

Lecture 20:

Router Design

CSE 123: Computer Networks

Aaron Schulman





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)
 - ◆ **Delay** packets (QoS)
 - ◆ **Transform** packets? (Encapsulation, Tunneling)

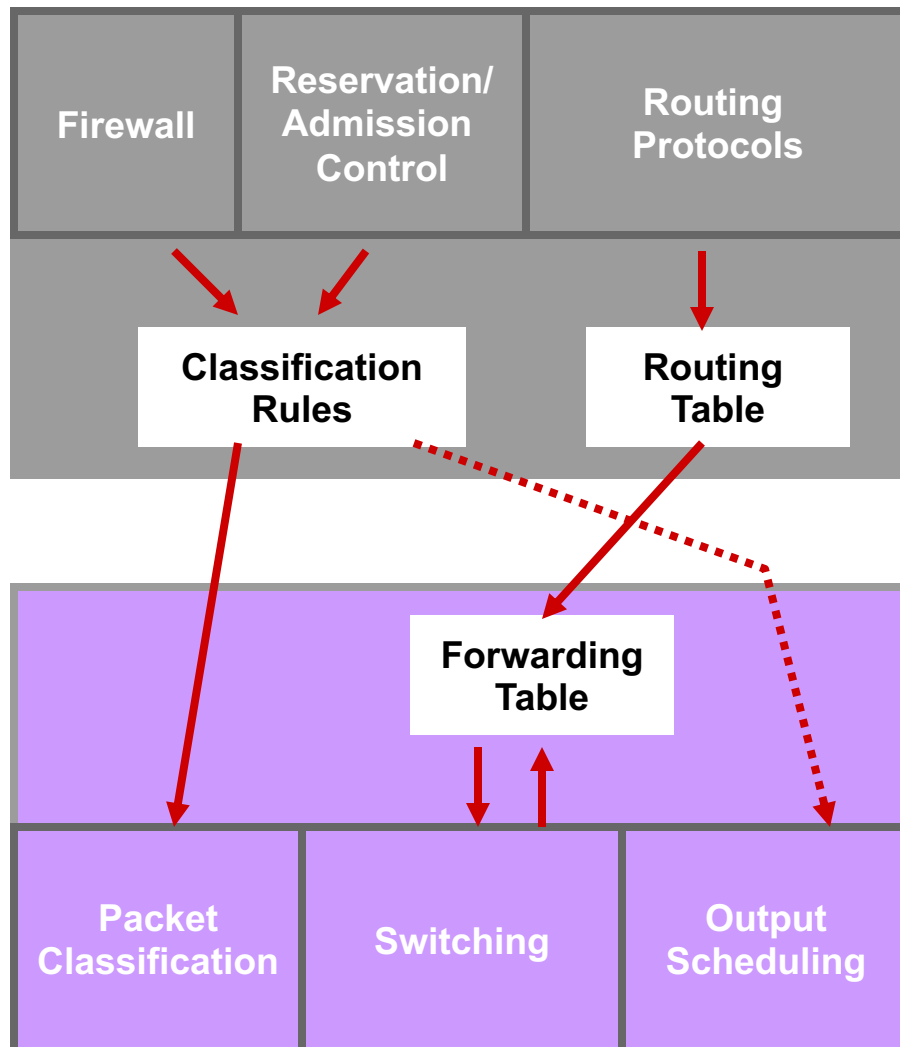


