A Programmable Routing Controller for Flexible Communications in Point-to-Point Networks

Stuart Daniel, Jennifer Rexford, James Dolter,

and Kang Shin

Real-Time Computing Laboratory

The University of Michigan

http://www.eecs.umich.edu/RTCL

Outline

1. Motivation for flexible network policies
2. Introduction to routing and switching
3. Implementation of flexibility
4. PRC architecture
5. Conclusions and future work

Why a Flexible Router?

Applications demand different "qualities of service":

- Average communication latency
- Throughput
- Predictability

Applications generate distinct traffic patterns:

- Packet size
- Destination distributions
- Interarrival times

$\longrightarrow$ Programmable Routing Controller (PRC)

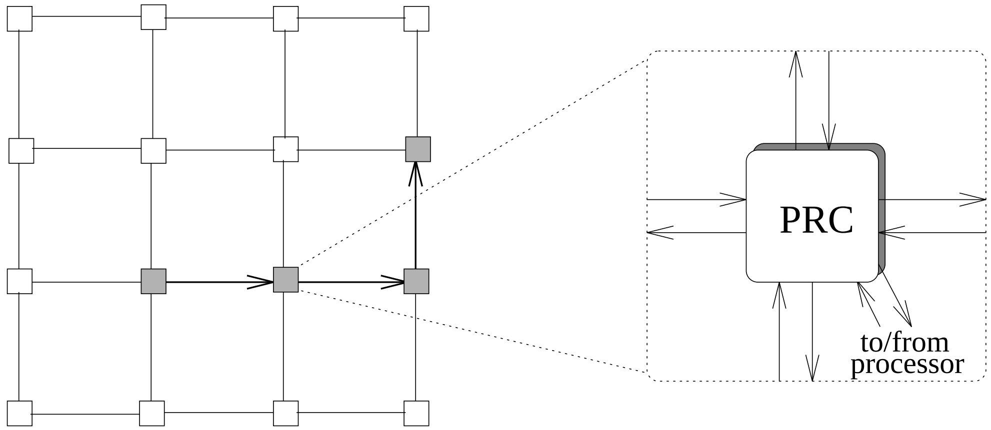
Router Policies

Network performance is sensitive to the policies chosen for:

Routing: Which links a packet traverses between its source and destination

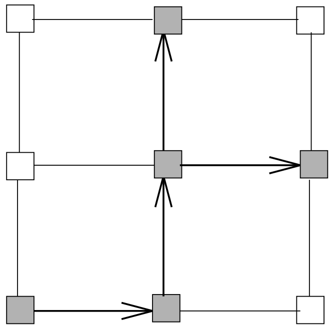
Switching: How the packet flows through intermediate nodes.

Selection: *Order* in which to consider potential links.



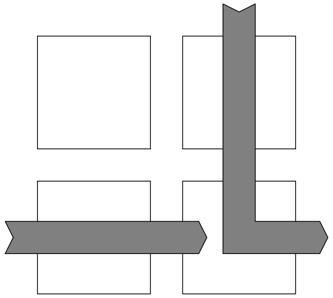
PRC Routing Schemes

- Static routing
  → in-order delivery, predictable resource usage
- Adaptive routing
  → congestion-avoidance, fault tolerance
- Multicast
  → efficient group communication
- Irregular topologies
  → general networks, reconfiguration

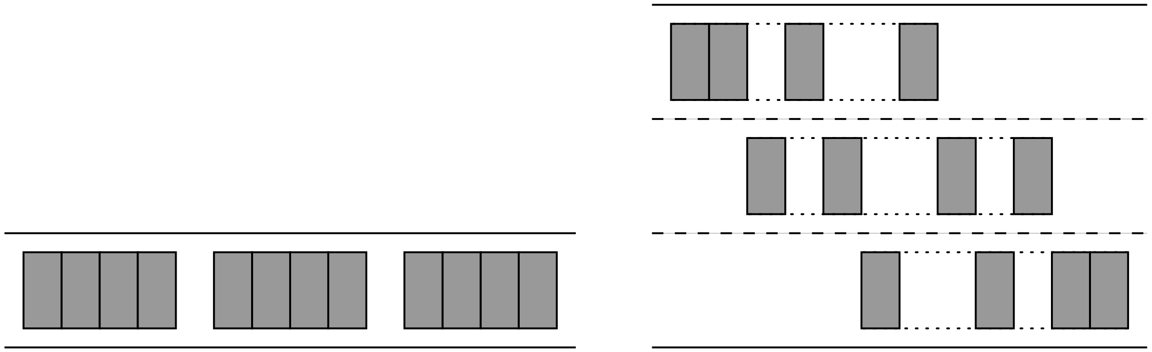


PRC Switching Schemes

- Circuit switching
  → guaranteed network access
- Packet switching
  → control over scheduling and bandwidth allocation
- Virtual cut-through switching
  → low latency
- Wormhole switching
  → low latency, reduced buffer requirements



