



南京大學

NANJING UNIVERSITY



# Computer Networks

Wenzhong Li, Chen Tian

Nanjing University

*Material with thanks to James F. Kurose, Mosharaf Chowdhury, and other colleagues.*



# Chapter 1. Introduction of Networking

- Basic Concepts and Questions
- Internet History
- Protocol Layers and Service Model
- Network Performance



# Internet History

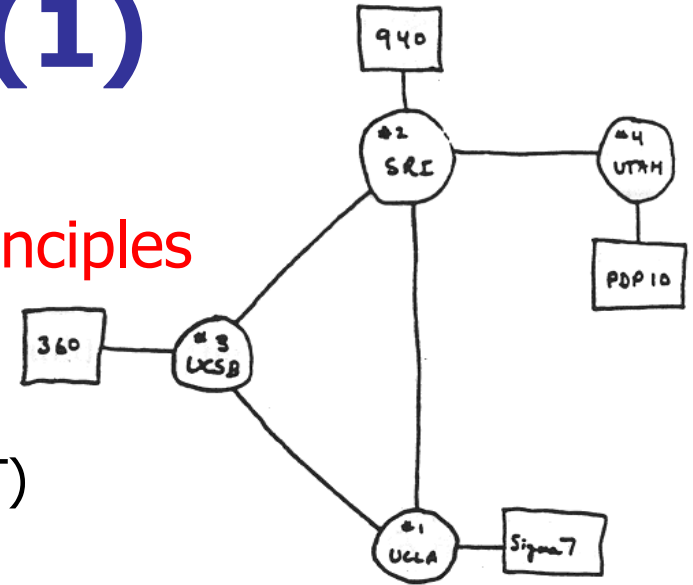


# Internet History (1)

## 1961-1972: Early packet-switching principles

### 60年代：诞生-分组交换网络

- 1961: Kleinrock – queuing theory shows effectiveness of packet-switching (PhD@MIT)
- 1964: Baran – packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational (UCLA, Stanford, UCSB, UTAH), Kleinrock
- 1972:
  - ARPAnet demonstrated publicly
  - NCP (Network Control Protocol) first host-host protocol [RFC001]
  - First email program
  - ARPAnet has 15 nodes



THE ARPA NETWORK



In the Press About Publications History Twitter Students

### The Day the Infant Internet Uttered its First Words

Below is a record of the first message ever sent over the ARPANET. It took place at 22:30 hours on October 29, 1969. This record is an excerpt from the "IMP Log" that was kept at UCLA. Professor Kleinrock was supervising his student/programmer Charley Kline (CSK) and they set up a message transmission to go from the UCLA SDS Sigma 7 Host computer to another programmer, Bill Duvall, at the SRI SDS 940 Host computer. The transmission itself was simply to "login" to SRI from UCLA. They succeeded in transmitting the "l" and the "o" and then the system crashed! Hence, the first message on the Internet was "lo", as in "lo and behold! They were able to do the full login about an hour later."

Leonard Kleinrock	
Born	June 13, 1934 (age 82) New York City
Residence	Los Angeles
Nationality	United States
Fields	Engineering Computer science
Institutions	UCLA

100	LOADED	OP. PROGRAM	CSK	
	FOR	SEN	BARBER	



# Internet History (2)

## 1972-1980: Internetworking, new and proprietary nets

70年代：成型 单一、封闭网络 -> 开放互连网络

- 1970: ALOHAnet satellite network in Hawaii, Norman Abramson (无线分组网络)
- 1973: Robert Metcalfe's PhD thesis (@Harvard) proposes Ethernet (以太网), at Xerox PARC in 1976 (局域网诞生)
- 1974: Cerf and Kahn – architecture for interconnecting networks (Internet构架)
- Late70's:
  - Proprietary architectures: DECnet, SNA, XNA
  - Switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Vint Cerf, Robert E. Kahn and George W. Bush



- Cerf and Kahn's internetworking principles:
  - Minimalism, autonomy – no internal changes required to interconnect networks
  - Best effort service model
  - Stateless routers
  - Decentralized control
- Define today's Internet architecture
- Design of TCP/IP suits



# Internet History (3)

## 1980-1990: new protocols, a proliferation of networks

### 80年代：持续发展

- 新协议： **NCP**-> **TCP/IP**
- **DNS**：实现域名解析
- 应用： **Email**, **Ftp**

- 1983: deployment of TCP/IP
- 1982: SMTP email protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: FTP protocol defined
- 1988: TCP congestion control
- New national networks: Cset, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks



# Internet History (4)

## 1990's, 2000's: commercialization, the Web, new apps

### 90年代：因特网爆炸

- 万维网出现：**www** (**http**, **HTML**, **Web Server**, **Browser**)
- 商用化，逐渐普及
- 新型应用：**Email**, **Web**, **IM** (**instant messaging**)，**MP3**文件共享

- **Early 1990's:** ARPAnet decommissioned
- **1991:** NSF lifts restrictions on commercial use of NSFnet (decommissioned in 1995)
- **Early 1990's:** Web
  - Hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
- **1994:** Mosaic, later Netscape Browser

Late 1990's: commercialization of the Web

Late 1990's ~ 2000's:

- More killer apps: instant messaging, peer2peer file sharing (e.g. Napster)
- Network security to forefront
- Est. 50 million host, 100 million+ users
- Backbone links running at Gbps

蒂姆·伯纳斯-李爵士  
Sir Tim Berners-Lee



出生 1955年6月8日 (61歲)<sup>[1]</sup>  
 + 英格兰伦敦  
 机构 万维网联盟  
 南安普敦大学  
 Plessey  
 麻省理工学院  
 知名于 发明万维网  
 麻省理工学院计算机科学及人工智能实  
 验室创办主席

**2016 Turing Award**



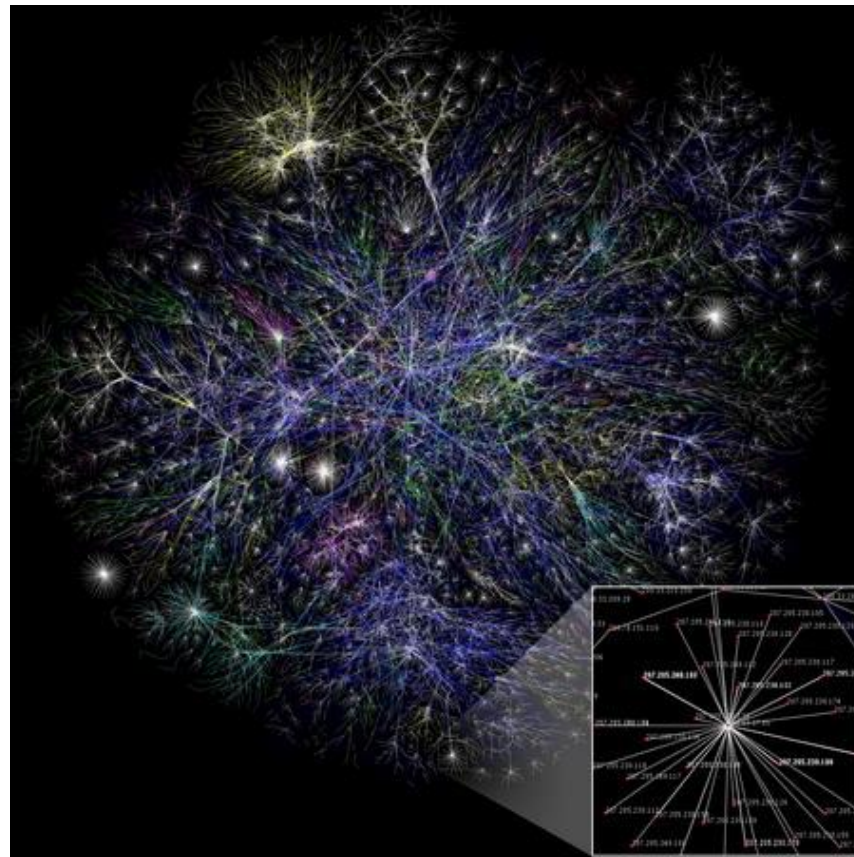
# Internet History (5)

## 2000年以后，新型应用涌现

- 多媒体
- **P2P**网络
- 社交网络 (**Facebook**, **Twitter**, 人人, 微博, 微信, ...)

## 2007

- ~500 million hosts
- Voice, Video over IP
- **P2P applications**: BitTorrent (file sharing), Skype (VoIP), PPLive (video)
- More applications: YouTube, online gaming
- Wireless and mobility
- 2015- , blockchain, AINet, 5G,



# THE INTERNET AGE

中央电视台大型电视纪录片

# 互联网时代



CCTV2  
财经

央视网  
CCTV.com 经济

首页

边看边聊

主创团队

开播仪式回顾

大调查启动回顾

采访嘉宾

CCTV2

经济频道



## 点播

《互联网时代》第十集 眺望

《互联网时代》第九集 世界

《互联网时代》第八集 忧虑

《互联网时代》第七集 控制

《互联网时代》第六集 迁徙

《互联网时代》第五集 崛起

《互联网时代》第四集 再构

《互联网时代》第三集 能量

《互联网时代》第二集 浪潮

《互联网时代》第一集 时代



# Protocol Layers and Service Model



# Protocol “layers”

*Networks are complex,  
with many “pieces” :*

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

*Question:*

is there any hope of  
*organizing* structure  
of network?

.... or at least our  
discussion of  
networks?



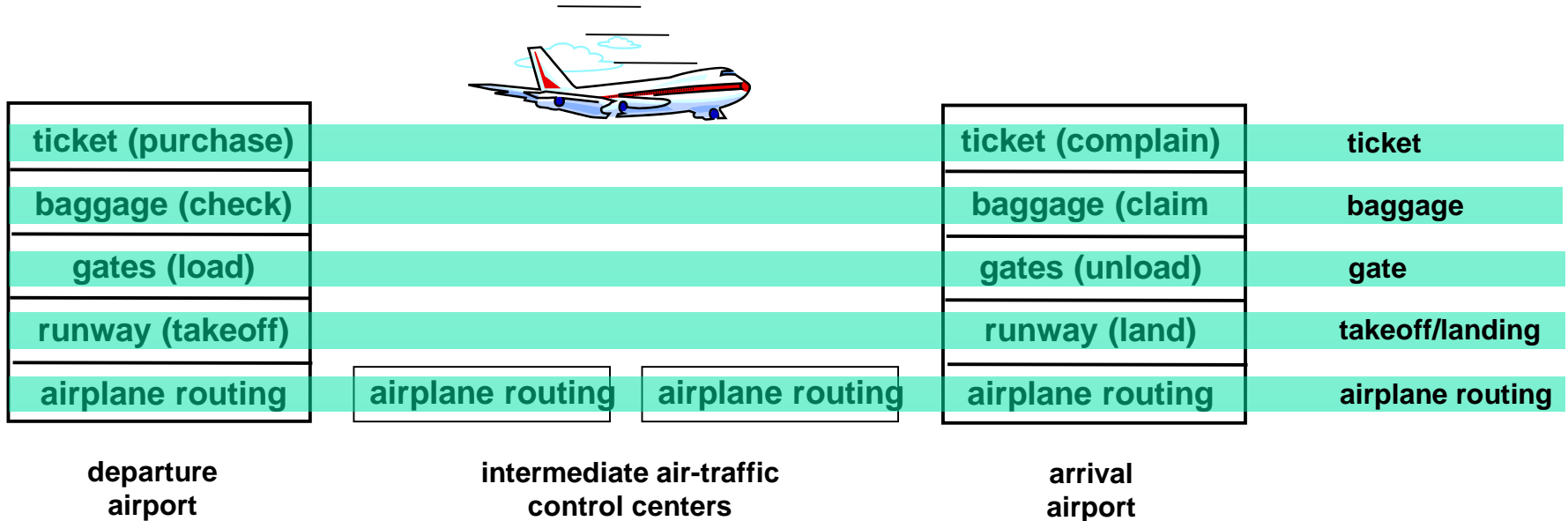
# Organization of air travel



- a series of steps



# Layering of airline functionality



*layers:* each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below



# Why layering?

dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system



# Standard Protocol Architectures

- Two standards:
  - OSI Reference model
    - Never lived up to early promises
  - TCP/IP protocol suite
    - Most widely used
  
