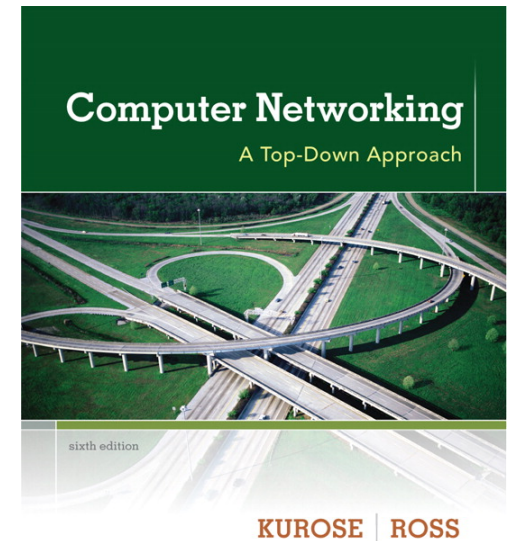


Chapter 4

Network Layer

Reti degli Elaboratori
Canale ALProf.ssa Chiara
Petrioli
a.a. 2019/2020

We thank for the support material Prof. Kurose-Ross
All material copyright 1996-2012
© J.F Kurose and K.W. Ross, All Rights Reserved



*Computer
Networking: A Top
Down Approach*
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Chapter 4: network layer

chapter goals:

- ❖ understand principles behind network layer services:
 - network layer service models
 - forwarding versus routing
 - how a router works
 - routing (path selection)
 - broadcast, multicast
- ❖ instantiation, implementation in the Internet

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

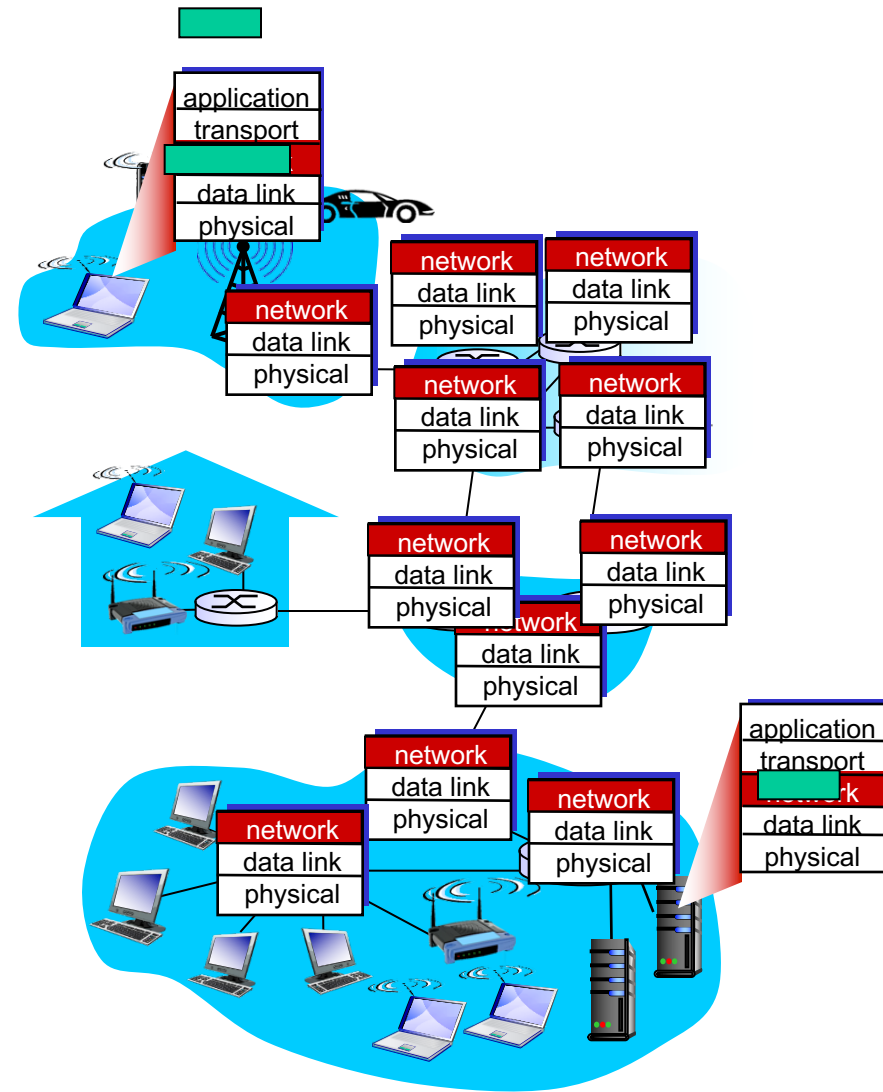
4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

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

- ❖ *forwarding*: move packets from router's input to appropriate router output

- ❖ *routing*: determine route taken by packets from source to dest.