Virtual Channels



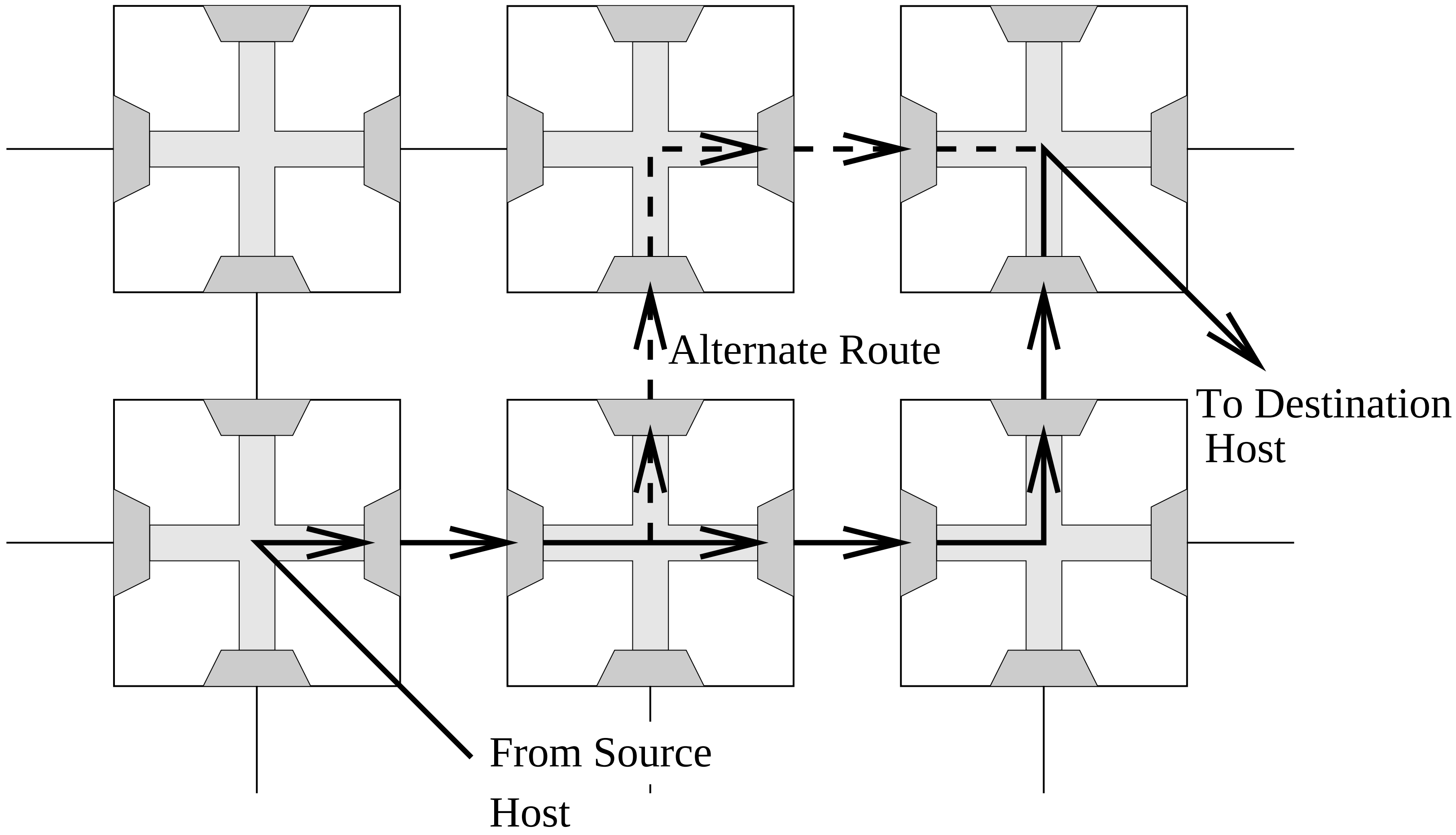
(a) Single physical channel

(b) Virtual channels