- Others
  - IBM Systems Network Architecture (SNA)
  - DECNet, Netware



# ISO-OSI

- Open Systems Interconnection (OSI)
- Developed by the **International Organization for Standardization (ISO)**
- Seven layers structure
  
- A theoretical system delivered **too late**
- TCP/IP is the de facto standard now



# OSI – The Model

- A layer model, and flow structure
- Each layer **performs a subset** of the required communication functions
- Each layer **relies on the next lower layer** to perform more primitive functions
- Each layer **provides services** to the next higher layer
- **Changes** in one layer should not require changes in other layers



# OSI Layers

**Example: Alice invite Bob to lunch**



"请客吃饭"

语言表达

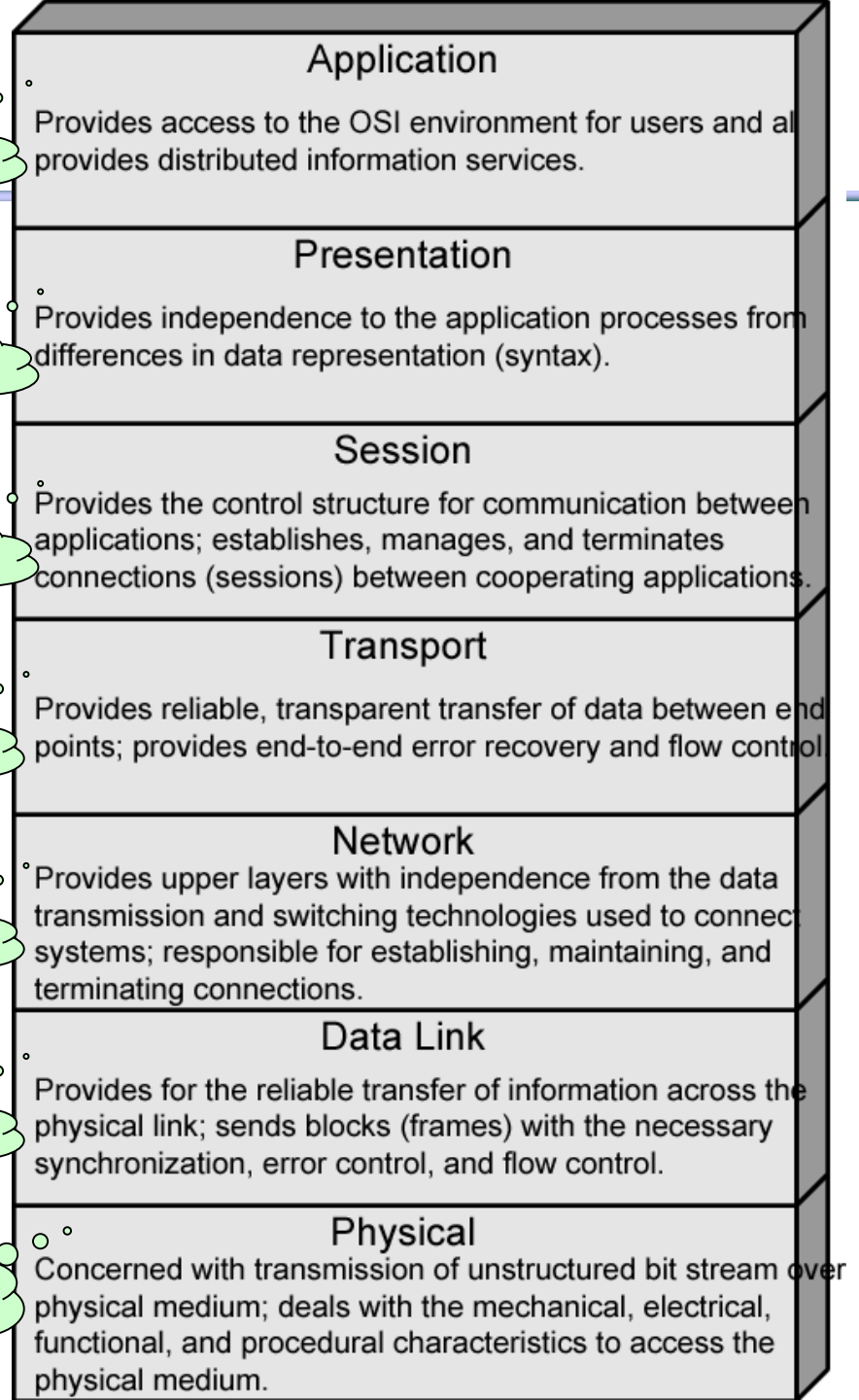
听说同步

摘机拨号

PBX 中转

信号传输

插口、双绞线





# Physical Layer



- Transfers bits across link
- Specification of the **physical aspects** of a comm link
  - **Mechanical**: cable, plugs, pins...
  - **Electrical/optical**: modulation, signal strength, voltage levels, bit times, ...
  - **Functional/procedural**: activate, maintain, deactivate physical links...
- **Physical interface** between devices
  - Ethernet, DSL, cable modem, telephone modems, ...
  - Twisted-pair cable, coaxial cable, optical fiber, radio, infrared, ...



# Data Link Layer

- Groups bits into **frames**
- Activation, maintenance, & deactivation of data link **connections**
- **Transfers** frames across direct connections
- **Medium access control** for local area networks
- **Detection** of bit errors; **Retransmission** of frames
- End-to-end **flow control**
- Higher layers may assume **error free transmission**



# Network Layer

- Transfers packets across **multiple links / multiple networks**
- **Addressing** must scale to large networks
- Nodes jointly execute **routing** algorithm to determine paths across the network
- **Forwarding** transfers packet across a node
- **Congestion control** to deal with traffic surges
- **Connection** setup, maintenance, and teardown when connection-based



# Transport Layer

- Exchange of data **between end systems**
  - Transfers data end-to-end from process in one host to process in another host
- **Reliable** stream transfer or quick-and-simple single-block transfer
  - Error free
  - In sequence
  - No losses
  - No duplicates
- **Connection** setup, maintenance, and release



# Upper Layers

## ■ Session

- Control of dialogues between applications
- Dialogue discipline
- Grouping data
- Checkpoint recovery

Incorporated into  
Application Layer Now

## ■ Presentation

- Machine-independent representation of data
- Data formats and coding
- Data compression & encryption

