Packet Switch Architectures
Part 2

Tutorial Outline

• Introduction:
  What is a Packet Switch?
• Packet Lookup and Classification:
  Where does a packet go next?
• Switching Fabrics:
  How does the packet get there?
• Output Scheduling:
  When should the packet leave?
Switching Fabrics

- Output and Input Queueing
- Output Queueing
- Input Queueing
  - Scheduling algorithms
  - Combining input and output queues
  - Other non-blocking fabrics
  - Multicast traffic

Basic Architectural Components

*Datapath: per-packet processing*

1. Forwarding Table
   - Forwarding Decision
2. Interconnect
3. Output Scheduling
Interconnects

Two basic techniques

Input Queueing

Output Queueing

Usually a non-blocking switch fabric (e.g. crossbar)

Usually a fast bus

Interconnects

Output Queueing

Individual Output Queues

Centralized Shared Memory

Memory b/w = (N+1).R

Memory b/w = 2N.R

Copyright 1999. All Rights Reserved
Output Queueing

The “ideal”

- 5ns per memory operation
- Two memory operations per packet
- Therefore, up to 160Gb/s
- In practice, closer to 80Gb/s

How fast can we make centralized shared memory?
Switching Fabrics

- Output and Input Queueing
- Output Queueing
- Input Queueing
  - Scheduling algorithms
  - Other non-blocking fabrics
  - Combining input and output queues
  - Multicast traffic

Interconnects

*Input Queueing with Crossbar*

![Diagram of Interconnects]

Memory b/w = 2R

Scheduler

Data In → Scheduler → Memory b/w = 2R → Data Out

configuration
Input Queueing

Head of Line Blocking

---

Head of Line Blocking
Input Queueing

Virtual output queues

Input Queues

Virtual Output Queues
Input Queueing

Scheduler

Memory b/w = 2R

Can be quite complex!

Input Queueing

Scheduling

Copyright 1999. All Rights Reserved
Wave Front Arbiter

Scheduling Algorithm

Requests → Match

Wave Front Arbiter

Requests → Match
Other Non-Blocking Fabrics

*Clos Network*

Expansion factor required = $2 - 1/N$ (but still blocking for multicast)
Other Non-Blocking Fabrics

Self-Routing Networks

The Non-blocking Batcher Banyan Network

Batcher Sorter

Self-Routing Network

• Fabric can be used as scheduler.
• Batcher-Banyan network is blocking for multicast.
Switching Fabrics

- Output and Input Queueing
- Output Queueing
- Input Queueing
  - Scheduling algorithms
  - Other non-blocking fabrics
  - Combining input and output queues
  - Multicast traffic

Speedup

- Context
- input-queued switches
  - output-queued switches
  - the speedup problem
- Early approaches
- Algorithms
- Implementation considerations
Speedup: Context

A generic switch

The placement of memory gives
- Output-queued switches
- Input-queued switches
- Combined input- and output-queued switches

Output-queued switches

Best delay and throughput performance
- Possible to erect “bandwidth firewalls” between sessions

Main problem
- Requires high fabric speedup \( (S = N) \)

Unsuitable for high-speed switching
Input-queued switches

Big advantage
- Speedup of one is sufficient

Main problem
- Can’t guarantee delay due to input contention

Overcoming input contention: use higher speedup

A Comparison

Memory speeds for 32x32 switch

<table>
<thead>
<tr>
<th>Line Rate</th>
<th>Memory BW</th>
<th>Access Time Per cell</th>
<th>Memory BW</th>
<th>Access Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 Mb/s</td>
<td>3.3 Gb/s</td>
<td>128 ns</td>
<td>200 Mb/s</td>
<td>2.12 µs</td>
</tr>
<tr>
<td>1 Gb/s</td>
<td>33 Gb/s</td>
<td>12.8 ns</td>
<td>2 Gb/s</td>
<td>212 ns</td>
</tr>
<tr>
<td>2.5 Gb/s</td>
<td>82.5 Gb/s</td>
<td>5.12 ns</td>
<td>5 Gb/s</td>
<td>84.8 ns</td>
</tr>
<tr>
<td>10 Gb/s</td>
<td>330 Gb/s</td>
<td>1.28 ns</td>
<td>20 Gb/s</td>
<td>21.2 ns</td>
</tr>
</tbody>
</table>
The Speedup Problem

Find a compromise: $1 < \text{Speedup} \ll N$
- to get the performance of an OQ switch
- close to the cost of an IQ switch

If speedup $> 1$, then output buffers are Needed
→ Switch with Combined Input-Output Queueing (CIOQ)
Intuition

Speedup = 1
Bernoulli IID inputs
Fabric throughput = .58

Speedup = 2
Bernoulli IID inputs
Fabric throughput = 1.16

Intuition (continued)

Speedup = 3
Bernoulli IID inputs
Fabric throughput = 1.74

Speedup = 4
Bernoulli IID inputs
Fabric throughput = 2.32
The Ideal Solution

Question: Can we find
- a simple and good algorithms
- that exactly mimics output-queueing
- regardless of switch sizes and traffic patterns?

What is exact mimicking?

Apply same inputs to an OQ and a CIOQ switch
- packet by packet

Obtain same outputs
- packet by packet