Computer Communications and Networks

Part 1: Introduction

Summary

- Contact information
- Course objective and topics
- Course workload
- Internet overview
- The layering principle of network protocols
- Reference models
- Connection-oriented and connectionless services
- Service primitives
- Performance measures

(1) Contact Information

- Course web page: https://bright.uvic.ca

- After you log in Brightspace, you should be able to see the course tab of CSC 361. If not, please send me an email and I will add you manually into the system.

Instructor: Jaya Prakash Champati

Email: jpchampati@uvic.ca

(2.1) Course Objectives

1) To help you gain a general understanding of the principles and concepts governing the operations of computer networks
2) To provide you with the opportunity to become skillful in the implementation and use of communication protocols
3) To help you grasp the basic research methodologies in the field of computer networks.

(2.2) Topics

- Introduction

  Internet overview;
  Access network technologies: basics;
  Backbone network technologies and structures: basics;
  Network architectures, services and protocols

- Application layer

  Client-server model;
  World-Wide Web (WWW);
  Hyper-Text Transfer Protocol (HTTP);
  Domain Name System (DNS);
  Socket Application Programming Interface

- Transport layer

  Transport layer services;
  User Datagram Protocol (UDP);
  Transmission Control Protocol (TCP);
  TCP connection management techniques;
  TCP flow, error and congestion control basics

- Network layer

  Network layer services;
  Internet Protocol (IP);
  Basic routing algorithms: distance vector and link state;
  Internet routing protocols: basics

- Link layer

  Link layer services;
  Link layer error and flow control techniques;
  Medium Access Control (MAC) techniques;
  Link layer interworking techniques;
  IEEE 802.3 and 802.11 basics

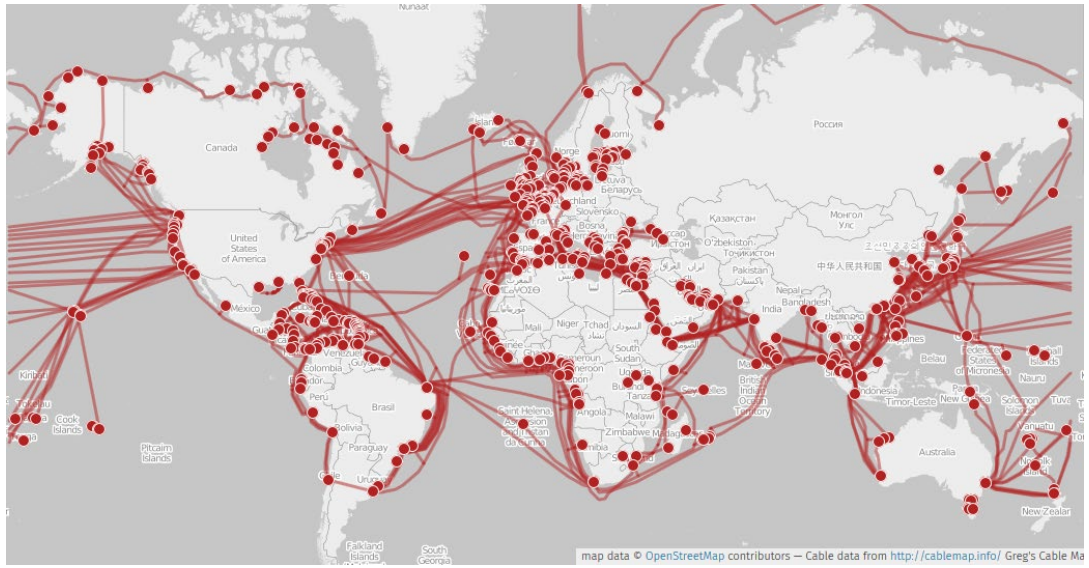
(3.1) Course Workload

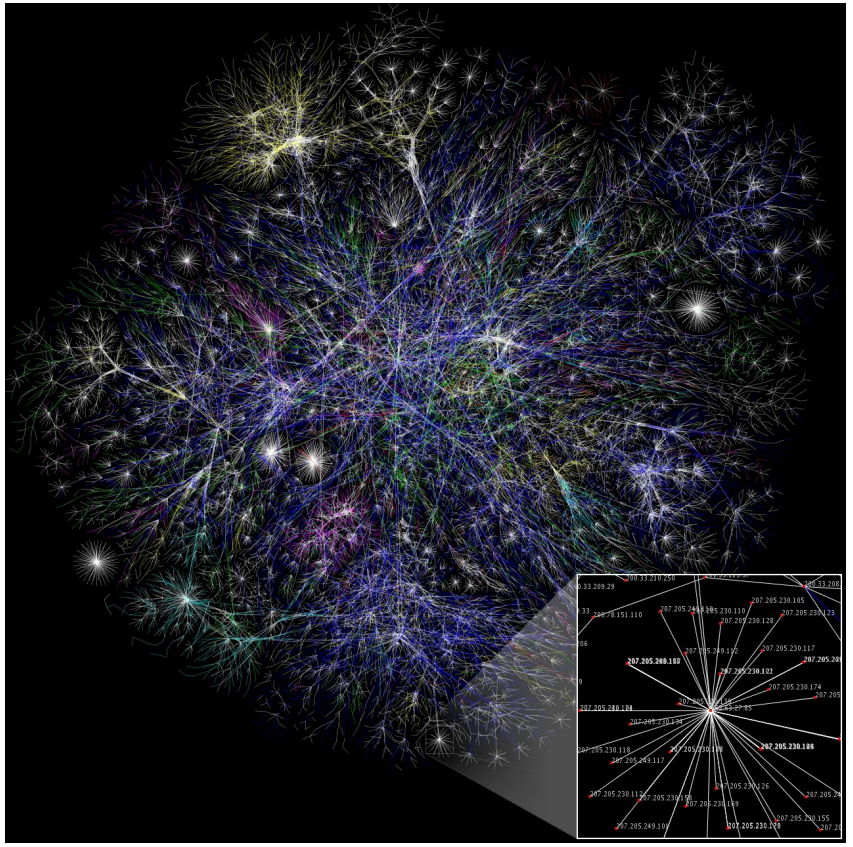
Assignment/Exams	Weight
Programming Assignment 1	13%
Midterm 1	15%
Programming Assignment 2	16%
Midterm 2	15%
Programming Assignment 3	16%
Midterm 3	15%
Homework Assignments	10%

(3.2) Suggested Approach

- Before lectures
  - read required sections in textbook; preview reference video
  - write down your questions
- Attend lectures
  - take notes, ask questions, and interact
- After lectures
  - explore further, get help and help others
  - attend labs and tutorials
  - start assignments early according to weekly schedule!

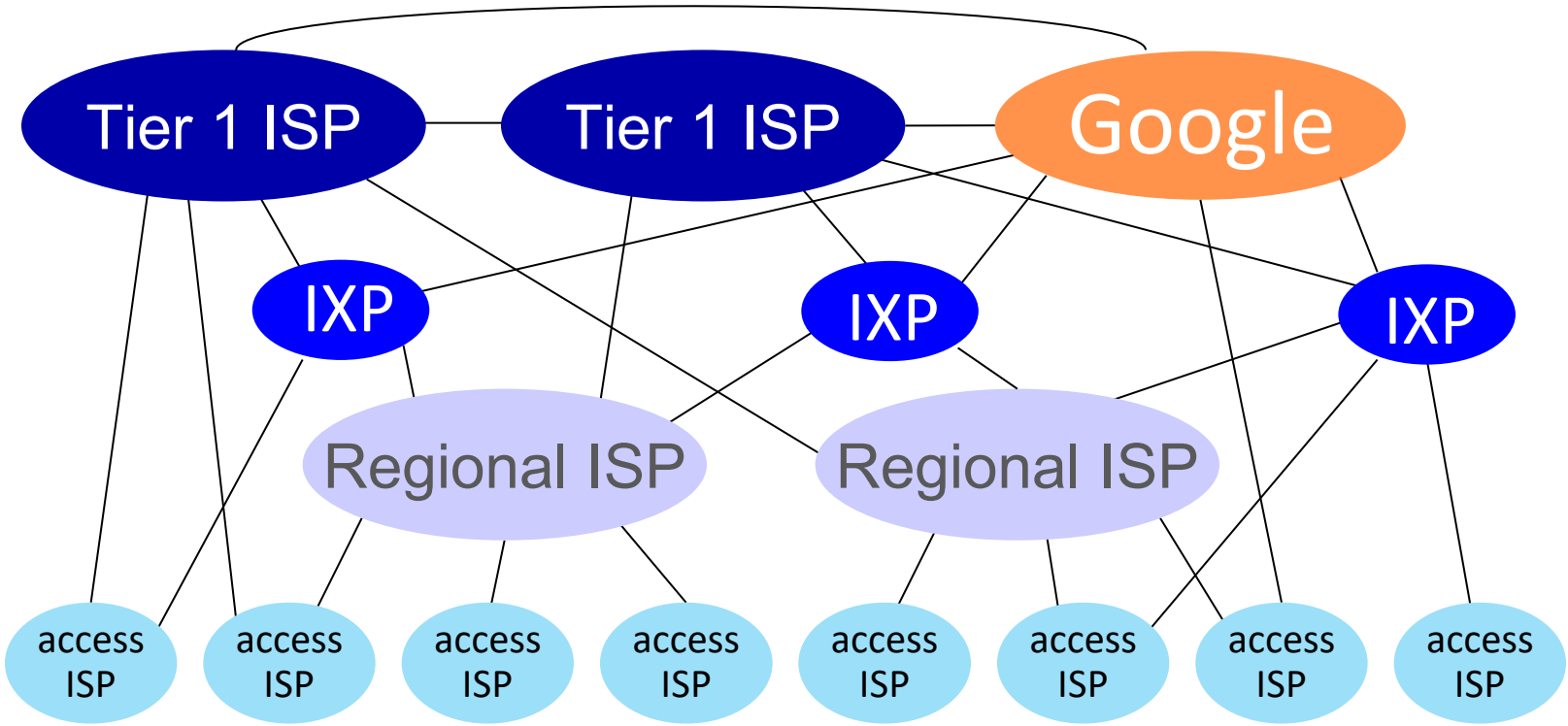
(4) What is Internet?





