Switching

Lecture 7:

- switch architectures
- buffering
- queuing

What is it all about?

- How do we move traffic from one part of the network to another?
- Connect end-systems to switches, and switches to each other
- Data arriving to an input port of a switch have to be moved to one or more of the output ports

Types of switching elements

- Telephone switches
  - switch samples
- Datagram routers
  - switch datagrams
- ATM switches
  - switch ATM cells

Classification

- Packet vs. circuit switches
  - packets have headers and samples don't
- Connectionless vs. connection oriented
  - connection oriented switches need a call setup
  - setup is handled in *control plane* by *switch controller*
  - connectionless switches deal with *self-contained* datagrams

	Connectionless (router)	*Connection-oriented (switching system)*
Packet switch	Internet router	ATM switching system
Circuit switch		Telephone switching system

Other switching element functions

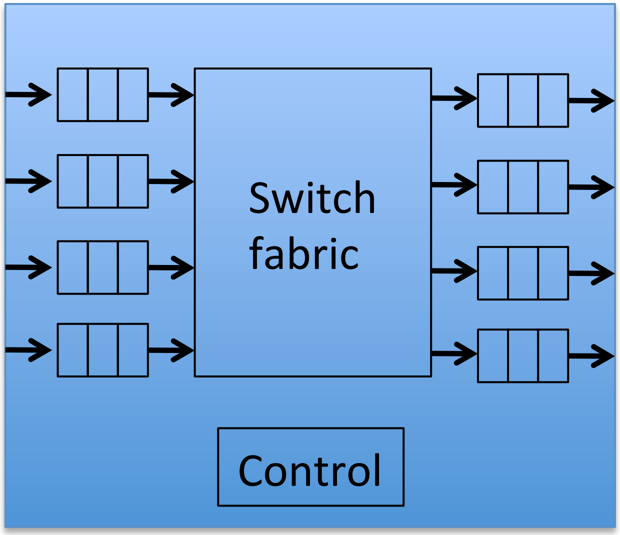
- Participate in routing algorithms
  – to build routing tables
- Resolve contention for output trunks
  – scheduling
- Admission control
  – to guarantee resources to certain streams

Requirements

- Capacity of switch is the maximum rate at which it can move information, assuming all data paths are simultaneously active
- Primary goal: maximize capacity
  - subject to cost and reliability constraints
- Secondary:
  - Circuit switch must reject call if can't find a path for samples from input to output
    - goal: minimize call blocking
  - Packet switch must reject a packet if it can't find a buffer to store it awaiting access to output trunk
    - goal: minimize packet loss
    - Don't reorder packets

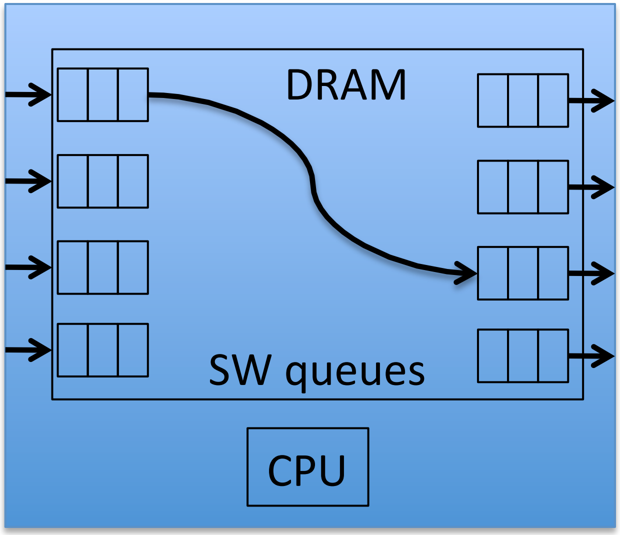
A generic switch

- Input buffers
- Output buffers
- A *line-card* has both input and output buffers and transmission interfaces
- *Switch fabric* or *Interconnect*
- Some processor for control funcitons (routing etc)



