# Lecture 13: Router Implementation

CSE 123: Computer Networks

Stefan Savage

# Last week

- Routing
  - Intra-domain
    - Distance vector
    - Link state
  - Inter-domain
    - BGP (path vector)
  - Multicast
    - One-to-many communication
    - Source-based tree routing
    - Shared tree routing
    - Tunneling

# Today

- Router implementation
  - Router basics
  - Interconnection architectures
    - » Input Queuing
    - » Output Queuing
    - » Virtual Output Queuing
  - Future bottlenecks

- Aside: Network Address Translation

# What's in a 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 – Quality of Service)
  - **Delay** packets (QoS)
  - **Transform** packets? (Encapsulation, Tunneling)

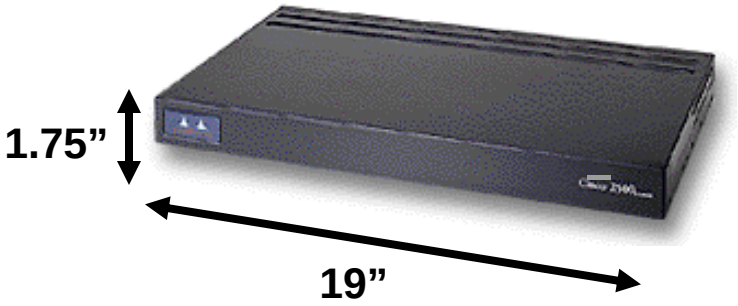
- Today we'll focus on the basic case:
  - FIFO scheduling
  - If queue full, packets dropped from tail

# What an IP router does: The normal case

1. Receive incoming packet from link input interface
2. Lookup packet destination in forwarding table
   (destination, output port(s))
3. Validate checksum, decrement ttl, update checksum
4. Buffer packet in input queue
5. Send packet to output interface (interfaces?)
6. Buffer packet in output queue
7. Send packet to output interface link

