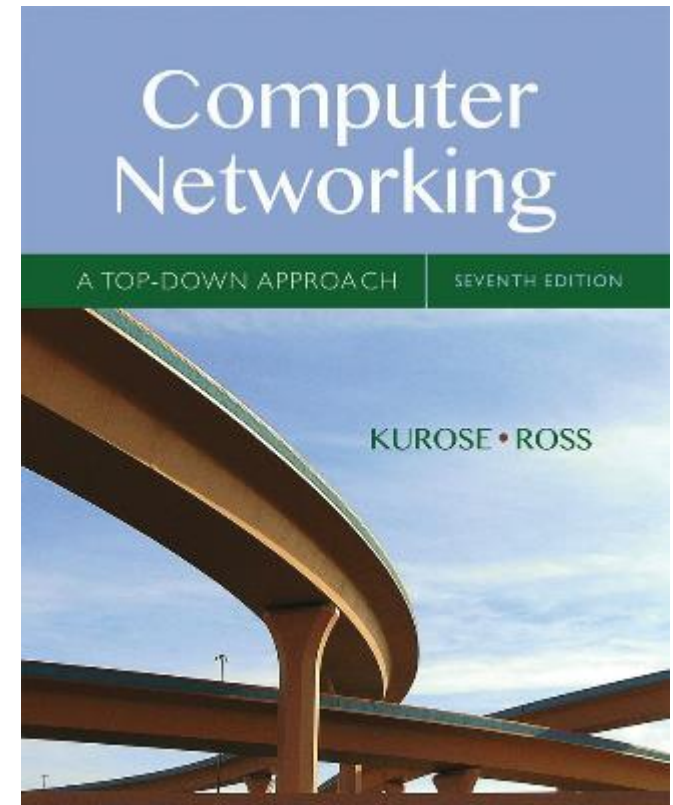


Review



Computer Networking: A Top Down Approach

7th edition

Jim Kurose, Keith Ross

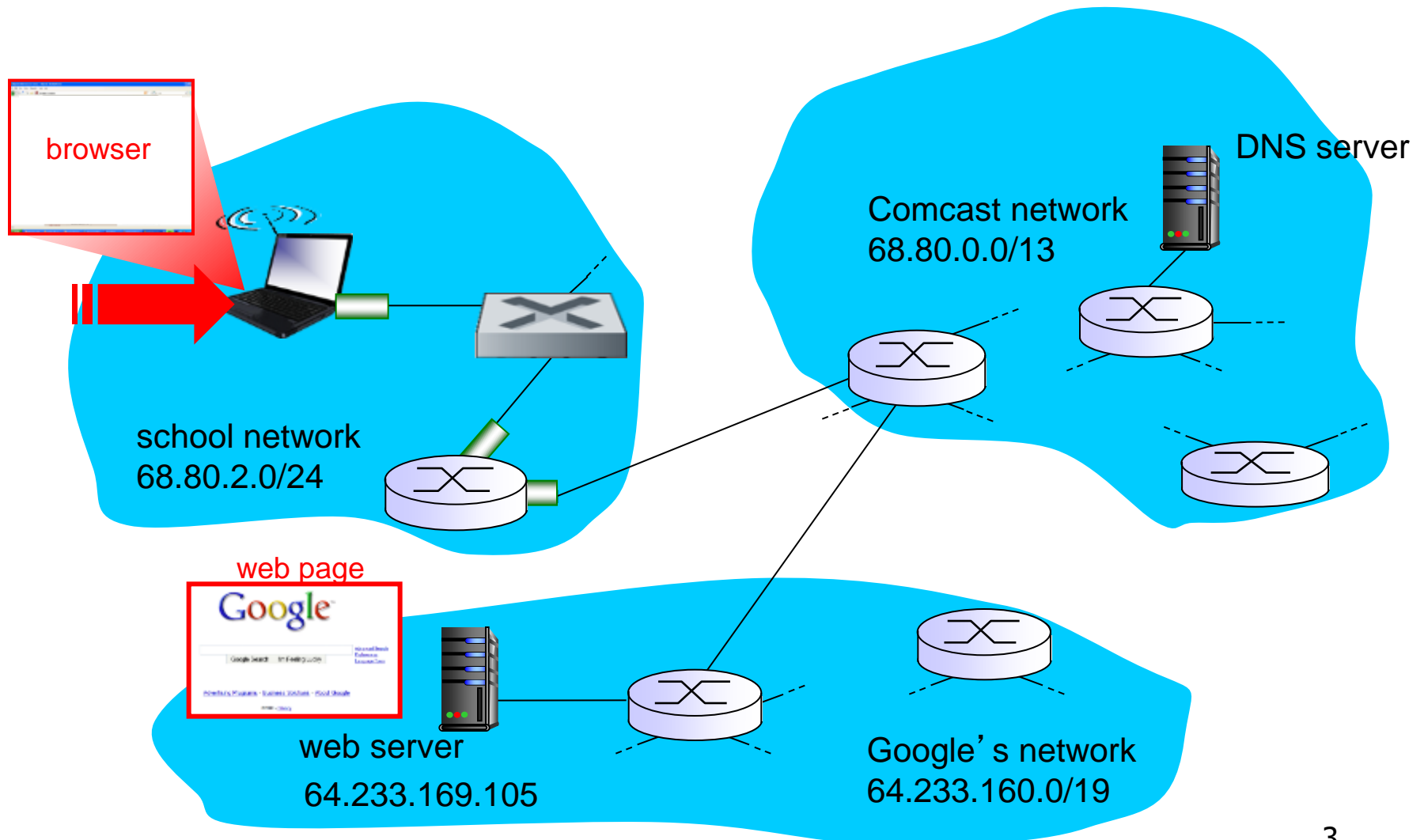
Pearson/Addison Wesley

April 2016

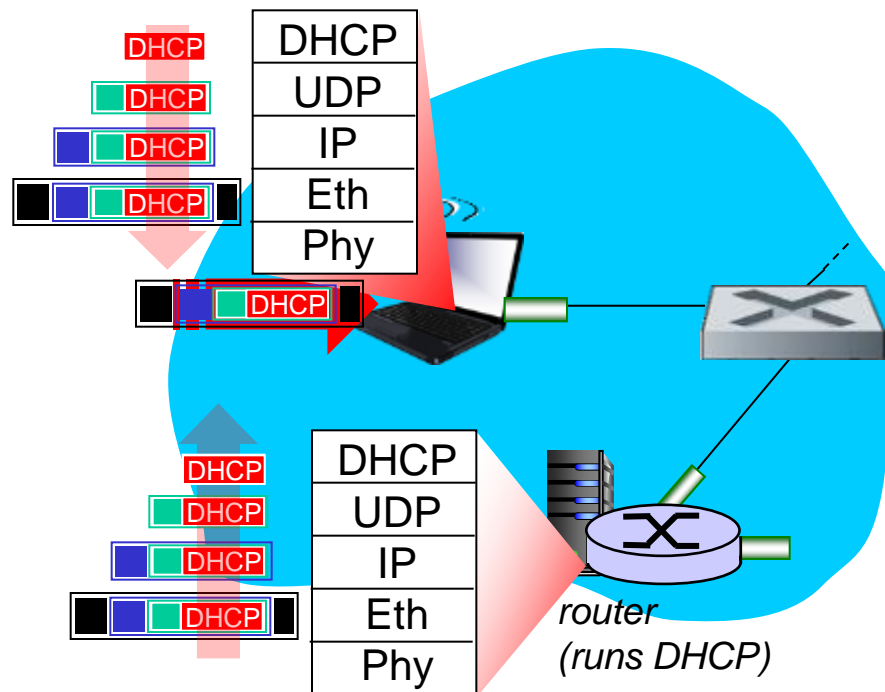
6.7 *Synthesis*: a day in the life of a web request

- journey down protocol stack complete!
 - application, transport, network, link
- putting-it-all-together: synthesis!
 - *goal*: identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
 - *scenario*: student attaches laptop to campus network, requests/receives www.google.com

A day in the life: scenario

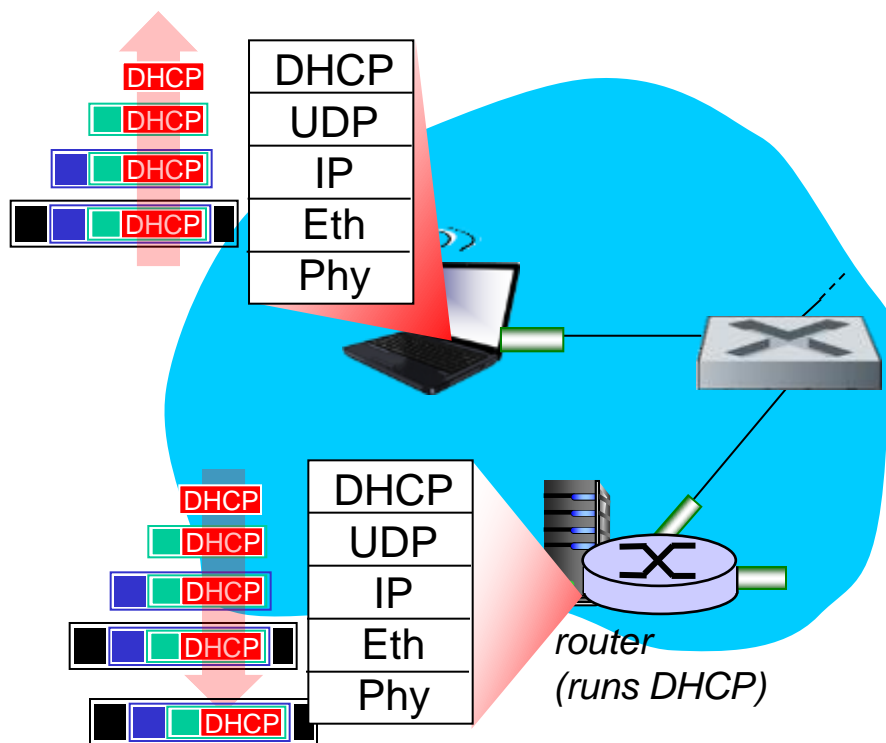


A day in the life... connecting to the Internet



- connecting laptop needs to get its own IP address, 网关地址, DNS server 地址: use **DHCP**
- DHCP request **encapsulated** in **UDP**, encapsulated in **IP**, encapsulated in **802.3** Ethernet
- Ethernet frame **broadcast** (dest: FFFFFFFFFFFFFFFF) on LAN, received at router running **DHCP** server
- Ethernet **demuxed** (多路分解) to IP demuxed, UDP demuxed to DHCP

A day in the life... connecting to the Internet



- DHCP server formulates **DHCP ACK** containing **client's IP address**, **IP address of first-hop router for client**, **name & IP address of DNS server**(DHCP提供什么)
- encapsulation at DHCP server, frame forwarded (**switch learning**) through LAN, demultiplexing at client
- DHCP client (67端口) receives DHCP (68端口) ACK reply

DHCP能提供: IP address, name & addr of DNS server, 网关的IP地址

DHCP client-server scenario

DHCP server: 223.1.2.5

DHCP discover

arriving
client



Broadcast: is there a
DHCP server out there?

DHCP offer

Broadcast: I'm a DHCP
server! Here's an IP
address you can use

DHCP request

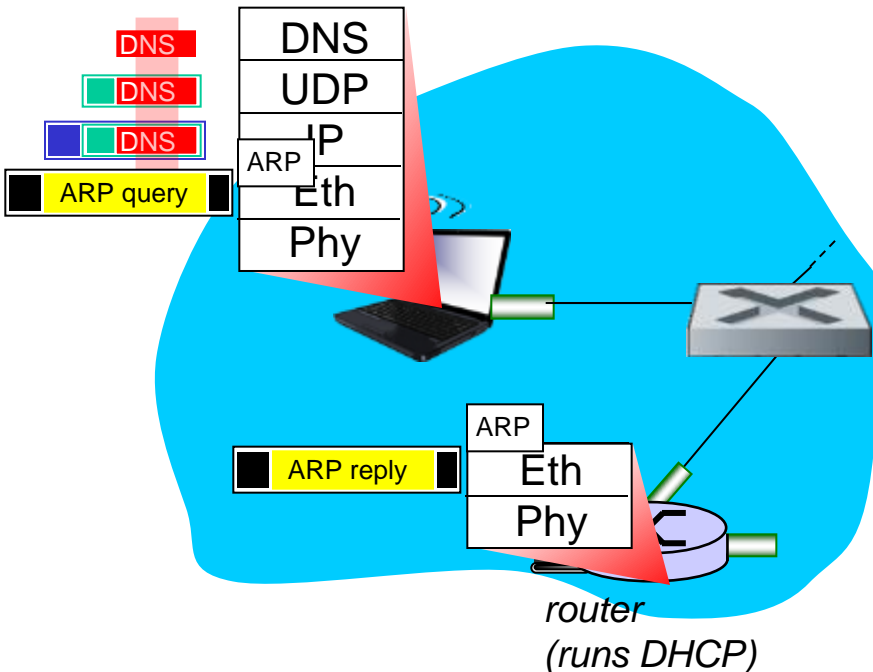
Broadcast: OK. I'll take
that IP address!