Source: Wikipedia

Internet structure: a “network of networks”



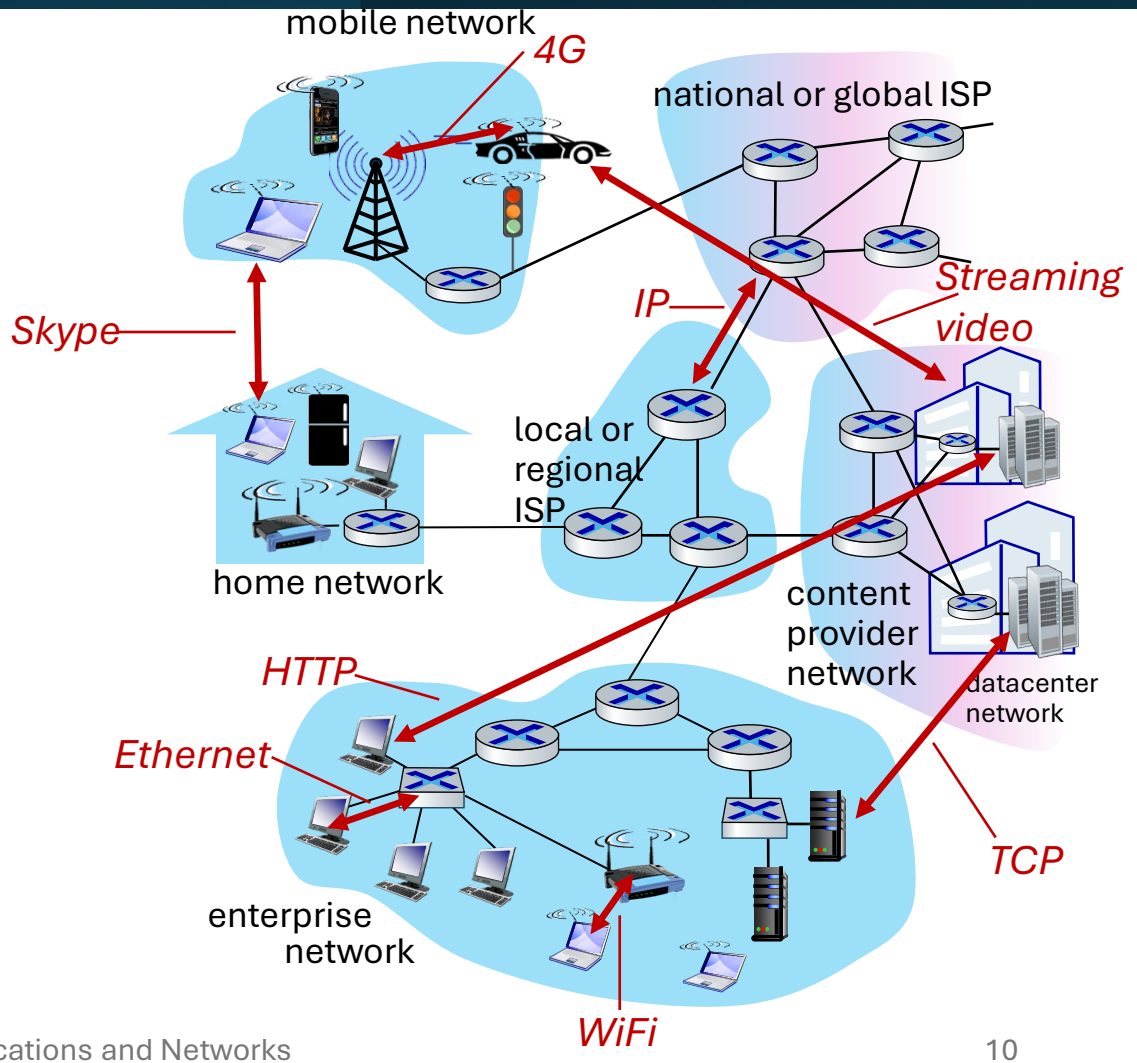
At “center”: small # of well-connected large networks

- “tier-1” commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
- content provider networks (e.g., Google, Facebook): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs

Source: Kurose & Ross

The Internet: a “nuts and bolts” view

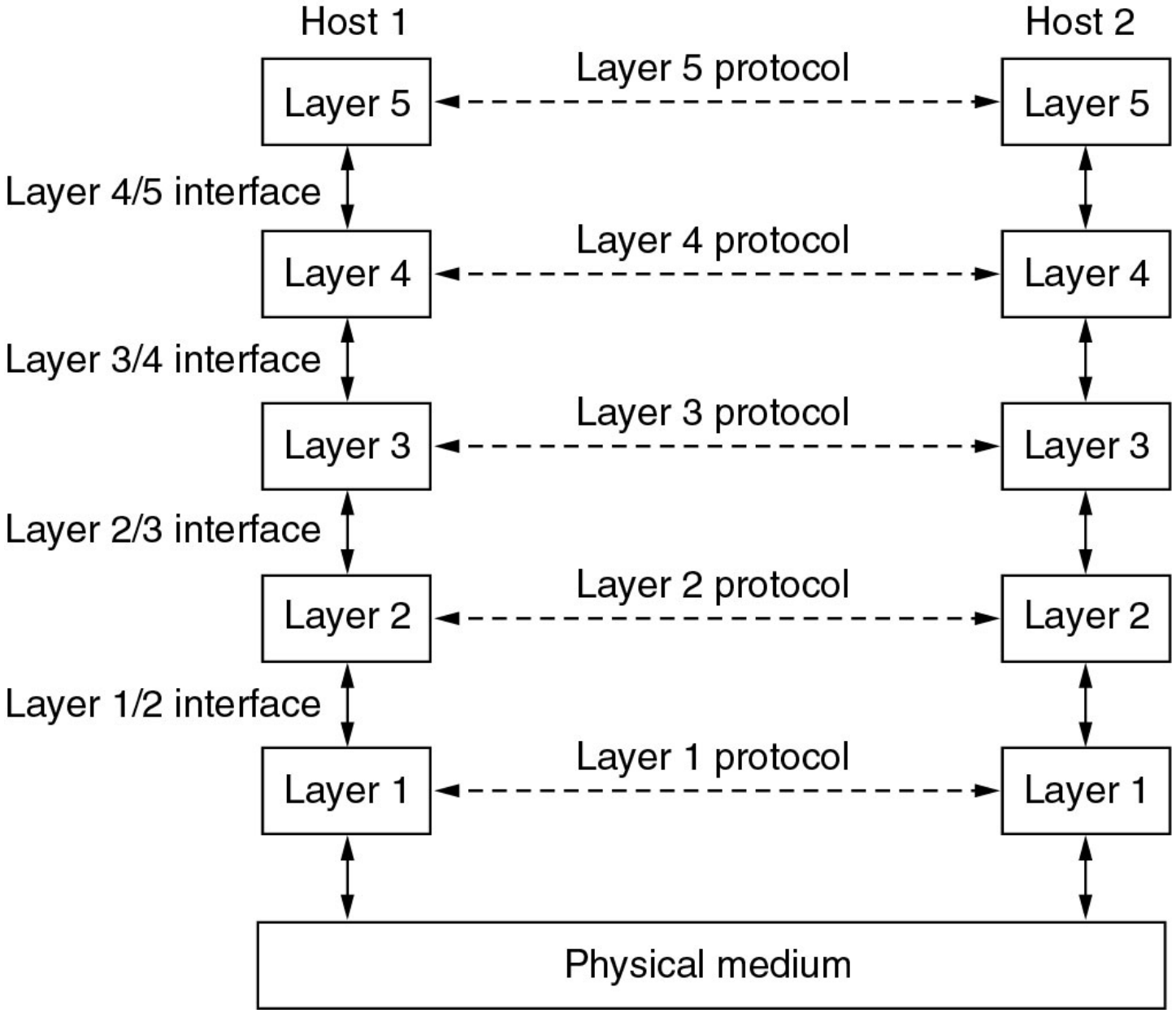
- *Internet:* “network of networks”
  - Interconnected ISPs
  - *protocols* are *everywhere*
    - control sending, receiving of messages
    - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4/5G, Ethernet
  - *Internet standards*
    - RFC: Request for Comments
    - IETF: Internet Engineering Task Force



