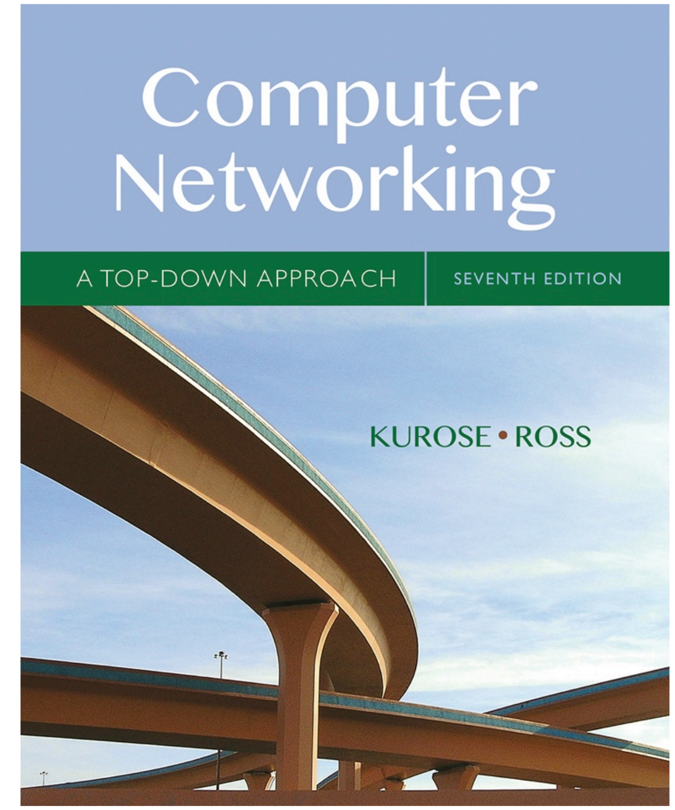
CN-Basic
L30

Router Architecture

Dr. Ram P Rustagi
rprustagi@ksit.edu.in
http://www.rprustagi.com
https://www.youtube.com/rprustagi

Chapter 4
Network Layer



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

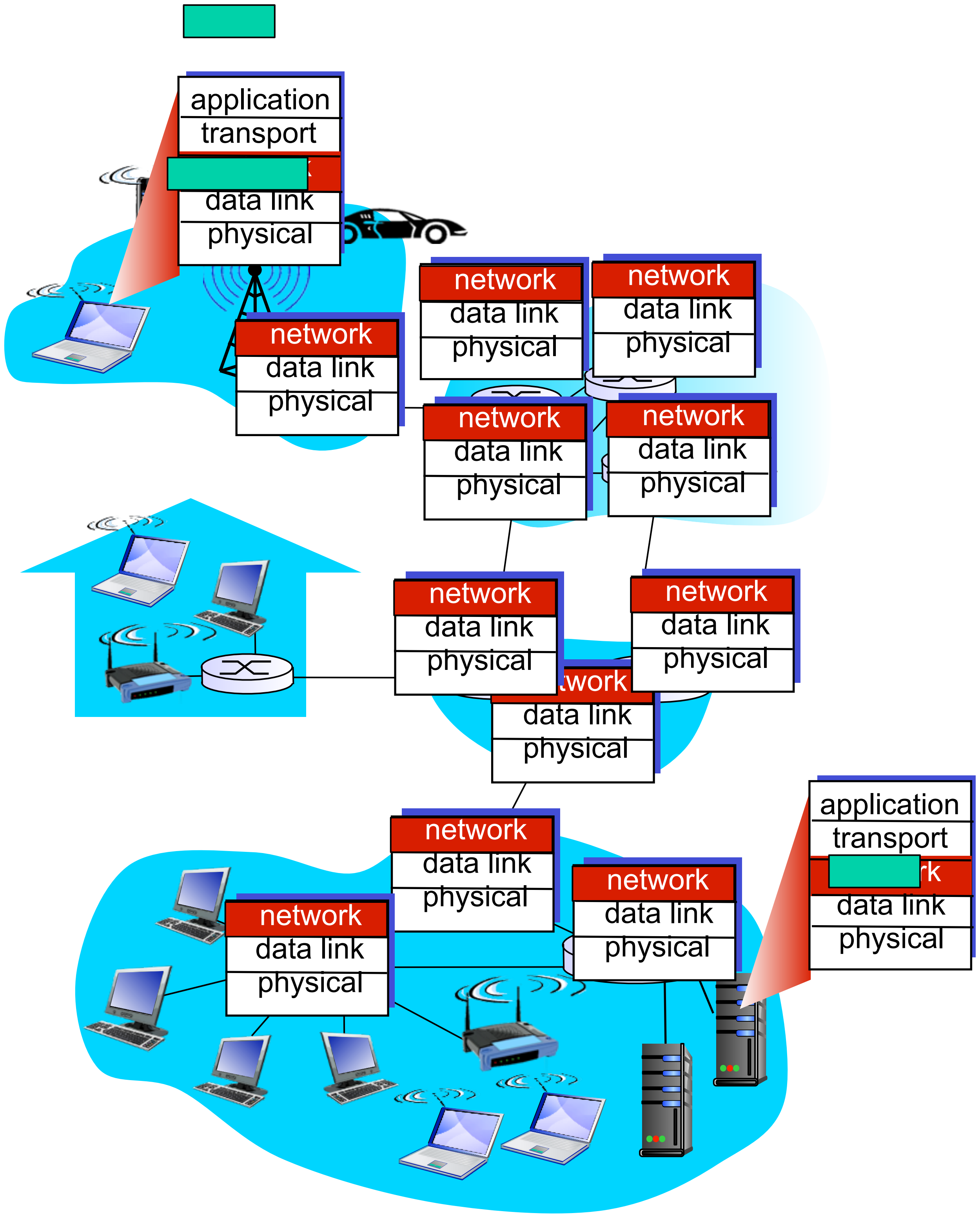
Network Layer

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
  - Either forwards it or drops



Two key network-layer functions

- *Forwarding & Routing*
- *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

