

Lecture 20:

Router Design

CSE 123: Computer Networks

Aaron Schulman





Lecture 20

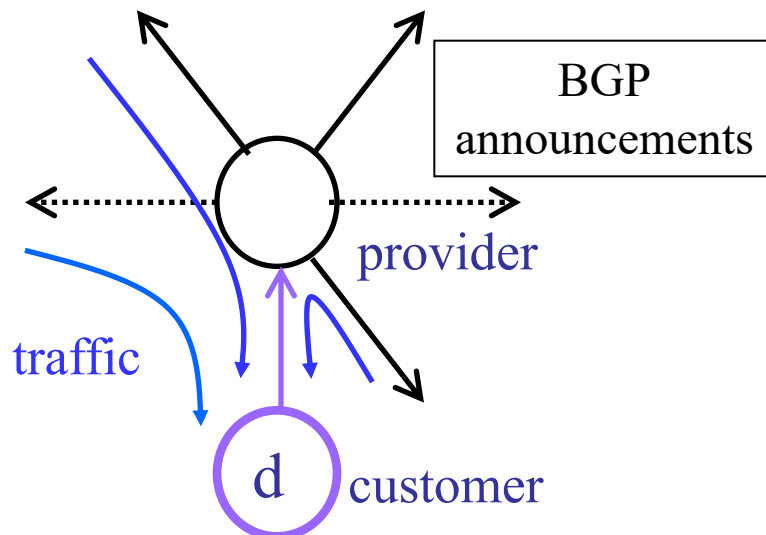
- Wrap up Interdomain routing
- Router implementation
 - ◆ Switching fabrics
 - ◆ Queueing



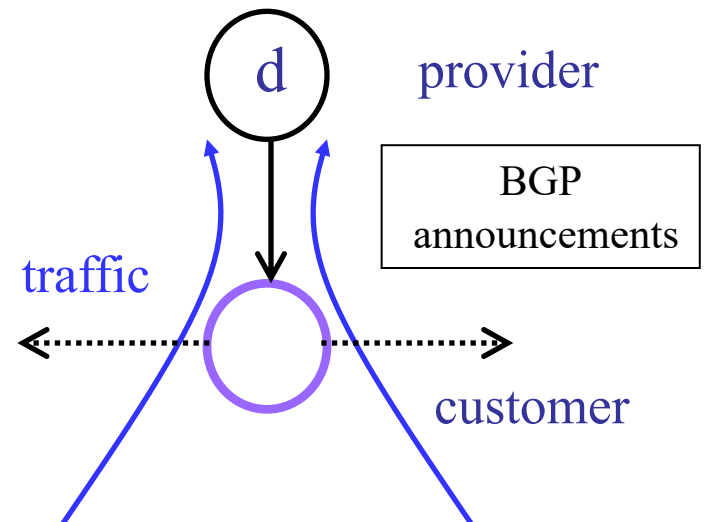
Customer/Provider

- Customer needs to be reachable from everyone
 - ◆ Provider tells all neighbors how to reach the customer
- Customer does not want to provide transit service
 - ◆ Customer does not let its providers route through it to other providers

