CS 552 Computer Networks

IP forwarding

Fall 2004

Rich Martin

(Slides from D. Culler and N. McKeown)

Position Paper

- Goals:
  - Practice writing to convince others
  - Research an interesting topic related to networking.
  - Generate reactions amongst fellow classmates/professors
- Size of the paper:
  - 4000-5000 words, 5-7 pages, 10 pt. Font
- Must have title and abstract
- Name on the paper is optional
- You will evaluate 2 papers
- You will revise your paper
- Hand in in PDF (preferred) or PS only
- Due Oct 8th

Evaluation Criteria

- Is the position well defined?
  - Is it narrow enough to be manageable?
  - Are the communities of people involved with the position (and their positions) identified?
- Are the opposing positions articulated?
- Are rebuttals given to the opposing positions?
- What evidence is used to support the position?
  - Quantitative evidence based on experimentation?
  - General facts about the systems in question?
  - Anecdotes only?
- Is the paper logically organized?
- Most importantly, does you paper influence someone on the position?

Position Topics I

- Peer to Peer technologies equals pirating.
  - (suggested by Thu Nguyen)
- SANs vs. LANs.
- Distributed hash tables (DHTs): What are they good for?
- Ipv4 is sufficient for the next 30 years.
- IP over direct links.

Position Topics 2

- Over-provisioning vs. QoS.
  - (Suggested by Badri Nath).
- Multicast vs. P2P.
- Mobile IP is dead.
- Wireless Ad-hoc networks.
- Information will be free.
- Others Possible (e.g. on GPL, on standards, security)
  - Must convince the instructor position is good.

Academic Integrity

- DO: think about the position
  - Helps if you pick a position you care about (at least a little bit)
- DO: write your own text
- DO: cite sources at points embedded in the next.
- DON'T rip/off copy text
  - Longer quotes (100-150 words) ok, IF properly done.
- Use papers and samples as models.

Outline

- Where IP routers sit in the network
- What IP routers look like
- What do IP routers do?
- Some details:
  - The internals of a “best-effort” router
    - Lookup, buffering and switching
  - The internals of a “QoS” router

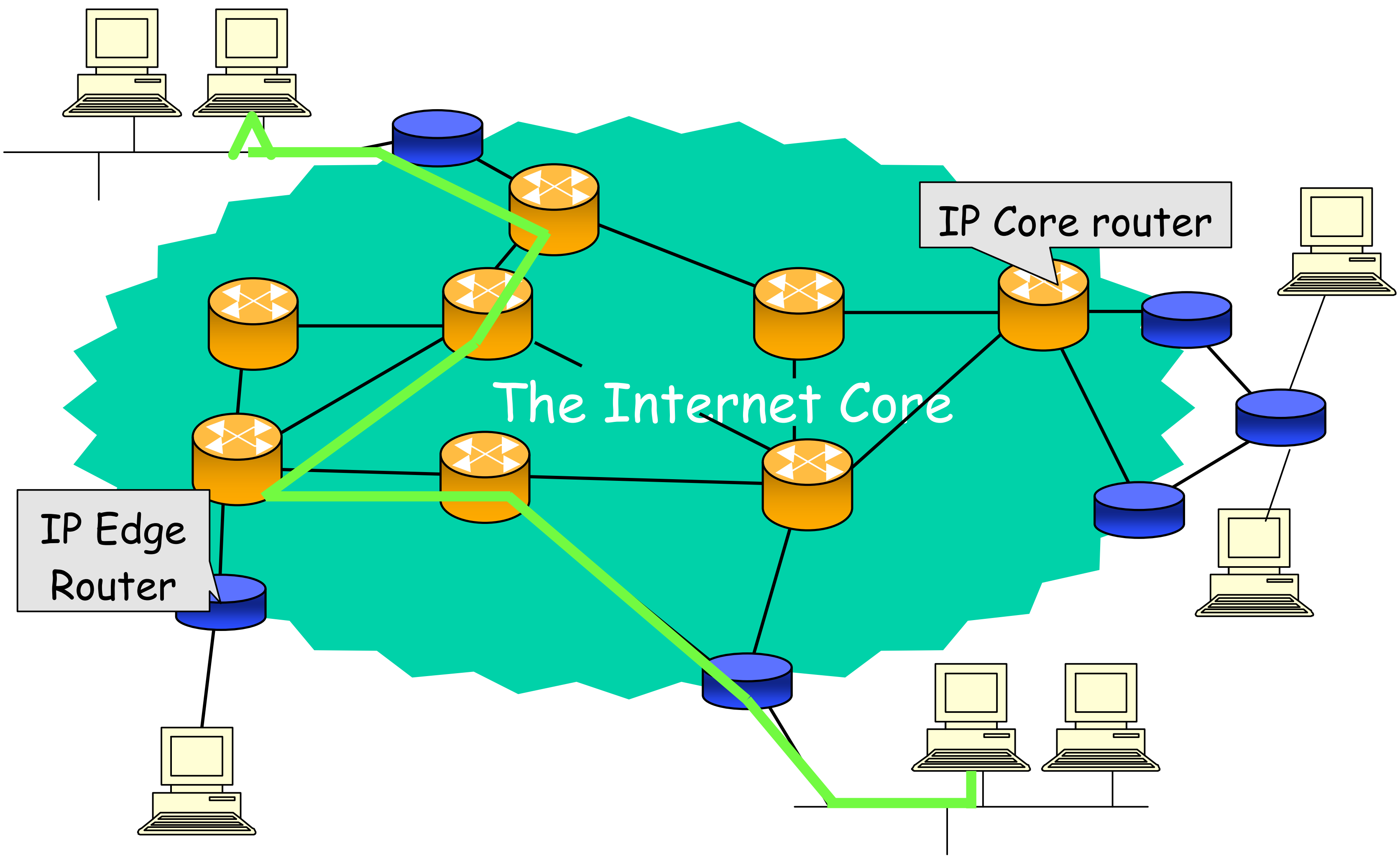
Outline (next time)

- The way routers are *really* built.
- Evolution of their internal workings.
- What limits their performance.
- The way the network is built today

Outline

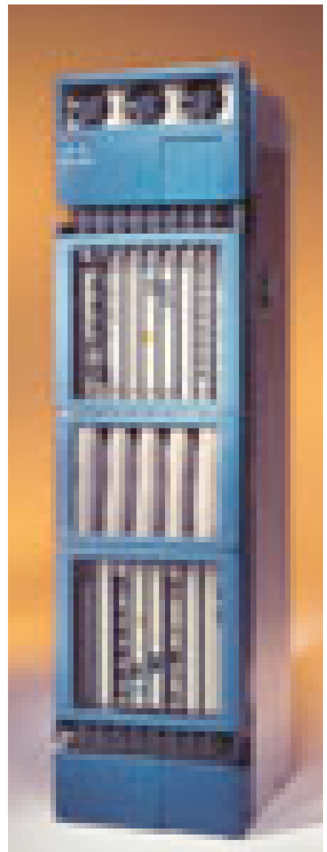
- Where IP routers sit in the network
- What IP routers look like
- What do IP routers do?
- Some details:
  - The internals of a "best-effort" router
    - Lookup, buffering and switching
  - The internals of a "QoS" router
- Can optics help?

The Internet is a mesh of routers (in theory)



What do they look like?

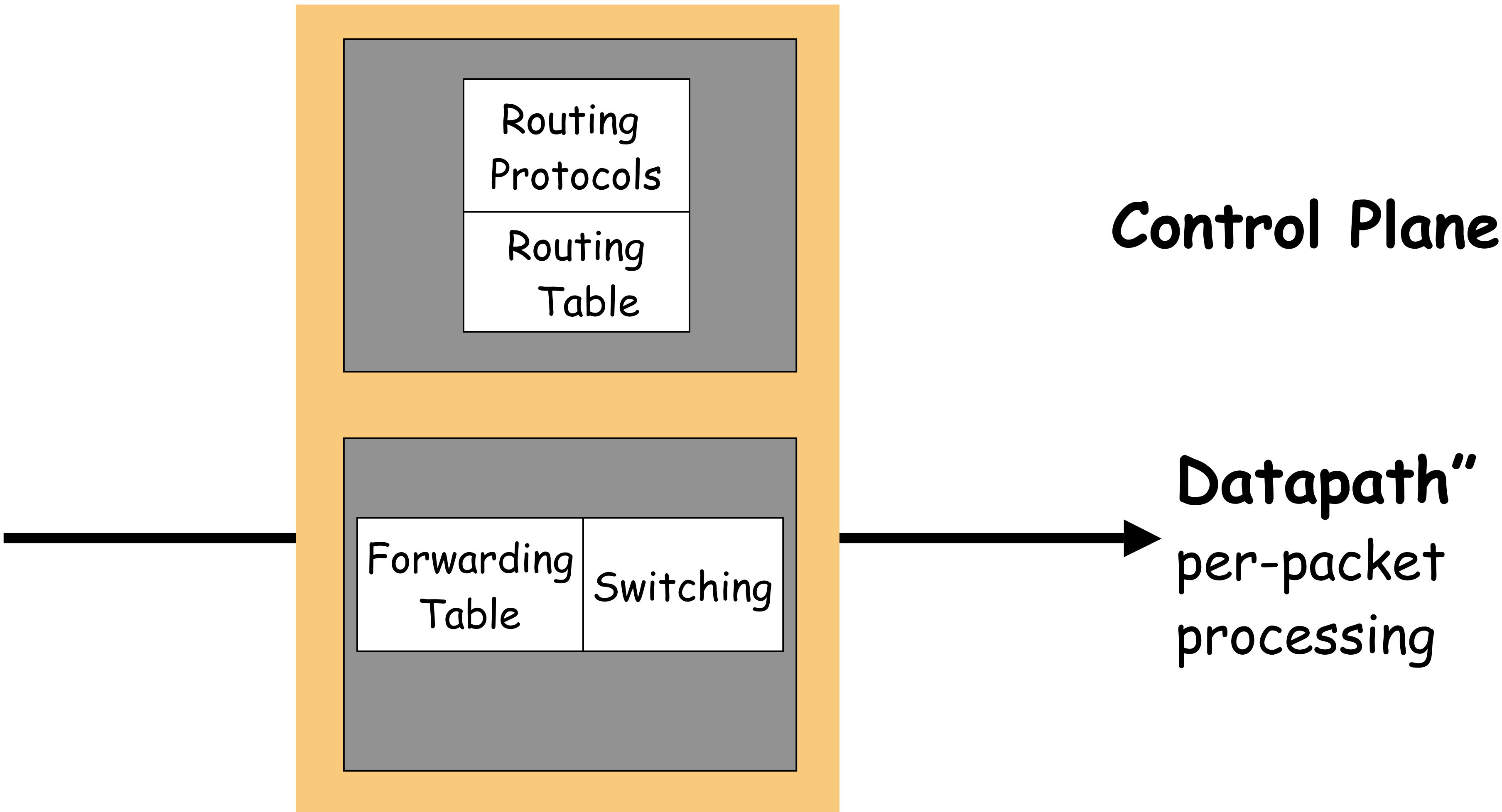
Access routers
e.g. ISDN, ADSL



Core router
e.g. OC48c POS

Core ATM switch

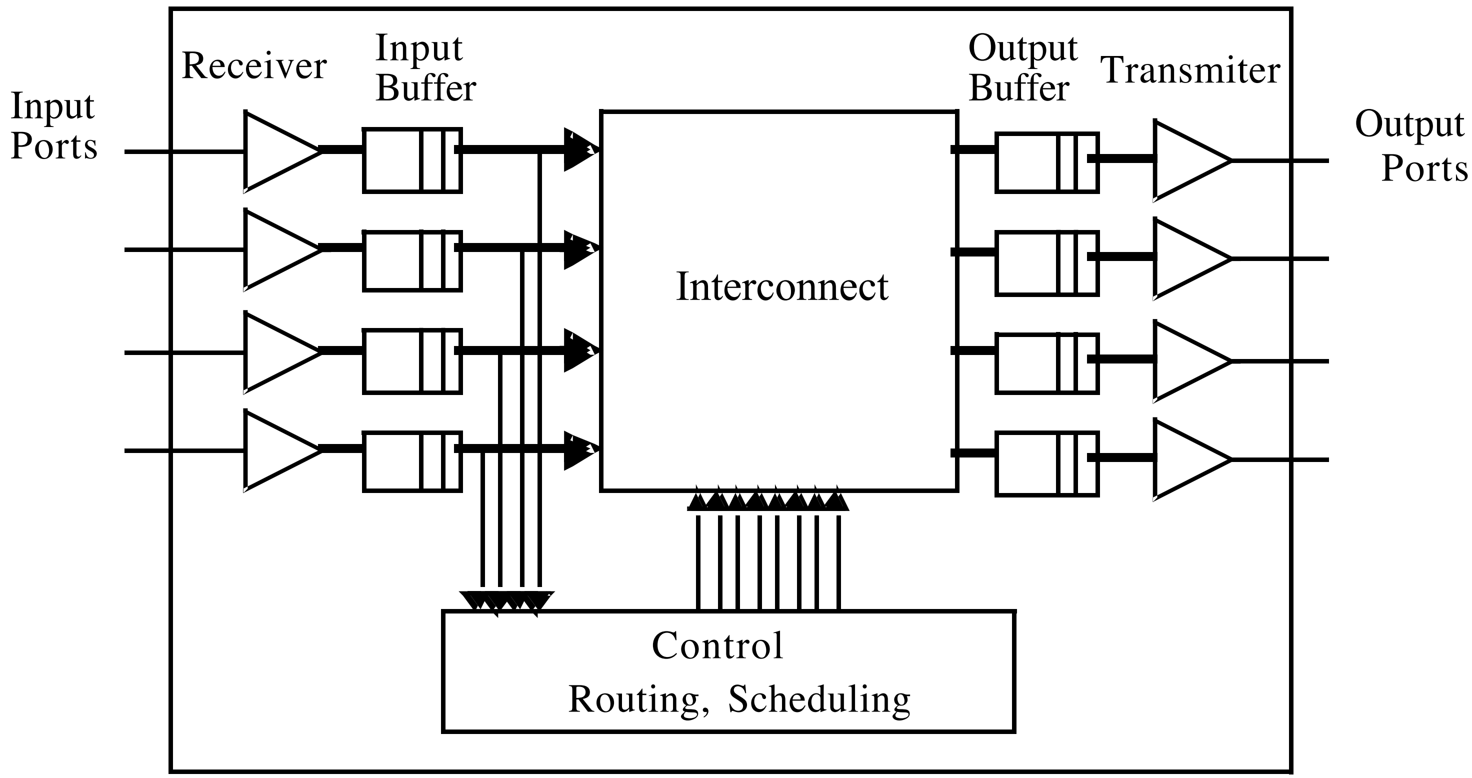
Basic Architectural Components of an IP Router



