The GigaRing Channel:

A new supercomputer-class system interconnect

Steve Scott

Cray Research, Inc.

Project Requirements

- Next generation channel for I/O and inter-system communication
- Interoperability between all CRI machines
- High bandwidth (on the order of 1 GB/s)
- Scalability: cost *and* performance
- High reliability
- On-line client maintenance

    (must be able to add, remove and swap nodes while operating)

Approach:

- Start with basic SCI topology and channel arbitration protocol
- Add features to enhance performance, reliability and flexibility

GigaRing Overview

Topology

- Ring based $\Rightarrow$ very flexible and scalable

Functionality

- DMA transfers
- Messaging
- Remote client maintenance

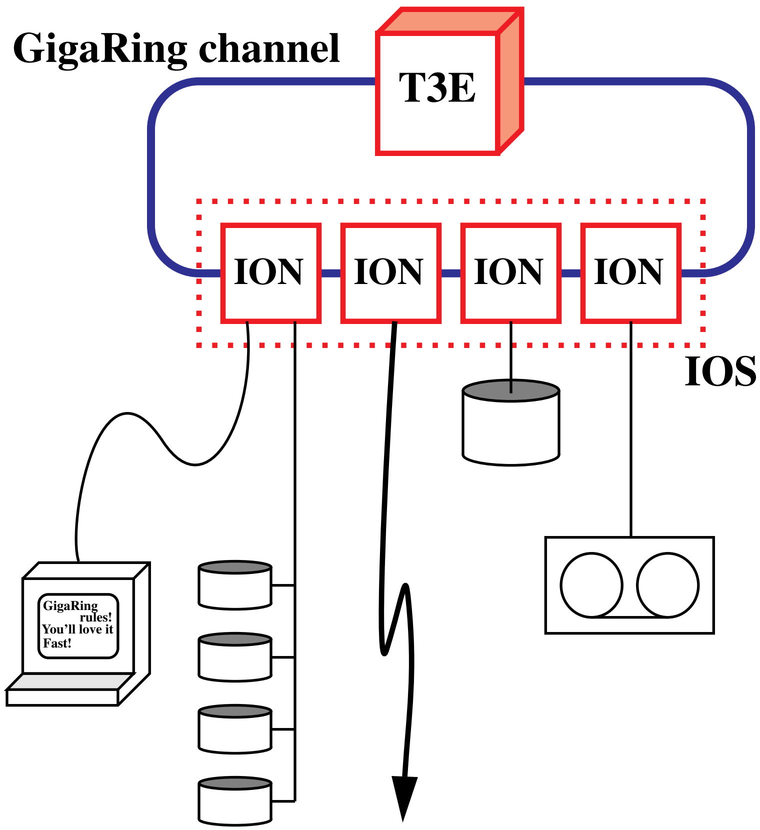
Performance

- Payload bandwidth over 900 MB/s half duplex or 2×800 MB/s full duplex
- Multiple virtual channels and adaptive congestion control
- Integrated control and data streams with priority mechanism

Reliability

- Conservative transmission rates
- Excellent error detection
- On-line reconfiguration to map out faulty resources

