Network Layer: Router

A/PROF. DUY NGO

Learning Objectives

4.1 Overview of Network layer

- **data plane**
- **control plane**

4.2 What's inside a router

Chapter 4: Network Layer

chapter goals:

understand principles behind network layer services, focusing on data plane:

- network layer service models
- forwarding versus routing
- how a router works
- generalized forwarding

instantiation, implementation in the Internet

Network Layer

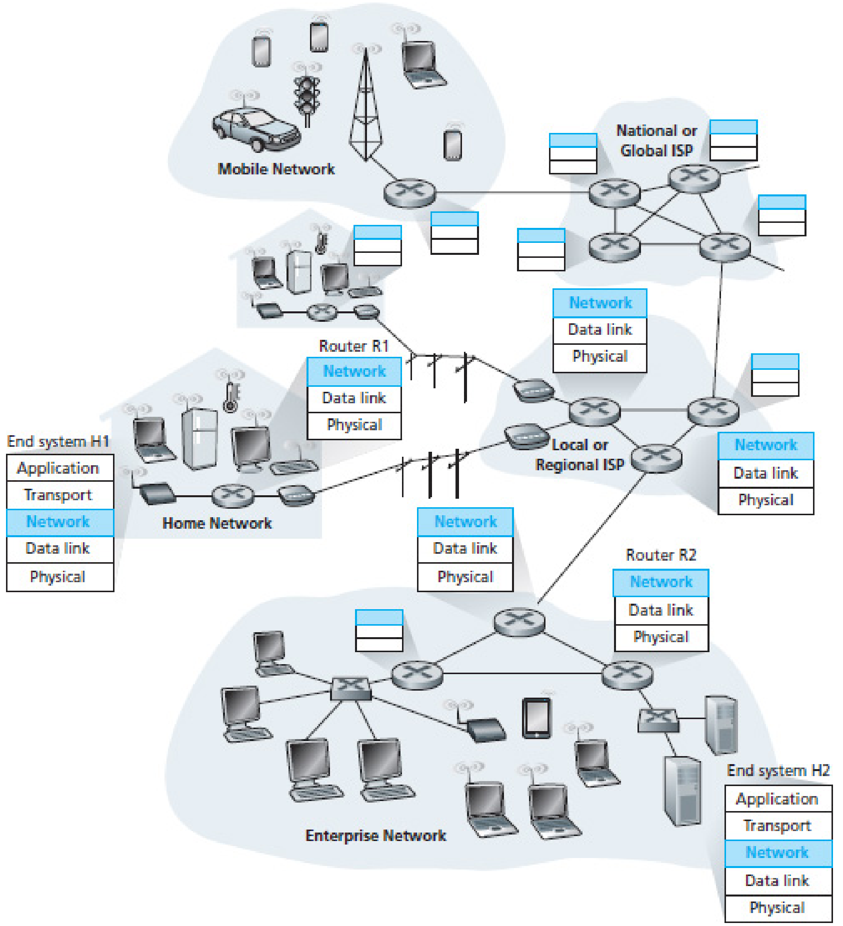
transport segment from sending to receiving host

on sending side encapsulates segments into datagrams

on receiving side, delivers segments to transport layer

network layer protocols in **every** host, router

router examines header fields in all IP datagrams passing through it



Two Key Network-Layer Functions

network-layer functions:

- **forwarding:** move packets from router’s input to appropriate router output
- **routing:** determine route 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

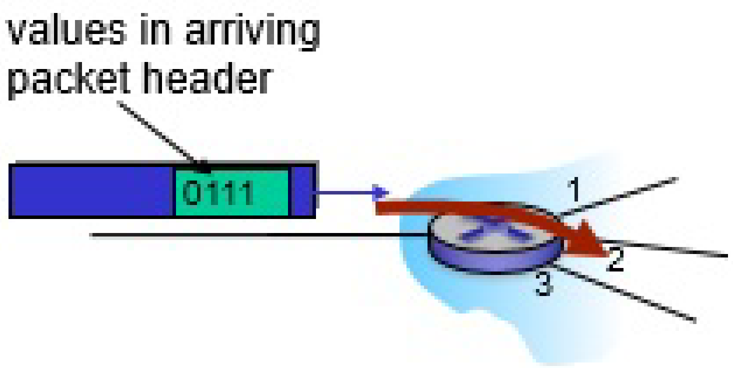
Network Layer: Data Plane, Control Plane

