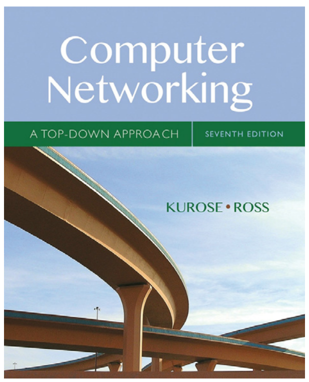
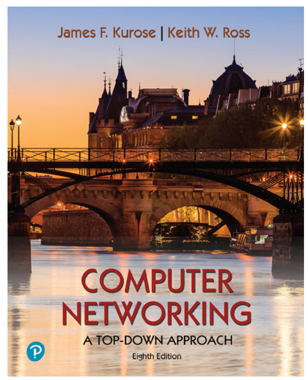
Chapter 4
Network Layer: Data Plane

Courtesy to the textbooks' authors and Pearson Addison-Wesley because many slides are adapted from the following textbooks and their associated slides.



Jim Kurose, Keith Ross, "Computer Networking: A Top Down Approach", 7th Edition, Pearson, 2016.



Jim Kurose, Keith Ross, "Computer Networking: A Top Down Approach", 8th Edition, Pearson, 2020.

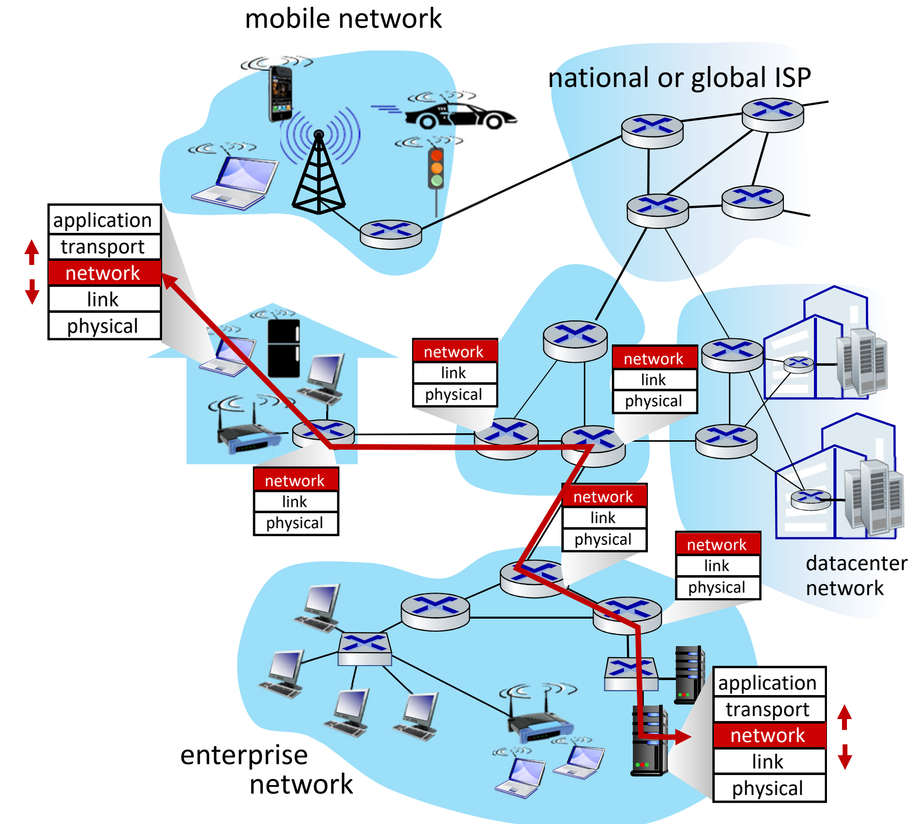
Network layer: “data plane” roadmap

- Network layer: overview
  - forwarding
  - routing
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes

Network-layer services and protocols

- network layer protocols in *many Internet device*, including
  - hosts, routers
    - unlike transport layer protocols (not on routers)
- routers:
  - examines header fields in all IP datagrams passing through it
  - moves datagrams from input ports to output ports to transfer datagrams along end-to-end path



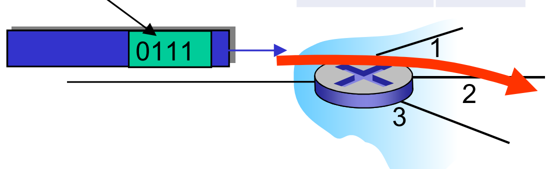
Two key network-layer functions

network-layer functions:

