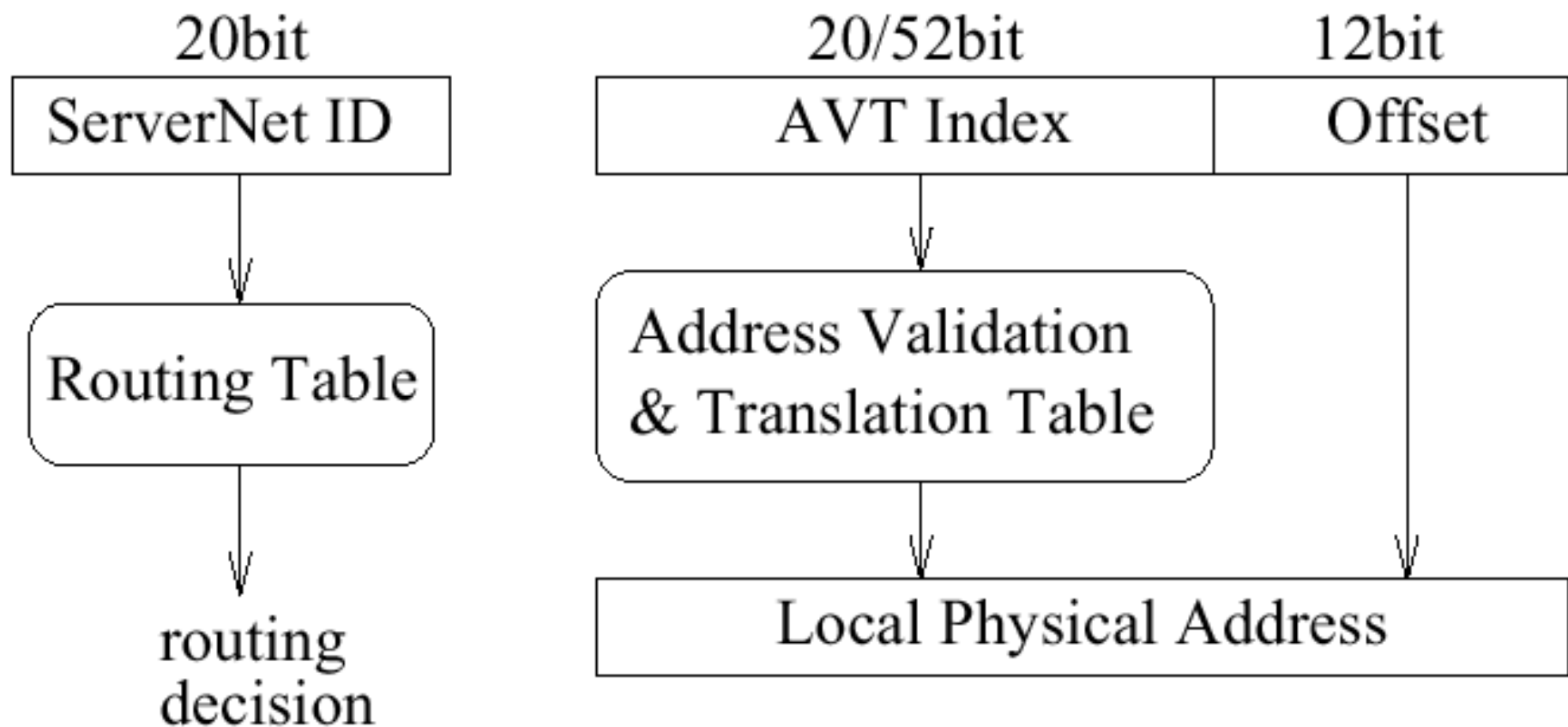
- full duplex, wormhole switched network
- 1st version: 9 bps , 50MHz
- 2nd version: 125MByte/s
- serial copper cable => longer distance
- with additional converter, ServerNet I and II components can be mixed within one system
- operate asynchronously and avoid buffer overrun through periodic insertion of SKIP control symbols, which are dropped by the receiver

