

CSE 123b
Graduate Networking

Spring 2003

Last Lecture: Potpourri

Stefan Savage

Final

- Check Studentlink to be sure, but I believe that its June 11th 3-6pm in this room.
- Closed book
- You can bring on 8.5x11 sheet of paper and you can do just about anything to that piece of paper you want (i.e. write on both sides, print on it, etc)
 - ◆ You can't tape your textbook to the sheet of paper though
 - ◆ Or your laptop
- **Same style as midterm**
 - ◆ FYI: You may be asked questions about project

Review session

- **Monday, June 9th, 4:00-5:30pm in AP&M 4301**
- **I will also be available on the 10th from 2-3:30 (normal class hours)**
 - ♦ Or by appointment

Stuff we'll look at today

- **Skim**
 - ◆ IPv6
 - ◆ IPSec
 - ◆ Active Networks
 - ◆ Sensor Networks

- **A bit more detail:**
 - ◆ Network measurement
 - ◆ Router design

IPSEC

- **Security at lowest layer: IP**
- **Framework for security**
 - ◆ Select different encryption algorithms, security protocols
 - ◆ Select security services (e.g., integrity, authentication, etc.)
 - ◆ Select granularity (e.g., connection, all end-point flows)
- **Two parts**
 - ◆ Authentication Header (AH)
 - » Access control, integrity, authentication, anti-replay
 - ◆ Encapsulating Security Payload (ESP)
 - » Confidentiality in addition to above
 - » Relies upon an encryption algorithm

IPSEC (2)

- **Security Association (SA) binds AH and ESP**
 - ◆ The “association” defines a set of security services between end-points (security gateways)
 - ◆ Simplex – only defined for a single direction
 - » Need two for both directions of a connection
- **Negotiation**
 - ◆ Internet Security Association and Key Management (ISAKMP)
 - » Procedures and formats to establish, negotiate, modify, delete security associations
 - » Again, a framework
 - ◆ Internet Key Exchange (IKE)
 - » Specific protocol for exchanging keys

IPv6

- **Originally motivated by crisis in IP address space**
 - ◆ 32 bits not enough if everything gets an IP address
 - ◆ Subnetting and CIDR helped alot, but not a long-term solution (mobile phones)
- **Solution: Increase the size of IP addresses**
 - ◆ Originally to 64, then to 128 bits
 - ◆ Requires changes to IP header
- **If we're going to change the header, might as well change other aspects of IP**
- **Rule of thumb: IPv6 = IPv4 + big addresses + IPSEC, Mobile IP + DHCP + ARP + improvements to these**

IPv6 (2)

- **IP addresses are 128 bits**
 - ◆ Unicast, multicast, local, etc.
 - ◆ 1234:ABCD:1234:ABCD:1234:ABCD:1234:ABCD
 - ◆ Registry, provider, subscriber, subnetwork, interface
 - ◆ Interface ID used as lowest 6 bytes
- **Simplified headers**
 - ◆ 40 bytes (20 for IPv4)
 - ◆ Use header extensions for options
- **Autoconfiguration**
 - ◆ Multiple addresses per host (link-local address)
- **FlowLabel, priority, security, mobility**

IPv6 and IPv4 Interoperability

- **Dual-stack operation**
 - ♦ IPv6 nodes that support both IPv4 and IPv6
- **Tunneling**
 - ♦ IPv6 packets encapsulated within IPv4 packets
 - ♦ End-points speak IPv6, but use IPv4 packet to use standard Internet routing
 - ♦ Easy mapping if IPv6 embeds IPv4 address, otherwise need to configure a table

Active Networks

- **Problem: How can we change the network without replacing the network?**
 - ♦ Routing algorithms (multicast), queueing disciplines (WFQ, RED), mobility, measurement, etc.
- **Approach: Active Networks**
 - ♦ Make routers **programmable**
 - ♦ Example:
 - » Packets carry programs or pointers to programs\
 - » Programs executed on arrival of packets
 - ♦ Need platform, execution environment, resource control, security, storage, etc. – essentially, a whole new OS
 - ♦ Many different incarnations, some small testbeds
 - ♦ Controversial (security, management, overhead)
 - ♦ A point in the design space of non-“client/server” models

Sensor Networks

- **Scenario**
 - ◆ Hundreds or thousands of low-powered wireless sensors distributed haphazardly over an area
 - ◆ Applications to environmental monitoring, object tracking, etc.
- **Research problems**
 - ◆ How to communicate information through network efficiently and with lower overall power (**application-specific** multihop)
 - ◆ How to locate nodes and provide time sync among them
 - ◆ How to write an application for such a network

Network Measurement

- **The Internet is an artifact of sufficient complexity that we don't understand it**
 - ◆ Difficult to measure many things directly -> inference
 - ◆ Too big to measure all of it
 - ◆ No such thing as “typical” (sampling difficult)
 - ◆ Constantly changing
- **A whole community focused on these problems**
 - ◆ Packet characteristics: application usage; dynamic behavior of existing protocols;
 - ◆ Path characteristics: delay, loss rate, queuing, bandwidth [end-to-end vs hop-by-hop]
 - ◆ Network topology: connections between routers, between ISPs; graph analysis of Internet topologies
- **Big challenge: no support for measurement**

For example:

- How to measure one-way **network path measurements** (e.g. **packet loss rate**)?
 - ◆ Critical for network performance analysis
 - ◆ Requires measurement from **both** endpoints
- **Remote hosts are indifferent to problem**
 - ◆ **Unrealistic** to deploy new software
 - ◆ Remote hosts may block measurements

Key idea

- **Exploit services needed by remote host**
 - ◆ Web, E-mail, news, file transfer, etc.
 - ◆ All use *Transmission Control Protocol (TCP)*
 - » Reliable, in-order, data transfer protocol
- **Exploit standard protocol behavior**
 - ◆ TCP has rich specification
 - ◆ Can be leveraged to perform measurements

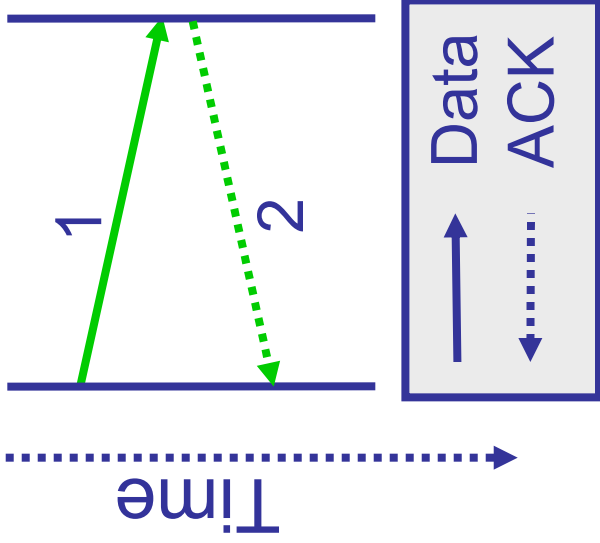
Simple example: Deducing one-way loss rate