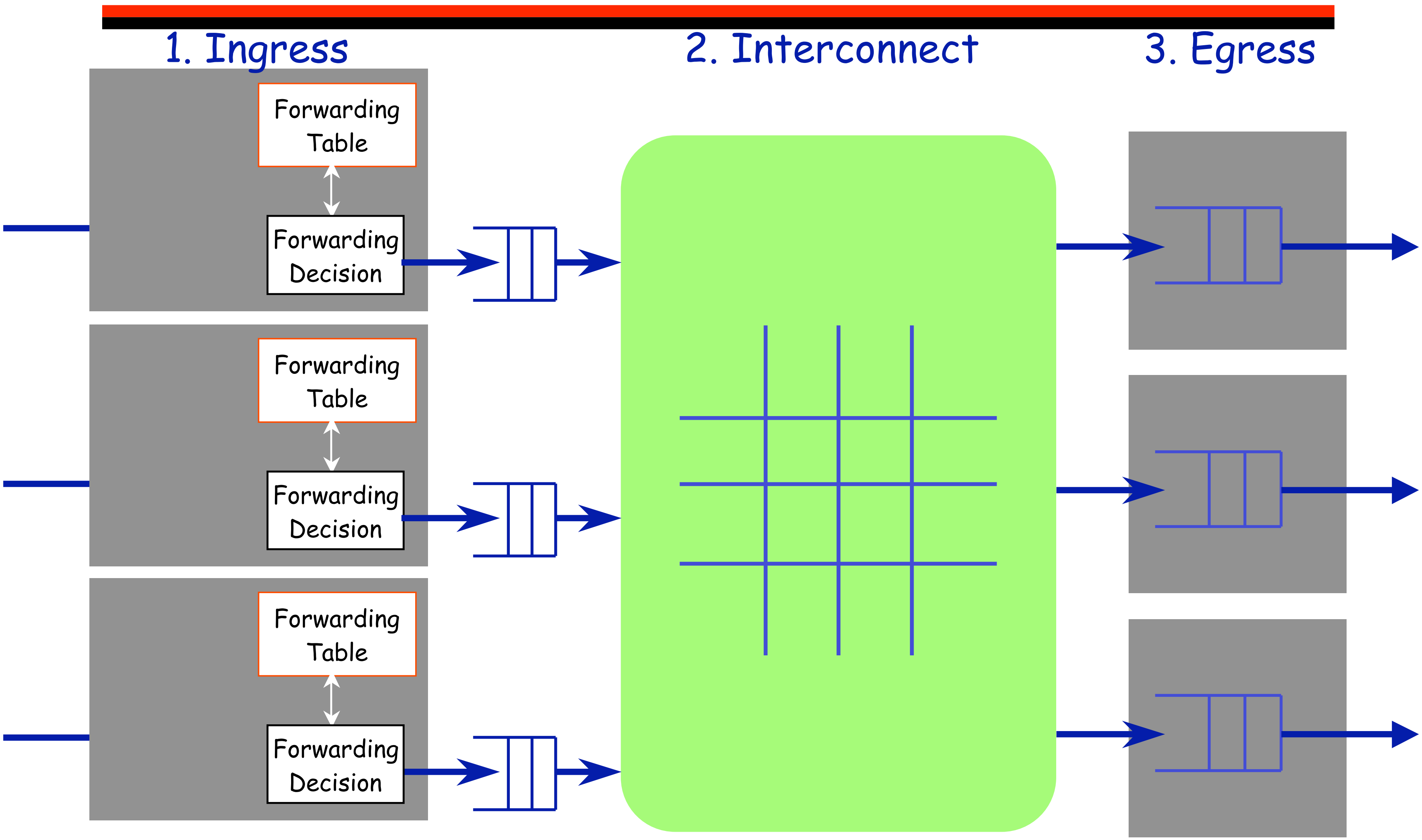
Per-packet processing in an IP Router

1. Accept packet arriving on an incoming link.
2. Lookup packet destination address in the forwarding table, to identify outgoing port(s).
3. Manipulate packet header: e.g., decrement TTL, update header checksum.
4. Send packet to the outgoing port(s).
5. Buffer packet in the queue.
6. Transmit packet onto outgoing link.

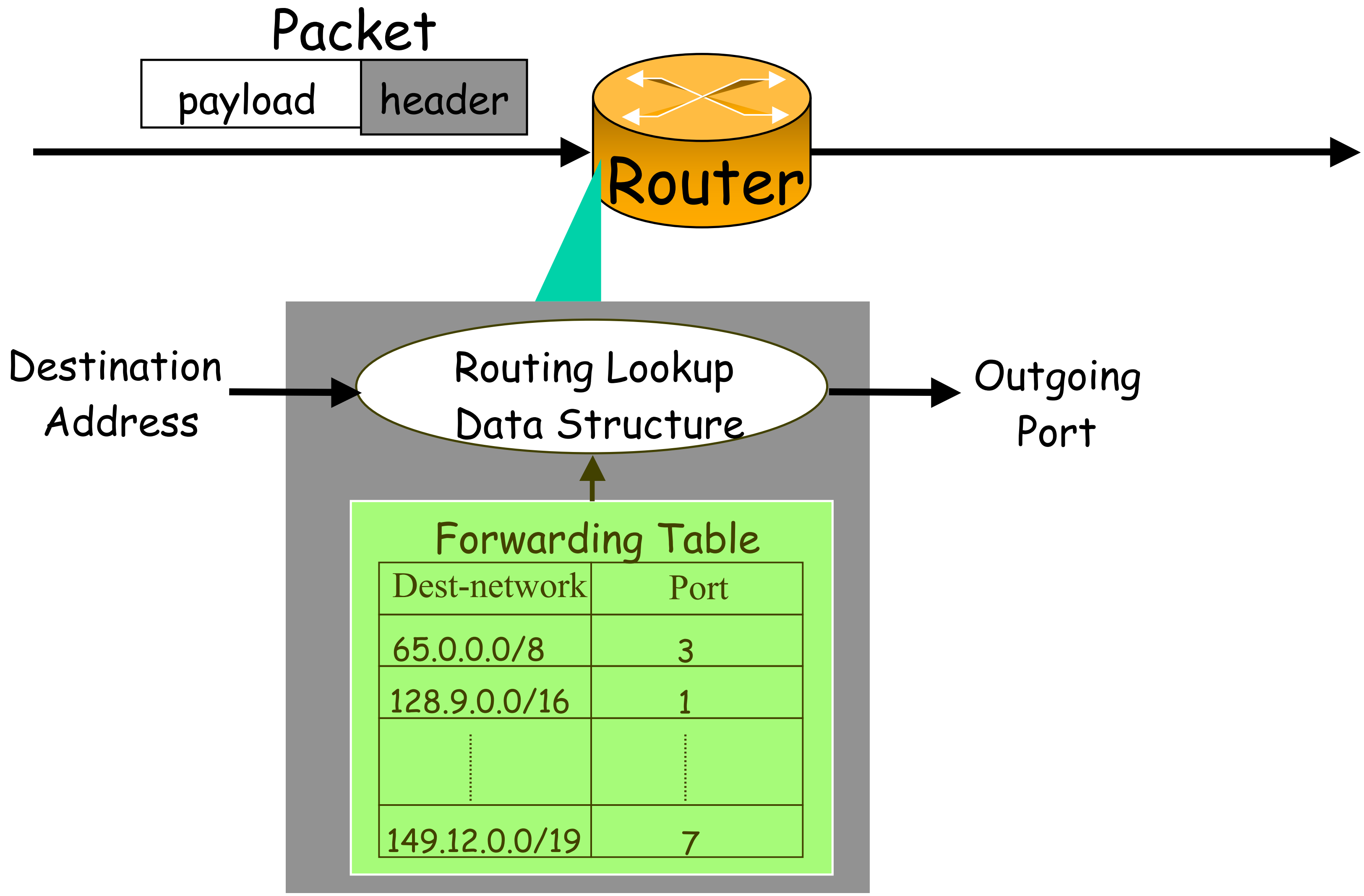
General Switch Model



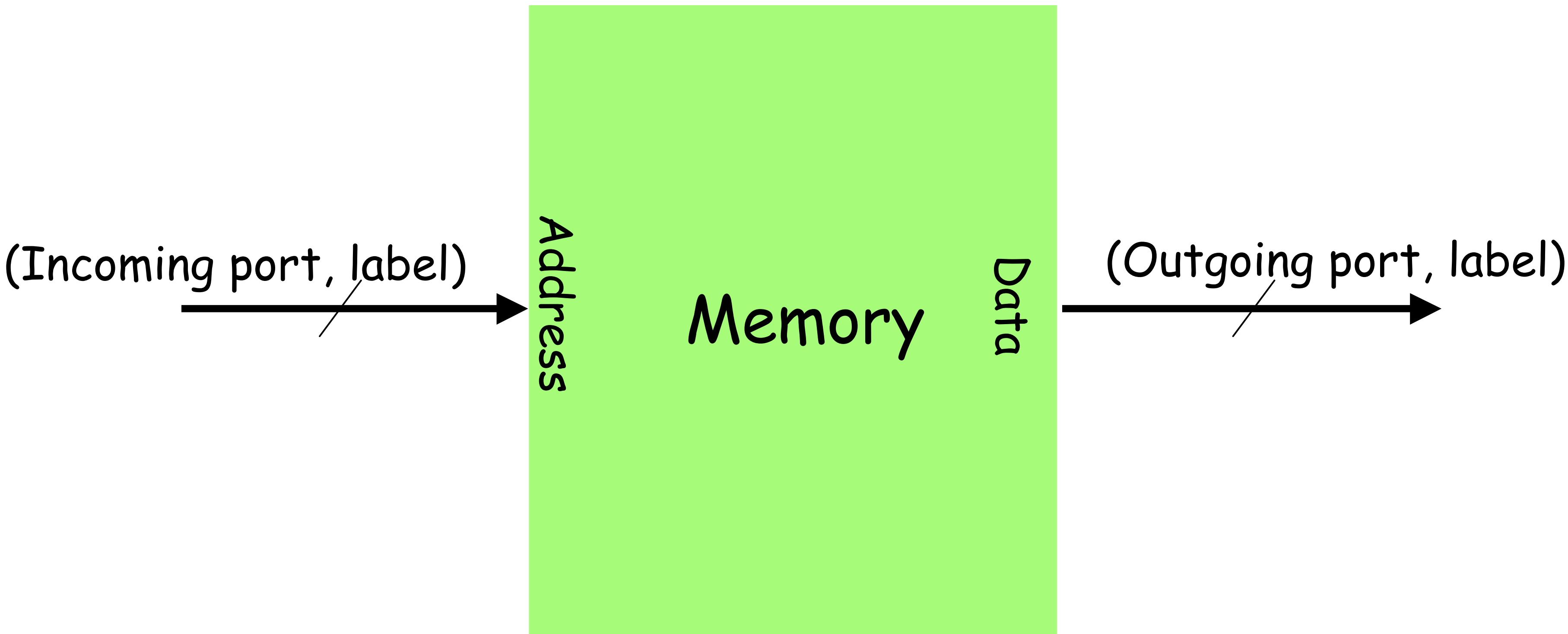
IP Switch Model



Forwarding Engine



The Search Operation is *not* a Direct Lookup



IP addresses: 32 bits long 4G entries

The Search Operation is also not an Exact Match Search

Exact match search: search for a key in a collection of keys of the same length.

Relatively well studied data structures:

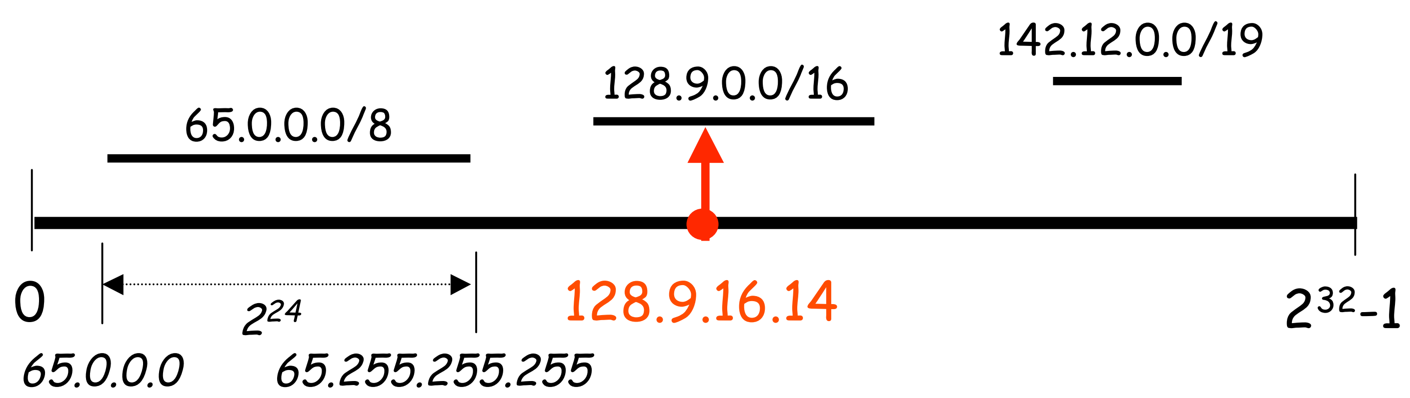
- Hashing
- Balanced binary search trees

Example Forwarding Table

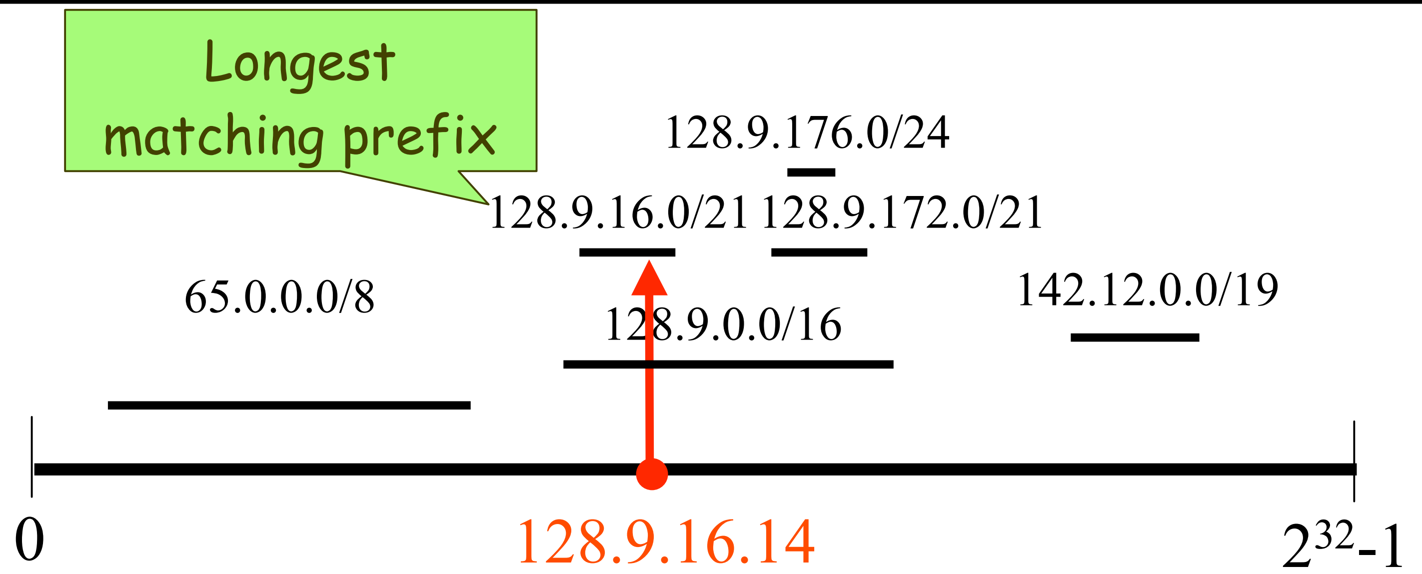
Destination IP Prefix	Outgoing Port
65.0.0.0/8	3
128.9.0.0/16	1
142.12.0.0/19	7