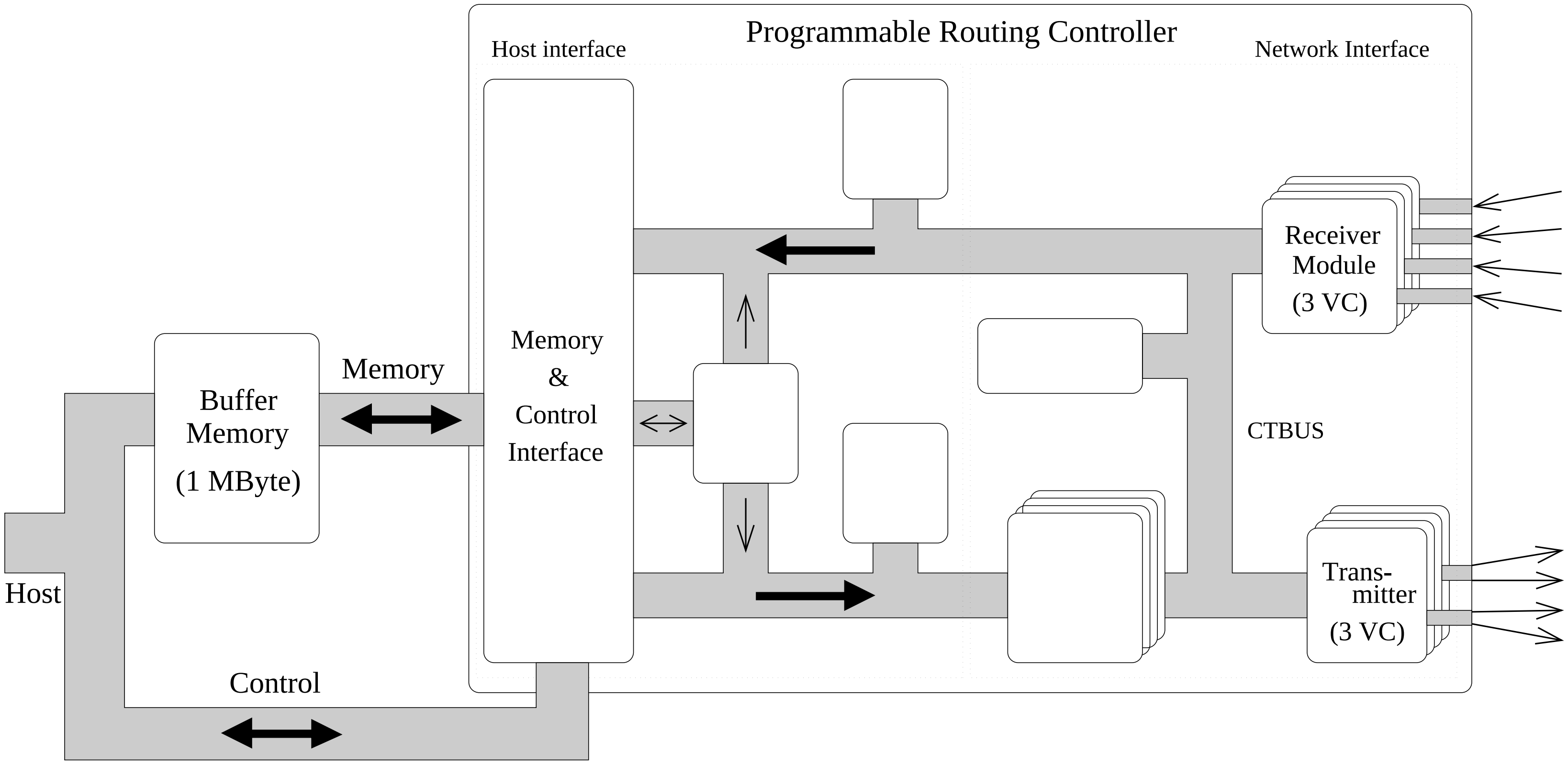
- Deadlock prevention in wormhole-switched networks
- Improving throughput
- Traffic partitioning

