

# Lesson 06-01: Network Layer Intro and Router

CS 326E Elements of Networking

Mikyung Han

[mhan@cs.utexas.edu](mailto:mhan@cs.utexas.edu)

## Example Protocols

## Responsible for

## Internet Reference Model



FTP, HTTP, SMTP

Application

application specific needs

TCP, UDP

Transport

process to process data transfer

IP

Network

host to host data transfer across different network

Ethernet, WiFi

Link

data transfer between physically adjacent nodes

802.3 PHY

Physical

bit-by-bit or symbol-by-symbol delivery

# Outline

 I. Why Network Layer?

# Why network layer?

- **Responsible for delivering a packet from src host to dst host**
- **Figures out which intermediate hops (route) to take from src to dst**
  - Global action of routing determined by routing algorithm
  - Computed either in a distributed manner (traditional) or centrally (SDN)
  - Control plane's job
- **Each hop should know where to send to the packet received**
  - Includes src, dst, and ALL intermediate hops
  - Local action of forwarding dictated by routing algorithm
  - Data plane's job

# Why **ONLY ONE** protocol in Network layer?

- **Each hop MUST understand this routing/forwarding**
  - Everyone needs to speak the "same language"
- **ONE protocol: IP**
  - IPv4 and IPv6

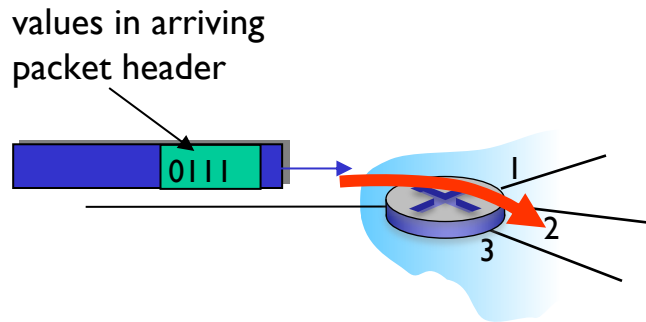
# A packet in network layer is called ...

- Datagram!

# Data plane vs control plane

## Data plane

- **local**, per-router function
- determines which **output port** to forward for a given datagram arriving at router's input port



## Control plane

- **network-wide** logic
- determines the **end-to-end route** from src to dst that this datagram should travel
- two approaches:
  - Computed in **distributed** manner by each router
  - Done **centrally** by SDN controllers

Which (forwarding vs computing routes) should be done faster?