DHCP ACK

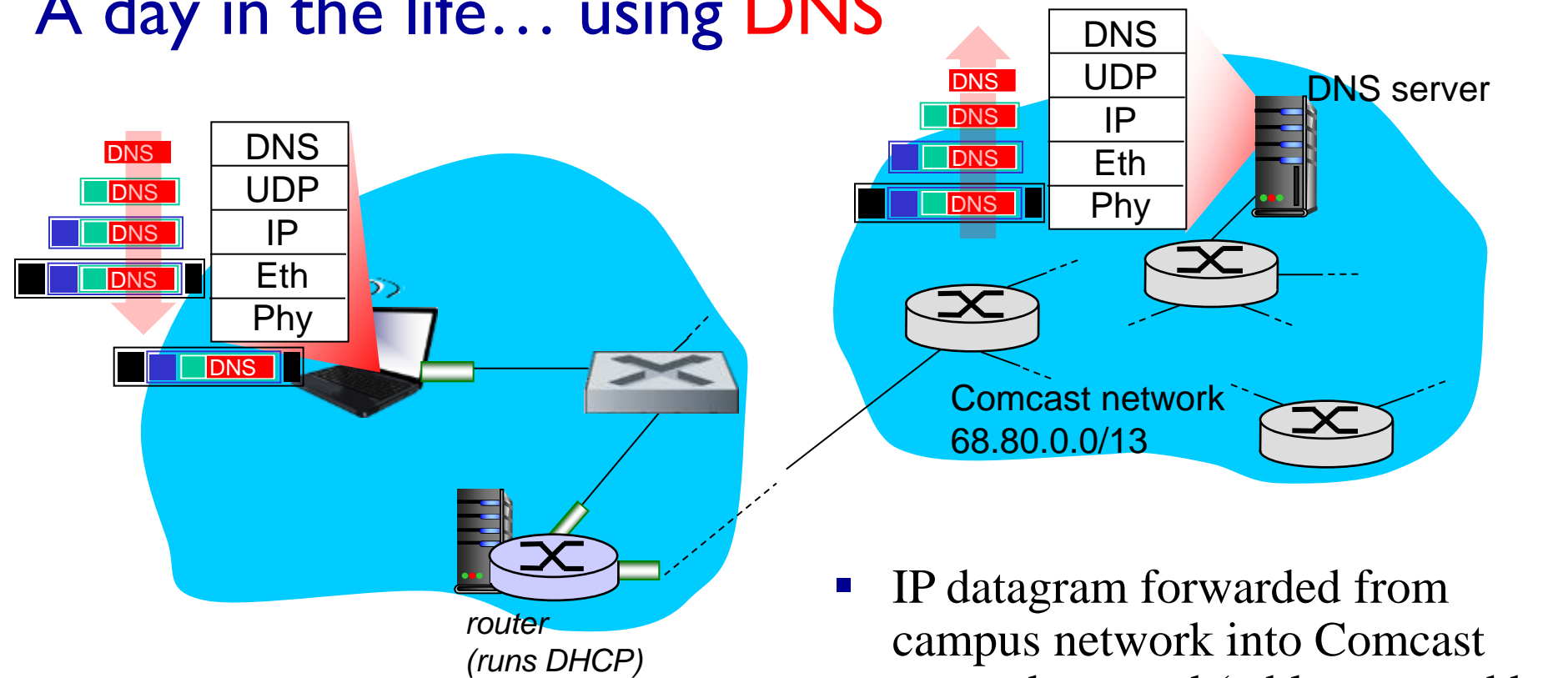
Broadcast: OK. You've
got that IP address!

A day in the life... **ARP** (before DNS, before HTTP)

- before sending **HTTP** request, need IP address of `www.google.com`:
DNS
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. 为了把帧发送到路由器, 需要知道路由器的接口: **ARP**
- **ARP query** broadcast, received by router, which replies with **ARP reply** giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query



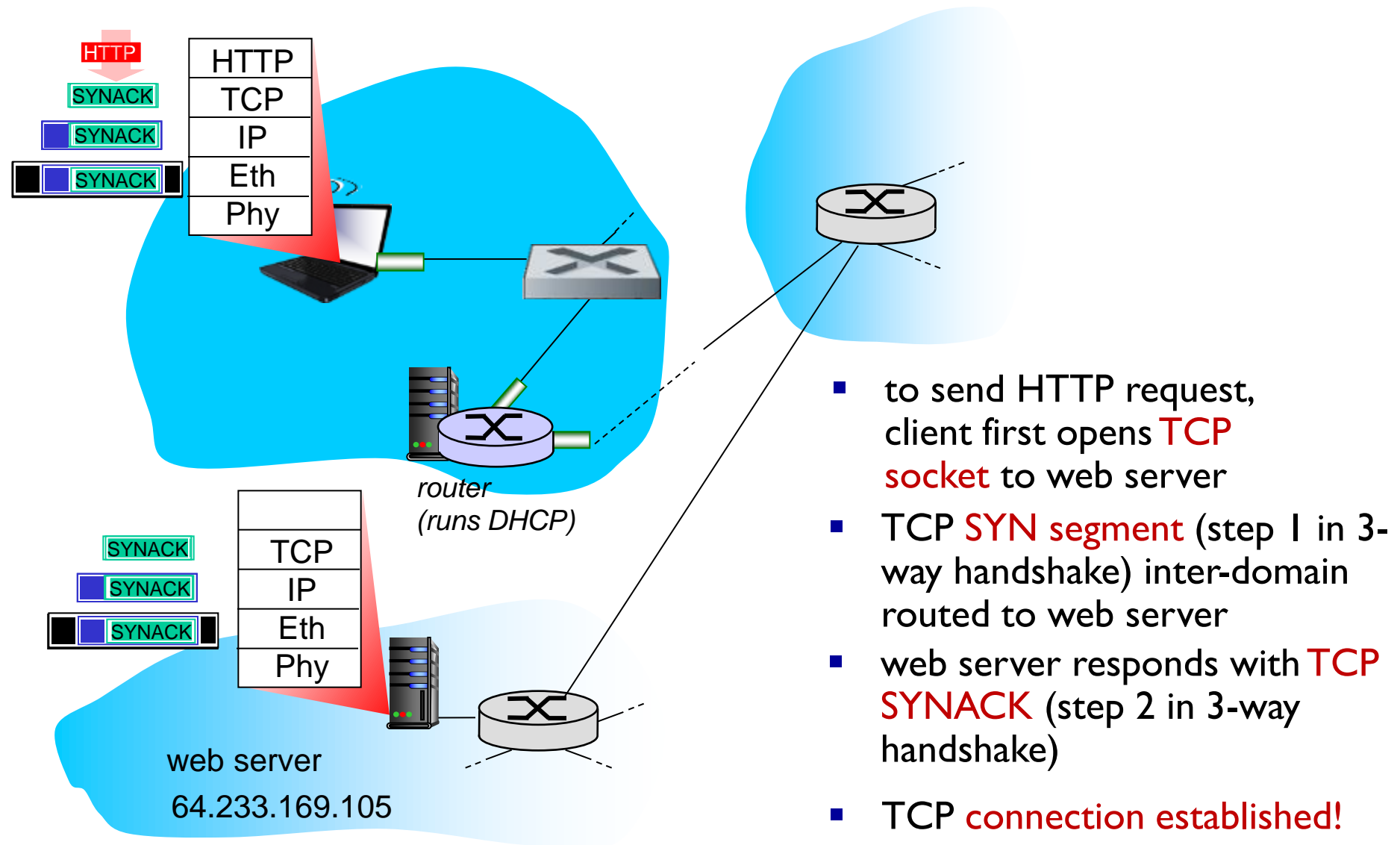
A day in the life... using DNS



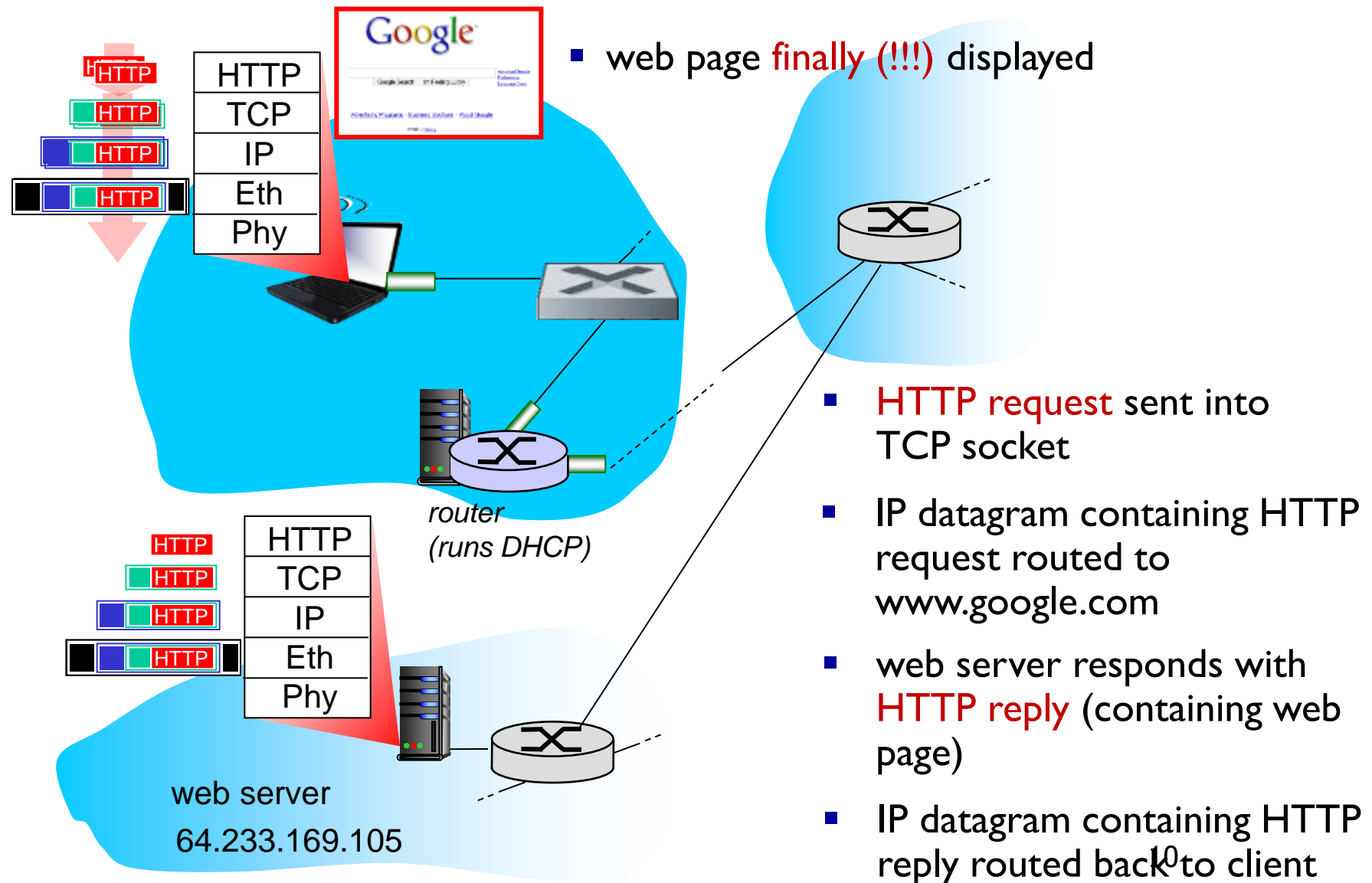
- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

- IP datagram forwarded from campus network into Comcast network, routed (tables created by **RIP**, **OSPF**, **IS-IS** and/or **BGP** routing protocols) to DNS server
- demuxed to DNS server
- DNS server replies to client with IP address of **www.google.com**

A day in the life...TCP connection carrying HTTP



A day in the life... HTTP request/reply



考试

- 上课考勤 (10%)
- 作业 (15%)
- 实验 (15%)
- 考试 (60%)
 - 考试形式 (闭卷, 卷面100分)
 - 选择题: 20题 (40分)
 - 判断题: 10题 (10分)
 - 简答题: 5题 (30分)
 - 综合题: 2题 (20分)

Ch1: 协议、服务、接入网、核心网

网络构成, 协议要素, 分组交换, 电路交换, 时延, 丢包, 吞吐量

Ch2: 应用层

应用层程序体系结构, 进程通信, 进程寻址, 网络应用需要的运输服务, 典型的应用层协议: Telnet, HTTP, SMTP, POP3, IMAP, DNS, P2P, web中的cookie机制, Web缓存, Email

Ch3: 传输层

传输层的目的是功能, TCP, UDP, 数据的封装过程, 多路复用和多路分解, UDP和TCP多路分解和多路复用的区别, 可靠性数据传输原理(包括校验和, 确认机制(ACK), 停等, 重传, 编号, 定时器, 滑动窗口机制或流水线协议(回退N步, 选择重传)), TCP的特点, 流量控制, TCP拥塞控制

Ch4: 网络层: 数据平面

网络层的功能, 网络层提供什么样的服务? 路由器的工作原理, 三种交换结构, 输入端口和输出端口, 调度机制, IP分片, IPv4的地址, NAT, IPv6与IPv4的区别, 地址段的划分(CIDR), 最长前缀匹配

Ch5: 网络层: 控制平面

路由算法分类(链路状态和距离矢量), OSPF, BGP(iBGP和eBGP), RIP, ICMP, 为什么有BGP, ICMP和两种应用

Ch6: 链路层和局域网

链路层的功能, 差错纠正技术(奇偶校验, CRC), 多路访问协议, 局域网(地址, ARP协议), 以太网(拓扑结构)交换机(switch), VLAN划分动机