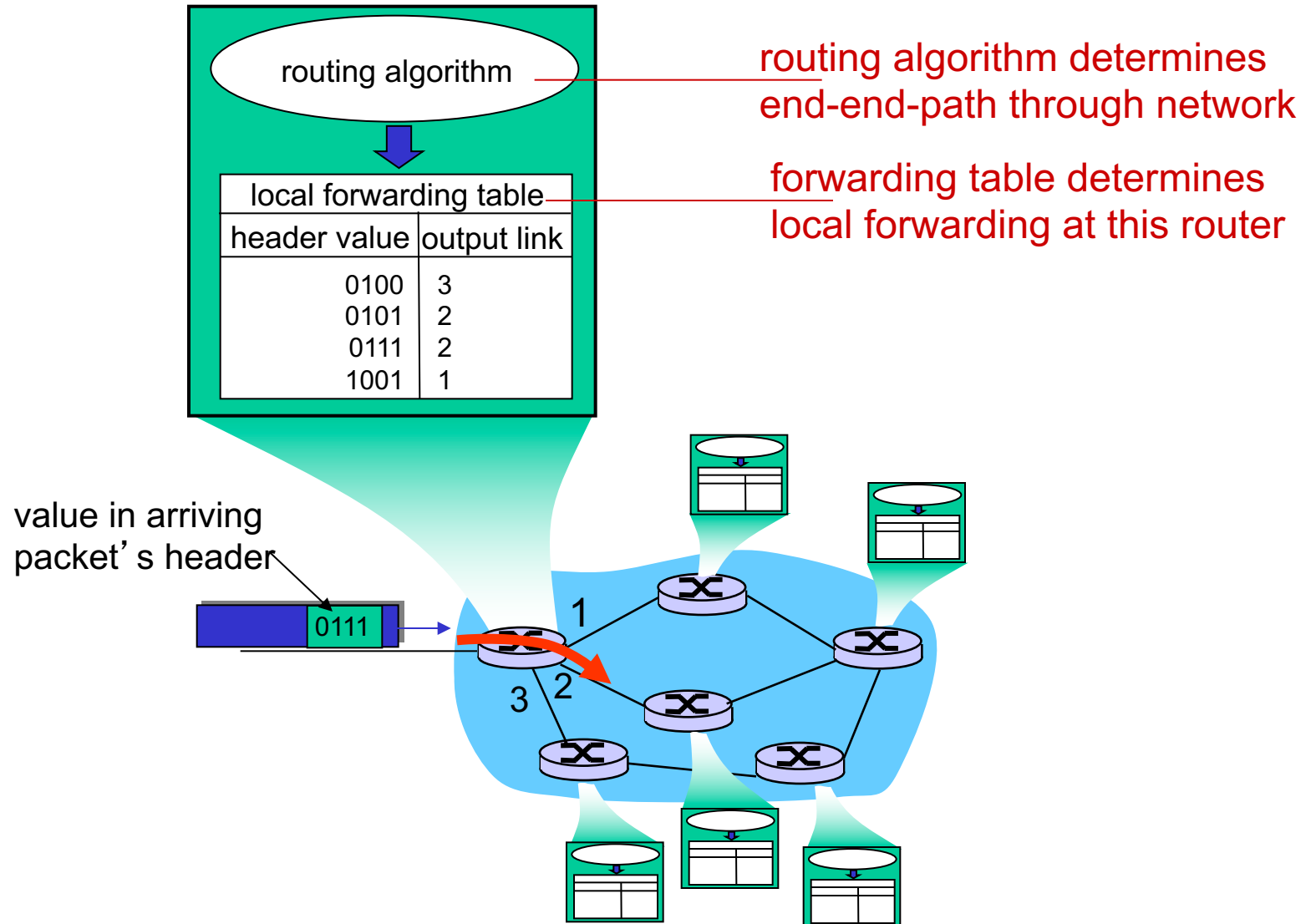
 - *routing algorithms*

analogy:

- ❖ *routing*: process of planning trip from source to dest

- ❖ *forwarding*: process of getting through single interchange

Interplay between routing and forwarding



Connection setup

- ❖ 3rd important function in *some* network architectures:
 - ATM, frame relay, X.25
- ❖ before datagrams flow, two end hosts *and* intervening routers establish virtual connection
 - routers get involved
- ❖ network vs transport layer connection service:
 - *network*: between two hosts (may also involve intervening routers in case of VCs)
 - *transport*: between two processes

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

Network layer service models:

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Connection, connection-less service

- ❖ *datagram* network provides network-layer *connectionless* service
- ❖ *virtual-circuit* network provides network-layer *connection* service
- ❖ analogous to TCP/UDP connection-oriented / connectionless transport-layer services, but:
 - *service*: host-to-host
 - *no choice*: network provides one or the other
 - *implementation*: in network core

Virtual circuits

“source-to-dest path behaves much like telephone circuit”

- performance-wise
- network actions along source-to-dest path

- ❖ call setup, teardown for each call *before* data can flow
- ❖ each packet carries VC identifier (not destination host address)
- ❖ every router on source-dest path maintains “state” for each passing connection
- ❖ link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

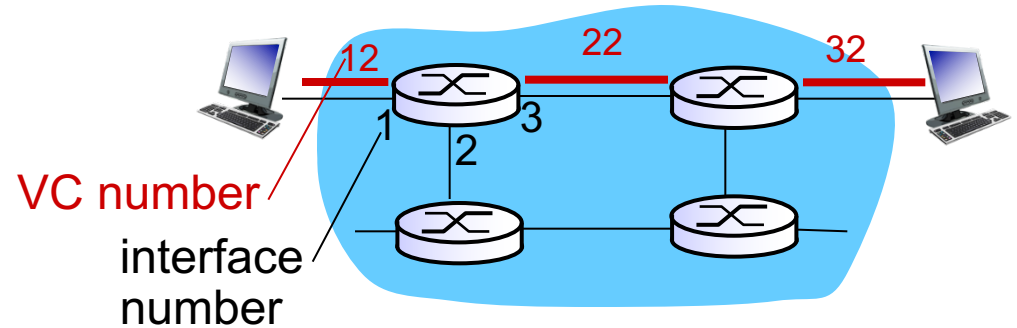
VC implementation

a VC consists of:

1. *path* from source to destination
 2. *VC numbers*, one number for each link along path
 3. *entries in forwarding tables* in routers along path
- ❖ packet belonging to VC carries VC number (rather than dest address)
 - ❖ VC number can be changed on each link.
 - new VC number comes from forwarding table

VC forwarding table

*forwarding table in
northwest router:*

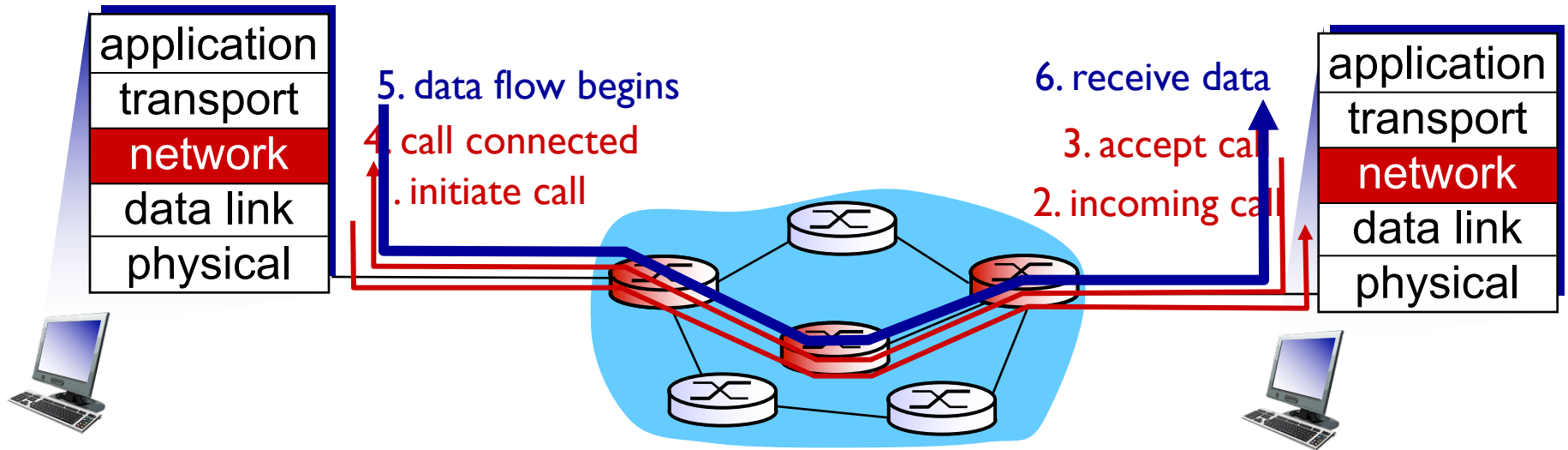


Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

VC routers maintain connection state information!