Router Functions

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



Functional architecture



Control Plane

- Complex
- Per-control action
- May be slow

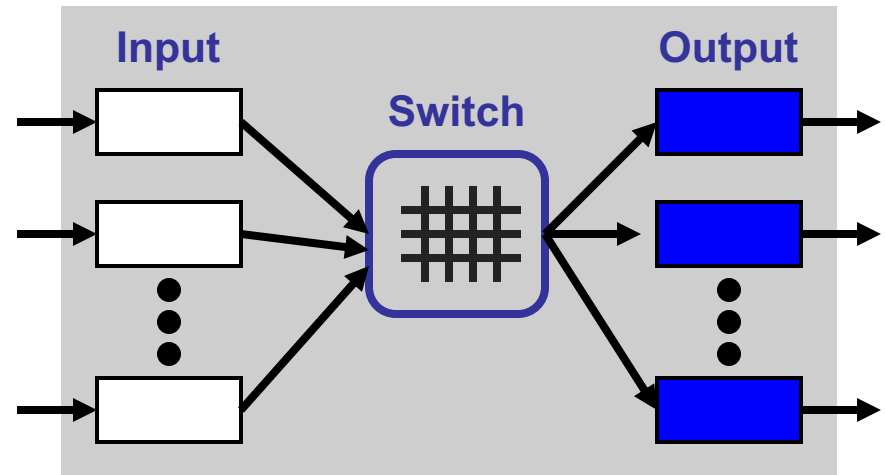
Data plane

- Simple