- *forwarding:* move packets from a input port/link to appropriate output port/link

header	port
0111	2
0010	1

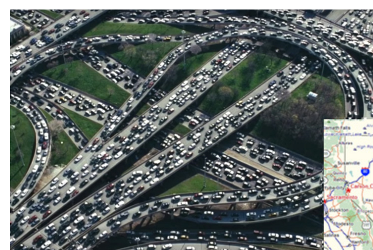
values in arriving packet header



- *routing:* determine route/path taken by packets from source to destination
  - *routing algorithms*

analogy: taking a trip

- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination



forwarding

routing

Network service model

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

example services for *individual* datagrams:

- guaranteed delivery (or not)
- guaranteed delivery with less than 40 msec delay (or not)

example services for a *flow* of datagrams:

- in-order datagram delivery (or not)
- guaranteed minimum bandwidth to a flow (or not)
- restrictions on changes in inter-packet spacing (or not)

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no

Internet "best effort" service model

No guarantees on:

i. successful datagram delivery to destination
ii. timing or order of delivery
iii. bandwidth available to end-to-end flow

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet	Diffserv (RFC 2475)	possible	possibly	possibly	no

Reflections on best-effort service:

- simplicity of mechanism has allowed Internet to be widely deployed and adopted
- sufficient provisioning of bandwidth allows performance of real-time applications (e.g., interactive voice, video) to be “good enough” for “most of the time”
- Internet’s basic best-effort service model combined with adequate bandwidth provisioning have arguably proven to be more than “good enough”
  - to enable an amazing range of applications including
    - streaming video services such as Netflix and
    - video-over-IP, real-time conferencing applications such as Skype and Google Meet

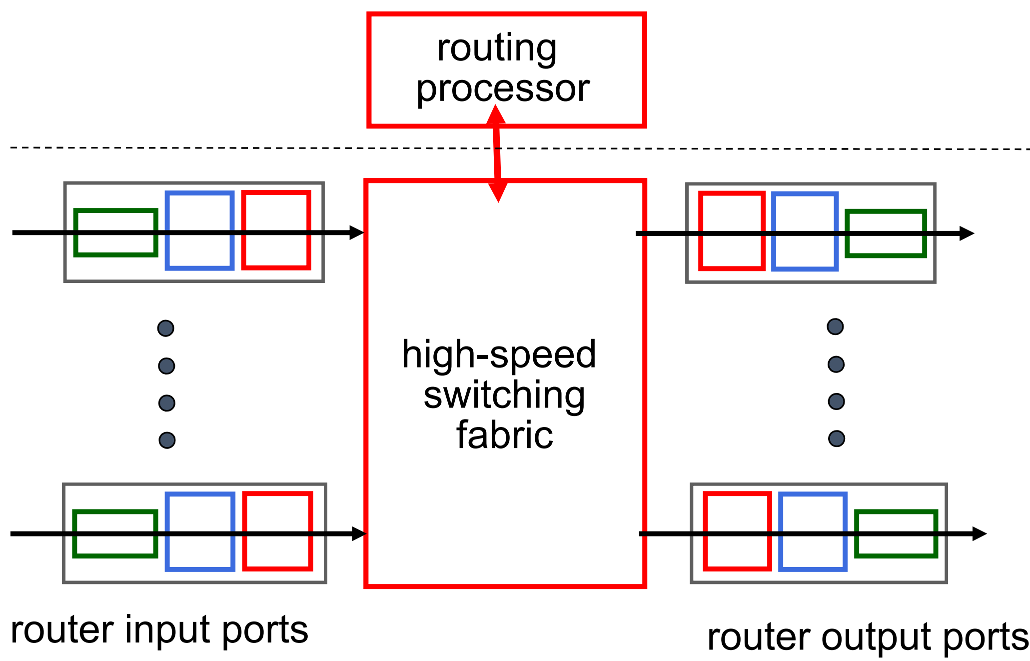
Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized Forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes