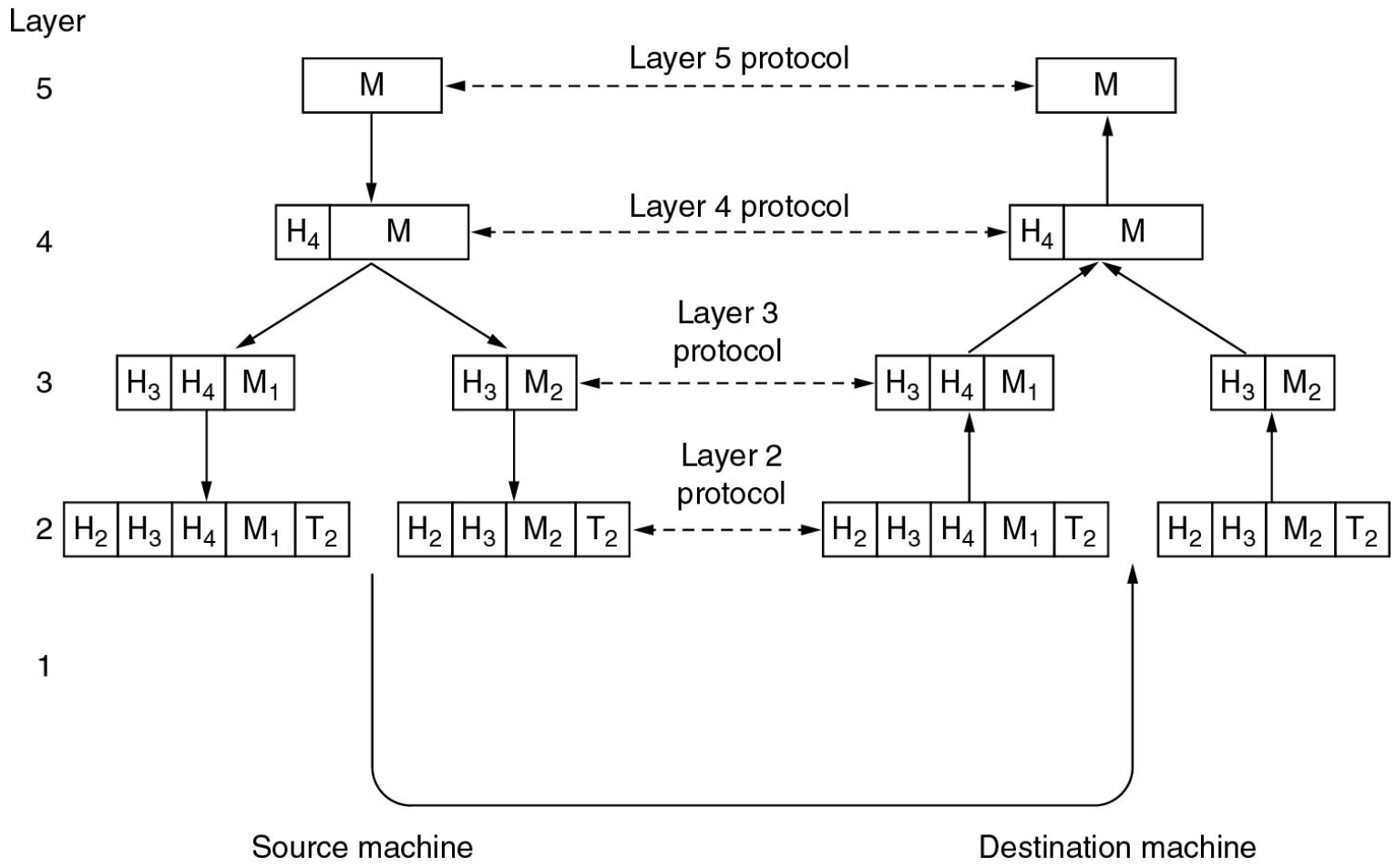
(4.1) Protocols

Basically, a protocol is an **agreement** between the communicating **peers** on how communication is to proceed.

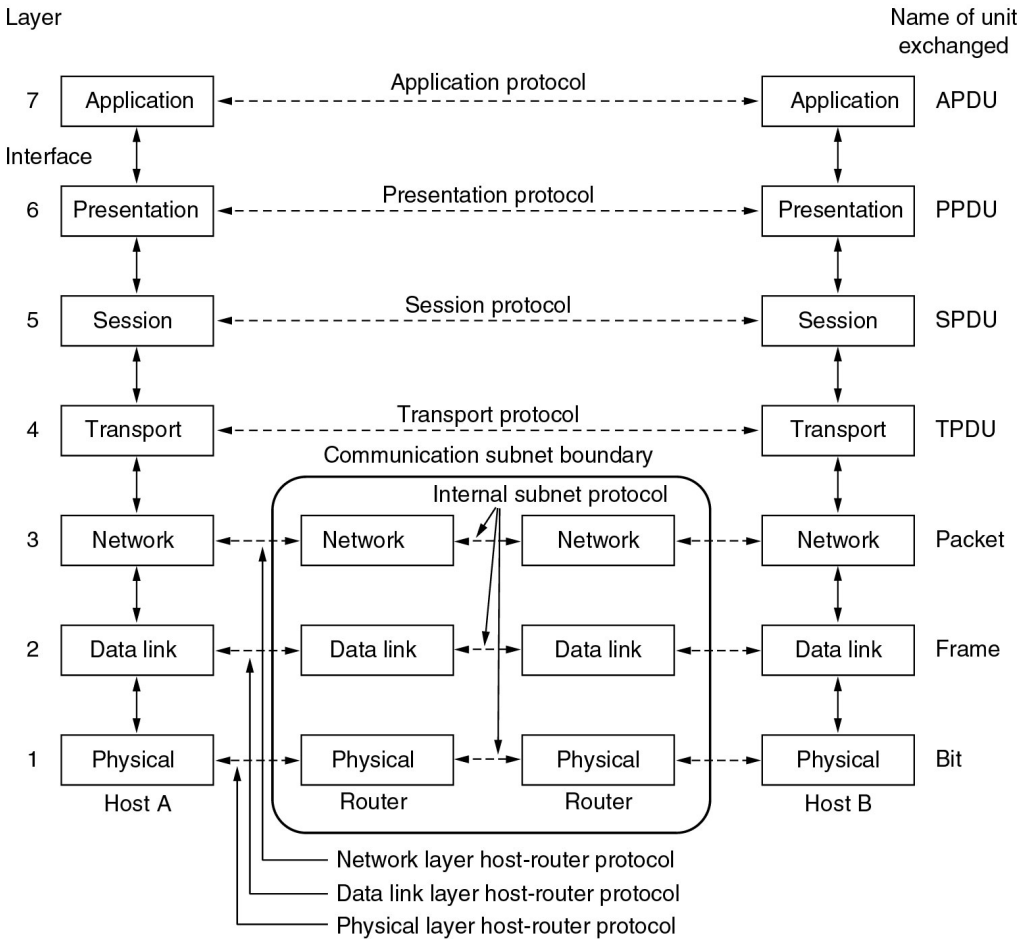
(4.2) The layering Principle (1)