Traffic to the customer



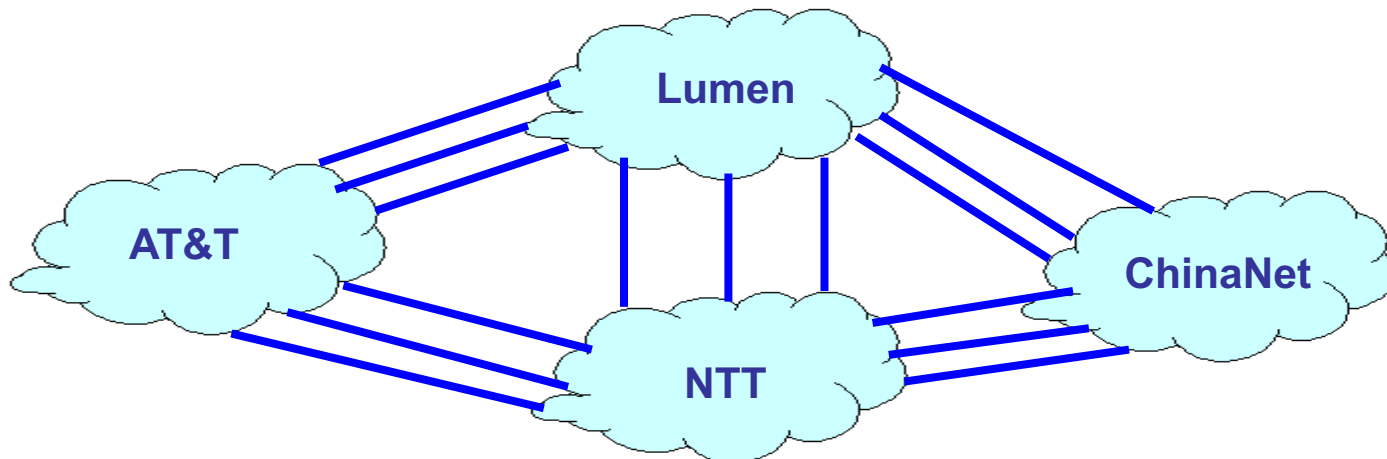
Traffic from the customer





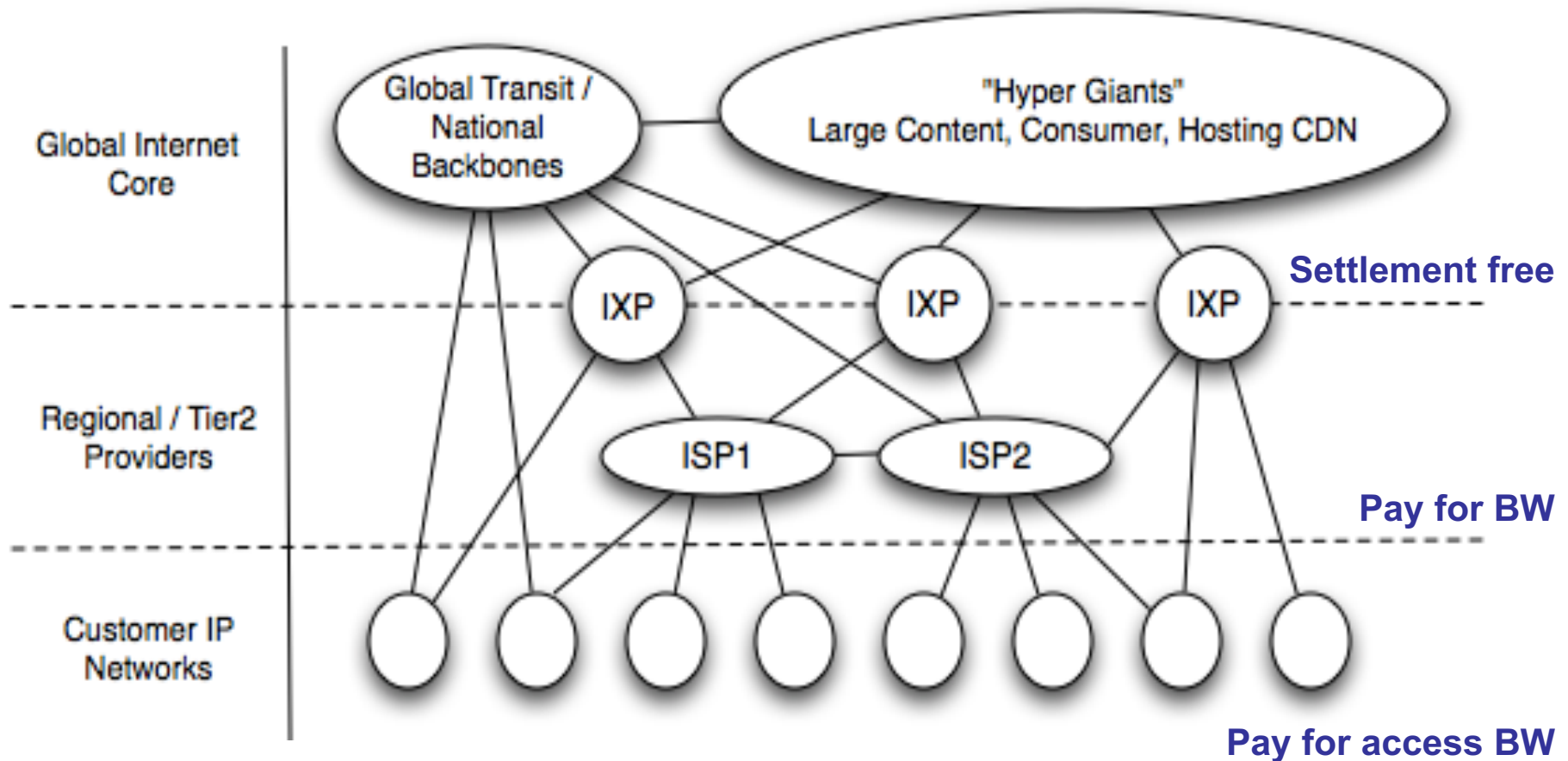
Tier-1 Providers

- Make up the “core” of the Internet
 - ◆ Has no upstream provider of its own
 - ◆ Typically has a national or international backbone
- Top of the Internet hierarchy of ~10-20 ASes
 - ◆ E.g., AT&T, Lumen (CenturyLink), NTT/Verio, Verizon, ChinaNet
 - ◆ Full peer-peer connections between tier-1 providers