⟶ PRC provides 3 virtual channels per link

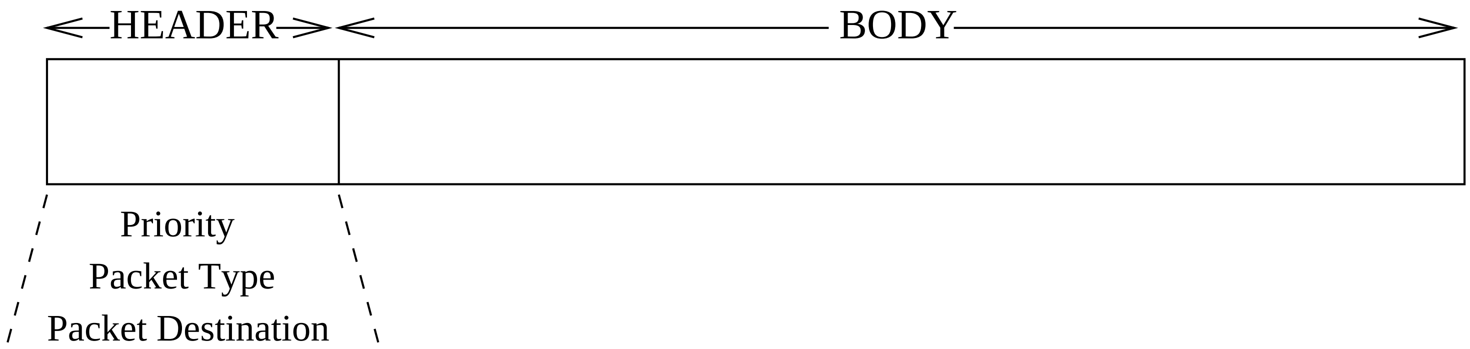
Where to Locate Flexibility



PRC Architecture

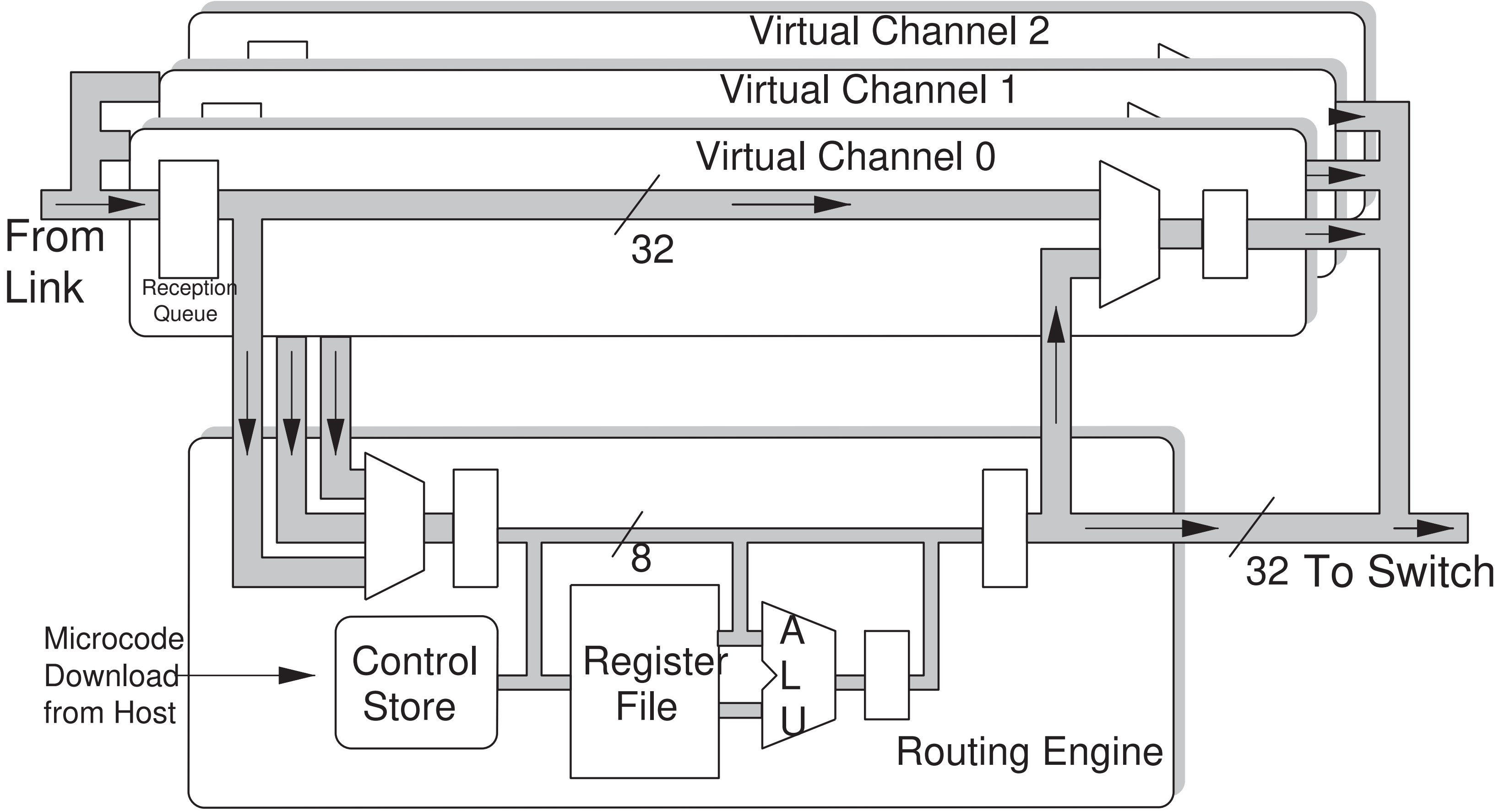


Flexible Routing and Switching



- Parsing packet header
- Reserve transmission channel(s)
- Data transfer

Receiver Module Architecture



Routing Primitives

Address: Bit mask of destination channels, including memory port.