## ■ Application

- Means for applications to access OSI environment

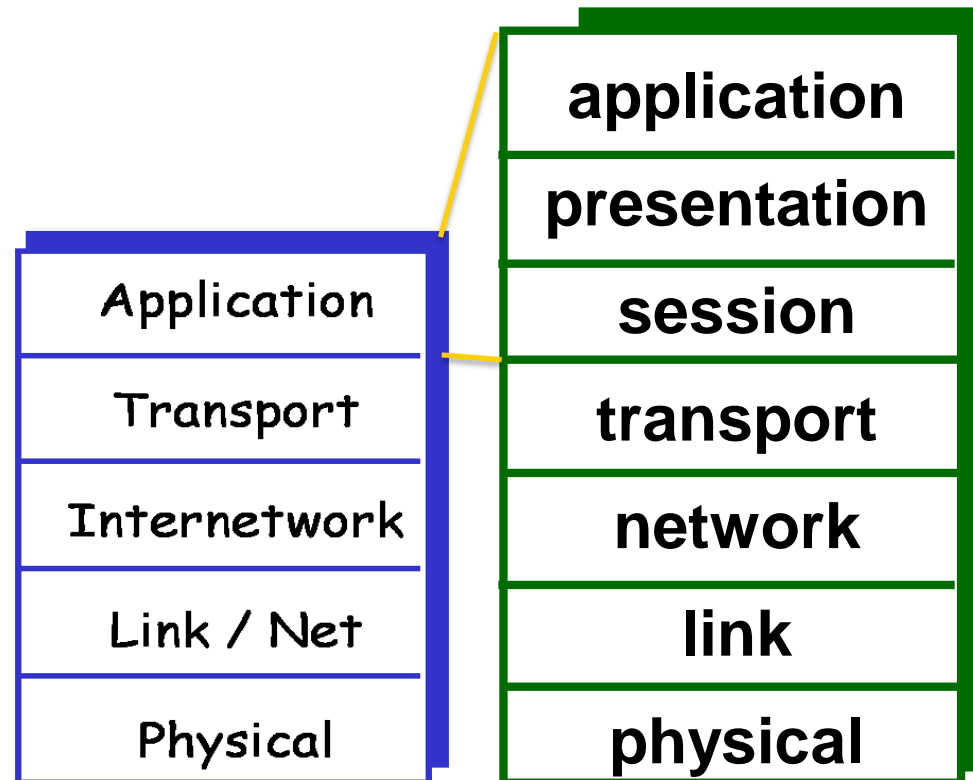


# TCP/IP Protocol Architecture

Used by the global **Internet**

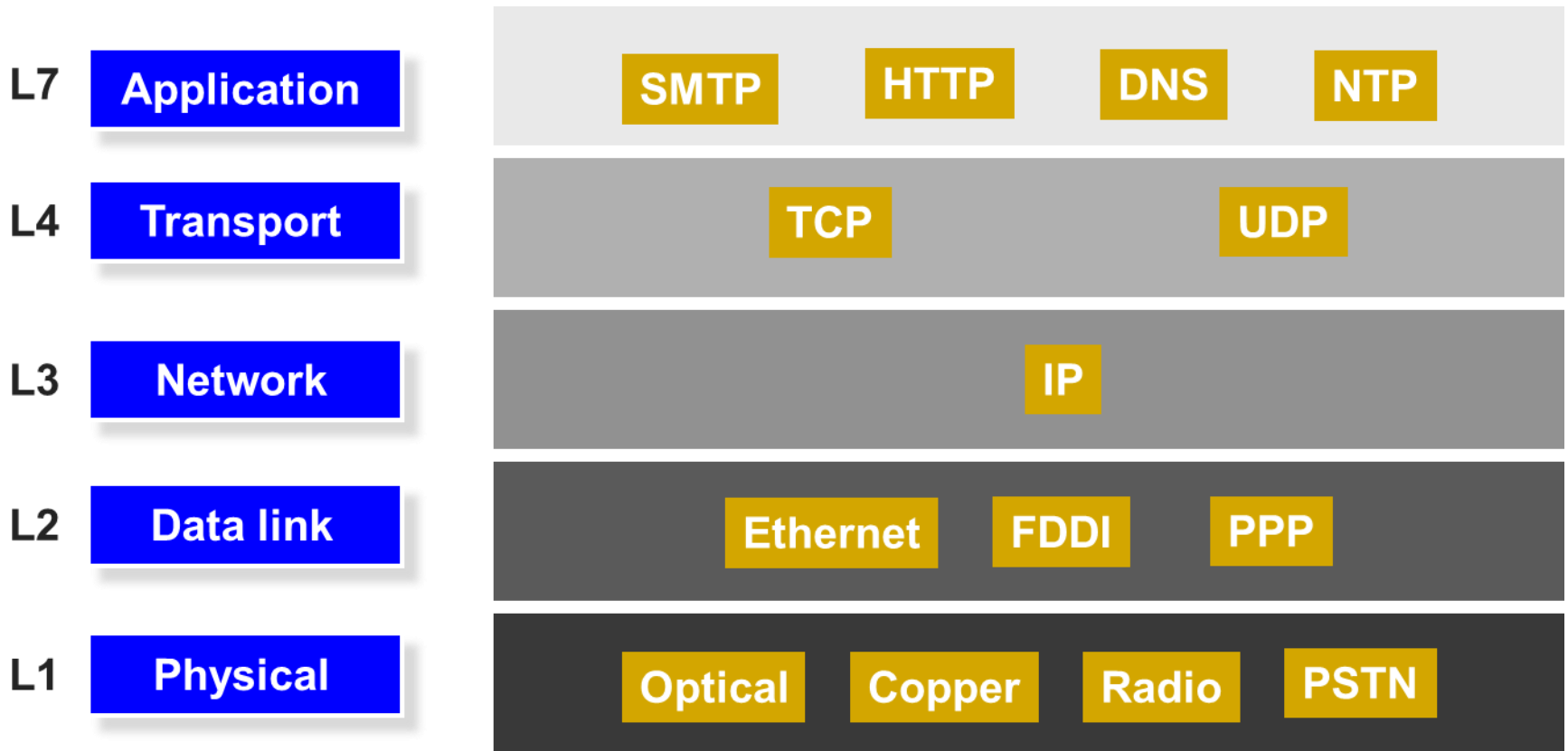
- **Application:** supporting network applications
  - FTP, SMTP, HTTP
- **Transport:** process-process data transfer
  - TCP, UDP
- **Internetwork:** routing of datagrams across net of nets
  - IP, routing protocols
- **Link:** data transfer between neighboring routers / hosts
  - PPP, Ethernet
- **Physical:** bits “on the wire”