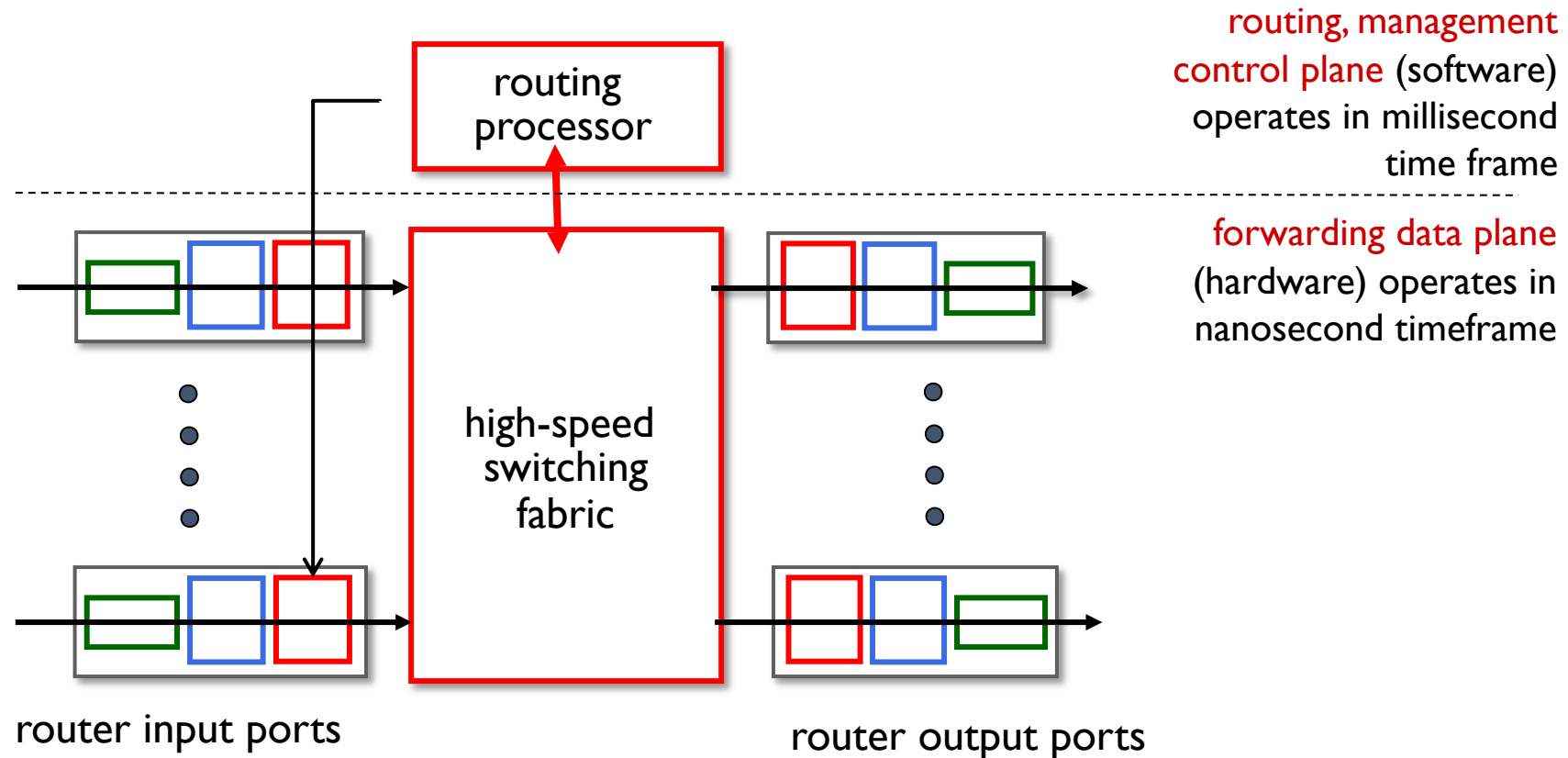
# Outline

1. Why Network Layer?

 2. Router architecture

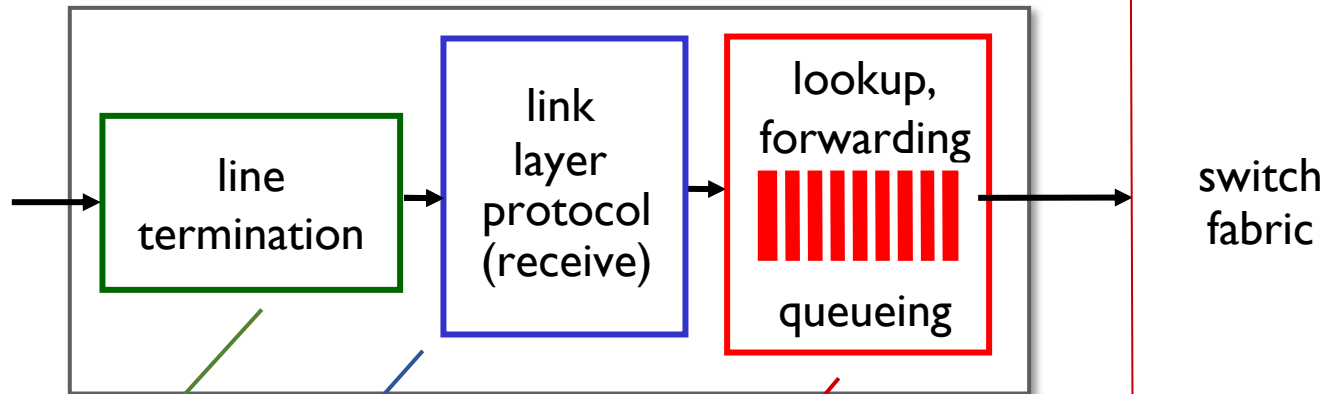


# Router architecture



False! Router performs **physical link** and **network** layer functions

# Input port functions



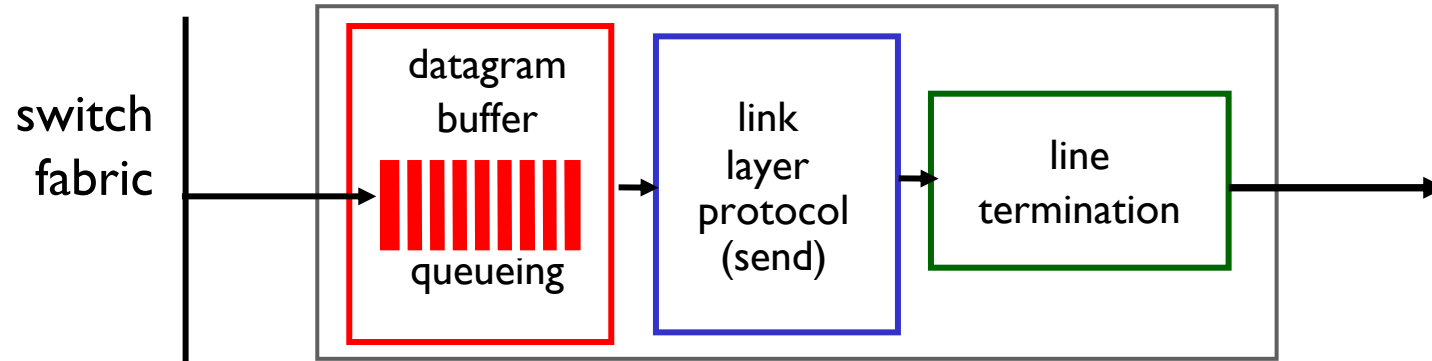
physical layer:  
bit-level reception

link layer:  
e.g., Ethernet  
(chapter 6)

## Match plus action forwarding

- Read header fields, lookup output port in forwarding table
  - **destination-based forwarding**: forward based only on dst IP address
  - **generalized forwarding**: forward based on any header fields including other layers (IP header, Ethernet header, Transport header, etc...)