ServerNet (IV)

■ Data transfer

- DMA-based remote memory read/write
- endnode: 64 (512 in ServerNet II) byte from/to a remote memory
- check read/write permissions
- (advance) specify one of several packet queues

ServerNet Address Space



ServerNet (V)

■ Fault Tolerance

- support guaranteed & error free in-order delivery of data on various levels
- check validation at each stage
- sending acknowledge

■ Switches

- 1st: 6 port switches, 2nd: 12 ports
- separate the data channel and control channel
- ability to form so called Fat Pipes
 - Several physical link can be used to form one logical link and choose dynamically

ServerNet (VI)

- Driver and Management Software
 - low overhead protocol layers and driver software
 - mechanism to efficiently support the message passing model of the VIA
 - IBC (In Band Control): same links as normal data packets
 - IBC protocol is responsible for initialization, faulty node isolation and several other management issues
 - IBC packets are used to gather status or scatter control data to all ServerNet components

ServerNet (VII)

- Focus on the business server market, poorly accepted by researchers so far
- A lot of properties, extremely useful for cluster computing
 - error handling on various levels
 - a kind of protection scheme (AVT)
 - standard physical layers (1000BaseX cables)
 - support for network management (IBC)
- Will be one of the leading SANs
 - several companies will use ServerNet for their servers & clusters
 - considerable influence on the VIA specification

Myrinet (I)

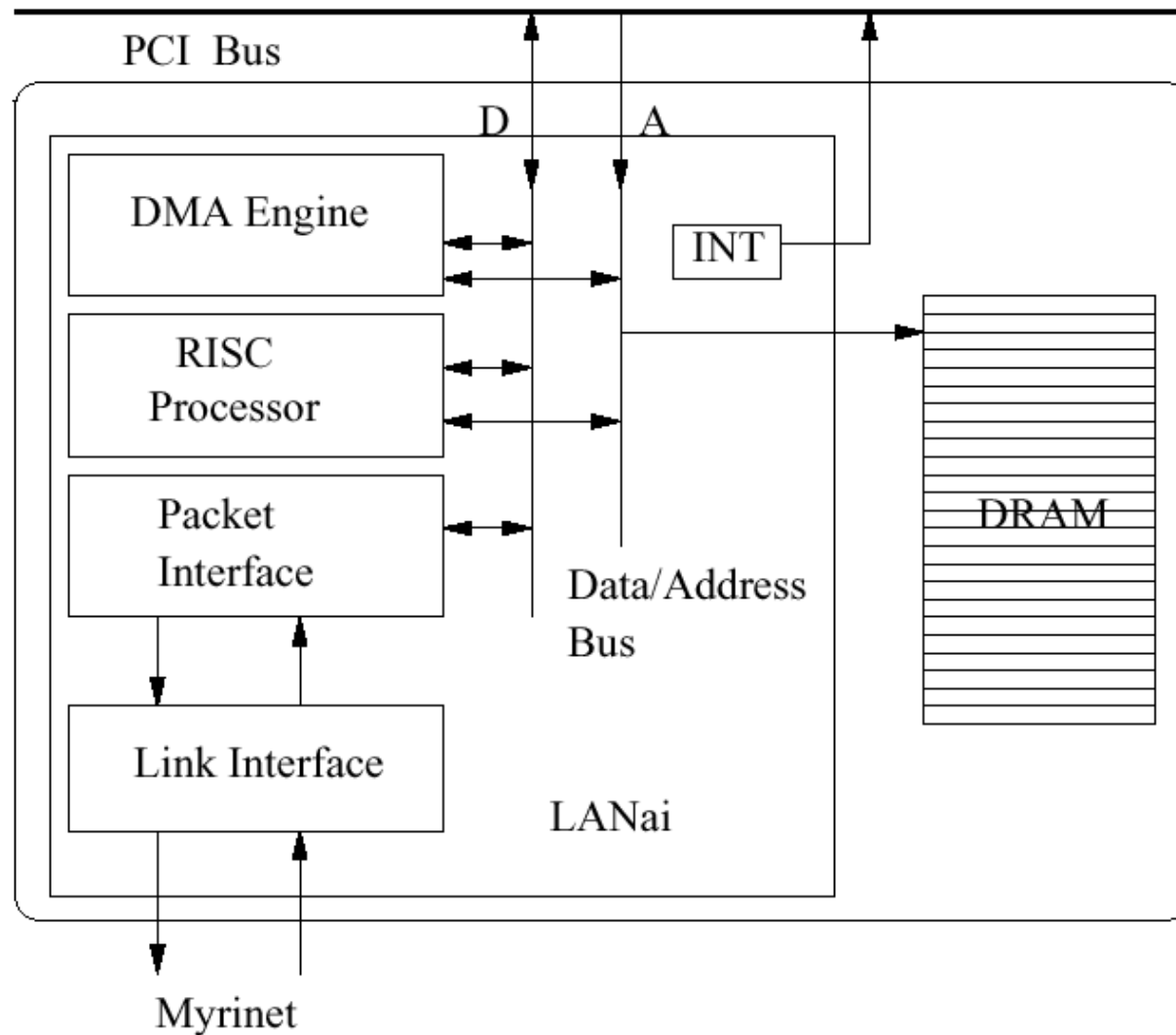
- A SAN evolved from supercomputer technology
- A main product of Myricom (founded in 1994)
- Quite popular in the research community
 - all HW & SW specifications are open & public
- Based on 2 research projects
 - Mosaic by Caltech
 - a fine grain supercomputer, need a truly scalable interconnection network with lots of bandwidth
 - Atomic LAN by USC
 - based on Mosaic technology, a research prototype of Myrinet
- Speed: 1.28 Gbps
- Good price/performance ratio