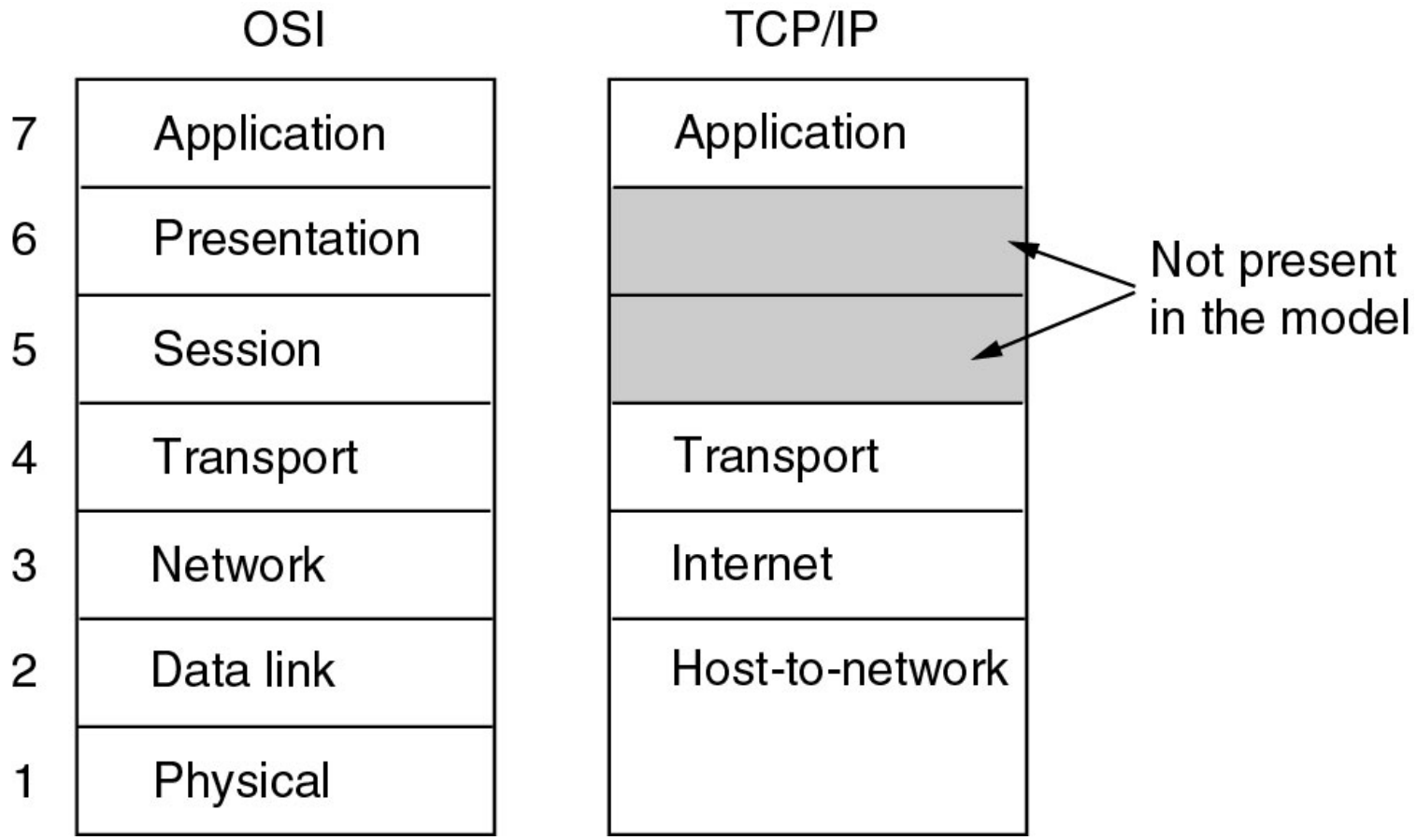
(4.2) The layering Principle (2)



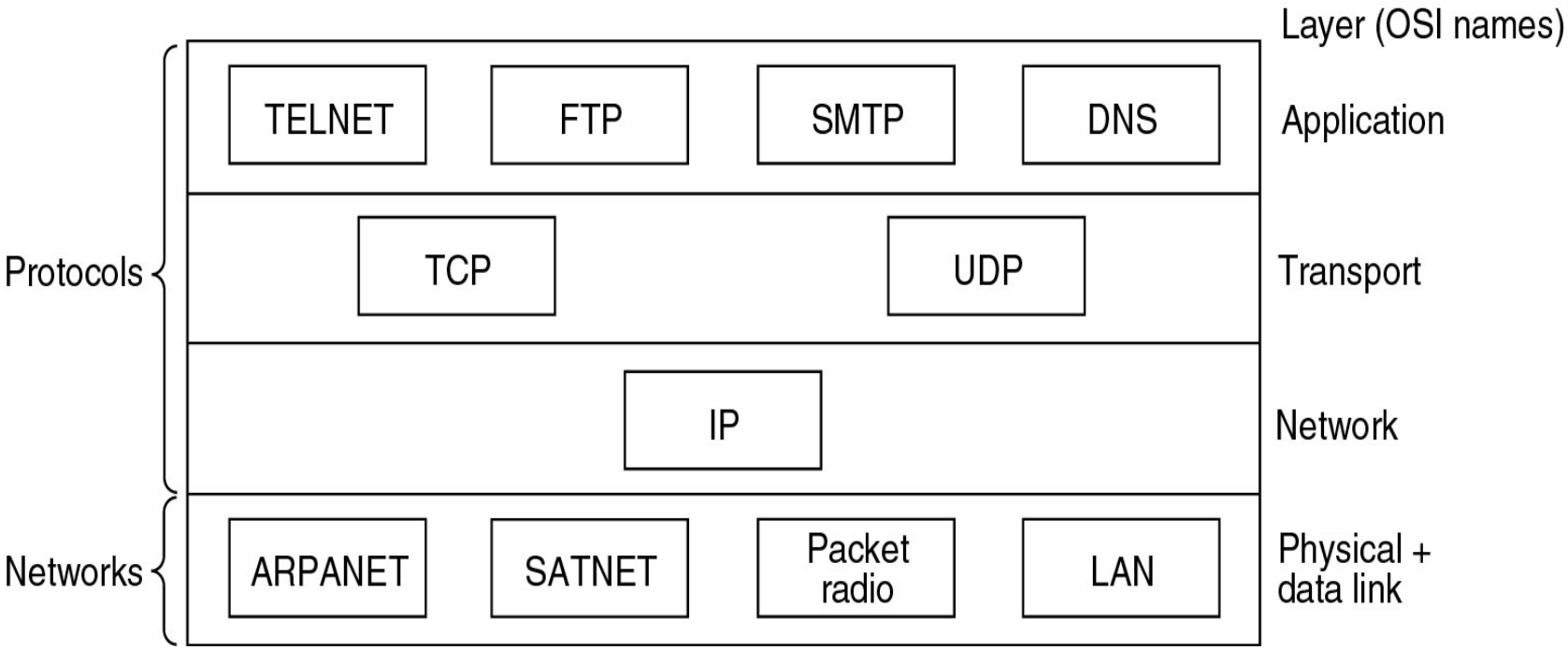
(5.1) The OSI Reference Model



(5.2) The TPC/IP Reference Model (1)



(5.2) The TPC/IP Reference Model (2)



