

CS 325 I - Computer Networks I: Network Layer

Professor Patrick Traynor
Lecture 11
9/24/13

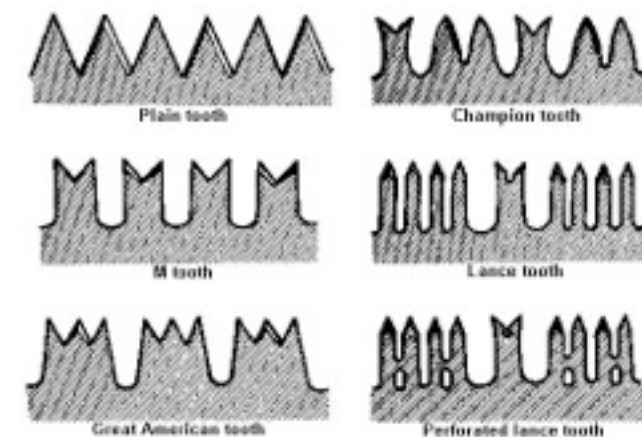
Reminders

- Check the course calendar!
 - Homework 2 due in one week
 - Project 2 due in two weeks
- Project 1 returned
 - Summary: You either got it or you didn't.
 - The difficulty of the Projects increases *steeply* from here on out. It may be time for some self-evaluation.

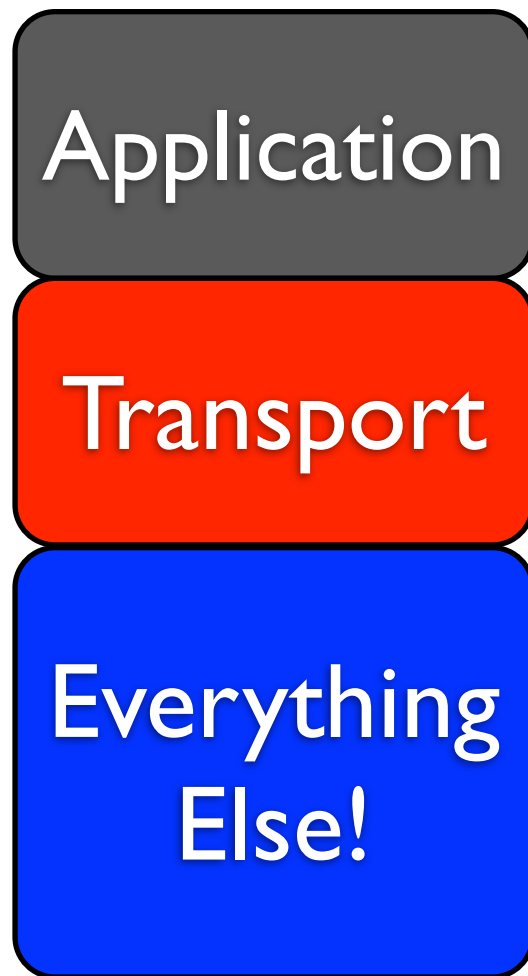


Review

- What is AIMD?
 - When do we use it?
- What is the “steady state” profile of TCP?
 - Is this good or bad?
- Why not use a mechanism like ATM ABR everywhere?
- Is TCP fair?



Where in the Stack...



The Web, DNS, Bittorrent, etc

Process to Process -
Guarantees?

Everything Else? What's that?



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)
 - dealing with scale
 - advanced topics: IPv6, mobility
 - expect to hear more on mobility later!
- instantiation, implementation in the Internet