Router architecture overview

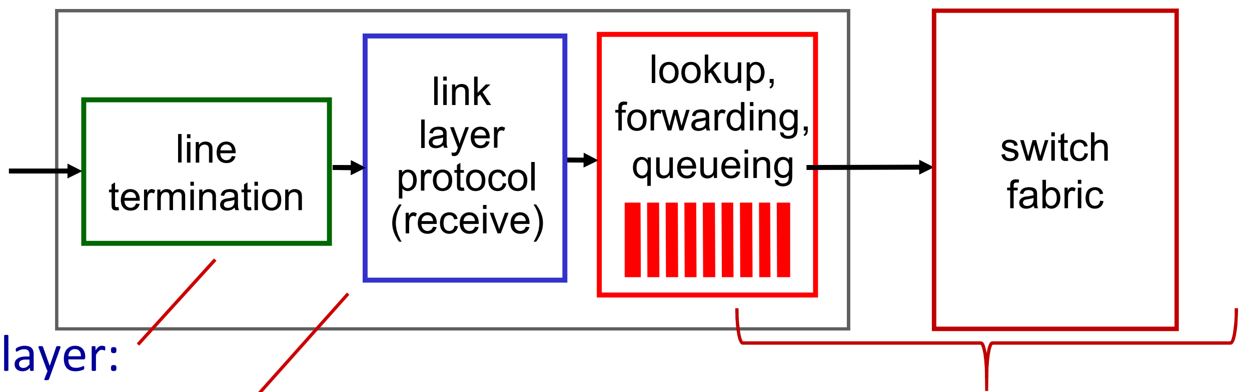
high-level view of generic router architecture:



routing, management
control plane (software)
operates in millisecond
time frame

forwarding data plane
(hardware) operates
in nanosecond
timeframe

Input port functions



physical layer:
bit-level reception

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

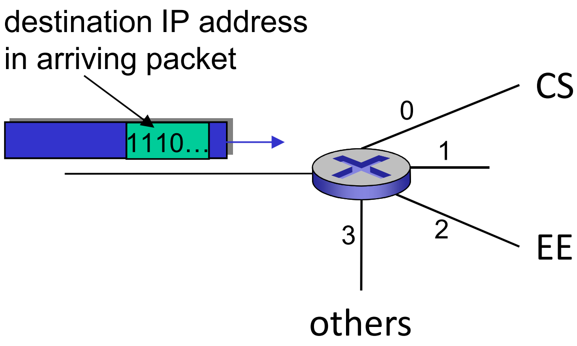
decentralized switching:

- goal: forwarding (from a input port to a proper output port)
  - read header field(s)
  - table-lookup using forwarding table (match plus action)
- which output port to go based on
  - destination IP address (traditional)
  - any set of header field values (SDN)
- queueing at input and output ports happens

IP addressing

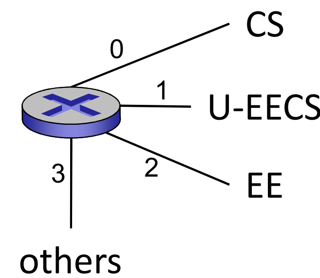
- A IP address (version 4) is 32-bit long
- (obsolete) Classful addressing
  - Class A: /8 (0******* ******** ******** ********)
    - Network number field is 8-bit
    - # of IP addresses per network is 2^{32-8} = 16,777,216
  - Class B: /16 (10****** ******** ******** ********)
  - Class C: /24 (110***** ******** ******** ********)
  - Class D (multicast, 1110*) and class E (reserved, 1111*)
- CIDR (Classless Inter-Domain Routing): /n
  - As long as n is a reasonable integer

IP address assignment



	Destination address range	port	
CS	11001000 00010111 00010000 00000000 **Through** 11001000 00010111 00010111 11111111	0	200.23.16~23.x
EE	11001000 00010111 00011000 00000000 **Through** 11001000 00010111 00011111 11111111	2	200.23.24~31.x
	otherwise	3	others

	Destination address range	port	
CS	11001000 00010111 00010000 00000000 **Through** 11001000 00010111 00010111 11111111	0	200.23.16~23.x
EE	11001000 00010111 00011000 00000000 **Through** 11001000 00010111 00011111 11111111	2	200.23.24~31.x
	otherwise	3	others



	Destination address range	port		
CS	11001000 00010111 00010000 00000000 **Through** 11001000 00010111 00010111 11111111	0	200.23.16~23.x	
U-EECS	11001000 00010111 00011000 00000000 **Through** 11001000 00010111 00011000 11111111	1	200.23.24.x	
EE	11001000 00010111 00011001 00000000 **Through** 11001000 00010111 00011111 11111111	2	200.23.25~31.x	200.23.24~31.x except 200.23.24.x
	otherwise	3	others	

Destination-based forwarding: 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 000[illegible]0111 00011000 ********	1
11001000 [illegible]1 00011*** ********	2
otherwise	3

match!

example 1: 11001000 00010111 00010110 10100001 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!

match!