Data: First word of data for the packet — usually, the modified packet header.

Make Reservations: Signals whether channels are already reserved. If false, the state machine will wait until it can reserve channel(s).

Multicast: Selects either multicast (all of) or unicast (only one of) semantics.

Routing Engine Instruction Set

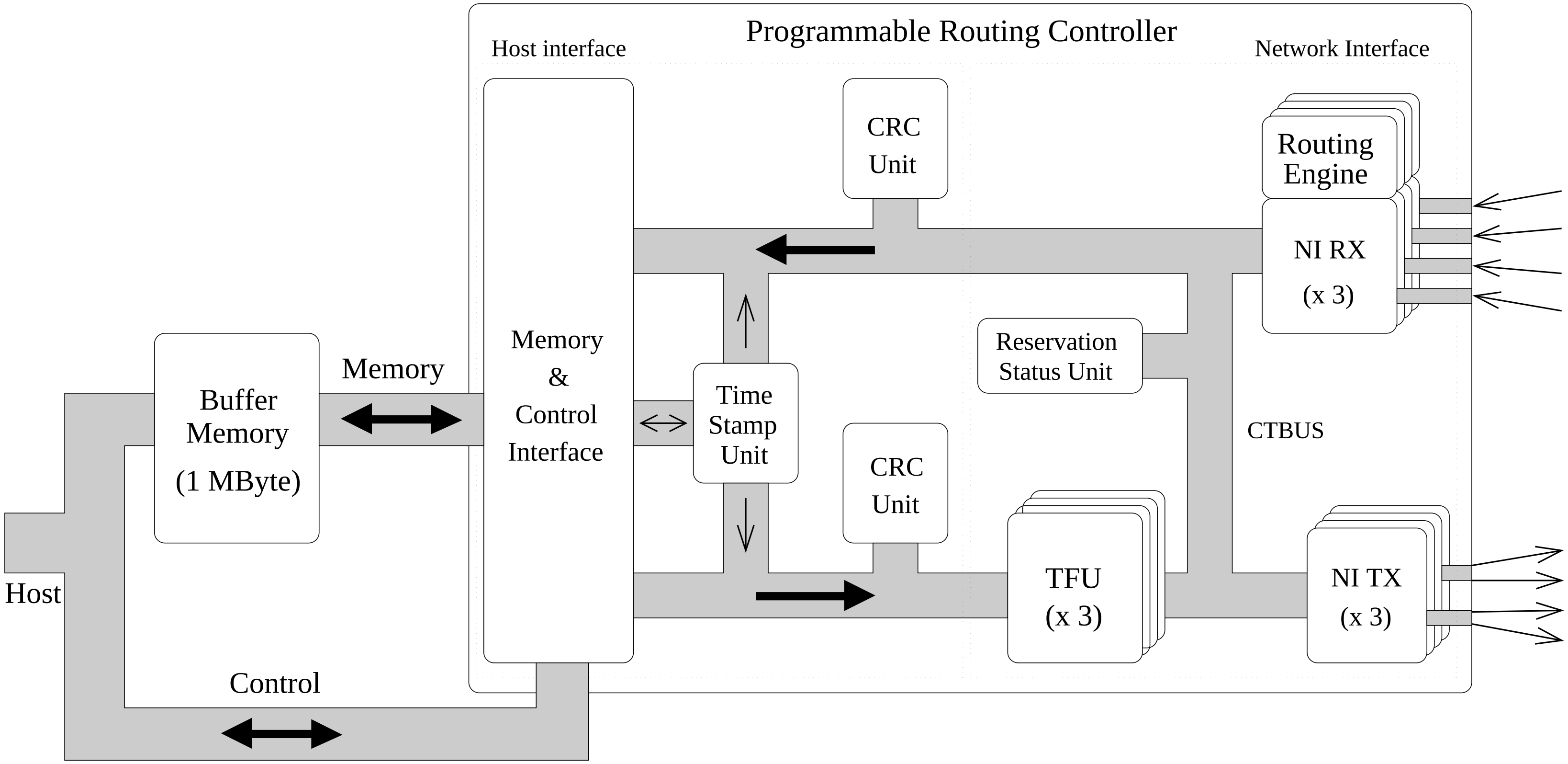
Data Movement: data transfer, load constant, wait for data