Prefix length

IP prefix: 0-32 bits

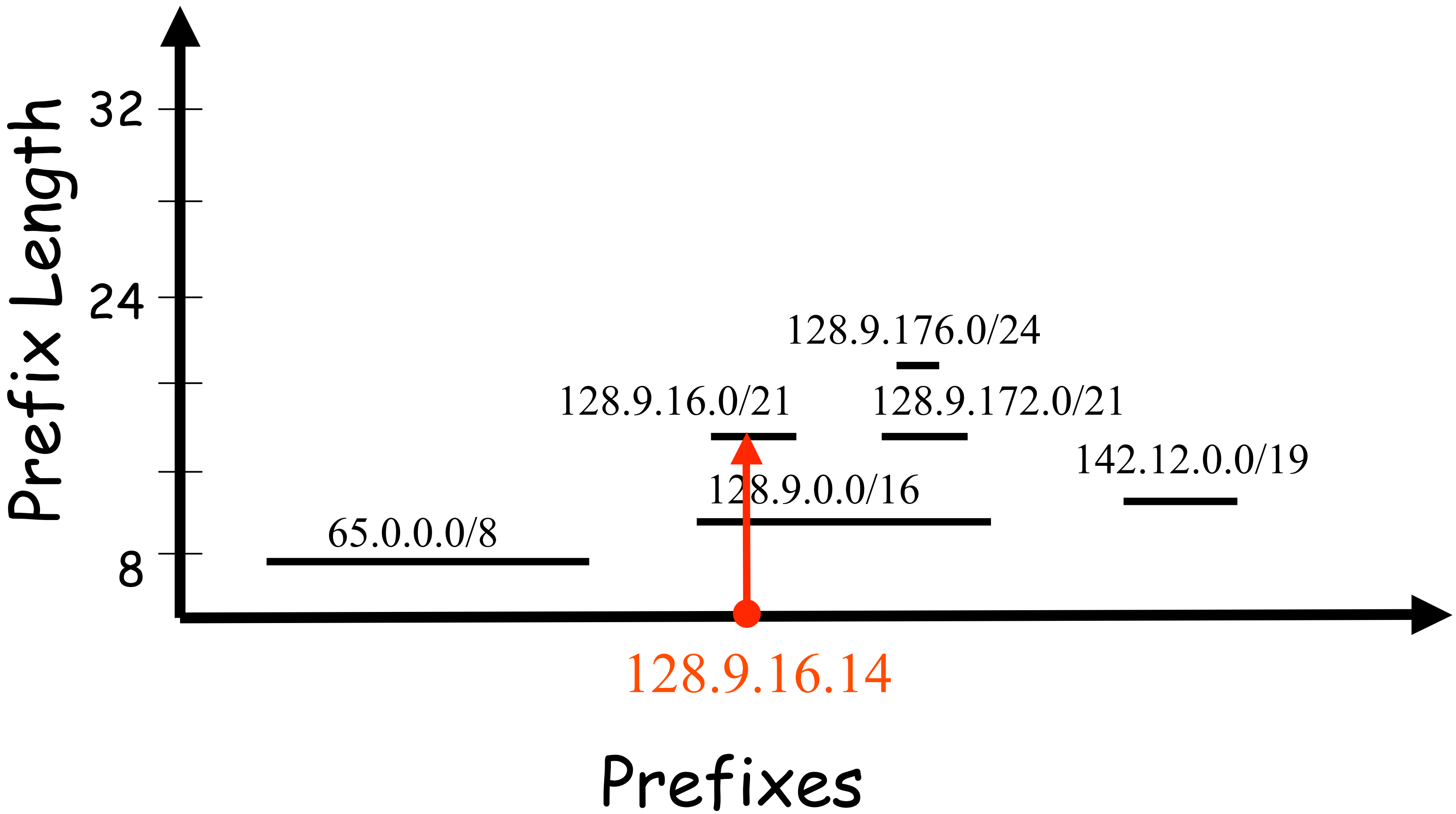


Prefixes can Overlap



Routing lookup: Find the **longest** matching prefix (the most specific route) among all prefixes that match the destination address.

Difficulty of Longest Prefix Match



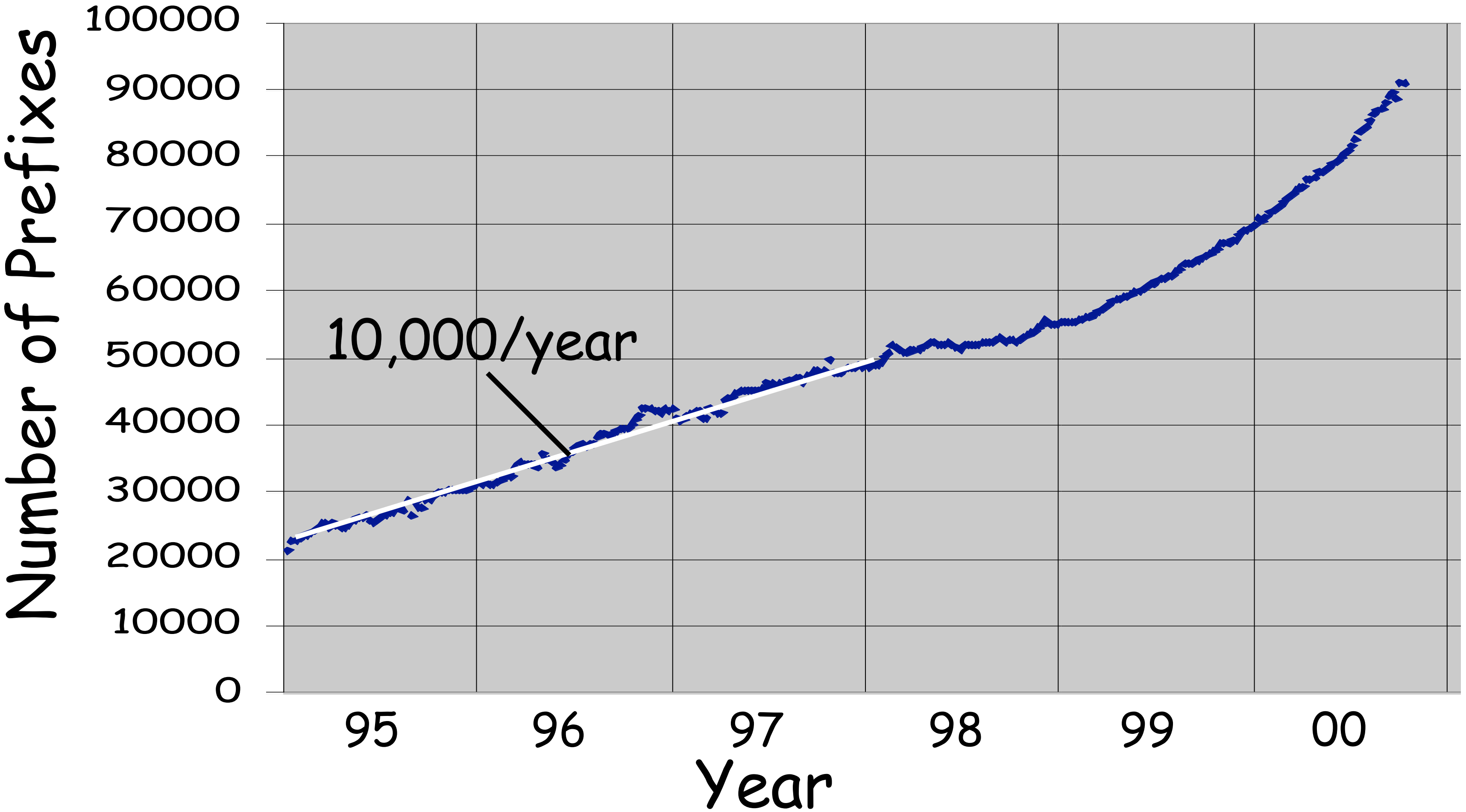
Lookup Rate Required

Year	Line	Line-rate (Gbps)	40B packets (Mpps)
1998-99	OC12c	0.622	1.94
1999-00	OC48c	2.5	7.81
2000-01	OC192c	10.0	31.25
2002-03	OC768c	40.0	125

31.25 Mpps → 33 ns

DRAM: 50-80 ns, SRAM: 5-10 ns

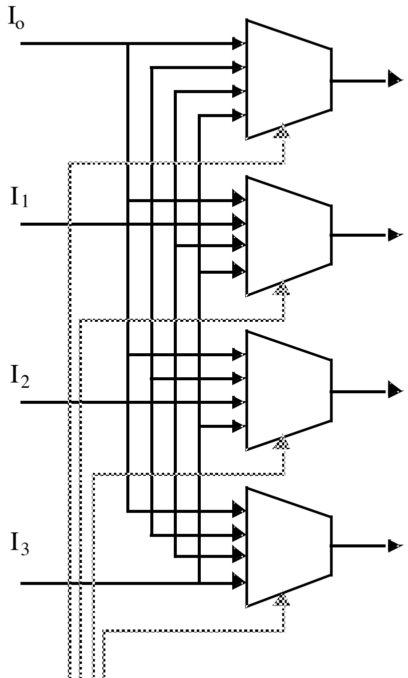
Size of the Forwarding Table



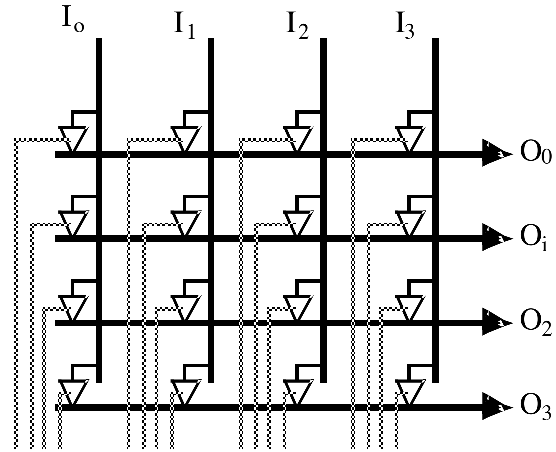
Source: http://www.telstra.net/ops/bgptable.html

Internal Interconnects

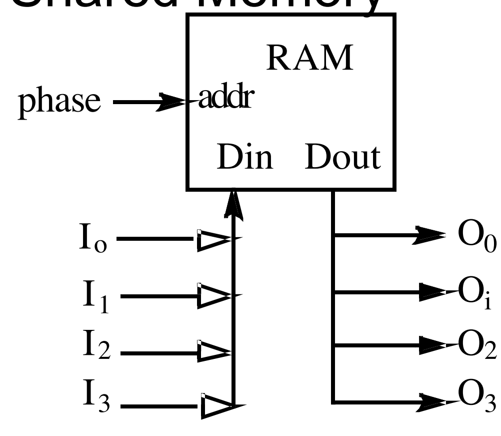
1. Multiplexers



2. Tri-State Devices



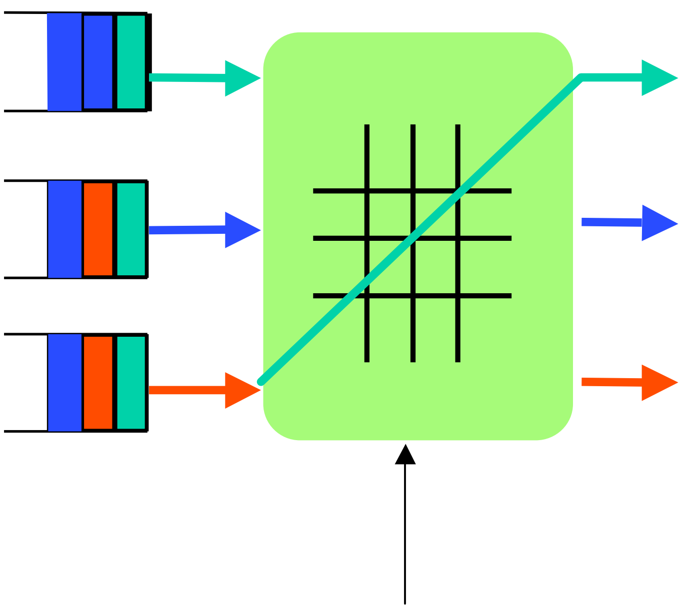
3. Shared Memory



Interconnects

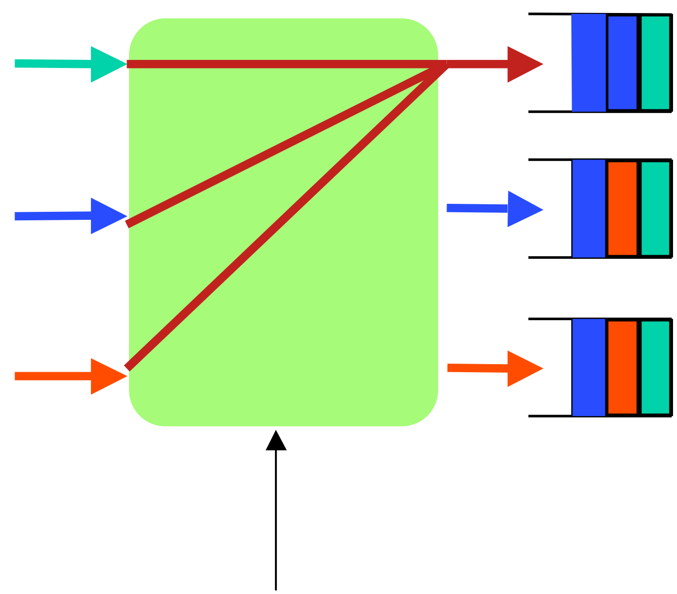
Two basic techniques

Input Queueing



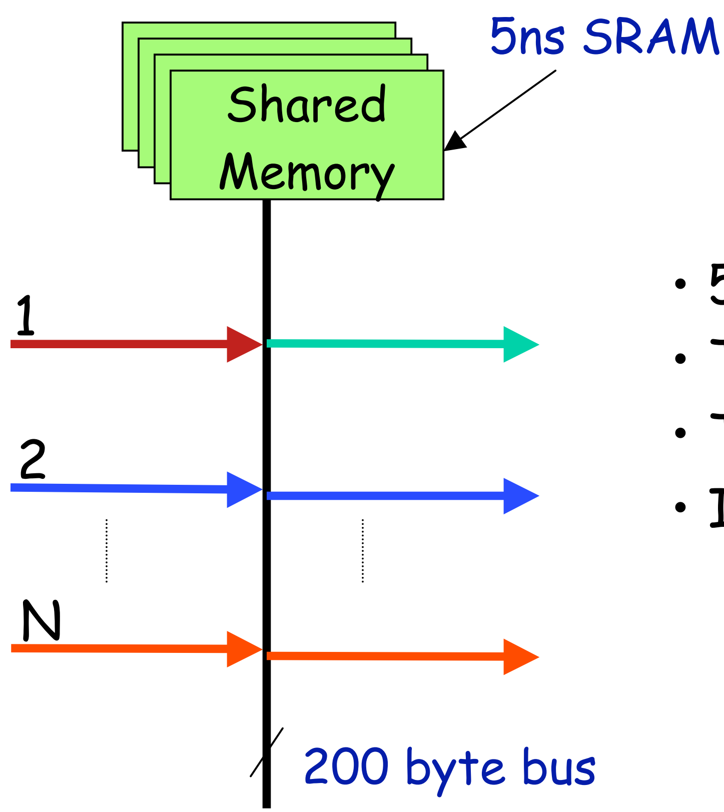
Usually a non-blocking switch fabric (e.g. crossbar)