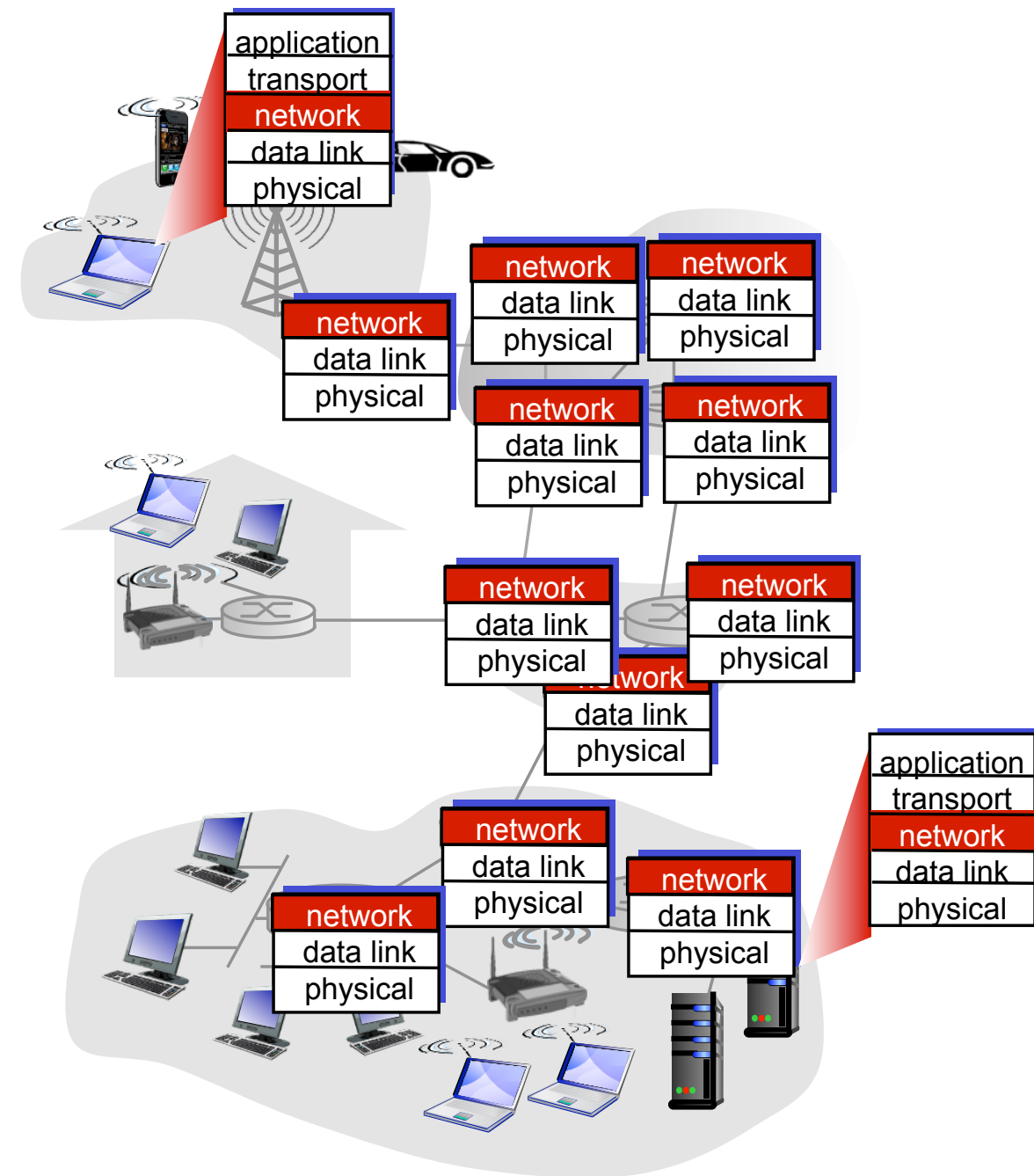


Chapter 4: Network Layer

- 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/Multicast

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
- *this is the first layer that everything between you and the receiver looks at!*



Two Key Network-Layer Functions

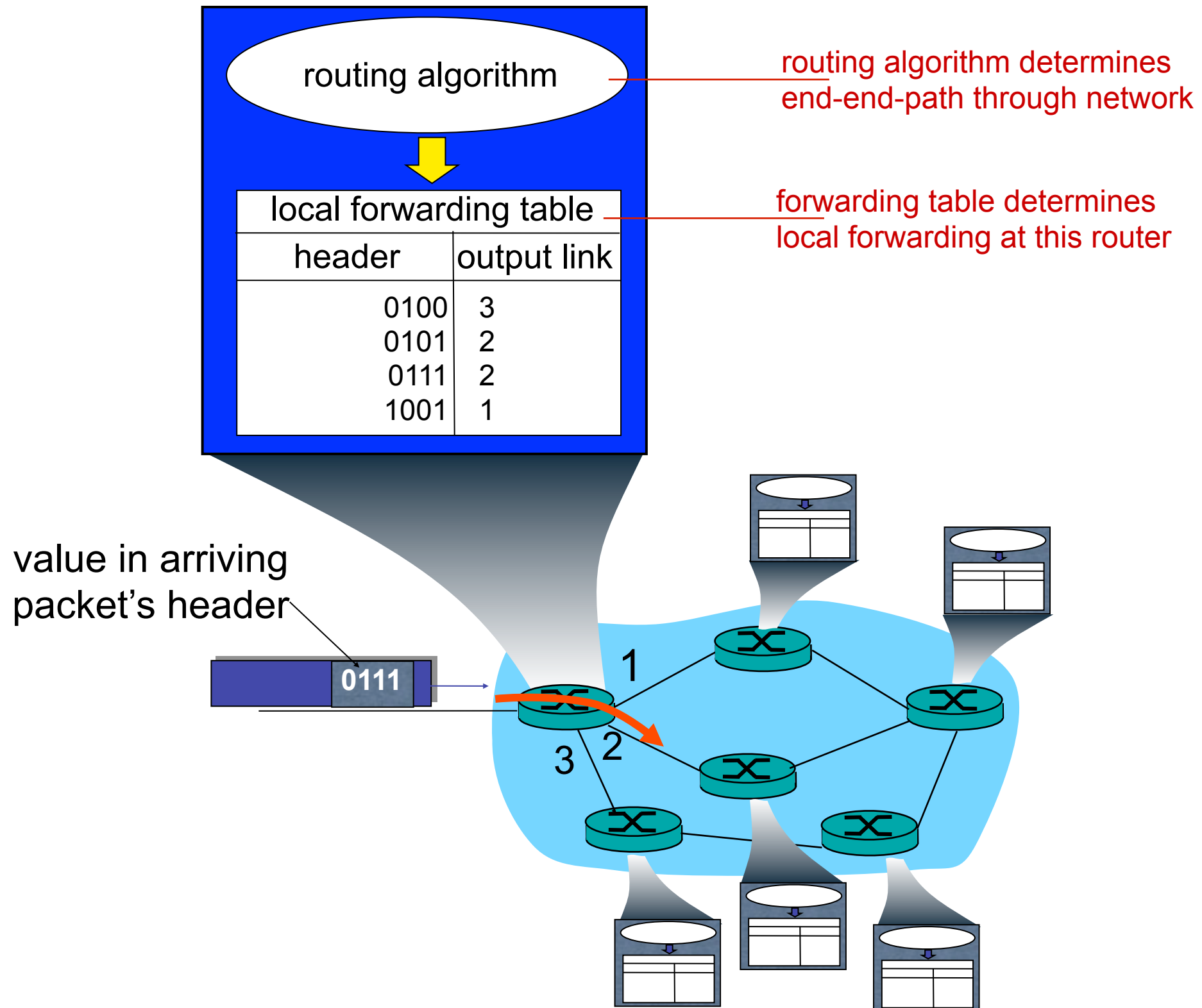
- *forwarding*: move packets from router's input to appropriate router output
- *routing*: determine route taken by packets from source to dest.
 - *routing algorithms*

analogy:

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

Think of it this way - this is the difference between figuring out the way to your destination using Google Maps and completing the next step in those directions!