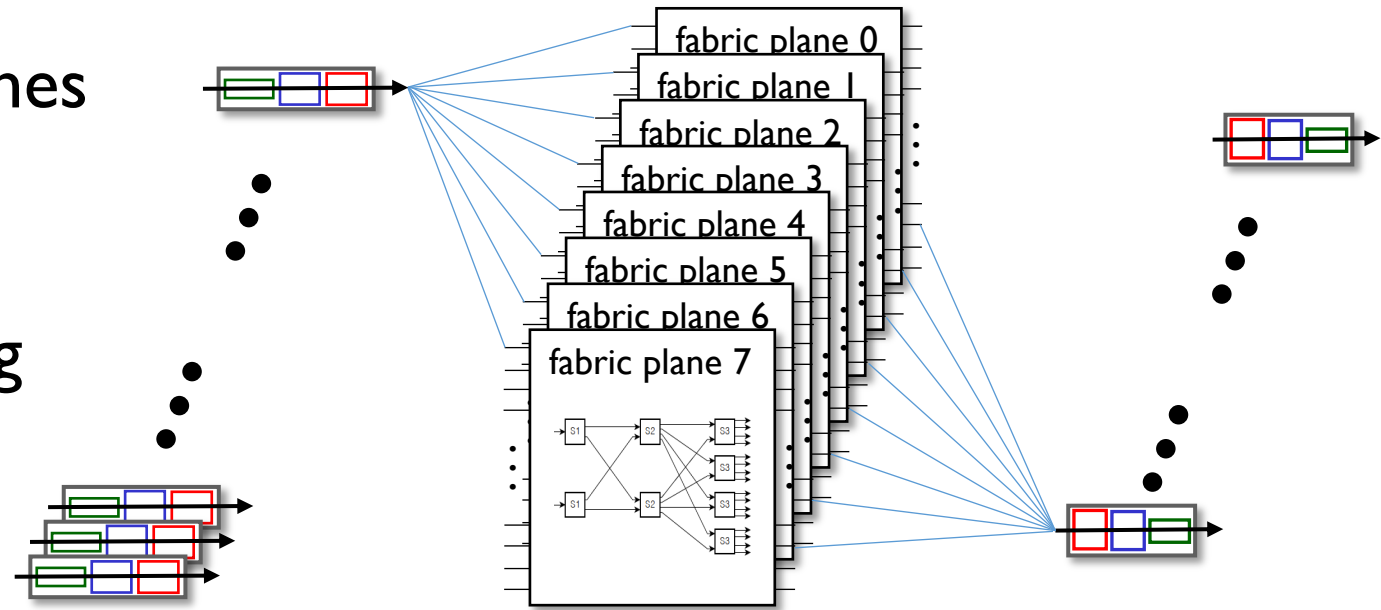
# Output port functions



- **Buffering** required when datagrams arrive from fabric faster than link transmission rate.
- **Drop policy:** which datagrams to drop if no free buffers?
- **Scheduling discipline** chooses among queued datagrams for transmission

# Fabric Switch

- scaling, using multiple switching “planes” in parallel:
  - speedup, scaleup via parallelism
- Cisco CRS router:
  - basic unit: 8 switching planes
  - each plane: 3-stage interconnection network
  - up to 100’s Tbps switching capacity

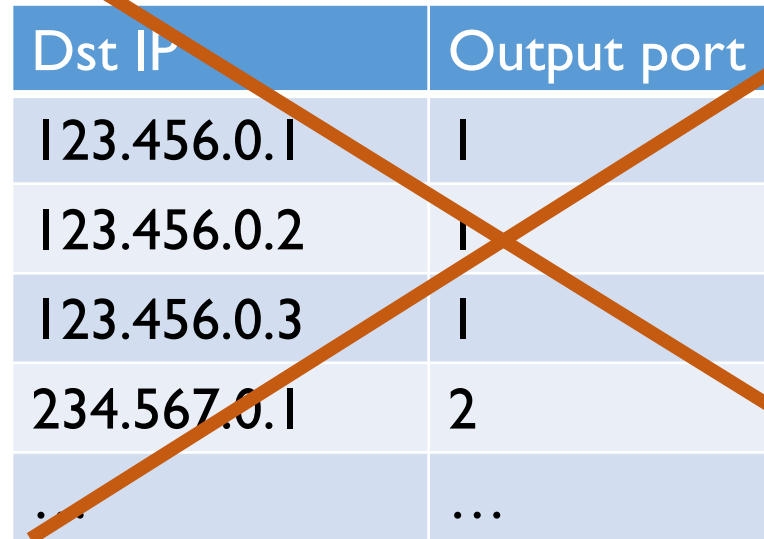


# Outline

1. Why Network Layer?
2. Router architecture
-  3. Destination based forwarding

# Destination-based forwarding considers dst IP address only when it does lookup from the forwarding table

- How does forwarding table look like?



Dst IP	Output port
123.456.0.1	1
123.456.0.2	1
123.456.0.3	1
234.567.0.1	2
...	...

# Longest prefix matching

## longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

examples:

11001000 00010111 00010110 10100001    which interface?

11001000 00010111 00011000 10101010    which interface?

# Longest prefix matching

## longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 <b>match!</b> 1 00011*** *****	2
otherwise	3

examples:

11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?



# Longest prefix matching

## longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

↑  
match!

examples:

11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?

# Longest prefix matching

## longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

match!

examples:

11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?

# Forwarding table uses ranges of IP address

- What if dst IP is 345.10.2.1?

Dst IP ranges	Output port
123.456.0.*	1
234.567.*.*	2
345.10.*.*	3
345.*.*.*	4
otherwise	5

Longest prefix matching: Look for the most specific requirement!

# Longest prefix matching and TCAMs

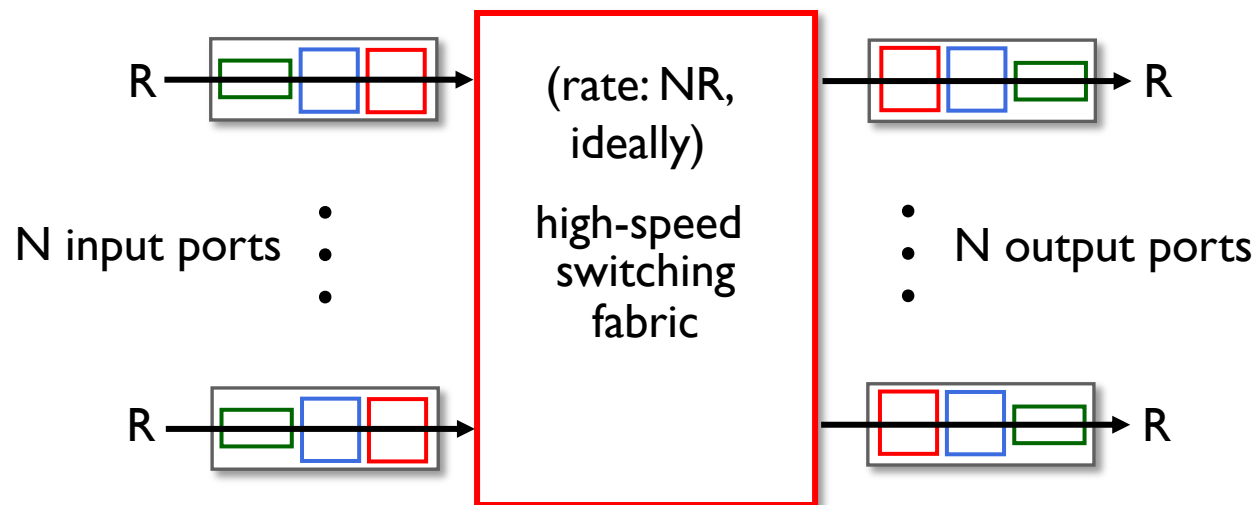
- Often performed using ternary content addressable memories (TCAMs)
  - *content addressable*: present address to TCAM: retrieve address in one clock cycle, regardless of table size
  - Cisco Catalyst: ~1M routing table entries in TCAM