- Send TCP data packets to remote host
- What we know
 - ◆ Number of data packets sent
 - ◆ Number of acknowledgments received
- What we need to know
 - ◆ How many data packets were received?
 - » Extract from ACKs; TCP is a reliable protocol
 - ◆ How many acknowledgments were sent?
 - » Arrange that **one ACK** sent for each data packet

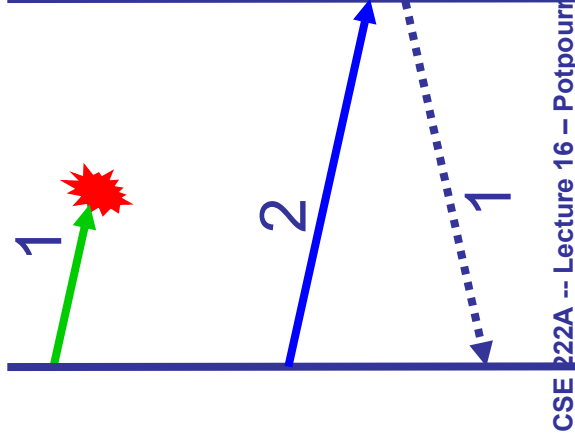
How TCP reveals packet loss

- Data packets ordered by sequence #'s
- ACK packets specify next seq# expected

Nothing lost

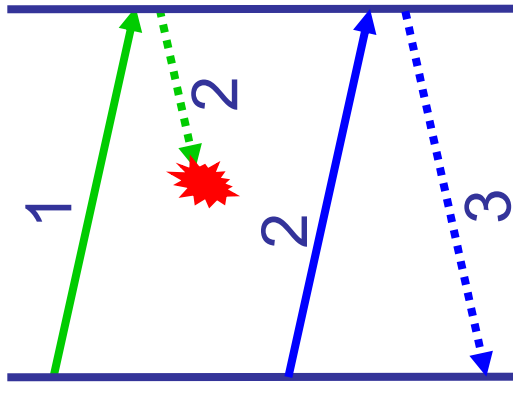


Data lost



CSE 222A -- Lecture 16 -- Potpourri

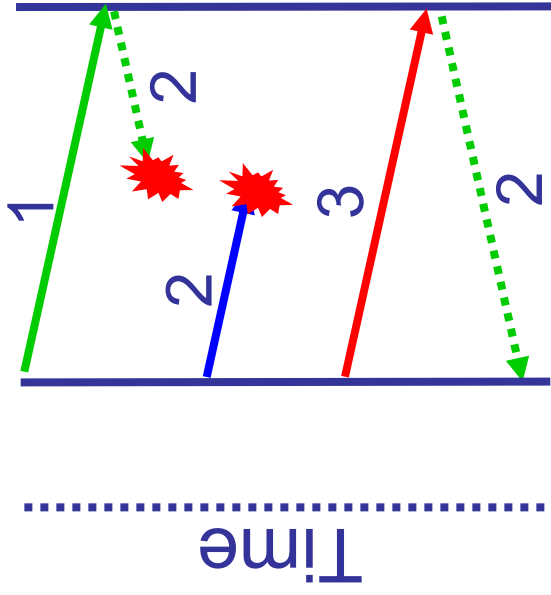
ACK lost



Thanks to Nick Weaver for some slides

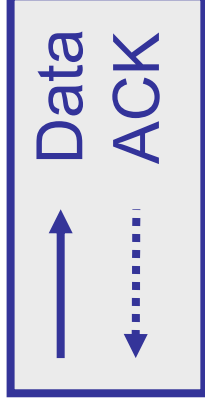
Loss deduction example

Measurement

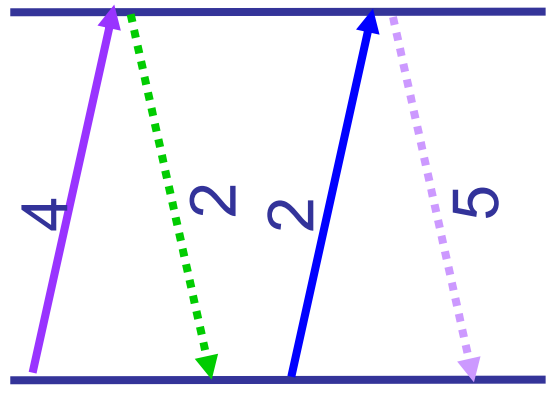


dataSent = 3

ackReceived = 1



Loss deduction



dataLost = 1

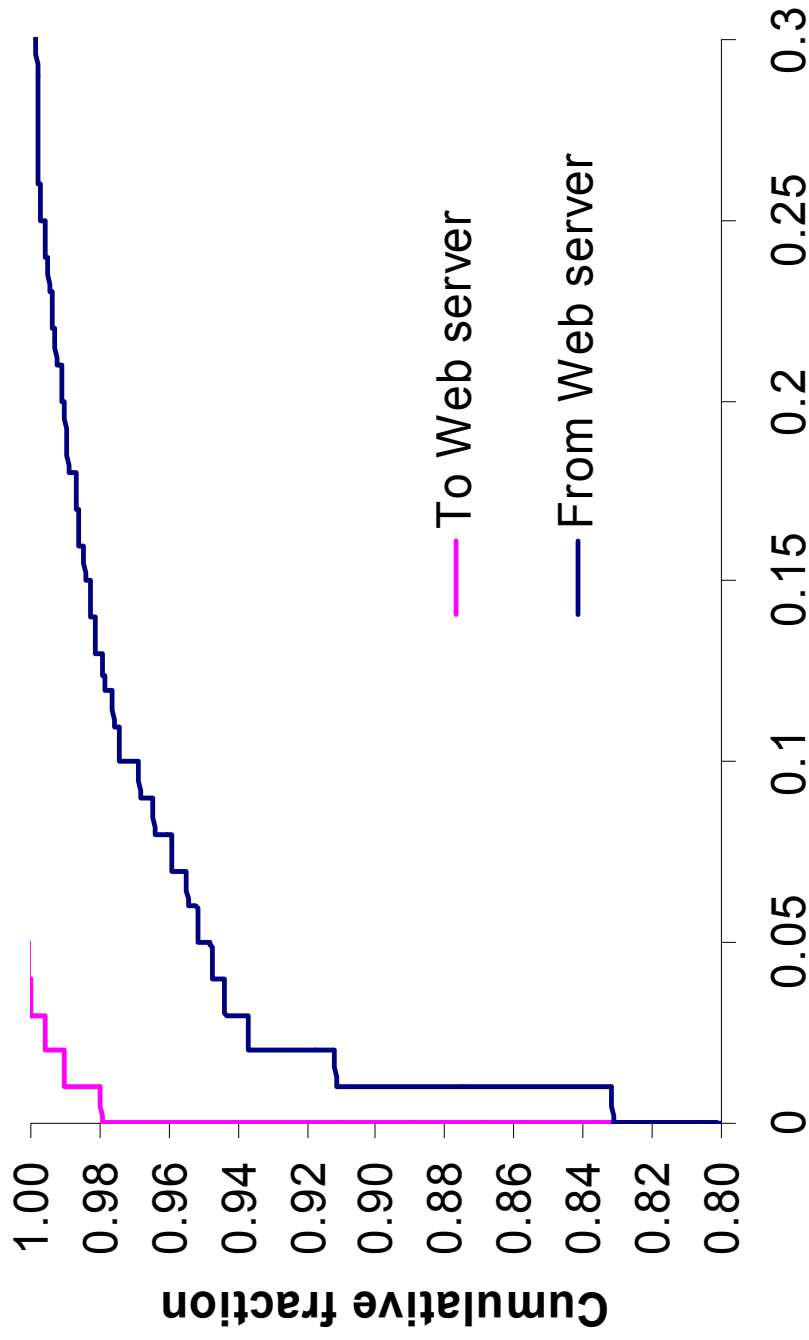
dataReceived, ackSent = 2

$$\text{Loss}_{\text{data}} = 1 - (\text{dataReceived}/\text{dataSent}) = 33\%$$

$$\text{Loss}_{\text{ack}} = 1 - (\text{ackReceived}/\text{ackSent}) = 50\%$$