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
- security

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	ABR	guaranteed minimum	no	yes	no	yes

Chapter 4: Network Layer

- 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 connection and connection-less service

- Datagram network provides network-layer connectionless service
- Virtual-Circuit network provides network-layer connection service
- Analogous to the transport-layer services (e.g., TCP), 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)

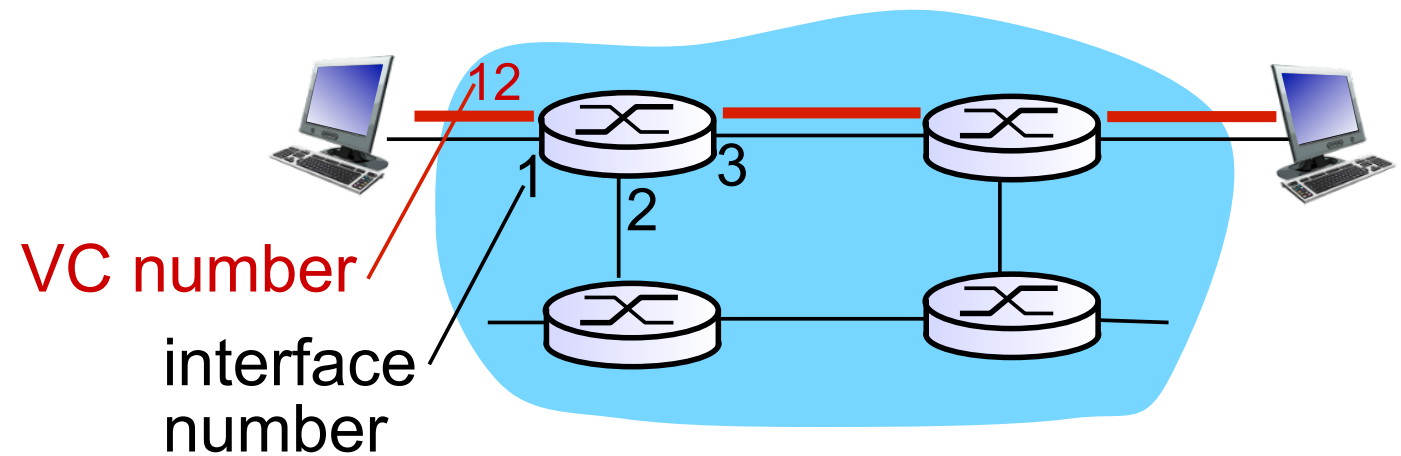
VC implementation

- 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 (why?)
 - New VC number comes from forwarding table



Forwarding table

Forwarding table in northwest router:

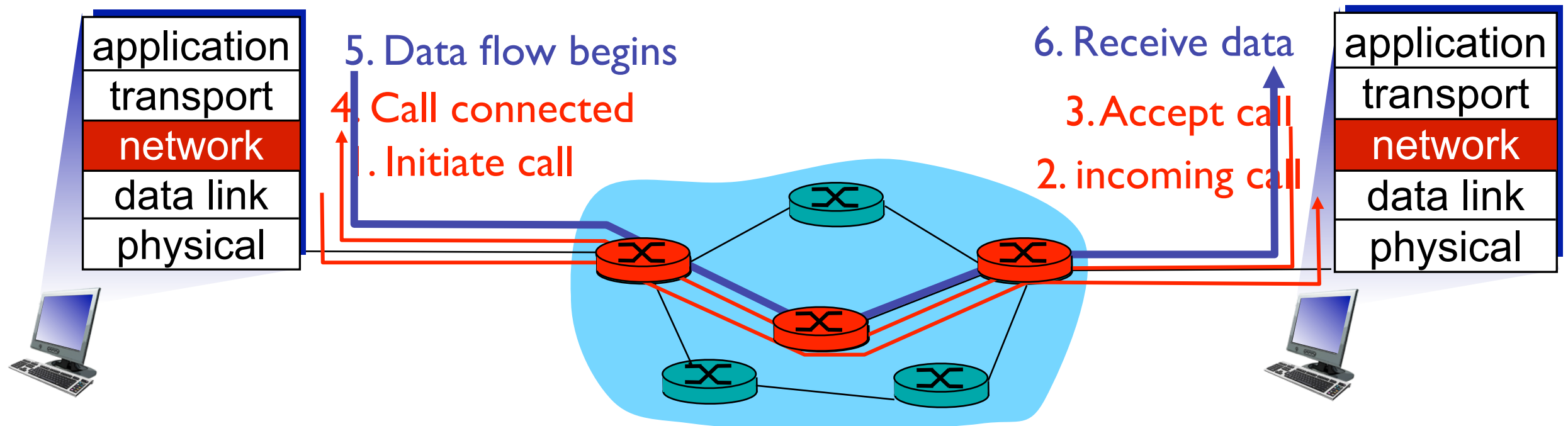


Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

VC routers maintain connection state information!