Ch7: 无线网: CSMA-CA

Chapter 1 计算机网络概述

- 计算机网络、Internet(构成、通信基础设施)
- 网络边缘：位置，家庭接入、企业接入、移动接入
- 网络核心：位置，分组交换（存储转发），电路交换，多路复用（时分、频分），电路交换和分组交换的比较，网络结构
- 网络体系结构：协议的定义（3要素），分层的优点与原则，层间通信（相邻（服务），对等（协议）），数据封装，网络体系结构定义，OSI体系结构模型、TCP/IP体系结构模型、及二者的比较
- 网络性能：时延（处理、排队、传输、传播），丢包，吞吐量
- 计算机网络的发展

Ch1. 协议、服务、接入网、核心网

1. **Internet协议的作用**：规定着通信活动规范。
2. **网络架构包括**：网络边缘，接入网，核心网。
3. **数据传输速率和带宽的关系**：带宽越大，速率越高
4. **接入网有哪几种类型？**：DSL，电缆，拨号，无线
5. **在分组交换和电路交换中，数据进入网络后分别如何传输？**
 - 在电路交换中，当两台主机要通信时，网络在两台主机之间创建一条专用的端到端连接，并进行资源(缓存、传输速率)预留。
 - 在分组交换中，采用存储转发机制，报文被分割成多个分组再通过网络进行发送，资源按需分配(而不是预分配)。

6. 分组交换的优缺点

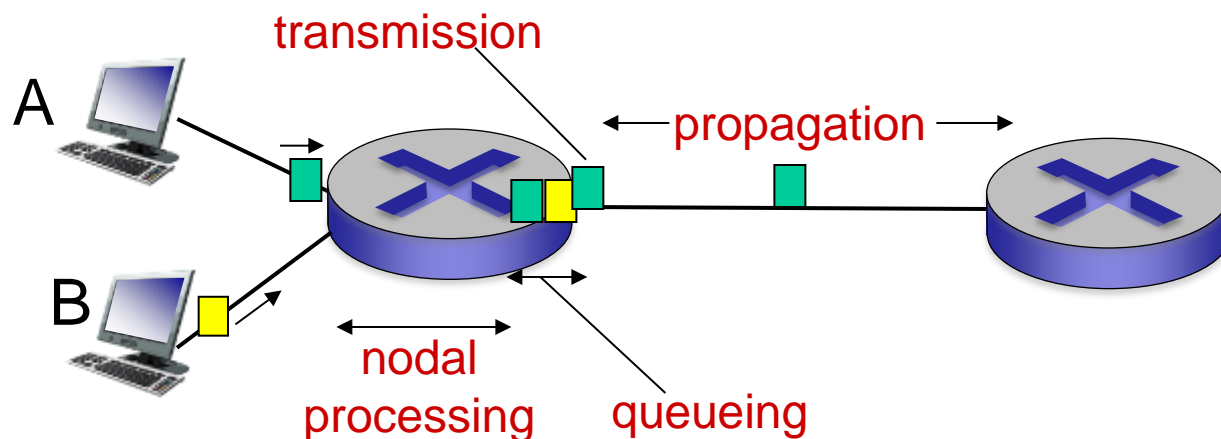
■ Advantages

- No bandwidth division
(不需共享带宽)
- No dedicated allocation
(按需使用)
- No reservation
(不需预留)

■ Disadvantages

- Resource Contention
(资源竞争)
- Congestion
(拥塞)
- Store and Forward
(存储转发)

7. 四种时延类型



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

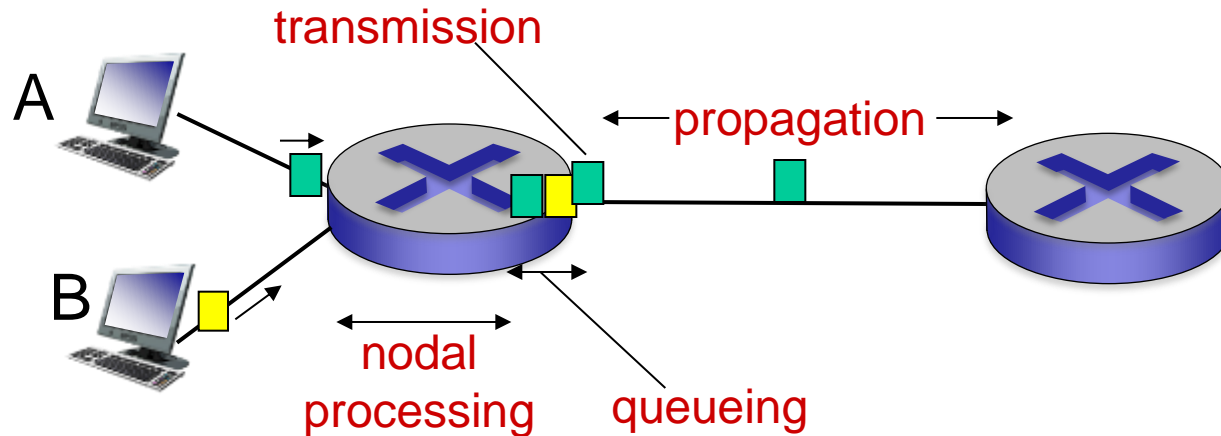
处理时延 (d_{proc}) : nodal processing

- 查错
- 确定输出链路
- typically < msec

排队时延 (d_{queue}) : queueing delay

- 在输出链路上等待的时间
- 依赖于路由器的阻塞程度

7. 四种时延类型



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

传输时延 d_{trans} : transmission delay:

- L : packet length (bits)
- R : 接口发射速率 (bps)
- $d_{\text{trans}} = L/R$

传播时延 d_{prop} : propagation delay:

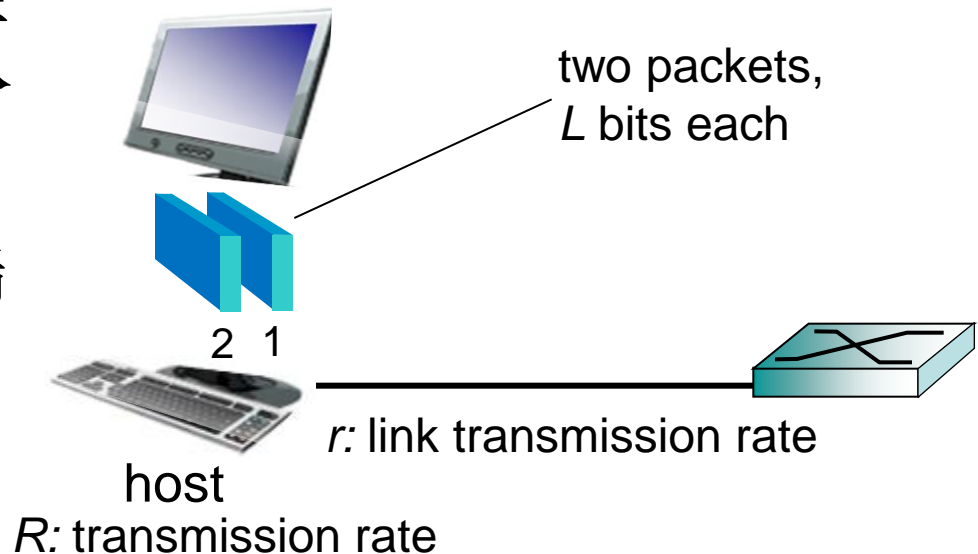
- d : length of physical link
- r : propagation speed ($\sim 2 \times 10^8$ m/sec)
- $d_{\text{prop}} = d/r$

← d_{trans} and d_{prop} →
very different

传输时延 vs. 传播延时

■ **Transmission Delay** 是指节点在发送数据时使数据块从结点进入到传输媒体所需的时间。

■ **Propagation Delay** 是指一个bit或一个signal从源传送到目的端的时间。

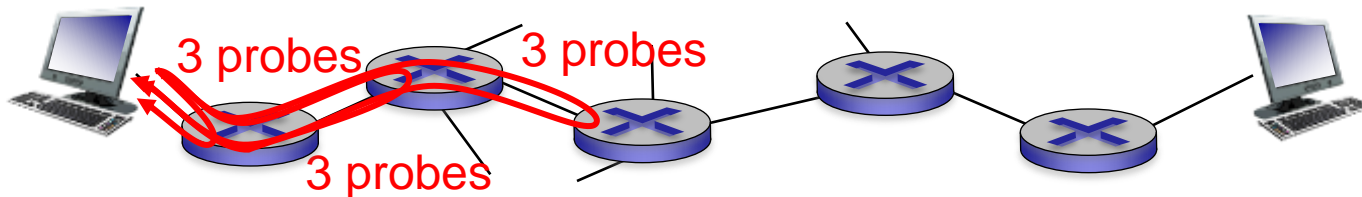


$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

$$\text{Packet Propagation delay} = \text{time needed to transmit } D\text{-distance from source into destination} = \frac{D \text{ (meter)}}{V \text{ (m/sec)}}$$

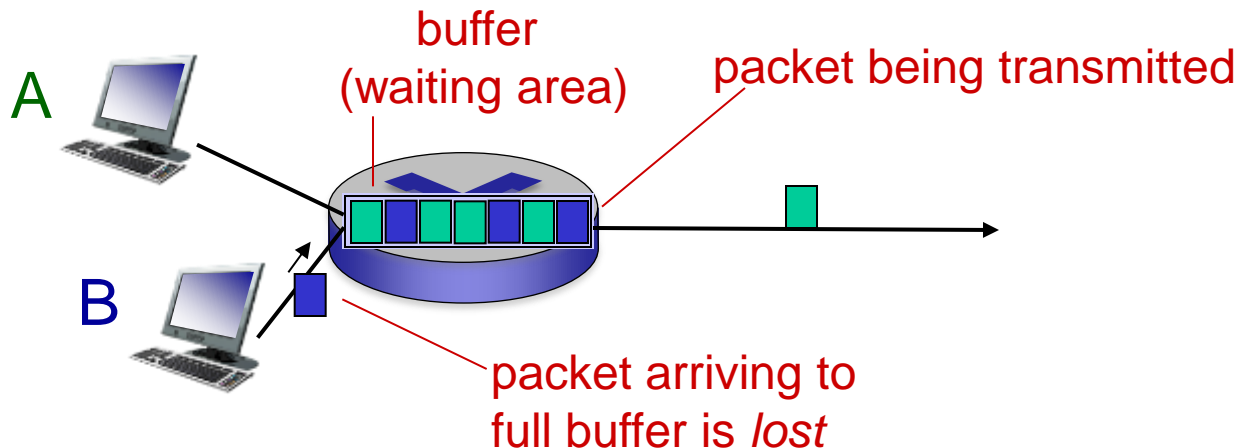
8. “Real” Internet delays and routes

- **tracert:**
- 通过发送小的ICMP数据包到目的设备直到其返回，来测量其需要多长时间。一条路径上的每个设备Traceert要测3次。输出结果中包括每次测试的时间(ms)和设备的名称（如有的话）及其IP地址。
- **traceroute**



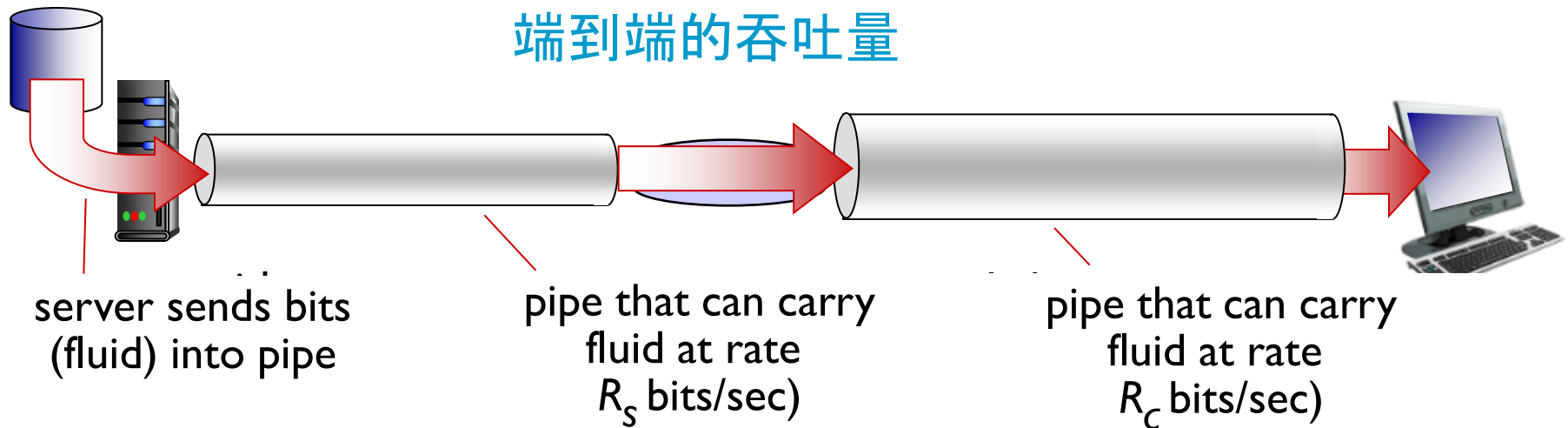
9. Packet loss(丢包)