Data plane

local, per-router function

determines how datagram arriving on router input port is forwarded to router output port

forwarding function



Control plane

network-wide logic

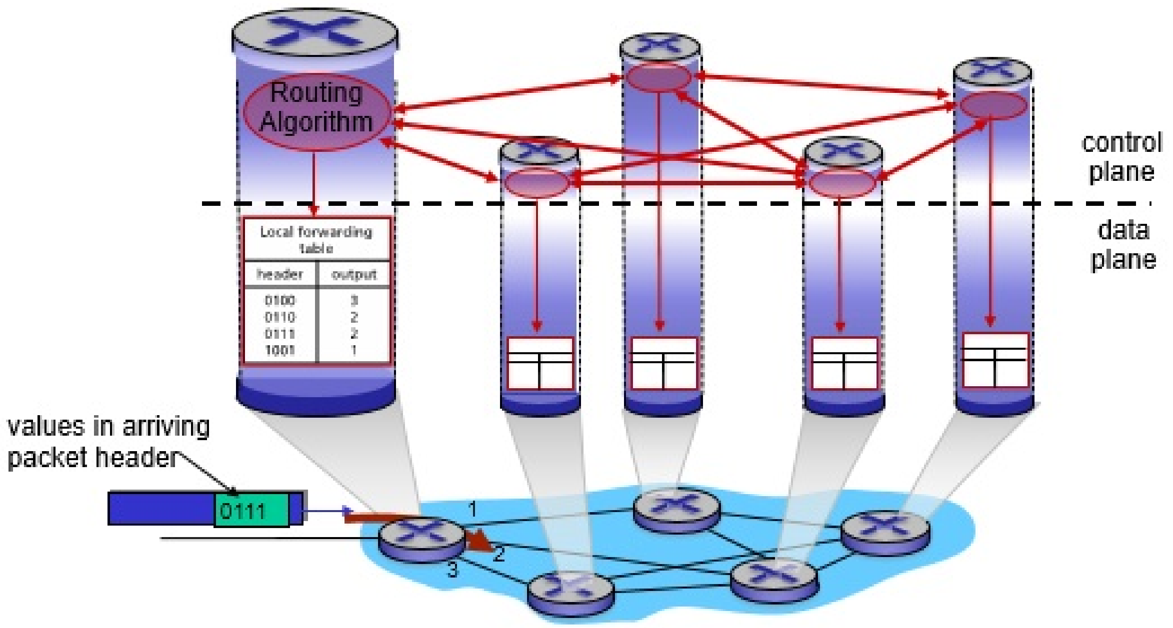
determines how datagram is routed among routers along end-end path from source host to destination host

two control-plane approaches:

- **traditional routing algorithms:** implemented in routers
- **software-defined networking (SDN):** implemented in (remote) servers

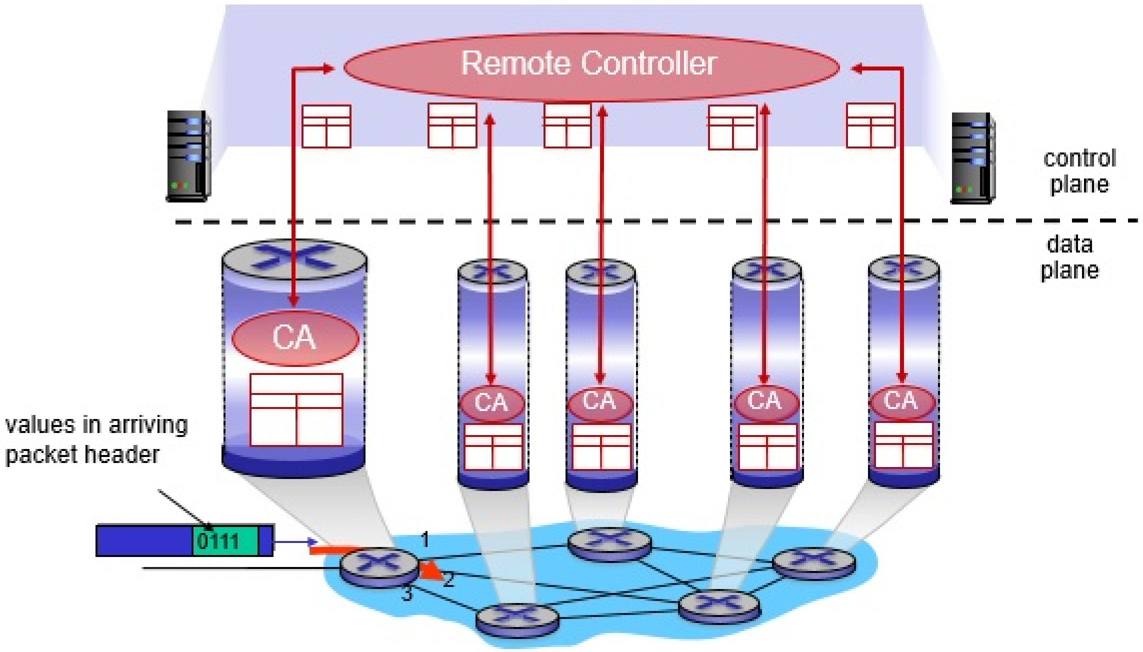
Per-Router Control Plane

Individual routing algorithm components **in each and every router** interact in the control plane