Experimental finding: Packet loss is highly asymmetric

25 Popular Web servers



What's in a router? (how to make a very fast router?)

- **Physical components**
 - ◆ One or more **input interfaces** that receive packets
 - ◆ One or more **output interfaces** that transmit packets
 - ◆ A chassis (box + power) to hold it all
- **Functions**
 - ◆ **Forward** packets
 - ◆ **Drop** packets (congestion, security, QoS)
 - ◆ **Delay** packets (QoS)
 - ◆ **Transform** packets? (Encapsulation, Tunneling)

What a router does: the normal case

- Receive incoming packet from link input interface
- Lookup packet destination in forwarding table
 - ◆ (destination, output port(s))
- Validate checksum, decrement ttl, update checksum
- Buffer packet in input queue
- Send packet to output interface (interfaces? Mcast)
- Buffer packet in output queue
- Send packet to output interface link

What a router looks like?

Cisco 2500



Capacity: <10Mbps

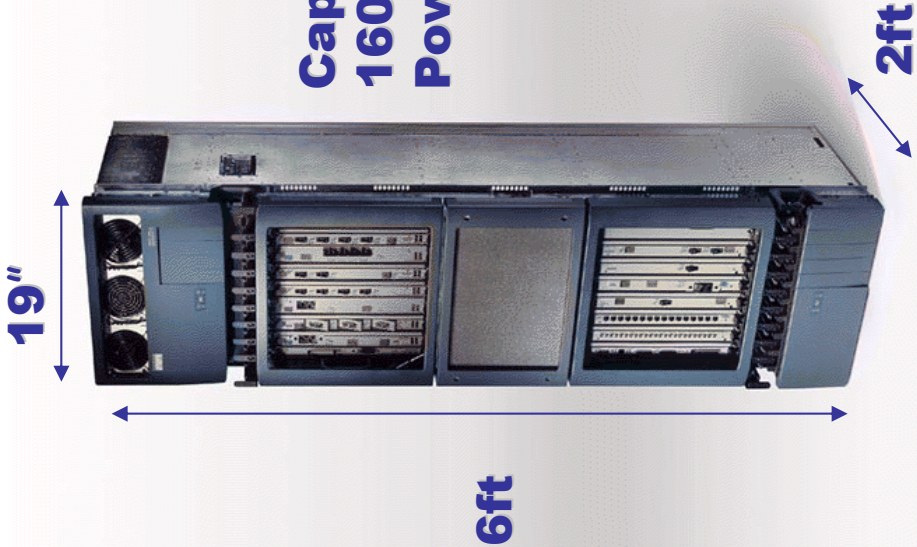
Linksys DEFSR81



Capacity: <10Mbps

What a router looks like (2)

Cisco GSR 12416



Juniper M160



What a router looks like (3)