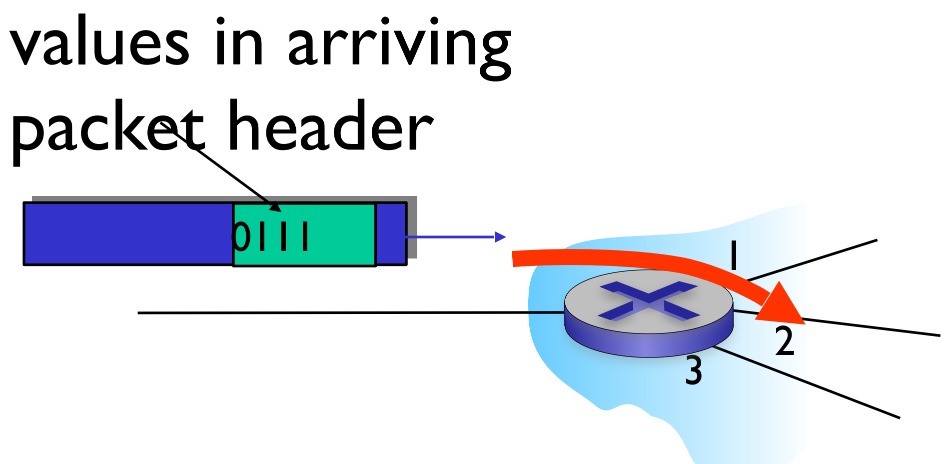
- Link layer Switches or layer-2 switching
- Network layer switches, called routers

Q: Is there another n/w layer function?