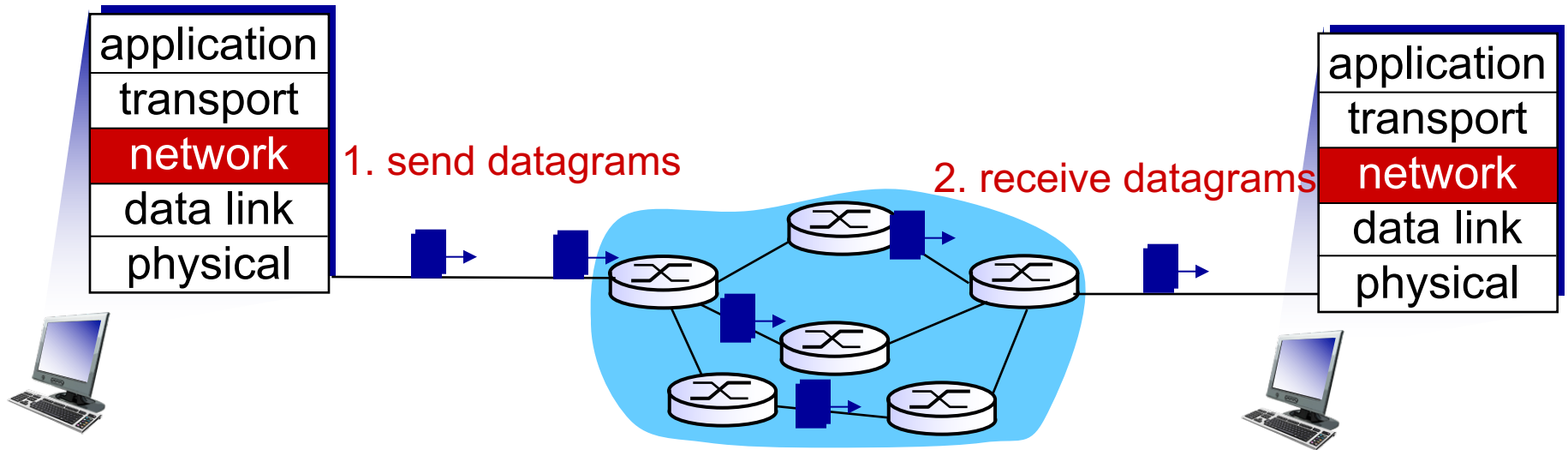
Virtual circuits: signaling protocols

- ❖ used to setup, maintain teardown VC
- ❖ used in ATM, frame-relay, X.25
- ❖ not used in today's Internet

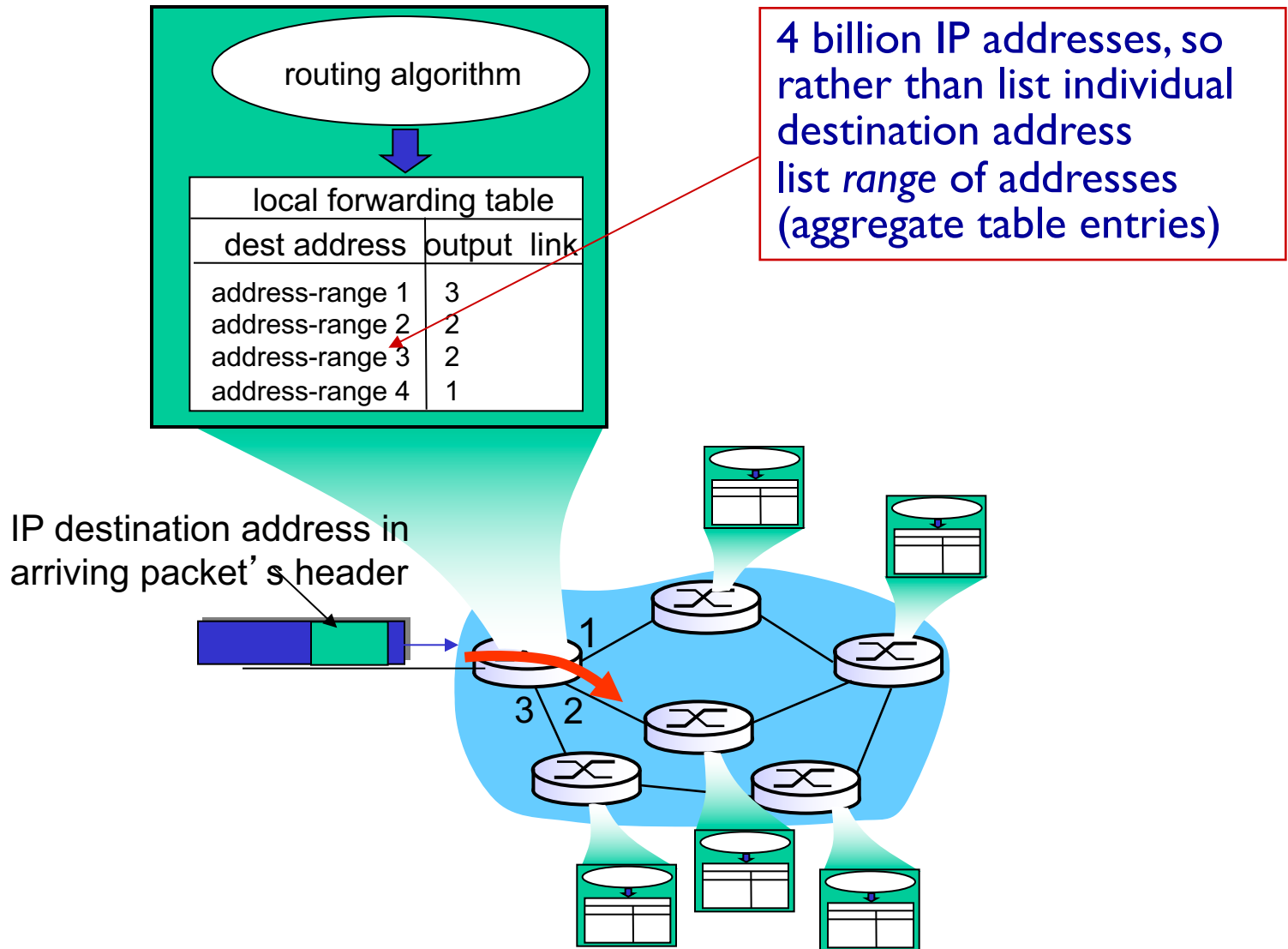


Datagram networks

- ❖ no call setup at network layer
- ❖ routers: no state about end-to-end connections
 - no network-level concept of “connection”
- ❖ packets forwarded using destination host address



Datagram forwarding table



Datagram 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?