Pluris Teraplex 20 w/7 Racks



**Capacity:
>1Tb/s**

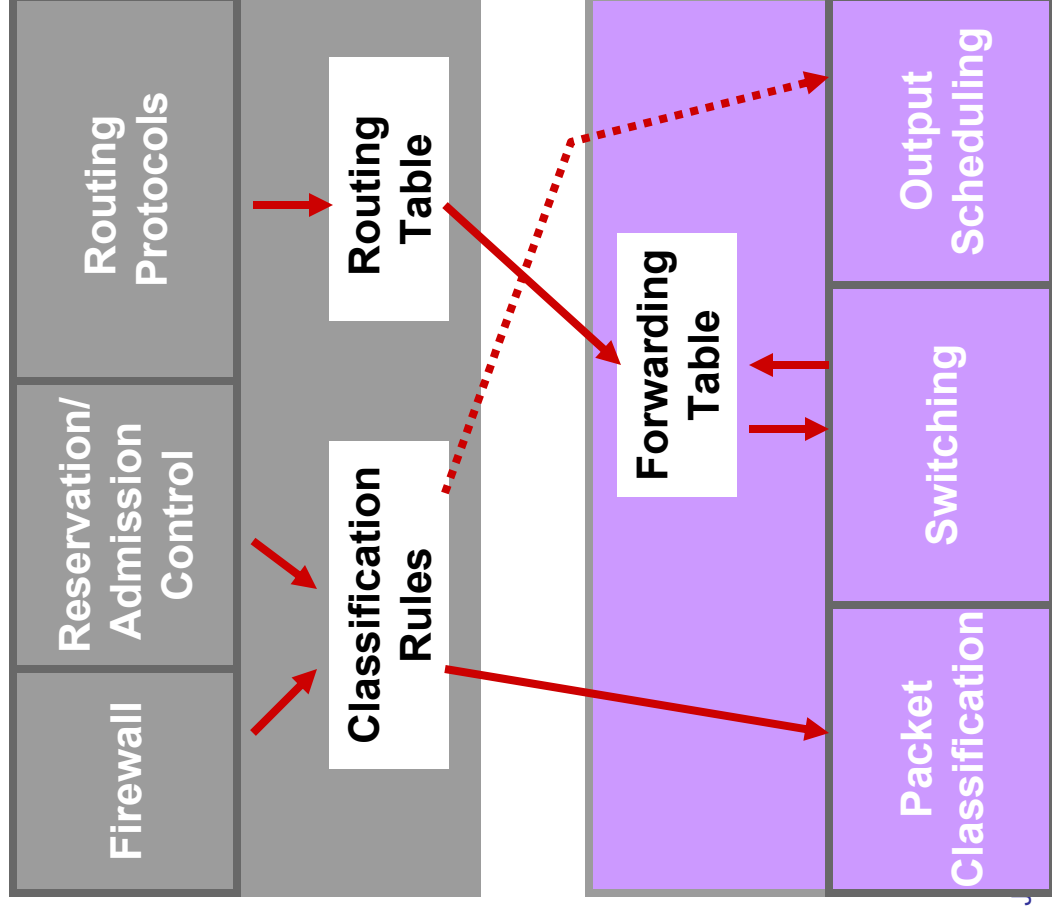
**Power: 45kW
(~250 homes?)**

1 room

High-performance routers

- **Geared to core and distribution service needs**
 - ♦ Requirements: high speed & high density
- **Why do we care?**
 - ♦ Moore's Law slower than link speed growth (and BW demand)
 - » OC48c (2.5Gbps), now, 128ns/packet
 - » OC192c (10Gbps), in deployment, 33ns/packet
 - » OC768c (40Gbps), 2002-3, 8ns/packet
 - ♦ Need high density/low power to manage POP complexity
 - » \$20-100k & 2-400W per port, 50% ports frequently for internal connectivity
 - » DWDM can help with the former, but requires more interfaces

Functional architecture



- Control Plane**
- Complex
 - Per-control action
 - May be slow

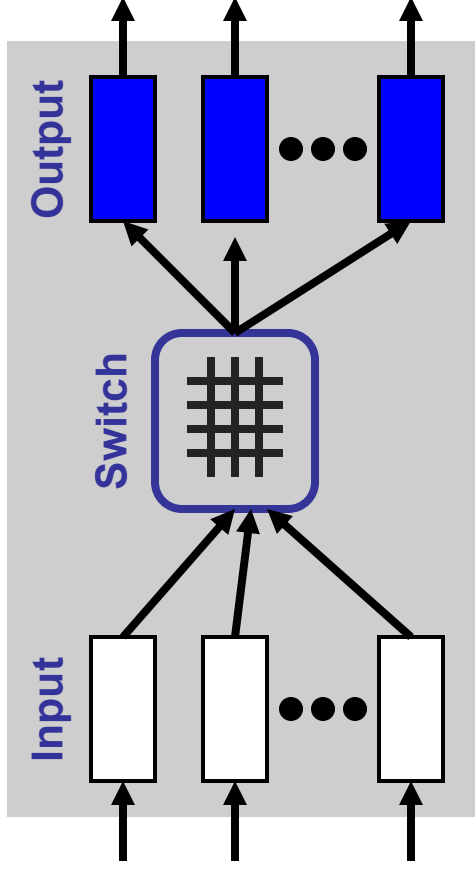
- Data plane**
- Simple
 - Per-packet
 - Must be fast

Packet classification

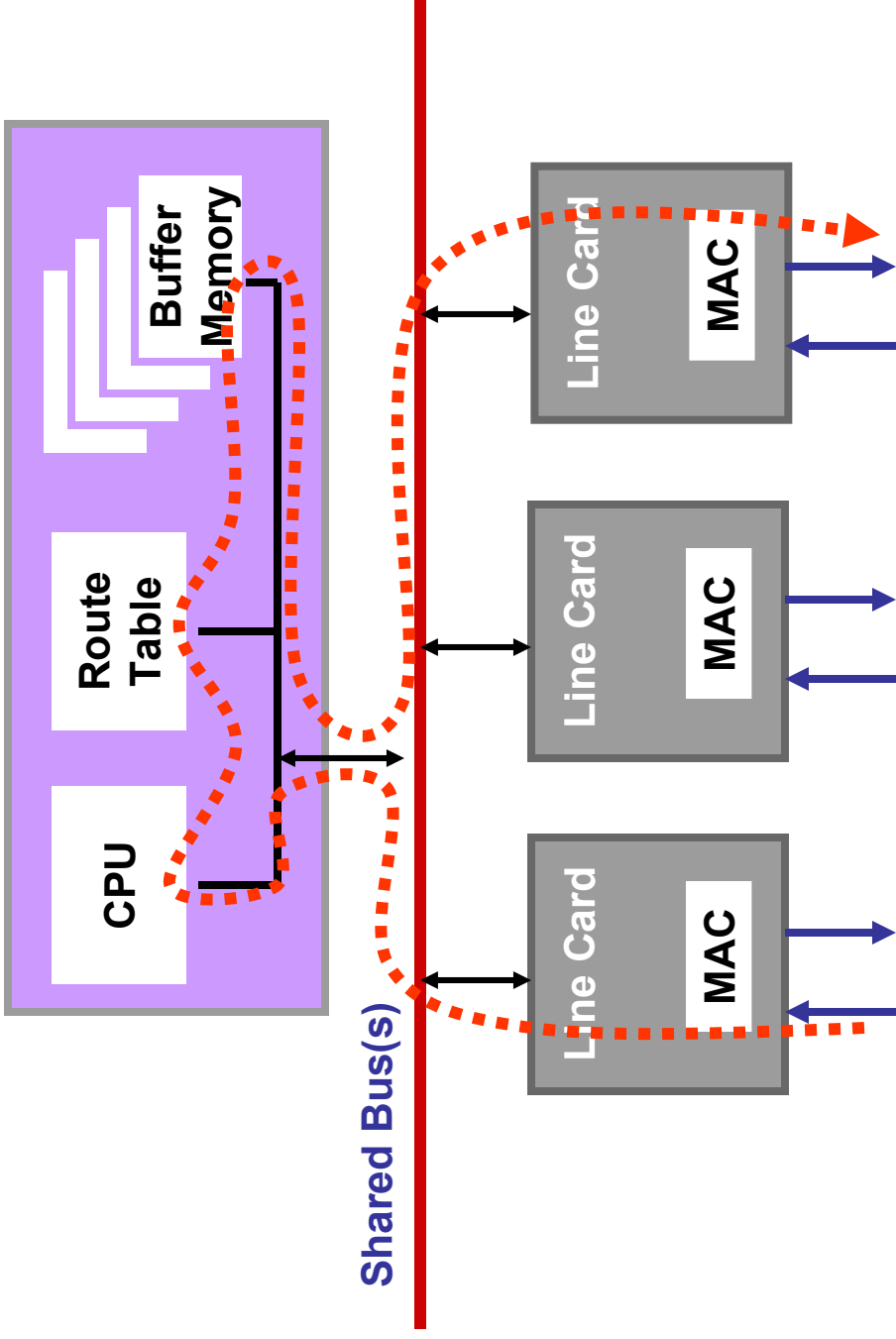
- **Forwarding**
 - ◆ Longest prefix match of destination against forwarding table
 - ◆ Returns (output port, Next-hop MAC header) tuple
 - ◆ Key issue: forwarding table growth
 - ◆ George will talk about this next time
- **QoS tagging**
 - ◆ Certain traffic tagged with higher priority
 - ◆ Per flow (src ip, src port, dst ip, dst port), pre source or dest prefix, per protocol (Napster, etc...)
- **Firewall rules**
 - ◆ Block access to TCP packets with dst port != 80