# What does a router look like?

**Cisco 2500**




**Capacity: <10Mbps**

**Linksys DEFSR81**


**Capacity: <10Mbps**

# What does a router look like (2)?

## Cisco GSR 12416


Capacity: 160Gb/s
Power: 4.2kW

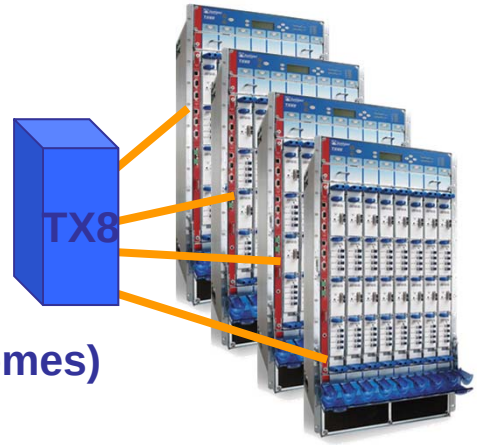
## Juniper M160


Capacity: 80Gb/s
Power: 2.6kW

# What does a router look like (3)?

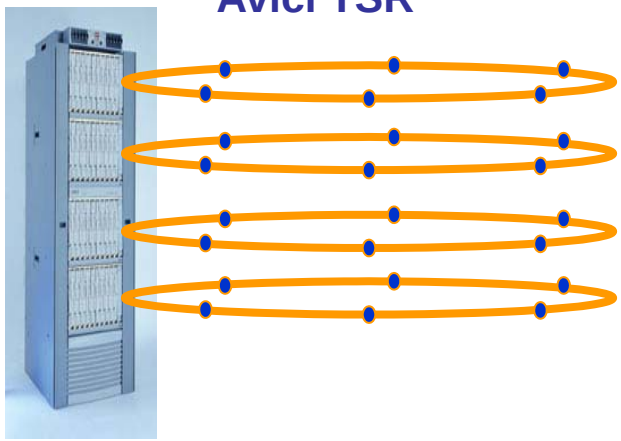
**Alcatel 7670 RSP**

**Juniper TX8/T640**



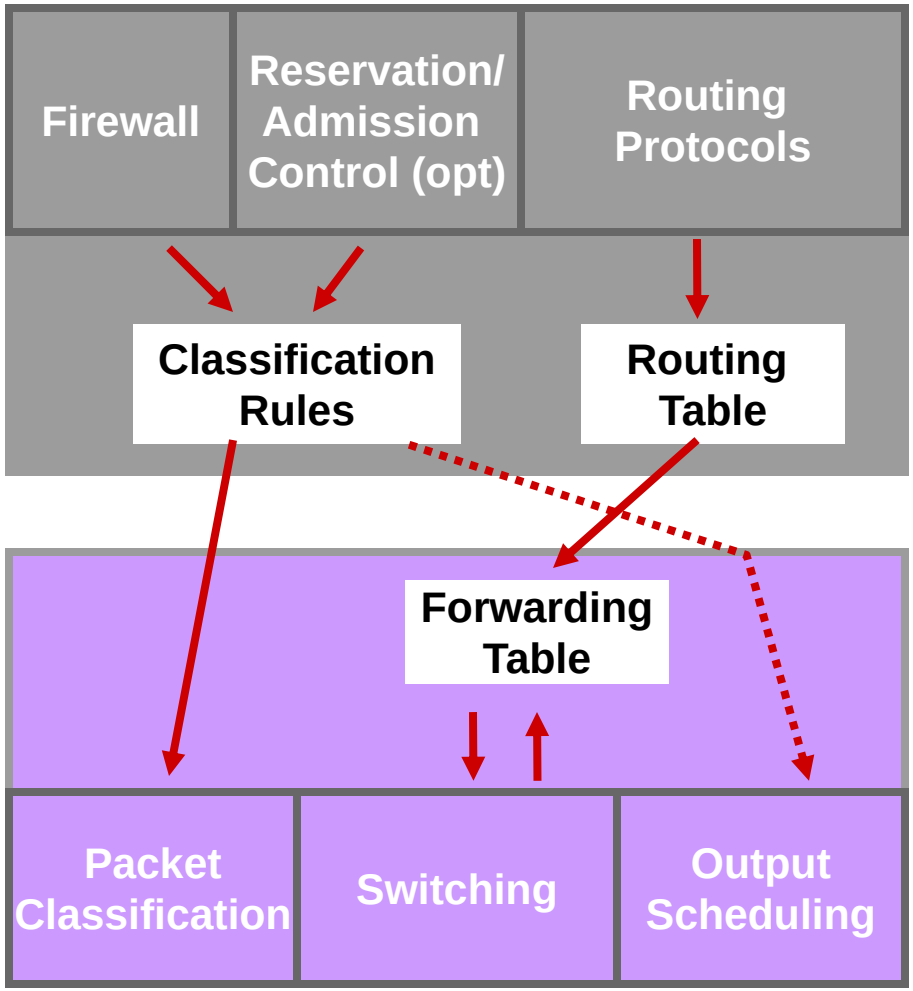

**Capacity: nTb/s**
**Power: 10s of kW (~100's of homes)**

**Avici TSR**



**Chiaro**

# Functional architecture




**Control Plane**

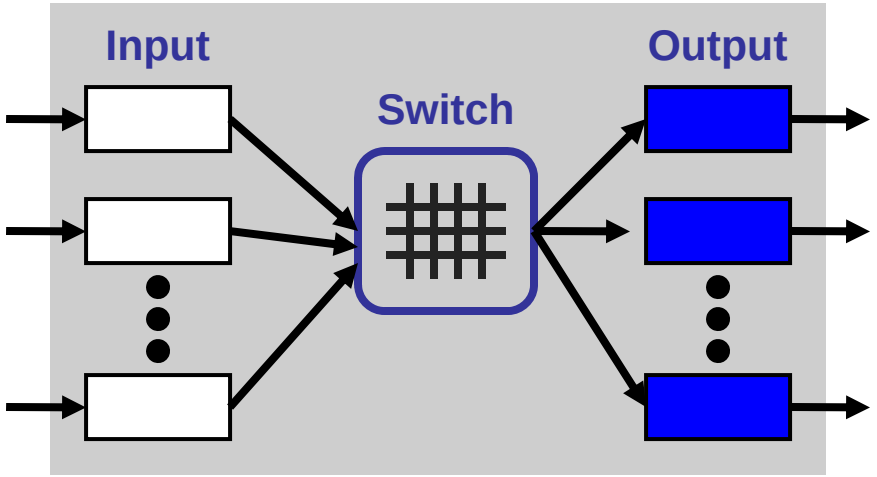
- Complex
- Per-control action
- May be slow