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

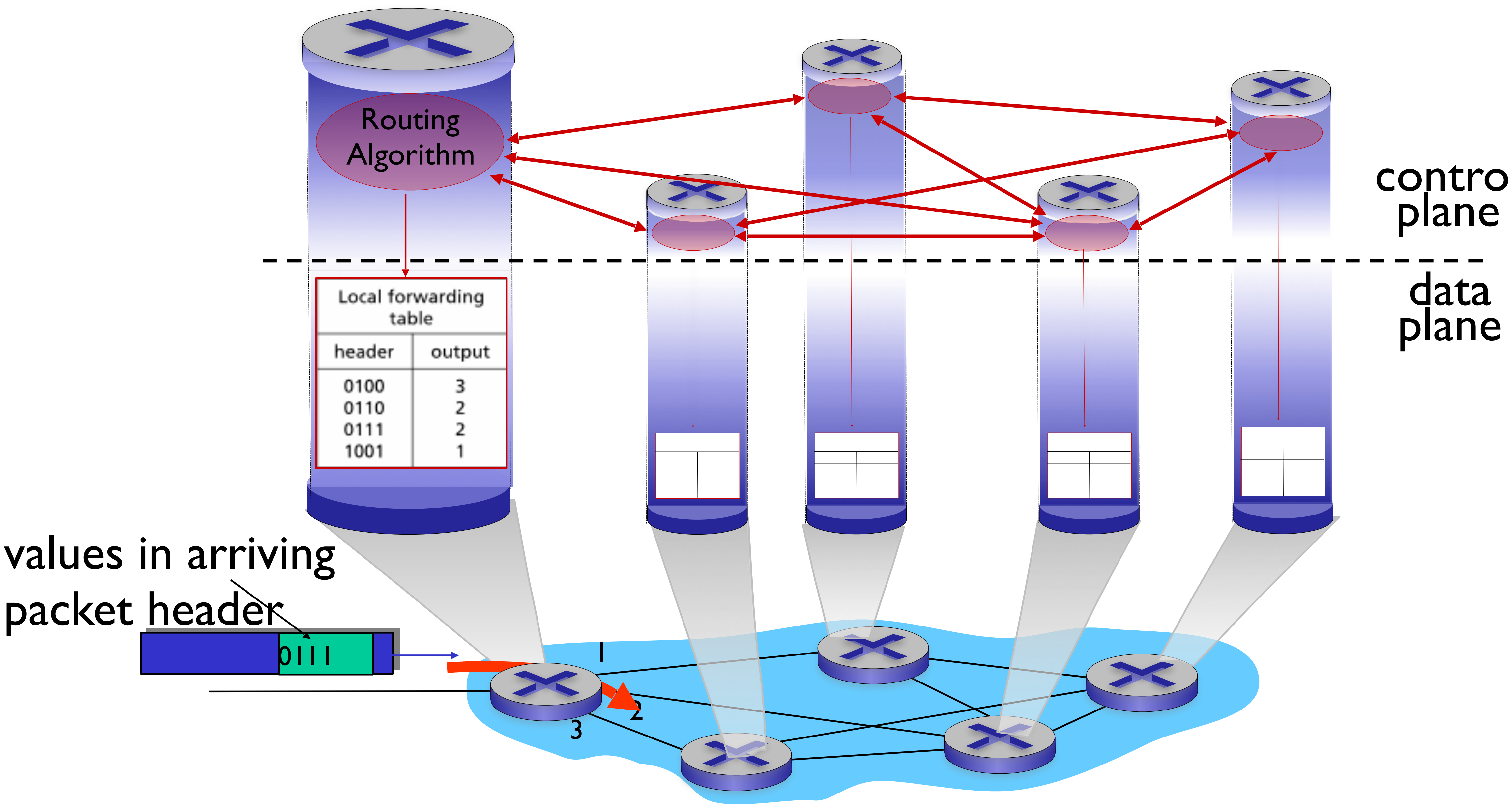


Control plane

- 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

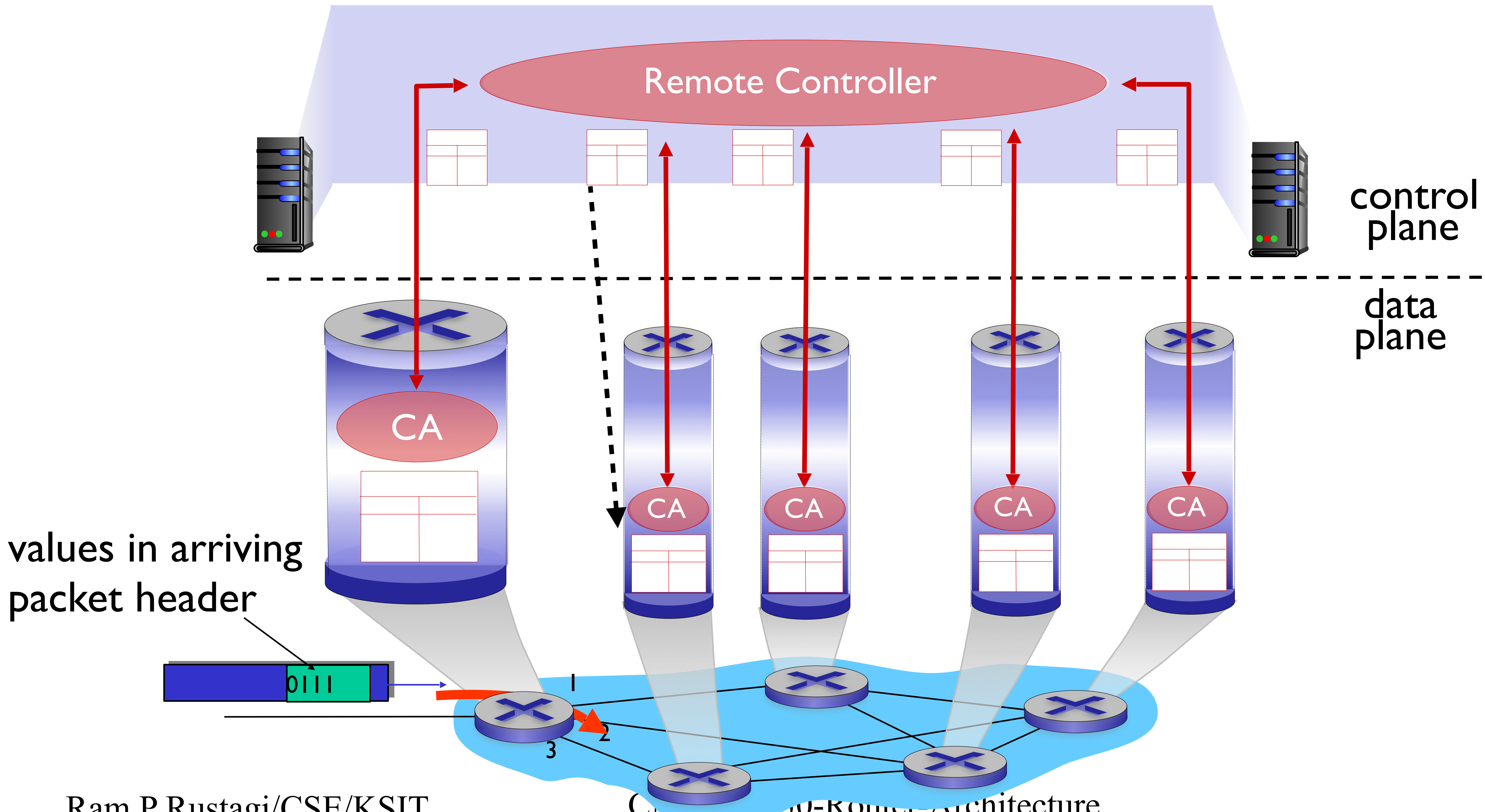
Per-router control plane

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: Service model for college?
  - What does it provide to prospective students?

    Grades?

    Jobs?

    Joining delay/time (jobs)?

    Class size?

    Teaching quality?

    Attendance?

    Fee benefits?

    Migration to other disciplines/colleges?

Network service model

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

- Can transport layer rely on n/w layer?
  - Will the packets be in order?
  - Will the time gap between two pkts be maintained?
  - Will network provide any congestion information?
  - Will network provide any time gurantees?
  - Will network provide any BW guarantees?

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 **(jitter)**
- Security services?

Q: Is there only one service model ?
Q: What does n/w layer of internet provides?

Destination-based forwarding

forwarding table

Destination Address Range	Link I/f
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 I/f
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00010110 10100001 which interface?

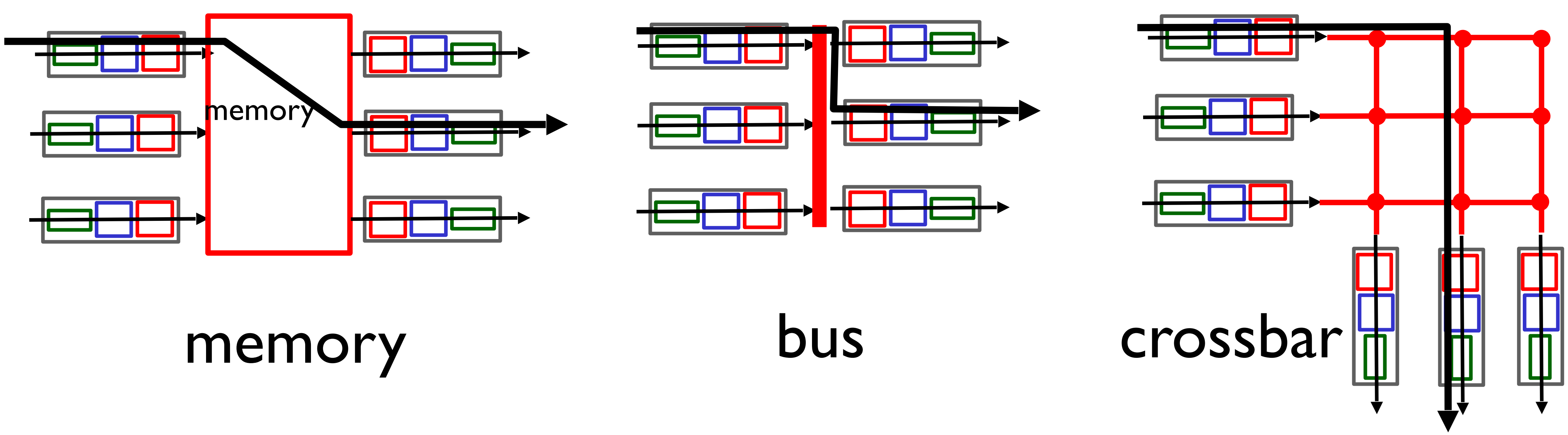
DA: 11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