Protocols and networks in the TCP/IP model initially

(6) Connection-Oriented vs.Connectionless

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

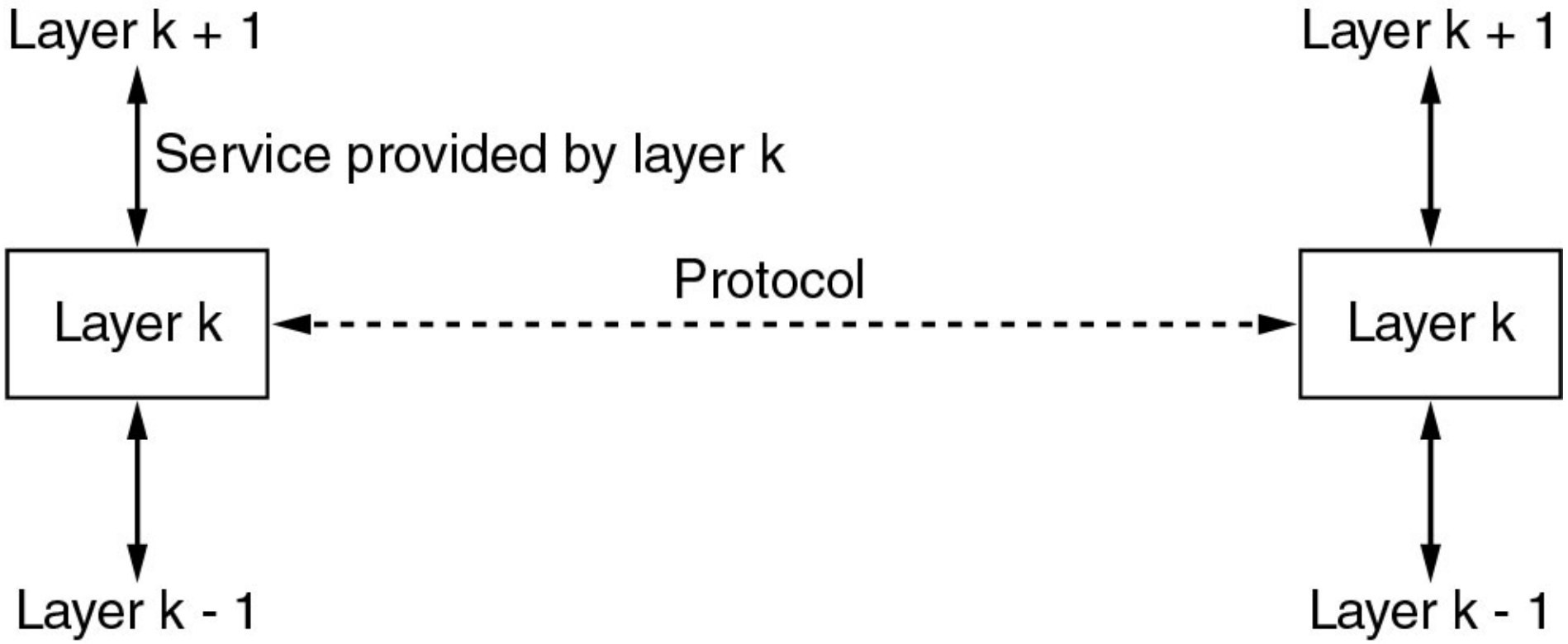
(7) Service Primitives

A service is formally specified by a set of primitives (basic operations) available to a user or other entity to access the service.

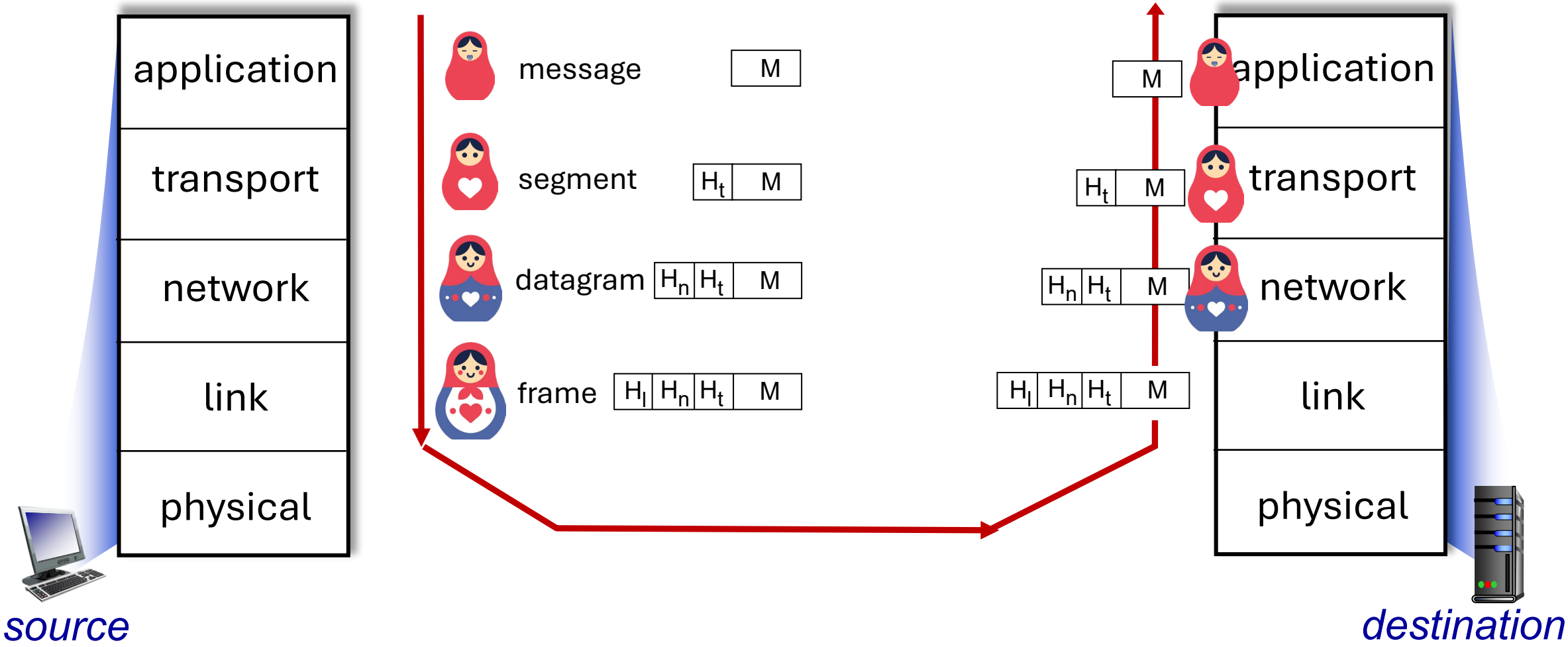
Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Example: five service primitives for implementing a simple connection-oriented service.