**Data plane**

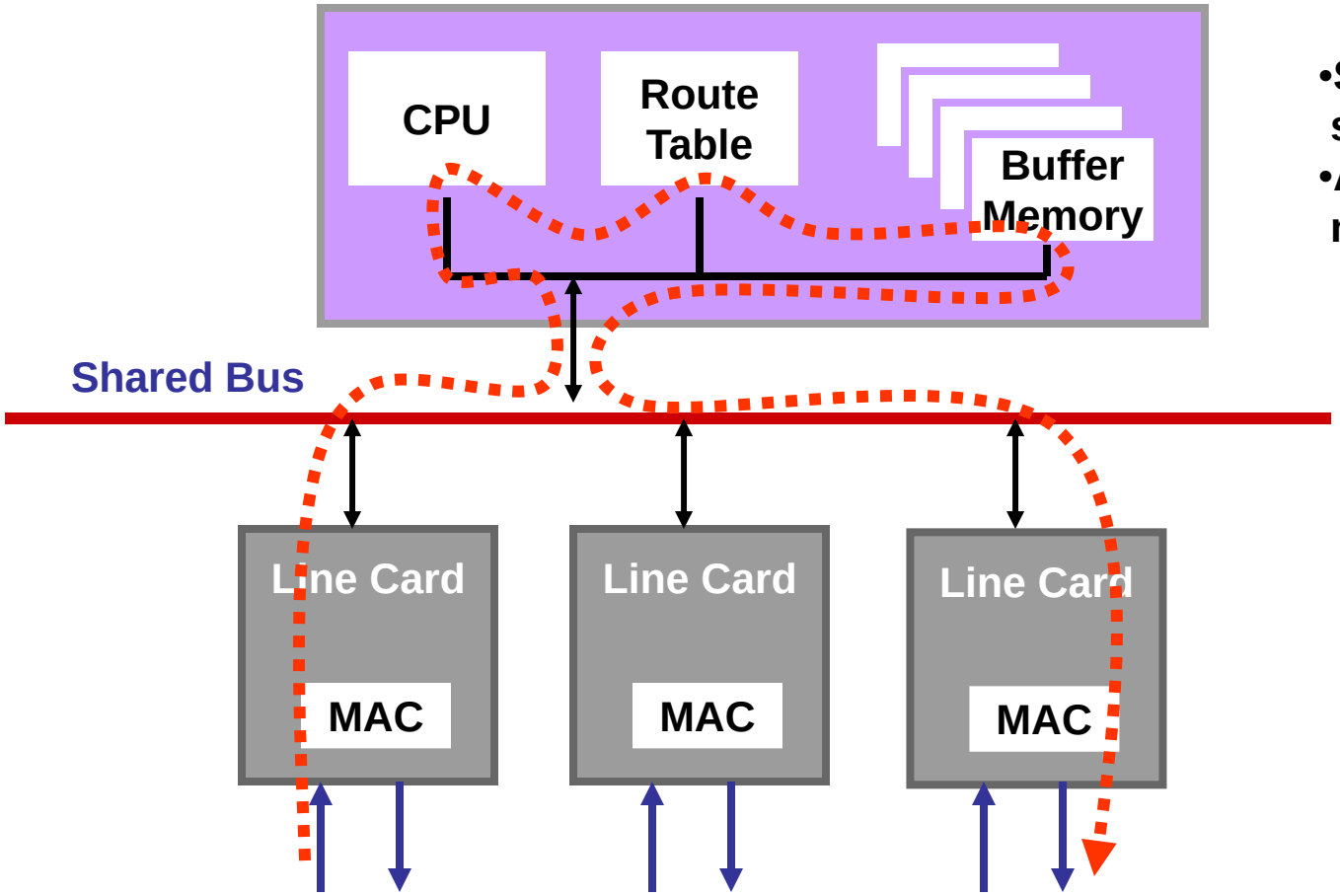
- Simple
- Per-packet
- Must be fast

# Interconnect (Router) architecture

- Input & output connected via switch fabric
  - Transfers packets from one port to another
- Kinds of switch fabric
  - Shared Memory – Low capacity
  - Bus – Medium capacity
  - Crossbar – High capacity
- How to deal with transient contention?
  - Input queuing
  - Output queuing