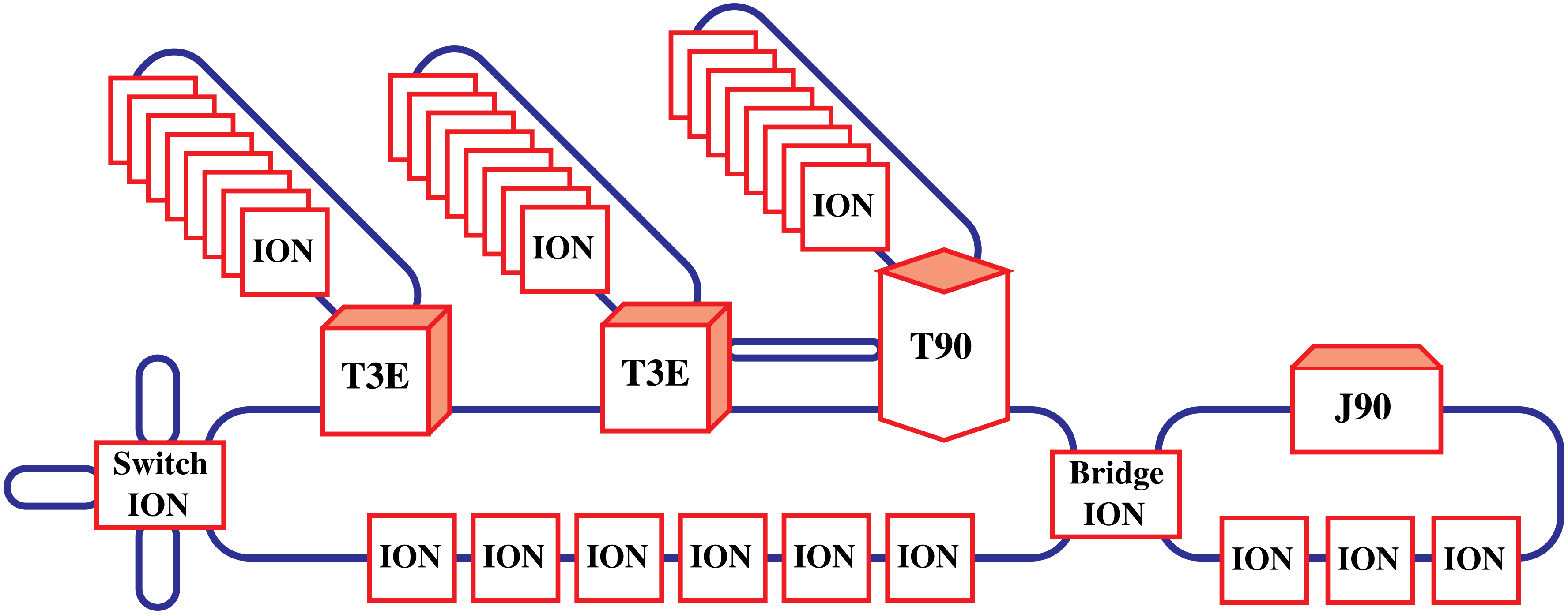
System Configurations



Supported peripherals and networks:

- Ethernet
- HIPPI
- Fibre Channel disks and arrays
- IPI disks, arrays and tapes
- Block mux tapes
- ESCON tapes
- FDDI
- SCSI
- ATM
- etc.

System Configurations



- Both private and shared I/O Subsystems
- Dedicated point-to-point links
- Bridges logically couple channels while physically separating them for bandwidth
- Switches for greater expandability with high performance

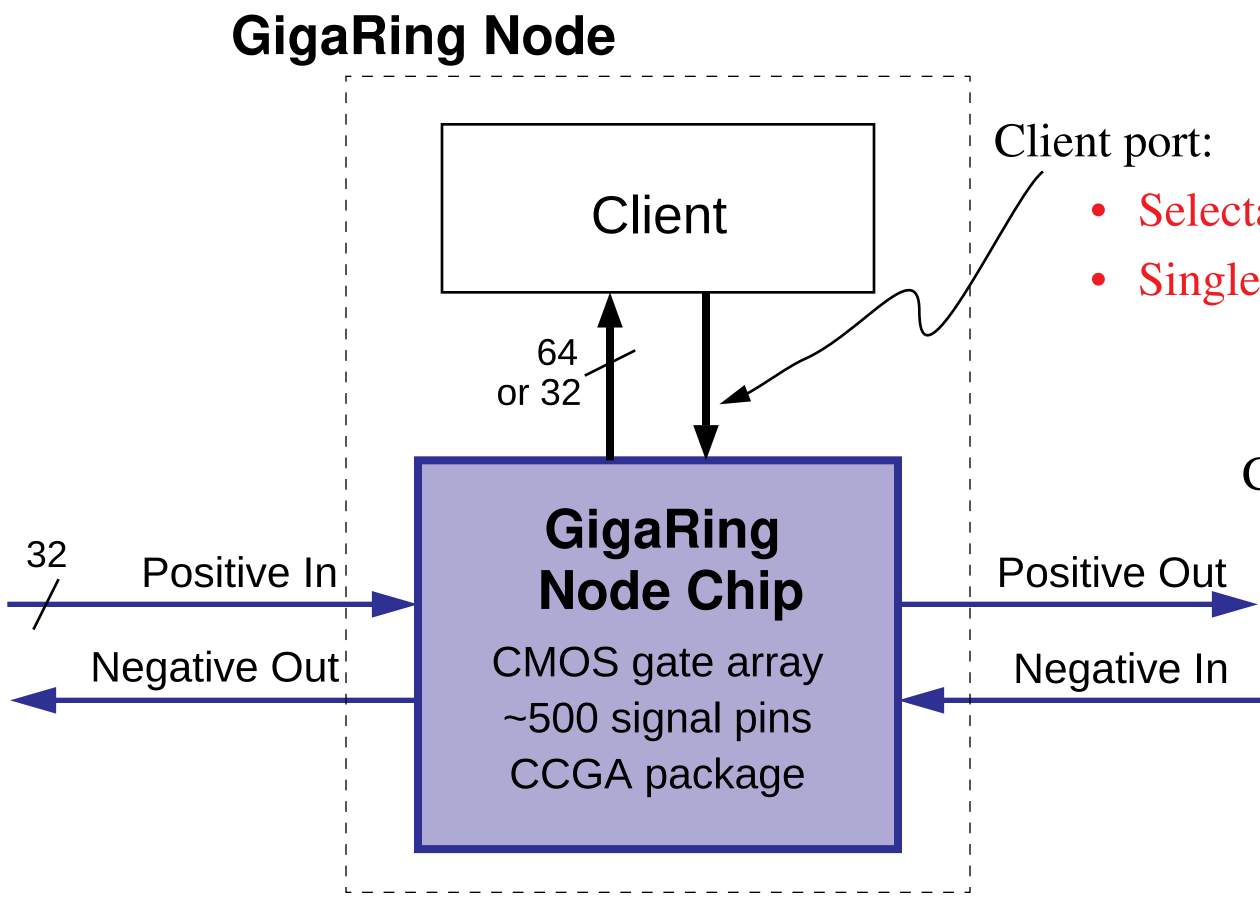
Outline of Talk

- Highlights

- Physical layer - *provides a physical communication medium*
- Logical layer - *efficiently delivers packets between clients*
- Protocol layer - *standard for communication: DMA, messages, etc.*