- 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

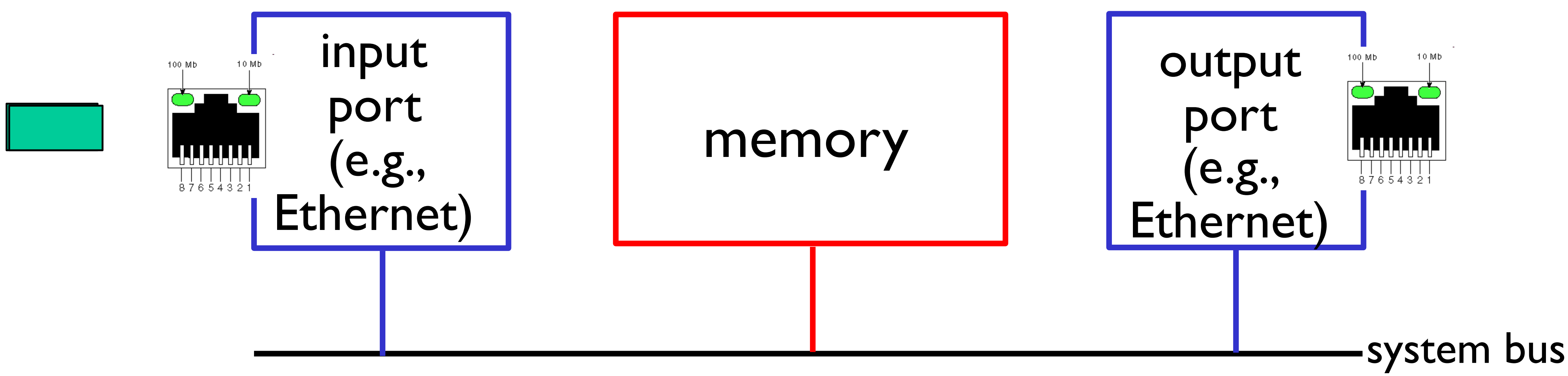
- Transfer packet from input buffer to appropriate o/p buf
- Switching rate: rate at which packets can be transferred from inputs to outputs
  - Often measured as multiple of input/output line rate
  - N inputs lines each of rate R
    - Desirable switching rate N times R
- Three types of switching fabrics



Switching via memory

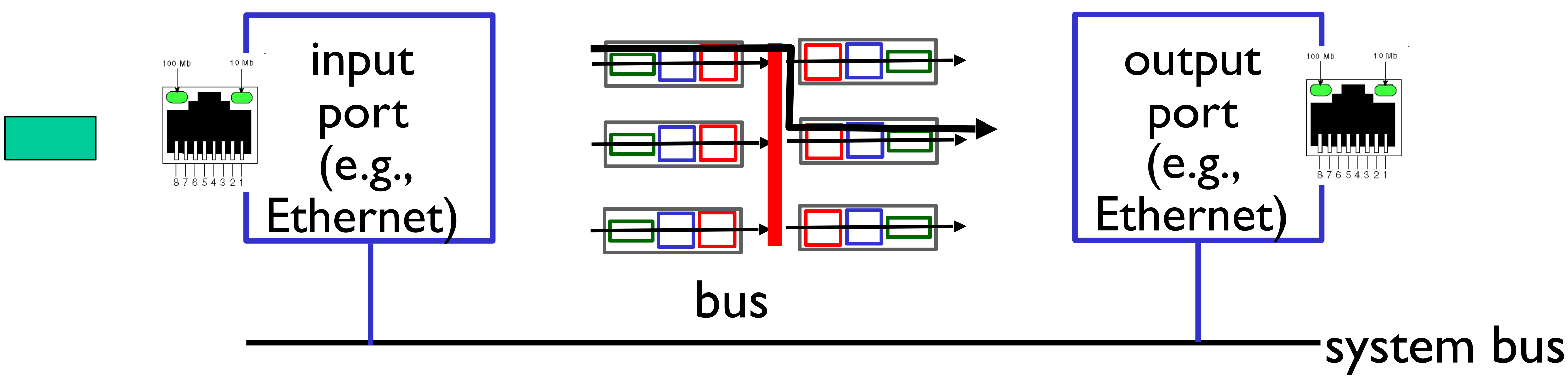
first generation routers:

- 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)