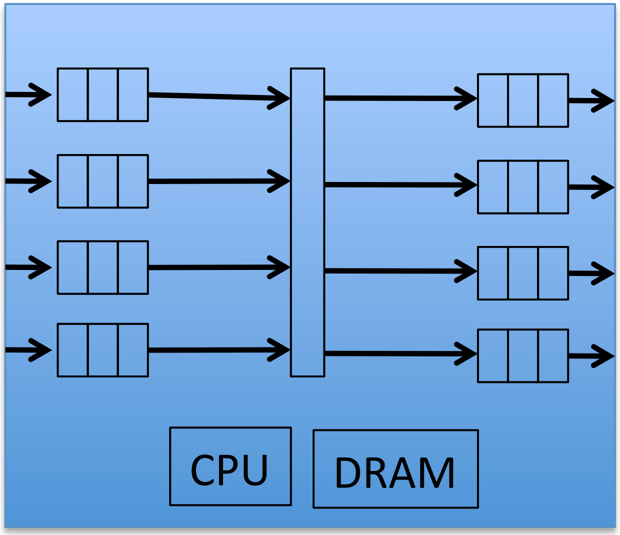
First generation

- All buffers in one simple memory sysem
- CPU makes forwarding decision and copies packets / manipulates queues
- Most simple Ethernet switches and cheap packet routers (e..g home routers)
- Bottleneck can be CPU, host-adaptor or I/O bus
- First Cisco routers were built sing 200MHz SPARC boards



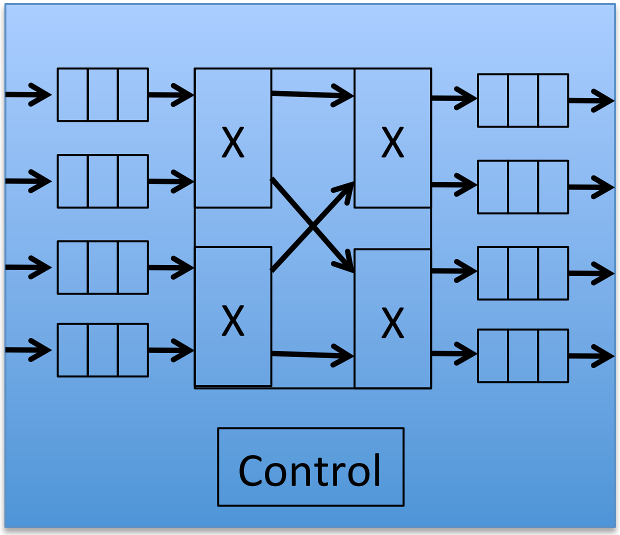
Second generation

- Input & output buffers on line cards
- Forwarding intelligence in line cards
- Simple bus interconnect
- CPU used to populate forwarding tables from routing protocol or connection set up
- FORE ATM switches used 2.4Gbps bus
- Bottleneck is bus

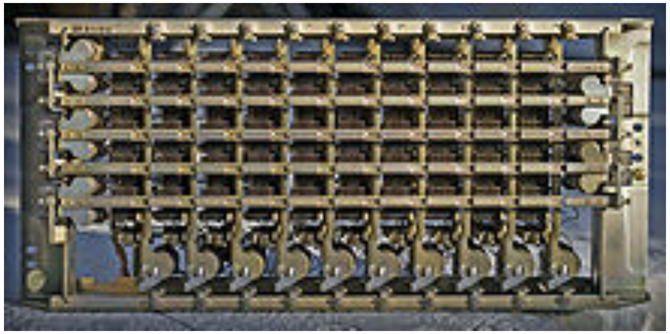


Third generation

- Simple bus does not scale well
- Need parallel paths
- Introduce interconnection networks

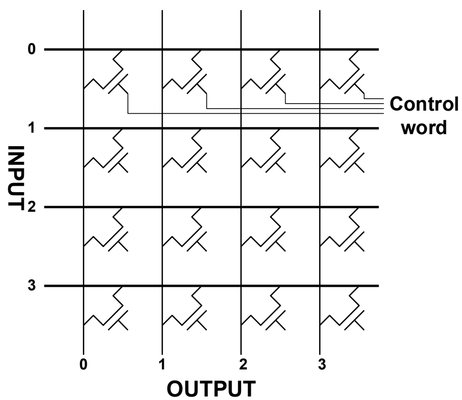


Crossbar



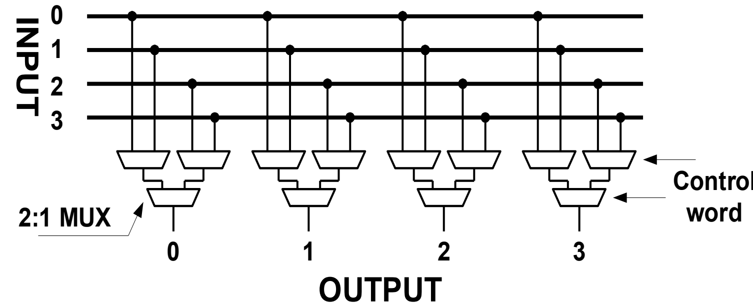
- Simplest possible space-division switch