- Per-packet
- Must be fast

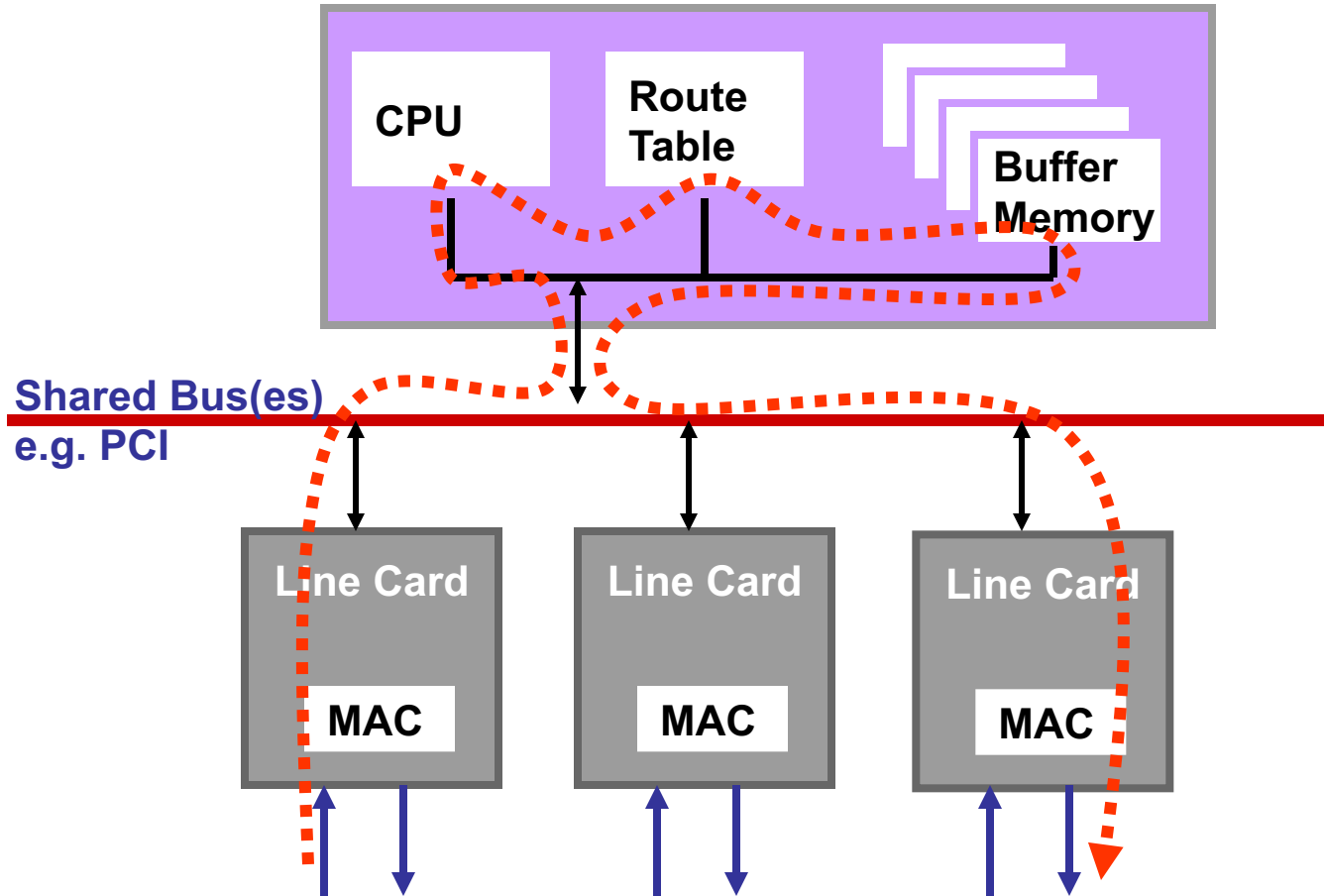


Interconnect architecture

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



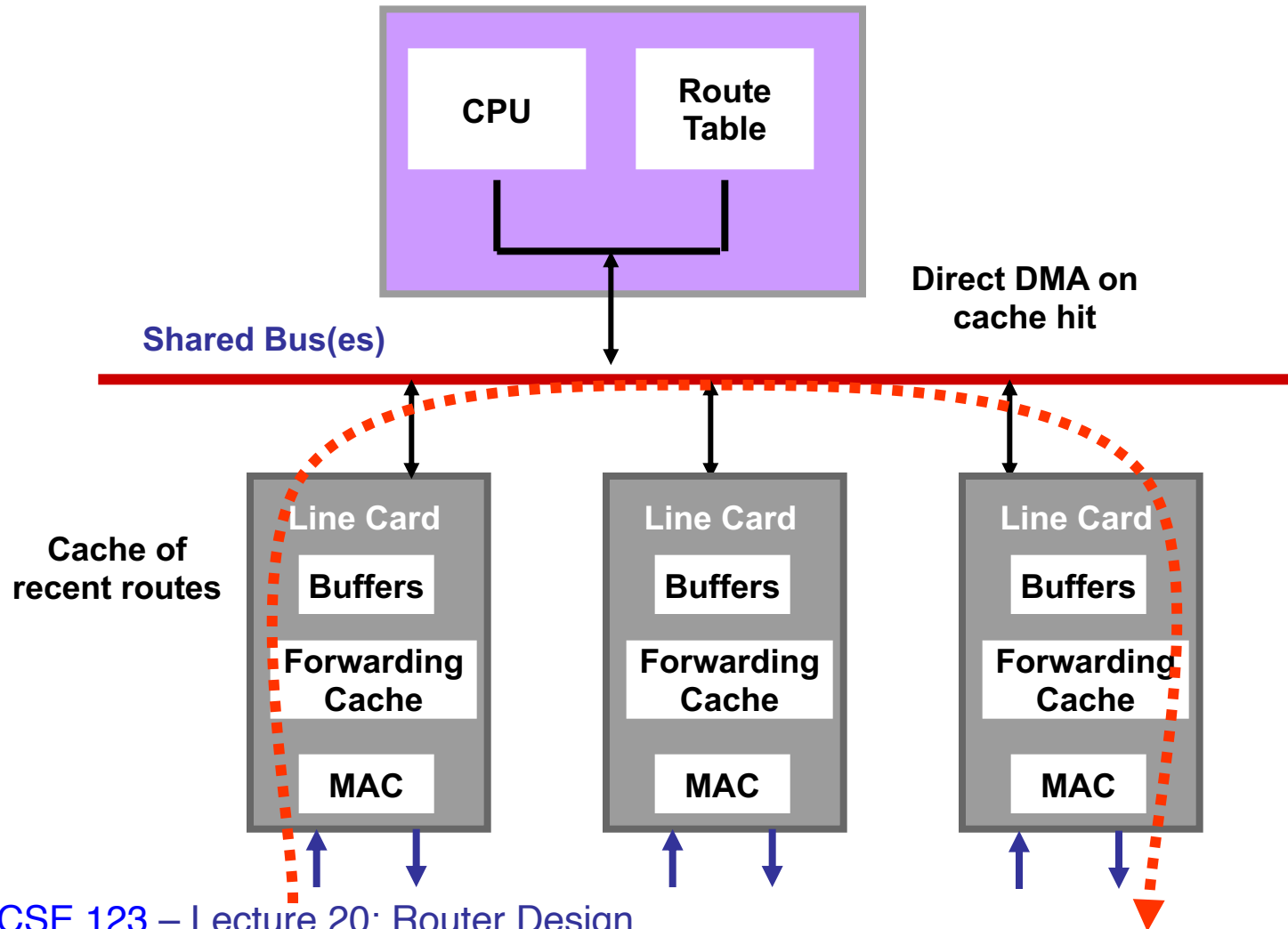
First Generation Routers (mostly software)