Interconnect architecture

- **Input & output connected via switch fabric**
- **Kinds of switch fabric**
 - ◆ Bus
 - ◆ Crossbar
 - ◆ Shared Memory
- **How to deal with transient contention?**
 - ◆ Input queuing
 - ◆ Output queuing

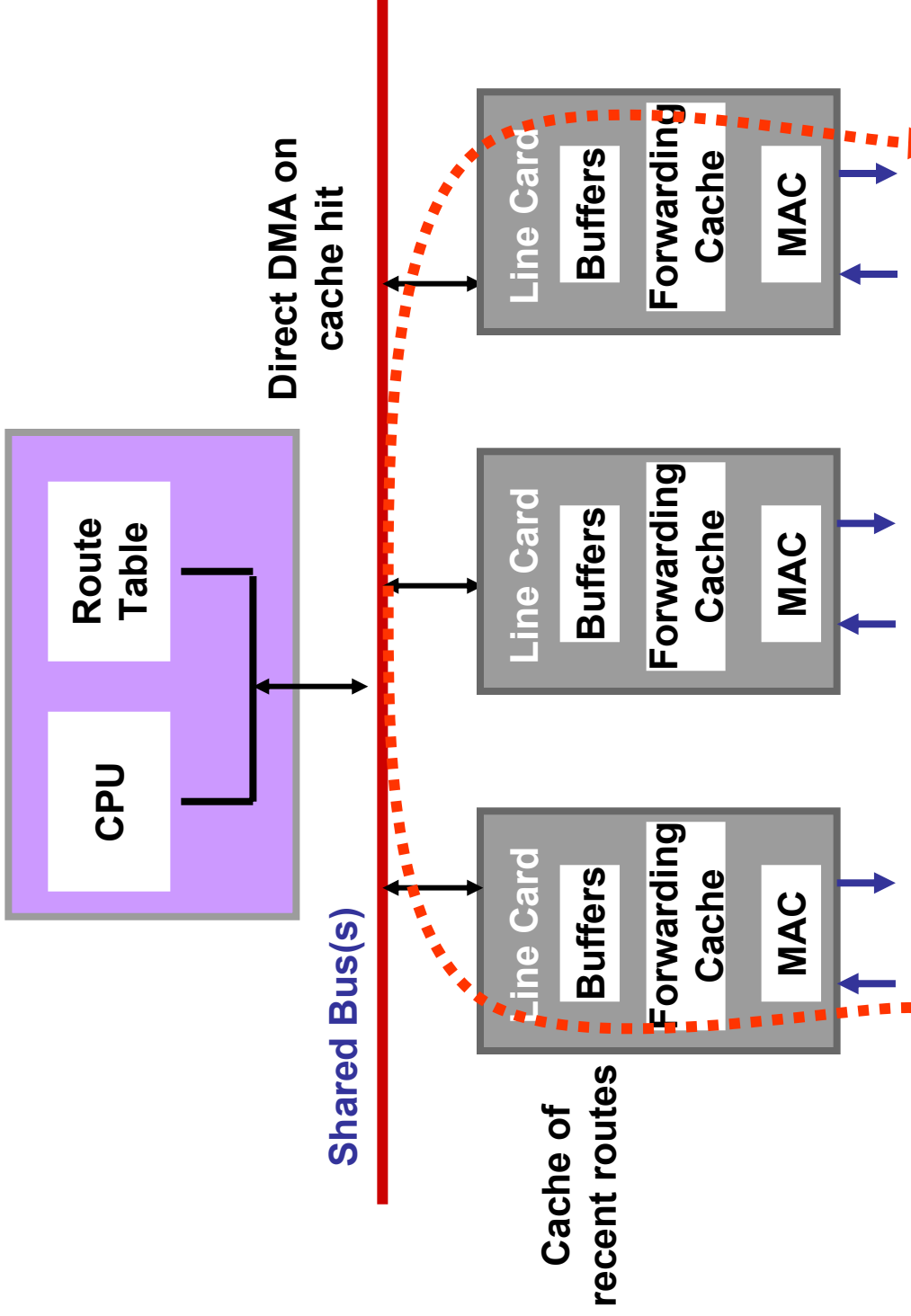


First Generation Routers

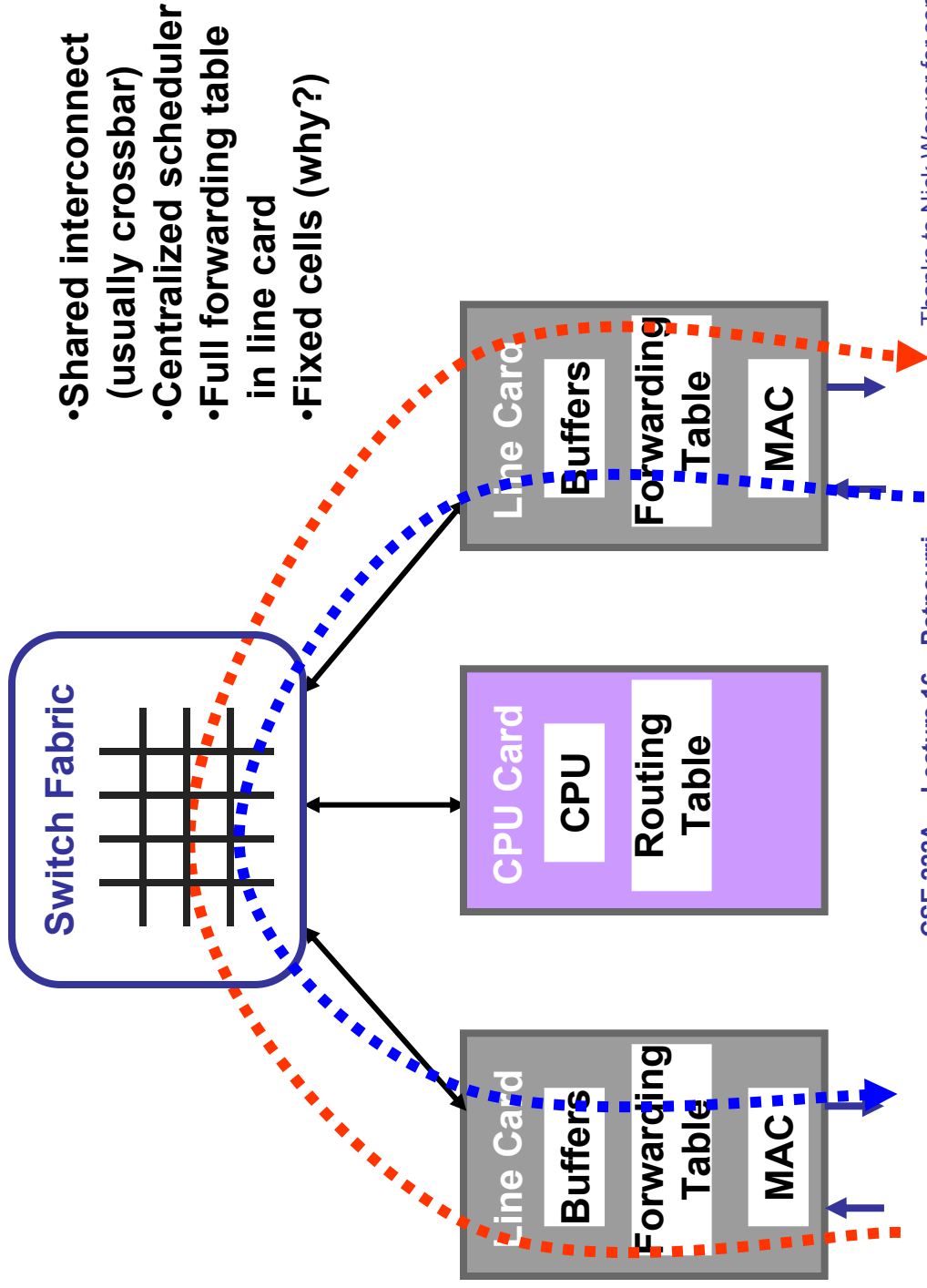


- Single CPU and shared memory;