Header Parsing: boolean flags, addition, subtraction, boolean operators

Control Flow: conditional branch, subroutines

External Interface Control: trigger routing primitive, access switch

PRC Architecture



CTBUS

Channel reservations

- CTBUS accesses used to reserve transmission channels
- Requests serialized by CTBUS arbitration; allows efficient implementation
- Reservation status can be "snooped" without CTBUS access

Data Transfer

- Interconnects receivers, transmitters, and memory ports
- Demand-slotted access
- Operates at twice the speed of links
- Peak bandwidth equal to that of links

External Interfaces

Network:

- 4 pairs of unidirectional links
- Links are parallel, and support serial via AMD TAXI chips
- One byte of data and 2 bits control transmitted each link cycle

Host:

- *Control*: Asynchronous, 32-bit interface allows host to monitor and control PRC operation
- *Memory*: Synchronous, 32-bit interface to static RAM buffer

Host Interface

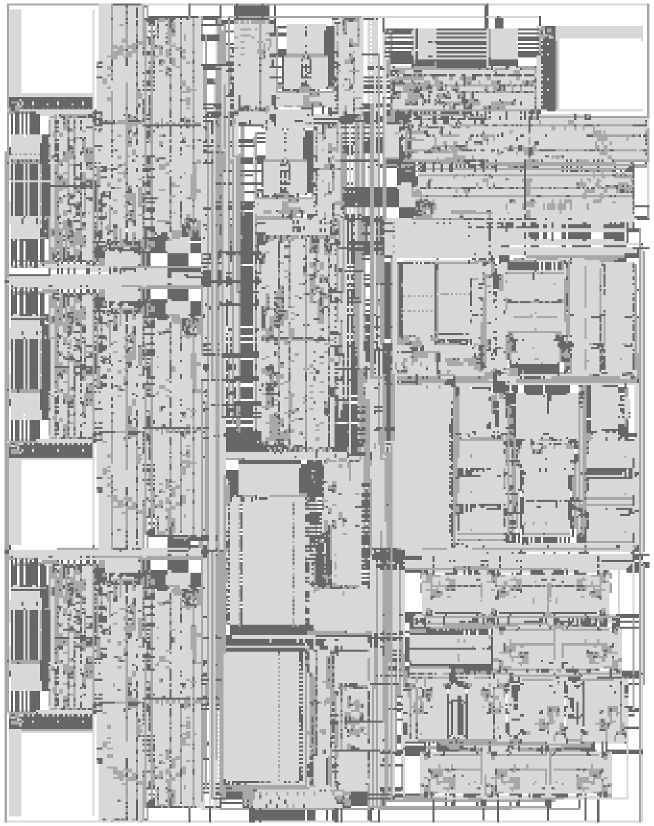
Host exercises *page*-level control over SPIDER:

- Page tag queues for each transmission/reception channel
- Event queue logging page-level activities

Flexibility:

- Scatter-gather DMA for each channel
- Variable packet size
- Interrupt masking
- Transparent error detection
- Transparent timestamping of packets

PRC Layout



Process: 0.5u HP CMOS

Size: 9.1 x 8.0 mm

Transistors: 490,000

Power: 0.6 watts

Pins: 256

Speed: 50 MHz

Conclusions

- Flexible routing and switching
- Efficient programmable architecture
- Single-chip solution

Future Work

- Chip fabrication and testing
- Board and system level evaluation