Output Queueing



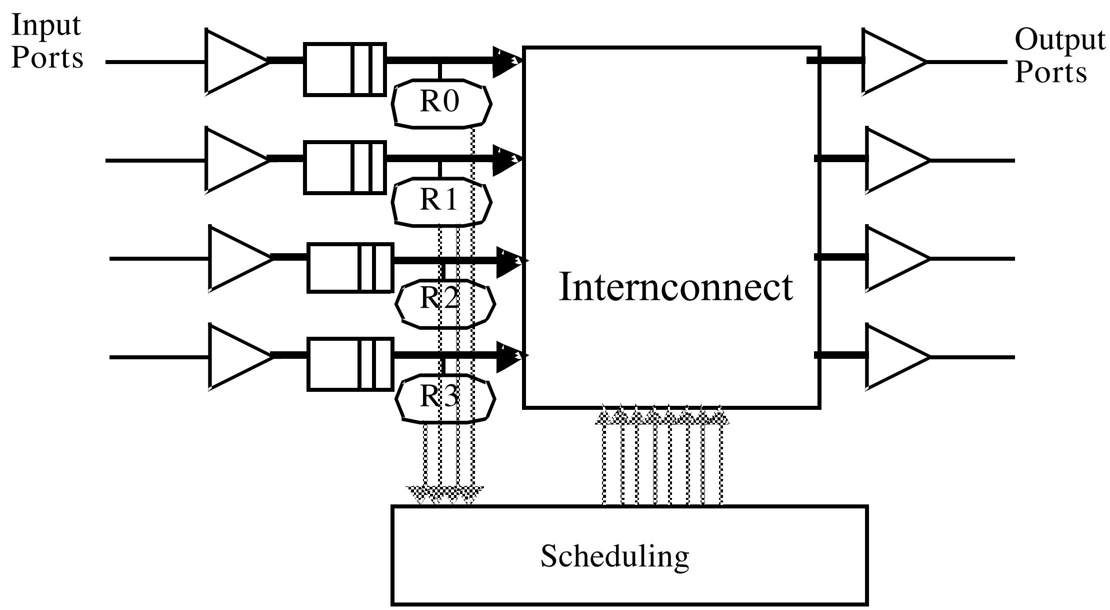
Usually a fast bus

Shared Memory Bandwidth



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

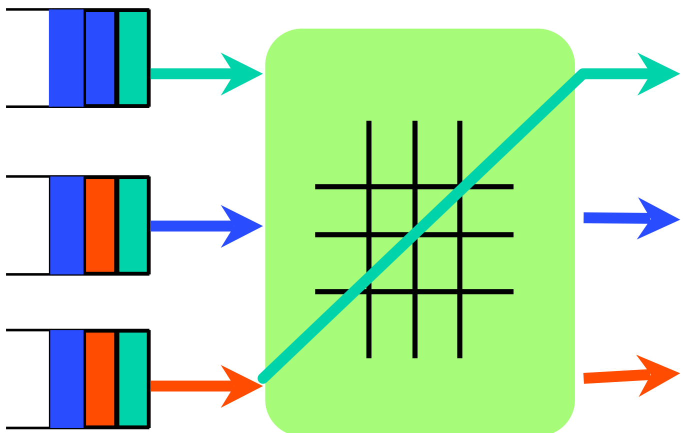
Input buffered swtich

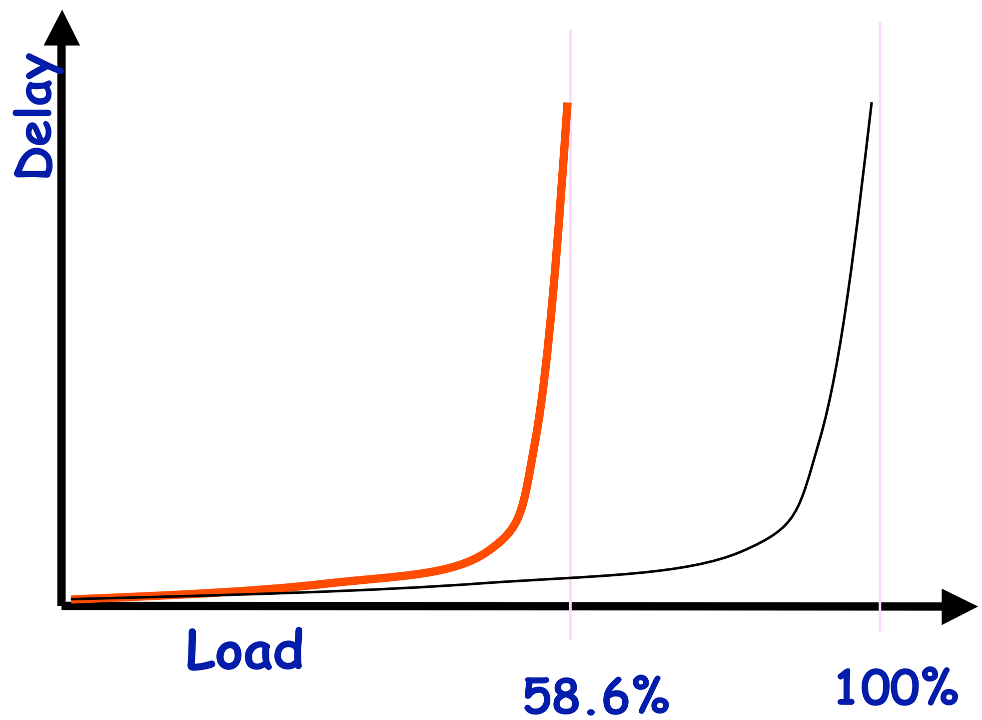


- Independent routing logic per input
  - FSM
- Scheduler logic arbitrates each output
  - priority, FIFO, random
- Head-of-line blocking problem

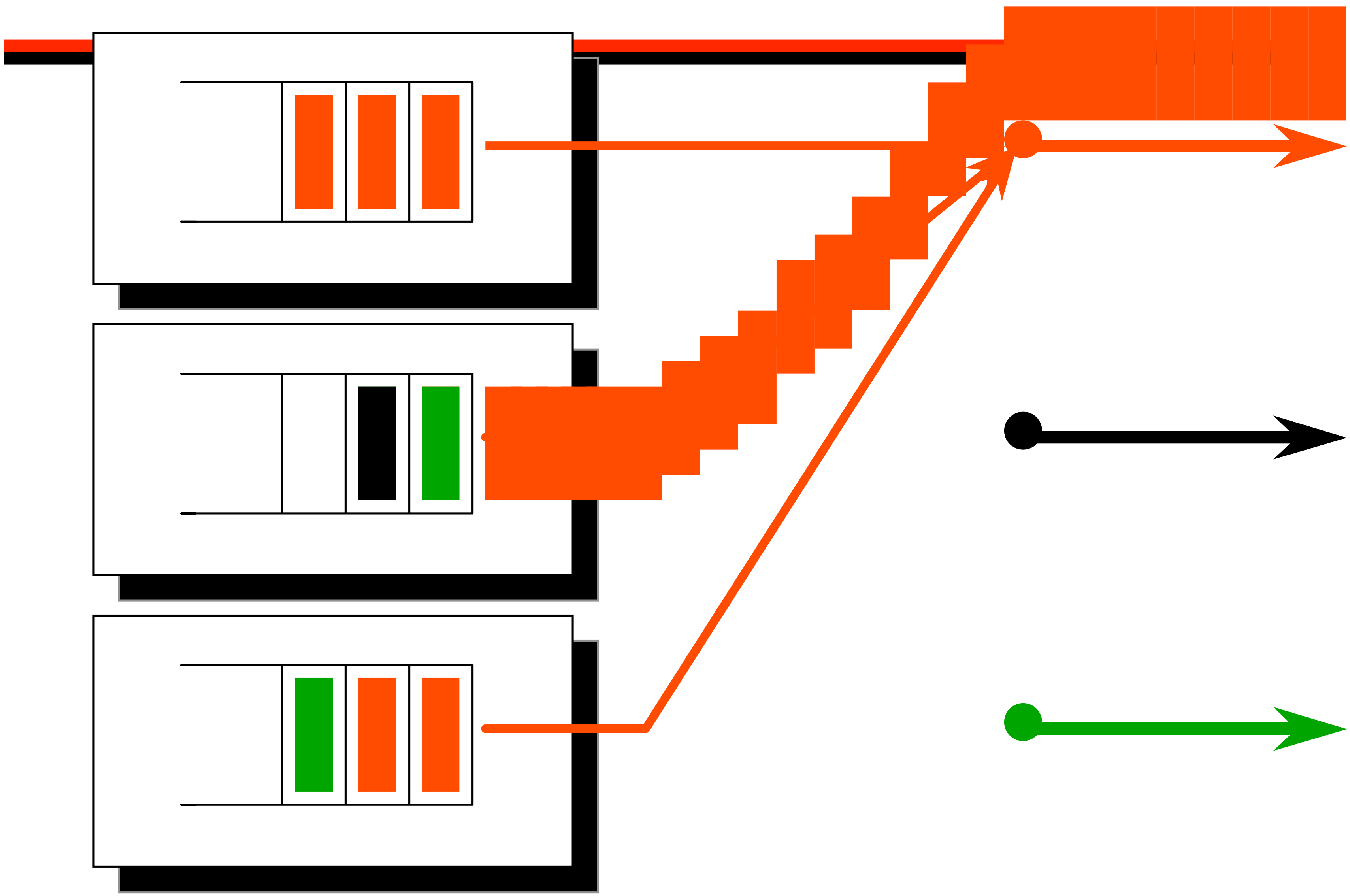
Input Queueing

Head of Line Blocking



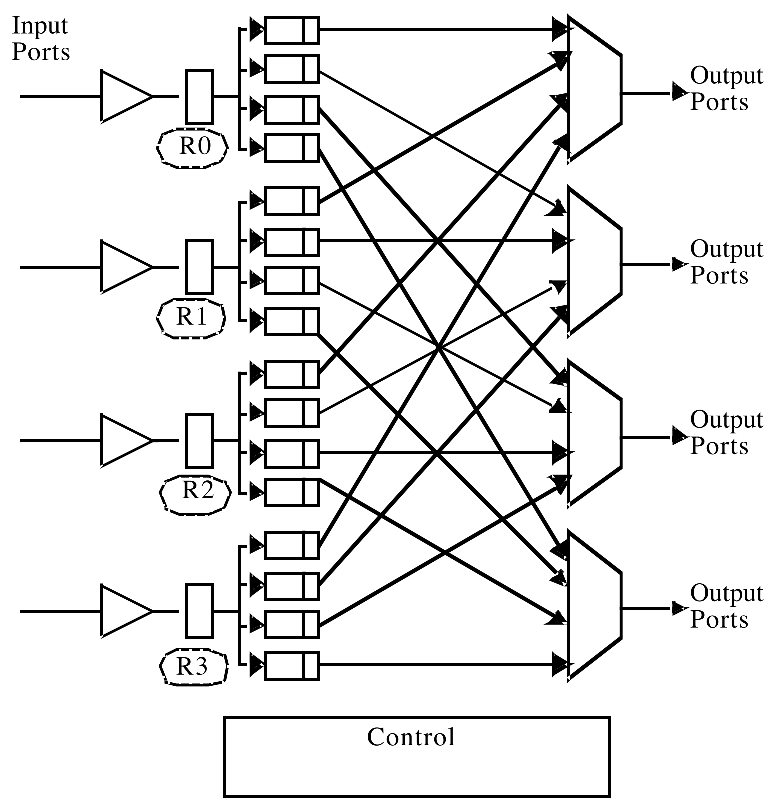


Head of Line Blocking



(Virtual) Output Buffered Switch

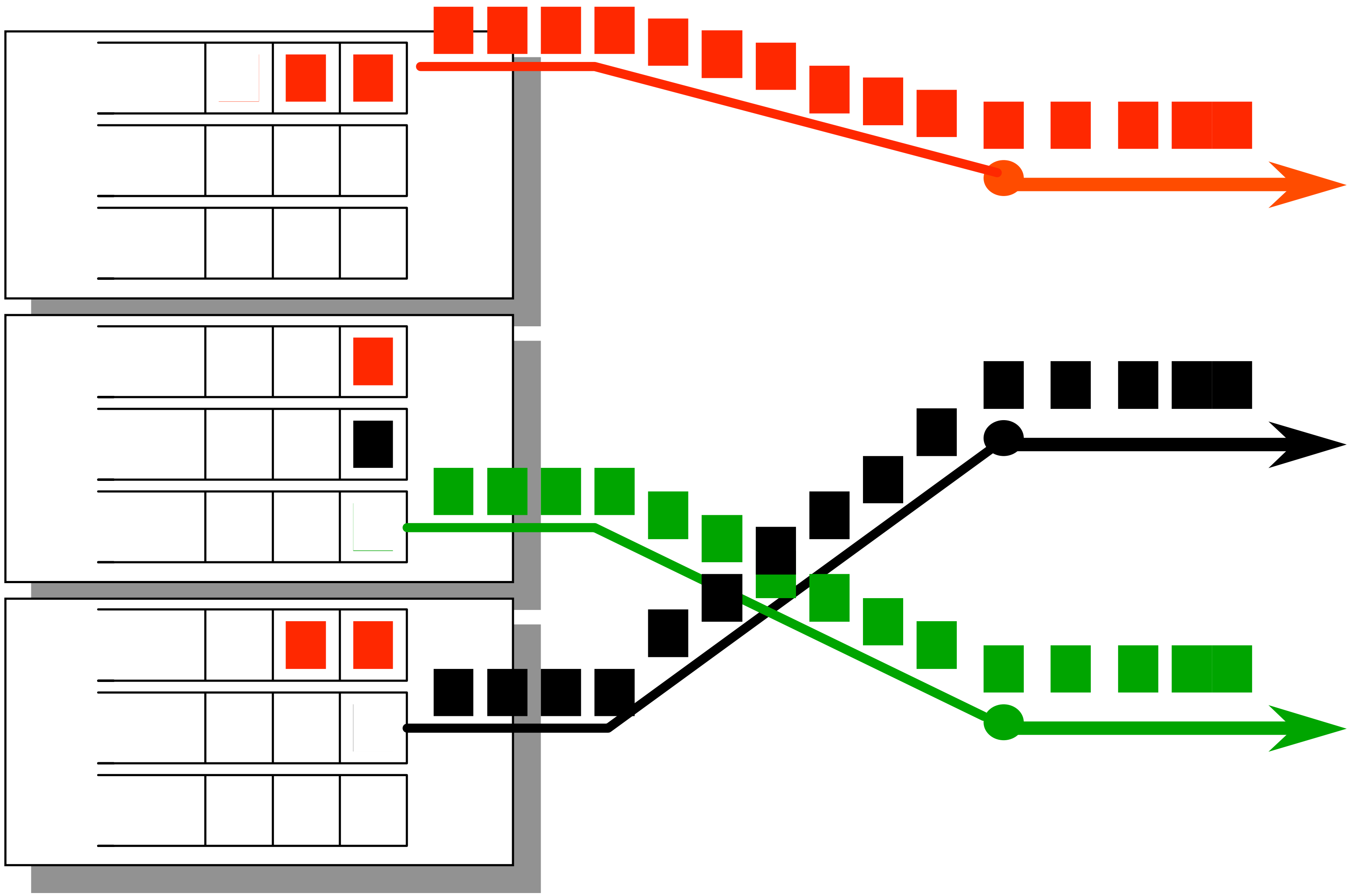
N buffers per input



- How would you build a shared pool?