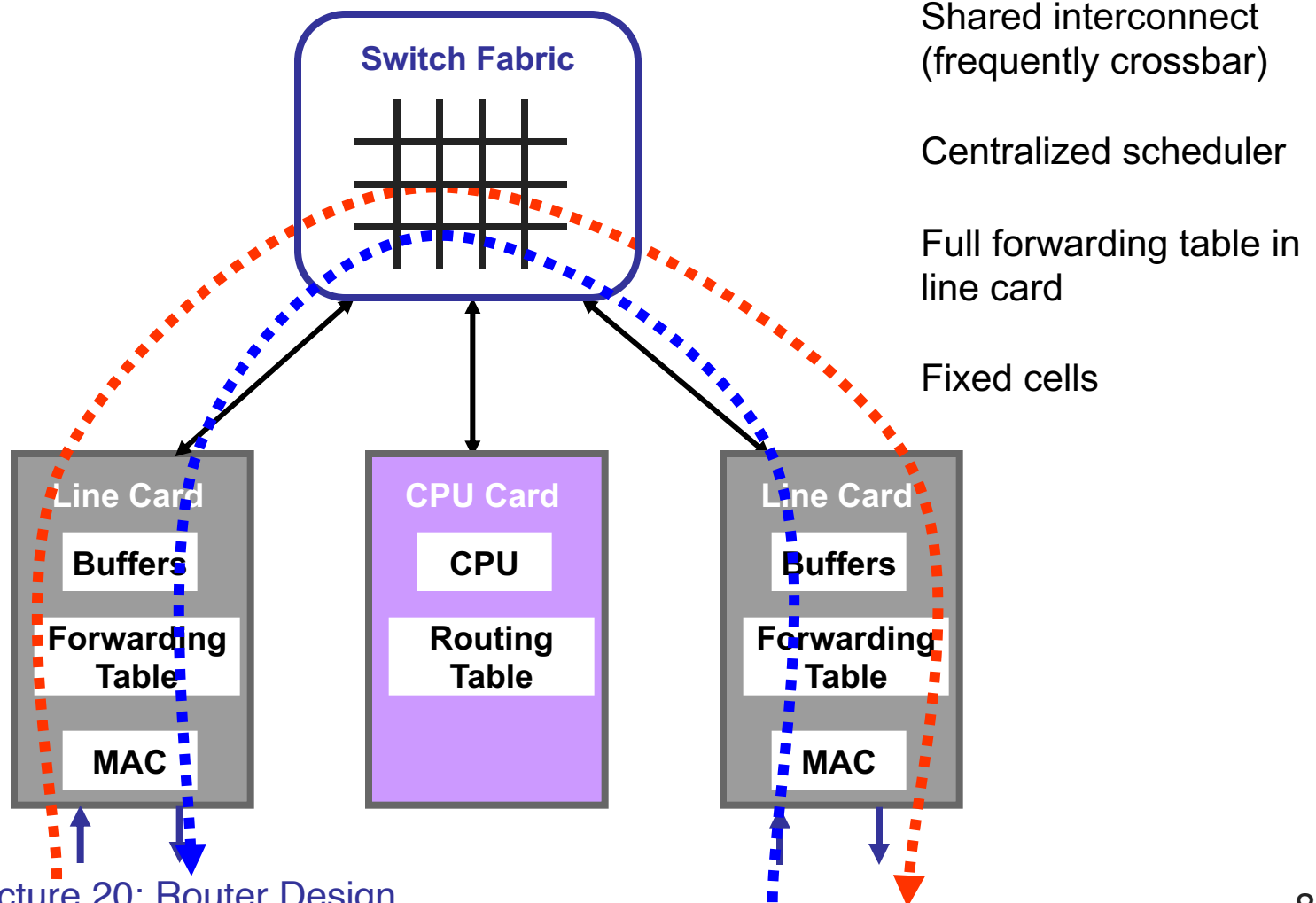
Single CPU and
shared memory;

All classification by
main CPU

Second Generation Routers (HW caching forwarding table)