- X-Y based crossbar



- Scalability : **N^2**
- Speed : **limited by Cap at input and output lines**
- Control: **N^2 bits**

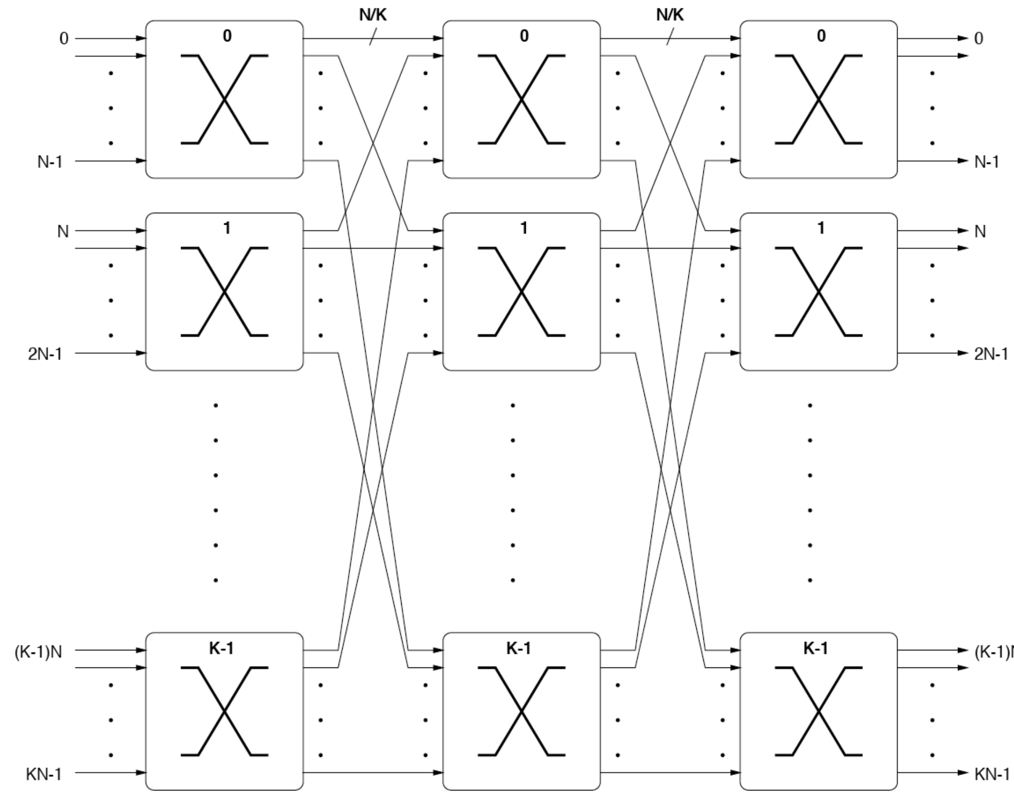
- Mux based crossbar



- Scalability : **N^2**
- Speed : **limited by Cap only at input line**
- Control: **$N*Log_2N$ bits**