# Outline

1. Why Network Layer?
2. Router architecture
3. Destination based forwarding
-  4. **Switching Fabrics**

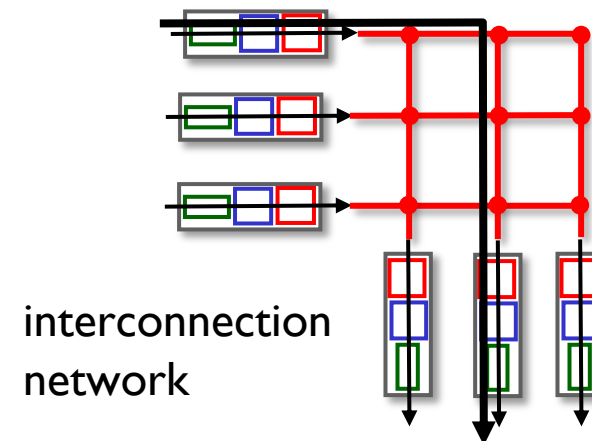
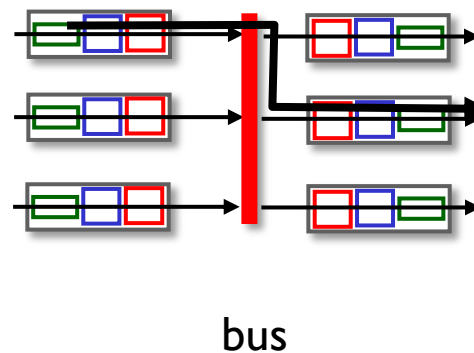
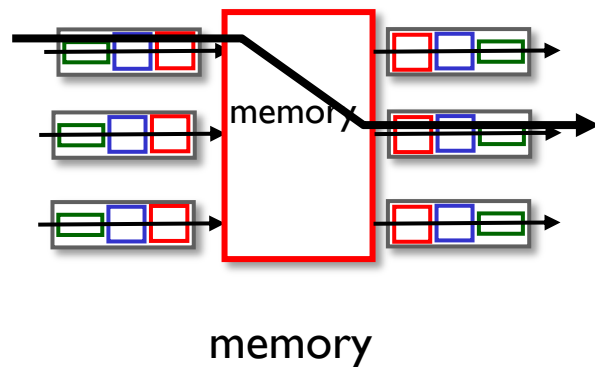
# Switching fabrics

- transfer packet from input link to appropriate output link
- **switching rate:** rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable



# Switching fabrics

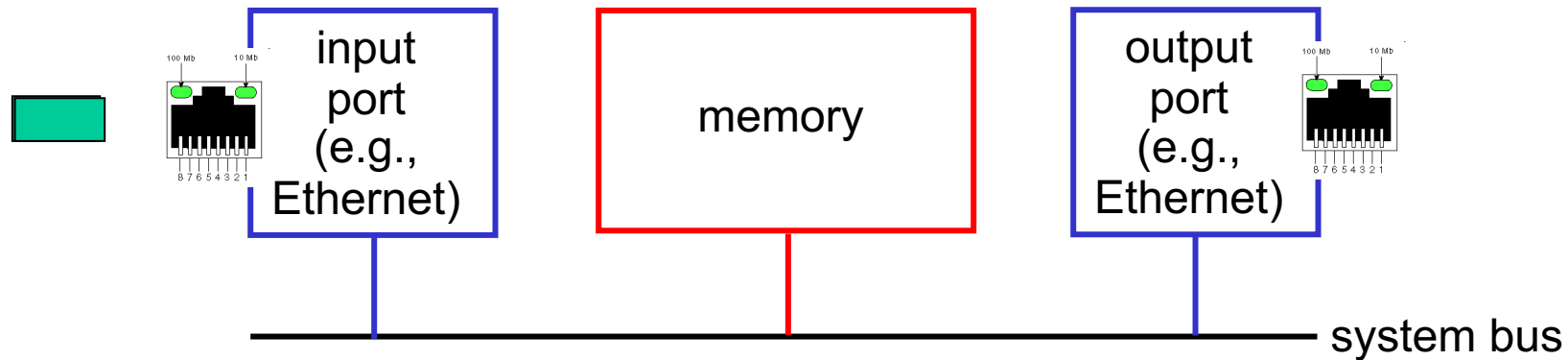
- transfer packet from input link to appropriate output link
- **switching rate:** rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- three major types of switching fabrics:



# Switching via memory

## first generation routers:

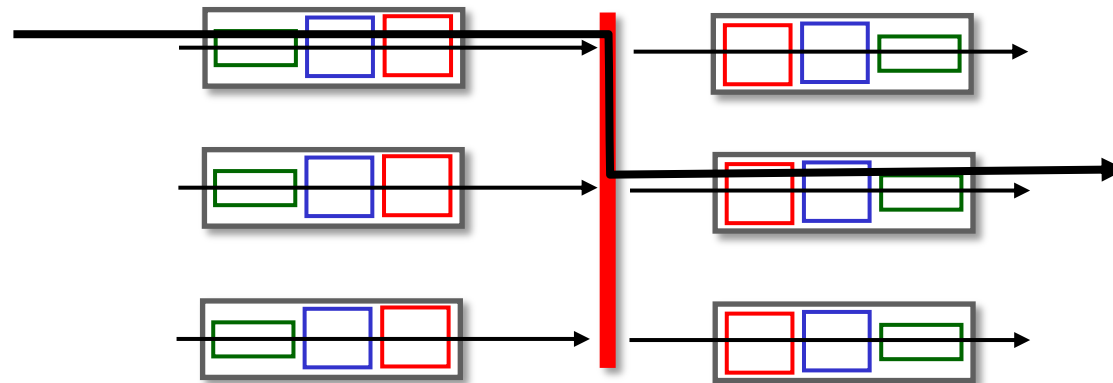
- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)