- Performance
- Status and future directions

Physical Layer: The GigaRing Node Chip



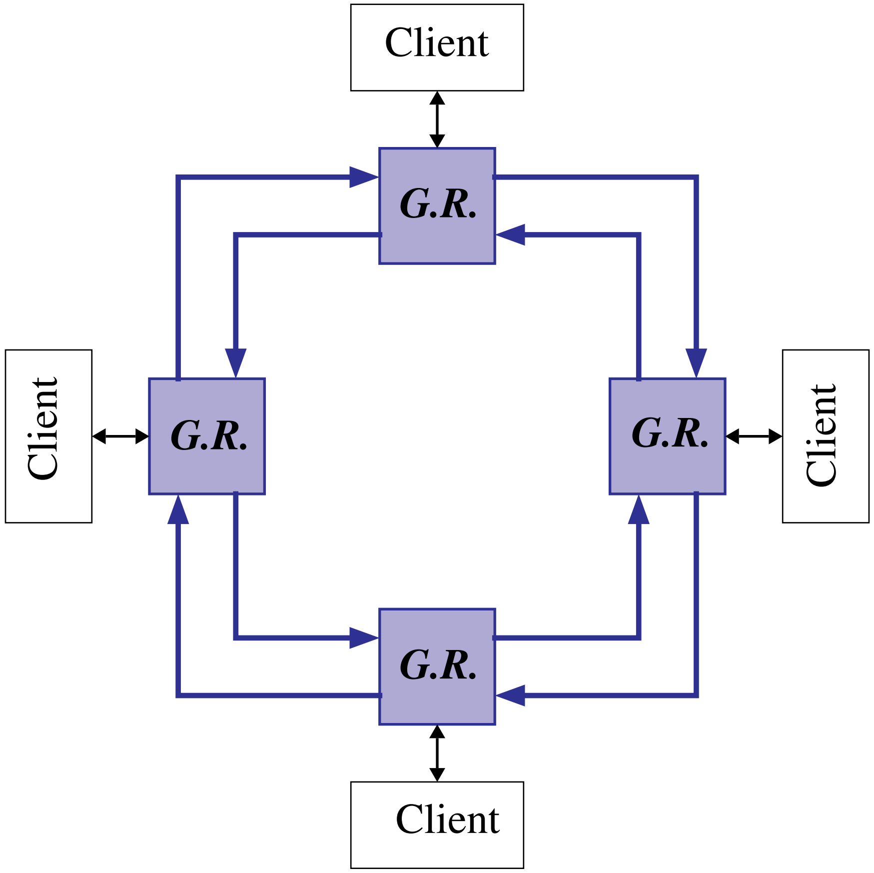
Client port:

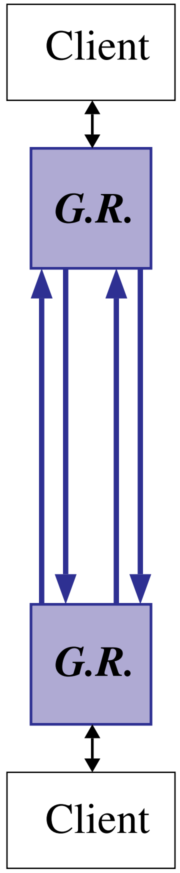
- Selectable speed and width
- Single ended

Channel:

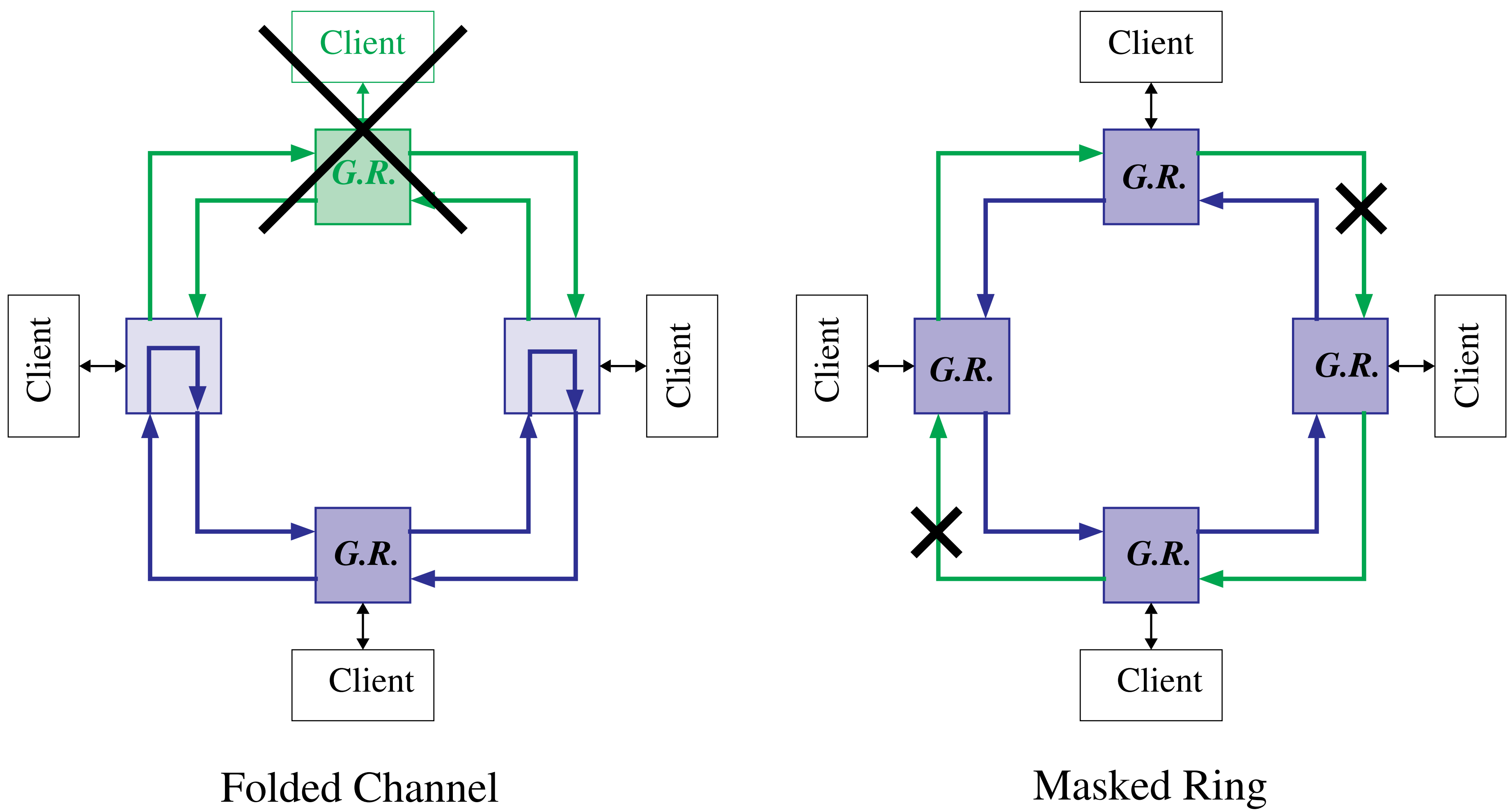
- AC coupled, differential
- Data encoding for
  - DC balance
  - switching frequency
- 150 Mb/s (75 MHz clock)
- Plesiochronous interface
- 11 meter external cables
- 200 meter fiber extender