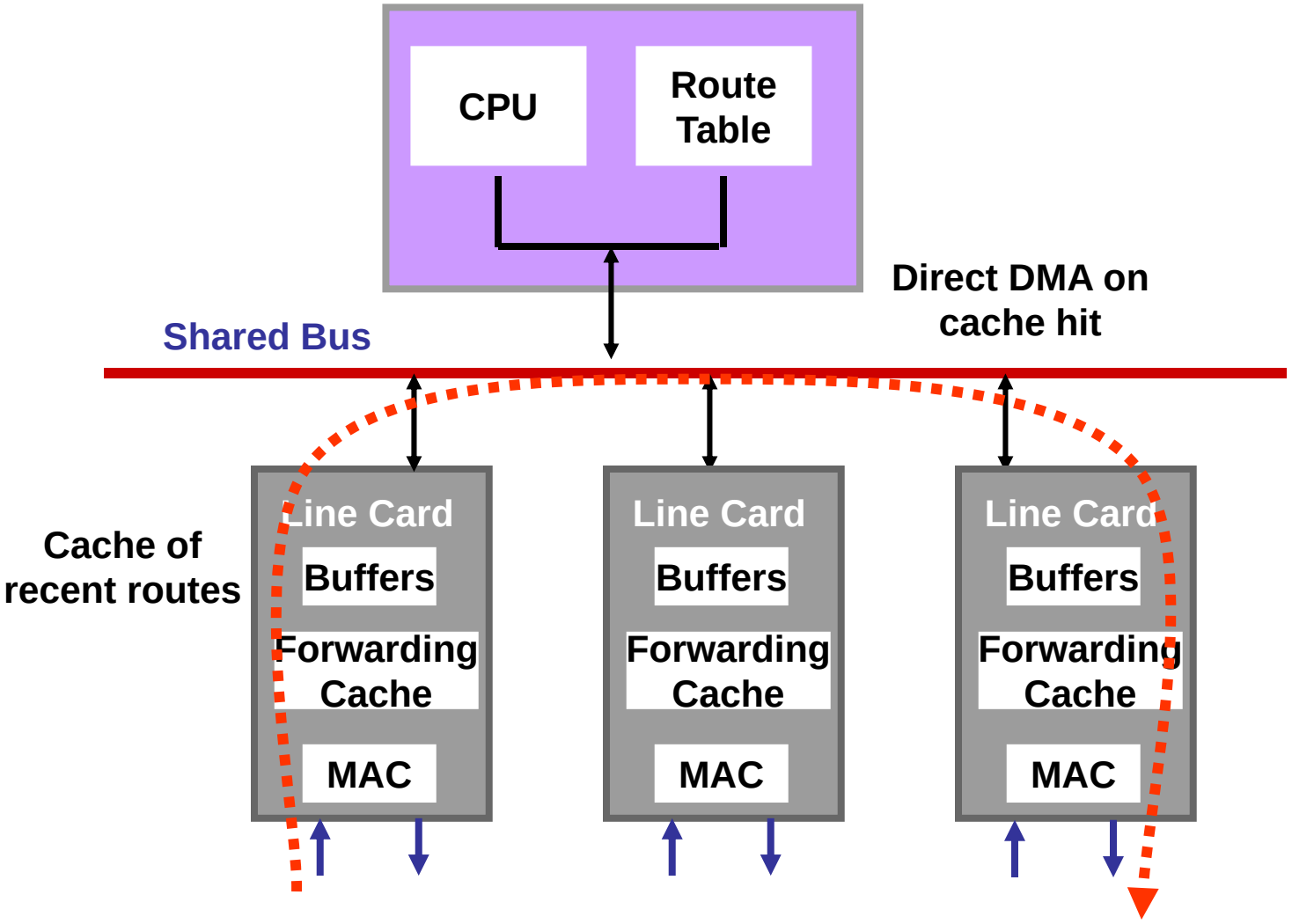


# First Generation Routers

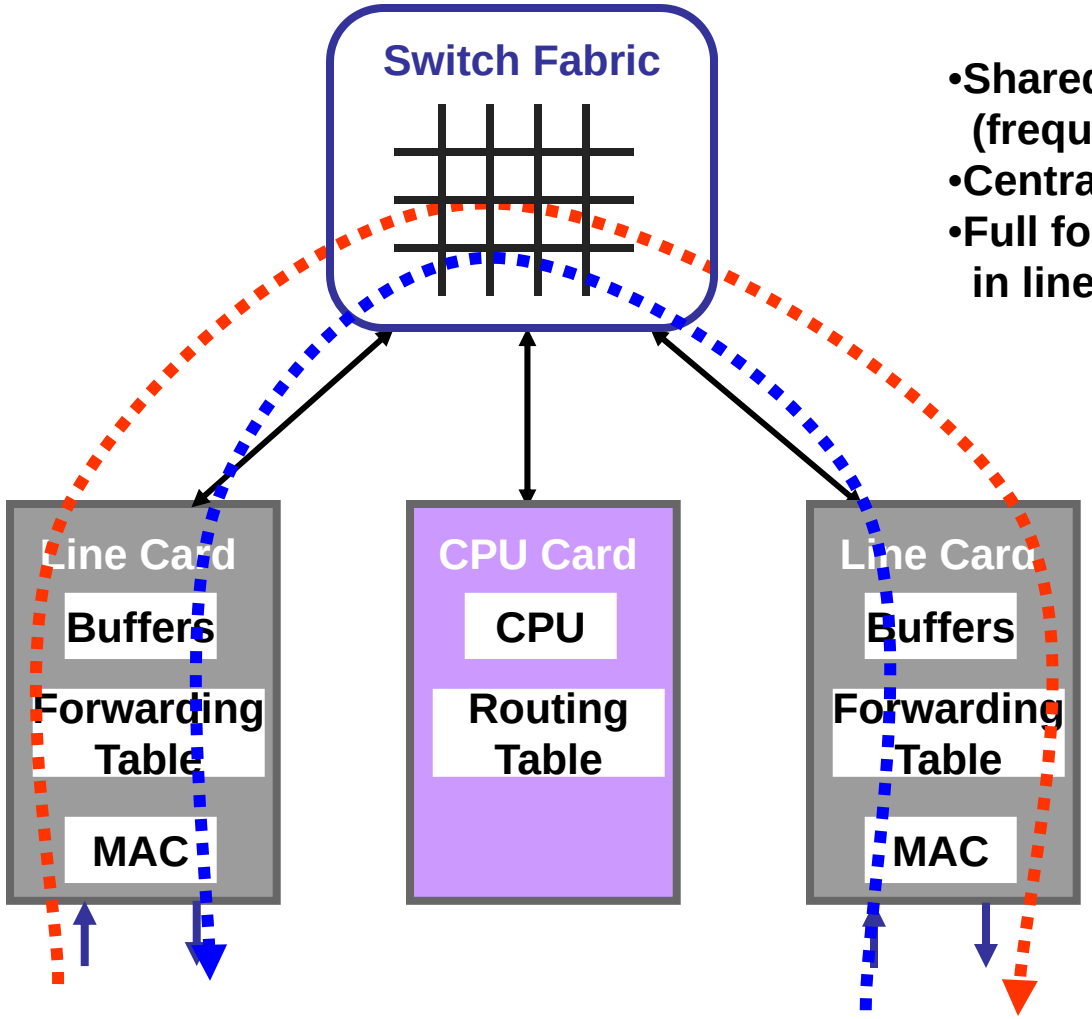



- Single CPU and shared memory;
- All classification by main CPU