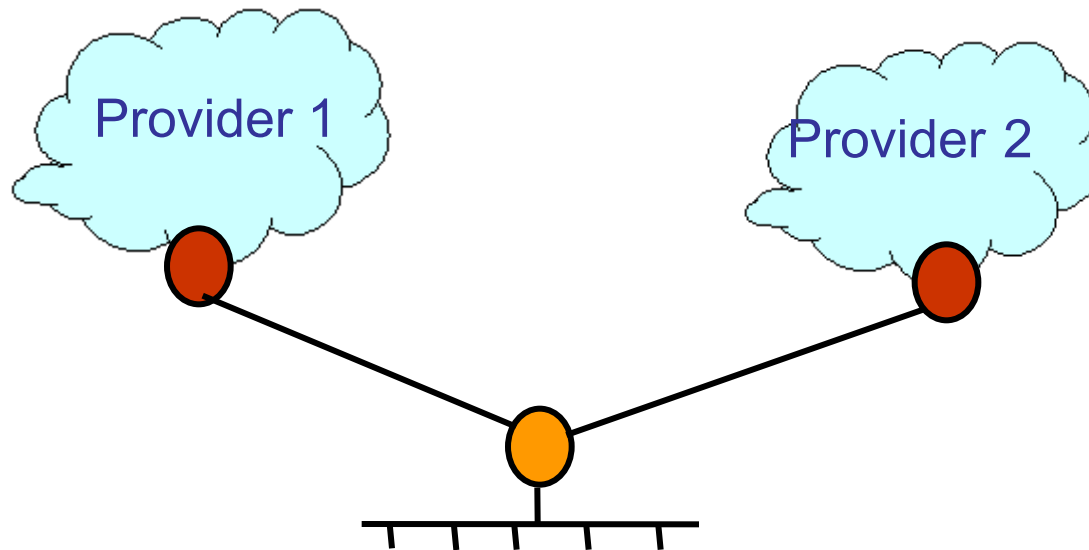
The Internet Hierarchy





Multi-Homing

- Customers may have more than one provider
 - ◆ Extra reliability, survive single ISP failure
 - ◆ Financial leverage through competition
 - ◆ Better performance by selecting better path
 - ◆ Gaming the 95th-percentile billing model





BGP Summary

- Interdomain-routing
 - ◆ Exchange reachability information (plus hints)
 - ◆ BGP is based on path vector routing
 - ◆ Local policy to decide which path to follow
- Traffic exchange policies are a big issue (\$\$\$)
 - ◆ Complicated by lack of compelling economic model (who creates value?)
 - ◆ Can have significant impact on performance



What's in a Router?

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

Router Implementation



**Cisco CSR Carrier-grade Router
(Tier-1)**



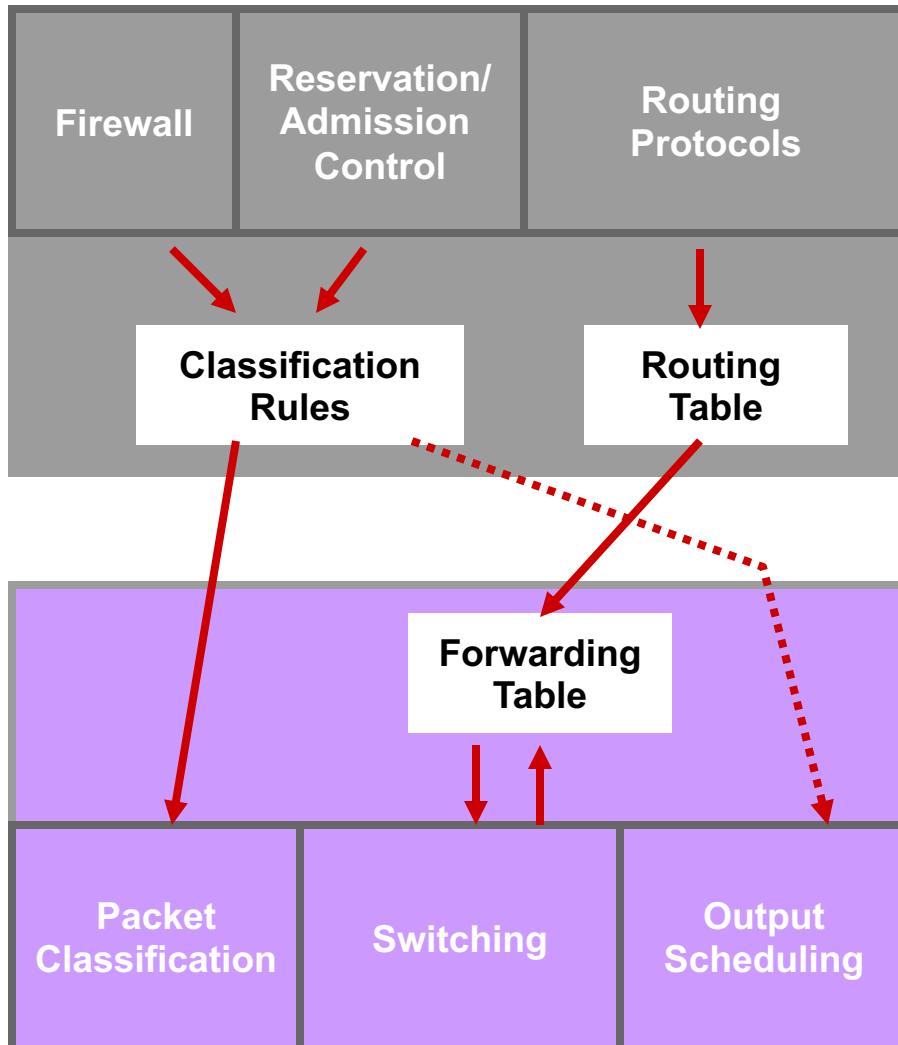
**Cisco ASR Aggregation Router
(Regional ISP)**