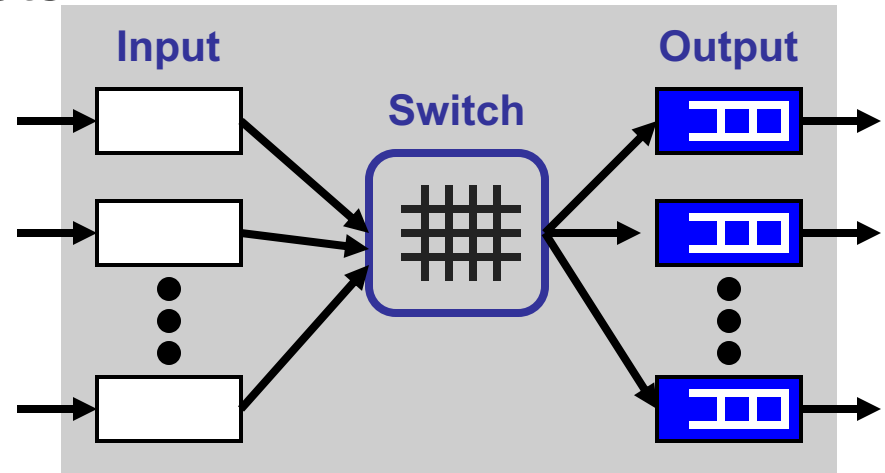
Third Generation Routers (improved HW shared bus)





Output queuing

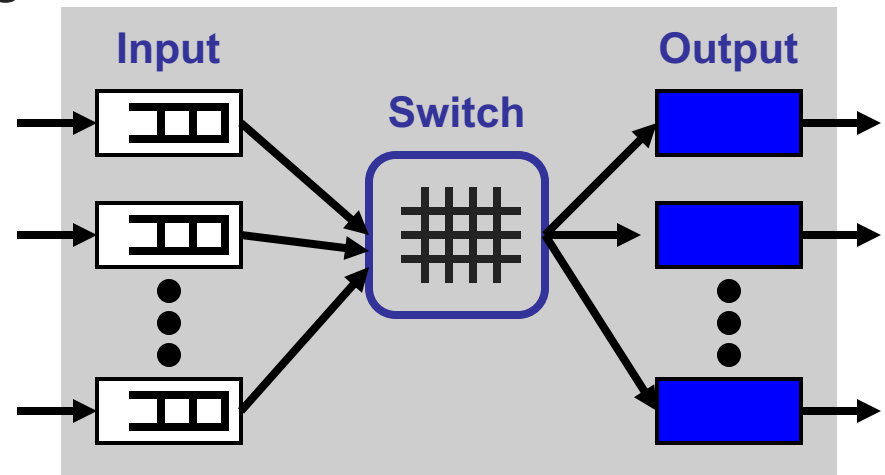
- Output interfaces buffer packets
- Pro
 - ◆ Simple algorithms
 - ◆ Single congestion point
- Con
 - ◆ N inputs may send to the same output
 - ◆ Requires **speedup** of N
 - » Output ports must be N times faster than input ports