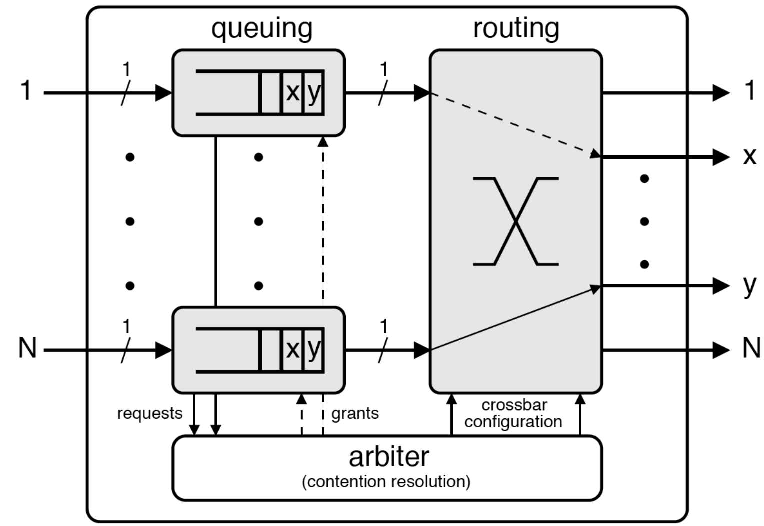
Multistage networks

- Build large fabrics from smaller crossbars
- What might be some problems here? (more later…)
- Number of crosspoints?
- What is input and output capacity… depends on buffering strategy



Input buffering

- One queue per input
- Input and output capacity is same as transmission line rate
- Needs *arbitration* to decide who wins
- Problem: *head of line blocking*
  - with randomly distributed packets, utilization at most 58.6%
  - worse with *hot spots*

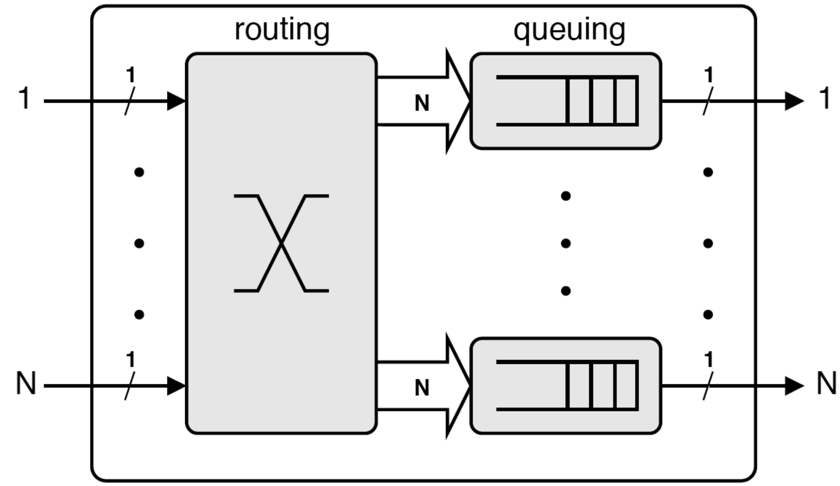


HoL - 58.6% utilization

Aggregate fabric capacity 2N

Output buffering

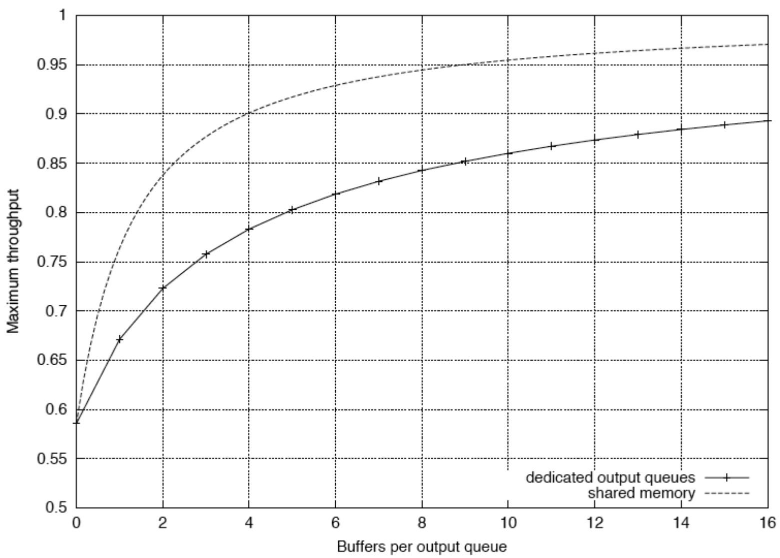
- One queue per output
  - No queues at input
- Input capacity is same as transmission line rate
- Output capacity is N times line rate
- Switch fabric output grows as N^2
- Can achieve 100% throughput