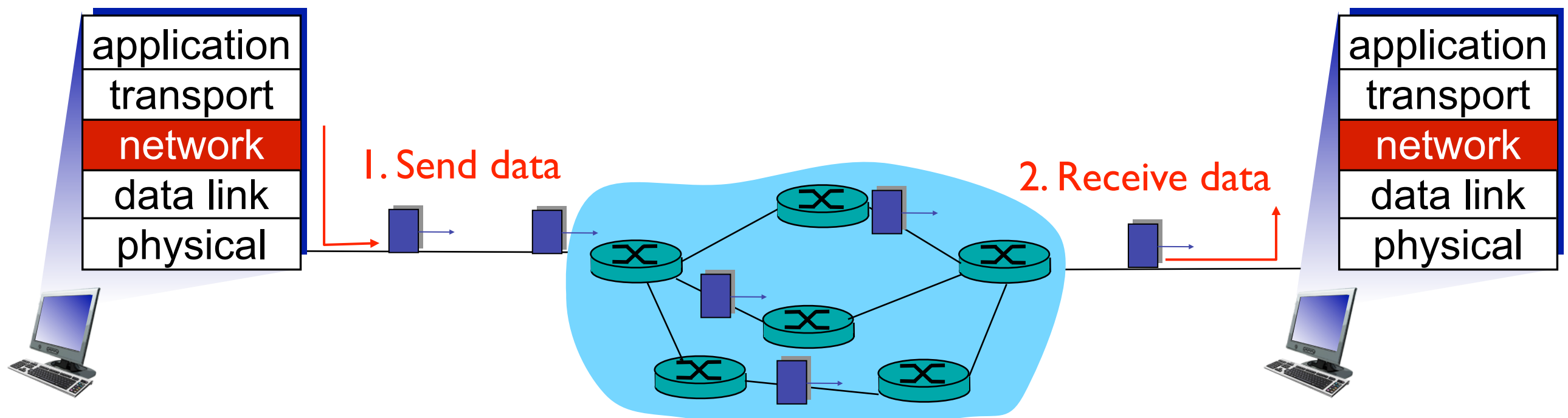
Virtual circuits: signaling protocols

- Used to setup, maintain teardown VC
- Used in ATM, frame-relay, X.25
- Not used in today's Internet (at the network layer)
 - Remember this for the next layer though!