Datagram or VC network: why?

Internet (datagram)

- ❖ data exchange among computers
 - “elastic” service, no strict timing req.
- ❖ many link types
 - different characteristics
 - uniform service difficult
- ❖ “smart” end systems (computers)
 - can adapt, perform control, error recovery
 - ***simple inside network, complexity at “edge”***

ATM (VC)

- ❖ evolved from telephony
- ❖ human conversation:
 - strict timing, reliability requirements
 - need for guaranteed service
- ❖ “dumb” end systems
 - telephones
 - ***complexity inside network***

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

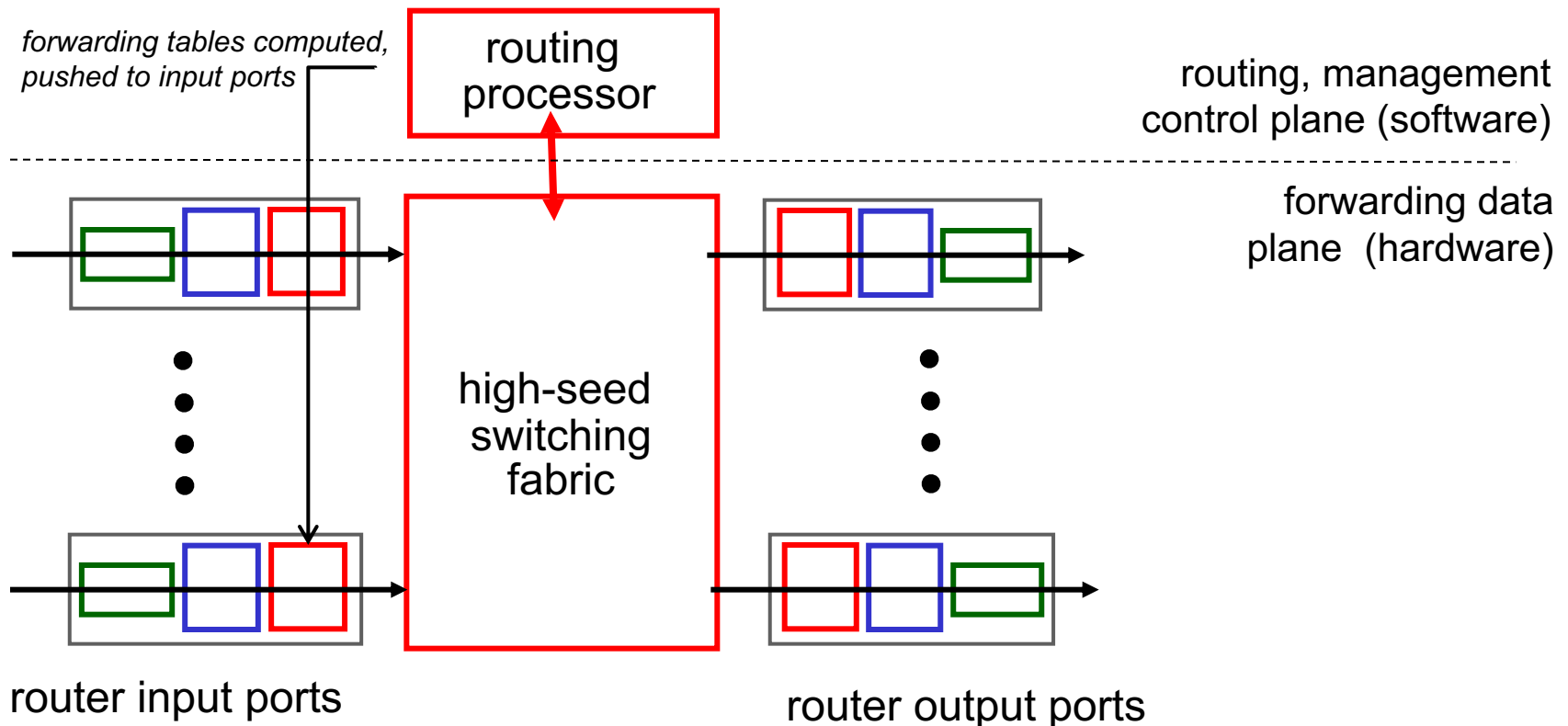
- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

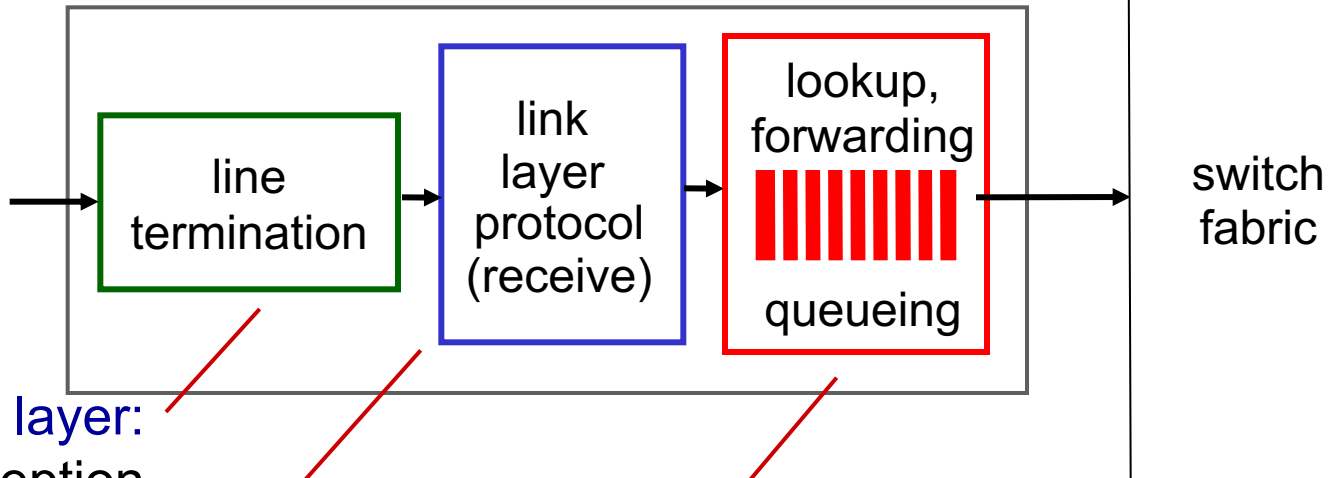
Router architecture overview

two key router functions:

- ❖ run routing algorithms/protocol (RIP, OSPF, BGP)
- ❖ *forwarding* datagrams from incoming to outgoing link



Input port functions



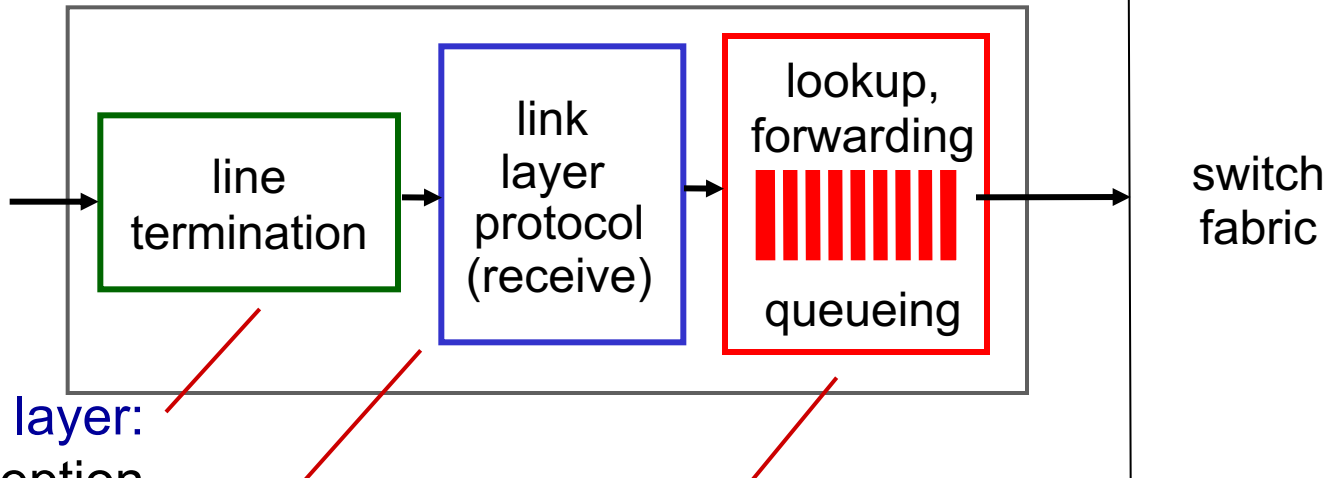
physical layer:
bit-level reception

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

decentralized switching:

- ❖ given datagram dest., 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



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*”)
- ❖ **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

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

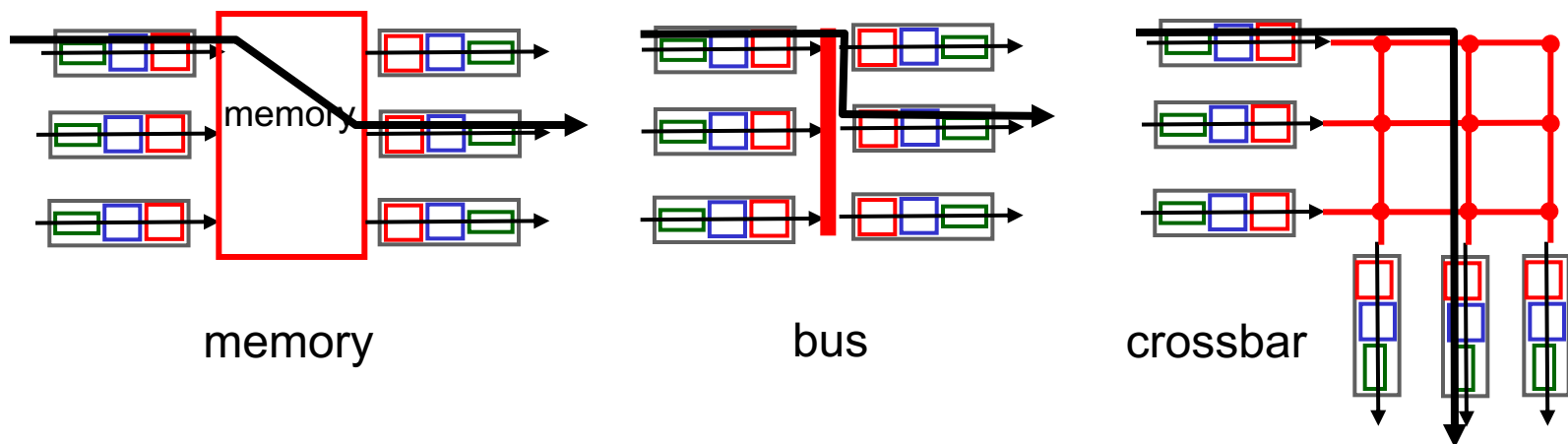
which interface?

Longest prefix matching

- ❖ 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

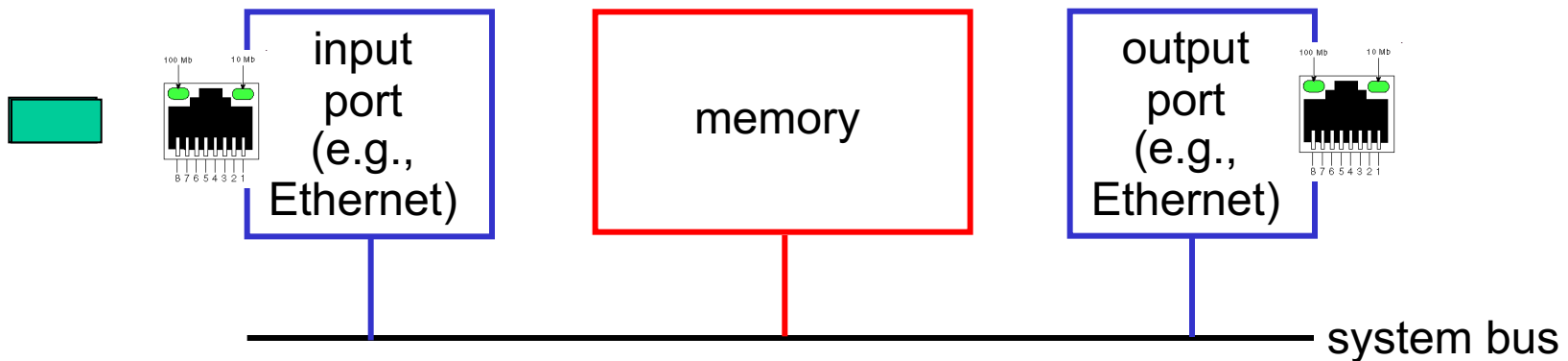
- ❖ transfer packet from input buffer to appropriate output buffer
- ❖ switching rate: rate at which packets can be transfer from inputs to outputs
 - often measured as multiple of input/output line rate
 - N inputs: switching rate N times line rate desirable
- ❖ three types of switching fabrics