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

Router functional architecture



Control Plane

- General-purpose CPU
- Complex software
- Per-control action
- May be slow

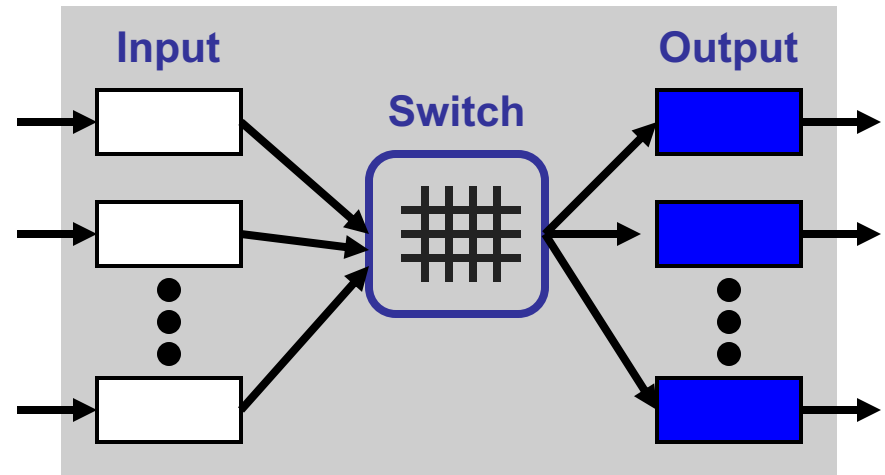
Data plane

- Simple
- Per-packet
- Must be incredibly fast (Tbps)

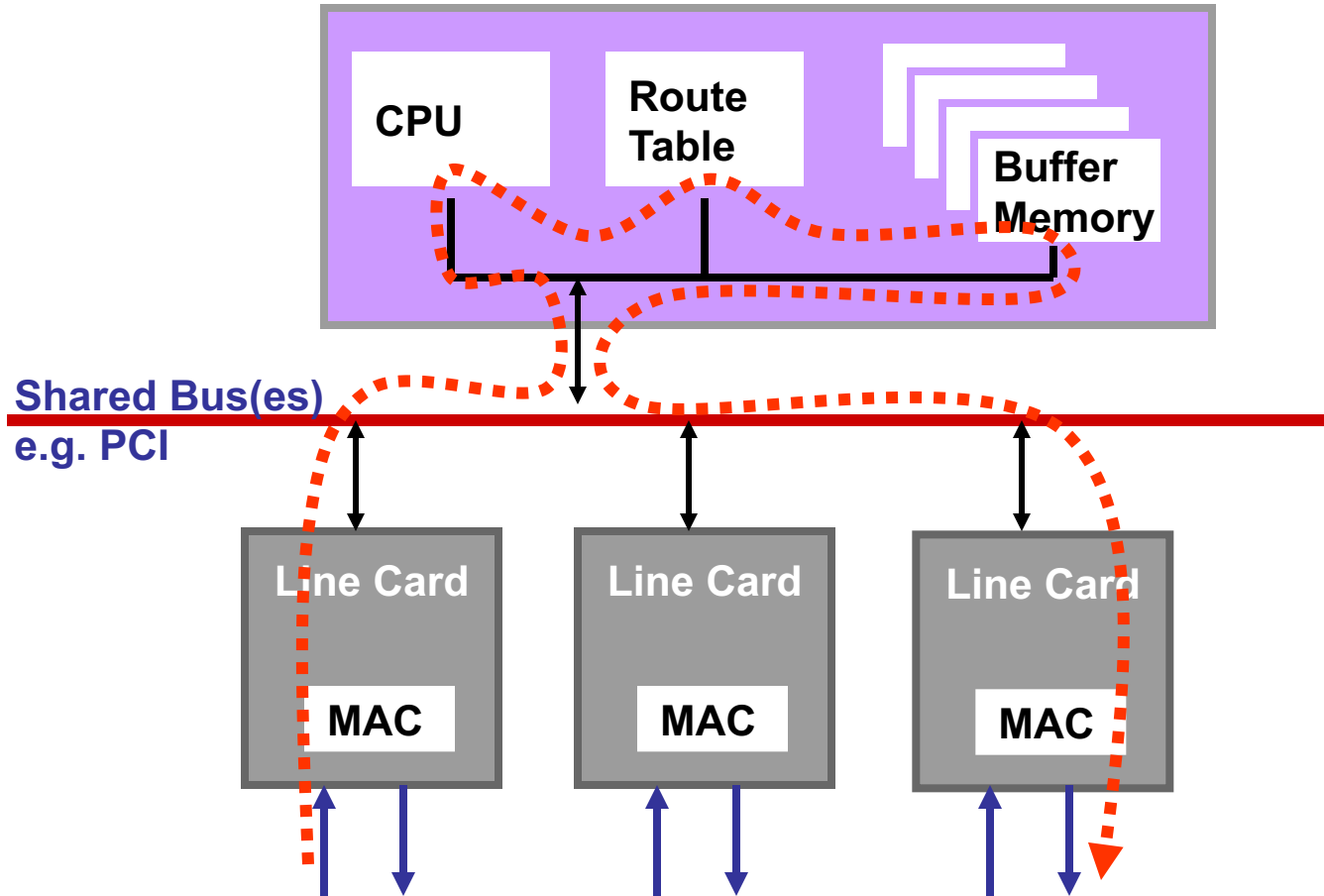


Interconnect architecture

- Input & output ports connected via switch fabric
 - ◆ Ports grouped together on “line cards”
- Kinds of switch fabric
 - ◆ Shared Memory
 - ◆ Bus
 - ◆ Crossbar
- How to deal with transient contention?
 - ◆ Input queuing
 - ◆ Output queuing