Myrinet (II)

■ Host interface

- LANai chip
 - a custom VLSI chip, a programmable microcontroller
 - control the data transfer between the host & the network
- SRAM memory
 - Message data must first be written to the NI SRAM, before it can be injected into the network
- (+) the great flexibility of the HW due to a programmable microcontroller,
- (-) but can also be a bottleneck with respect to performance since the LANai runs only at moderate frequencies

Myrinet Host Interface

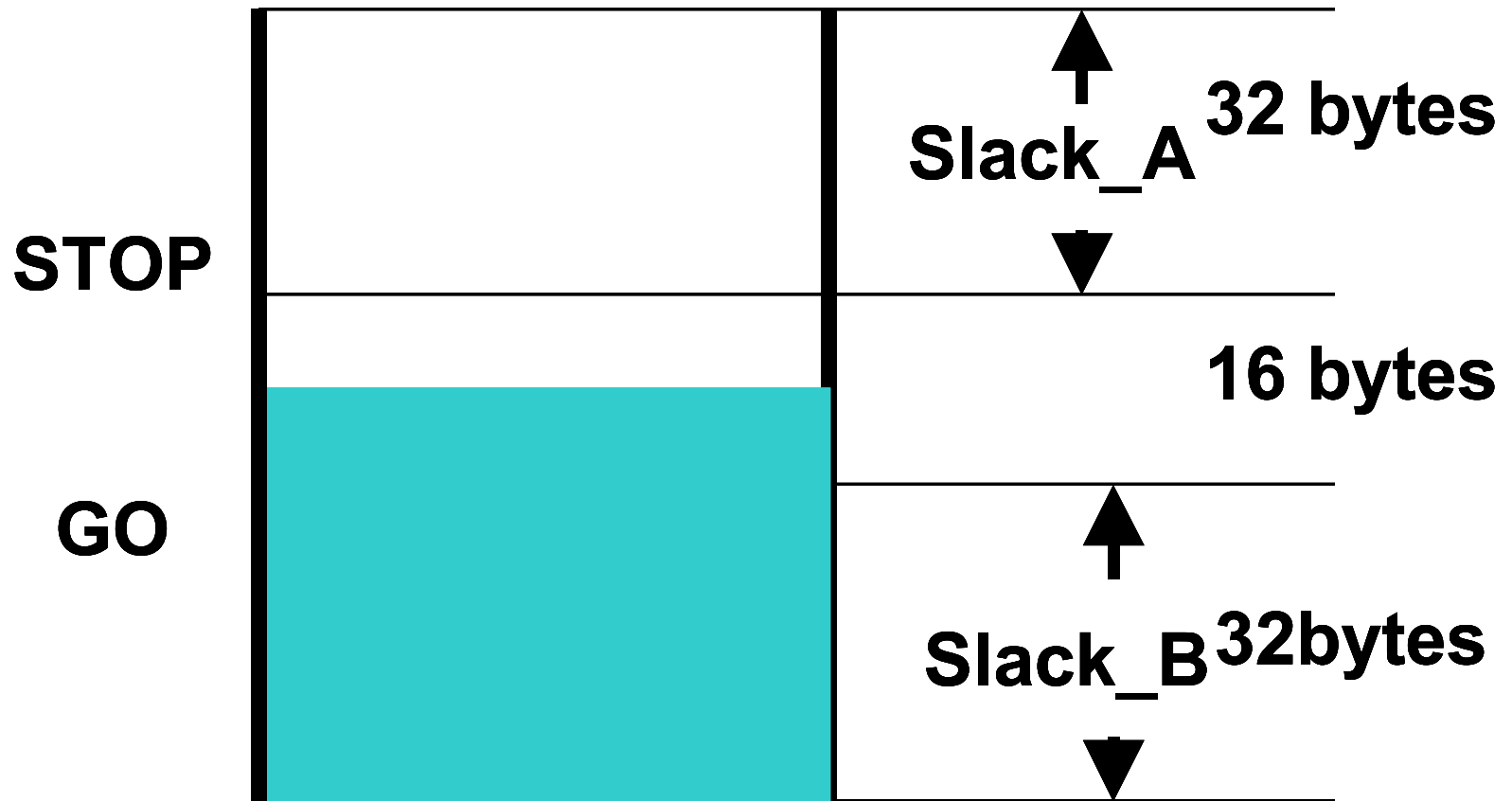


Myrinet (III)

■ Link and Packet Layer

- similar ServerNet
- full duplex 9 bit parallel channel in one direction running at 80MHz
- network offer 160Mbyte/s physical bandwidth over one channel
- two different cable type (SAN, LAN)
 - 3m SAN link, 10m LAN link
- variable length data format
- route with wormhole switching
- source path routing
- consist of routing header
- special control symbols (STOP, GO)

Flow Control (Slack Buffer Operation)



Myrinet (IV)

■ Switches

- 4, 8 and 16 ports, mixable SAN and LAN
- any network topology
- autodetect the absence of a link
- starting up, host interface detect network topology automatically

■ Error Handling

- MTBF: million hours are reported
- cable fault and node failure
 - alternative routing by LANai
- prevent deadlock: time out generates a forward reset (FRES) signal

