



计算机网络 Computer Networks

主讲：洪佩琳
plhong@ustc.edu.cn

Computer

课程主页

<http://if.ustc.edu.cn/course/>

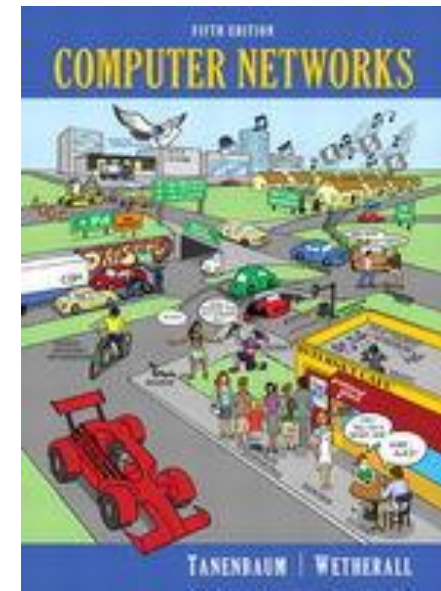
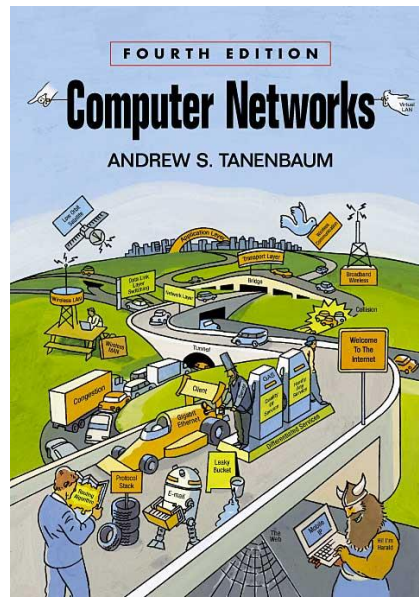
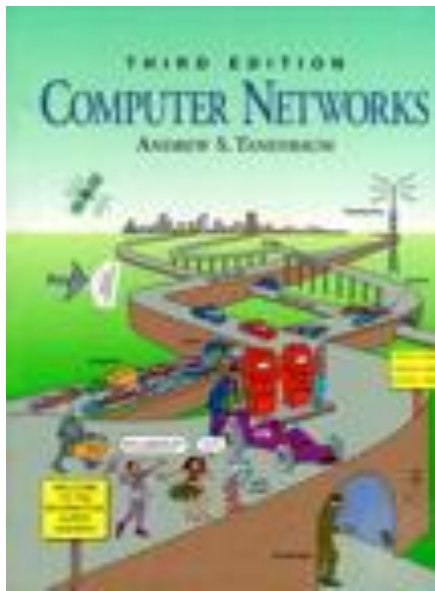
辅导：王航，盛家华
[yuyingju](mailto:yuyingju@mail.ustc.edu.cn), jhsheng@mail.ustc.edu.cn

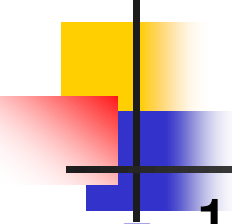
教科书和参考书

Textbook & Reference

- Textbook: Andrew S. Tanenbaum, Computer Networks, 5rd ed, Prentice Hall, 2011

中文版：“计算机网络”（第5版），潘爱民译，清华大学出版社



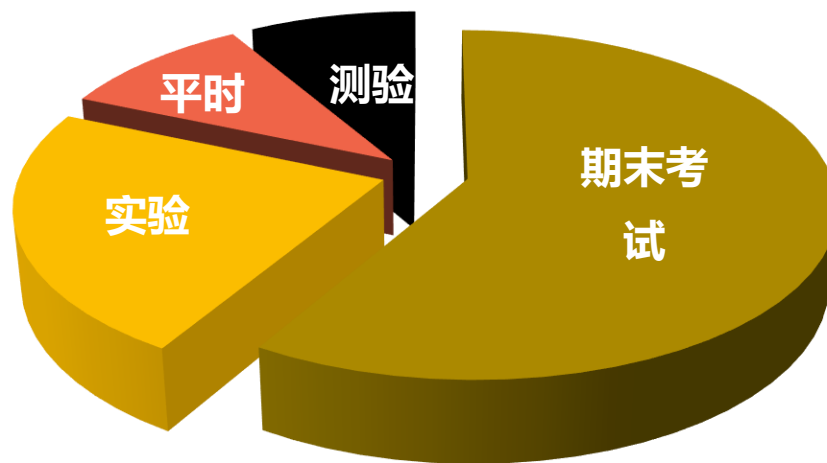


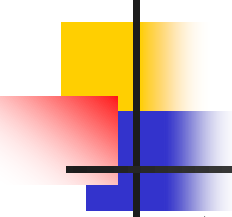
References:

- 1. JAMES F.KUROSE,KEITH W.ROSS, “计算机网络——自顶向下方法”（第7版），陈鸣等译，机械工业出版社，2018.6
- 2. 谢希仁，“计算机网络”，（第7版）电子工业出版社，2017

课程安排和成绩评定

- 课程安排：60学时授课，20学时实验
- 成绩评定：
 - 期末考试:60%
 - 平时作业+平时测验:20%
 - 实验成绩:20%





什么是计算机网络？

- 通过同一种技术（通信线路和网络设备按某些协议/通信技术和通信协议）相互连接起来的（具有通信功能的）一组自主计算机的集合^[1]。
- 什么是因特网（Internet）？
 - 指全球最大的、开放的，由众多网络相互连接而成的计算机网络，它由美国阿帕网（ARPAnet）发展而成，主要采用TCP/IP协议。（96年的定义）
 - 起源：美国阿帕网（Advanced Research Projects Agency Network, ARPAnet）
 - 协议：TCP/IP——版本升级
 - 端系统：计算机——手机、电话、PDA、信息家电、汽车等一切需要联网的物体

互联网应用产生的商机



课程内容

1. 概述
2. 网络的体系结构与参考模型
3. 物理层
4. 数据链路层
5. 局域网——以太网和WIFI
6. 网络层
7. 互联网——Internet
8. 传输层
9. 应用层
10. 网络安全

课程内容

■ CH1. 概述

- 主要介绍互联网的发展历史及其应用、标准化组织。

■ CH2. 网络的体系结构与参考模型

- 网络体系结构(**architecture**)就是网络的顶层设计, 将网络功能分层。后续3-9章从低到高描述各层功能及相关协议。
- 主要介绍其基本概念和两种参考模型。

■ CH3. 物理层

- 主要介绍网络**通信**相关的基本概念, 如传输速率、传输介质、调制、复用以及交换。

■ CH4. 数据链路层

- 主要介绍数据链路层的基本功能与协议设计。

■ CH5. 局域网——以太网和WIFI

- 例举两个局域网的数据链路层协议, 链路层的互连。

数据链路层

课程内容

- CH6. 网络层

- 网络层的基本功能，路由转发、拥塞控制。

- CH7. 互联网——Internet

- Internet的网络层协议IP及其它相关协议。

- CH8. 传输层

- Internet的传输层协议TCP\UDP。

- CH9. 应用层

- 列举几个经典的Internet应用层协议。

- CH10. 网络安全

- 介绍网络安全的基本概念。

网络层



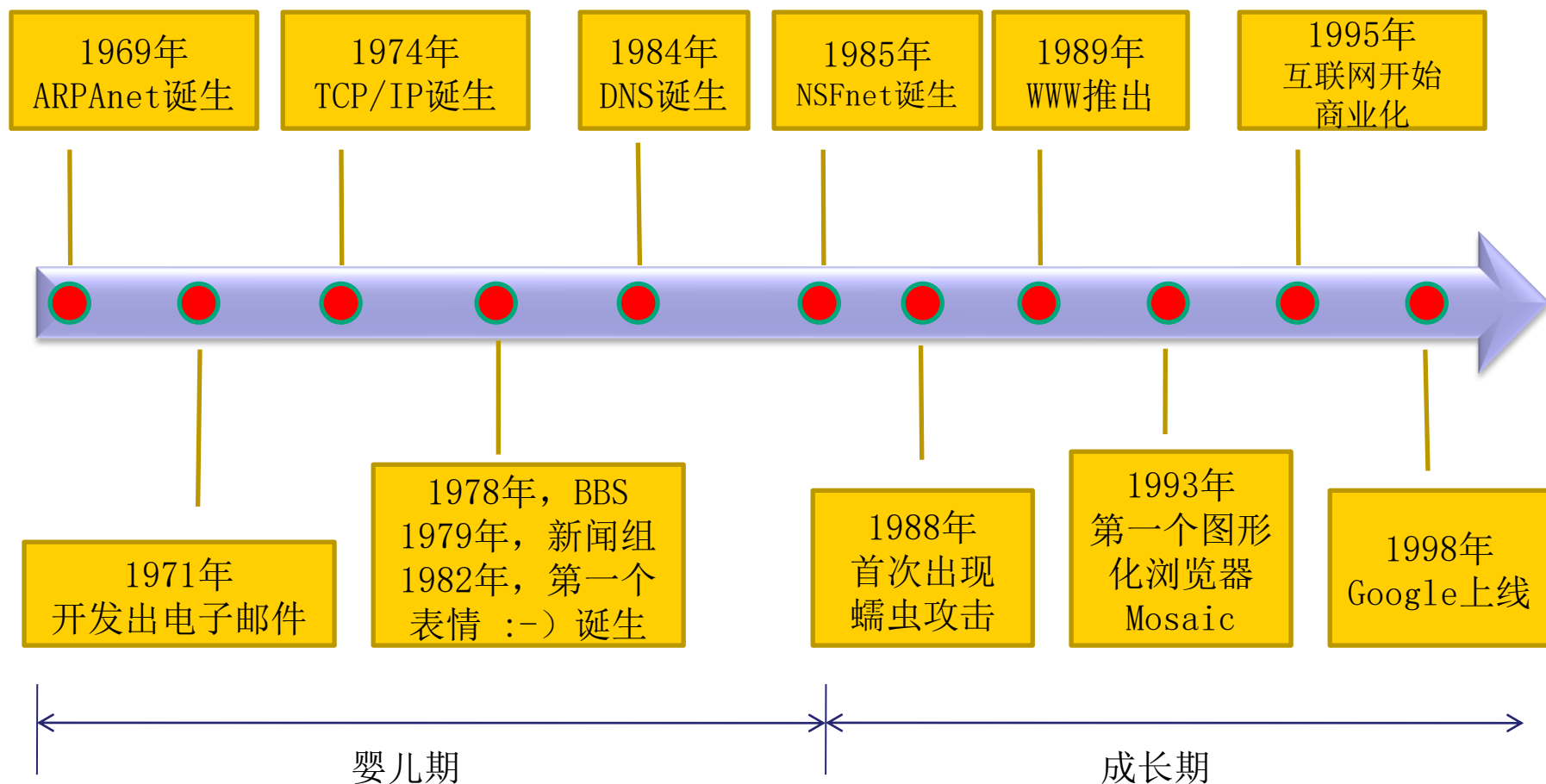
Chapter 1 概述

- **1.1 Internet 的发展史**
- **1.2 中国Internet的发展史**
- **1.3 IPv6发展史**
- **1.4 Internet的应用**
- **1.5 互联网与物联网**
- **1.6 网络的标准和标准化组织**

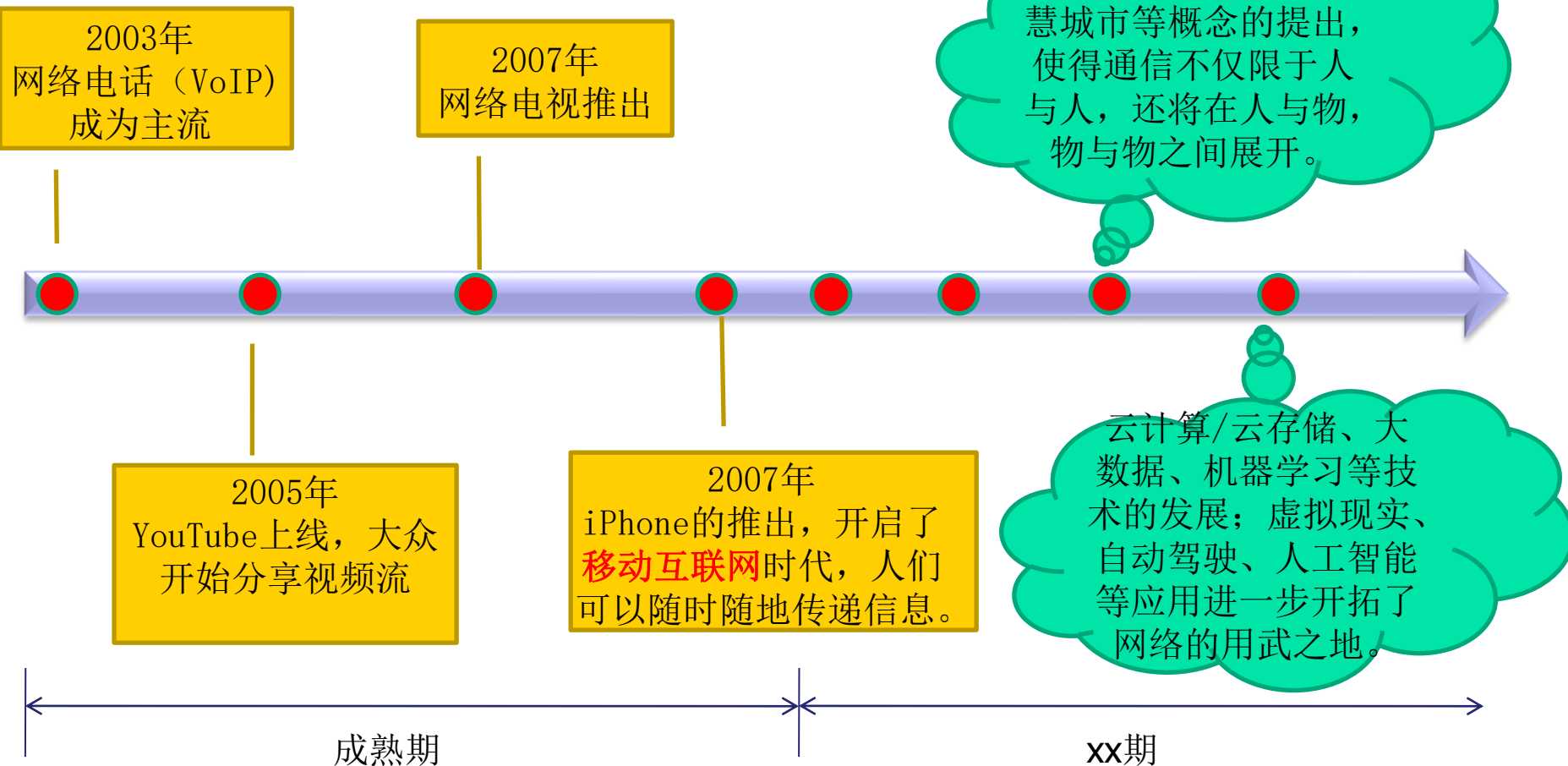
1.1Internet 的发展史

- **Internet——因特网（国际互连网，互联网（**internet**））**
 - 指全球最大的、开放的，由众多网络相互连接而成的计算机网络，它由美国阿帕网（ARPAnet）发展而成，主要采用TCP/IP协议。
- **历史**
 - 1969年起源于美国国防部高级研究计划署的ARPAnet
 - 1979年，研究开发TCP/IP，80年代TCP/IP-》UNIX
 - 1985年，美国国家科学基金会NSF投入TCP/IP的研发，建成NSFNet
 - 90年代后，Internet飞速发展，入网的用户数呈指数增长
 - 93年美国提出建立信息高速公路（国家信息基础设施，NII，National Information Infrastructure），一个贯通全美各大学、研究机构、企业及家庭的全国性网络。之后，GII（Global Information Infrastructure）在全球掀起
 - 1997年7月，美国提出了NGI（Next Generation Internet）
 - 2005年美国NSF启动GENI（Global Environment for Networking Innovation）和FIND（Future INternet Design）
 - 2005年国际电联的报告上提出了物联网概念（**Internet of Things**）

Internet技术及应用的发展历程（1）



Internet技术及应用的发展历程（2）



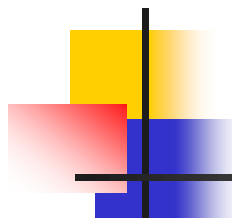


Chapter 1 概述

- **1.1 Internet 的发展史**
- **1.2 中国Internet的发展史**
- **1.3 IPv6发展史**
- **1.4 Internet的应用**
- **1.5 互联网与物联网**
- **1.6 网络的标准和标准化组织**

1.2 中国Internet的发展史

- 1989年9月，国家计委向世界银行贷款建设中关村地区**教育与科研示范网络**（**NCFC, The National Computing and Networking Facility of China**），1992年**NCFC**工程全部完成。（后改名中国**科技网****CSTNet**）
- 1994年5月，**NCFC**代表中国正式加入**Internet**，向**InterNIC**注册**CN**域名。
- 1994年9月，中国**公用计算机互联网**（**Chinanet**）建设正式启动，**96年1月**正式开通。
- 1994年10月，中国**教育科研网**（**Cernet, China Education and Research Network**）开始启动。
- 1996年9月，中国**金桥网**（**ChinaGBN**）正式开通。
- 1997年，**Chinanet**与**CSTNet**、**Cernet**、**ChinaGBN**互连互通。



1.2中国Internet的发展史（续）

- 1997年底，中国互联网络信息中心(CNNIC)发布了**第一次**《中国互联网络发展状况统计报告》：截止到**1997年10月31日**，中国共有上网计算机**29.9万台**，上网用户数**62万**，**CN**下注册的域名**4066个**，**WWW**站点约**1500个**，国际出口带宽**25.408M**。
- 2006年1月，**CNNIC第十七次**统计报告：中国共有上网计算机总数**4950万台**，用户数**11100万**，**CN**下注册的域名**1096924个**，**WWW**站点约**694200个**，国际出口带宽**136,106M**。
- 2016年1月，**CNNIC第三十七次**统计报告：全国网民规模**6.88亿**，手机网民规模**6.20亿**，**CN**下注册的域名**1636万个**，**WWW**站点约**423万个**，国际出口带宽**5,392,116M**，年增长率为**30.9%**。

中国互联网应用的发展史

中国互联网发展史

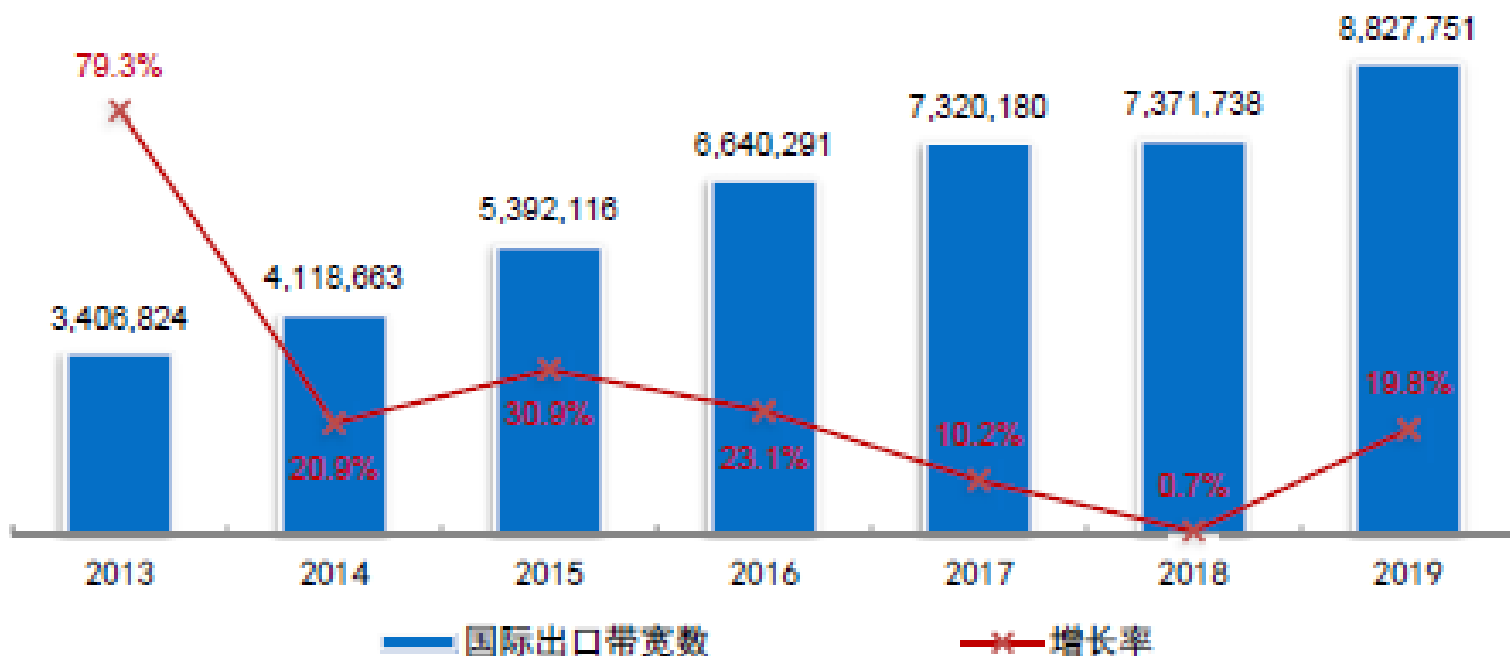


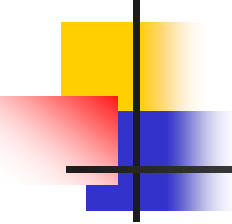
网络出口带宽（截至到2019年12月）

■ 第45次中国互联网络发展状况统计报告

国际出口带宽数及增长率

单位：Mbps



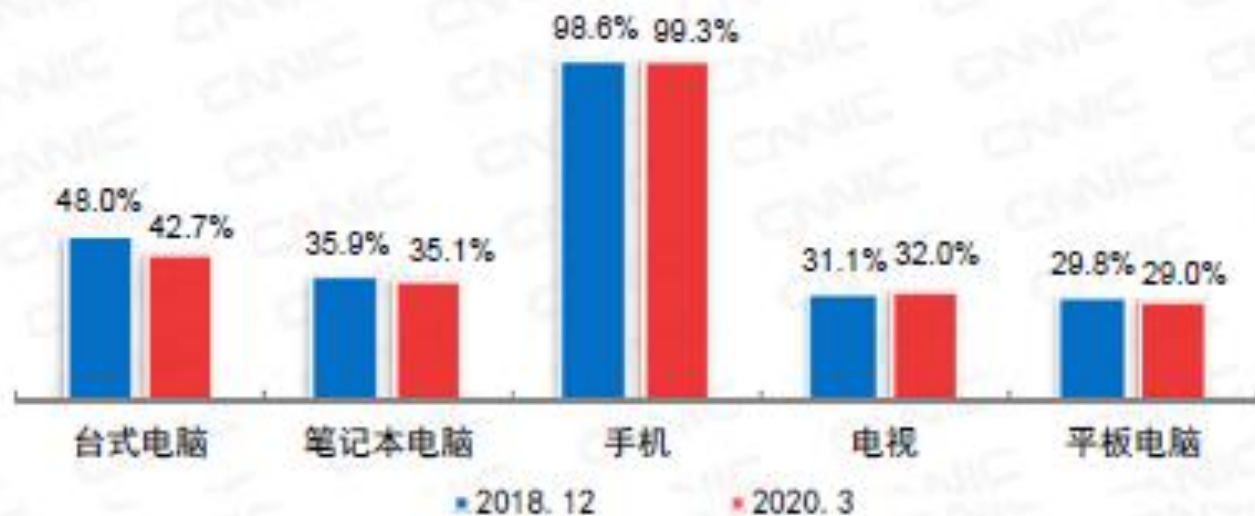


国际出口带宽

	国际出口带宽数 (Mbps)
中国电信 中国联通 中国移动	8,651,623
中国科技网	114,688
中国教育和科研计算机网	61,440
合计	8,827,751

互联网接入环境

互联网络接入设备使用情况



来源: CNIC 中国互联网络发展状况统计调查

2020.3



来源: CNIC 中国互联网络发展状况统计调查

2017.12

网民平均每天上网时间4.4小时

网民人均每周上网时长

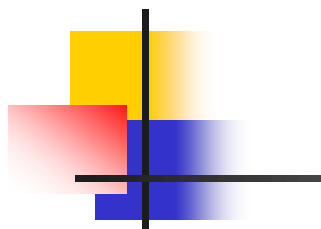
单位：小时



来源：CNIC 中国互联网络发展状况统计调查

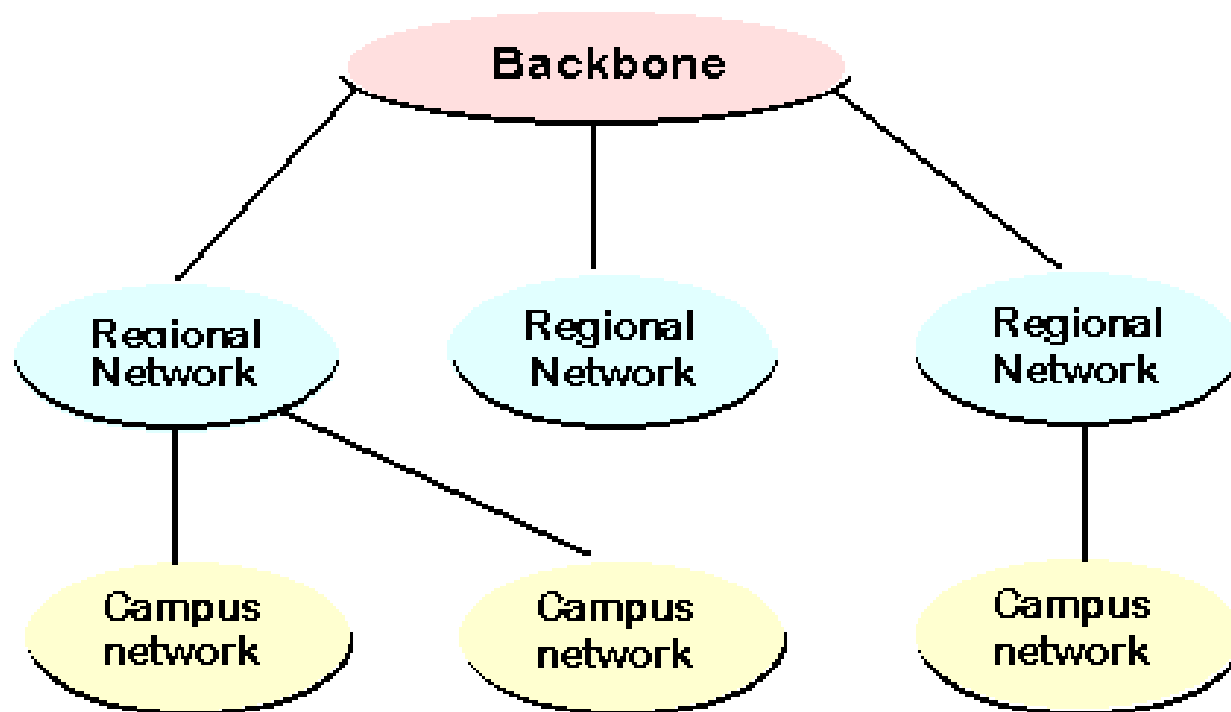
2020.3

近年来互联网各类应用用户规模及使用率

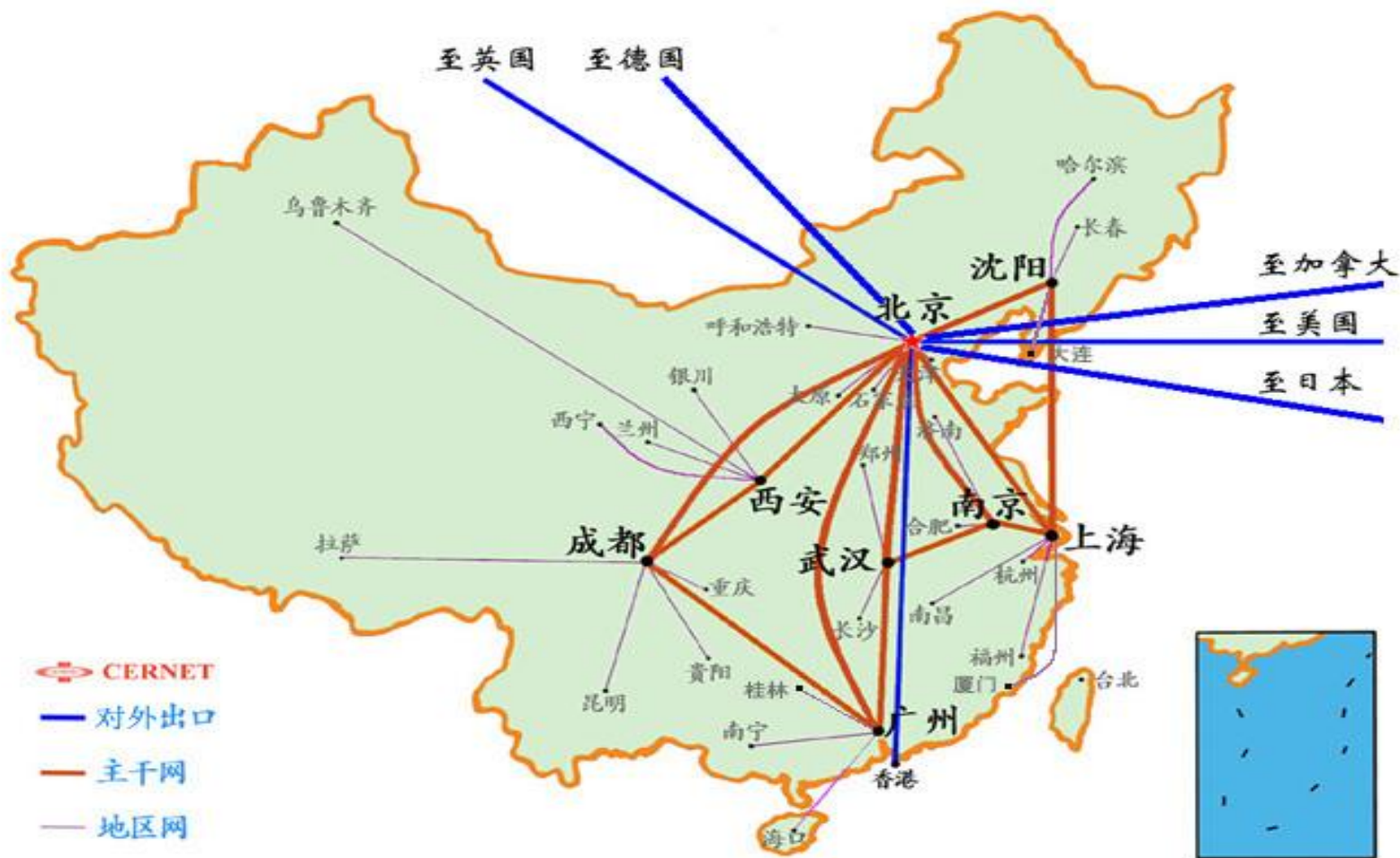


	2020.3		2018.12		
应用	用户规模 (万)	网民使用率	用户规模 (万)	网民使用率	增长率
即时通信	89613	99.2%	79172	95.6%	13.2%
搜索引擎	75015	83.0%	68132	82.2%	10.1%
网络新闻	73072	80.9%	67473	81.4%	8.3%
网络支付	76798	85.0%	60040	72.5%	27.9%
网络购物	71027	78.6%	61011	73.6%	16.4%
网上外卖	39780	44.0%	40601	49.0%	-2.0%
旅行预订 ³⁶	37296	41.3%	41001	49.5%	-9.0%
网约车	36230	40.1%	38947	47.0%	-7.0%
在线教育	42296	46.8%	20123	24.3%	110.2%
网络音乐	63513	70.3%	57560	69.5%	10.3%
网络文学	45538	50.4%	43201	52.1%	5.4%
网络游戏	53182	58.9%	48384	58.4%	9.9%
网络视频 (含短视频)	85044	94.1%	72486	87.5%	17.3%
短视频	77325	85.6%	64798	78.2%	19.3%
网络直播 ³⁷	55982	62.0%	39676	47.9%	41.1%
互联网理财	16356	18.1%	15138	18.3%	8.1%

Cernet的网络结构

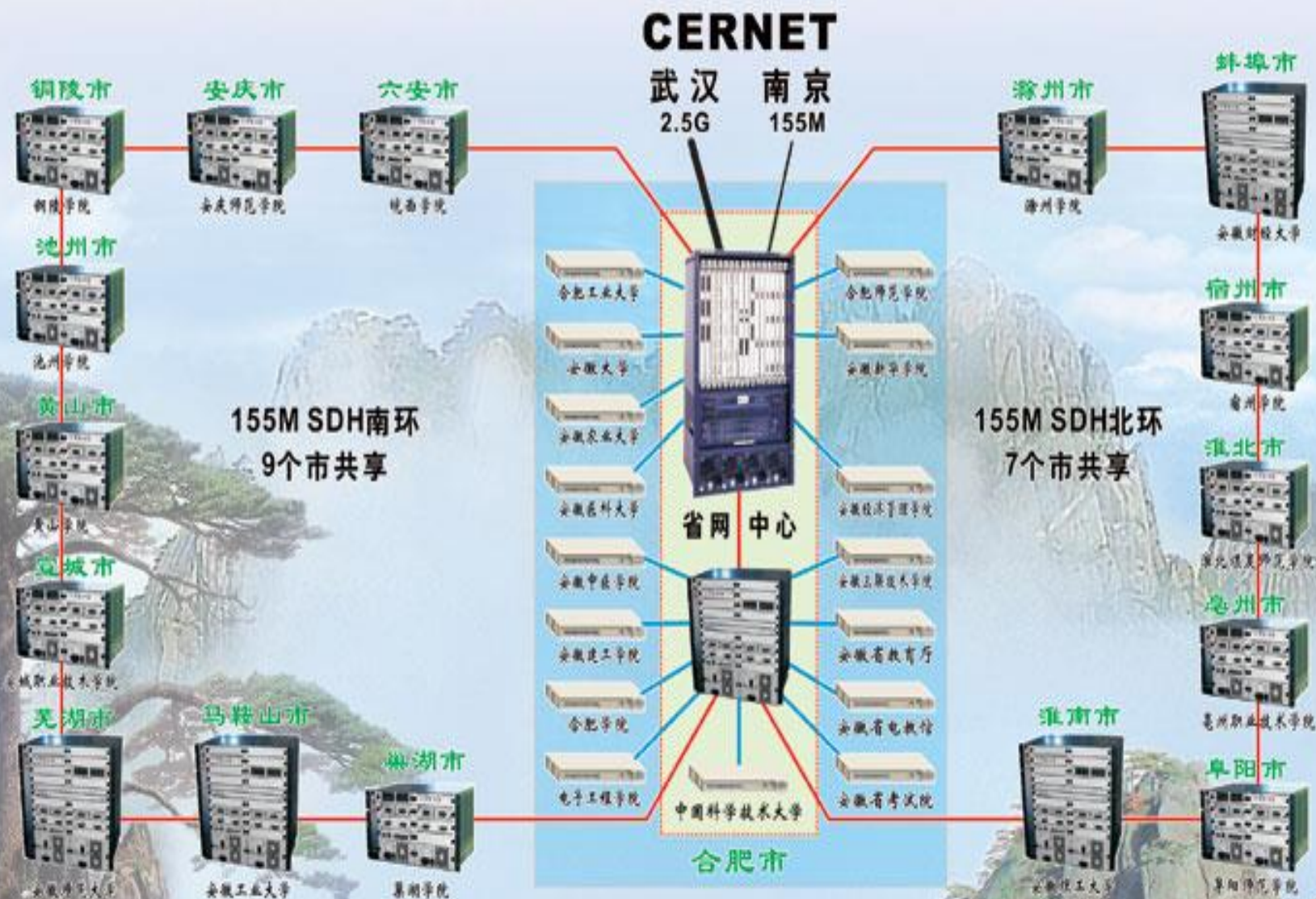


CERNET拓扑结构

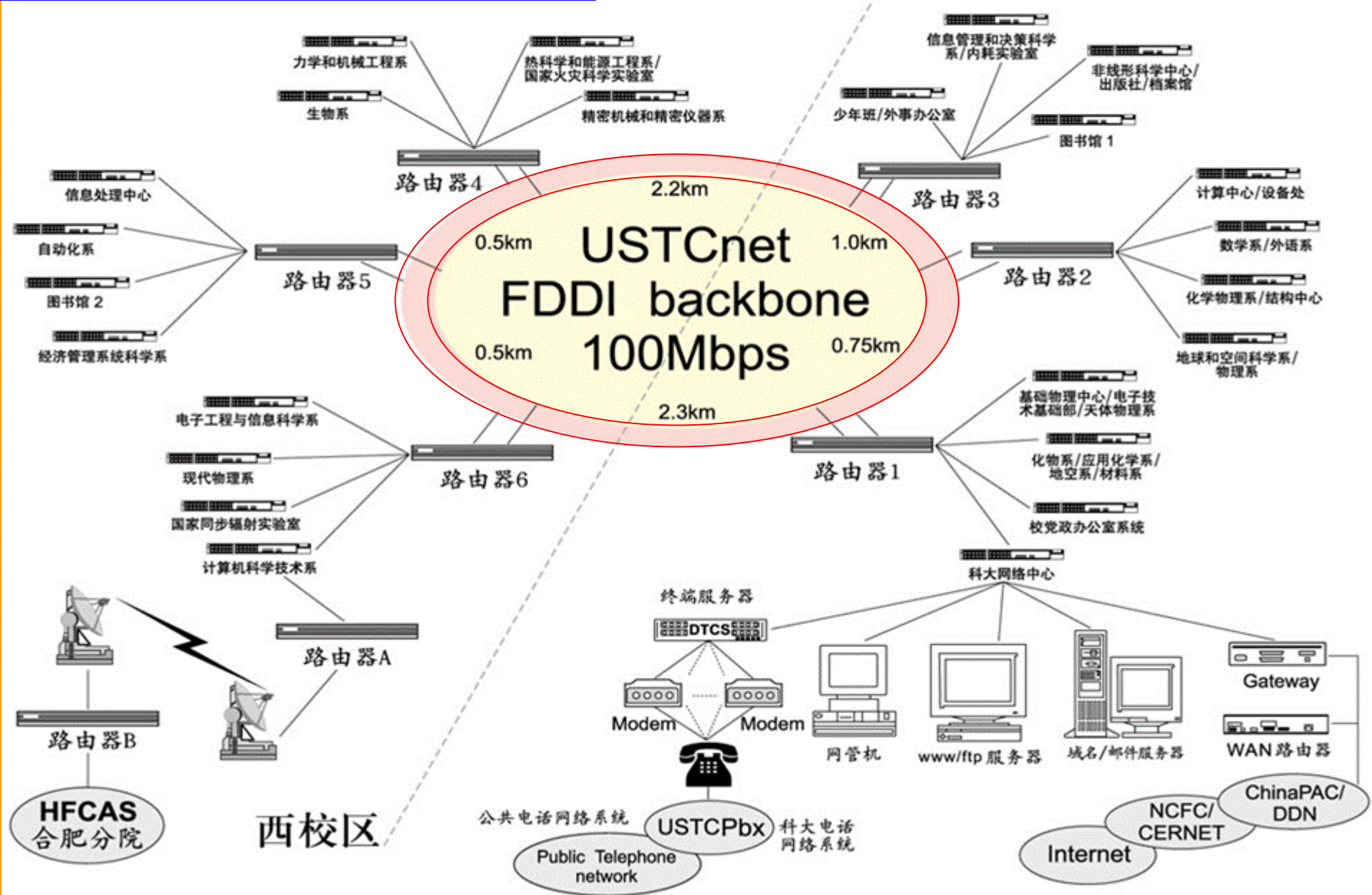




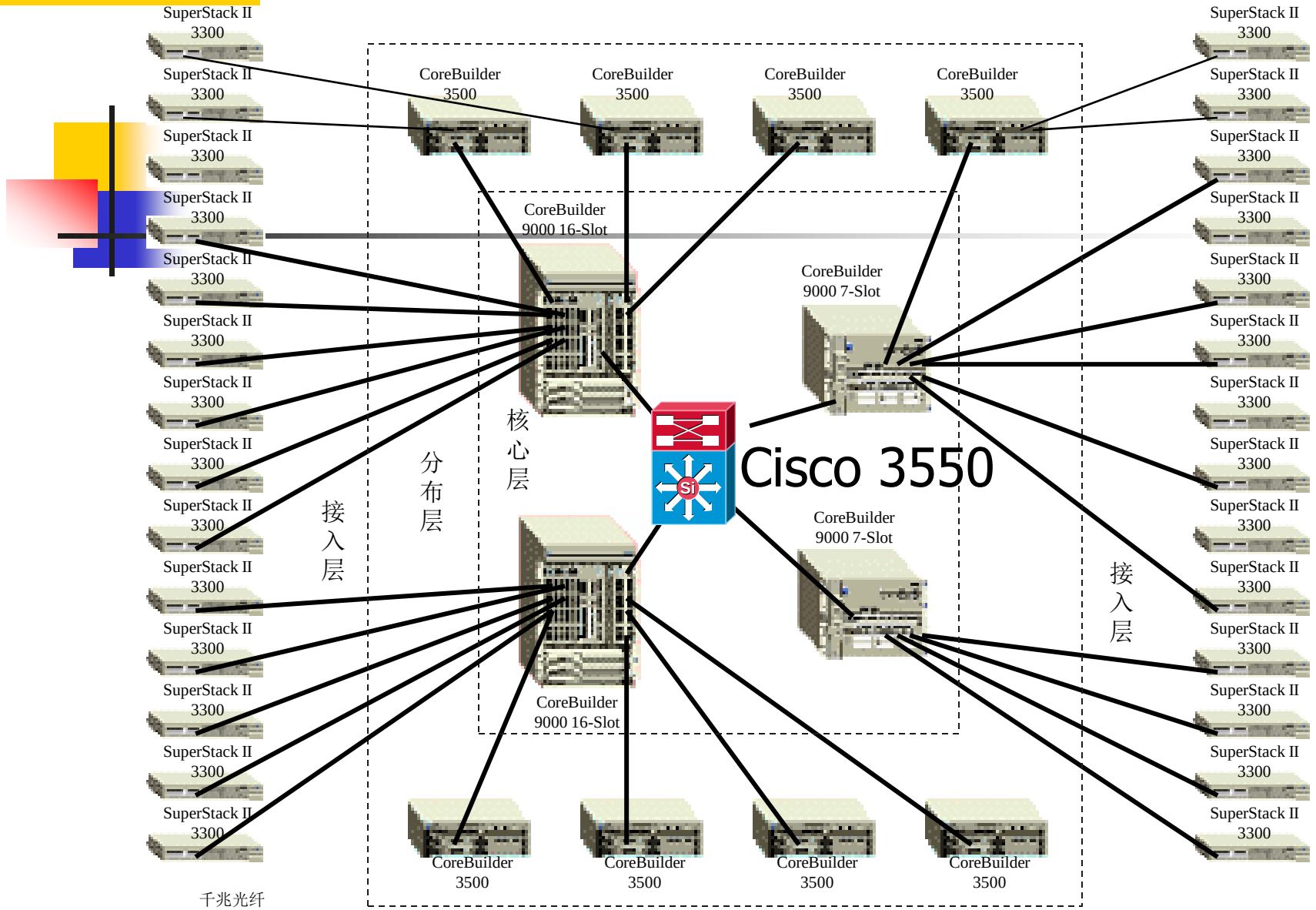
安徽省教育科研网络主干示意图



FDDI 主干, 24个10M子网, 运行IP、IPX协议(1期)

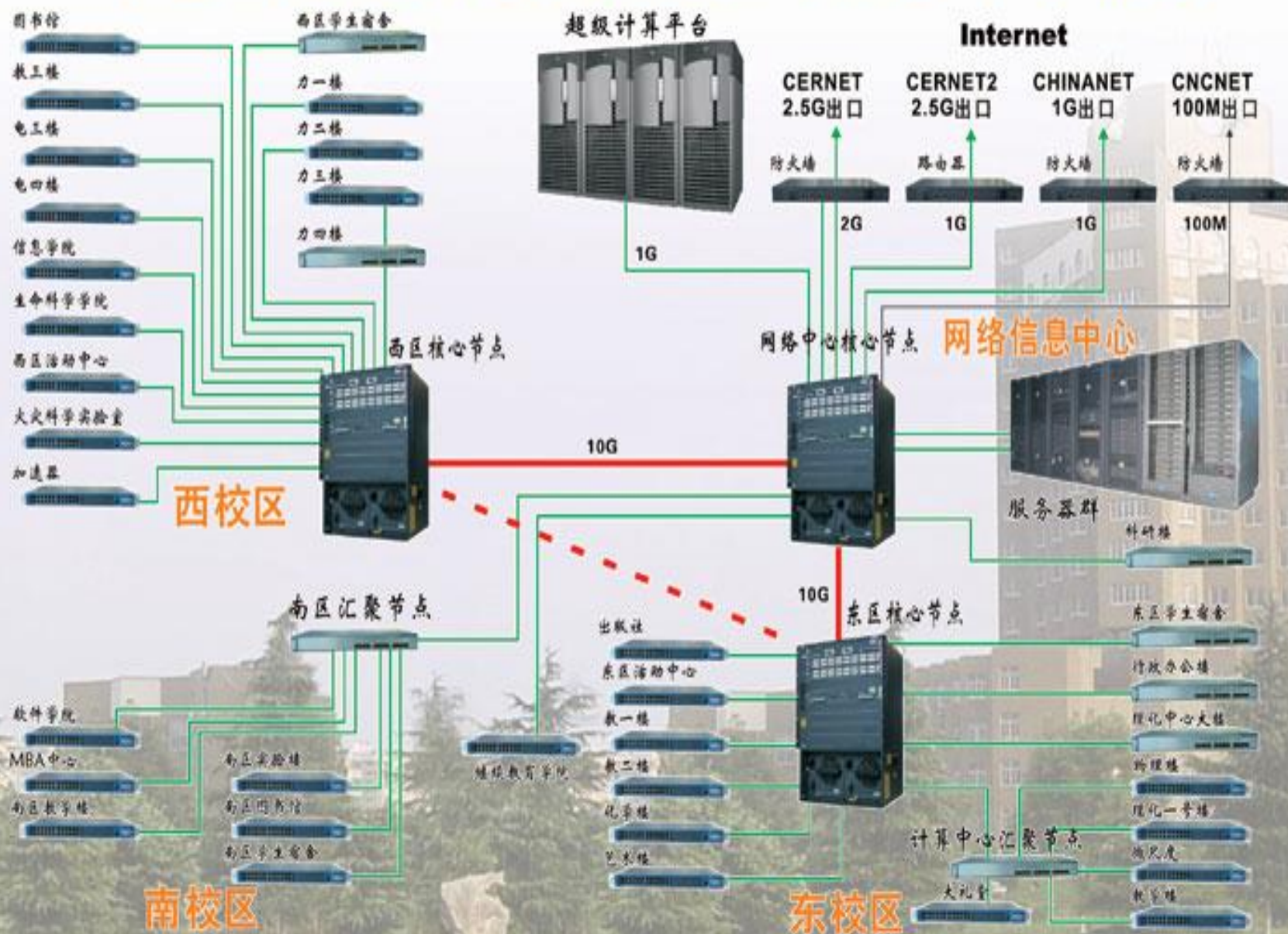


第一章：概述



中国科大校园网主干修改结构图 2002

中国科学技术大学校园网络主干示意图



中国科技大学无线网络分布图



■ <http://linux.ustc.edu.cn/ap/>



Chapter 1 概述

- **1.1 Internet 的发展史**
- **1.2 中国Internet的发展史**
- **1.3 IPv6发展史**
- **1.4 Internet的应用**
- **1.5 互联网与物联网**
- **1.6 网络的标准和标准化组织**

1.3 IPv6发展史

■ **IPv6 is sometimes also called the Next Generation Internet Protocol or IPng.**

■ **国际:**

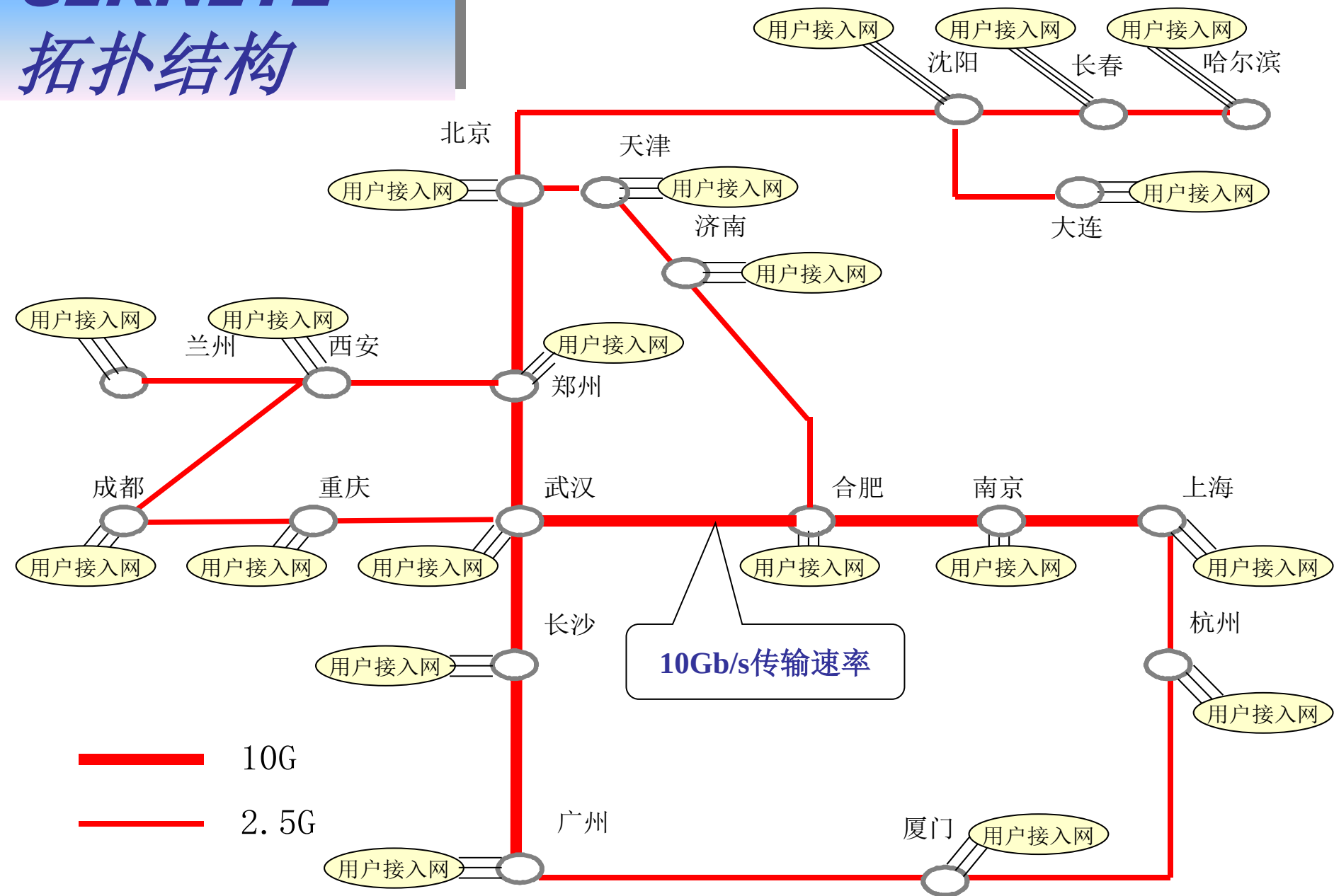
- **IPv6 was recommended by the IPng Area Directors of the Internet Engineering Task Force at the Toronto IETF meeting on July 25, 1994 in RFC 1752**
- **The core set of IPv6 protocols were made an IETF Draft Standard on August 10, 1998. , RFC2460,“Internet Protocol, Version 6 (IPv6), DEC,1998 Specification ”**
- **1996年3月国际IPv6主干网（6bone）建成**

■ **国内:**

- **1998年4月，CERNET正式参加下一代IP协议(IPv6)试验网6BONE。8月国家863计划启动“IPv6示范系统”**
- **2002年1月,国家启动“下一代互联网中日IPv6合作项目”。**
- **2003年8月，国务院批复同意国家发改委等八部委“关于推动我国下一代互联网发展有关工作的请示”，正式启动“中国下一代互联网示范工程CNGI”。建设目标是在2003年到2005年的时间内，采用IPv6技术，完成CNGI主干网（覆盖20个城市30个核心节点）**

CERNET2

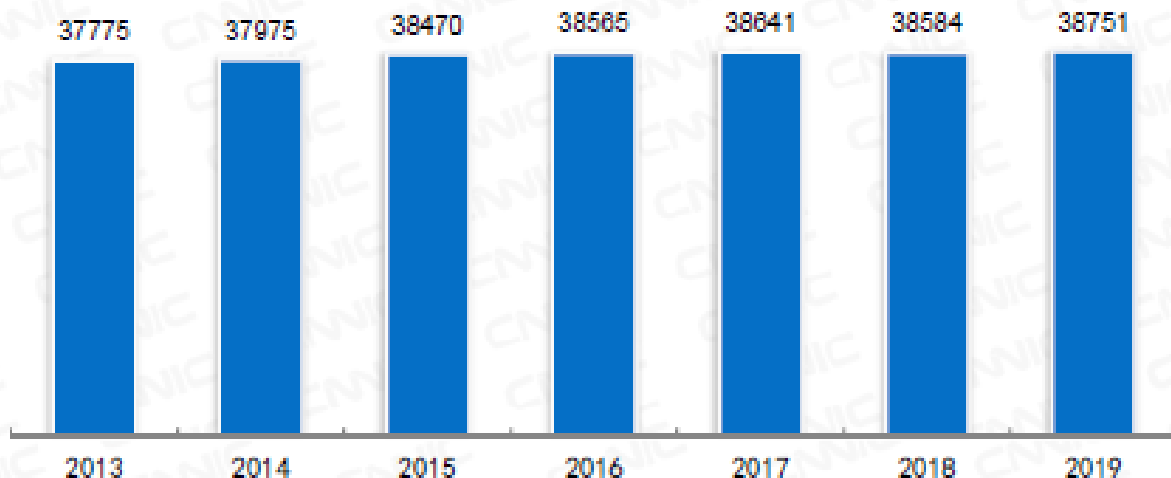
拓扑结构



中国 IPv4 地址 与 IPv6 地址 数量

IPv4地址数量

单位：万个



来源：CNIC 中国互联网络发展状况统计调查

2019.12

IPv6地址数量

单位：块/32



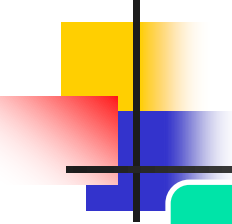
来源：CNIC 中国互联网络发展状况统计调查

2019.12



Chapter 1 概述

- **1.1 Internet 的发展史**
- **1.2 中国Internet的发展史**
- **1.3 IPv6发展史**
- **1.4 Internet的应用**
- **1.5 互联网与物联网**
- **1.6 网络的标准和标准化组织**



1.4Internet的应用（1）

传统应用

- 远程登陆，文件传输，电子邮件，WWW

音视频类

- IP电话，可视电话，电话会议，视频点播，短视频，直播，远程监控，线上教学等

电子商务

- 金融，证券，网购，网银，电子支付

社交、网游类

- 即时通信，网络游戏

智能家居

智慧城市

.....

- 物联网
- 云计算/存储、雾计算/边缘计算
- 智能交通、自动驾驶
- 虚拟现实

- 大数据
- 机器学习/深度学习
- 人工智能

1.4Internet的应用（2）

---移动互联网

■ 特点：

- 永远在线，基本实名（手机号码），手机摄像头和GPS定位。

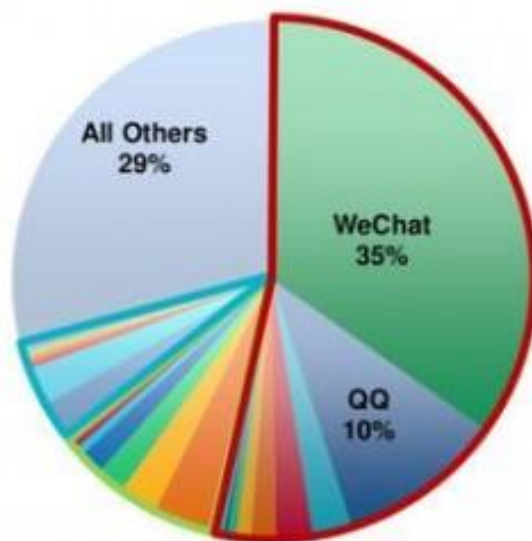
■ 应用类型

- 手机新闻、广告：良好的互动性，用户行为跟踪
- 电子商务：
 - 团购和优惠促销，公交、出租
 - 预订服务，机票、酒店、景点门票的预订、查询、点评等
 - 手机支付，刷脸验证
- 娱乐：手机游戏，音乐、视频
- 移动社区：微信等
- 移动搜索：
 - 比较查询，条形码查询和可视化搜索
 - 基于地理文字的服务，如签到，周边服务搜索、点评，
-

移动互联网中BAT用户的占比

中国移动互联网领导者：腾讯+阿里巴巴+百度，
占据71%移动互联网使用时长

移动互联网使用时间占比，2016年4月每位用户日均移动互联网使用时间约200分钟



Tencent

- WeChat
- QQ
- QQ Browser
- Tencent Video
- Tencent News
- Tencent Games
- QQ Music
- JD.com
- QQ Reading

Alibaba

- UCWeb Browser
- Taobao
- Weibo
- YouKu Video
- Momo
- Shuqi Novel
- AliPay
- AutoNavi

Baidu

- Mobile Baidu
- iQiyi / PPS Video
- Baidu Browser
- Baidu Tieba
- 91 Desktop
- Baidu Maps
- All Other

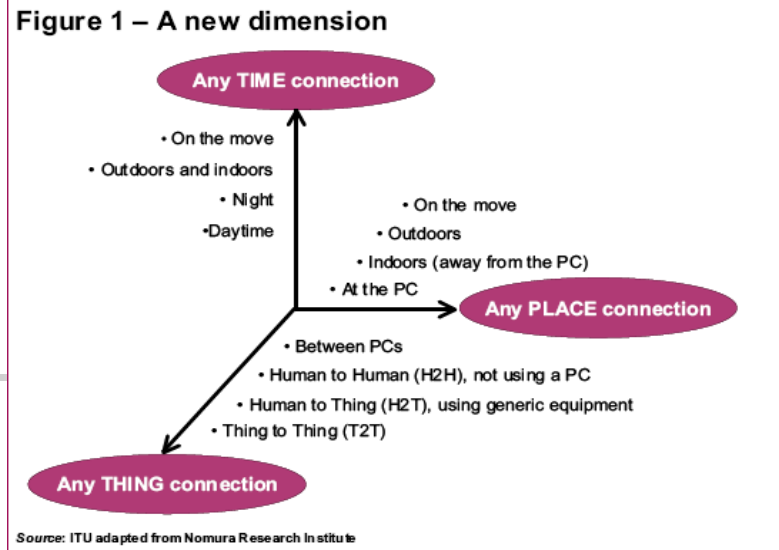
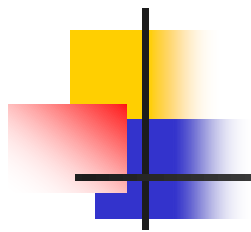


Chapter 1 概述

- **1.1 Internet 的发展史**
- **1.2 中国Internet的发展史**
- **1.3 IPv6发展史**
- **1.4 Internet的应用**
- **1.5 互联网与物联网**
- **1.6 网络的标准和标准化组织**

1.5 互联网与物联网

- 什么是物联网（目前业界对物联网还没有一个完全统一的概念，有以下几种说法）
 - 物联网以互联网为载体，允许人们访问和控制任意联网物品、动物，实现对其智能化的识别、定位、跟踪、监控和管理。
 - 通过射频识别（RFID）、红外感应器、全球定位系统（GPS）、激光扫描器、环境传感器、图像感知器等信息传感设备，按约定的协议，把任何物品与互联网连接起来，进行信息交换和通讯，以实现智能化识别、定位、跟踪、监控和管理的一种网络。
 - 全面信息感知和获取、无缝互联与协同、高度智能化的新型网络形态。物联网是具有标识、感知和智能处理能力的物体基于通信技术相互连接形成的网络，这些物体可以在无需人工干预的条件下实现协同和互动，目的在于为人们提供智慧和集约的服务。
 - 一般的互联网是不带传感跟感知的，当互联网扩展到物联网的时候，可以增加很多传授、感知的功能。互联网平台是通用的，而物联网是单独的。还有互联网的本身并不负责决策，而物联网则要对信息进行分析 and 决策，因此需要很多提炼。



■ 什么是物联网？（2005年国际电联的报告上指出）

- A new dimension has been added to the world of information and communication technologies (ICTs): **from *anytime*, *any place* connectivity for *anyone*, we will now have connectivity for *anything***. Connections will multiply and create an entirely new dynamic network of networks –**an Internet of Things**.



物联网特点

- 物品、动物的相关属性信息来源于射频识别（RFID）、红外感应器、全球定位系统、激光扫描器等信息传感设备。
- 联网实体载有智能芯片，具有一定计算能力。
- 用于信息交换的互联网接入技术和传输技术：WLAN、3G、BlueTooth、ZigBee、UWB等，以及互联网。
- 高度分布和集中处理：
 - 物联网的信息来源及种类多种多样。
 - 物联网的信息载体遍布生活的每一个角落，高度分散。
 - 在这个整合的网络当中，存在能力超级强大的中心计算机群，能够对网络内的人员、机器、设备和基础设施实施实时的管理和控制。

物联网应用

- 将广泛应用于智能交通、环境保护、政府工作、公共安全、平安家居、个人健康等诸多领域。



物联网体系结构分析

目前比较公认的结构是将物联网分为三层

■ 感知层

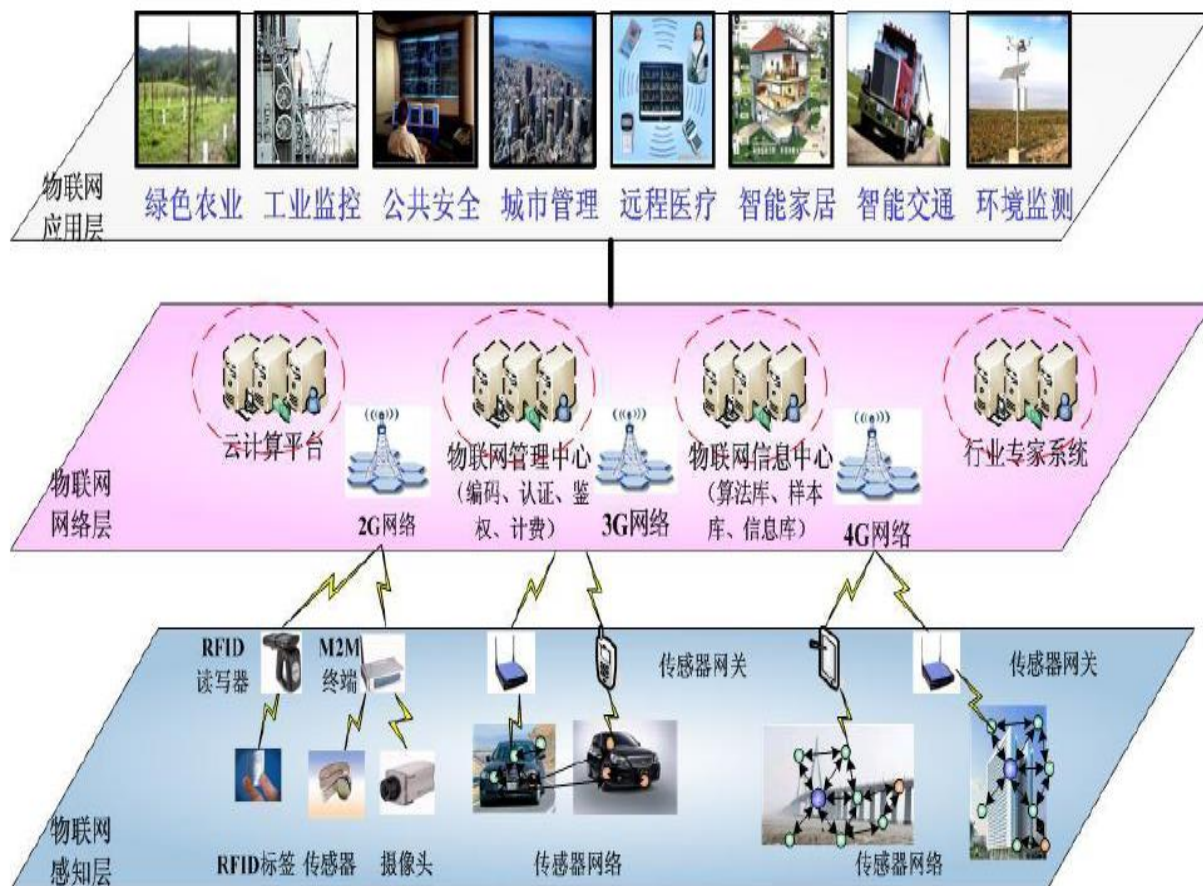
主要是识别物体，采集信息

■ 网络层

网络层将感知层获取的信息进行传递和处理

■ 应用层

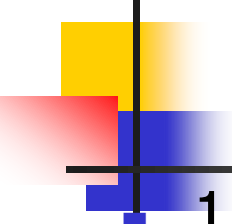
与行业需求结合，实现行业智能化





Chapter 1 概述

- **1.1 Internet 的发展史**
- **1.2 中国Internet的发展史**
- **1.3 IPv6发展史**
- **1.4 Internet的应用**
- **1.5 互联网与物联网**
- **1.6 网络的标准和标准化组织**



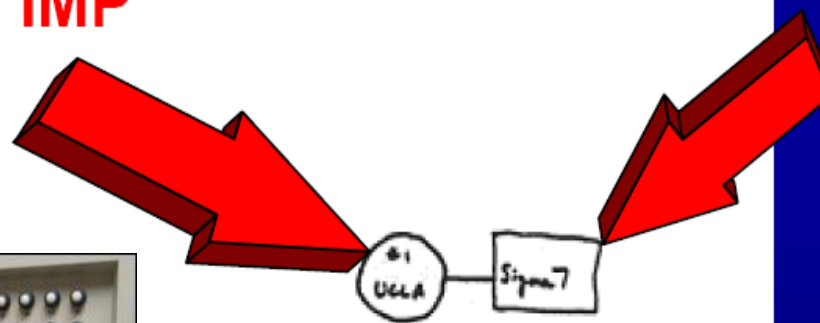
1.6网络的标准和标准化组织

- 1. 标准化的必要性及类型
 - 事实标准：没有正式规划，由一些大厂商产品形成的标准。
 - 法规标准：由一些权威标准化组织制定的标准。
- 2. 国际标准界最有影响的组织
 - 国际电信联盟**ITU**（如H.323， G.723.1等）(<http://www.itu.int>)
 - 国际标准化组织**ISO** (<http://www.iso.org>)
 - 电气电子工程师协会**IEEE**（如IEEE 802）(<http://www.ieee.org>)
- 3. Internet的标准化组织
 - **Internet**工程任务组**IETF**（如请求评注RFC）(<http://www.ietf.org>)
 - Internet研究任务组**IRTF**
- 4. 其他组织
 - 各种论坛**Forum**（如ATM Forum， <http://www.atmforum.com>）
 - 美国国家标准协会**ANSI**
 - 地区标准化组织（如欧洲、日本的一些标准）

What It Looked Like in 1969

The
Interface
Message
Processor
IMP

UCLA



THE ARPA NETWORK

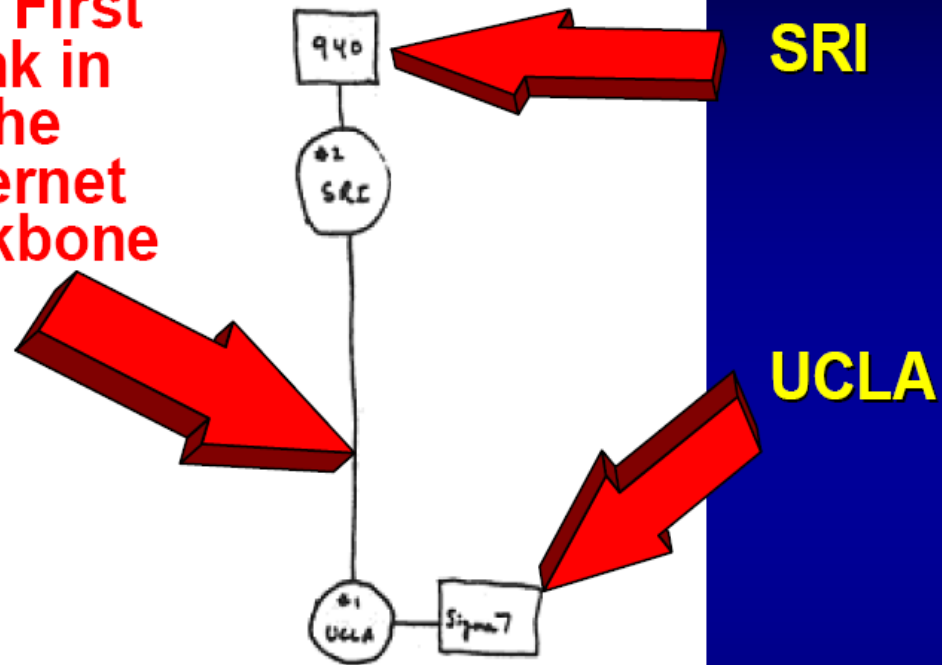
September 1969



The 1969 IMP



The First Link in the Internet Backbone



THE ARPANET

October 1969

What was the first message even sent on the Internet?

- 1844年，电报发明人Morse的第一份电报
“What hath God Wrought”
- 1876年，电话发明人Bell的第一句话
“Watson, come here. I want you.”
- 1969年，Armstrong登上月球的第一句话
“One Giant Leap for Mankind.”

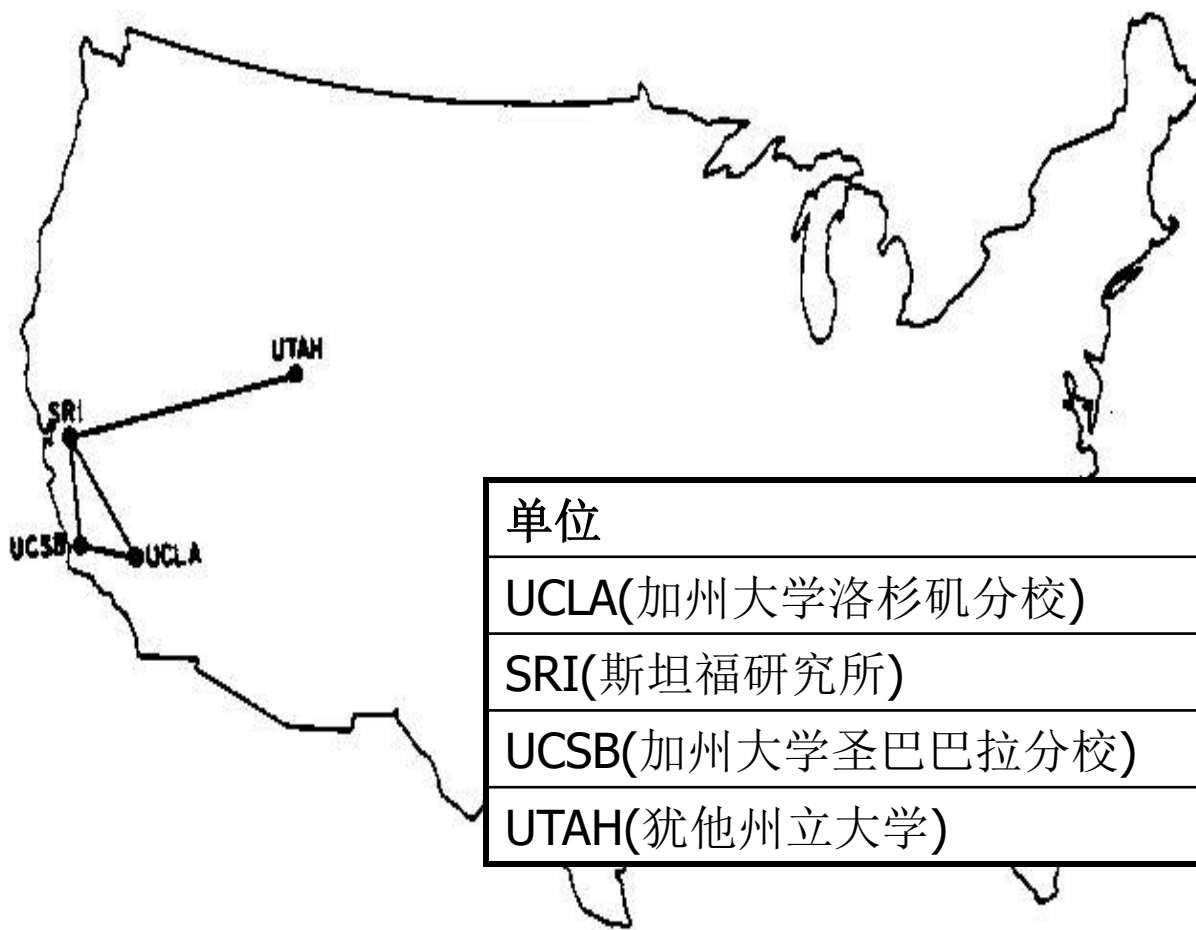
It was simply a **LOCTN**

from

Nobody noticed!!

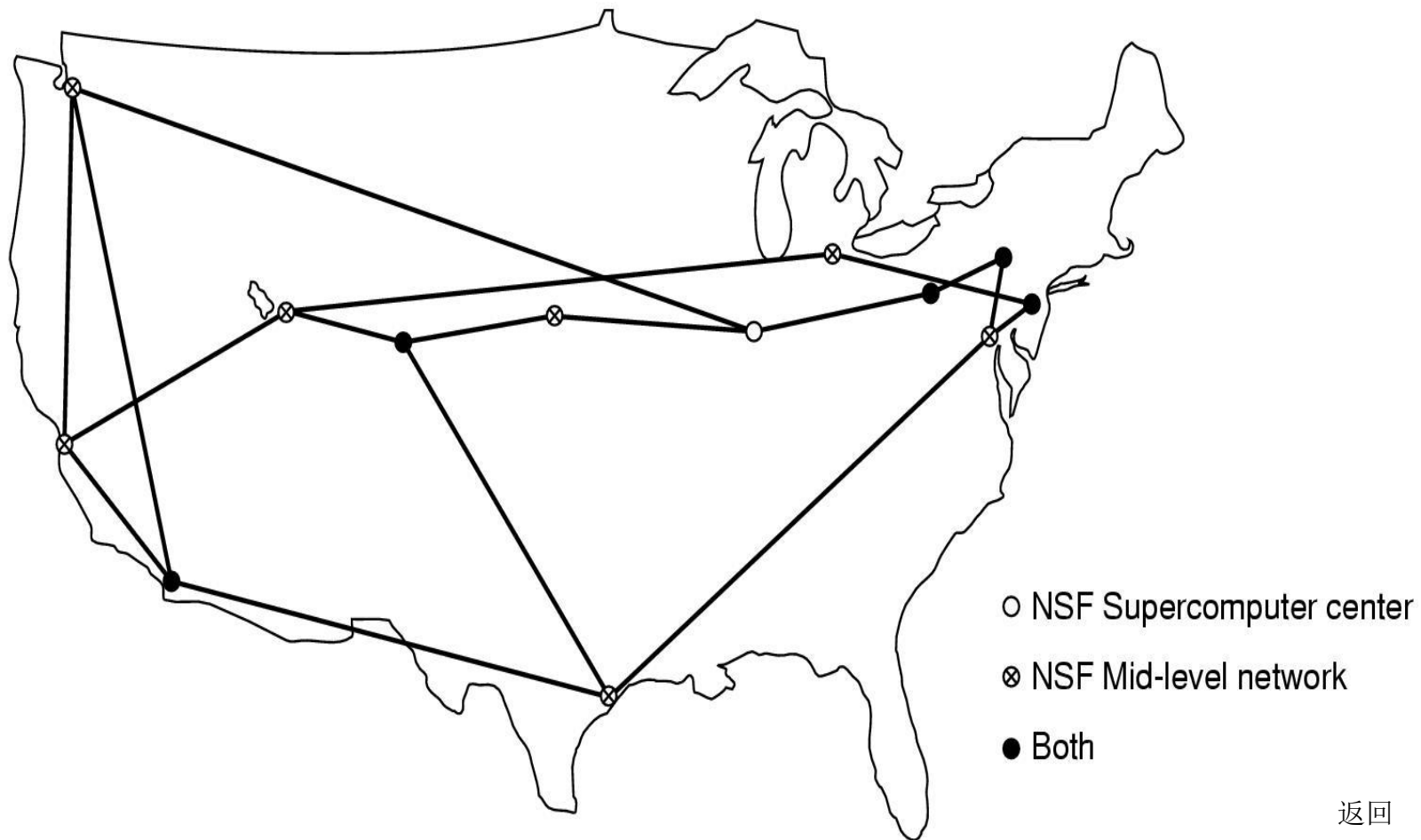
er.

1969年12月的ARPAnet



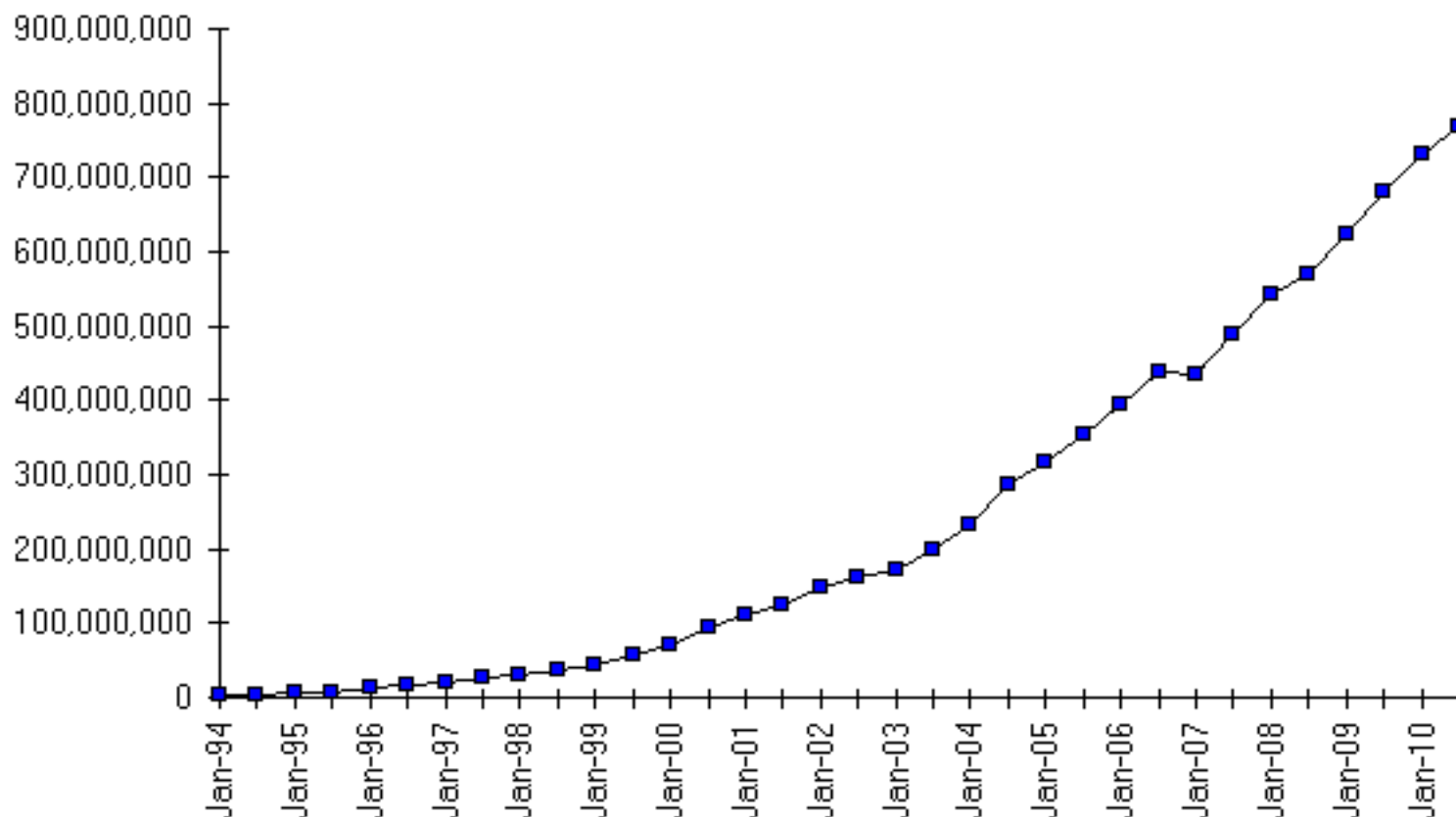
单位	机型	OS
UCLA(加州大学洛杉矶分校)	SDS Sigma 7	SEX
SRI(斯坦福研究所)	XDS-940	Genie
UCSB(加州大学圣巴巴拉分校)	IBM 360/75	OS/MVT
UTAH(犹他州立大学)	Digital PDP-10	TENEX

1988年的NSFNet

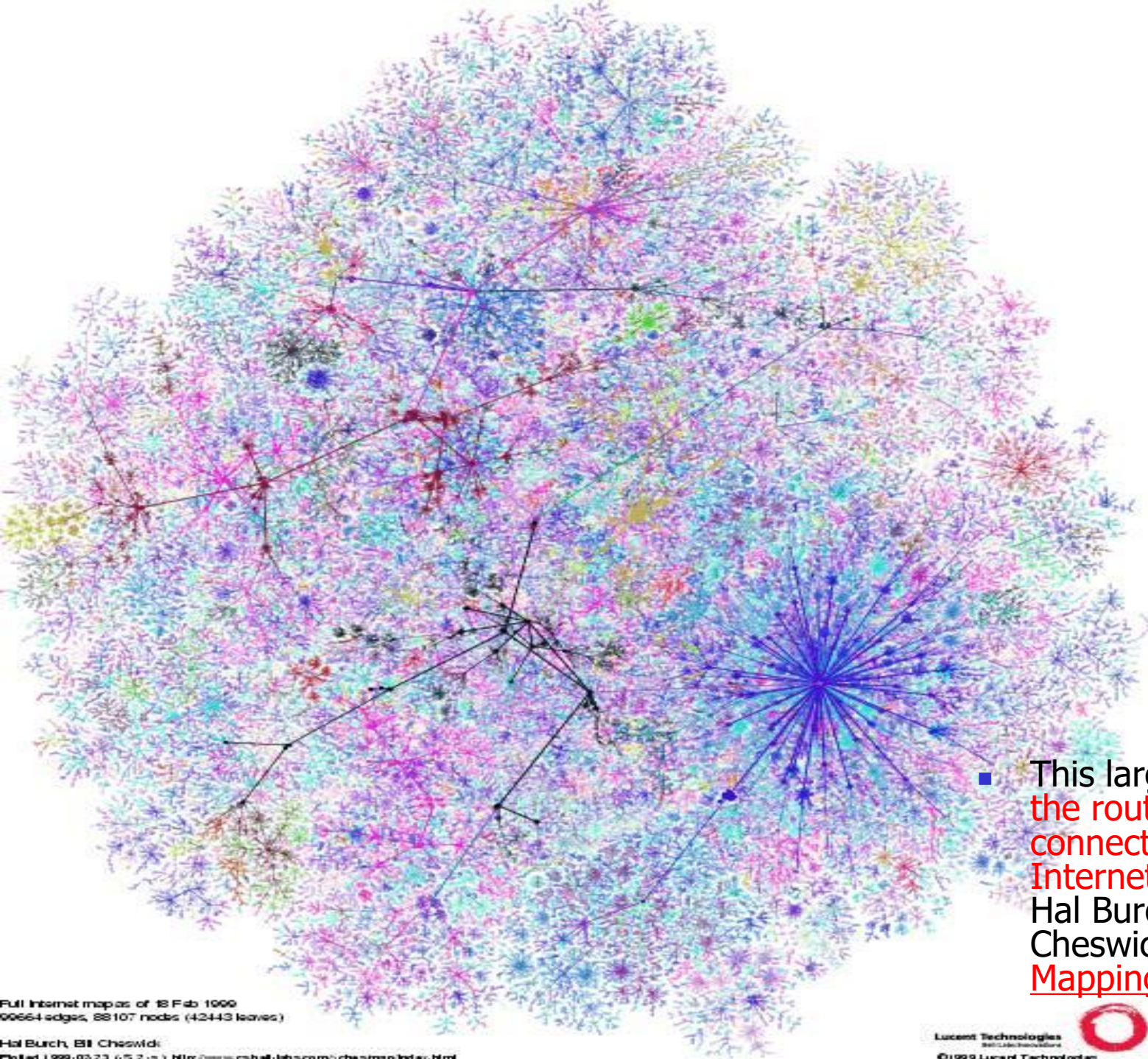


接入因特网的主机数

Internet Domain Survey Host Count



Source: Internet Systems Consortium (www.isc.org)



- This large graph shows the router level connectivity of the Internet as measured by Hal Burch and Bill Cheswick's Internet Mapping Project

Full Internet map as of 18 Feb 1999
90664 edges, 88107 nodes (42443 leaves)

Hal Burch, Bill Cheswick
Mailed 1999-02-23 (15:22:00) <http://www.cs.cmu.edu/~lab/ucm/~chew/imap/index.html>

Lucent Technologies
Bell Laboratories
© 1999 Lucent Technologies

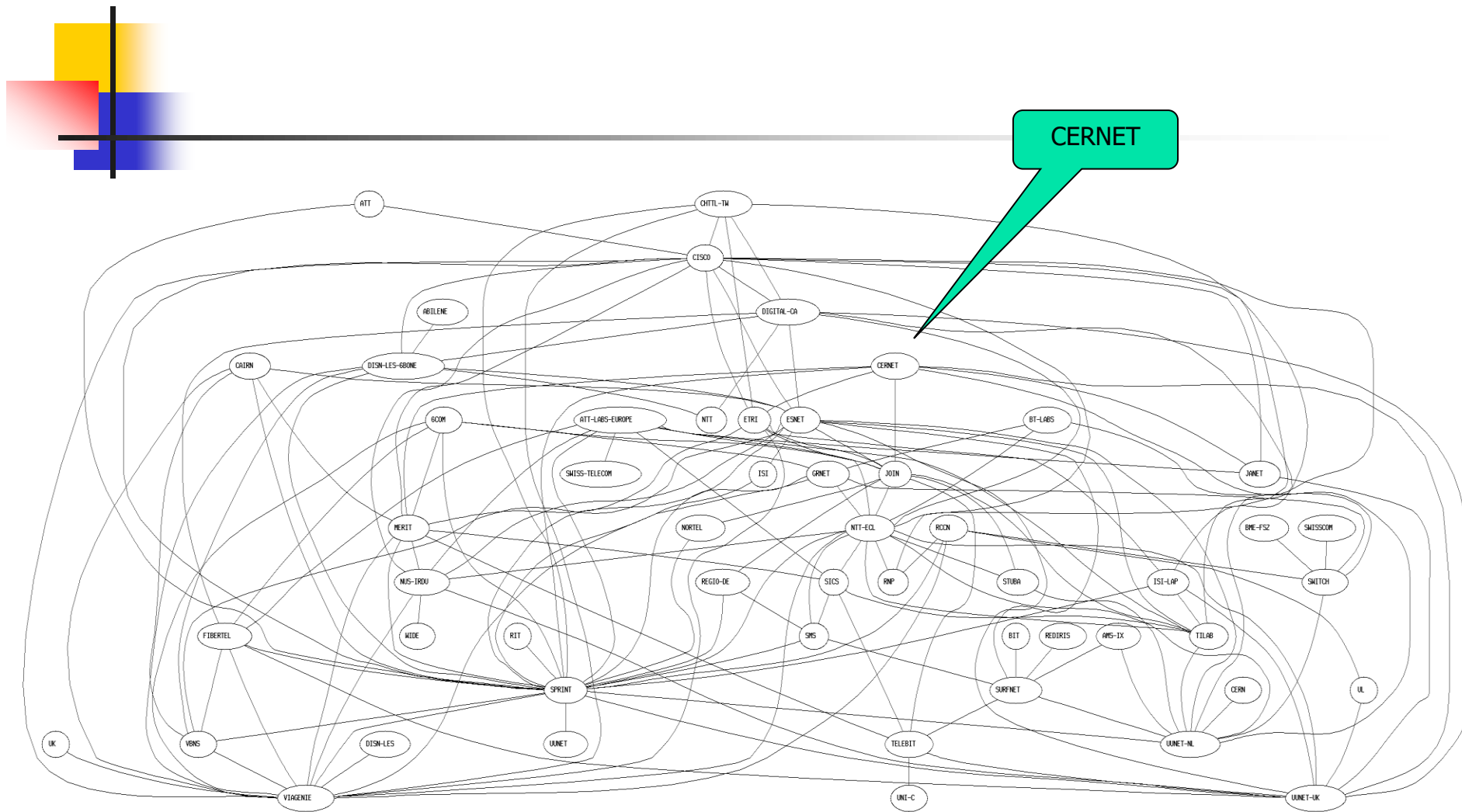




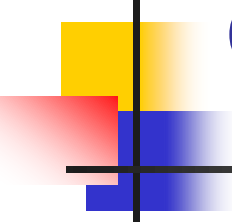
[返回](#)

Asia Pacific - Red
Europe/Middle East/Central
Asia/Africa - Green
North America - Blue
Latin American and Caribbean -
Yellow
RFC1918 IP Addresses - Cyan
Unknown - White

6BONE



The 6Bone p2PAs and their tunnels. adamo@lab.ece.ntu.gr, Sun Feb 29 08:00:19 EET 2005



Chapter 2 网络的体系结构与参考模型

- 2.1 网络的构成
- 2.2 网络的体系结构
- 2.3 OSI模型
- 2.4 TCP/IP模型
- 2.5 OSI与TCP/IP模型比较
- 2.6 网络的分类

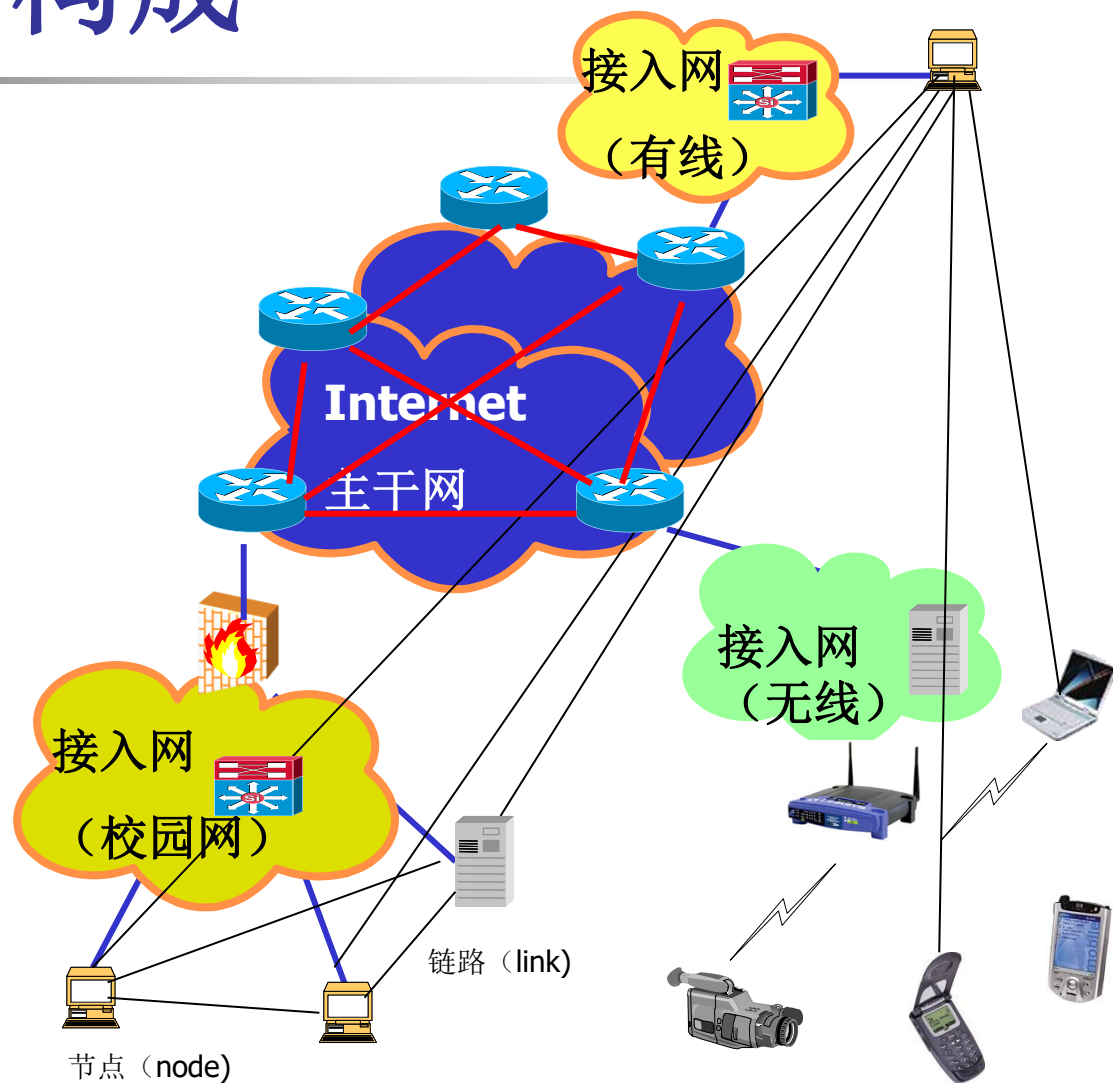
2.1 网络的构成

■ 网络的构成

- 端系统（主机等）：
信息的收发
- 通信链路（电缆等）：
信息的传输
- 中间交换系统（路由器等）：
信息的转发

■ 网络功能上分

- 主干网（核心网络）
- 接入网（边缘网络）



2.1 网络的构成

■ 网络协议上分：

■ 局域网（ Local Area Network ）

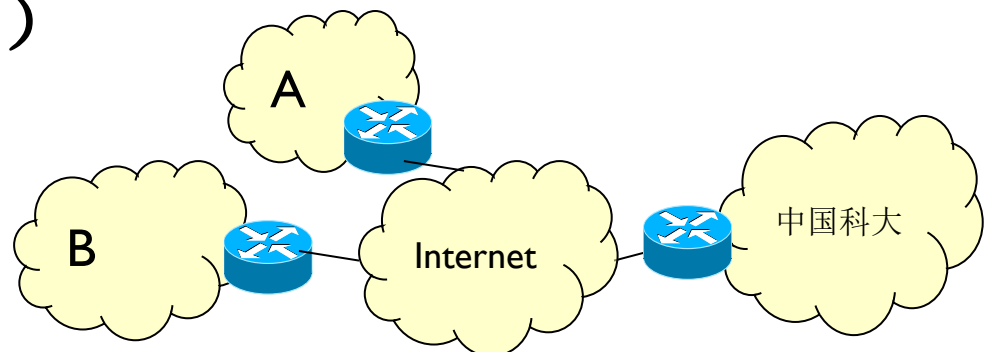
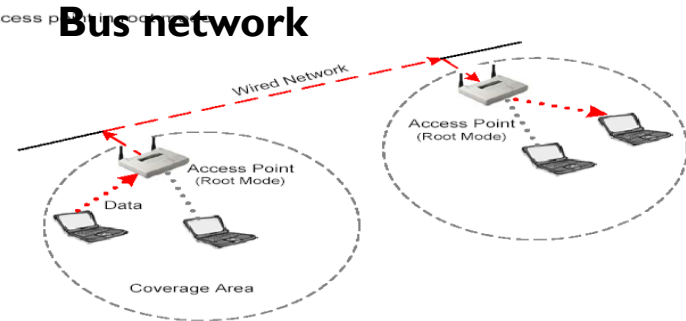
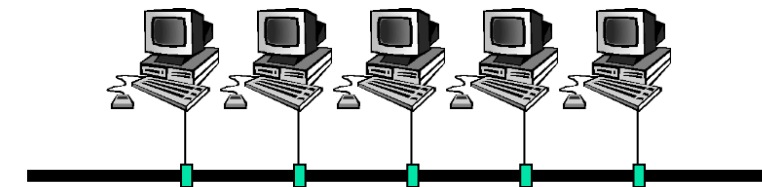
- 有线LAN：以太网（IEEE 802.3）
- 无线LAN：WiFi（IEEE 802.11）

■ 广域网（ Wide Area Network ）

- PPP, X.25, ATM

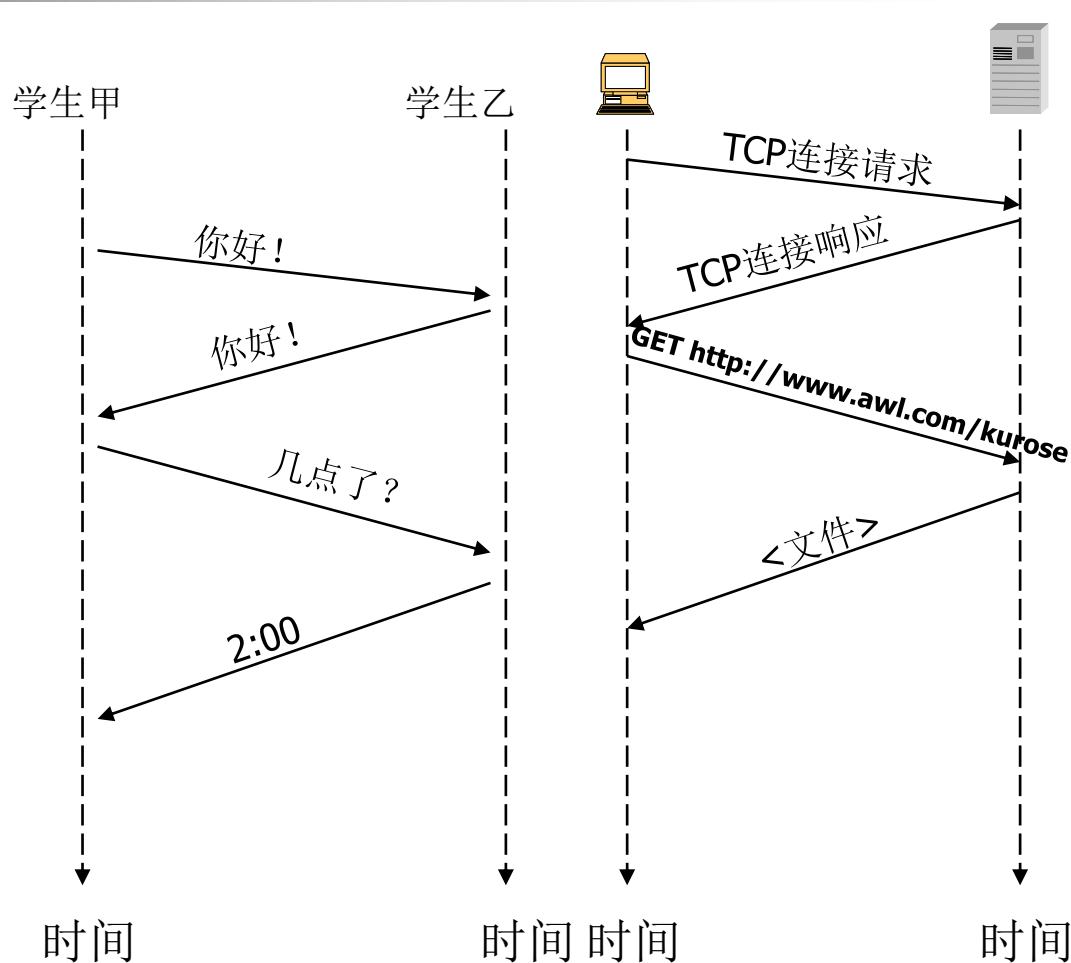
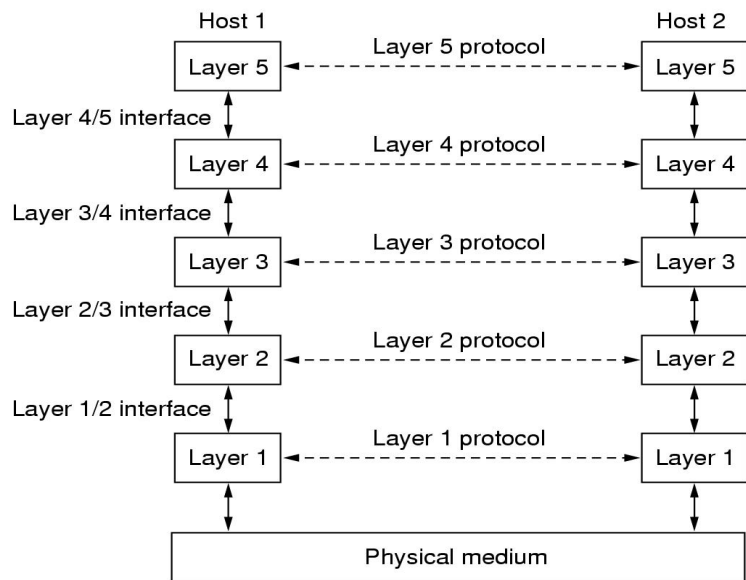
■ 互联网

- IPv4 (IETF RFC 791\919\922)
- IPv6 (RFC 2406)



2.2网络体系结构

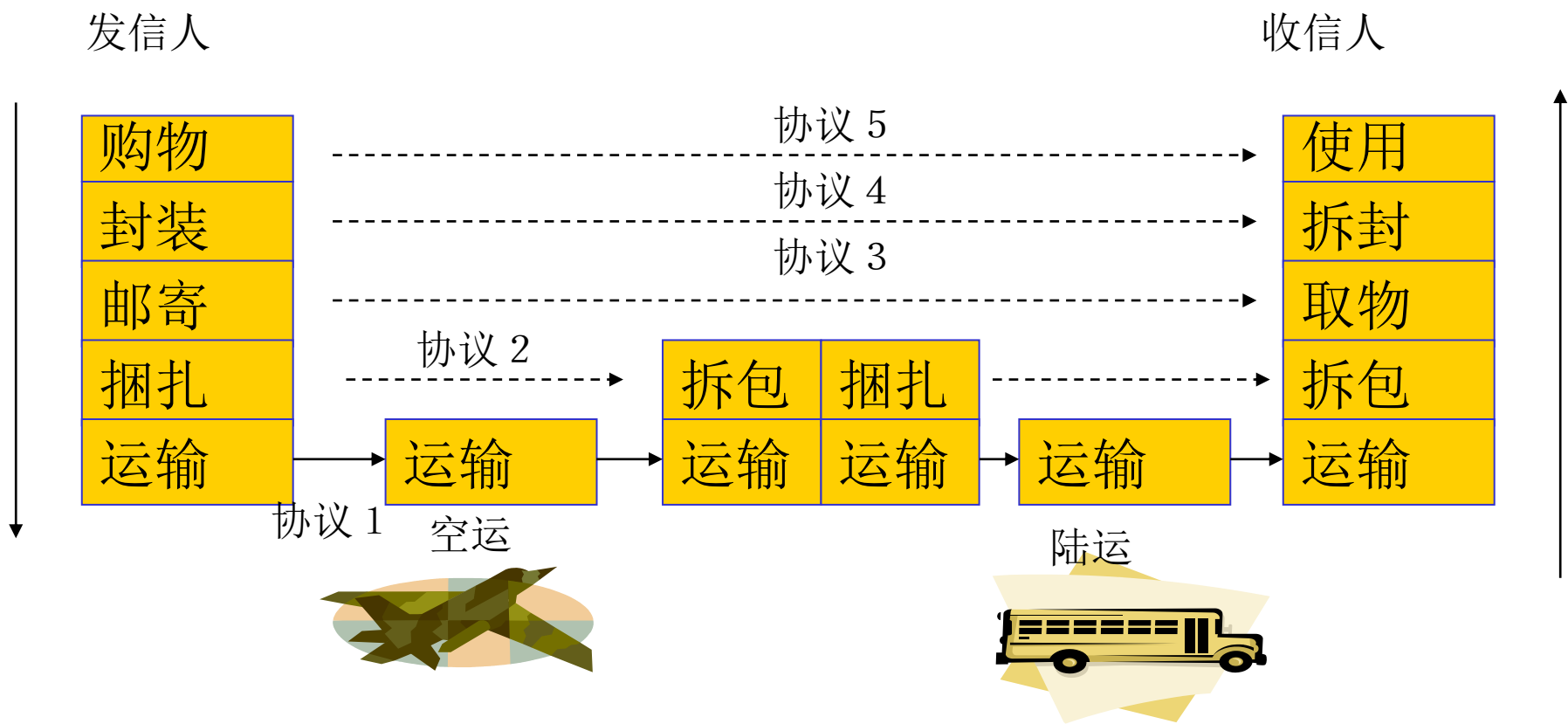
(网络的总体设计) —— 层和协议的集合^[1]



■ 协议 (protocol)

- 什么是协议?
- 协议为什么要分层?