- All classification by main CPU

Second Generation Routers

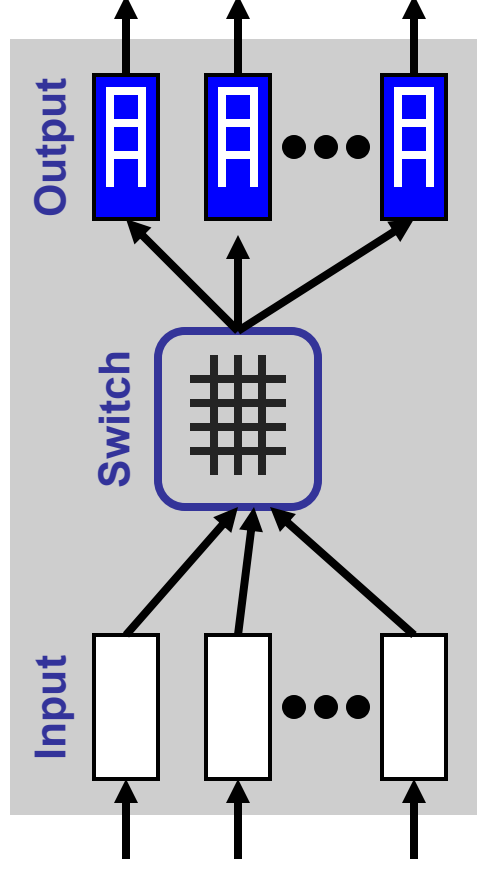


Third Generation Routers



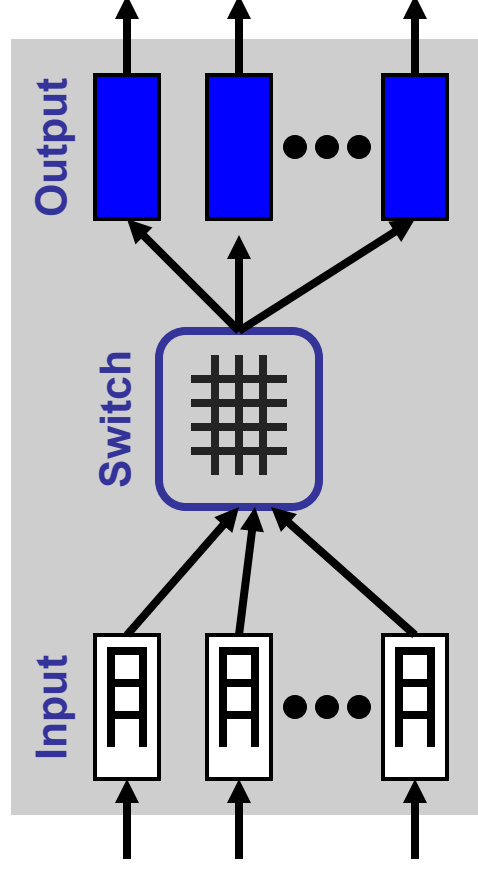
Output queuing

- Output interfaces buffer packets
- **Pro**
 - ♦ Simple algorithms
 - ♦ Single congestion point
- **Con**
 - ♦ N inputs may send to the same output
 - ♦ Requires speedup of N