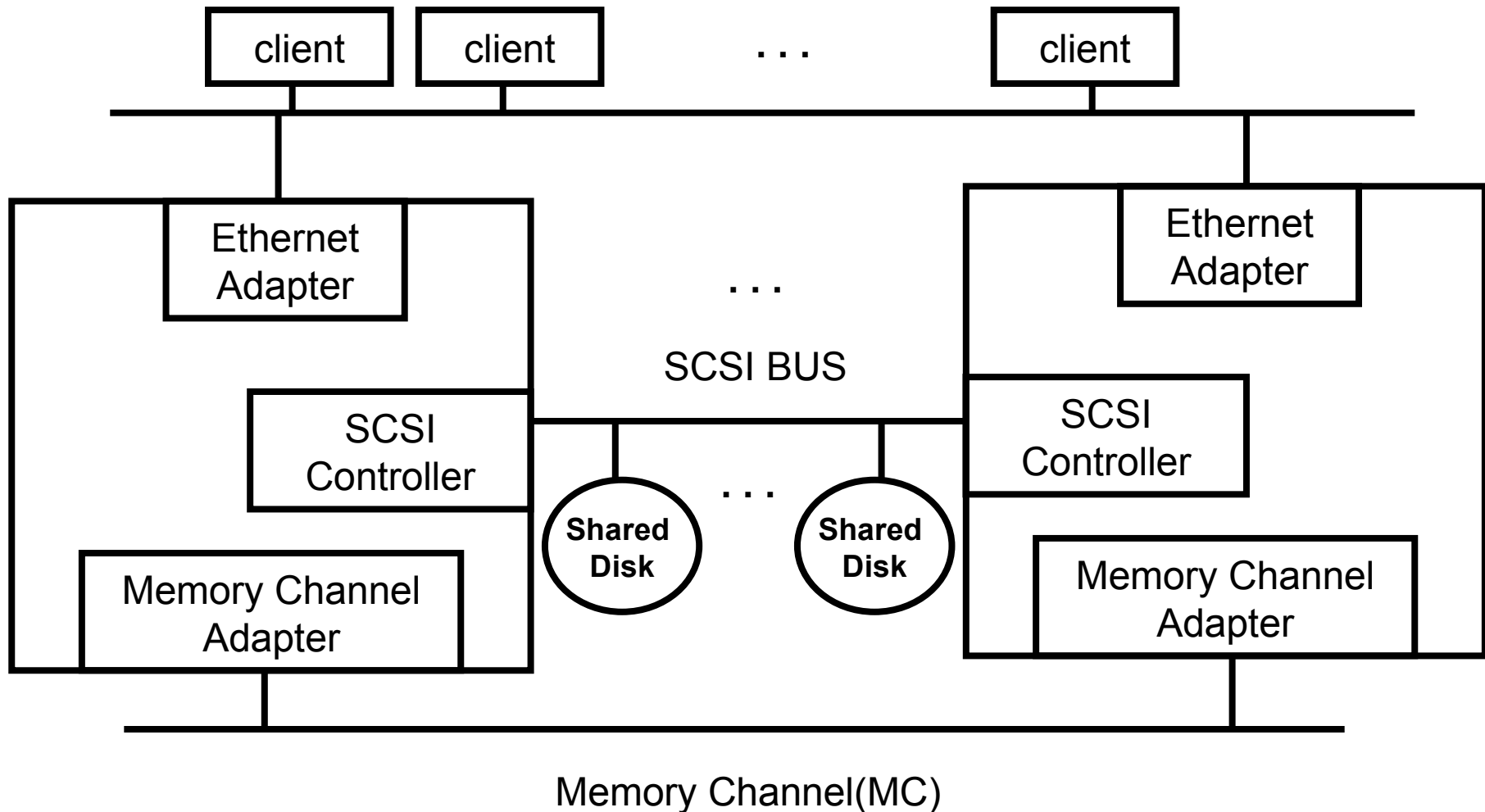
Performance of Message Layers over Myrinet

Machine	API	Latency (μ s)	Bandwidth (Mbit/s)	Ref.
200 MHz PPro	BIP	4.8	1009	LHPC
166 MHz Pentium	PM	7.2	941	RWCP
Ultra-1	AM	10	280	GAM
200 MHz PPro	TCP (Linux/BIP)		293	LHPC
200 MHz PPro	UDP (Linux/BIP)		324	LHPC
DEC Alpha 500/266	TCP (Digital Unix)		271	Duke
DEC Alpha 500/266	UDP (Digital Unix)		404	Duke

Memory Channel (I)

- A completely different approach towards SANs
- Provide a portion of global virtual shared memory by mapping portions of remote physical memory as local virtual memory
- Obtained from Encore & several other projects such as VMMC or SHRIMP
- Speed: 800 Mbps

