Switch Design Goals

• **Throughput – performance advantage**
  - Limited by contentions and variable packet sizes
  - Incorporation various Traffic models

• **Size or scalability**
  - With the goal of tens of thousands of ports

• **Cost**
General Organization

Input port

Processor

Output port

Input port

Fabric

Output port

Input port

Buffer

Output port

Input port

Buffer

Buffer
Packet Movement

Input port

Processor

Output port

Input port

Fabric

Buffer

Input port

Buffer

Input port

Buffer

Output port

Output port

Output port

Output port
Buffering at the Input Port

Switch

Port 1

Port 2
Head-of-Line Blocking

- For uniformly distributed traffic, throughput reduced to 59% of theoretical maximum (sum of the link BWs)
- So - use output or fabric buffering
Design Options to Shift Buffering Point

• **Shared bus**
  - Bus bandwidth determines the throughput
  - So build special busses

• **Shared memory**
  - Memory BW determines throughput
  - Build special high-speed memory and memory bus

• **Cross-bar**
  - Each output port accepts packets from all input ports *at once*
  - So each port has a BW = total switch throughput
Crossbar

- Cost
- Memory B/W requirement
  → non-blocking design
Crossbar with Redundancy

- Every-port is duplicated

⇒ Hence the design is robust but costly
Multi-Stage Banyan Network

Switch elements

Switching stage

Switching matrix

Input Ports

Output Ports
The self-routing is internal to the switch - it is used to minimize contention of ports.
Blocking in Multi-Stage Switches

• When multiple connections going to the same output port
  - Buffer
  - Increase switching speed
Attempts to Minimize Blocking

Module

Two 2-shuffles

Input buffer

or

Output buffer

Banyan

Baseline

Omega

Flip
A Banyan Network

Blocking occurs if two incoming packets want to go to the same output terminal

- Solution, include buffer in every switching element → expensive
- To avoid packet loss, buffer size must be large enough to hold the worst case pattern

Since buffering is an expensive solution first try to minimize *hot spots* (contention points)

Two techniques:

1. Use of *randomization* stage in *front* of Banyan switch
2. Use of *sorting* network in *front* of Banyan switch
Use of Randomization Stage

• Additional banyan switch is added in front - called the randomization stage.
  - When a cell arrives at an input port, its destination port
  - Is replaced by another (Random) destination port

• On leaving the randomization stage, each packet's original destination is restored

• Since randomization is statistical
  - this technique does not completely eliminate all hot spots
  - thus, buffers at each stage will still be required - but less expensive versions
Randomized Banyan Network

Ran = 011
(randomly selected)

001 110
Out = 100

Randomization

Switching

Out
Sorting - Batcher Banyan Network

- The technique is based on the observation that if the destination N Routing-vector is sorted (in increasing order) by output port, the corresponding N routes will be disjoint.

- With sorting the two routes have no links in common:
  - so both cells can be transmitted simultaneously
  - there will be no contention or need for buffers within elements

- Thus the Banyan network is preceded by a sorting network
- However, if two input cells with the same destination output port appear at the Batcher network, there will be contention.
  - So one of those cells cannot be allowed to enter the network and must be buffered.
  - Thus a Batcher-banyan switch also must be equipped with buffer...
Randomized Banyan Network

Sorted

Out = 110
Out = 111

Sorting
ATM LANs

LAN Emulation
ATM in LAN

• As backbone
• Interworking with other LANS
  - ATM not a broadcast technology
  - ATMARP and LAN Emulation
Protocol Stack for LAN Emulation

Higher-layer protocols (IP, ARP, ...)

Signalling + LANE

AAL5

ATM

PHY

Host

Ethernet-like interface

Switch

ATM

PHY

Host

Higher-layer protocols (IP, ARP, ...)

Signalling + LANE

AAL5

ATM

PHY

Host
Emulation Components

- LE clients
- Servers
  - Configuration servers (LECS)
  - Emulation Server (LES)
  - Broadcast and Unknown Server (BUS)