# Second Generation Routers



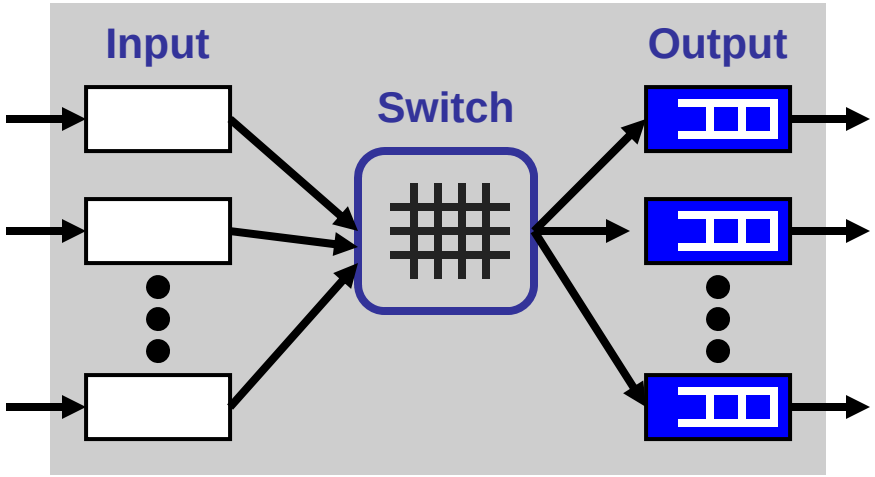
# Third Generation Routers




- Shared interconnect (frequently crossbar)
- Centralized scheduler
- Full forwarding table in line card