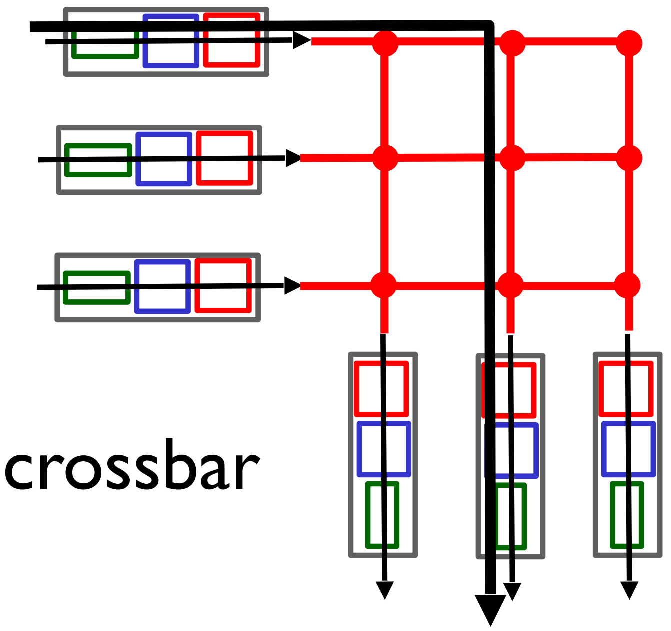
Switching via a bus

- 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

- Overcomes 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



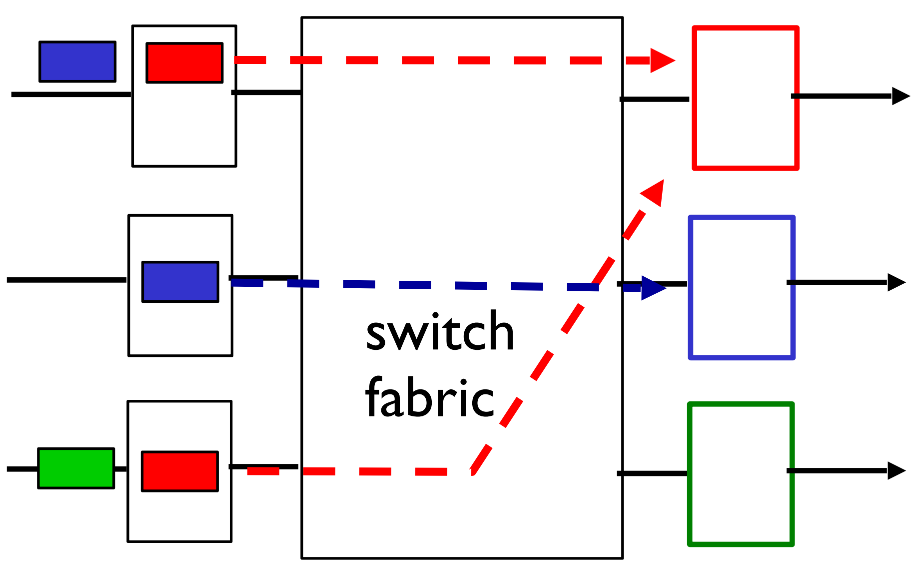
crossbar

Queuing

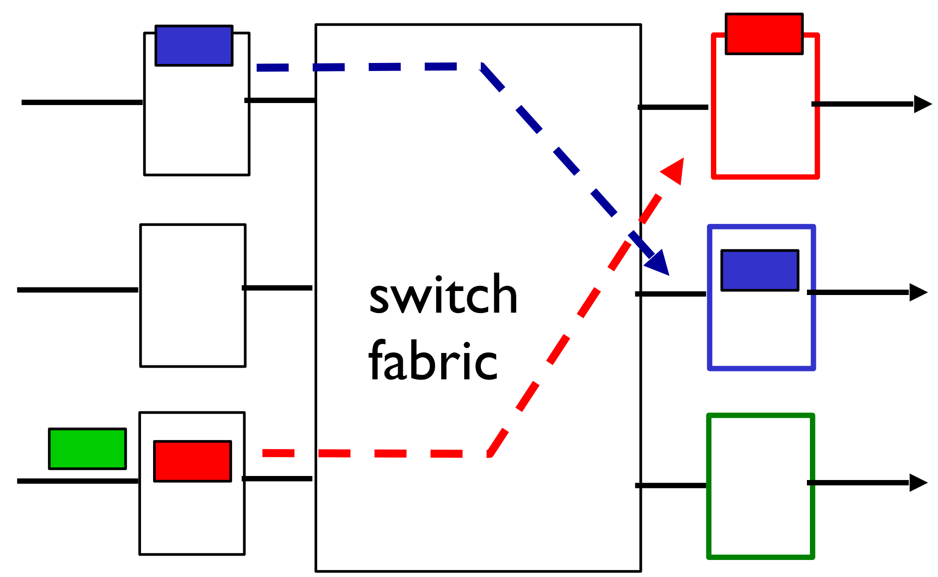
- At Input Port
- At Output Port

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



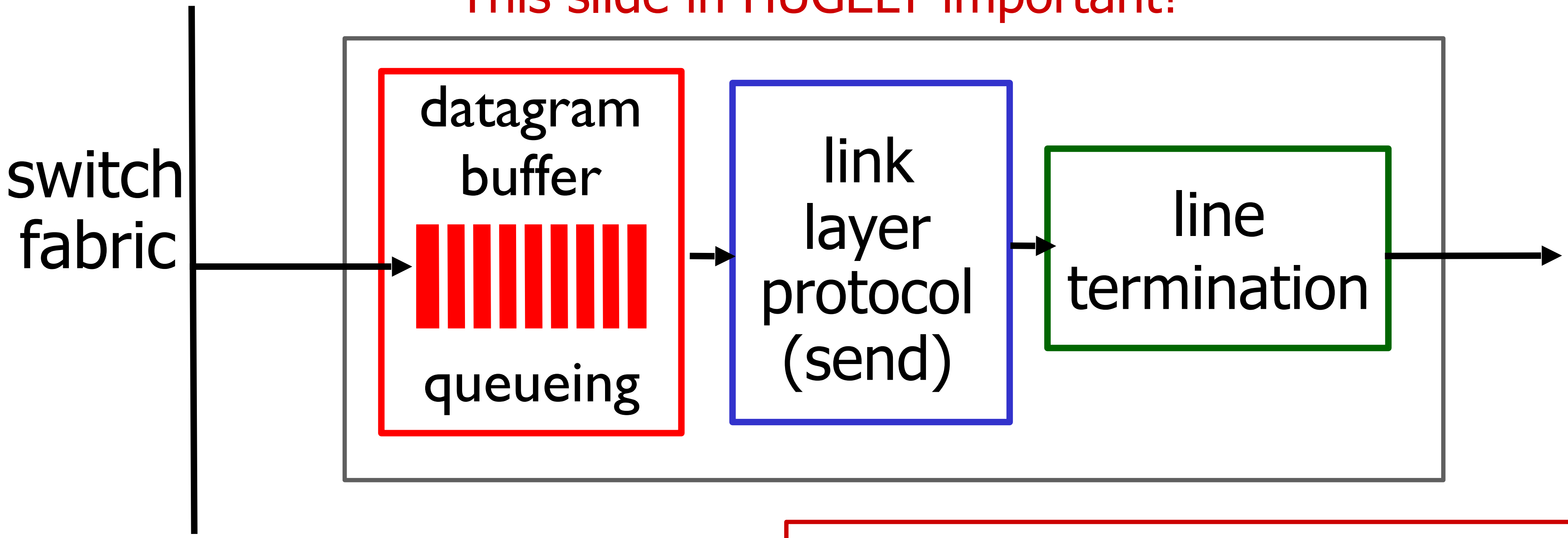
output port contention:
only one red datagram can be transferred.
lower red packet is blocked



one packet time later:
green packet experiences HOL blocking

Output ports

This slide in HUGELY important!