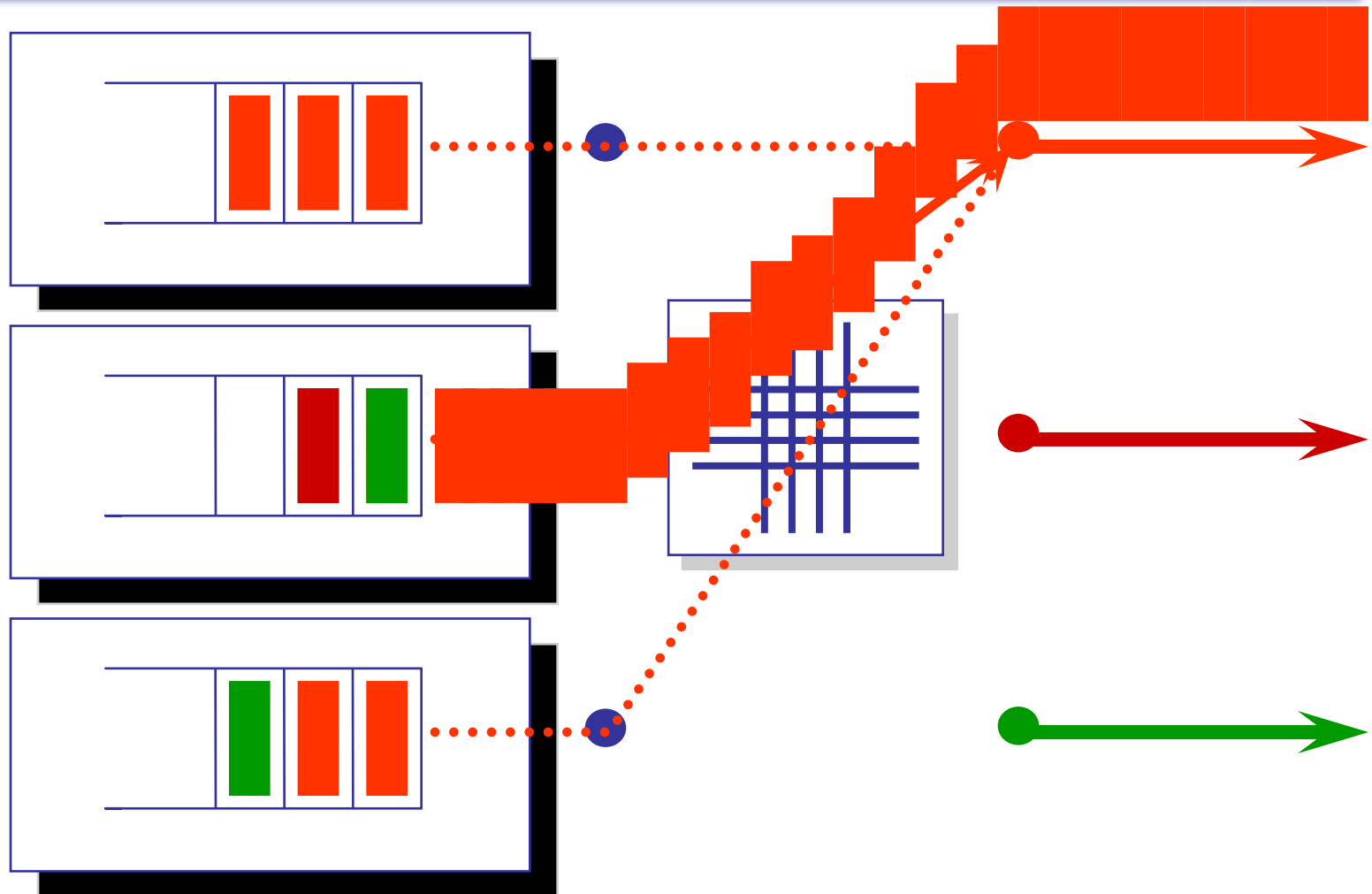
Input queuing

- 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





Head-of-Line Blocking



IQ + Virtual Output Queuing



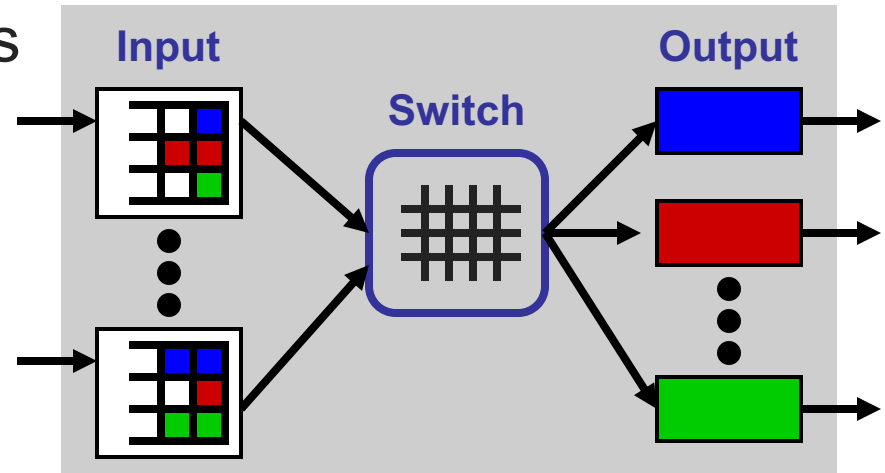
- Input interfaces buffer packets in per-output virtual queues