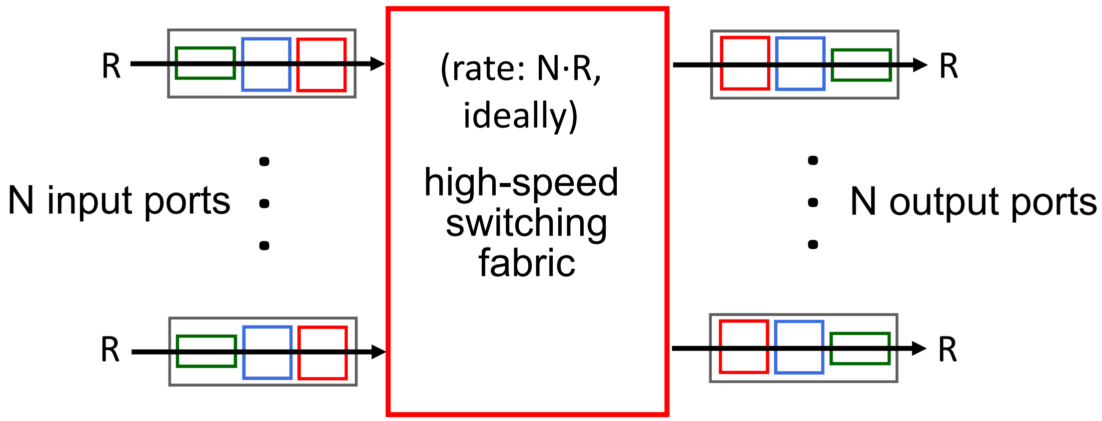
example 2: 11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

- we'll see *why* longest prefix matching is used shortly, when we study addressing
- longest prefix matching: 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

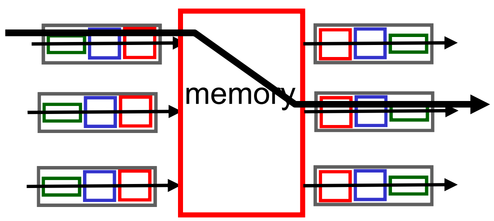
Switching fabrics

- transfer packet from input link/port to appropriate output link/port
- switching rate: rate at which packets can be transferred from inputs to outputs
  - often measured as multiple of input/output line rate/speed
  - N inputs: it is desirable to have switching rate N times faster than the line rate



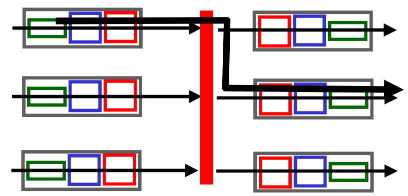
Switching fabrics

- three major types of switching fabrics:



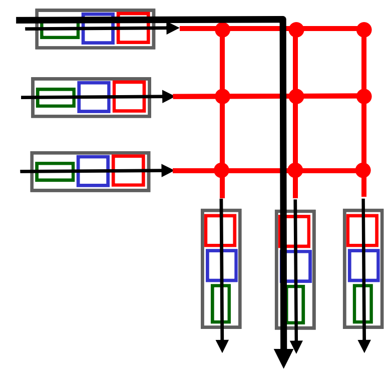
memory

slow speed,
low cost



bus

medium speed (e.g., 32 Gbps),
medium cost

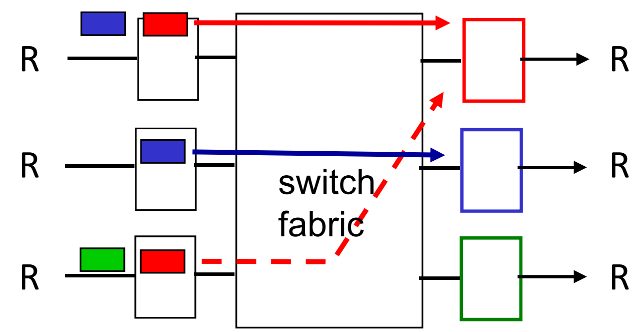


crossbar
(interconnection network)