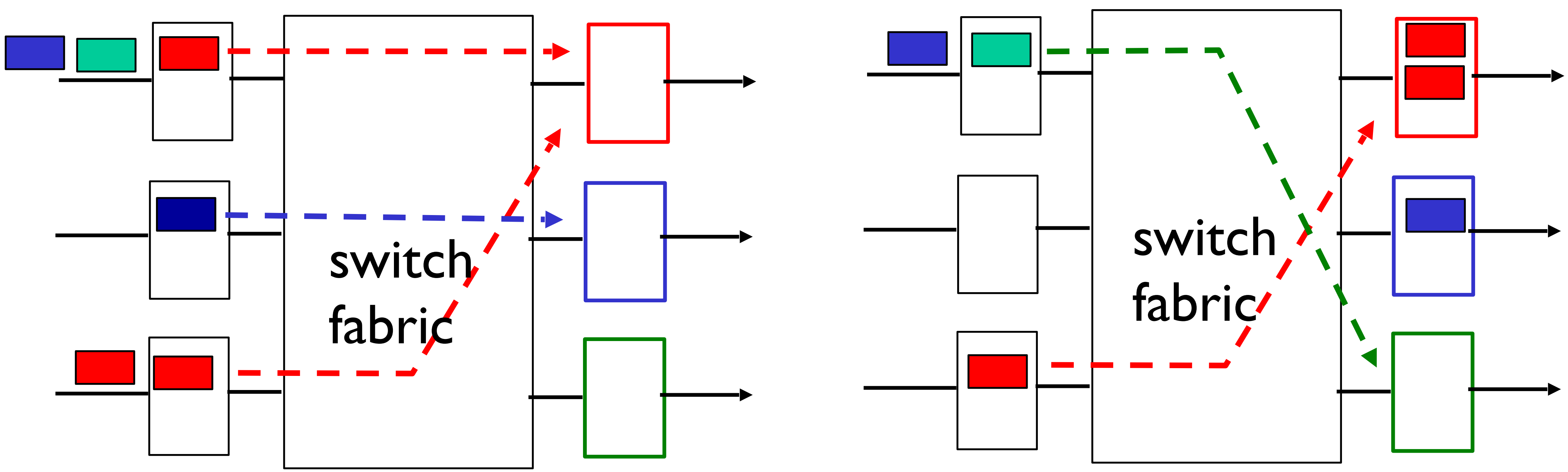


Datagram (packets) can be lost due to congestion, lack of buffers

Priority scheduling – who gets best performance, network neutrality

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

Output port queueing



at *t*, packets move 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!*

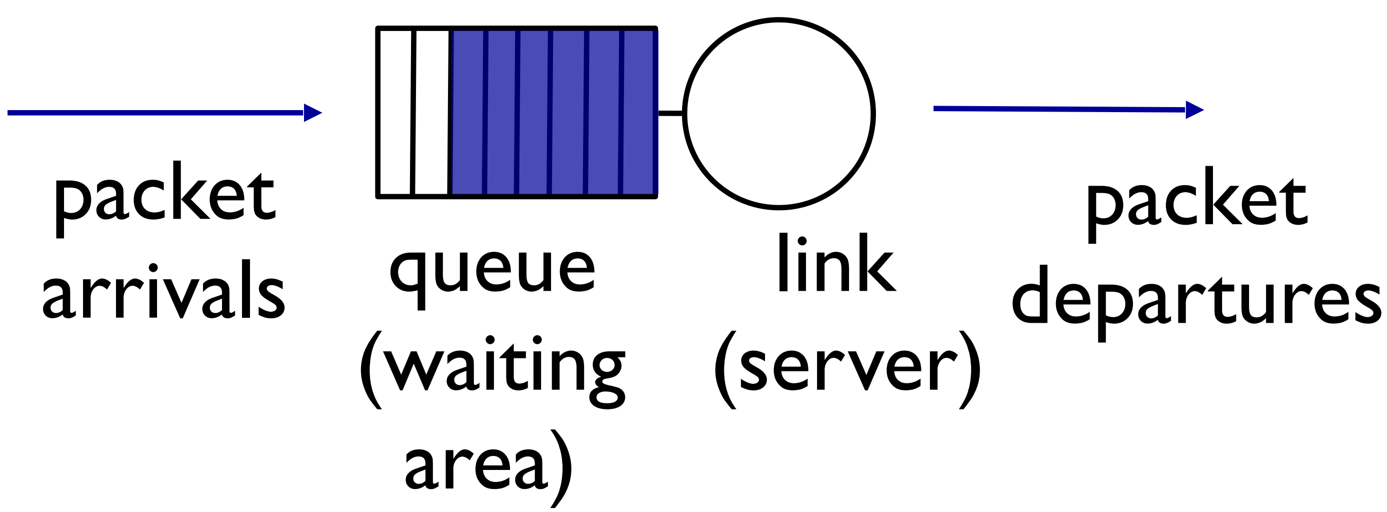
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to "typical" RTT (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