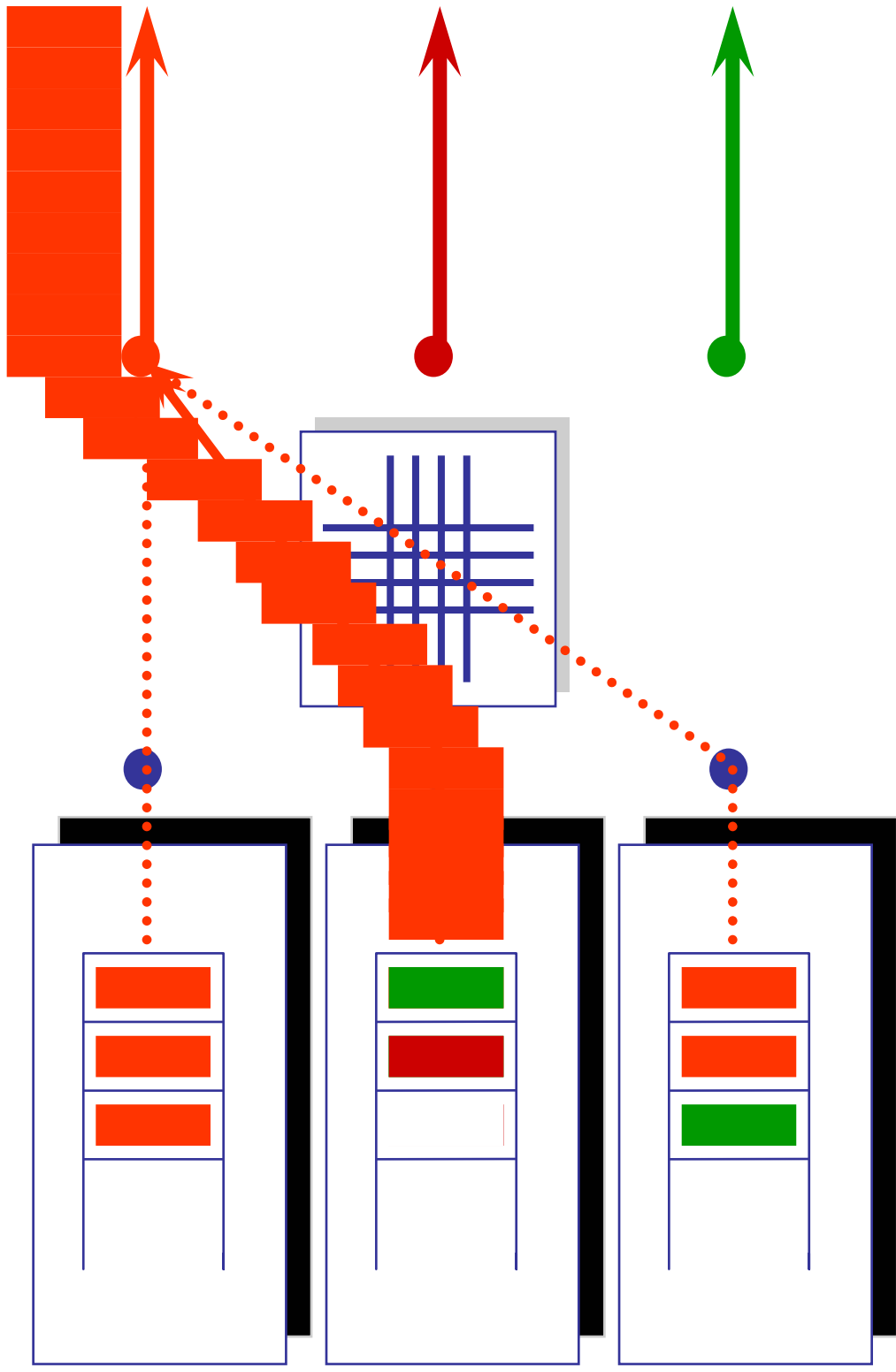


Input queuing

- Input interfaces buffer packets
- **Pro**
 - ◆ Single congestion point
 - ◆ Simple to design algorithms
- **Con**
 - ◆ Must implement flow control
 - ◆ Low utilization due to Head-of-Line (HoL) Blocking
 - » Utili limited to 2- $2^{.5}=58\%$