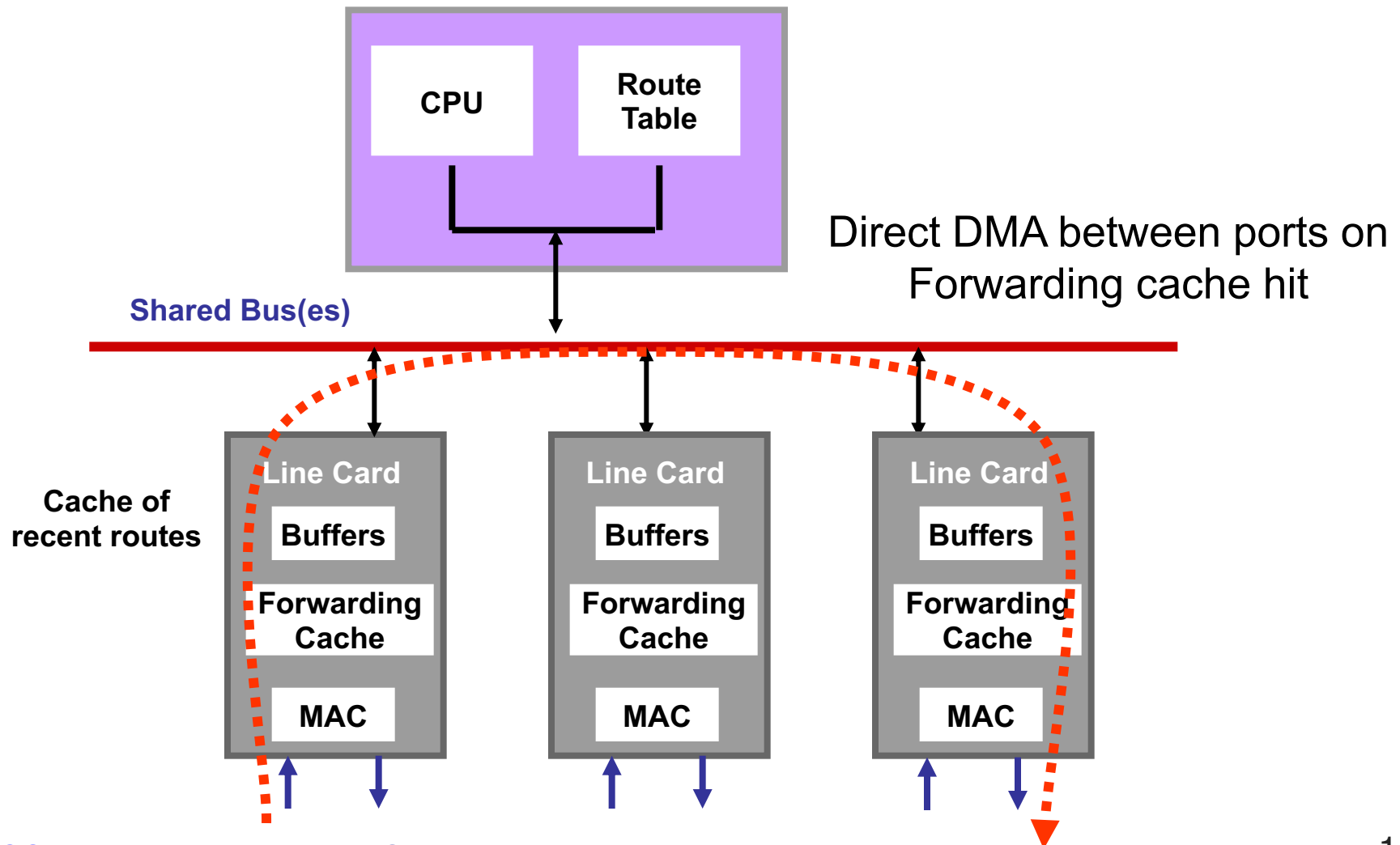
First Generation Routers (mostly software)



Single CPU and
shared memory;

All forwarding
hits main CPU

Second Generation Routers (HW caching forwarding table)



What if input rate exceeds output rate?

Output queuing



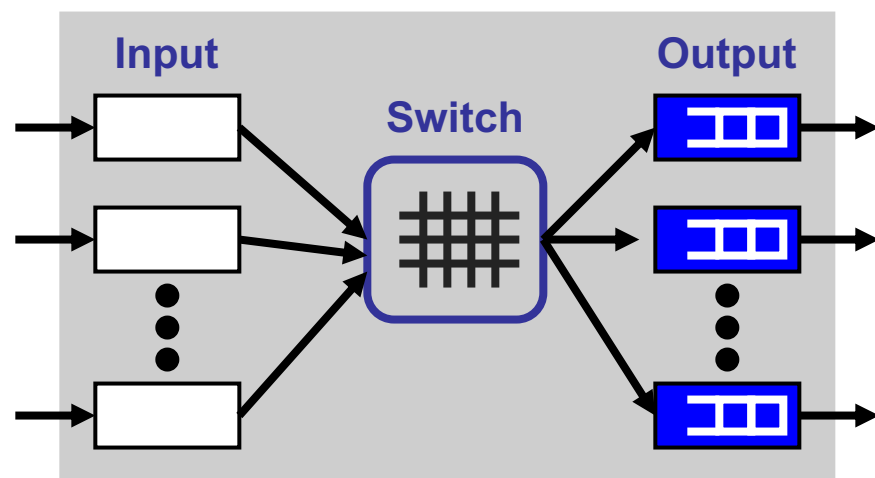
- Output interfaces buffer packets

- Pro

- ◆ Simple queuing 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



What if input rate exceeds output rate?