## TCP/IP protocol stack vs. OSI



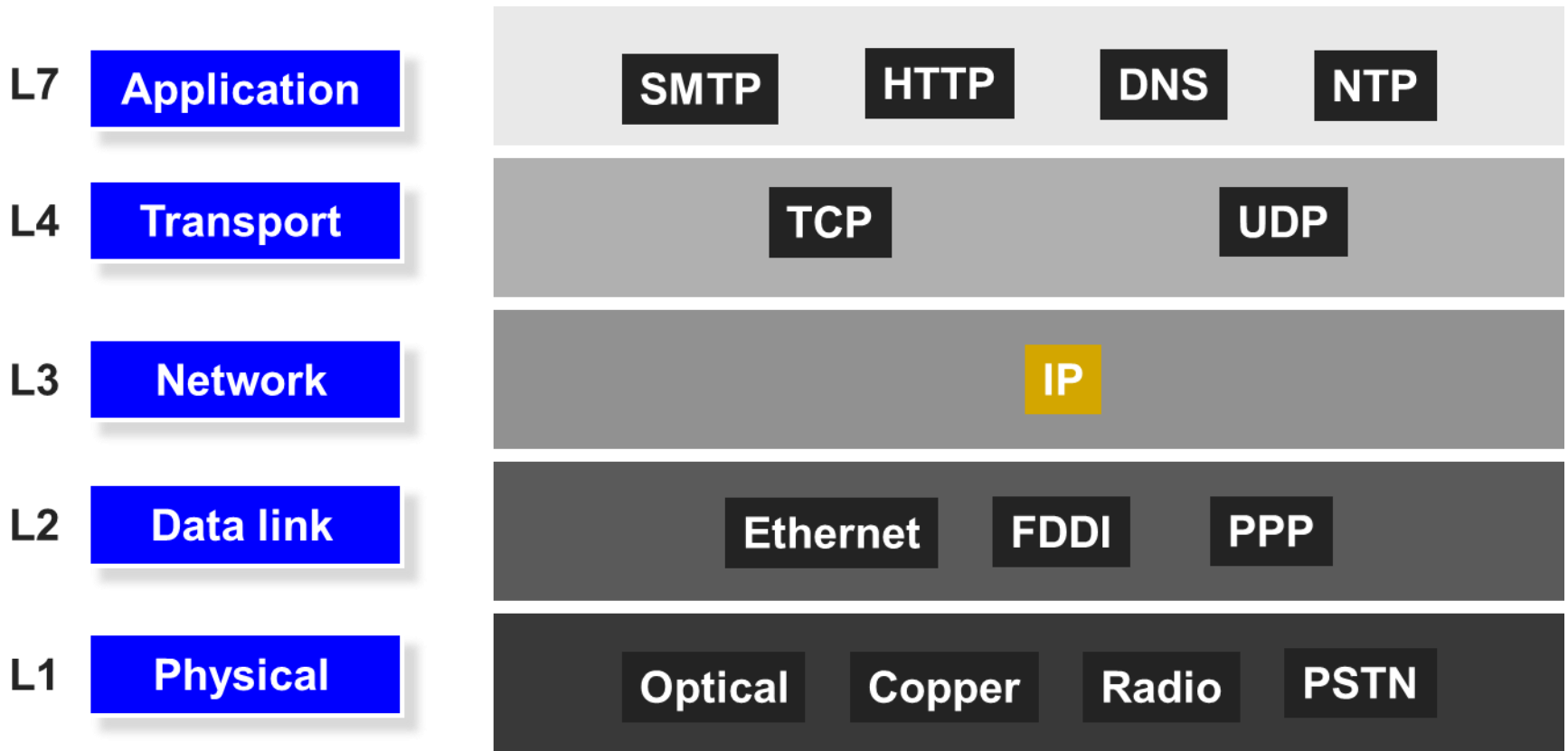


# Protocols at different layers



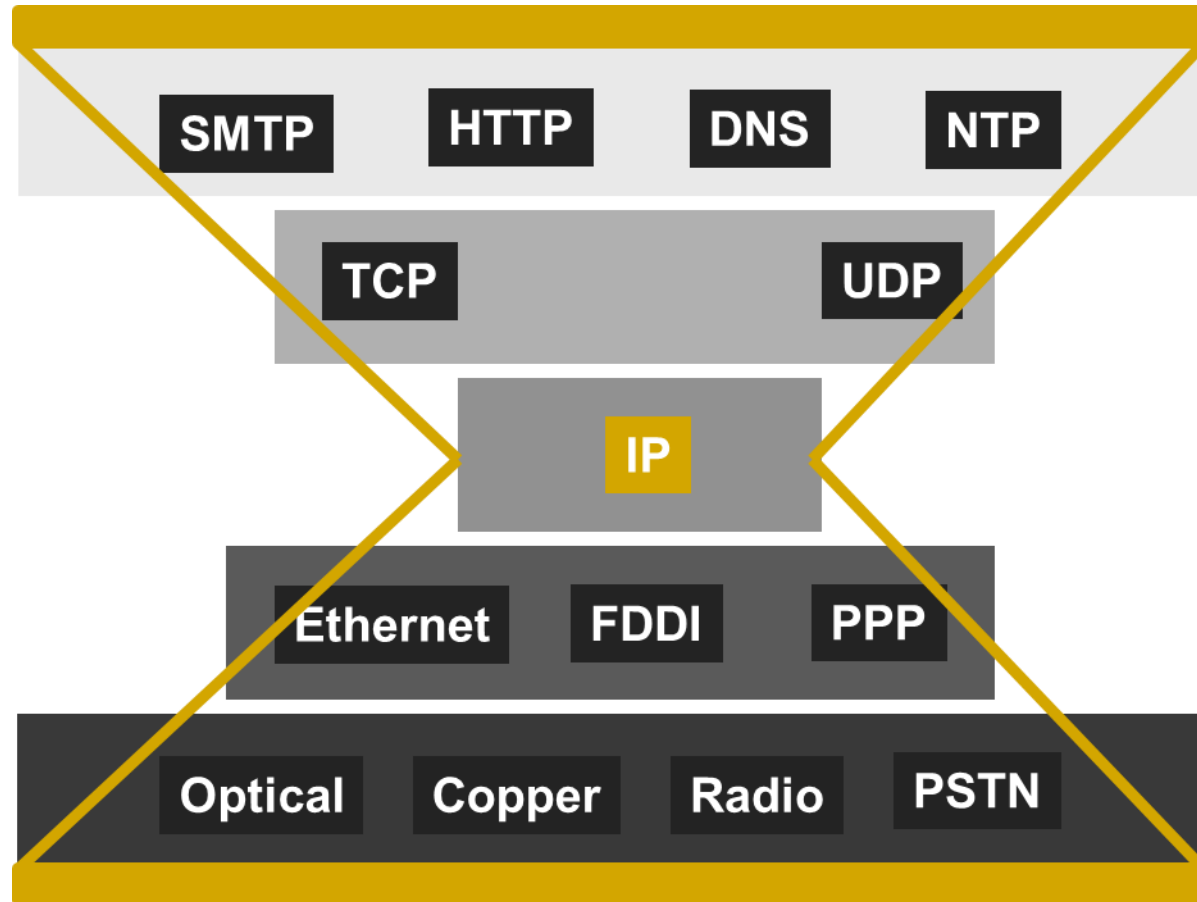


# ONE network layer protocol





# IP is the narrow waist of the layering hourglass



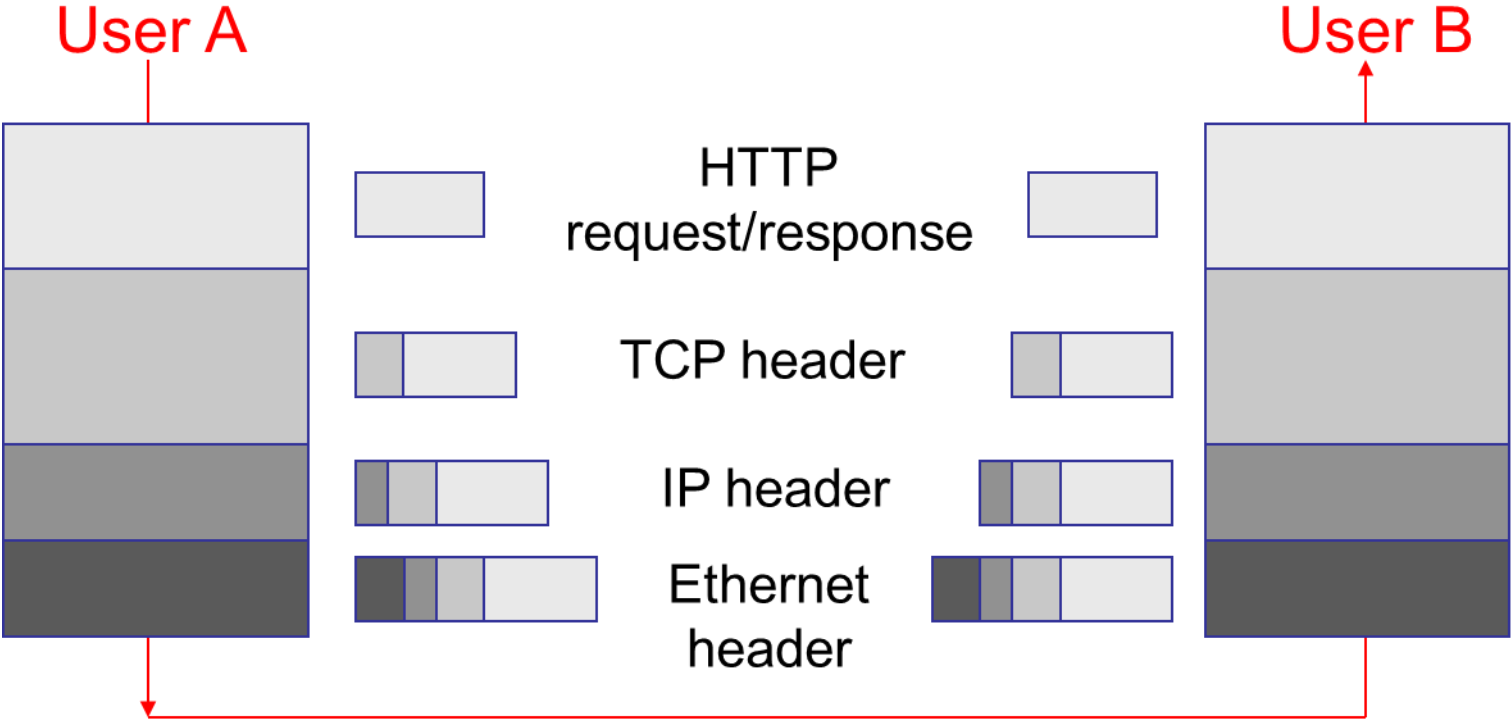


# Implications of hourglass

- Single network-layer protocol (IP)
- Allows arbitrary networks to interoperate
  - Any network that supports IP can exchange packets
- **Decouples** applications from low-level networking technologies
  - Applications function on all networks
- Supports simultaneous innovations above and below IP
- But changing IP itself is hard (e.g., IPv4 → IPv6)



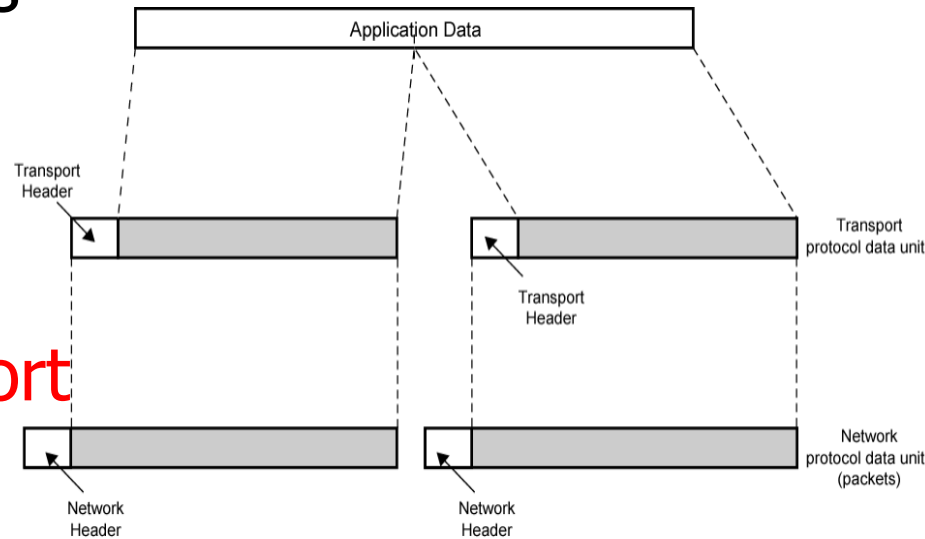
# Layer encapsulation: Protocol headers





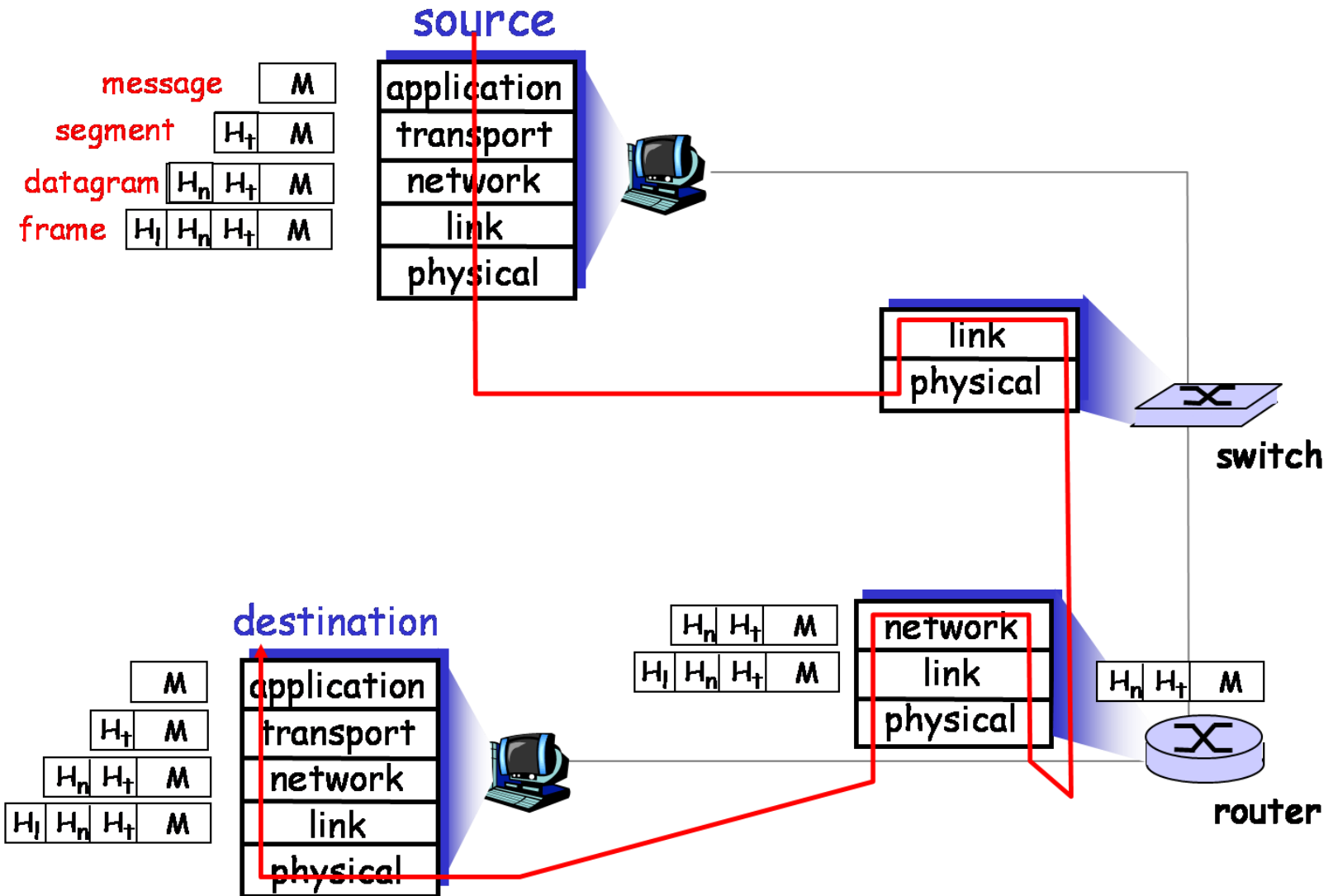
# Protocol Data Units

- At each layer, **Control info** is added to **user data** to ease communication, e.g.
- Transport layer segments application data
- Each segment has **a transport header** added
  - Destination port
  - Sequence number
  - Error detection code
- This gives a **transport protocol data unit (PDU)**





# Encapsulation





# Pros and cons of layering

## Why layers?

- Reduce complexity
- Improve flexibility

## Why not?

- Higher overheads
- Cross-layer information often useful



# Network Performance



# How to evaluate the performance of a network?



# Performance metrics

- Delay
- Loss
- Throughput



# Delay

- How long does it take to send a packet from its source to destination?



# Delay

- Consists of four components

- Transmission delay
- Propagation delay
- Queuing delay
- Processing delay

due to link properties

due to traffic mix and  
switch internals



# A network link

**bandwidth**



**delay x bandwidth**

**Propagation delay**

- Transmission delay (传输时延)
  - How long does it take to push all the bits of a packet into a link?
  - Packet size / Transmission rate of the link
    - e.g., 1000 bits / 100 Mbits per sec =  $10^{-5}$  sec
- Propagation delay
  - How long does it take to move one bit from one end of a link to the other?
  - Link length / Propagation speed of link
    - E.g., 30 kilometers /  $3 \times 10^8$  meters per sec =  $10^{-4}$  sec