举例（寄包裹）



2.2网络体系结构

——层和协议的集合

■ 协议体系(protocol architecture)

- 网络协议：是指通信双方（或多方）关于如何进行通信的一种约定^[1]。
- 协议分层：为了降低设计的复杂度，增加网络的可扩展性，具有概念化、结构化的优点，有利于新业务的导入。

■ 分层的原则：

- 将相似的功能集中在同一层内，必要时可将层的功能再分成子块，层数不宜过多，以避免层间接口的开销变大。
- 当功能差别较大时应分层处理。
- 各层只对相邻的上下层定义接口。

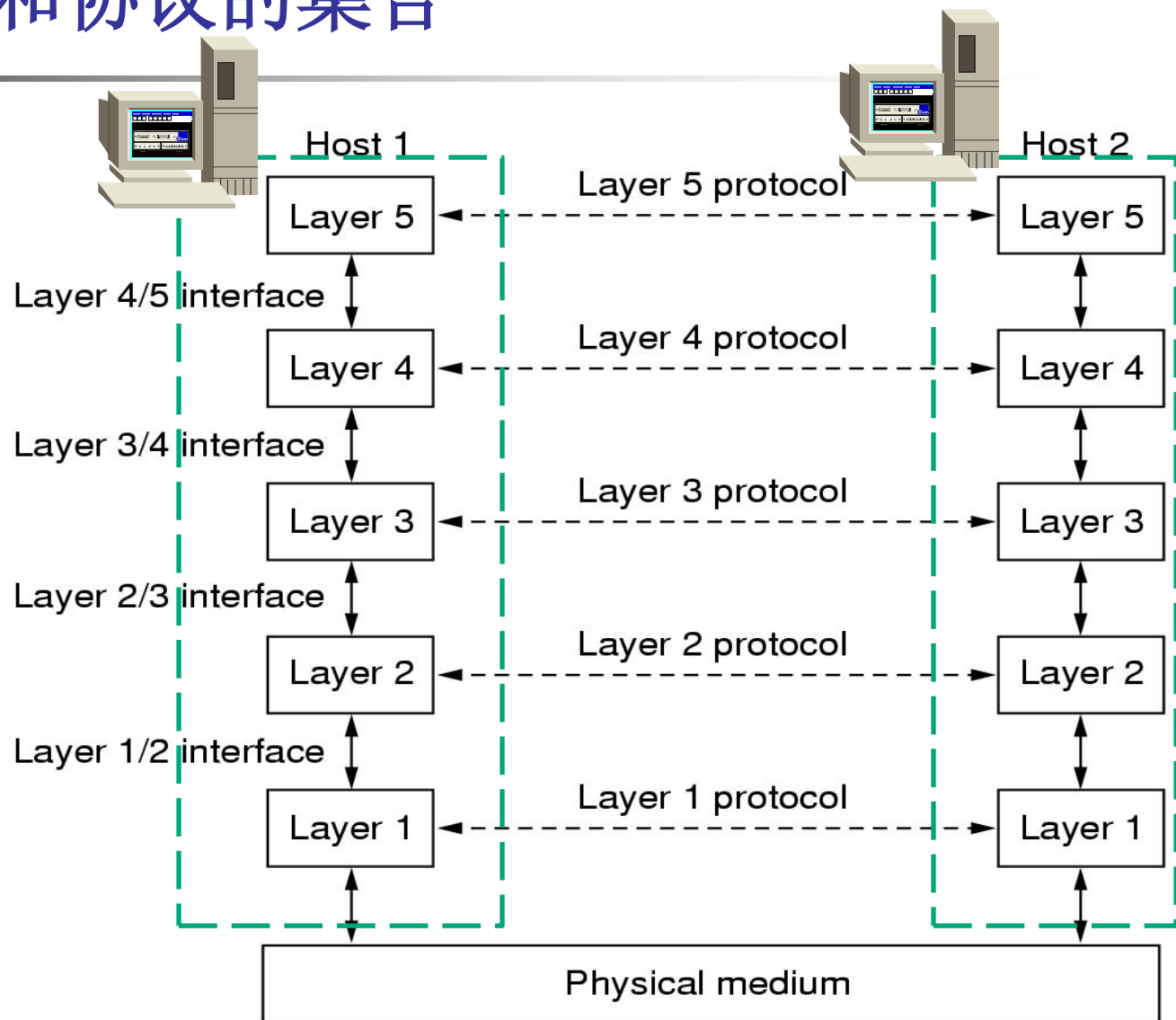
■ 协议栈(protocol stack)：一个特定的系统所使用的一组协议（每一层一个或几个协议），网络协议图（[参见其它文件](#)）

2.2网络体系结构

——层和协议的集合

参考模型的核心:

- 服务(service)
- 接口(interface)
- 协议(protocol)

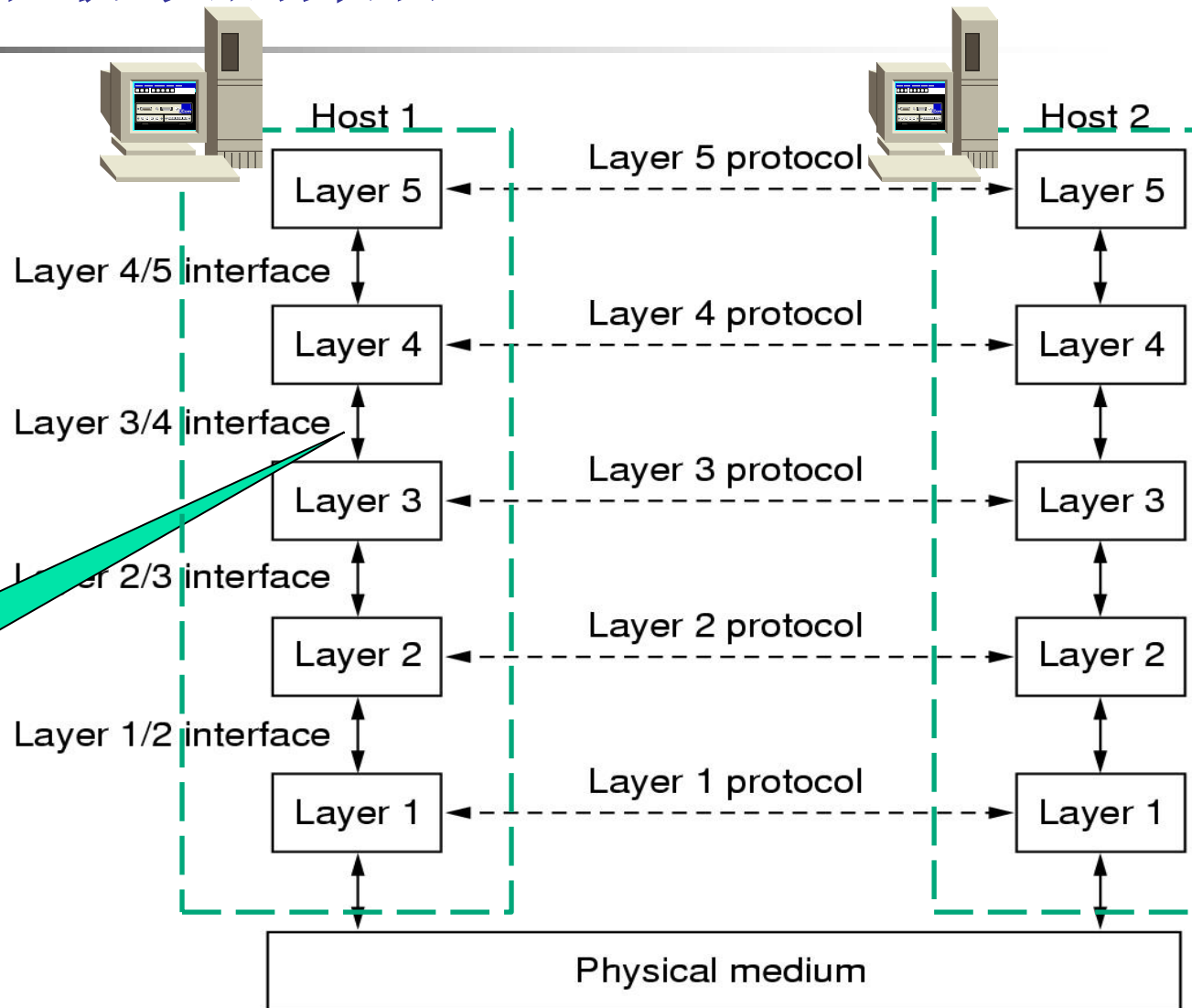


2.2网络体系结构

——层和协议的集合

参考模型的核心:

- 服务(service)
- 接口(interface)
- 协议(protocol)



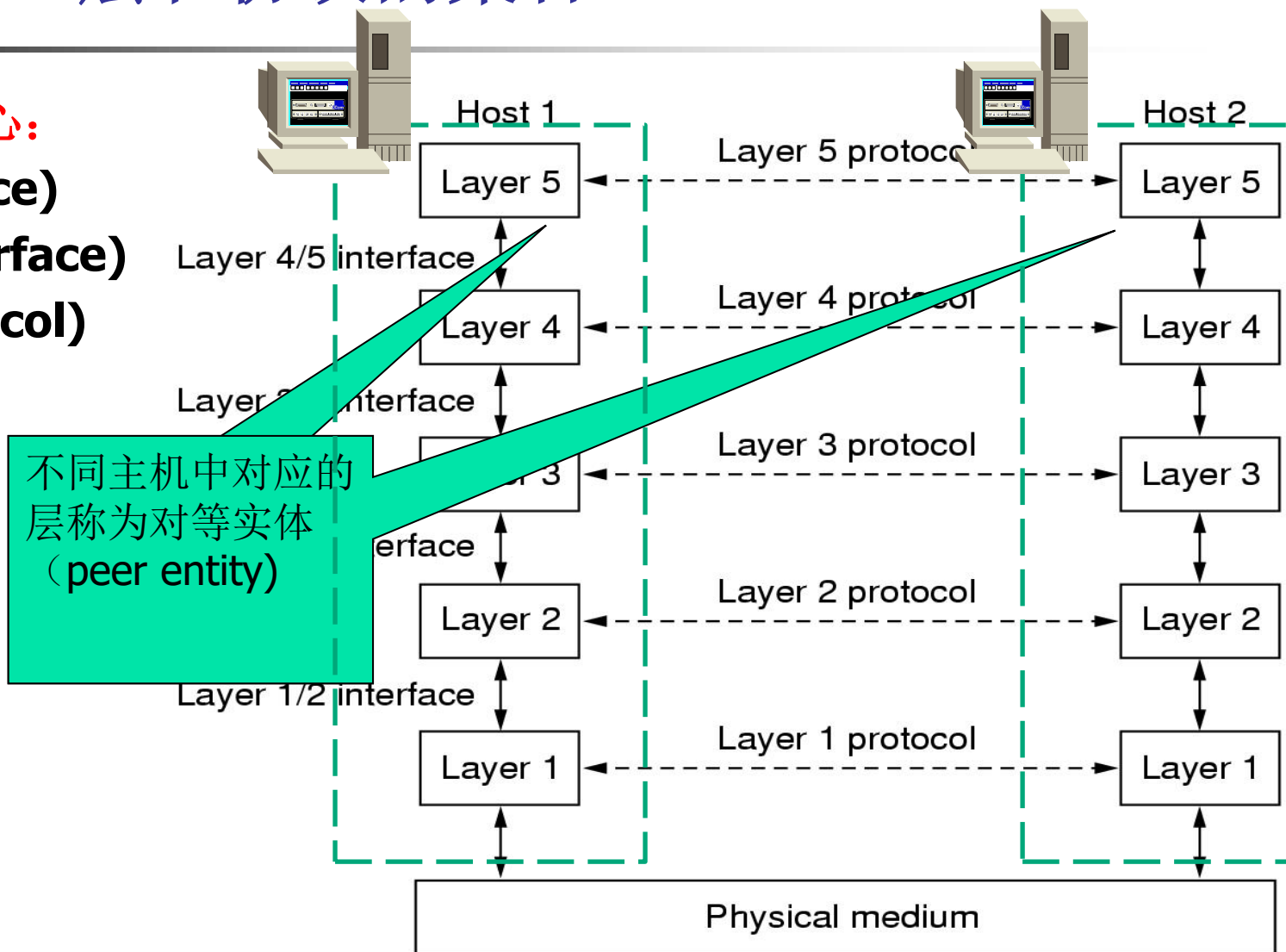
相邻上下层之间都有接口，接口定义下层向上层提供的服务。

2.2网络体系结构

——层和协议的集合

参考模型的核心:

- 服务(service)
- 接口(interface)
- 协议(protocol)



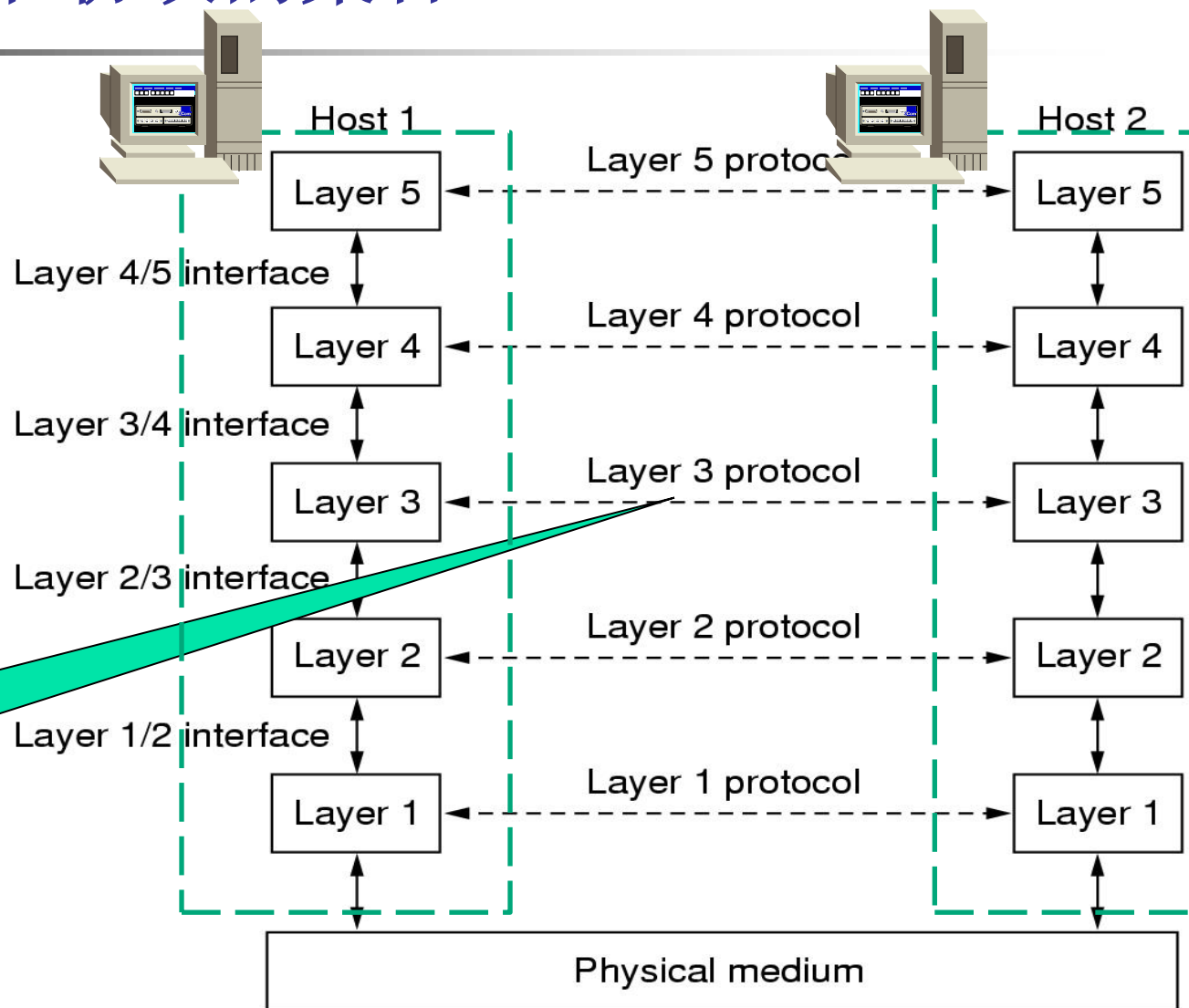
2.2网络体系结构

——层和协议的集合

参考模型的核心:

- 服务(service)
- 接口(interface)
- 协议(protocol)

第n层的通信规则
和功能由该层的协
议描述



数据单元的名称与关系

- 业务接入点 service access point

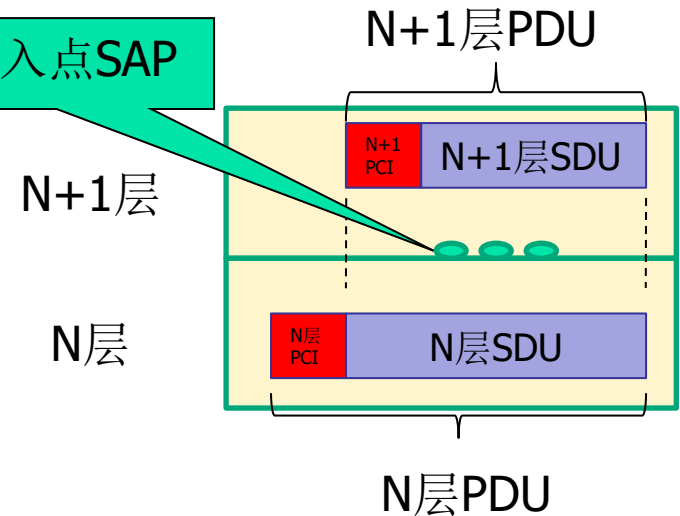
业务接入点SAP

- 数据单元的名称

- SDU: service data unit
- PDU: protocol data unit
- PCI: protocol control info.

- 数据单元的关系

- $N\text{层的PDU} = N\text{层PCI} + N\text{层SDU}$
- $N\text{层的SDU} = N+1\text{层的PDU}$





服务 (service)

■ 面向连接的服务

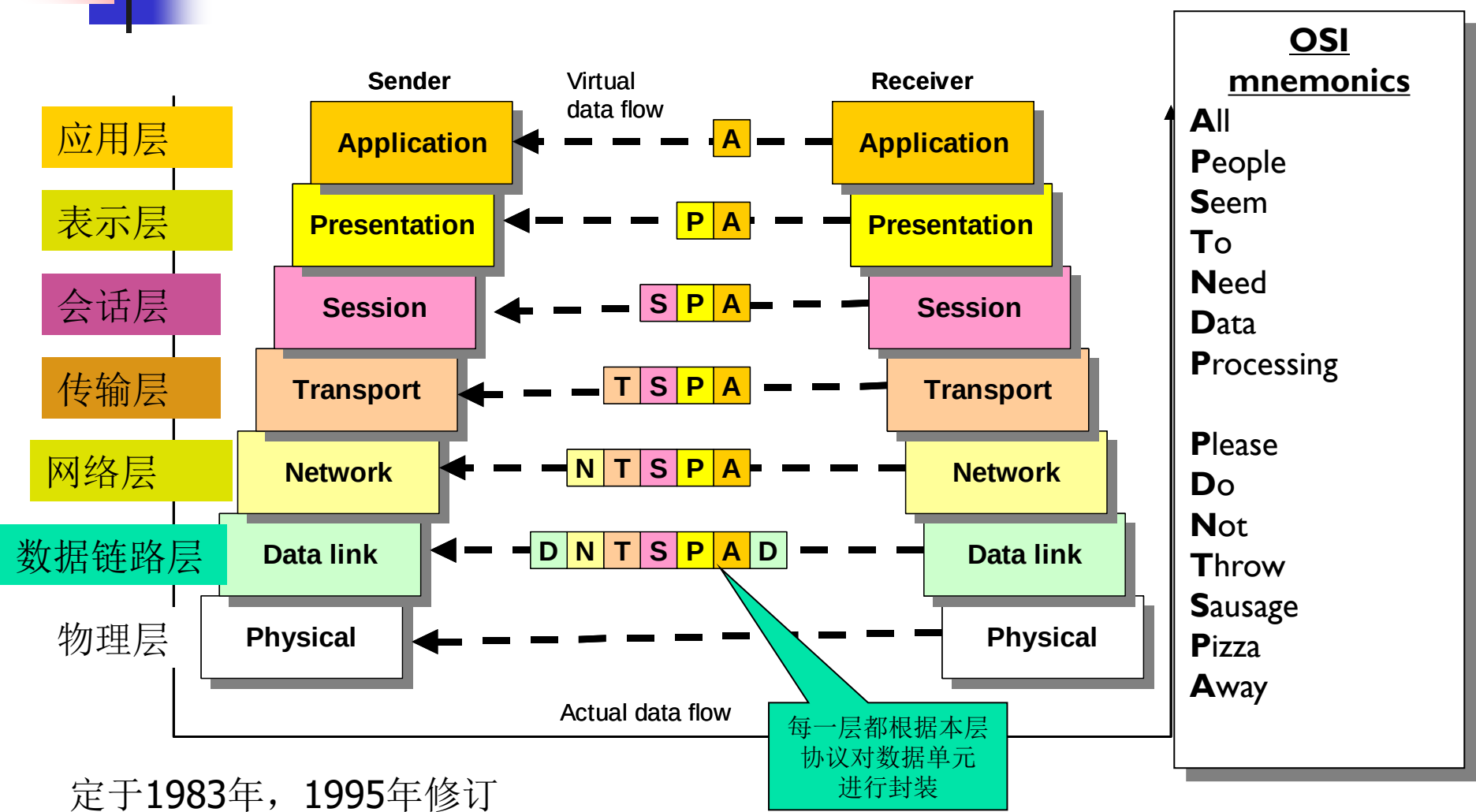
- 首先要在信源与信宿之间建立连接，然后在此连接上通信，最后拆除连接。
- 特点：占用一定的资源，可靠，按序传送。（不同的层占用的资源不同，传输层服务会占用主机的资源，网络层或链路层会占用转发节点的资源）

■ 非连接服务

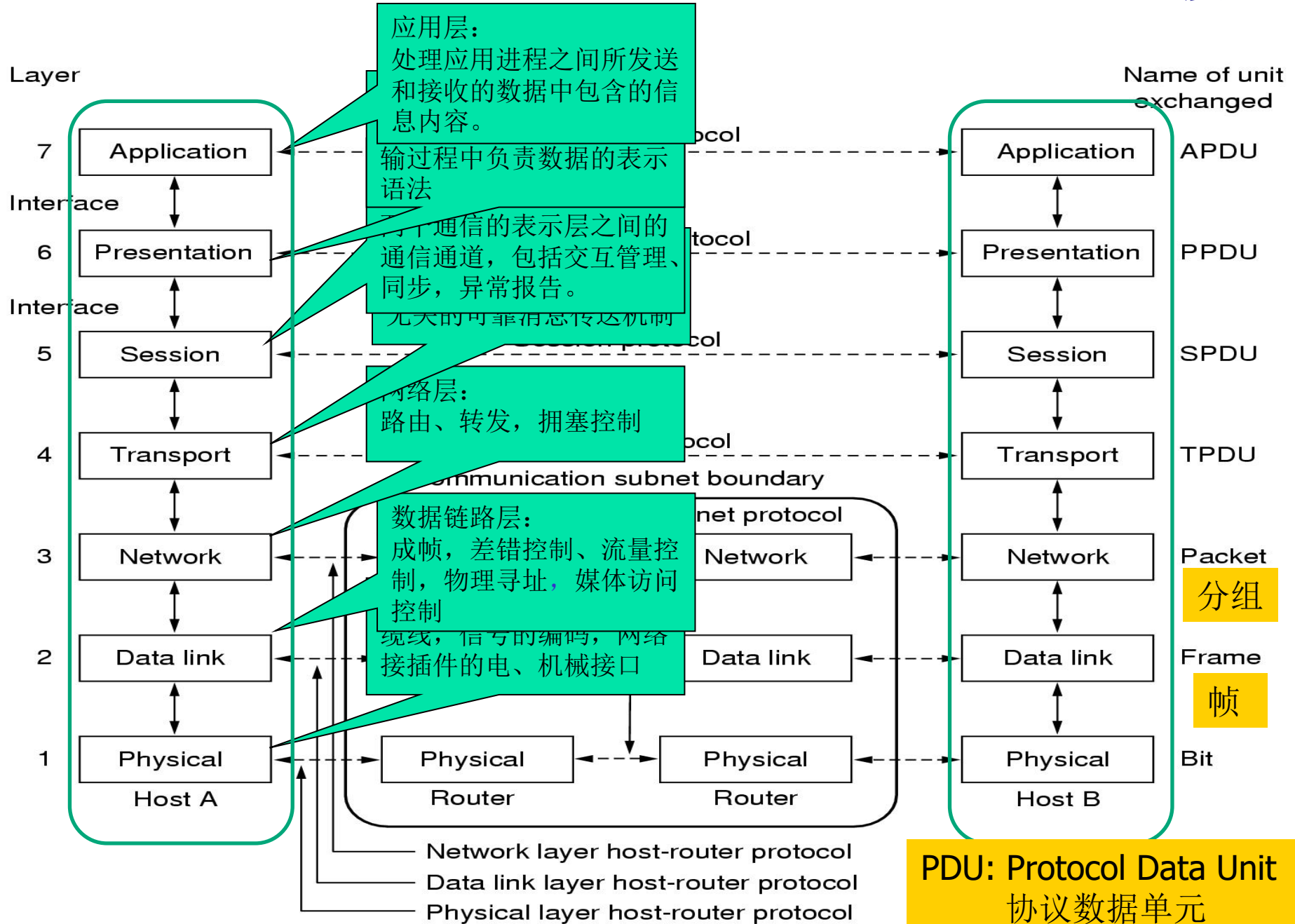
- 传送数据前不需要建立连接，即有即送。
- 将每个数据单元打包，在包头添加地址信息。
- 特点：每个数据包独自寻路（重复劳动），同一数据流的包可能经由不同的路径到达目的地，到达的顺序也可能颠倒。（这是网络层或链路层服务的特点）

国际标准化组织 (International Standard Organization)
开放系统互连 (Open System Interconnection)

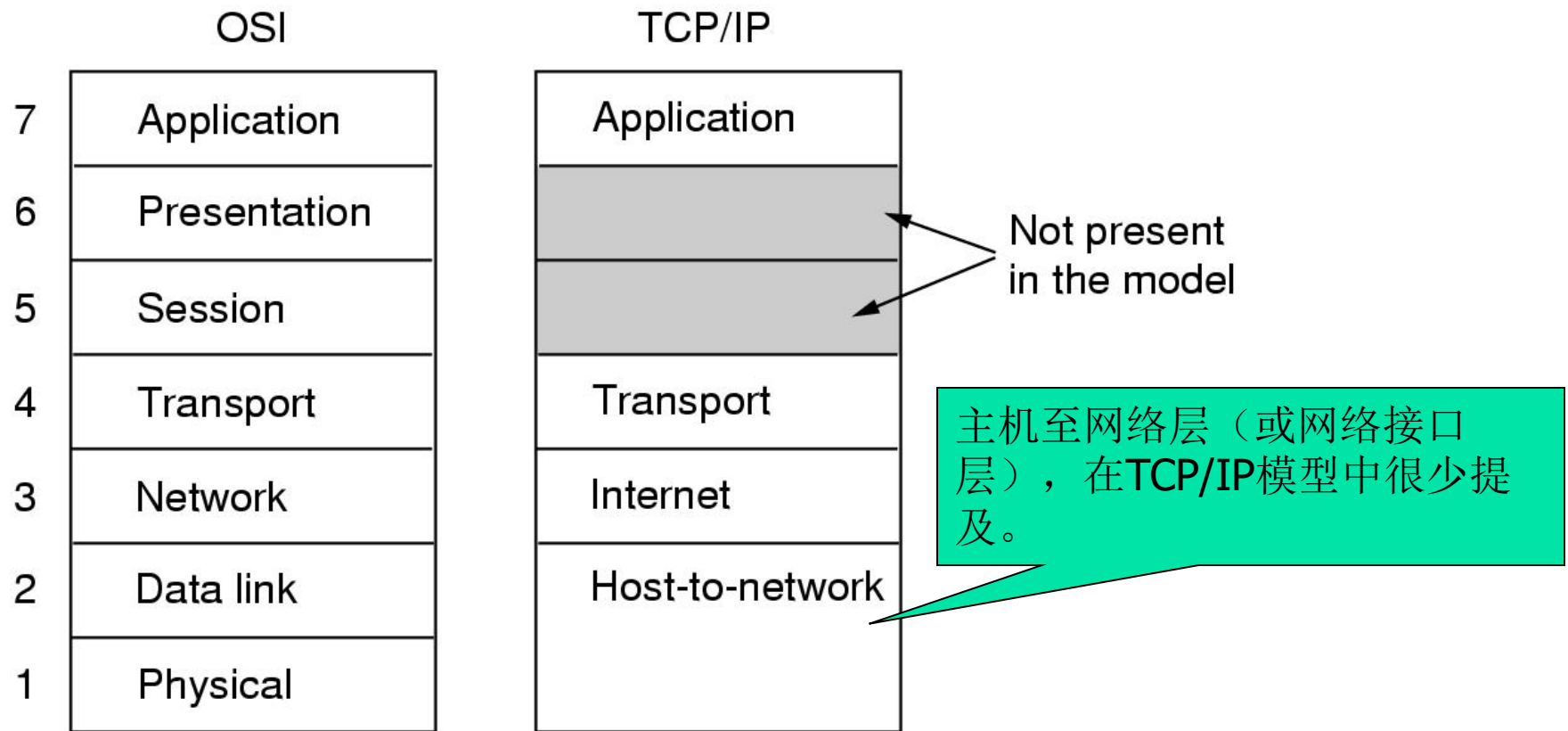
2.3 ISO-OSI模型



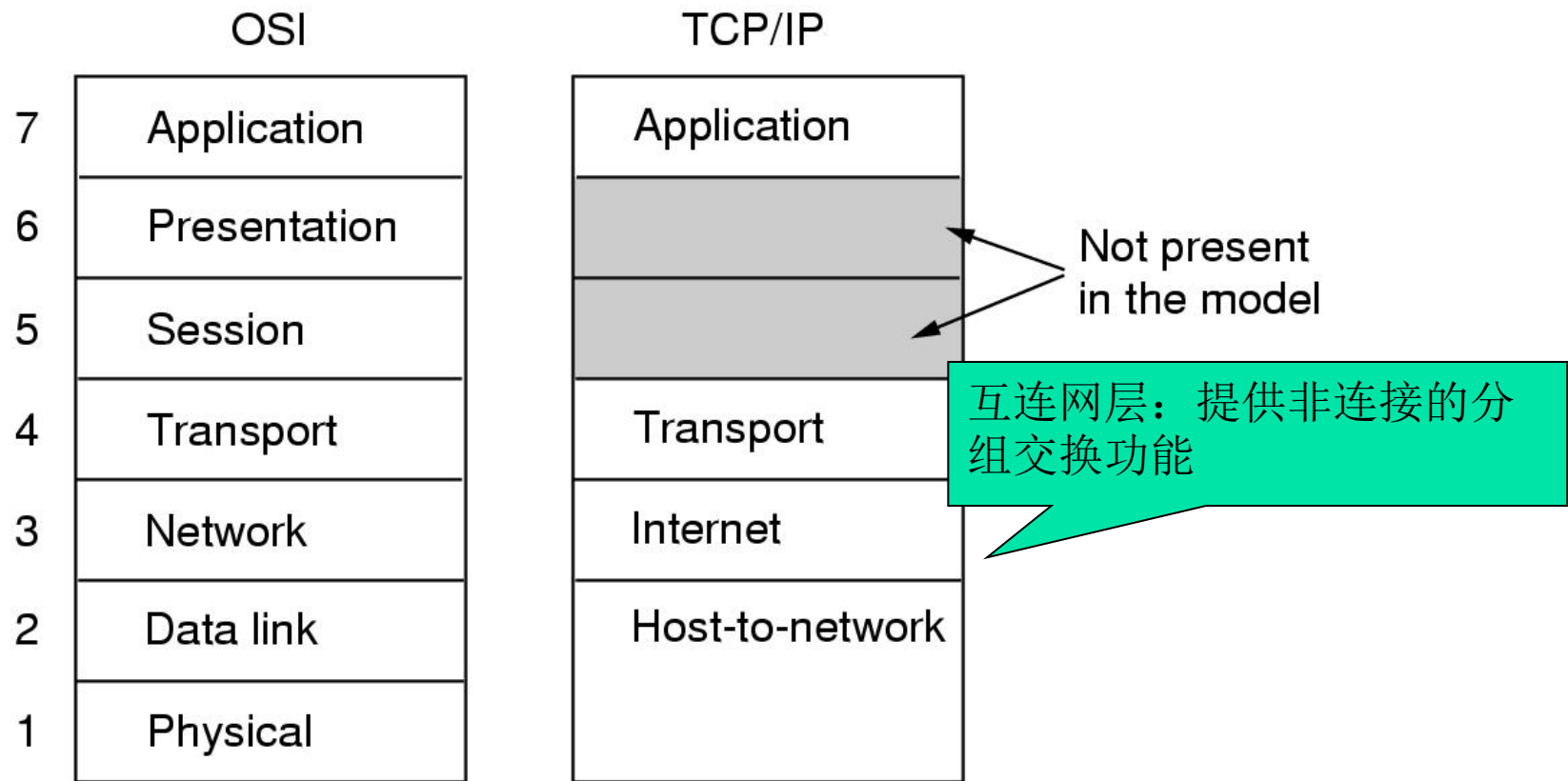
ISO-OSI模型



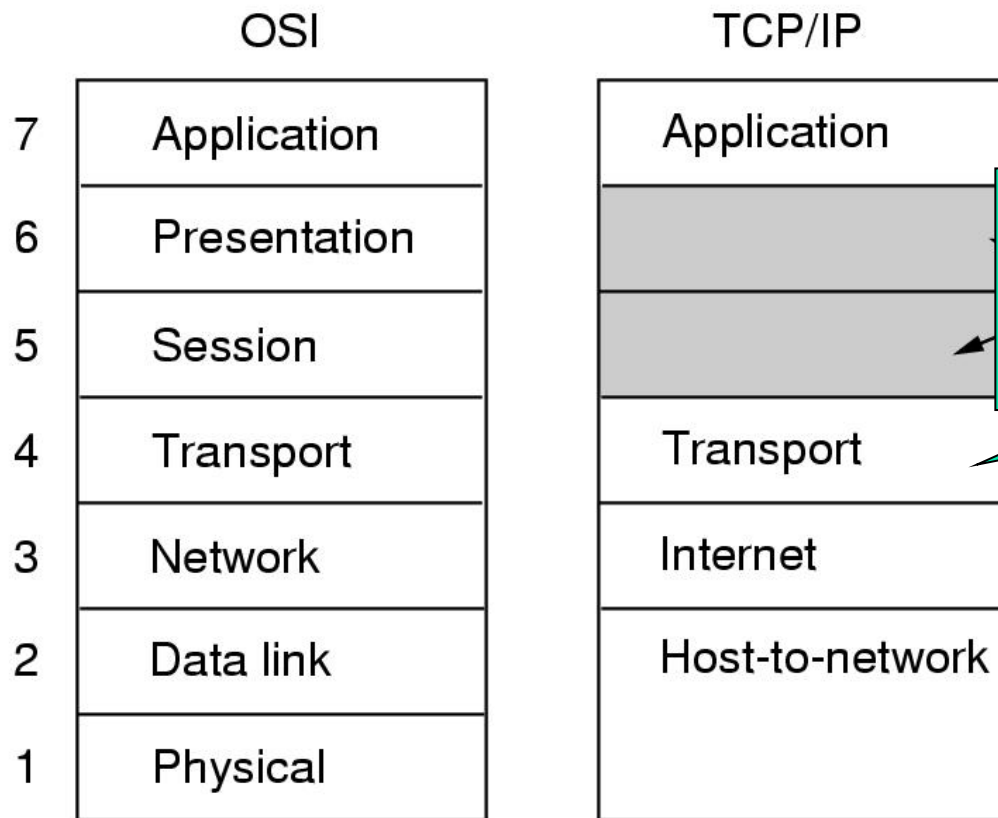
2.4 TCP/IP模型



2.4 TCP/IP模型

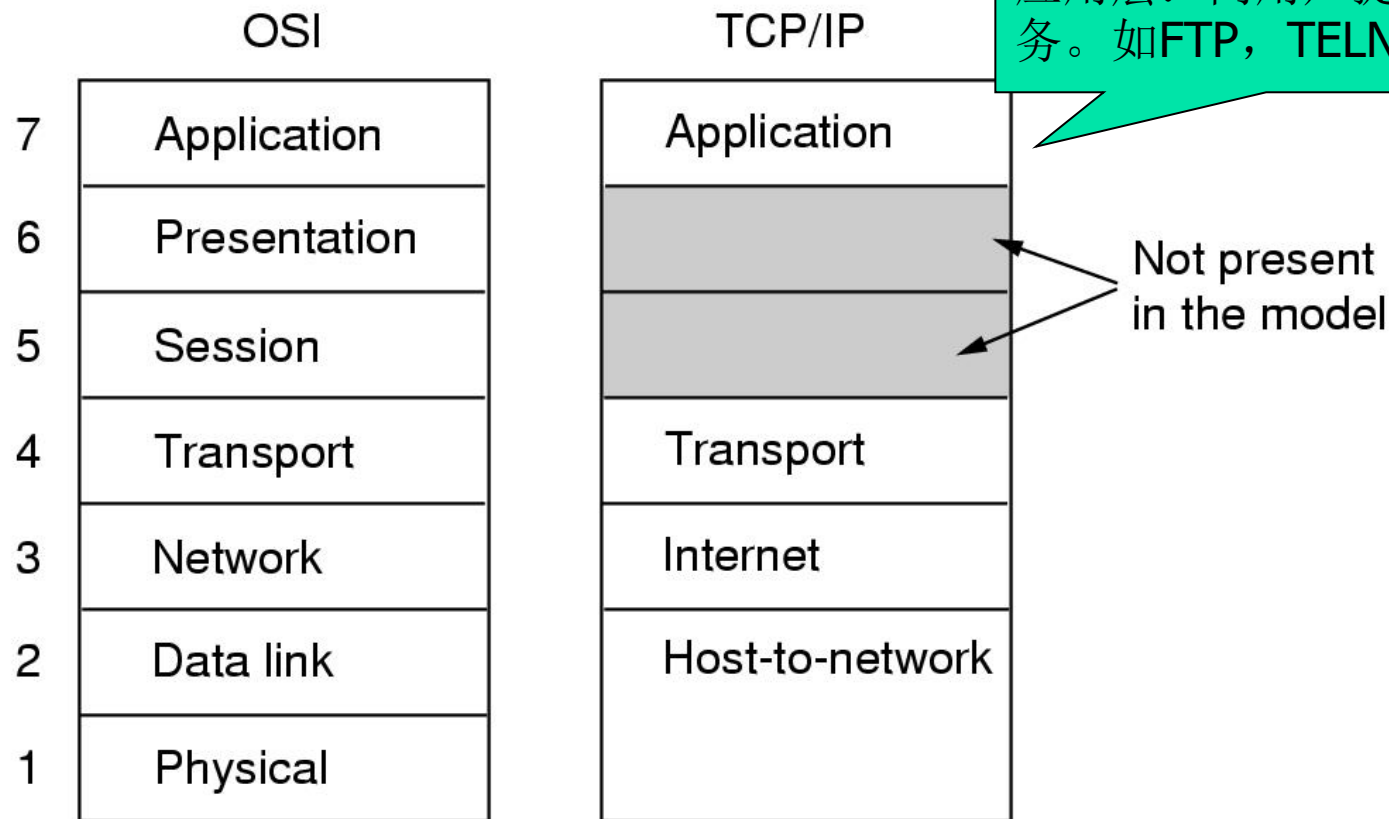


2.4 TCP/IP模型

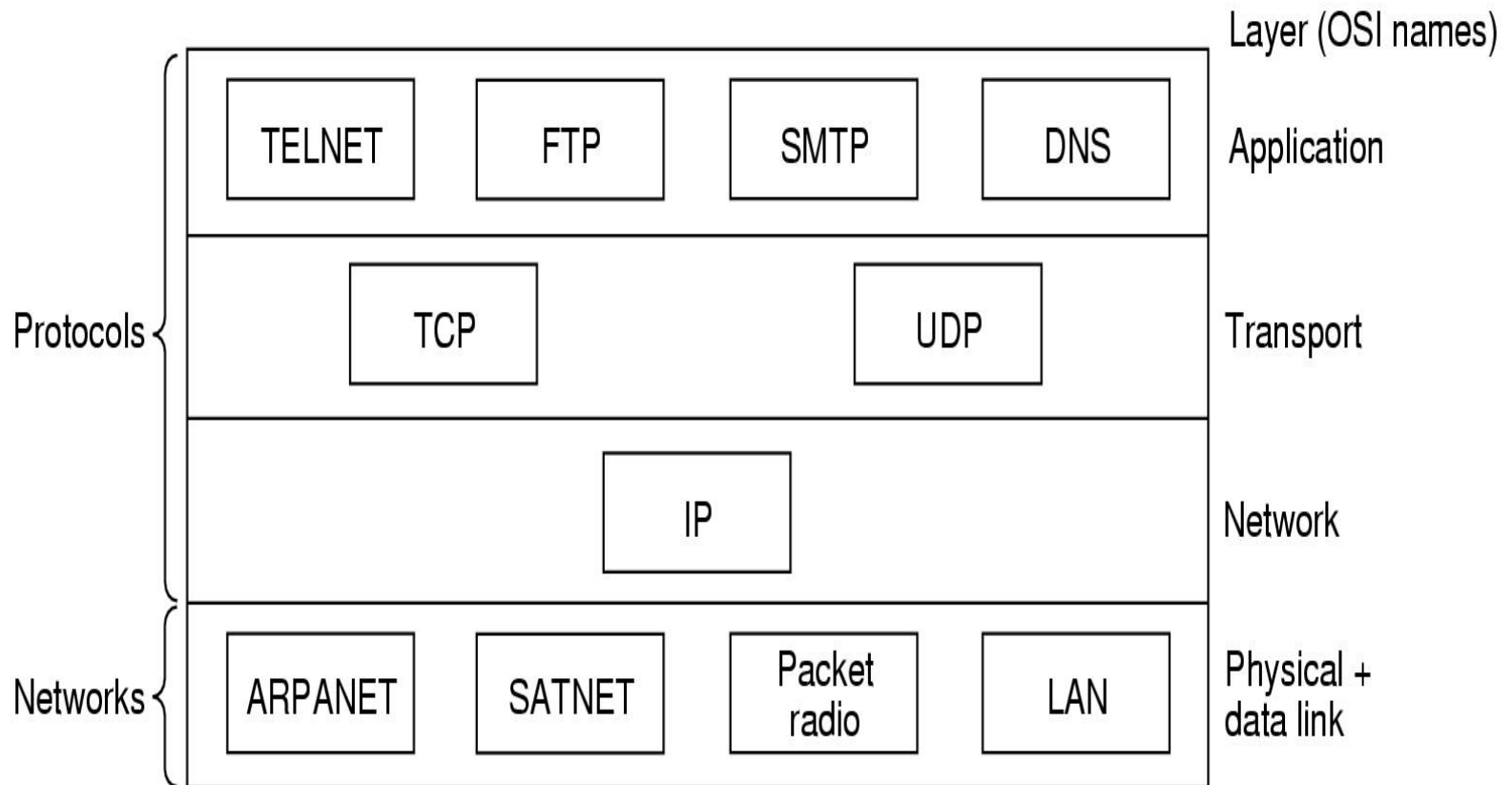


传输层：提供可靠的面向连接的传输层协议TCP和不可靠的非连接传输层协议UDP。

2.4 TCP/IP模型

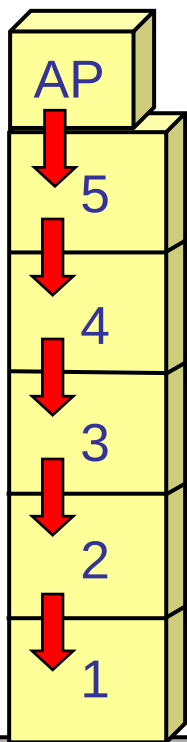


TCP/IP模型



计算机 1 向计算机 2 发送数据 (数据的封装与拆封)

计算机 1



应用数据先传送到应用层

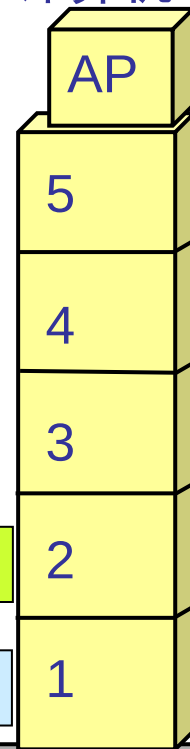
加应用层头部, 成为应用层 PDU

加传输层头部,
成为传输层 PDU

加网络层头部,
成为分组

加数据链路层头、
尾部, 称为帧

计算机 2



应用数据

H₅ 应用数据

H₄ H₅ 应用数据

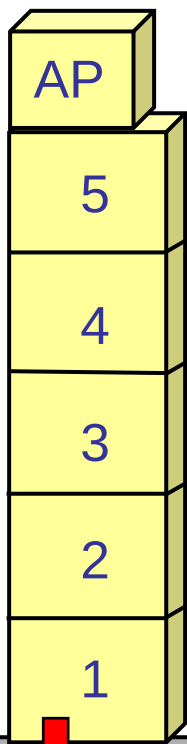
H₃ H₄ H₅ 应用数据

H₂ H₃ H₄ H₅ 应用数据 T₂

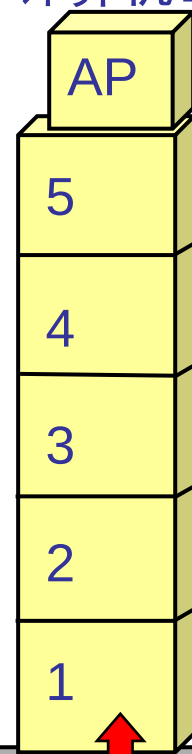
10100110100101 比特流 110101110101

计算机 1 向计算机 2 发送数据 (数据的封装与拆封)

计算机 1



计算机 2

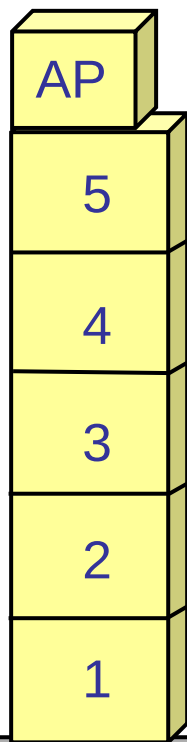


物理传输媒体

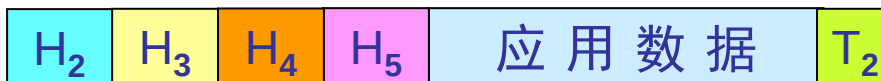
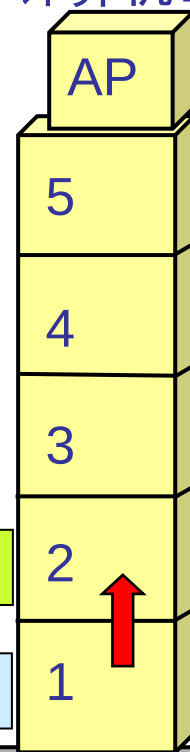


计算机 1 向计算机 2 发送数据 (数据的封装与拆封)

计算机 1



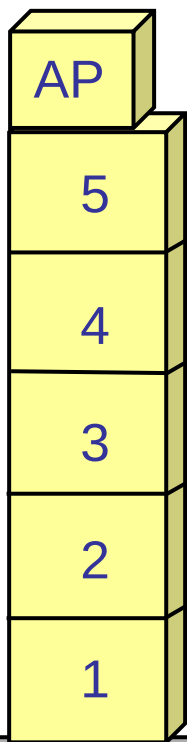
计算机 2



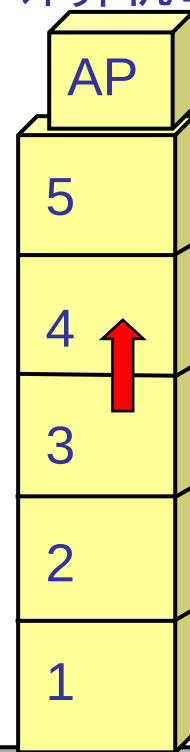
10100110100101 比特流 110101110101

计算机 1 向计算机 2 发送数据 (数据的封装与拆封)

计算机 1

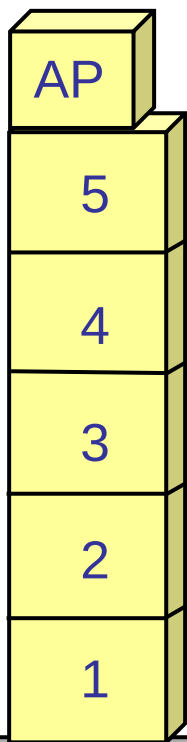


计算机 2

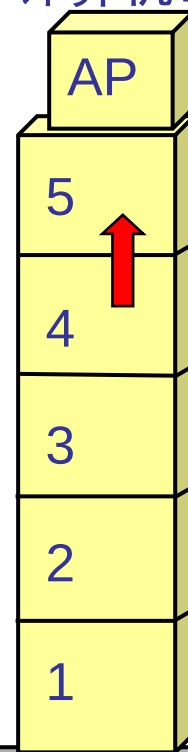


计算机 1 向计算机 2 发送数据 (数据的封装与拆封)

计算机 1

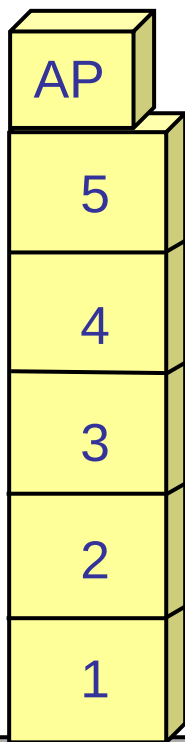


计算机 2

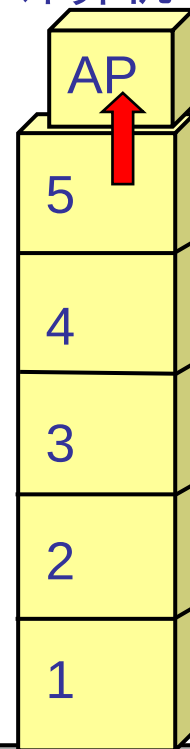


计算机 1 向计算机 2 发送数据 (数据的封装与拆封)

计算机 1



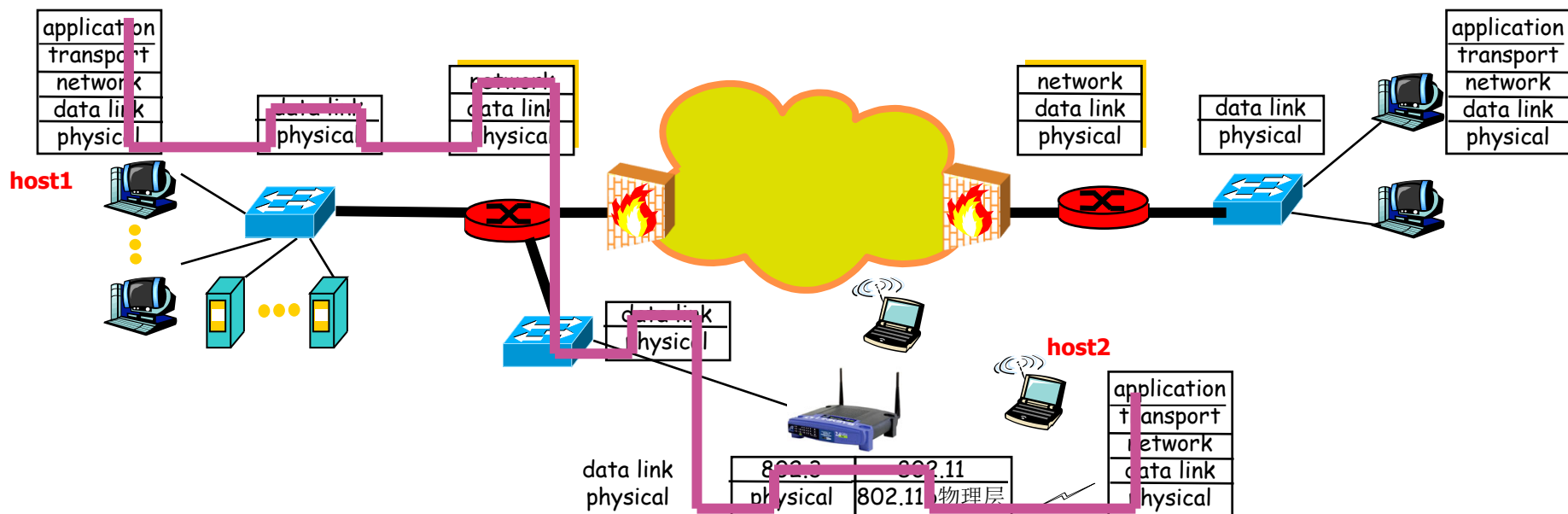
计算机 2

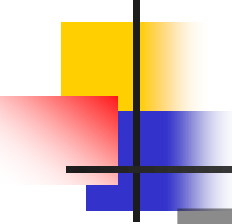


用数据

网络的哪些层是必要的？
网络的分层与哪些因素
有关？什么是跨层设计
和扁平化设计？

Internet体系与数据流程示意图





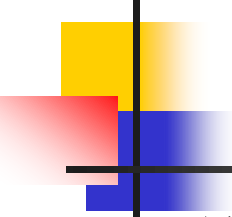
2.5 OSI与TCP/IP模型的比较

相同点：

1. 都是基于独立的协议栈概念。
2. 两者都有功能相似的应用层、传输层、网络层。

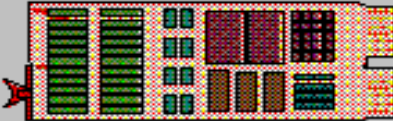
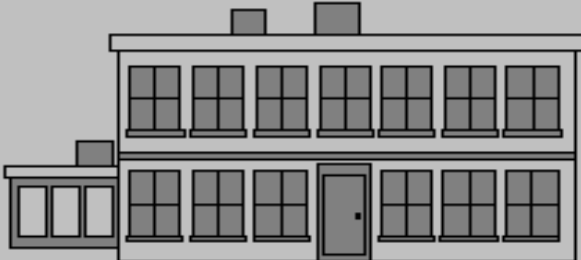
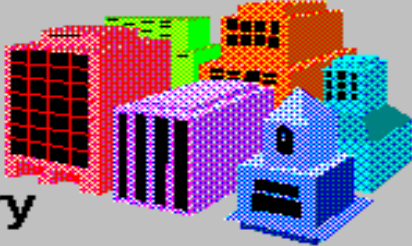
不同点：

1. 在OSI模型中，严格地定义了服务、接口、协议；在TCP/IP模型中，并没有严格区分服务、接口与协议。
2. OSI模型支持非连接和面向连接的网络层通信，但在传输层只支持面向连接的通信；TCP/IP模型只支持非连接的网络层通信，但在传输层有支持非连接和面向连接的两种协议可供用户选择。
3. TCP/IP模型中不区分、甚至不提起物理层和数据链路层。



2.6 网络的分类

- 从网络连接的区域看网络可分为：
 - 个域网（Personal Area Network, PAN）：只有数十米的区域，有蓝牙（802.15）、ZigBee（802.15.4）以及UWB（超宽带）等技术
 - 局域网（Local Area Network, LAN）：10m~nkm，大楼，校园
 - 城域网（Metropolitan Area Network, MAN）：城市，几十公里
 - 广域网（Wide Area Network, WAN）：国家和地区，100km~1000km
 - 互连网（internet）：由众多网络互连而成

Distance	Processors located in same	Network Type
0.1 m	board 	Data Flow Machine
1 m	system 	Multiprocessor
10 m	 room, building, campus	L A N Local Area Network
100 m		
1 km		
10 km	city country 	W A N Wide Area Network
100 km		
1000 km	continent	internet Interconnection of WANs
10000 km	planet 	

■ 从传输媒体看

- 有线网络
- 无线网络

■ 从功能上分

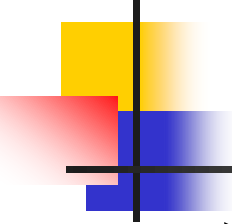
- 主干网(backbone): 高速, 功能简捷
- 接入网(access network): 功能复杂, 种类繁多

■ 从网络使用者分

- 公用网 (public network):指国家的电信公司或ISP出资建造的大型网络, 公众通过缴费即可使用。
- 专用网 (private network): 指某个单位所有, 仅为本单位人员提供的网络。

■ 从连接方式看

- 广播型网络: 多个用户共享同一信道的网络, 也称共享媒体型网络
- 点-点网络: 由许多点-点的链路连接而成, 也称交换型网络



小结

■ 本章要求

- 重点掌握**OSI**七层模型和**TCP/IP**模型，熟悉网络体系结构（层、服务、协议等基本概念）

■ 作业

- 第四版，第一章：13，18，20，22，33，36
- 或第五版，第一章：16，17，31，34，
 - 补充：1.在无连接通信和面向连接的通信两者之间，最主要的区别是什么？
 - 2.**OSI**的哪一层处理以下问题：（a）把传输的位流分成帧，（b）在通过子网的时候决定使用哪条路由路径。



Chapter 3 物理层

- **3.1**数据通信的理论基础
- **3.2**传输介质
- **3.3**数字调制与多路复用
- **3.4**交换技术

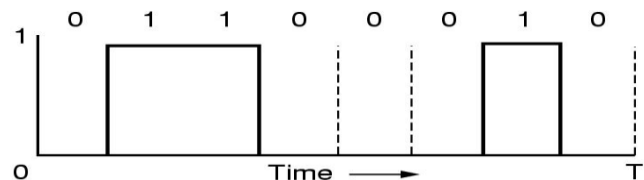


3.1 数据通信的理论基础

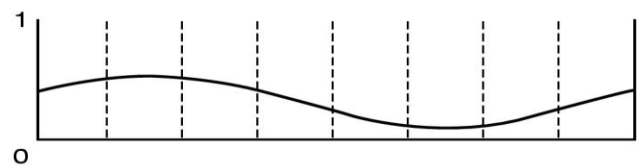
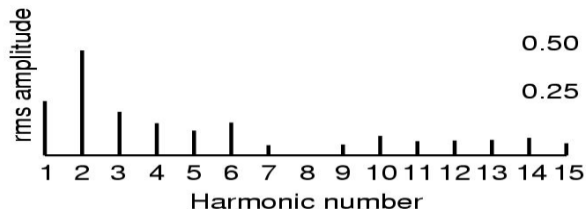
■ 3.1.1 Fourier 级数

- 任何正常的周期为T的函数 **$g(t)$** 都可以展开成**Fourier级数**，即

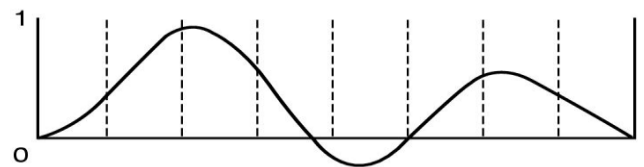
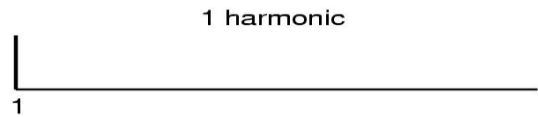
$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft)$$



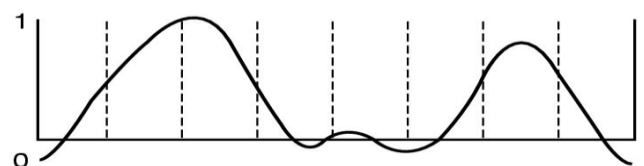
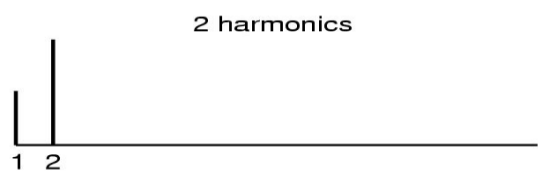
(a)



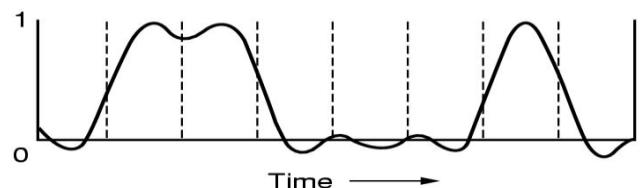
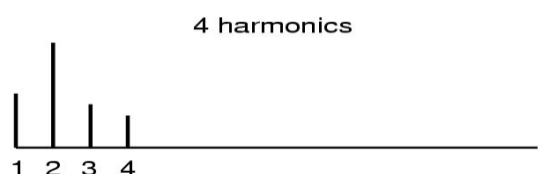
(b)



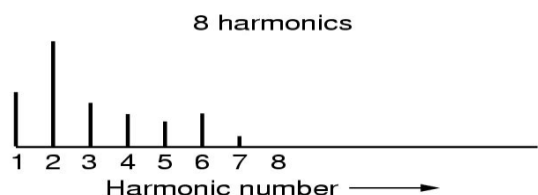
(c)



(d)



(e)



Example: 8 bit的ASCII码
“b”的位模式 01100010,
利用傅里叶级数求得
 $(a_n^2 + b_n^2)^{1/2}$

假定比特率为: **a** 比特/秒
则周期 **T=8/a** 秒

$$f = 1/T = a/8 \text{ Hz}$$

对于普通话音级线路, 截止
频率大约为 **3000 Hz**。

即能通过的最高次谐波

$$nf = 3000$$

$$n = 3000 / (a/8) \\ = 24000/a$$

传输速率与信道的带宽成正比

限制信道的带宽就是限制传输速率

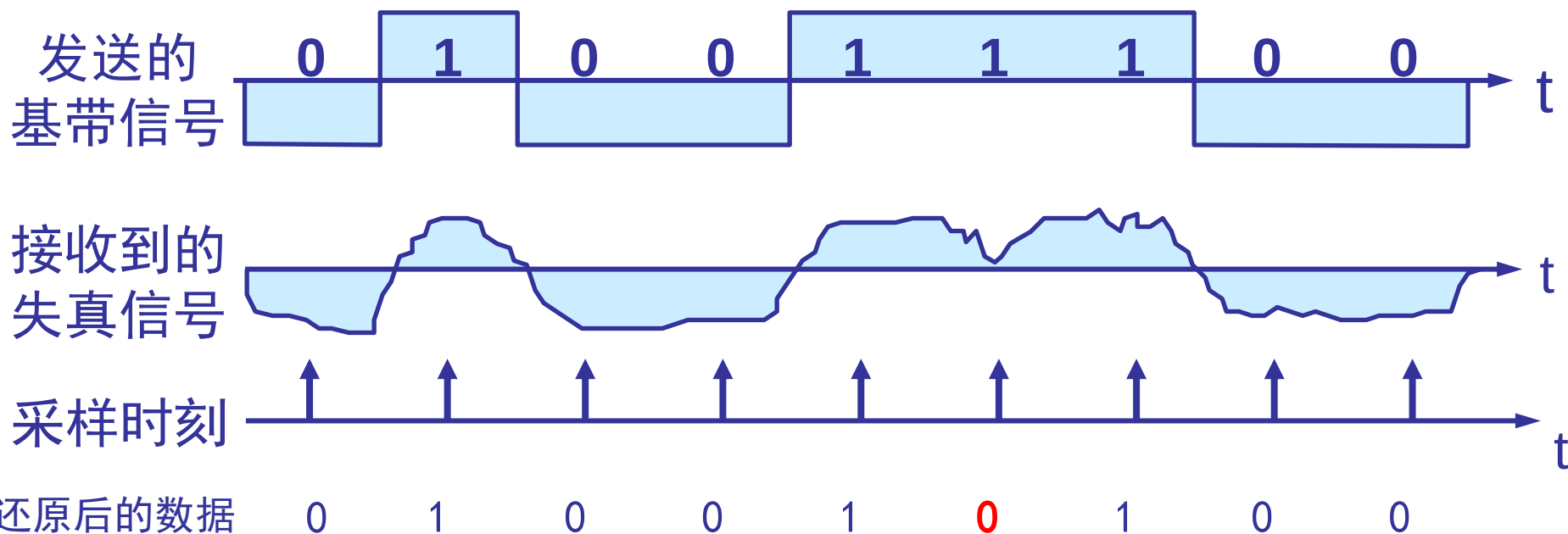
■ 带宽与传输速率

- 模拟信道的带宽以赫兹（**Hz**）为单位。
- 数字信道的带宽（传输速率）以每秒多少比特为单位（**bps**），有时候也以字节/秒（**B/s**）

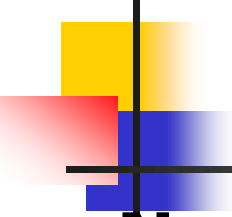
a	8/a	1/T	$n=f_{\text{截}}/f_1=8f_{\text{截}}/a$
bps	T(ms)	f_1	# harmonics
300	26.67	37.5	80
600	13.33	75.0	40
1200	6.67	150.0	20
2400	3.33	300.0	10
4800	1.67	600.0	5
9600	0.83	1200.0	2
19200	0.42	2400.0	1
38400	0.21	4800.0	0

信号在信道传输中的其他干扰因素

- 信号的衰减、噪声、干扰等。



误比特率=错误比特数/传输总比特数，若误比特率 p ，帧长 n 比特，误帧率 $1-(1-p)^n$
设 $p=10^{-4}$, $n=1518$ B, 误帧率0.7, $p=10^{-5}$, 误帧率0.11, $p=10^{-10}$, 误帧率0.0000012.



3.1.2信道的最大传输速率

■ Nyquist 定理（有限带宽的无噪声信道）

- 如果一个任意的信号通过带宽为**B**的低通滤波器，那么每秒采样**2B**次就能完整地重现通过该滤波器的信号。
- 最大数据传输率= $2B\log_2 V$ （b/s）

■ Shannon定理

- 对任何带宽为**B Hz**，信噪比为**S/N**的信道，
最大数据传输速率= $B \log_2 (1+S/N)$ （b/s）
- 信道的带宽或信道中的信噪比越大，则信息的极限传输速率就越高。
- 通常信噪比表示形式 $10\log_{10} S/N$ ，以分贝（**dB**）为单位，
20分贝的信噪比**S/N=100**。



Chapter 3 物理层

- **3.1**数据通信的理论基础
- **3.2**传输介质
- **3.3**数字调制与多路复用
- **3.4**交换技术



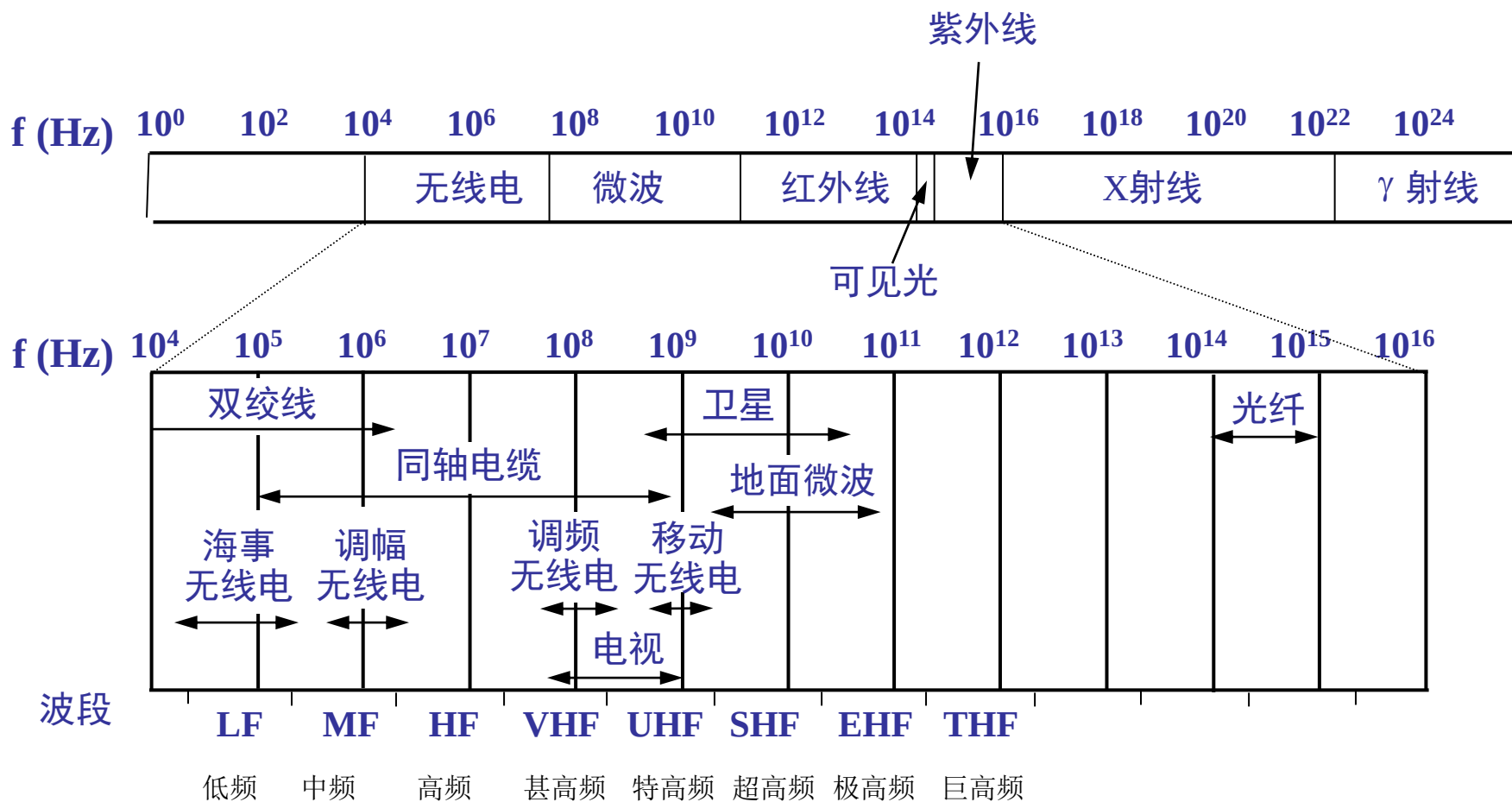
3.2 传输介质

- **3.2.1 双绞线**
- **3.2.2 同轴电缆**
- **3.2.3 电力线**
- **3.2.4 光纤**
- **3.2.5 无线传输**

导向性传输媒体

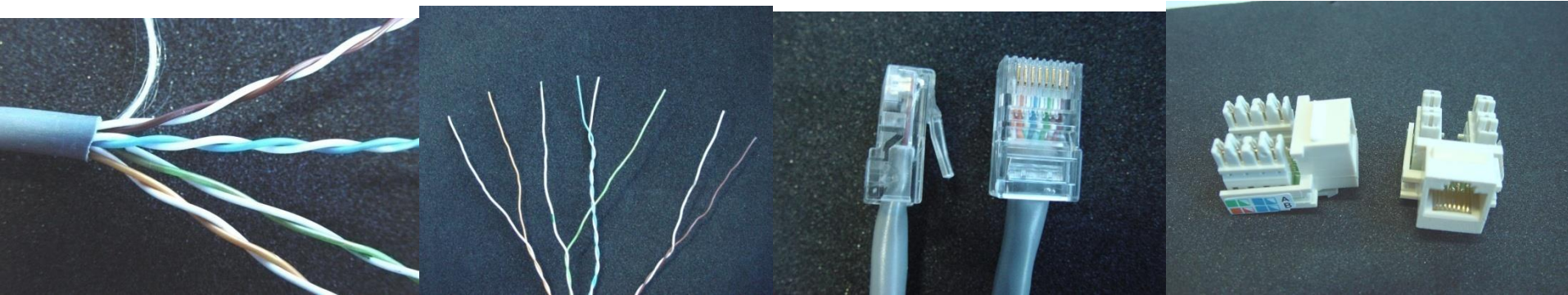
非导向性传输媒体

电磁波的频谱



3.2.1 双绞线 TP(Twisted Pair)

- 由两条相互绝缘的铜线组成，其典型粗细约**1mm**，两条象螺纹一样绞在一起。
 - **Shielded Twisted Pair(STP)** 屏蔽双绞线
 - **Unshielded Twisted Pair(UTP)** 无屏蔽双绞线
- 双绞线的传输距离一般为**100M**，既可传输模拟信号，也可传输数字信号。

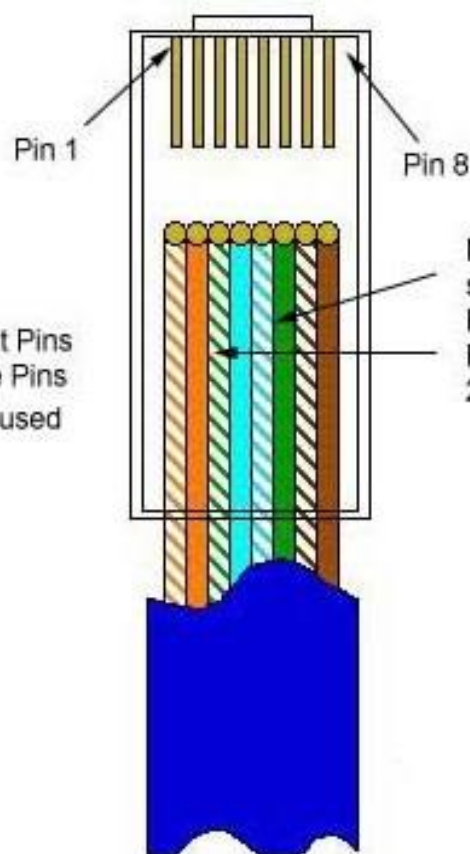


Categories of Unshielded Twisted Pair

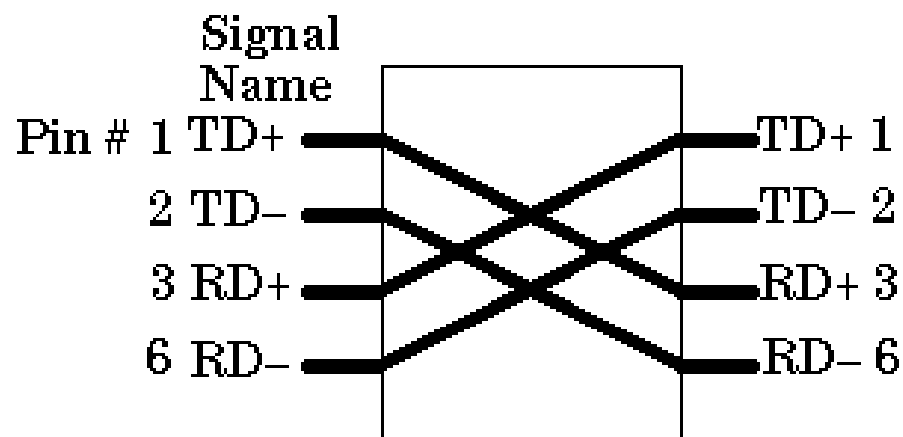
The EIA/TIA (Electronic Industry Association/Telecommunication Industry Association) has established standards of UTP.

Type	Use
Category 1	Voice Only (Telephone Wire)
Category 2	Data to 4 Mbps (LocalTalk)
Category 3	Data to 10 Mbps (Ethernet)
Category 4	Data to 20 Mbps (16 Mbps Token Ring)
Category 5	Data to 100 Mbps (Fast Ethernet)
Category 5e	Data to 100 Mbps (Fast Ethernet)
Category 6	Data to 250 MHz (Gigabit Ethernet?)
Category 7	Data to 600 MHz

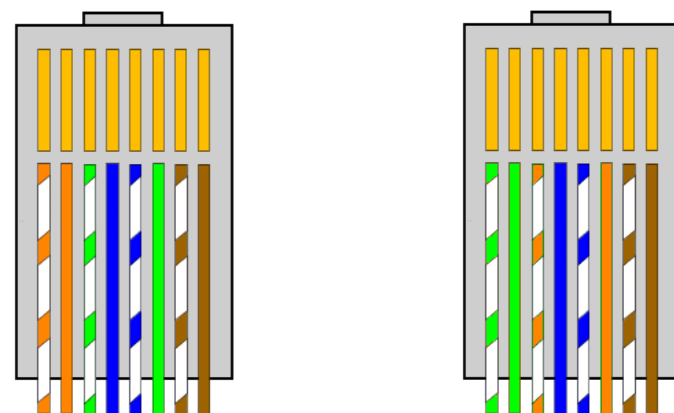


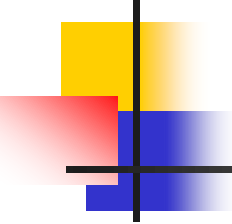


Notice that the Green pair has to be separated before crimping on the RJ45 plug. If this is not done, the Receive Data will be split between 2 pairs and performance will degrade!



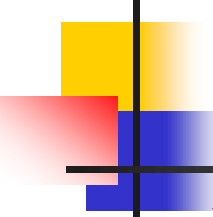
双绞线头的制作过程





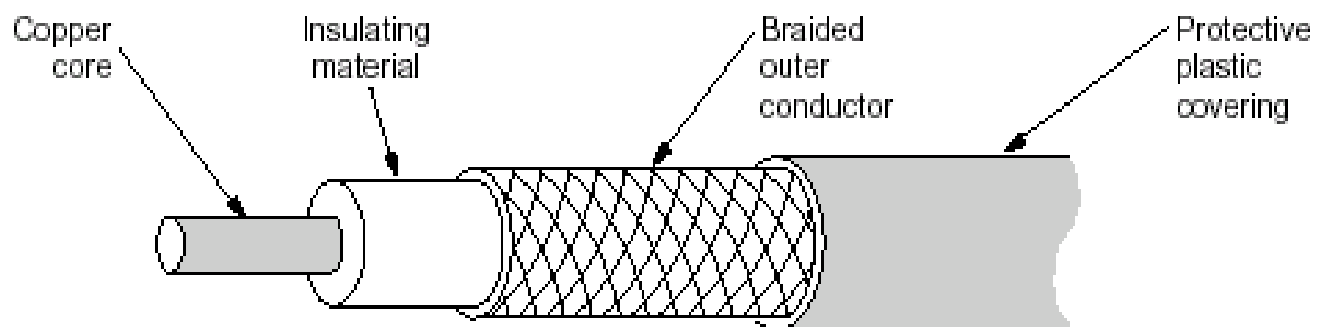
STP



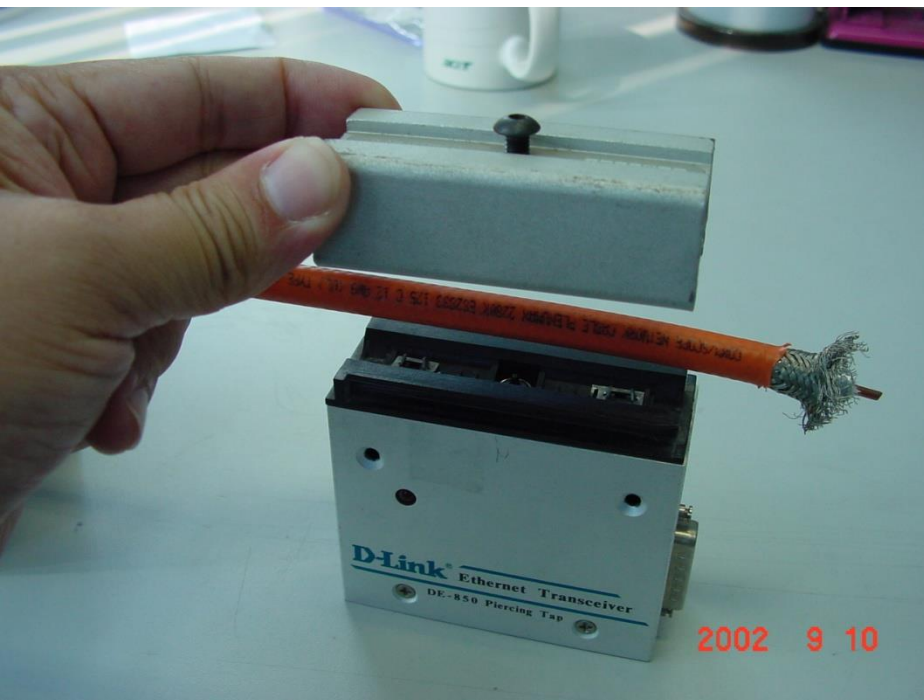


3.2.2同轴电缆

- **基带同轴电缆 (Baseband Coax)**：阻抗匹配为 **50Ω**，用于**数字传输**，1公里电缆可达**1~2Gbps**的传输速率。又分为：
 - 粗缆 (**Thick**)：如**10Base-5**，单段长度≤**500**米，最长**5**段达**2.5**公里。
 - 细缆 (**Thin**)：如**10Base-2**，单段长度≤**185**米，最长**5**段达**925**米。
- **宽带同轴电缆 (Broadband Coax)**：阻抗匹配为 **75Ω**，用于电视信号的**模拟传输 (CATV)**，带宽可达**800MHz**以上，采用**FDM**技术。传输数字信号时，要使用**Cable MODEM**这样的特殊设备，现在综合有线电视网络已成为**MAN**的一种形式。



粗、细同轴电缆的比较





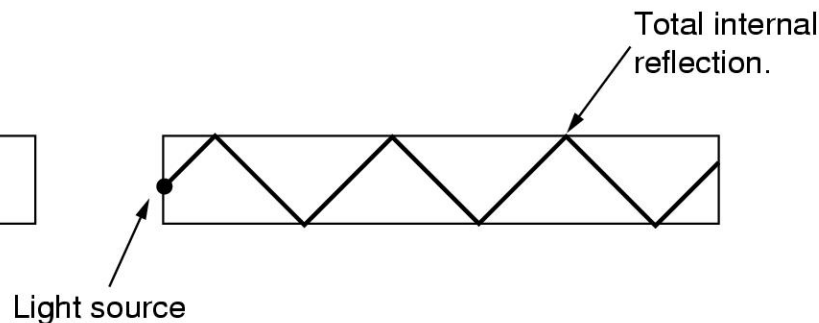
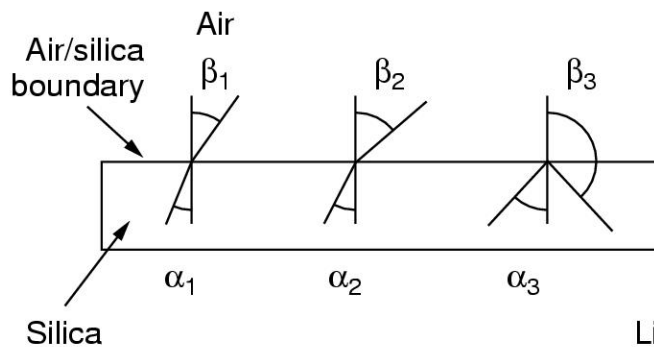
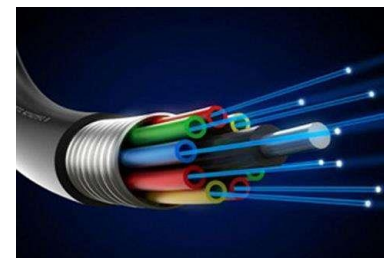
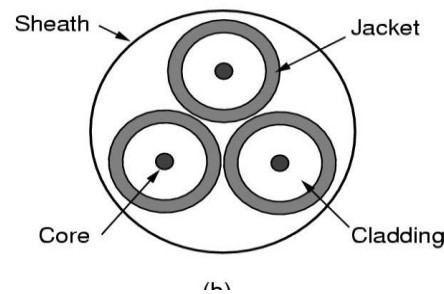
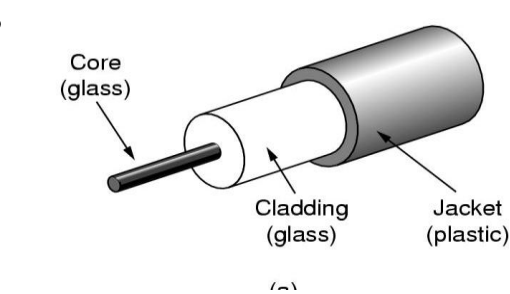
3.2.3 电力线

- PLC（power line communication），用**电力线**作为信号的传输媒介，通过载波方式来传输模拟或数字信号的技术。
- 自**20世纪50年代**起，电力线通信技术开始应用于中压和低压电网上，主要面向电力行业的运用，包括**电力线自动抄表、电网负载控制和供电管理**等。
- 特点：
 - 用户数目最多，不用去重新铺设专用线路
 - 负载阻抗呈现随机性的特点，造成电力线载波通信的收、发信机的输出阻抗与输入阻抗很难和线路的阻抗相匹配，信号在传输过程中**衰减**很严重。电力线网络的拓扑具有时变性。

3.2.4 光纤

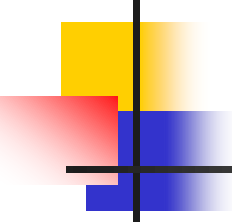
1966年7月，英籍、华裔学者高锟博士(K.C.Kao)发表了论文《光频率的介质纤维表面波导》，从理论上分析证明了用光纤作为传输媒体以实现光通信的可能性，因此获得**2009年诺贝尔物理学奖**。

- 光纤由传导光波的高纯石英玻璃纤维和保护层（**jacket**）构成，其中**纤芯（core）**的折射率大于包裹着它的**包层（cladding）**折射率，这样光信号就被保持在纤芯中不会散播出去。
- 经常将多根光纤封装在坚固的外壳（**sheath**）中，形成所谓的**多芯光缆**。



(a)

(b)

- 
- 光发射系统：光源、传输介质、检测器
 - 有两种光源可用于信号源：**LED**（发光二极管）和**ILD**（注入型激光二极管）。以光信号的有和无来表示二进制的“**1**”和“**0**”。
 - 接收端由光电二极管构成的光检测器将光信号转换成电信号。
 - 光纤的连接：
 - 机械式：快速，一般不需特殊设备，新技术和连接器改善了接合的损耗（有些 $< 0.1 \text{ dB}$ ），适合于小数量和应急的应用。
 - 熔结：需要昂贵的特殊设备，极低的损耗（有的可达 $< 0.05 \text{ dB}$ ），长距离链路的唯一方法。

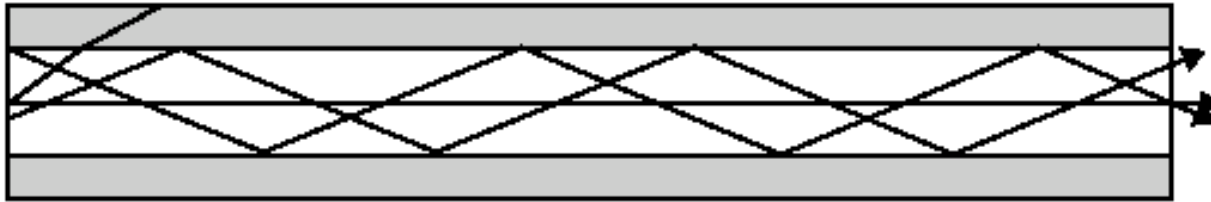


光纤的类型

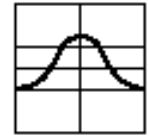
- 多模光纤
 - 突变型**SI**（ **Multimode step-index fiber** ）：
the reflective walls of the fiber move the light pulses to the receiver
 - 渐变型**GI**（ **Multimode graded-index fiber** ）：
acts to refract the light toward the center of the fiber by variations in the density
- 单模光纤（ **Single mode fiber** ）
 - the light is guided down the center of an extremely narrow core

Optical Fiber Transmission Modes

Input pulse

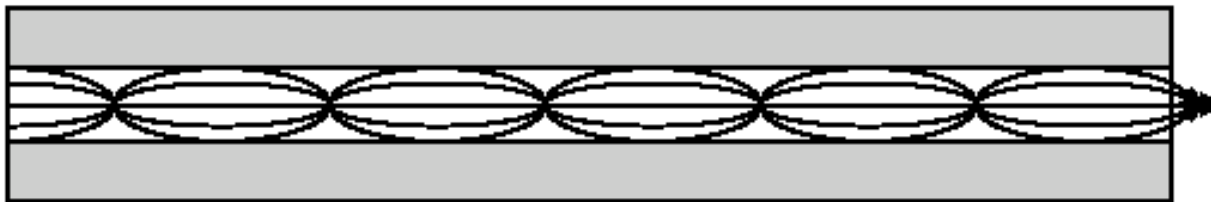


Output pulse



(a) Step-index multimode

Input pulse

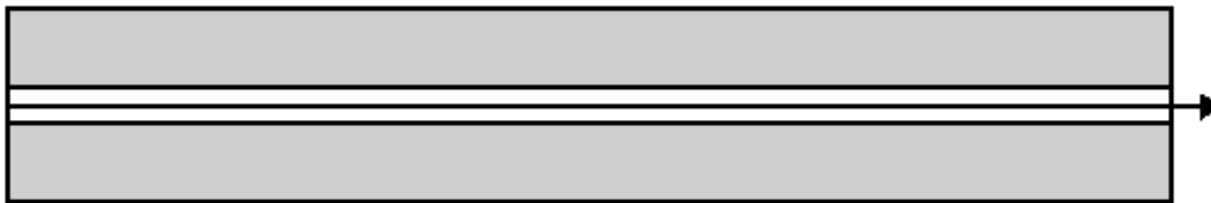


Output pulse

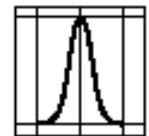


(b) Graded-index multimode

Input pulse

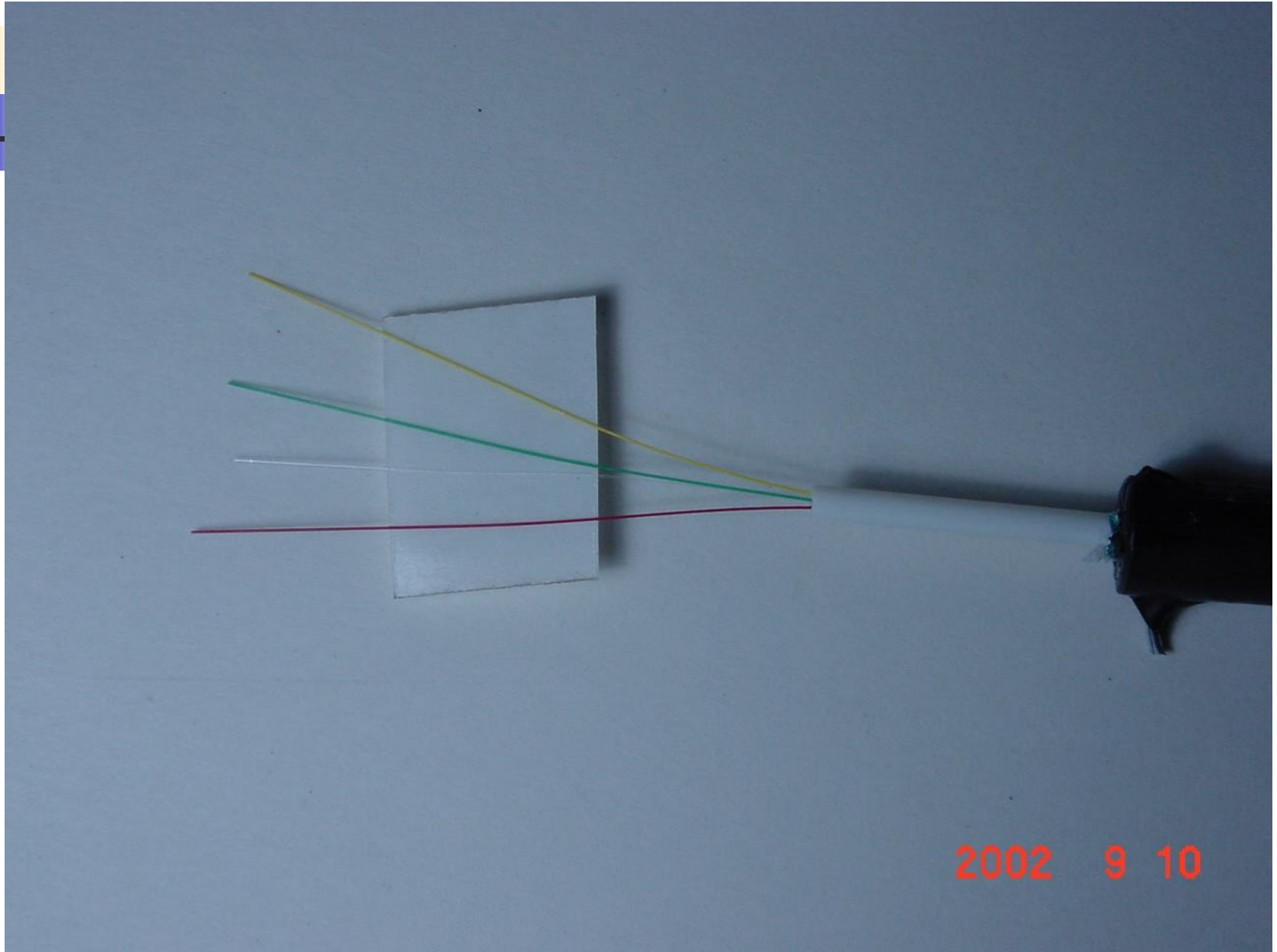


Output pulse

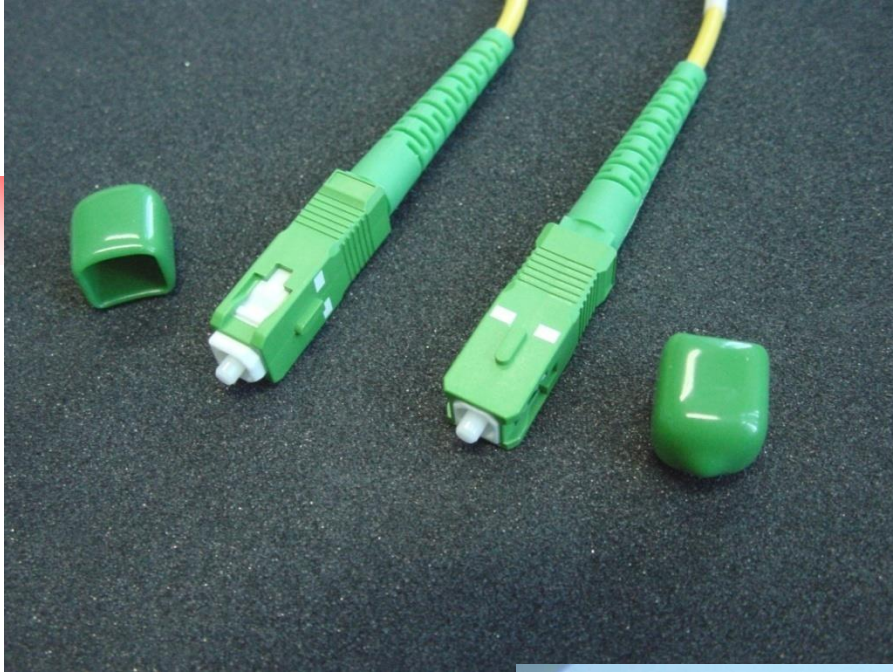


(c) Single mode

实际的光纤

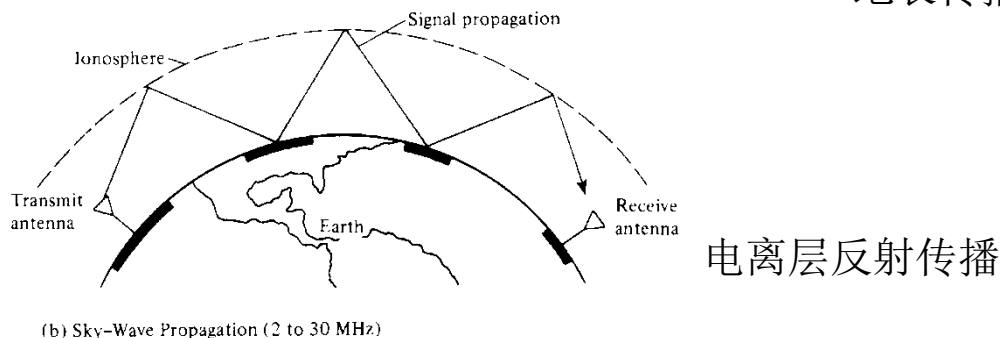
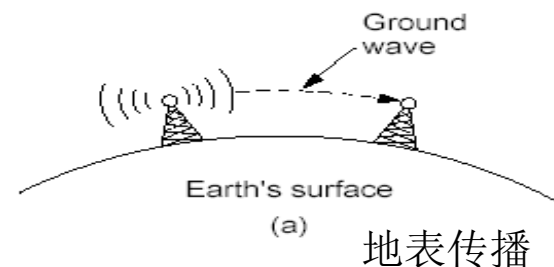


2002 9 10



3.2.5 无线传输

- 无线传输所使用的**频段**很广。
- **低频**LF、**中频**MF波段电波沿**地表传播**，**高频**HF和**甚高频**VHF波段的地表电波会被地球吸收，但可通过地球上空的**电离层反射**实现**长距离传播**。
- **微波**在空间主要是**直线传播**。
 - 地面微波接力通信
 - 卫星通信
- 光通信



微波传输

- 在**300MHz**以上，微波通过抛物状天线把能量集中于一小束，具有极高的信噪比，沿直线实现可视距传输。
- 在地面因地表弯曲，需中继站接续微波信号，**100米**高的塔可接续约**80公里**。
- 天气和频率的影响可造成多路减弱。
- 开放的波段：**2.4 GHz ~ 2.484 GHz**，即工业/科学/医学波段。

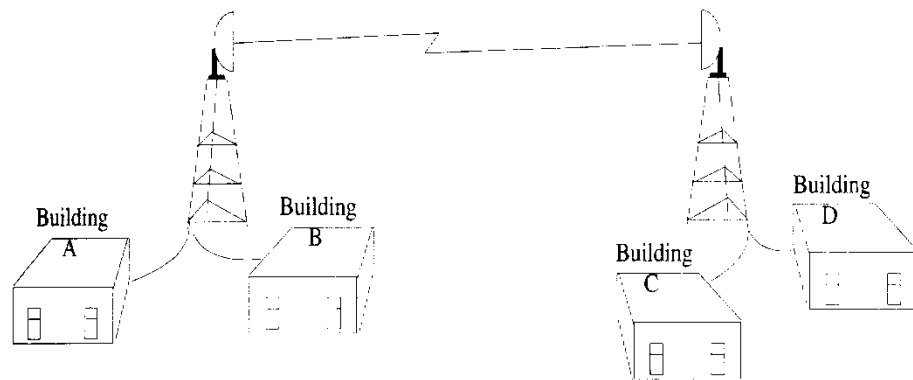


Figure 1.13 Conceptual view of microwave communication.