GigaRing Channel Connections

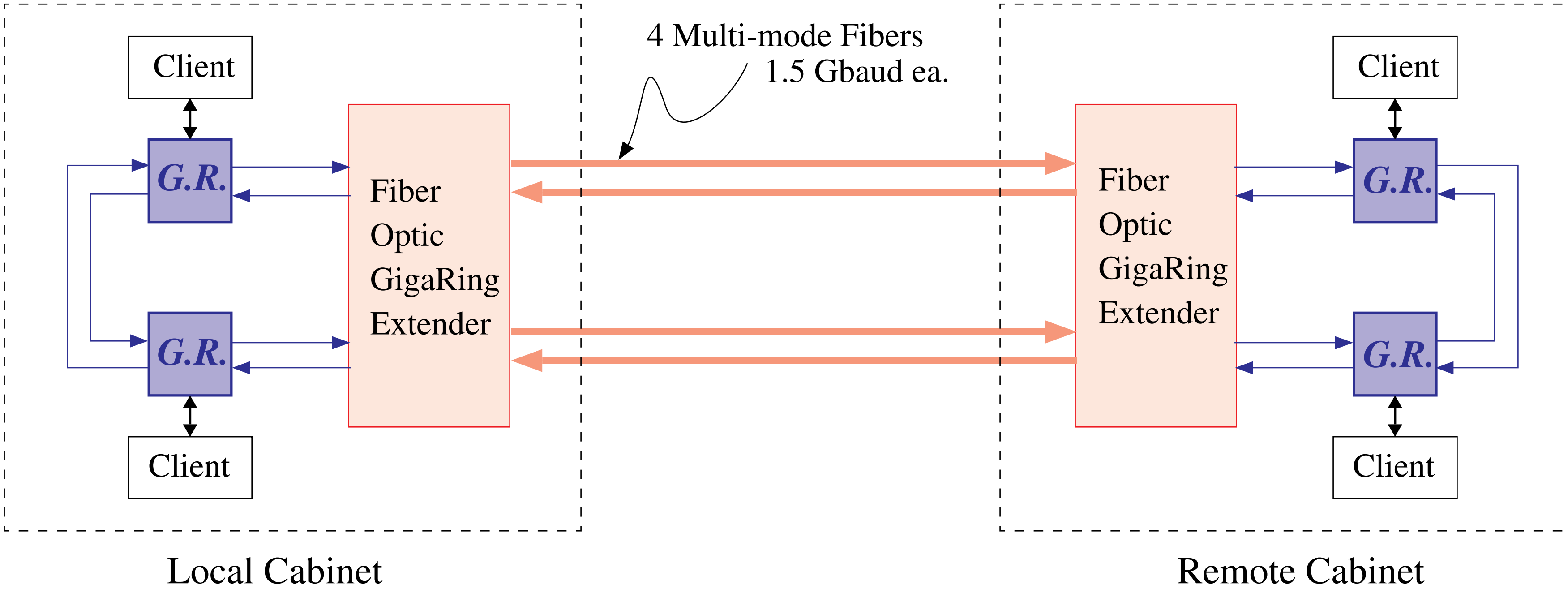




Reconfigured GigaRing Channels



Fiber-Optic Extender



- Transparently inserted between node chips
- Allows up to 200 meter separation

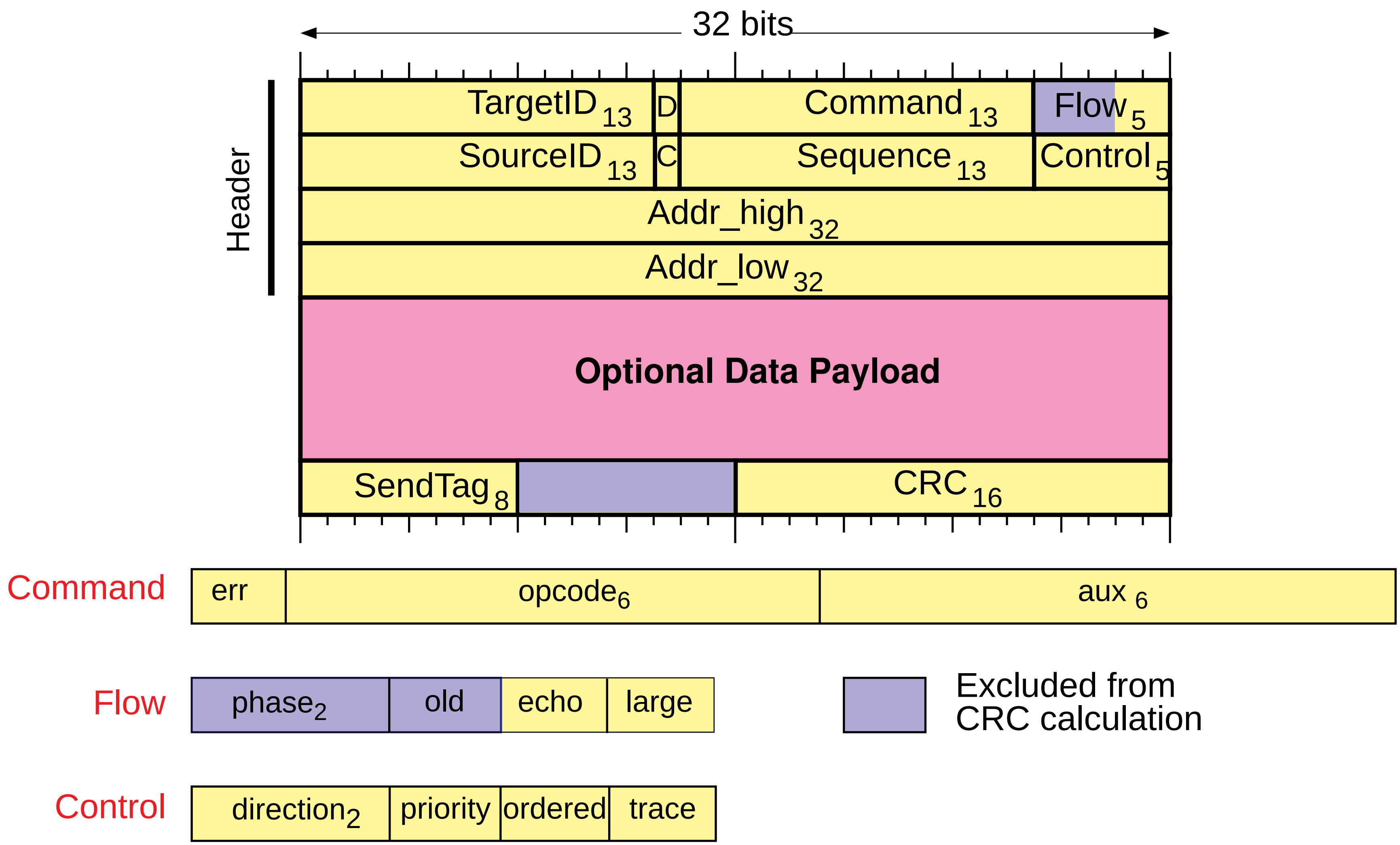
GigaRing

Logical Layer Features

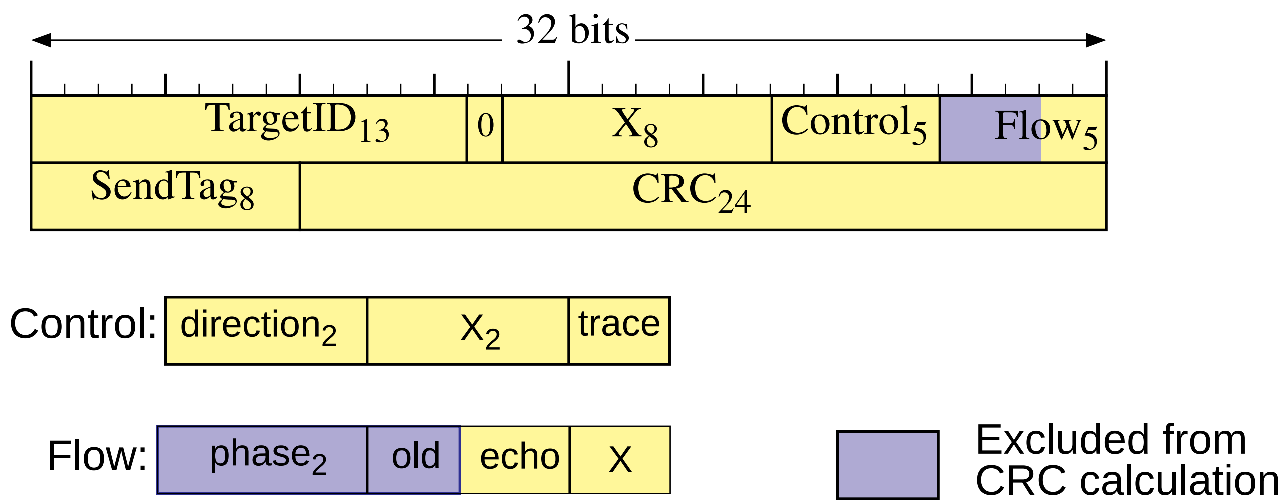
- Dual counter-rotating rings for fault tolerance and hot swap
- SCI/GigaRing arbitration protocols
  - fair bandwidth allocation
  - forward progress of all traffic