Logically Centralized Control Plane

A distinct (typically remote) controller interacts with local control agents (CAs)



Network Service Model

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

example services for individual datagrams:

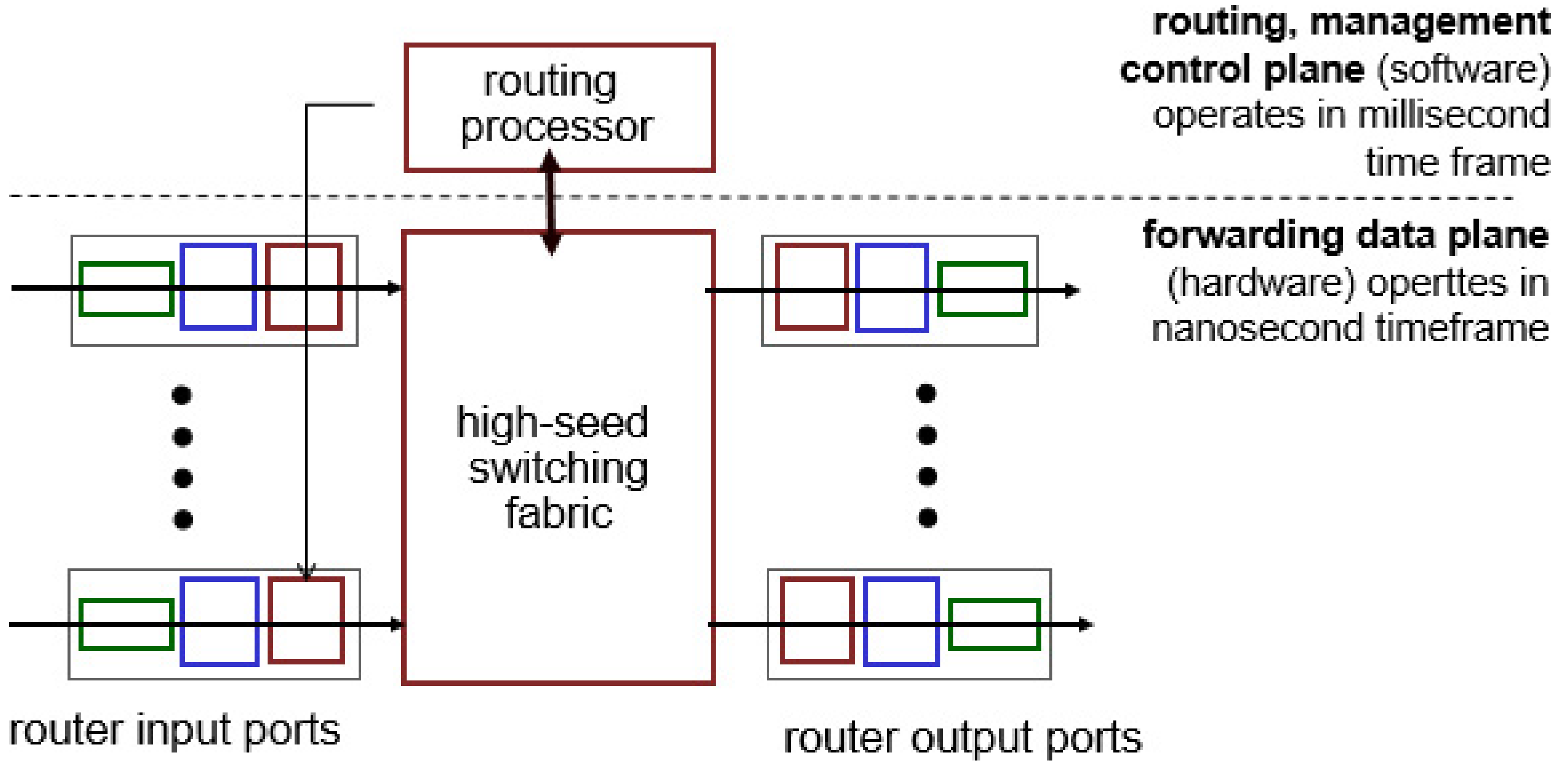
- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

example services for a flow of datagrams:

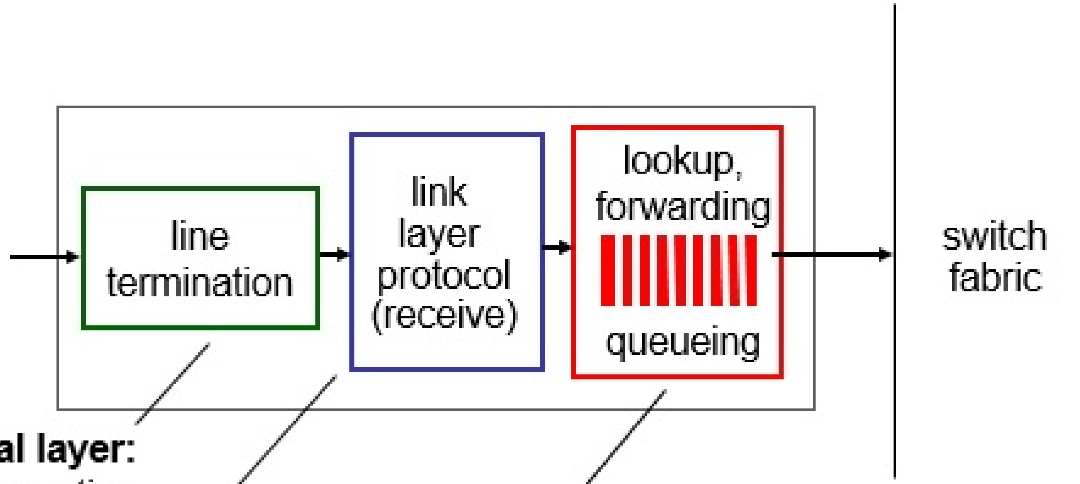
- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

Router Architecture Overview

high-level view of generic router architecture:



Input Port Functions (1 of 2)



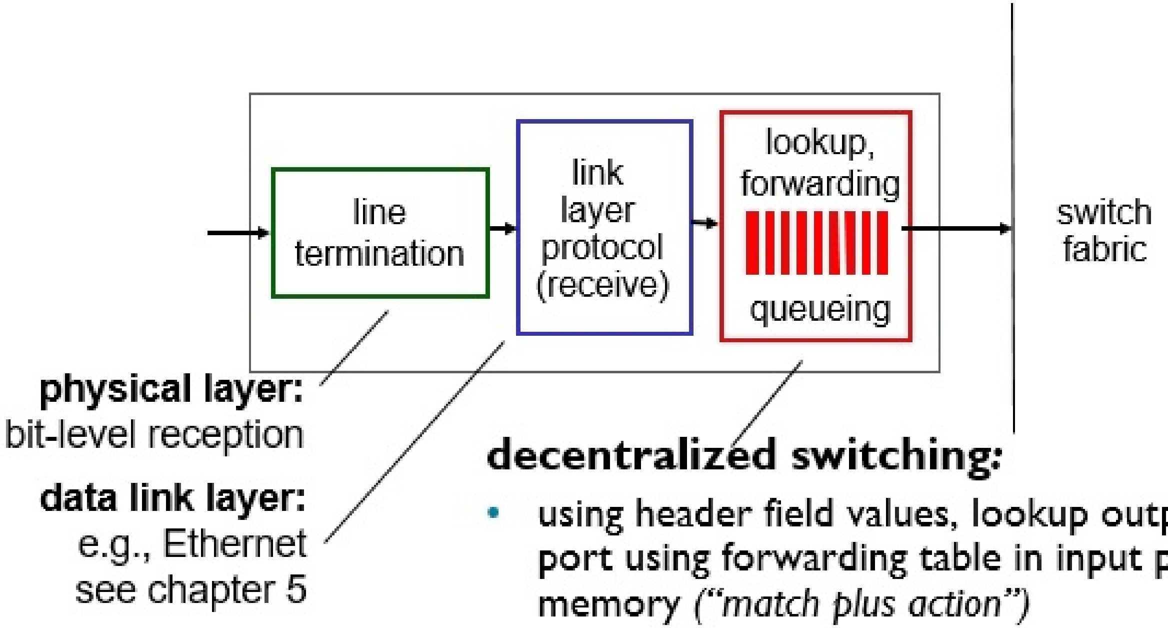
physical layer:
bit-level reception

data link layer:
e.g., Ethernet
see chapter 5

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory **("match plus action")**
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Input Port Functions (2 of 2)



decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- **destination-based forwarding:** forward based only on destination IP address (traditional)
- **generalized forwarding:** forward based on any set of header field values

Destination-Based Forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Longest Prefix Matching (1 of 2)

longest prefix matching

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:

DA: 11001000 00010111 00010110 10100001 **which interface?**

DA: 11001000 00010111 00011000 10101010 **which interface?**

Longest Prefix Matching (2 of 2)

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)