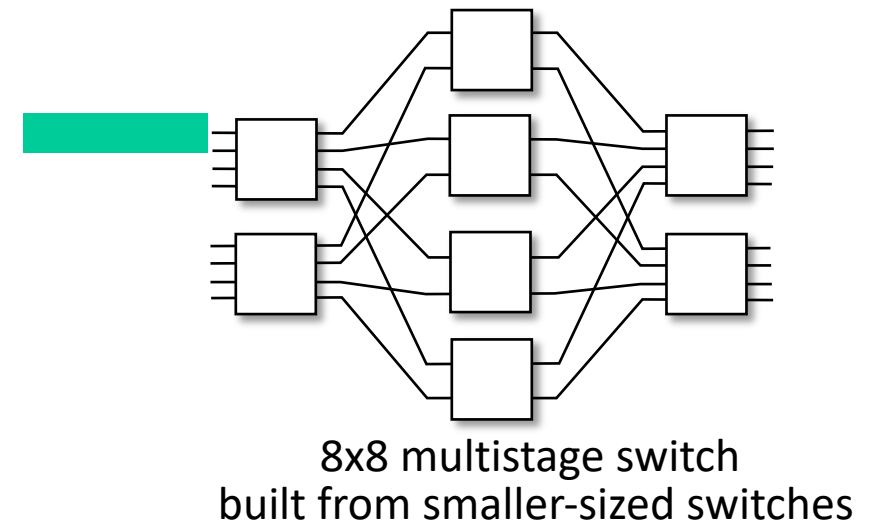
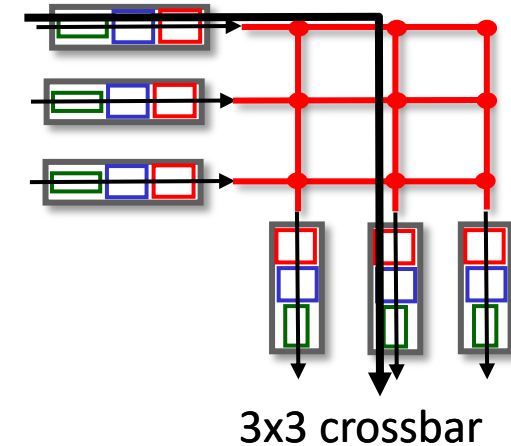
# Switching via a bus

- datagram from input port memory to output port memory via a shared bus
- *bus contention*: switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access routers



# Switching via interconnection network

- Crossbar, Clos networks, other interconnection nets initially developed to connect processors in multiprocessor
- **multistage switch**:  $n \times n$  switch from multiple stages of smaller switches
- **exploiting parallelism**:
  - fragment datagram into fixed length cells on entry
  - switch cells through the fabric, reassemble datagram at exit

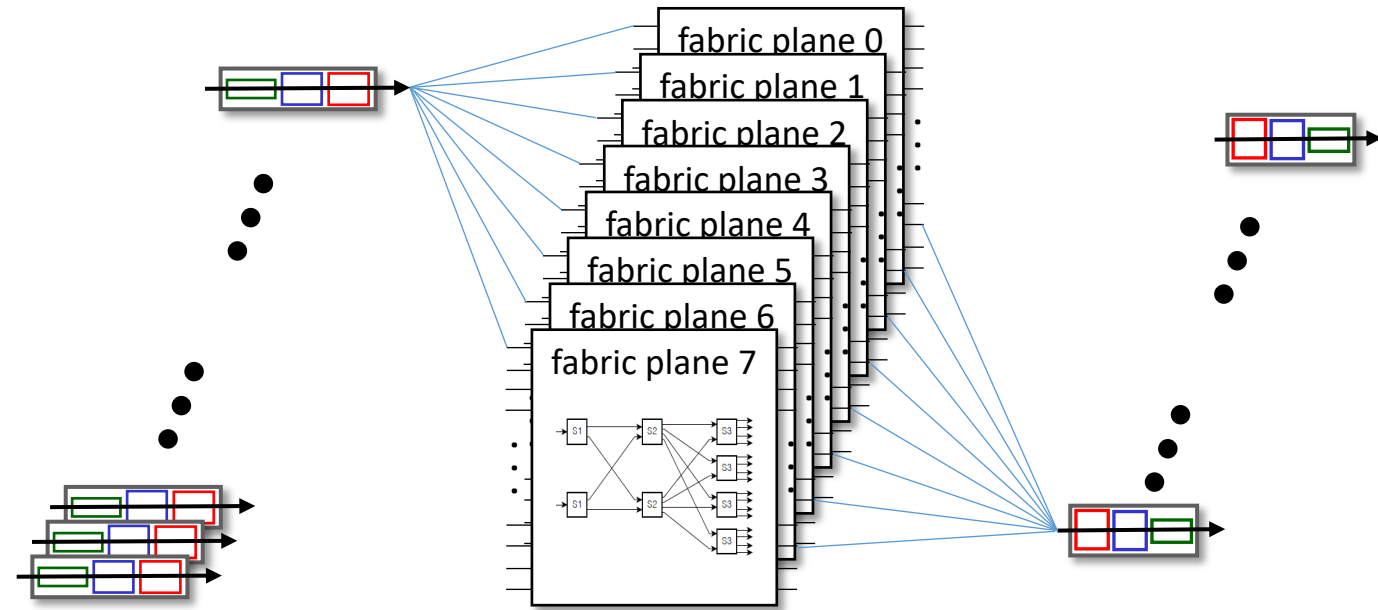


# Switching via interconnection network

- scaling, using multiple switching “planes” in parallel:
  - speedup, scaleup via parallelism

- Cisco CRS router:

- basic unit: 8 switching planes
- each plane: 3-stage interconnection network
- up to 100's Tbps switching capacity