- Multiple virtual channels
- Adaptive congestion control mechanism
- Support for multiple channels with bridges
- Protection mechanisms for channel and client security
- Ring scrubber for routine channel maintenance and error detection

Basic service: deliver packets efficiently from one client to another

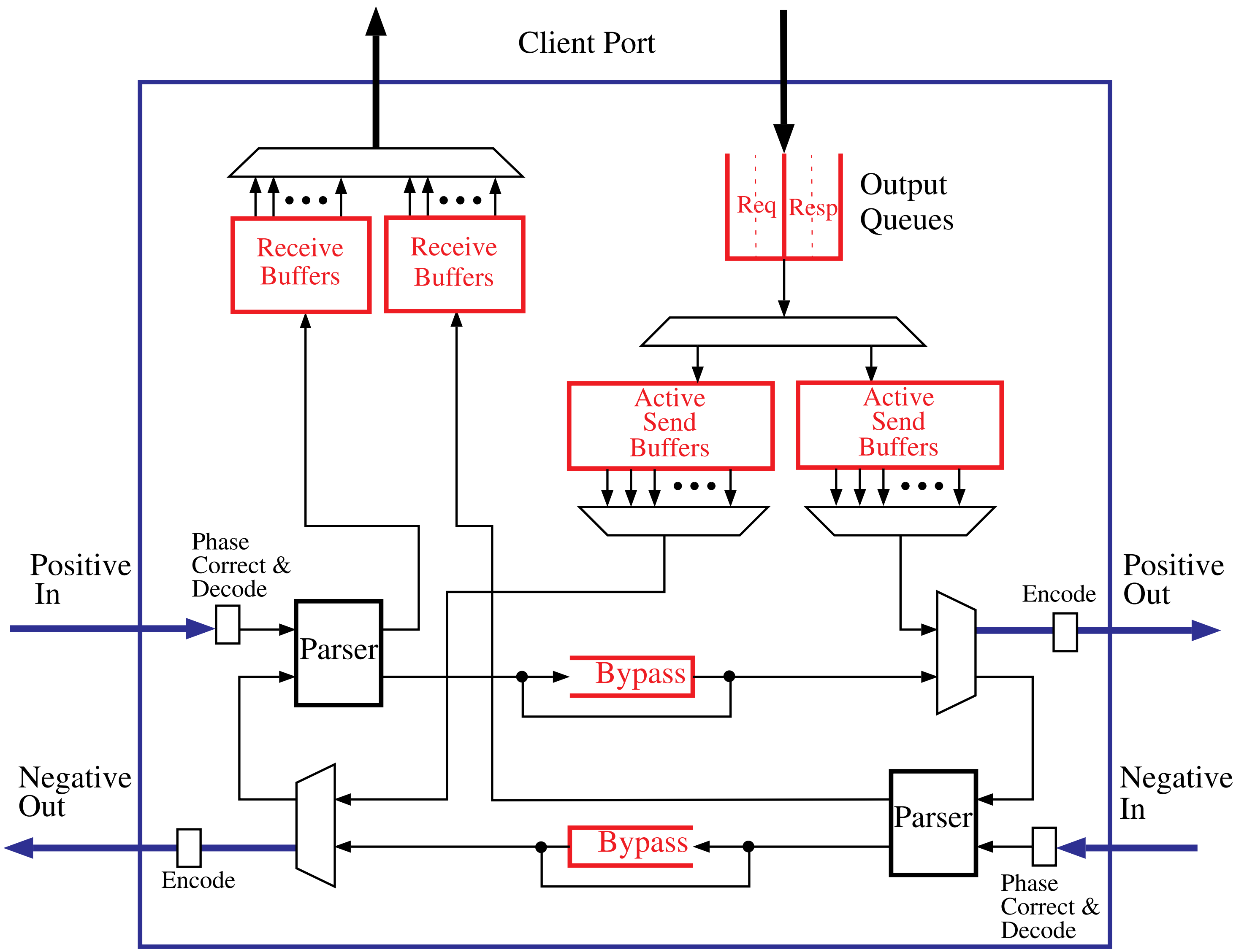
Send Packet Format



Echo and Idle Formats



GigaRing Node Chip Block Diagram



SCI/GigaRing Arbitration Protocols

Bandwidth allocation:

- When a node strips a packet, it creates bandwidth for transmission
  - ⇒ A node that *receives* more traffic gets more opportunities to *send*
  - ⇒ A node could starve or hog the ring bandwidth
- Go-bit arbitration protocol assures fair bandwidth allocation under contention

Queue reservation:

- A node could be unlucky and have its packets repeatedly busied
  - ⇒ No guarantee of forward progress (could livelock)
- Queue reservation protocol periodically assures older, busied packets will be accepted before new packets
- Applied independently to each VC at each receiver

Flow Control Overview

GigaRing

- No back-pressure on channel links
  - ⇒ End-to-end flow control only
- Busying send packets provides basic flow control, but:
  - - Congested streams hog send buffers and waste channel bandwidth
- Four virtual channels used over client port
  - - Read requests, read responses, write requests, write responses
  - - Send and receive buffers flexibly partitioned among VCs
  - - Decouple independent, dissimilar streams
- Adaptive client stream congestion control
  - - Speed negotiation on block transfers establish initial transfer rate
  - - Feedback allows master to monitor request *and* response congestion
  - - Master automatically adjusts transmission rate to adapt to congestion
  - - Fully software tunable

Adaptive Congestion Control

GigaRing

- For each stream, master monitors and limits:
  - Number of requests on the channel (max = Nc)
  - Number of requests in the system (max = Ns)
- Feedback provided by channel:
  - *ClientEcho*: request has been delivered, busied?
  - Response packet: response has been delivered, busied?
- Master uses feedback to adjust transmission
  - If requests are busied too much, then reduce Nc

    (works well for *write* DMA transfers)
  - If responses are busied too much, then reduce Ns

    (works well for *read* DMA transfers)
  - As congestion clears, ratchet rates back up again
- Software can tune the mechanism at each node
  - Operating point
  - Sensitivity

Client Port Protection

GigaRing supports six levels of isolation between a client and the rest of the channel:

1. ISOLATE: Total isolation from channel.
2. STANDBY: Total isolation, save that client can unlock port.
3. MAINTENANCE: Allow *MaintReadResp* and *MaintWriteResp* packets only.
4. SLAVE: Allow response packets only.
5. USER: Allow all traffic except *MMR_Writes*di and client maintenance requests.
6. MASTER: Full permissions.

- Illegal packets from a client are discarded
- GigaRing node chips power on in the STANDBY state
- Channel master unlocks its port, initializes the channel, optionally initializes other clients, and unlocks their ports as desired.

Memory-Mapped Registers

GigaRing node chip MMRs

- Used to configure the channel, set parameters, access state, etc.
- Accessible via *MMR_Read* and *MMR_Write* packets
- Simply send the appropriate node chip an MMR packet, and it performs the desired operation and returns an MMR response packet

Client MMRs

- Accessible via *MaintRead* and *MaintWrite* packets
- Handful of GigaRing-defined registers
  - Client information
  - Tuning registers for congestion control mechanism
- Can be used for any sort of client maintenance
  - Booting, diagnosis, scan control, memory testing, etc.

Features/Functions Accessible via MMRs

- Channel initialization
- Setting "trust" to isolate clients or groups of clients from each other
- Client isolation for protection or on-line client diagnosis/maintenance
- Channel folding for live node insertion, deletion or swap
- Ring masking for fault tolerance or during reconfiguration
- Error diagnosis
- Remote client reset
- Changing channel operating parameters (*e.g.*: buffer partitioning)
- Reading channel status
- Using performance monitor
- Setting up bridges
- Etc.

IO Protocol

Name	Description
MsgPkt	Send a self-contained message with a payload of 0-32 64-bit words to another client
ReadBlkInit	Set up transfer from target to source
ReadBlkInitResp	Acknowledge ReadBlkInit and negotiate speed or buffer requirements
Read	Request 1-32 64-bit words from target memory
ReadResp	Return Read data
ReadBlkDone	Notify completion of ReadBlk
ReadBlkDoneResp	Acknowledge ReadBlkDone
WriteBlkInit	Set up transfer from source to target
WriteBlkInitResp	Acknowledge WriteBlkInit and negotiate speed or buffer requirements
Write	Transfer 1-32 64-bit words to target memory
WriteResp	Acknowledge receipt of Write
WriteBlkDone	Request completion of WriteBlk
WriteBlkDoneResp	Acknowledge WriteBlkDone, signalling completion of WriteBlk

IO Example

- Program issues a *write*() system call
- OS sends a *MsgPkt* to a disk I/O node describing desired operation
- Software on ION processes the MsgPkt, and fires up ION hardware
- ION performs a *ReadBlk* transaction from mainframe to ION, writes the data out to disk, then notifies the ION software
- ION software sends OS a *MsgPkt* with completion status
- OS processes *MsgPkt* and returns to user (assuming synchronous IO)

Comparison to SCI

Used as is:

- Basic ring arbitration, bandwidth allocation and queue reservation protocols

Modifications:

- Removed cache coherence layer completely
- Physical layer: slower wider paths, electrical isolation via AC coupling, greater error protection, tunable client port, etc.
- Packet formats: larger payload size, larger intra-node address, enhanced control, new commands, integrated error reporting, etc.
- Lots of buffering for high-performance, heavily pipelined transfers