Switching via memory

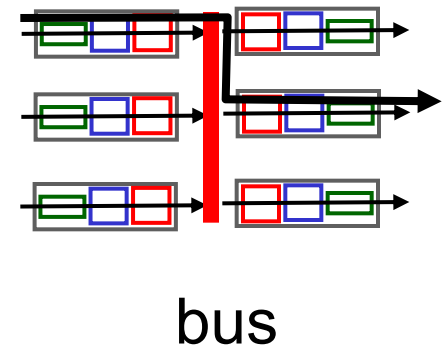
first generation routers:

- ❖ 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)



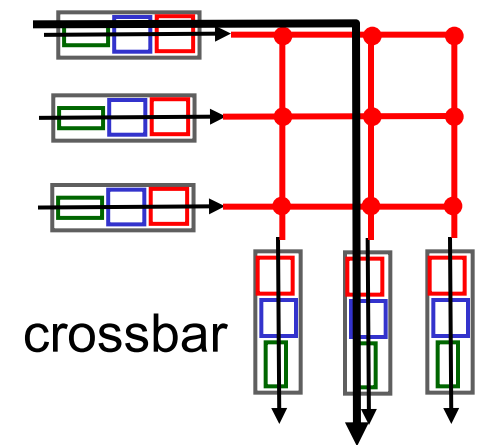
Switching via a bus

- ❖ 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



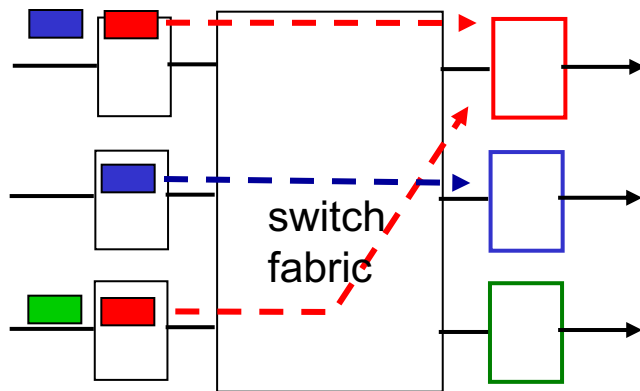
Switching via interconnection network

- ❖ 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 I2000: switches 60 Gbps through the interconnection network

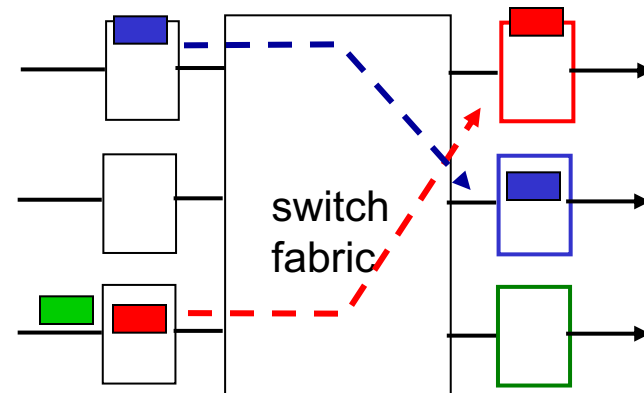


Input port queuing

- ❖ 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

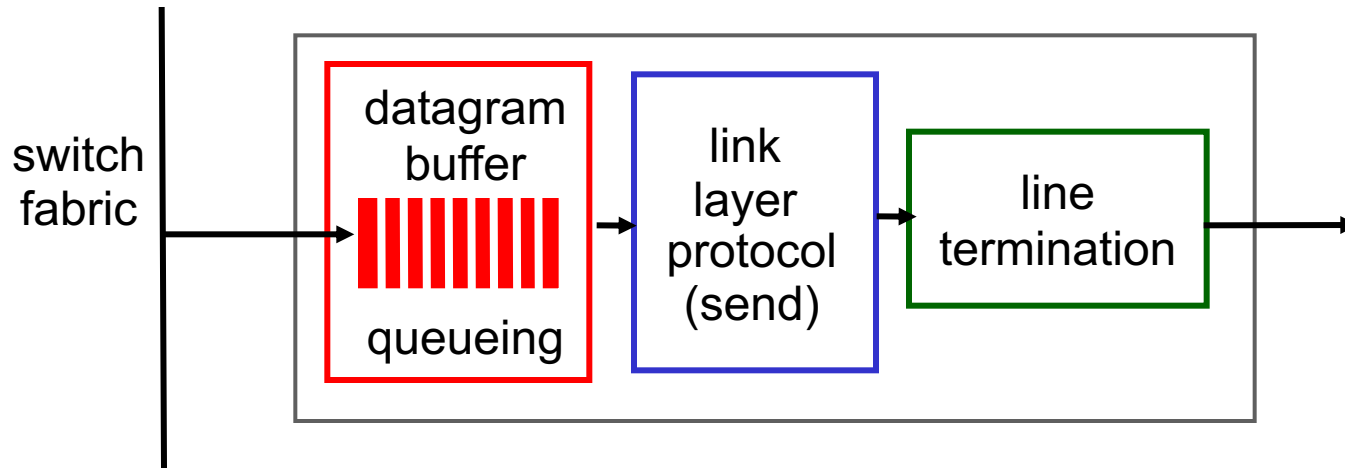


output port contention:
only one red datagram can be
transferred.
lower red packet is blocked