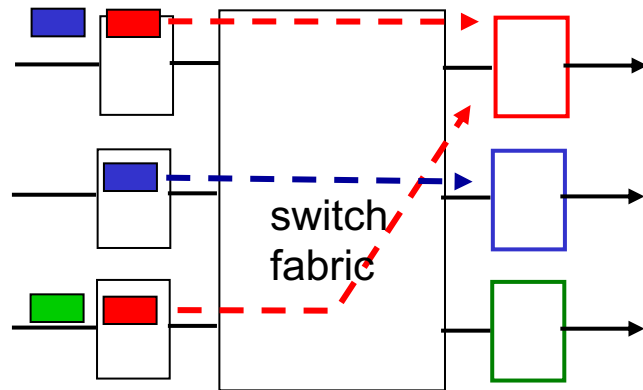


# Outline

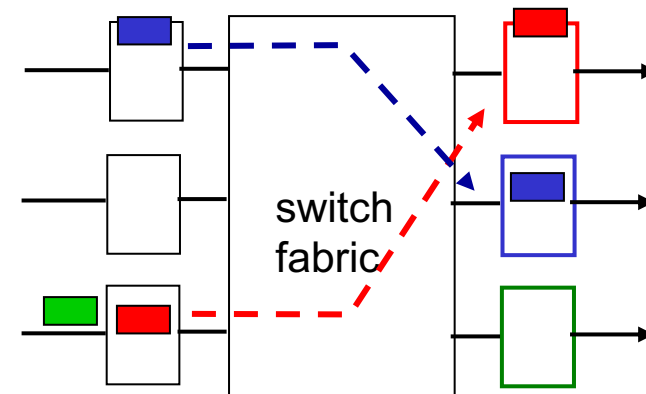
1. Why Network Layer?
2. Router architecture
3. Destination based forwarding
4. Switching Fabrics
-  5. **Queuing and Scheduling**

# Input port queuing

- If switch fabric slower than input ports combined -> queuing may occur at input queues
  - queuing delay and loss due to input buffer overflow!
- **Head-of-the-Line (HOL) blocking:** queued datagram at front of queue prevents others in queue from moving forward

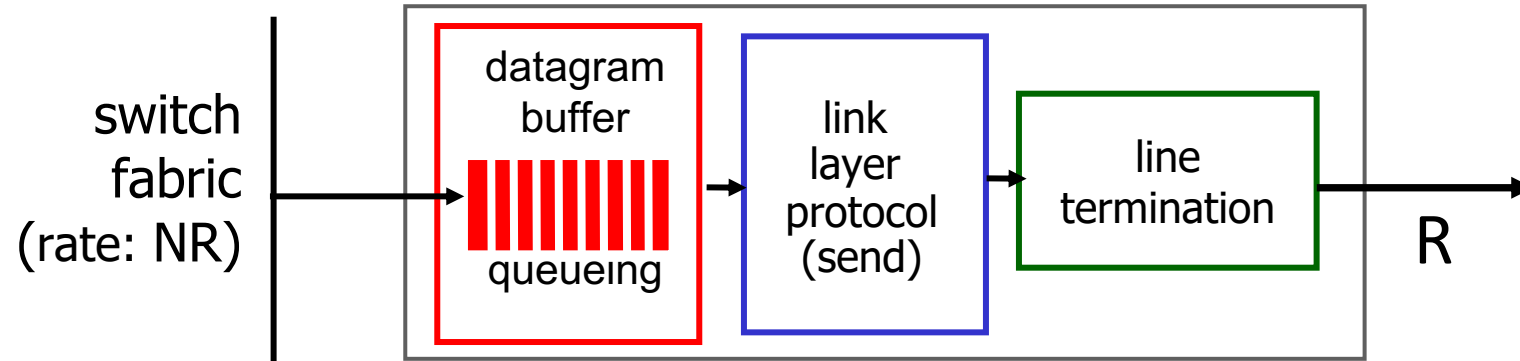


output port contention: only one red datagram can be transferred. lower red packet is *blocked*



one packet time later: green packet experiences HOL blocking

# Output port queuing



This is a really important slide

- **Buffering** required when datagrams arrive from fabric faster than link transmission rate. **Drop policy**: which datagrams to drop if no free buffers?
- **Scheduling discipline** chooses among queued datagrams for transmission

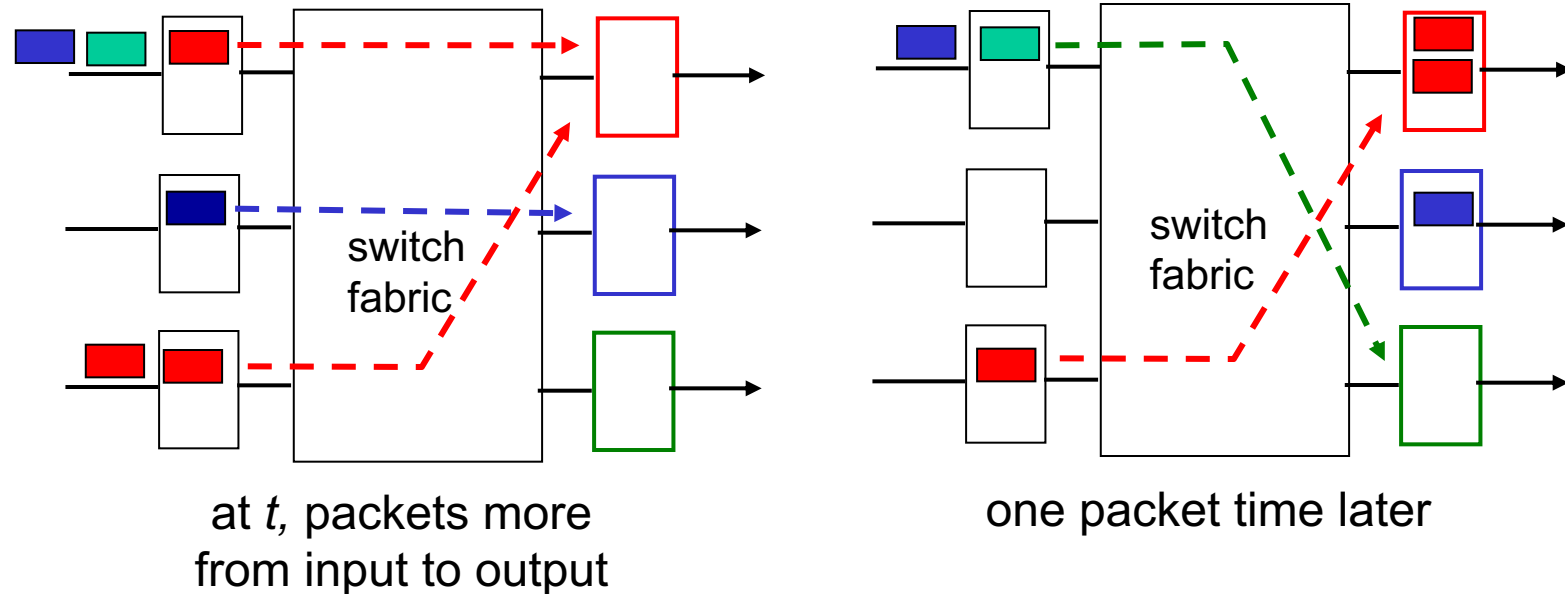


Datagrams can be lost due to congestion, lack of buffers



Priority scheduling – who gets best performance, network neutrality

# Output port queuing



- buffering when arrival rate via switch exceeds output line speed
- queueing (delay) and loss due to output port buffer overflow!