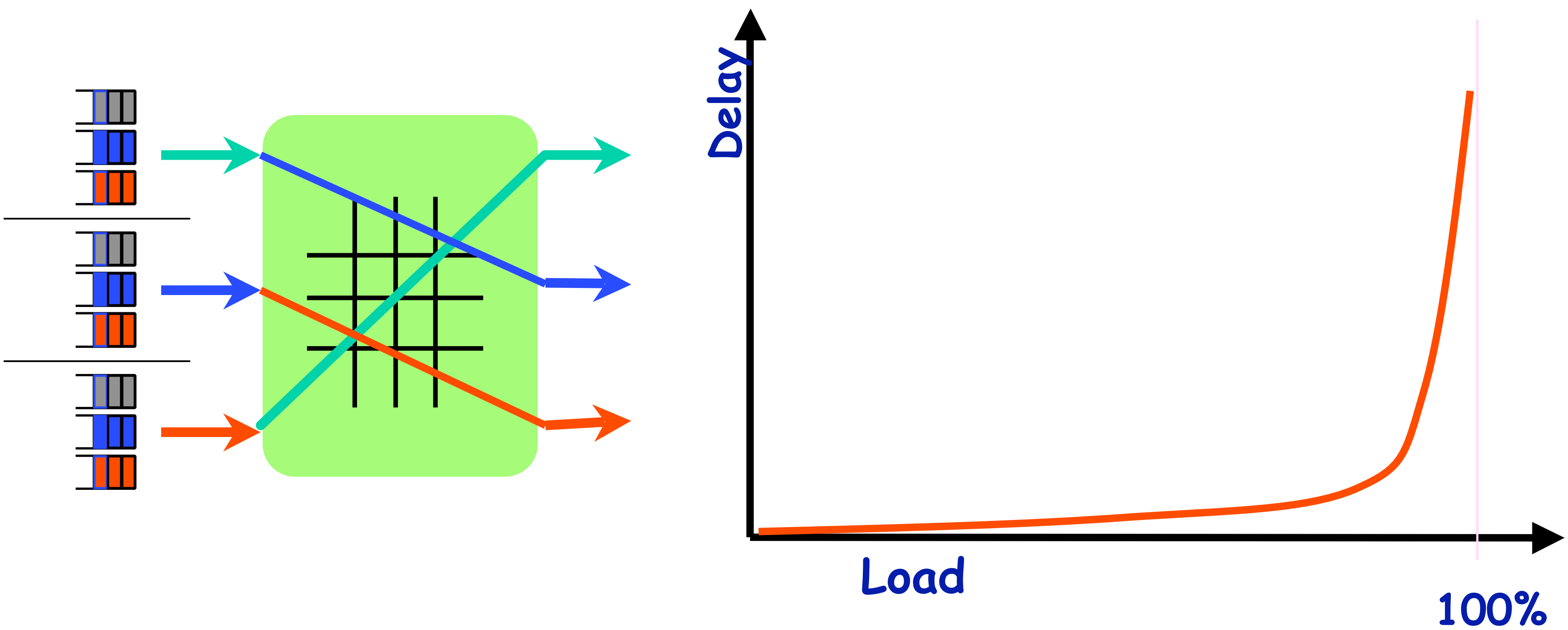
Solving HOL with Input Queueing

Virtual output queues

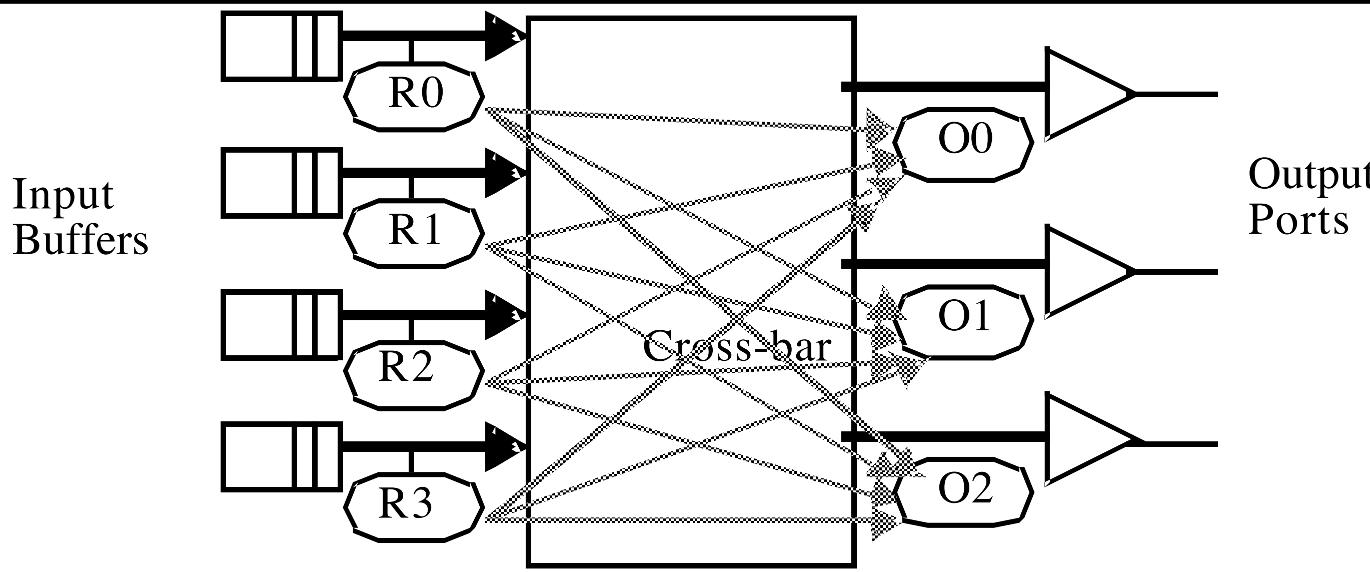


Input Queueing

Virtual Output Queues



Output scheduling

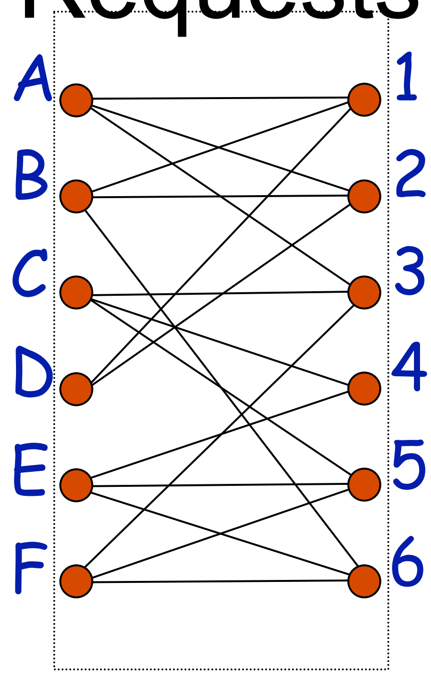


- n independent arbitration problems?
  - static priority, random, round-robin
- simplifications due to routing algorithm?
- general case is max bipartite matching

Finding a maximum size match

Inputs

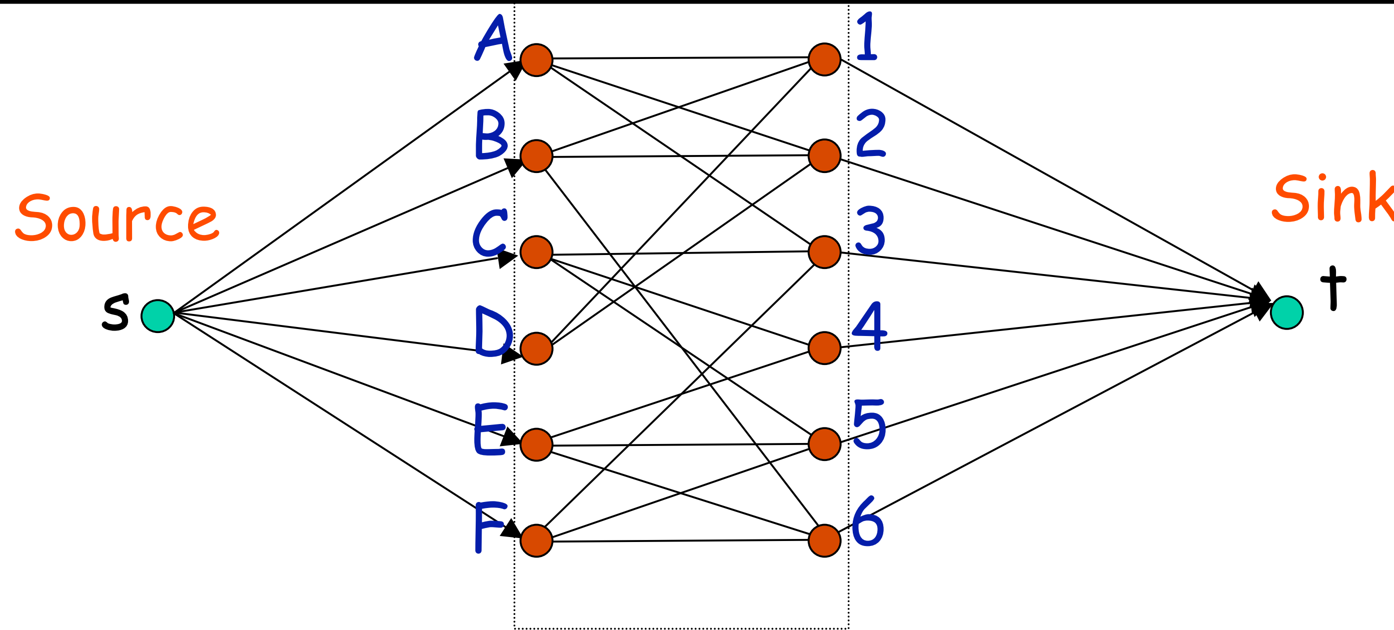
Requests



Outputs

- How do we find the maximum size (weight) match?

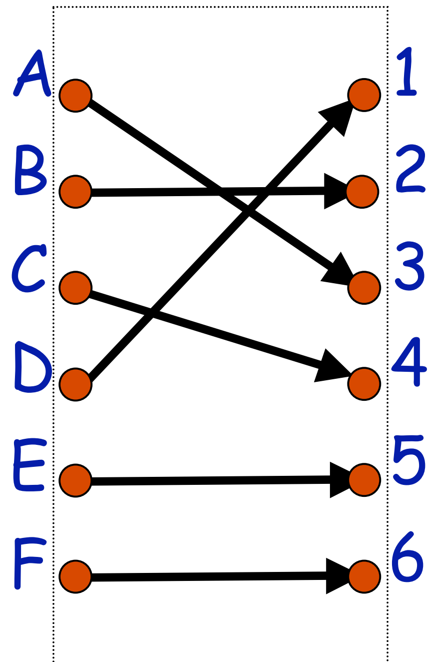
Network flows and bipartite matching



Finding a maximum size bipartite matching is equivalent to solving a network flow problem with capacities and flows of size 1.

Network flows and bipartite matching

Maximum Size Matching:



Complexity of Maximum Matchings

- Maximum Size Matchings:
  - Algorithm by Dinic $O(N^{5/2})$
- Maximum Weight Matchings
  - Algorithm by Kuhn $O(N^3)$

- In general:
  - Hard to implement in hardware
  - Too Slow in Practice
    - But gives nice theory and upper bound

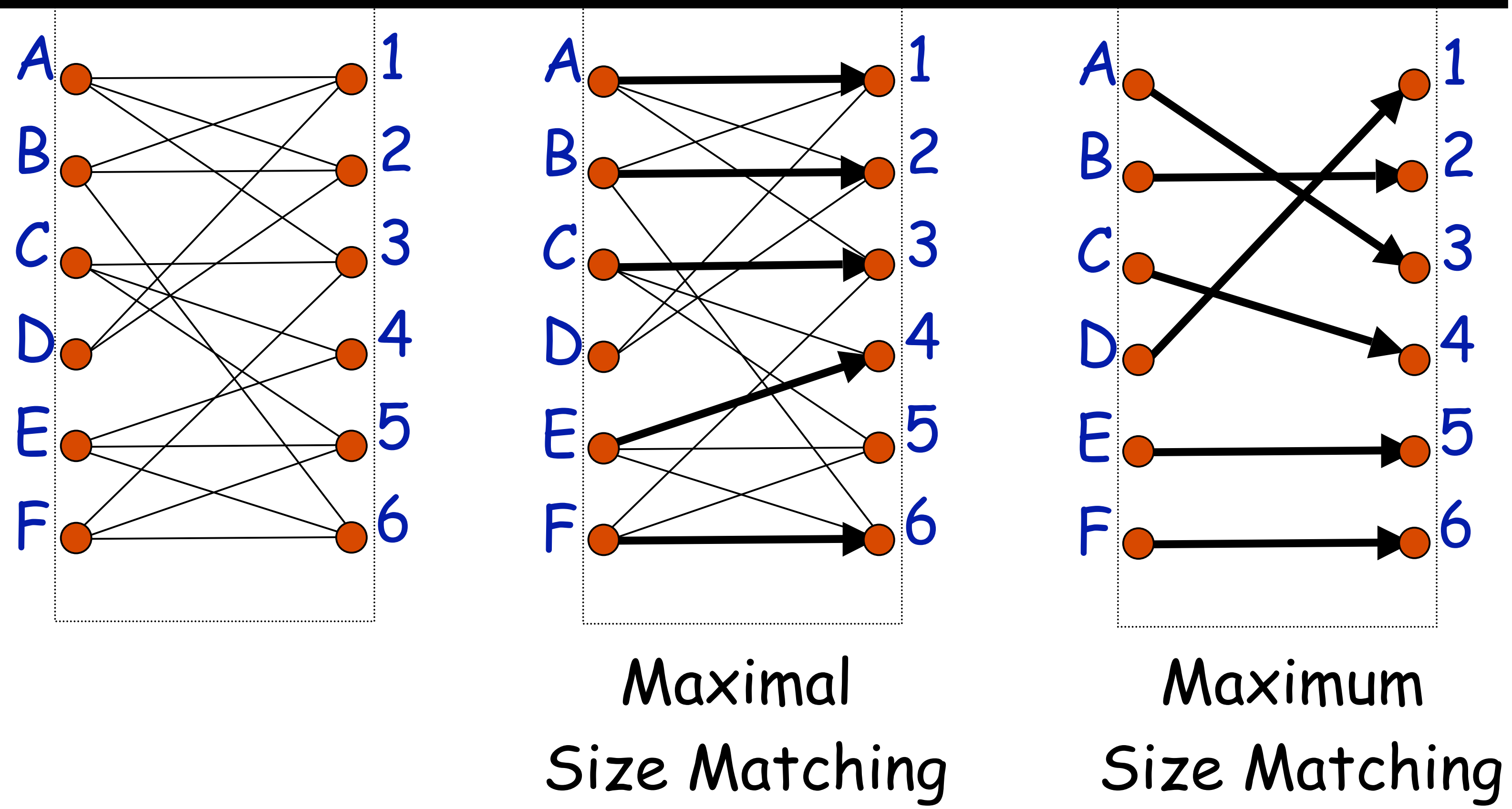
Aribitration

- Maximal Matches
- Wavefront Arbiter (WFA)
- Parallel Iterative Matching (PIM)
- *i*SLIP

Maximal Matching

- A maximal matching is one in which each edge is added one at a time, and is not later removed from the matching.
- i.e. no augmenting paths allowed (they remove edges added earlier).
- No input and output are left unnecessarily idle.