Memory Channel (II)



Memory Channel (III)

- Bringing Together Simplicity and Performance
 - consist of a PCI adapter & a Hub
 - 1st version: shared medium => bottleneck
 - 2nd version: point-to-point, full-duplex 8x8 cross bar
 - heartbeat signal and flow control
 - detect node failure or blocked data transfers

Comparison of Memory Channel 1 and 2

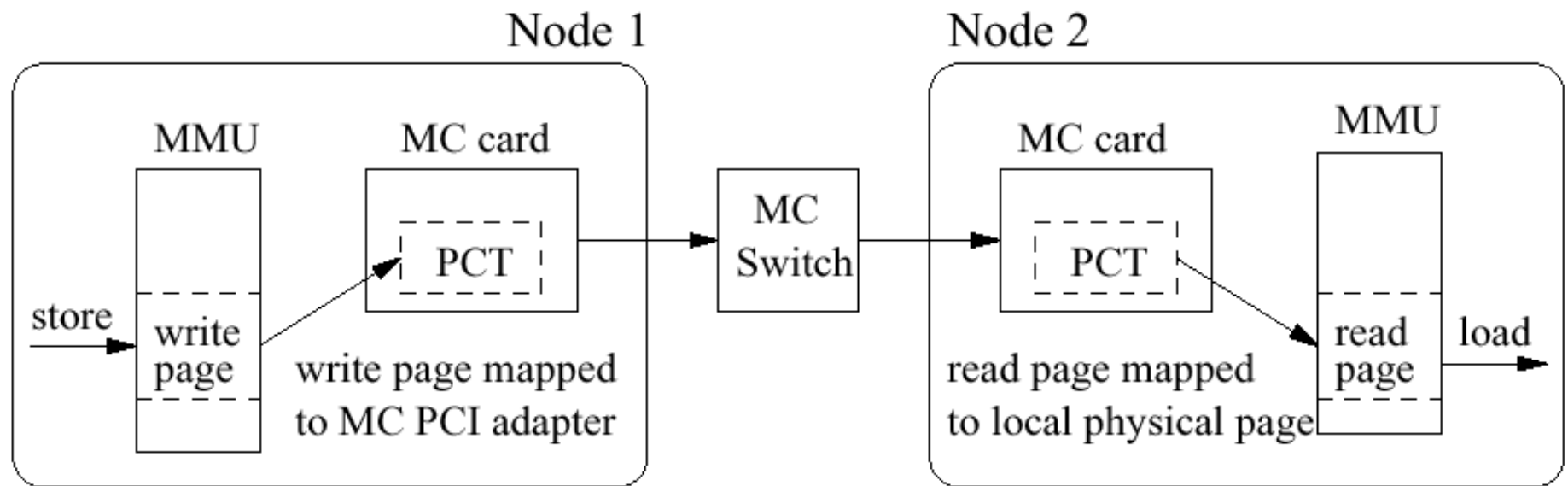
Characteristics	Memory Channel 1	Memory Channel 2
channel data width	37 bit (half-duplex)	16 bit (full-duplex)
link frequency	33 MHz	66 MHz
max. copper cable length	4 m	10 m
max. one way transfer rate	133 Mbyte/s	133 Mbyte/s
sustained pt2pt bandwidth	66 Mbyte/s	100 Mbyte/s
max. packet size	32 byte	256 byte
remote read	no	yes
supported page size	8 Kbyte	4/8 Kbyte
hub architecture	shared bus	crossbar

Memory Channel (IV)

■ Data Transfer

- Map pages as read- or write-only into virtual address space
- Read-only page: a page is pinned down in local physical memory
- Write-only: page table entry is created in PCI interface
 - store a local copy of each packet
 - request acknowledge message from receiver
 - define the packet as broadcast or point-to-point packets

Data Transfer over Memory Channel



Memory Channel (V)