# How much buffering?

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity  $C$ 
  - e.g.,  $C = 10$  Gbps link: 2.5 Gbit buffer

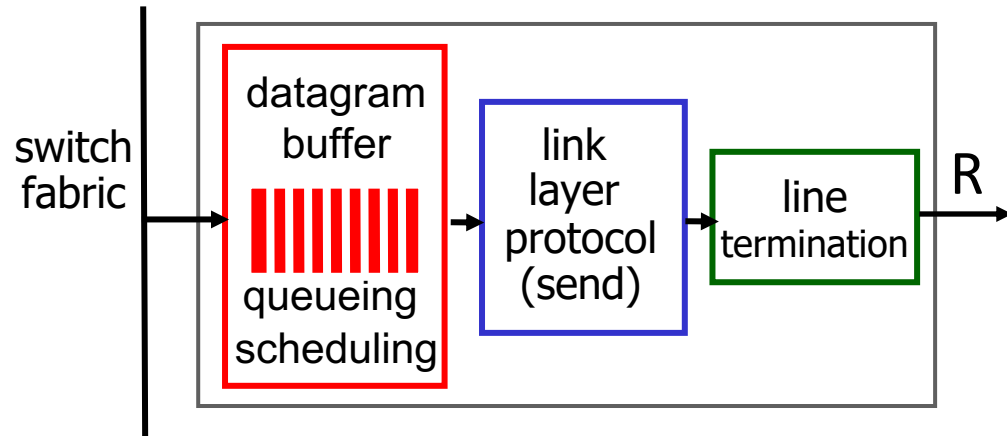
- more recent recommendation: with  $N$  flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

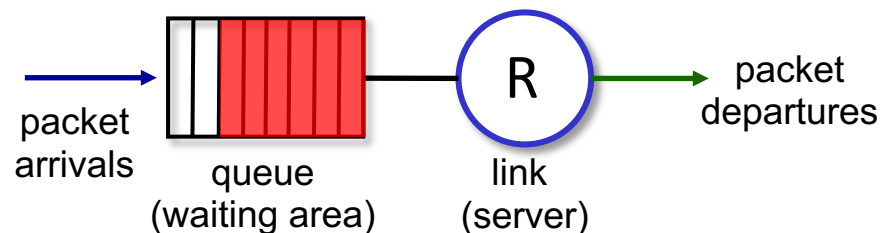
- but too much buffering can increase delays (particularly in home routers)
  - long RTTs: poor performance for realtime apps, sluggish TCP response
  - recall delay-based congestion control: “keep bottleneck link just full enough (busy) but no fuller”



# Buffer Management



## Abstraction: queue



## buffer management:

- **drop:** which packet to add, drop when buffers are full
  - **tail drop:** drop arriving packet
  - **priority:** drop/remove on priority basis
- **marking:** which packets to mark to signal congestion (ECN, RED)

# Packet Scheduling: FCFS

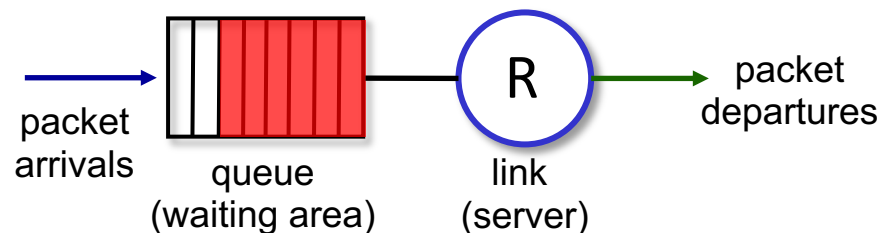
**packet scheduling:** deciding which packet to send next on link

- first come, first served
- priority
- round robin
- weighted fair queueing

**FCFS:** packets transmitted in order of arrival to output port

- also known as FIFO

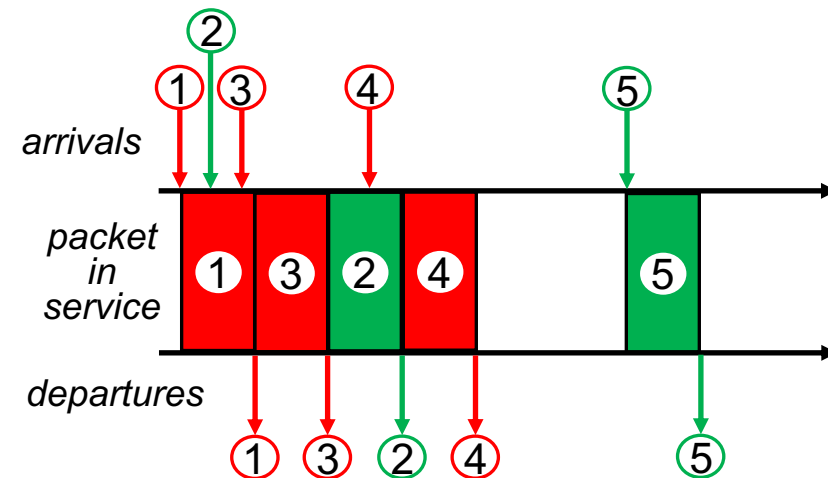
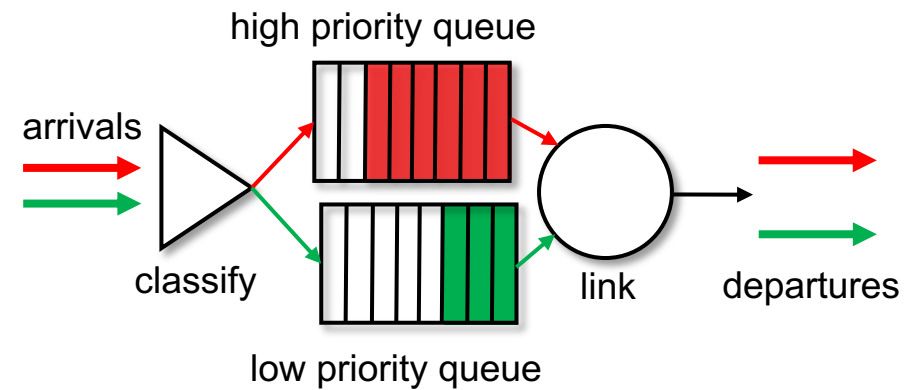
Abstraction: queue