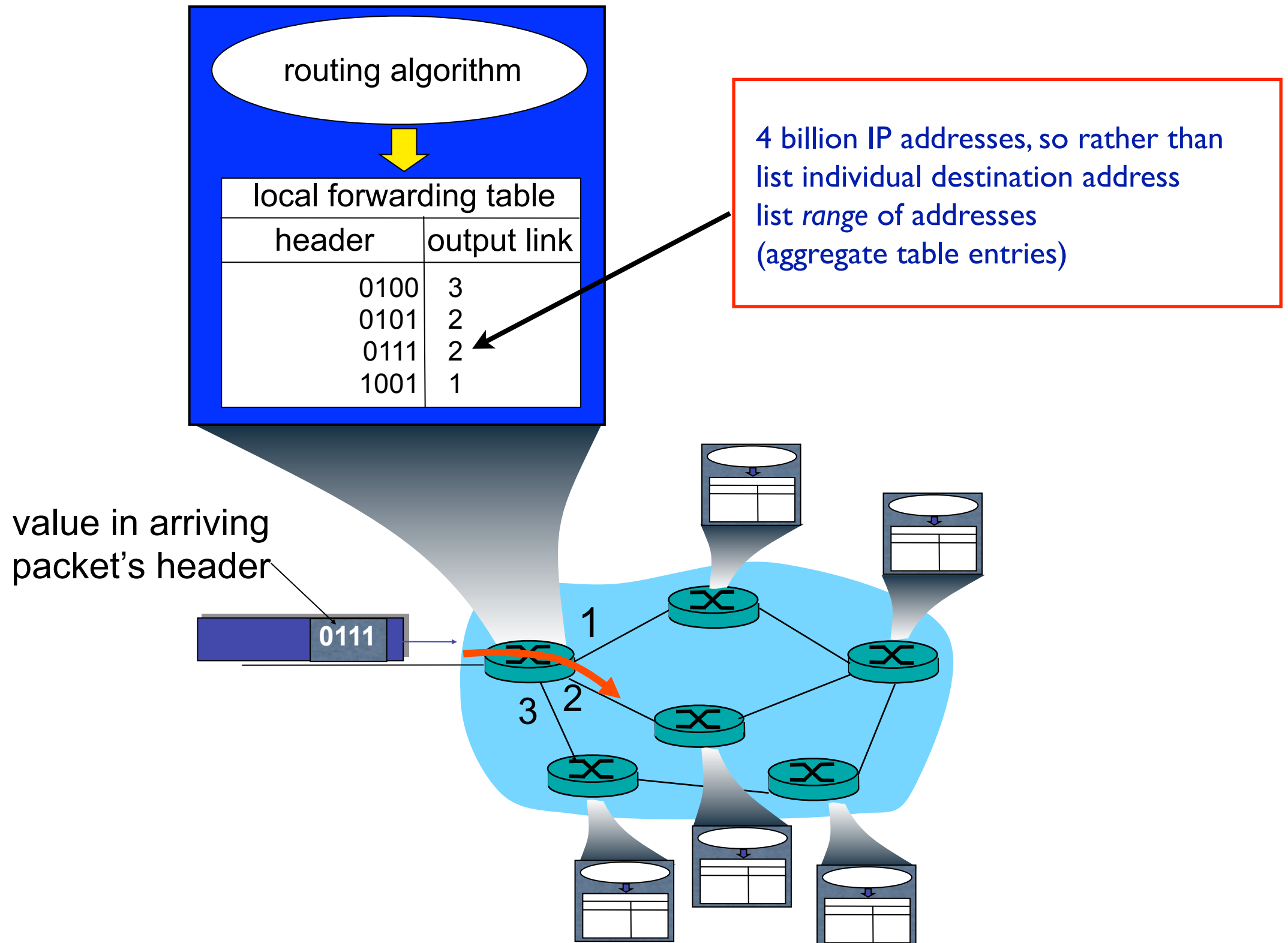


Datagram Networks

- No individual 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
 - packets between same source-dest pair may take different paths



Datagram 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:

Datagram or VC network: why?

Internet (datagram)

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

ATM (VC)

- evolved from telephony (human conversation):
 - strict timing and reliability
 - need for guaranteed service
- “dumb” end systems
 - telephones
- complexity inside network