- 队列具有有限的容量。
- 包到达一个满队列时，将丢弃该包。
- 丢失的包可能要重传，保证所有数据被送达目的地。



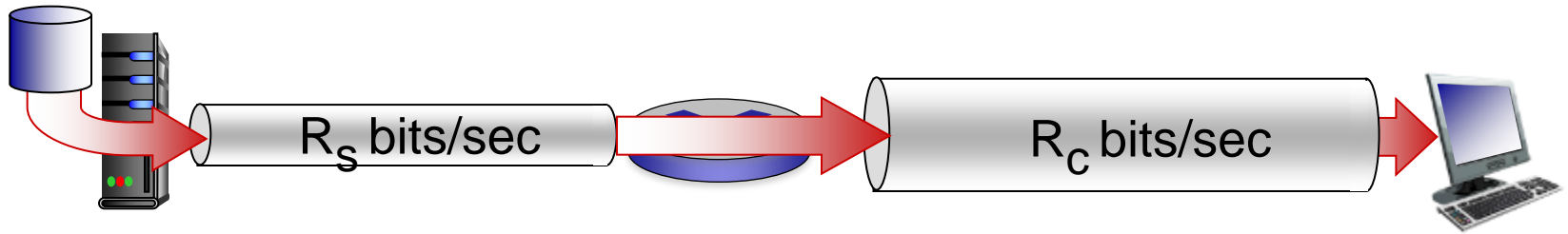
10. Throughput (吞吐量1)

- **吞吐量**: 为用户流量中转所能达到的峰值。
(bits/time unit)
 - **瞬时吞吐量**: 瞬时值rate at given point in time
 - **平均吞吐量**: 平均值**rate over longer period of time**

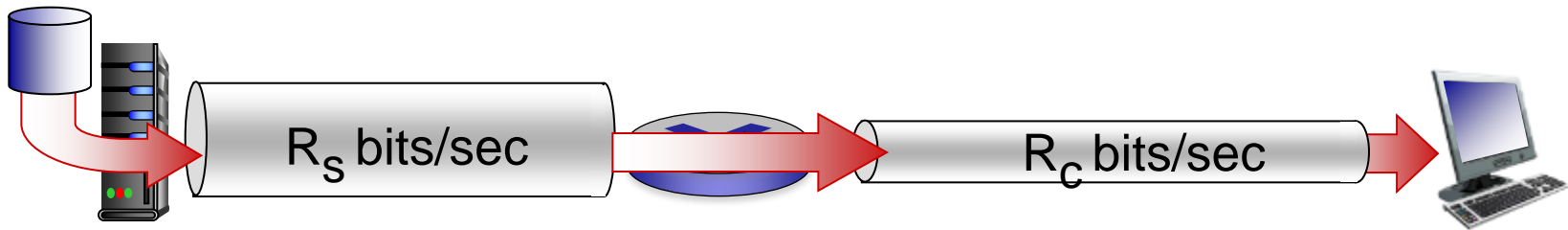


Throughput(吞吐量2)

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



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



bottleneck link

瓶颈链路限制了端到端的吞吐量。

11. Internet网的体系结构与OSI体系结构

- 复杂系统清楚的分层，有利于管理
- 功能的模块化，其变化不影响其它层

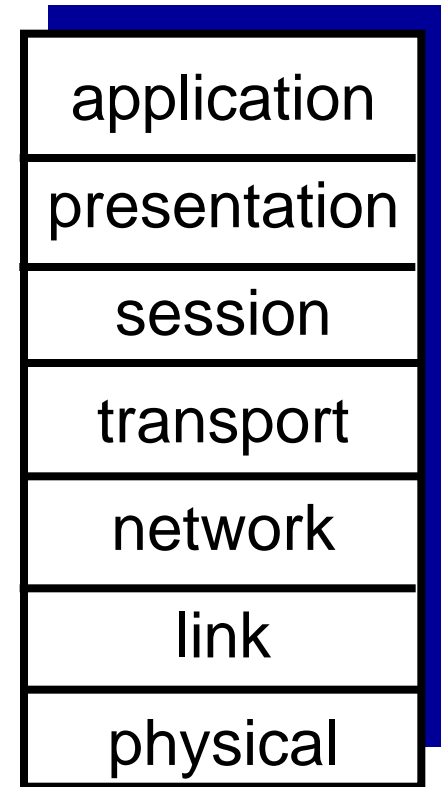
e. g., change in gate procedure doesn't affect rest of system

- 分层的缺点？
 - 功能的冗余，每层都提供查错控制
 - 跨层的依赖

- **Application(应用层):** supporting network applications
 - FTP, SMTP, HTTP
- **Transport (传输层):** process-process data transfer
 - TCP, UDP
- **Network (网络层):** routing of datagrams from source to destination
 - IP, routing protocols
- **Link (链路层):** data transfer between neighboring network elements
 - Ethernet, 802.111 (WiFi), PPP
- **Physical (物理层):** bits “on the wire”

ISO/OSI 参考模型

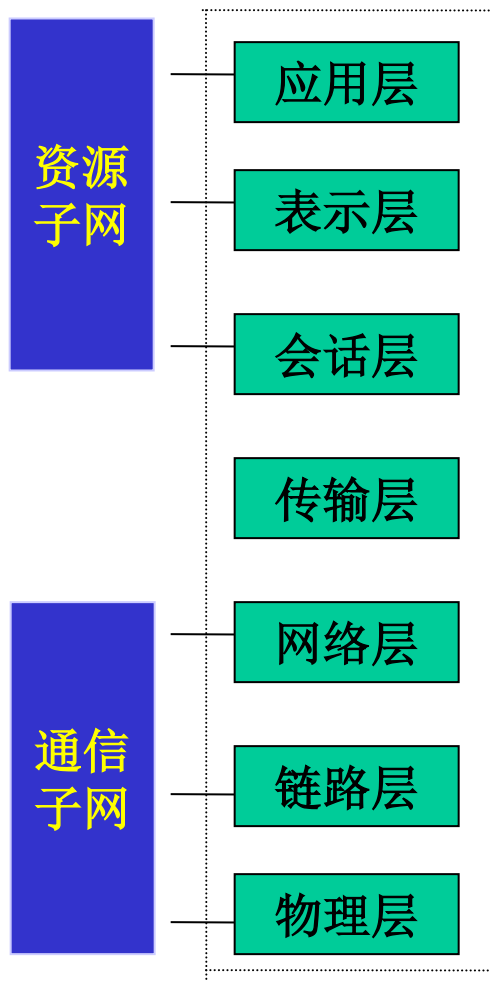
- **表示层**: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- **会话层**: synchronization, checkpointing, recovery of data exchange
- Internet stack “missing” these layers!
 - these services, *if needed*, must be implemented in application
 - needed?



网络体系结构

- 系统分层实现的好处（降低复杂性、提高灵活性）
 - a. 独立性强
 - b. 适应性强
 - c. 易于实现和维护
 - d. 有利于促进标准化
- 系统分层实现的原则
 - a. 层数应适中
 - b. 每层功能应明确
 - c. 层与层应相互独立
 - d. 层间单向引用

网络体系结构——OSI模型



做什么：应用程序（FTP、E-mail、Telnet）

内容像什么：数据结构表示、数据转换、加密、压缩

该谁讲话、从哪讲起：双工、半双工、单工、断点续发

对方在哪：提供端到端可靠的数据传输，报文，流量控制、差错控制、拥塞控制

走哪条路：如何将分组传给正确的主机，寻址定位、路由选择

每步怎么走：将数据组成帧，点到点可靠传输，帧同步、差错控制、流量控制、媒体访问控制

如何利用物理介质：规定物理信号、接口、信号形式、速率等，比特

OSI和TCP/IP模型的比较（1）

■ OSI参考模型优点

- OSI结构清晰，概念明确，适合作为教学内容
- 产生在协议发明之前，非常通用

■ OSI参考模型缺点

- 设计者没有实际经验，层次功能划分不是十分合理
- 协议实现复杂，运行效率低
- 标准制订周期长，符合该标准的设备无法及时进入市场

OSI和TCP/IP模型的比较（2）

■ TCP/IP参考模型优点

- 一开始就考虑到异构网互联问题
- 一开始就对面向连接的服务和无连接的服务并重
- 有很好的网管功能

■ TCP/IP参考模型缺点

- 对服务、协议等概念并没有很清楚地区分开
- 通用性差
- 缺乏对物理层和链路层的描述

Chapter 2 应用层

- 应用层协议：位置，含义，网络应用架构（C/S、P2P、混合），进程通信（接口、地址），网络应用需要的传输服务
- Web和HTTP：Web页面，HTTP协议（80，TCP），非持续连接和持续连接，HTTP报文，Cookie，Web缓存，条件Get
- E-mail：组成，SMTP协议（25，TCP），邮件报文格式，从邮件服务器获取报文的三种方式（POP3（110，TCP）、IMAP、HTTP）
- DNS：功能（53，UDP），DNS提供的服务、实现方式（两种查询），4类域名服务器
- P2P应用（BitTorrent协议）
- 视频流与内容分发网络
- Socket编程

Ch2. 应用层

1. 网络应用程序的可能架构：客户服务器，P2P
2. 进程间的通信：同一主机，不同主机
3. 应用程序对传输服务的可能要求：可靠、安全、实时，吞吐量
4. TCP和UDP分别为应用层提供什么样的服务？

4. 传输层提供的服务

TCP service:

- *reliable transport* between sending and receiving process
(包括哪些机制?)
- *flow control*: sender won't overwhelm receiver
- *congestion control*: throttle sender when network overloaded
- *does not provide*: timing, minimum throughput guarantee, security
- *connection-oriented*: setup required between client and server processes

UDP service:

- *unreliable data transfer* between sending and receiving process
- *does not provide*: reliability, flow control, congestion control, timing, throughput guarantee, security, or connection setup,

Ch2. 应用层

5. 应用层与传输层的协议对应

6. Web和HTTP

1. HTTP overview
2. HTTP连接
3. HTTP报文格式
4. Cookie
5. Web缓存
6. 条件Get方法

User-server state: cookies

many Web sites use cookies

four components:

- 1) 在HTTP响应报文中有一个cookie header line
- 2) 在下一个请求报文中cookie header line
- 3) cookie file kept on user's host, managed by user's browser
- 4) back-end database at Web site

example:

- Susan always access Internet from PC
- visits specific e-commerce site for first time
- when initial HTTP requests arrives at site, site creates:
 - unique ID
 - entry in backend database for ID

Cookies的作用

what cookies can be used for:

- Authorization 认证
- shopping carts 购物车
- Recommendations 推荐
- user session state (Web e-mail) 会话

aside

cookies and privacy:

- cookies permit sites to learn a lot about you
- you may supply name and e-mail to sites

how to keep “state”: 保持状态

- 在多个事务处理过程中, 保持发送者和接受者之间的状态
- cookies: http messages carry state

Method types

HTTP/1.0:

- GET
- POST
- HEAD
 - asks server to leave requested object out of response

HTTP/1.1:

- GET, POST, HEAD
- PUT
 - uploads file in entity body to path specified in URL field
- DELETE
 - deletes file specified in the URL field

HTTP connections

非持续性连接

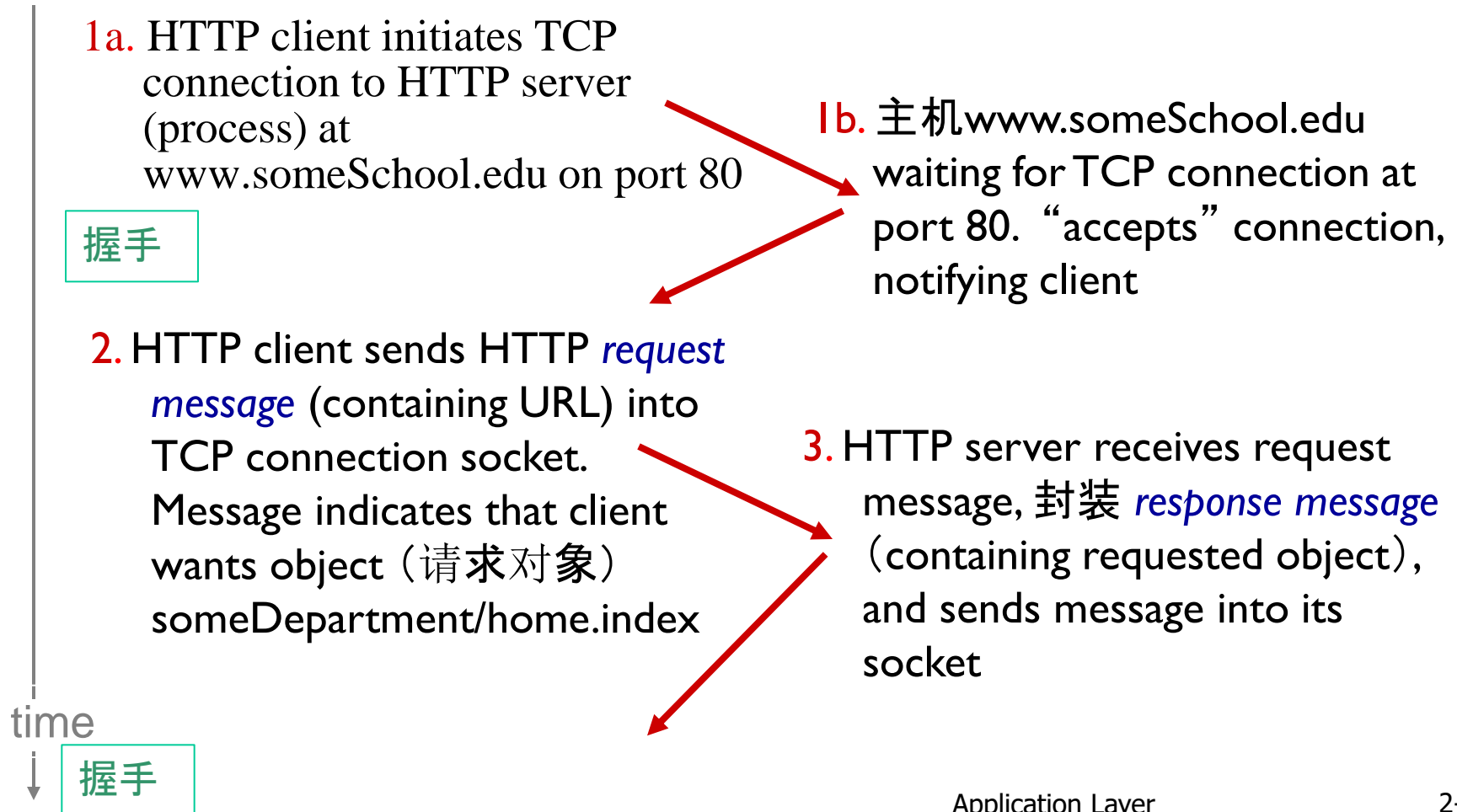
- 一次TCP连接最多发送一个object。
- downloading multiple objects required multiple connections
- HTTP 1.0