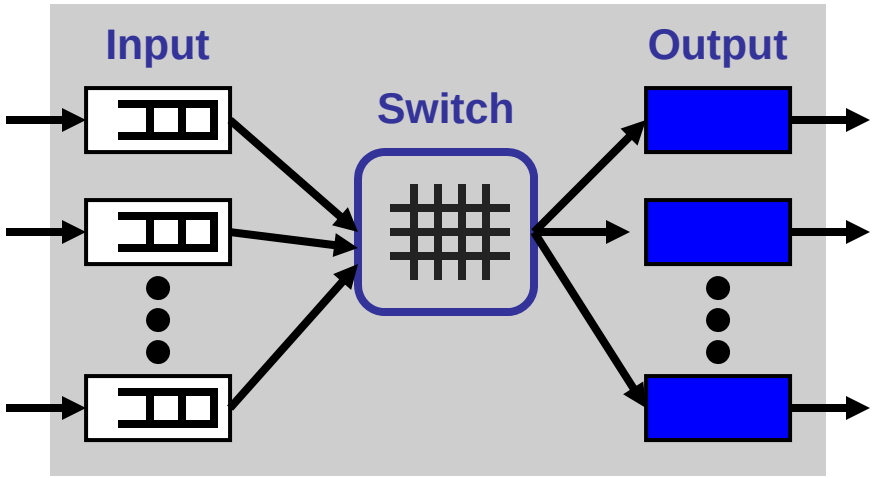
# Output queuing

- Output interfaces buffer packets
- Pro
  - Simple algorithms
  - Single congestion point
- Con
  - N inputs may send to the same output
  - Requires *speedup* of N
    - » *speedup* is ratio of output drain rate to input fill rate
    - » i.e., so output ports must run N times quicker than input ports to handle worst case