红外线与毫米波

- 有方向性，便宜，穿透能力差，传输距离短。
- 如：遥控器，防盗报警等



光通信

- **激光通信**，波束窄，需要对准，受天气（风和温度）干扰大
- **LED可见光通信(Visible Light Communication)**，给 LED灯泡装上驱动芯片，就可以控制它每秒数百万次闪烁，亮了表示1，灭了代表0。2011年7月份Ted大会(全球科技娱乐设计大会)上，英国爱丁堡大学工程学院教授哈拉尔德·哈斯，向人们展示了他制造的可以用来通信的VLC灯泡。他首次将VLC称为“**LiFi**”

Satellite

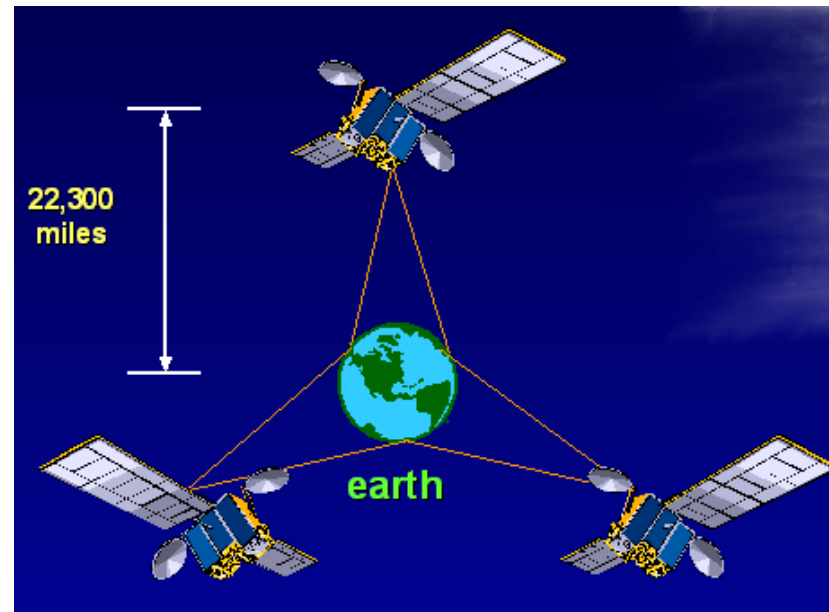
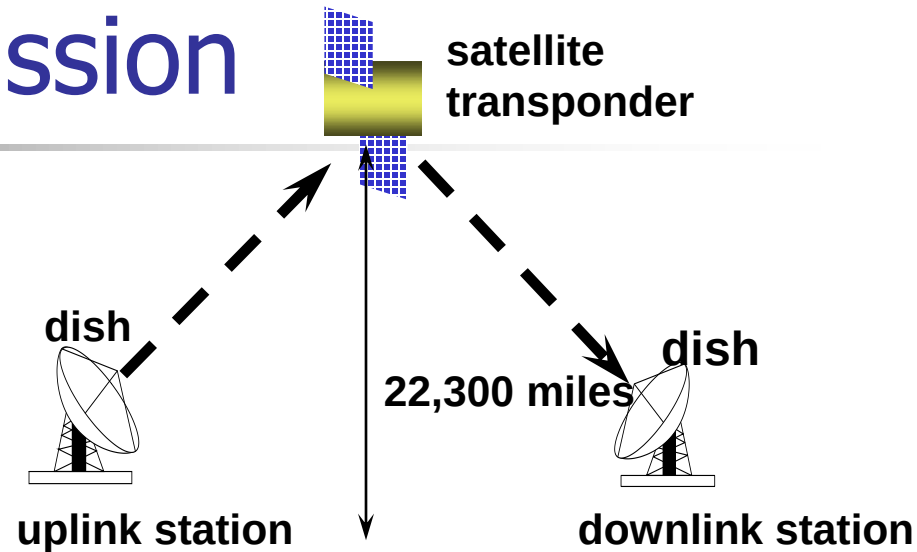
Microwave Transmission

■ 地球同步卫星

- 赤道上方约**36000km**处
- 卫星作为空间微波中继站
- 端-端传输时延**250~300ms**
- 广播信道

■ 主要卫星波段

- **C band: 4**(下行**3.7~4.2 GHz**) - **6**(uplink **5.925~6.425GHz**),
问题: 拥挤串扰
- **Ku band: 11**(下行**11.7~12.2 GHz**) - **14**(上行**14.0~14.5 GHz**),
问题: 雨水
- **Ka band: 20**(下行**17.7~21.7 GHz**) - **30**(上行**27.5~30.5GHz**),
问题: 雨水, 设备造价



低轨卫星

■ 铱星

- **66颗**，卫星高度**750km**，采用圆形极地轨道，上下行链路运行在**L波段（1.6GHz）**。

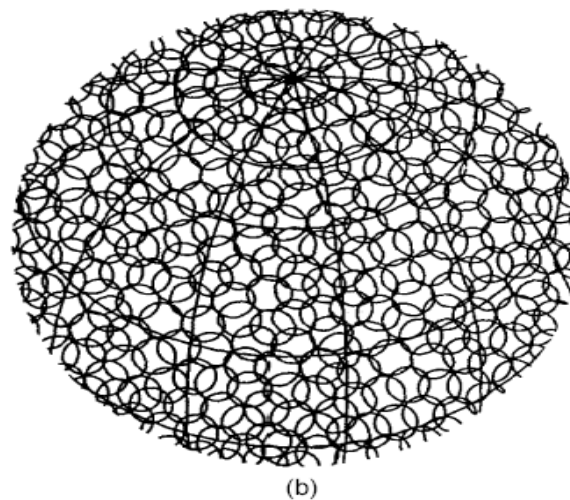
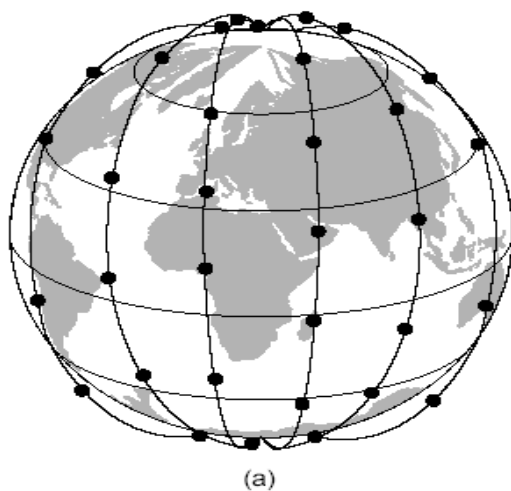


Fig. 2-57. (a) The Iridium satellites form six necklaces around the earth. (b) 1628 moving cells cover the earth.



Chapter 3 物理层

- **3.1**数据通信的理论基础
- **3.2**传输介质
- **3.3**数字调制与多路复用
- **3.4**交换技术



3.3 数字调制与多路复用

■ 3.3.1 数字调制

- **调制**是对信号源进行处理，使其变为适合传输形式的过程。即把**基带信号**变为一个相对基带频率而言频率非常高的通带信号。**通带信号**叫做已调信号，而基带信号叫做调制信号。
- **数字调制**就是把要发送的数字信号转换到适合于传输的通带模拟信号的过程。

■ 3.3.2 多路复用

- **多路复用**是指**多个用户共享一条信道**的一种机制。多路复用有**频分多路复用（FDMA）**，**时分多路复用（TDMA）**，**码分多路复用（CDMA）**几种。



3.3.1 数字调制

- 由于基带 (**baseband**) 信号在长距离的传输信道上会受到衰减、畸变及噪音等的影响, 因此在发送端必须转换成一种适合于在信道上传输的信道信号, 这个转换过程就叫调制 (**modulation**)。在接收端的相反转换过程称为解调 (**demodulation**)。
- 调制解调器 (**MODEM**) 就是调制器 (**MOdulator**) 和解调器 (**DEModulator**) 的组合。

基带传输

■ 不归零码 (NRZ)

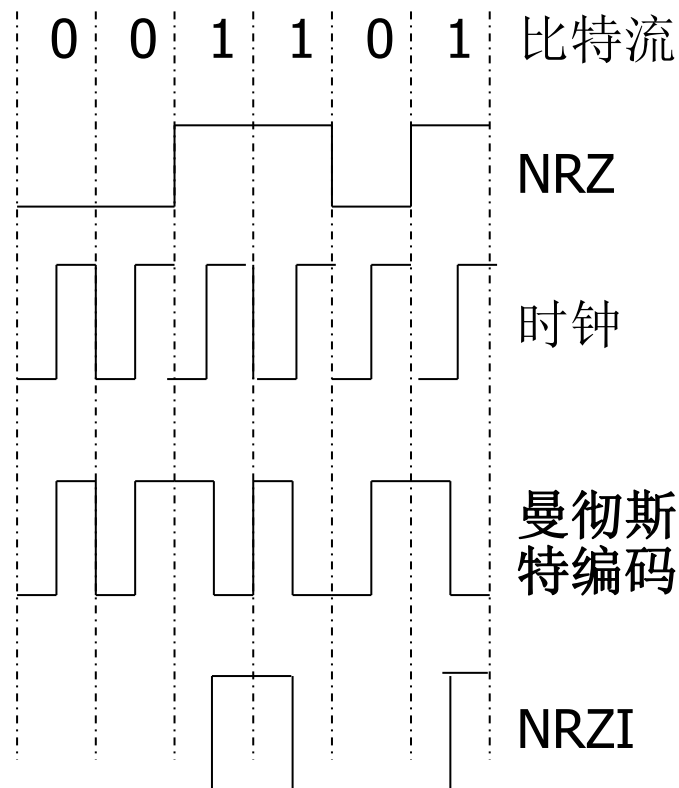
- 数字信号的最直接表示是用正负电压分别表示二进制的**0**、**1**值，对于光纤，有光表示**1**，没光表示**0**，这种编码方案称为**NRZ**。

■ 曼彻斯特编码

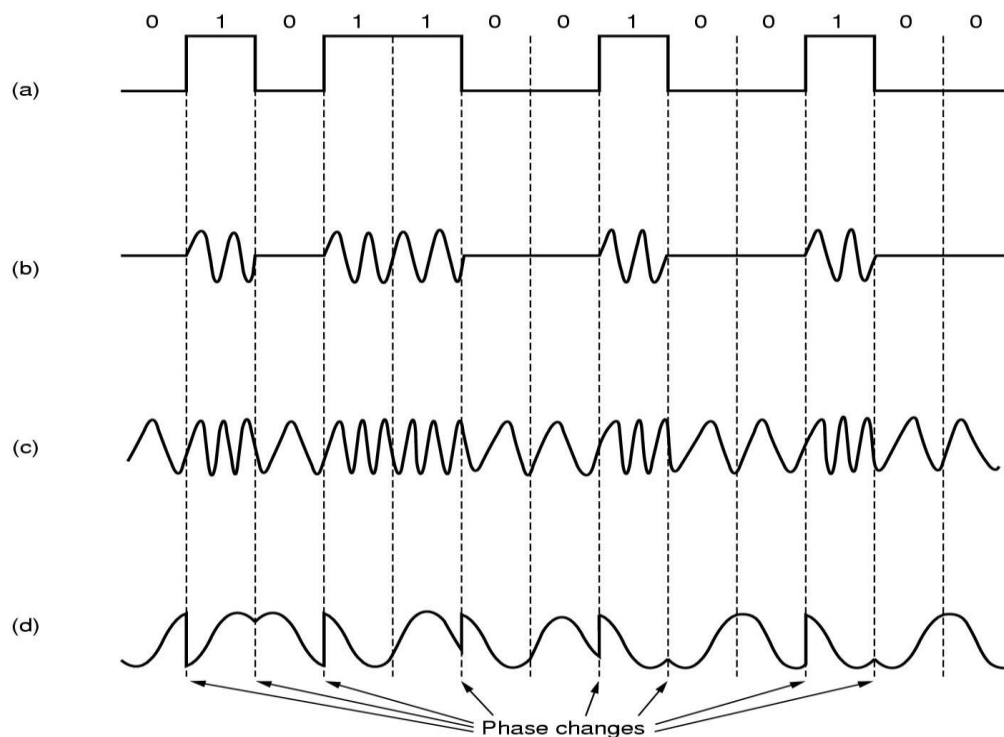
- 每一比特的中央有一变迁，可用做时钟，由高到低为“**1**”，由低到高为“**0**”，相当于时钟信号与比特流的异或

■ 不归零逆转

- 信号有跳变表示**1**，无跳变表示**0**，长串**1**不会有时钟恢复问题，但**长串0仍存在时钟恢复问题**。解决该问题可用**4B/5B**编码。



通带传输（三种基本调制）



二进制信号

幅度调制，**AM**；幅移键控，**ASK**

载波的振幅随基带数字信号而变化

频率调制，**FM**；频移键控，**FSK**

载波的频率随基带数字信号而变化

相位调制，**PM**；相移键控，**PSK**

为了提高传输速率，采用更为复杂的调制技术，如载波的初始相位随基带数字信号而变化，有BPSK、QPSK，QAM

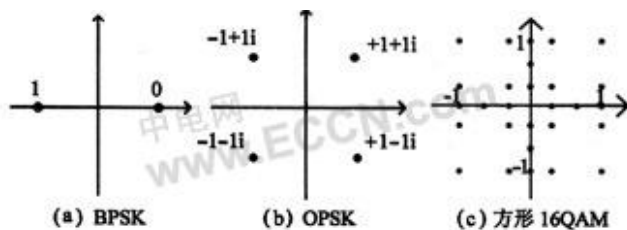
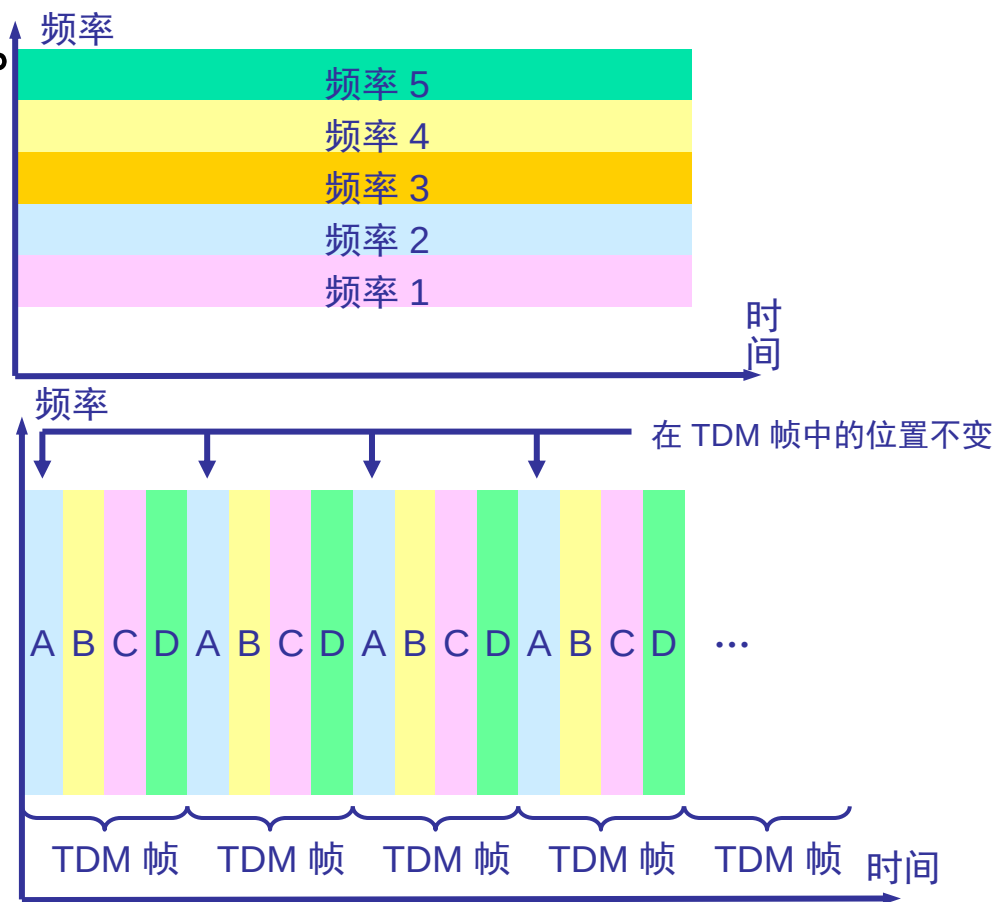


图 2 各种调制的波形图

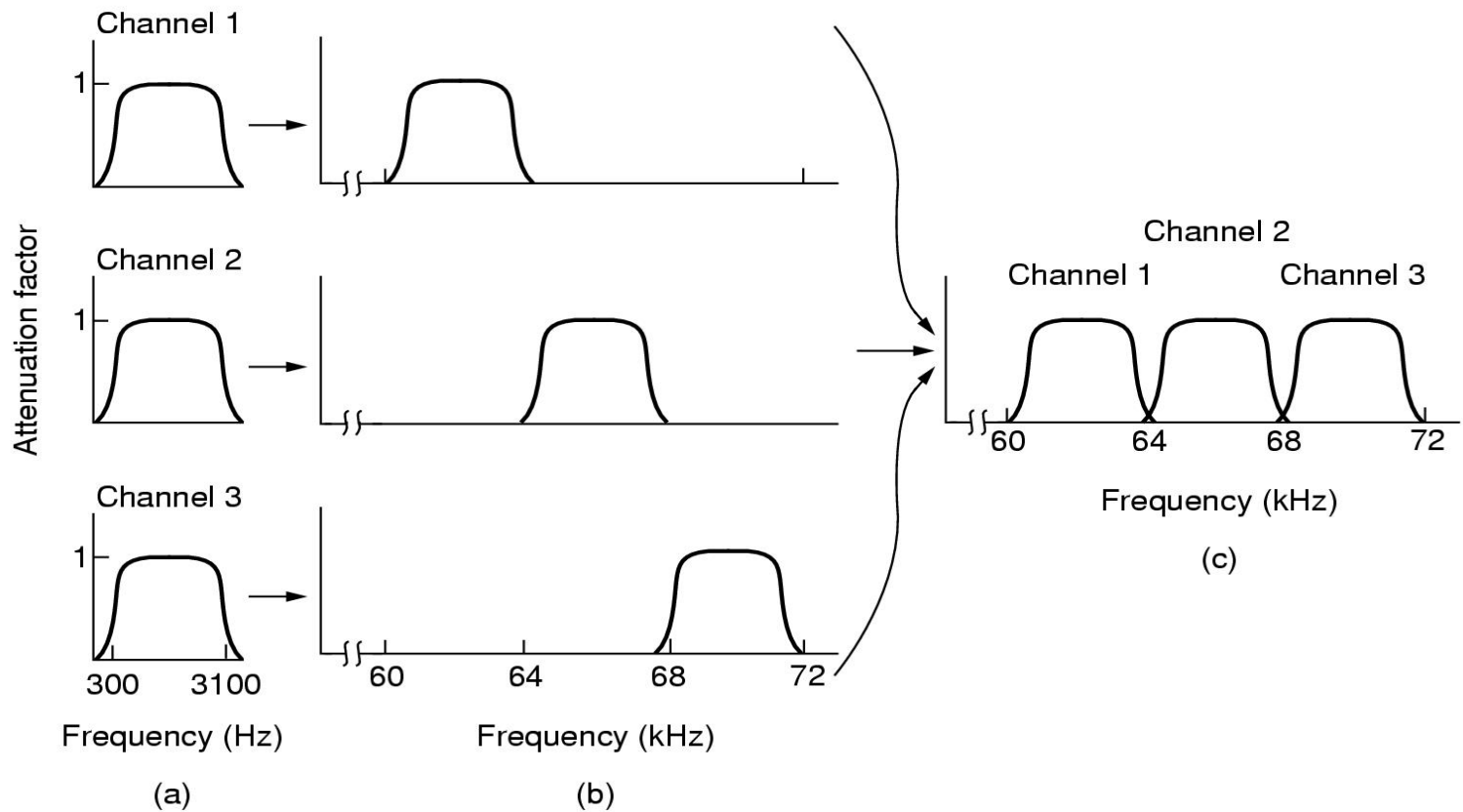
3.3.2 复用技术

- 复用技术是将多路信号组合在一条物理信道（主干 **Trunk**）上进行传输。

- 频分复用
- 时分复用
- 波分复用
- 码分复用



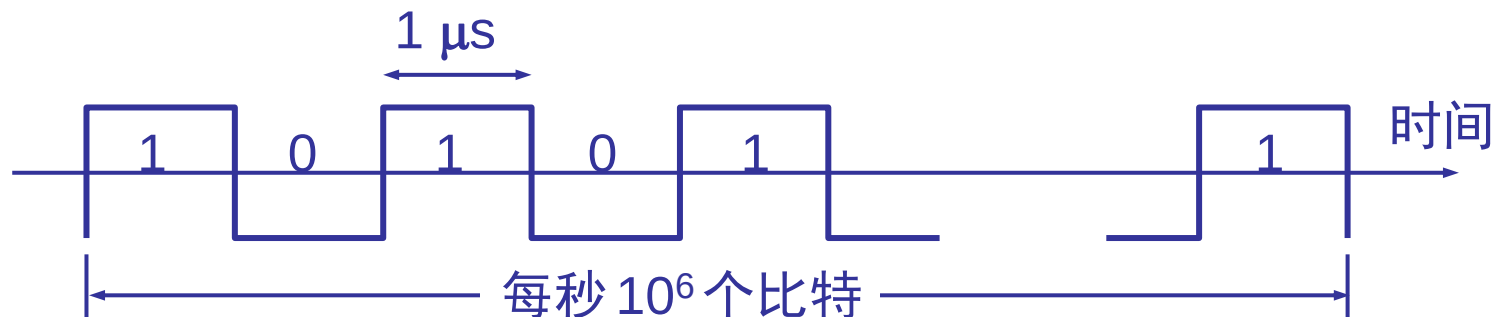
频分复用FDM



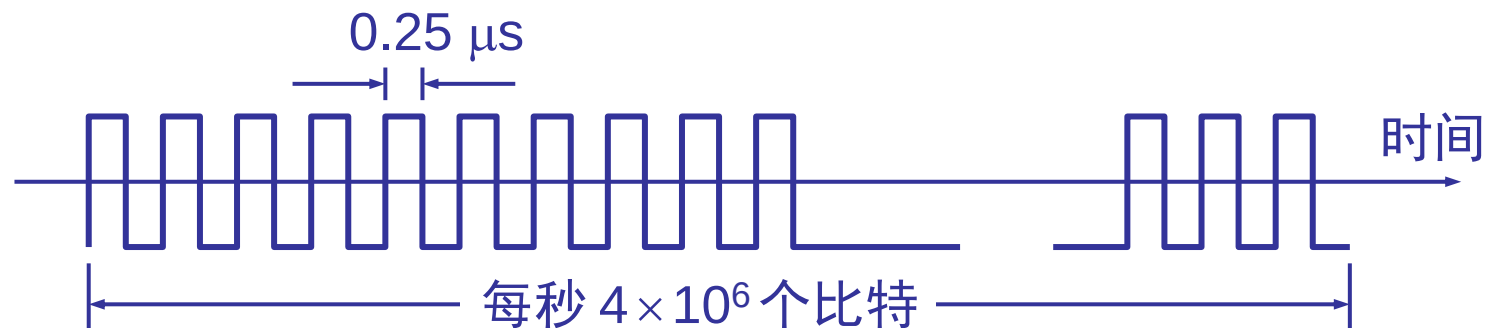
数字传输中不同速率信号的变化

- 在时间轴上信号的宽度随带宽的增大而变窄。

带宽为
1 Mb/s



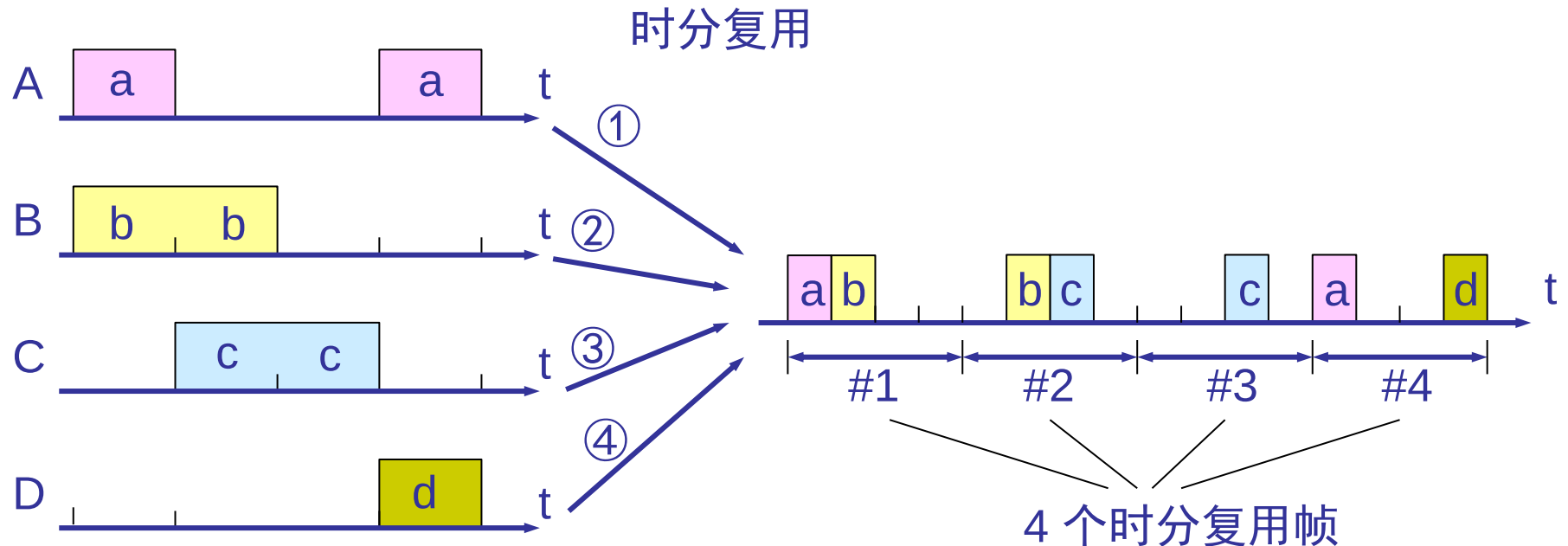
带宽为
4 Mb/s



基于时隙的时分复用

可能会造成线路资源的浪费

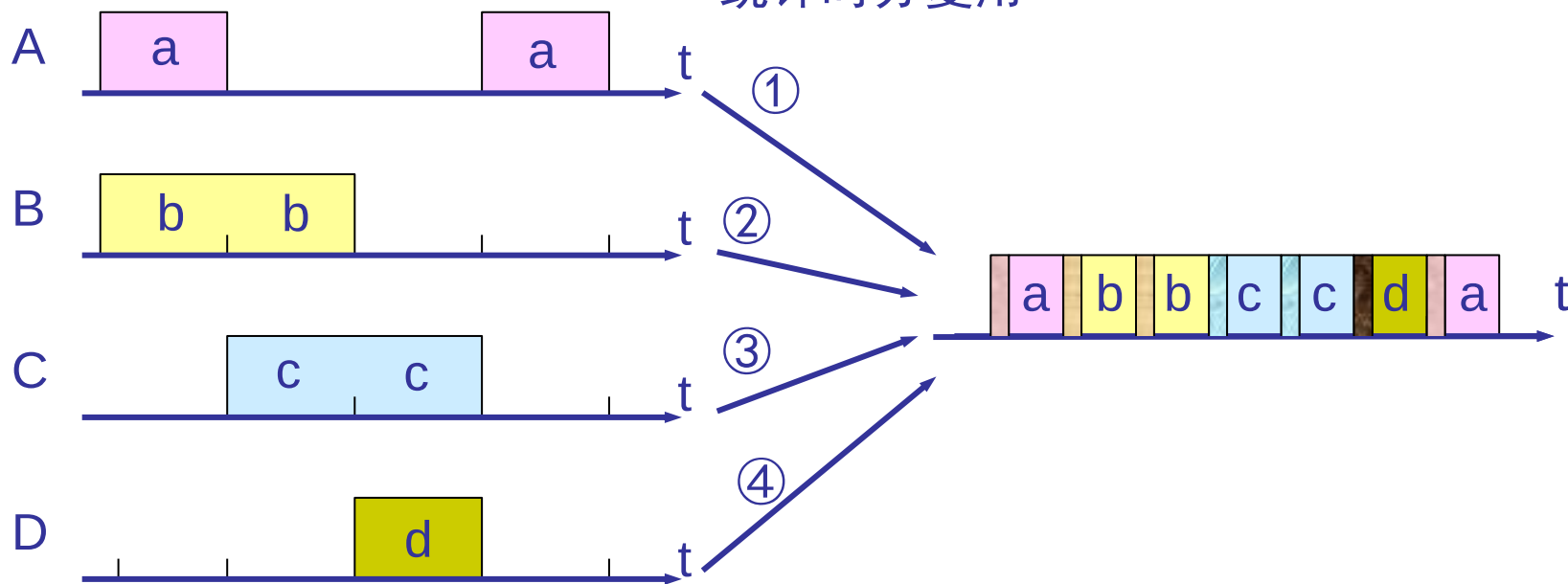
用户



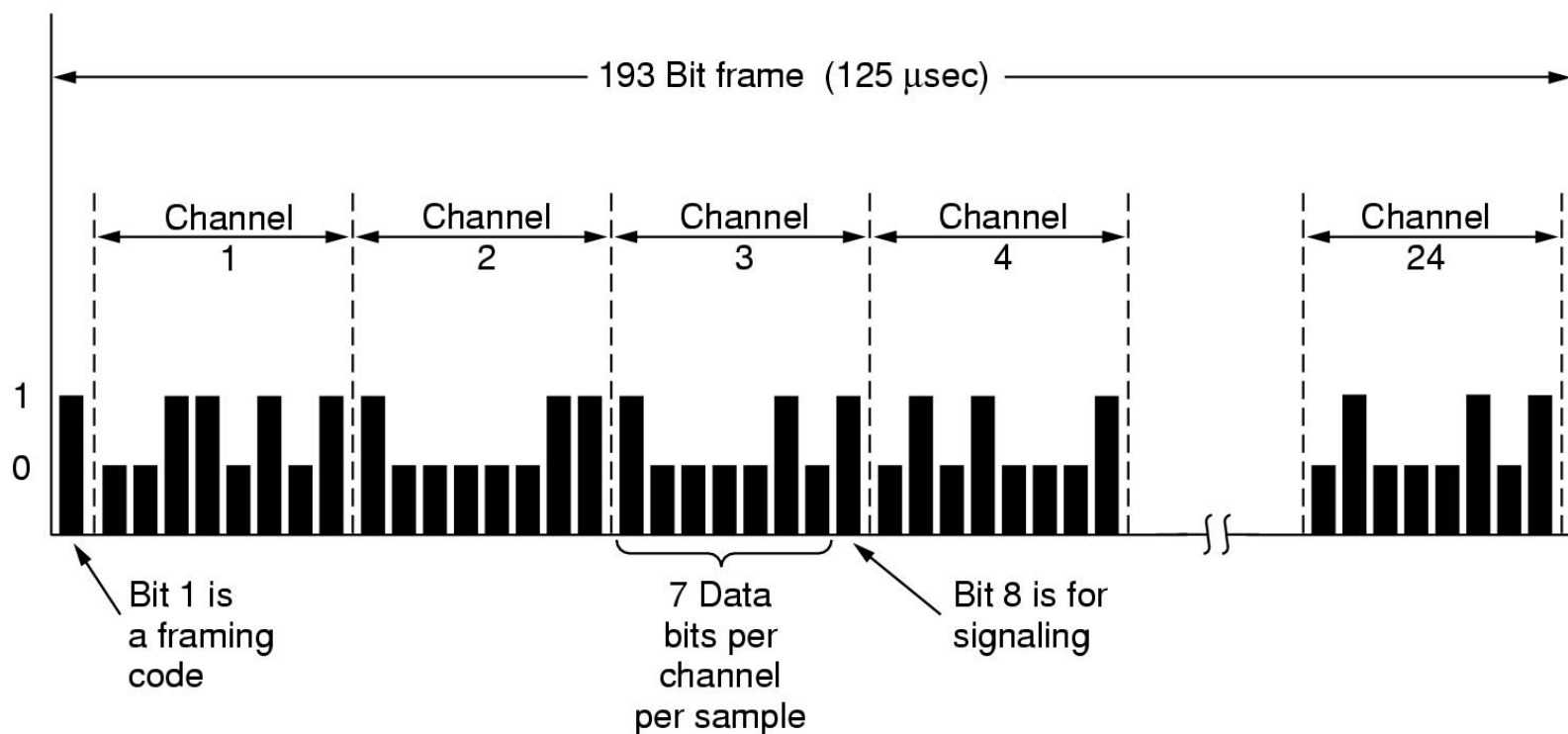
统计时分复用 STD

用户

统计时分复用

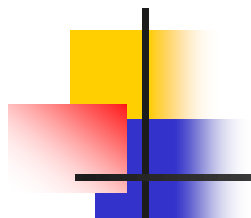


T1信道的时分复用



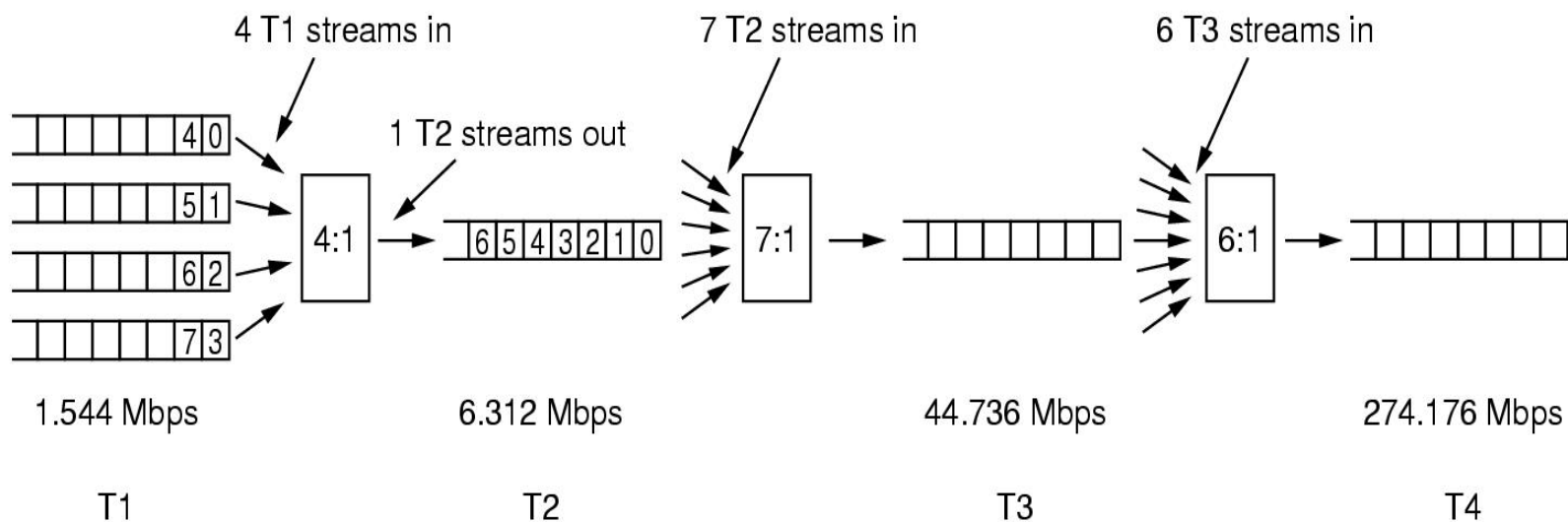
T1信道（北美、日本）：1.544Mbps

E1信道（欧洲，中国）：2.048Mbps



- **T1** supports a total of
$$[(7+1) \times 24 + 1] \text{ bit} / 125 \times 10^6$$
$$= 193 \text{ bit} / 125 \times 10^6$$
$$= \textcolor{red}{1.544M} \text{bit/sec or } 1.544 \text{Mbps}$$
- **E1**: 2.048Mbps
$$8 \times 32 / 125 \times 10^6 = \textcolor{red}{2.048M} \text{bps}$$
- 二次群速率**T2: 6.312 Mbps; E2: 8.848Mbps**
三次群速率**T3: 44.736Mbps; E3: 34.304Mbps**
四次群速率**T4: 274.176Mbps; E4: 139.264Mbps**

T系列的群路复用

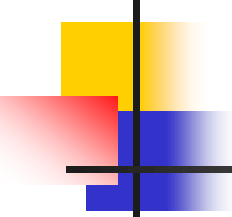


数字传输网的传输技术体制

■ 数字传输网上传输的信号一般都是经过**TDM**以后形成的数字信号的**群路信号**。

■ 有两类速率等级信号：

- **PDH**（**Plesiochronous Digital Hierarchy**，准同步数字系列）：复接成群路信号的各支路信号的时钟频率有一定的偏差，复接时通过在各支路信号中插入一定数量的脉冲来实现各支路的同步。主要有北美的**T**系列和欧洲的**E**系列这两个互不兼容的标准。
- **SDH**（**Synchronous Digital Hierarchy**，**同步数字系列**）：1985年美国贝尔通信研究所提出，称之为**SONET**（**Synchronous Optical Network**，**同步光网络**），1986年成为美国数字体系的新标准。**CCITT**于1988年接受**SONET**概念，并与**ANSI**达成协议，将**SONET**修改后重新命名为**SDH**，使之成为同时适应于光纤、微波、卫星传送的通用技术体制（**G.707**，**G.708**，**G.709**）。整个系统由一个主时钟控制，精度达 10^{-9} ，因此处于非常精准的同步状态，易于复接和分接。



SDH的特点

- **SDH**拥有全世界统一的网络节点接口（**NNI**），是真正的**数字传输体制上的国际标准**。**SDH**有一套开放的标准化**光接口**，因而使现有**PDH**中的两大标准得以兼容，可以很方便地在光路上实现互通，使信号传输、复用和交换过程得到**简化**，大大降低了联网成本。
- **SDH**拥有一套标准化的信息结构等级，称为同步传送模块（**STM**），采用同步复用方式，利用软件就可以从高速复用信号中一次分出或插入低速支路信号，使交叉连接分支得以方便实现。
- **SDH**拥有丰富的开销比特，约占信号的**5%**，用于网络的**OAM（Operation、Administration、Maintenance）**，从而大大提高了传输的质量和应急能力。另外其网络结构适应性强，各种数字传输体系均可进入其**帧结构**。

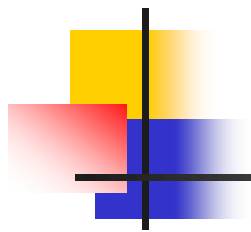
SONET/SDH的速率及复用

■ SONET的基本帧:

- 是每**125 μ s**内传输**810**个字节（用**90列**×**9行**描述），因此数据速率 = **8 bit** × **810 Byte** × **8000 Hz** = **51.84 Mbps**，这是SONET最基本的信道，称作同步传输信号**STS-1**（**Synchronous Transport Signal-1**）。
- 多条**STS-1**支流的复用可构成其它的SONET干线，如**3条STS-1**支流被合成为**1条** **3 × 51.84 Mbps = 155.52 Mbps**的**STS-3**流。对应于**STS- n** 的光纤线路被称作**OC- n** （**Optical Carrier Level**）。

■ SDH的基本帧:

- 从**OC-3 155.52 Mbps**开始，称作同步传输模块**STM-1**（**Synchronous Transport Module-1**）。
- **OC- n** 表示由 **n** 条单独的**OC-1**线路组成。



段开销

线路开销

路径开销

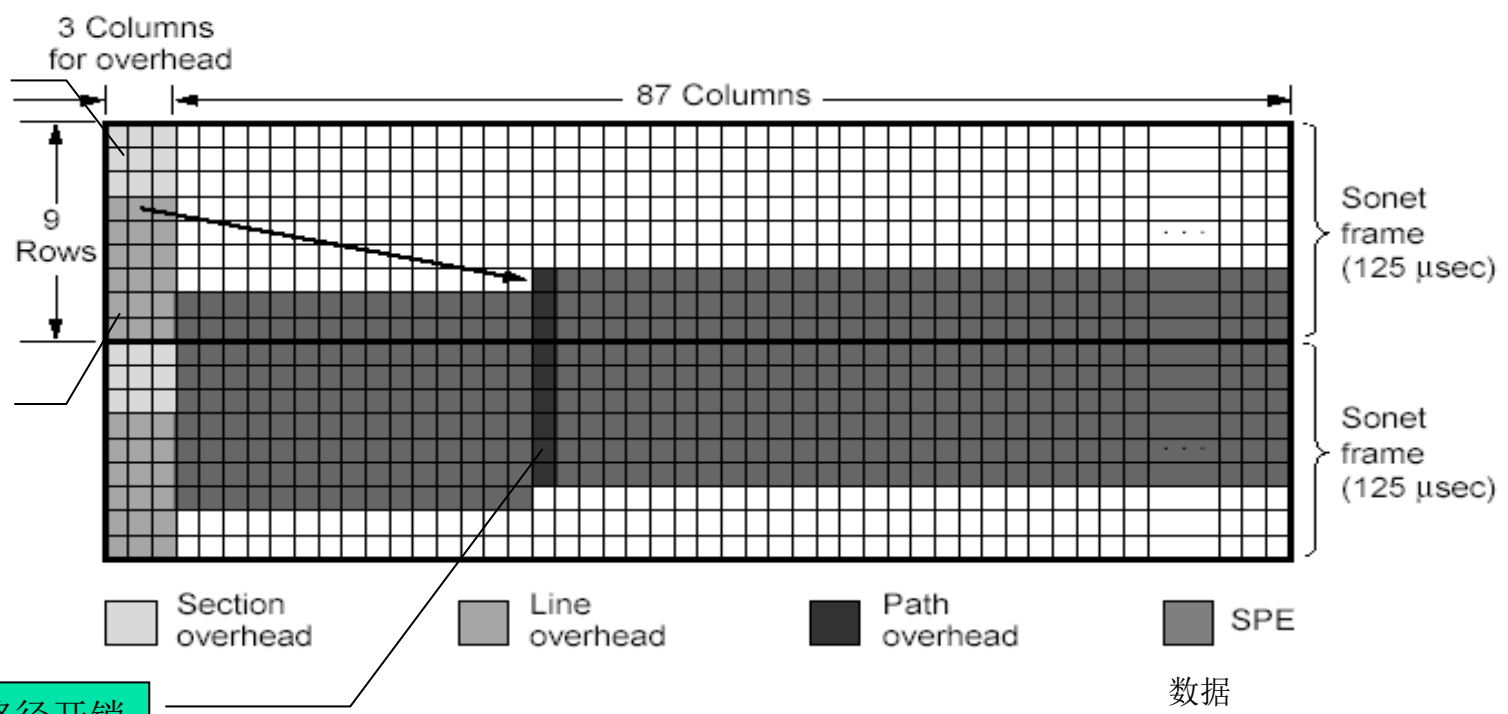
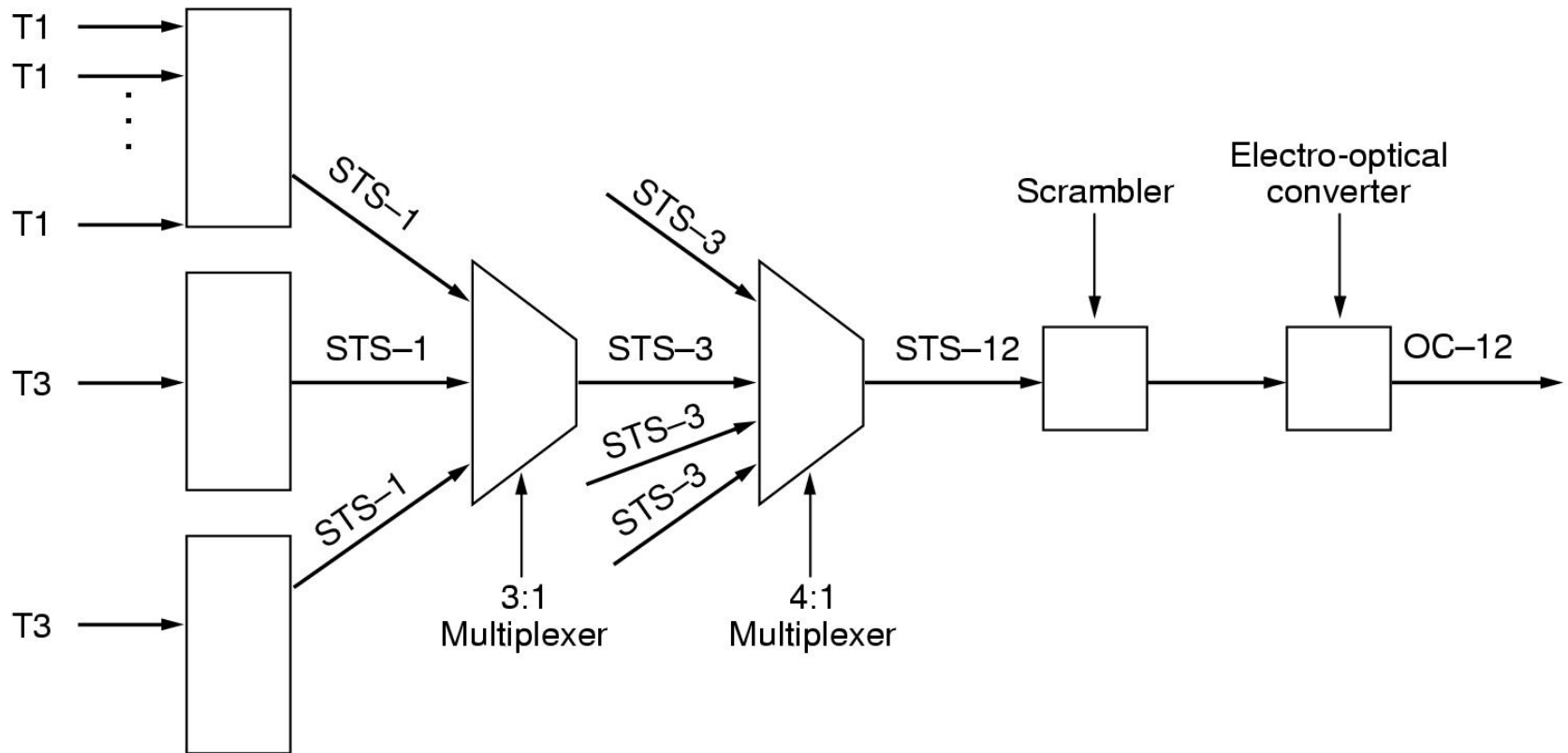
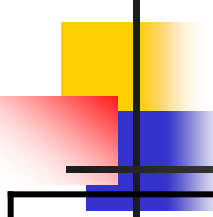


Fig. 2-30. Two back-to-back SONET frames.

SONET/SDH

(Synchronous Optical NETWORK/
Synchronous Digital Hierarchy)

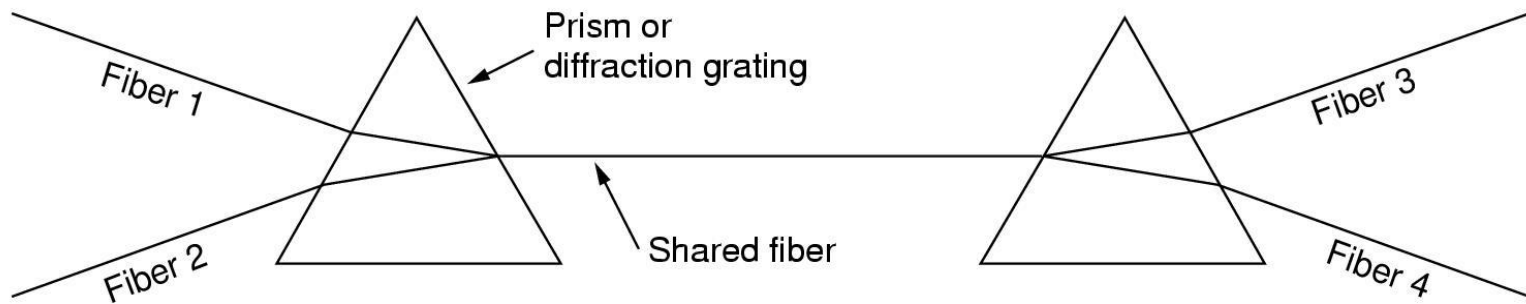
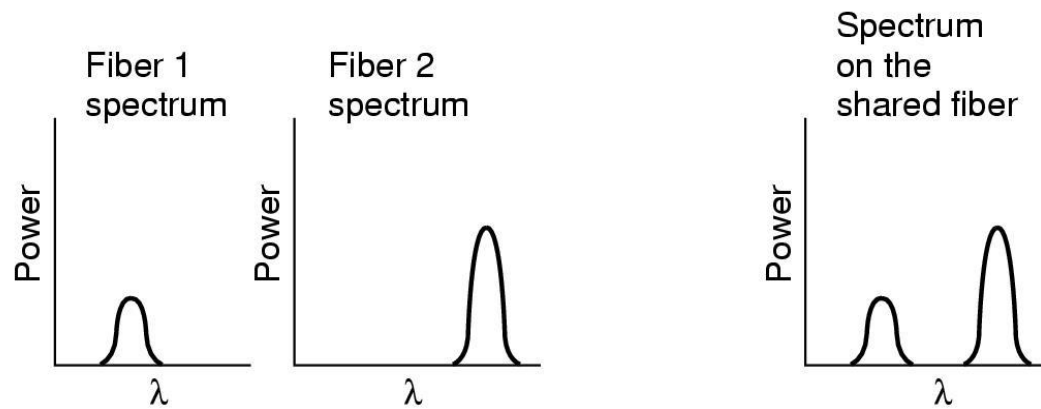


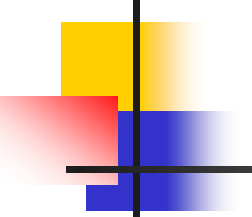


SONET/SDH的速率等级

SDH等级		SONET等级	标准速率 (Mbps)
		OC-1/STS-1 (480 CH)	51.8400
155 M	STM-1 (1920 CH)	OC-3/STS-3 (1440 CH)	155.520
	STM-3	OC-9/STS-9	466.560
622 M	STM-4 (7696 CH)	OC-12/STS-12 (8046 CH)	622.080
	STM-6	OC-18/STS-18	933.120
	STM-8	OC-24/STS-24	1244.160
	STM-12	OC-36/STS-36	1866.240
2.5 G	STM-16 (30720 CH)	OC-48/STS-48 (32356 CH)	2488.320
10 G	STM-64 (122880 CH)	OC-192/STS-192 (129024 CH)	9953.280

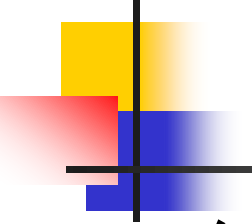
波分复用WDM





码分复用 CDM

- 码分复用是扩频通信的一种形式，常用的名词是码分多址 CDMA (Code Division Multiple Access)。
- 每一个比特时间划分为 m 个短的间隔，称为码片(chip)。
- 各用户使用经过特殊挑选的彼此正交的码型，因此不会造成干扰。
- 例如，采用双极符号把码片序列写成-1, +1的序列组合，设 $m=8$ ，A站分配得到的码片序列为
(-1 -1 -1 +1 +1 -1 +1 +1)，若A站发送的是该序列，则表示发出的是“1”比特，若发送的是
(+1 +1 +1 -1 -1 +1 -1 -1)，则表示发送的是“0”。



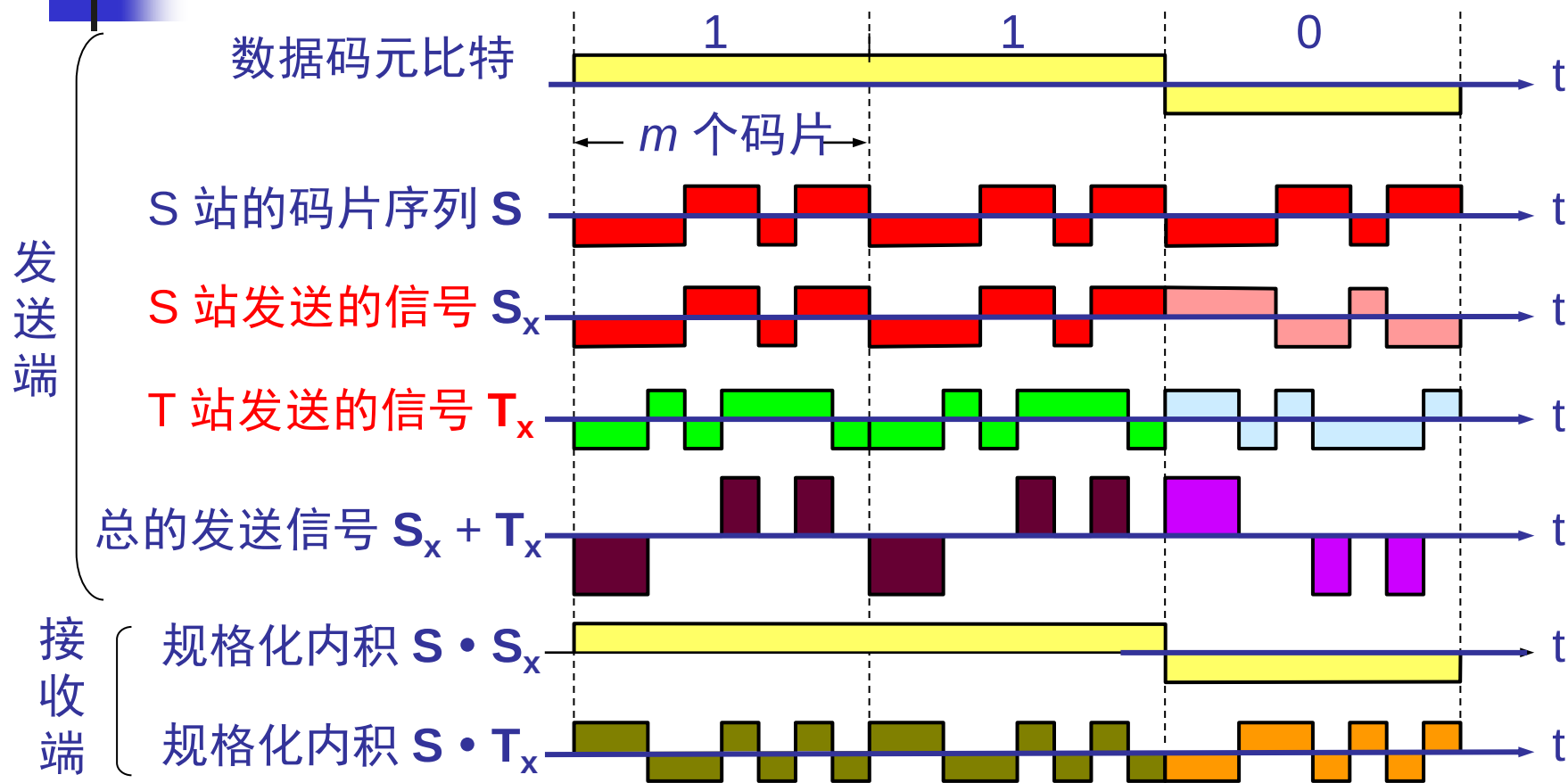
码片序列的正交关系

- 令向量 **S** 表示站 **S** 的码片向量，令 **T** 表示其他任何站的码片向量。
- 两个不同站的码片序列正交，就是向量 **S** 和 **T** 的规格化内积(inner product)都是 0:

$$\mathbf{S} \bullet \mathbf{T} \equiv \frac{1}{m} \sum_{i=1}^m S_i T_i = 0 \quad (2-5)$$

$$\mathbf{S} \bullet \mathbf{S} \equiv \frac{1}{m} \sum_{i=1}^m S_i S_i = 1$$

CDMA 的工作原理





Chapter 3 物理层

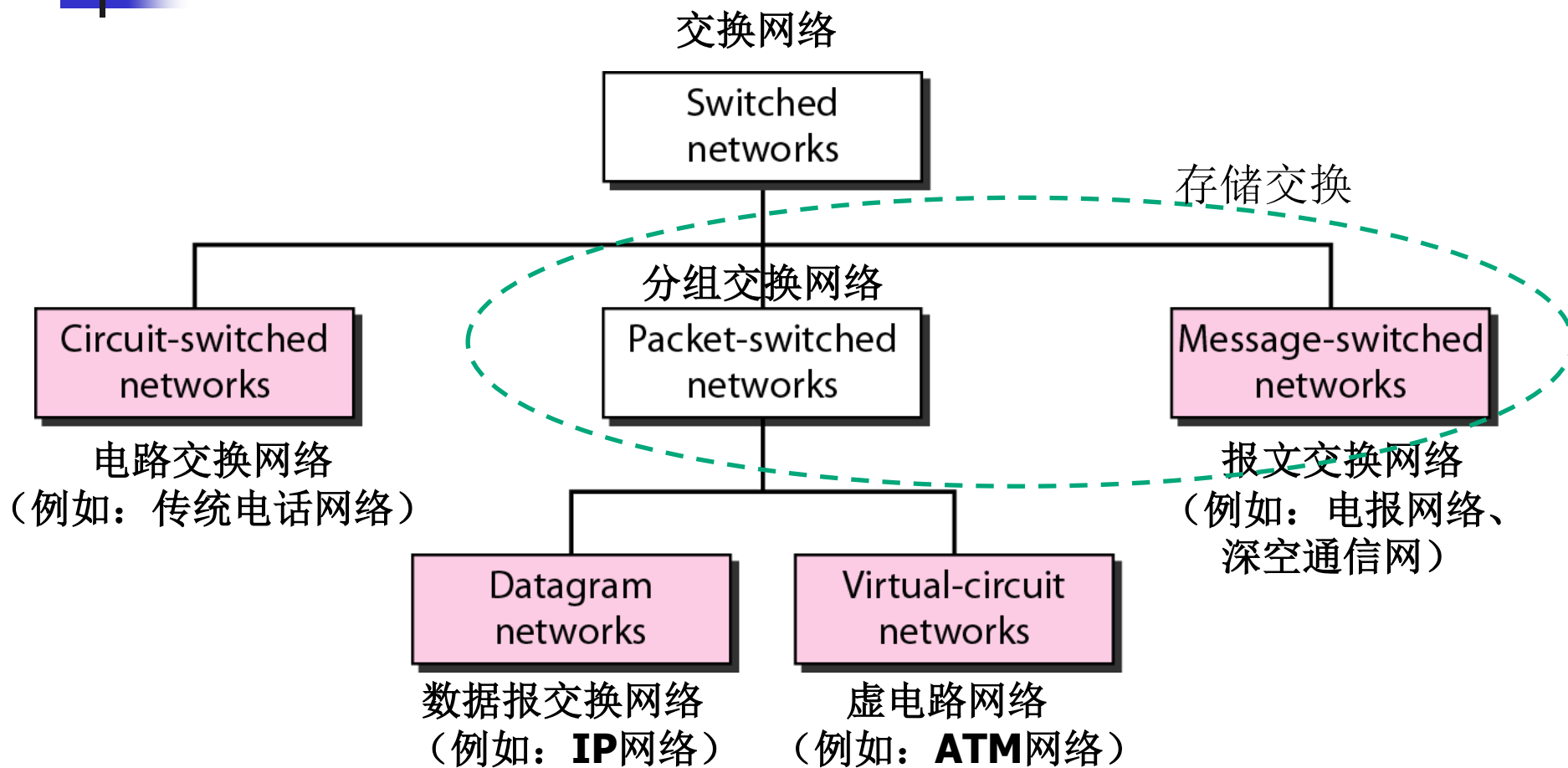
- **3.1**数据通信的理论基础
- **3.2**传输介质
- **3.3**数字调制与多路复用
- **3.4**交换技术

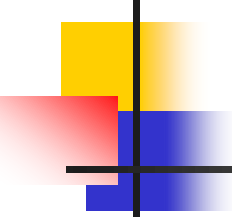


3.4 交换技术

- 通信网包括用户终端、传输链路和交换系统这三个基本元素，交换系统是通信网进行信息交换的控制枢纽。
- 交换类型
 - 电路交换（circuit switching）（TDM或FDM）
 - 存储交换（统计时分复用）
 - 报文交换（message switching）
 - 分组交换（packet switching）

交换类型





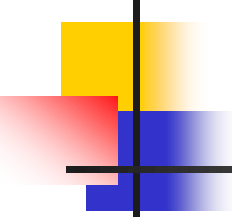
电路交换(circuit switching)

- 电路交换方式在通信对端之间建立**物理连接（预留物理信道）**，该通路在通信期间一直维持着，并且为该通信双方专用，直到通信结束。
- 连接过程中，连接请求从发送端传向接收端，并被接收端确认，一旦连接建立，数据传输的唯一延时是电磁波的传播时间。
- 电路交换的优点是传输可靠，迅速，不丢失而又保序。
- 计费方式按预订带宽、距离和时间计算。



存储交换方式

- 报文交换(**message switching**)
 - 数据传输以**报文的整体**为单位，即端点系统一次性发送数据块，长度不限且可变。
- 分组交换(**packet switching**)
 - 将报文**划分成**若干个可以存放在内存中的**分组**来进行传送。
 - 数据报交换：无连接的分组交换
 - 虚电路交换：面向连接的分组交换



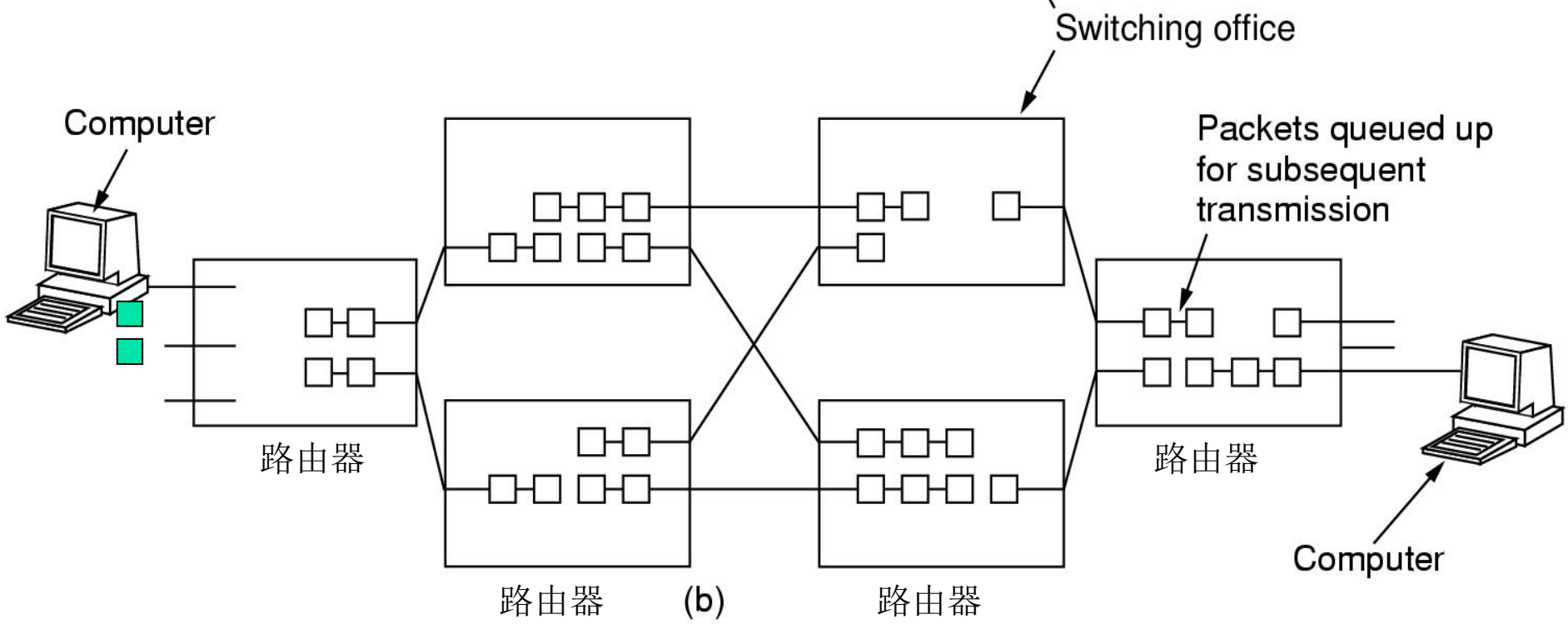
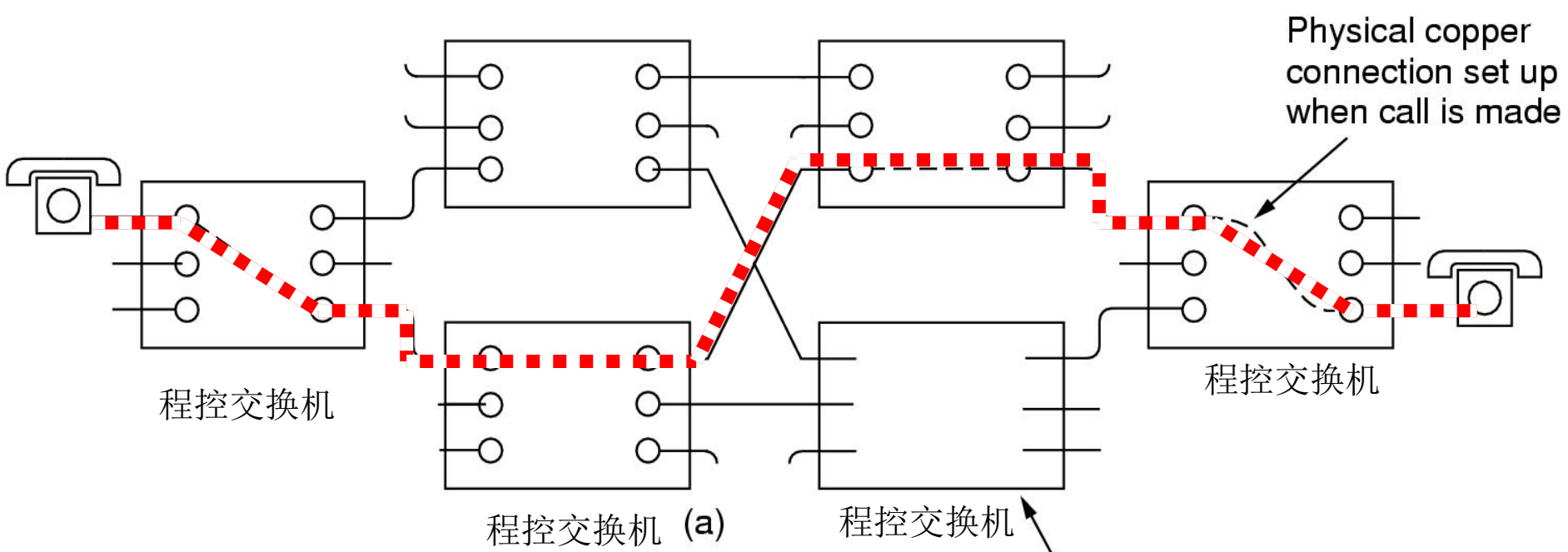
分组交换

■ 数据报 (datagram)

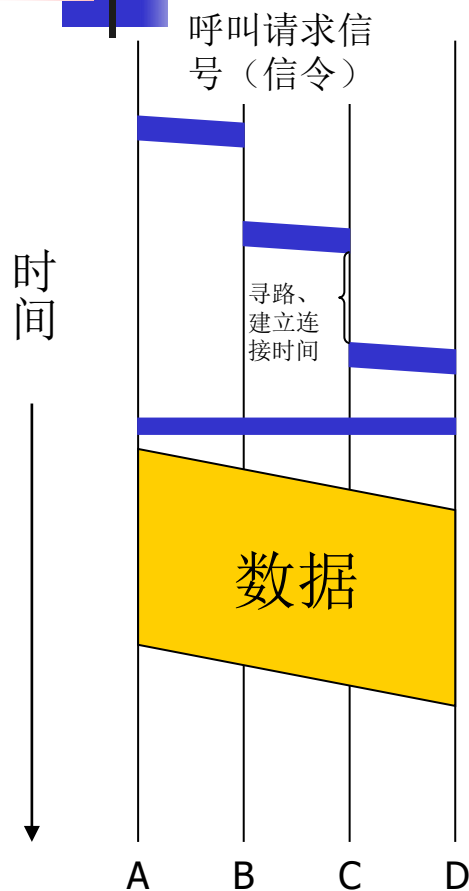
- 在数据报交换中，每个分组的传送被单独处理，每个**分组头中都包含目的节点的地址**，一个节点收到数据报后，根据报头中的地址信息和节点所存储的路由信息，找出一个合适的出口，把数据报发送到下一个节点。同一报文中的数据报可以走不同的路径，也可以按不同的顺序到达。

■ 虚电路(virtual circuit)

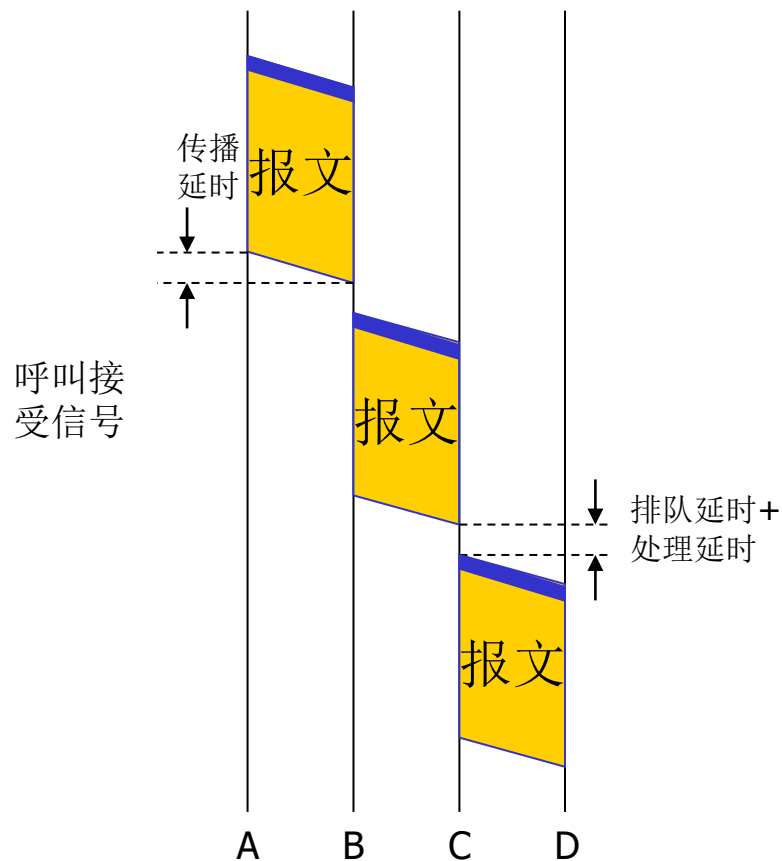
- 在虚电路交换中，首先在源节点和目的节点间建立一条**逻辑通路**（既虚电路，在沿路的链路上分配了一个虚通道号，相应的节点上占用了一条表项），然后进行通信，最后拆除虚电路。



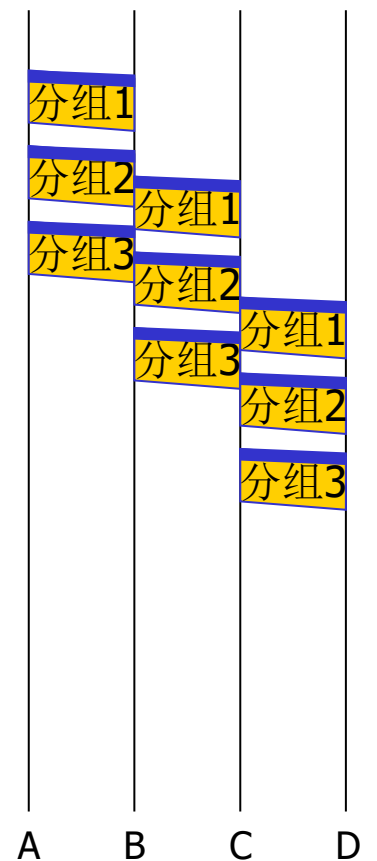
三种交换技术的时序图



■ 电路交换



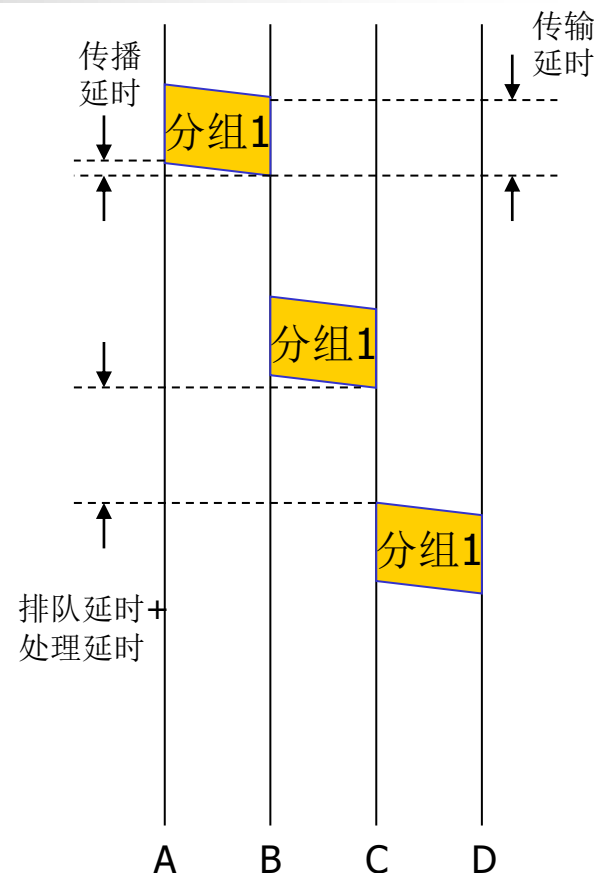
■ 报文交换



■ 分组交换

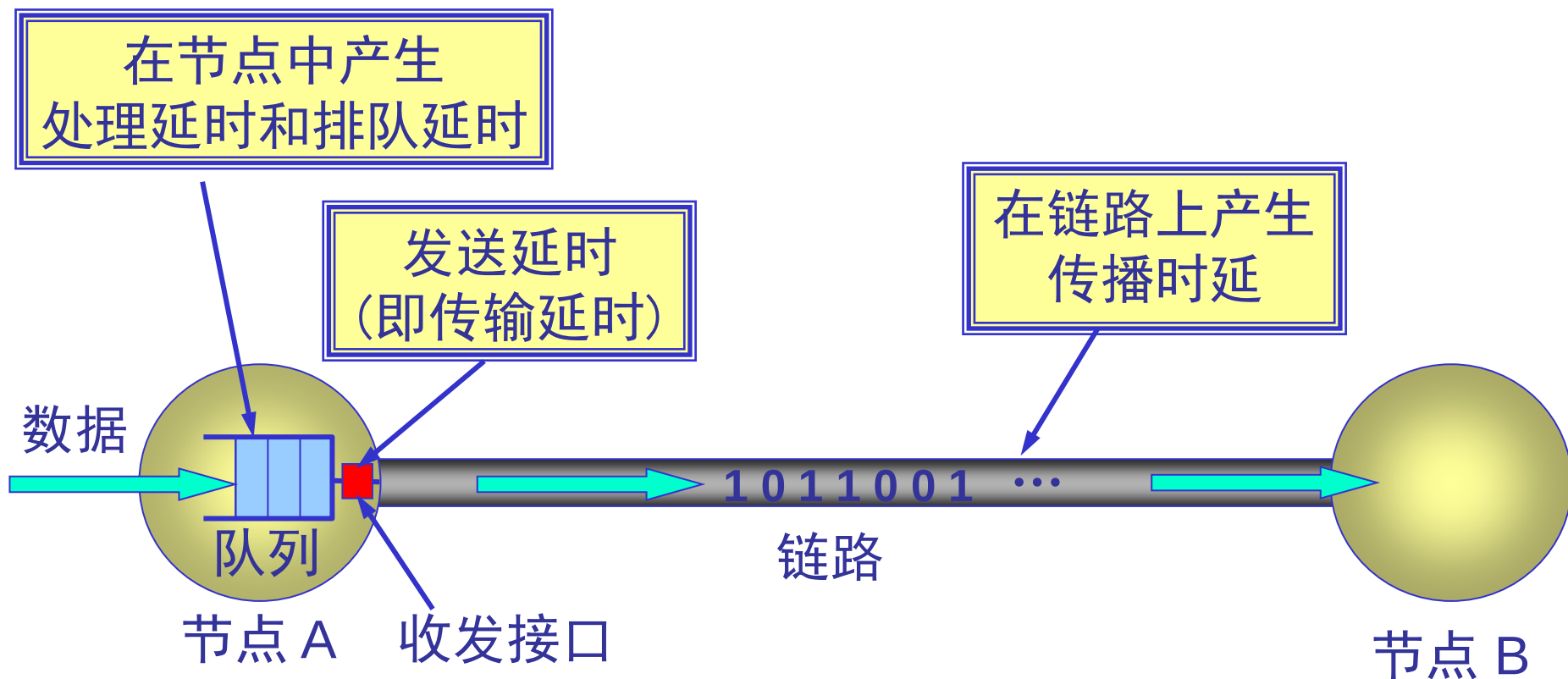
分组交换中的延时分析

- 传播延时 (**propagation delay**)
 - 数据某比特从A节点到B节点传播所需的时间，取决于物理媒体**传播速度** ($2\sim 3 \times 10^8 \text{m/s}$) 和**AB两点间距离**。
- 传输延时 (**transmission delay**)
 - 节点发送 (或接受) 整个分组所需时间，用L比特表示**分组长度**，用Rb/s表示**链路速率**，则L/R为传输延时。
- 处理延时 (**processing delay**)
 - 检查分组头部，寻路，差错校验等，与**CPU速度**和处理**算法**有关。
- 排队延时 (**queue delay**)
 - 在节点上等待处理所需时间，与**网络拥塞状况**有关

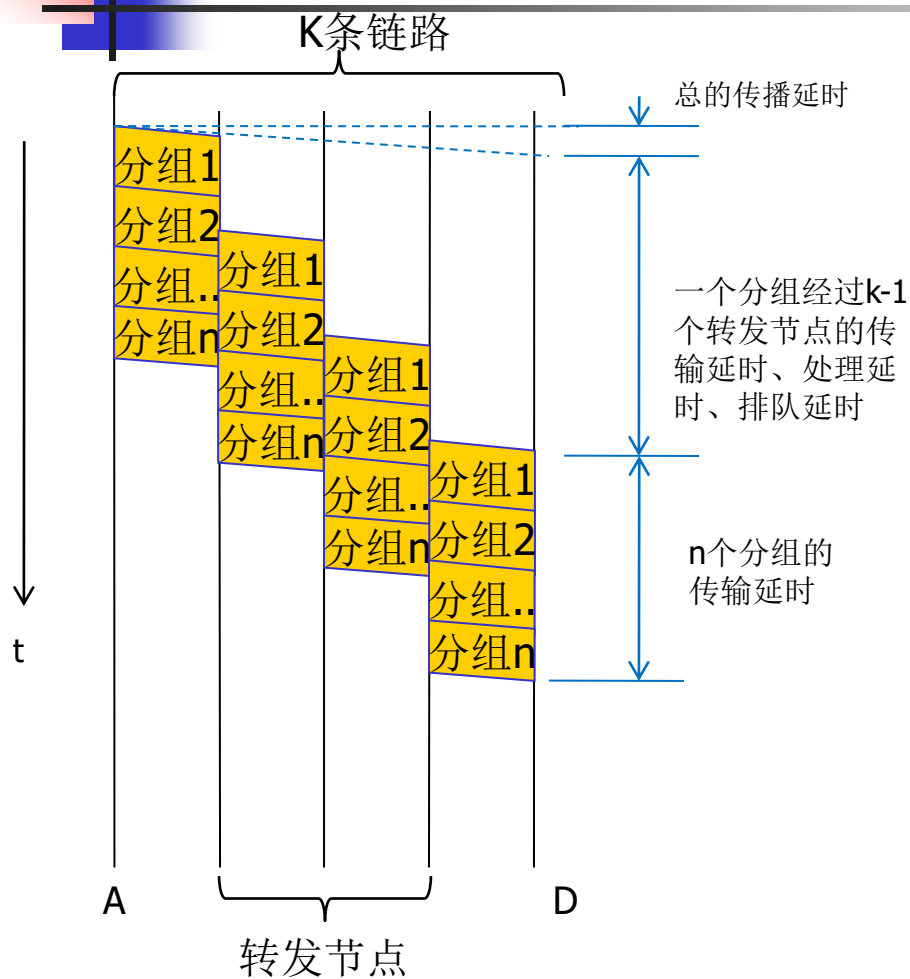


■ 分组交换

延时产生的地方



分组交换的总延时



■ 一个分组在路径上的总延时

- $D = \text{总}d_{\text{prop}} + \text{转发节点数} * (d_{\text{trans}} + d_{\text{pro}} + d_{\text{queue}}) + d_{\text{trans}}$

■ n个分组在路径上的总延时

- $D = \text{总}d_{\text{prop}} + \text{转发节点数} * (d_{\text{trans}} + d_{\text{pro}} + d_{\text{queue}}) + n * d_{\text{trans}}$



电路交换与分组交换的比较

- 电路交换预先静态地保留所要带宽；而分组交换却是根据需要动态地获得和释放带宽。
- 在电路交换中，已分配给线路上未用的带宽只能浪费掉；而分组交换中，未用的带宽可以被别的传输所利用。
- 电路交换对数据传输是完全透明的，而分组交换则要利用分组所携带的参数进行路由转发。
- 分组交换是存储转发的，会增加传输延时；电路交换则是连续的通过物理线路传输。
- 分组交换中，分组到达目的地可能不按原顺序；在电路交换中，不会发生乱序现象。
- 分组交换可以按传输的字节或连接时间计费；电路交换按时间、距离计费。



本章小结

- 了解物理层的基本功能，掌握调制/解调、编码/解码、复用/解复用、电路交换、分组交换等基本概念。
- 习题：
 - 第5版，第二章，3，4，9，26，39，46
 - 第4版，第二章，3，4，9，29，43，53



Chapter 4 数据链路层

- 4.1 数据链路层的基本概念与功能
- 4.2 差错检测与纠正
- 4.3 基本数据链路协议
- 4.4 滑动窗口（Slide Windows）协议
- 4.5 面向位的协议HDLC
- 4.6 面向字节的数据链路层协议-PPP



Chapter 4 数据链路层

- 4.1 数据链路层的基本概念与功能
- 4.2 差错检测与纠正
- 4.3 基本数据链路协议
- 4.4 滑动窗口（Slide Windows）协议
- 4.5 面向位的协议HDLC
- 4.6 面向字节的数据链路层协议-PPP



4.1 数据链路层的基本概念

■ **链路 (link)**: 是连接相邻节点的通信信道, 中间没有任何其他的交换节点。

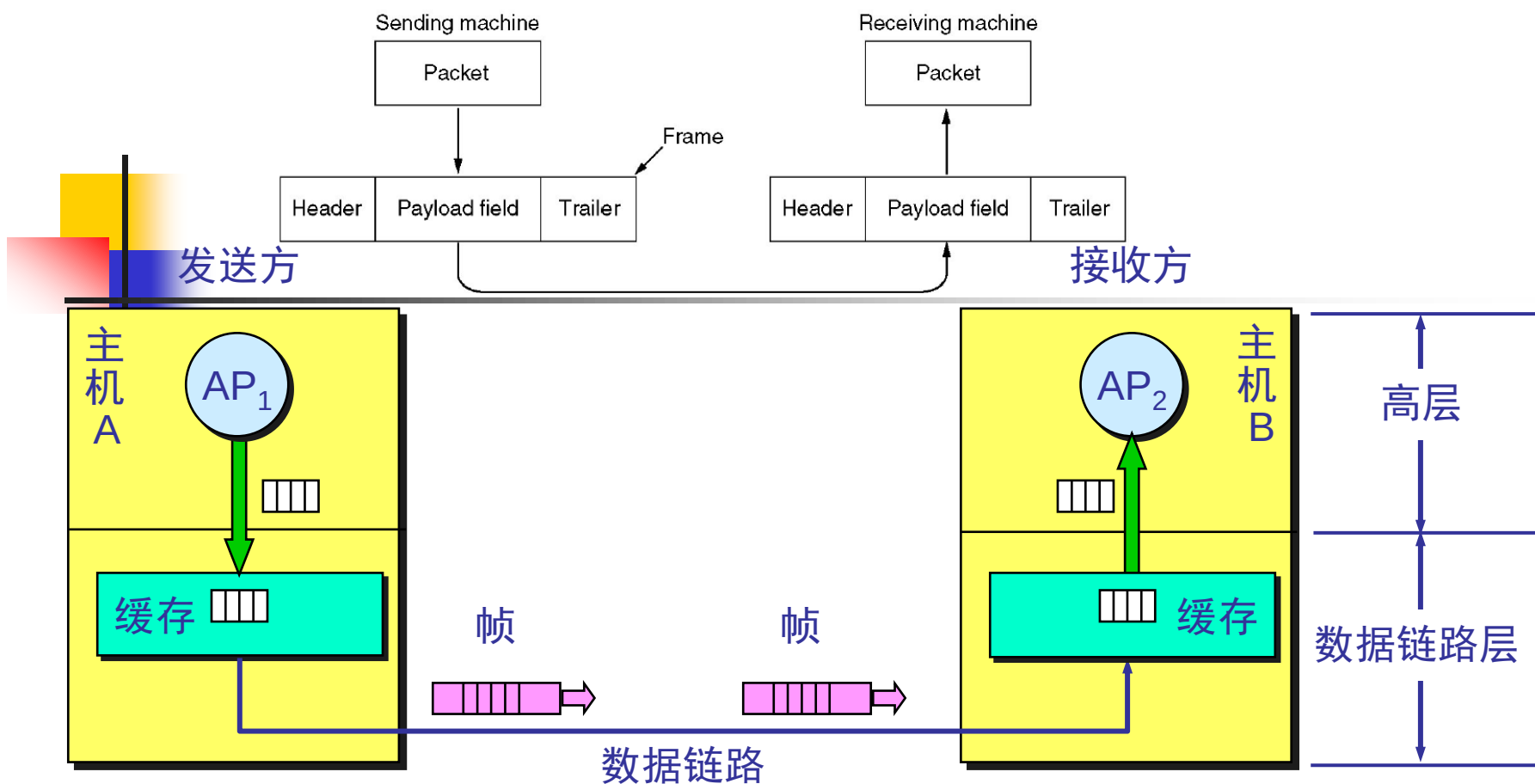
- **点对点链路**: 链路的两端都只有一个节点, 一个节点发送, 另一个节点接收, 带宽独占。
- **点对多点链路** (广播链路, 共享链路): 链路上有多个节点, 带宽共享, 需要媒体访问控制 (MAC) 机制来控制多个用户对信道的访问。

■ 除了物理线路外, 还必须有**通信协议**来控制数据的传输。这些用**软件或硬件**实现的协议构成了网络的数据链路层。

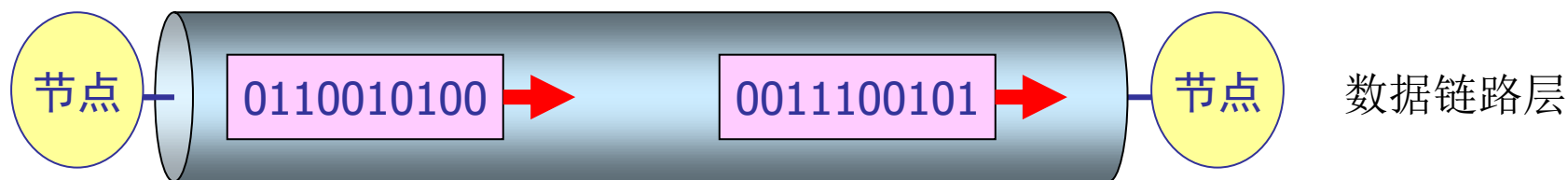
- 现在最常用的方法是使用适配器 (即网卡) 来实现这些协议的硬件和软件。
- 一般的适配器都包括了数据链路层和物理层这两层的功能