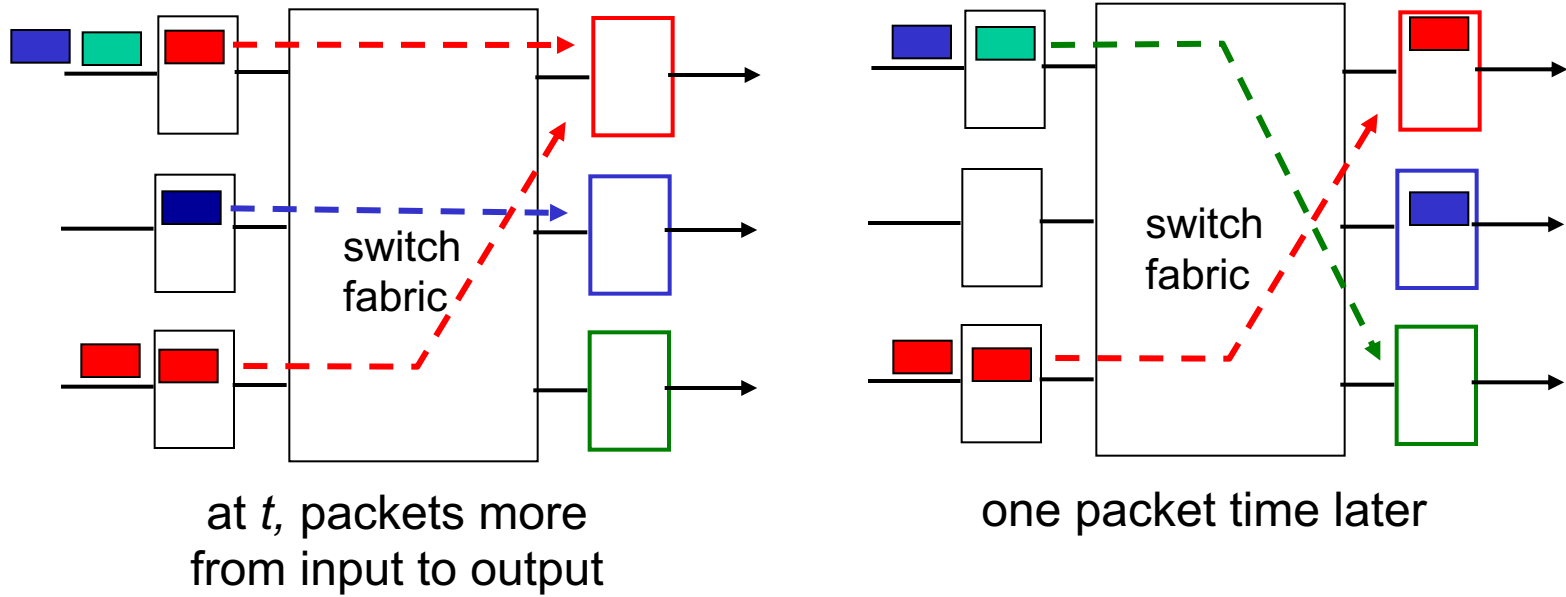
one packet time later:
green packet
experiences HOL
blocking

Output ports



- ❖ *buffering* required when datagrams arrive from fabric faster than the transmission rate
- ❖ *scheduling discipline* chooses among queued datagrams for transmission

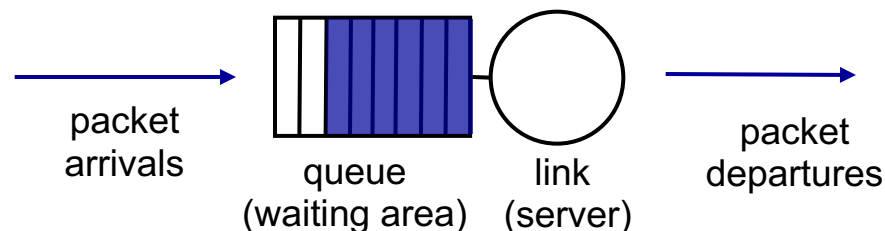
Output port queueing



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

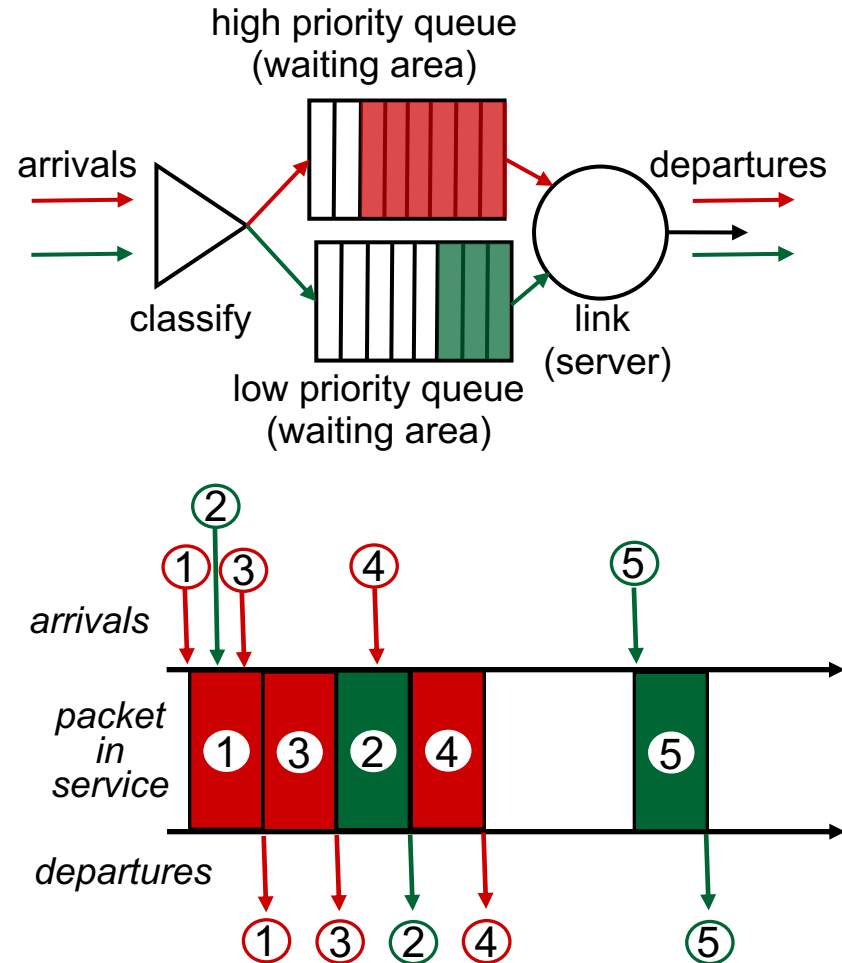
Scheduling mechanisms

- ❖ *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

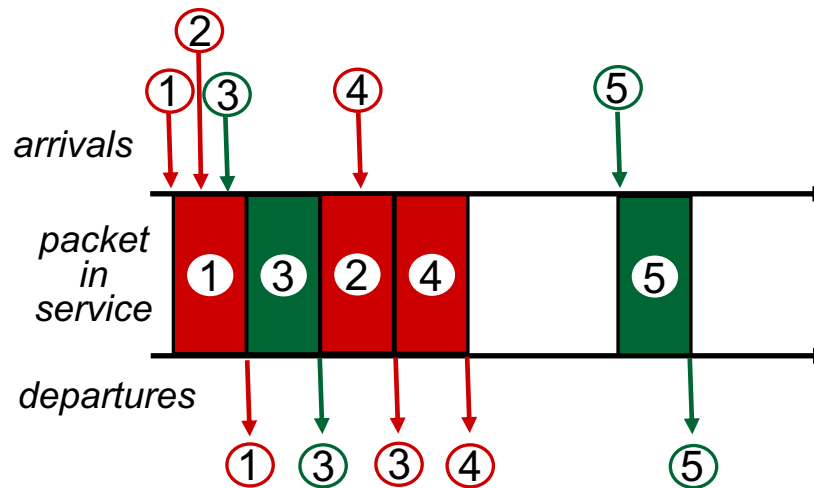
- priority scheduling*: send highest priority queued packet
- ❖ multiple *classes*, with different priorities
 - class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc.
 - real world example?