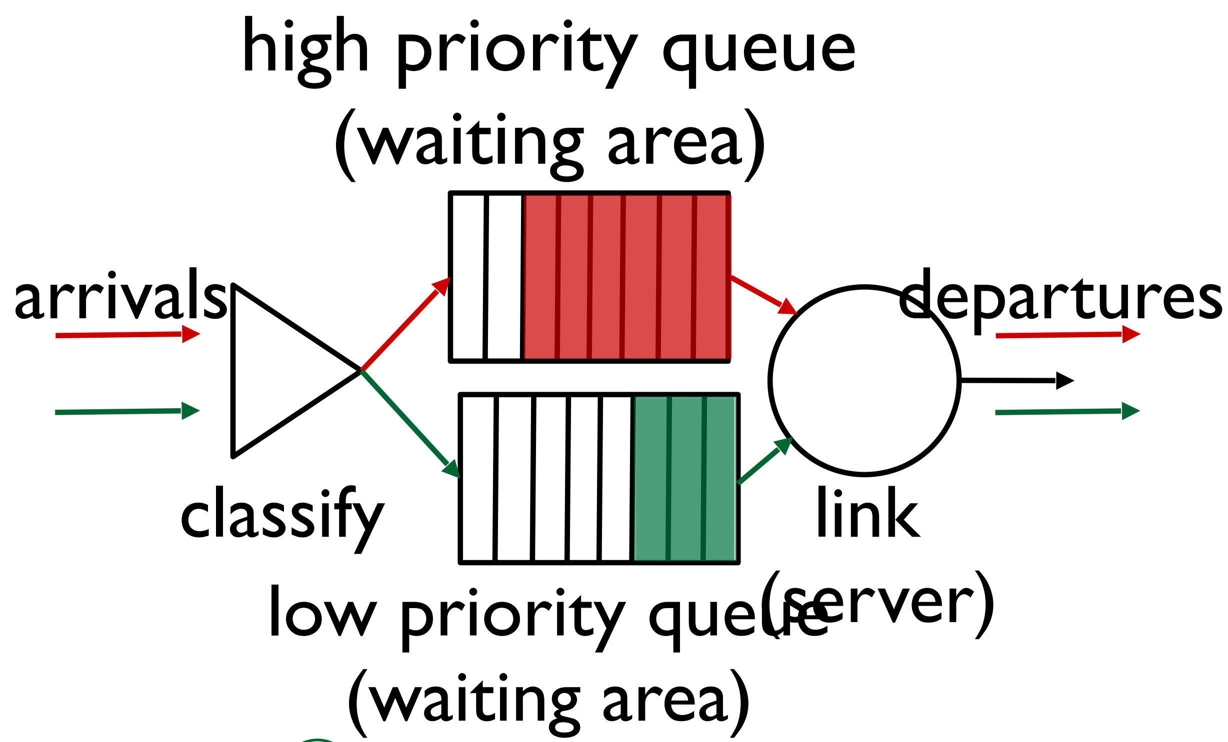
- *scheduling:* choose next packet to send on link
- *FIFO (first in first out) scheduling:* send in order of arrival to queue
  - real-world example?
  - *discard policy:* if packet arrives to full queue: who to discard?
    - *tail drop:* drop arriving packet
    - *priority:* drop/remove on priority basis
    - *random:* drop/remove randomly

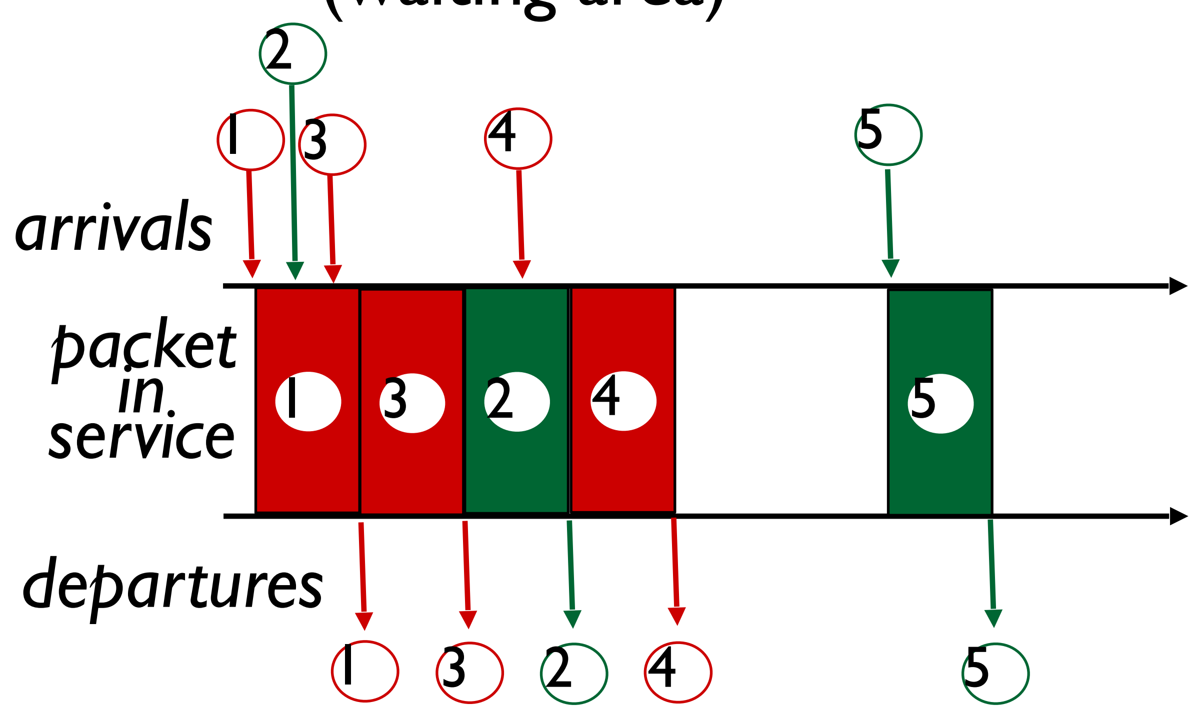


Scheduling policies: priority

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?