Input queuing



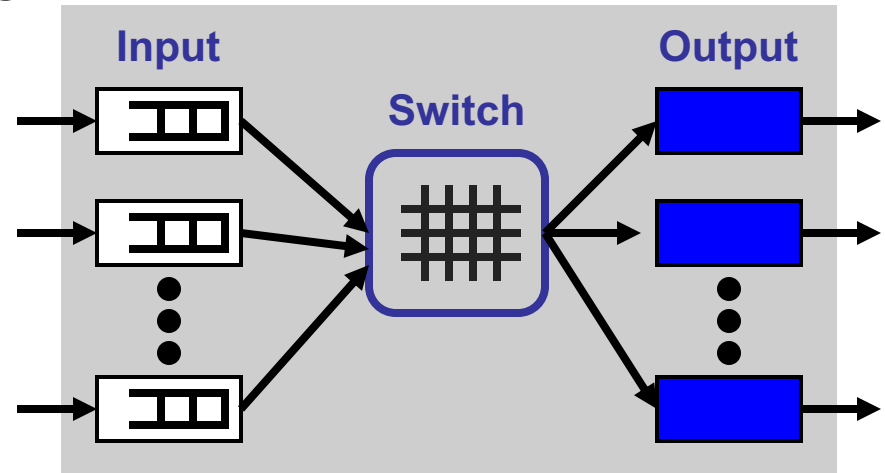
- Input interfaces buffer packets

- Pro

- ◆ Single congestion point
- ◆ Simple to design algorithms

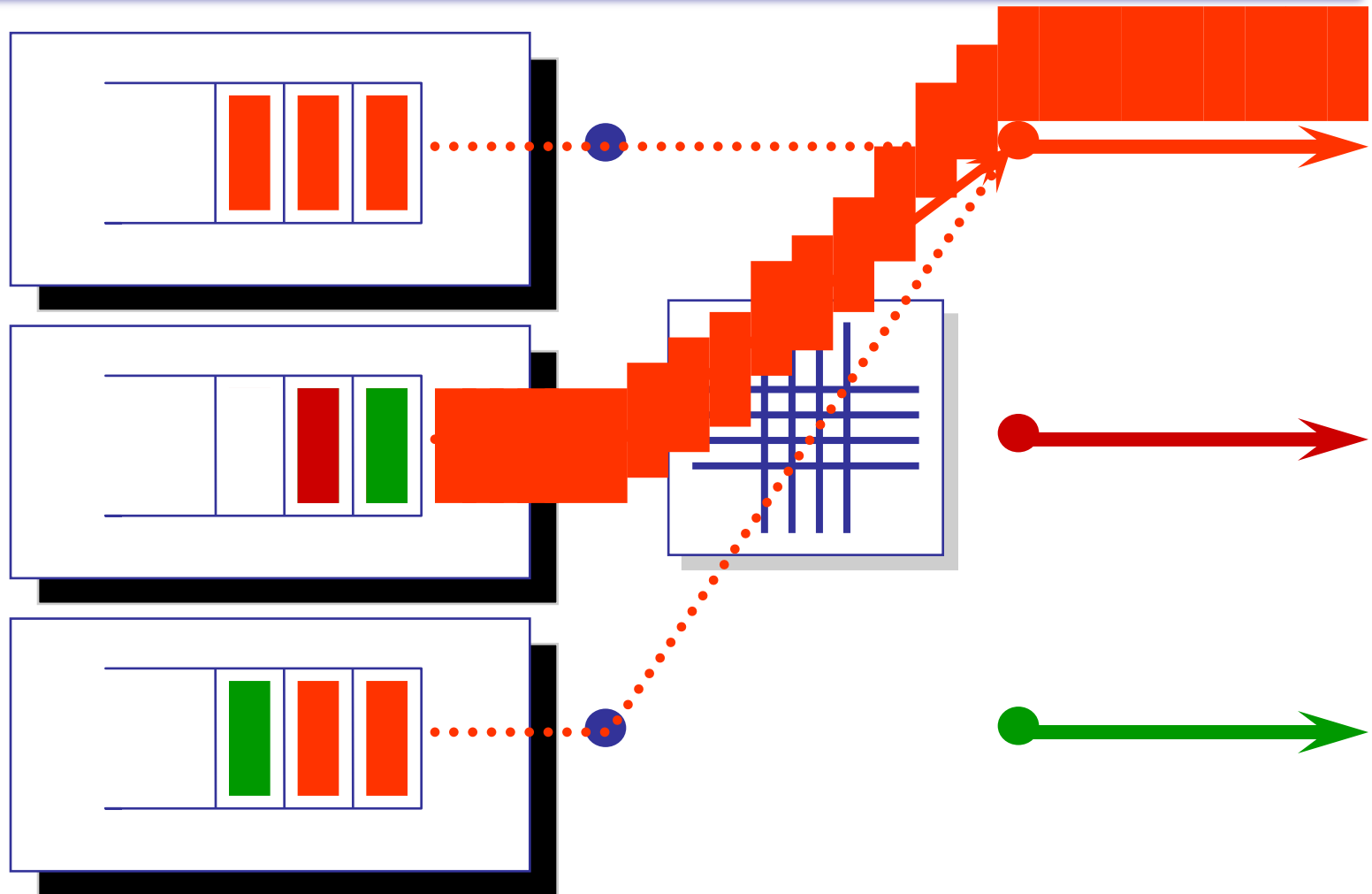
- Con

- ◆ Must implement flow control to outputs
- ◆ Low output utilization due to Head-of-Line (HoL) Blocking





Head-of-Line Blocking



What if input rate exceeds output rate?