# Packet delay

## Sending a 100-byte packet



**Time to transmit**

**Time to transmit  
800 bits =  $800 \times 1/10^6$ s**

**100Byte packet**

**Time when that  
bit reaches B  
=  $1/10^6 + 1/10^3$ s**

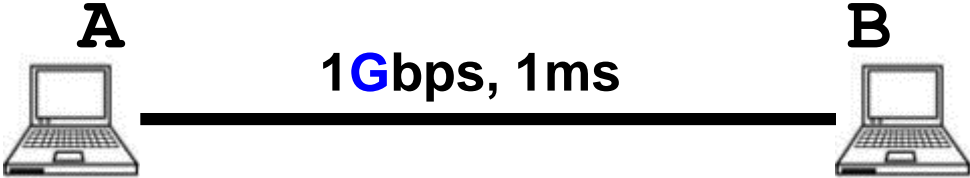
**Time**

**The last bit  
reaches B at  
 $(800 \times 1/10^6) + 1/10^3$ s  
= 1.8ms**



# Packet delay

## Sending a 100-byte packet



**Time to transmit**

**Time to transmit  
800 bits =  $800 \times 1/10^9$ s**

**100Byte packet**

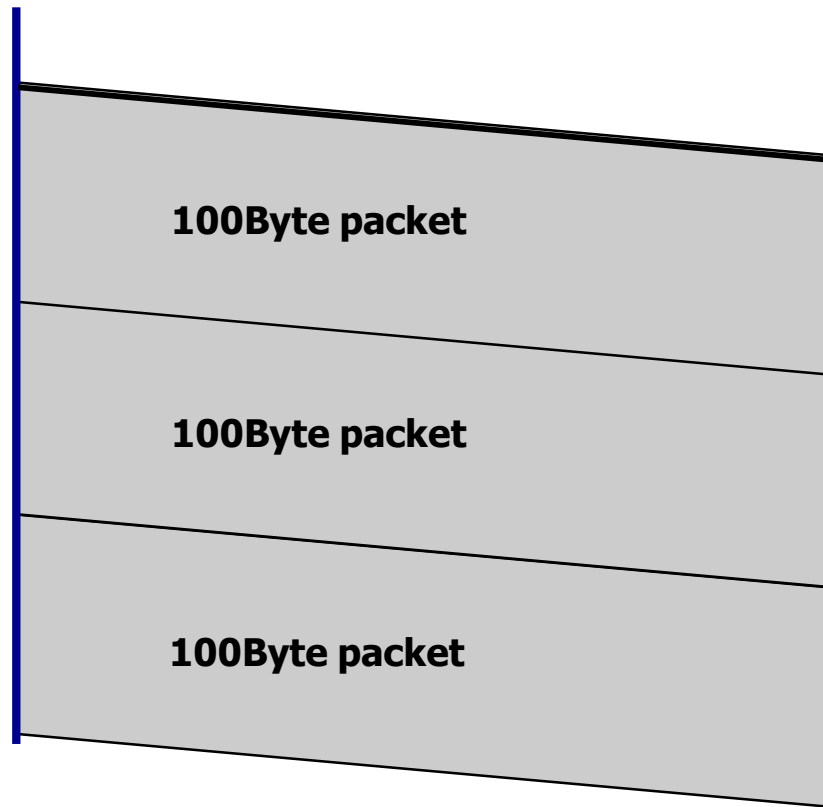
**Time when that  
bit reaches B  
=  $1/10^9 + 1/10^3$ s**

**The last bit  
reaches B at  
 $(800 \times 1/10^9) + 1/10^3$ s  
= 1.0008ms**

**Time**  
↓

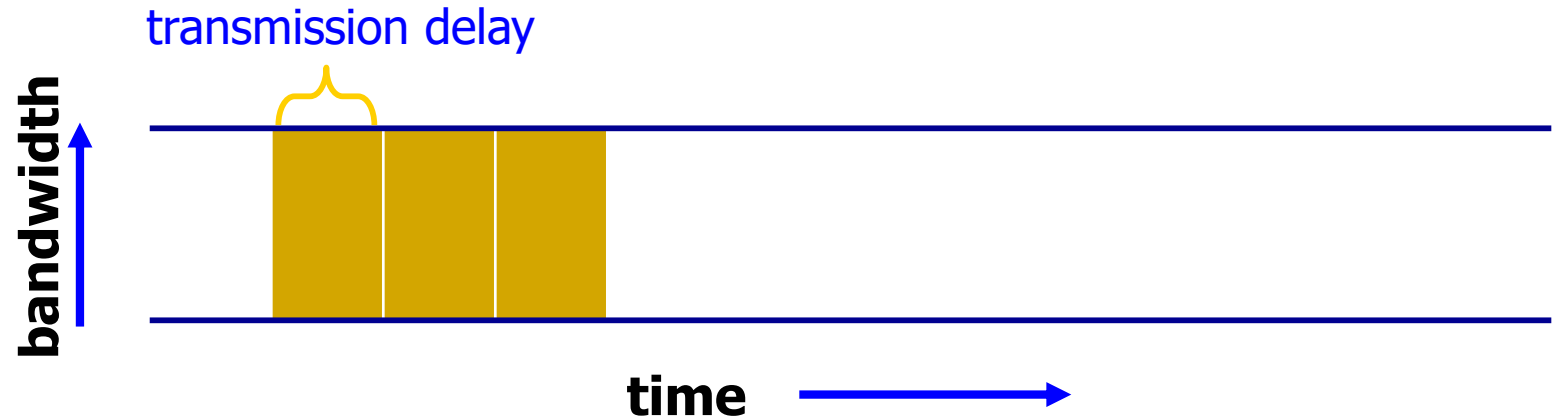


# Sending a large file using 100-byte packets





# Pipe view of a link



- Transmission delay decreases as bandwidth increases

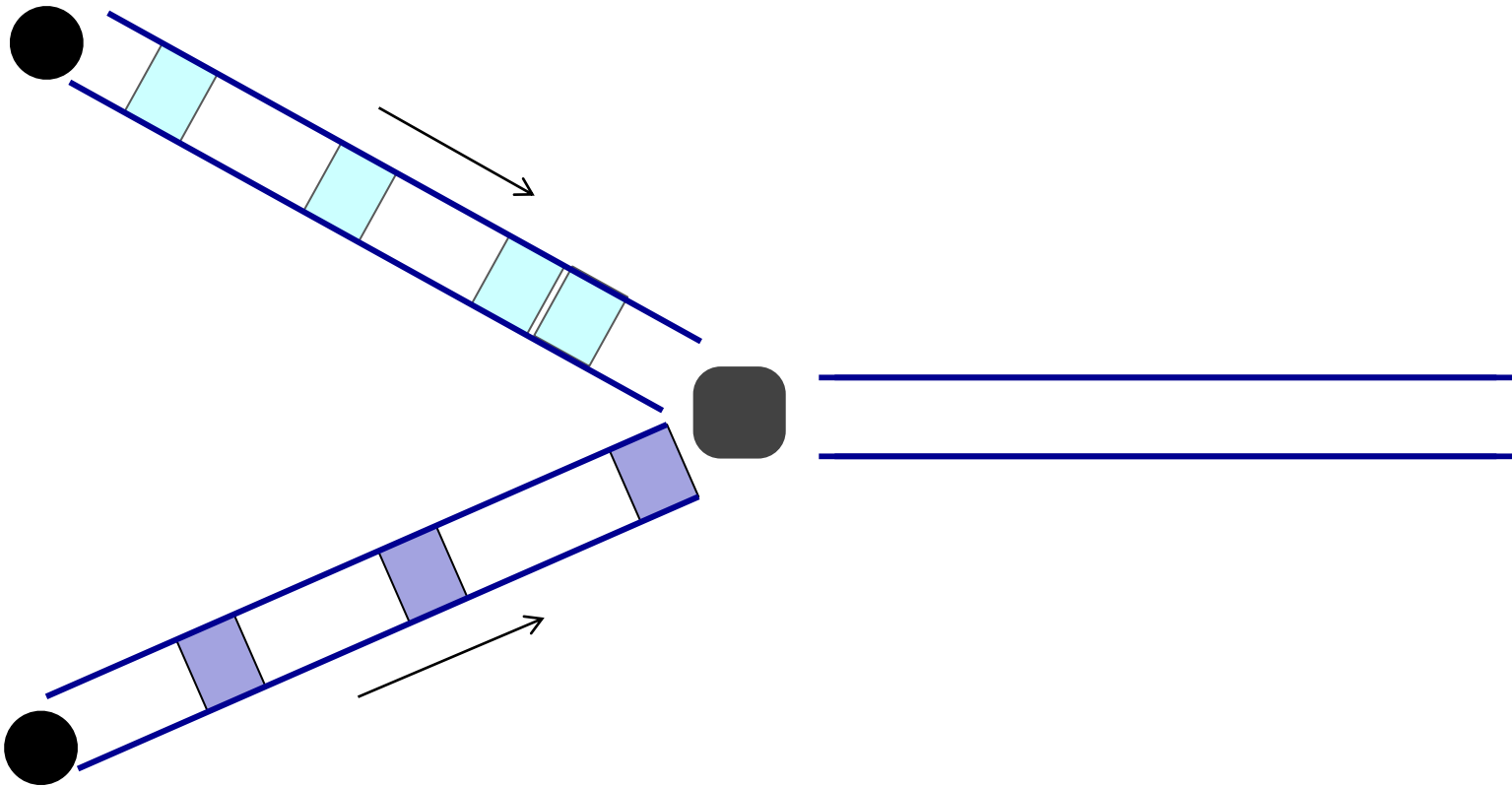


# Queuing delay

- How long does a packet have to sit in a buffer before it is processed?