# Scheduling policies: priority

## Priority scheduling:

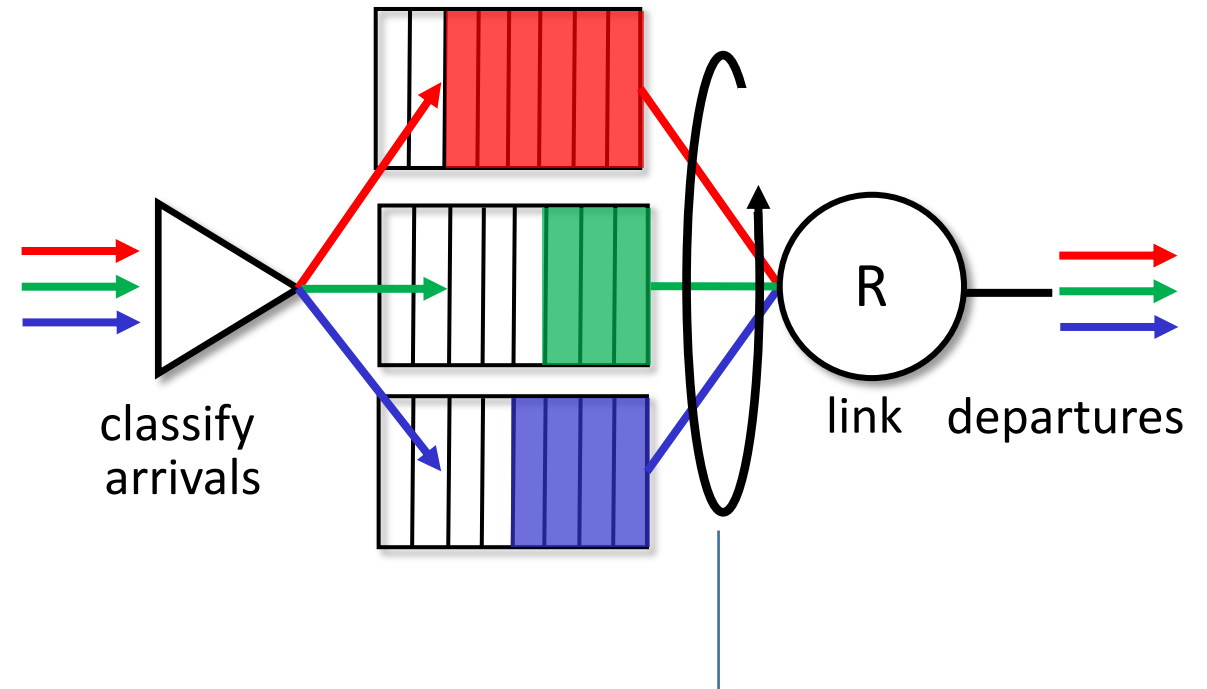
- arriving traffic classified, queued by class
  - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
  - FCFS within priority class



# Scheduling policies: round robin

## *Round Robin (RR) scheduling:*

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn



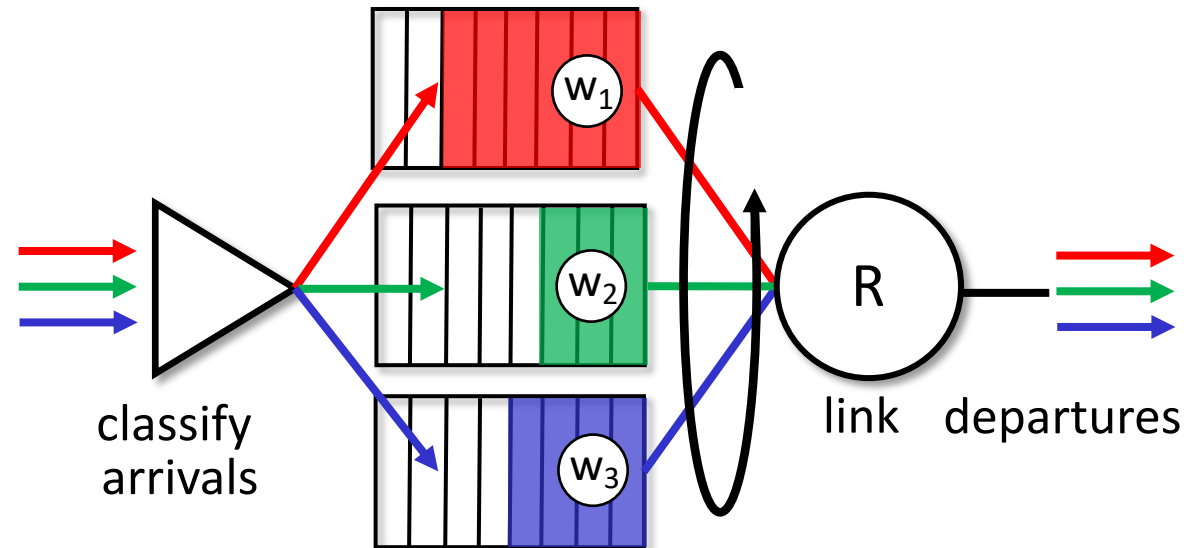
# Scheduling policies: weighted fair queueing

## Weighted Fair Queuing (WFQ):

- generalized Round Robin
- each class,  $i$ , has weight,  $w_i$ , and gets weighted amount of service in each cycle:

$$\frac{w_i}{\sum_j w_j}$$

- minimum bandwidth guarantee (per-traffic-class)



# Acknowledgements

Slides are adopted from Kurose' Computer Networking Slides