■ 通信方式: **单工**通信、**半双工**通信和**全双工**通信

- **单工通信**: 是指消息只能单方向传输的工作方式。
- **半双工通信**: 是指数据可以沿两个方向传送, 但同一时刻一个信道只允许单方向传送双工。
- **全双工 (Full Duplex)** 是指在通信的任意时刻, 两个节点间可以同时双向传输信号。**TDD (Time Division Duplex, 时分双工)** 是移动通信中上下行在同一频段上按照时间分配交叉进行传输, **FDD (Frequency Division Duplex, 频分双工)** 是指上下行在不同频段同时进行传输。

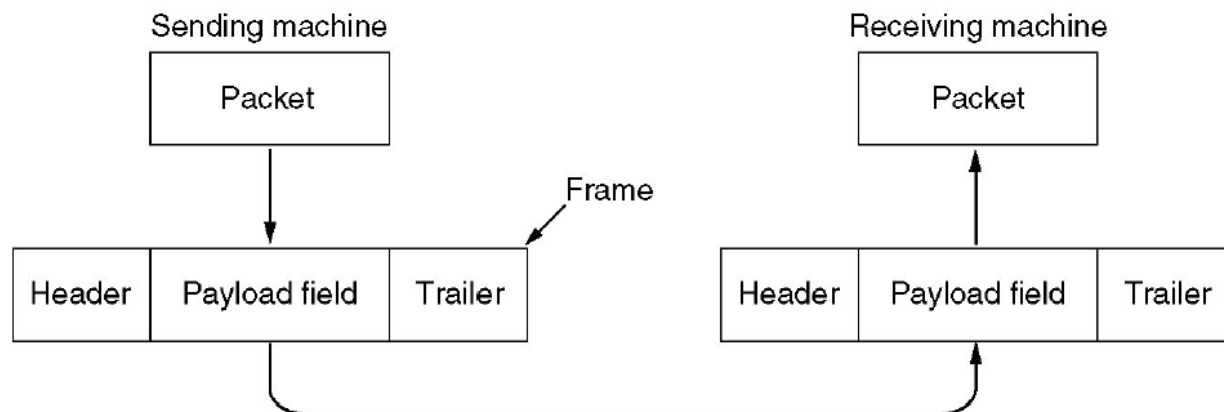


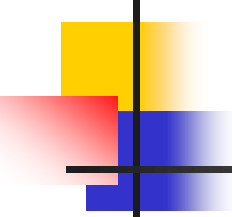
- 在两个对等的的数据链路层之间画出一个数字管道，而在这条数字管道上传输的数据单位是帧。



数据链路层的功能

- 向网络层提供良好的服务接口
- 将物理层的比特流编成帧
- 差错控制和可靠性
- 流量控制
- 介质访问控制(下一章)



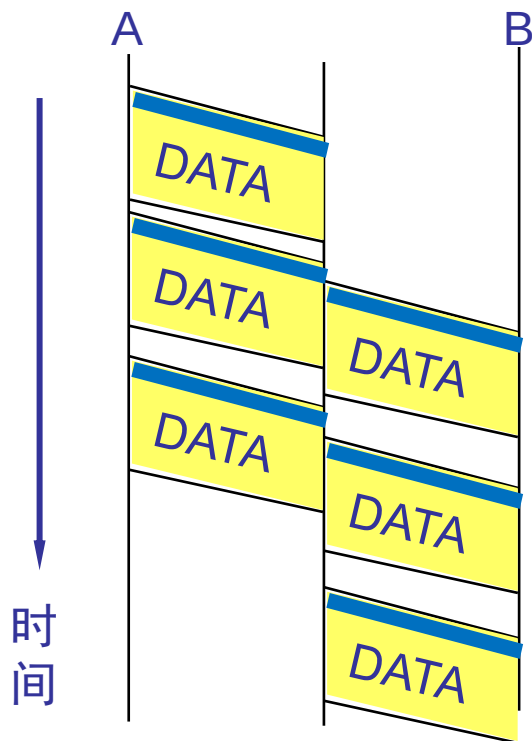


4.1.1 为网络层提供服务

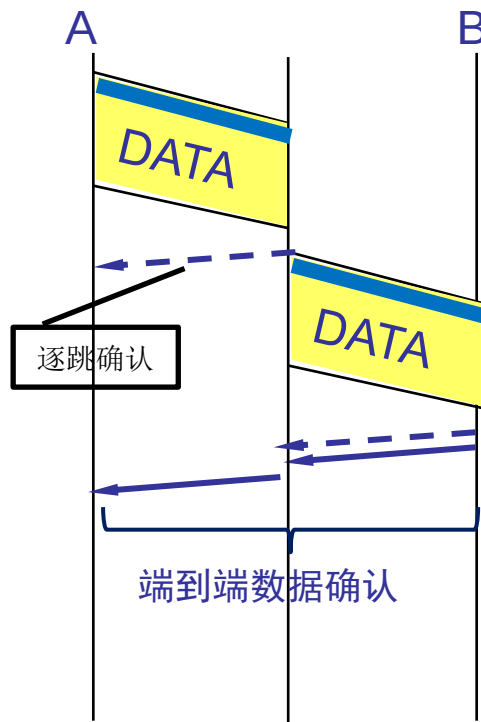
- 无确认、无连接的服务（有线局域网）
 - 源端可以不需要建立连接就向目的端发送独立的数据帧，而目的端也不需要收到帧进行确认。
- 有确认、无连接的服务（无线局域网）
 - 源端可以不需要建立连接就向目的端发送独立的数据帧，但目的端需要对收到的帧进行确认。
- 面向连接的服务（无线城域网）
 - 源端与目的端在通信前要先建立连接，然后在此连接上互相传输数据帧，每一个帧都被编号，数据链路层保证传送的帧被对方收到，且只收到一次，双方通信完毕后拆除连接。

三种服务的对比（无差错传输时）

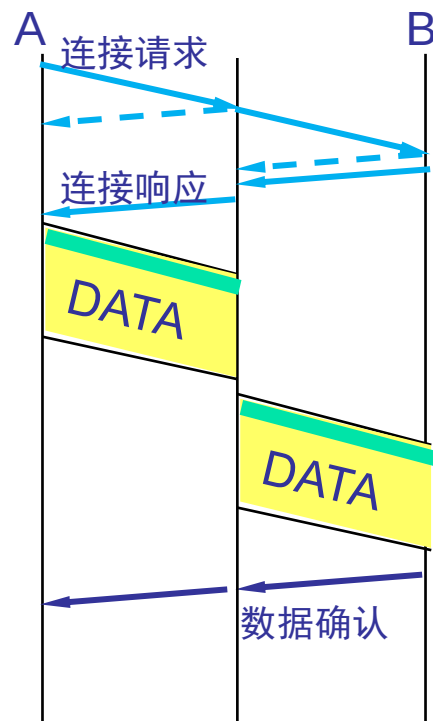
无连接无确认



无连接有确认



面向连接



识别或表示一个帧的起始和结尾

4.1.2 成帧(framing)

- 为什么要成帧？
 - 物理层传输比特流时可能发生错误，如何处理？→ 在数据链路层重发！
 - 数据链路层以帧为单位进行发送
 - 在共享链路上，发送方以帧为单位来竞争对共享链路的访问
- 将比特流划分成帧的方法
 - 字符计数法
 - 含字节填充的分界符法
 - 含位填充的分界符法
 - 物理层编码违例法

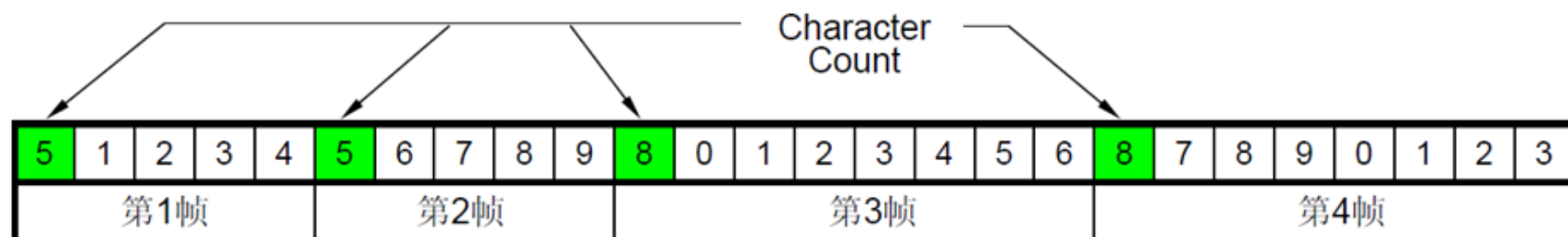
标识每一帧的起始和结束位置

识别或表示一个帧的起始和结尾

4.1.2 成帧(framing)

■ 1. 字符计数

- 每一个帧的头上描述帧的长度



缺点：帧头出错不光影响本数据帧，还影响后续的帧。



识别或表示一个帧的起始和结尾

4.1.2 成帧(framing)

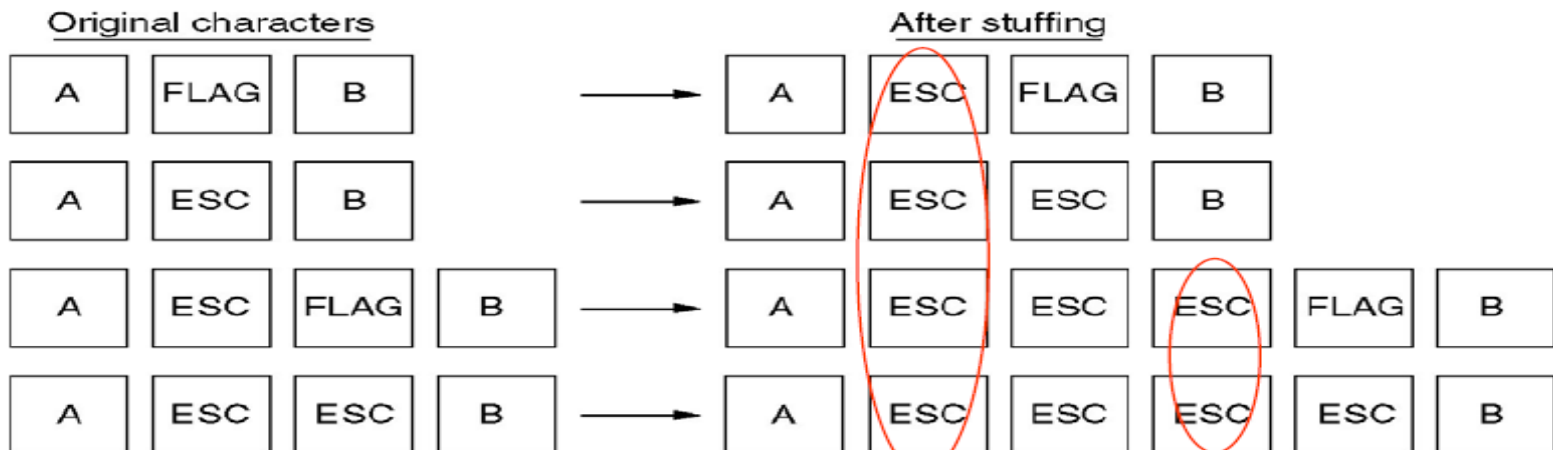
■ 2. 字节填充的分界符法

- 在帧的起始和结尾用特殊的字符标志，称为标志字节（Flag）

若发送方数据链路层发送的数据中有标志字节，则在它前面插入转义字节（**ESC**），接收方的数据链路层将数据递交给网络层之前删除转义字节



(a)



识别或表示一个帧的起始和结尾

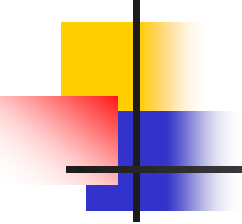
4.1.2 成帧(framing)

■ 3. 位填充的分界符法

- 用特殊的位序列表示帧的起始和结尾

例如，用**01111110**表示帧的起始和结尾

为了避免起始、结尾标志与帧中的数据混淆，发送端的数据链路层在检测到数据比特流中有连续**5个1**出现时，就在其后加一个**0**，接收端的数据链路层在将**5个1**后面的**0**取走还原。



(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits



(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

模式**01111110**不会在普通正常数据中出现

识别或表示一个帧的起始和结尾

4.1.2 成帧(framing)

■ 4.物理层编码

- 在物理层用1.5或2个物理位表示一个数据位来表示帧的起始与结尾。

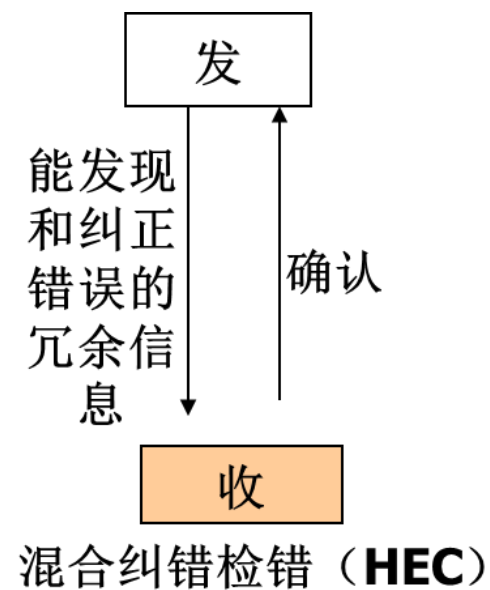
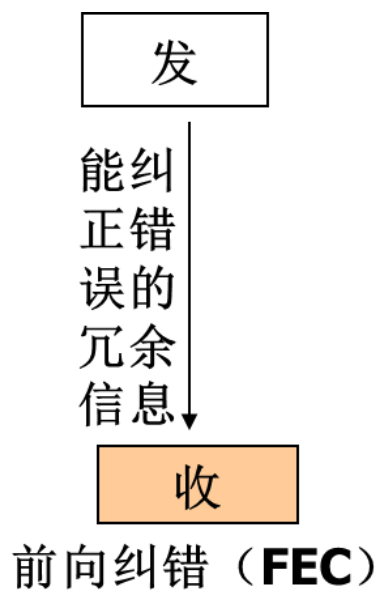
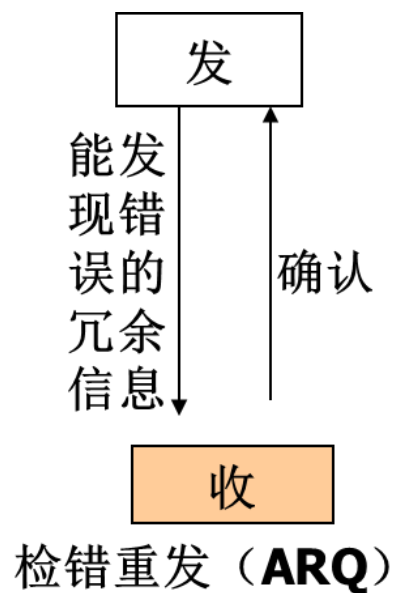
例如在物理层编码中，用「」表示1，用「」表示0
则可用 __，— 作为起始、结尾标志。



4.1.3 差错控制

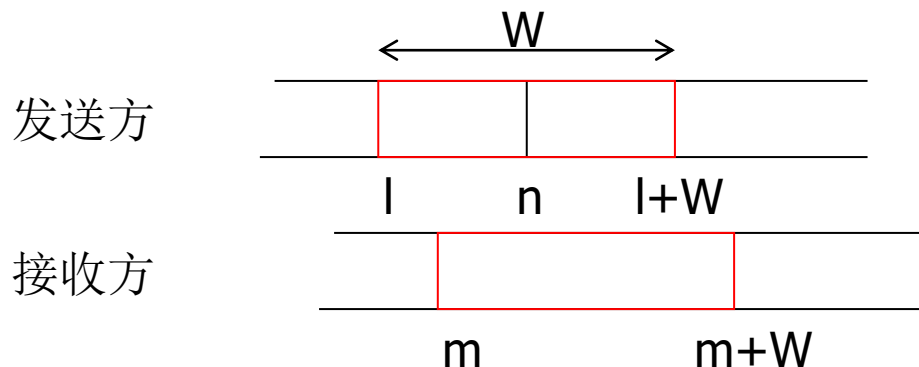
- 在**有确认**的服务中，接收方需要对收到的数据帧进行确认，通常以返回**特别控制帧**来告诉发送方收到的数据有没有出错；
- 对于端到端的确认还可以通过返回的数据帧中的某些位来**捎带**地进行**确认**。
- 如果发送的数据丢失，那么接收端是不可能进行确认的，怎么办呢？
 - 在发送端引入计数器，进行**超时重发**。为了避免相同的帧收到多次，需要对帧进行编号。

4.1.3 差错控制



4.1.4 流量控制

- 当发送端的发送速度大于接收端的接收速度，或发送端所在的网络传输速率大于接收端所在的网络传输速率时，就需要流量控制。
- 可以在数据链路层之上的其他各层进行实现，例如传输层
- 窗口机制
 - 在任意时刻，发送方发出 n 帧，其中有 l 帧已得到确认。而同一时刻，接收方收到并应答了 m 帧，显然 $l \leq m \leq n$ ，等待确认的帧的数目不允许超过 W ，即 $n - l \leq W$ ，该 W 就是窗口。





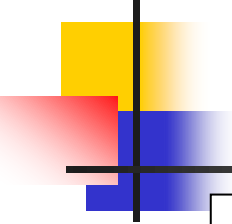
Chapter 4 数据链路层

- 4.1 数据链路层的基本概念与功能
- 4.2 差错检测与纠正
- 4.3 基本数据链路协议
- 4.4 滑动窗口（Slide Windows）协议
- 4.5 面向位的协议HDLC
- 4.6 面向字节的数据链路层协议-PPP



4.2 差错检测与纠正

- 信号在任何信道上传输都存在着传输差错，这些差错由多种物理现象引起，解决差错问题的方法有**两种**：
- 一种是在要发送的数据中加入一定的冗余位，使接收方能知道数据是否出错，但不知道是哪里出错，这种编码方法叫**差错检测码**，或简称**检错码**。
- 另一种是在要发送的数据中加入足够多的冗余位，使接收方能纠正出错的位，这种编码方法叫**差错纠正码**，或简称**纠错码**。



什么时候用检错码？什么时候用纠错码？

噪声小的可靠链路，例如电缆或者光纤，使用检错码+确认重传

噪声大的不可靠链路，例如无线链路，或者单工链路使用纠错码

信号在任何信道上传输都存在着传输差错，这些差错由多种物理现象引起，解决差错问题首先要能检测到错误，在某些情况下还要求能够纠正错误，对应着**检错码**和**纠错码**，还有将两者功能结合起来的码，称为**纠错删码**



4.2.1 纠错码

- 定义：一帧由 m 个数据位（即报文）和 r 个冗余位（即校验位）组成，设总长度为 n （ $n=m+r$ ），此长度为 n 的单元常称为 n 位码字。
 - 定义：
 - 两个码字不同的位的数目称为**海明距离(Hamming Distance)**
- 例如，10001001 与10110001它们的海明距离为3。
- 对于 n 位码字的集合，只有 2^m 个码字是有效的，在任意两个有效码字间找出具有最小海明距离的两个码字，该海明距离便定义为全部码字的海明距离。

一种编码的检错和纠错能力取决于编码后码字的海明距离的大小。

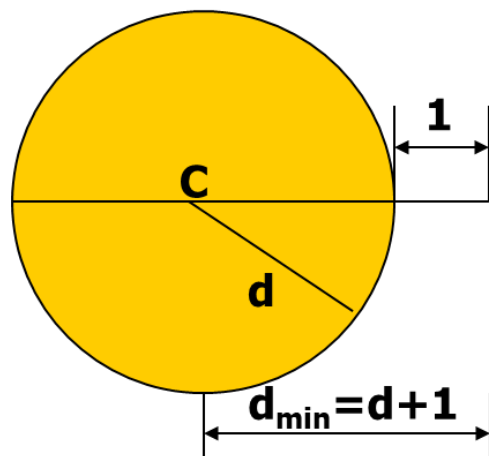
■ 为了检测出 d 个比特的错，需使用距离为 $d+1$ 的编码。

例如：数据后加奇偶校验位，编码后的海明距离为2，能检测1比特错。

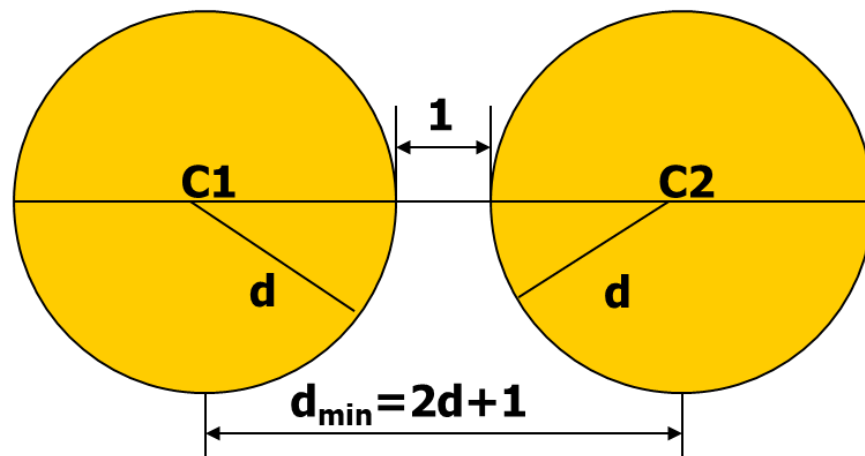
■ 为了纠正 d 个比特的错，必须用距离为 $2d+1$ 的编码。

例如有4个有效码字：它们是0000000000，0000011111，1111100000，1111111111，海明距离为5，能纠正2比特错。

检错



纠错

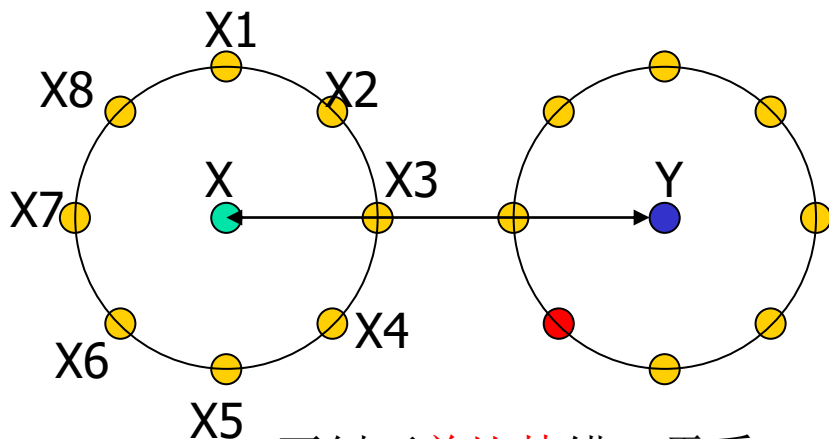
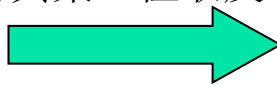


纠正单比特错的校验位下界

- 设计一种编码，它有 m 个信息位和 r 个校验位，当 r 满足什么条件时，能纠正所有单比特错？
- 码字长度 $n=m+r$ ，总共有 2^n 个种组合，其中有效码字数为 2^m

每一个有效码字 X ，
例如 $X=10101010$

对其某一位取反



要纠正单比特错，需采用距离至少为3的编码

有 n 个与其距离为1的无效码字

$X1=00101010$, $X2=11101010$

$X3=10001010$, $X4=10111010$

$X5=10100010$, $X6=10101110$

$X7=10101000$, $X8=10101011$

$$(n+1) 2^m \leq 2^n, \quad n=m+r \text{ 代入} \\ 2^r \geq n+1$$

纠正单比特误码的校验位下界



海明编码

- 海明在1950年提出一种编码来纠正单比特错的编码。该编码是将码字内的位从左到右依次编号，编号为2的幂的位是校验位（如第1，2，4，8...），其余为信息位。
- 每个校验位的取值应使得包括自己在内的一些集合服从规定的奇偶性。
- 集合的选取如下：
 - 对编号为K的信息位来说，K可以分解成2的幂的和，如编号为11， $11=1+2+8$ ，即第11位由1，2，8校验位校验，它同时属于1，2，8所在的集合。

海明编码举例

Char.	ASCII	Check bits
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	11111001111
c	0100000	10011000000
o	1100011	11111000011
d	1101111	00101011111
e	1100100	11111001100
	1100101	00111000101

Order of bit transmission

上例中, $m=7$, $r=4$, $n=11$, 显然 $2^4 >= 11+1$, 采用偶校验

$$3=1+2, 5=1+4$$

校验位: $1 \in (3, 5, 7, 9, 11)$

$$6=2+4, 7=1+2+4$$

$2 \in (3, 6, 7, 10, 11)$

$$9=1+8, 10=2+8$$

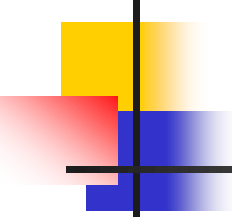
$4 \in (5, 6, 7)$

$$11=1+2+8$$

$8 \in (9, 10, 11)$

能纠正单比特错!

在接收方, 如果校验位1不满足偶校验, 而其他校验位都满足, 则第1位出错, 如果1/2/4位出错, 则第7位出错...



4.2.2 检错码

- 在实际通信中使用纠错码好还是检错码好呢？
- 例题：假设一个信道误码率是 10^{-6} ，且出错是孤立产生的（即只有单比特错），数据块长度为1000比特，如果采用纠错编码，需要10个校验位（ $2^{10} > 1011$ ），传送1M数据需要10000个校验位；如果采用检错编码，每个数据块只需一个奇偶校验位，传送1M数据只需1000个校验位和一个重传的数据1001位，共需要2001比特。
- 在多数通信中采用检错编码，但在单工信道中需要纠错编码。

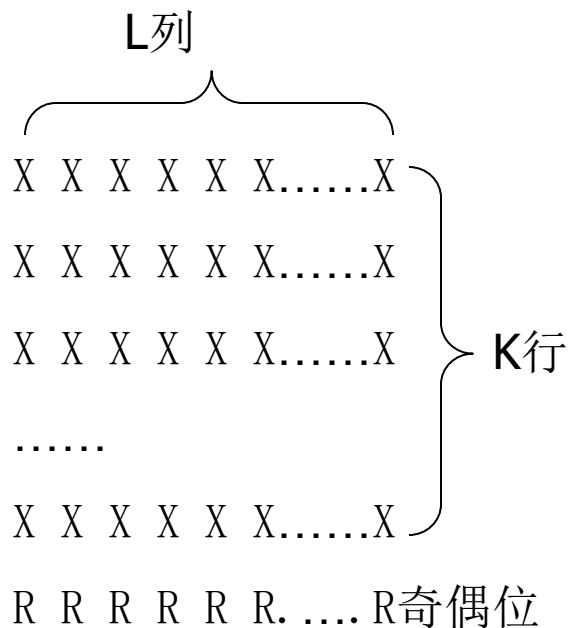


4.2.2 检错码

- 奇偶校验码 / 改进的奇偶校验码
- 多项式编码 / 循环冗余校验码
- 校验和

改进的奇偶校验

- 对数据位组成一个L位宽，K位高的长方形矩阵来发送，然后对每一列单独计算奇偶位，并附在最后一行作为冗余位。



差错检测能力分析：

- 该方法可以检测所有长度为L的突发性错误。
- 当出现错误长度大于L时，假设L列中任意一列检测出错的概率为 $1/2$ ，那么，整个数据块的错判率（实际出错却判为正确的概率）为 $(1/2)^L$ 。

该方法用在ICMP报头检验中。

注：突发性错误指最少第1位和最后一位是错误的，并不是指所有位都错误



多项式编码（循环冗余码CRC）

- CRC（Cyclic Redundancy Check）
- CRC编码的一般操作：给定一个 m 比特的帧或报文，发送方生成 r 比特的序列（也称为帧检验序列FCS，Frame Check Series），形成 $(m+r)$ 的码字，该码字能被某个事先确定的数整除。接收方用相同的数去除收到的帧，如果无余数，则认为数据帧无差错。



CRC也称多项式编码

- 多项式表示：即将k比特的数据用k项多项式表示，它的各项为 $X^{k-1} \dots X^0$ ，它的系数为数据中对应位的0或1。
- 例如：110001可表示成 $X^5 + X^4 + 1$

注：多项式编码是建立在多项式运算基础之上的，**多项式算术运算的加减法都等同于异或运算（没有借位和进位）**

多项式编码的推导

生成多项式，
最高和最低位
必须为1

设数据帧多项式表示为 $M(X)$ m 位序列

事先约定的除数为 $G(X)$ $r+1$ 位序列

帧检验序列FCS为 $F(X)$ r 位序列

加了检验序列的帧为 $T(X) = X^r M(X) + F(X)$
 $m+r$ 位序列

设 $X^r M(X) / G(X) = P(X) + R(X) / G(X)$

$[X^r M(X) - R(X)] / G(X) = P(X)$

在模2运算中，加法与减法等同，取 $F(X) = R(X)$
就能使 $T(X)$ 被 $G(X)$ 整除

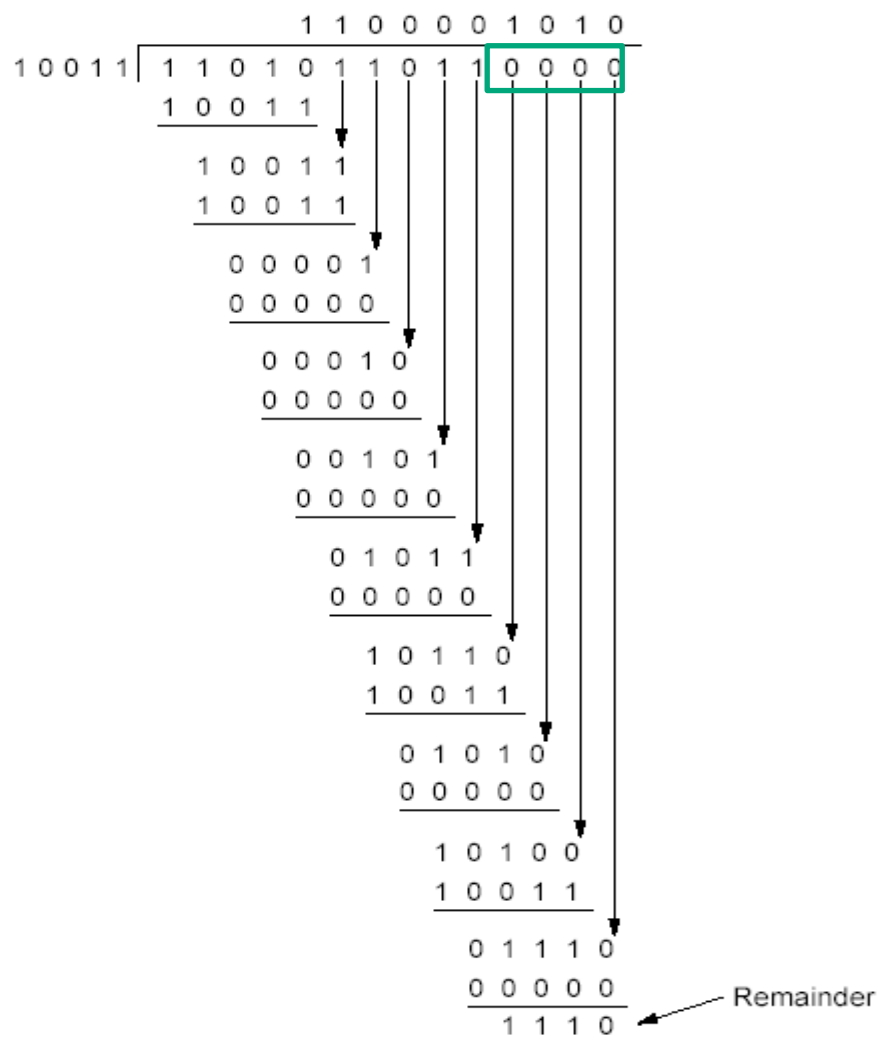
Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after appending 4 zero bits: 1 1 0 1 0 1 1 0 0 0 0

$r+1$ 位

补 r 位0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0 r 位

CRC的检测能力分析

- 能检验出所有长度小于等于 r 的错误（即检测率100%）
- 如果突发长度为 $r+1$ ，当且仅当差错与 $G(X)$ 相同时才被整除。根据突发错误长度的定义，其第1位和最后1位必须是1，因此与 $G(X)$ 完全相同的概率为 $(1/2)^{r-1}$ 。（检测率 $1-(1/2)^{r-1}$ ）
- 对于长度大于 $r+1$ 的差错，其错判率为 $1/2^r$ 。（检测率 $1-(1/2)^r$ ）

例： 第五版，书166页

待发的帧	1 1 0 1 0 1 1 1 1 1	生成多项式 $G(X)=X^4+X+1$ ，除数 1 0 0 1 1
发出的帧	1 1 0 1 0 1 1 1 1 0 0 1 0	得到余数为10，这里 $r=4$ ，所以检错码一定是4位
传输出错的帧	1 1 0 1 0 1 0 1 0 1 0 0 1 0	相当于异或了一个突发错误 $E(X)$
突发错误	1 0 1	$E(X)=(X^2+1)X^1$ ， $E(X)/G(X)$ 不能整除
	1 0 0 1 1	$E(X)=G(X)X^1$ ， $E(X)/G(X)$ 能整除
	1 0 1 1 1 1 1	$E(X)=G(X)(X^2+1)X^1$ ， $E(X)/G(X)$ 能整除



常用的多项式国际标准

- $\text{CRC-12} = X^{12} + X^{11} + X^3 + X^2 + X + 1$
- $\text{CRC-16} = X^{16} + X^{15} + X^2 + 1$
- $\text{CRC-CCITT} = X^{16} + X^{15} + X^5 + 1$
- $\text{CRC-32} = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$



校验和（Checksum）

- 多用于**Internet**上的数据校验，例如**IP**、**TCP/UDP**
 - 比较弱的差错检测能力，主要还是依赖于低层（数据链路层）的检错
- 数据以**16**比特为一组进行加法运算，产生**进位加到最后，然后求反码**

校验和计算实例

```
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      .... ..0. = ECT bit - transport protocol will ignore the CE bit
IP:      .... ...0 = CE bit - no congestion
IP: Total length   = 52 bytes
IP: Identification = 8143
IP: Flags          = 4X
IP:      .1.. .... = don't fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live    = 64 seconds/hops
IP: Protocol        = 6 (TCP)
IP: Header checksum = 3CA9 (correct)
```

```
00000000: 00 01 30 ff b7 c0 00 fd 00 00 1b 45 08 00 45 00 ..
00000010: 00 34 1f cf 40 00 40 06 3c a9 da f6 bc fd da 1e .4.
00000020: 6c 39 06 2f 00 50 c5 2a fa 5a 00 00 00 00 80 02 19.
00000030: ff ff ca c5 00 00 02 04 05 b4 01 03 03 02 01 01
00000040: 04 02 ..
```

```
45 00
00 34
1F CF
40 00
40 06
00 00
DA F6
BC FD
DA 1E
+ 6C 39
-----
3 C3 53
```

把进位3加到结果后: **3 + C3 53 = C3 56**

C 3 5 6
1100 0011 0101 0110 求反码: 0011 1100 1010 1001 得到3CA9



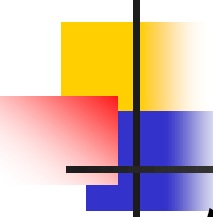
Chapter 4 数据链路层

- 4.1 数据链路层的基本概念与功能
- 4.2 差错检测与纠正
- 4.3 基本数据链路协议
- 4.4 滑动窗口（Slide Windows）协议
- 4.5 面向位的协议HDLC
- 4.6 面向字节的数据链路层协议-PPP



4.3基本数据链路协议

- 4.3.1一种无限制的单工协议（An Unrestricted Simplex Protocol）
- 4.3.2无错信道的单工停-等协议（Stop-and-Wait Protocol）
- 4.3.3有噪音信道的停-等协议（Stop-and-Wait Protocol for a Noisy Channel）



4.3.1 一种无限制的单工协议

- 假设：数据单向传输，收发双方的网络层一直处于就绪状态，处理时间可忽略不计，接收缓冲空间无限大，信道不会损坏或丢失帧
- 发送方无限循环地重复三个动作：
 - 从网络层取分组→构造帧→发出帧。

没有无需任何差错控制和流量控制
- 接收方也是无限循环地重复三个动作：
 - 等待事件（一个未损坏帧的到达）发生→帧到达后，从硬件缓冲中取出新到的帧→将帧的数据部分传给网络层

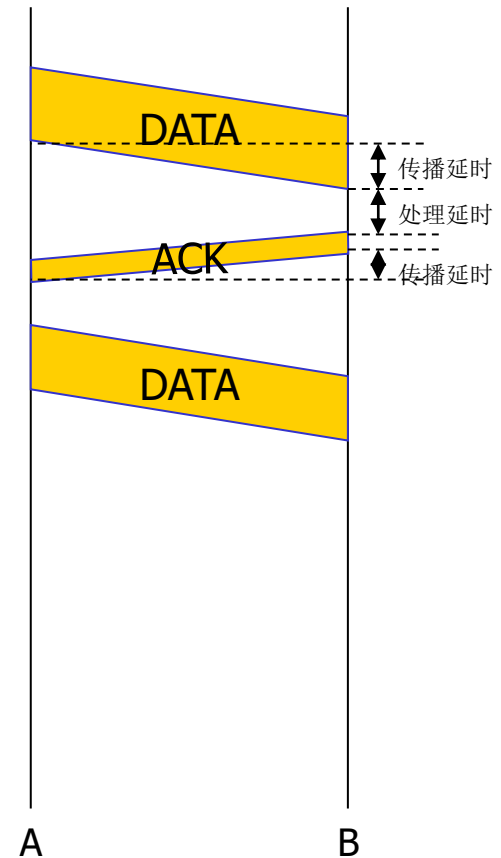
全双工：可以同时两个方向上发送数据

半双工：可以在两个方向上发送数据，但某一时刻只能在一个方向上发送数据

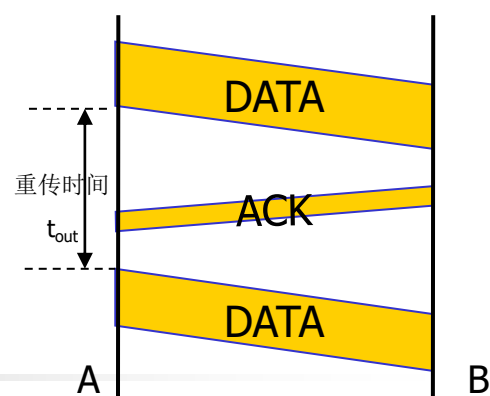
单工：只能在某一个方向上发送数据

4.3.2 无错信道的停-等协议

- 条件基本同协议1，
 - 修改：信道无噪，但接收端需要一定的接收处理时间，**接收缓冲只能存放一个帧**。
- 为了**防止发送快于接收**而造成数据丢失，发送端发送一帧后必须停止发送，等待接收端发回的**反馈确认**；
- 接收端在收到一个帧并发送网络层后，需向发送端发一反馈确认短帧，表示可发新帧。
- 由于需要反馈，且帧的发送和反馈是严格交替进行的，所以相当于采用半双工信道。



有噪音信道所涉及的问题



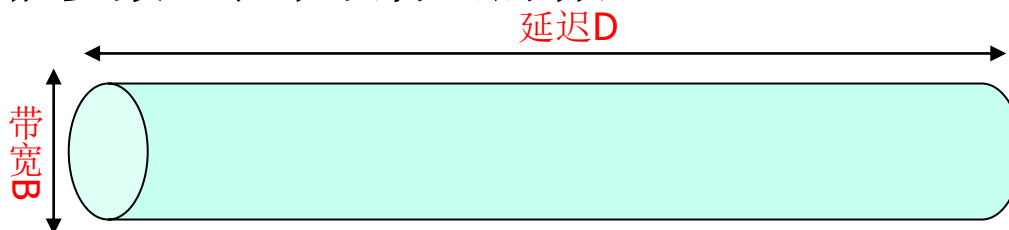
- 考虑实际有噪信道，帧既可能损坏（接收端可通过校验检查出错误），也可能完全丢失。
- 由于帧会丢失，发送端可能收不到反馈的确认帧，因此发送端必须引入超时机制（time out），即增加一个定时计数器，在一定时间后对没有确认的帧进行重发，也称作ARQ（Automatic Retransmit reQuest）。
 - 时间值应选择稍大于两倍端到端的信号传播时间和接收端的接收处理时间之和。
- 由于帧会丢失（假如确认帧丢失），为避免接收端收取重复的数据帧，必须通过为帧编制序号来解决重复帧的问题。
 - 帧的序号位数应尽可能的短从而少占用帧头的空间，在简单停-等协议中只需1个比特位（“0”→“1”，“1”→“0”）即可。

停-等协议对信道利用率的影响

- 在时延大的信道（如卫星通信）中，停-等协议的效率是很低的。
- 考虑卫星通信，典型的传播时间约为270ms。假设一个帧的发送时间为20ms，则从发送站开始发送算起，经 $20\text{ms} + 270\text{ms} = 290\text{ms}$ ，数据帧才能到达目的站。假设不考虑目的站的处理时间，且认为确认帧非常短，其发送时间可忽略不计，则又需270ms确认帧才能被发送站收到。因此信道的利用率为： $20\text{ms} / (290\text{ms} + 270\text{ms}) = 1/28$ ，非常低。
- 为了提高传输效率，可以设想让发送站连续不断地发送数据帧，当发完第28个帧数据后，恰好第1帧的确认帧到达，根据确认可紧接着发第29帧或重发第1帧。以后，每过20ms（发一个帧）就有一个确认帧到达，这样信道的利用率就大大地提高了。
- 允许发送站连续发送多个帧而不需等待确认的做法称作管道化（pipelining），属于一种窗口（windows）机制。

延迟*带宽乘积

- 单向延迟 × 带宽——**带宽延迟乘积**： BD ，（单位：比特\字节\帧）
 - 第一个比特到达接收端之前，发送者能发送的比特数，是一个有用的链路性能参数，表示该管道的容量。



- 往返延迟 × 带宽： $2BD$
 - 假设发送窗口为 W ，它是发送端在接收到接收端给出的确认之前能够发送的帧（或字节，比特）数。
 - 若要链路利用率100%，则 $W=2BD+1$ （单位：帧，注意，这里的 BD 单位要换成帧），
 - **链路利用率** $\leq W/(1+2BD)$ ，注意，这里的+1是因为必须接收完整帧后确认帧才会被发出，所以 W 和 BD 的单位也应该是**帧**。该值是上限，因为没有考虑处理时间。

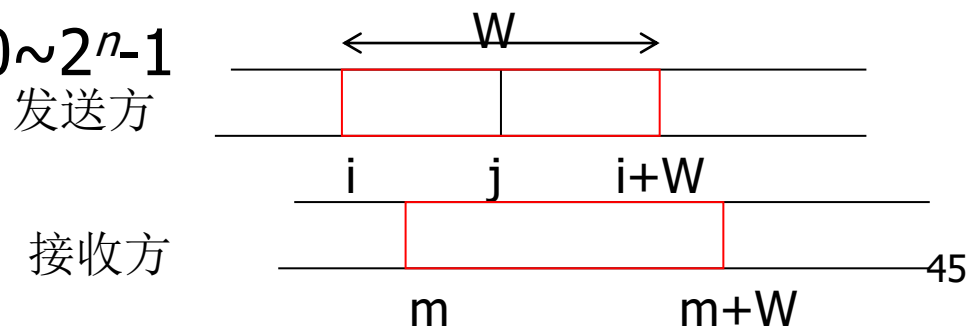


Chapter 4 数据链路层

- 4.1 数据链路层的基本概念
- 4.2 差错检测与纠正
- 4.3 基本数据链路协议
- 4.4 滑动窗口（Slide Windows）协议
- 4.5 面向位的协议HDLC
- 4.6 面向字节的数据链路层协议-PPP

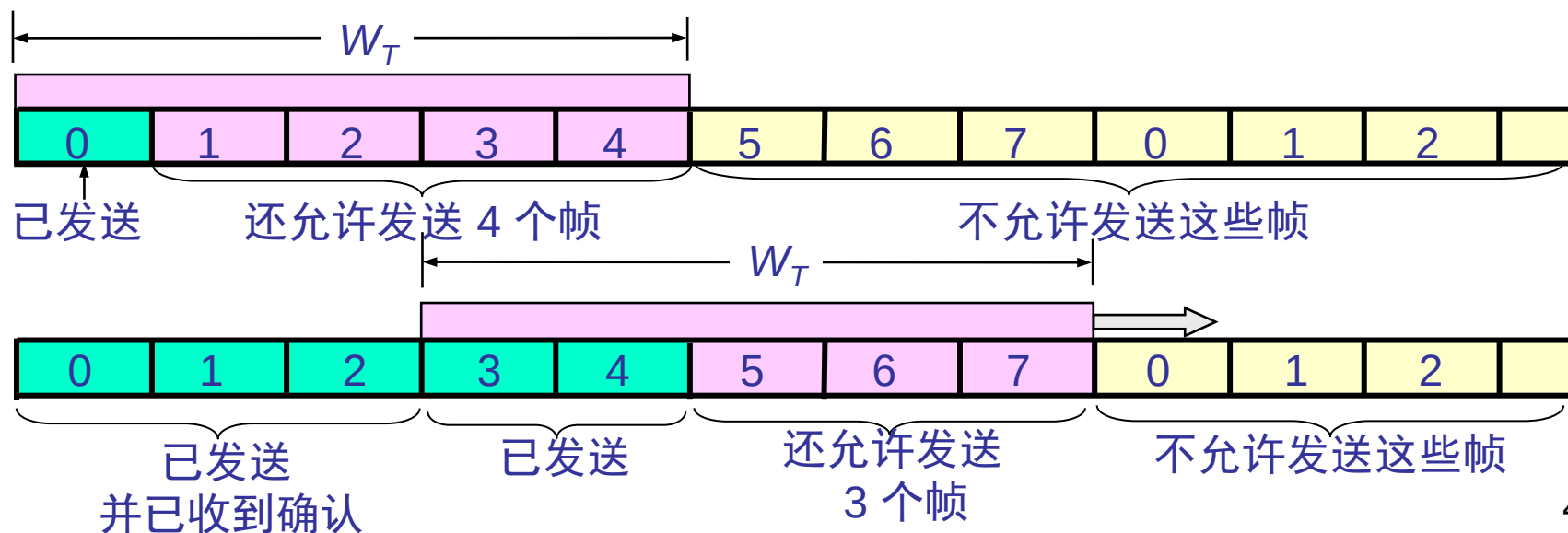
4.4 滑动窗口（Slide Windows）协议

- 滑动窗口协议是一种非常可靠、适用于各种条件的通用流量控制协议，特别是在效率、复杂性及对缓冲区的需求等方面可作灵活调配。
- 发送端和接收端分别设定发送窗口和接收窗口。
- 发送窗口用来对发送端进行流量控制。
- 主要的滑动窗口协议有回退N协议（go-back-n，出错全部重发）和选择重发协议，停-等协议是窗口大小为1的滑动窗。
- 帧序号用 n 位编码，范围为 $0 \sim 2^n - 1$



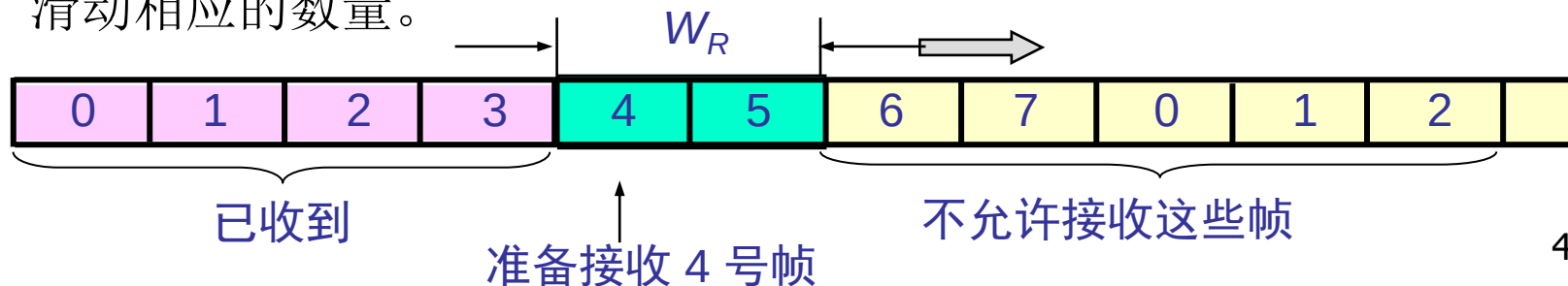
发送窗口

- **发送窗口**就是发送端允许不等确认而连续发送的帧的**序号表**。
- 允许连续发送的帧的数量称为**发送窗口尺寸**，表示为 W_T 。发送端必须有 W_T 个**输出缓冲区**来存放 W_T 个数据帧的副本以备数据帧的重发。
- 当发送端收到发送窗口下沿帧的肯定确认时，将发送窗口整体向前滑动一个序号，并从输出缓冲区中将相应的数据帧副本删除。



接收窗口

- **接收窗口**是接收端**允许接收**的帧的**序号表**。
- 允许接收的帧的数量称为**接收窗口尺寸** W_R 。同样接收端也必须设置相应数量的**缓冲区**来支持接收窗口。
- 对接收端收到的帧的序号落在接收**窗口外的帧**被直接**丢弃**。只有落在接收窗口内的帧才会被接收端进行校验处理，若校验正确：
 - 当接收的帧不是接收窗口下沿帧时，必须暂存在输入缓冲区。
 - 当接收到接收窗口下沿帧时，**会将其连同后面连续的若干个检验过的正确帧按顺序交给网络层**，在发回确认帧的同时将接收窗口向前滑动相应的数量。



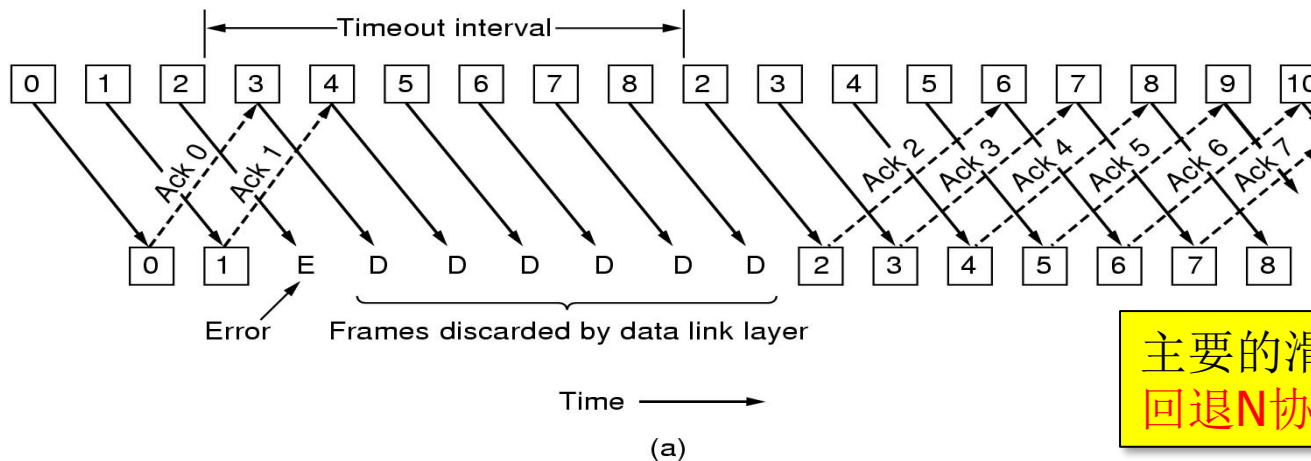


数据的捎带确认(如何捎带?)

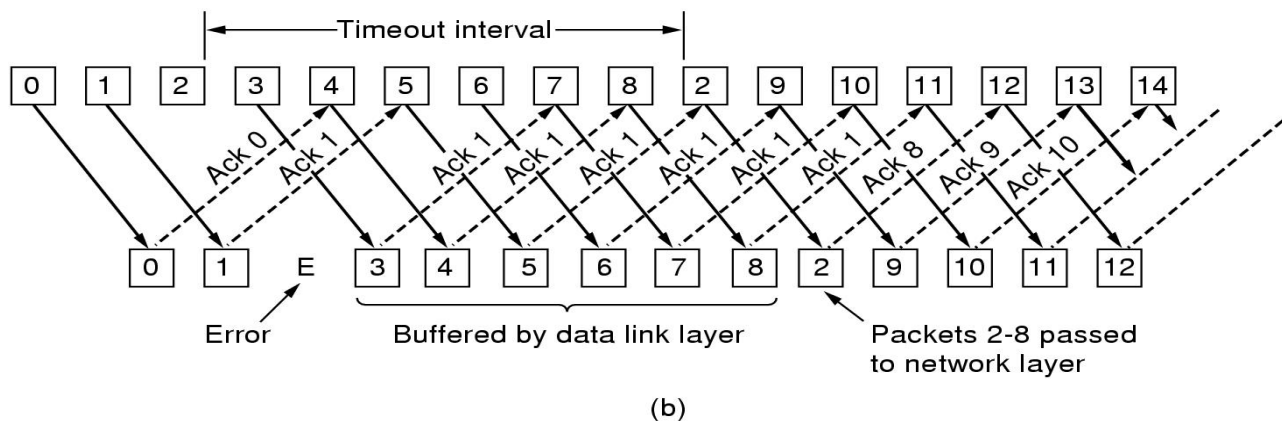
- 在实际通信中，通常收发**双方都相互发送数据**。
- 为了**提高效率**，可以将确认信息放在数据帧中作为一个控制字段连同数据一起发送给对方，这种方式称为**捎带应答**（piggybacking）。
- 当一方收到对方的数据帧后：
 - 若正好也有数据需发给对方，则立即可使用捎带应答。
 - 若暂时没有数据需发给对方或数据还未准备好，则等待一定的时间，如果在该时间内准备好了数据，则可以使用捎带应答。如果未准备好，为了防止对方等待时间过长而超时重发，必须立即发送一个单独的确认帧。
- 使用**捎带应答**就不用对每一个帧都作确认，可以用**对下一帧的期待来代替对该帧之前的所有帧的确认**。

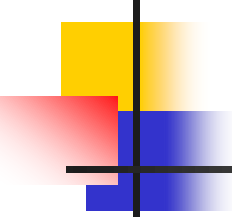
实际网络中大多用ACK n 表示
“第 $n-1$ 号帧已经收到，现在期望接收第 n 号帧”

回退N协议和选择重发协议



主要的滑动窗口协议有：
回退N协议和选择重发协议



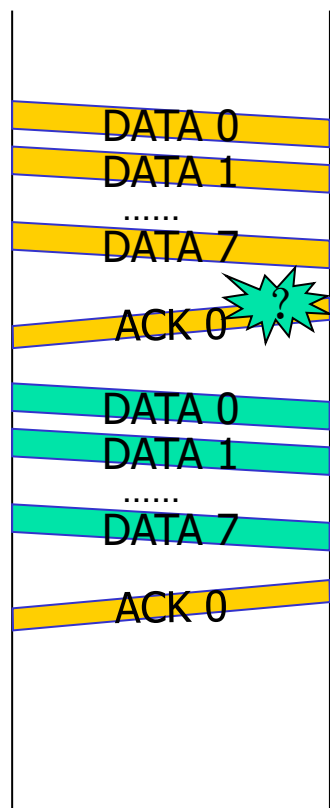


回退N协议 (go-back-n)

- 回退N协议(也叫**出错全部重发协议**)中，发送窗口的尺寸是大于**1**，而**接收窗口**的尺寸则等于**1**。
- 由于接收窗口的尺寸为**1**，接收端只能按顺序接受数据帧，一旦某个帧出错或丢失，只能丢弃该帧及其所有的后续帧（因为发送窗口的尺寸是大于**1**的）。发送端超时后需重发出错或丢失的帧及其后续所有的帧。
- 发送端需要为每个待确认的帧都各自设置一个定时计数器。
- **发送窗口的尺寸不能超过 2^n-1** (这里的**n**为序号的编码位数)，否则会造成接收端无法分辨新、旧数据帧。
- 出错全部重发协议只要求发送端保持一定数量的缓存来保存没有确认的数据帧，对接收端没有缓存的要求。但在误码率高的情况下，会大大降低信道的利用率。

发送窗口尺寸超过 2^n-1 造成帧的混淆

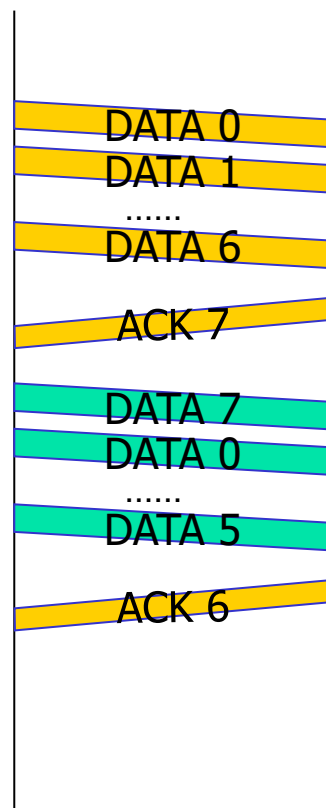
- 设序号位数为3，序号0~7，发送窗口假设=8.
- 发送方考虑两种情况：
 - 所有确认帧都收到，发下一批数据。
 - 所有确认帧都丢失，重传当前窗口所有数据帧。



发送方

接收方

- 设发送窗口假设=7.



发送方

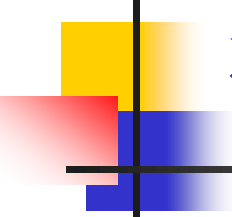
接收方

接收方：
是新数据还是旧数据？

如果前一批确认顺利到达，则是新数据，收；

否则，发送端超时重传，是旧数据，不收。

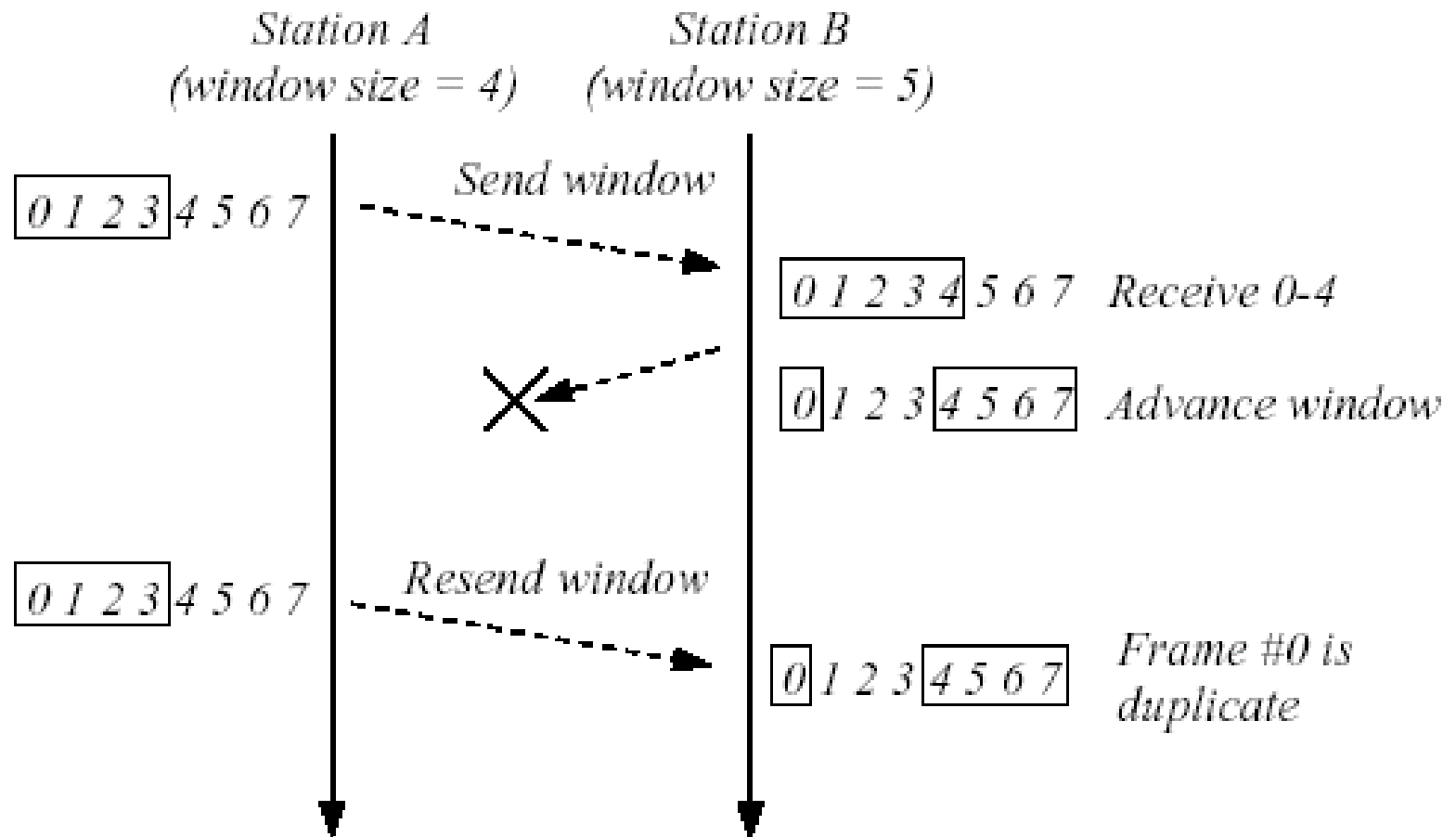
累计确认：当确认序号为n时，意味着n之前的帧如n-1、n-2等帧都自动得到确认。



选择重发协议 (selective repeat)

- 选择重发协议中，**发送和接收窗口的尺寸都大于1**。
- 由于接收窗口的尺寸大于1，接收端可存储坏帧之后的其它数据帧（**落在接收窗口**），接收端对错帧发否定确认帧，因此发送端只需重发出错的帧，而不需重发其后的所有后续帧。
- 接收端正确收到重发的帧后，可对其后连续的已接收的正确帧作一次总体确认（最大序号的确认），并交送网络层。大大提高了信道的利用率。
- **接收窗口的尺寸不能超过 2^{n-1}** （即序号范围的1/2），否则可能造成帧的重叠。
- 发送窗口的尺寸一般和接收窗口的尺寸相同，发送端为每一个输出缓存区设置一个定时计数器，定时器一旦超时，相应输出缓存区中的帧就被重发。

接收窗口的尺寸超过 2^n-1 造成帧的重叠





Chapter 4 数据链路层

- 4.1 数据链路层的基本概念与功能
- 4.2 差错检测与纠正
- 4.3 基本数据链路协议
- 4.4 滑动窗口（Slide Windows）协议
- **4.5 面向位的协议HDLC**
- 4.6 面向字节的数据链路层协议-PPP



面向位
有确认
无连接



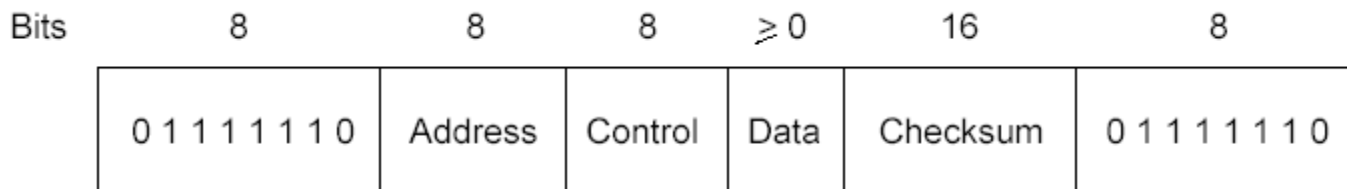
面向字节
无确认
无连接



4.5面向位的协议HDLC

- 高级数据链路控制（**High-Level Data Link Control**）是由国际标准化组织制定的**面向位**的有序链路层协议。
- 是为非平衡的链路级操作而研制的，采用主从结构，链路上一个主站控制多个从站，主站向从站发命令，从站向主站返回响应。
- **HDLC**中只有一个地址域，即从站的地址，在命令帧中，它是目的地址，在响应帧中，它是源地址。

HDLC的帧格式



1. 帧标志序列：**01111110**，作为起始和结束标志，在数据位有**5**个连续的**1**出现时，就插入**1**个**0**（位填充）
2. 地址段：在命令帧中表示目的地址，在响应帧中表示源地址，全**1**为广播地址，全**0**为测试地址

3. 控制字段

用来表示帧的种类，执行信息传送，监控功能：

信息帧 (I: Information)：用来实现信息的传送，含有信息字段

监控帧 (S: Supervision)：帧中不包含信息字段，具有监控链路的作用，并能对收到的帧进行确认。

无编号帧 (U: Unnumbered)：对数据链路进行附加控制。

I帧	Bits	1	3	1	3
	(a)	0	Seq	P/F	Next
S帧					
	(b)	1	0	Type	P/F
U帧					
	(c)	1	1	Type	P/F

Seq：发送端发送序列编号，这里是3比特，采用模8循环编号。

Next：表示发送端准备接收的序列号，也采用模8循环编号。

Type：表示监控功能的类型。**Modifier**：附加的修改功能。

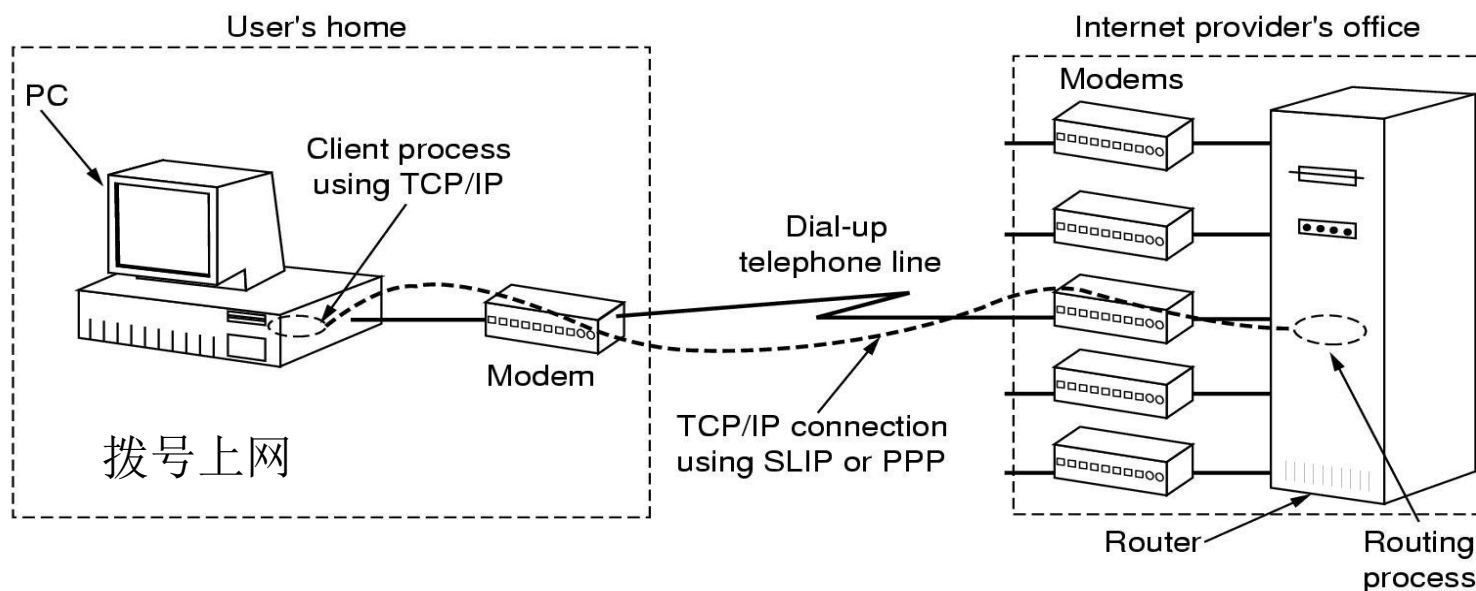
P/F：在命令帧中作为询问比特，在响应帧中作为终止比特。



Chapter 4 数据链路层

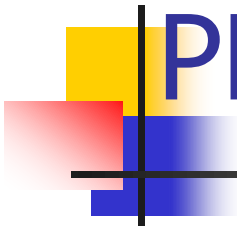
- **4.1**数据链路层的基本概念
- **4.2**差错检测与纠正
- **4.3**基本数据链路协议
- **4.4**滑动窗口（**Slide Windows**）协议
- **4.5**面向位的协议**HDLC**
- **4.6**面向字节的数据链路层协议-**PPP**

4.6 面向字节的数据链路层协议- PPP(Point to Point Protocol)



SONET上的数
据链路层封装





PPP(Point to Point Protocol)

- PPP由以下3部分组成：
 - 串行链路上的数据报封装方法
 - 链路控制协议（LCP: Link Control Protocol），用来建立、配置、测试数据链路连接
 - 对不同的网络层协议定义了网络控制协议族

PPP的帧格式

FLAG 7E	ADDR FF	CONTROL 03	PROTOCOL	INFORMATION (<=1500)	CRC	FLAG 7E
------------	------------	---------------	----------	----------------------	-----	------------

0021	IP数据报
C021	链路控制数据
8021	网络控制数据

- PPP有以下优点：
 - 可在一条串行线上支持多种网络层协议
 - 有CRC检错
 - 用网络控制协议动态协商每一端的IP地址
 - 链路控制协议协商多种数据链路选项。

PPP协议有同步传输和异步传输两种方式，在SONET同步传输网上，使用位填充方式，而在异步传输时，是一种面向字节的协议，采用字节填充。



异步传输：字节填充

- **PPP**帧都以标志字节**0x7E**开始和结束
- 字符 **0x7D**用作转义字节，转义算法为当 **0x7d** 出现在信息域中时，紧接着的字节的第 6 个比特要取其补码，即与 **0x20** 做异或运算
 - **0x7E**转变为 2 字节序列(**0x7D, 0x5E**)
 - **0x7D** 转变为 2 字节序列(**0x7D, 0x5D**)
 - **ASCII** 码的控制字符（即数值小于 **0x20** 的字符），则在该字符前面要加入一个 **0x7D** 字节，同时将该字符的编码加以改变

同步传输：位填充

- **PPP** 协议用在 **SONET/SDH** 链路时，是使用同步传输（一连串的比特连续传送），采用位填充（零比特填充）

信息域中出现了和
标志字节完全一样的
8 比特组合

0 1 0 0 1 1 1 1 1 0 0 0 1 0 1 0
会被误认为是标志字段

发送方在 5 个连 1 之后
填入 0 比特再发送出去

0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0
发送方填入 0 比特

在接收方把 5 个连 1
之后的 0 比特删除

0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0
接收方删除填入的 0 比特



小结

数据链路层

物理层

- 向网络层提供的服务
 - 面向连接，无连接有确认，无连接无确认
- 成帧
- 差错检测：纠错码，检错码，超时重传
- 流量控制：滑动窗口
- 协议：基本协议，HDLC，PPP



小结

- 熟悉数据链路层的基本功能，掌握常用的几种检错码和数据链路层协议。
- 习题：第三章，3,6,9-13,16,17,20,23



Chapter 5 LAN & MAC Sub layer

- 5.1 传统LAN的基本概念
- 5.2 多路访问协议
- 5.3 以太网的MAC协议
- 5.4 无线LAN
- 5.5 网桥
- 5.6 交换式局域网



Chapter 5 LAN & MAC Sub layer

- 5.1 传统LAN的基本概念
- 5.2 多路访问协议
- 5.3 以太网的MAC协议
- 5.4 无线LAN
- 5.5 网桥
- 5.6 交换式局域网



5.1 传统LAN的基本概念

■ 5.1.1 LAN的模型

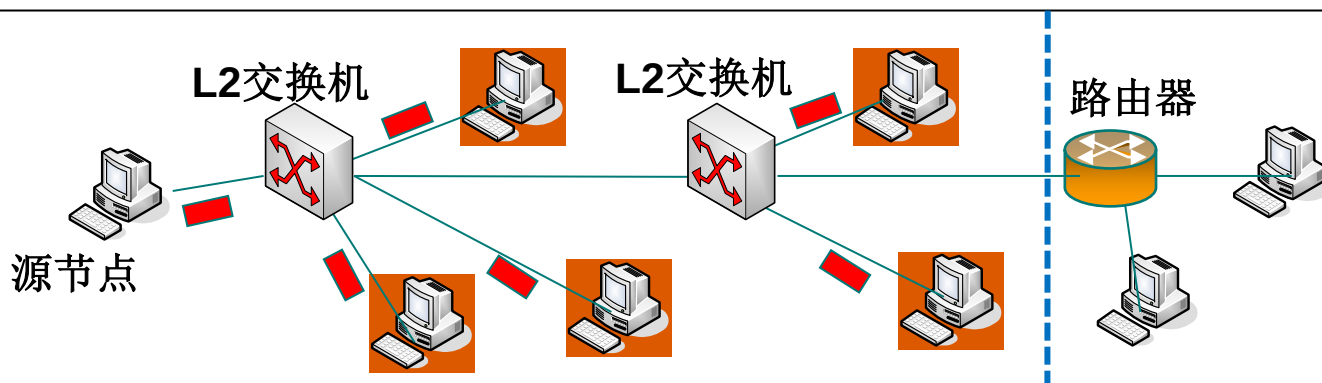
- **局域网（LAN）**是在一个小区域范围内对各种数据通信设备提供了互连的信息网。
- 决定**局域网特性**的主要技术：
 - 用以传输数据的**传输媒体**
 - 用以连接各种设备的**拓扑结构**
 - 用以共享资源的**媒体访问控制方法**

5.1传统LAN的基本概念

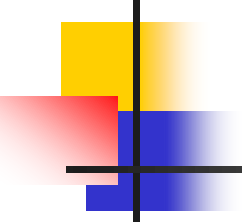
- 局域网**LAN**（**Local Area Network**）是在一个小区域范围内（家庭、实验室、企业部门等）对各种设备提供了互连的网络

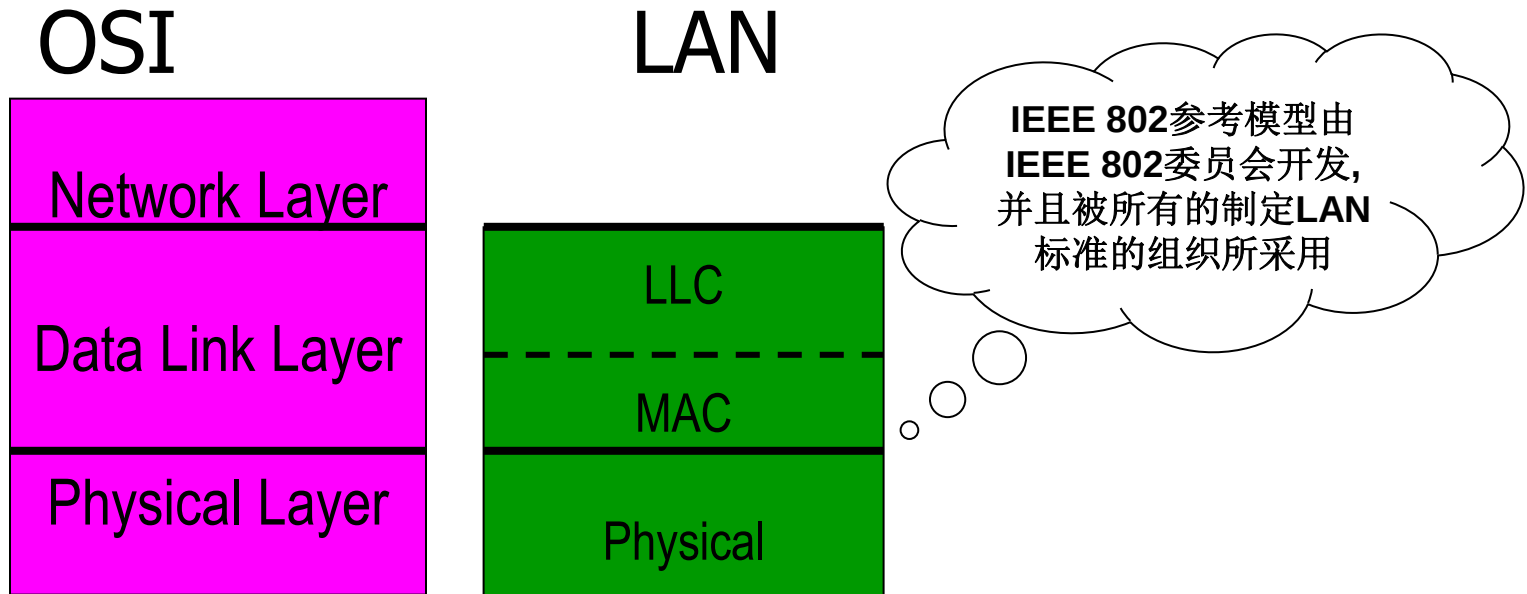
局域网被定义为一个广播域，早期局域网的构建基于**共享（广播式）链路**，当前的局域网的采用**L2交换机**互连。

广播域：在局域网内目的地址为广播地址数据链路层帧会被所有的网络节点接收和处理



- 局域网因**Internet**的发展而出现并普及，是最常用的**Internet**接入网技术之一
 - 基于**802.3**的有线局域网络（以太网）、基于**802.11**的无线局域网

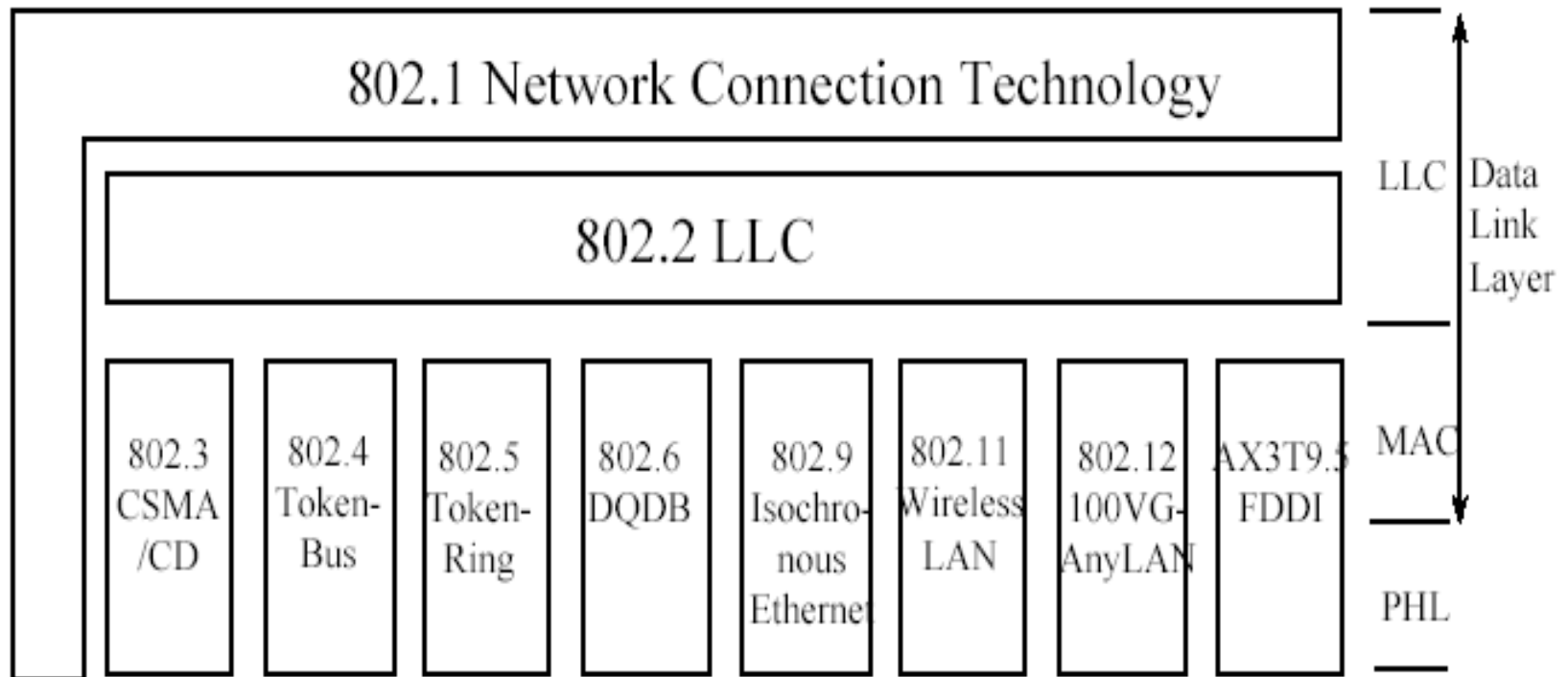
- 
- 也称为**IEEE 802**参考模型



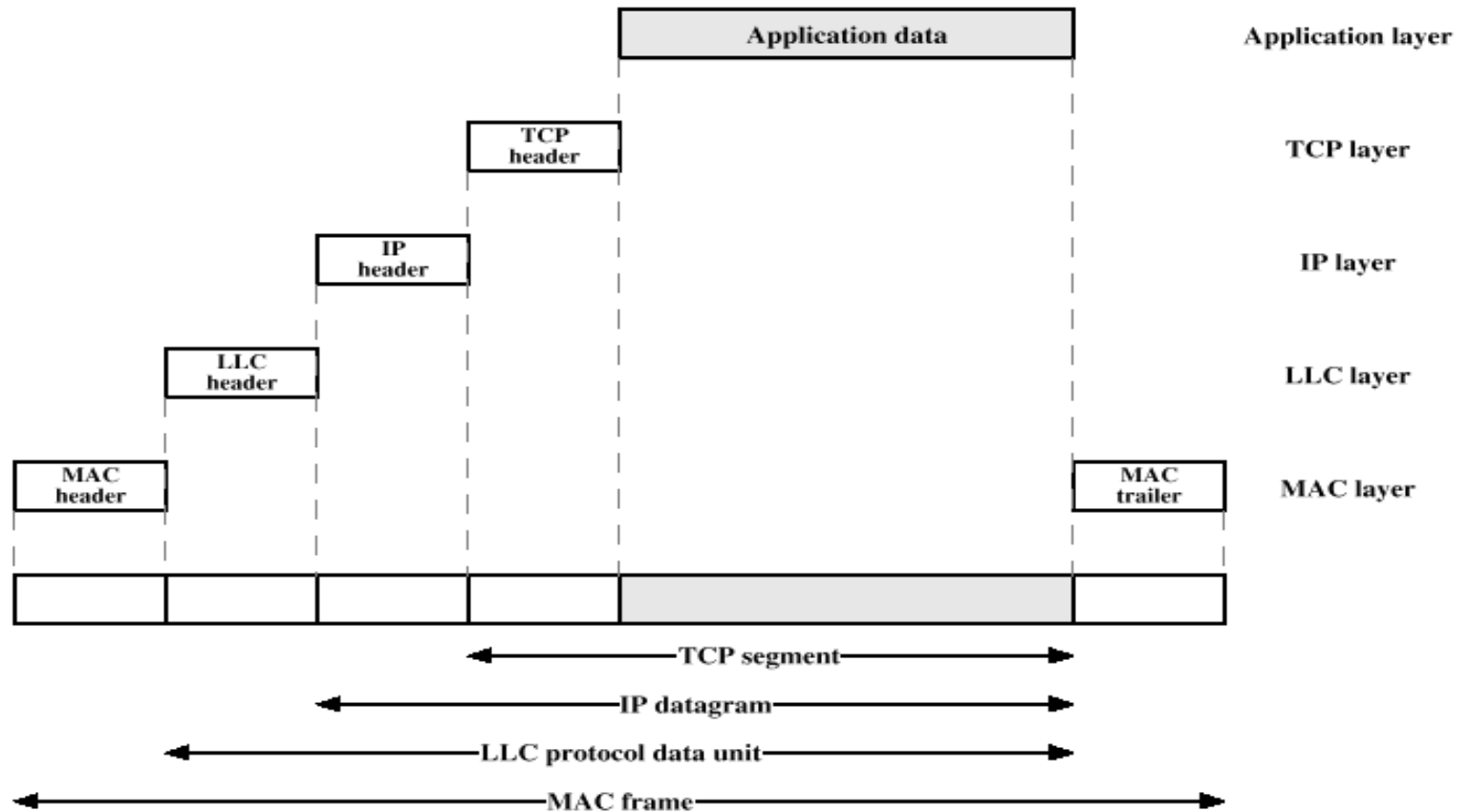
- ✓物理层对应着OSI参考模型的物理层
 - ✓对传输介质和拓扑的规定
- ✓OSI参考模型的数据链路层被划分为两个子层, 即逻辑链路控制(LLC)子层和介质访问控制(MAC)子层



IEEE 802 Architecture



Operation of higher layers over LAN



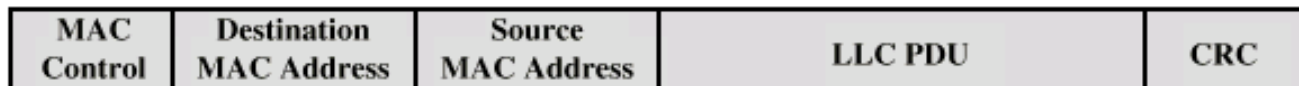


LAN的传输媒体

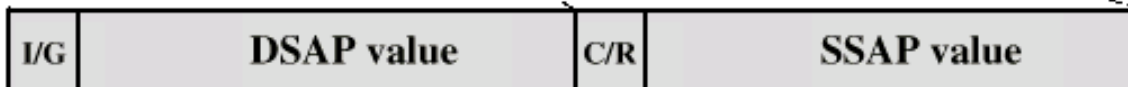
- 双绞线
- 同轴电缆
- 光纤
- 无线电波
- 红外

帧格式

**MAC
Frame**



**LLC
PDU**



**LLC
Address Fields**

I/G = Individual/Group

C/R = Command/Response

LLC - Logical Link Control

PDU - Packet Data Unit

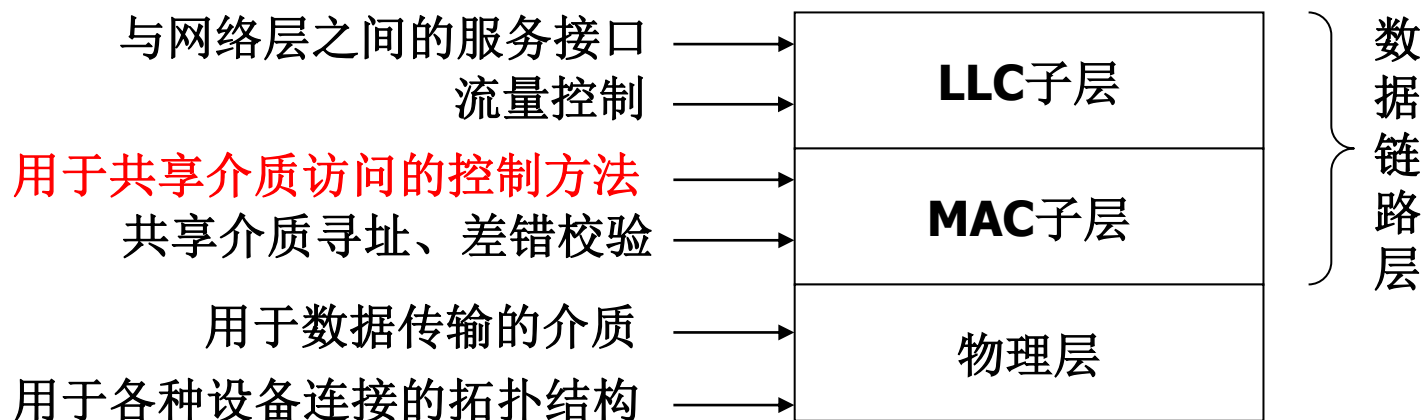
DSAP - Destination Service Access Point

SSAP - Source Service Access Point

SAP indicates the user

局域网主要技术

- 用于数据传输的传输介质
- 用于各种设备连接的拓扑结构
- 用于共享介质访问的控制方法





MAC和LLC

- **LLC: Logical Link Control**
- **MAC: Medium Access Control**
- 将数据链路层划分为两个子层的原因
 - 在传统的链路层控制中没有管理访问共享（广播式）介质的逻辑需求，**MAC**子层负责对共享介质的访问进行控制
 - 对于相同的**LLC**子层，可以提供不同的**MAC**子层
 - 常用以太网、**802.11**无线局域网等局域网具有相同的**LLC**子层，只是**MAC**子层不同



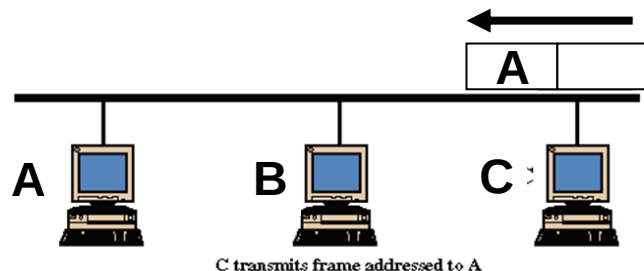
LLC子层—服务

- **LLC**是一个对所有**LAN**都相同的链路层协议，它定义了两个用户之间数据交换的机制，向网络层提供了三种类型的服务
 - 无确认、无连接的服务
 - 有确认、无连接的服务
 - 面向连接的服务

对于**LAN**，无确认的非连接服务是优先考虑的选项，但在一些不可靠网络环境，例如无线网络，一般使用有确认无连接服务

MAC子层

- **多路访问控制**：**MAC**对由多个节点共享的传输介质（有线介质或者无线信道）的进行**访问控制**
- **寻址**：在**MAC**层定义了数据链路层地址，在共享介质上，节点根据帧的目的**MAC**地址来决定是否处理接收到的帧
 - 节点上的网络接口设备都有一个**MAC**地址，将接收：
 - （1）帧的目的**MAC**地址与自己相同。
 - （2）目的**MAC**地址为广播地址
- 经过局域网络数据链路层处理后的数据单元也被称为**MAC**帧，**MAC**帧的通用格式



MAC地址是网络接口设备例如网卡的物理地址，购买这些设备时一般就固化在设备的内部，它用来标识连接在**LAN**上的网络接口的物理接入点

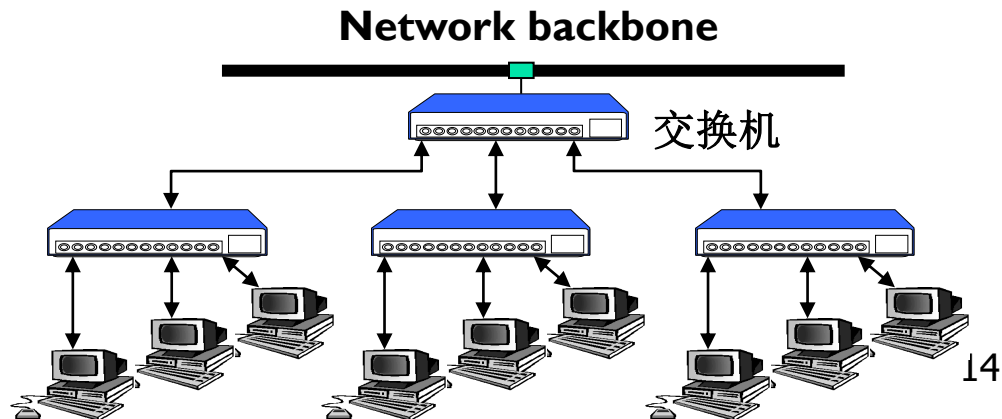
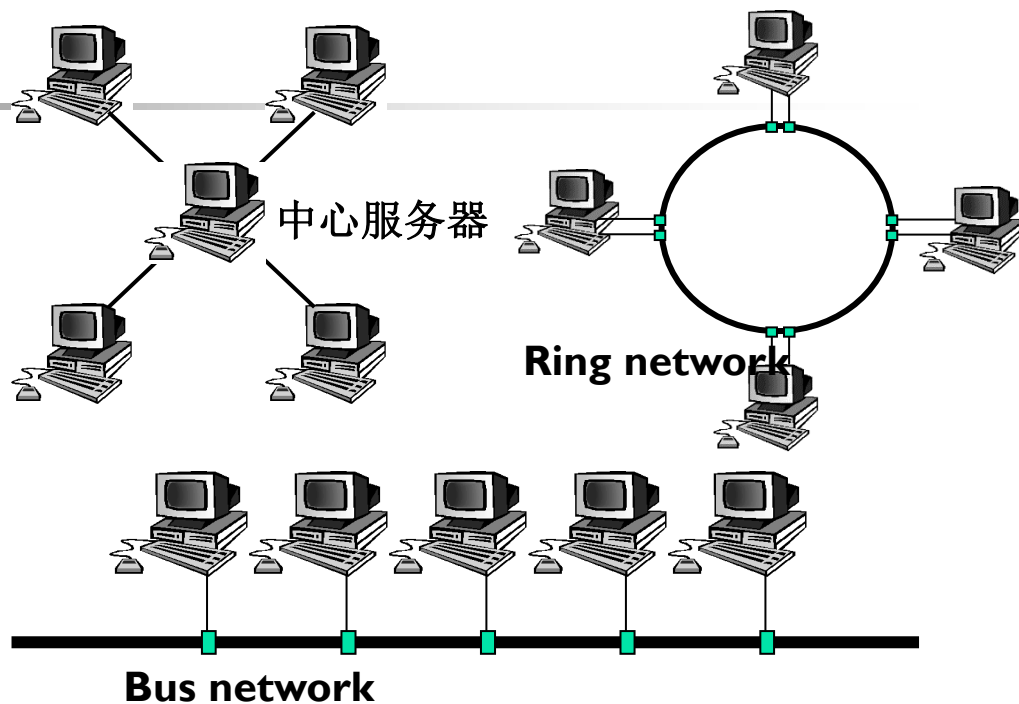
物理层技术

■ 传输介质

- 电缆，包括双绞线、同轴电缆等，最常用的是（增强型）五类非屏蔽双绞线，六类非屏蔽双绞线等
- 光纤，包括单模光纤和多模光纤
- 无线传输，包括无线电波、红外线等

■ 网络拓扑

- 星型、环形、总线型、树型





Chapter 5 LAN & MAC Sub layer

- 5.1 传统LAN的基本概念
- 5.2 多路访问协议(multiple access)
- 5.3 以太网的MAC协议
- 5.4 无线LAN
- 5.5 网桥
- 5.6 交换式局域网

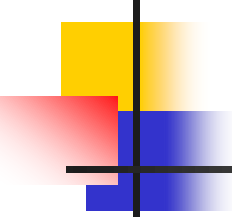


概述

- 多路访问信道（也称为广播信道）：多个用户共享的同一条链路，如果一个用户发送，则其它用户都能侦听到
- 多路访问协议：控制多个用户对广播信道的访问，尽量避免多个用户同时开始帧发送，因为这样会导致冲突（**Collision**）

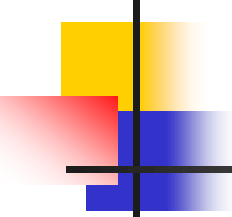
多路访问协议的作用就是在多个竞争的用户之间分配单个广播信道

- 设计原则
 - 简单，易于实现（集中式 **vs.** 分布式）
 - 有效的介质利用率
 - 对用户的公平合理



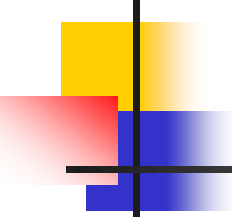
5.2 多路访问协议

- 对于广播信道，需要解决信道分配问题，信道的分配方案有：
 - 静态分配：如传统的FDM或TDM，如果有N个用户，把带宽或时间分成N份，每个用户静态地占用一个。缺点是不能有效地处理突发数据，有的用户无通信量时白白浪费资源。
 - 动态分配：异步时分多路复用。分为两种：
 - 随机访问（争用，contention）：只要有数据，就可直接发送，发生冲突后再采取措施解决冲突。适用于负载轻的网络，负载重时效率低，例如ALOHA、CSMA等。
 - 控制访问：发送站点必须先获得发送的权利，再发送数据，不会发生冲突。在负载重的网络中可获得很高的信道利用率。主要有轮询（round-robin）和预约（reservation）两种方式，例如802.11的PCF模式等。



动态信道分配的假设

- 单信道：所有用户都共享同一个信道，在该信道上发送数据、接收数据。
- 时间连续或分槽：
 - 时间连续：任意时刻都可发送。
 - 分槽：只在某个时槽的起始点可以发送。
- 发送端能否检测冲突：如果多个用户同时发送数据，多路信号在时间上叠加使接收端无法正常解码，称为冲突。
 - 对有线信道，节点硬件可设计成边发送边检测冲突，而无线信道很难做到。
- 载波侦听或不侦听：
 - 有载波侦听：发送前先侦听信道，可知道当前时刻信道是否“忙”
 - 无载波侦听



争用协议一：ALOHA协议

- 20世纪70年代，美国夏威夷大学的**ALOHA网**通过无线广播信道将分散在各个岛屿上的远程终端连接到本部的主机上，是**最早采用争用协议**的网络。
- 在**ALOHA**系统中，每个站给中央计算机发送帧之后，中央计算机把该帧重新广播给所有站。
- 有两个版本：
 - **纯ALOHA协议**（Pure ALOHA）：每个站点只要有**数据就可发送**；通过**监听**来自集线器的广播，发现是否发生冲突；若冲突，则**等待一段随机时间**，再重新发送。
 - **时隙ALOHA协议**（Slotted ALOHA）：将信道时间分为离散的时间片（Time Slot），每个时间片可以用来发送一个帧。一个站点有数据发送时，必须等到下个时间片的开始才能发送。与**纯ALOHA**相比信道的利用率提高一倍。