持续性连接

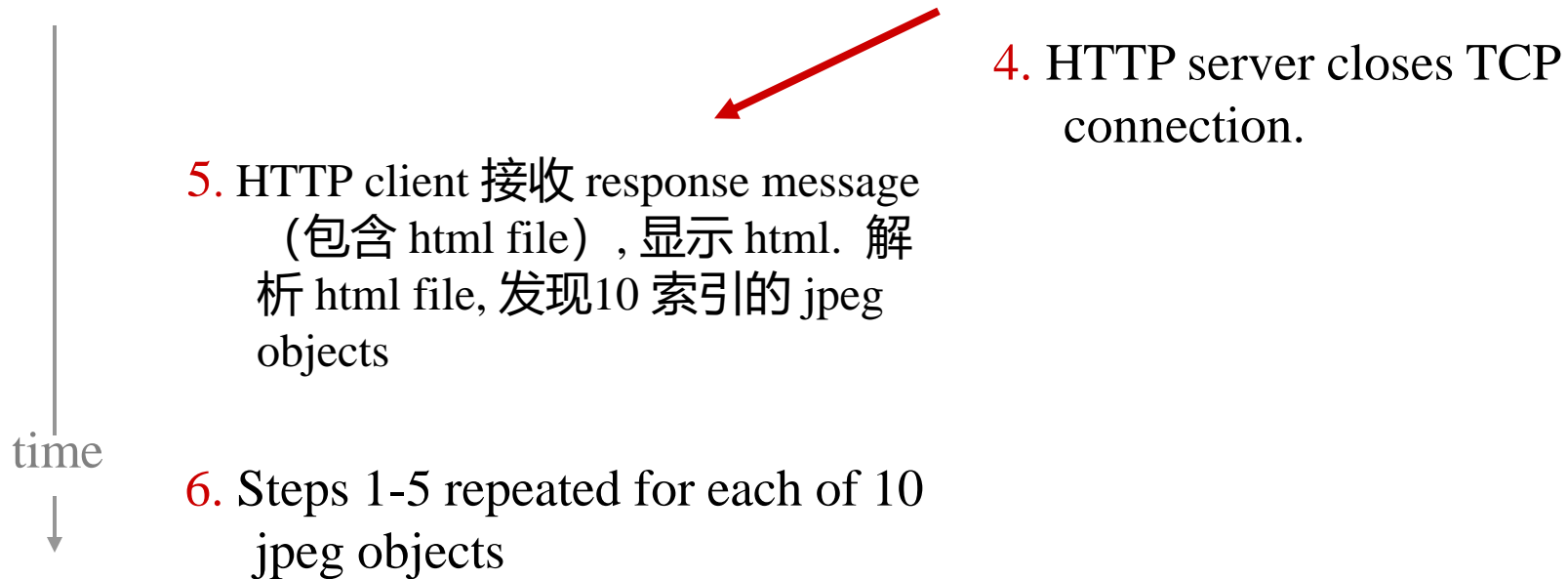
- 一次TCP连接可以发送多个object。
- HTTP 1.1
- 大部分浏览器默认支持。

非持续性连接的HTTP

suppose user enters URL: `www.someSchool.edu/someDepartment/home.index` (contains text, references to 10 jpeg images)



非持续性连接的HTTP(cont.)

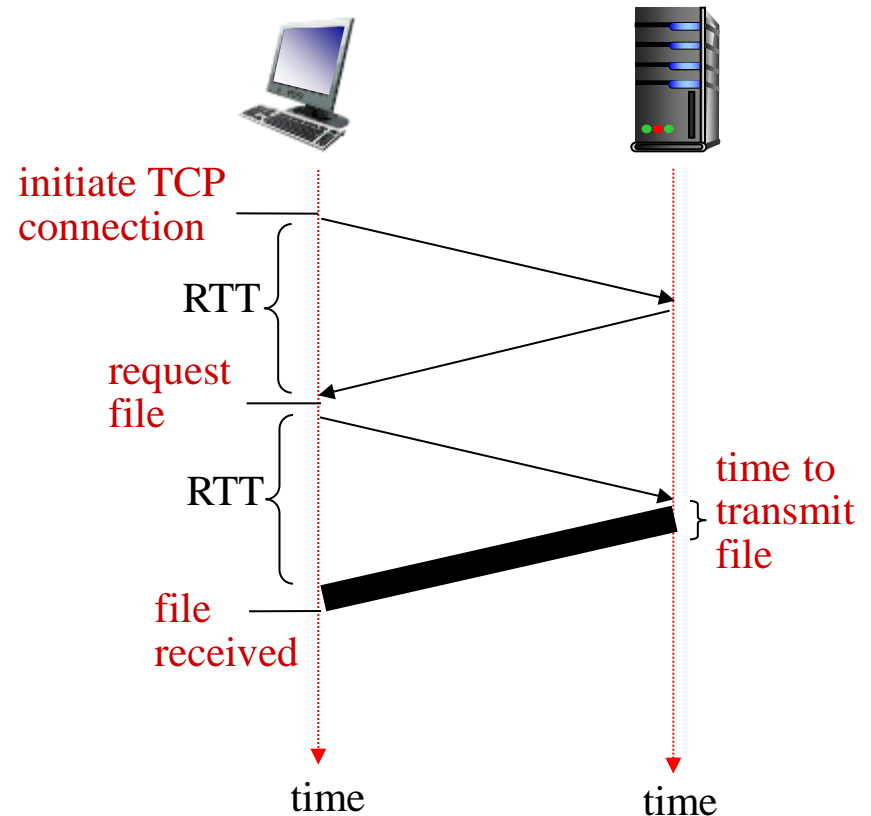


非持续性连接的HTTP : response time

RTT (往返时间): 一个小数据包从客户到服务器然后再返回客户所花费的时间。

HTTP response time:

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- file transmission time
- 总响应时间=
 $2\text{RTT} + \text{file transmission time}$



二者比较

non-persistent HTTP

issues:

- 每个对象需要2个RTT
- 每个TCP连接由OS负担
- 浏览器只有使用并行的TCP 连接来获取索引的web objects。

persistent HTTP:

- server 发送response 后保持连接
- subsequent HTTP messages **between same client/server** sent over open connection
- client 解析到页面中索引的object后马上发送 requests
- 1个object仅需要一个RTT

Ch2. 应用层

7. SMTP和POP3

8. DNS：两种查询方式，缓存技术，工作原理

9. P2P体系结构，了解BitTorrent特点

10. 了解内容分发网的结构

11. TCP和UDP套接字编程：流程

Chapter 3 运输层

- 传输层概述：传输层的功能、位置、数据单元、和网络层关系
- 多路复用与多路分解
- UDP协议：无连接、不可靠，报文格式，校验和
- 可靠数据传输机制：rdt状态图，差错检测、应答机制、序号、定时器、滑动窗口、流水线协议（回退N步，选择重传）
- TCP协议：报文格式、工作原理、流量控制、连接建立（3次握手）、连接释放（4次挥手）
- 拥塞控制：拥塞原因与代价，拥塞控制方法（端到端、网络辅助），TCP拥塞控制（加性增、乘性减，慢启动、拥塞避免、快速恢复、公平性）

Ch3. 传输层

1. 传输层和网络层的关系
2. 理解数据封装的过程
3. 多路复用和多路分解
4. TCP和UDP的报文格式
5. 无连接和有连接的多路分解和复用
根据不同的元组区分服务进程
6. UDP使用校验和
7. 理解可靠数据传输的原理

数据的封装过程（由上至下）：

1.data

2.segment/datagram

3.packet

4.frame

5.bit



应用层

传输层

网络层

数据链路层

物理层

多路复用/多路分解

多路复用

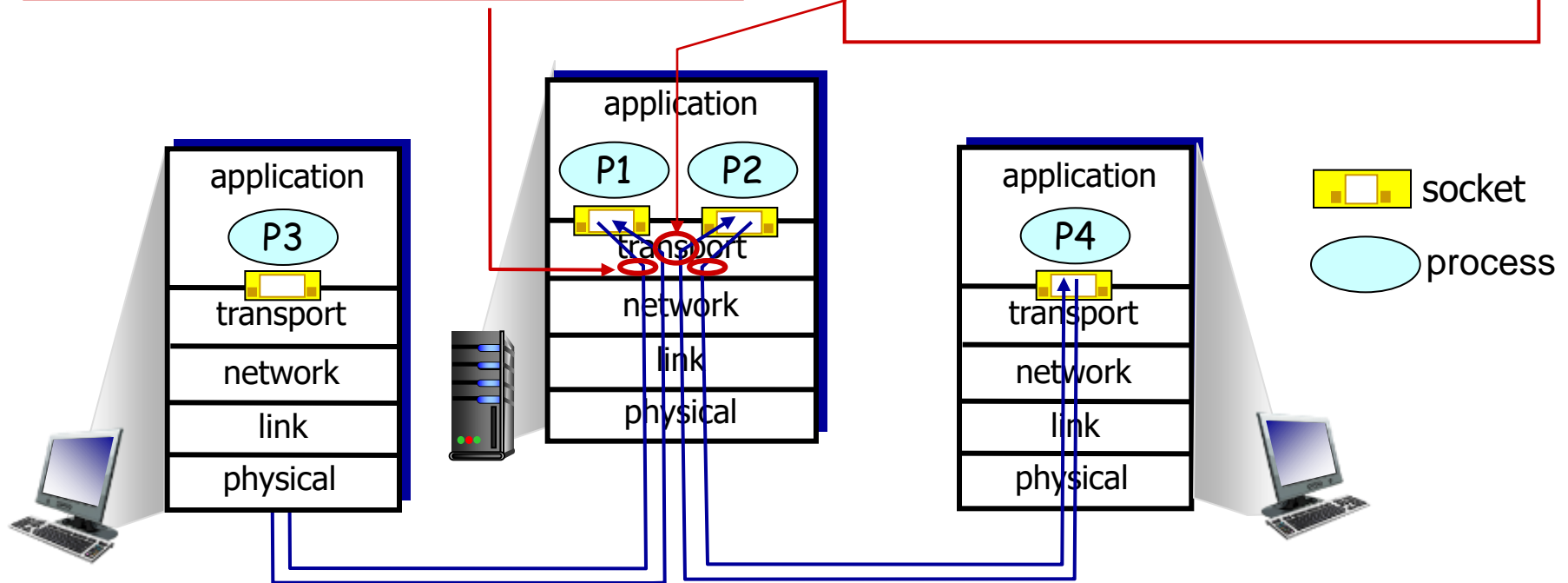
multiplexing at sender:

从上到下，从套接字中收集数据，加报文头，传递到网络层。

多路分解

demultiplexing at receiver:

从下到上，将运输层的报文段交给正确的套接字。



面向连接的多路分解

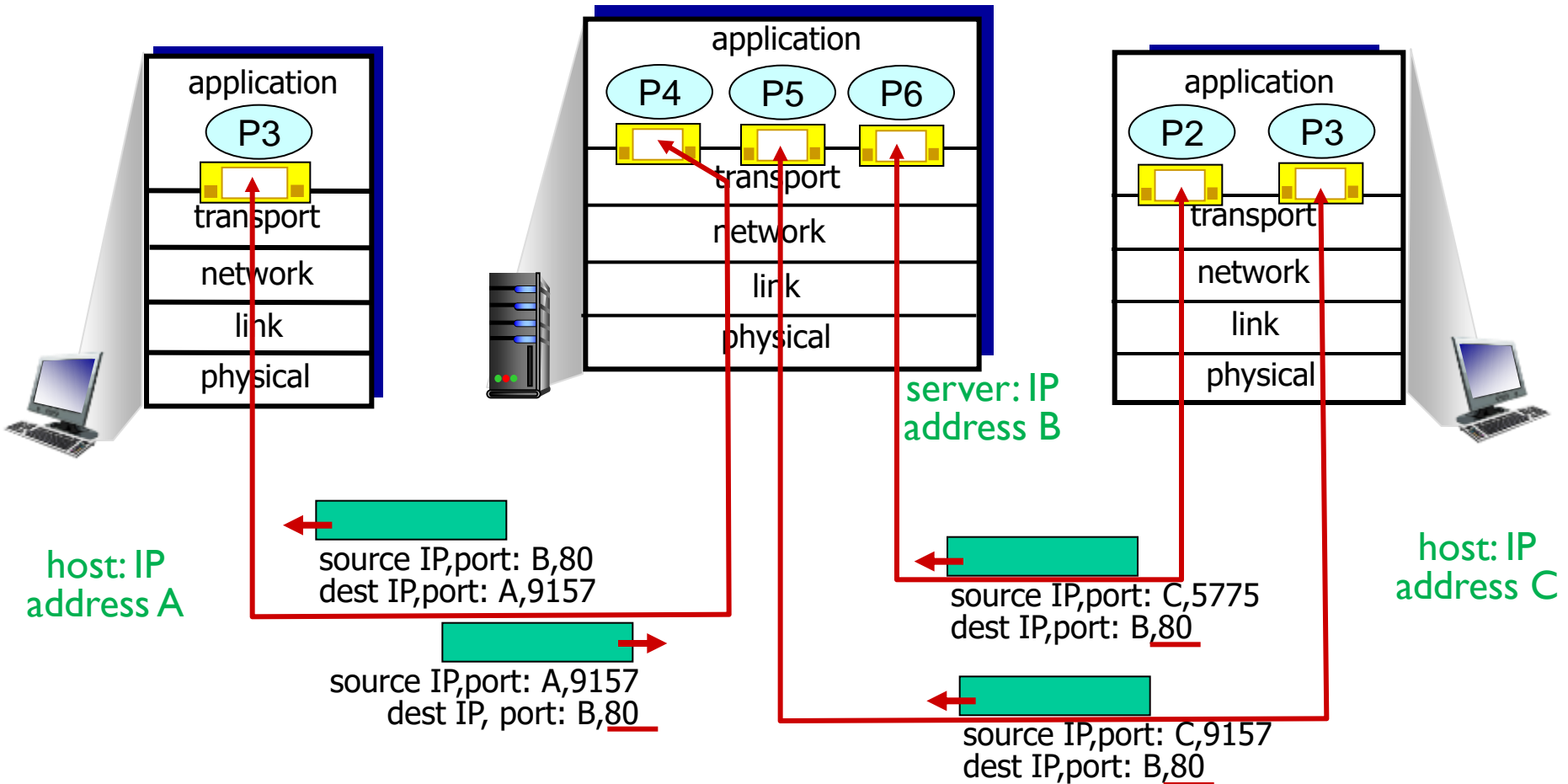
- TCP socket identified by 4-tuple:
 - source IP address
 - source port number
 - dest IP address
 - dest port number
- demux: receiver uses all four values to direct segment to appropriate socket(多路分解, 自下而上)
- 服务器支持很多同时的 TCP sockets:
 - each socket identified by its own 4-tuple
- web servers have different sockets for each connecting client
 - non-persistent HTTP will have different socket for each request

四元组标识

服务器支持更多的socket连接

面向连接的多路分解: 和UDP不同

three segments, all destined to IP address: B,
dest port: 80 are demultiplexed to *different* sockets



Ch3. 传输层

8. 流水线协议：回退N步和选择重传

Go-back-N（回退N步）：

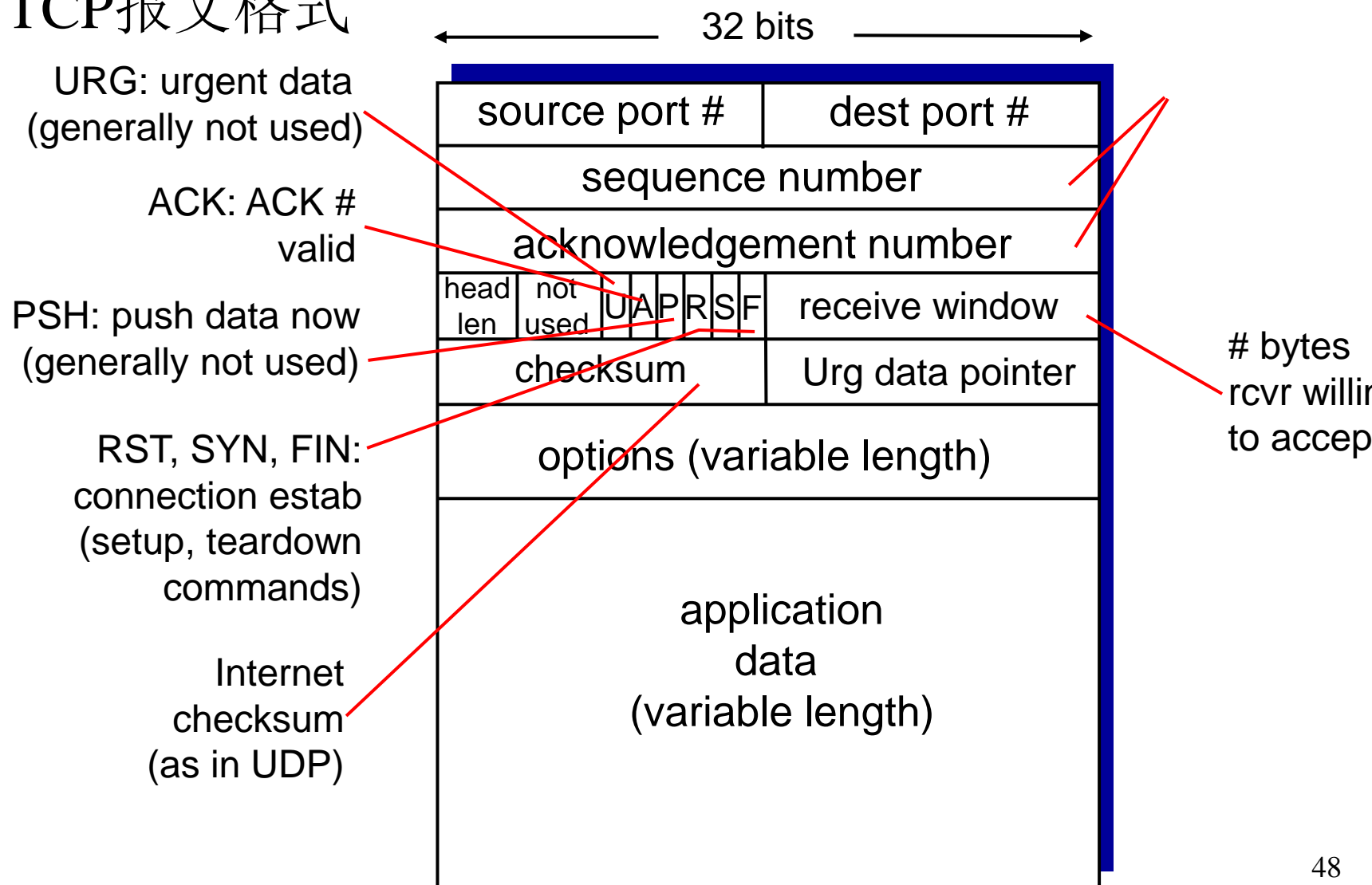
- 发送端可以有N个未被确认的分组在 pipeline 中
- receiver only sends *cumulative ack*
 - Doesn't ack packet if there's a gap（如果序号不连续）
- sender has timer for oldest unacked packet
 - when timer expires, retransmit *all* unacked packets

Selective Repeat(选择重传):

- sender can have up to N unack'ed packets in pipeline
- 接收端 sends *individual ack* for each packet
- sender maintains timer for each unacked(未确认的) packet
 - when timer expires, retransmit only that unacked packet（只重传那些未被确认的包）

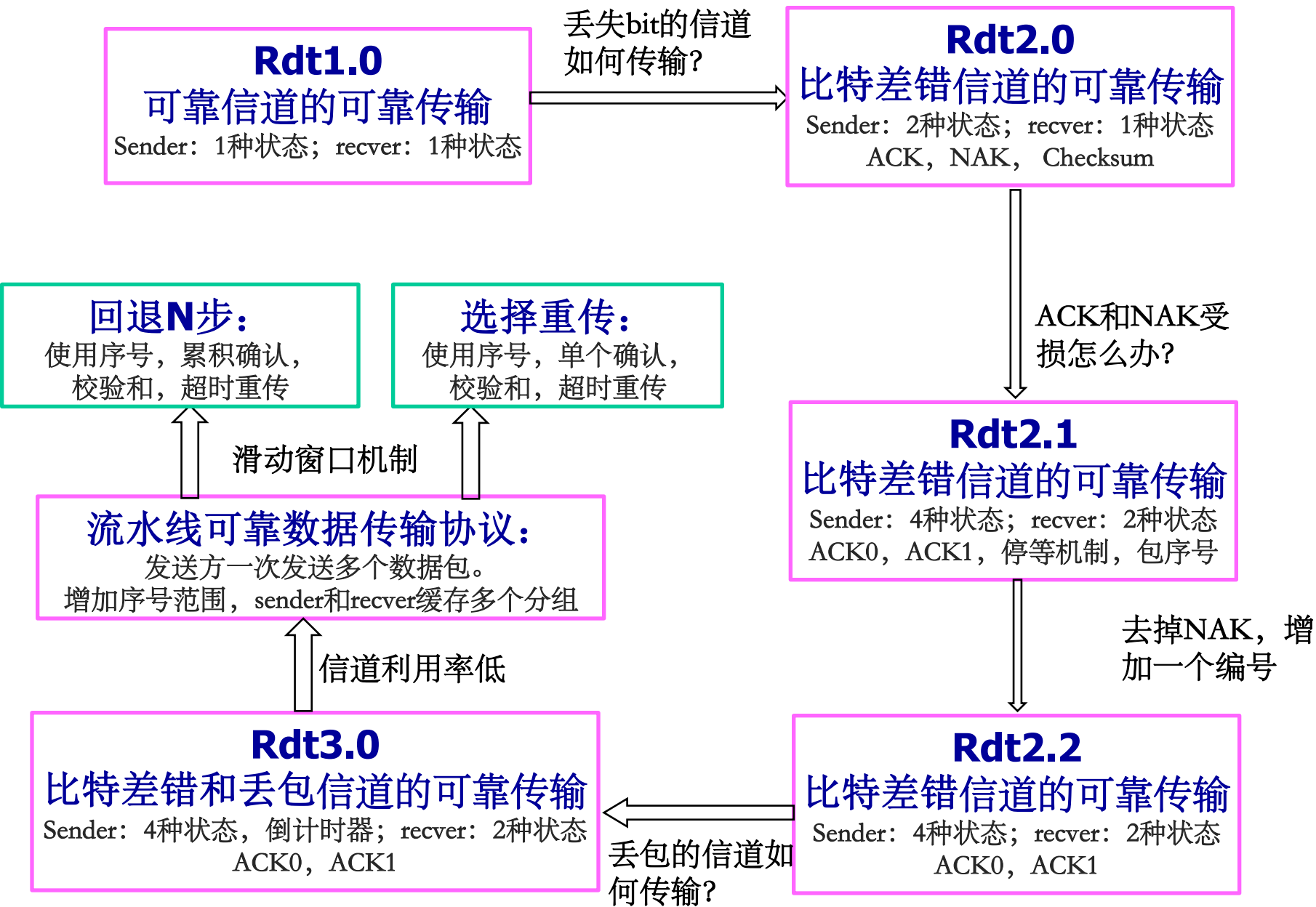
Ch3. 传输层

9. TCP报文格式



Ch3. 传输层

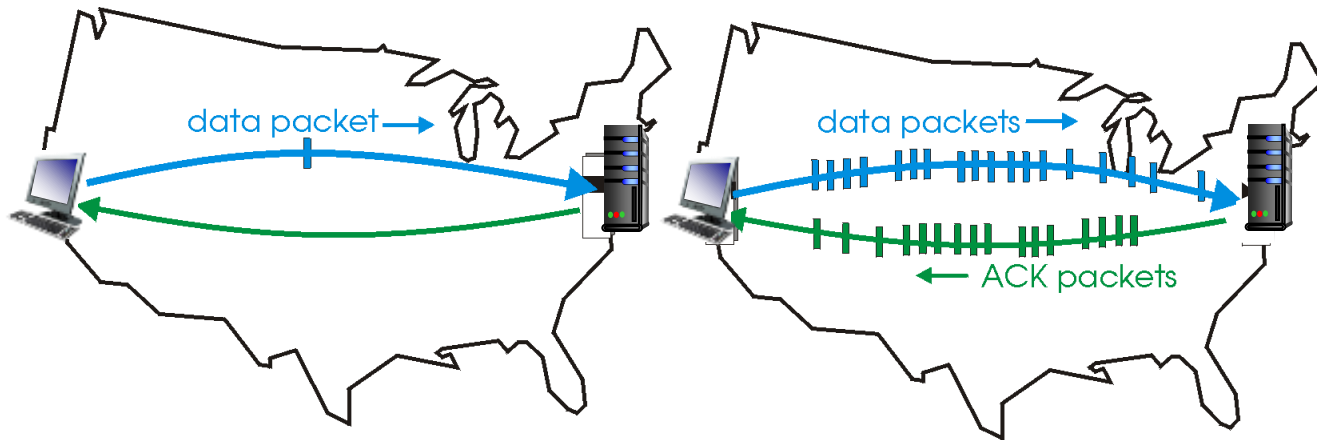
- 10. **TCP**的流量控制和连接：3次握手和4次挥手
- 11. **TCP**的拥塞控制：慢启动等措施
- 12. TCP的公平性