(8) The Relationship of Service

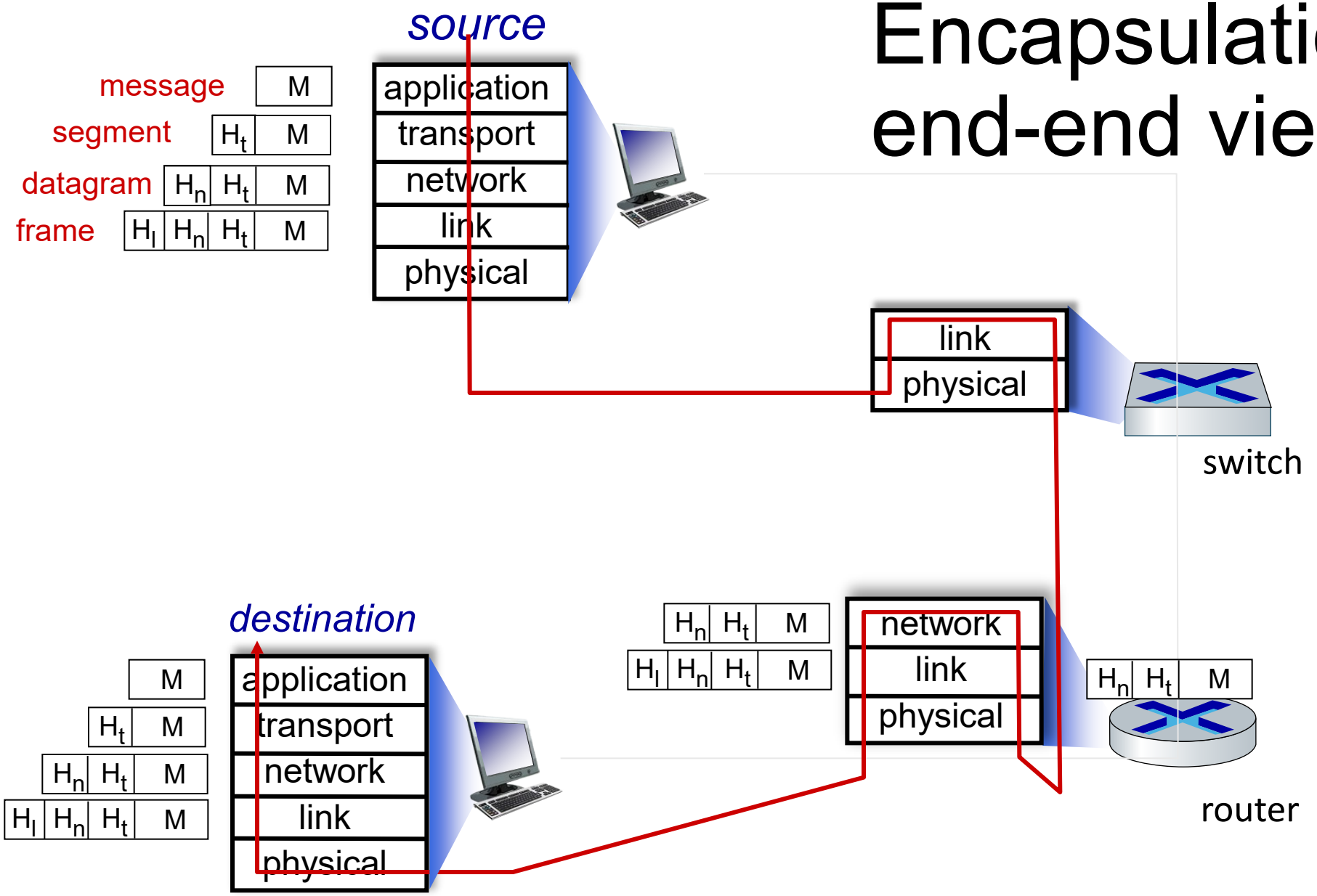


(8) Services, Layering and Encapsulation



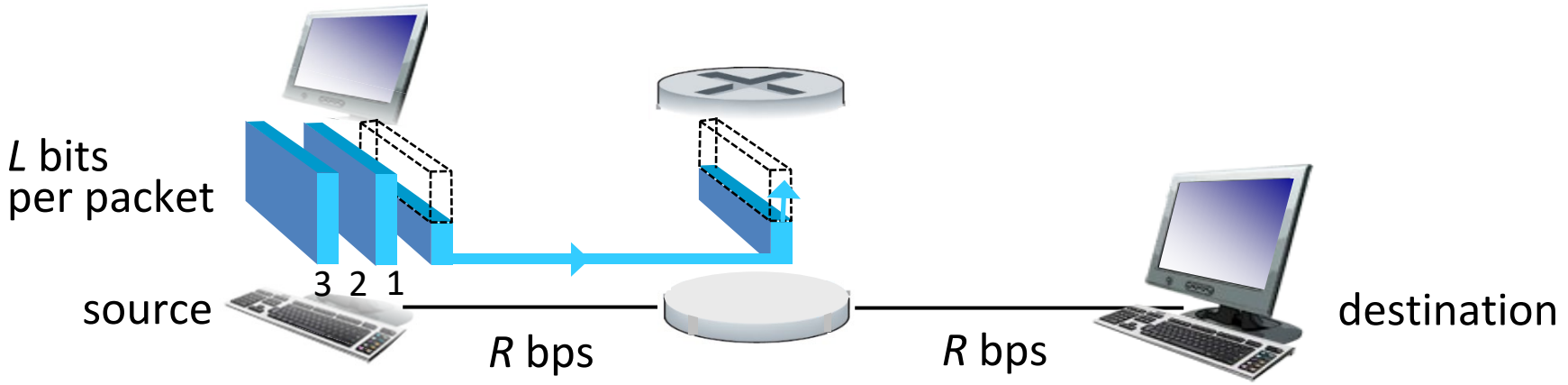
Source: Kurose & Ross

Encapsulation: an end-end view



Source: Kurose & Ross

(9.1) Packet-switching: store-and-forward



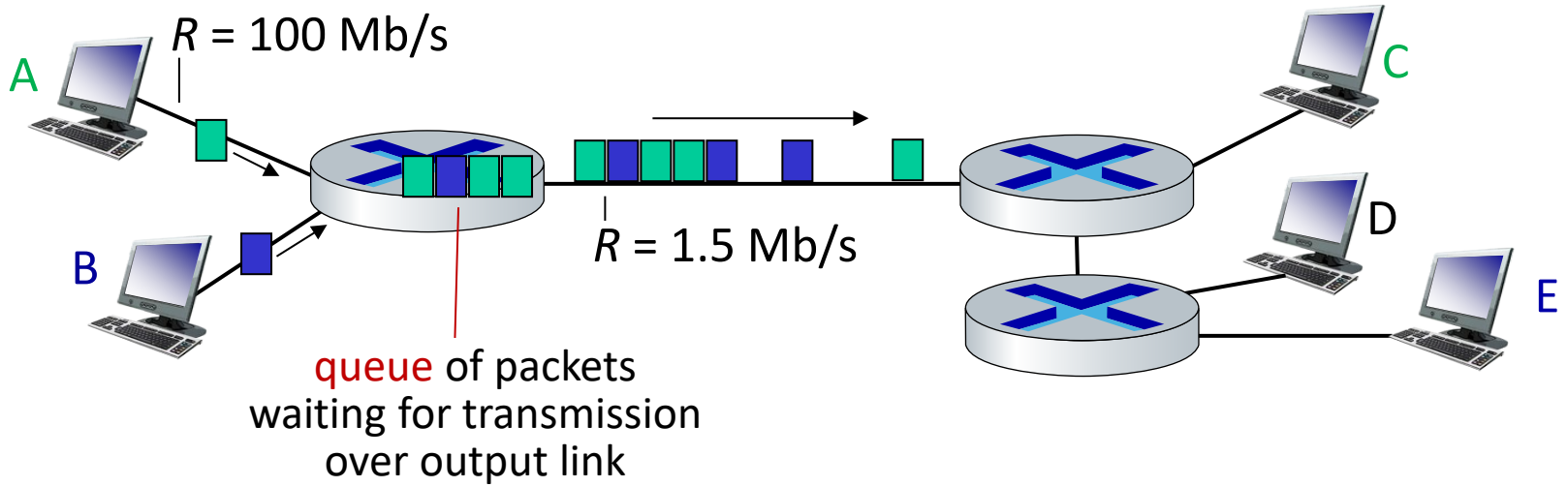
- packet transmission delay: takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward: entire* packet must arrive at router before it can be transmitted on next link

One-hop numerical example:

- *L* = 10 Kbits
- *R* = 100 Mbps
- one-hop transmission delay = 0.1 msec

Source: Kurose & Ross

Packet-switching: queueing



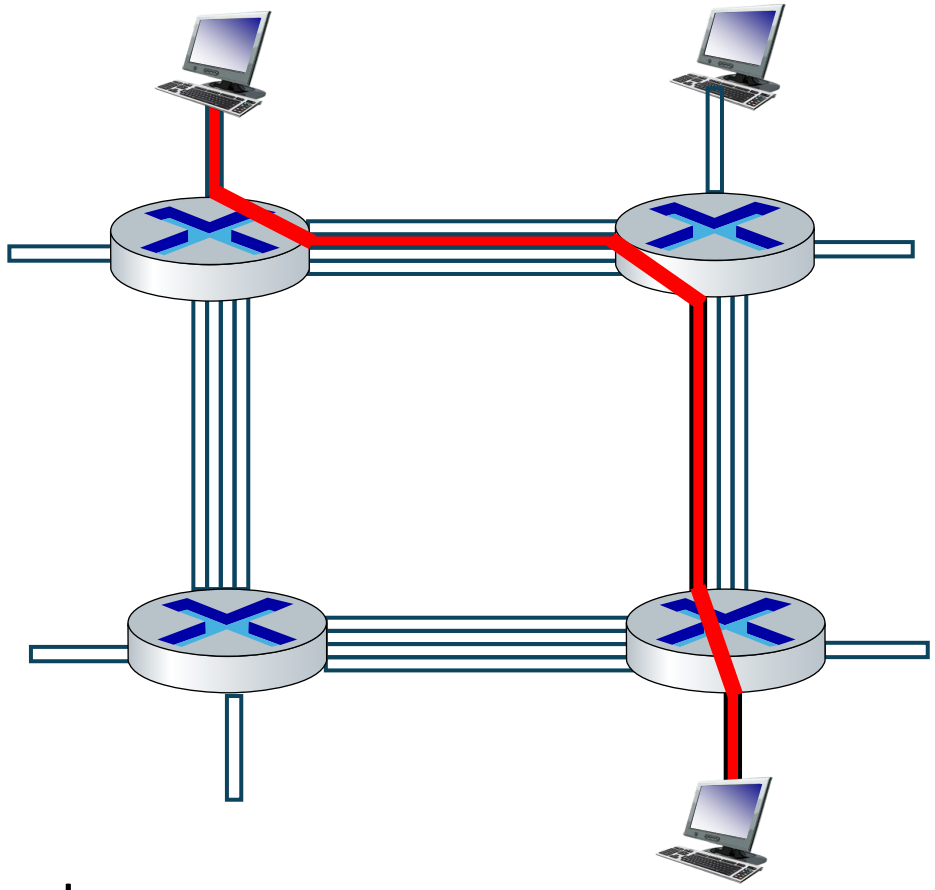
Queueing occurs when work arrives faster than it can be serviced:

Source: Kurose & Ross

(9.2) Alternative to packet switching: circuit switching

end-end resources allocated to, reserved for "call" between source and destination

- in diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (no sharing)
  - commonly used in traditional telephone networks

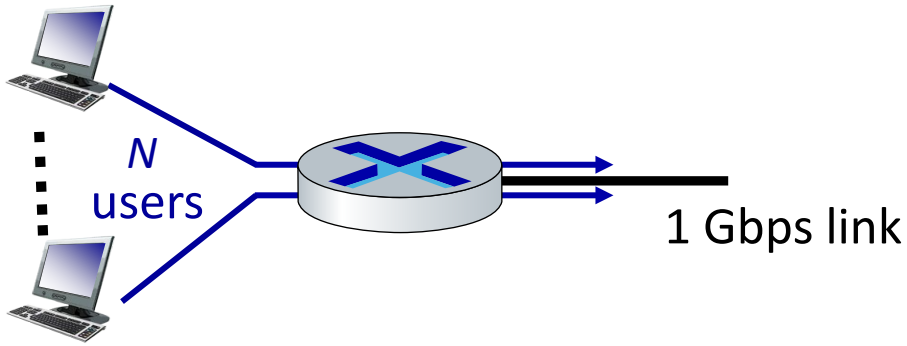


Source: Kurose & Ross

Packet switching versus circuit switching

example:

- 1 Gb/s link
- each user:
  - 100 Mb/s when “active”
  - active 10% of time



Q: how many users can use this network under circuit-switching and packet switching?