Example of Maximal Size Matching

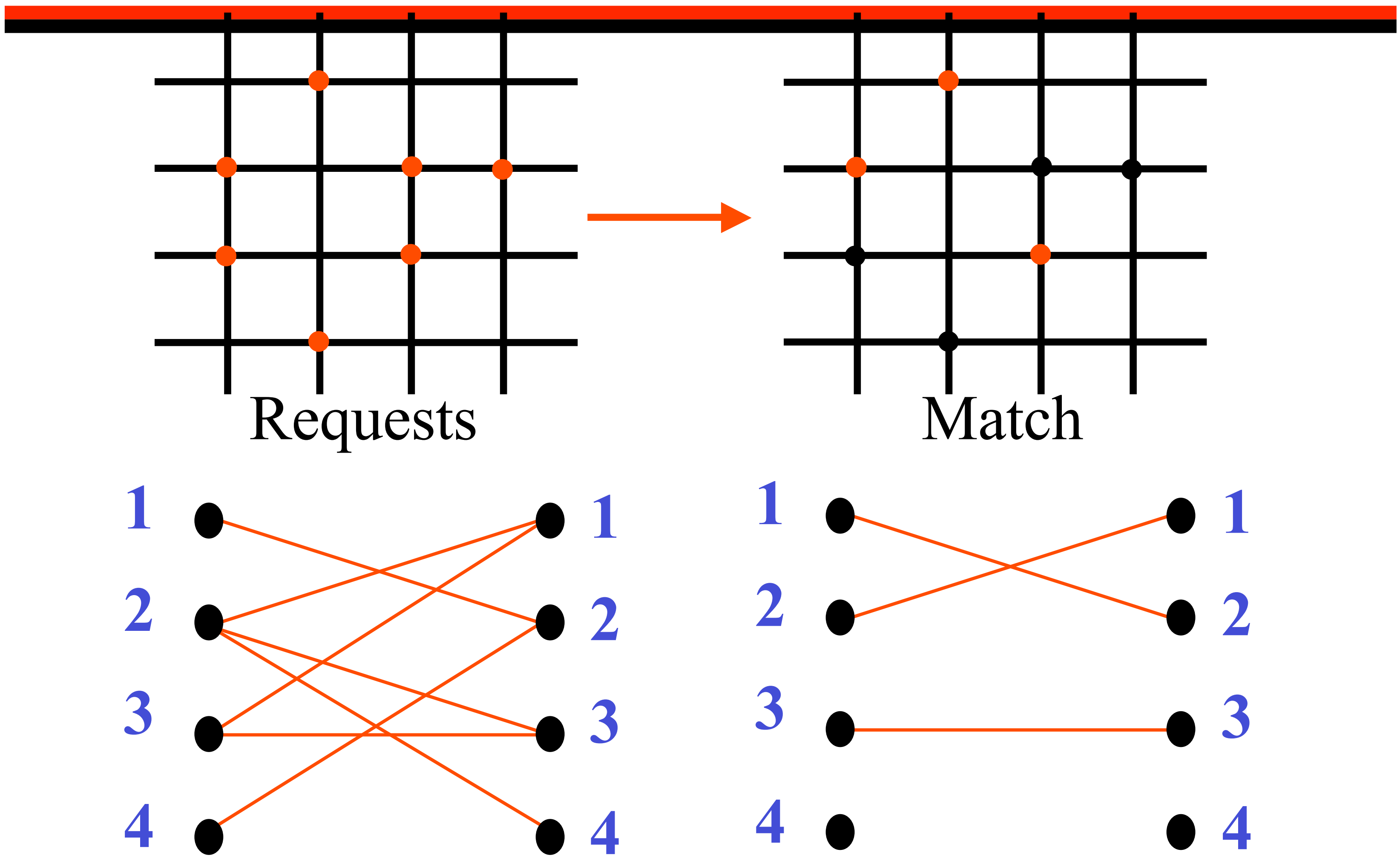


Maximal Matchings

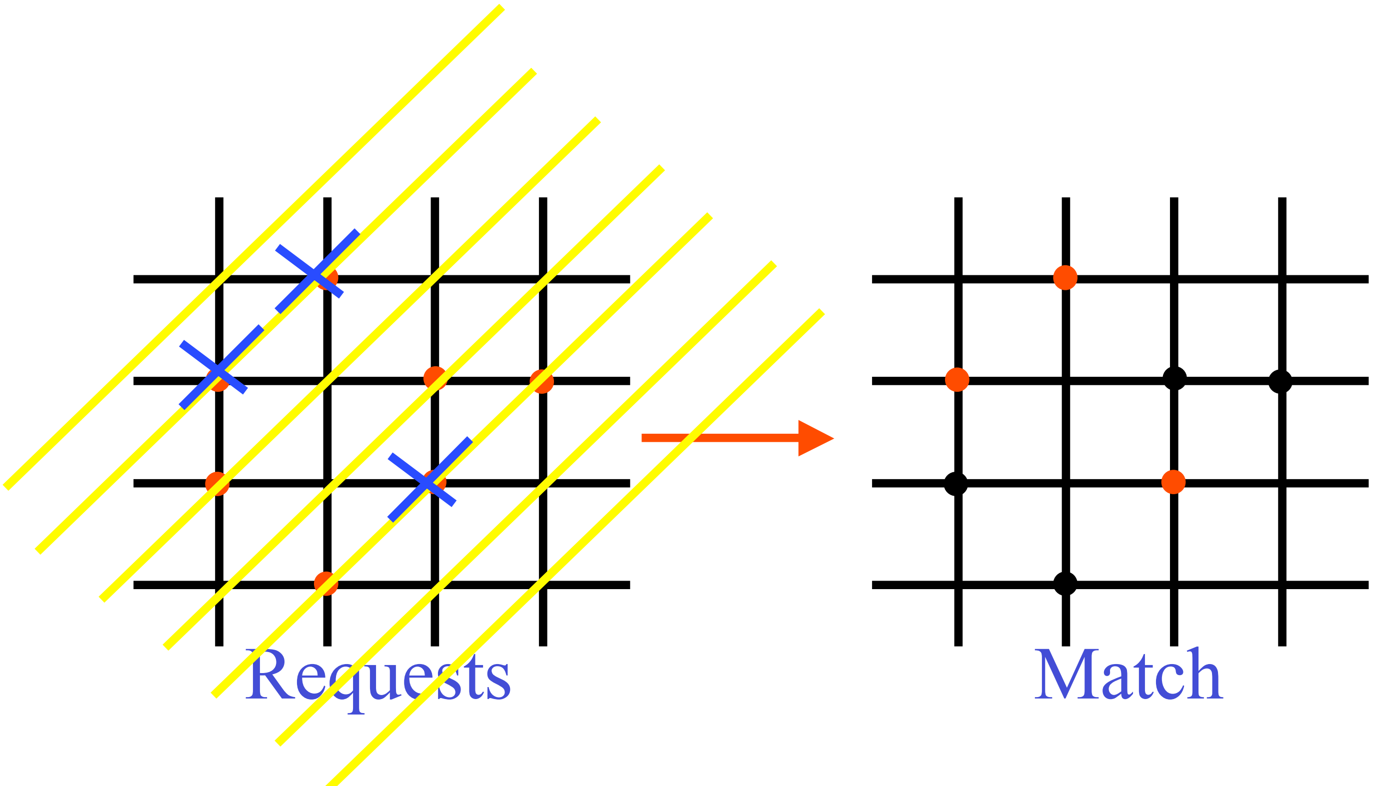
- In general, *maximal* matching is simpler to implement, and has a faster running time.
- A maximal size matching is at least half the size of a maximum size matching.
- A maximal weight matching is defined in the obvious way.
- A maximal weight matching is at least half the weight of a maximum weight matching.

Wave Front Arbiter

(Tamir)



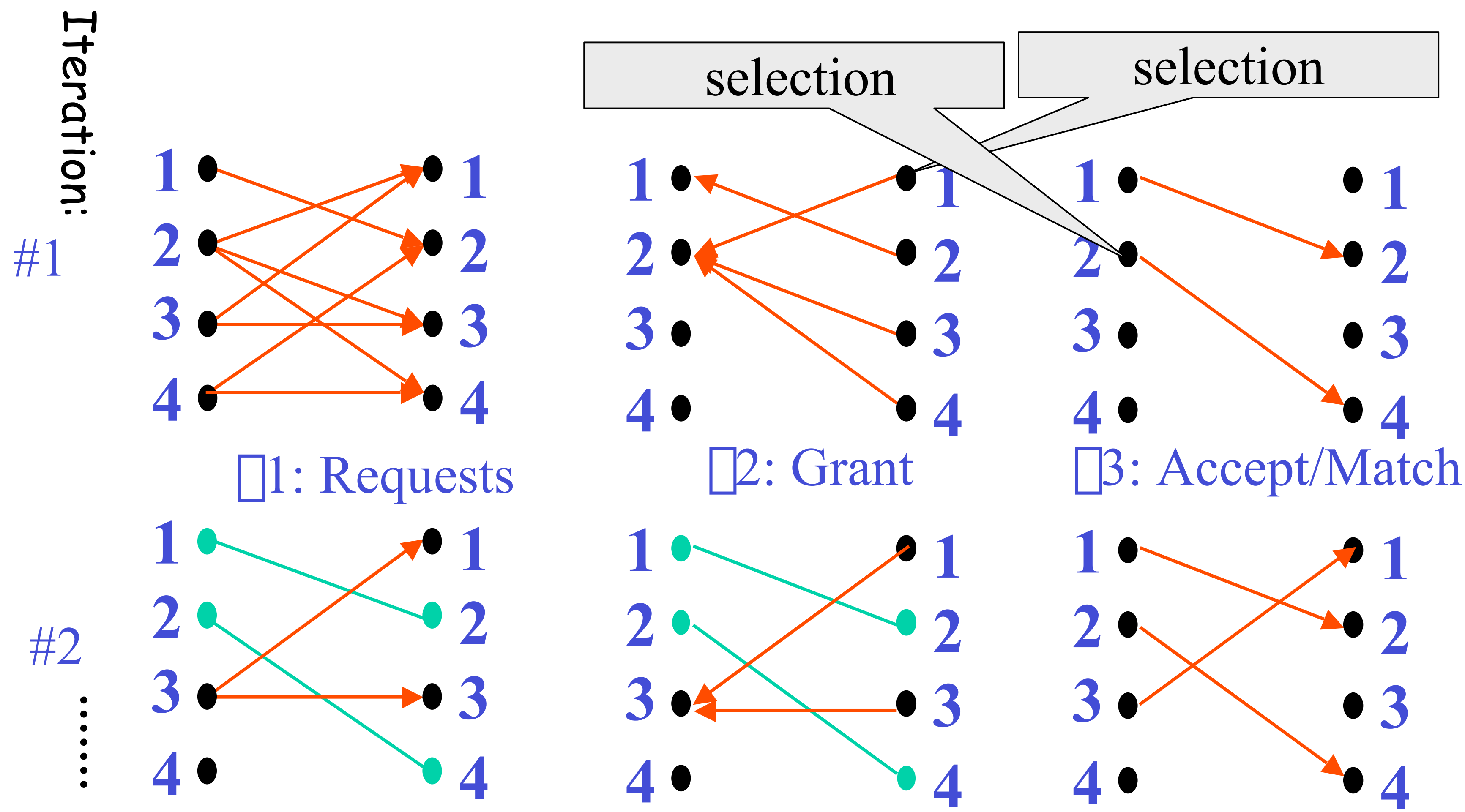
Wave Front Arbiter



Wavefront Arbiters
Properties

- Feed-forward (i.e. non-iterative) design lends itself to pipelining.
- Always finds maximal match.
- Usually requires mechanism to prevent inputs from getting preferential service.
  - What the 50Gbs router does:
    - Scramble (permute) inputs each cycle