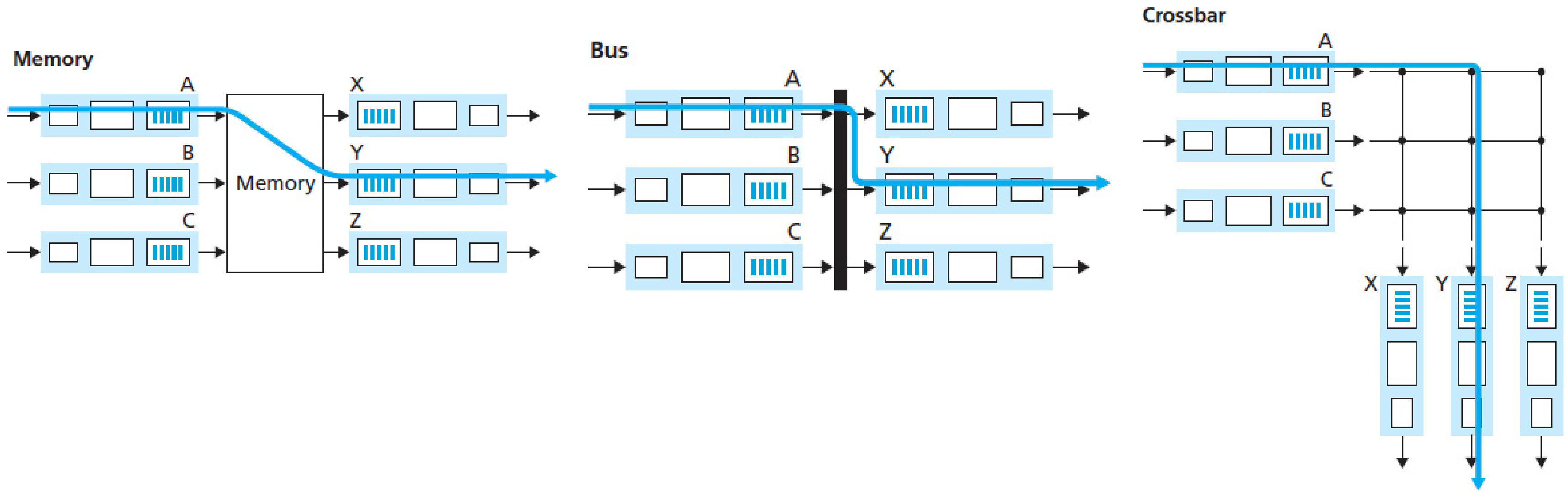
- **content addressable:** present address to TCAM : retrieve address in one clock cycle, regardless of table size
- Cisco Catalyst: can up ~1M routing table entries in TCAM

Switching Fabrics

transfer packet from input buffer to appropriate output buffer

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 types of switching fabrics



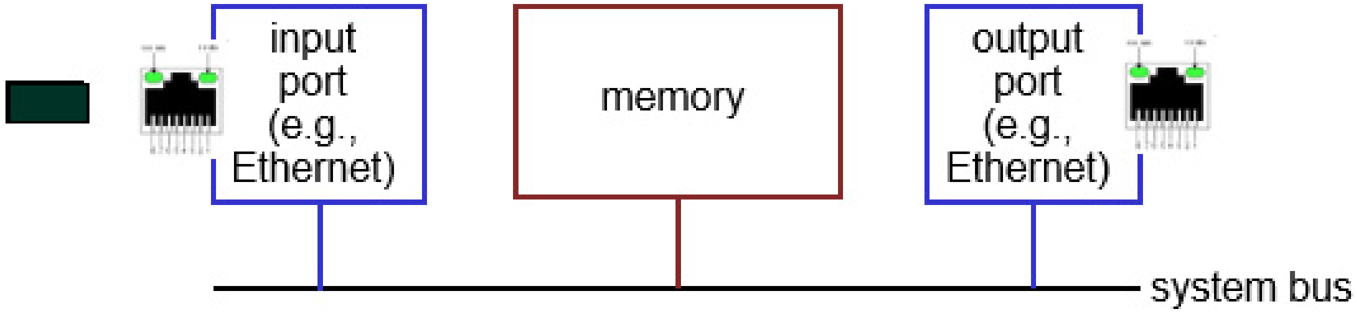
Switching via Memory

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)