争用协议二：CSMA协议

- **载波侦听多路访问（Carrier Sense Multiple Access）**协议中，各站点不是随意发送数据帧，而是**先要监听**一下信道，根据信道的状态来调整自己的动作，只有发现信道空闲后再可发送数据。即“**讲前先听**”
- 常见的四种CSMA协议：
 - 非坚持式CSMA（non-persistent）
 - 1-坚持式CSMA（1-persistent CSMA）
 - p -坚持式CSMA（ p -persistent CSMA）
 - **带有冲突检测的CSMA**（CSMA with Collision Detection）



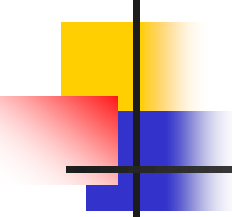
非坚持式CSMA

- 当一个站点要发送数据时，先监听信道，若信道忙，就随机等待一段时间后再开始监听信道（非坚持）；一旦发现信道空闲，就立即发送数据。
- 缺点：
 - 无法准确识别空闲的时刻点，导致信道利用率低。
 - 相隔短的时间内都检测到空闲同时发送。。



1-坚持式CSMA

- 当一个站点要发送数据时，首先监听信道，若信道忙，就坚持监听，一旦发现信道空闲，就立即发送数据（发送数据的概率为1）。若发生冲突，就等待一随机长时间，再重新开始监听信道。
- 两种发生冲突的可能：
 - 信号传输的延迟造成的冲突。
 - 多个站点在监听到信道空闲时，同时发送。



p -坚持式CSMA

- 用于**时隙信道**。当一个站点要发送数据时，先监听信道，若信道忙则等到下个时间片再监听信道；若信道空闲，则以概率 p 发送数据，概率 $q=1-p$ 推迟到下个时间片再重复上述过程，直到数据被发送。
- 概率 p 的目的就是试图降低**1-坚持式**协议中多个站点**同时发送而造成冲突的概率**。
- 采用**坚持监听**是试图克服非坚持式协议中造成的**时间延迟**。
- p 的选择直接关系到协议的性能。

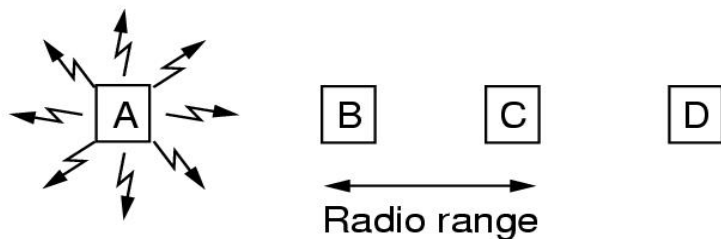


CSMA/CD

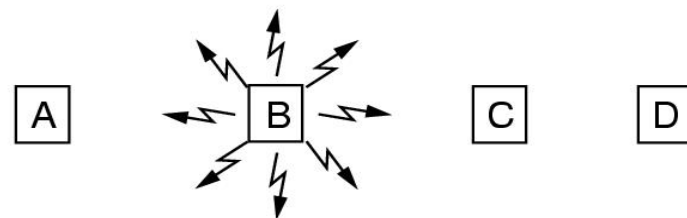
- CS协议的“**讲前先听**”对ALOHA系统进行了有效的改进，但在发送过程中若发生冲突，仍要将剩余的无效数据发送完，既浪费了时间又浪费了带宽。
- CD协议的“**边讲边听**”可对CSMA作进一步的改进。发送过程中，**仍然监听信道，通过检测回复信号的能量或脉冲宽度并将之与发送的信号作比较**，就可判断是否发生冲突。一旦发生冲突，立即取消发送，等待一随机时间后再重新尝试发送。

无线局域网（Wireless LAN）使用CSMA时存在的问题

- WLAN基本使用CSMA协议，但由于各个站点发出的信号**范围有限**（不像有线网络中一个站点发出的信号可到达所有的站点），因此会造成：
 - **隐藏站点**问题（hidden station problem）：图(a)中，A向B发送时，由于C听不到误以为可发送数据，造成B接收失败。
 - **暴露站点**问题（exposed station problem）：图(b)中，B向A发送时，C听到信道忙误认为它不能向D发送数据，实际上并不影响A和D两站的接收。



(a)



(b)

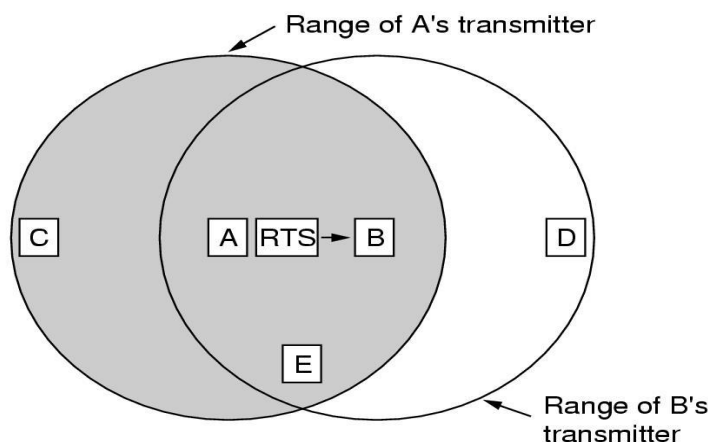


CSMA with Collision Avoidance协议

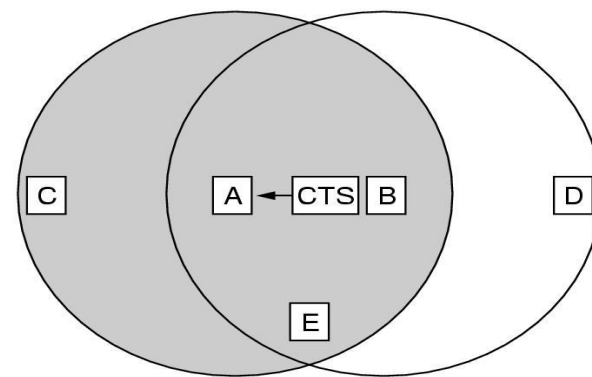
- CSMA/CA（避免冲突的CSMA协议）是WLAN采用的介质访问控制协议，其相应的国际标准为IEEE 802.11。
- 发送方先发送**短帧RTS**（Request to Send），接收方响应一个**短帧CTS**（Clear to Send），使接收方周围的站点不会在即将到来的数据帧期间发送数据而导致冲突（**避免冲突**）。
- 当多个站点同时发短帧时仍会发生冲突，在预定时间内没有收到**CTS**的**发送方采用二进制指数退避算法**，在等待一随机时间后再次重试。

CSMA/CA的分析

- A首先向B发送一包含后继数据帧长度的RTS短帧。B回复一个也包含数据帧长度（从RTS中得到）的CTS短帧。A一旦收到CTS，就开始发送数据。
- 侦听到RTS的其它站点均保持足够长的沉默时间使A可无冲突地收到CTS。侦听到CTS的站点也在后续的一定时间内（从CTS中可知）保持沉默。
 - C位于A范围内，B范围外：能听到RTS，听不到CTS，也需保持静默。
 - D和E都位于B范围内：听到CTS后，关闭所有的发送，直到A到B的帧被认为发送完毕。



(a)



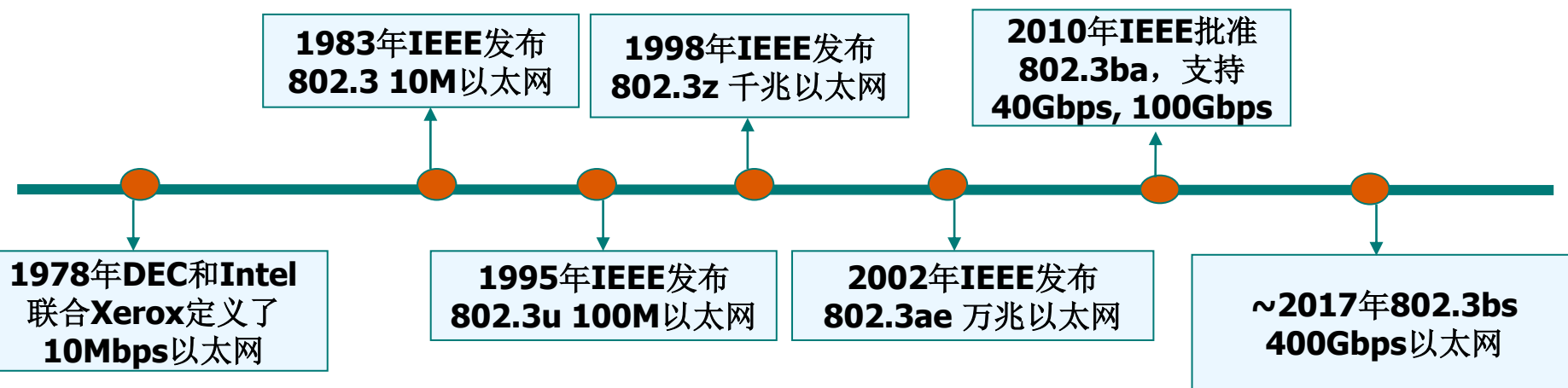
(b)



Chapter 5 LAN & MAC Sub layer

- 5.1 传统LAN的基本概念
- 5.2 多路访问协议
- 5.3 以太网的MAC协议
- 5.4 无线LAN
- 5.5 网桥
- 5.6 交换式局域网

发展历程



注意：一般使用时常常不区分“以太网”和“IEEE 802.3”



传输介质

- **10base-T**
 - **10**表示速率为**10Mbps**
 - **base**表示使用基带信号传输
 - **T**表示传输介质是双绞线
- **10Base-F**表示传输介质是光纤
- **100Base-TX**
 - **100**表示速率为**100Mbps**
 - **TX**表示传输介质为双绞线，通信模式为全双工

由于以太网使用的传输介质的误码率很小，因此以太网的LLC一般提供的是无确认的非连接服务



以太网

- 5.3.1 经典以太网
- 5.3.2 快速以太网
- 5.3.3 千兆以太网
- 5.3.4 万兆以太网
- 5.3.5 以太网帧格式



以太网

- 5.3.1 经典以太网
- 5.3.2 快速以太网
- 5.3.3 千兆以太网
- 5.3.4 万兆以太网
- 5.3.5 以太网帧格式



概述

- 经典以太网在总线型共享链路上以**10Mbps**的速率工作，也称为**10M**以太网，对应标准为**802.3**
- 传输介质
 - 同轴电缆（**10Base5**，**10Base2**，已经很少使用了），双绞线（**10Base-T**）及光纤（**10Base-F**）
- **MAC**层使用**CSMA/CD**来实现多台主机竞争访问共享链路

CSMA/CD: Carrier Sense Multiple Access with Collision Detection，带冲突检测的载波侦听多路访问

共享链路问题(1)

MAC
Frame



问题1: 在共享链路上, 任何主机都能够接收其它主机发送的数据, 因此需要一种机制来指明数据的接收者

解决方法: 主机发送数据时以帧为单位。数据帧的头标中包含了接收这个数据帧的目的地址, 即目的MAC地址

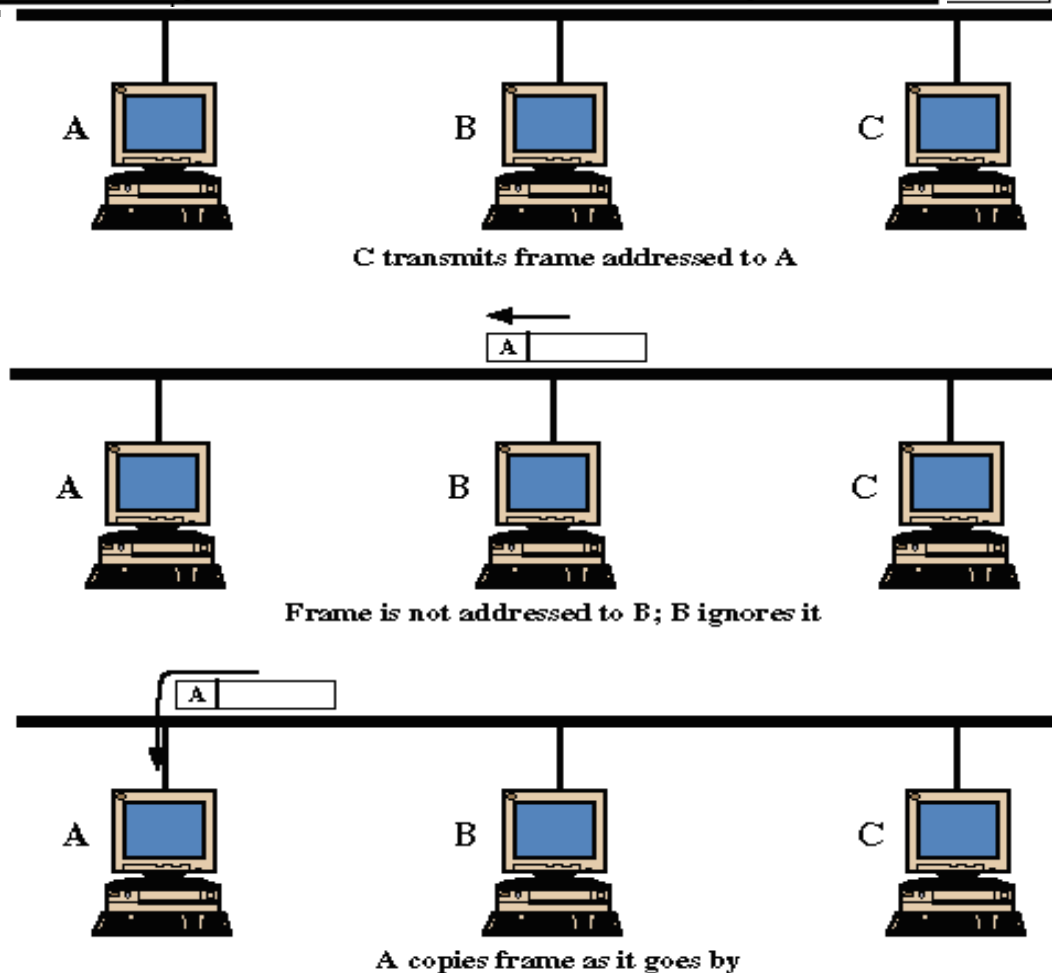


Figure 9.4 Frame Transmission on a Bus LAN



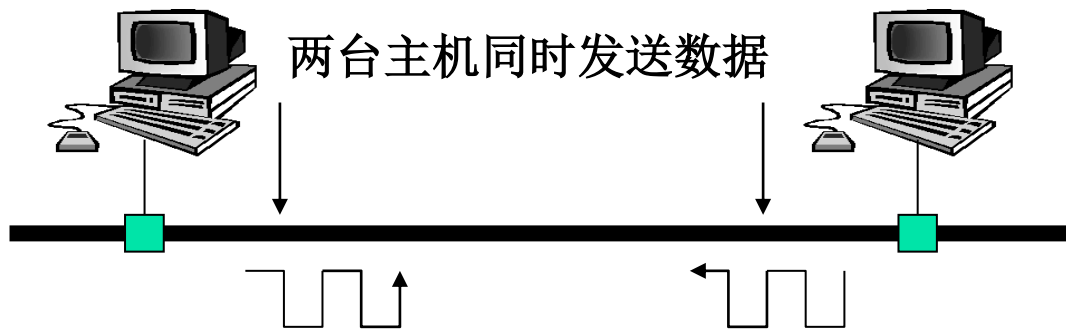
共享链路问题(2)

问题2：在共享链路上，需要一种机制协调所有主机之间的数据发送，以公平竞争访问链路

解决方法：每个数据帧都有最大长度限制（数据域部分不超过**1500**字节），发送完后主机之间要重新使用**CSMA/CD**机制协调对共享链路的访问

碰撞

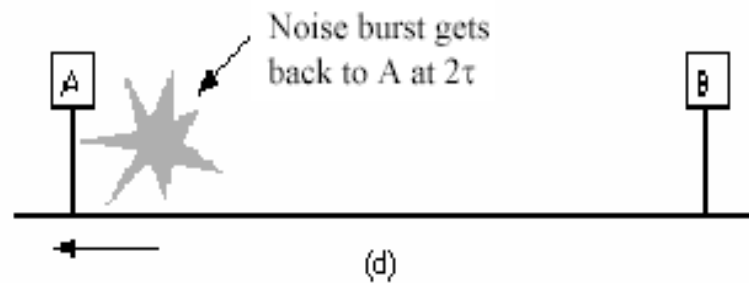
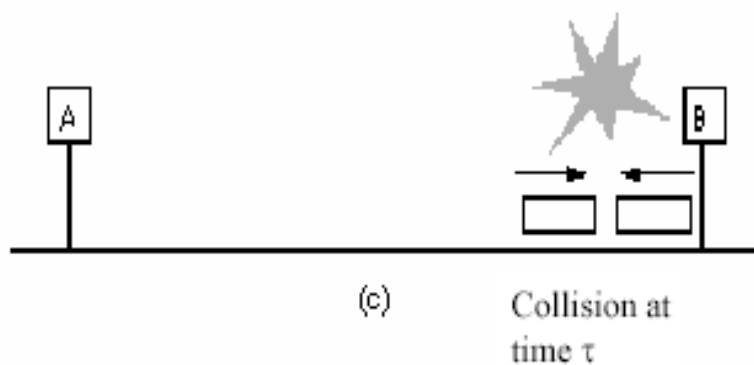
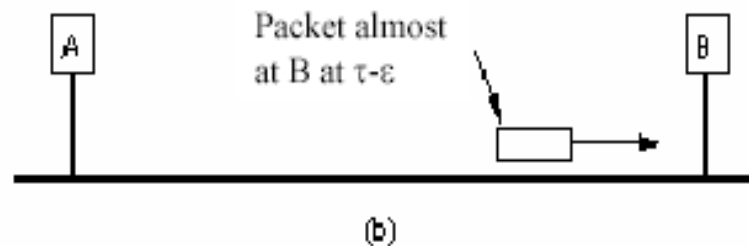
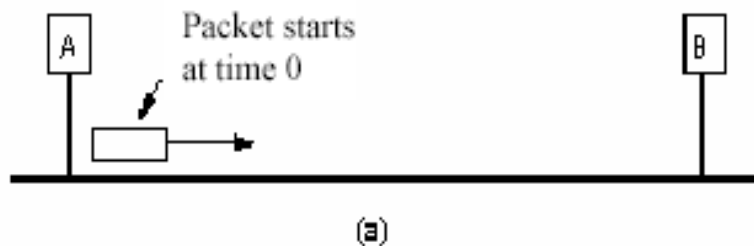
- 碰撞（**Collision**），也叫做冲突，由多台主机争用传输介质引起的。当连接在共享链路上的两台或两台以上主机同时发送数据时，表示这些数据的信号将在同一段传输介质上叠加，从而导致无法被接收主机正确接收



- 主机边发送数据边检测信道上电压大小，如果信号电压摆动值超过一定的门限值时，就认为传输介质上至少有两台主机同时在发送数据，表明产生了碰撞

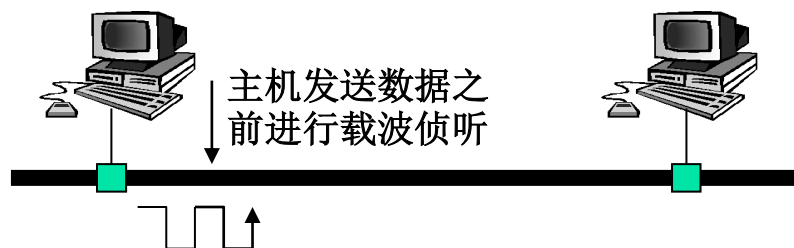
碰撞检测

碰撞域：不同站点同时发送数据可能产生碰撞的最大网段

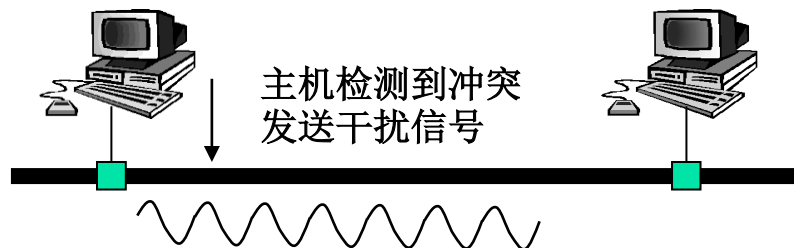


CSMA/CD 步骤

- 载波侦听 (**Carrier Sense**)：当某台主机想要发送数据时，它首先侦听介质是否忙，如果是，表示当前有其它主机正在传输数据，因此将继续侦听，直到介质空闲，然后发送数据



- 冲突检测 (**Collision Detection**)：如果主机在数据发送过程中检测到冲突，则发送干扰信号 (**jamming signal**)，以确保所有主机都知道发生了冲突，并且取消发送

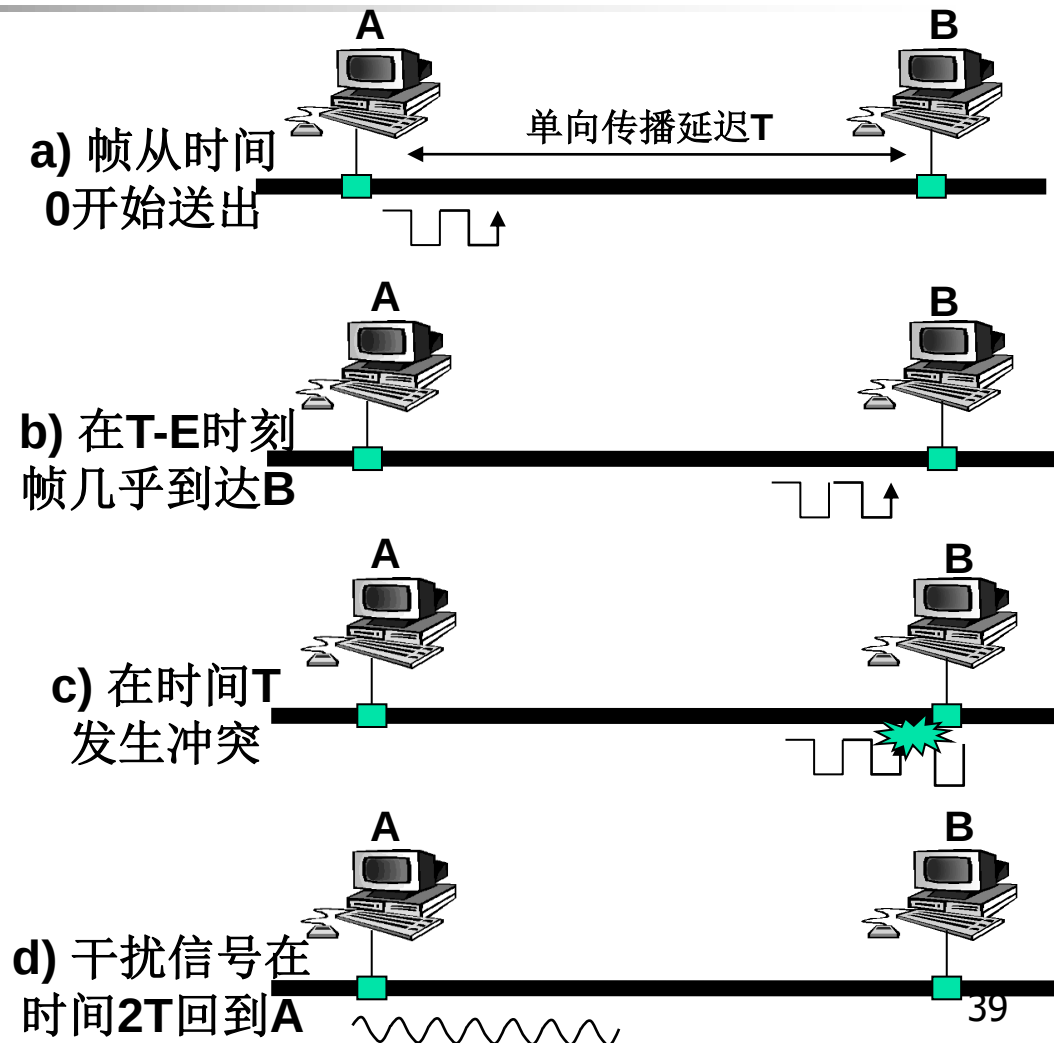


CSMA/CD机制—传播延迟

冲突检测过程还与数据在传输介质上的传播延时相关，检测到冲突的必要条件是当干扰信号到达发送方的时候发送方必须还在发送数据

- 最大电缆长度限制：**2500米**
- 最小帧长限制：**64字节**

往返传播延迟 $2T=51.2\mu s$ ，发送1比特延迟 $=100ns$
由帧发送延迟 $>2T$ 得到最小帧长64字节





CSMA/CD机制—退避算法

- 当主机检测冲突后，它将等待一个随机的时间，也称为退避（**backoff**）
- 二进制指数退避算法（**binary exponential backoff**）来确定冲突发生后的随机等待时间
 - 在一次冲突之后，时间被划分为离散的时槽，其长度等于最差情况下在以太网介质上往返传播所需要的时间，时槽与以太网所允许的介质最大长度有关，一般设置为**51.2us**
 - 当第**k**次冲突发生后，从离散整数集合 **$\{0, 1, \dots, 2^k-1\}$** 中随机地选择一个数**r**，然后等待**r**个时槽。因此对于第一次冲突，就是等待**0**或者**1**个时槽后再次尝试发送，对于第二次冲突，随机选择**0, 1, 2**或者**3**，然后等待这么多个时槽。等等
 - 如果 **$k \geq 10$** ，则保持离散整数集合 **$\{0, 1, \dots, 2^{10}-1\}$** （**1023**）不变，从其中选择一个随机数**r**，然后等待**r**个时槽
 - 如果 **$k > 16$** ，也就是说尝试**16**次后仍然没有成功发送，则主机放弃发送，并且产生一个错误



以太网

- 5.3.1 经典以太网
- 5.3.2 快速以太网
- 5.3.3 千兆以太网
- 5.3.4 万兆以太网
- 5.3.5 以太网帧格式

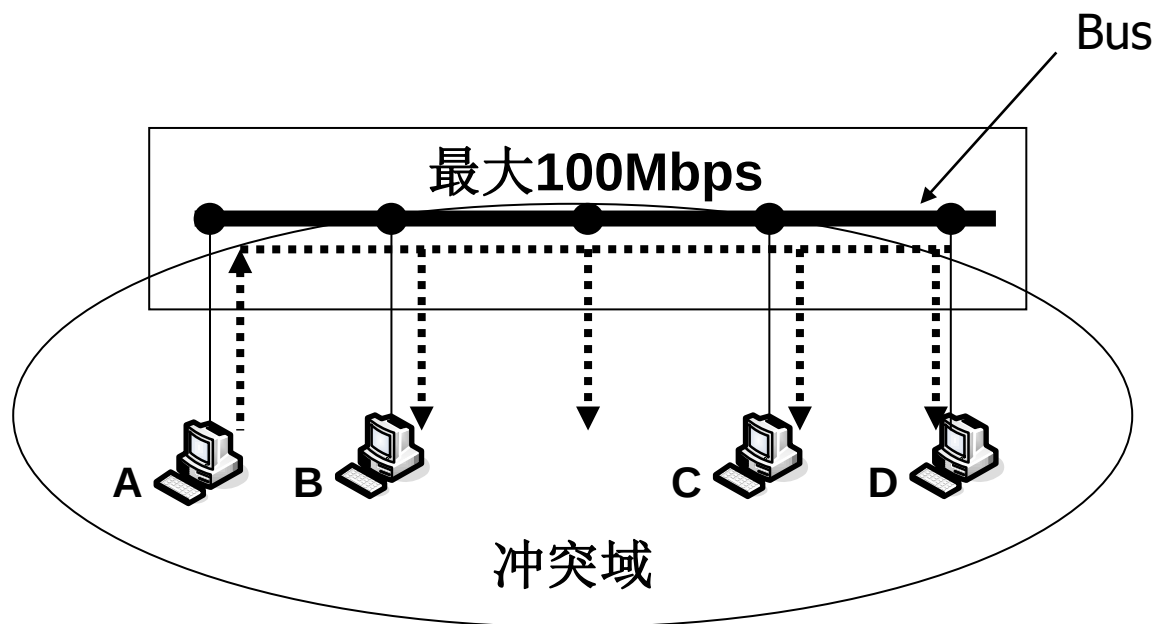


概述

- **100Mbps**以太网也称为快速以太网，对应的标准为**802.3u**
 - 向前兼容**802.3**，保留原来**802.3**的帧格式、接口和过程规则，只是将发送**1bit**所需的时间从**100ns**降低到**10ns**，因此电缆的最大长度也降低到原来的十分之一，即**250米**
- 传输介质
 - 双绞线（**10Base-TX**，全双工，非屏蔽**5**类双绞线**UTP**）、光纤（**100Base-FX**）
- 两种工作模式
 - 主机通过集线器互连（**Hub**）
 - 主机通过二层交换机互连（**Switch**）

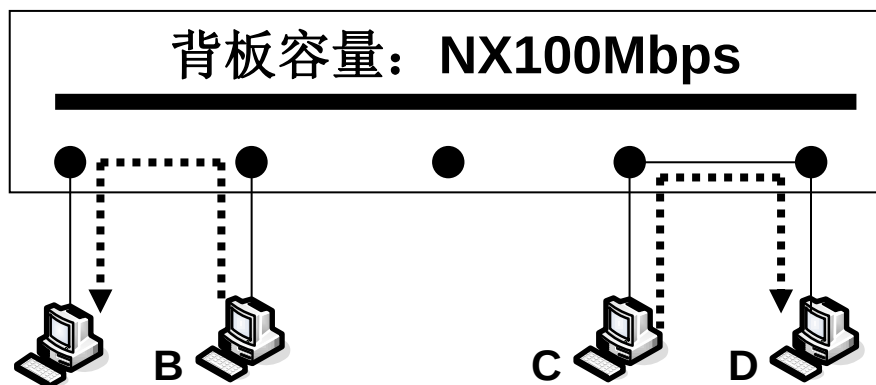
集线器互连

- **CSMA/CD**机制：集线器的所有端口在逻辑上都连在一起，构成一个冲突（碰撞）域
- 半双工通信：在同一时刻只有一台主机可以发送数据



交换机互连

- 不需要使用**CSMA/CD**机制：连接在交换机上的每台主机都构成了一个单独的冲突域
- 全双工：发送和接收过程是分开的
 - 例如**100Base-TX**，有两对双绞线，一对用于发送，一对用于接收



交换机中，每个输入的数据帧都被缓存起来，然后根据目的**MAC**地址通过高速背板被转发到相应的输出端口

目前快速以太网主要使用二层交换机互连



以太网

- 5.3.1 经典以太网
- 5.3.2 快速以太网
- 5.3.3 千兆以太网
- 5.3.4 万兆以太网
- 5.3.5 以太网帧格式



概述

- 千兆以太网比快速以太网快**10**倍，保持与现有以太网标准的向前兼容，对应的标准为**802.3z**
- 传输介质
 - 双绞线（**1000Base-T**，标准的非屏蔽**5**类双绞线**UPT**，**1000Base-CX**，屏蔽双绞线）、光纤（**1000Base-SX**，多模光纤，**1000Base-LX**，单模或者多模光纤）
- 两种工作模式
 - 主机交换机互连，全双工（**Switch**）
 - 主机通过集线器互连，半双工（**Hub**）



交换机互连

- 不需要使用**CSMA/CD**机制：连接在交换机上的每台主机都构成一个单独的冲突域，不会和连接在其它端口上的主机冲突
 - 电缆的最大长度由信号的强度决定
- 全双工：主机可以同时发送和接收数据

绝大多数的千兆以太网使用交换机互连



集线器互连

问题：发送1比特所需时间为1ns，为检测到冲突，如果使用64字节最小帧长，电缆最大长度降为25米！

解决方法，将电缆最大长度扩展到200米

- 1) 载荷扩充 (**Carrier extension**)，它的本质是让硬件在普通的帧后面增加一些填充数据，以便将数据帧的长度扩充到512字节，这种填充对于上层软件是透明的
- 2) 帧串 (**frame bursting**)，它允许发送方将多个帧连接在一起，把它们串起来一起传输出去。如果整个串小于512字节，则硬件会再次对它们进行填充。

半双工通信：连接在集线器上的所有主机构成一个冲突域



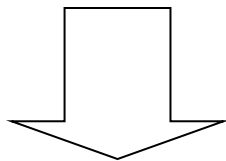
以太网

- 5.3.1 经典以太网
- 5.3.2 快速以太网
- 5.3.3 千兆以太网
- 5.3.4 万兆以太网
- 5.3.5 以太网帧格式



概述

- 万兆以太网工作速率为**10Gbps**，对应的标准为**802.3ae**
- 传输介质
 - 万兆以太网的传输介质只是使用光纤，包括长波和短波
- 传输距离增强
 - 从**300m**到**40km**，可以用来构建**MAN**和**WAN**
- 只支持全双工通信



- 更高速以太网：**40Gbps**、**100Gbps**、**400Gbps**、.....

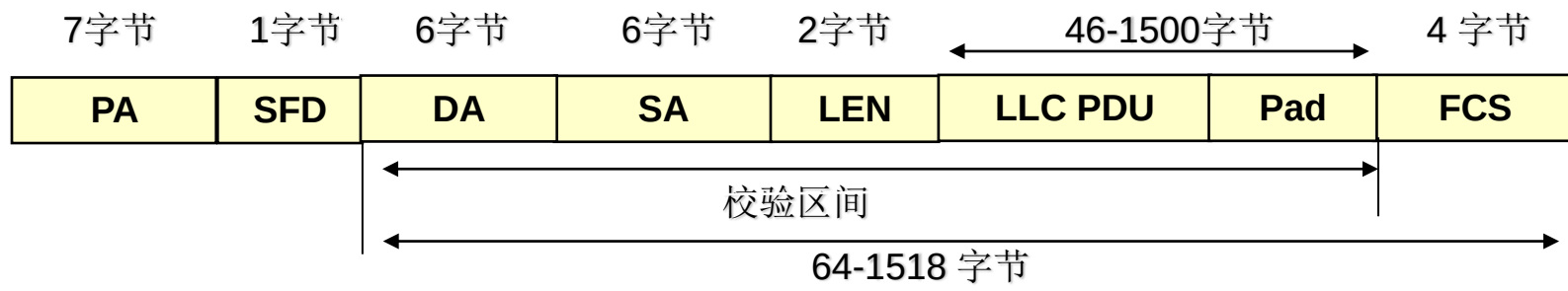


以太网

- 5.3.1 经典以太网
- 5.3.2 快速以太网
- 5.3.3 千兆以太网
- 5.3.4 万兆以太网
- 5.3.5 以太网帧格式

MAC帧格式

- 所有类型的以太网具有相同的**MAC**帧格式



PA: 先导域 - 10101010序列, 用于使接收方与发送方同步

SFD: 帧起始定界符 -- 10101011

DA: 目的MAC地址; SA: 源MAC地址

LEN: 数据长度 (数据部分的字节数) (0-1500B)

大于1500标识高层协议 (0X0800,上层使用IP)

LLC PDU+pad – LLC协议数据单元加填充, 最少46字节, 最多1500字节

Pad: 填充字段, 保证帧长不少于64字节(0~46字节, 若数据域 ≥ 46 字节, 则无Pad)

FCS: 帧校验序列 (CRC-32)

MAC地址

Cisco Systems, Inc.

00:05:9A:2C:7B:00

24 bits

24 bits

ccccccug cccccccc cccccccc

xxxxxxxx xxxxxxxx xxxxxxxx

IEEE administered company ID

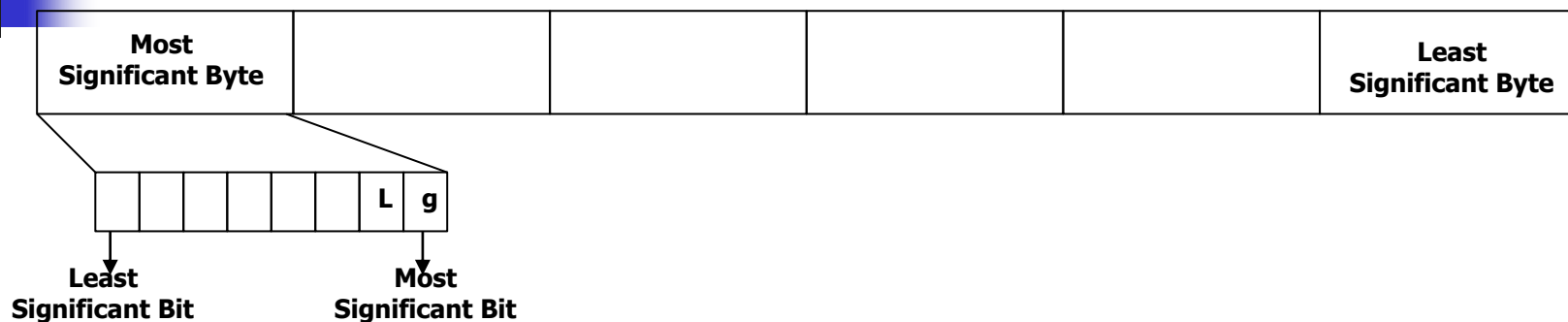
Manufacturer selected extension ID

MAC地址前24比特由IEEE统一分配

U/L：全局(0)还是本地管理(1)
I/G：单播(0)还是组播(1)
对于一般的802网络接口，**U/L**
和**I/G**比特都设置为0

1. **MAC**地址用于链路寻址，只需要在链路范围内唯一
2. **MAC**地址一般由厂商固化在网络接口设备内，很少变化，也称为硬件地址

组播和广播



- 组播（**Multicast**）：MAC地址的最高有效位如果为**0**，表示的是普通的地址，如果为**1**，则表示组地址。利用组地址，在以太网内，可以将一个数据帧发给加入了这个组的多台主机，这种数据发送方式也称为组播
- 广播（**Broadcast**）：全**1**的MAC地址（**FF:FF:FF:FF:FF:FF**）被称为广播地址。如果数据帧的目的地址为广播地址，那么以太网内所有的主机都会接收该帧



Chapter 5 LAN & MAC Sub layer

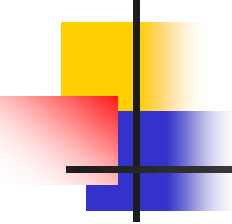
- 5.1 传统LAN的基本概念
- 5.2 多路访问协议
- 5.3 以太网的MAC协议
- 5.4 无线LAN
- 5.5 网桥
- 5.6 交换式局域网



无线局域网协议标准

- 无线局域网**WLAN (Wireless Local Area Network)**
 - 无线电波按照工作的频率的不同被划分为信道，每个信道对应着一个频率范围(频段)，**WLAN**通信是建立在无线信道的基础上的
- **WLAN标准**
 - **802.11**系列标准：**IEEE802.11**工作组制定
 - **Hiper LAN**系列标准：**ETSI**的**BRAN**小组制定

下面我们的讨论都是基于**802.11**系列标准的无线局域网



Wi-Fi



Wi-Fi CERTIFIED™ makes it Wi-Fi.

■ Wi-Fi

- 最初是**Wi-Fi**联盟（**Wi-Fi Alliance**）的注册商标，用来描述基于**802.11**系列协议的无线局域网技术
- 现在也被用来描述**WLAN**中移动设备的无线接口

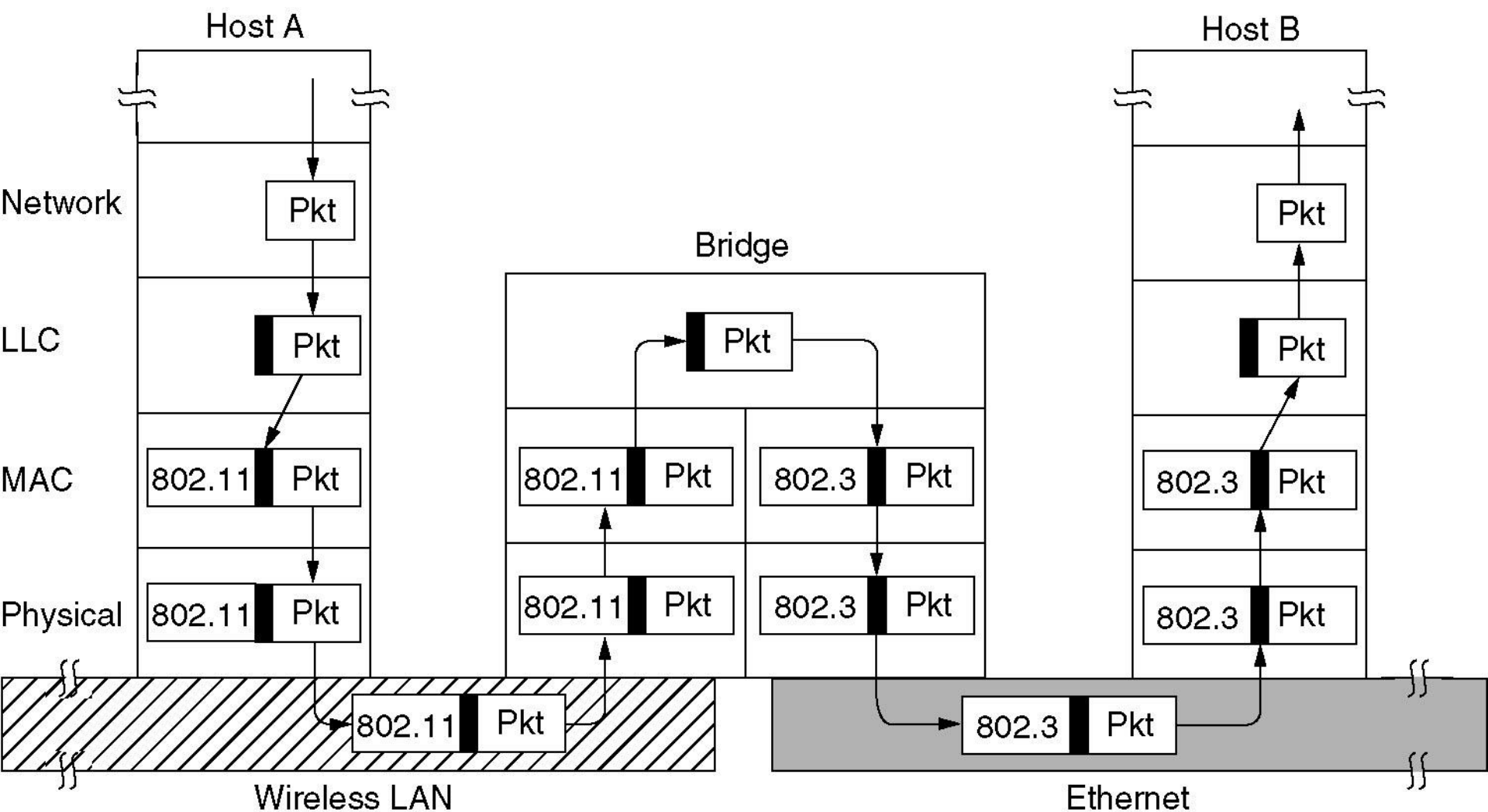
■ Wi-Fi联盟

- 由工业界发起成立的一个全球性的非盈利组织，负责测试和认证兼容**IEEE 802.11**系列协议的设备，并且对于通过测试和认证的设备打上**Wi-Fi**认证标志

网络组成

- 无线站点**STA**（**wireless STAtion**）
 - 配置支持**802.11**协议的无线网卡的终端
 - **STA**通过**MAC**层机制来共享同一个无线信道，最简单的**WLAN**仅仅由**STA**组成，它们之间直接通信
- 无线接入点**AP**（**wireless Access Point**），有时也被称为基站**BS**（**Base Station**）
 - 数据链路层设备
 - **AP**为**STA**之间的通信提供帧转发功能





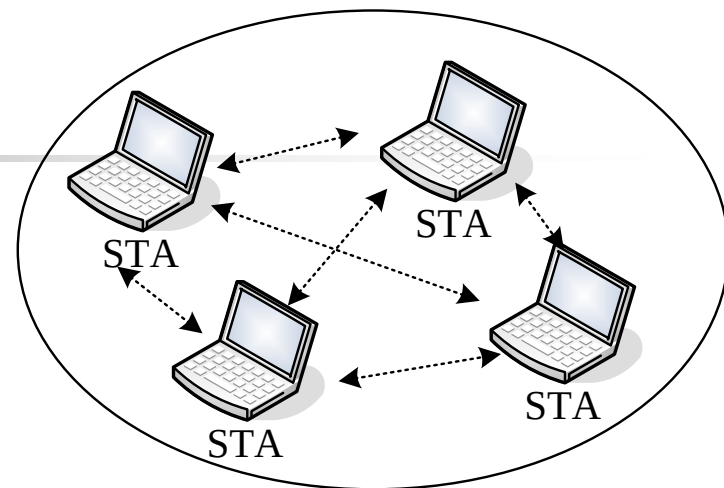
在无线局域网中，**AP**一般同时有**802.3**和**802.11**接口，实现网桥功能，包括

- 1) 不同帧格式转换：由于帧格式的不同，需要重新组帧，某些域需要丢弃或产生还需要重新生成校验和。
- 2) 不同的数据速率适配：由于**802.11**速率低，网桥需要用缓冲区存储来不及传输的帧

网络拓扑

■ Ad Hoc网络，也称为自组织网络

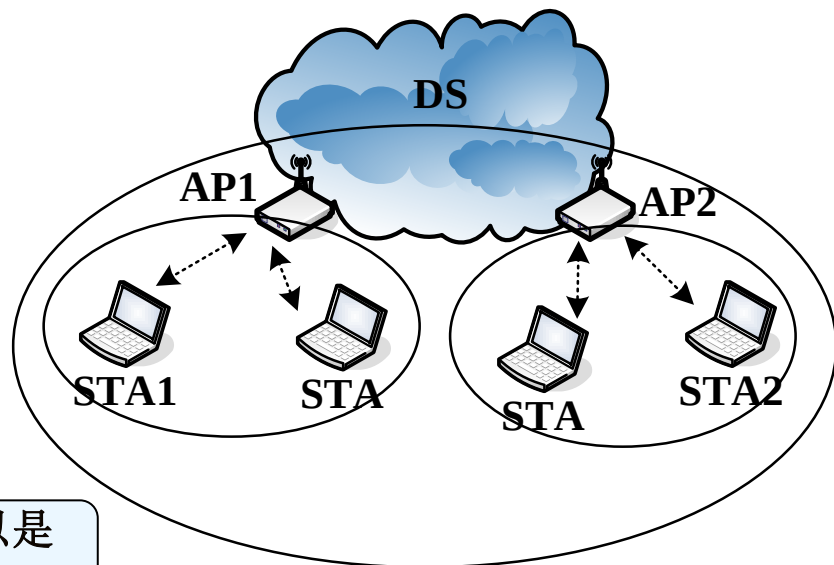
- 仅由**STA**之间通过无线传输介质组成的网络，这些**STA**在彼此无线信号覆盖范围内，因此之间能够互相通信



AD Hoc网络

■ 基础设施网络

- 在基础设施网络中，**STA**之间不能直接通信，所有的通信都必须经过**AP**
- 每个**AP**都配置了一个网络名（**SSID**），**STA**必须配置和**AP**相同的网络名



基础设施网络

DS (Distributed System): 分布式系统，可以是一个交换机，一个有线网络或者一个无线网络

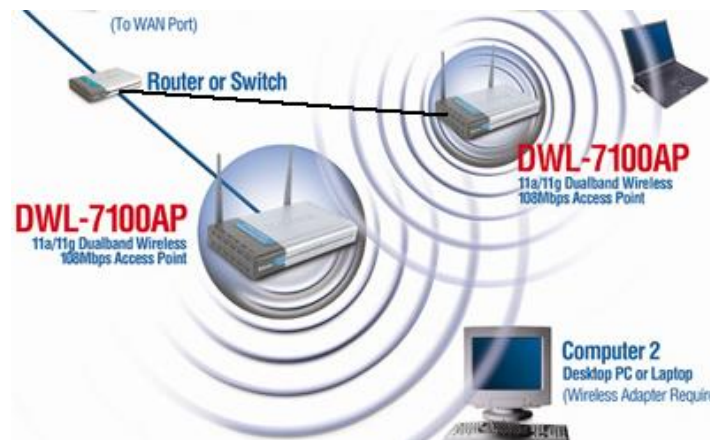
无线局域网特点

■ 优势

- 快速部署（**deployment**）：不需要进行繁琐的布线工作
- 支持移动漫游（**roaming**）：在无线信号覆盖范围内，用户可以随意的移动漫游
- 易于扩展（**scalability**）：根据应用的需求来配置相应的网络拓扑结构

■ 问题

- 传输速率和效率
- 移动终端功耗
- 安全性



发展历程

802.11 history

1997年IEEE发布第一个802.11标准, 工作在**2.4GHz**的**ISM**频段, 支持速率为**1M**或者**2M**, 室内约**25米**, 室外约**75米**

1999年IEEE发布802.11a, 工作在**5GHz**的**ISM**频段, 支持最大速率为**54M**, 室内约**30米**, 室外约**100米**

1999年IEEE发布802.11b, 工作在**2.4GHz**的**ISM**频段, 支持最大速率为**11M**, 室内约**35米**, 室外约**110米**

2003年IEEE发布802.11g, 工作在**2.4GHz**的**ISM**频段, 支持最大速率为**54M**, 室内约**35米**, 室外约**110米**

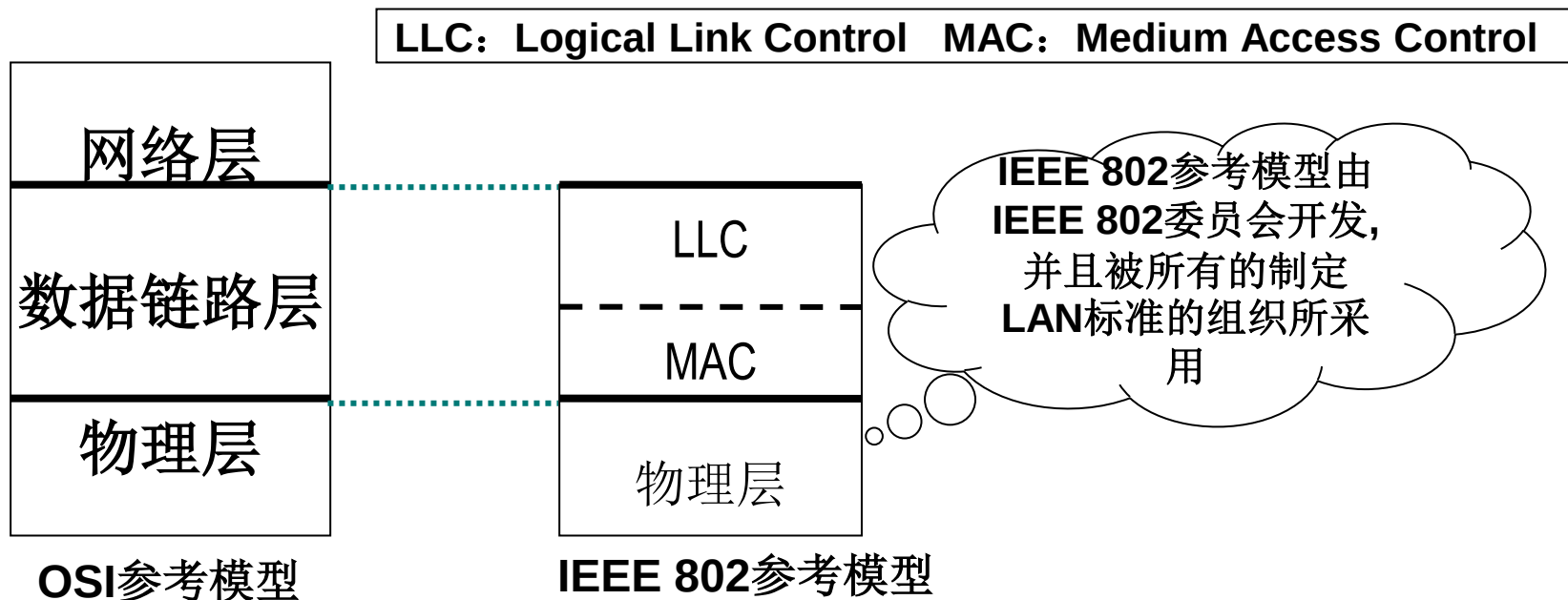
2009IEEE发布802.11n, 工作在**2.4GHz/5GHz**的**ISM**频段, 支持最大速率为**150M**, 室内约**70米**, 室外约**250米**

2014年1月IEEE批准802.11ac, 工作在**5GHz**的**ISM**频段, 支持最大速率为**866.7M**

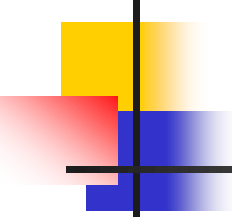
注意: 如果没有特别说明, 一般用“802.11”来表示“802.11系列协议”

体系结构

采用IEEE802参考模型



- 所有802系列协议具有相同的MAC子层, 不同的是物理层
- 所有的802.11系列协议的制定都是以最初发布的802.11协议为基础, 因此, 802.11b/a/g/n实际上都不是一个独立的协议, 它们是对最初802.11协议的修订和增强



信道划分

- **802.11**工作在两个**ISM**频段（非授权频段），即**2.4GHz**频段和**5GHz**
- 对于**802.11b**整个**2.4G**频段被划分为**14**个信道，每个信道的带宽为**22MHz**
- 在**802.11**中，每个站点或者**AP**使用某个信道，也被称为工作信道

信道	美国和加拿大 (MHz)	欧洲 (MHz)	中国和日本 (MHz)
1	2412	2412	2412
2	2417	2417	2417
3	2422	2422	2422
4	2427	2427	2427
5	2432	2432	2432
6	2437	2437	2437
7	2442	2442	2442
8	2447	2447	2447
9	2452	2452	2452
10	2457	2457	2457
11	2462	2462	2462
12	X	2467	2467
13	X	2472	2472
14	X	X	2484



信道分配

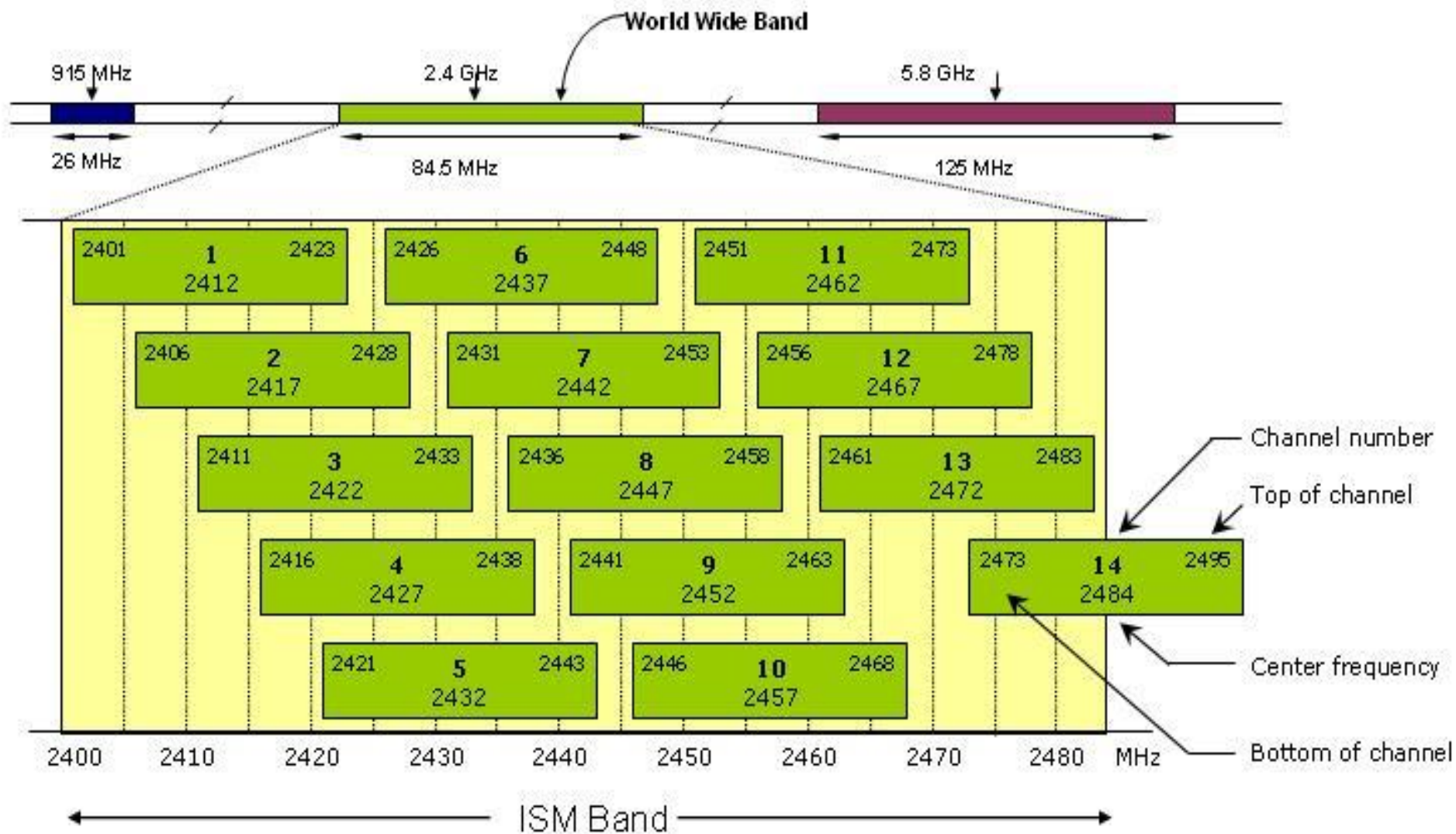
- 对于**Ad Hoc**网络，要求所有的**STA**都工作在同一个信道
- 对于基础设施网络，**STA**必须和其关联的**AP**工作在同一个信道

信道分配问题：

如何在同一个区域部署多个**Ad Hoc**网络或者多个**AP**，使得它们之间的干扰尽量小？

访问控制问题：

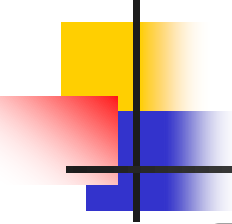
如何控制多个**STA**对同一个信道的访问？



为了避免信道间干扰，同一区域内的**Ad Hoc**网络或**AP**的工作信道不能重叠，中心频率间隔最少为**25MHz**，因此同一区域最多只能有**3**个工作在不同信道的**Ad Hoc**网络或**AP**

11个信道：每个信道占用**22 MHz**，有且只有**1**个信道集合中有**3**个互不重叠的信道{**1, 6, 11**}

14信道：每个信道占用**22MHz**，可划分为**4**个具有**3**个互不重叠信道的信道集合 {**1, 6, 11**}, {**2, 7, 12**}, {**3, 8, 13**}, {**4, 9, 14**}，但每次只能使用一个信道集合



MAC子层

由于无线环境的不可靠性，**802.11**在**LLC**子层使用确认无连接的服务。**LLC**子层对于所有基于**802.11**参考模型的协议都是一样的，我们将主要关注**MAC**子层的功能

- **MAC子层功能**
 - 无线信道的访问控制
 - 无线站点的移动管理
 - 寻址
 - 差错控制



访问控制—概述

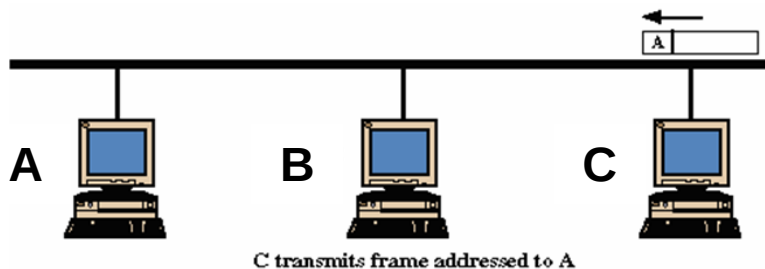
- **MAC子层功能**
- 确定由多个无线站点共享的无线信道的使用方式，也就是当信道空闲时，决定下一个该由谁传输数据
- 信道的传输能力被划分为一个个时槽（**time slot**），由访问控制来决定每个时槽由哪个无线站点或者接入点使用
- 采用带冲突避免的载波侦听多路访问机制（**CSMA/CA**）

访问控制—CSMA/CA

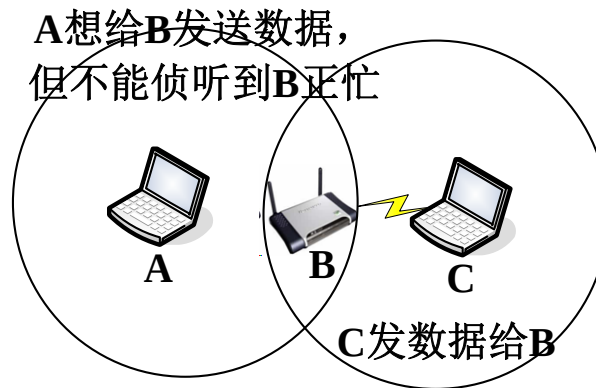
■ Why CSMA/CA?

- 以太网采用的是带冲突避免的载波侦听多路访问机制 (CSMA/CD)

CSMA/CD VS. CSMA/CA



有线：任何主机传输数据其它主机都能侦听得到



无线：由于覆盖范围有限

- 1) A无法侦听到数据传输--> A CS失败
- 2) C无法侦听到冲突--> C CD失败



访问控制—操作模式

- **802.11中MAC定义了两种操作模式来实现对共享信道的访问**
 - **分布式协调功能（DCF, Contention）**
 - 无任何中心控制手段，所有站点竞争访问信道
 - 强制要求实现的操作
 - **点协调功能（PCF, Contention-Free）**
 - 使用**AP**或者基站来控制所有的通信
 - 可选的操作



访问控制—分布式协调功能

- 分布式协调功能**DCF** (**Distributed Coordination Function**)
 - 带冲突避免的载波侦听多路访问机制**CSMA/CA** (**CSMA with Collision Avoidance**)
 - 冲突避免**CA**而不是冲突检测**CD**，在无线环境下更容易发生冲突，并且很难检测到
 - 两种**CSMA/CA**操作
 - 基本**CSMA/CA**：采用物理信道侦听方法，该操作强制要求必须使用
 - 扩展**CSMA/CA**：采用基于**RTS/CTS**的虚拟信道侦听方法，该操作是可选的，可以根据需要配置

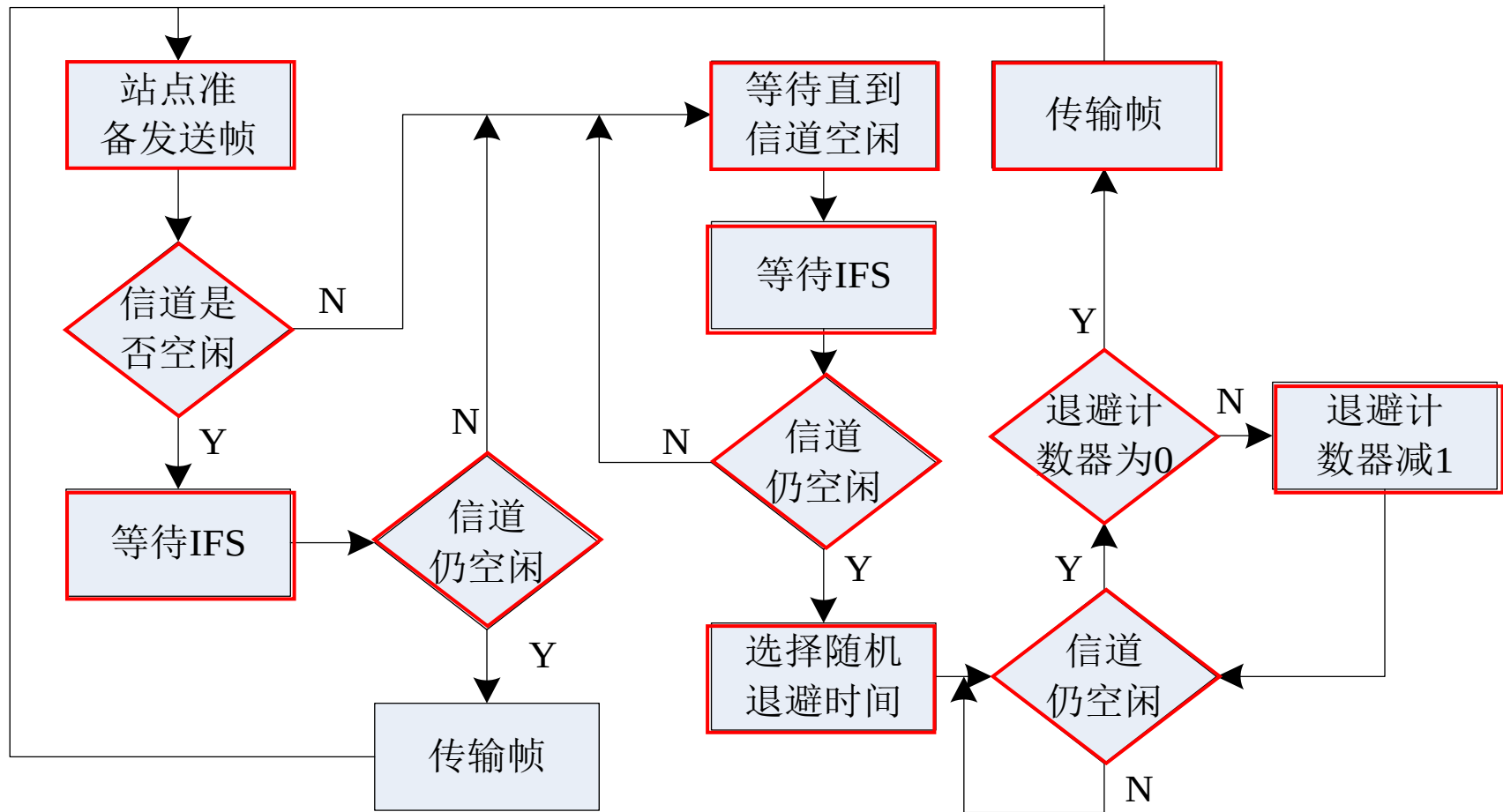
CSMA/CA, CSMA with Collision Avoidance, 载波侦听多路访问机制



基本CSMA/CA（1）

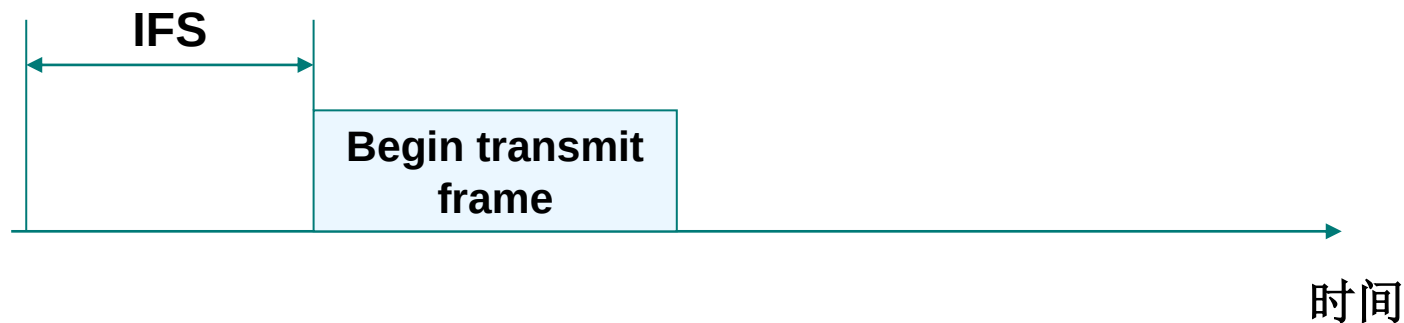
- 包括载波侦听和冲突避免两个过程
 - 载波侦听：站点在发送帧之前侦听无线信道是否空闲，如果是，则进入冲突避免阶段，如果当前信道忙，说明现在有其它站点正在传输数据，则延迟发送帧直到侦听到信道空闲
 - 冲突避免：站点在发送帧之前要先等待一个帧间间隔**IFS**（**InterFrame Spacing**），并且确保在**IFS**时间内信道空闲
 - 为了防止多个站点在等待**IFS**时间后同时发送而导致冲突，与以太网类似，引入了一个随机退避算法来选择一个退避时间（**backoff time**）

具体流程

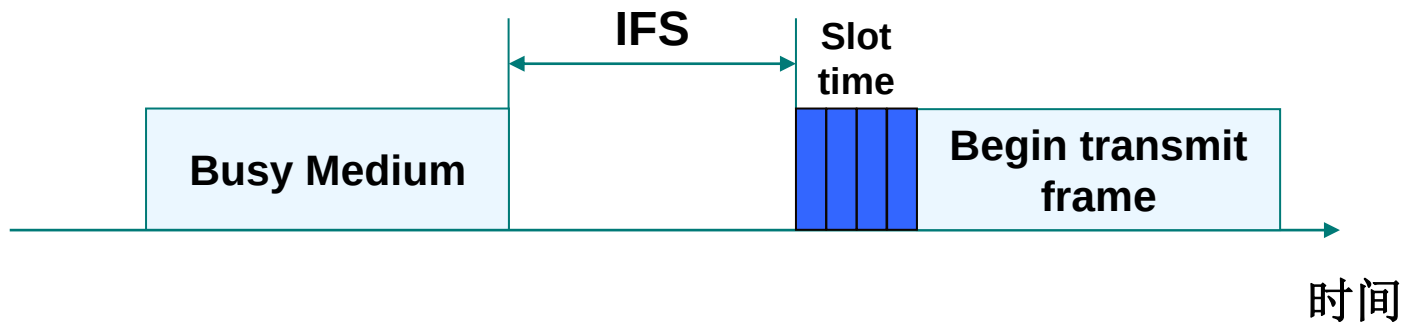


基本CSMA/CA (2)

■ 信道始终空闲



■ 侦听到信道忙



基本CSMA/CA (3)

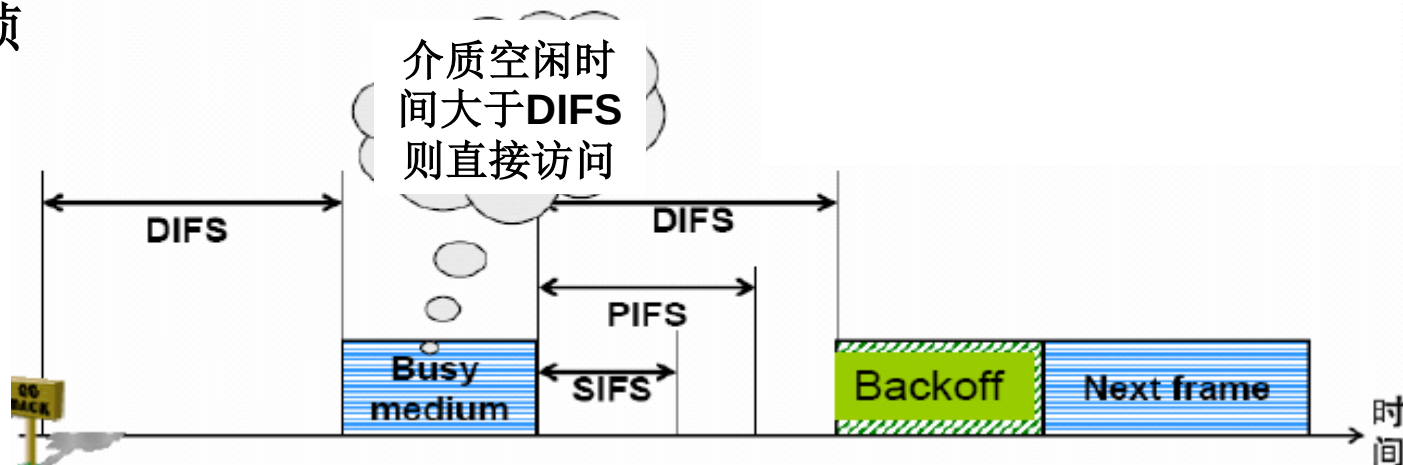
■ 帧间间隔 (**IFS**) 控制了发送帧之前的等待时间

- 不同类型的帧具有不同的**IFS**，确定了帧的发送优先级

IFS越短 $\leftrightarrow$ **帧的优先级就越高**

■ 三种**IFS**

- **SIFS: Short IFS**，最高优先级，用于**CTS**、**ACK**等控制帧
- **PIFS: PCF IFS**，等于**SIFS+1**，中等优先级，用于**PCF**操作模式下的帧
- **DIFS: DCF IFS**，等于**SIFS+2**，最低优先级，用于**DCF**操作模式下的数据帧





基本CSMA/CA（4）

- 站点执行随机退避算法来确定退避时间
 - 从**0**到竞争窗口**CW**之间的随机选择一个值**r**，则退避时间为**r**倍的时槽
- 竞争窗口**CW**（**Contention Window**）采用与以太网类似的指数退避算法来设置
 - **CW**初始化为一个最小值**CWmin**
 - 当发送方认为发送的帧发生冲突时将**CW**加倍增大，直到到达最大值**CWmax**
 - 初始为 2^k-1 ，下一次为 $2^{k+1}-1$ ，



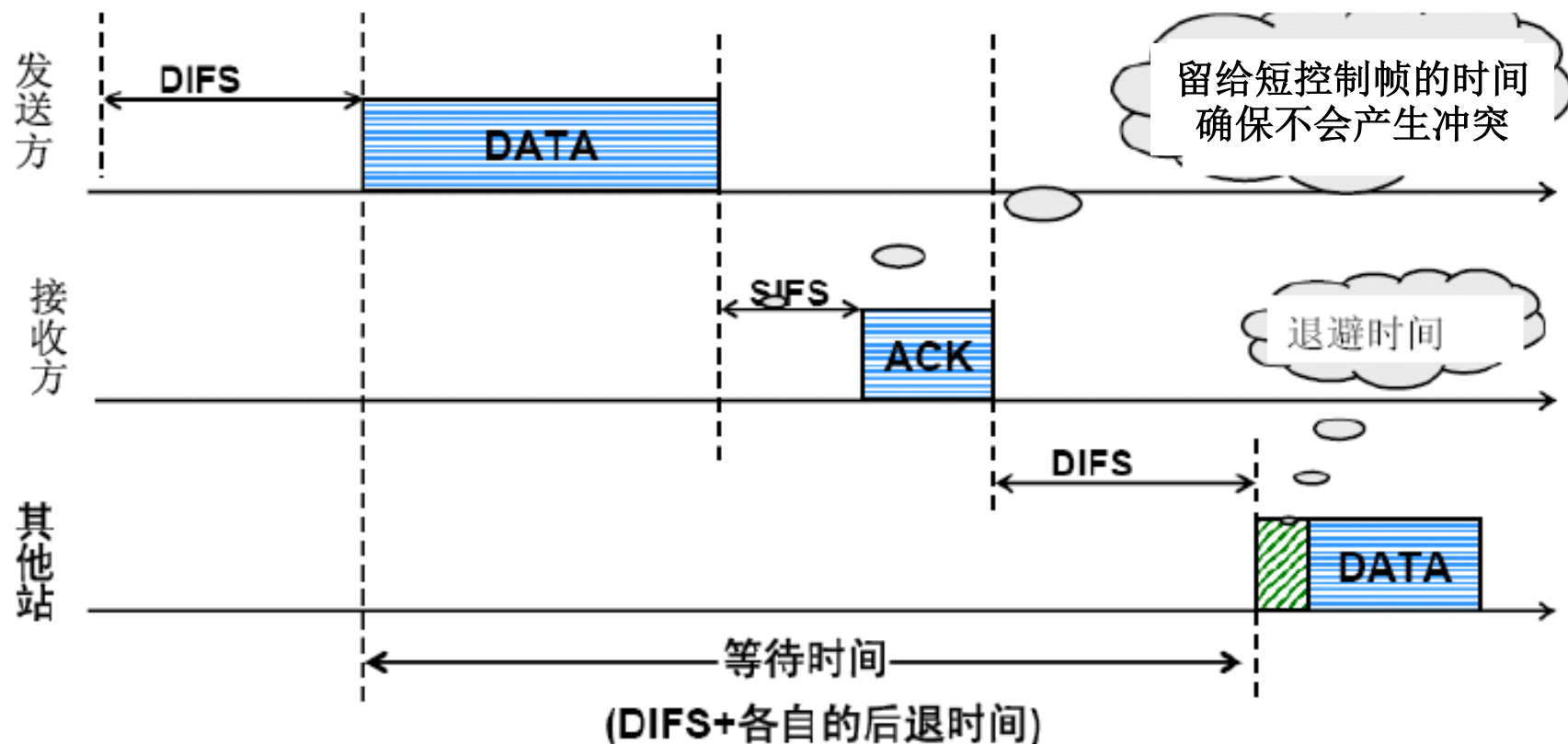
接收方确认

- 接收方发送**ACK**帧
 - 接收方正确接收来自发送方的数据帧后发送**ACK**帧进行确认
 - **ACK**帧不需要执行随机退避算法，而是直接等待一个**SIFS**时间后就发送
- 发送方未收到**ACK**帧
 - 发送方执行指数退避算法，将**CW**的值加倍，并且将重传计数器加**1**，然后重传数据帧，此时需要再次使用**CSMA/CA**机制竞争信道
 - **CW**从**CWmin**开始，每次重传都将**CW**加倍，直到到达**CWmax**
 - 重传计数器到达一个设定的门限值时，发送方将放弃发送数据帧，并且返回一个错误

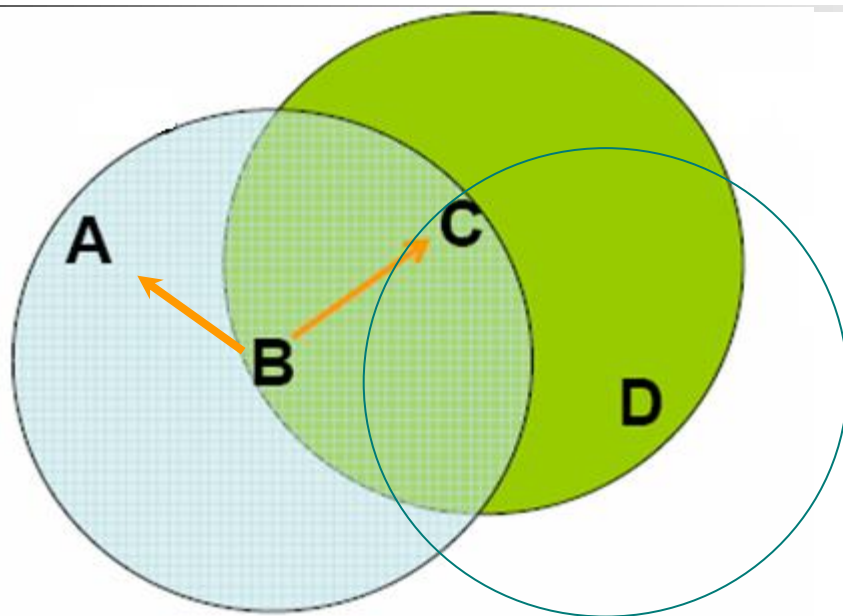
发送方用随机退避算法选择退避时间，也就是说从**0**到**CW**之间选择一个随机数**r**，退避时间即位**r**倍的时槽。而**CW**的确定采用的是指数退避算法，从**CWmin**开始，每次重传都加倍，直到到达**CWmax**

CSMA/CA+ACK

- 发送方在**DIFS**时间内侦听到信道空闲直接发送数据帧，不执行退避算法



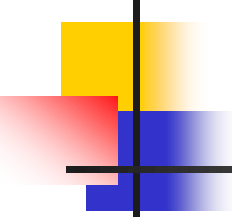
隐藏站点和暴露站点问题



隐藏站点问题（**hidden station problem**）：图中，**B**向**C**发送时，由于**D**听不到误以为可发送数据，造成**C**接收失败

暴露站点问题（**exposed station problem**）：图中，**B**向**A**发送时，**C**听到信道忙误认为它不能向**D**发送数据，实际上并不影响**A**和**D**两站的接收

基本的**CSMA/CA**机制无法解决隐藏站点和暴露站点问题

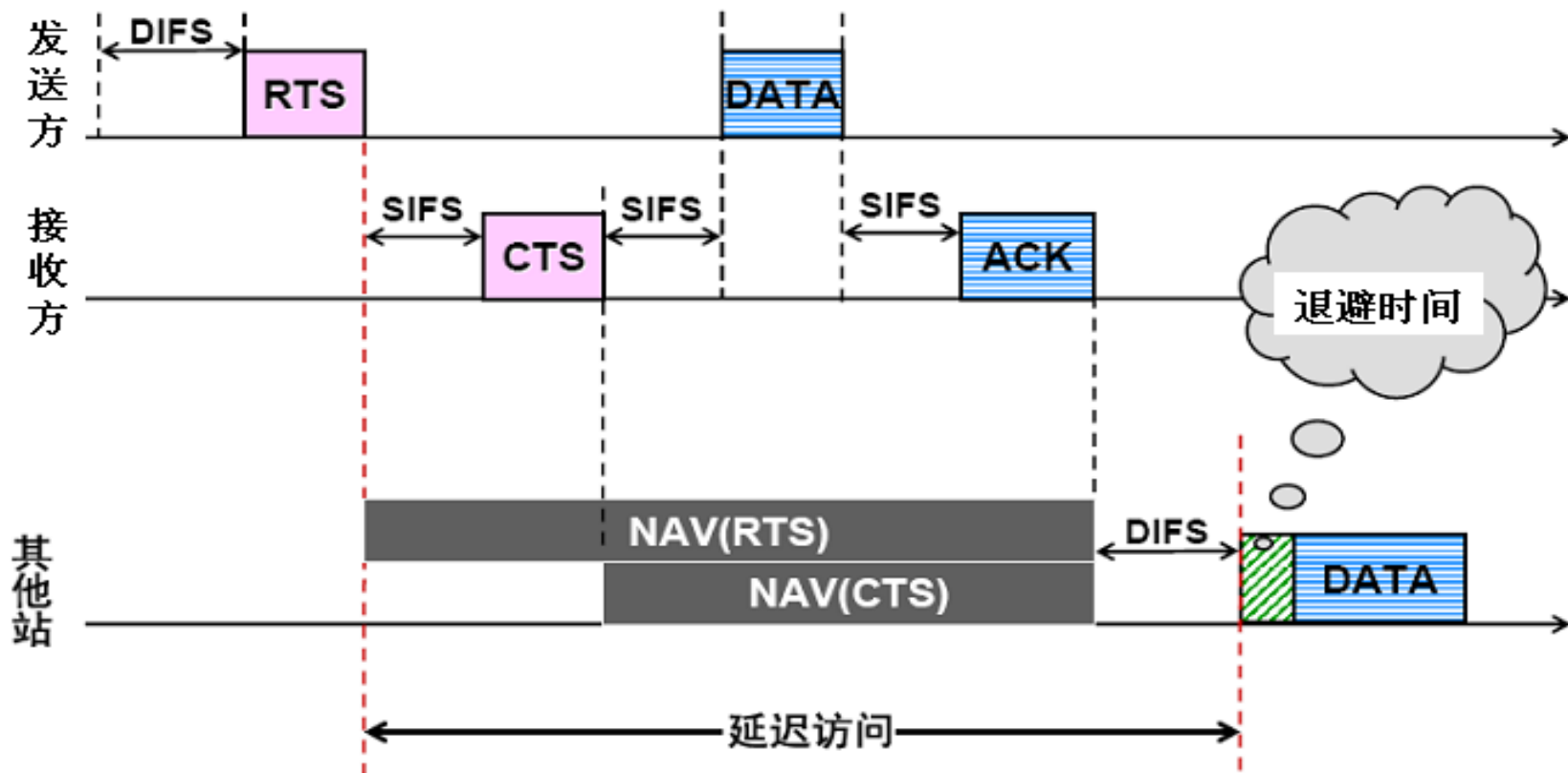


扩展CSMA/CA—原理

- 可选的，但**绝大多数的802.11产品都支持**，可根据需要配置
- 基于**RTS/CTS**的虚拟信道侦听
 - 发送方发送**RTS**帧，其中包含了一个持续时间域，该域的值表明发送方完成帧交换所需要的时间，包括从发送数据帧到接收**ACK**帧所需要的时间
 - 收到**RTS**的站点根据其中的持续时间为自己声明一个虚拟信道，并且该信道正忙，用网络分配向量（**NAV: Network Allocation Vector**）来表示，在**NAV**时间内，该站点不会尝试发送帧
 - 接收方响应**CTS**帧中也包含一个持续时间域，该域的值足够大，以保证发送方能够完成数据帧交换
 - 收到**CTS**的站点根据其中的持续时间为自己声明一个虚拟信道，并且该信道正忙，用网络分配向量（**NAV: Network Allocation Vector**）来表示，在**NAV**时间内，该站点不会尝试发送帧
 - **NAV更新**：只有当**RTS/CTS**中的持续时间域的值大于当前存储的**NAV**时，该**NAV**才会被更新

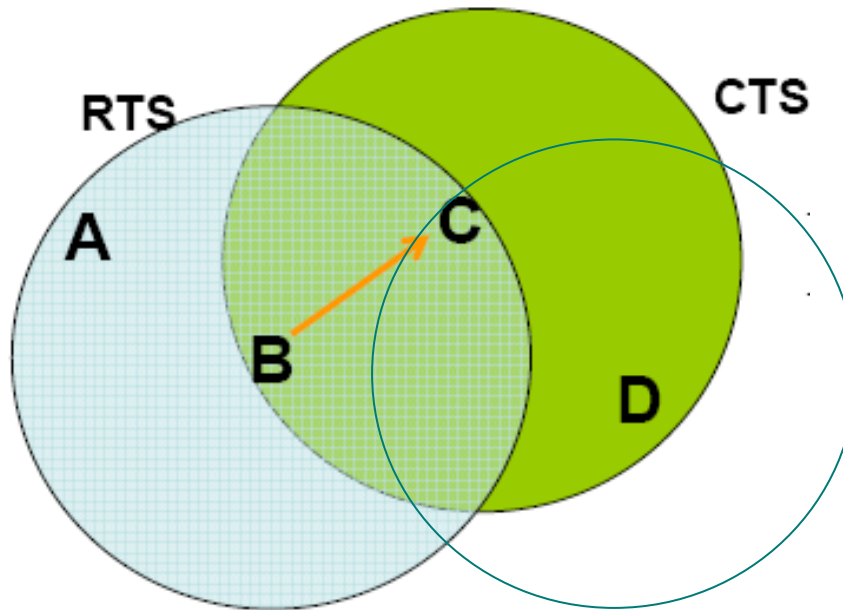
扩展CSMA/CA—过程

1. 发送方像发送一般数据帧一样发送**RTS**帧
2. 接收方等待**SIFS**时间后直接发送**CTS**帧
3. **NAV**减为0后，其它站点就开始使用基本的**CSMA/CA**机制发送数据帧



扩展CSMA/CA—发现隐藏站点

- 侦听RTS
- 设置NAV



- 侦听CTS
- 设置NAV

侦听到RTS--> 发送方附近
侦听到CTS--> 接收方附近

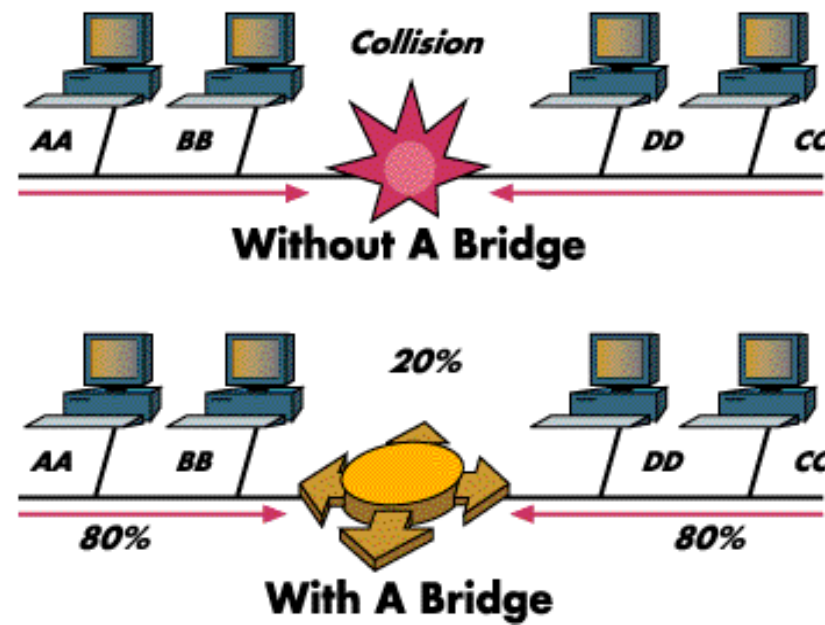


Chapter 5 LAN & MAC Sub layer

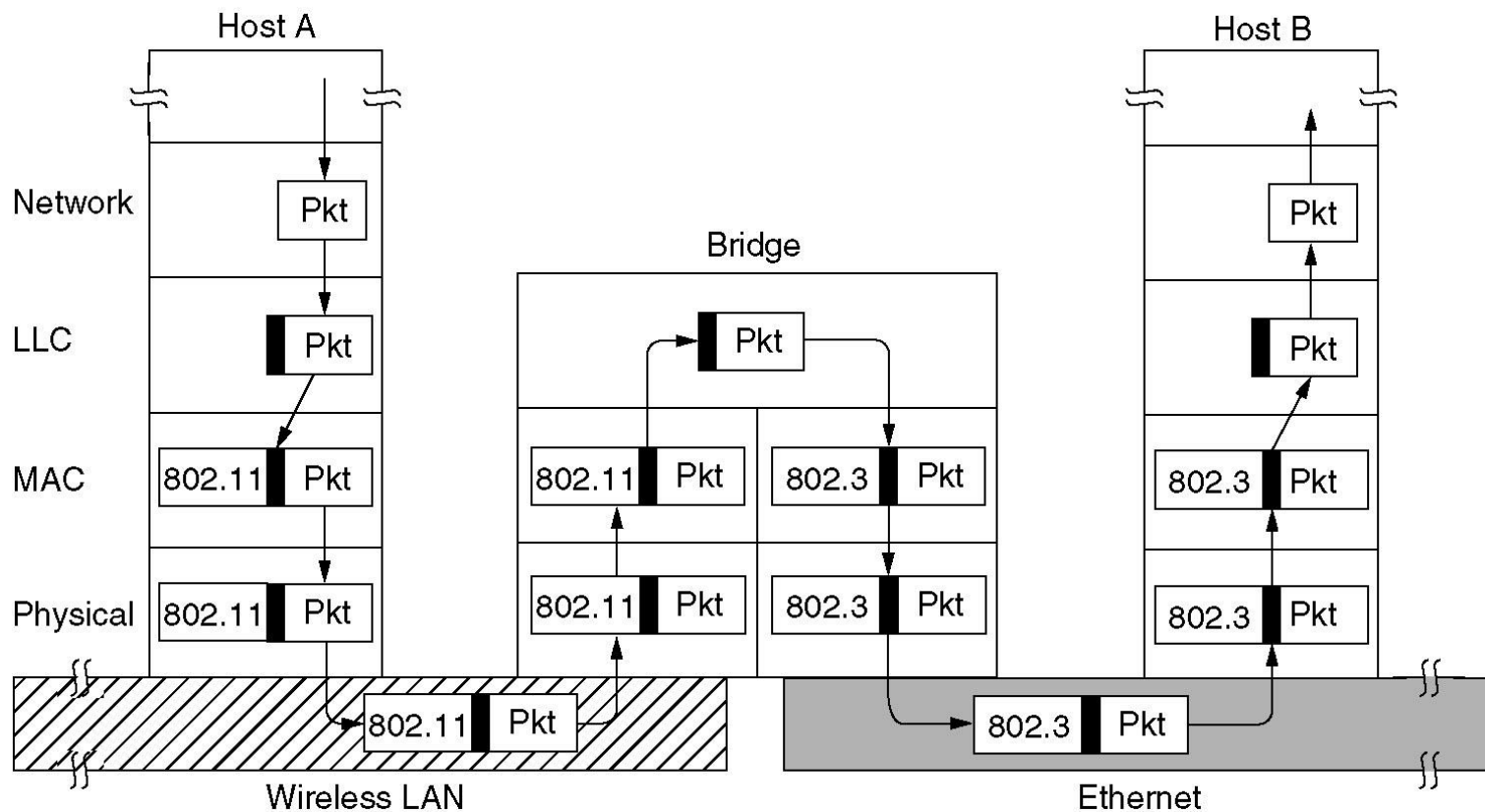
- 5.1 传统LAN的基本概念
- 5.2 多路访问协议
- 5.3 以太网的MAC协议
- 5.4 无线LAN
- 5.5 网桥
- 5.6 交换式局域网

5.5 网桥

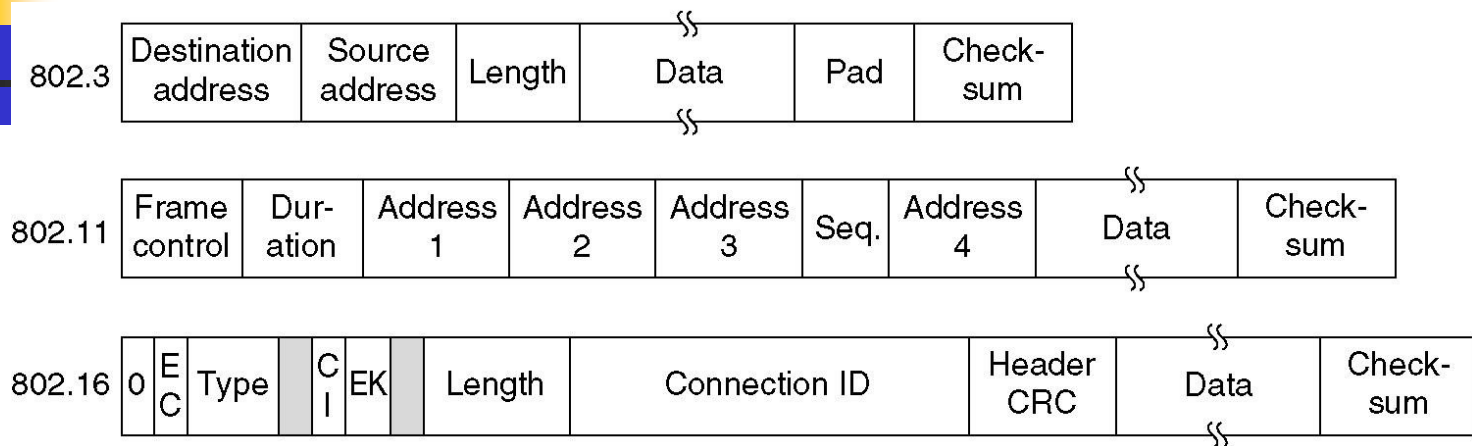
- 网桥是用来连接局域网的互连设备，工作在数据链路层。
- 转发局域网之间的数据帧，必要时进行帧格式转换
- 能隔离以太网中的碰撞域，但不能隔离广播域
- 丢弃出错帧



网桥的工作原理



网桥所存在的问题



- 在连接不同类型的局域网时，网桥的转发存在以下主要问题（以802.3和802.11为例）：
 - 不同的帧格式
 - 由于帧格式的不同，需要重新组帧，某些域需要丢弃或产生（如802.11帧的持续时间），还需要重新生成检验和。
 - 不同的数据速率（802.11速率较低）
 - 由于速率不同，网桥需要用缓冲区存储来不及传输的帧。



网桥所存在的问题（续）

- 最大帧的长度
 - 由于802标准本身不提供把长帧分片的功能，因此对于太长的帧只好丢弃。
- 安全性问题
 - 802.11可以支持加密，以太网不支持
- 服务质量问题
 - 802.11采用PCF可以支持QoS，802.3不支持。