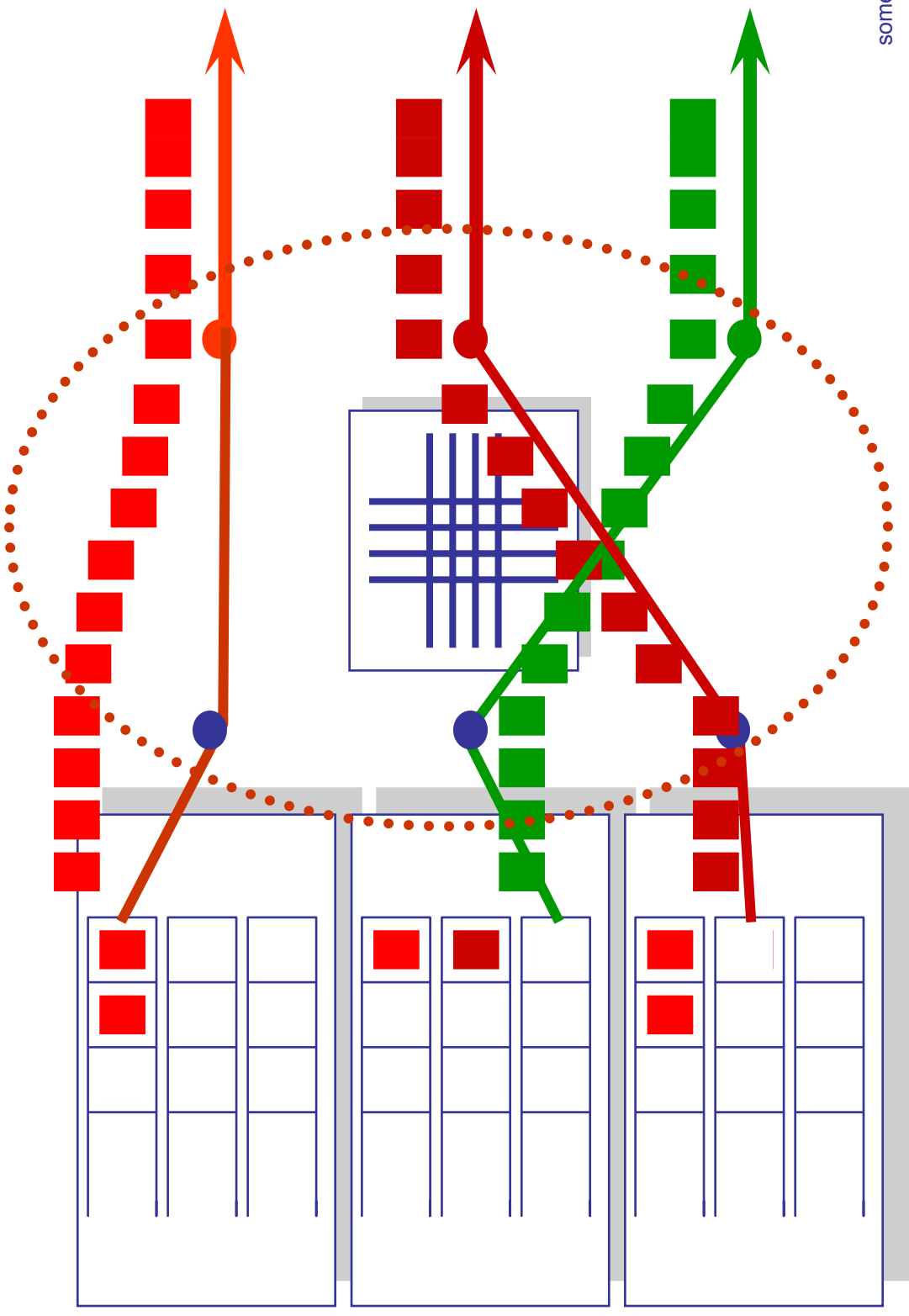


Head-of-Line Blocking



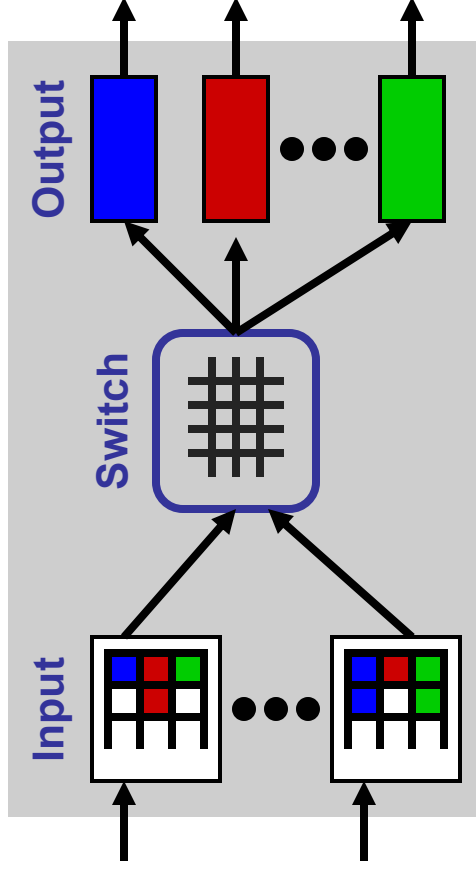
Virtual Output Queues

(courtesy Nick McKeown)



IQ + Virtual Output Queuing

- Input interfaces buffer packets in per-output virtual queues
- **Pro**
 - ◆ Solves blocking problem
- **Con**
 - ◆ More resources per port
 - ◆ Complex arbiter
 - ◆ Still limited by input/output contention (scheduler).
 - ◆ RR: $1=1/e = 63\%$

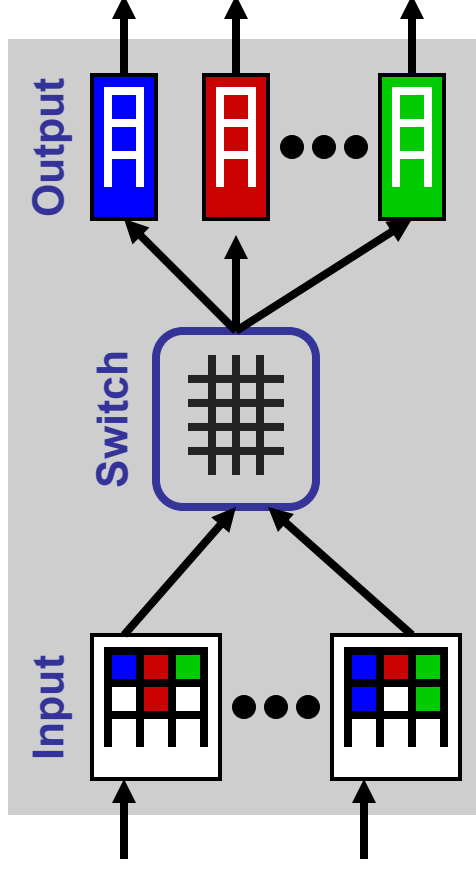


Switch scheduling

- **Problem**
 - ♦ Match inputs and outputs
 - ♦ Resolve contentions
 - ♦ No output packet drops
 - ♦ Maximize throughput
 - ♦ Do it in constant time...
- **Many algorithms for uniform traffic assumption**
 - ♦ E.g. TDM, Maximum size bipartite match
 - ♦ Approximate answers (e.g. iSLIP, submaximal match)
- **Recent result (Dai et al, 2000)**
 - ♦ Maximal size matching + speedup of two guarantees 100% utilization for most traffic assumptions

Modern router

- **IQ + VoQ + OQ**
 - ◆ Speedup of 2
 - ◆ Central scheduler
 - ◆ Fixed-sized internal cells
- **Pro**
 - ◆ Can achieve utilization of ~ 1
 - ◆ Can scale to multiple Tb/s
- **Con**
 - ◆ Multiple congestion points
 - ◆ Complexity



Typical function breakdown

- **Input interface**
 - ◆ Forwarding
 - ◆ Virtual output queuing
- **Switch**
 - ◆ Scheduling input interface requests to output interfaces
 - ◆ Multicast scheduling
- **Output interface**
 - ◆ Queue packets for transmission
 - ◆ Classification
 - ◆ Buffer management (which pkt to drop)
 - ◆ Scheduling (which pkt to send from buffer)

Next bottlenecks

- **Buffering at high speed**
 - ◆ SRAM density too low for $BW \cdot D$ of 40Gbps link
 - ◆ DRAM too slow
 - ◆ SRAM memory management as cache for DRAM
- **Scheduler and arbiter overhead**
 - ◆ Limits size of switch and link BW
 - ◆ Two-state switch (Chang et al, 2000); no scheduler
- **High density (100's-1000's of line cards)**
 - ◆ Physical distance to support density; electrical links degrade
 - ◆ Optical links; optical cross connect (MEMs, tunable lasers)
- **Time to market, Power/Heat**

Next time...

- I will be in Princeton NJ (not my choice)
- Geoff Voelker will give a lecture on distributed web caching (don't skip this, there may be a related question on the final)
 - ♦ I will arrange to have his slides put on the Web
- I will see you folks at the review session or the final
- Good luck, its been a pleasure.