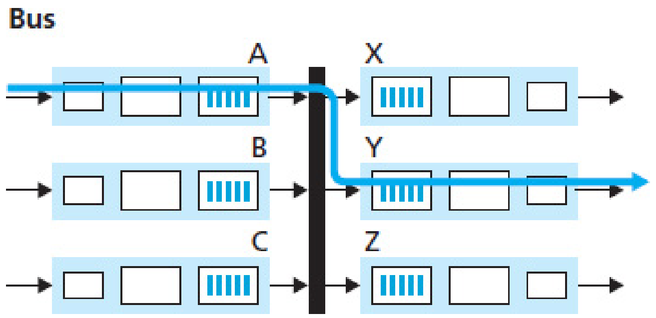


Switching via a Bus

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 and enterprise routers



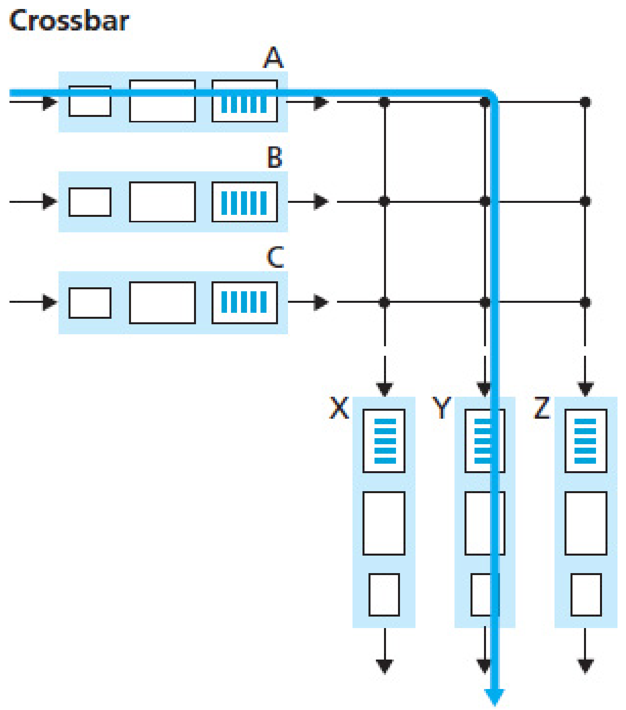
Switching via Interconnection Network

overcome bus bandwidth limitations

banyan networks, crossbar, other interconnection nets initially developed to connect processors in multiprocessor

advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.

Cisco 12000: switches 60 Gbps through the interconnection network

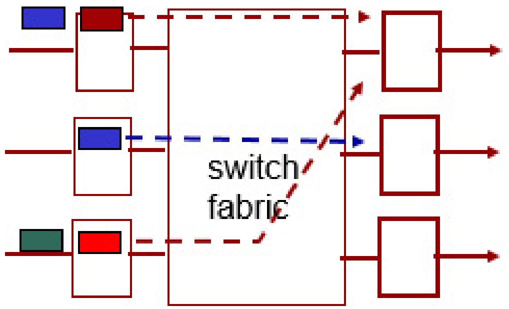


Input Port Queuing

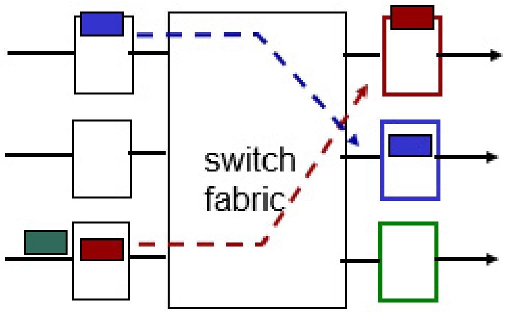
fabric slower than input ports combined -> queueing may occur at input queues

- **queueing 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

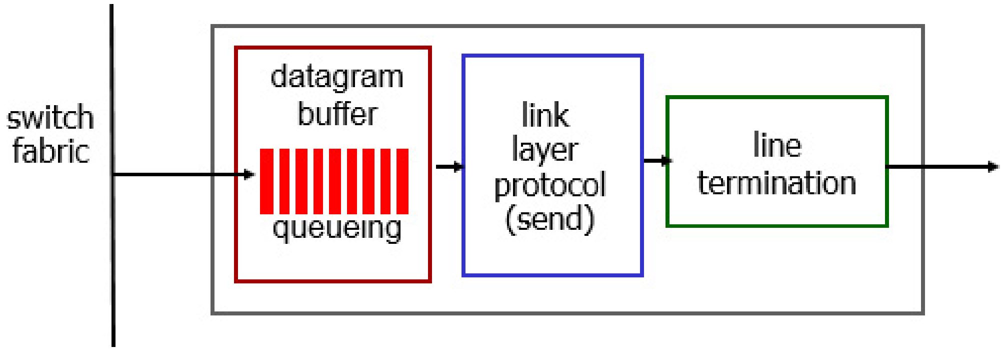


output port contention:
only one red datagram can be transferred.
lower red packet is blocked