■ Software and Performance

- Memory Channel and UMP(Universal Message Passing)
 - MPI, PVM, HPF
 - Alpha 4100 nodes (300 MHz 21164 CPU) in a two node configuration
- Reduce communication to the minimum: simple store operations
 - latencies for single data transfers are very low
 - One way latency for an 8 byte ping-pong test: 2.2 μ s (raw), 5.1 μ s (HPF) and 6.4 μ s (MPI)
 - reach the max sustained datarate of 88 Mbyte/s with relative small data packets of 32 byte
- Largest possible configuration
 - 8 12-CPU Alpha server nodes: a 96-CPU cluster

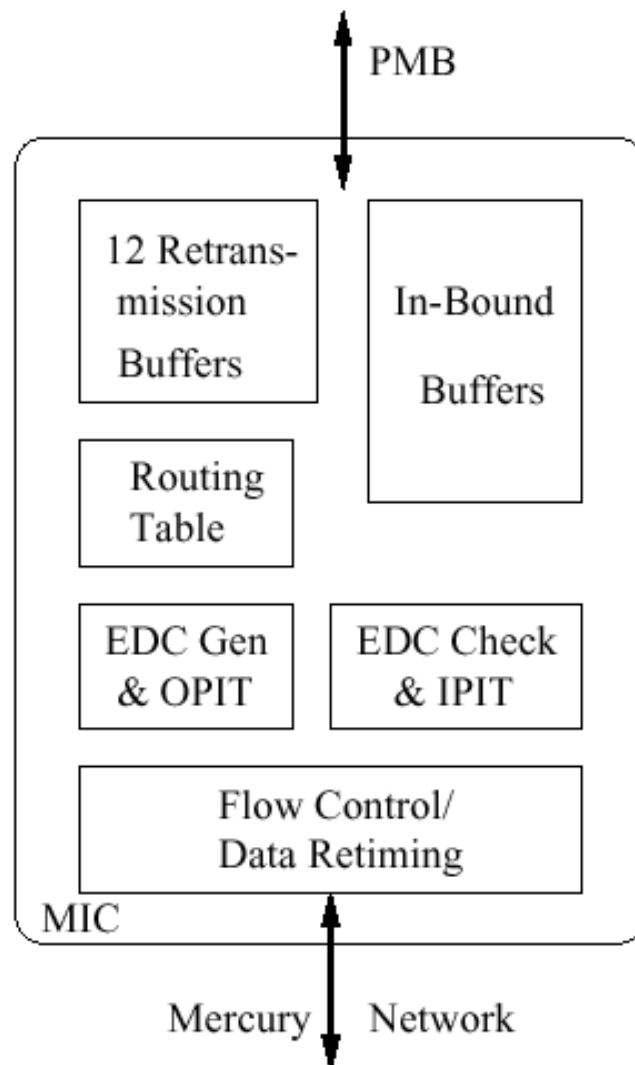
Synfinity (I)

- Developed & marketed by Fujitsu System Technologies (FJST)
 - a business unit of HAL computer Systems & Fujitsu
 - based on the technology of HAL's Mercury interconnect
- Support both parallel programming models: Message Passing & Shared Memory
- 3 components
 - Synfinity NUMA
 - an adapter to connect SMP nodes together to one ccNUMA cluster
 - Synfinity CLUSTER
 - a host adapter intended for Message Passing
 - Synfinity NET
 - a six-port switch to connect several interfaces together to 1 cluster