Pipelined protocols (流水线协议)

pipelining: sender allows multiple, range of sequence numbers must be increased (包的序号要增加)

- sender 和 receiver 需要缓存

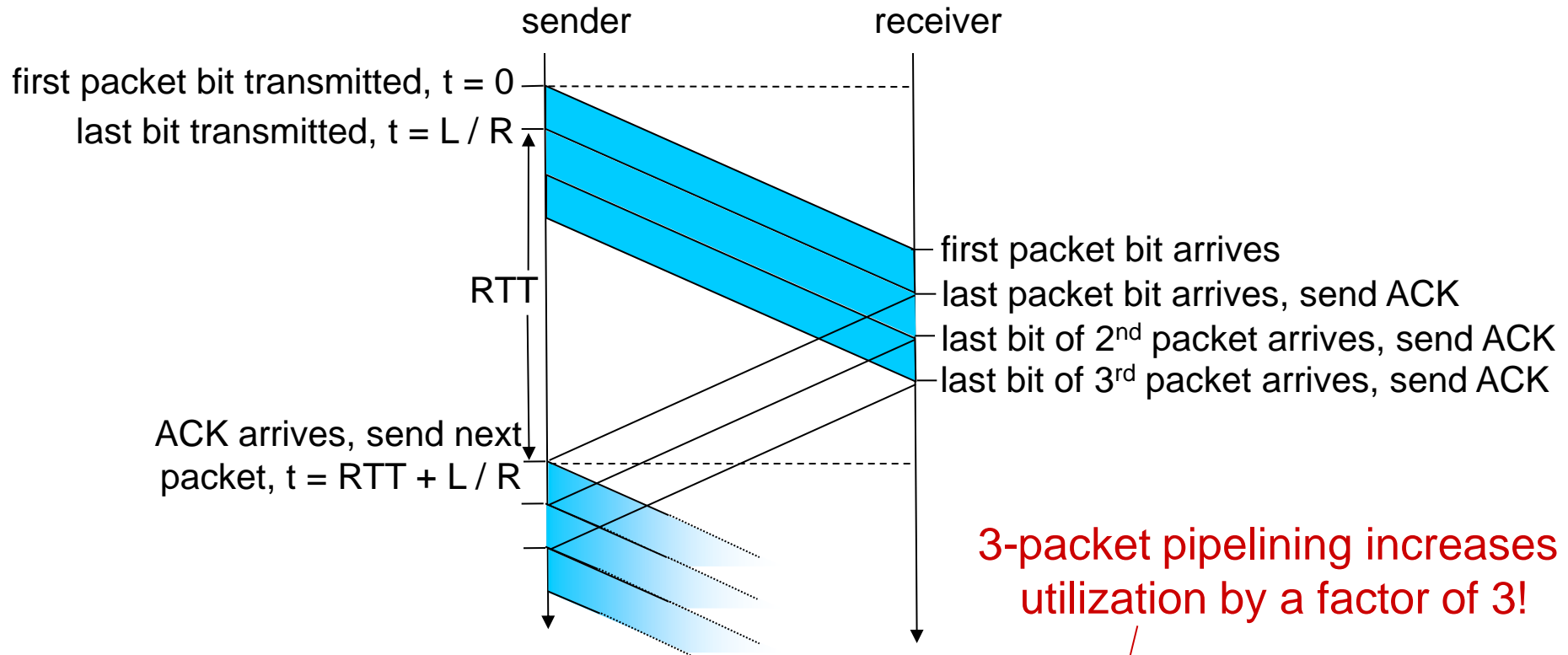


(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

- two generic forms of pipelined protocols: *go-Back-N* (回退N步), *selective repeat* (选择重传)

流水线的利用率



$$U_{\text{sender}} = \frac{3L / R}{RTT + L / R} = \frac{.0024}{30.008} = 0.00081$$

Chapter 4 网络层：数据平面

- 网络层概述：位置，数据单元，两大功能（转发——数据平面、路由——控制平面（两种实现方式）），网络服务模型（Internet网络层提供尽力而为服务）
- 路由器结构：输入端口（功能）、输出端口（功能）、交换结构（功能、三种交换结构）、路由选择处理器（路由协议）
- 分组调度：先进先出、优先权、循环(轮询)、加权公平排队
- IPv4协议：报文格式、分片、IP地址（表示、组成）、子网划分、子网掩码、特殊IP地址
- DHCP的功能和工作原理，NAT的功能和工作原理
- IPv6：产生的动机（3个方面），与IPv4的区别
动机：扩大地址空间；需要进一步加快处理/转发速度；促进QoS。

Ch4. 网络层：数据平面

- 网络层概述：网络层的位置，两大功能（转发——数据平面、路由——控制平面（两种实现方式）），网络服务模型（Internet网络层提供尽力而为服务）
- 路由器结构：输入端口（功能）、输出端口（功能）、高速交换结构（功能、三种交换结构）、路由选择处理器（路由协议）
 - 三种交换方式：总线型、共享内存型、Cross-bar空分结构型
- 分组调度：先进先出、优先级、轮询、加权公平排队
- IPv4协议：报文格式、分片、IP地址（表示、组成）、子网划分、子网掩码、特殊IP地址
- DHCP的功能和工作原理，NAT的功能和工作原理
- IPv6：产生的动机（3个方面），与IPv4的区别

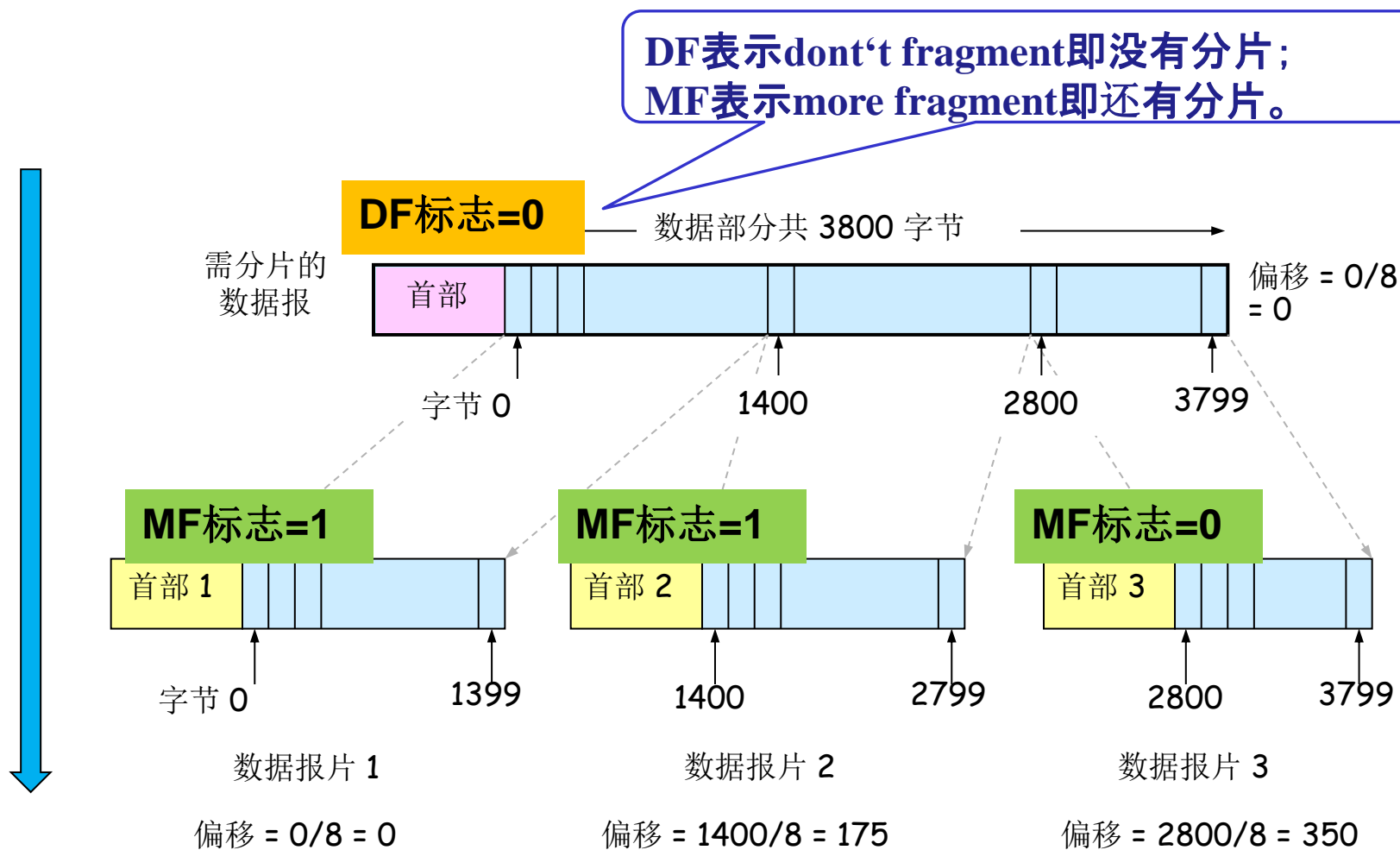
Ch4. 网络层：数据平面

- 网络层协议存在于哪里？
 - 网络层协议存在于每一个主机和路由器
- 一个IP地址和一个接口对应（不一定）
- 路由器中转发发表的主要作用是什么？
 - 路由表的作用是确定分组下一步应该发送到哪里？路由器会根据分组首部的一个或几个字段作为索引，并在转发表中根据索引找到对应的输出链路接口。
- 转发是指将我们对网络层执行的转发功能和路由选择功能进行区别。路由选择和转发的主要区别是什么？
 - 分组从一个输入链路接口转移到适当的输出链路接口的路由由本地动作。转发发生的时间很短，通常用硬件实现。
 - 路由选择是指确定分组从源到目的地所采取的端到端路径的网络范围处理过程。发生的时间较长，通常为几秒，因此用软件实现。

Ch4. 网络层：数据平面

- 路由器核心的功能是什么？
转发和路由。
- 交换结构最常见有哪几种？
总线型、共享内存型、Cross-bar空分结构型

Link layer: MTU=1420 bytes, Network layer: Datagram=3820 bytes
How to make fragments?



原始报文和分片报文具有相同的IP标识 (IP头部字段)

Chapter 5 网络层：控制平面

- 路由选择算法：链路状态算法（主要思想、实例），距离矢量算法（主要思想、实例），两者比较，环路问题和毒性逆转
- 域内路由协议：OSPF的工作原理和功能
- 域间路由协议：BGP（eBGP+iBGP）的工作原理和功能、BGP路由选择策略、区分域内域间路由的原因
- Internet控制报文协议的功能（ICMP）

Ch5. 网络层：控制平面

- 路由选择算法：链路状态算法（主要思想、实例），距离矢量算法（主要思想、实例），两者比较
- 域内路由协议：OSPF的工作原理和功能
- 域间路由协议：BGP（eBGP+iBGP）的工作原理和功能、BGP路由选择策略、区分域内域间路由的原因
- Internet控制报文协议的功能
- SDN的核心思想、架构组成、优点、流表、匹配+动作

全局算法LS 和距离向量DV算法的比较

消息复杂度 (发送报文个数)

- **LS:** with n nodes, E links, $O(nE)$ msgs sent
- **DV:** 在邻居间交换数据
 - 收敛时间不同情况会变化

收敛速度

- **LS:** $O(n^2)$ algorithm requires $O(nE)$ msgs 消息量大
 - 有可能出现震荡现象
- **DV: convergence time varies**
 - 可能出现路由环路
 - **count-to-infinity problem** 计数到无穷大问题

robustness: 路由器发生故障时，算法的健壮性

LS:

- 节点可以发送不正确的链路开销；
- 每个节点计算自己的转发表。

DV:

- 节点可以发送不正确的链路开销。
- 每个节点的转发表给其它节点使用。

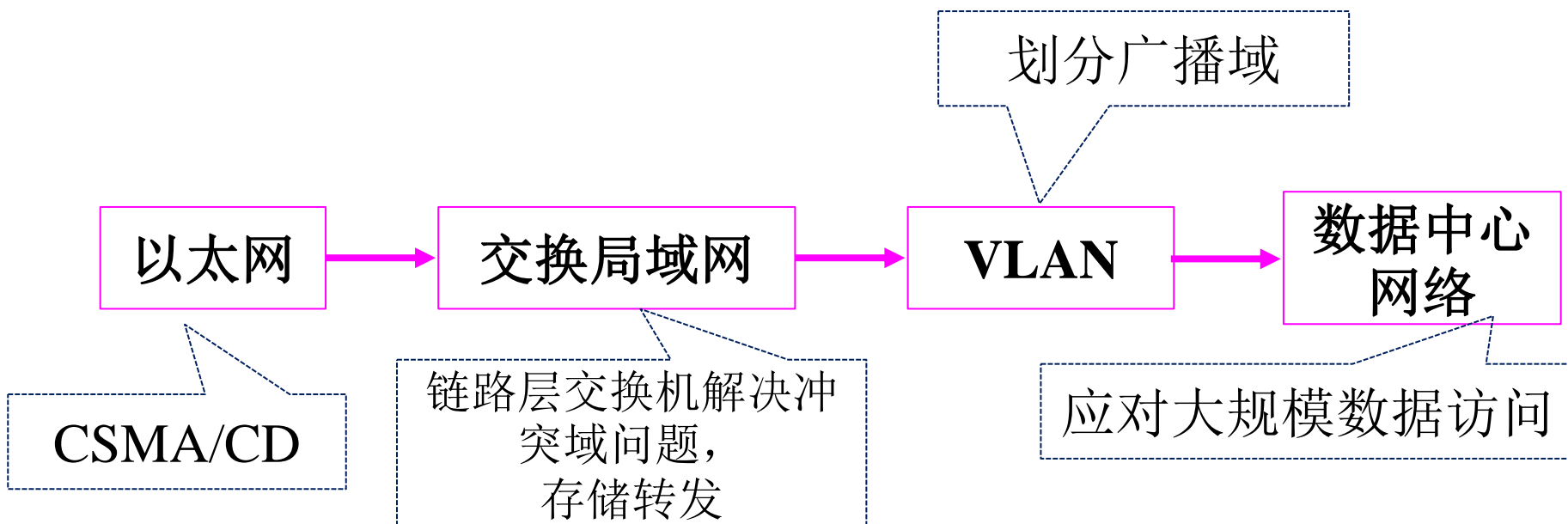
Chapter 6 链路层和局域网

- 链路层概述：功能，数据单元，链路层服务，实现位置
- 差错检测和纠正技术：奇偶校验、检验和、循环冗余检测（CRC）
- 多路访问链路和协议：多路访问的含义，多路访问控制协议类型（3类），ALOHA、时隙ALOHA、轮流协议、CSMA、CSMA/CD的工作原理
- 局域网：局域网的定义和特点，局域网地址（表示和作用），ARP（功能、工作原理、ARP表），发送数据报到子网以外
- 以太网：拓扑结构，帧结构、服务类型（无连接、不可靠）
- 交换机：位置、功能、交换机表（即插即用、自学习），交换机和路由器的比较
- VLAN（划分动机、方法）
- 综合示例(Web页面请求的历程)

Ch6. 链路层和局域网

- 链路层概述：功能，数据单元，链路层服务，实现位置
- 差错检测和纠正技术：奇偶校验、检验和、循环冗余检测（CRC）
- 两种类型网络链路：点对点链路（PPP、HDLC）和广播链路
- 多路访问链路和协议：多路访问的含义，多路访问控制协议类型（3类），ALOHA、时隙ALOHA、令牌环网，轮询，CSMA、CSMA/CD的工作原理
- 局域网：局域网的定义和特点，局域网地址（表示和作用），ARP（功能、工作原理、ARP表），局域网的描述特性（4种），局域网参考模型
- 以太网：拓扑结构，帧结构、服务类型（无连接、不可靠）
- 交换机：位置、功能、交换机表（即插即用、自学习），交换机和路由器的比较
- VLAN（划分动机、方法）、综合示例

Ch6. 链路层和局域网



链路层

差错检测和纠正

奇偶校验

单比特奇偶校验差错检测率为50%

循环冗余校验

每个CRC标准都能够检测小于 $r+1$ 比特的突发差错

多路访问链路协议

信道划分协议

TDM, FDM, CDMA

随机接入协议

时隙ALOHA, 纯ALOHA, CSMA(载波侦听多路复用), CSMA/CD

轮流协议

轮询协议, 令牌传递协议

ARP协议

理解链路层报文传输的4种情况

Chapter 7 无线网络

- CSMA/CA协议
 - 利用RTS和CTS解决了隐藏的终端问题

IEEE 802.11 MAC Protocol: CSMA/CA

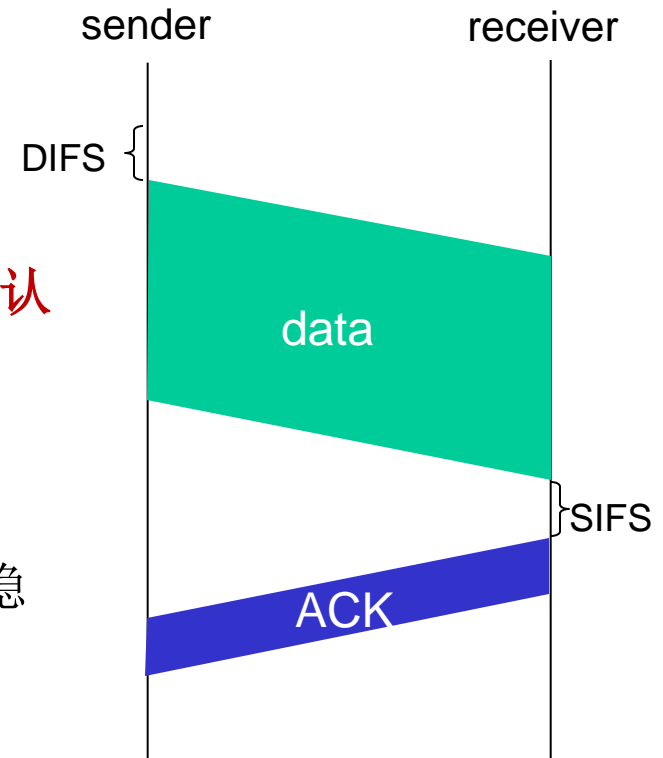
802.11 sender

- 1 如果检测到信道空闲，将在 **DIFS**(分布式帧间间隔) 后发送 entire frame (no CD)
- 2 如果监测到信道繁忙，
选择一个随机回退值，
当信道空闲时，计数器减少
计数器为0时，该站点发送整个帧并**等待确认**
如果未收到确认，循环第2步

802.11 receiver

- 如果接收成功

return ACK after **SIFS** (短帧间间隔) (因为隐藏终端问题，所以要有ACK)



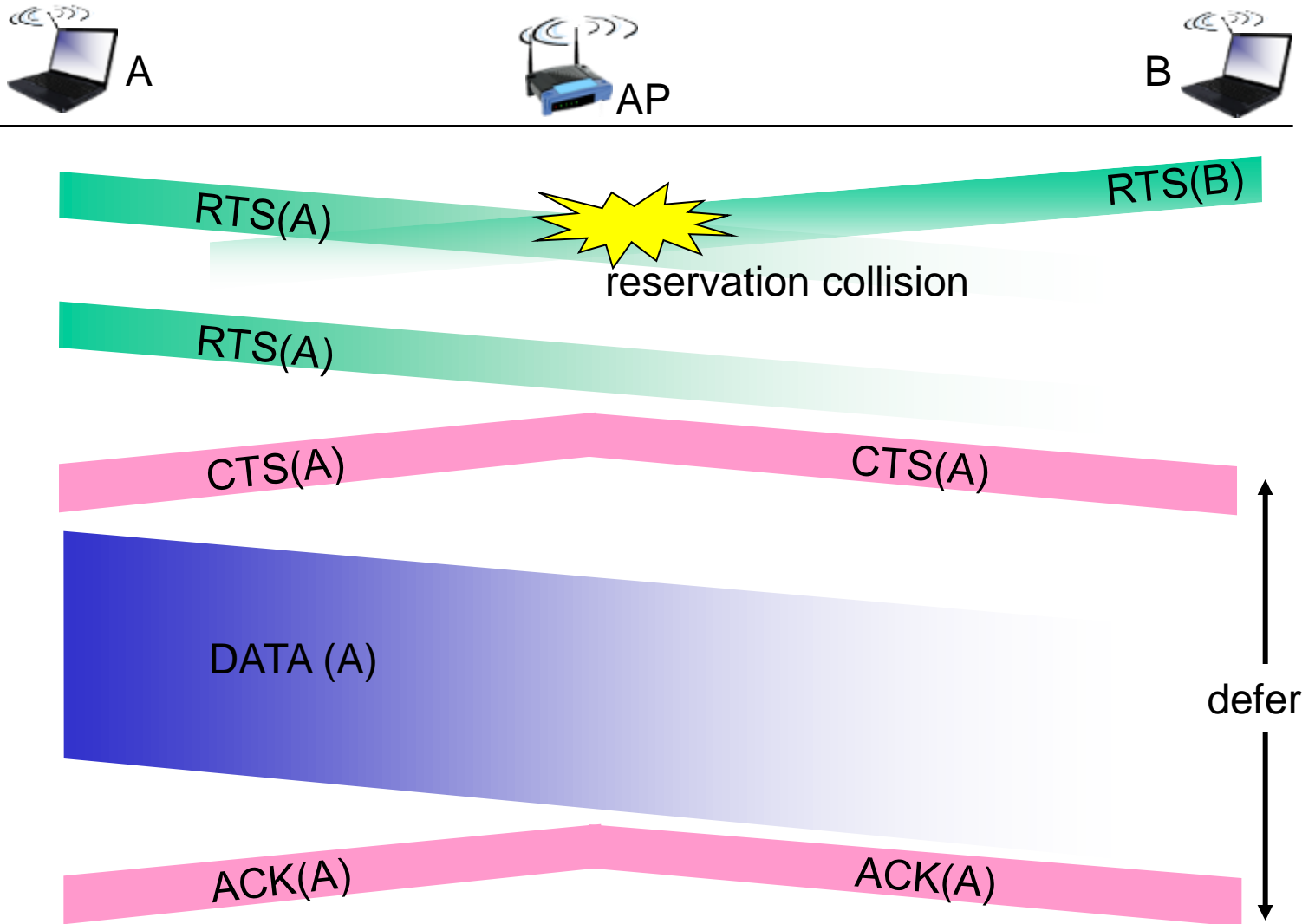
冲突避免 (CA) (使用RTS和CTS)

idea: 发送者为长的数据帧预留信道

- sender first transmits *small* request-to-send (RTS) packets to BS(基站) using CSMA
 - RTSs仍然可能碰撞(but they're short)
- BS broadcasts clear-to-send(CTS) in response to RTS
- CTS 被所有节点接收
 - 发送者发送数据
 - 其它节点延迟发送

预约机制几乎完全避免了冲突!

Collision Avoidance: RTS-CTS 交换



HTTP, SMTP, POP3,
IMAP, DNS, C/S, P2P
socket programming
with UDP and TCP

应用层协议

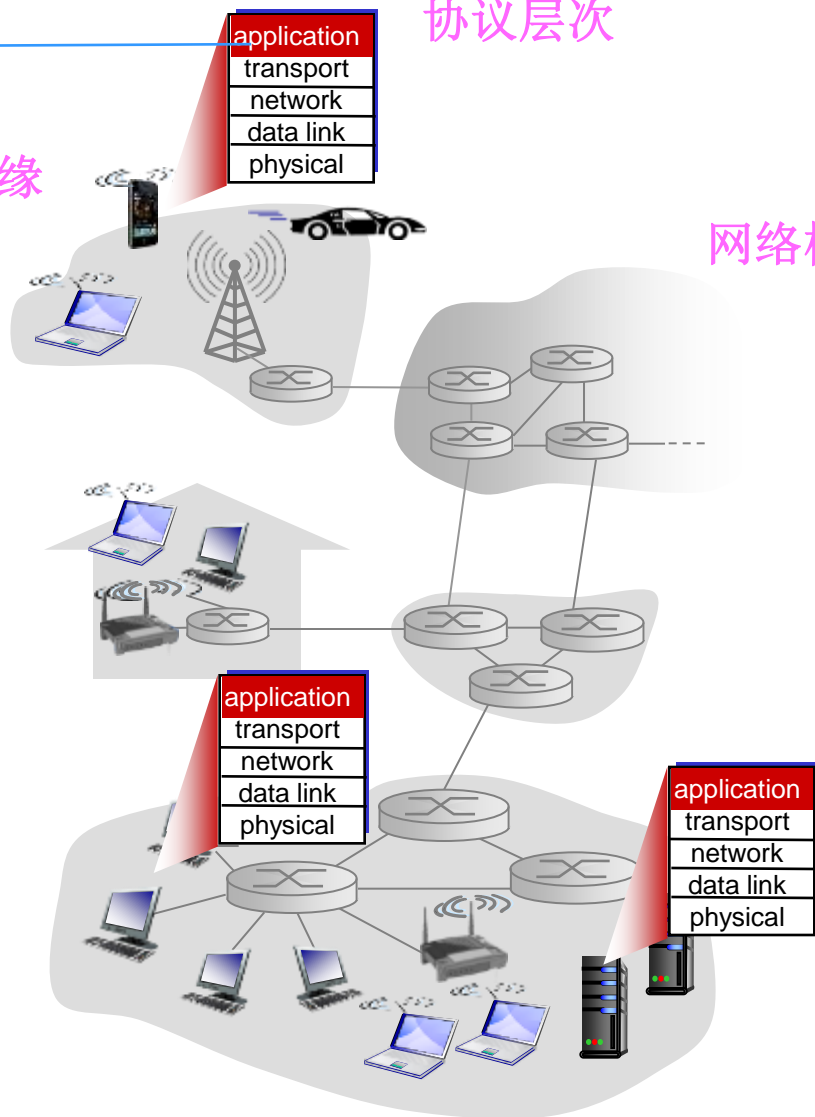
协议层次

网络边缘

网络核心



Delay
Loss
Throughput



各个协议层次所处的物理位置？
数据报文的分布？ 协议的功能？

Some questions

- How many kinds of addresses in computer networks?
- How many kinds of data in computer networks?
- How many kinds of rdt protocols in computer networks?
- How many kinds of MAP in computer networks?
- How many kinds of