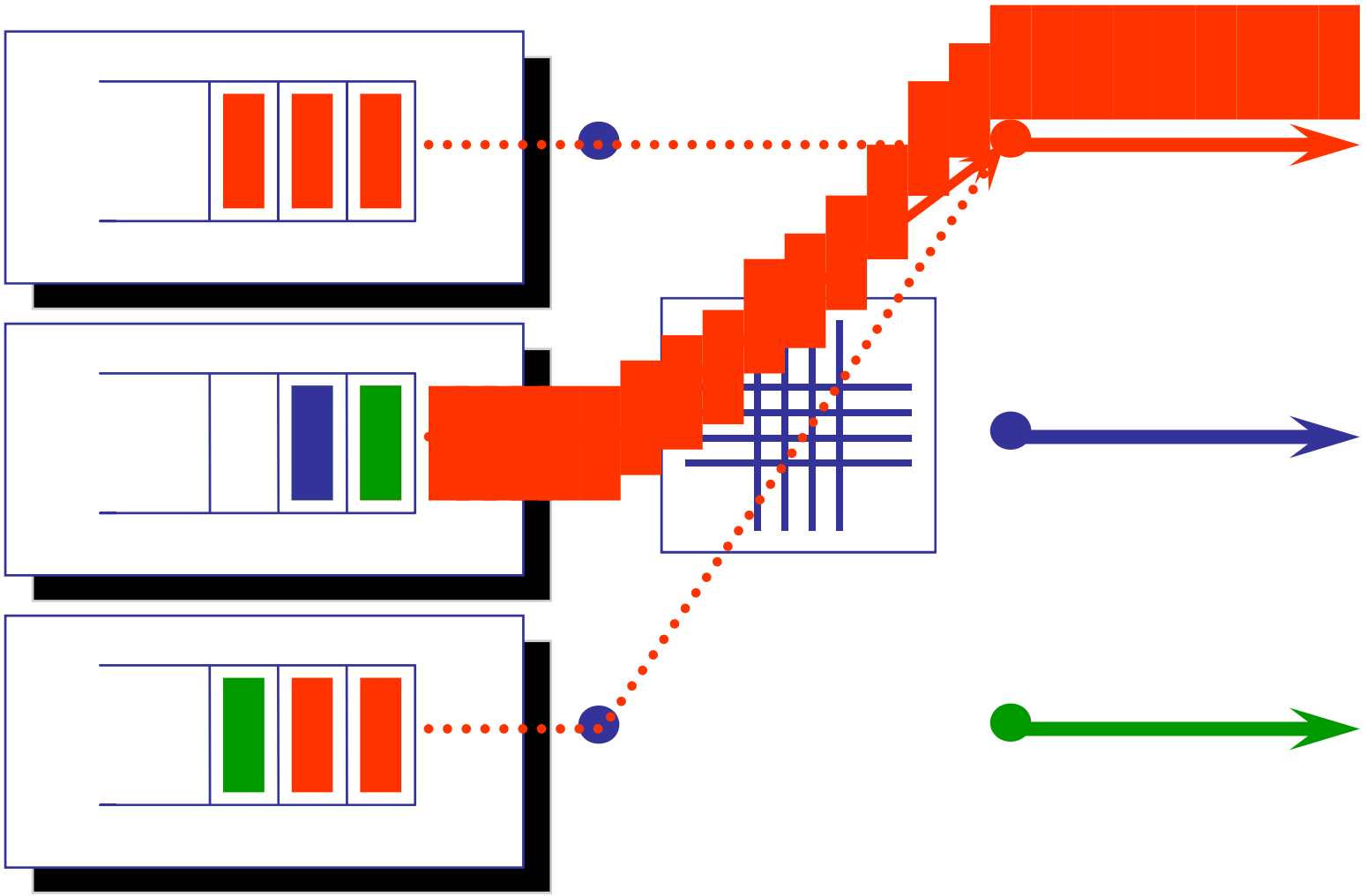


# Input queuing

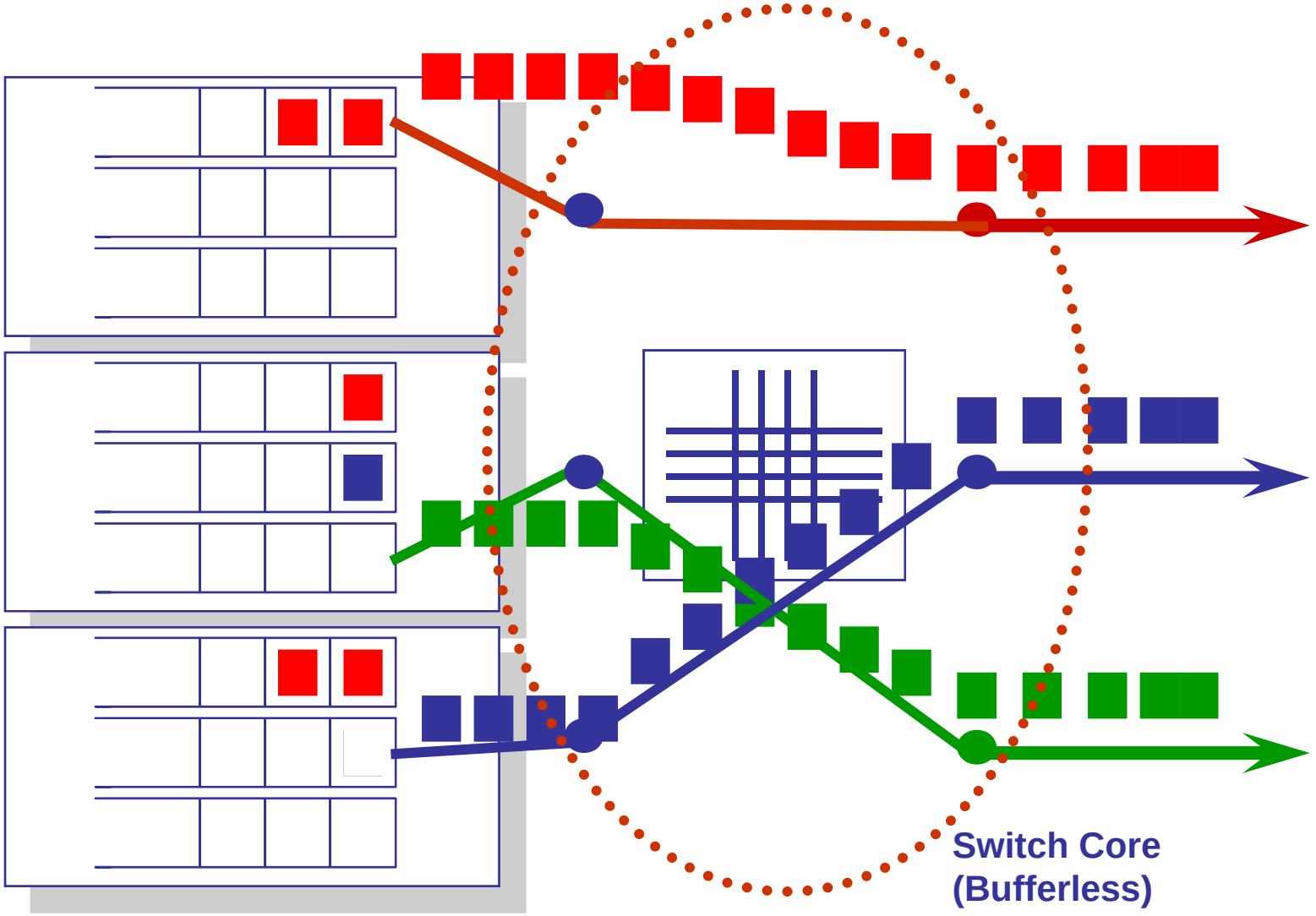
- Input interfaces buffer packets
- Pro
  - Single congestion point
  - Simple to design algorithms
- Con
  - Must implement flow control
  - Low utilization due to Head-of-Line (HoL) Blocking
    - » Utilization limited to 58%