网桥的类型

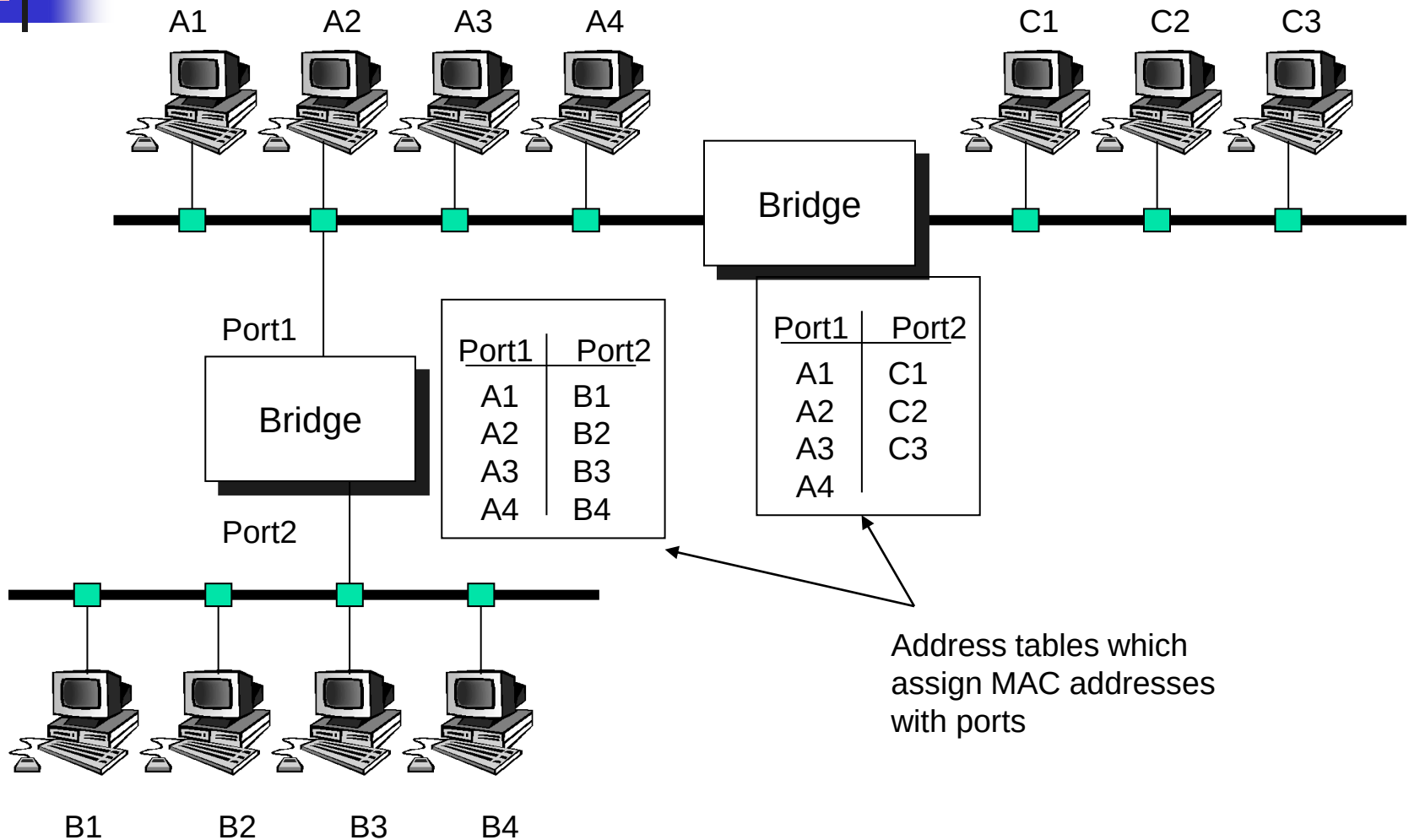
- 透明网桥（transparent bridge）
- 源路由选择网桥（source routing bridge）



透明网桥（transparent bridge）

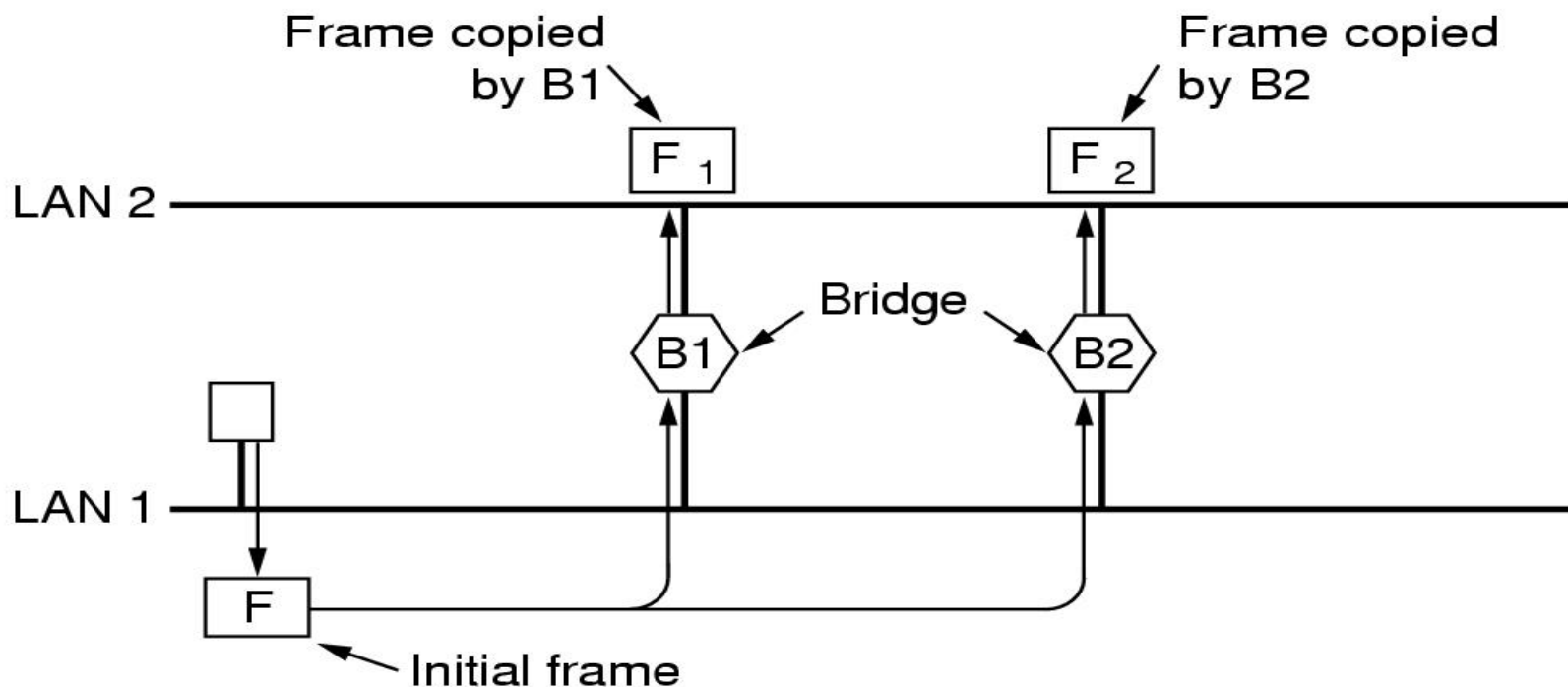
- 当网桥连接局域网的初期，网桥的地址查找表的所有表项均为空。
- 采用逆向学习法（**backward learning**）产生目的地址查找表。
- 如果数据帧的源地址和目的地址属于同一个**LAN**，则丢弃该帧。
- 当数据帧的目的地址在网桥地址表上查找不到时，采用扩散法进行转发，即向除接收端口以外的所有端口转发。
- 网桥的地址查找表定期更新。

Transparent bridges with address tables



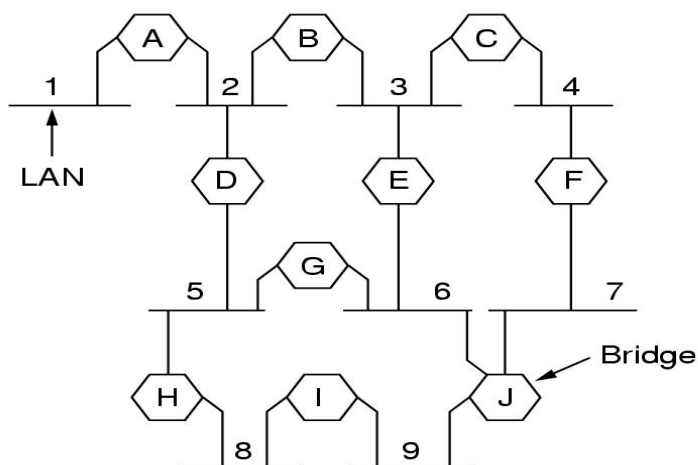
并行（parallel）透明网桥的问题

- 为了提高可靠性，可设置并行的两个或多个网桥，但会在拓扑结构上产生回路，导致阻塞。
- 解决的办法是采用生成树算法。

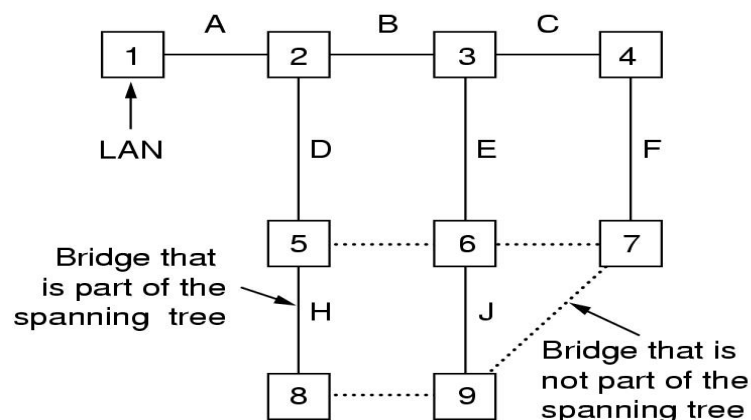


生成树（spanning tree）网桥

- 通过网桥之间的相互通信，去掉一些会造成回路的冗余链路，构造一棵可覆盖每个LAN的生成树去取代实际的拓扑结构。确保任意两个LAN之间只有唯一的一条路径。
- 首先每个网桥广播其序列号（唯一），选取序列号最小的网桥作为生成树的根。然后再按根到每个网桥的最短路径来构造生成树。此算法一直工作，自动地检查拓扑结构的变化并更新生成树。



(a)



(b)



Chapter 5 LAN & MAC Sub layer

- 5.1 传统LAN的基本概念
- 5.2 多路访问协议
- 5.3 以太网的MAC协议
- 5.4 无线LAN
- 5.5 网桥
- 5.6 交换式局域网

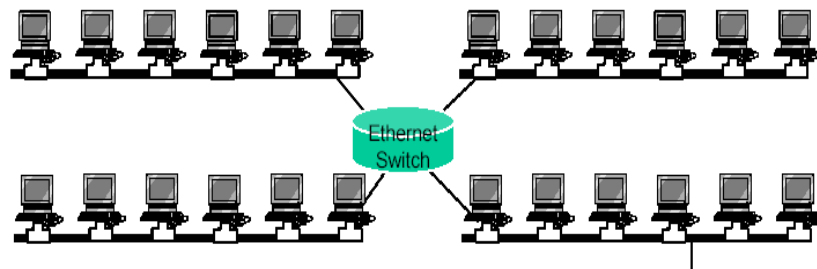
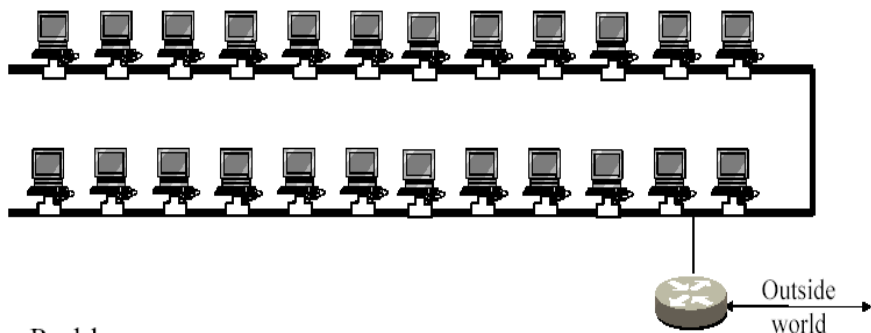
5.6 交换式局域网

■ 共享媒体的局域网问题

- 碰撞降低了信道的传输效率
- 媒体共享其实就是带宽共享，每个主机的带宽不能保证

■ 交换式局域网

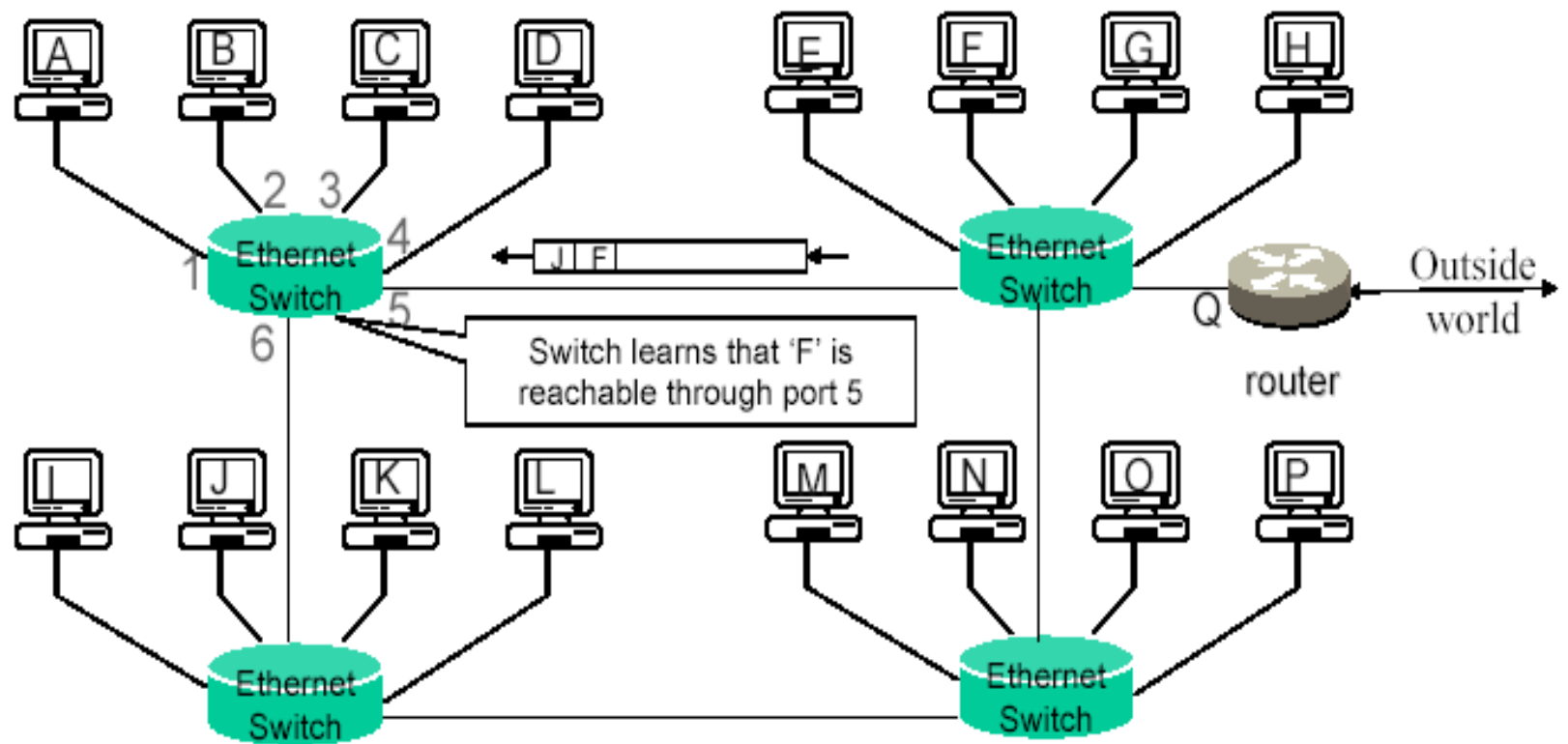
- 采用星型拓扑结构，用交换机连接主机
- 交换机工作在数据链路层，能隔离碰撞域

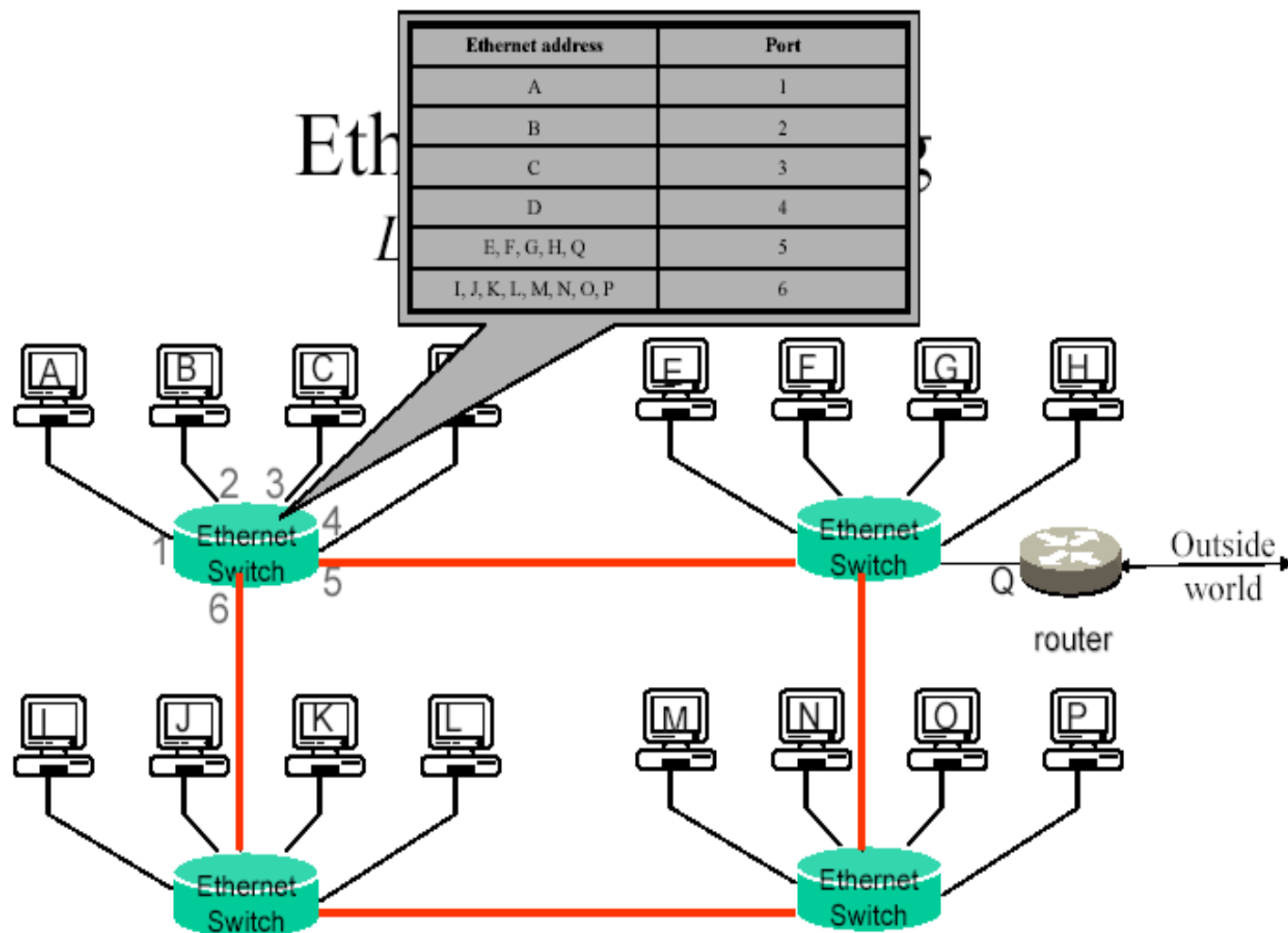
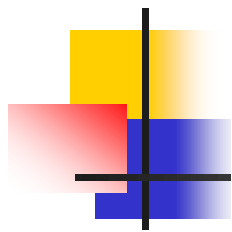




以太网的交换技术

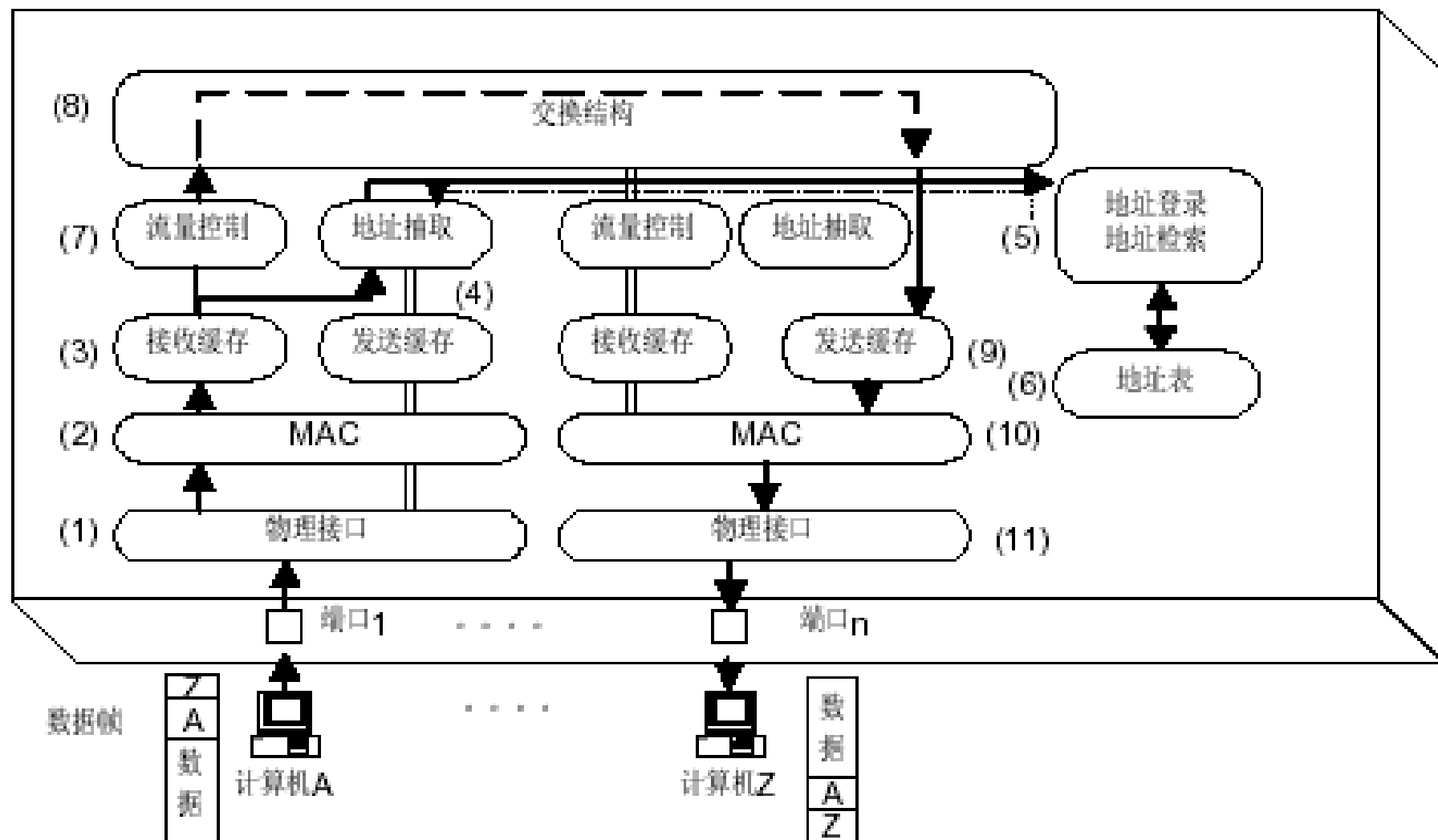
- 交换机（**switch**）源自于多端口网桥（**bridge**），采用存储-转发方式在各端口之间进行数据帧的交换
- 交换机检测每个到达数据的帧头
- 根据数据帧的目的地址查找输出端口
- 如果地址查找表中没有该表项，交换机就向所有端口（除接收端口）转发
- 地址查找表是通过帧的源地址与到达端口的对应关系建立的



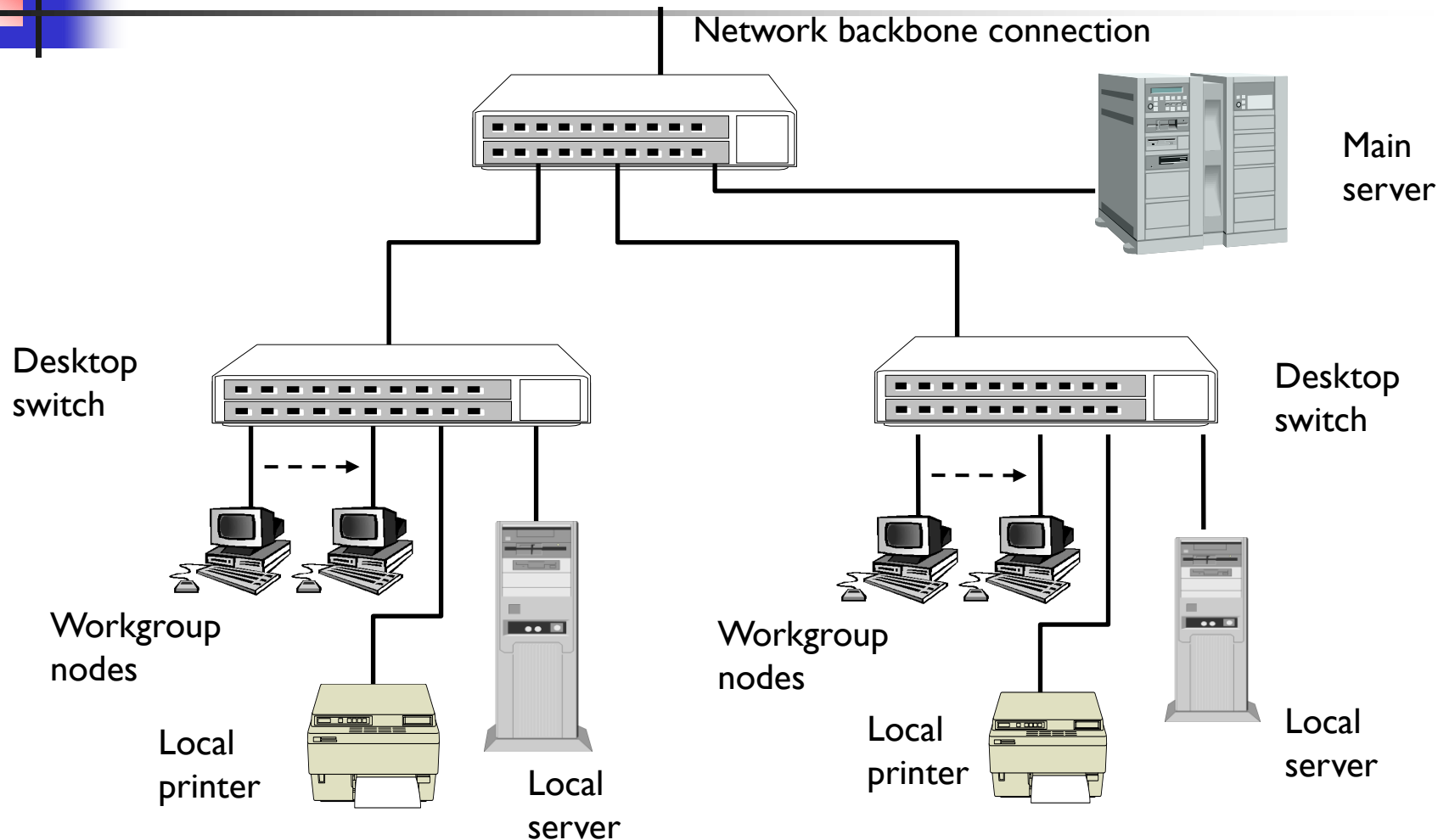


Q: How do we prevent loops?

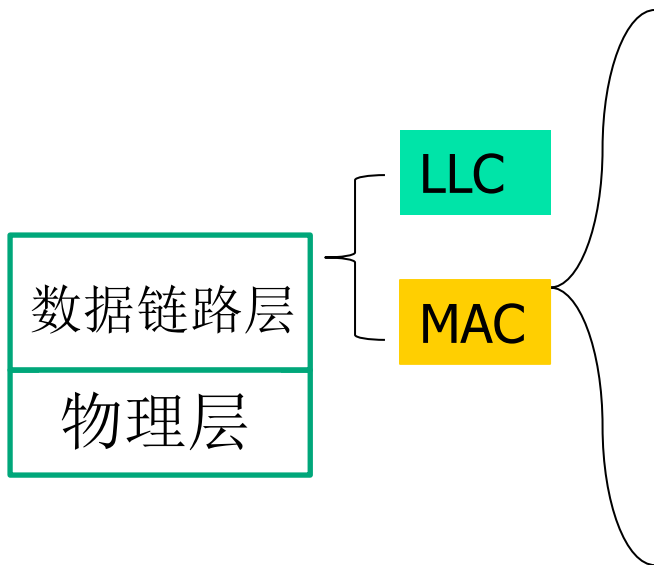
交换机的工作原理



交换机的连接方式



小结



- 传统局域网的基本概念
 - 传输媒体，拓扑结构
- 媒体访问控制方法
 - 动态信道分配：
CSMA/CD, CSMA/CA
- 以太网/802.3
- WiFi/802.11
- 局域网互联设备-网桥/交换机



小结

- 本章介绍的是数据链路层的两种典型的局域网——以太网和**WiFi**。掌握局域网的基本概念，重点掌握以太网、**WiFi**的基本协议，熟悉网桥、交换机的工作原理。
- 习题：
 - 第五版，3，10, 12，17，18，27



Chapter 6 Network layer

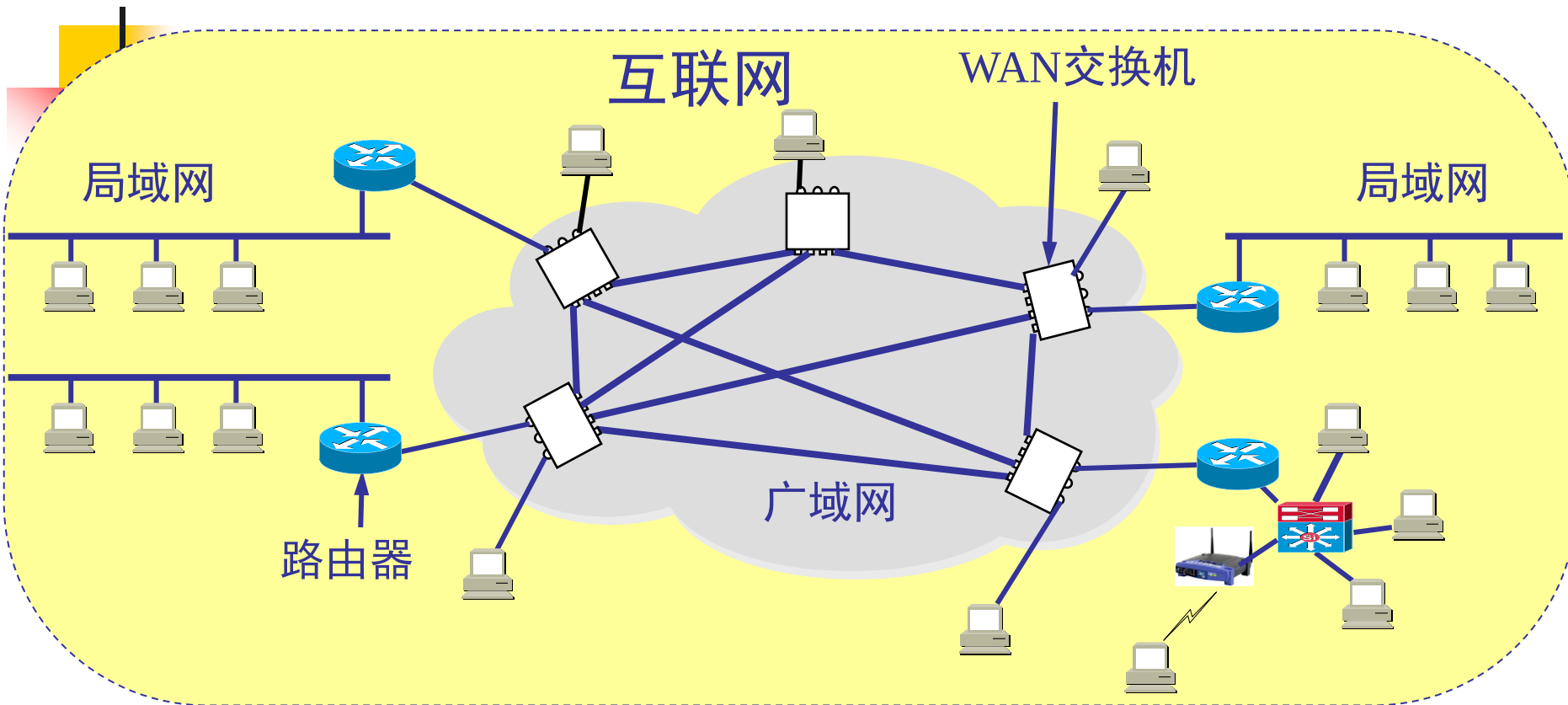
- 6.1网络层向传输层提供的服务
- 6.2虚电路与数据报
- 6.3路由算法
- 6.4拥塞控制
- 6.5网络互连



Chapter 6 Network layer

- 6.1网络层向传输层提供的服务
- 6.2虚电路与数据报
- 6.3路由算法
- 6.4拥塞控制
- 6.5网络互连

6.1网络层向传输层提供的服务



网络层的必要性：

1. 相距较远的局域网之间的互联互通需要有结构化的统一的地址编排，而MAC地址是硬件地址，与位置无关，无法用来路由。
2. 二层交换机不隔离广播域，网络规模较大时易形成广播风暴。



6.1网络层向传输层提供的服务

- 网络层设计目标：
 - 服务应与通信子网的技术无关
 - 对于传输层而言，通信子网的数量、类型和拓扑结构是隐蔽的
 - 网络地址应该采用统一的编号模式
- 争论：
 - 网络层提供面向连接的服务还是非连接的服务



连接与无连接

- 无连接的服务
 - **Internet**社区：“网络简单、终端智能”，高效和可扩展性是主要考虑因素
 - 数据报网络：**IP**
- 面向连接的服务
 - 电话公司：“网络智能，终端简单”，管理和服务质量是主要考虑因素
 - 虚电路网络：**ATM** (Asynchronous Transfer Mode)

存储
转发



Chapter 6 Network layer

- 6.1网络层向传输层提供的服务
- 6.2虚电路与数据报
- 6.3路由算法
- 6.4拥塞控制
- 6.5网络互连



6.2虚电路与数据报

- OSI的网络层提供两种服务
 - 面向连接——虚电路 (**virtual circuit**):
 - 首先要发出连接请求, 与目的端建立连接
 - 数据通信
 - 拆除连接
 - 非连接——数据报(**datagram**)
 - 每个分组头都必须包含目的地址
 - 每个分组在途径节点上被单独处理
 - 同一数据流的分组可以走不同的路径

虚电路网络

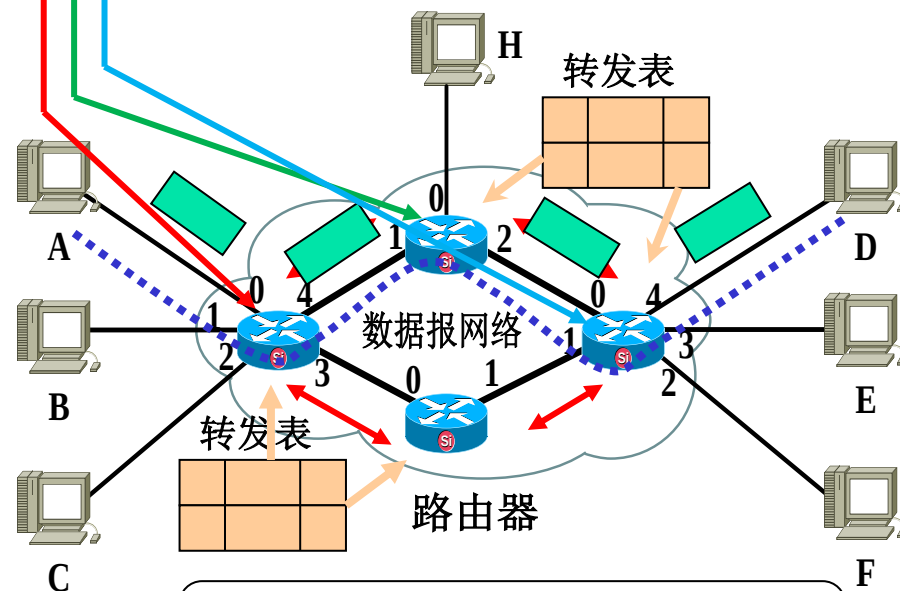
- 分组发送前建立虚连接或者虚电路(**VC: Virtual Circuit**)，即在源和目的主机之间的每个交换机上建立“连接状态”
- 分组只需携带链路范围有效的**VCI (VC Identifier)**，在交换机上标识自己所属的**VC**
- 服务质量保证：在建立连接阶段为每条虚电路分配足够的资源，以保证带宽、延时等需求

VC表记录：

由VC到达路由器的分组的输入端口
分组输入时的VCI

从VC离开路由器的分组的输出端口
用于输出分组的一个可能不同的VCI

输入端口	输入VCI	输出端口	输出VCI
0	3	4	5
输入端口	输入VCI	输出端口	输出VCI
1	5	2	7
输入端口	输入VCI	输出端口	输出VCI
0	7	4	4



路由器之间运行路由协议或者静态配置转发（路由）表

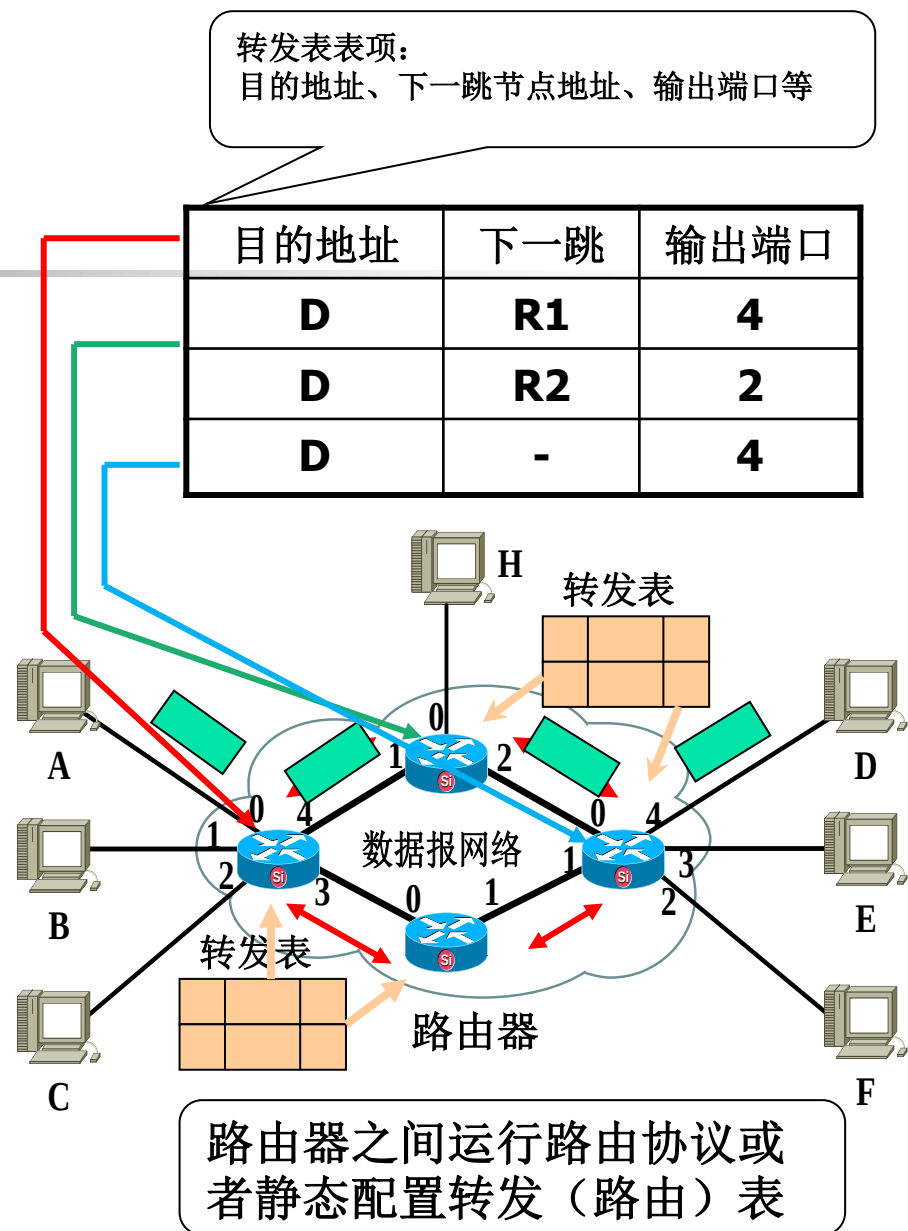


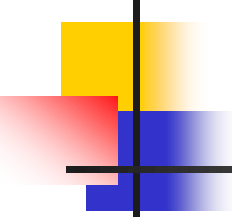
虚电路的特点

- 一条物理链路可以对应多条逻辑信道
- 一条虚电路由各物理链路上的逻辑信道级联而成，占用了节点上的一条逻辑信道实际上就是占用了该节点上缓存器内的一个存储空间
- 分组靠逻辑信道号（LCN）选择路由，因LCN只有局部意义，所以减少了分组头标的开销和处理的复杂度
- 能有效的防止拥塞

数据报网络

- 来自传输层的数据经过网络层处理后直接发送
- 每个分组中携带完整的目的地址，路由器根据该地址查找转发表来实现分组转发





数据报的特点

- 每个分组的寻路是独立的，可以合理利用网络资源
- 如果途中一个节点或一条链路发生故障，能给分组重选路由
- 分组头需要包含地址字段，也会增加开销（overhead）
- 各分组途经的路径可能不同，因此有可能出现先发后到现象
- 分组必须有生存时间限制，当生存期满时，分组则被抛弃，免得在网络内死转



虚电路与数据报的比较

问题	数据报	虚电路
连接建立	不需要	需要
路由与转发	每个分组都包含源/目的地址，独立路由，路由器根据分组的目的地地址进行路由转发	建立连接时需要路由，一旦建立逻辑通道，后续分组都根据分组头的逻辑通道号转发
路由器状态	不保留分组路由过的状态信息	路由器需要保留每个连接的状态信息，以便后续分组的转发
分组路径与到达顺序	同一流的分组可能沿不同的路径到达目的端，有可能乱序	同一流的分组沿建立好的虚电路到达按序目的端
路由器故障影响	通过相邻节点的路由表更新，能迂回故障节点	故障路由器上的所有连接都将中断
服务质量	较难提供	易于在建立连接时在路径上的各个节点预留资源
拥塞控制	较难提供	在建立连接时，可根据网络的资源选择路径，避免网络拥塞

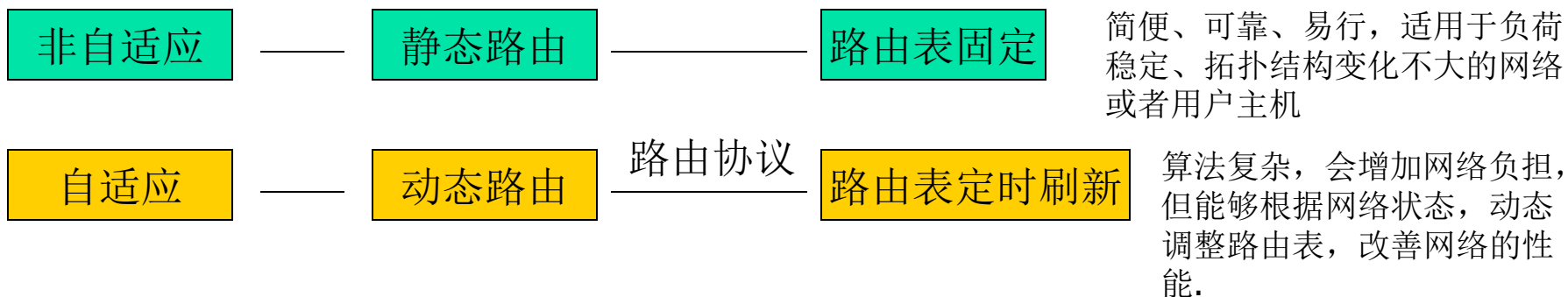


Chapter 6 Network layer

- 6.1网络层向传输层提供的服务
- 6.2虚电路与数据报
- 6.3路由算法
- 6.4拥塞控制
- 6.5网络互连

6.3 路由算法(routing algorithm)

- 网络层的主要功能是根据分组目的地址选择路径，对数据报，每个分组都要在途径的节点上被单独寻路；而虚电路，则在建立连接时要进行寻路。
- 路由算法有两类：非自适应和自适应



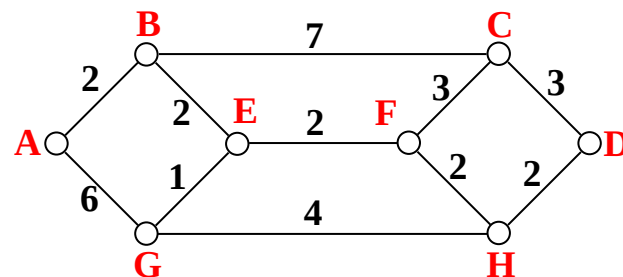


自适应路由算法

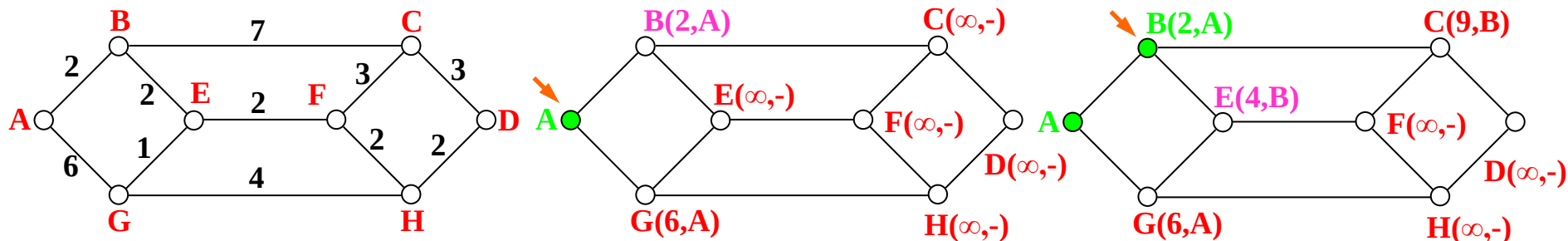
- 链路状态算法
- 距离矢量算法

6.3.1 Dijkstra最短通路搜索算法

- 最短通路算法的基本准则：
 - 在全双工链路连接的网络上，每条链路的每个方向上都有一个与之相关的权值。两个节点之间一条路由的代价是它所经过的链路权值之和，所以，这两个节点间的最佳路由为其所有可能路径中具有最小代价的那条路径。
- Routing metrics 度量(weight, cost)
 - Number of hops
 - Delay
 - Bandwidth
 - Load



利用Dijkstra算法求A到D的最短通路



变量:

$c(i,j)$: 从节点i到j链路代价(初始状态下非相邻节点之间的链路代价为无限大)

$D(v)$: 从源节点到节点V的当前路径代价(节点的代价)

$p(v)$: 从源到节点V的路径前序节点

节点标记: 每一个节点使用 $(V(D(v), p(v)))$ 如: B(2,A)标记

2类节点:

临时节点(红色): 还没有找到从源节点到此节点的最优路径的节点

永久节点(绿色): 已经找到了从源节点到此节点的最优路径的节点

步骤:

初始化

除了源节点外, 所有节点都为临时节点

节点代价除了与源节点代价相邻的节点外, 都为 ∞

从所有临时节点中找到一个节点代价最小的临时节点(紫色), 将之变成永久节点(当前节点)W

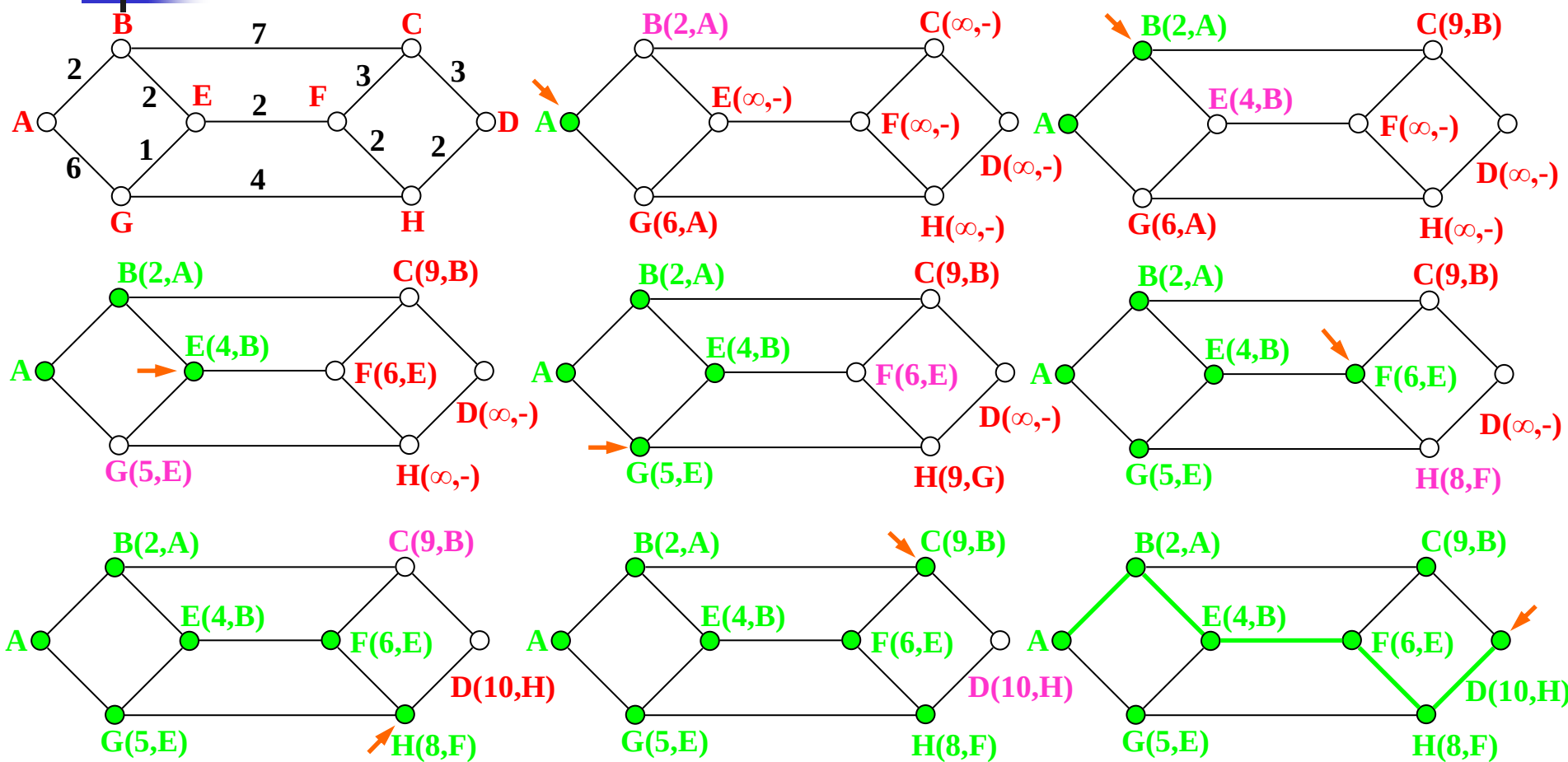
对此节点的所有在临时节点集合中的邻节点(V)

如 $D(v) > D(w) + c(w,v)$, 则重新标注此点, $V(D(w) + C(w,V), W)$

否则, 不重新标注

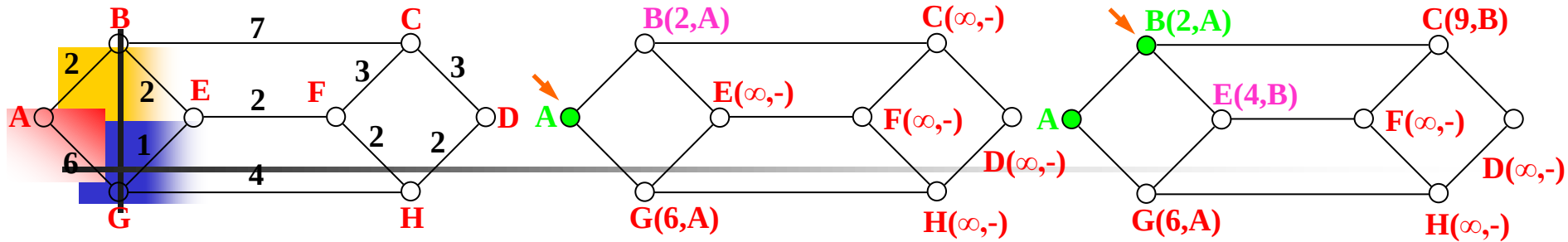
开始一个新的循环

利用Dijkstra算法求A到D的最短通路



最短通路为：A-B-E-F-H-D，权值为10

利用Dijkstra算法求A到D的最短通路

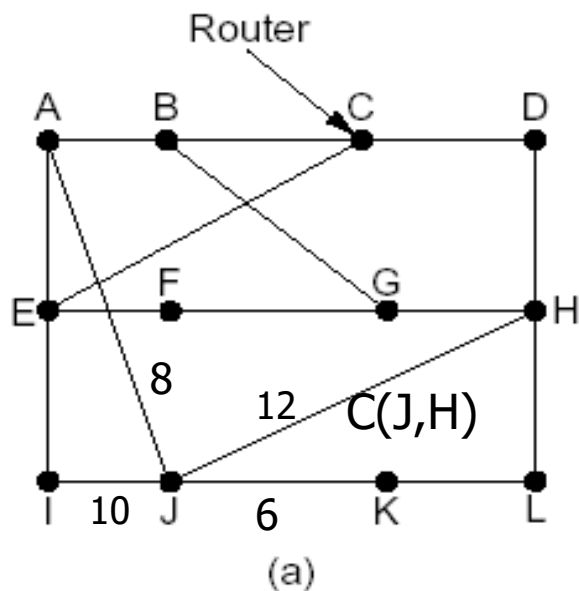


步骤	D(B),p(B)	D(G),p(G)	D(E),p(E)	D(C),p(C)	D(F),p(F)	D(H),p(H)	D(D),p(D)
1	2,A	6,A	∞	∞	∞	∞	∞
2	2,A		4,B	9,B			
3		5,E	4,B		6,E		
4		5,E				8,F	
5				9,B	6,E	8,F	
6						8,F	10,H
7				9,B			10,H
8							10,H

最短通路为：A-B-E-F-H-D，权值为10

6.3.2 距离矢量（Distance Vector）路由算法

- **距离矢量路由**选择（distance vector routing, DVR）算法是现代计算机网络两个最常使用的动态路由选择算法之一。
- Internet的一种内部网关协议**RIP**（Route Information Protocol）使用了距离矢量路由选择算法；Cisco开发的另一种改进的协议，叫作**IGRP**（Interior Gateway Routing Protocol）也使用距离矢量路由算法。
- 每个路由节点**定期地**向它的**相邻路由节点交换路由表**，这里的路由表所包含的内容有：
 - 每条路径的**目的地址——矢量**
 - 本节点到该目的地址的**代价——距离**
- 每个节点根据收到的相邻节点的路由信息**更新自己的路由表**



$d_A(y), d_I(y), d_H(y), d_K(y)$ New estimated delay from J

To	A	I	H	K		Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K

JA delay is 8 JI delay is 10 JH delay is 12 JK delay is 6

Vectors received from J's four neighbors

New routing table for J

Bellman-Ford算法

节点x到y的最低费用

$$d_x(y) = \min_v \{C(x,v) + d_v(y)\}$$

例题中, $x=J, v=A, I, H, K$

距离矢量路由算法举例

存在问题

- 寻路环 (routing loops)
- 慢收敛 (slow convergence)
- 无穷计算 (count to infinity):
 - 它对好消息的反应迅速，但对坏消息却反应迟钝。

A	B	C	D	E	
●	●	●	●	●	
	∞	∞	∞	∞	Initially
	1	∞	∞	∞	After 1 exchange
	1	2	∞	∞	After 2 exchanges
	1	2	3	∞	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
●	●	●	●	●	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges

$\vdots$
 ∞ ∞ ∞ ∞

(b)



解决方案——水平分割算法

- **水平分割**（split horizon）算法是解决无穷计算问题的一种方法。
- 其工作过程与距离矢量法一样。
- 不同之处在于：如果节点**C**通过节点**B**向**A**发送分组，则节点**B**不会再试图通过**C**向**A**发送分组。换句话说，防止路由器向邻居返回一个从该邻居获得的最佳路径。

6.3.3 链路状态（ Link State ） 路由算法

- 链路状态路由算法于1979年出现在ARPAnet上，作为一种用来取代DVR的动态路由选择算法，得到了广泛的应用。
- 算法：
 - 主动测试邻接节点的状态
 - 定期地将相邻节点的状态信息传送给所有节点
 - 每个节点都有完整的网络拓扑信息，然后计算到每个节点的最佳路径
- 该方法也叫最短路径优先（shortest path first），简称SPF算法



Distance Vector vs. Link State

■ DV

- 节点向相邻节点告诉它所知道的**所有节点的路由信息**
- 节点根据相邻节点的路由信息更新自己的路由表
- 分布式计算
- 可扩展性差

■ LS

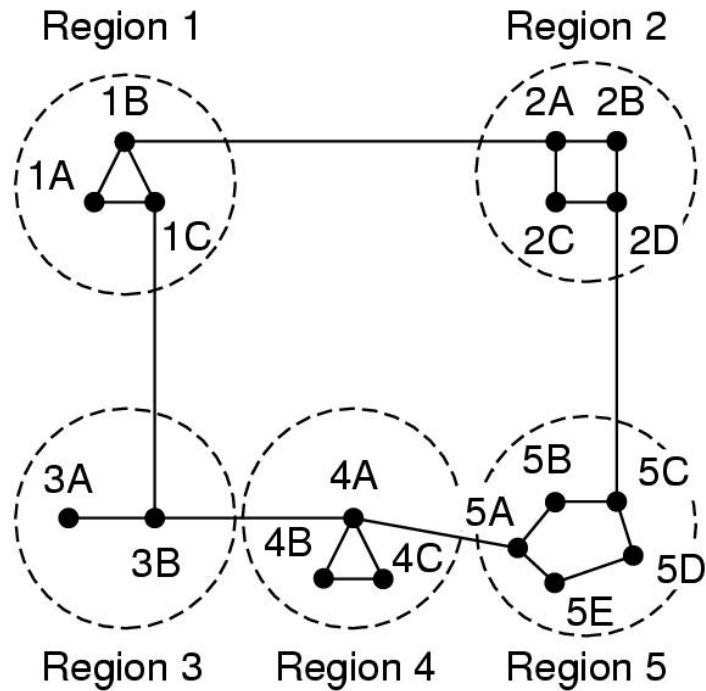
- 节点向所有节点告诉其**相邻节点**的状态信息
- 每个节点都有一个全局的拓扑结构
- 根据此拓扑结构计算路由表
- 可扩展性好，可靠



层次路由（hierarchical routing）

- 路由表项大小与网络规模成正比，当网络规模变大时，路由表也变大，会消耗更多的内存、带宽、路由表查找时间。
- 采用层次化路由解决以上问题。
 - 将网络分区域，区域内的路由器只负责本区域内的分组转发，目的地址不在本区域内的分组都发给指定的区域路由器（边界路由器）去处理。
 - 路由信息的交换只在域内进行，路由器内部需存储的路由表项大大减少。
 - 对于大型网络，可以分成多层次结构，每一层次的边界路由器相互之间交流路由信息。
- 缺点是选择的路由可能不是最佳的。

层次路由示例



(a)

Full table for 1A

Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

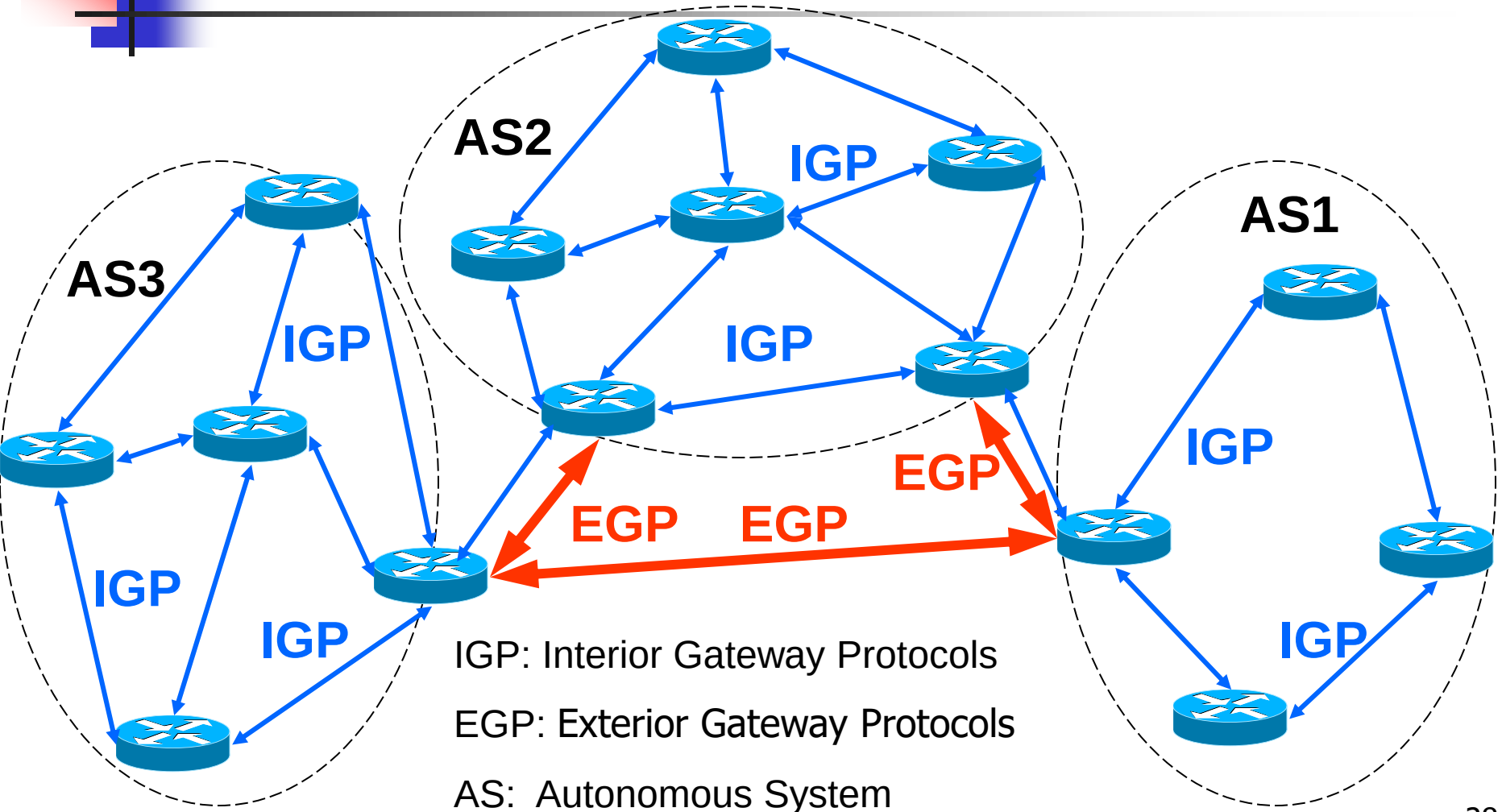
(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

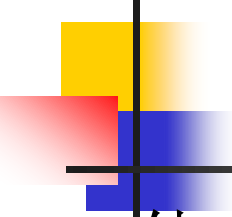
IGP vs. EGP





Chapter 6 Network layer

- 6.1网络层向传输层提供的服务
- 6.2虚电路与数据报
- 6.3路由算法
- 6.4拥塞控制
- 6.5网络互连



6.4 拥塞控制

■ 什么是网络的拥塞

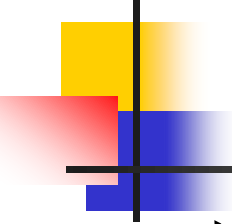
- 当大量分组进入通信子网，超出了网络的处理能力时，就会引起网络**局部或整体性能下降**，这种现象称为**拥塞**。
- 路由器的**队列溢出**，**分组丢失**

■ 拥塞会导致什么后果

- 拥塞使许多分组重传
- 导致更多的业务量，直至崩溃

■ 拥塞的原因

- 路由器的处理速度、存储空间、带宽不匹配
- 网络负载的不平衡



拥塞控制与流量控制

■ 拥塞控制

- 网络负载的不均衡，例如，某个路由器的多个输入端口向同一个输出端口传输分组
- 是**全局**问题，涉及的节点包括主机、路由器
- 拥塞发生在网络内，但对于无连接的网络很难在网络层单独控制拥塞，因此网络层和传输层共同承担拥塞控制。

■ 流量控制

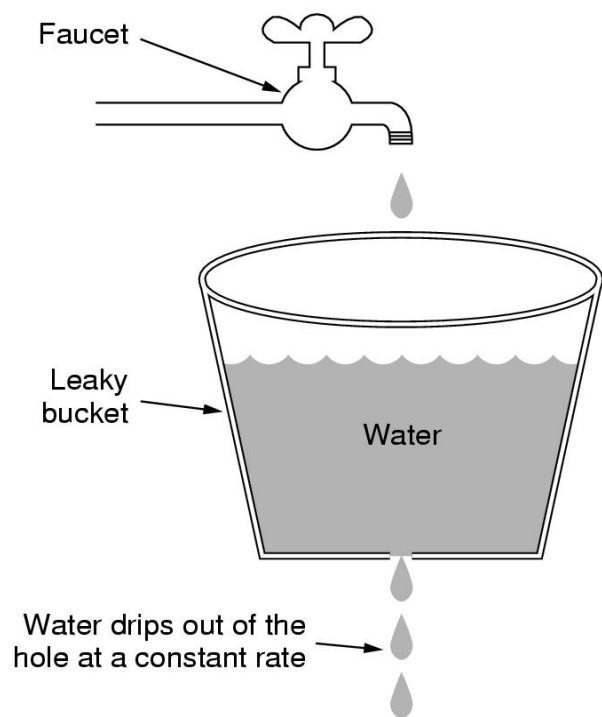
- 接收端或所在网络的接收速度小于发送端的发送速度
 - **只涉及收发两端**，是**局部**问题
- 由于解决方法相似，两者经常混淆



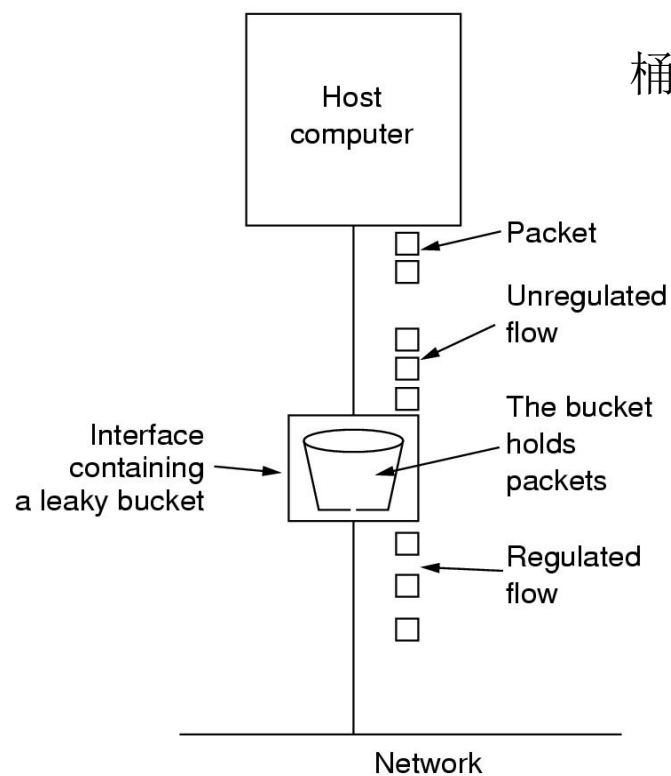
6.4.1 拥塞控制的一般原理

- 从控制理论观点出发，可分为两类：
 - 开环控制(Open Loop)，基于良好的设计
 - 业务量整形——漏桶算法、令牌桶法
 - 闭环控制(Close Loop)，基于反馈概念
 - 监测系统何时、何处发生拥塞
 - 将拥塞信息传到能控制它的地方
 - 调整系统操作
 - 指示网络拥塞的参数
 - 分组丢失率
 - 平均队列长度
 - 分组重传率
 - 平均分组延时

漏桶算法



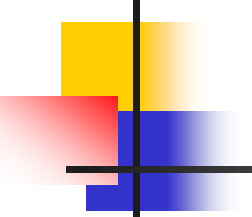
(a)



(b)

桶的容量为 B

以恒定的
速率 R 向
网络发送



举例

- 计算机以25MB/s（200Mbps）速率产生数据，向网络发送1MB的数据（即以25MB/s速率发送了40ms）。而网络的最佳传输速率不超过2MB/s，为了降低传送速率，令漏桶的 $R=2\text{MB/s}$ ，因此1MB的数据将传输500ms。

令牌桶算法

令牌桶：Token Bucket

- 桶中保存的是令牌，每隔 T 秒产生一个
- 只有当桶中有令牌时才能传输数据
- 允许突发流量

假设：

S : 突发长度 (秒)

C : 令牌桶容量 (字节)

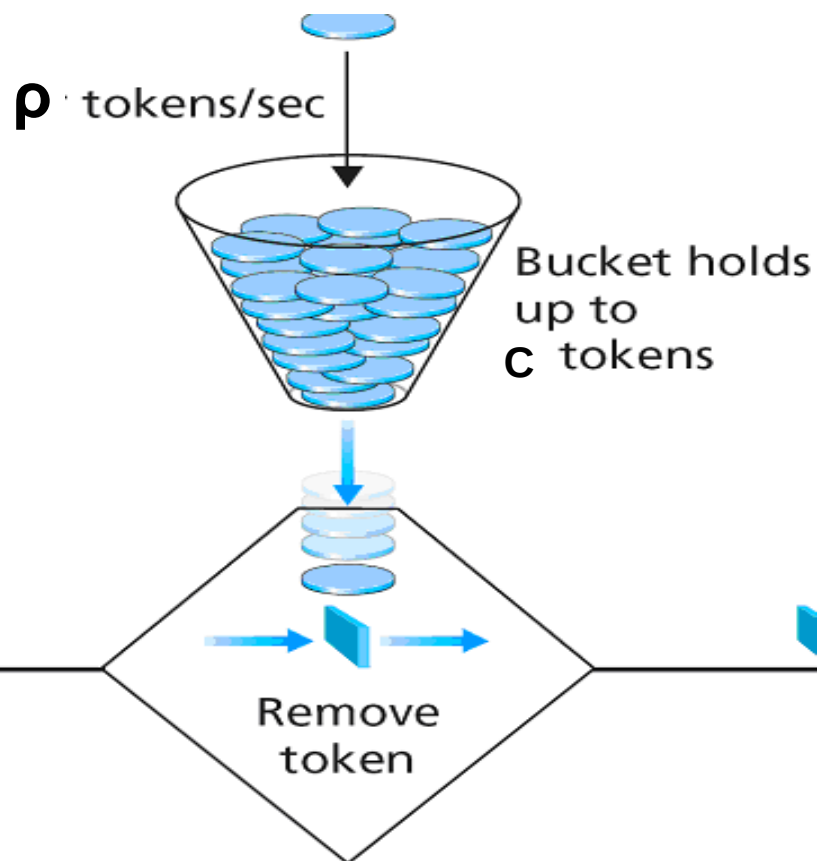
M : 最大输出速率 (字节/秒)

ρ : 令牌产生速率 (字节/秒)

$$C + \rho S = MS \quad S = C / (M - \rho)$$

Arriving packets

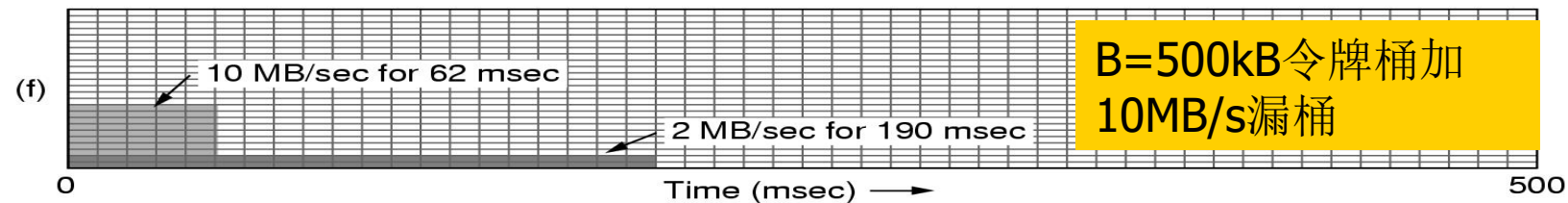
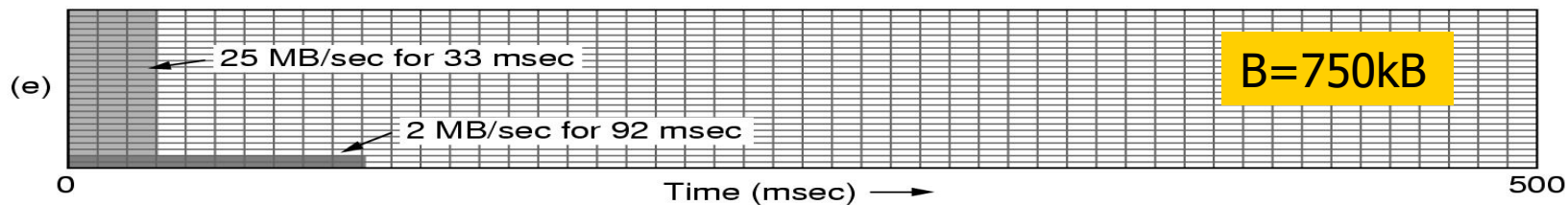
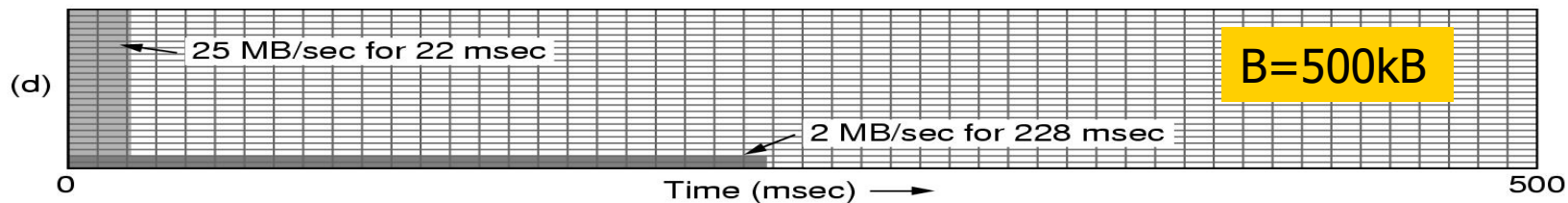
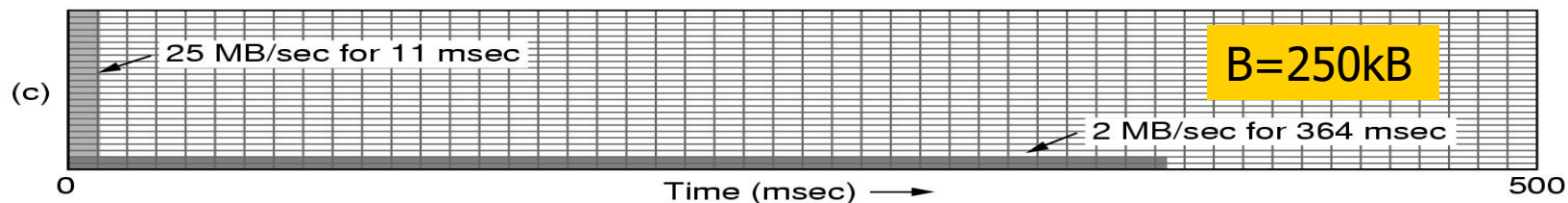
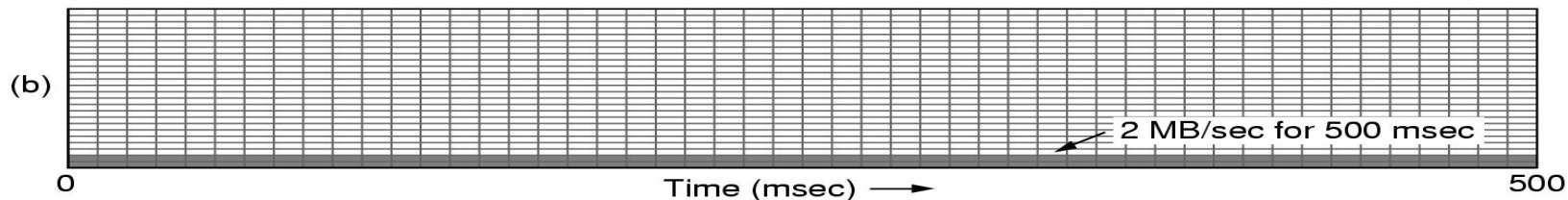
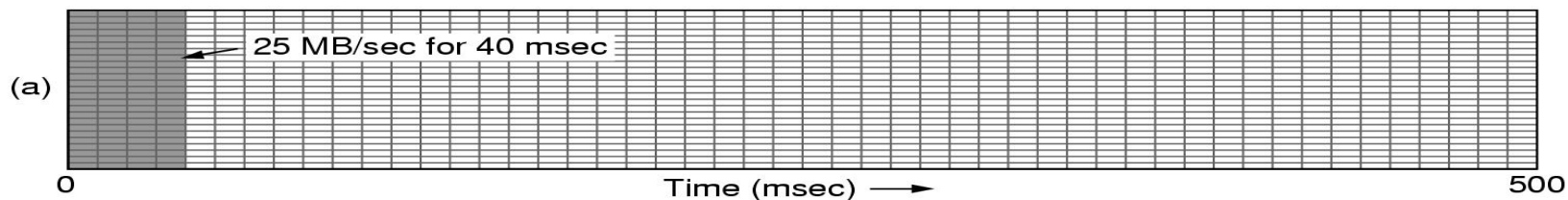
Token
wait area





举例

- 设突发长度为S秒，令牌桶容量B bytes，令牌产生速率为R bytes/s，计算机最大数据速率M bytes/s。
- 设前例中， $B=250\text{KB}$ ， $M=25\text{MB/s}$ ， $R=2\text{MB/s}$ ，发送1MB数据，问突发时间多少？以2MB/s的发送时间多少？
- $B + R S = M S$ $S = B / (M - R) = 11\text{ms}$
- $(1 - 0.011 * 25) / 2 = \text{剩余的以} 2\text{MB/s} \text{发送}$
 362.5ms





比较

- 漏桶算法的输出保持的是严格的均匀速率，不管业务流量的突发程度如何
 - 在漏桶算法中，不允许将空闲时的发送许可权保存起来以便发送大的突发数据（每个时钟嘀嗒后，漏桶的字节计数器都将被重置）
- 令牌桶算法在大量突发数据到来的时候，允许输出流适当的加快
 - 可以将发送许可权保存起来，直到到达桶的最大尺寸。这也就意味着只要突发数据不超过桶的大小，就可以一次发送出去
- 在漏桶算法中，桶中填充的是数据，所以当桶填满后将丢弃分组，而在令牌桶中，桶中填充的是令牌，所以当桶填满后将丢弃令牌，相当于是传输许可，而不是分组

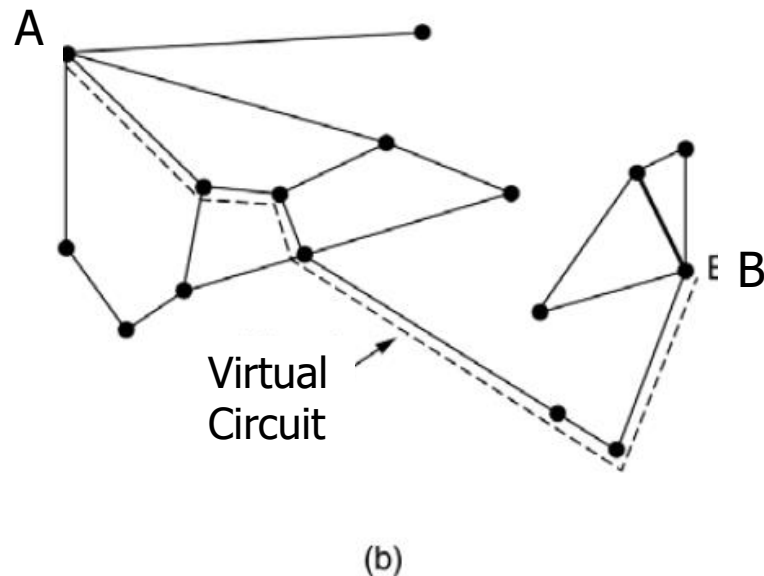
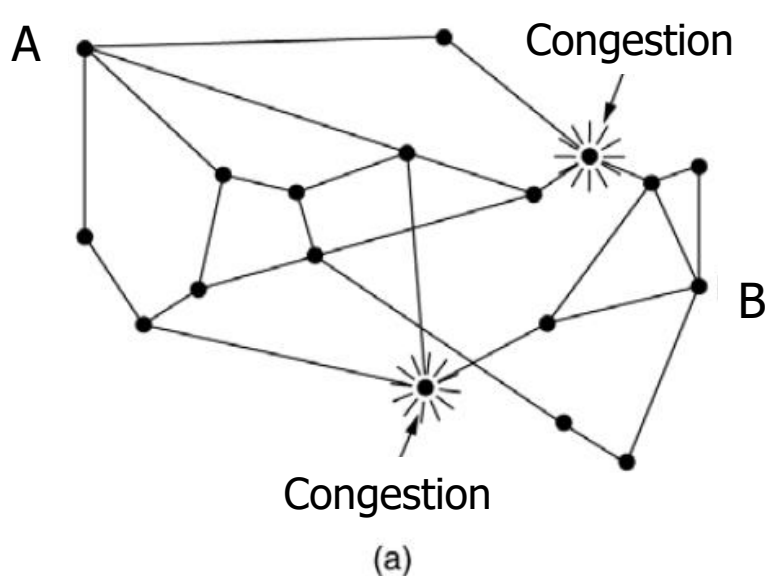


6.4.2 拥塞控制的策略

- 闭环控制(Close Loop)
 - 检测到拥塞时，就发一个分组给源端或向所有主机广播
 - 在每个分组头中保留一个位或域，当拥塞超过一定值时，路由器就在该位或域上填上拥塞信息通知网络节点
 - 由主机或路由器发送询问分组打听拥塞情况

虚电路网络中的拥塞控制

- 用户事先与网络服务提供商协商好服务等级，在虚电路建立过程中预留相应的资源
- 接纳控制 (**Admission Control**)：如果网络出现拥塞，拒接建立新的虚电路
- 虚电路建立时绕开发生拥塞的区域

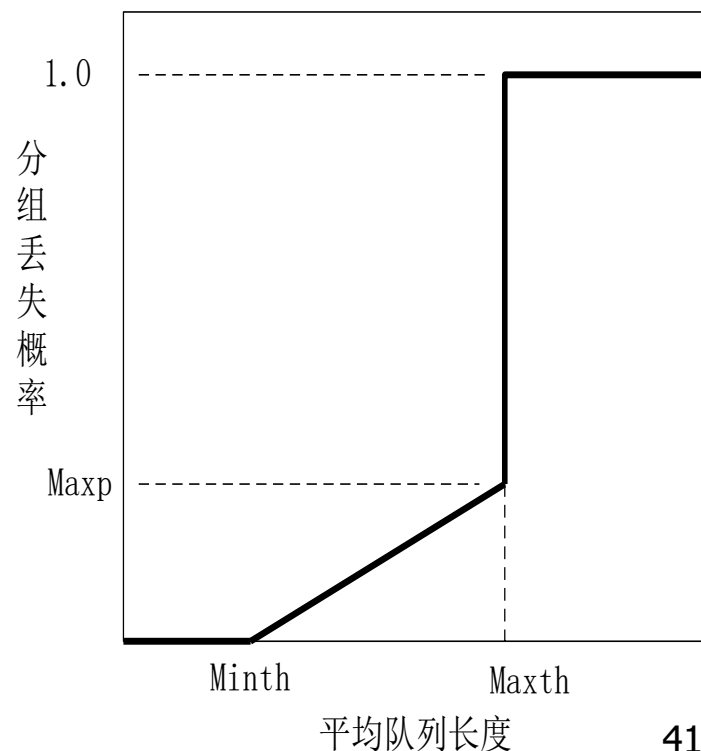


数据报网络中的拥塞控制

■ 抑制分组

- 源抑制：当路由器发生拥塞时，发送抑制分组给源主机，其中指明原分组的目的地址；源主机收到抑制分组后，减慢到指定目的地的流量
- 逐跳抑制：抑制分组将对其通过的每跳都起作用

- 采用随机早期检测（RED, Random Early Detection）的方法进行拥塞控制，该方法采用提早丢包来减轻网络负载。





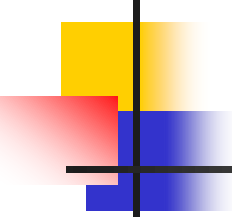
Chapter 6 Network layer

- 6.1网络层向传输层提供的服务
- 6.2虚电路与数据报
- 6.3路由算法
- 6.4拥塞控制
- 6.5网络互连



6.5 网络互连

- 为什么需要互连？
- 怎样互连？
- 有哪些互连设备？

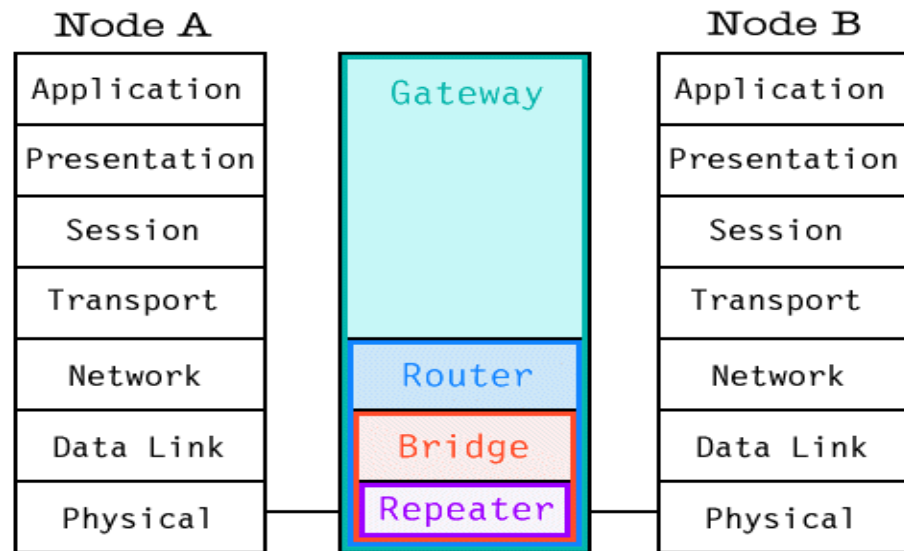


为什么需要互连？ / 怎样互连？

- 距离：由于媒体的传输距离有限（ 100m or 185m or 500m or 2.5km）
 - 中继器：延长传输距离
- 容量：同一网段的所有主机共享（10 or 100 Mbps），异种LAN的互连
 - 网桥/以太网交换机：隔离碰撞域，转发数据帧，第二层连接设备
- 路由：物理地址不具备广域可寻路特性，广播数据，异种网络的互连
 - 路由器：隔离广播域，转发分组，第三层连接设备
- 交换机（虚电路）：在面向连接的网络（X.25, ATM, 帧中继）中，级联逻辑链路

互连设备

- Repeater/Hub
- Bridge/Switch
- Router/L3 Switch
- Gateway

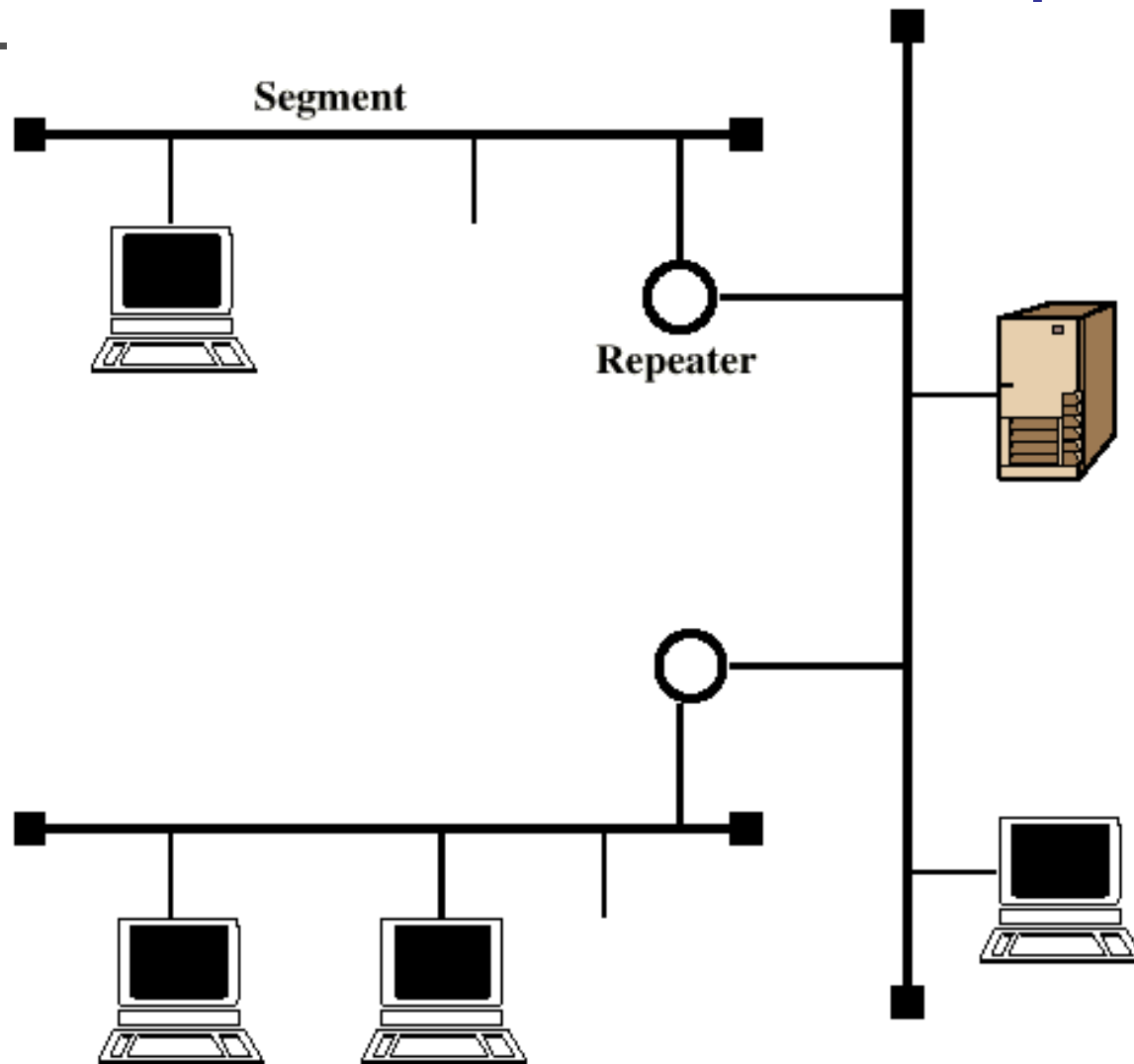




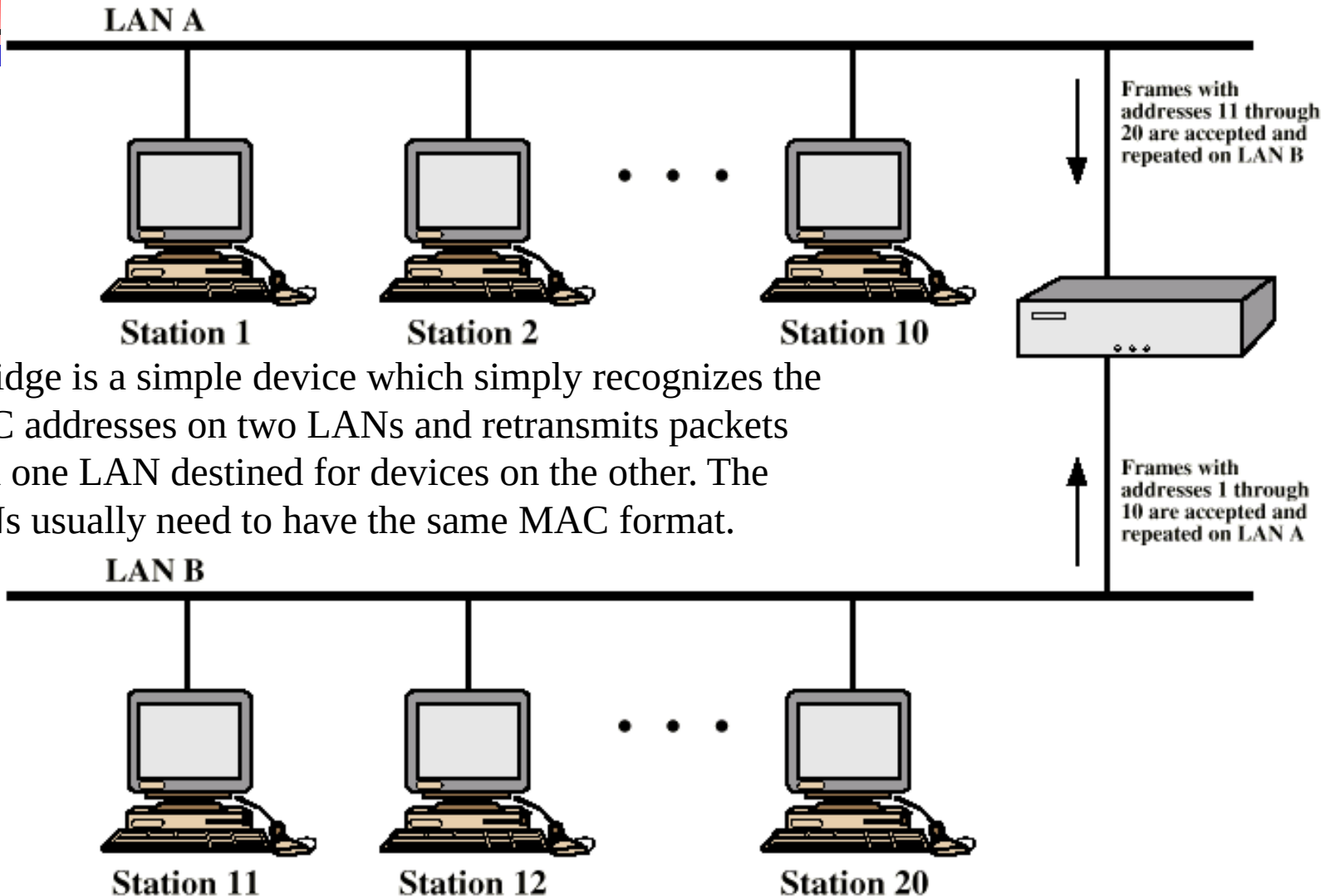
互连设备的功能

- 网络连接设备有
 - **repeater or hub**: 在物理层透明地复制比特流，以补偿信号的衰减
 - **bridge or switch**: 在不同的LAN间存储转发帧，必要时要进行链路层的协议转换
 - **router**: 在不同的网络间存储转发发组，必要时进行网络层协议转换
 - **Gateway**: 指对高层协议进行转换的协议转换器

Segmented Baseband with Repeaters

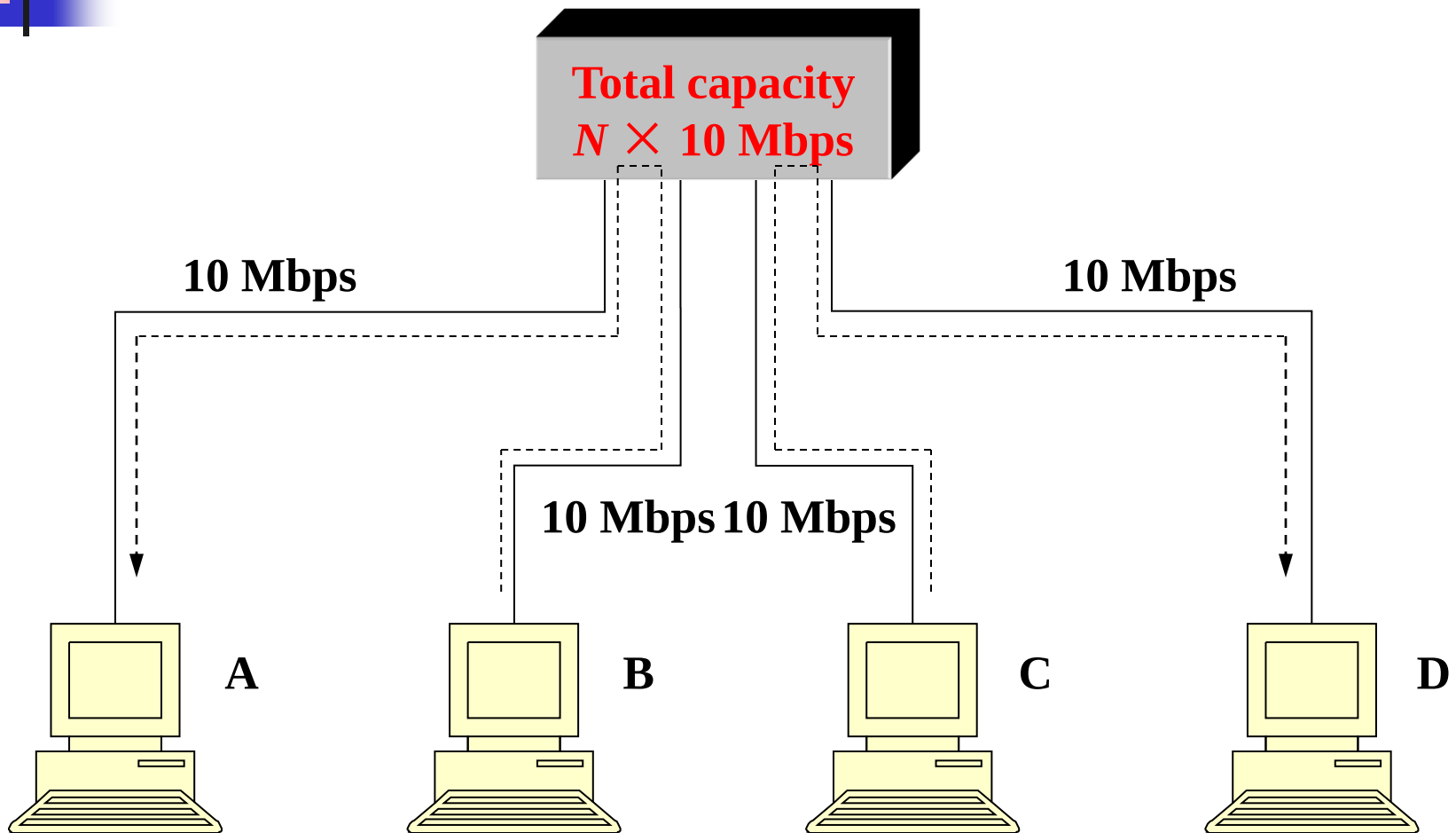


Bridge Operation



A bridge is a simple device which simply recognizes the MAC addresses on two LANs and retransmits packets from one LAN destined for devices on the other. The LANs usually need to have the same MAC format.