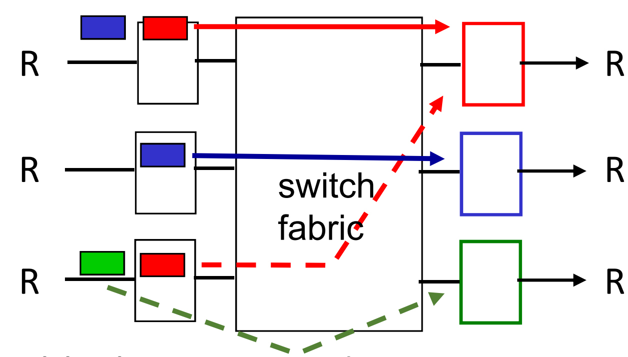
high speed (e.g., 100 Tbps),
high cost

Input port queuing

- queueing may occur at input queues, even when switch fabric is fast enough
  - queueing delay
  - loss due to input buffer overflow!
- output port contention
  - suppose: to an output port, switch fabric can transfer only one packet at a time
    - what if switch fabric can transfer multiple packets to an output port at a time?
- Head-of-the-Line (HOL) blocking
  - queued datagram at front of queue prevents others in queue from moving forward

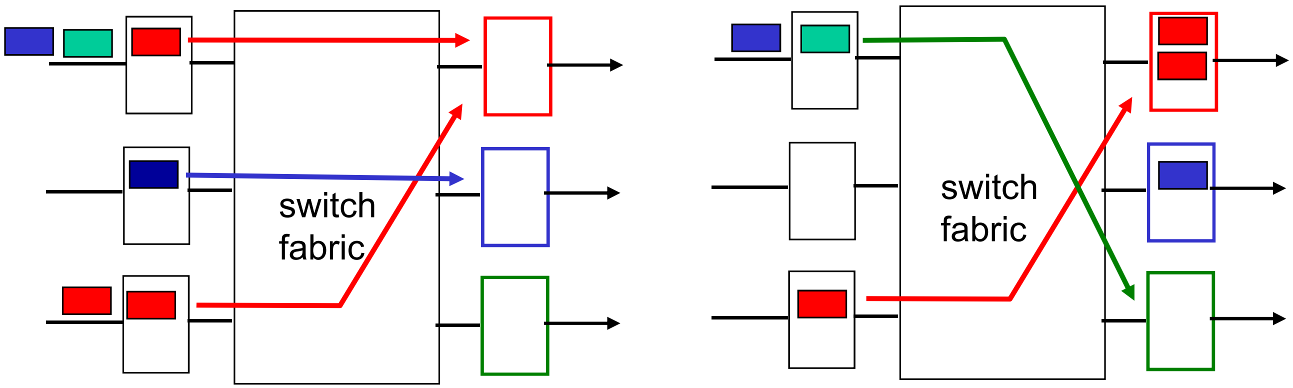


output port contention: Only one red datagram can be transferred to upper output port. Lower red one can't be forwarded at the same time.



HOL blocking: Green datagram experiences HOL blocking, since it has to wait for the red datagram.

Output port queuing



at *t*, packets more from input to output

one packet time later

- 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 times link capacity R
  - e.g., R = 10 Gbps and RTT = 0.25 s ➔ 2.5 Gbit buffer
- more recent recommendation: with *N* flows, buffering equal to

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

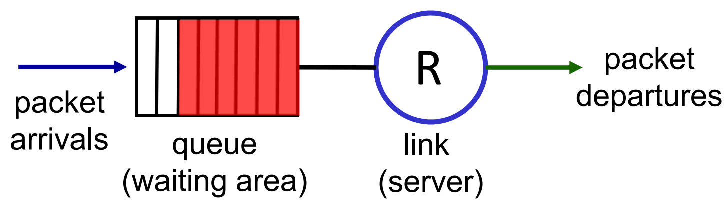
- but *too* much buffering can increase delays (particularly in home routers)
  - long RTTs: poor performance for real-time apps, sluggish TCP response
  - recall delay-based congestion control: "keep bottleneck link just full enough (busy) but no fuller"

Packet Scheduling: FCFS

packet scheduling: deciding which packet to send next on link

- first come, first served (FCFS)
- priority
- round robin
- weighted fair queueing