# Head-of-Line Blocking

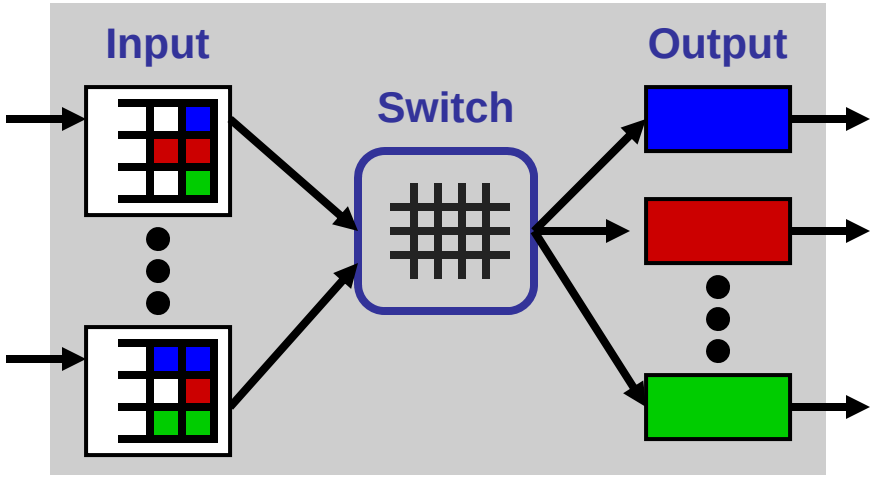


# Virtual Output Queues



# IQ + Virtual Output Queuing

- Input interfaces buffer packets in per-output virtual queues
- Pro
  - Solves blocking problem
- Con
  - More resources per port
  - Complex arbiter at switch
  - Still limited by input/output contention (scheduler)

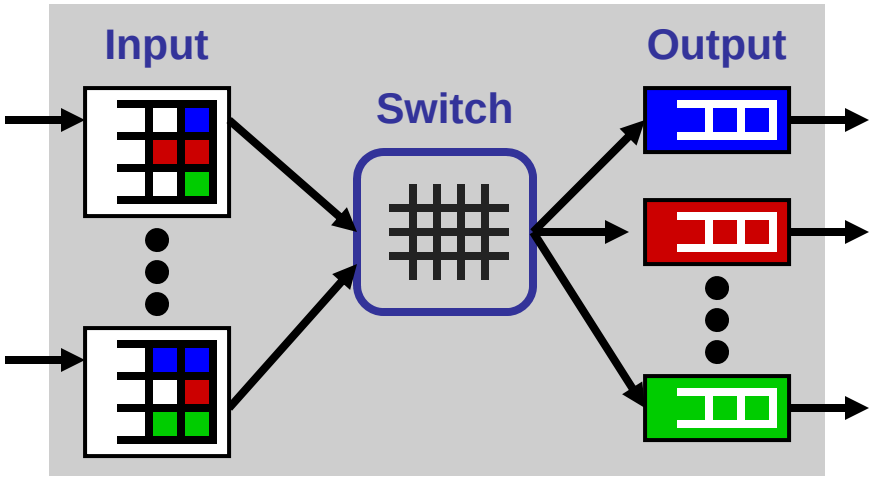


# Switch scheduling

- Problem
  - Match inputs and outputs
    (remember *matching problems* from algorithms?)
  - Resolve contentions, no packet drops
  - Maximize throughput
  - Do it in constant time…
- If traffic is uniformly distributed its easy
  - Lots of algorithms (approximate matching)
- Major result (Dai et al, 2000)
  - Maximal size matching + *speedup* of two guarantees
    100% utilization for most traffic assumptions

# Modern high-performance router

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



# Next bottlenecks

- Buffering at high speed
  - SRAM is fast enough, but density is too low for BW*D of 40Gbps or 100Gbps link (also expensive)
  - DRAM is too slow
  - Possible solution: SRAM memory management used as cache for DRAM
- Scheduler overhead
  - Hard to do central scheduler much over 1Tbps
  - Multi-stage load-balanced switches
- High density (100's-1000's of line cards)
  - Physical distance to support density; electrical links degrade
  - Optical links; optical cross connect

# For next time…

- Transport protocols & congestion control
- Read P&D 6.3