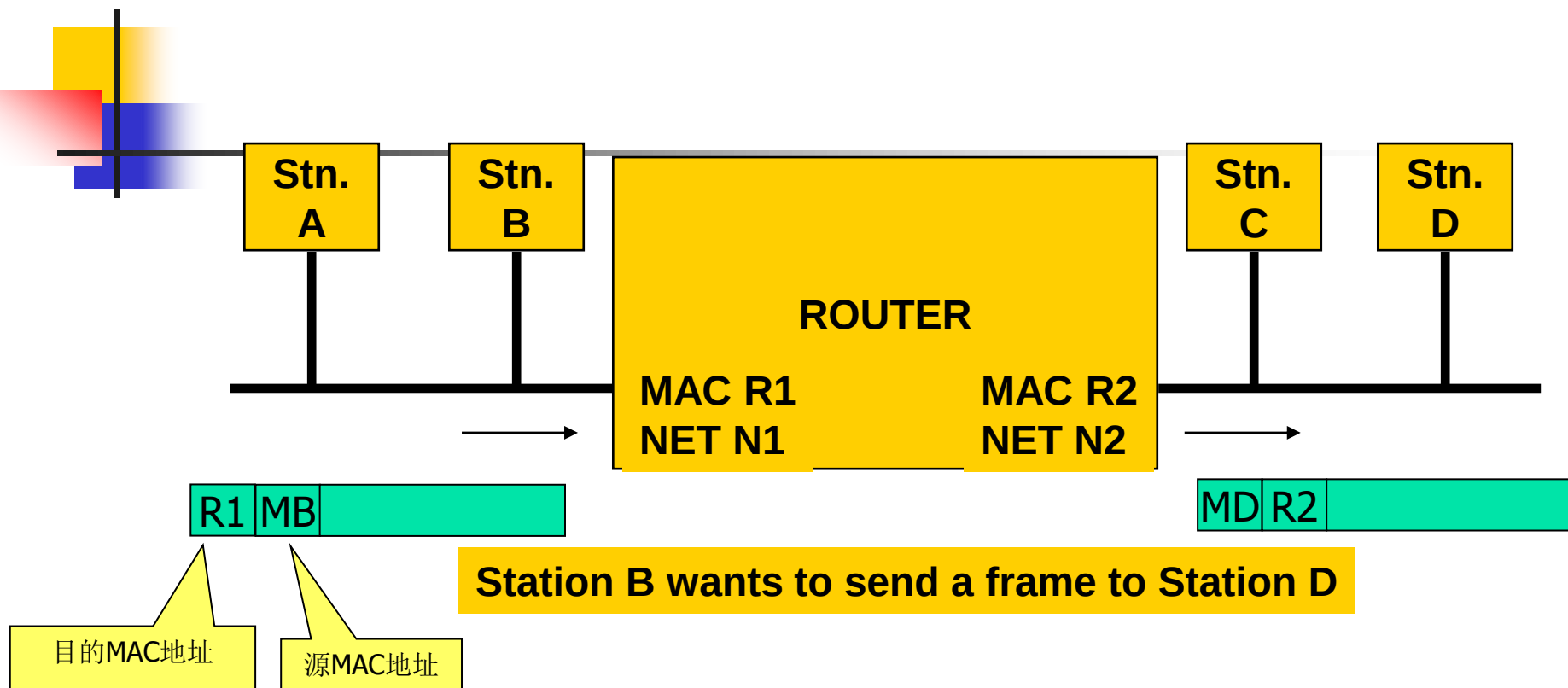
Capacity with Switched Hub





Routers

- Operate at the Network Layer
- Selectively forwards packets
- Error Frames will not be forwarded
- Collision domains are isolated
- Broadcast domains are isolated
- Protocols can be Filtered
- Has an effect on MAC Addresses but no effect on Network Addresses



- **R1/ R2** : 图中路由器的左/右接口**MAC**地址
- **MB/MD**: 站点**B/D**的**MAC**地址



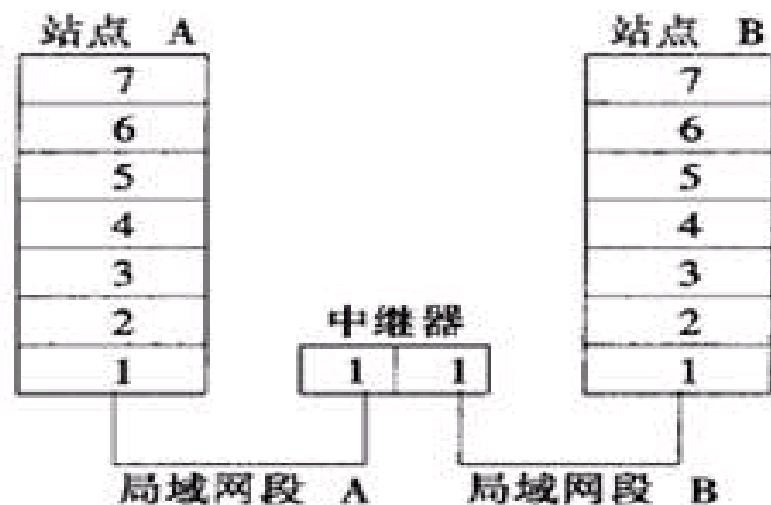
Per Packet Processing in an IP router

1. Accept packet arriving on an incoming link.
2. Lookup packet destination address in the forwarding table, to identify outgoing port(s).
3. Manipulate packet header: e.g., decrement TTL, update header checksum.
4. Send packet to the outgoing port(s).
5. Buffer packet in the queue.
6. Transmit packet onto outgoing link.

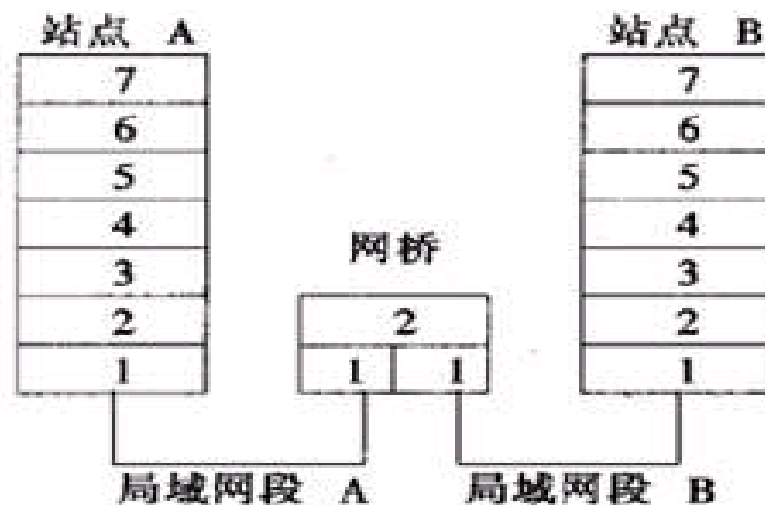


Gateway

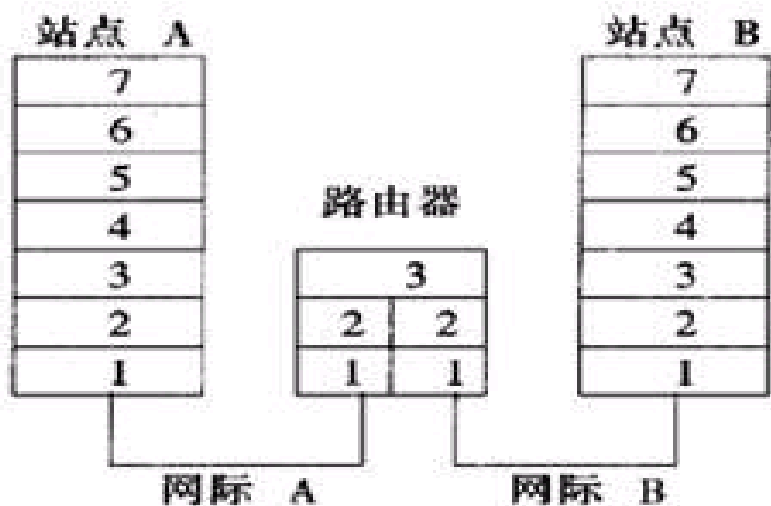
- 网关也称协议转换器，它是用于连接不同网络的高层网络互连设备
 - 应用层网关：在应用层连接两个网络，如**IP**电话网关
 - 传输层网关：在传输层连接两个网络
 - 安全网关，用于安全考虑的协议过滤，包过滤，内容过滤等，也称为防火墙，用于连接内外网络，保护内部网络的安全



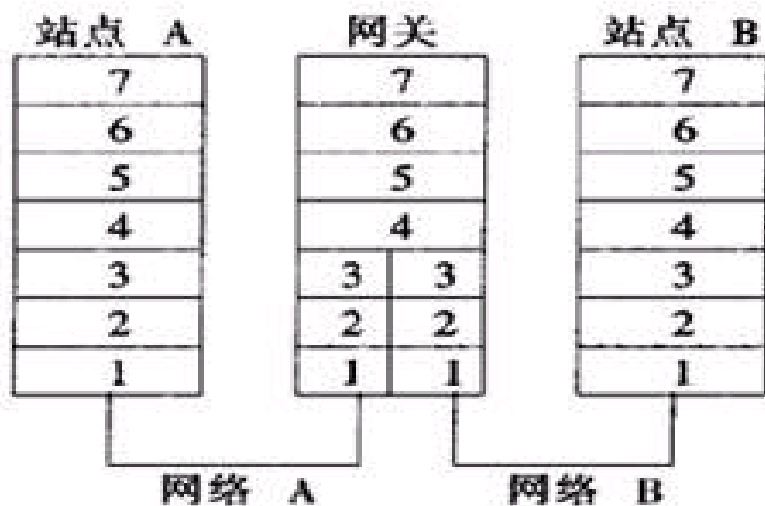
(a) 中继器



(b) 网桥



(c) 路由器



(d) 网关

小结



■ 向传输层提供的服务

- 面向连接-虚电路
- 无连接-数据报

注意两者之间的区别，同时注意虚电路与电路交换的区别

■ 路由算法

- 自适应/非自适应（动态/静态路由）
- 距离矢量与链路状态
- 层次化路由

■ 拥塞控制

- 拥塞控制与流量控制的区别
- 漏桶算法、令牌桶算法

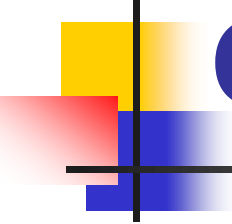
■ 网络互连

- 从物理层到应用层都有哪些经典的互连设备，它们的转发机制？
- 哪些设备能隔离碰撞域、广播域

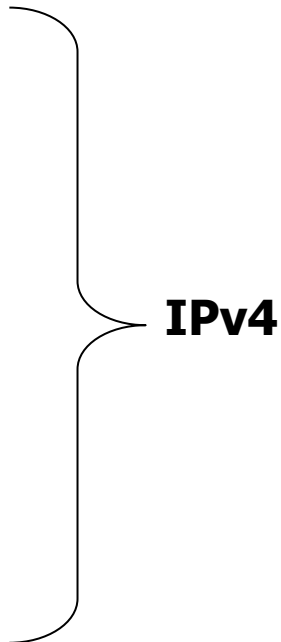


小结

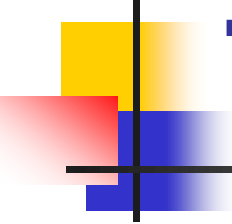
- 了解网络层的基本功能，掌握虚电路与数据报基本概念，掌握路由算法、拥塞控制。了解网络互连设备。
- 习题：
 - 第五章：2，4，6，19
 - 同样基于图5-12a，给出A节点到其他所有节点基于Dijkstra的最短路径求解



Chapter 7 Internet Protocol

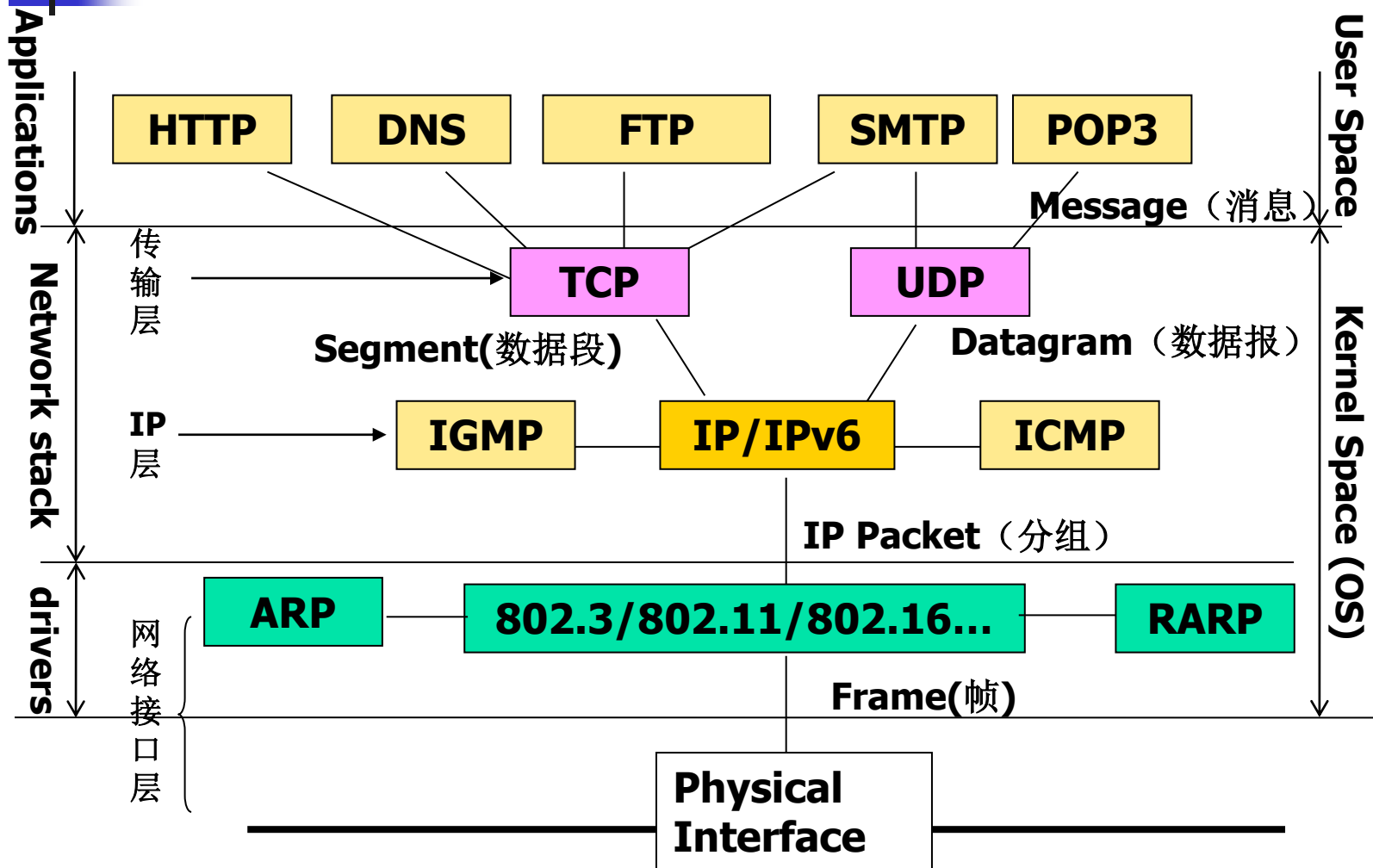
- **7.1 Internet基本协议栈**
 - **7.2 IP地址**
 - **7.3 地址解析协议**
 - **7.4 IP协议**
 - **7.5 IP路由和转发**
 - **7.6 ICMP协议**
 - **7.7 IP组播**
 - **7.8 IPv4的可扩展性**
 - **7.9 IPv6协议**
- 
- IPv4**

Chapter 7 Internet & TCP/IP



- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**

Internet基本协议栈





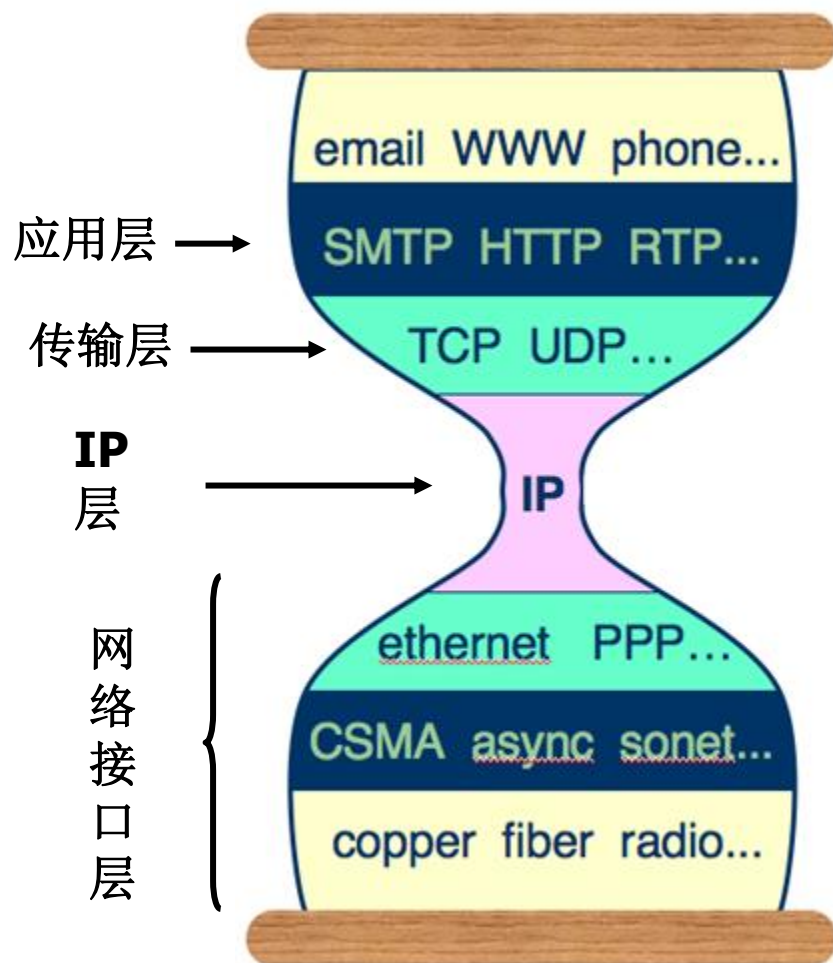
Internet为什么选择IP

- 对于**Internet**发展来说，可扩展性极其重要
 - **Internet**实际上可以看作是千千万万个网络互联而成，这些网络分布在世界各地，由不同的实体管理
 - 可扩展性主要要解决两个问题：**异构性**和**效率**
- **Internet**如何解决可扩展性问题
 - 以**IP为核心**，采用“细腰”体系架构，互联各种异构网络
 - 采用基于**IP的尽力服务模型**，实现简单，保证大规模网络下的传输效率

“细腰”体系架构

- 向下，可以互联采用各种通信技术的网络
- 向上，可以承载各种类型网络业务
- **IP over Everything, Everything over IP**

所有接入到**Internet**的节点都必须运行**IP**协议



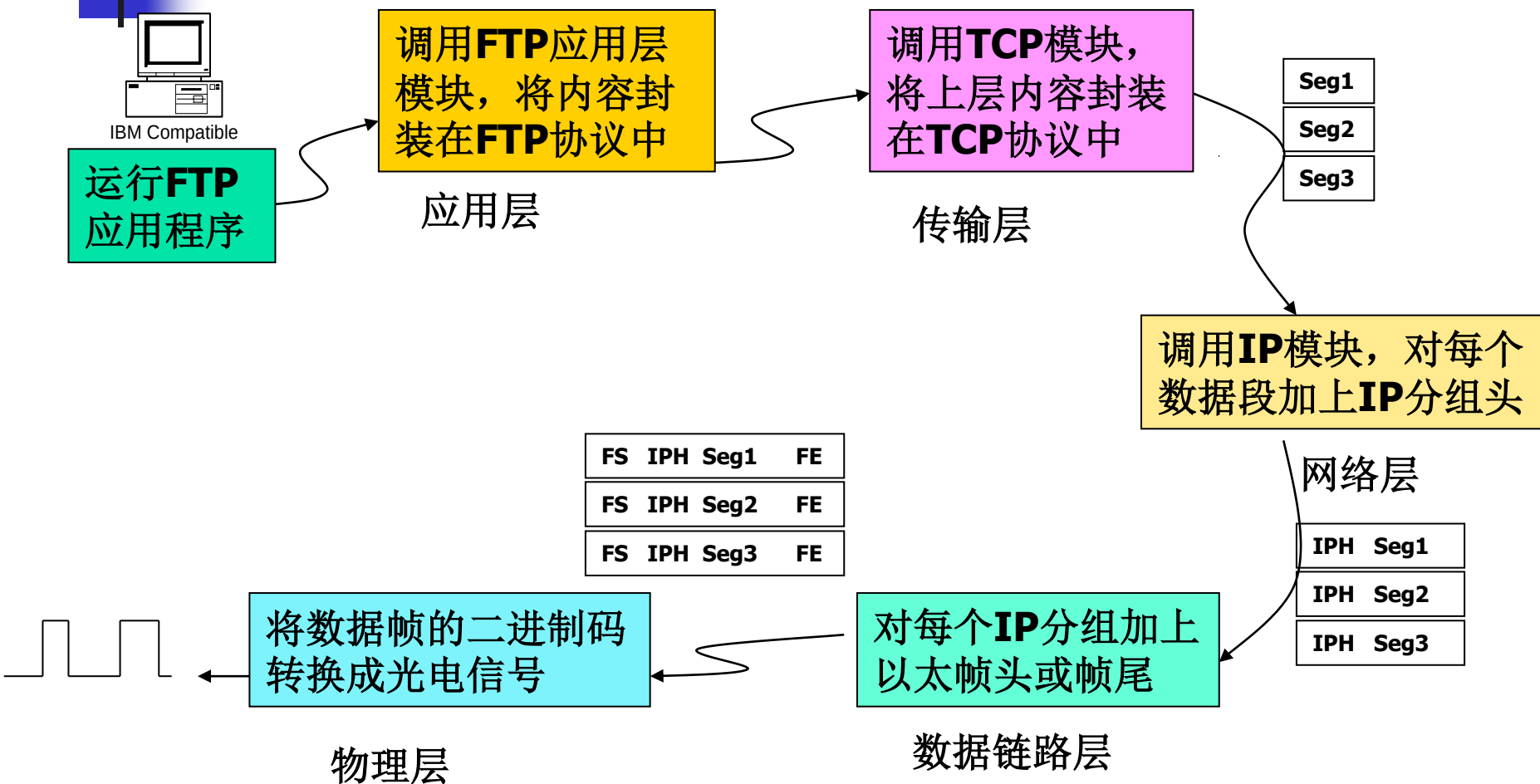


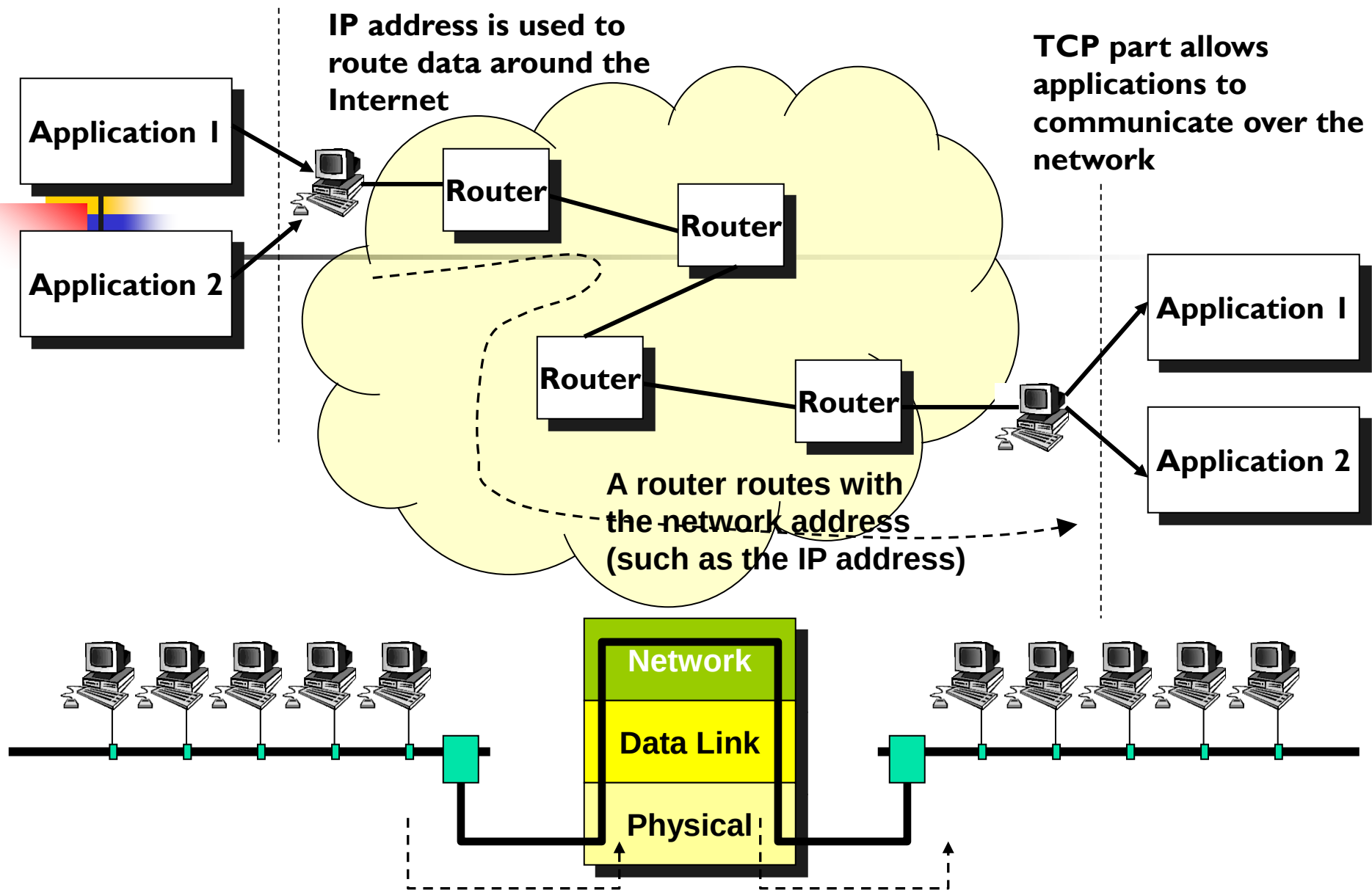
Internet服务模型

- 以**IP**为核心，根据**IP**协议信息转发分组
- **IP**提供的是尽力传送服务（**best-effort**）
 - **不可靠**：不保证分组可靠到达目的地，可能出现分组的丢失
 - 无连接，无确认
 - 乱序：分组可能不是按照发送顺序到达目的地的
 - 分组携带足够的信息使得网络可以将分组传送到目的地，每个分组独立进行转发

设计原则：使网络尽可能的简单

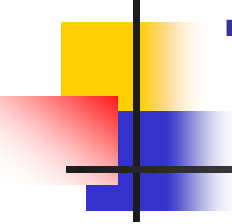
数据封装流程



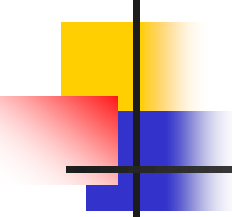


IP协议本质上是定义了一种数据的封装格式，在数据前面加上一些协议相关的控制信息，在这些信息里最重要的就是**IP地址** 8

Chapter 7 Internet & TCP/IP



- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**



IP地址分配

- Internet 上的 IP 地址 由 **IANA**(Internet Assigned Numbers Authority, 互联网编号分配机构)负责分配和管理
- **IANA**向地区性注册机构（如亚太网络信息中心，**APNIC**）分配**IP**地址
- 互联网服务提供商(**ISP**)和企业向这些注册机构申请**IP**地址
- **2011**年**2**月，**IANA**一名官员表示，**IPv4**地址耗尽，呼吁所有互联网行业的成员部署**IPv6**网络

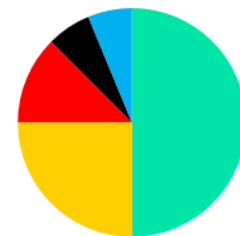
Table 1: The Percentage of IP Address Ownership by Country in 2015



Country Code	Country Name	Total IPs	Percentage	2019 Ranking	2018 Ranking
US	UNITED STATES	1,254,235,289	37.7206%	1	1
CN	CHINA	340,482,033	10.2399%	2	2
JP	JAPAN	207,116,716	6.2289%	3	3
DE	GERMANY	124,725,791	3.7511%	4 	5
GB	UNITED KINGDOM	123,563,721	3.7161%	5 	4
KR	KOREA REPUBLIC OF	112,292,739	3.3772%	6	6
FR	FRANCE	85,297,376	2.5653%	7 	8
BR	BRAZIL	85,238,530	2.5635%	8 	7
CA	CANADA	70,635,390	2.1243%	9	9
IT	ITALY	54,892,973	1.6509%	10	10
NL	NETHERLANDS	51,468,273	1.5479%	11	11
AU	AUSTRALIA	51,082,445	1.5363%	12	12
RU	RUSSIAN FEDERATION	45,243,503	1.3607%	13	13
IN	INDIA	43,846,892	1.3187%	14	14
TW	TAIWAN PROVINCE OF CHINA	35,602,611	1.0707%	15	15
ES	SPAIN	33,228,648	0.9993%	16	16
ZA	SOUTH AFRICA	31,248,905	0.9398%	17	17
MX	MEXICO	29,179,104	0.8775%	18	18

From: <http://www.ip2location.com/reports/internet-ip-address-2019-report>

IP地址格式



■ A类地址
■ B类地址
■ C类地址
■ D类地址
■ E类地址

- IPv4地址由**4个字节（32位）**组成，每个字节之间由点号分割，用十进制表示，称为**点分十进制**表示。

- 分成**5类**：

单播地址

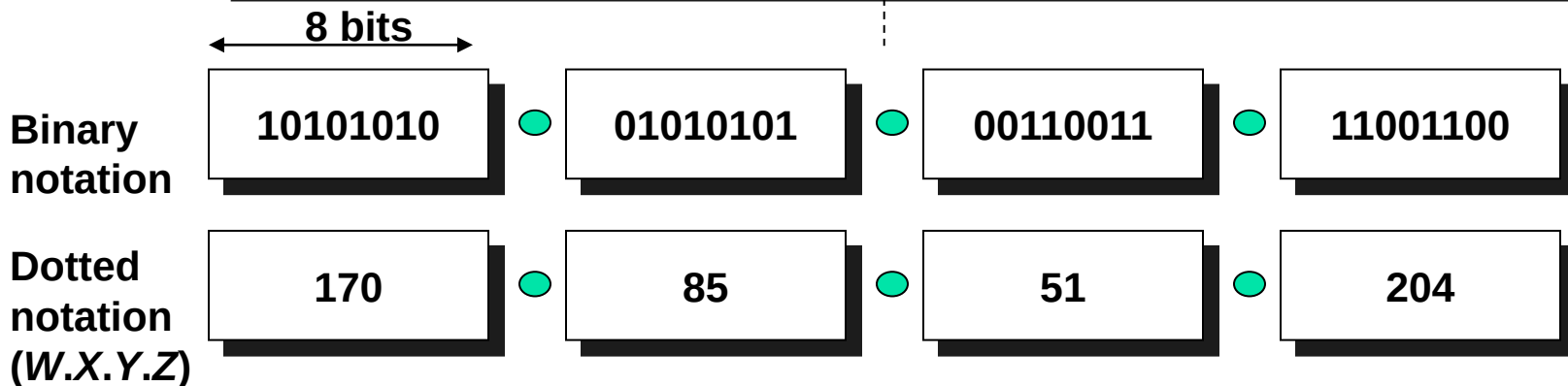
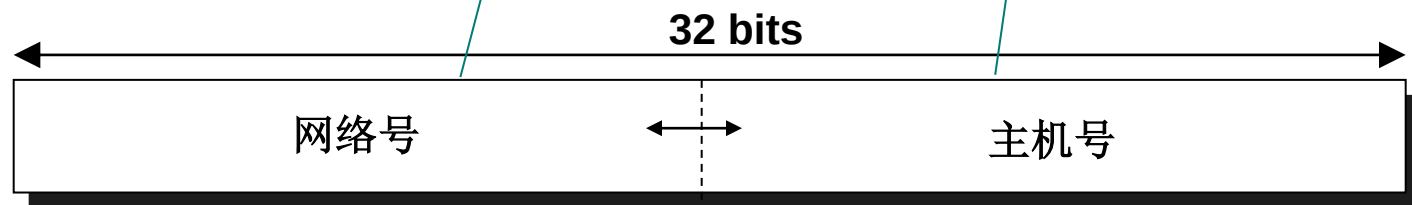
- A类：最高位为**0**，随后**7位**为网络号，最后**24位**表示主机号。可以标识126个A类网络，每个网络可以有 **$2^{24}-2$** （约**1600万**）个主机。
- B类：最高两位**10**，随后**14位**为网络号，最后**16位**表示主机号。可以标识 **$2^{14}-2$** （约**16000**）个B类网络，每个网络可以有 **$2^{16}-2$** （约**65000**）个主机。
- C类：最高三位为**110**，随后**21位**为网络号，剩下**8位**为主机号。可以标识**200万个C类网络**，每个网络最多只能有**254个**主机。

- 组播地址 →
- D类：最高四位为**1110**，是组播地址，标识一个组的地址。
 - E类：最高四位为**1111**，是保留地址。

IP地址格式

标识IP地址所对应节点所在的网络，也被称为**网络前缀**

在网络号所指定的网络中标识IP地址所对应的节点，**主机号的长度决定了网络所能容纳节点的数量！**



Class A	0	NET	●	HOST	●	HOST	●	HOST	(0-127)
Class B	10	NET	●	NET	●	HOST	●	HOST	(128 - 191)
Class C	110	NET	●	NET	●	NET	●	HOST	(192 - 223)
Class D	1110	组	●	播	●	地	●	址	(224-239)
Class E	1111	保	●	留	●	地	●	址	(240-255)

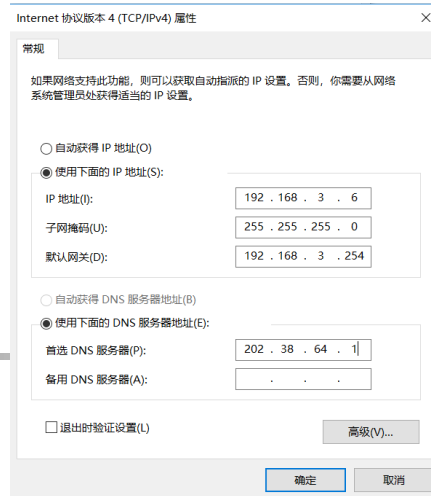
特殊IP地址

- **IP**地址中网络号或主机号为全**0**或全**1**的一般用做特殊处理，不用来标识网络或主机。
- 如：

广播地址	有限广播	全 1	
	定向广播	网络号	全 1
回环地址 (loopback)		127	任意
用作测试	{	全 0	主机号
		全 0	

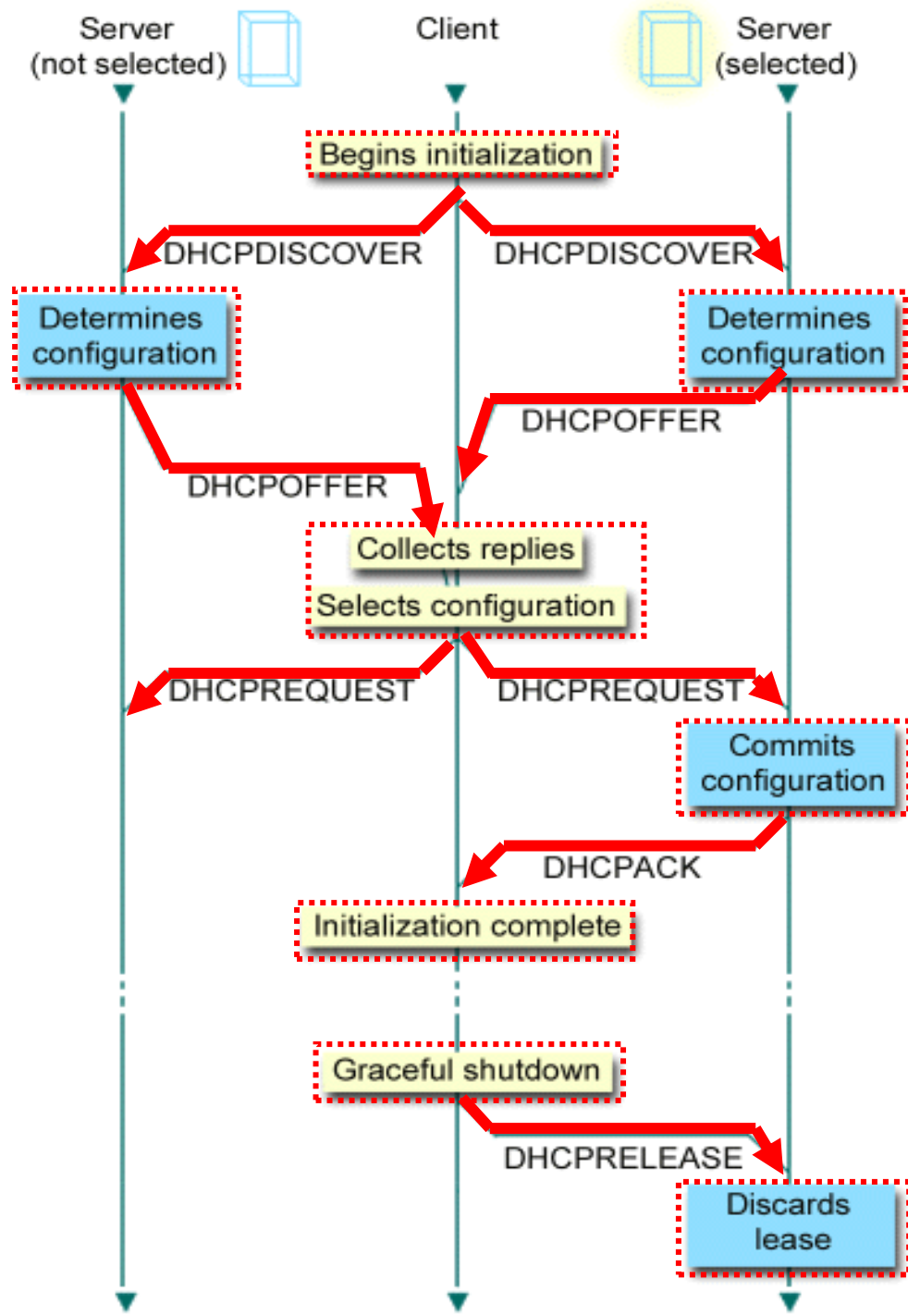
IP地址配置

- 手动配置
- 拨号用户通过PPP协议分配
- 基于DHCP（Dynamic Host Configuration Protocol）的配置
 - 采用客户/服务器模式，消息采用UDP协议封装
 - IP地址分配机制
 - 自动分配：DHCP给客户端分配一个永久的IP地址
 - 动态分配：DHCP给客户端分配一个一段时间内有效的IP地址
 - 手动分配：客户端IP地址由网络管理员分配，DHCP仅用来将分配的地址通报给客户端

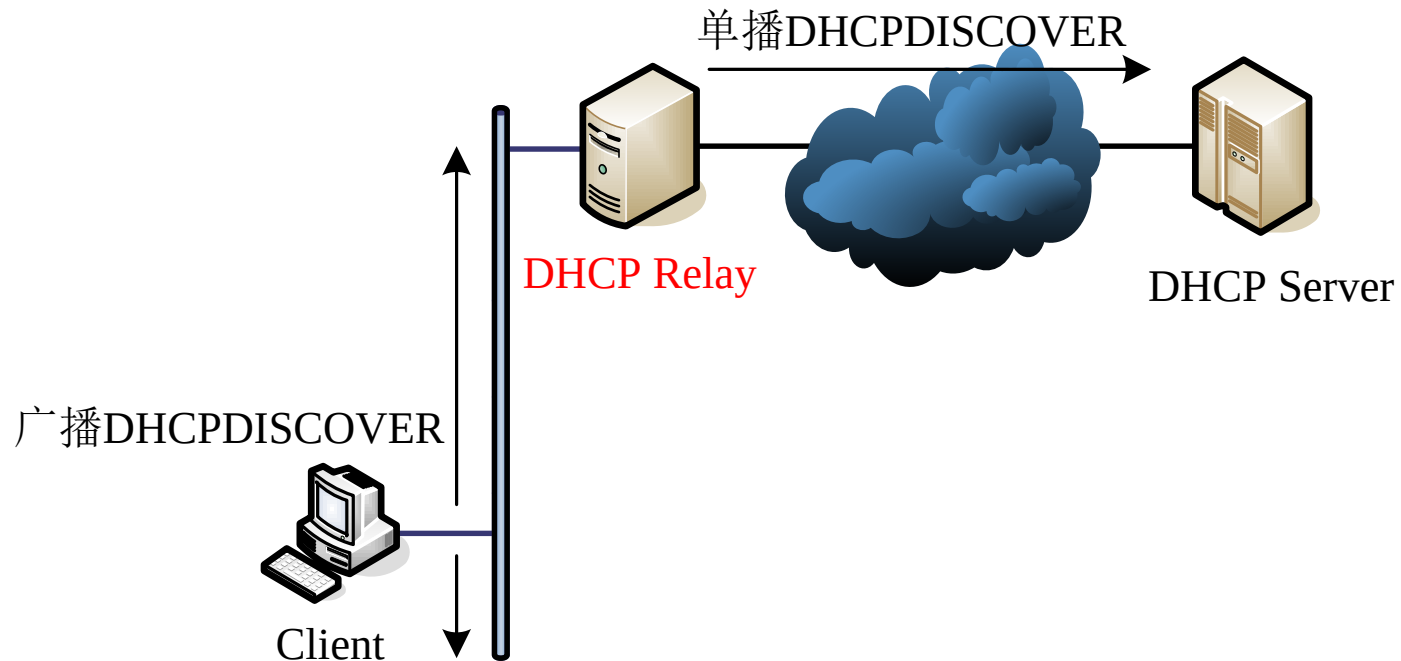


IP地址配置：DHCP过程

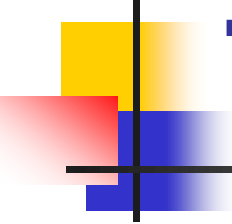
- 1) 客户端广播**DHCPDISCOVER**消息
源IP地址为**0.0.0.0**目的IP地址为**255.255.255.255**，使用**MAC**地址来标识客户端
- 2) **DHCP**服务器回应**DHCPOFFER**消息给客户端，其中包含分配的**IP**地址、租用期限和其它配置参数
- 3) 客户端选择一个**DHCP**服务器并且在整个网络上广播**DHCPREQUEST**消息，告诉其它**DHCP**服务器它选择哪个服务器提供的**IP**地址
- 4) **DHCPREQUEST**选定的服务器给客户端回应**DHCPACK**，其中包含有配置参数
- 5) 客户端接收到**DHCPACK**后，应该对参数进行最后的检查
然后才能开始使用分配的**IP**地址
- 6) 客户端可以通过向**DHCP**服务器发送**DHCPRELEASE**消息来释放对**IP**地址的租用，其中包含有客户端的**MAC**地址和租用**IP**地址



DHCP中继

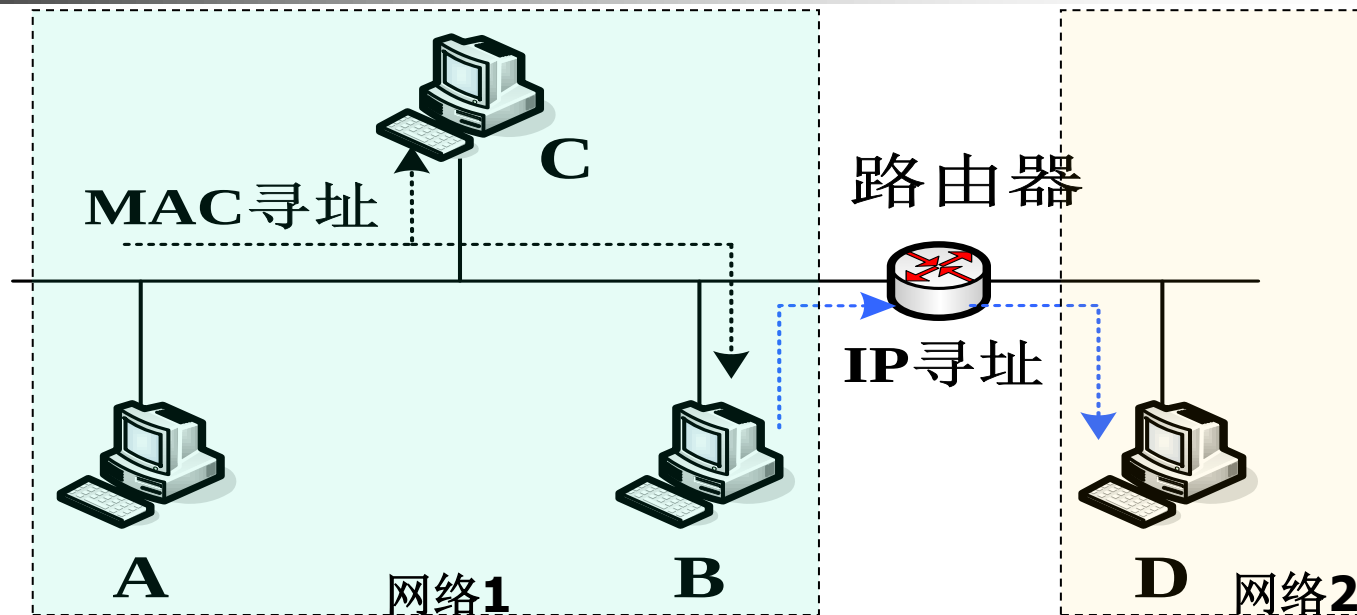


Chapter 7 Internet & TCP/IP



- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**

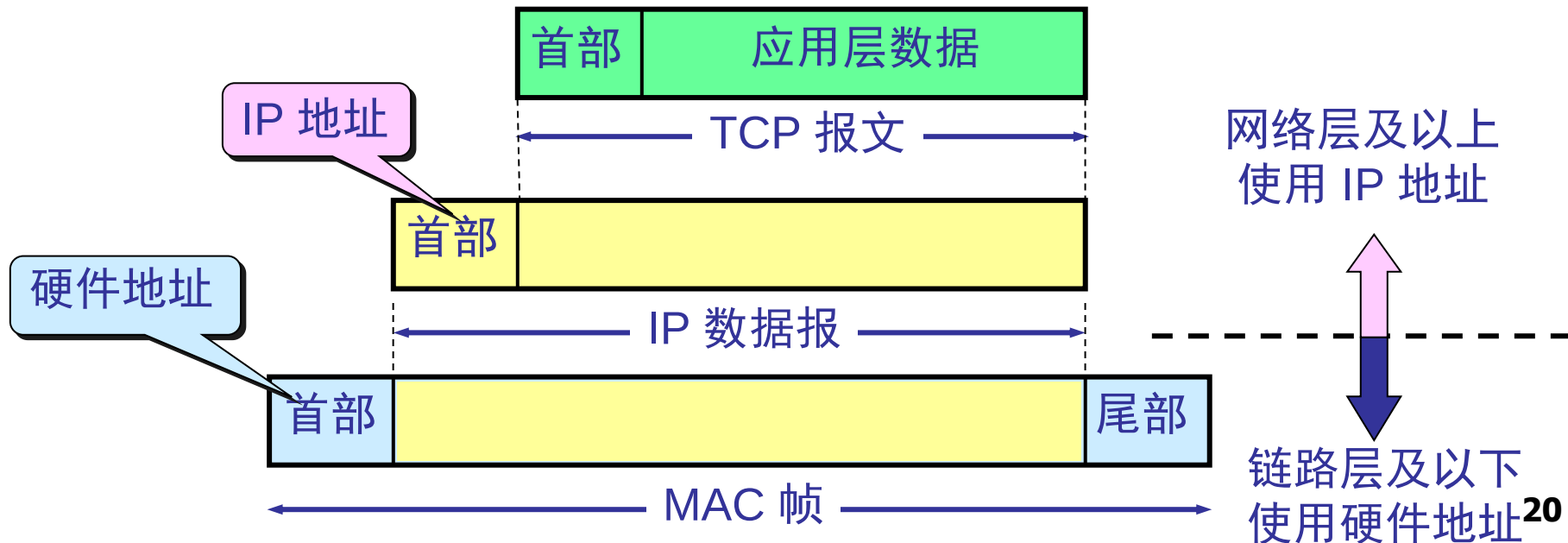
IP地址和MAC地址区别



- **IP**地址在整个网络中（全局）标识不同的节点，而**MAC**地址在链路或者局域网内标识不同的节点
- **IP**地址包含有网络号（前缀），因此节点配置的**IP**地址与其所在的网络相关，而**MAC**地址与节点位置无关，因此一般事先固化在网络设备中

IP地址与MAC地址的区别

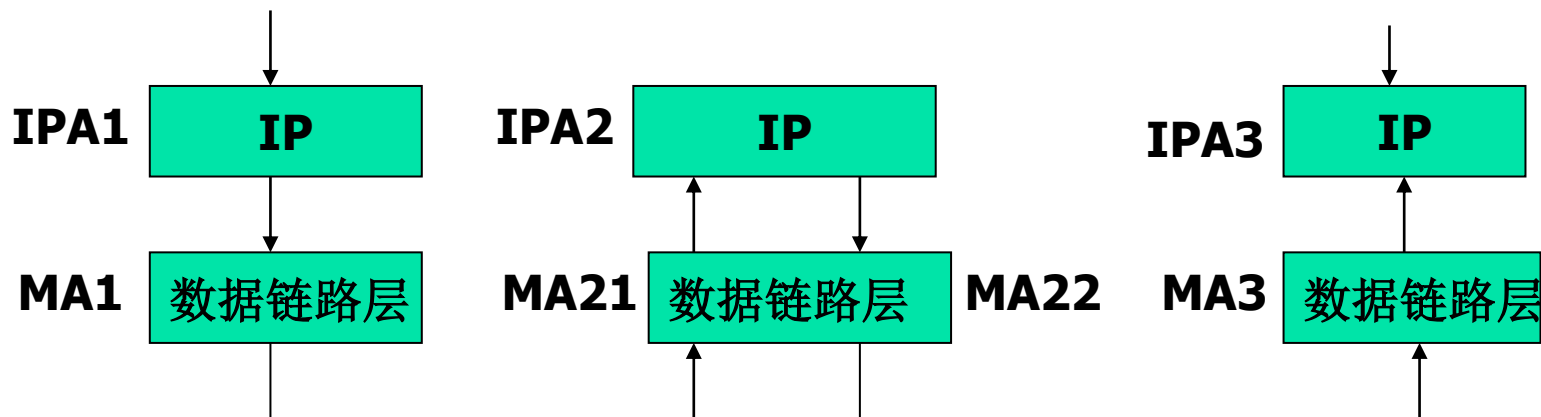
MAC	IP
物理地址（数据链路层地址）	逻辑地址（网络层地址）
局部意义	全局意义
随机获得	上级分配
48位, 如08:00:39:00:2f:c3	32位, 如202.38.64.1



地址解析协议

ARP: Address Resolution Protocol

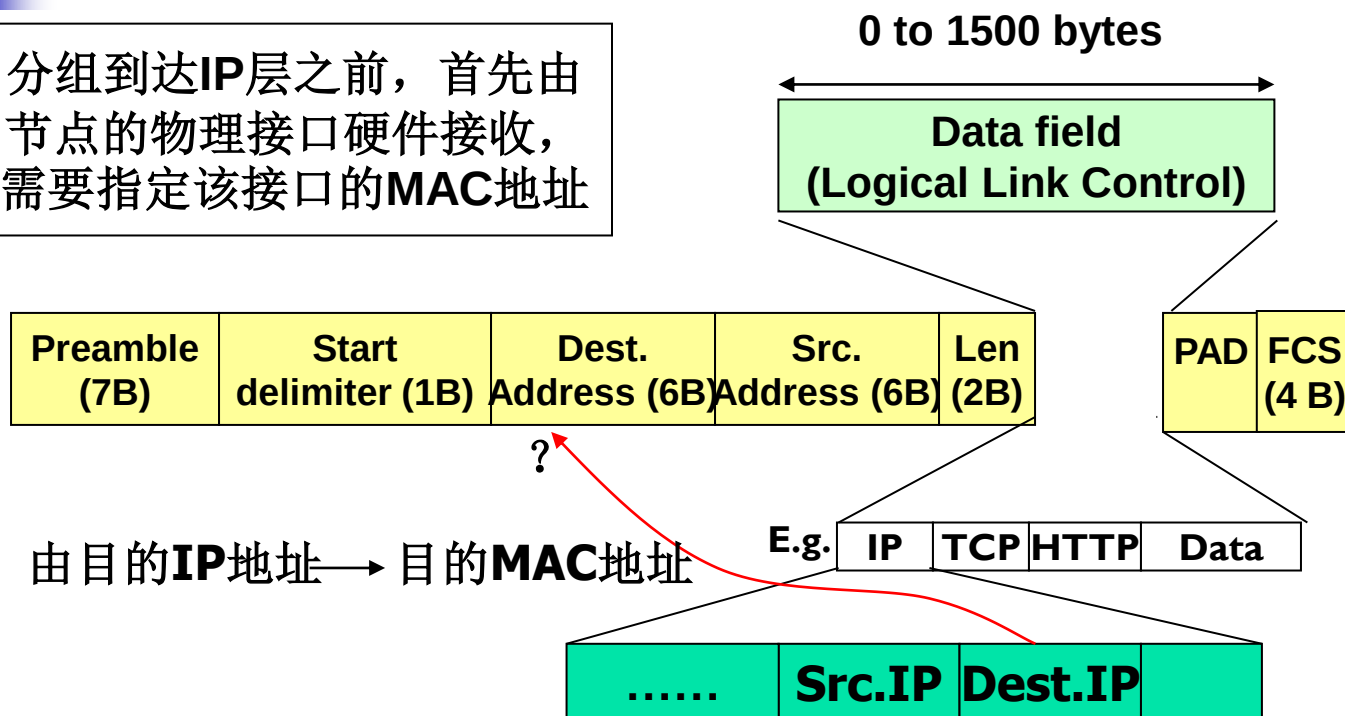
- 数据帧在链路上上传送使用**MAC**地址，源**MAC**地址为发送帧的接口的**MAC**地址，而目的**MAC**地址为同一链路上接收帧的接口的**MAC**地址
- **IP**分组在**Internet**中传输使用**IP**地址，源**IP**地址为发送**IP**分组的主机的**IP**地址，而目的**IP**地址为接收**IP**分组的主机的**IP**地址



数据帧在**不同链路**上传输时，源和目的**MAC**地址**变化**，而源和目的**IP**地址**不变**

地址解析需求—从数据帧的封装来理解

分组到达**IP**层之前，首先由节点的物理接口硬件接收，需要指定该接口的**MAC**地址

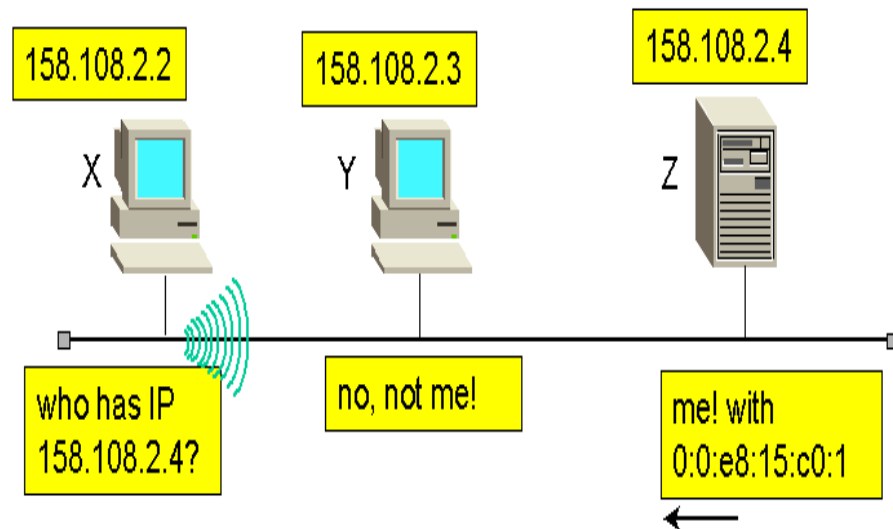


RFC 826: Address Resolution Protocol

地址解析过程

情况1：目的IP地址所对应的主机和发送主机在同一个网络内

- 首先，每个主机都有**ARP缓存**，用来存放一些IP地址与MAC地址的对应关系。主机根据分组头上的目的IP地址查阅自己的**ARP缓存**，如果没查到，就向广播地址发送**ARP请求**。
- 被请求的IP地址所对应的主机返回一个**ARP响应**。
- 主机收到**ARP响应**后，就可发送数据帧，并将该IP地址与MAC地址对存放在**ARP缓存**中



每个节点一般都维护**ARP缓存**，通过ARP缓存可以减少频繁的ARP操作，从而减少网络开销，提高性能



地址解析过程

情况2：目的IP地址所对应的主机和发送主机在两个不同的网络内

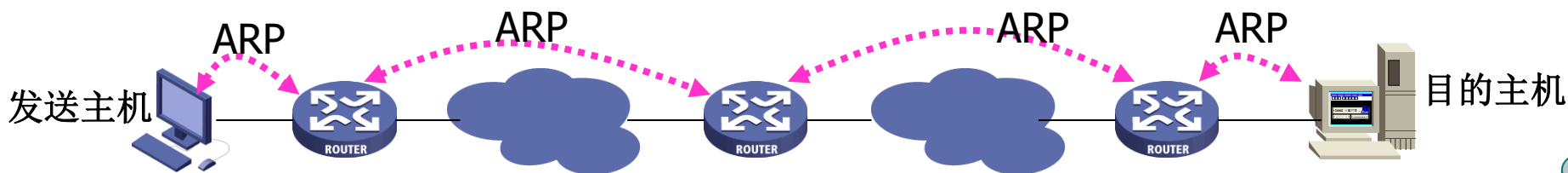
■ 两种方案

- **使用缺省路由**：主机通过识别目的IP地址的网络号，知道它是子网外的主机，直接发给缺省路由器
- **代理ARP**：路由器有ARP代理功能，它代理网络外的主机响应ARP请求，可以实现路由器拦截目的IP地址为其代理的主机的分组

不论哪种方案，目的IP地址对应的MAC地址均为路由器的MAC地址

地址解析过程

- 发送主机通过查找转发表得到**下一跳节点的IP地址**（第一跳路由器），对**该IP地址执行ARP过程**，然后发送数据
- 第一跳路由器查找**转发表**得到第二跳路由器的**IP地址**，对该**IP地址执行ARP过程**，然后发送数据；.....；目的网络的路由器查找转发表知道目的节点和自己在同一个网络中，对目的**IP地址执行ARP过程**，然后将数据发送到目的节点



ARP过程被限制在链路范围(同一个网络)内，始终是查找转发表得到转发的下一跳节点的**IP地址**，然后对该**IP地址做ARP**。在数据传输频繁时，**ARP缓存表中一般有对应的表项**，节点间不需要交互**ARP过程**

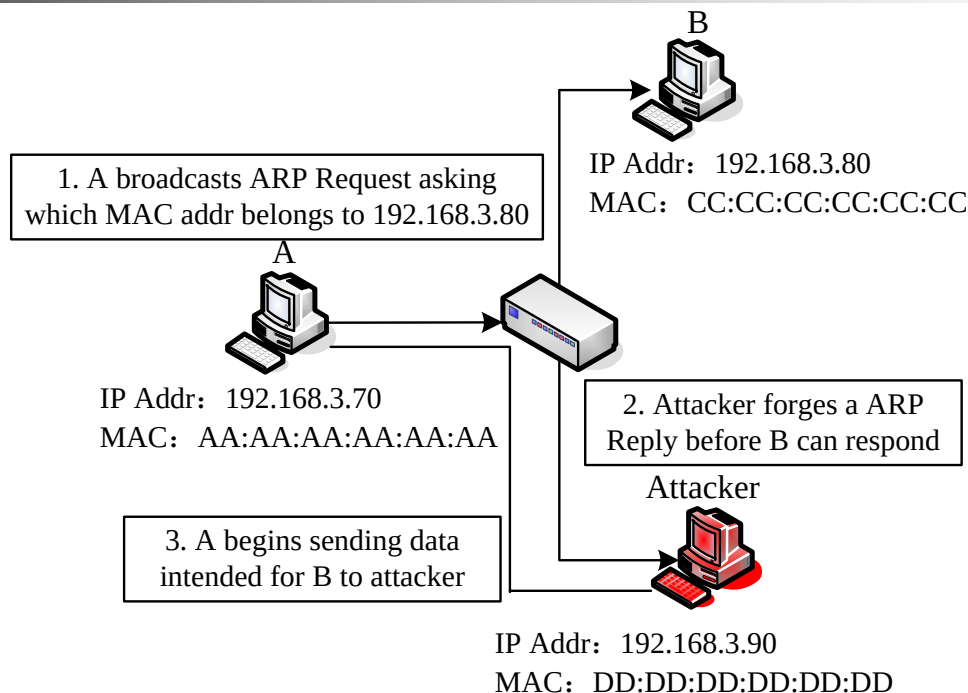
地址解析实例



■ 使用缺省路由

- B根据D的IP地址的网络号知道它不属于**NET N1**，直接将分组发送到缺省路由器**R1**(如果在**ARP Cache**中没有缺省路由器的**MAC**，则需要通过**ARP**过程来获取)
- B发送帧到路由器，帧的**SrcMAC=MB**，**SrcIP=N1B**，**DstMAC=R1**，**DstIP=N2D**
- 路由器查找自己的**ARP Cache**，获得D的**MAC**地址，将分组向D转发，在**N2**网络上的帧的**SrcMAC=R2**，**SrcIP=N1B**，**DstMAC=MD**，**DstIP=N2D**

ARP Spoofing

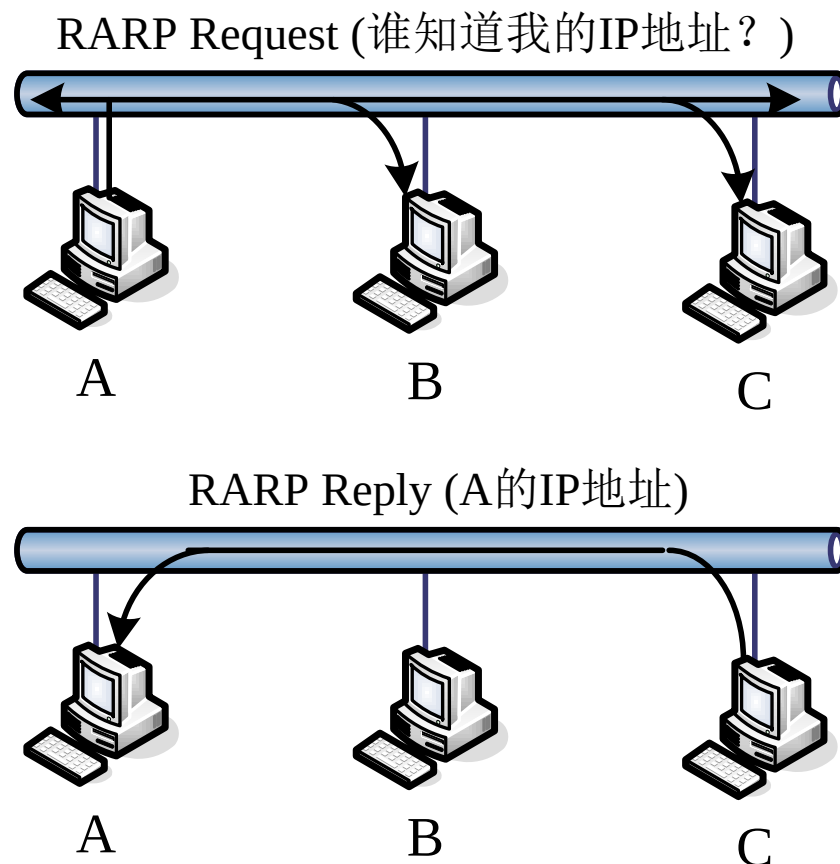


■ 防范措施

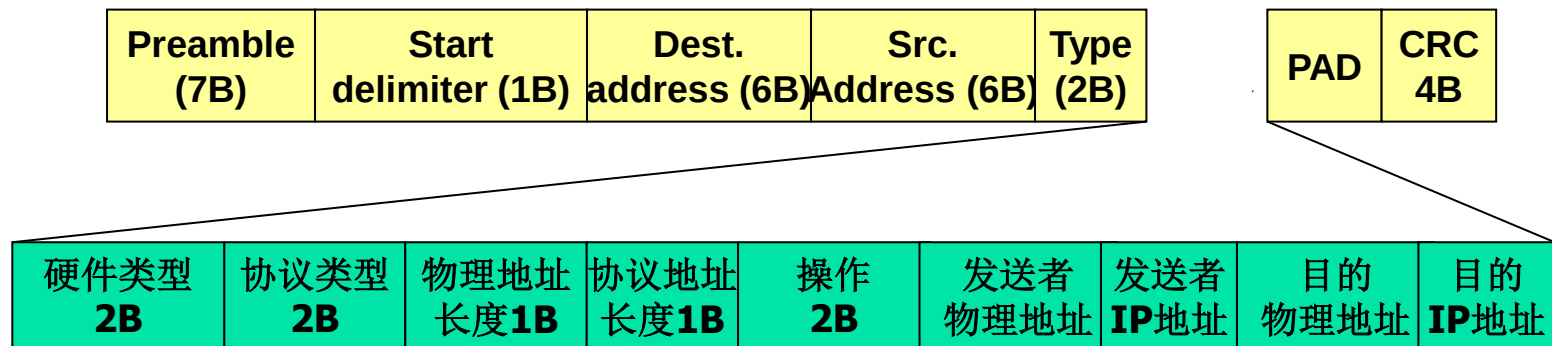
- 监测可疑的**ARP**流量，特别是网关和路由器等关键设备的**MAC**地址的变化。
- 通过划分子网、**VLAN**等措施限制**ARP**的广播域
- 在**ARP Cache**中静态配置**IP**和**MAC**地址的映射表项

反向ARP

- 反向**ARP**:
Reverse ARP
 - **RFC 903**
 - 用于查找物理地址所对应的**IP**地址，例如对于**无盘机**、小型网络服务设备，启动时需要知道自己的**IP**地址



ARP协议帧格式



帧类型（**Type**）：**ARP**请求及响应为**0X0806**

硬件类型：指发送者的网络接口类型，如以太网为**1**

协议类型：指发送者所采用的网络层协议类型，如**IP**协议为**0X0800**

操作：**ARP**请求——**1**，**ARP**响应——**2**，**RARP**请求——**3**，**RARP**响应——**4**

ARP工具

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\kpxue>arp -a

Interface: 192.168.3.66 --- 0x2
    Internet Address      Physical Address      Type
    192.168.3.148         00-50-8d-4c-1f-4f     dynamic
    192.168.3.177         00-11-2f-82-de-91     dynamic
    192.168.3.254         00-e0-4c-40-01-32     dynamic

C:\Documents and Settings\kpxue>
```



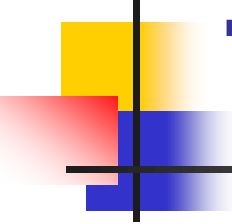
特殊情况

- 目的**IP**地址为广播地址
 - 目的MAC地址为FF:FF:FF:FF:FF:FF
- 目的**IP**地址为组播地址
 - 目的MAC地址：采用地址映射，即将**IP**组播地址的低**23**位映射到**MAC**地址的低**23**位，前面加上**01005E**，再加一个**bit 0**

01	00	5E	0+IP组播地址低23位
----	----	----	--------------

问题： **IP**组播地址是低**28**位不同（最高四位为**1110**），但低**23**位可能存在重复，这样的映射会造成多对一

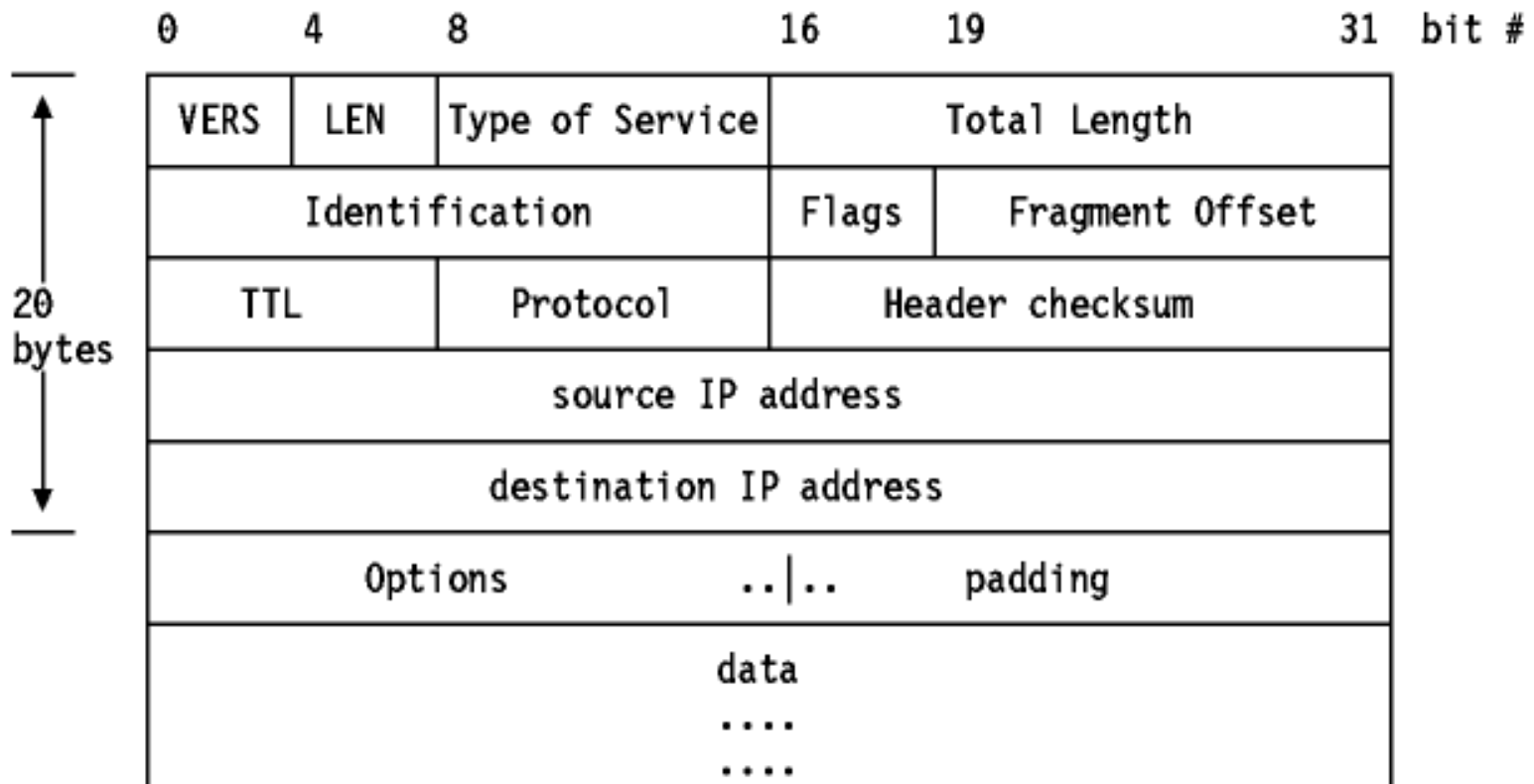
Chapter 7 Internet & TCP/IP



- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**

IPv4头标格式

IP: Internet Protocol

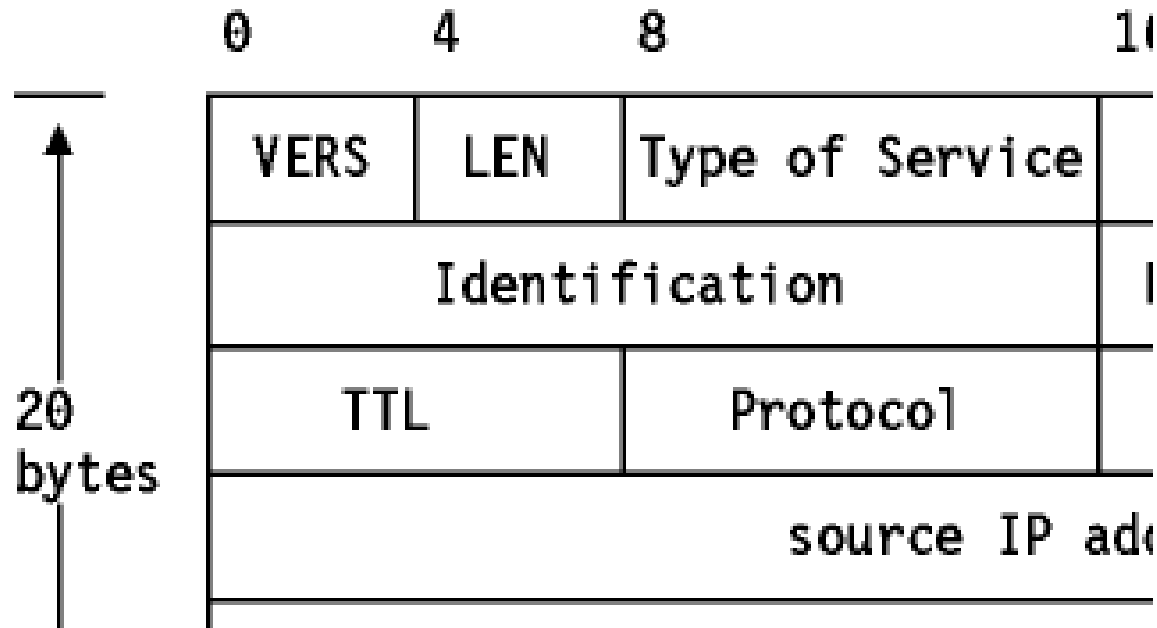


不同教科书中**IP头**、**头标**、**头部**、**首部**指的是同一个意思

版本号与头标长度

版本号（**VERS**）：**4bits**，**IPv4**协议填**4**，**IPv6**协议填**6**。

IP分组头长度（**LEN**）：**4bits**，单位为**4字节**，取值范围**5-15**（确省值为**5**，即标准头标长**20字节**），指示**IP**分组头的长度。



服务类型 (TOS)

优先级 d t r c 0

20 bytes

0	4	8	12
VERS	LEN	Type of Service	
Identification			
TTL		Protocol	
source IP address			

代价: **0**-一般代价; **1**-低代价

可靠性: **0**-一般可靠; **1**-高可靠

吞吐量: **0**-一般; **1**-高吞吐

延时: **0**-一般; **1**-低延时

000 Routine (普通)

001 Priority (优先)

010 Immediate (快速)

011 Flash (闪速)

100 Flash override (疾速)

101 Critical (关键)

110 Internetwork control (网间控制)

111 Network control (网络控制)

TOS域最初的设计并没有得到应用。后来在引入**区分服务**时,这前6个比特又被用来标记分组的服务类别。然而,实际的网络中,服务质量控制并没有在全网范围内流行。

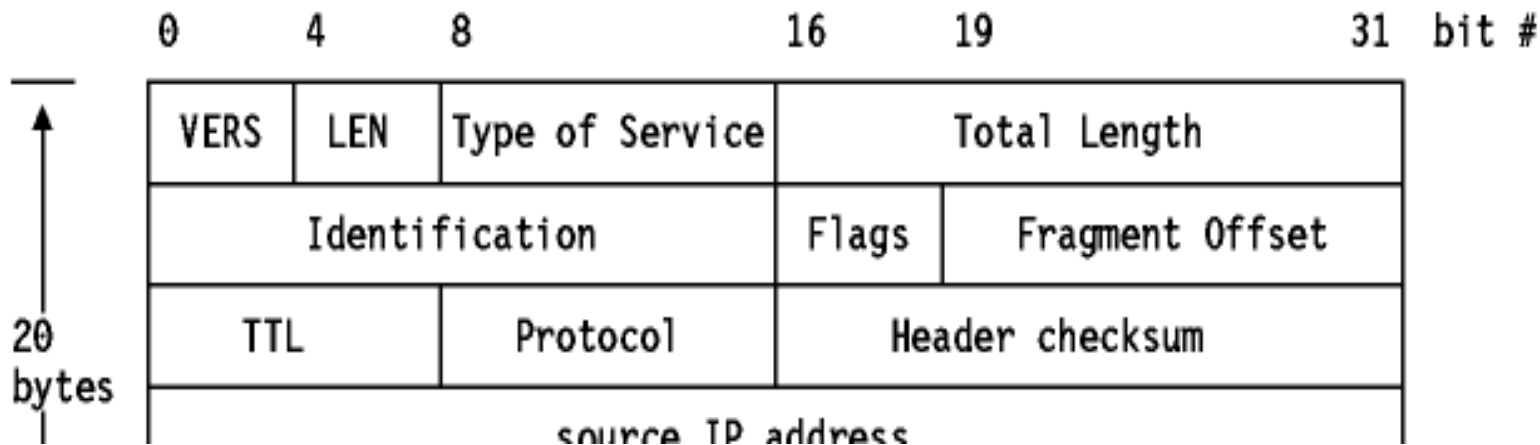
总长度、分段功能

总长度: **16bits** , 单位字节, 描述IP分组的总长 (包括头和数据), 最大分组长度为**65535**字节。

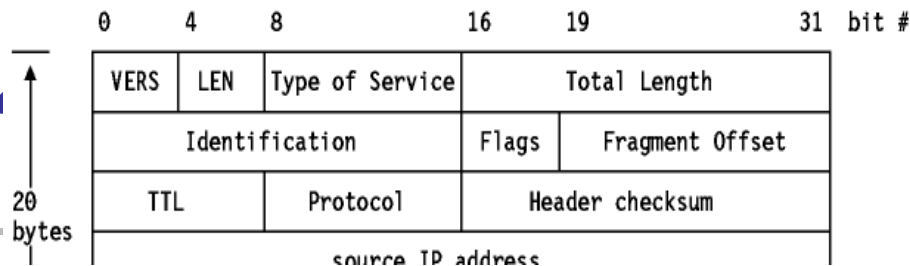
标识符: **16bits**, 用于唯一标识该分组。

标志: **3bits**, 第1位未定义, 第2位(**DF: Don't Fragment**)为**0**表示该分组可分段, 为**1**时表示不可分段; 第3位(**MF: More Fragment**)为**0**表示这是最后分段, 为**1**时表示还有后续分段。

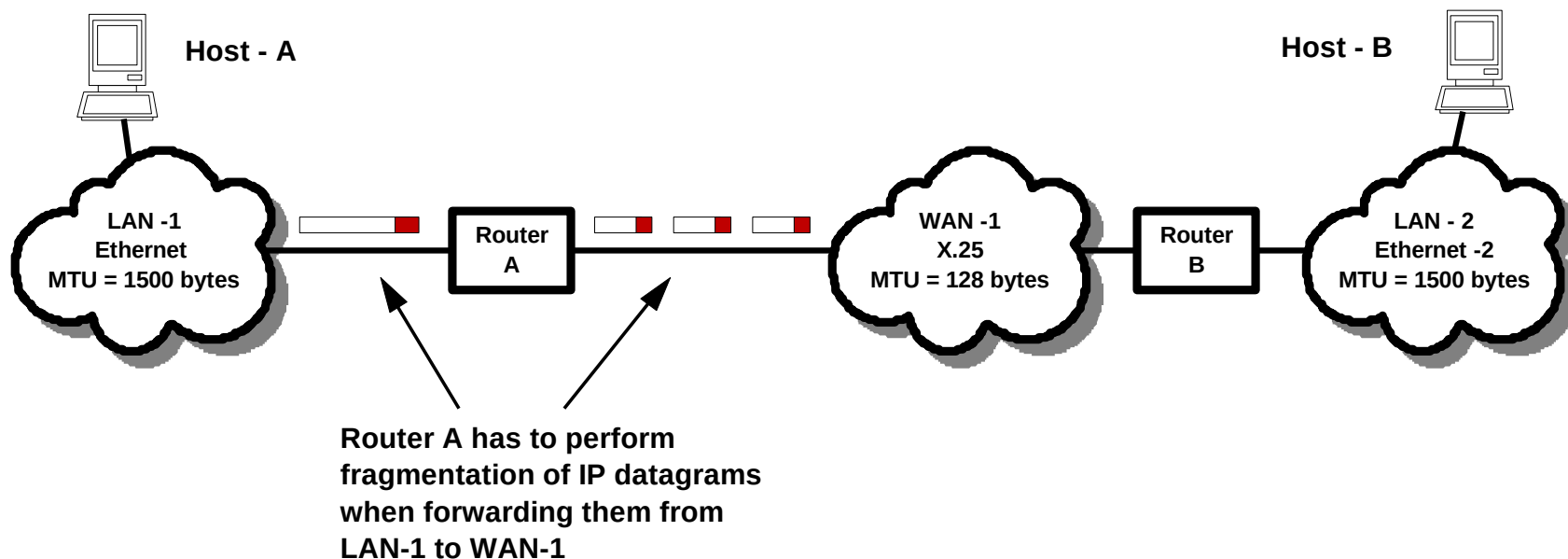
段偏移: **13bits**, 单位**8字节**。取值**0-8191**, 标明当前分段数据开始部分在原分组中数据部分的相对位置。



为什么要分段

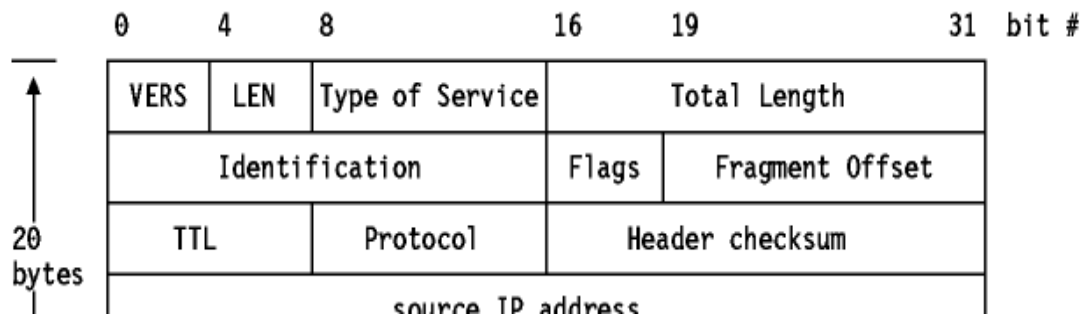


Host - A in LAN -1 is communicating with Host - B in LAN -2 using IP



MTU: Maximum Transmission Unit, 最大传输单元, 即一帧所能携带的最大IP分组, 包括IP头标。分段在路由器进行, 但分段后的重组只在目的端进行。

怎样分段？



MTU=620字节



段偏移=0



段偏移=75

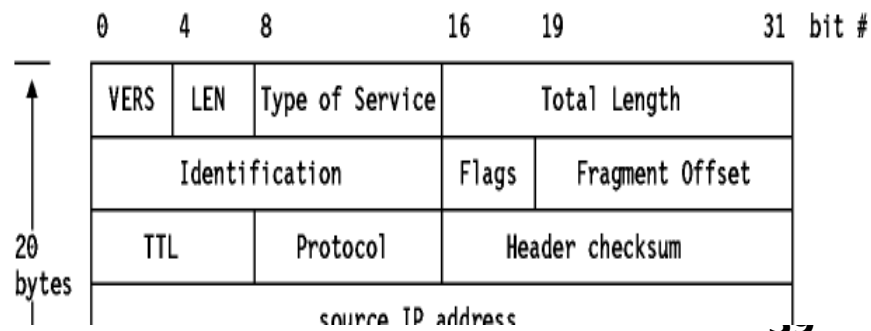


段偏移=150

MTU包括IP头标，此例中IP头标取固定长度20字节

TTL、Protocol

- 生存时间(**TTL: Time to Live**): **8bits**, 单位秒, 表示分组的生存时间。实际操作时, 分组每经过一个路由器, **TTL**值减一, 当**TTL**值为**0**时, 该分组被丢弃
 - 不同类型的操作系统设置默认值可能不一样
 - **64.121 unix: 255;**
 - **64.42windows: 127;**
 - **64.1linux: 64**
- 协议 (**Protocol**): 表示高层协议类型。
 - **0 Reserved**
 - **1 Internet Control Message Protocol (ICMP)**
 - **2 Internet Group Management Protocol (IGMP)**
 - **3 Gateway-to-Gateway Protocol (GGP)**
 - **4 IP (IP encapsulation)**
 - **5 Stream**
 - **6 Transmission Control (TCP)**
 - **8 Exterior Gateway Protocol (EGP)**
 - **9 Private Interior Routing Protocol**
 - **17 User Datagram (UDP)**
 - **89 Open Shortest Path First(OSPF)**



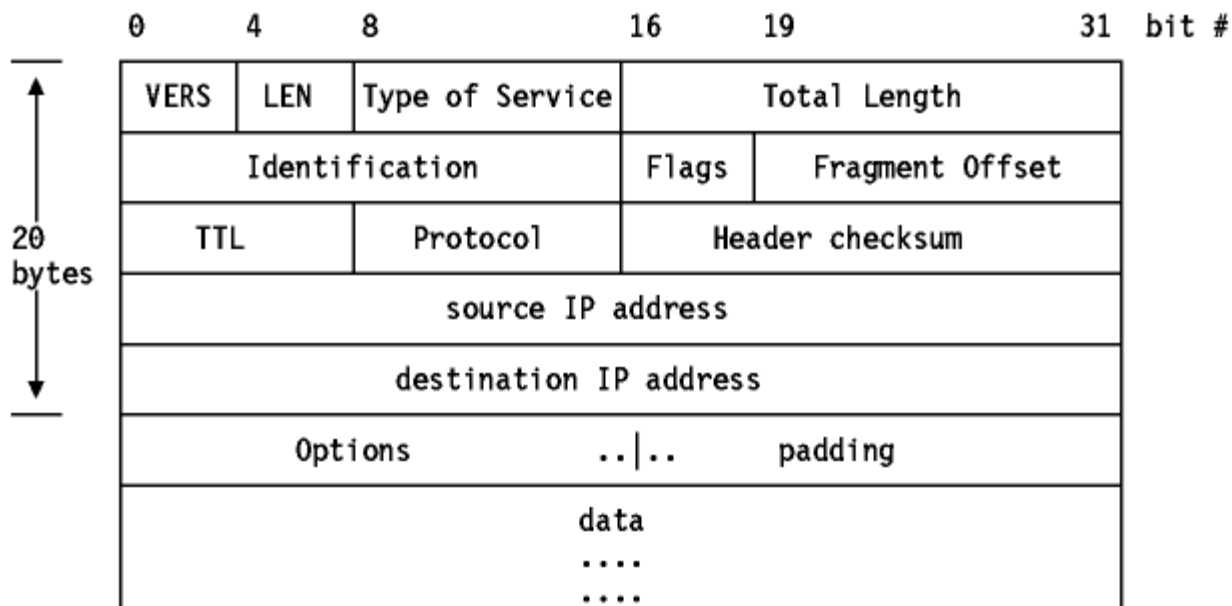
分组头校验、源/目的地址

分组头校验：**16bits**，用来检验**IP头标**在传输过程中是否被破坏。

源地址：**32bits**，分组发送者的**IP**地址。

目的地址：**32bits**，分组接收者的**IP**地址。

填充（**padding**）：分组头长度必须为**4字节**的整数倍，如果选项的长度不是**4字节**的整数倍，那么就要进行填充。



校验和计算过程

- 计算初始化
 - 校验和设置为0
- 计算过程

注意

- 1) 校验范围只包括IP头标
- 2) 由于在分组传输过程中的TTL值会变化，因此在路由器每次都必须重新计算校验和