IQ + Virtual Output Queuing



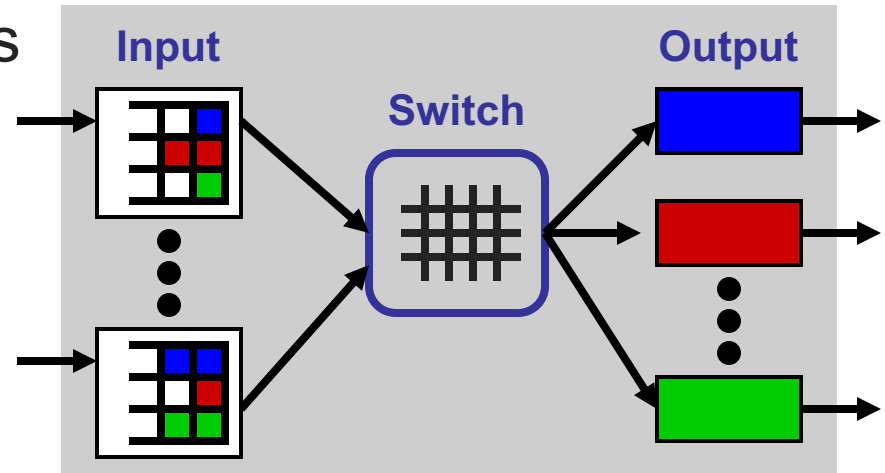
- Input interfaces buffer packets in per-output virtual queues