Scheduling policies: still more

Round Robin (RR) scheduling:

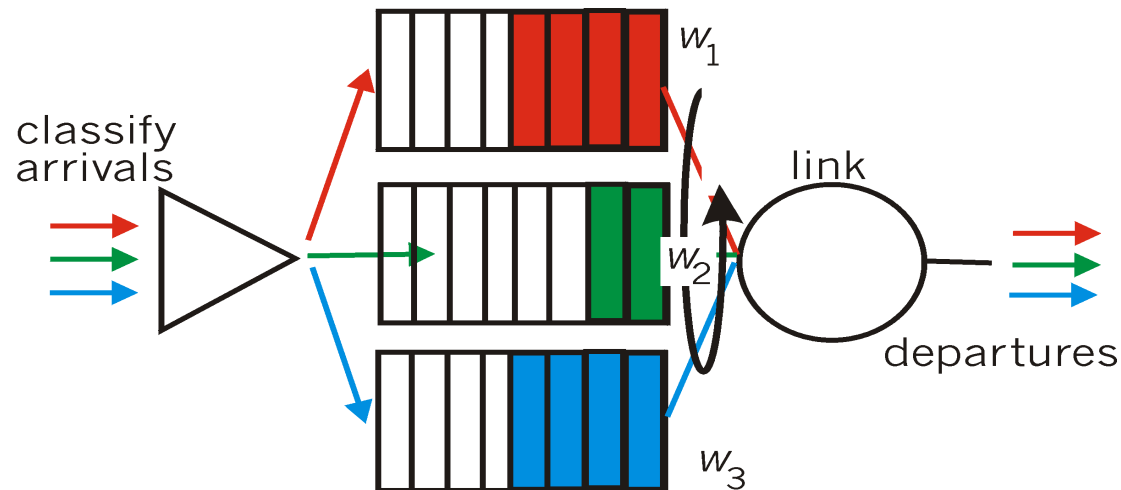
- ❖ multiple classes
- ❖ cyclically scan class queues, sending one complete packet from each class (if available)
- ❖ real world example?



Scheduling policies: still more

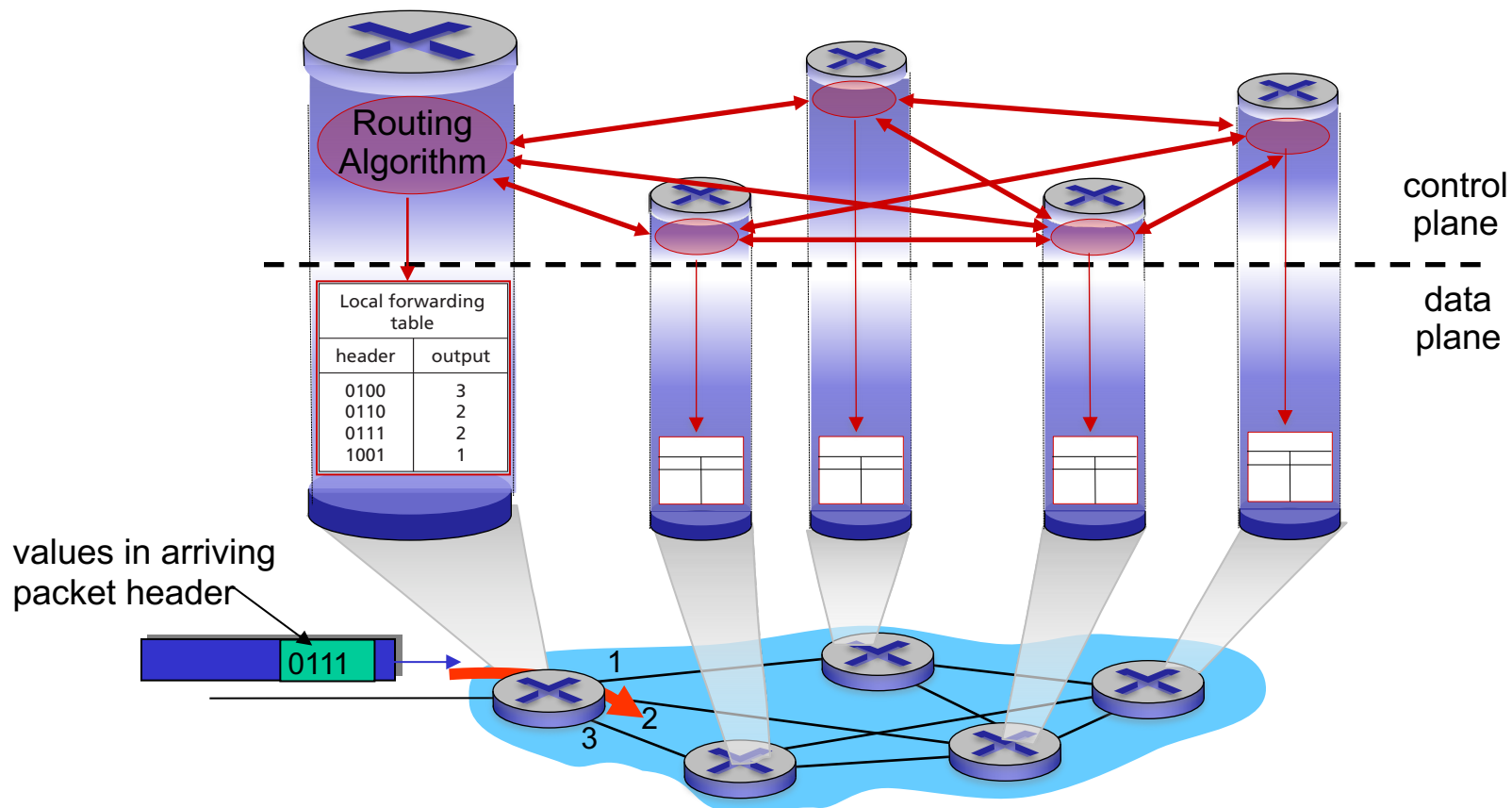
Weighted Fair Queuing (WFQ):

- ❖ generalized Round Robin
- ❖ each class gets weighted amount of service in each cycle
- ❖ real-world example?



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)