- Pro

- ◆ Solves blocking problem

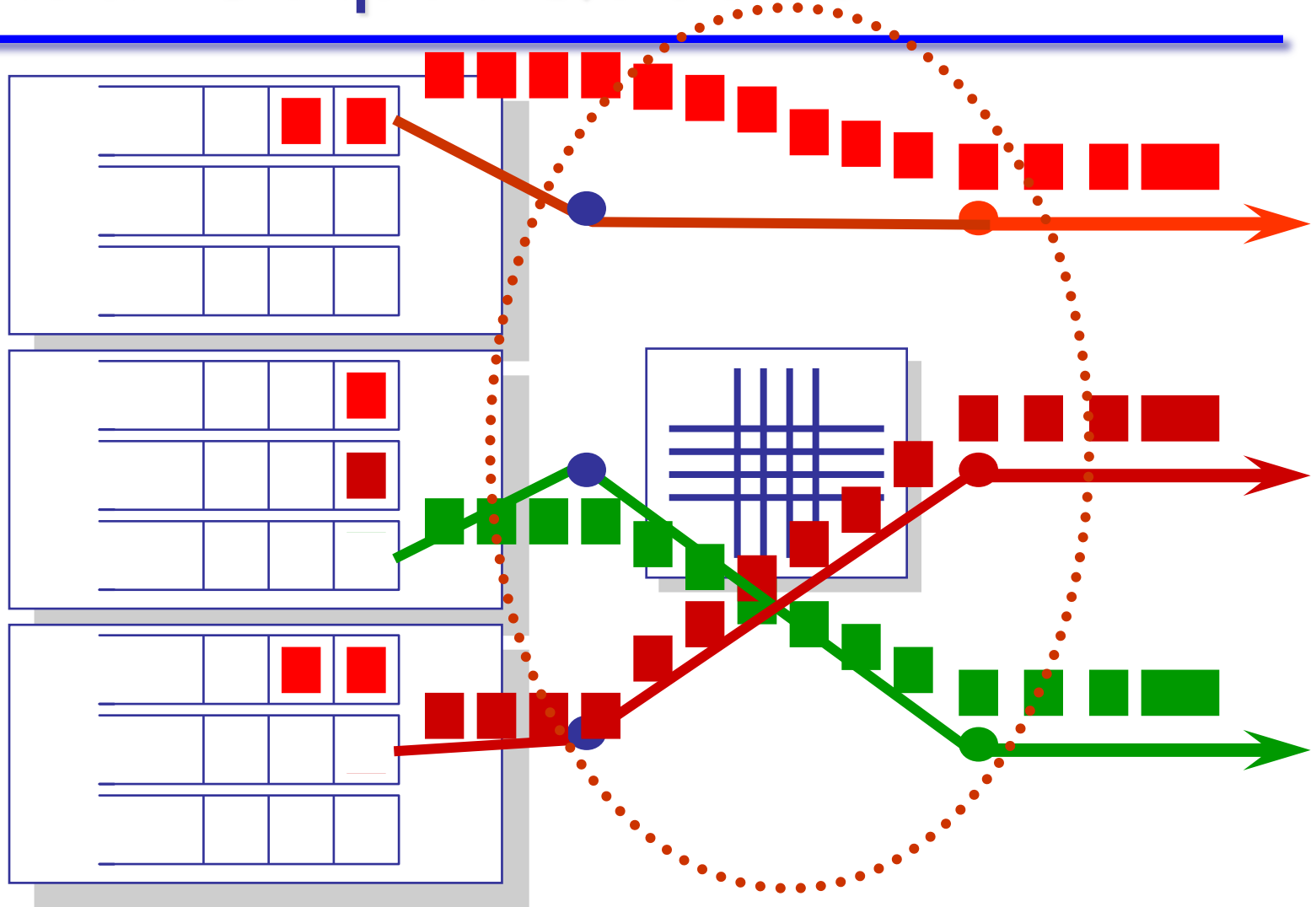
- Con

- ◆ More resources per port
- ◆ Complex arbiter at switch
- ◆ Still limited by input/output contention (scheduler)





