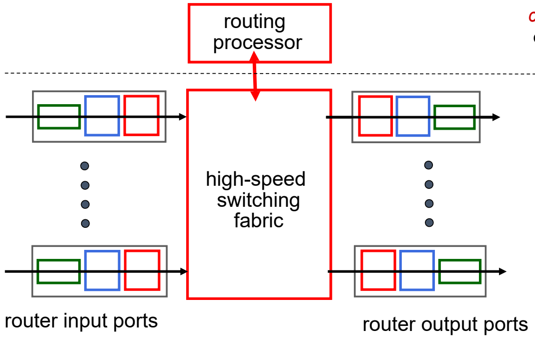
Network layer: “data plane” roadmap

- 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

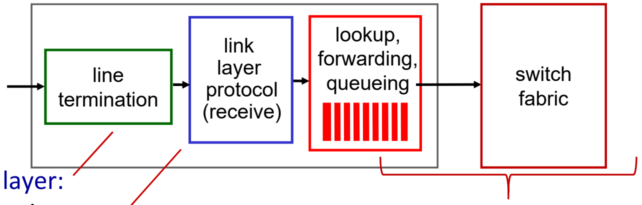
high-level view of generic router architecture:



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 address assignment

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

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

Destination address range	port	
11001000 00010111 00010000 00000000 Through 11001000 00010111 00010111 11111111	0	200.23.16~23.x
11001000 00010111 00011000 00000000 Through 11001000 00010111 00011000 11111111	1	200.23.24.x
11001000 00010111 00011001 00000000 Through 11001000 00010111 00011111 11111111	2	200.23.25~31.x
otherwise	3	others

200.23.24~31.x
except
200.23.24.x

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:

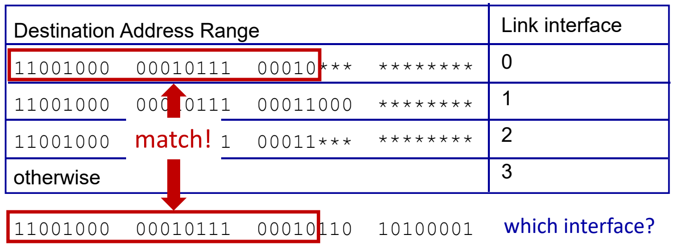
11001000 00010111 00010110 10100001 which interface?

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 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

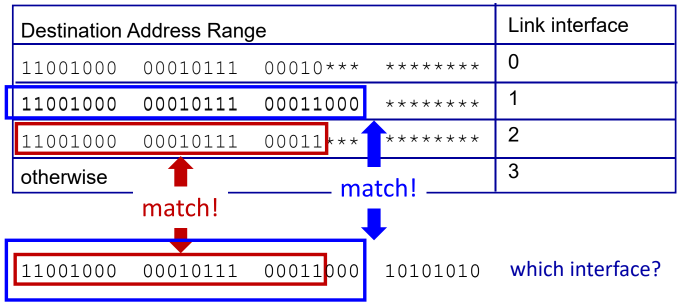
match!

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!

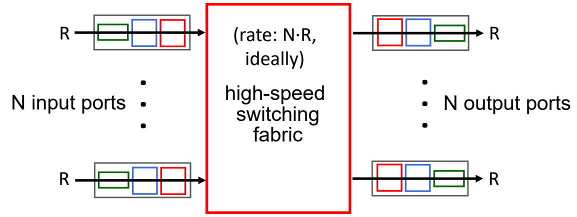
example 2: 11001000 00010111 00011000 10101010 which interface?

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

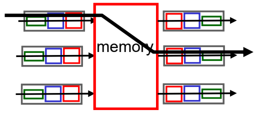
Switching fabrics

- 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



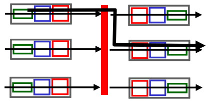
Switching fabrics

- three major types of switching fabrics:



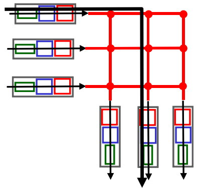
memory

slow speed, low cost



bus

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

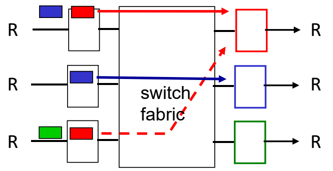


crossbar (interconnection network)

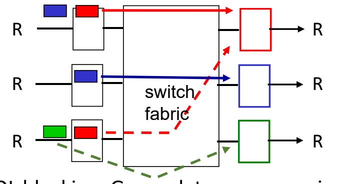
high speed (e.g., 100 Tbps), high cost

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

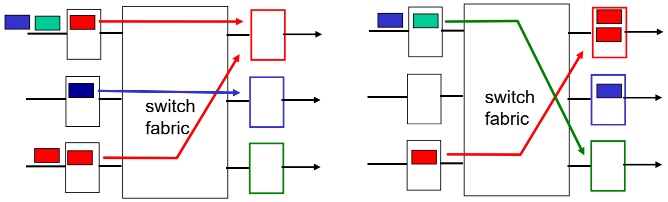


output port contention: Only one red datagram can be transferred to upper output port. Lower red one can't be forwarded at the same time.



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

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

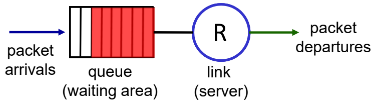
- but *too* much buffering can increase delays (particularly in home routers)
  - long RTTs: poor performance for real-time apps, sluggish TCP response
  - recall delay-based congestion control: "keep bottleneck link just full enough (busy) but no fuller"

Packet Scheduling: FCFS

packet scheduling: deciding which packet to send next on link

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

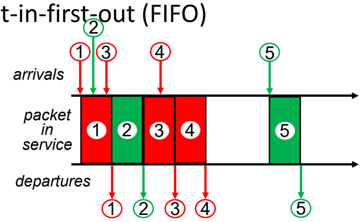
Abstraction: queue



Scheduling policies: FCFS

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

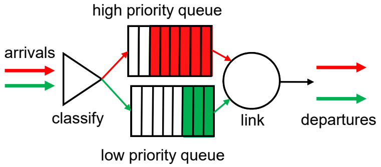
- also known as: First-in-first-out (FIFO)

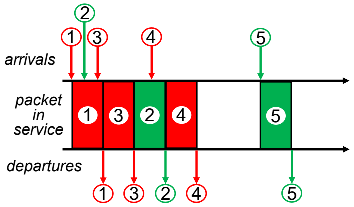


Scheduling policies: priority

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

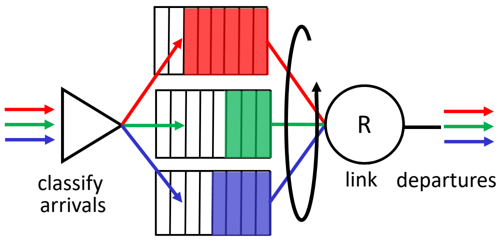




Scheduling policies: round robin

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



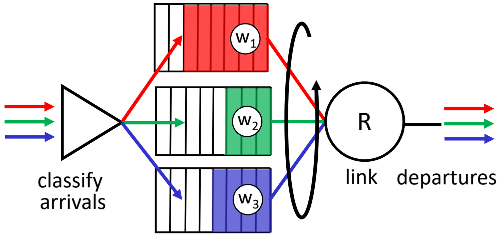
Scheduling policies: weighted fair queueing

Weighted Fair Queuing (WFQ):

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

$$\frac{w_i}{\Sigma_j w_j}$$

- it guarantees minimum bandwidth for each class


w_1
w_2
w_3
R
classify
arrivals
link
departures