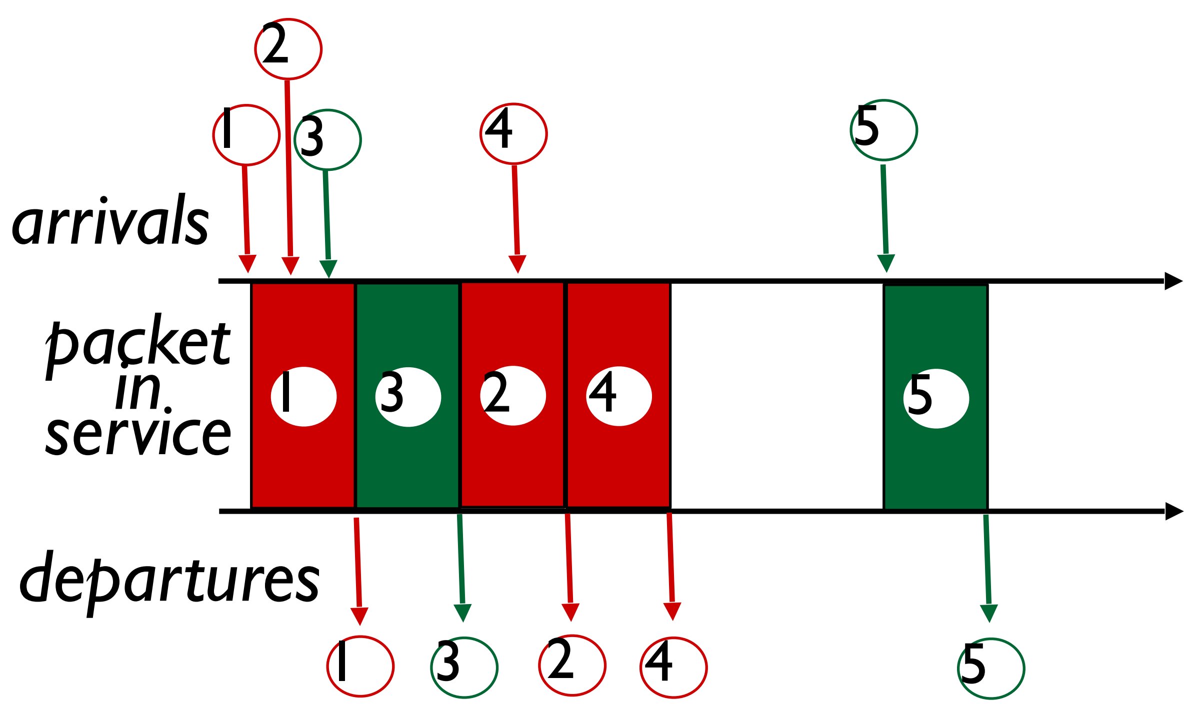




Scheduling policies: still more

Round Robin (RR) scheduling:

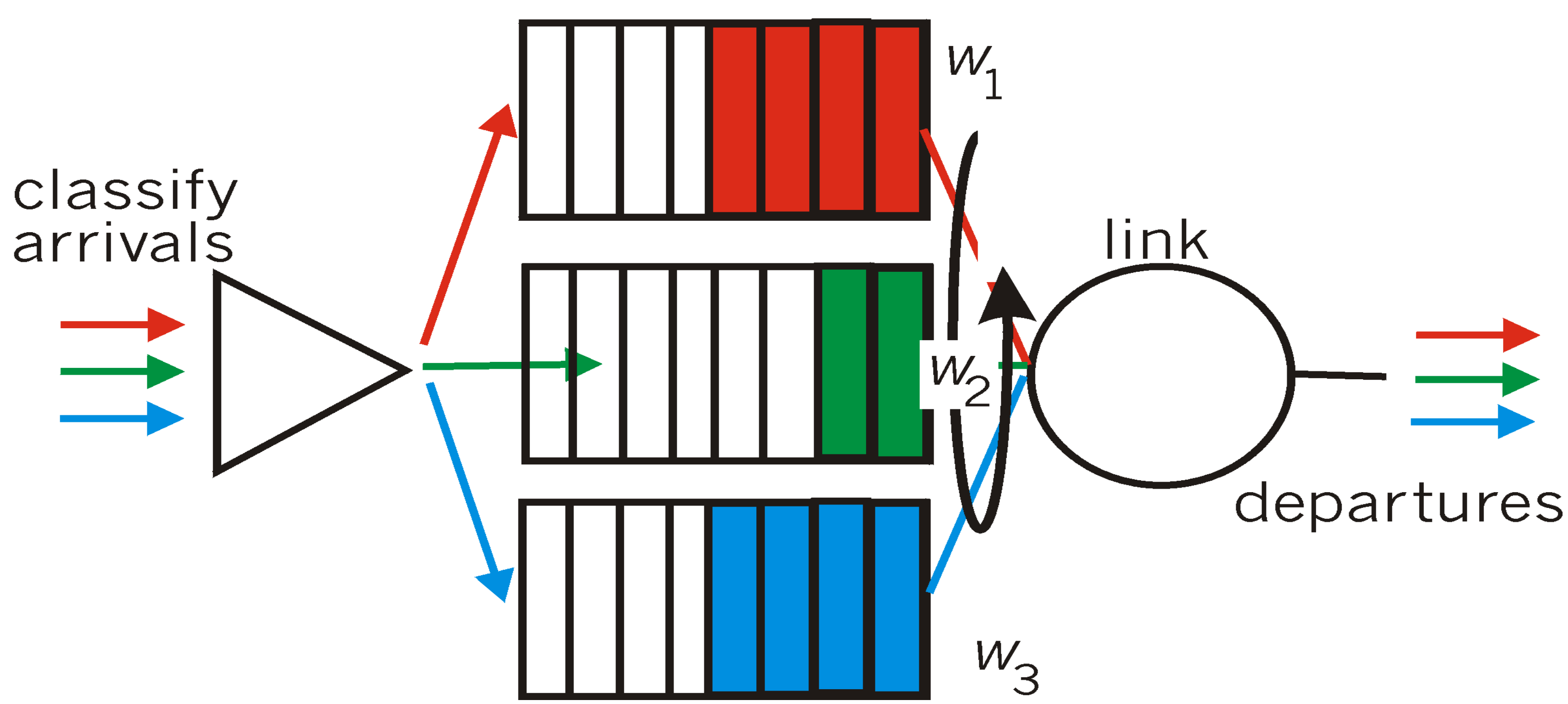
- multiple classes
- cyclically scan class queues, sending one complete packet from each class (if available)
- real world example?



Scheduling policies: still more

Weighted Fair Queuing (WFQ):

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



Summary

- Goals
- Network layer functions
- Data and Control plane
- Service model
- Destination based forwarding
  - Longest prefix match
- Switching fabrics
  - memory, bus, cross connect
- Queuing
- Scheduling