- *circuit-switching:* 10 users
- *packet switching:* with 35 users, probability > 10 active at same time is less than .0004 *

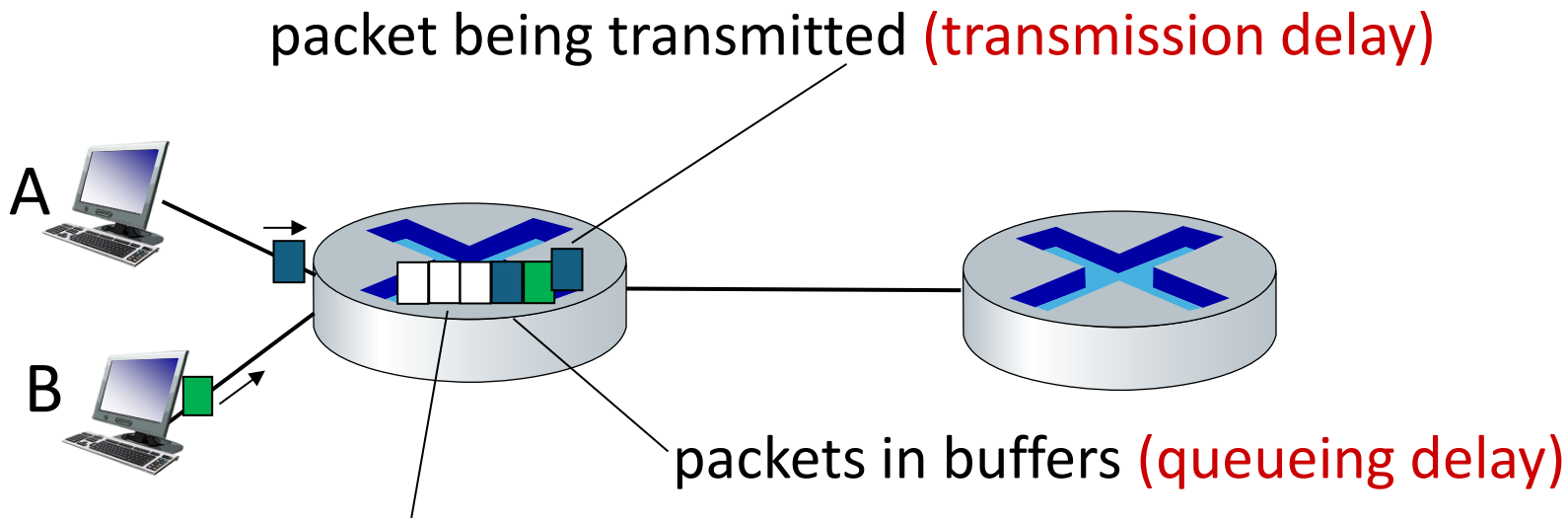
Q: how did we get value 0.0004?

A:

Source: Kurose & Ross

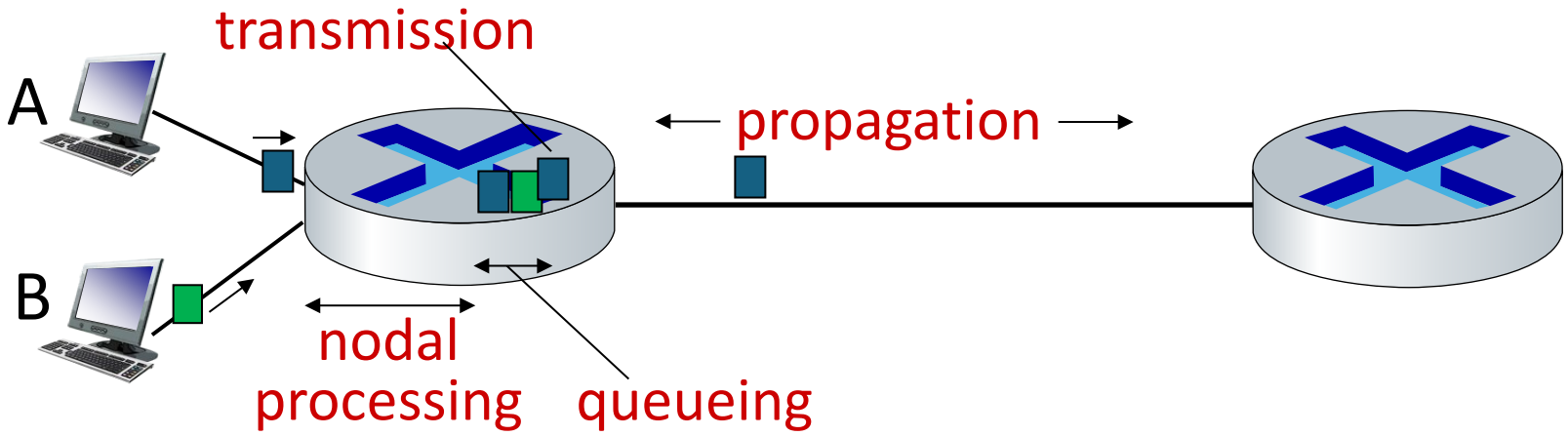
(10) Packet Delay and Loss

- packets *queue* in router buffers, waiting for turn for transmission
  - queue length grows when arrival rate to link (temporarily) exceeds output link capacity
- packet *loss* occurs when memory to hold queued packets fills up



Source: Kurose & Ross

Packet delay: four sources



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{proc}: nodal processing

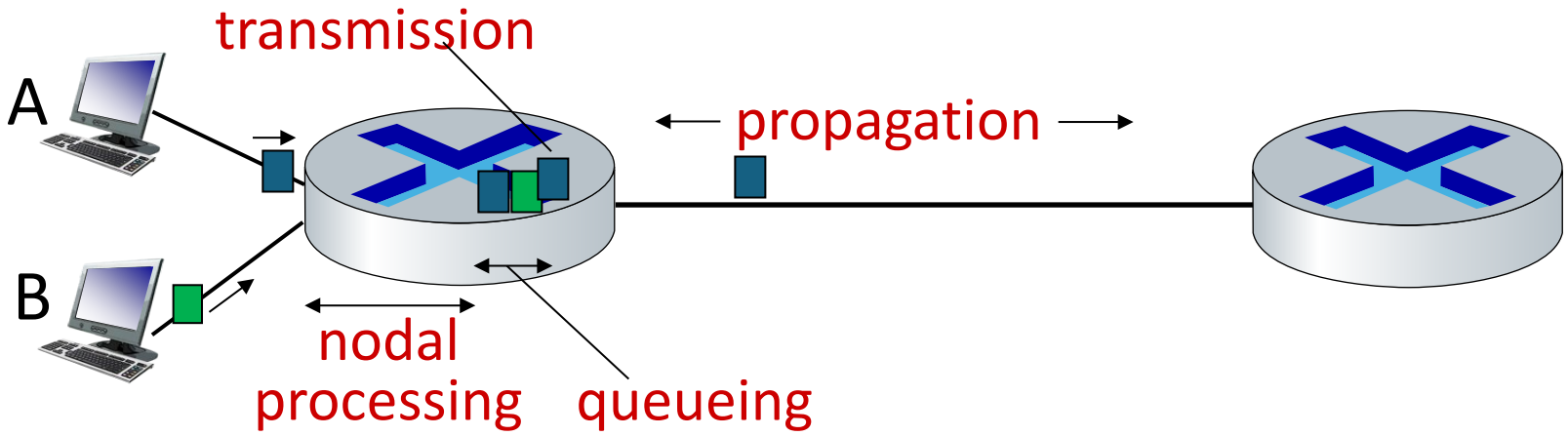
- check bit errors
- determine output link
- typically < microsecs