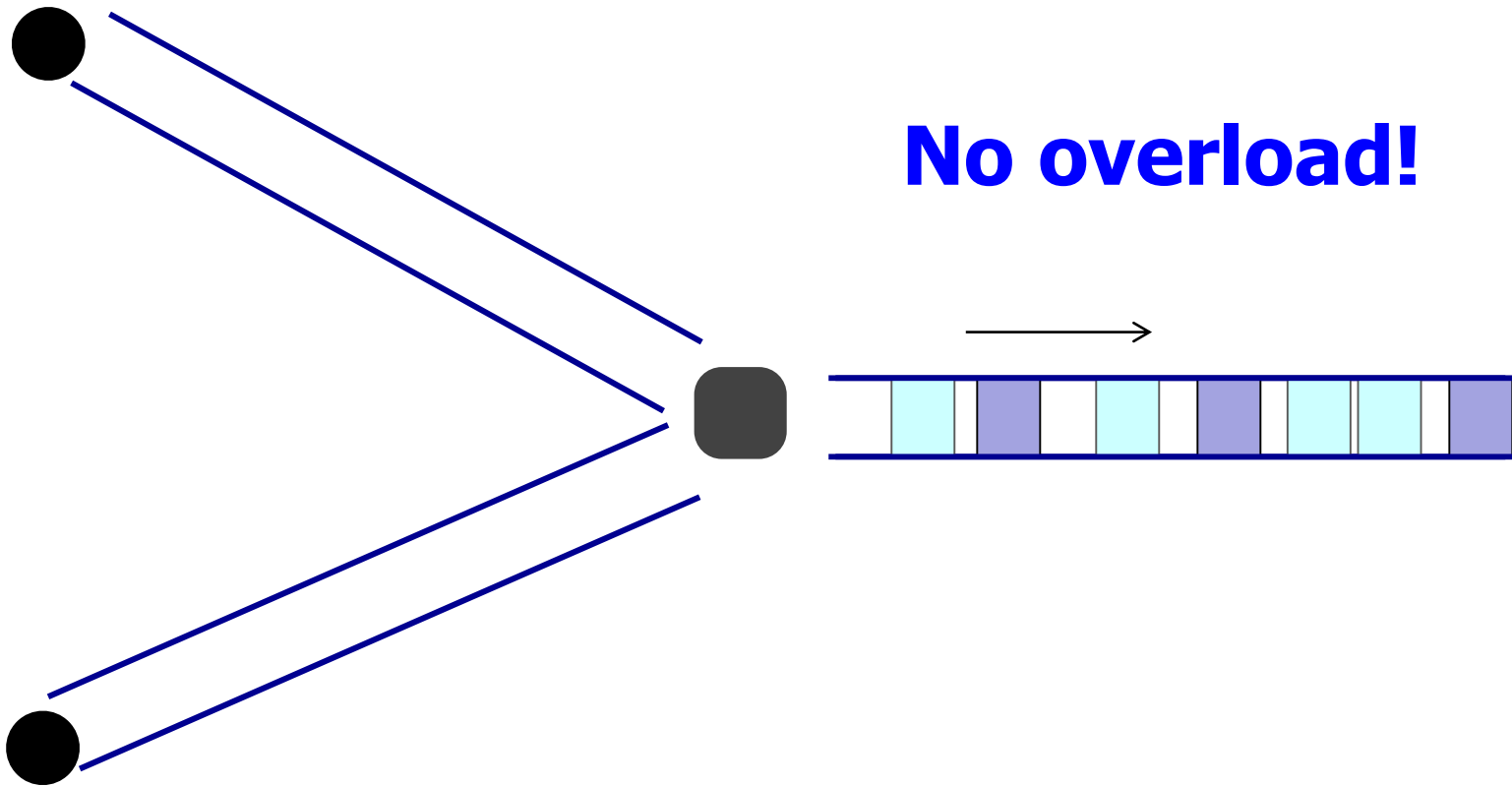


# Queueing delay: "pipe" view



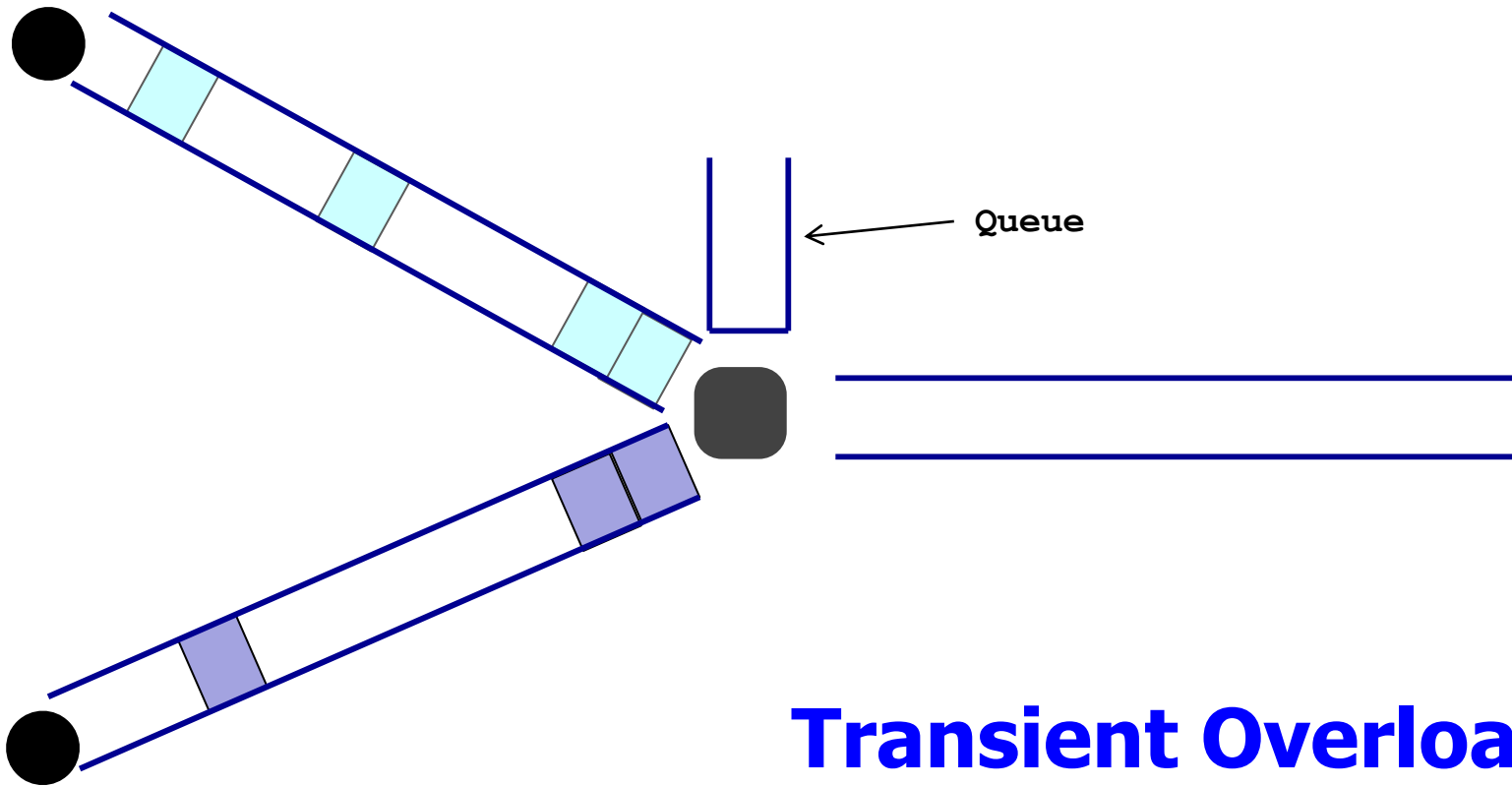


# Queueing delay: "pipe" view





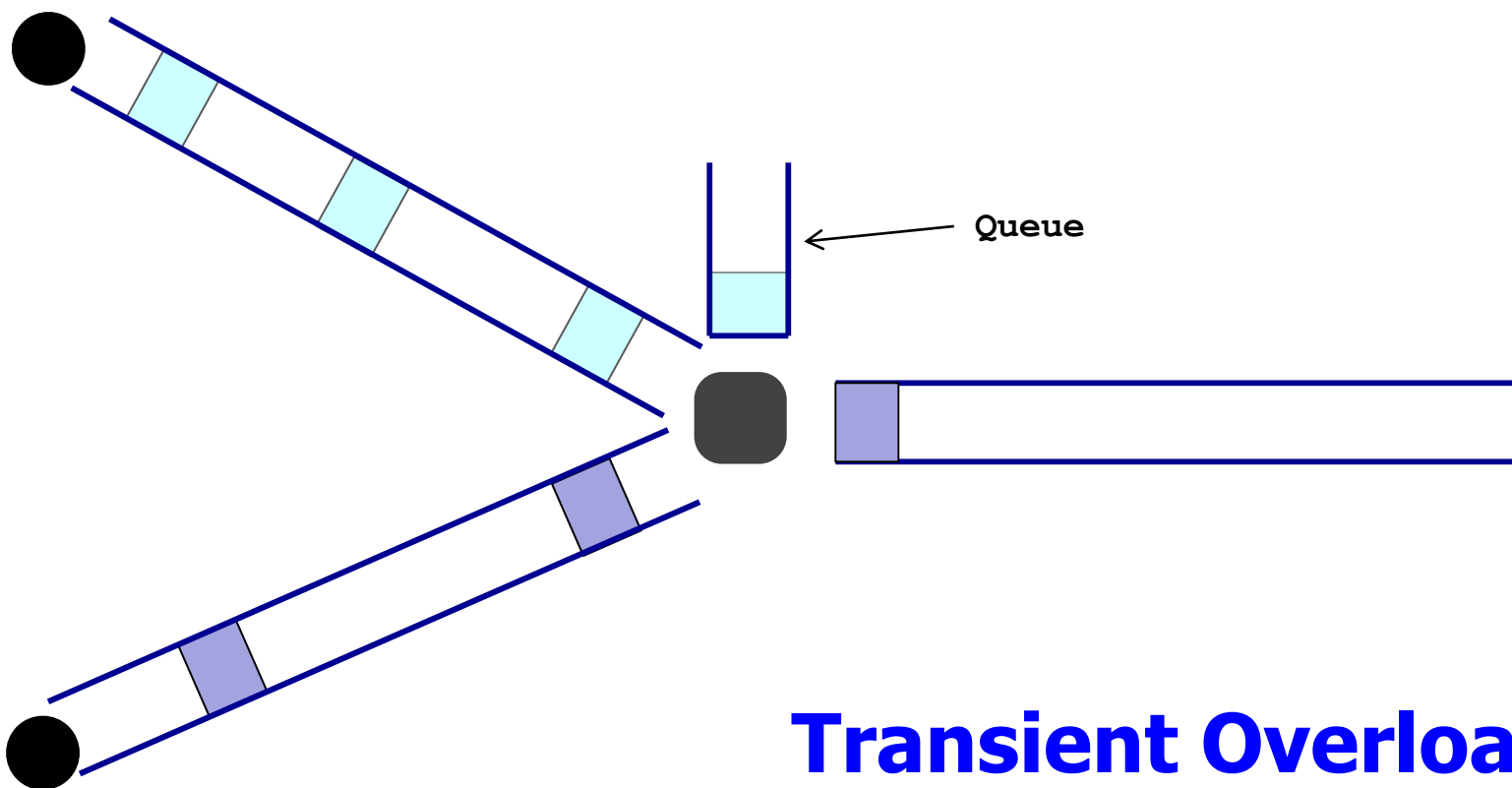
# Queueing delay: "pipe" view



**Transient Overload**  
**Not a rare event!**



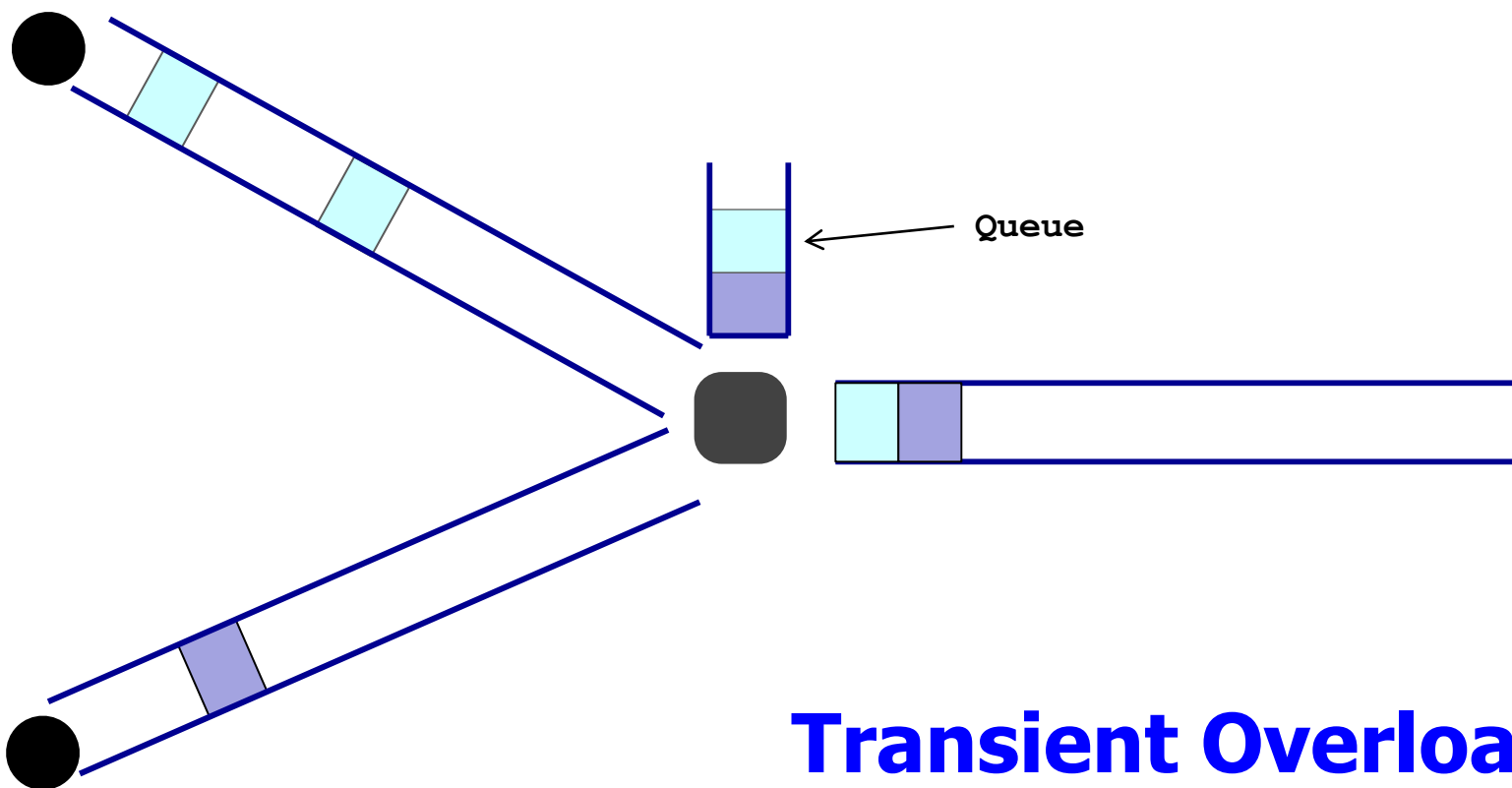
# Queueing delay: "pipe" view



**Transient Overload**

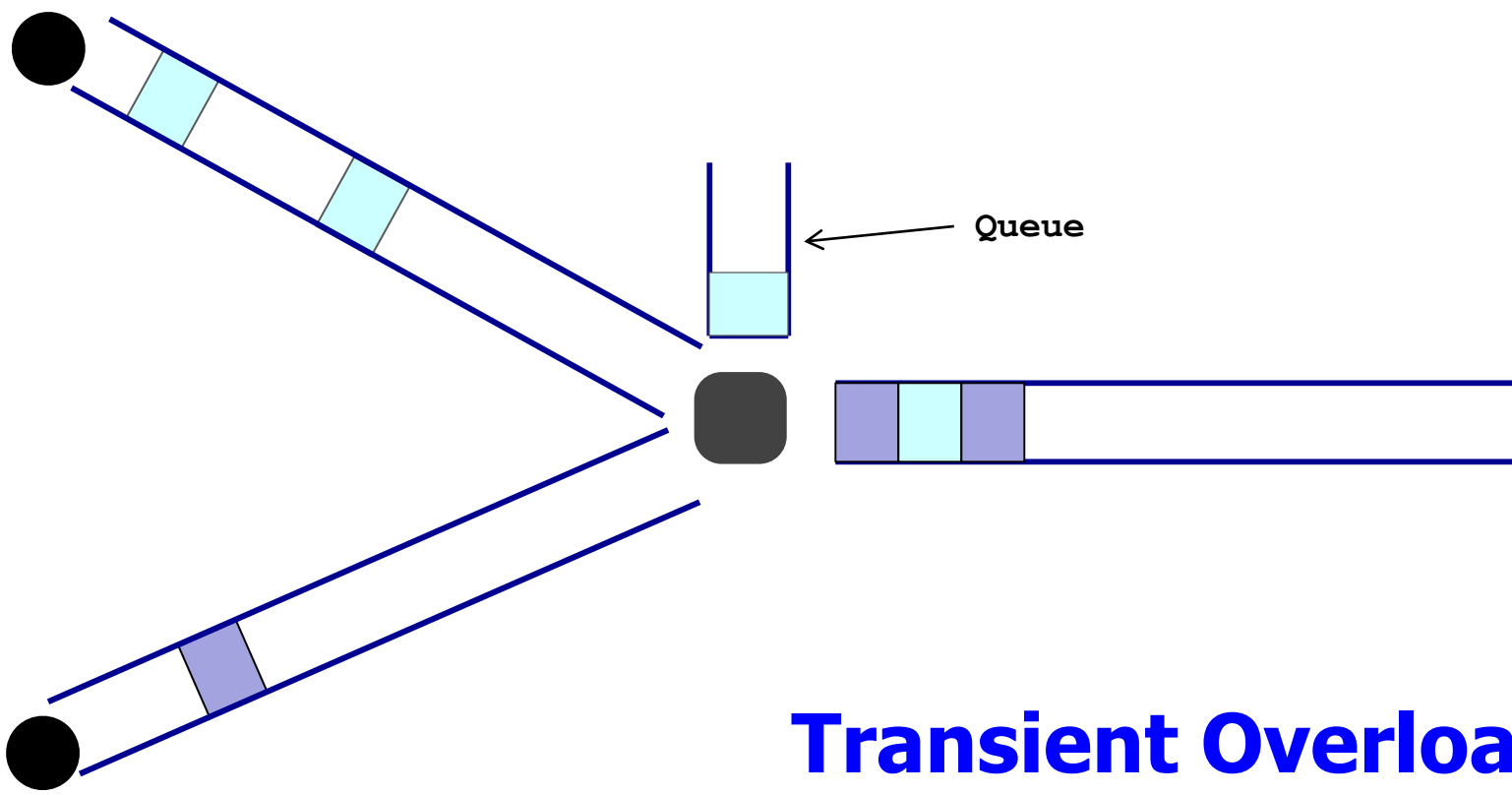


# Queueing delay: "pipe" view



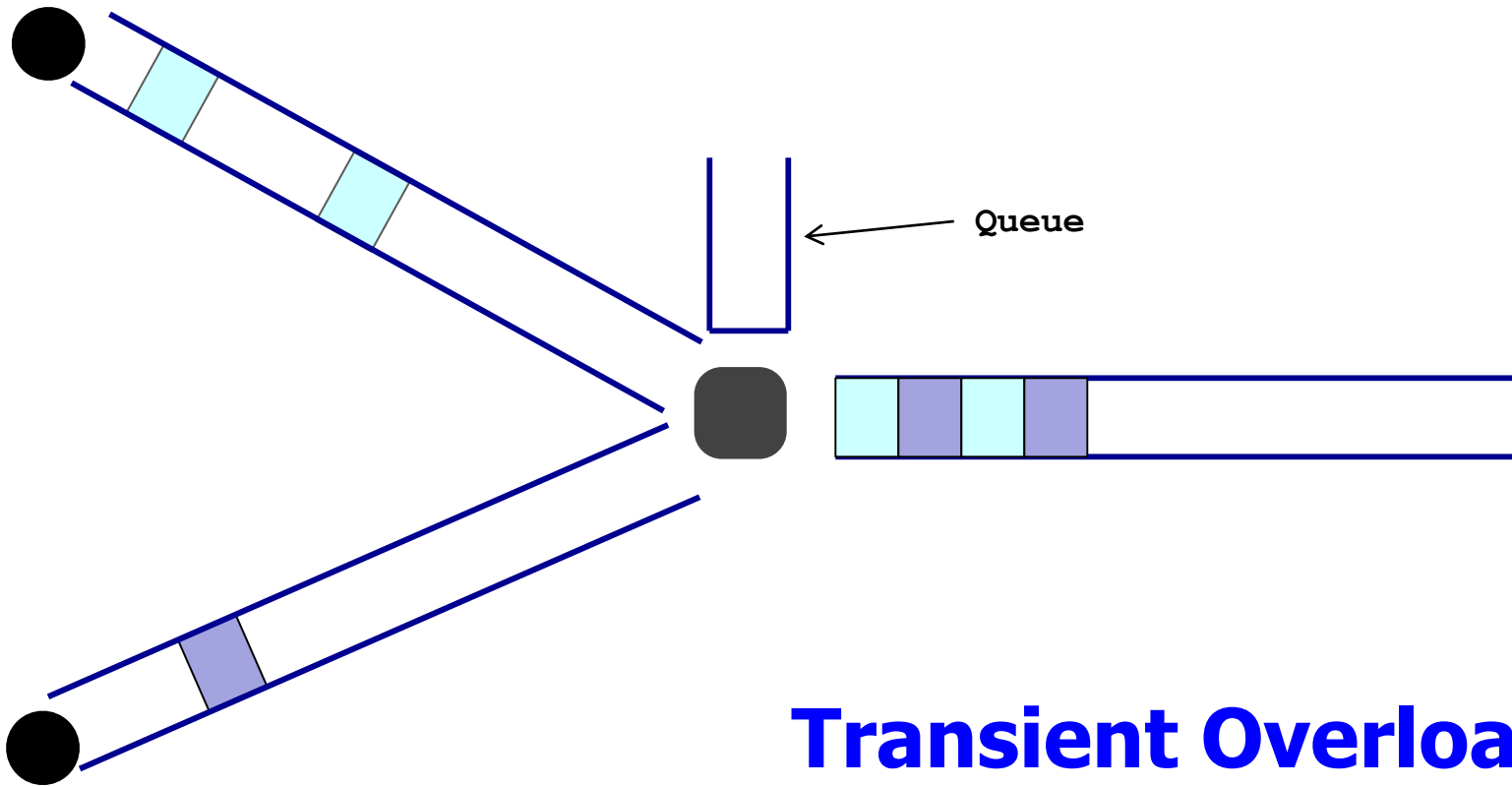


# Queueing delay: "pipe" view



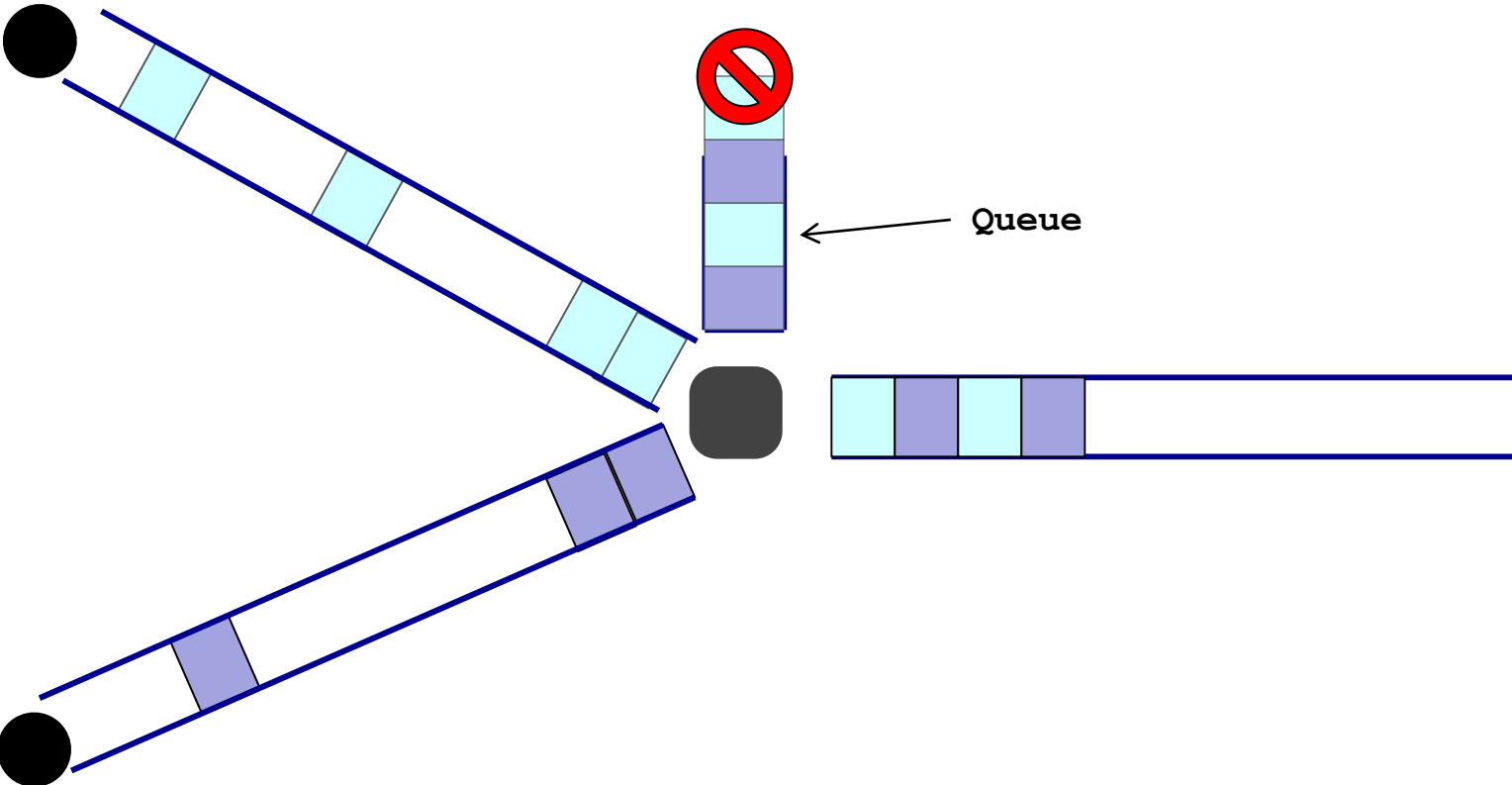


# Queueing delay: "pipe" view





# Persistent overload leads to packet loss





# Queueing delay

- How long does a packet have to sit in a buffer before it is processed?
- Depends on traffic pattern
  - Arrival rate at the queue
  - Nature of arriving traffic (bursty or not?)