Virtual Output Queues





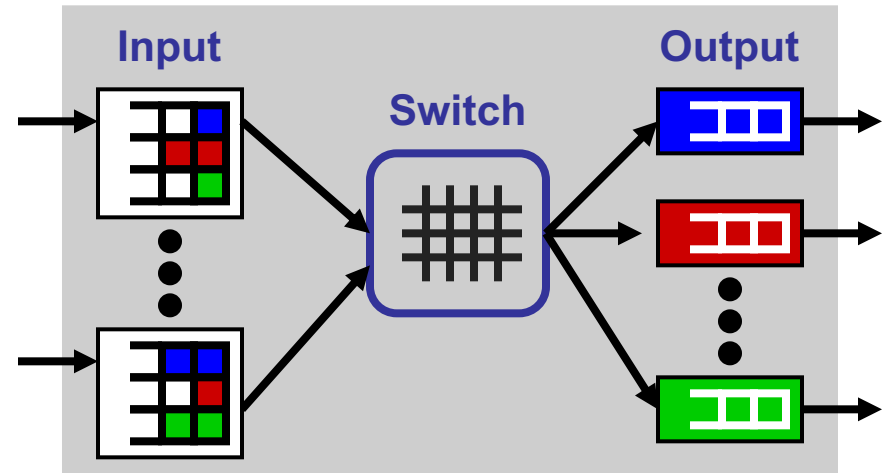
Switch scheduling

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



Typical high-performance router

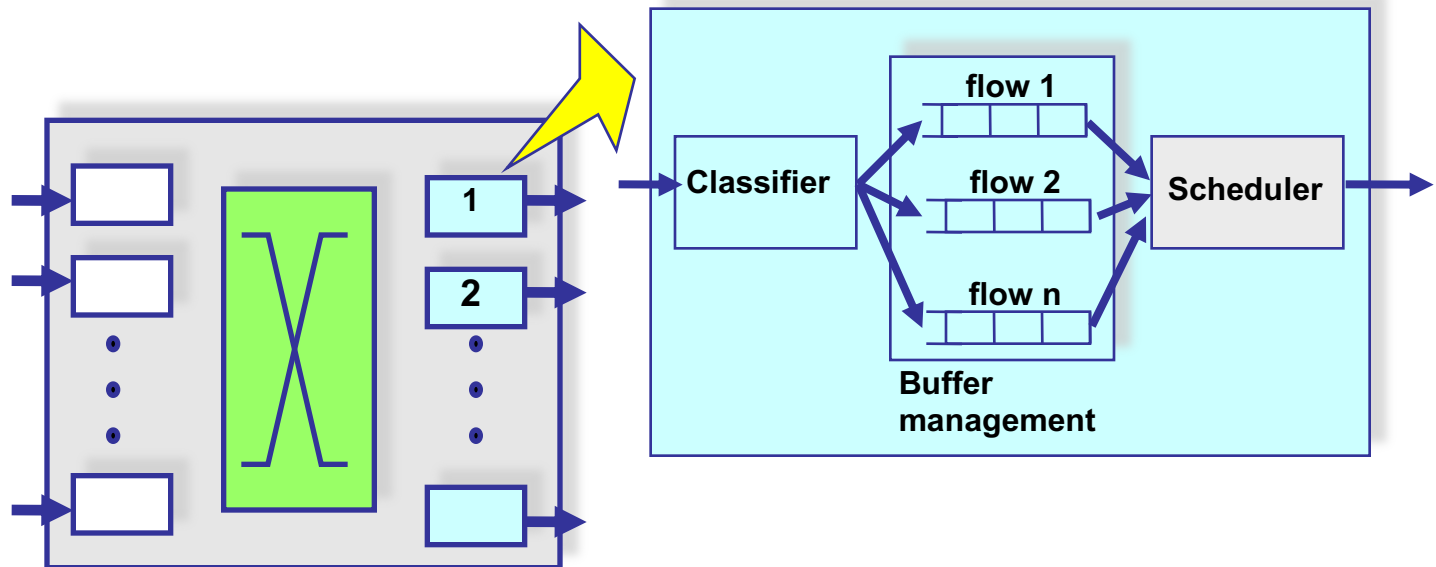
- IQ + VoQ + OQ
 - ◆ Central scheduler
 - ◆ Fixed-sized internal pkt cells
- Pro
 - ◆ Can achieve utilization of 1
 - ◆ Can scale to $> \text{Tb/s}$
- Con
 - ◆ Multiple congestion points
 - ◆ Complexity





Key Router Challenges

- **Buffer management**: which packet to drop when?
 - ◆ We only have finite-length queues
- **Scheduling**: which packet to transmit next?





Basic Buffer Management

- FIFO + drop-tail
 - ◆ Simplest choice
 - ◆ Used widely in the Internet
- FIFO (first-in-first-out)
 - ◆ Implies single class of traffic
- Drop-tail
 - ◆ Arriving packets get dropped when queue is full regardless of flow or importance
- Important distinction:
 - ◆ FIFO: scheduling discipline
 - ◆ Drop-tail: drop policy



FIFO/Drop-Tail Problems

- Leaves responsibility of preventing congestion completely to the edges
 - ◆ Transport protocols will need to make sure queues in routers are not so full that they drop, (called congestion control)
- Does not separate between different flows
 - ◆ Packet can be dropped regardless of what TCP or UDP flow it is part of
- **No policing:** send more packets → get more service
- Synchronization: end hosts react to same problems



For Next Time

- Read P&D 6.2
- Get started on Project 3!