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Source: Kurose & Ross

Packet delay: four sources



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{trans}: transmission delay:

- L: packet length (bits)
- R: link *transmission rate (bps)*
- $d_{trans} = L/R$

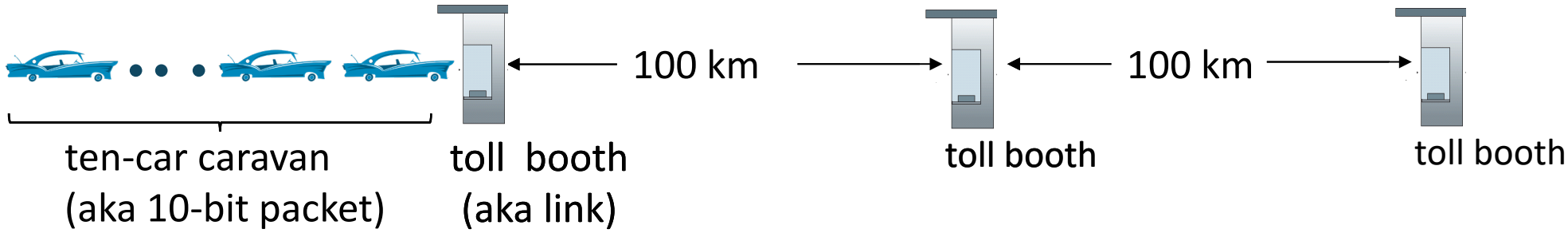
d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed (~2×10^8 m/sec)
- $d_{prop} = d/s$

d_{trans} and d_{prop} *very* different

Source: Kurose & Ross

Caravan analogy



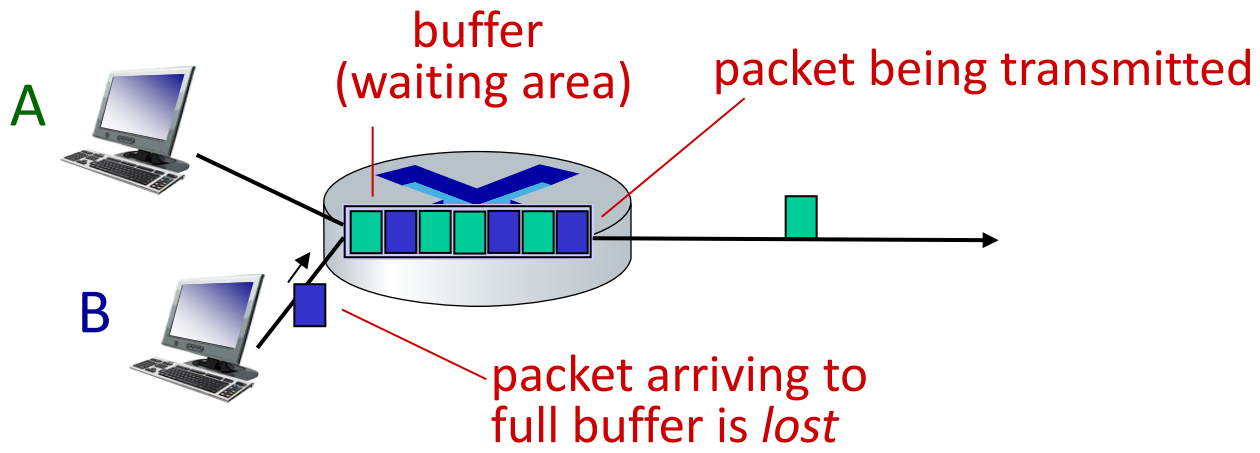
- car ~ bit; caravan ~ packet; toll service ~ link transmission
- toll booth takes 12 sec to service car (bit transmission time)
- "propagate" at 100 km/hr
- *Q:* How long until caravan is lined up before 2nd toll booth?

- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr) = 1 hr
- *A:* 62 minutes

Source: Kurose & Ross

Packet loss

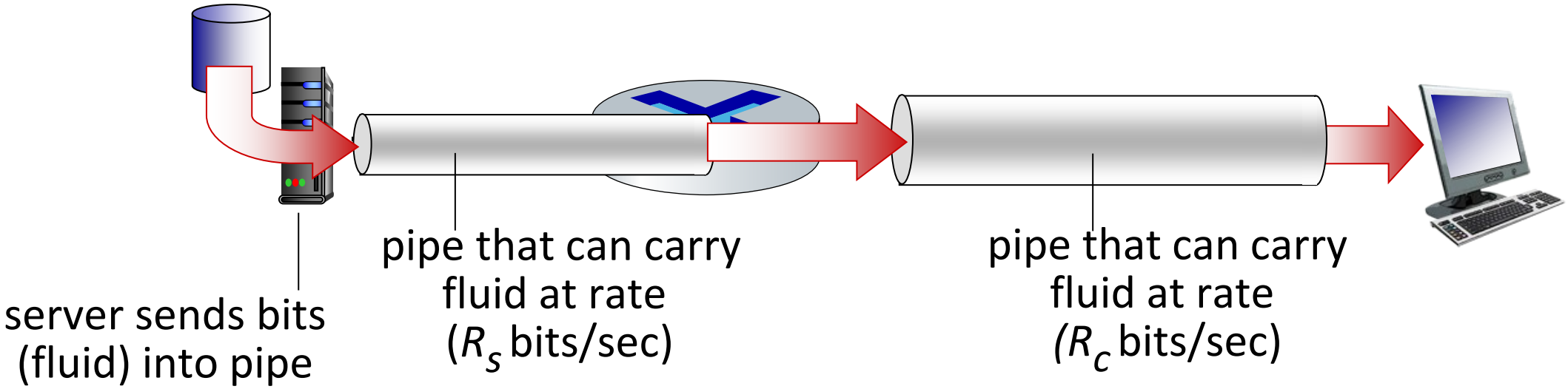
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



Source: Kurose & Ross

Throughput

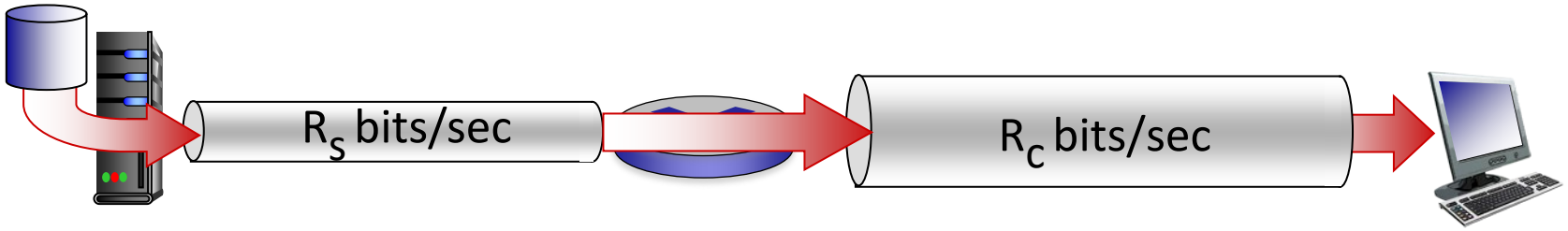
- *throughput:* rate (bits/time unit) at which bits are being sent from sender to receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time



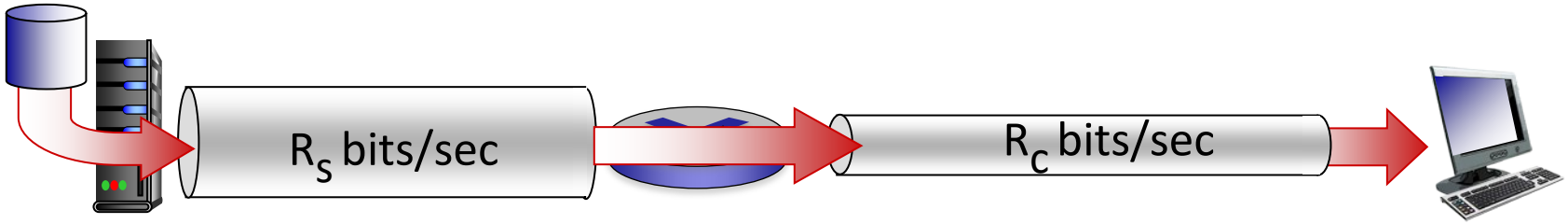
Source: Kurose & Ross

Throughput

$R_s < R_c$ What is average end-end throughput?



$R_s > R_c$ What is average end-end throughput?



bottleneck link

link on end-end path that constrains end-end throughput

Source: Kurose & Ross