- Pro

- ◆ Solves blocking problem

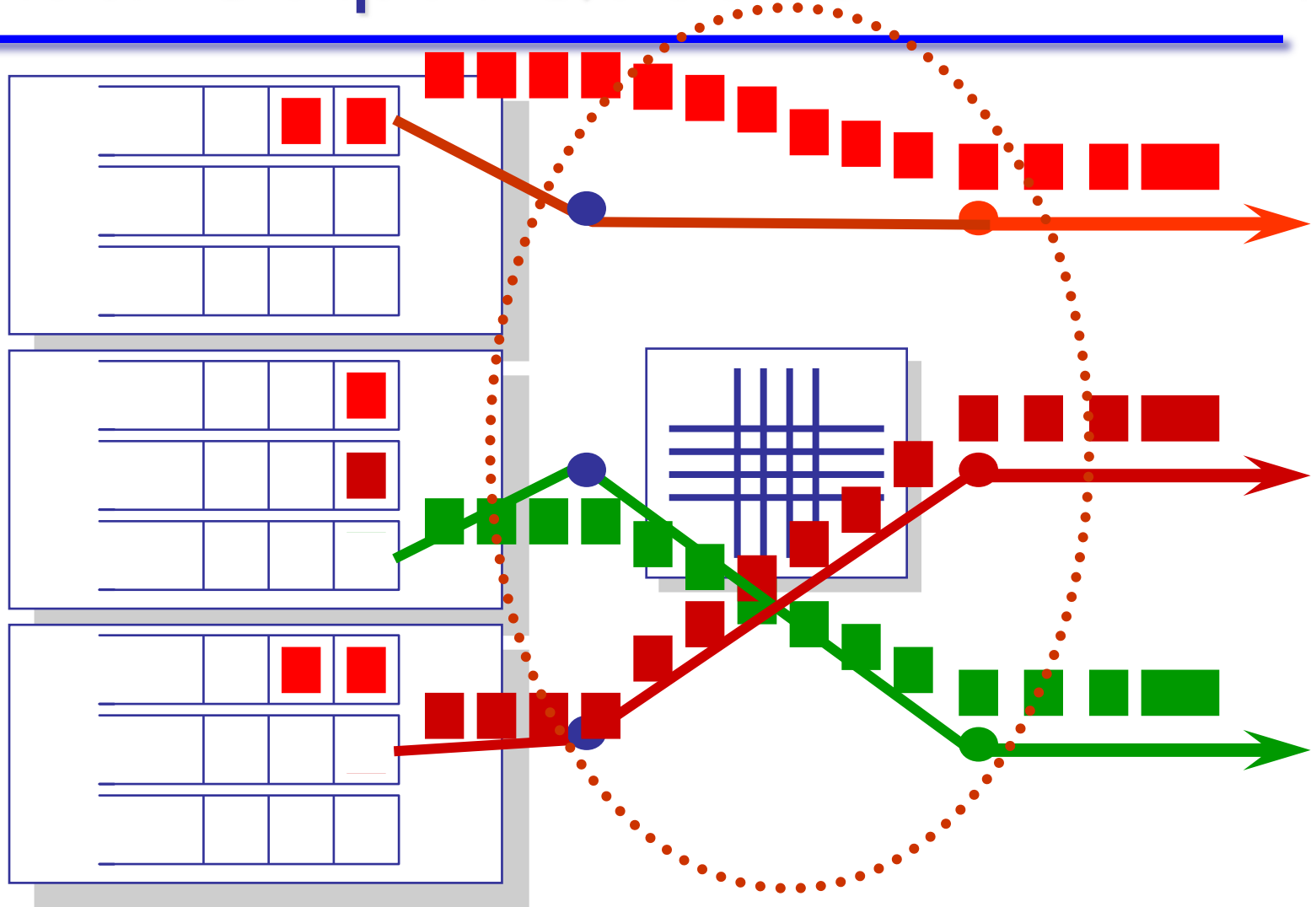
- Con

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





Virtual Output Queues





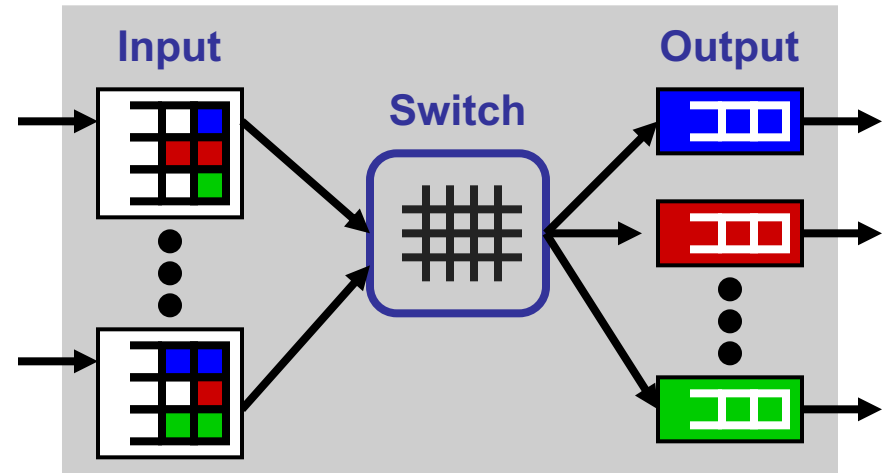
Switch scheduling

- 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

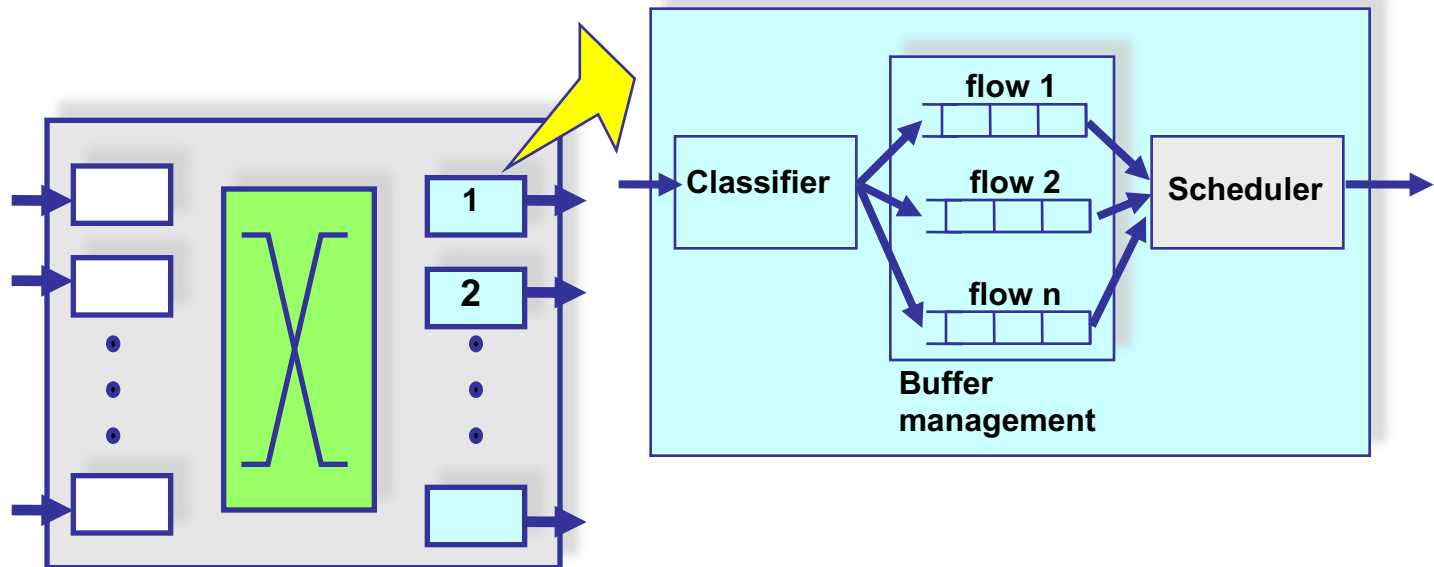


- But what do we do when queues are full (links are saturated)



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

Active Queue Management



- Design active router queue management to aid in reducing the congestion at routers
- Why?
 - ◆ Router has unified view of queuing behavior
 - ◆ Routers see actual queue occupancy (distinguish queue delay and propagation delay.
 - ◆ Routers can see if congestion is transient



Design Objectives

- Keep throughput high and delay low
 - ◆ High power (throughput/delay)
- Accommodate bursts
- Queue size should reflect ability to accept bursts rather than steady-state queuing (that's why they are there!)
- Help with transport protocol (e.g., TCP) performance with minimal hardware changes in router

Random Early Detection (RED)



- Detect incipient congestion
- **Assume hosts respond to lost packets**
 - ◆ We know they will retransmit based on losses
 - ◆ Soon we will see **losses will also make them slow down!**
- Avoid window/end host synchronization
 - ◆ Randomly mark packets
- Do not bias against bursty traffic



For Next Time

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