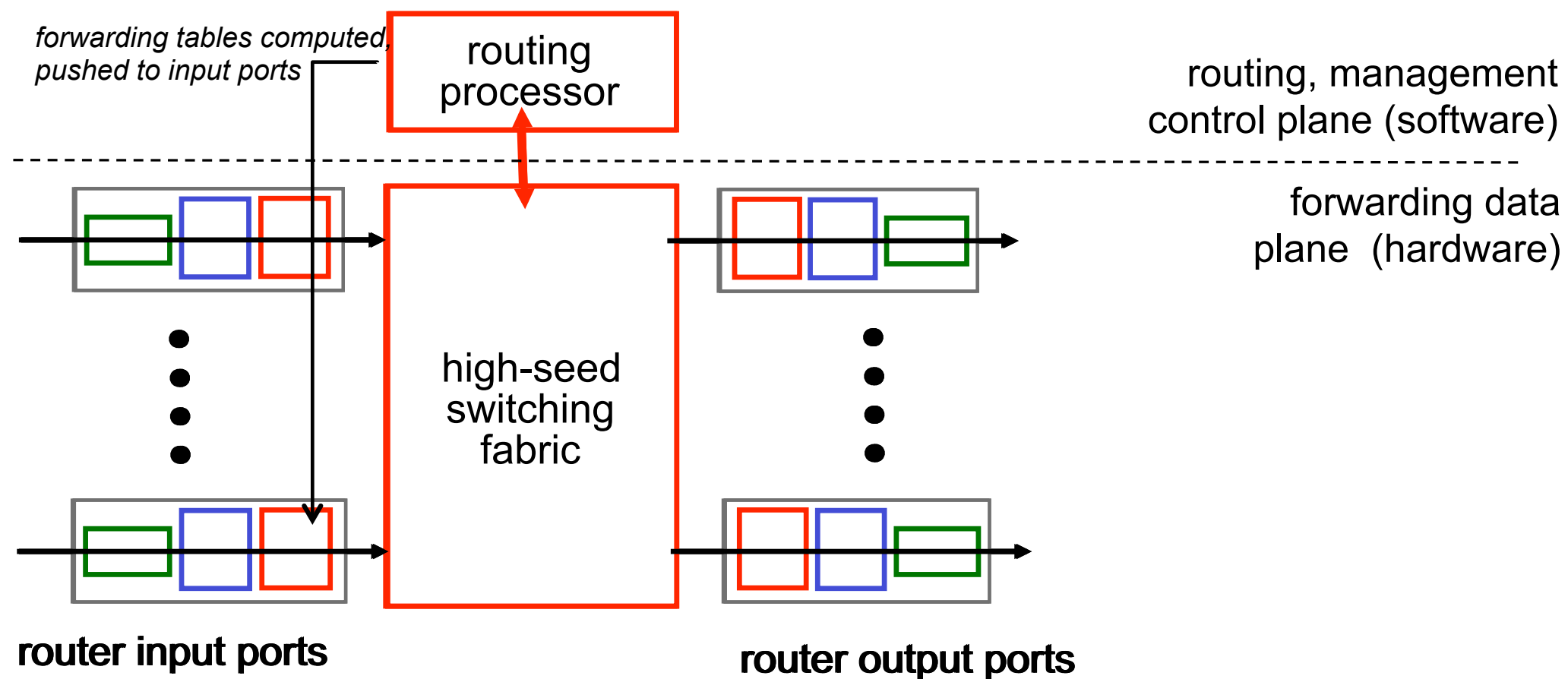
Chapter 4: Network Layer

- 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

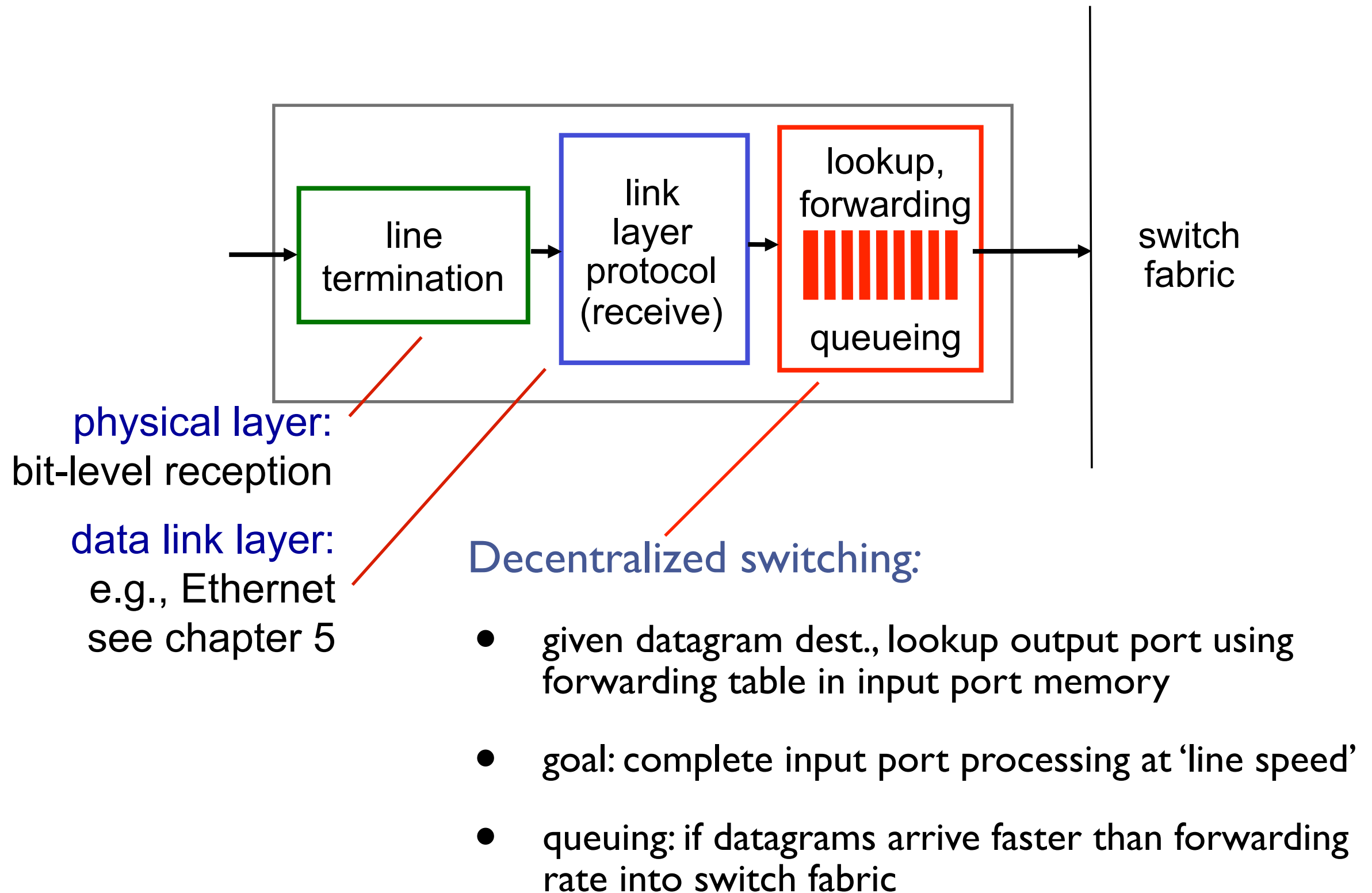
Router Architecture Overview

Two key router functions:

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



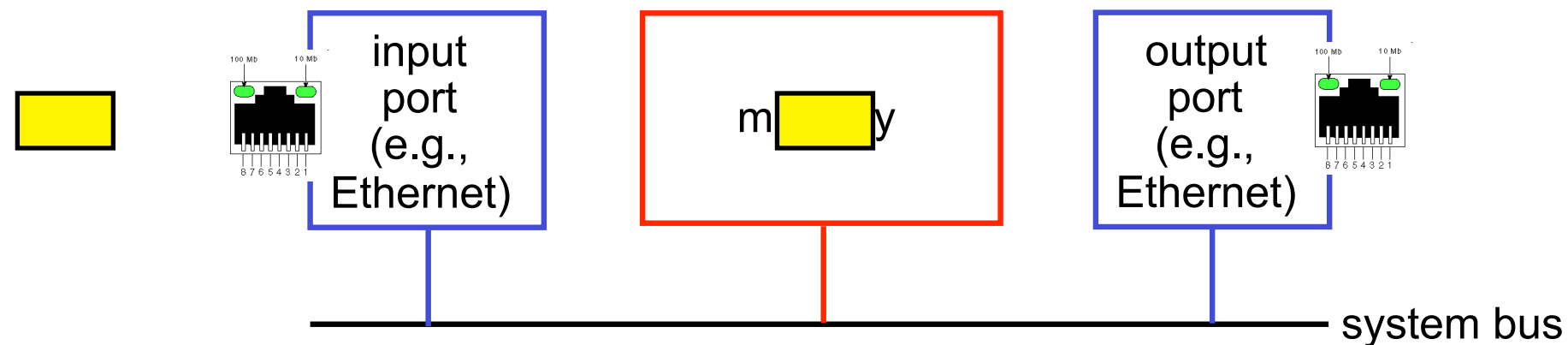
Input Port Functions



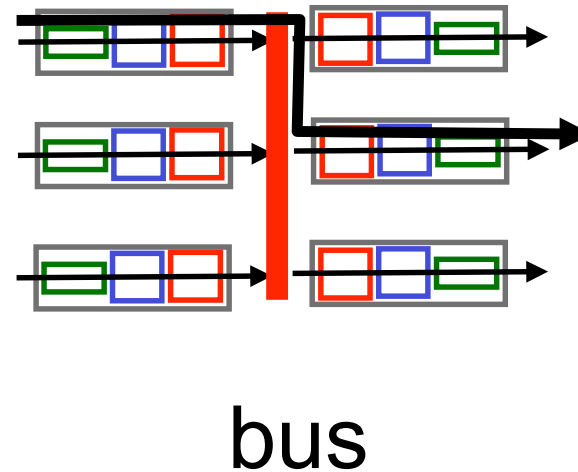
Switching via Memory

- *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)