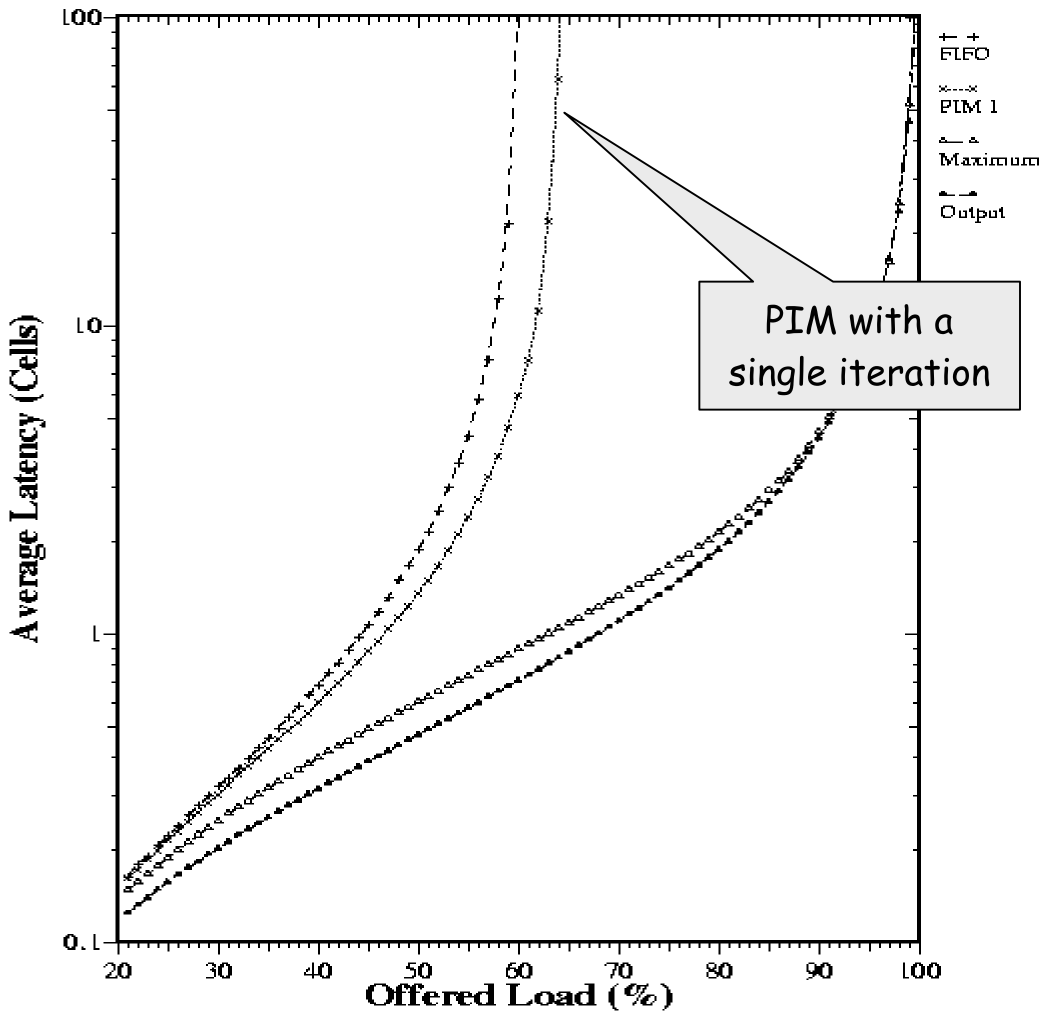
Parallel Iterative Matching



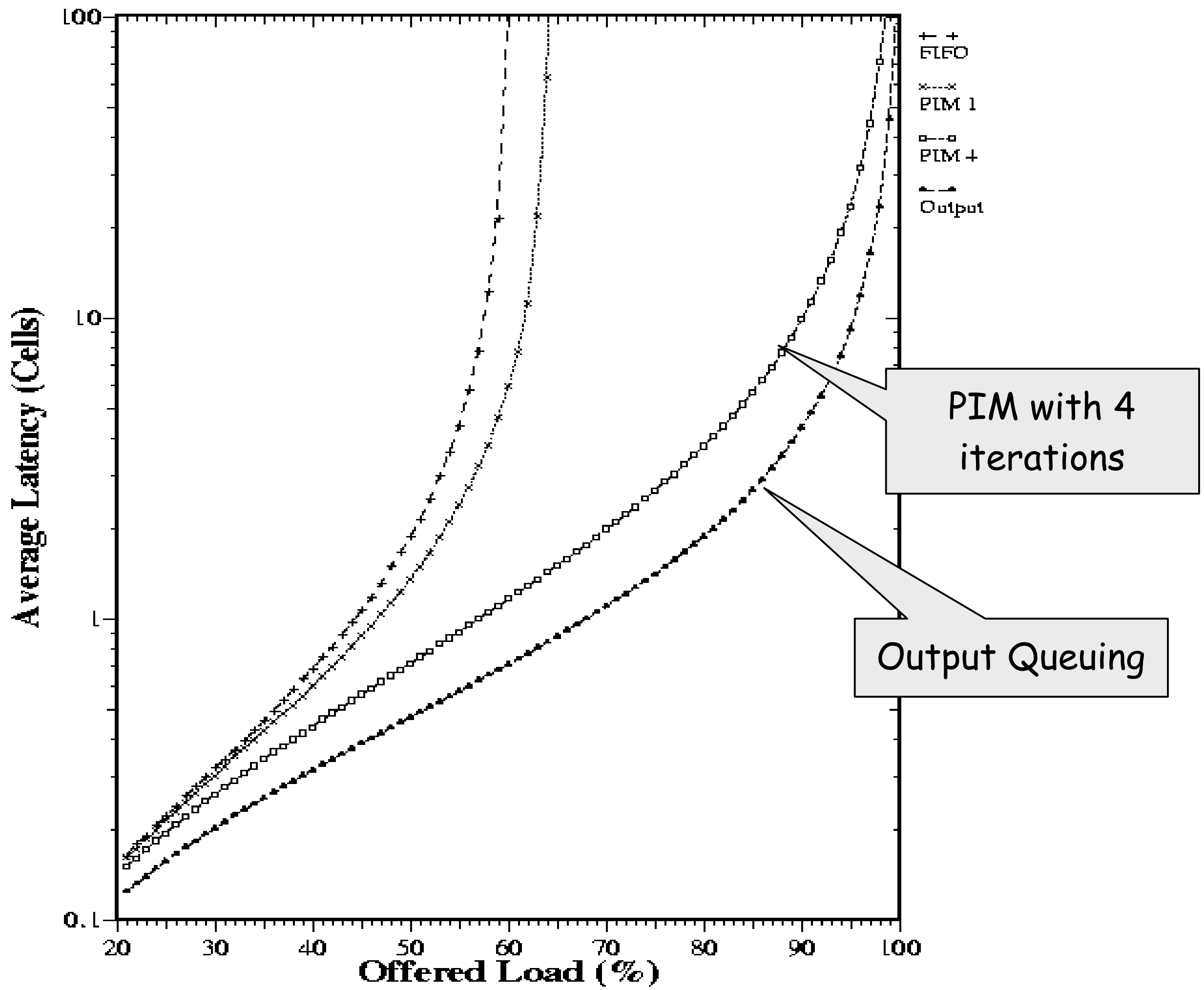
PIM Properties

- Guaranteed to find a maximal match in at most N iterations.
- In each phase, each input and output arbiter can make decisions independently.
- In general, will converge to a maximal match in $< N$ iterations.

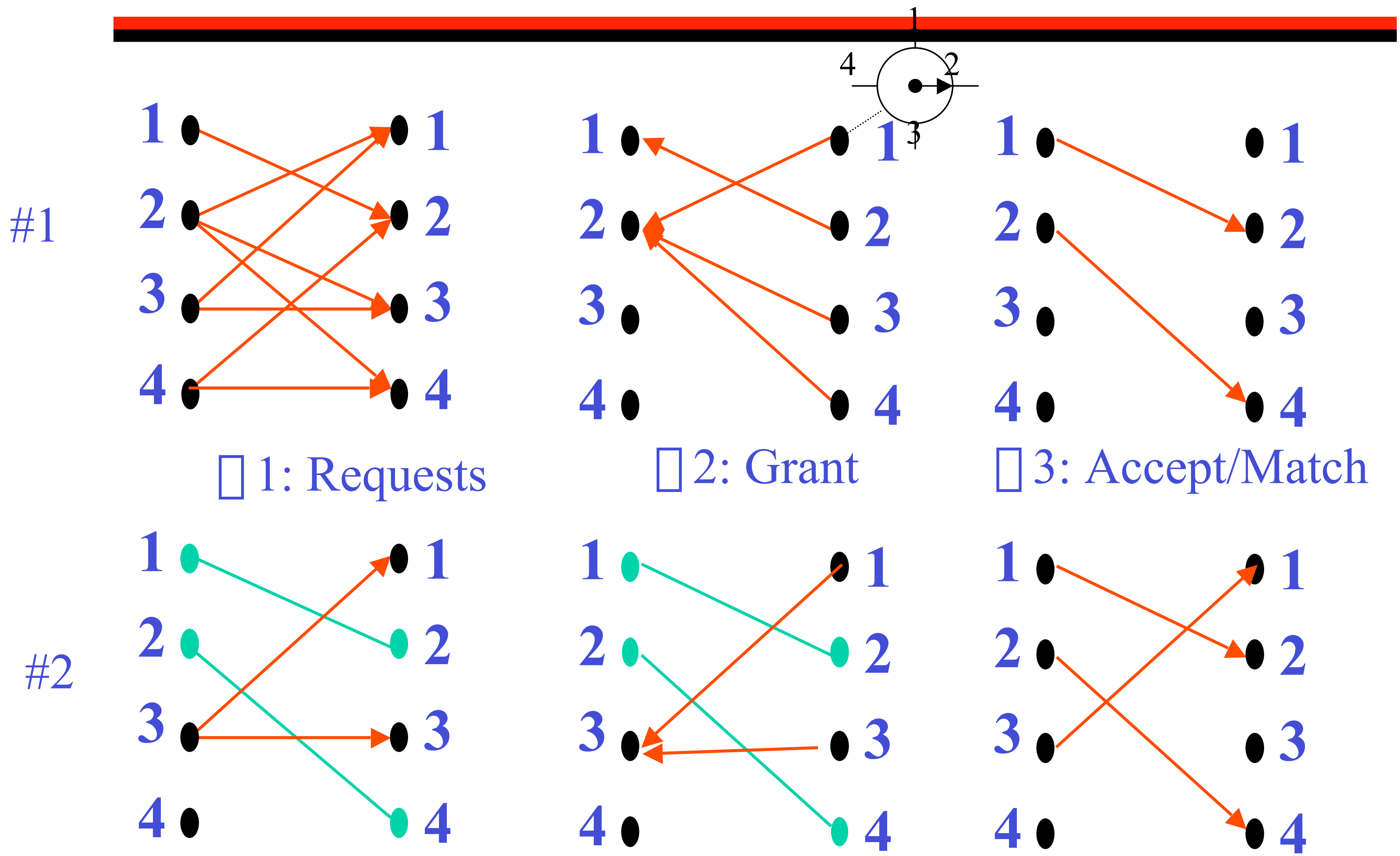
Parallel Iterative Matching



Parallel Iterative Matching



*i*SLIP



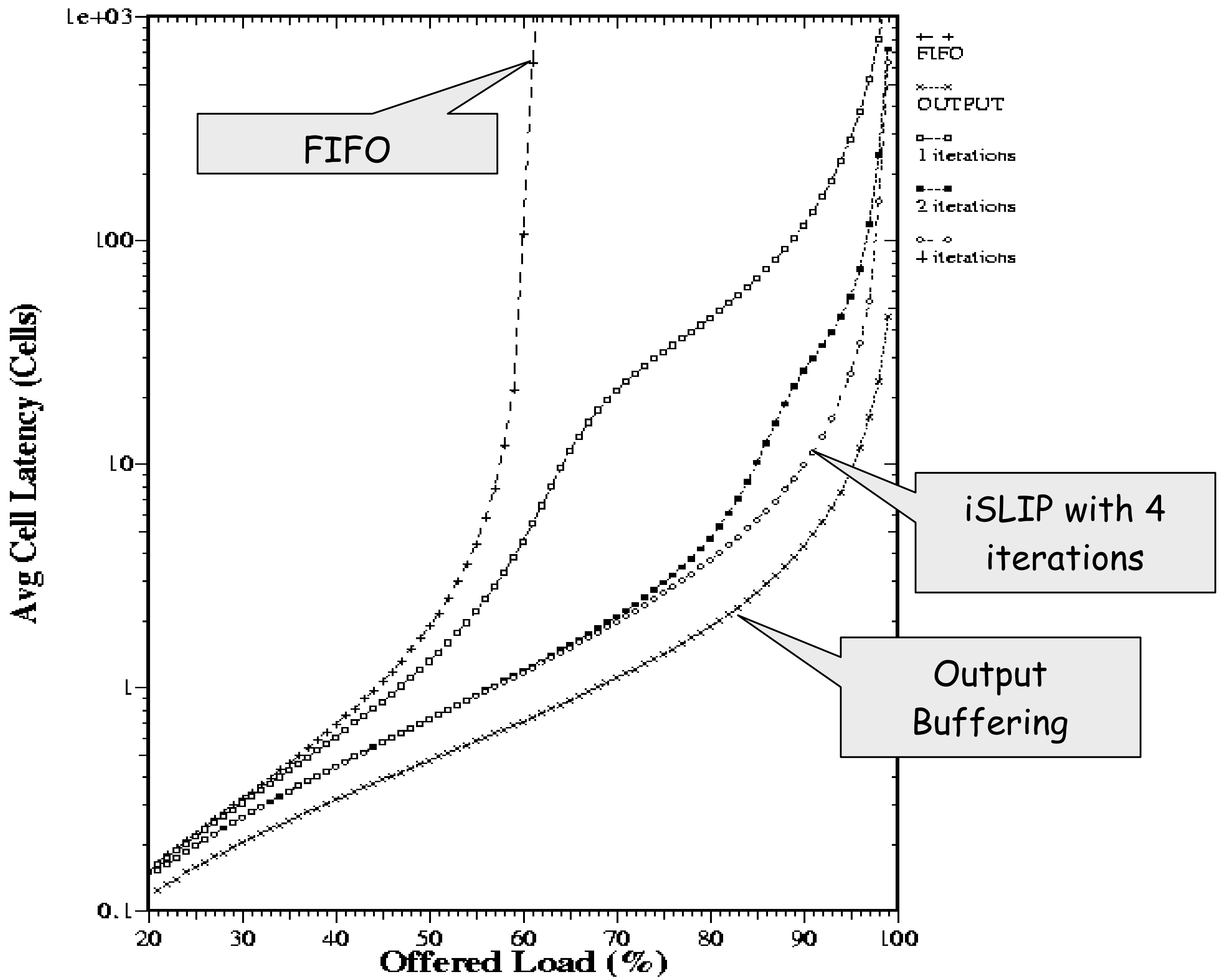
*i*SLIP Operation

- **Grant phase:** Each output selects the requesting input at the pointer, or the next input in round-robin order. *It only updates its pointer if the grant is accepted.*
- **Accept phase:** Each input selects the granting output at the pointer, or the next output in round-robin order.
- Consequence: Under high load, grant pointers tend to move to unique values.

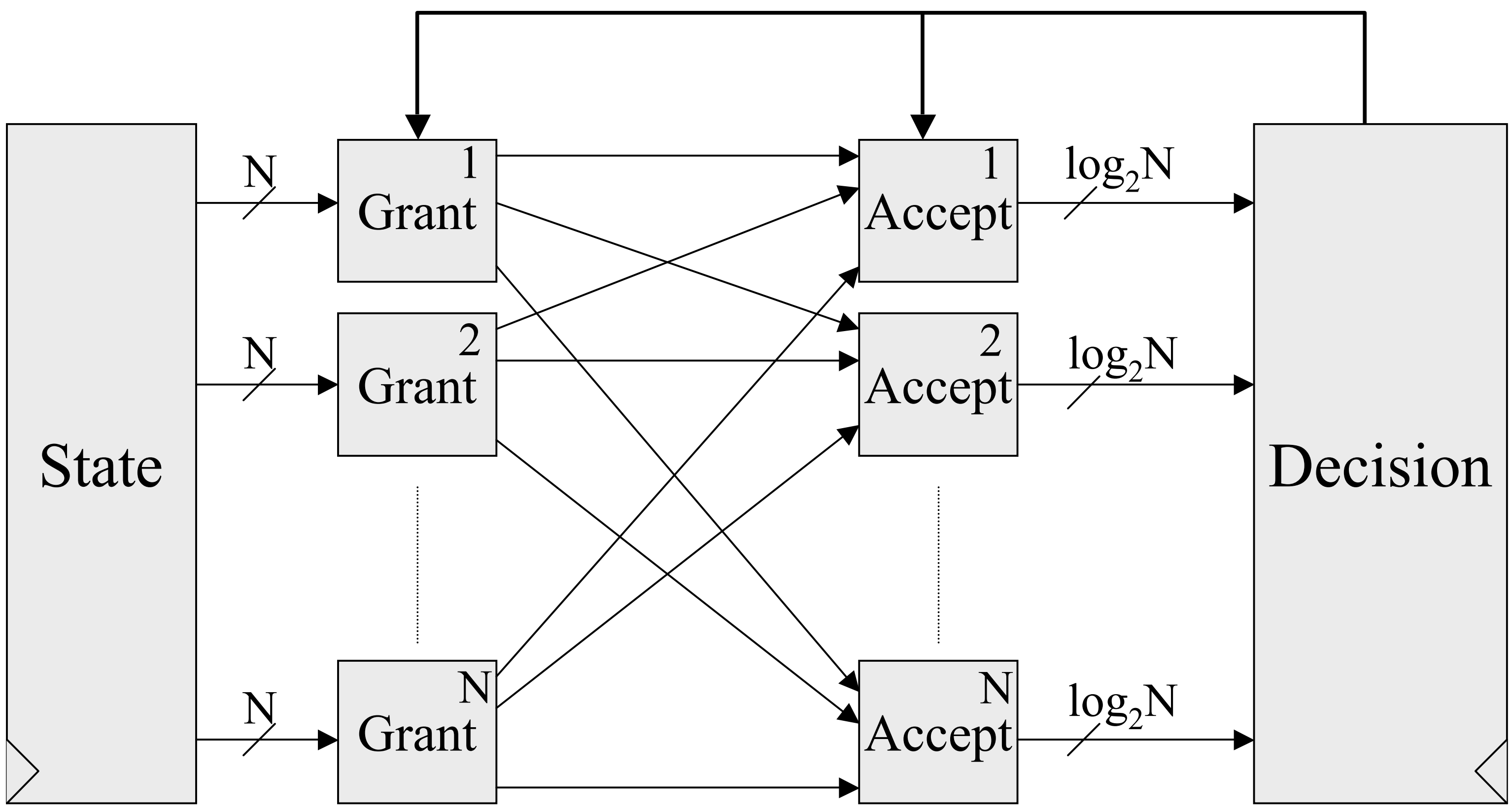
*i*SLIP
Properties

- Random under low load
- TDM under high load
- Lowest priority to MRU
- 1 iteration: fair to outputs
- Converges in at most N iterations. (On average, simulations suggest $< \log_2 N$)
- Implementation: N priority encoders
- 100% throughput for uniform i.i.d. traffic.
- But…some pathological patterns can lead to low throughput.