  - Transmission rate of outgoing link



# Queueing delay

- How long does a packet have to sit in a buffer before it is processed?
- Characterized with statistical measures
  - Average queuing delay
  - Variance of queuing delay
  - Probability delay exceeds a threshold value



# Basic queueing theory terminology

- Arrival process: how packets arrive
  - Average rate  $A$
  - Peak rate  $P$
- $W$ : average time packets wait in the queue
  - $W$  for "waiting time"
- $L$ : average number of packets waiting in the queue
  - $L$  for "length of queue"



# Little's Law (1961)

- $L = A \times W$
- Compute  $L$ : count packets in queue every second
  - How often does a single packet get counted?  $W$  times
- Why do you care?
  - Easy to compute  $L$ , harder to compute  $W$

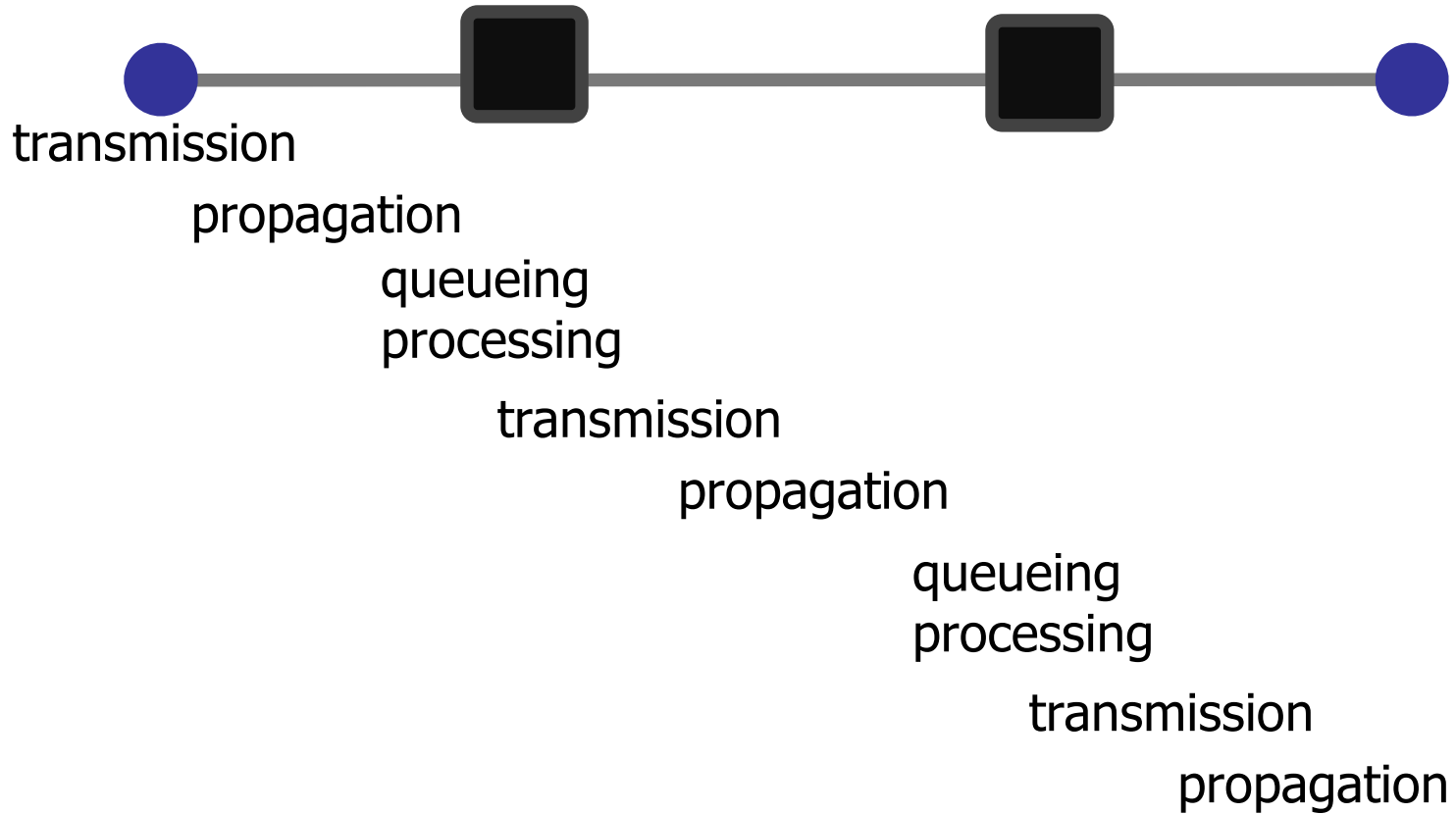