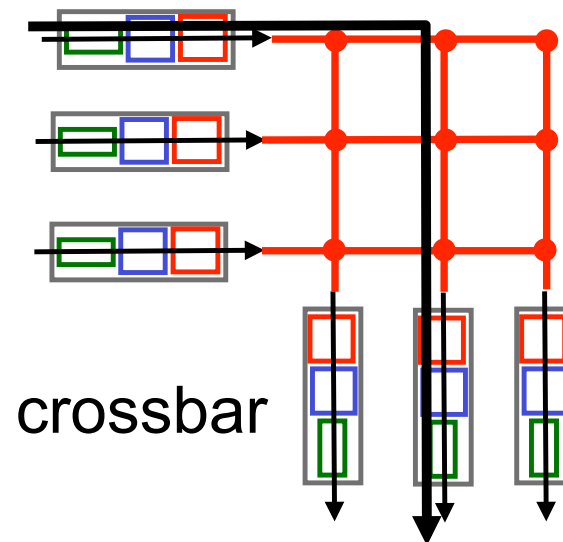


Switching Via a Bus



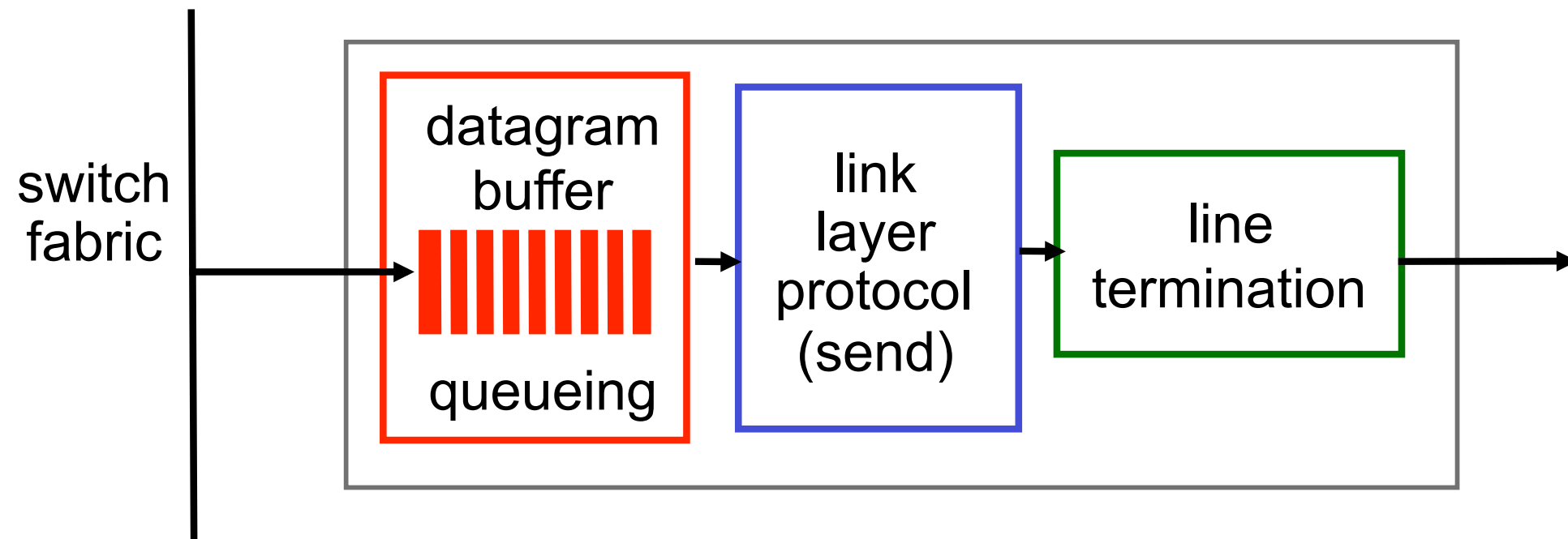
- datagram from input port memory to output port memory via a shared bus
- *bus contention*: switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers

Switching Via An Interconnection Network



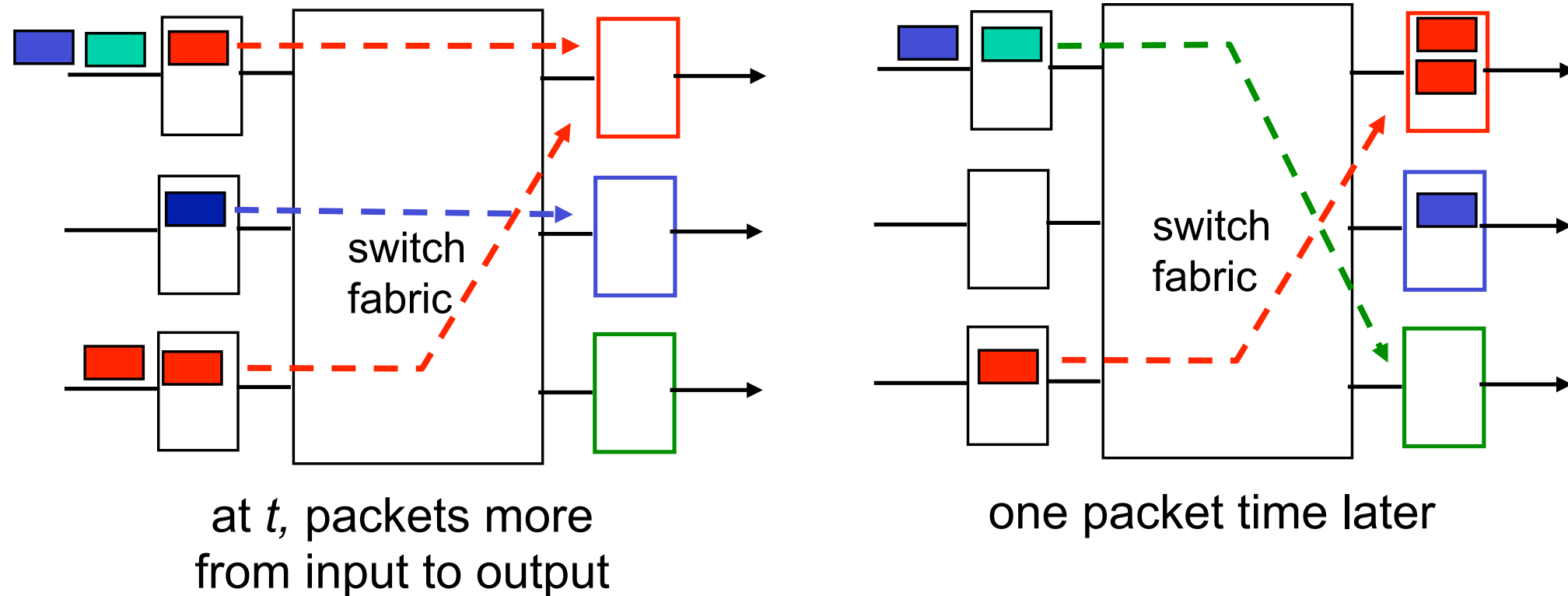
- overcome bus bandwidth limitations
- banyan networks, crossbar, other interconnection nets initially developed to connect processors in multiprocessor
- advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco I2000: switches 60 Gbps through the interconnection network

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 overflowing!**

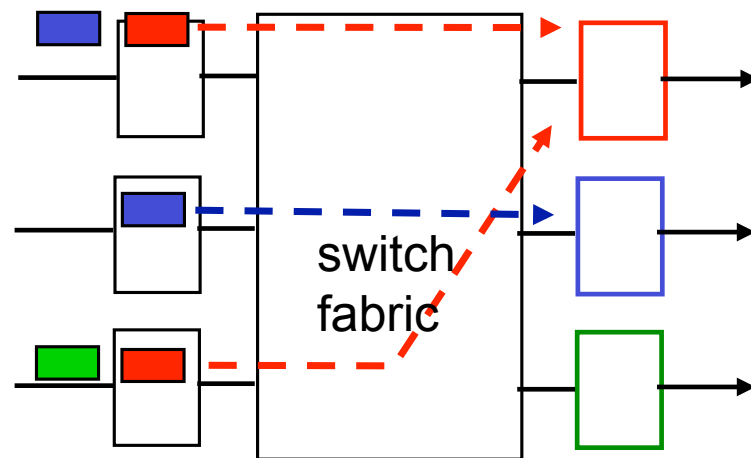
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity C
 - e.g., $C = 10$ Gps link: 2.5 Gbit buffer
- Recent recommendation: with N flows, buffering equal to:

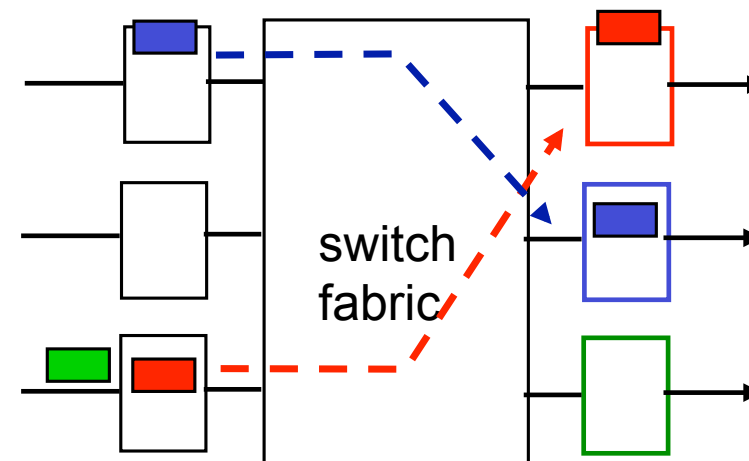
$$\frac{RTT \cdot C}{\sqrt{N}}$$

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

Until Next Time

- Project 2 - get started!
 - Everyone should be paired up at this point.
 - I have tried to help as best as I can. If you don't have a partner, come here now!!!
- Next Class
 - Read Section 4.4: IP

© Cartoonbank.com