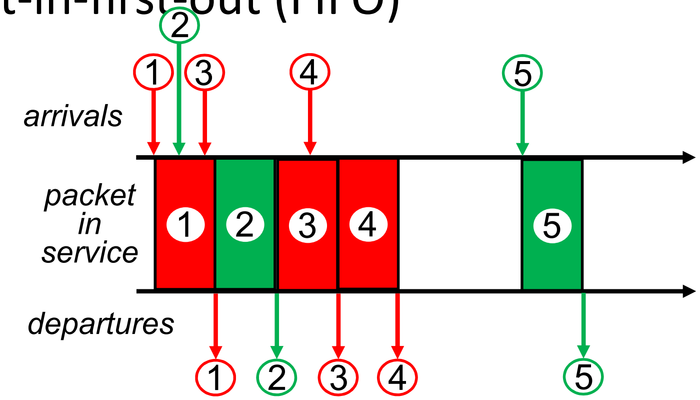
Abstraction: queue



Scheduling policies: FCFS

FCFS: packets are transmitted in the order of arrival to output port

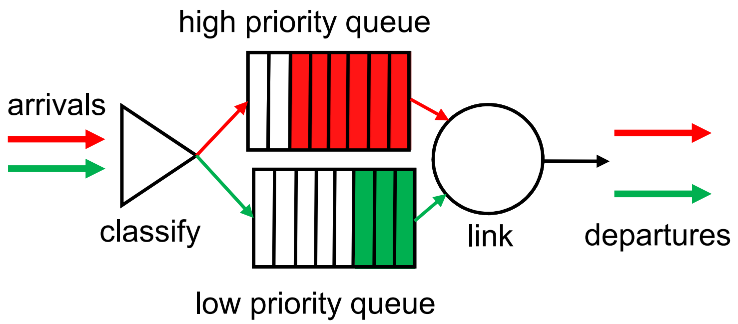
- also known as: First-in-first-out (FIFO)

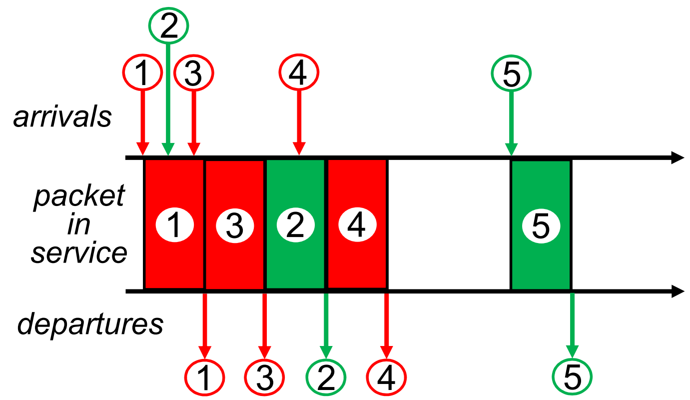


Scheduling policies: priority

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 the same priority class

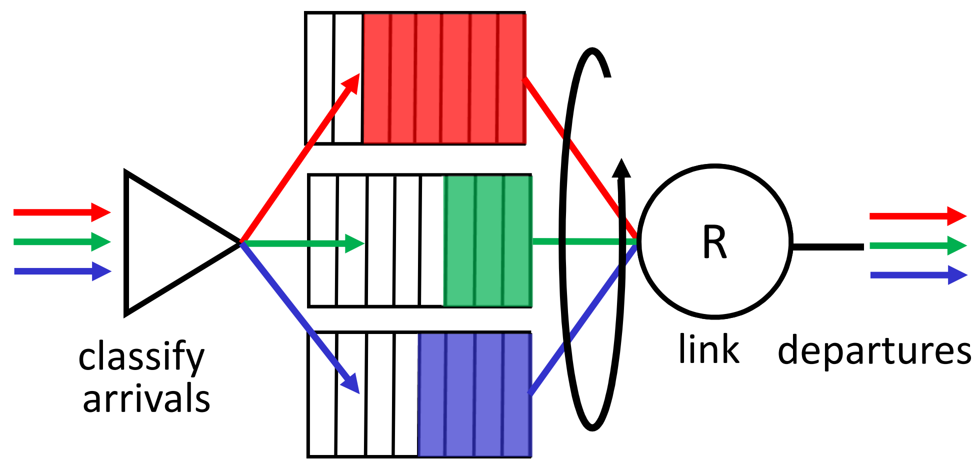




Scheduling policies: round robin

Round Robin (RR) scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- cyclically and 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:

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

- it guarantees minimum bandwidth for each class