# Processing Delay

- How long does the switch take to process a packet?
  - Negligible



# End-to-end delay





# Loss

- What fraction of the packets sent to a destination are dropped?



# Throughput

- At what rate is the destination receiving data from the source



# Throughput

Transmission rate  $R$  bits/sec



File of size  $F$  bits

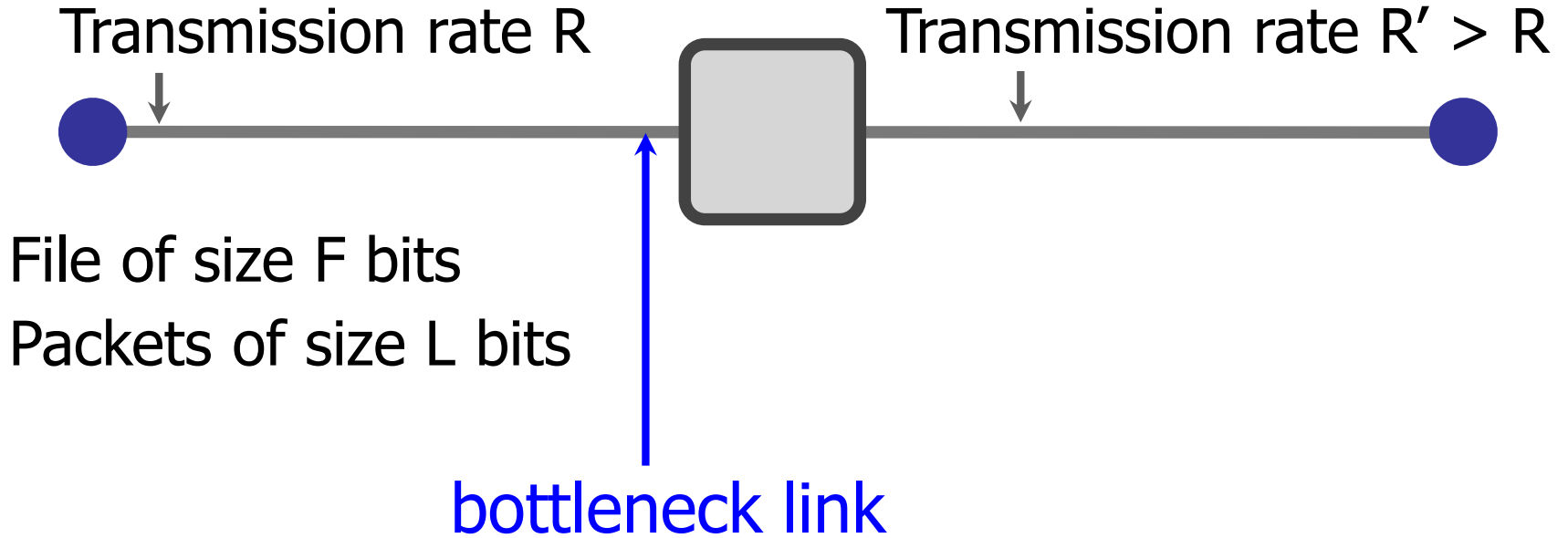
Packets of size  $L$  bits

Transfer time ( $T$ )  
=  $F/R$  + propagation delay

Average throughput =  $F/T \approx R$



# End-to-end throughput



$$\text{Average throughput} = \min\{R, R'\} = R$$



# Summary

- Internet基本概念
  - 什么是Internet
    - 组成、服务、协议
  - 网络边缘
  - 网络接入
    - 家庭、公司、无线
  - 网络核心
    - 电路交换、分组交换、虚电路
- Internet历史
- 协议层次及模型
  - OSI七层模型
  - TCP/IP协议栈五层模型
- 网络安全基本概念
- Internet应用
  - C/S构架
  - P2P构架