*i*SLIP


FIFO
iSLIP with 4 iterations
Output Buffering
Avg Cell Latency (Cells)
Offered Load (%)
FIFO
OUTPUT
1 iterations
2 iterations
4 iterations
1e+03
100
10
1
0.1
20
30
40
50
60
70
80
90
100

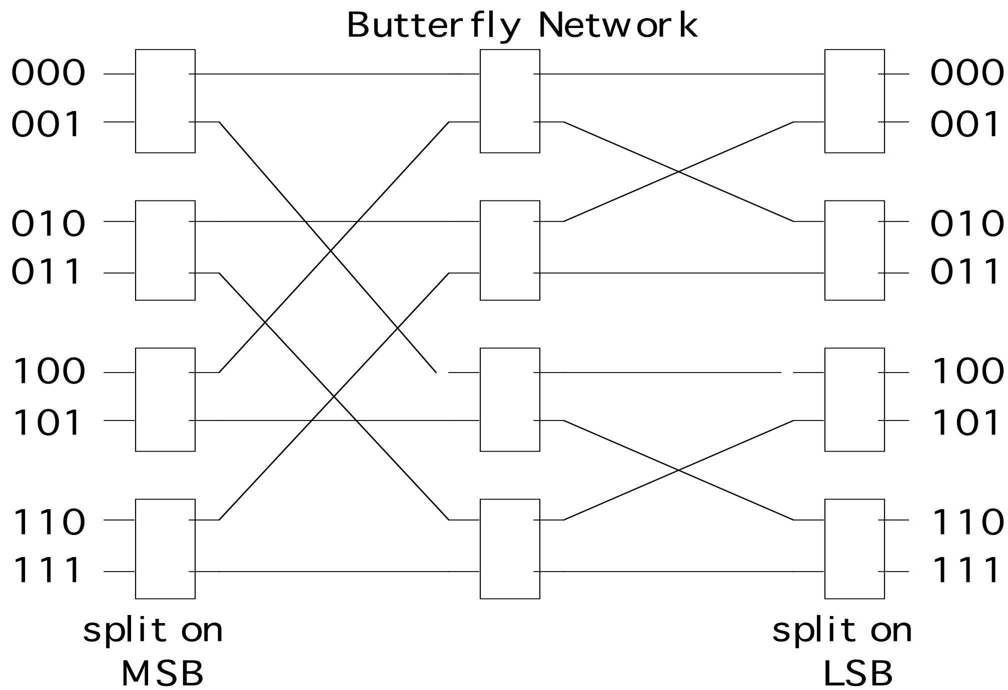
*i*SLIP
Implementation



Alternative Switching

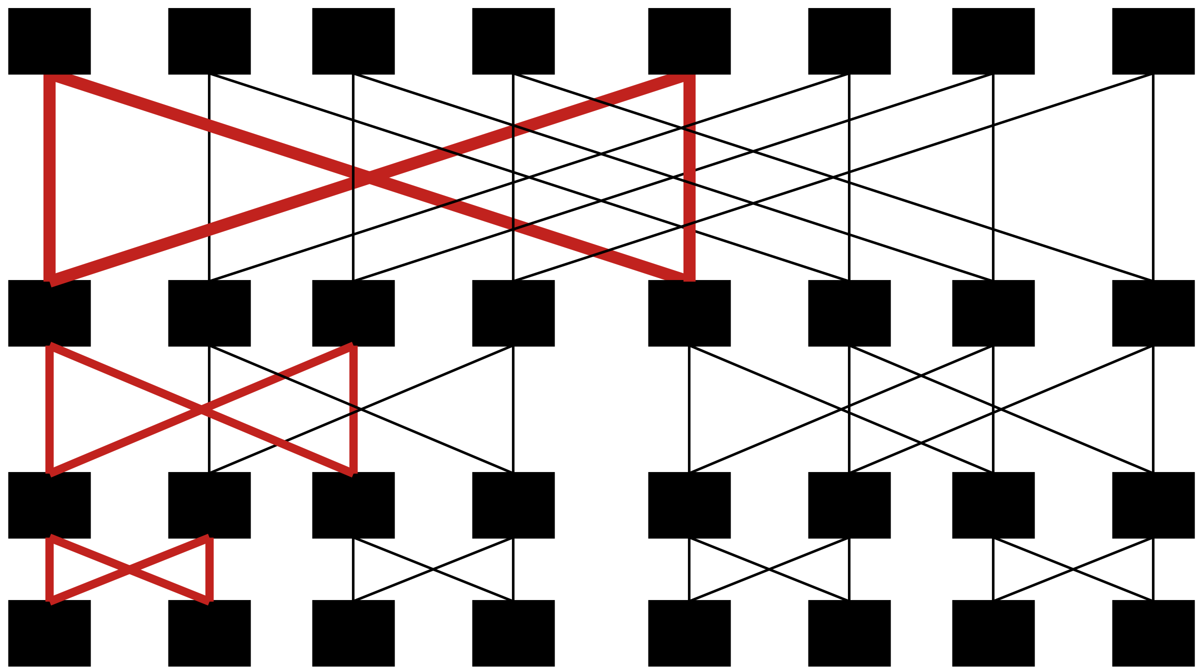
- Crossbars are expensive
- Alternative networks can match inputs to outputs:
  - Ring
  - Tree
  - K-ary N-cubes
  - Multi-stage logarithmic networks
    - Each cell has constant number of inputs and outputs

Example: Butterfly



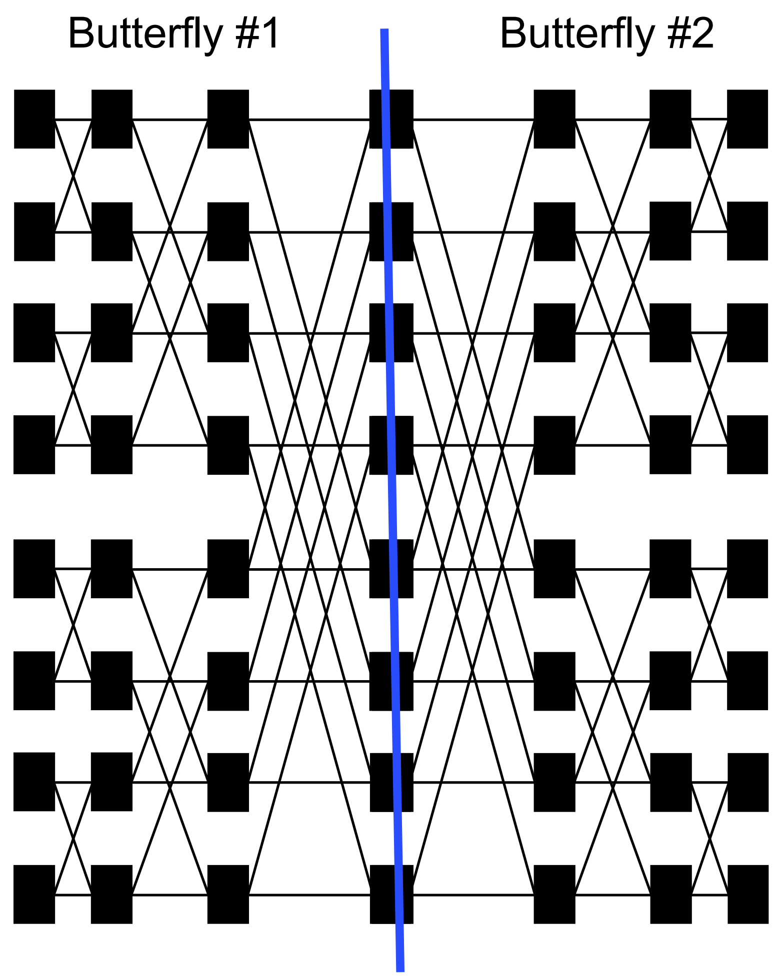
Butterfly

Outputs



Inputs

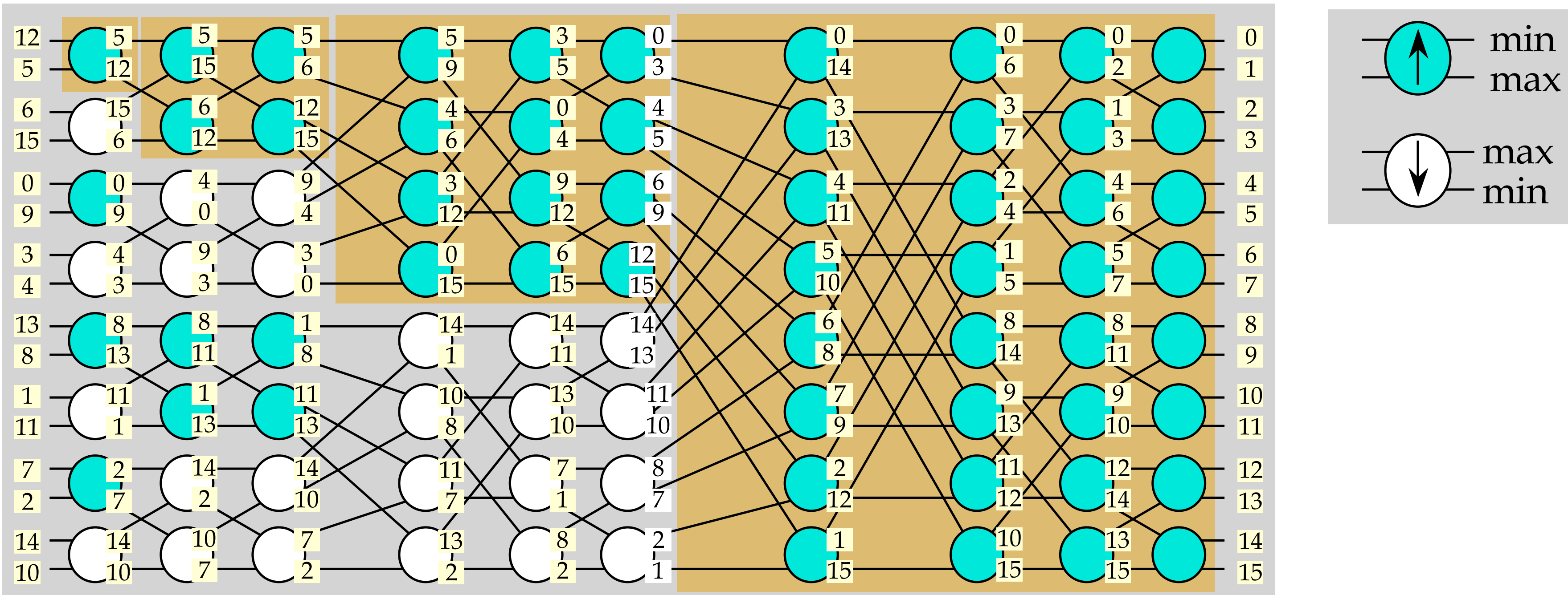
Benes Network



Benes networks

- Any permutation has a conflict free route
  - Useful property
  - Offline computation is difficult
- Can route to random node in middle, then to destination
  - Conflicts are unlikely under uniform traffic
  - What about conflicts?

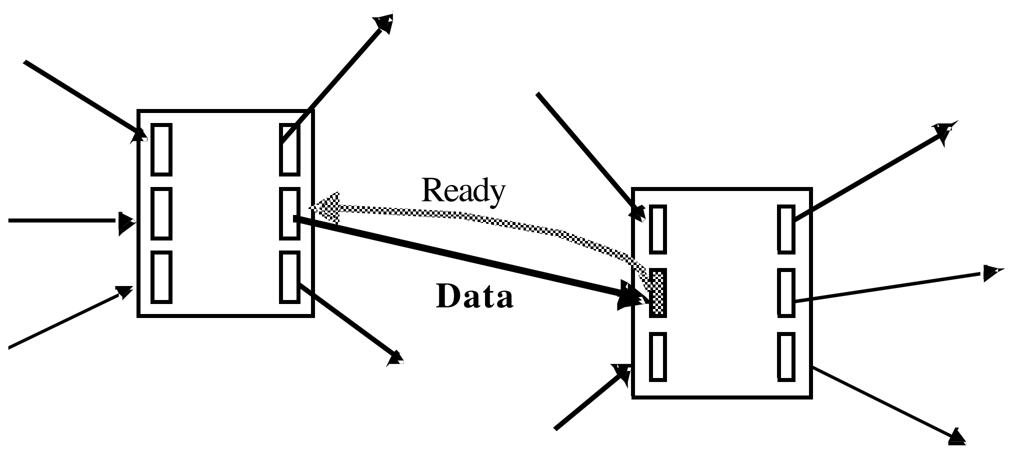
Sorting Networks



- Bitonic sorter recursively merges sorted sublists.
- Can switch by sorting on destination.
  - additional components needed for conflict resolution

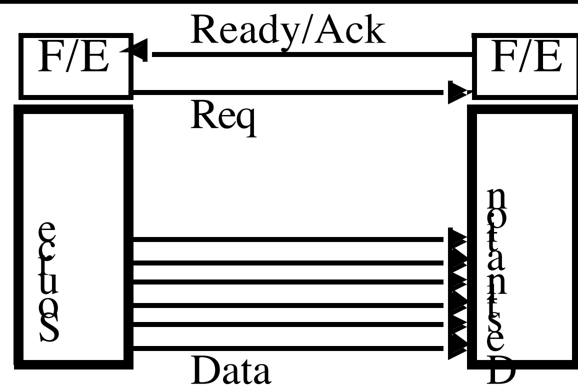
Flow Control

- What do you do when push comes to shove?
  - ethernet: collision detection and retry after delay
  - FDDI, token ring: arbitration token
  - TCP/WAN: buffer, drop, adjust rate
  - any solution must adjust to output rate
- Link-level flow control

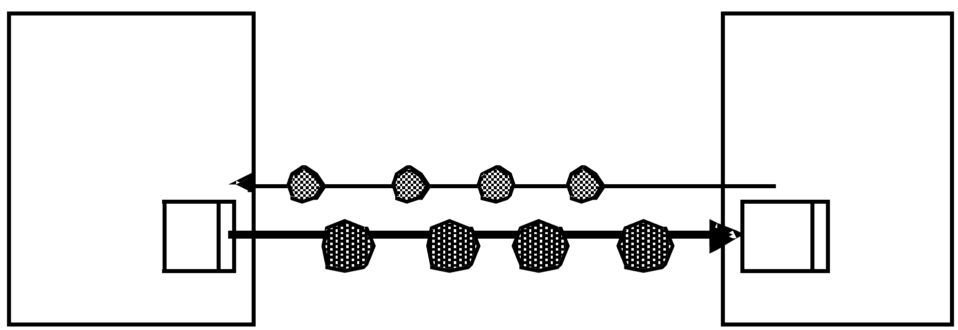


Link Flow Control Examples

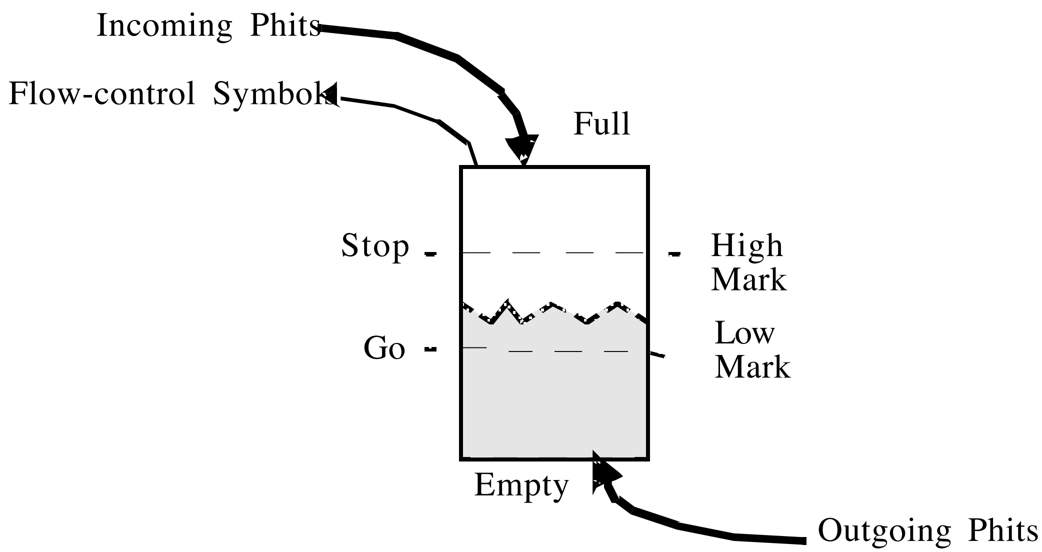
- Short Links



- long links
  - several flits on the wire



Smoothing the flow



- How much slack do you need to maximize bandwidth?