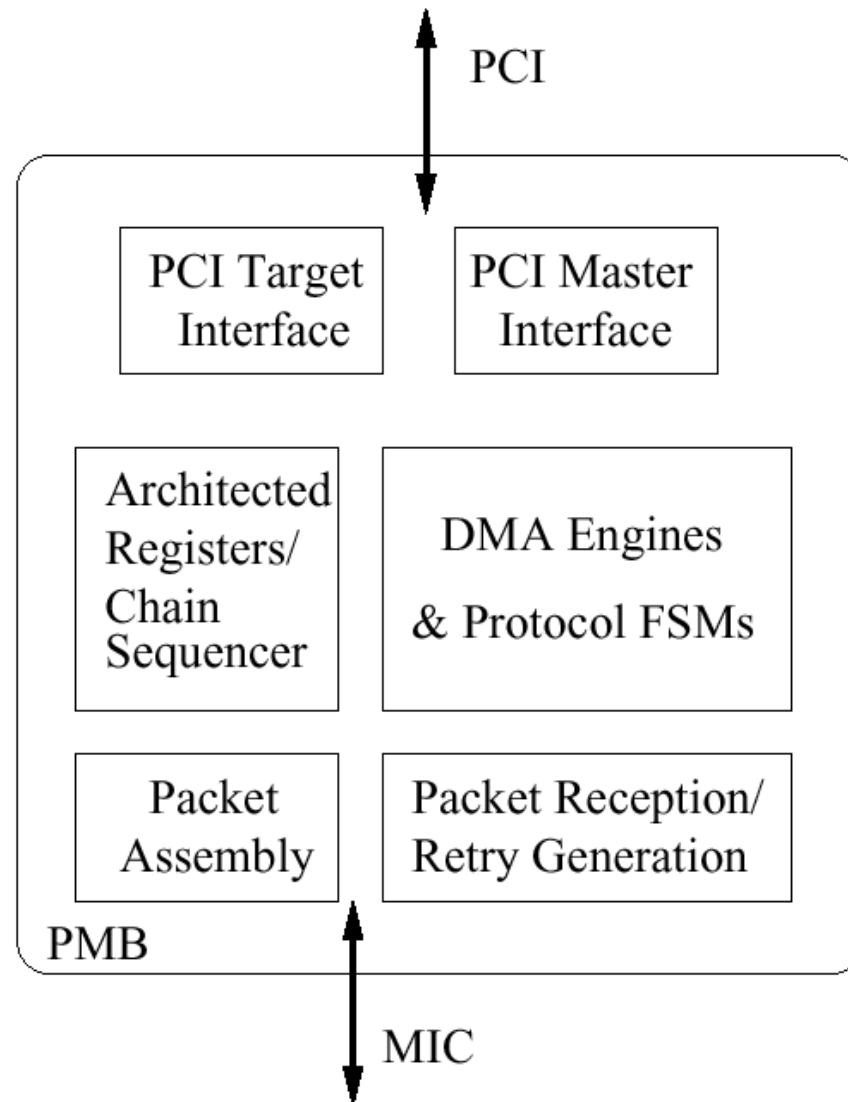
Synfinity (II)

- Pushing networking to the technological limits
 - Synfinity NUMA
 - with Xeon processor, forming ccNUMA
 - CLUSTER interface
 - 32/64 bit , 33/66MHz available
 - PMB (PCI to Mercury Bridge) and MIC (Mercury Interface Chip)

Synfinity Mercury Interface Chip (MIC)



Synfinity PCI to Mercury Bridge (PMB)



Synfinity (III)

■ Data Transfer

- various data transfer and communication services
- Plain Send/Receive
- Put/Get
- RMW (Remote Atomic Read Modify-Write)
- choose Fetch/Add or Compare/Swap
- special barrier
- Initiated chained descriptors
- credit-based mechanism
- Bust/Retry acknowledge

Synfinity (IV)

■ Switches

- 6port, virtual cut-through switch (200MHz)
- 64 nodes in an arbitrary topology
- Source-path routing
- To prevent blocking of small special-service packets, 3 arbitration priorities allow small message to bypass large Put/Get packets

Synfinity (V)

- Software is available as a VIA implementation under Window NT
 - 220 Mbyte/s bandwidth on a 64 bit/66 MHz PCI Sun Ultra
- Looks good for message passing system
- Several communication mechanism directly supported in hardware
 - enables thins and fast protocol layers
- Advanced Flow control in heavy load is very interesting

Reference Book