one packet time later: green packet experiences HOL blocking

Output Ports



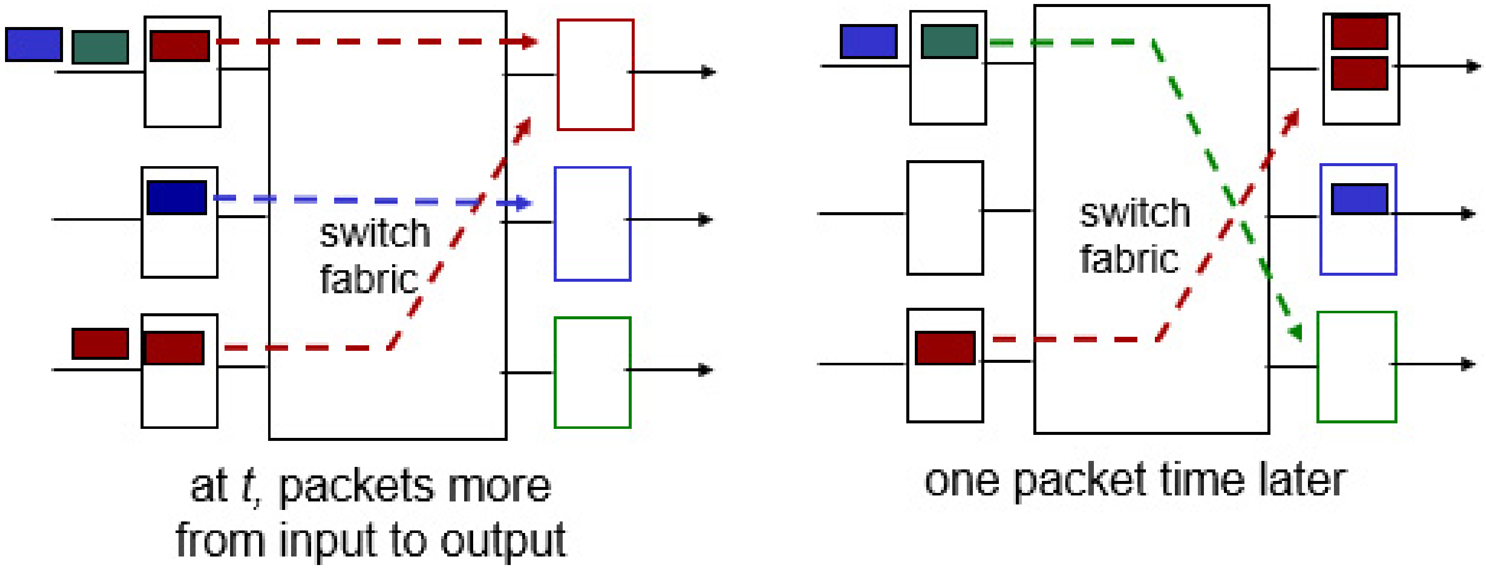
buffering required when datagrams arrive from fabric faster than the transmission rate

Datagram (packets) can be lost due to congestion, lack of buffers

scheduling discipline chooses among queued datagrams for transmission

Priority scheduling – who gets best performance, network neutrality

Output Port Queueing



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 Gpbs link: 2.5 Gbit buffer

recent recommendation: with **N** flows, buffering equal to

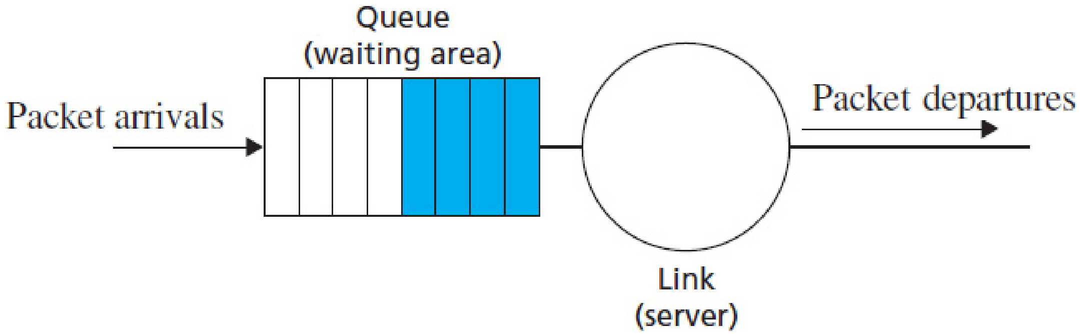
$$\frac{RTT \cdot C}{\sqrt{N}}$$

Scheduling Mechanisms

scheduling: choose next packet to send on link

FIFO (first in first out) scheduling: send in order of arrival to queue

- real-world example?
- **discard policy:** if packet arrives to full queue: who to discard?
  - **tail drop:** drop arriving packet
  - **priority:** drop/remove on priority basis
  - **random:** drop/remove randomly