New features:

- Counter-rotating rings for fault tolerance, hot swap and performance
- A variety of protection mechanisms for fault isolation and on-line maintenance
- Multiple virtual channels and end-to-end congestion control mechanism (both software configurable)
- MMR interface for controlling remote node chips and clients

Performance

Bandwidth

- "Raw" BW between nodes: 1200 MB/s
- Encoding for DC balance and plesiochronous interface

  ⇒ 1059 MB/s
- Protocol overheads (depend on traffic patterns and transfer types)

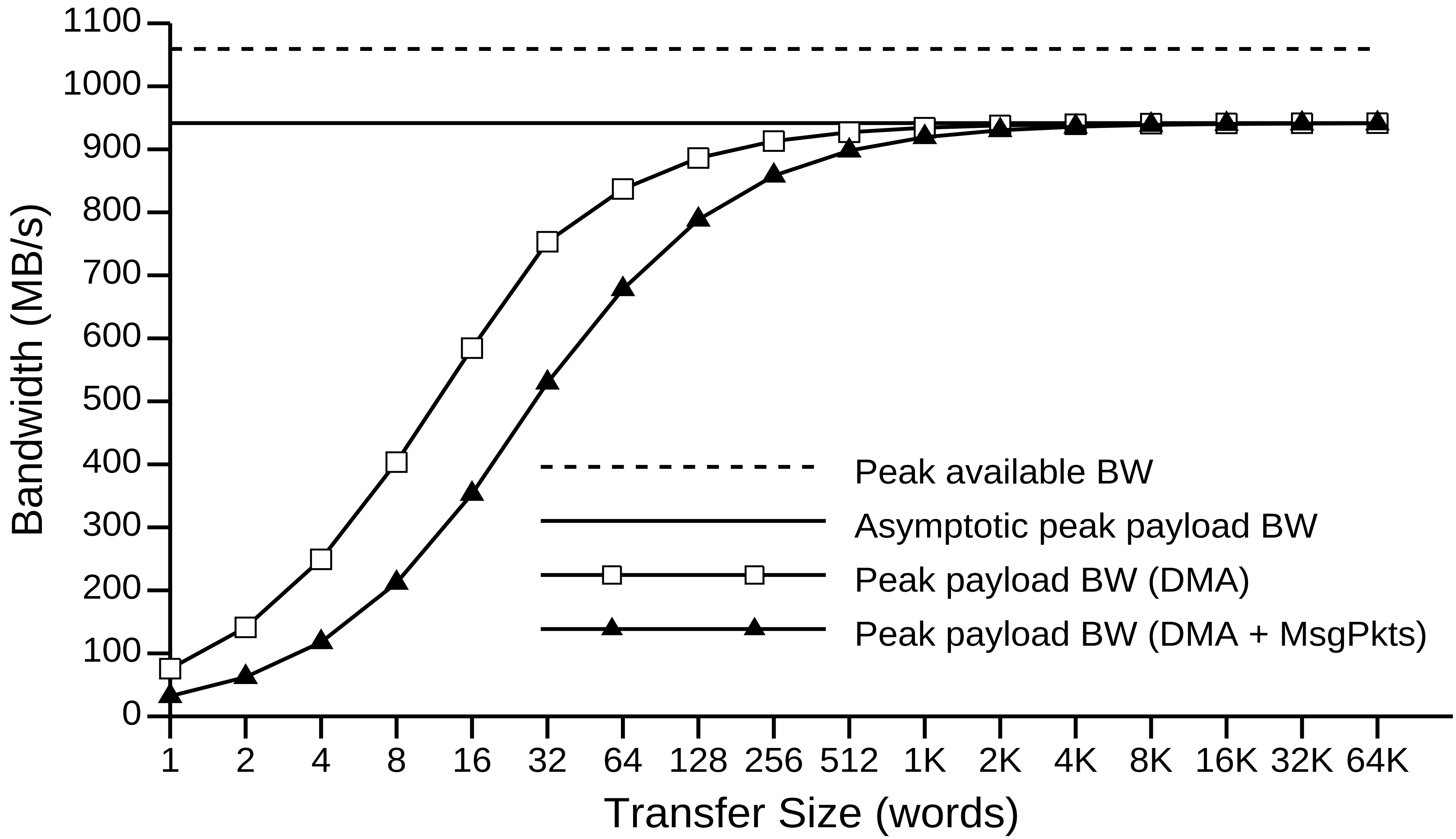
  ⇒ Half duplex client to client payload BW = 941 MB/s

  ⇒ Full duplex client to client payload BW = 2×827 MB/s
- Peak client port BW ranges from ~1/6 to 1 times full channel BW

Latency

- Per-hop latency ~ 225 ns
- DMA hardware latency ~ 10 - 15 µs + transfer time

Performance



Current Status and Future Directions

- Prototype version (incomplete functionality)
  - design completed in July 1995
  - parts received in November 1995
- Production version released to layout in October 1995
  - will support final system verification and product build in 1Q 1996
- First products to use GigaRing will be T3E and Scalable I/O system in 1H 1996

- Second generation GigaRing will likely focus on:
  - lower cost and power
  - higher bandwidth *or* equal BW with thinner links
  - additional functionality
  - input from standards activities
- Plan to make GigaRing an open standard