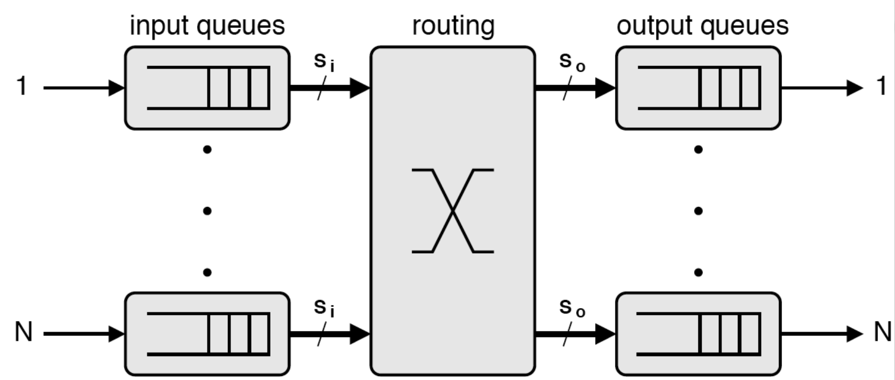


100% throughput

Aggregate fabric capacity is N(N+1)

CIOQ



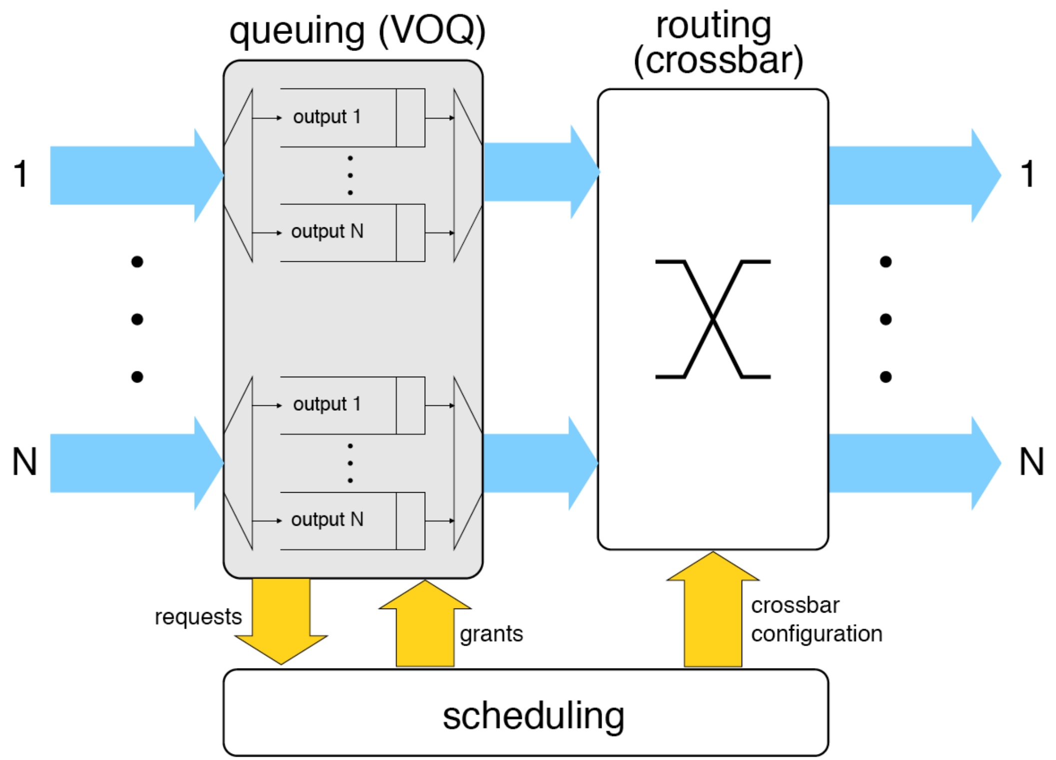


- CIOQ
  - Combined input and output queues
- Apply speed up S_i at input and S_o at output

Apply speed up S_i and S_o

Aggregate fabric $N(S_o + S_i)$

VOQ



- VOQ
  - Virtual output queues
- All the magic and complexity is in the arbitration…

As output queued

Aggregate fabric 2N

Blocking

- We have come across HoL

Where else?

- Internal blocking in complex switch fabrics
- Two packets need same internal link
- Dealing with Blocking:
  - Overprovisioning
    - internal links much faster than inputs
  - Buffers
    - at input or output
  - Backpressure
    - prevent packet from entering until path is available
  - Parallel switch fabrics
    - increases effective switching capacity