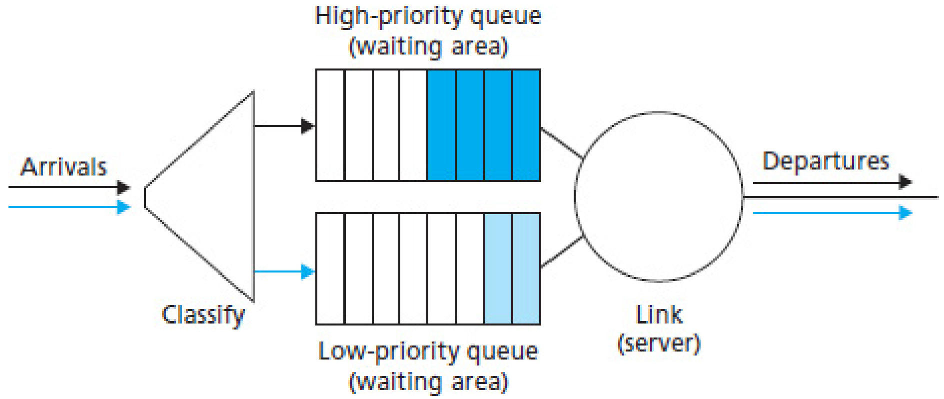


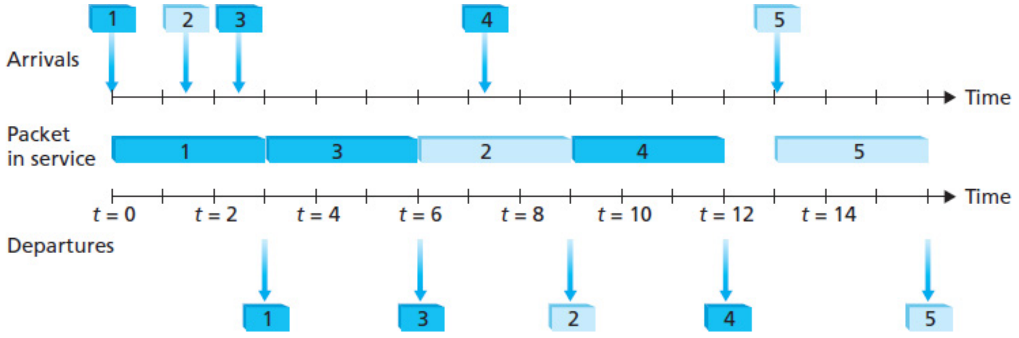
Scheduling Policies: Priority

priority scheduling: send highest priority queued packet

multiple **classes**, with different priorities

- class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc.
- real world example?





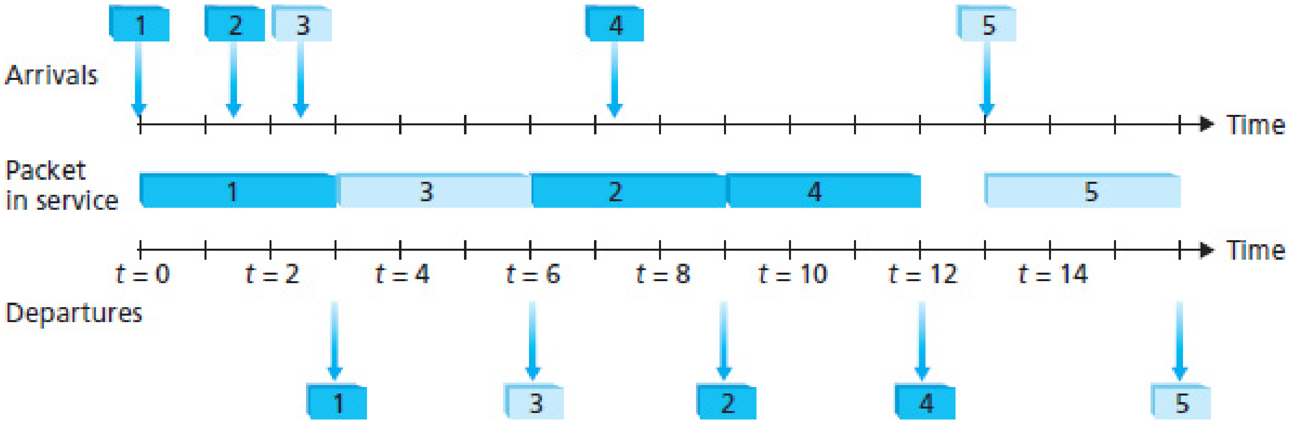
Scheduling Policies: Round Robin

Round Robin (RR) scheduling:

multiple classes

cyclically scan class queues, sending one complete packet from each class (if available)

real world example?



Scheduling Policies: Weighted Fair Queuing

Weighted Fair Queuing (WFQ):

generalized Round Robin

each class gets weighted amount of service in each cycle

real-world example?