```
u_short cksum(u_short *buf, int count)
{
    register u_long sum = 0 ;
    while (count--)
    {
        sum += *buf++;
        if(sum & 0xFFFF0000)
        {
            //产生进位，加到最后一位
            sum &= 0xFFFF;
            sum++;
        }
    }
    return ~(sum & 0xFFFF);
}
```


IP: ...0 = normal delay
 IP: 0... = normal throughput
 IP:0.. = normal reliability
 IP:0. = ECT bit - transport protocol will ignore the CE bit
 IP:0 = CE bit - no congestion
 IP: Total length = 52 bytes
 IP: Identification = 8143
 IP: Flags = 4X
 IP: .1... = don't fragment
 IP: ..0. = last fragment
 IP: Fragment offset = 0 bytes
 IP: Time to live = 64 seconds/hops
 IP: Protocol = 6 (TCP)
 IP: Header checksum = 3CA9 (correct)

00000000: 00 01 30 ff b7 c0 00 fd 00 00 1b 45 08 00 45 00
 00000010: 00 34 1f cf 40 00 40 06 3c a9 da f6 bc fd da 1e
 00000020: 6c 39 06 2f 00 50 c5 2a fa 5a 00 00 00 00 80 02
 00000030: ff ff ca c5 00 00 02 04 05 b4 01 03 03 02 01 01
 00000040: 04 02

0	4	8	16	19	31
VERS	LEN	Type of Service	Total Length		
Identification			Flags	Fragment Offset	
TTL		Protocol	Header checksum		
source IP address					
destination IP address					

45 00
 00 34
 1F CF
 40 00
 40 06
 00 00
 DA F6
 BC FD
 DA 1E
 + 6C 39

 3 C3 53

把进位3加到结果后: $3 + C3\ 53 = C3\ 56$

C 3 5 6
 1100 0011 0101 0110 求反码: 0011 1100 1010 1001 得到 3CA9

选项（option，可变部分）

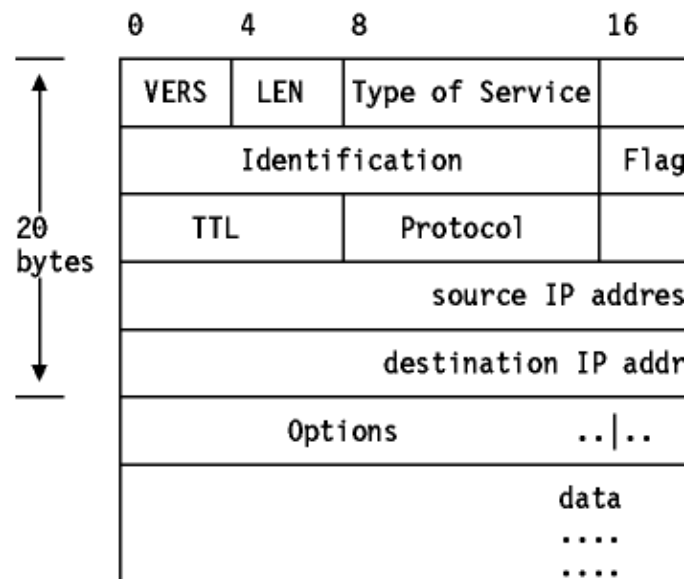
- 用于提供排错、测量、安全等措施，1-40字节
 - 安全性（Security）：指明分组的机密性
 - 严格的源路由选择（Strict Source routing）：给出分组经过的完整路由
 - 松散的源路由选择（Loose Source routing）：给出分组经过的某些路由器列表
 - 路由记录（Route recording）：使每个路由器都附上它的IP地址
 - 时间标记（Time stamping）：使每个路由器都附上它的IP地址和时间标记

优点：增加了IP数据报的功能，具有可选择性

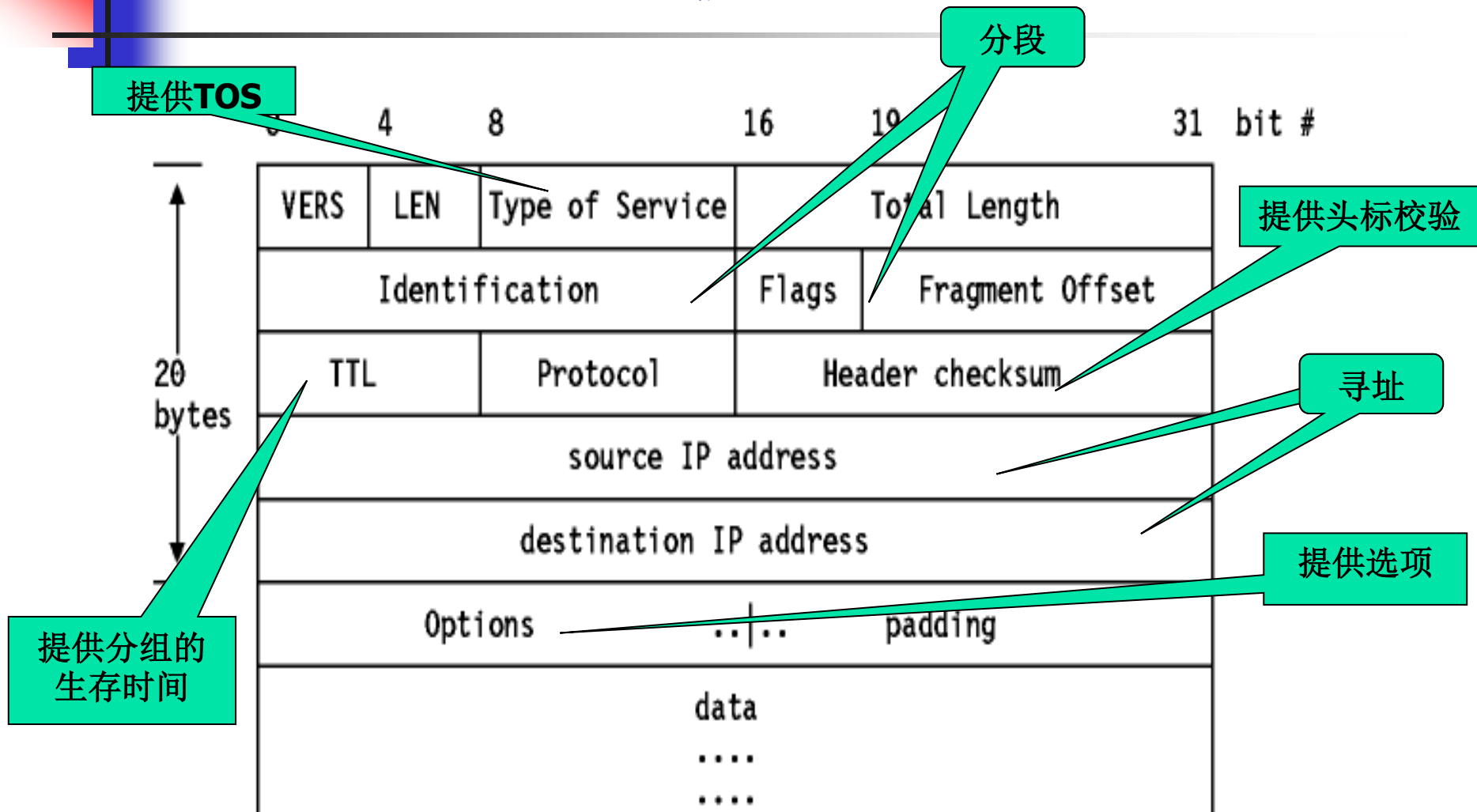
缺点：IP数据报首部长度的可变，增加了每个路由器处理开销，但实际上很少选用选项

TLV格式

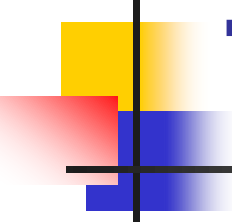
TYPE (1B)	LENTH (1B)	VALUE (nB)
--------------	---------------	---------------



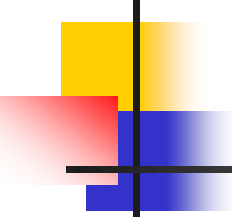
IPv4协议提供的功能总结



Chapter 7 Internet & TCP/IP



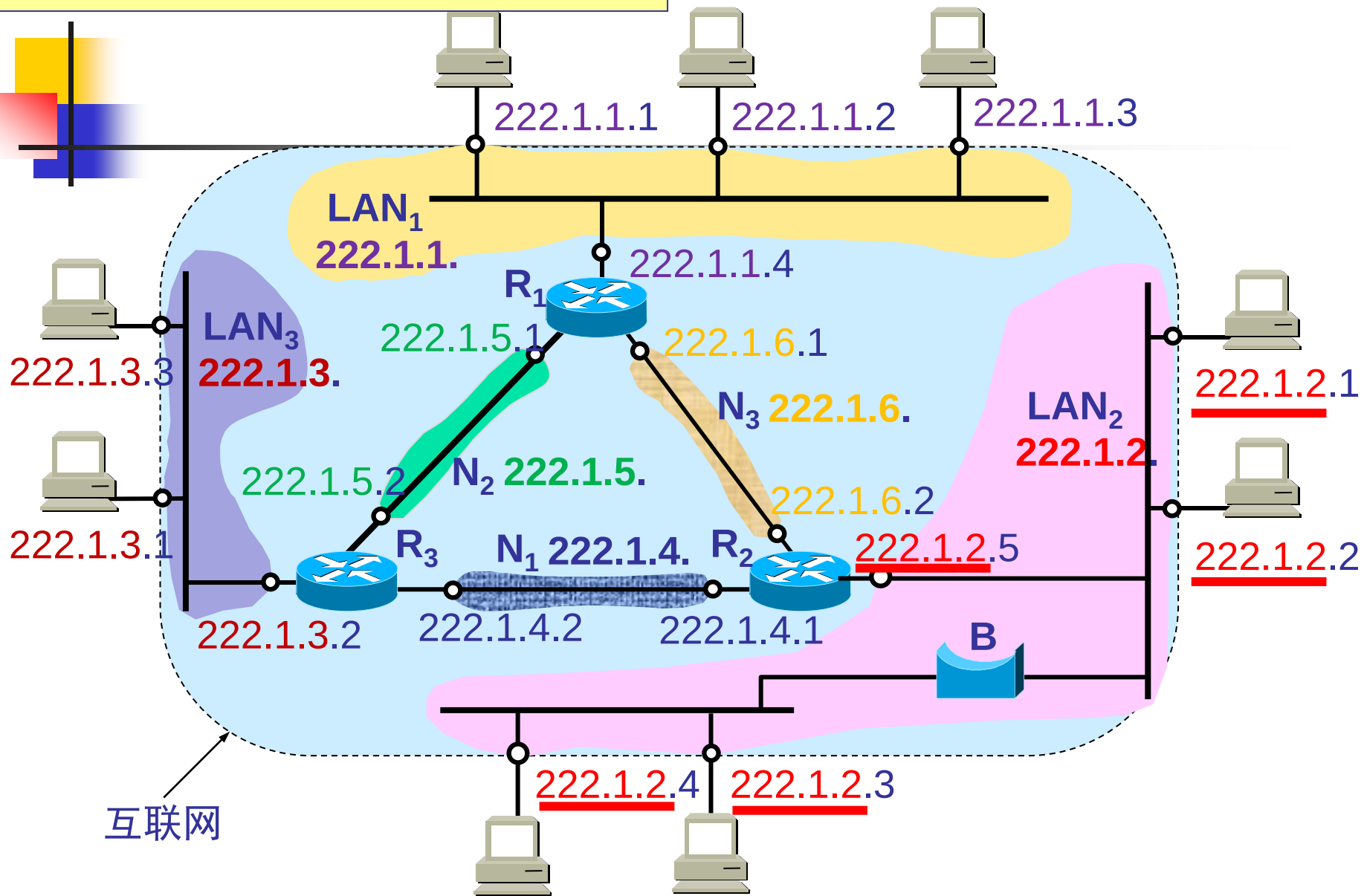
- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**

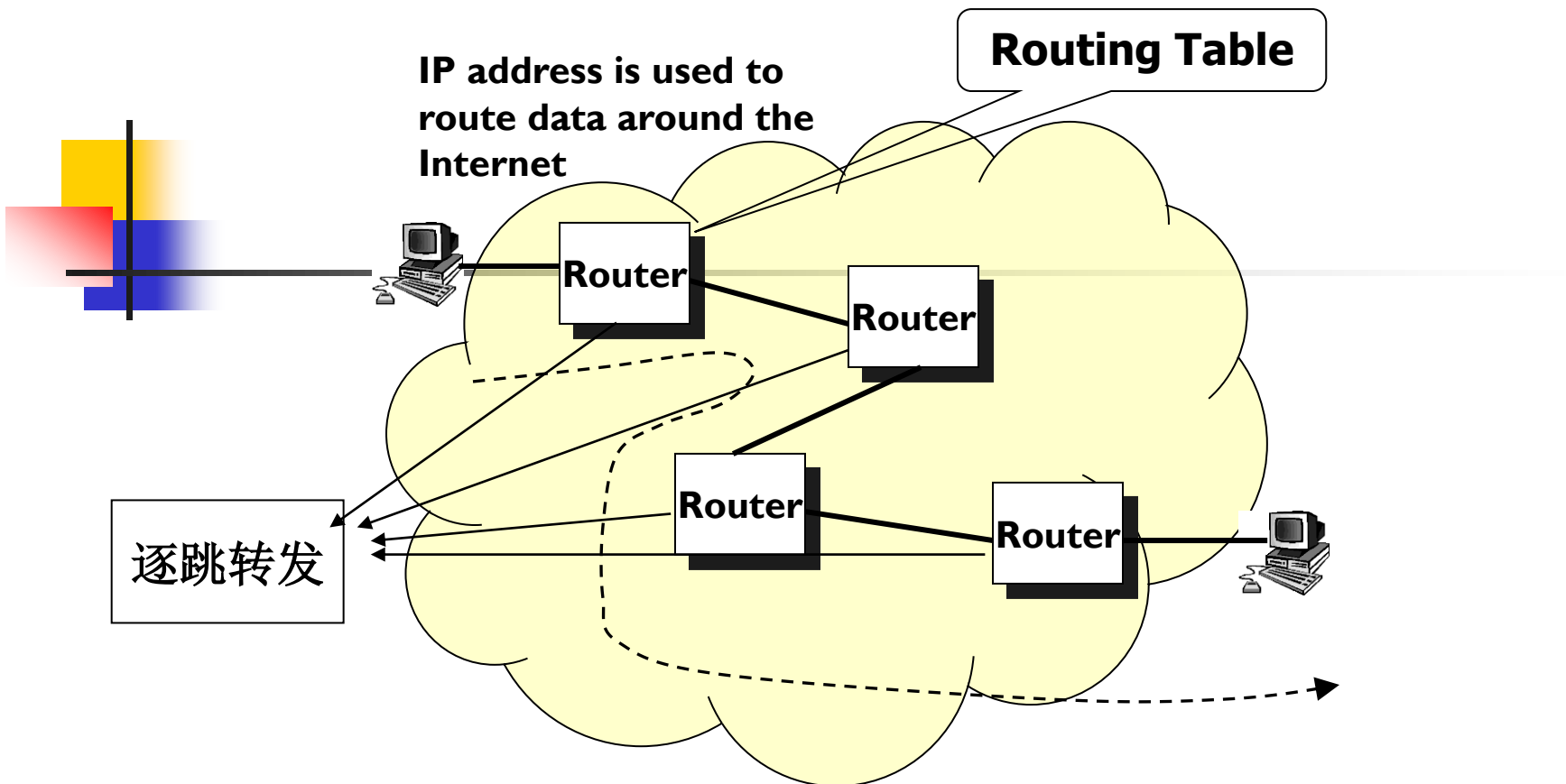


IP寻址 (IP addressing)

- IP 地址分为网络号+主机号
 - 地址管理机构只分配网络号，主机号由得到该网络号的单位自行分配。便于 IP 地址的管理。
 - 路由器只需根据目的主机的网络号来转发分组，可以减少路由表项从而减小路由表所占的存储空间。
- 一个 **IP** 地址标识一台主机，或路由器上的一个接口。
- 当一个主机同时连接到两个网络上时，该主机原则上同时具有两个相应的 **IP** 地址，这种主机称为多接口主机 (multihomed host)。
- 由于一个路由器至少应当连接到两个网络，因此原则上一个路由器至少应当有两个不同的 **IP** 地址。

在同一个局域网上的主机或路由器的
IP 地址中的网络号必须是一样的。
图中的网络号就是 IP 地址中的 net-id





1. 在**Internet**中，数据从源主机发送到目的主机，中间可能要经过多个路由器
2. 路由器根据传输数据的**IP**协议信息为其选择转发的下一跳网络节点，也就是说路由器是网络层（**IP**层）设备，而传输的数据为**IP**分组
3. 具体来说，路由器根据接收到的**IP**分组头标中的目的地址查找路由表，从而确定其转发的下一跳网络节点

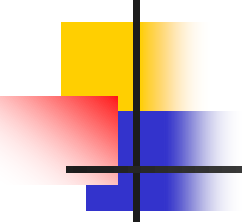


分组转发

- 路由表：**Routing Table**
 - 主机或者路由器根据分组的**目的IP地址**查找**路由表**，找出分组的下一跳节点。
 - 每台主机或者路由器上都有一个路由表
- 路由表每个表项至少包含两个内容
 - 目的网络(网络号)/目的主机的地址
 - 下一跳（**Next Hop**）地址
- 路由表可由管理员手工配置（静态路由表），也可通过路由协议动态建立和维护（动态路由表）
 - 静态路由一般用于主机，动态路由一般用于路由器
- 路由表的表项有显式路由和缺省路由两种，缺省路由可以缩小路由表的表项

转发表Forwarding Table：

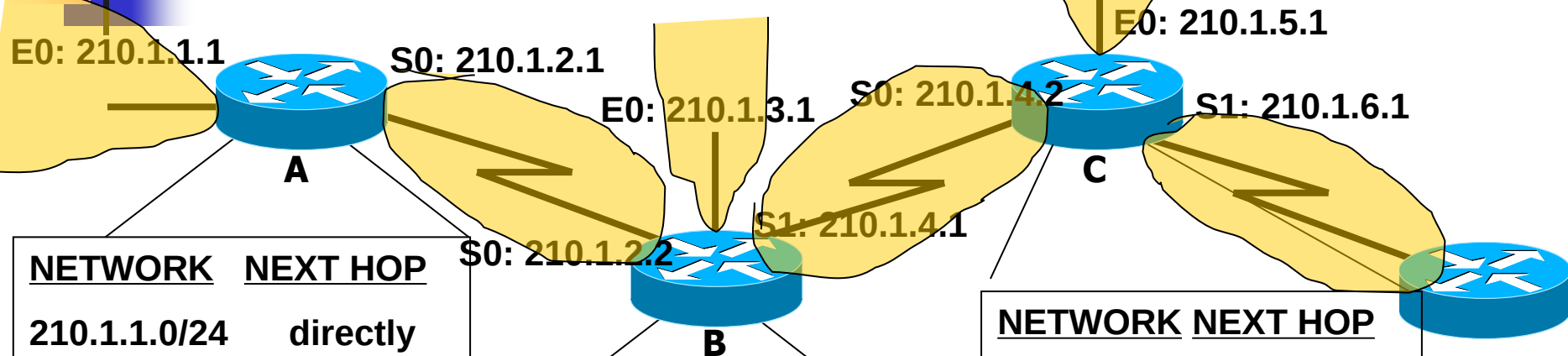
与路由表相对应的是转发表的概念，包含目的网络的网络号/目的主机的地址、输出接口以及一些**MAC**地址等信息



路由表	网络号	下一跳	
	10或者10.0.0.0/8	171.69.245.10	
转发表	网络号	接口	MAC地址
	10 或者10.0.0.0/8	0	00:02:2b:e4:b:1:2

路由表是建立转发表的前奏，也就是说转发表基于路由表的信息建立的。在以后的叙述中，如果不特别说明，我们将不区分路由表和转发表

路由表实例

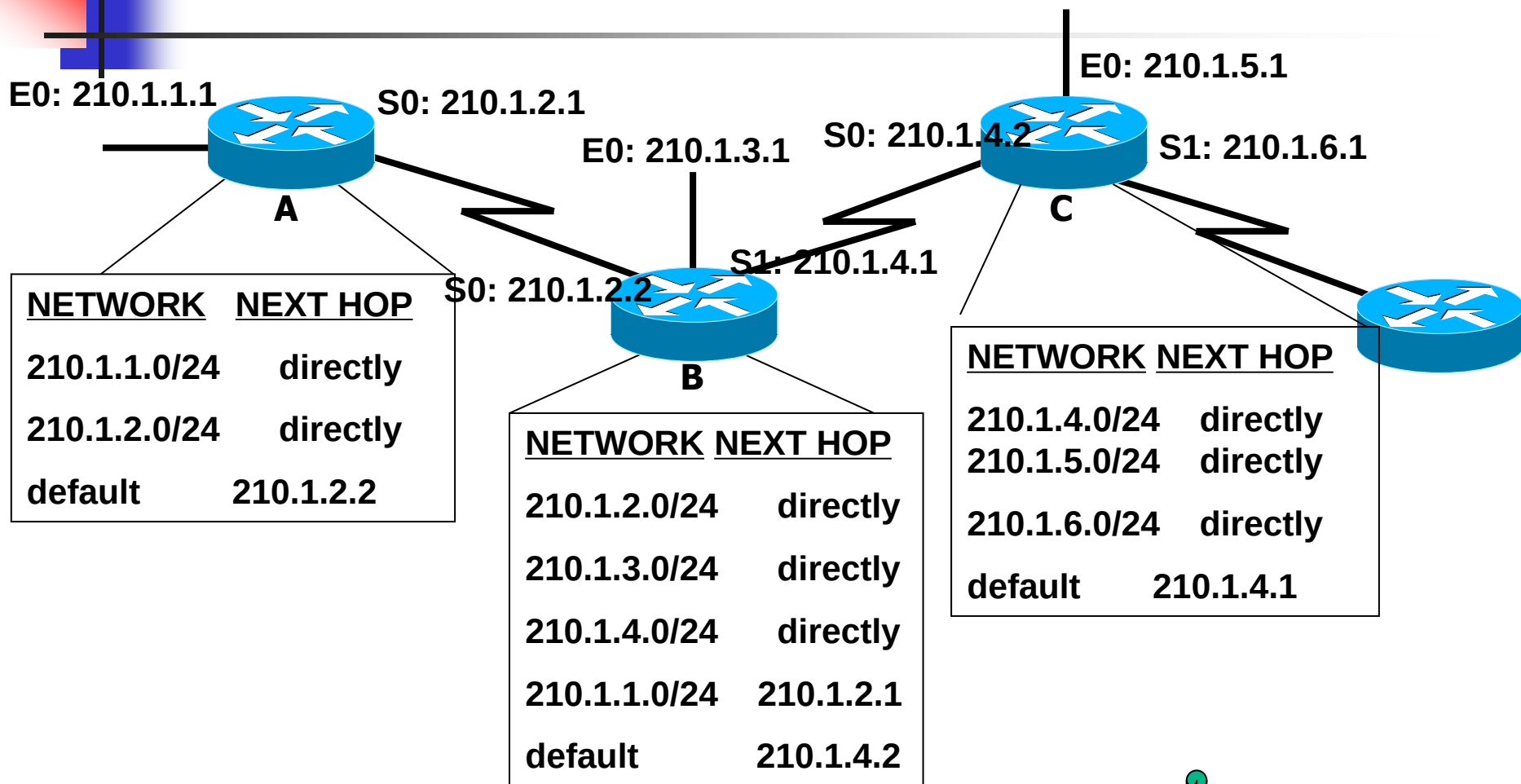


NETWORK	NEXT HOP
210.1.1.0/24	directly
210.1.2.0/24	directly
210.1.3.0/24	210.1.2.2
210.1.4.0/24	210.1.2.2
210.1.5.0/24	210.1.2.2
210.1.6.0/24	210.1.2.2

NETWORK	NEXT HOP
210.1.2.0/24	directly
210.1.3.0/24	directly
210.1.4.0/24	directly
210.1.1.0/24	210.1.2.1
210.1.5.0/24	210.1.4.2
210.1.6.0/24	210.1.4.2

NETWORK	NEXT HOP
210.1.4.0/24	directly
210.1.5.0/24	directly
210.1.6.0/24	directly
210.1.1.0/24	210.1.4.1
210.1.2.0/24	210.1.4.1
210.1.3.0/24	210.1.4.1

缺省路由



通过缺省路由可以减少路由表表项



查看路由表

- **Linux**下面
 - **route (ip route show)**
- **Windows**下面
 - 命令行下运行**route print**

```
[root@infonet root]# ip route show
192.168.6.0/24 via 202.38.75.75 dev eth0
192.168.3.0/24 dev eth1 scope link
202.38.75.0/24 dev eth0 scope link
127.0.0.0/8 dev lo scope link
default via 202.38.75.254 dev eth0
[root@infonet root]#
```

下一跳地址

用**IP**地址表示的接口

Active Routes:

Network	Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	0.0.0.0	192.168.3.254	192.168.3.66	20
127.0.0.0	255.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.3.0	255.255.255.0	255.255.255.0	192.168.3.66	192.168.3.66	20
192.168.3.66	255.255.255.255	255.255.255.255	127.0.0.1	127.0.0.1	20
192.168.3.255	255.255.255.255	255.255.255.255	192.168.3.66	192.168.3.66	20
224.0.0.0	240.0.0.0	240.0.0.0	192.168.3.66	192.168.3.66	20
255.255.255.255	255.255.255.255	255.255.255.255	192.168.3.66	192.168.3.66	1

Default Gateway: 192.168.3.254

Persistent Routes:

None



路由协议

- 路由选择和转发

- 路由选择（**Routing**）

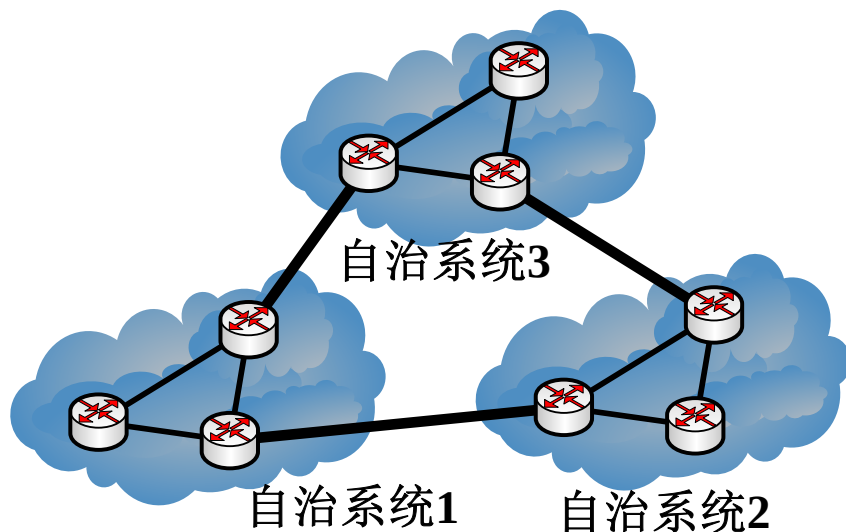
- 路由器之间通过路由协议建立路由表的过程

- 转发（**Fowarding**）

- 路由器或者交换机接收一个分组，根据分组的地址查询路由表，然后按表中指定的端口或者接口将分组转发出去

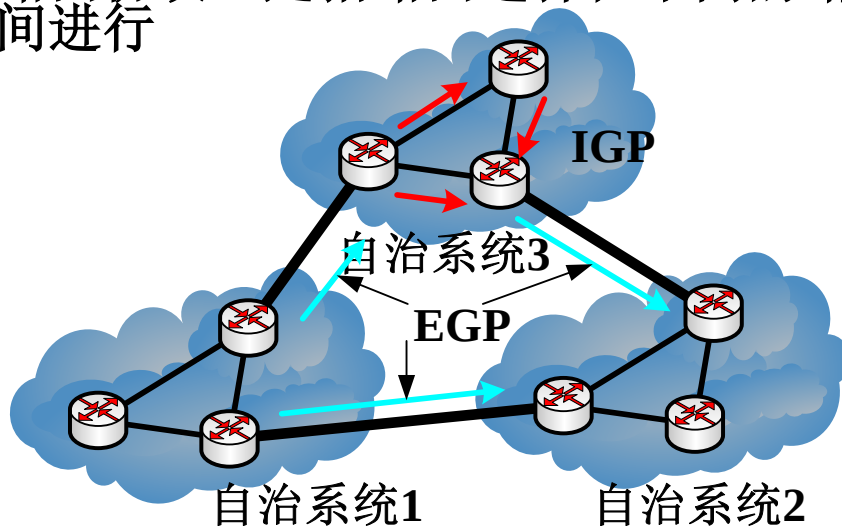
路由协议

- 自治系统**AS: Autonomous System**
 - 在**Internet**中，自治系统一般是在同一个管理实体控制之下并且具有相同路由选择策略的**IP**网络和路由器的集合
 - 在**Internet**中自治系统也称为路由选择域（**Domain**）



路由协议分类

- 内部网关路由协议**IGP** (**Interior Gateway Protocols**)
 - 也称为域内路由协议，是指路由选择的范围被限制在一个路由选择域内
- 外部网关路由协议**EGP** (**Exterior Gateway Protocols**)
 - 也称为域间路由协议，是指路由选择在不同的路由选择域中的路由器之间进行





内部网关路由协议

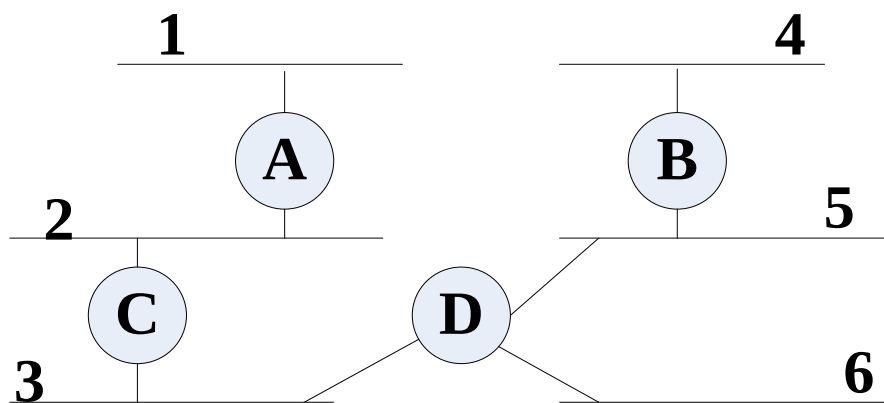
- 距离向量协议
 - 基于距离向量算法，是一种分布式算法，例如**Bellman-Ford**算法
- 链路状态协议
 - 基于链路状态算法，也称为最短路径算法，是一种全局算法。例如**Dijkstra**算法

- ❖ 在距离向量算法中，每个节点只和直接相连的节点进行通信，但是它把所有已知的全部信息（即到所有节点的距离）都告诉它们
- ❖ 在链路状态算法中，每个节点和其余各个节点通过扩散或者泛洪的方式都进行通信，但是它只告诉它们自己确切知道的信息（即与其直接相连的链路状态）
- ❖ 距离向量算法只适用于小规模网络，为了避免计数到无穷问题，网络直径一般不能超过**15**跳
- ❖ 与链路状态算法相比，距离向量算法对网络变化反应较慢

路由信息协议

RIP: Routing Information Protocol

- 距离向量算法的具体实现
- 路由器向邻居路由器通告到达所有网络的距离



路由器C通告路由器A它可以以距离0到达网络2和网络3，以距离1到达网络5和6，以距离2到达网络4

RIP分组格式

0 ----- 31

命令	版本	全零
地址类别		全零
网络 1 的 IP 地址		
全零		
全零		
到网络 1 的距离		
前 20 个字节的重复		

(a) Version 1

0 ----- 31

命令	版本	全零
地址类别		路由标记
网络 1 的 IP 地址		
子网掩码		
下一个站点的 IP 地址		
到网络 1 的距离		
前 20 个字节的重复		

20字节，
重复出现，
最多25次
重复

(b) Version 2

分组的主要部分采用
<网络地址、距离>对

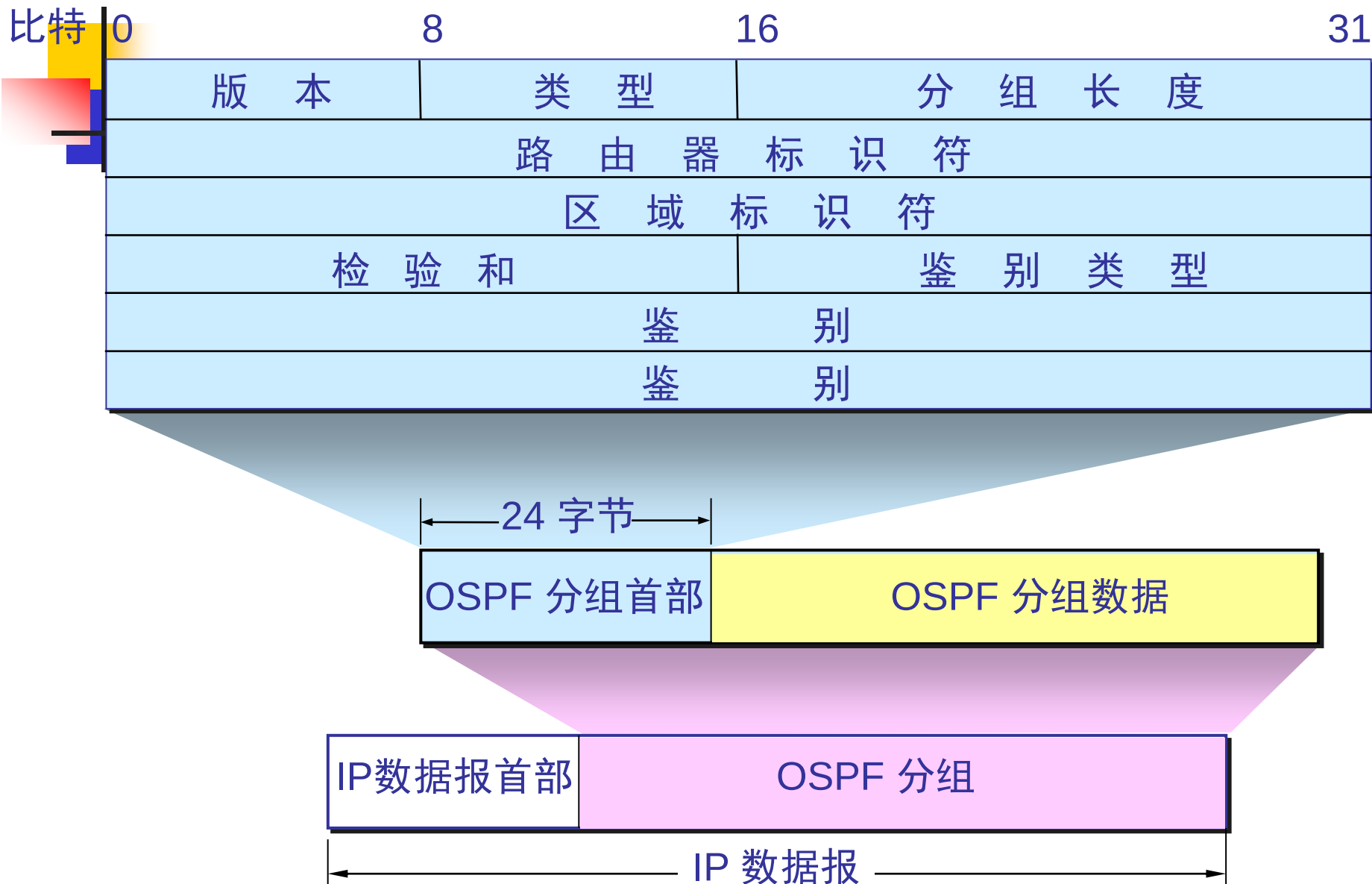


开放最短路径优先协议

OSPF: Open Shortest Path First

- 基于**链路状态算法**的具体实现，只有链路状态变化时才采用**flooding**的方式
- 在基本链路状态算法的基础上增加了特性
 - 路由选择消息的鉴别
 - 避免恶意的路由器通告恶意的路由选择消息，例如开销为**0**的路由选择消息
 - 直接采用**IP分组**（协议号为**89**），而不是**UDP**
 - 使得**OSPF**适合规模很大的网络，附加的层次性
 - 允许将路由选择域（**Domain**）划分为区（**Area**）
 - 负载均衡
 - 允许到同一位置的多条路由具有相同的开销

OSPF 分组

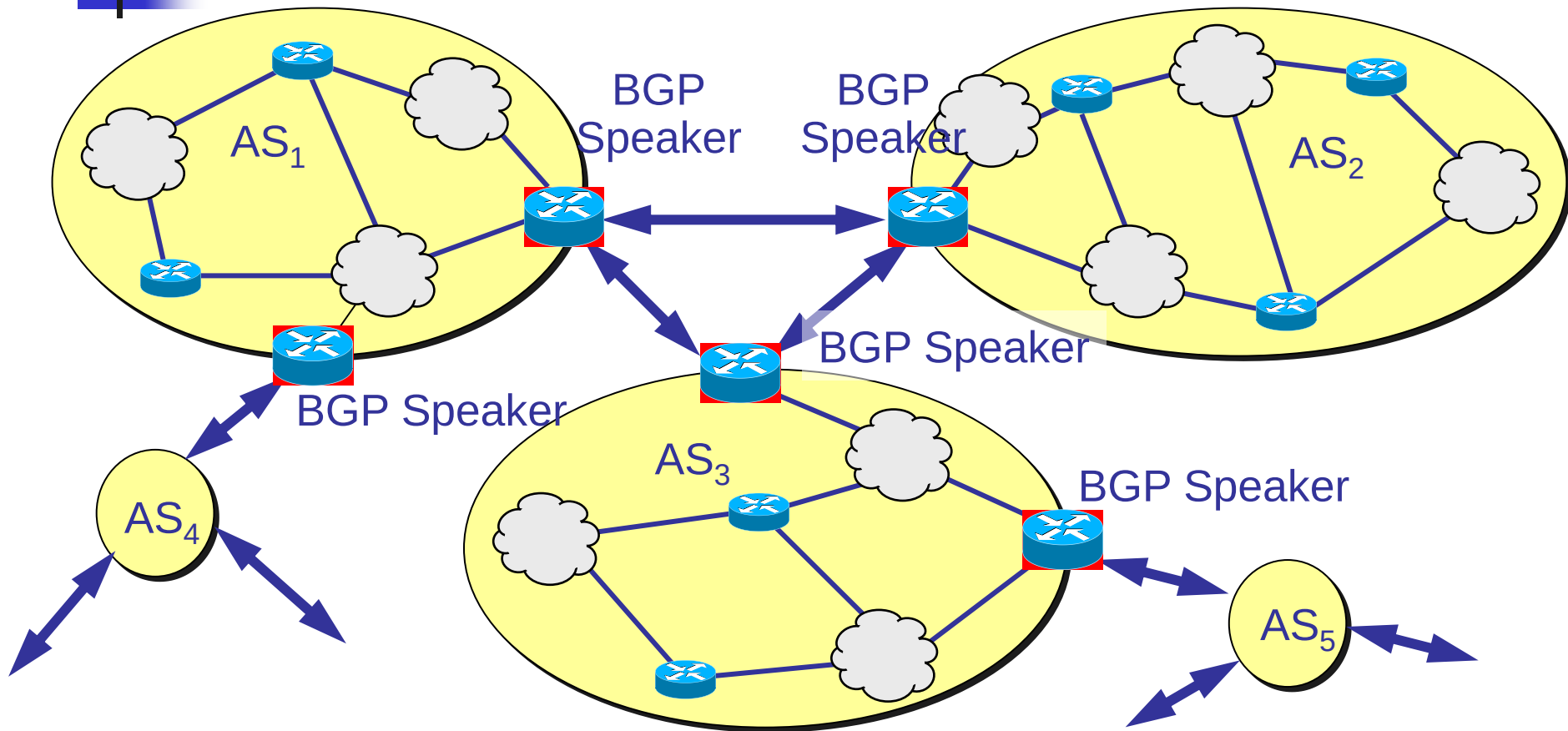




外部网关路由协议

- **BGP: Border gateway Protocol**, 边界网关路由协议
- **AS间为什么不采用RIP或者OSPF?**
 - **规模:** Internet规模庞大, 平坦式的路由协议不现实;
 - **策略:** 不同自治域之间还需要考虑其他因素, 例如政治因素;
 - **性能:** 不同自治系统的声明的路径代价不同, 同时由于策略的原因, 寻找最佳路由不现实。

外部网关路由协议

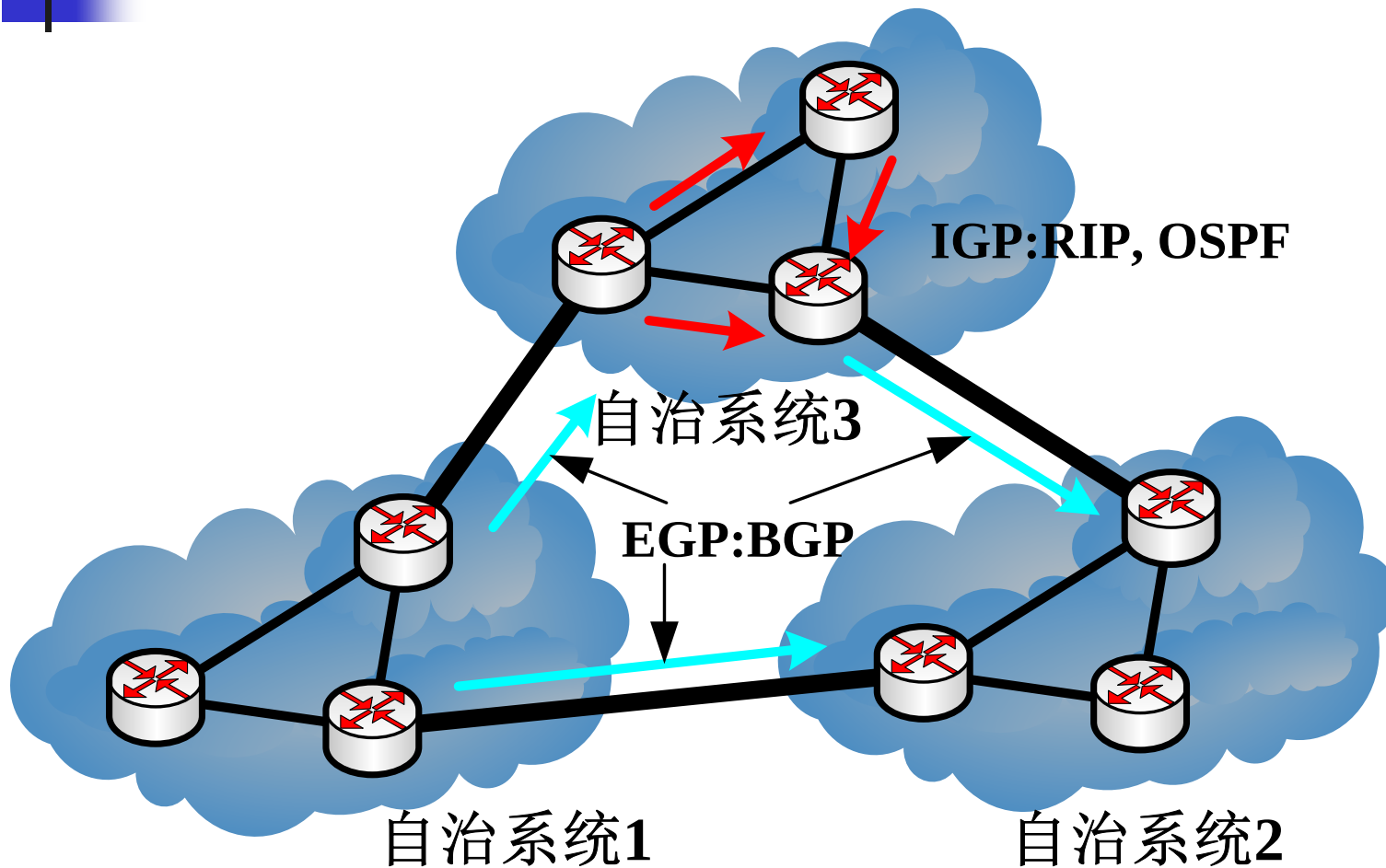




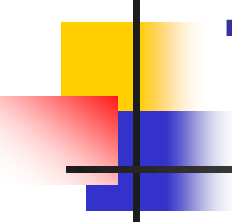
外部网关路由协议

- 基于**距离向量**的外部网关协议，增强了以下功能：
 - 使用**TCP**来实现更新消息的可靠性（端口**179**）
 - 触发更新消息
 - 周期性**keepalives**来确保**TCP**连接
 - 丰富的路由度量，被称为路径矢量或者属性
 - 能够扩展到大规模互联网络，例如**Internet**

路由协议作用范围



Chapter 7 Internet & TCP/IP

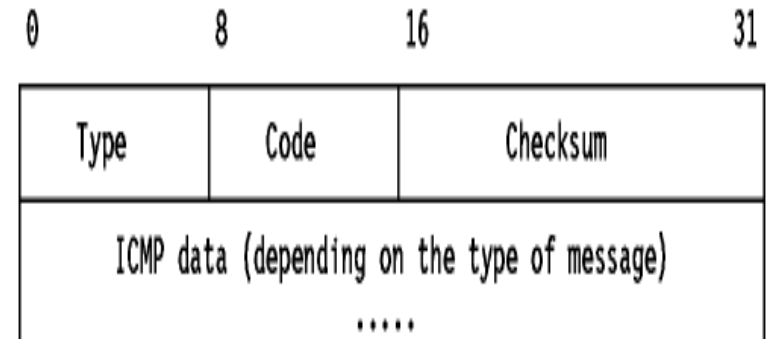
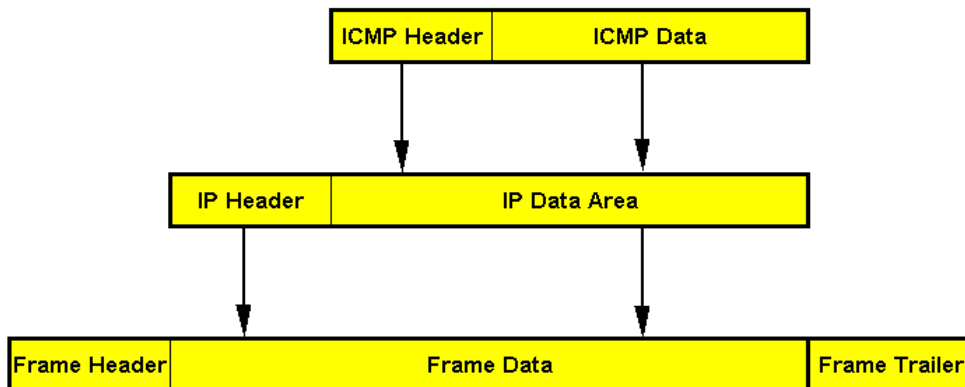


- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**

7.5 ICMP协议

(Internet Control Message Protocol)

- IP在网络层提供尽力服务 (best effort service)，当分组由于各种原因无法投递而遭丢弃时，就用**互连网控制报文协议** (ICMP) 发送**差错报告**。
- 尽管**ICMP也是网络层协议**，但它也需要经过IP协议封装。同样ICMP也不能保证可靠传输。ICMP协议定义在RFC 792 和RFC 950。





ICMP的主要功能

- **ICMP**定义了两类报文：**差错报文**和**信息报文**
- 差错报文
 - 源抑制（**Source quench**）：抑制发送过多分组的主机。
 - 超时（**Time exceeded**）：分组的**TTL为0**。
 - 信宿不可达（**Destination unreachable**）：报告子网、主机不能定位的信宿。
 - 重定向（**Redirect**）：路由重定向
 - 参数问题（**Parameter problem**）：分组头参数出错。
- 信息报文
 - 回音请求/响应（**Echo request/reply**）
 - 路由器发现（**Router discovery**）
 - 地址掩码请求/响应（**Address mask request/reply**）

ICMP Message Types

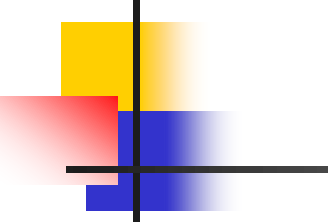
Type	Code	Description	Query	Error	Type	Code	Description	Query	Error
0	0	Echo reply	•		5		Redirect		
3		Destination unreachable:				0	Redirect for network		•
	0	Network unreachable		•		1	Redirect for host		•
	1	Host unreachable		•		2	Redirect for TOS and Net		•
	2	Protocol unreachable		•		3	Redirect for TOS and Host		•
	3	Port unreachable		•	8	0	Echo request	•	
	4	Fragmentation needed		•	9	0	Router advertisement	•	
	5	Source route failed		•	10	0	Router solicitation	•	
	6	Destination network unknown		•	11		Time exceeded		
	7	Destination host unknown		•		0	TTL equals 0 during transit		•
	8	Source host isolated		•		1	TTL equals 0 during reassembly		•
	9	Destination net prohibited		•	12		Parameter problem		
	10	Destination host prohibited		•		0	IP header bad		•
	11	Network unreachable for TOS		•		1	Required option missing		•
	12	Host unreachable for TOS		•	13	0	Timestamp request	•	
	13	Communication prohibited		•	14	0	Timestamp reply	•	
	14	Host precedence violation		•	15	0	Information request	•	
	15	Precedence cutoff in effect		•	16	0	Information reply	•	
4	0	Source quench		•	17	0	Address mask request	•	
					18	0	Address mask reply	•	



常用网络诊断工具

■ ping

- 功能：测试主机的可达性和往返延迟等网络信息
- **ping**程序采用**ICMP**的回音请求/响应（**Echo request/reply**）报文，通过向目的主机发送回音请求，返回响应报文，来测试目的主机的可达性、往返延迟以及丢包率



```
C:\Documents and Settings\THINKPAD USER>ping 202.38.64.1
```

```
Pinging 202.38.64.1 with 32 bytes of data:
```

```
Reply from 202.38.64.1: bytes=32 time=4ms TTL=56
```

```
Reply from 202.38.64.1: bytes=32 time=5ms TTL=56
```

```
Reply from 202.38.64.1: bytes=32 time=3ms TTL=56
```

```
Reply from 202.38.64.1: bytes=32 time=3ms TTL=56
```

```
Ping statistics for 202.38.64.1:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 3ms, Maximum = 5ms, Average = 3ms
```

```
C:\Documents and Settings\THINKPAD USER>ping www.sohu.com
```

```
Pinging pgctcsht01.a.sohu.com [222.73.123.6] with 32 bytes of data:
```

```
Reply from 222.73.123.6: bytes=32 time=13ms TTL=52
```

```
Reply from 222.73.123.6: bytes=32 time=14ms TTL=52
```

```
Reply from 222.73.123.6: bytes=32 time=13ms TTL=52
```

```
Reply from 222.73.123.6: bytes=32 time=12ms TTL=52
```

```
Ping statistics for 222.73.123.6:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 12ms, Maximum = 14ms, Average = 13ms
```

```
C:\Documents and Settings\THINKPAD USER>ping 202.38.75.44
```

```
Pinging 202.38.75.44 with 32 bytes of data:
```

```
Request timed out.
```

```
Request timed out.
```

```
Request timed out.
```

```
Request timed out.
```

```
Ping statistics for 202.38.75.44:
```

```
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```



常用网络诊断工具

■ traceroute

- 功能：发现从源主机到目的主机路径上的网络节点。
- **Traceroute**首先发送**TTL=1**的**echo request**,第一跳节点向源端返回**ICMP**的**Time Exceeded**错误,然后发送**TTL=2**的**echo request**,第二条节点返回**Time Exceeded**错误,不断递增**TTL**,直到到达目的主机


```
C:\Documents and Settings\kpxue>tracert -d 202.38.64.1
```

```
Tracing route to 202.38.64.1 over a maximum of 30 hops
```

1	<1 ms	<1 ms	<1 ms	192.168.3.254
2	<1 ms	<1 ms	<1 ms	202.38.75.254
3	<1 ms	<1 ms	<1 ms	202.38.96.181
4	<1 ms	<1 ms	<1 ms	202.38.64.1

```
Trace complete.
```

```
C:\Documents and Settings\THINKPAD USER>tracert -d 202.38.64.1
```

```
Tracing route to 202.38.64.1 over a maximum of 30 hops
```

1	2 ms	2 ms	2 ms	172.16.23.96
2	4 ms	3 ms	4 ms	61.191.23.17
3	5 ms	3 ms	3 ms	61.190.228.43
4	2 ms	3 ms	3 ms	61.190.194.89
5	3 ms	3 ms	5 ms	218.22.20.10
6	6 ms	6 ms	5 ms	202.141.161.252
7	6 ms	21 ms	5 ms	210.45.224.65
8	4 ms	4 ms	3 ms	202.38.64.1

```
Trace complete.
```

```
C:\Documents and Settings\kpxue>ping -r 9 -n 1 202.38.64.1
```

Pinging 202.38.64.1 with 32 bytes of data:

Reply from 202.38.64.1: bytes=32 time=1ms TTL=61

Route: 202.38.75.11 ->

202.38.96.175 ->

202.38.64.126 ->

202.38.64.1 ->

202.38.64.1 ->

202.38.96.181 ->

202.38.75.254 ->

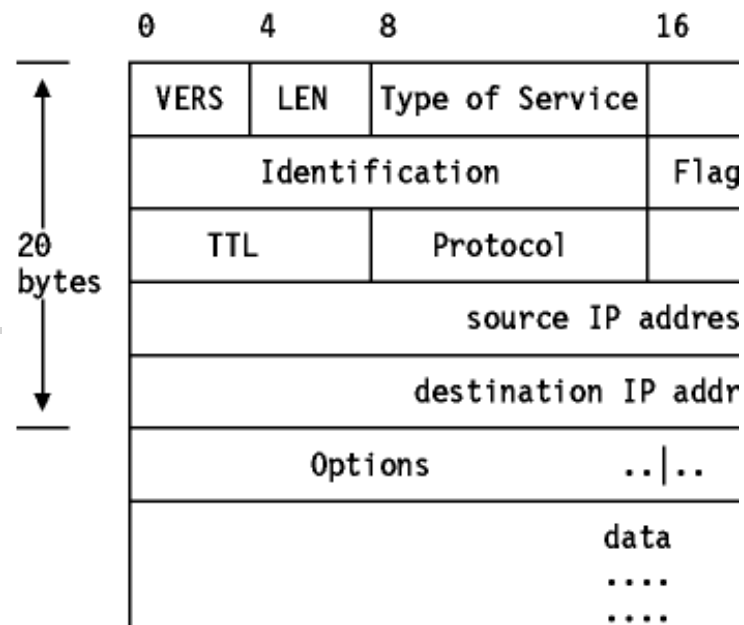
192.168.3.254

Ping statistics for 202.38.64.1:

Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),

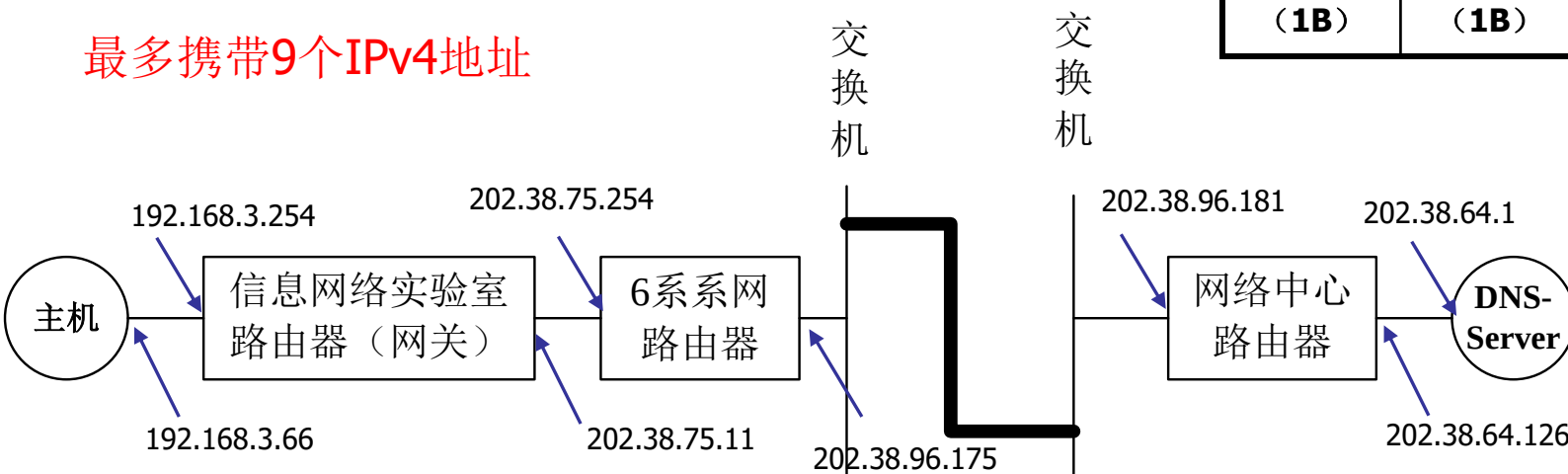
Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 1ms, Average = 1ms



路由记录选项（Route recording）

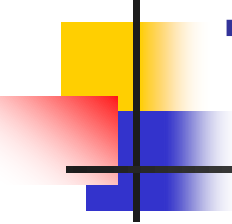
最多携带9个IPv4地址



TLV格式

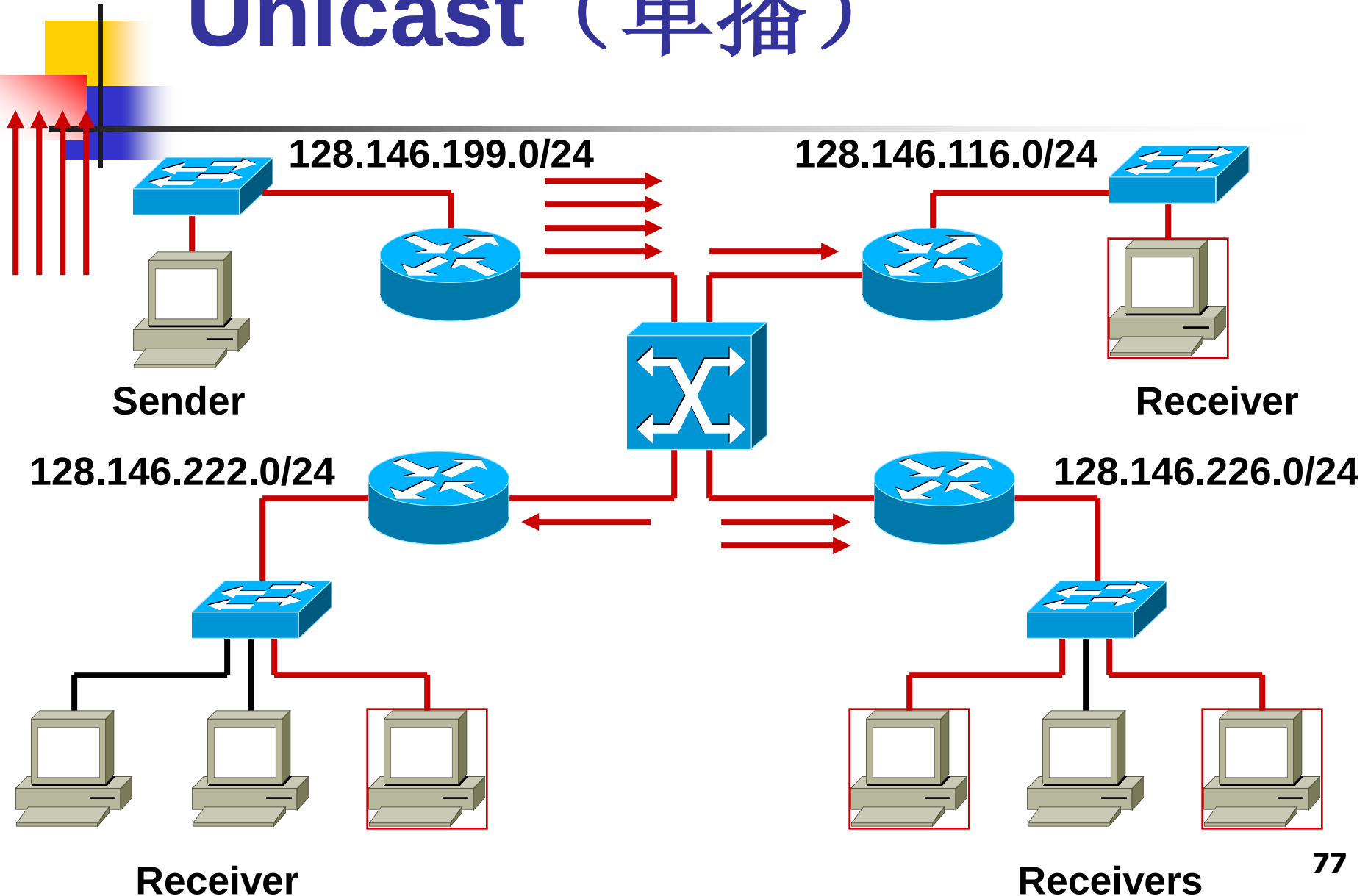
TYPE (1B)	LENTH (1B)	VALUE (nB)
-----------	------------	------------

Chapter 7 Internet & TCP/IP

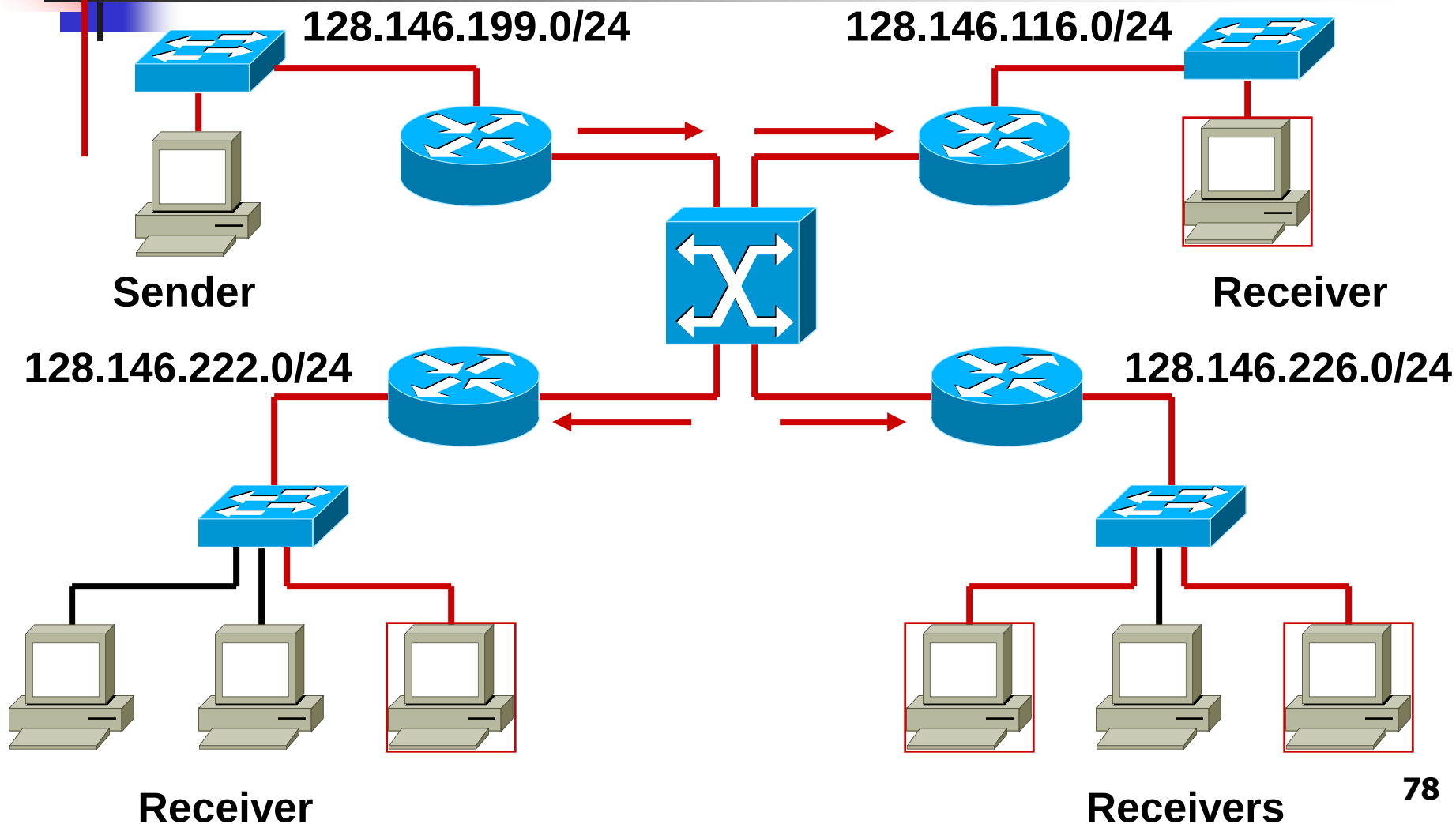


- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**

Unicast (单播)



Multicast (组播)



组播概念

单播 (unicast): 1对1
广播 (broadcast): 1对多

■ 组播 (multicast) : 1对多, 多对多

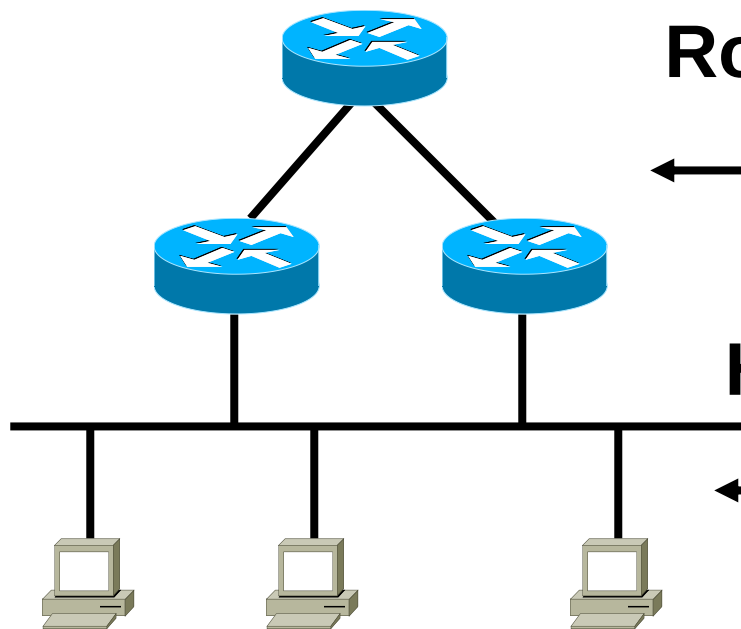
- 也称多播, 可用于会议电视, 分布式计算, 视频转播, 网络游戏, 讨论组
- IP采用D类地址作为组播地址, 每个D类地址代表一组主机, 共有**28**位可用来标识小组。因此可同时支持多达**25**亿个小组。
- Internet支持两类组地址: 永久组地址和临时组地址
 - 224.0.0.1——LAN上所有系统
 - 224.0.0.2——LAN上所有路由器
 - 224.0.0.5——LAN上所有OSPF路由器

} 永久组地址



(224-239)

组播相关协议



Router to Router

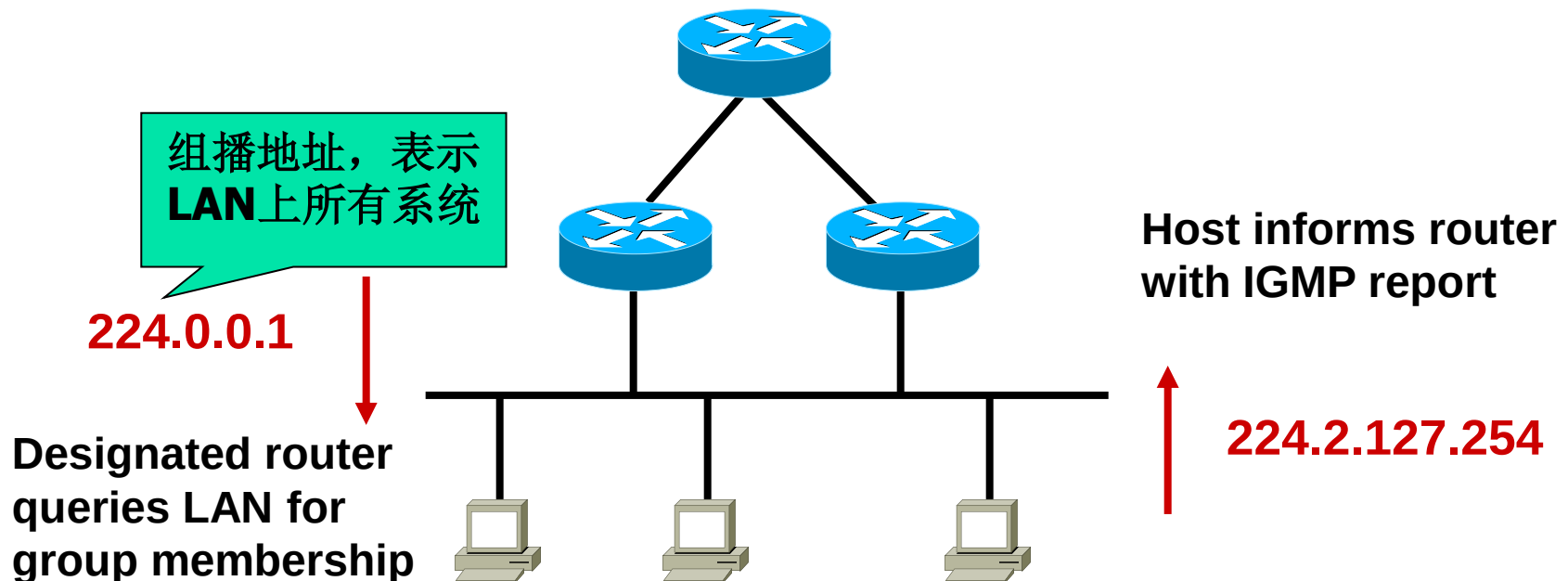
← 组播路由协议，如DVMRP, PIM
功能：生成组播路由表

Host to Router

← 组播管理协议，IGMP
功能：组成员的加入和退出

IGMP

- 组成员管理协议**IGMP**: Internet Group Management Protocol

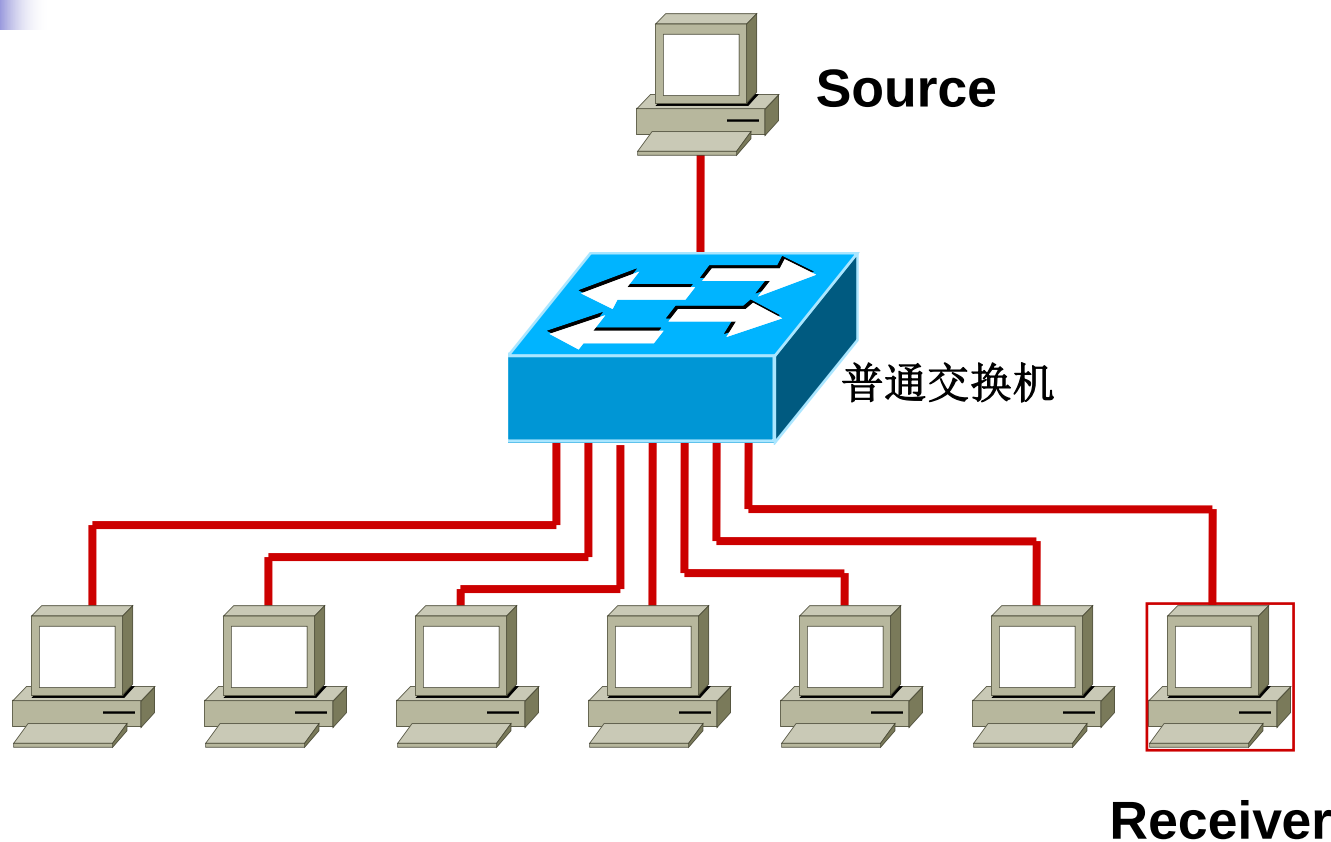




组播路由协议

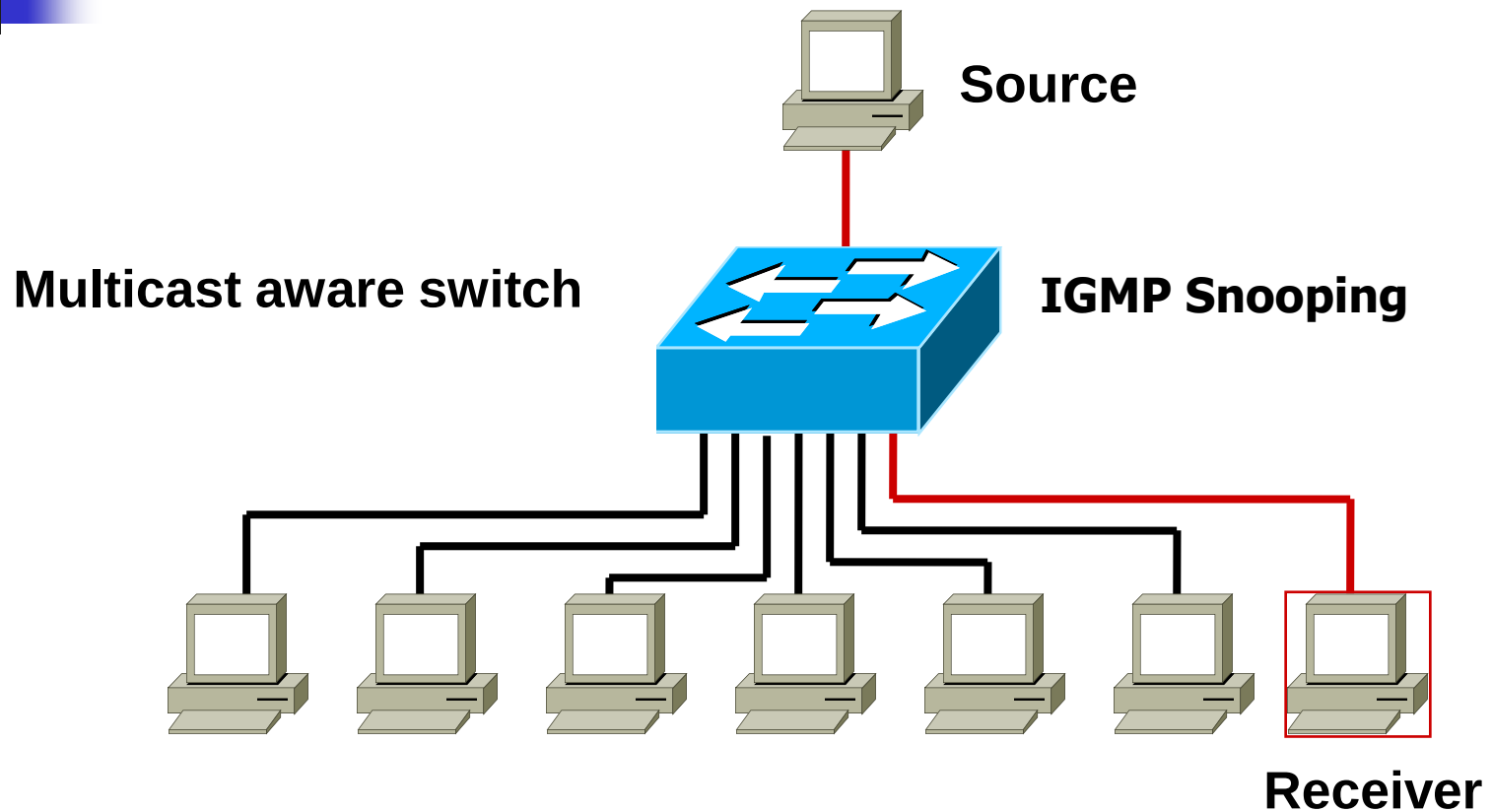
- 基于源的组播协议，以发送方为多播树的根，并且包含了所有的组成员
 - **DVMRP (Distance Vector Multicast Routing Protocol)**
 - **MOSPF (Multicast Extension to OSPF)**
- 基于共享树的协议，对于每个组使用同一颗树
 - **CBT (Core Based Trees)**
- 混合以上两种方法的协议
 - **PIM (Protocol Independent Multicast)**
 - **PIM-DM (Dense Mode), PIM-SM (Sparse Mode)**

IGMP Snooping



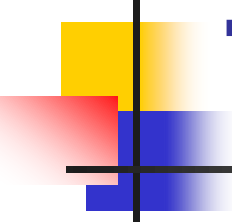
交换机的所有端口都转发组播分组，包括那些没有连接组成员的端口

IGMP Snooping



交换机只向连接有组成员的端口转发分组

Chapter 7 Internet & TCP/IP



- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**



IPv4寻址问题

- 地址空间出现不足
 - 32位的IPv4地址总共有 2^{32} (4,294,967,296) IP地址，但根据地址的分类方式，实际可用的地址要少得多。
 - 地址分配的不合理，A类地址有128个，每个A类可容纳1600万台主机，而C类有200万个，每个由256个地址。
- 非层次化的地址分类方式，使得骨干网路由表急剧膨胀
 - 路由器会对于每一个网络都需要在路由表中增加一条表项



解决思路

- 采用层次地址结构
 - 提高**IP**地址的利用率
 - 对**IP**地址进行汇聚，减少路由协议中携带的和路由器的路由表中储存的网络号的数量

1. 划分子网

2. 无类别域间寻路**CIDR**

- 多台主机共享一个全局的**IP**

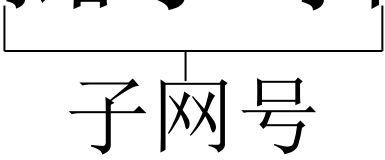
3. 网络地址转换**NAT**

最终的解决方案是采用**IPv6**



划分子网

- 大的**A/B/C**类网络划分为多个小的子网（**subnet**），作为一个整体对外仍然表现为一个单独的网络，只有一个网络号
- **IP地址=网络号+子网ID+主机号**



子网号
- 子网掩码（**subnet mask**）配置于每个子网中的所有的主机



划分子网

■ 子网掩码：Subnet Mask

- **32bits**的位模式，表示方法与**IP**地址类似
- 位模式中的**1**对应于**IP**地址的子网号，**0**对应于主机号

例如：11111111 11111111 11111111 00000000 255.255.255.0

11111111 11111111 11111111 11000000 255.255.255.192

某主机的**IP**地址为**140.128.34.79**，如果其子网掩码为**255.255.255.0**，则其子网号为**140.128.34.0**，主机号为**79**；如果子网掩码为**255.255.255.192**，则其子网号为**140.128.34.64**，主机号为**15**

另外也可以采用地址/掩码长度表示，例如：**140.128.34.79 /26**



划分子网

■ 步骤

1) 确定子网号的长度或者主机号的长度

- 子网数量决定子网**ID**的长度，从而决定了子网号的长度
 - 子网数量 $<2^n$ ，其中**n**为子网**ID**的长度
- 主机数量决定了主机号的长度
 - 主机数量 $<2^n-2$ ，其中**n**为主机号的长度

2) 根据子网号（网络号+子网**ID**）的长度确定子网掩码，然后根据子网掩码确定子网号

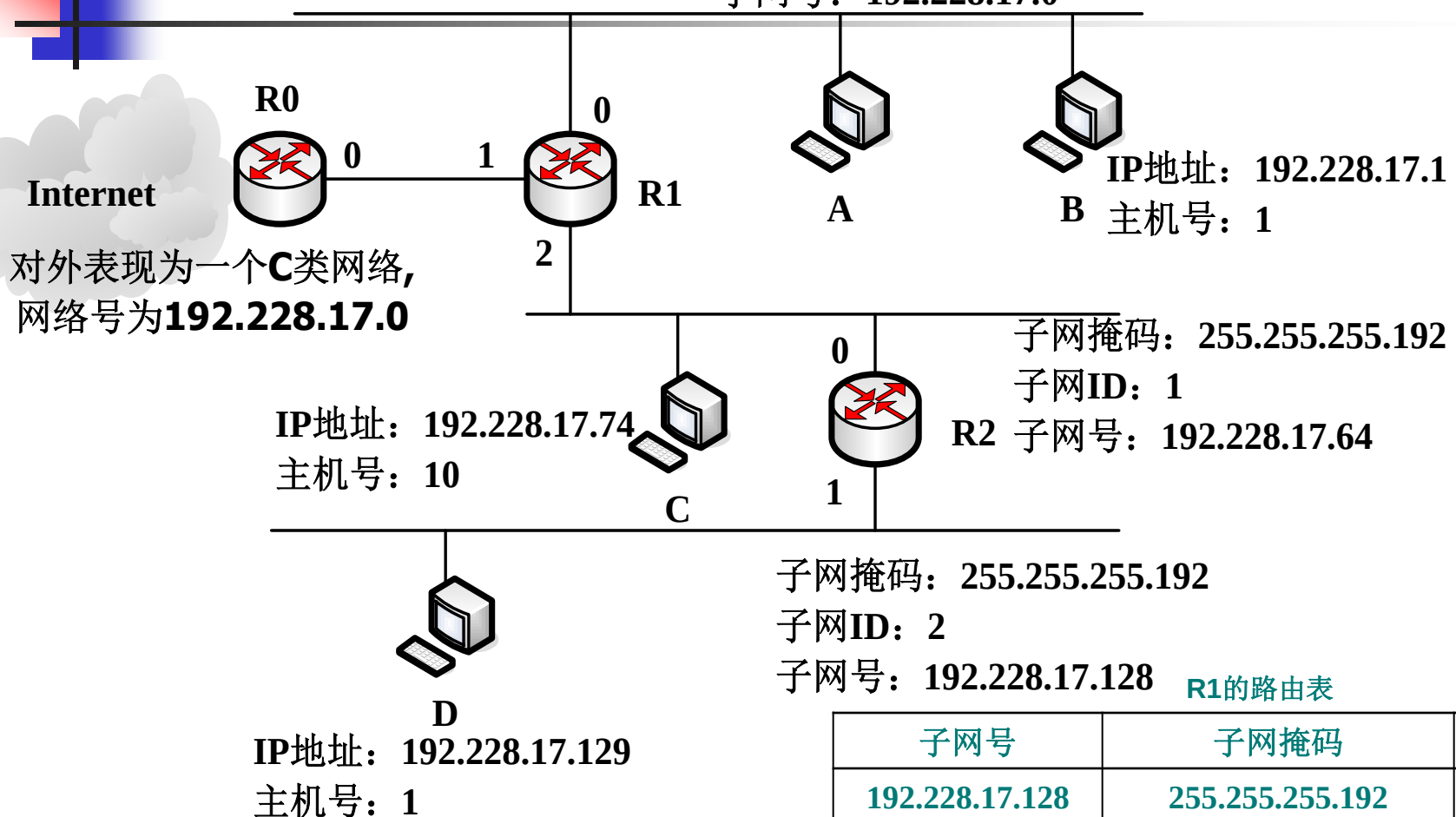
3) 根据子网掩码，子网号等信息配置每个路由器上的转发表

- 把一个**C类网络（192.228.17.0）**划分为三个子网
 - 根据子网数量 $<2^n$ ，选择 **$n=2$** ，从而得到子网号长度为**26**比特，子网掩码为**255.255.255.192**（**192=1100 0000**）

子网掩码：255.255.255.192

子网ID：0

子网号：192.228.17.0



R1的路由表

子网号	子网掩码	下一跳
192.228.17.128	255.255.255.192	R2
192.228.17.64	255.255.255.192	2
192.228.17.0	255.255.255.192	0

- 一个公司有一个C类网络**200.1.1.0**，并且希望建立**4**个部门的子网，其中部门**A**有**72**台主机，**B**有**35**台主机，**C**有**20**台主机，**D**有**18**台主机，共有**145**台主机

确定主机号的长度：

对于部门A，有 $2^n - 2 \geq 72$ 得到 $n \geq 7$ ，取 $n=7$ ，子网ID长度为1

对于部门B，有 $2^n - 2 \geq 35$ ，得到 $n \geq 6$ ，取 $n=6$ ，子网ID长度为2

对于部门C，有 $2^n - 2 \geq 20$ ，得到 $n \geq 5$ ，取 $n=5$ ，子网ID长度为3

对于部门D，有 $2^n - 2 \geq 20$ ，得到 $n \geq 5$ ，取 $n=5$ ，子网ID长度为3

确定子网掩码和子网号：

部门A：子网掩码**255.255.255.128**，子网号**200.1.1.0**(子网ID=0)

部门B：子网掩码**255.255.255.192**，子网号**200.1.1.128**(子网ID=10)

部门C：子网掩码**255.255.255.224**，子网号**200.1.1.192**(子网ID=110)

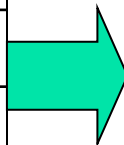
部门D：子网掩码**255.255.255.224**，子网号**200.1.1.224**(子网ID=111)

划分子网

■ 分组转发实例

路由表

子网号	子网掩码	下一跳
128.96.39.0	255.255.255.128	接口0
128.96.39.128	255.255.255.128	接口1
128.96.40.0	255.255.255.128	R2
192.4.153.0	255.255.255.192	R3
缺省		R4



目的地址	下一跳
128.96.39.10	接口0
128.96.40.12	R2
128.96.40.151	R4
192.4.153.17	R3
192.4.153.90	R4

主要操作：通过子网掩码得IP地址所对应的子网号，然后查找路由表



无类别域间寻路

CIDR: Classless Inter-Domain Routing

- **RFC1518, RFC1519**
- 采用可变长度的网络前缀(**network prefix**)来取代地址分类中网络号长度固定的做法
- 具有相同前缀的连续的**IP**地址组成**CIDR Block**, 表示为**A.B.C.D/N**, 其中**N**为前缀长度
 - 例如**202.38.208.0/20**
- **VLSM: Variable-Length Subnet Masking**, 等同于**CIDR**概念, 置**1**的比特数量等于前缀长度



无类别域间寻路

CIDR: Classless Inter-Domain Routing

■ **CIDR Block**分配

IANA (Internet Assigned Numbers Authority) -->

RIRs (Regional Internet Registries) -->

ISPs (Internet Service Providers) -->

Subscribers

无类别域间寻路

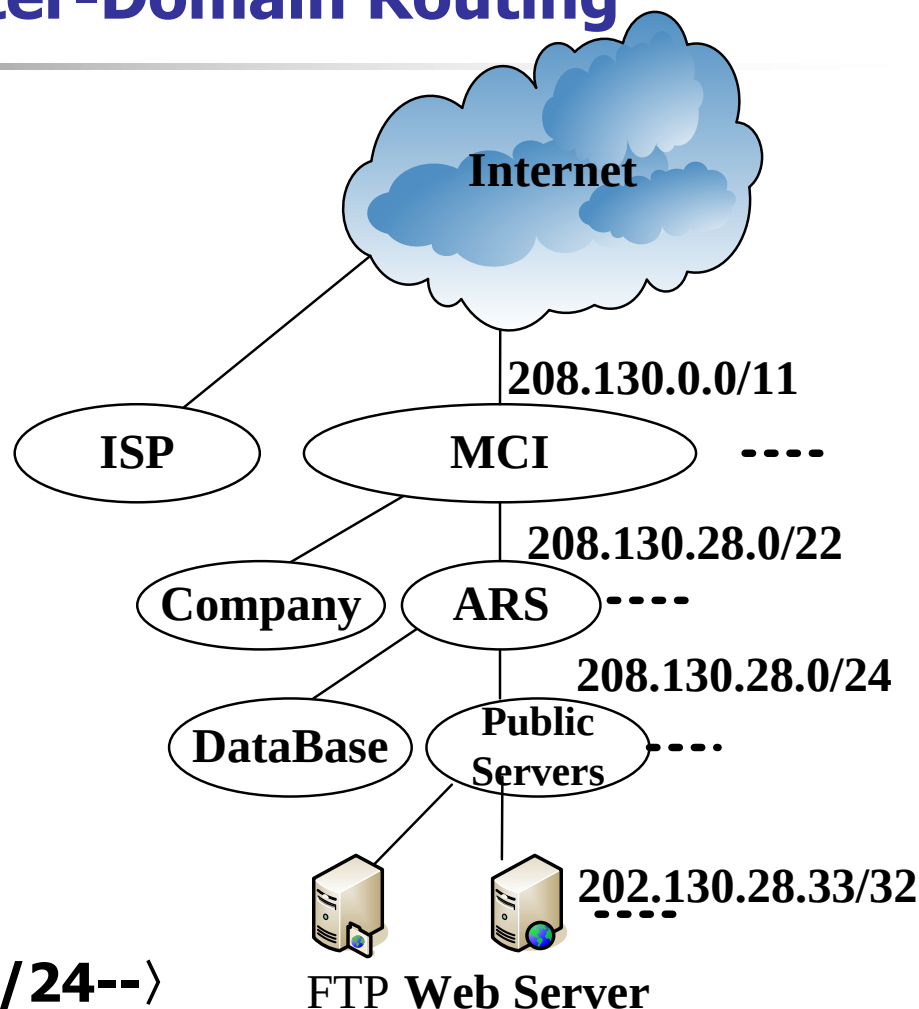
CIDR: Classless Inter-Domain Routing

- 前缀汇聚: **Prefix Aggregation**

- 具有相同前缀的多个连续的**CIDR Block**可以汇聚成一个更短前缀路由

- 16个连续的/24的**CIDR Block**可以汇聚成一条/20的路由

208.130.16.0/24~208.130.31.0/24-->
208.120.16.0/20



无类别域间寻路

CIDR: Classless Inter-Domain Routing

- 分组转发
 - 路由表结构
 - <网络前缀/前缀长度、下一跳>
 - 在**CIDR**中，如果路由器上的路由表中有多条表项满足要求，则采用**前缀最长匹配规则**

目的网络	下一跳
128.96.0.0/16	R0
128.96.192.0/18	R1
128.96.128.0/18	R2

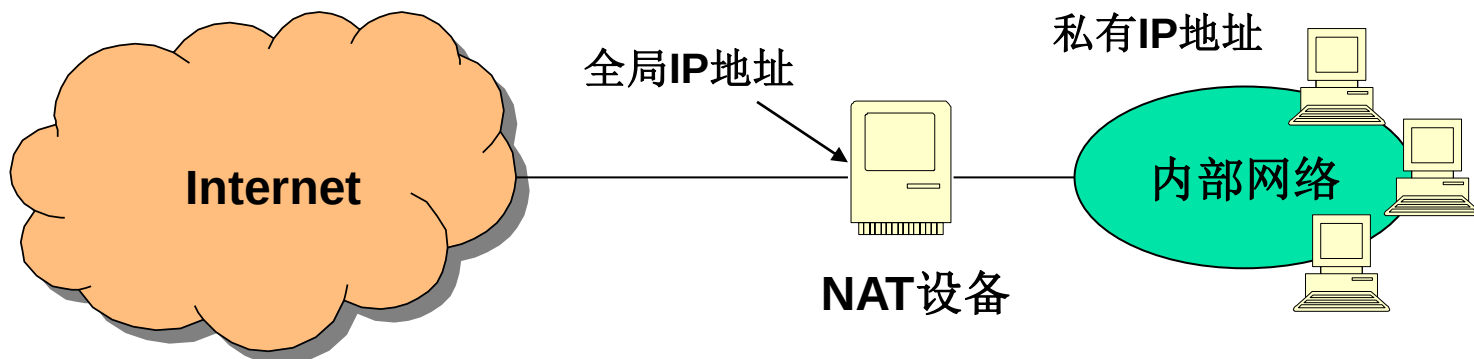
对于目的地址为**128.96.195.70**的分组，匹配的路由表表项包括两项，但是使用最长匹配规则选择下一跳为**R1**

网络地址转换

NAT: Network Address Translation

- 根据作用范围的不同，分两种**IP**地址
 - 全局**IP**地址：用于**Internet**上的分组转发，要求在**Internet**范围内唯一，
 - 私有**IP**地址：用于指定网络内的分组转发，只要求在指定网内部唯一

NAT: 实现网络内的多台主机共享一个全局的**IP**地址





网络地址转换

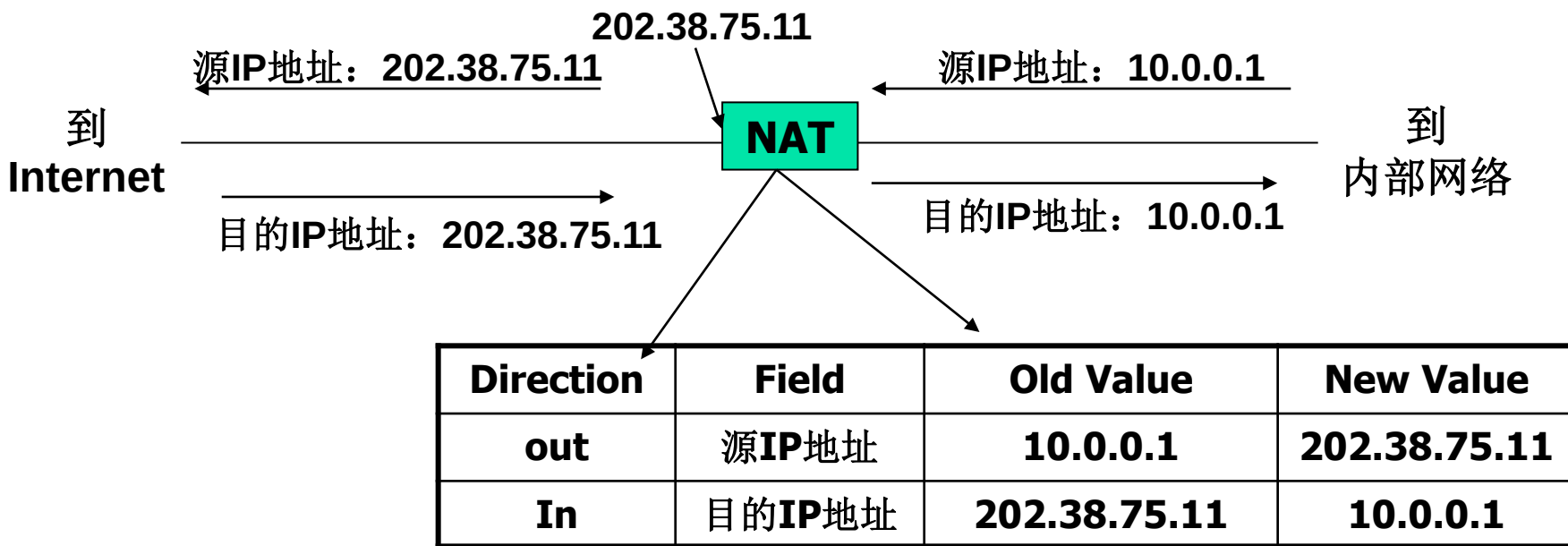
NAT: Network Address Translation

- 常用的私有**IP**地址
- **10.0.0.0 - 10.255.255.255**
A single Class A network
- **172.16 .0.0- 172.31 .255.255**
16 contiguous Class B networks
- **192.168.0.0 - 192.168.255 .255**
256 contiguous Class C networks

网络地址转换

NAT: Network Address Translation

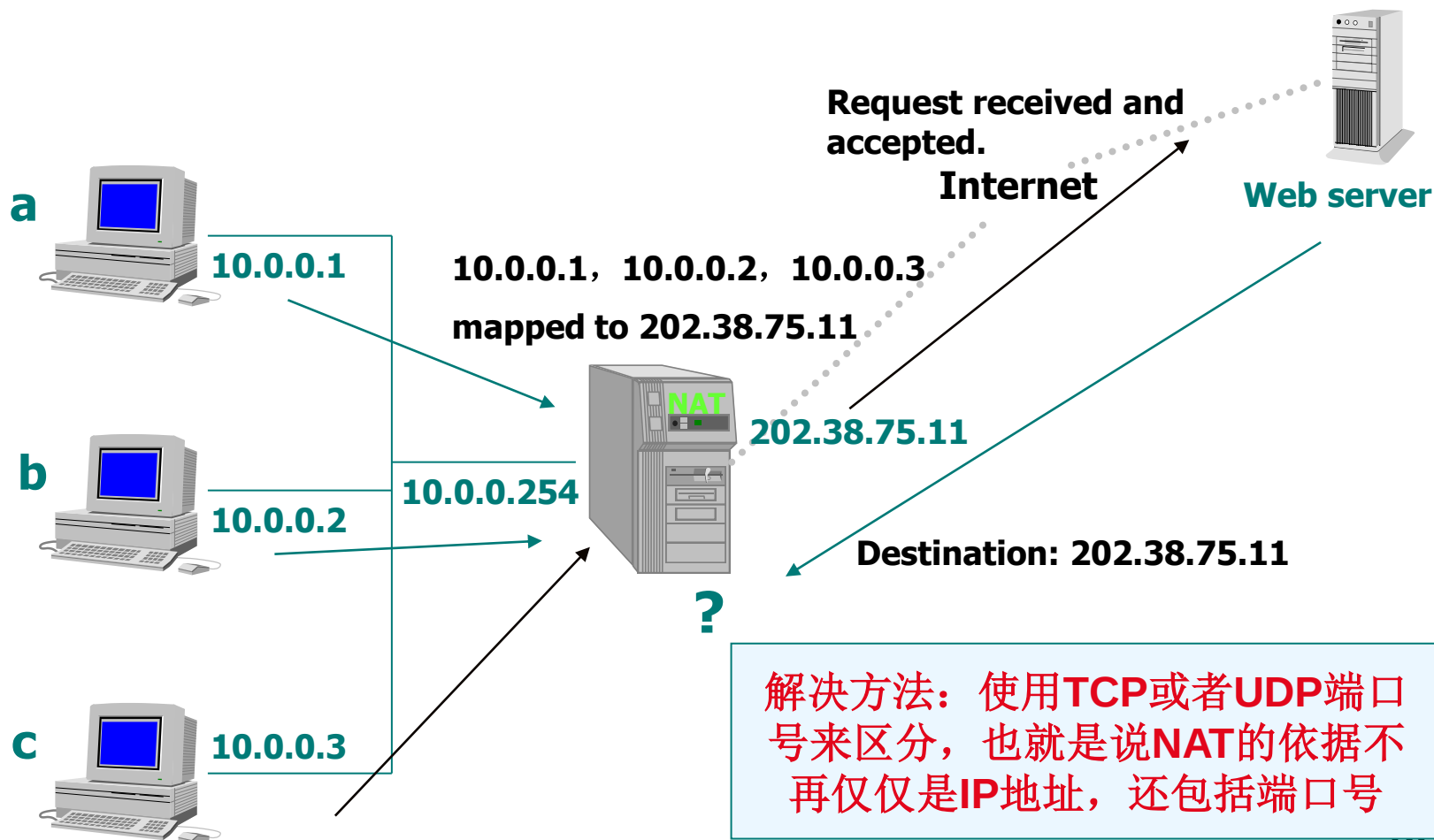
■ 基本的地址转换



转换表(自动生成)

NAT操作会改变分组的IP头标，因此要重新计算检验和

问题：基本的地址转换只使用**IP**地址信息，在只有一个全局**IP**地址的情况下，对于从**Internet**到内部网络上的分组，**NAT**设备无法知道分组所对应的内部主机，或者同一台内部主机上的应用



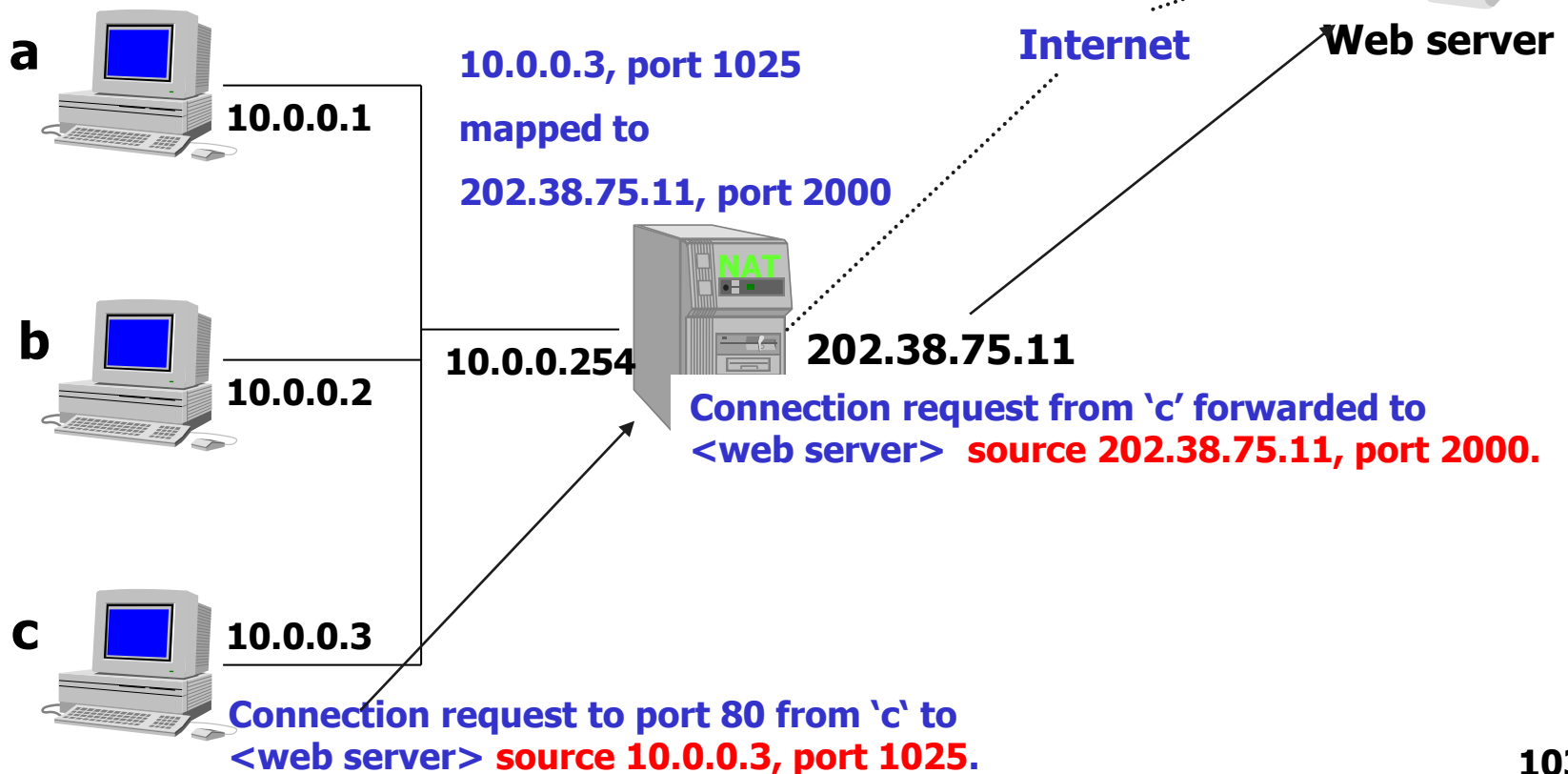


网络地址转换

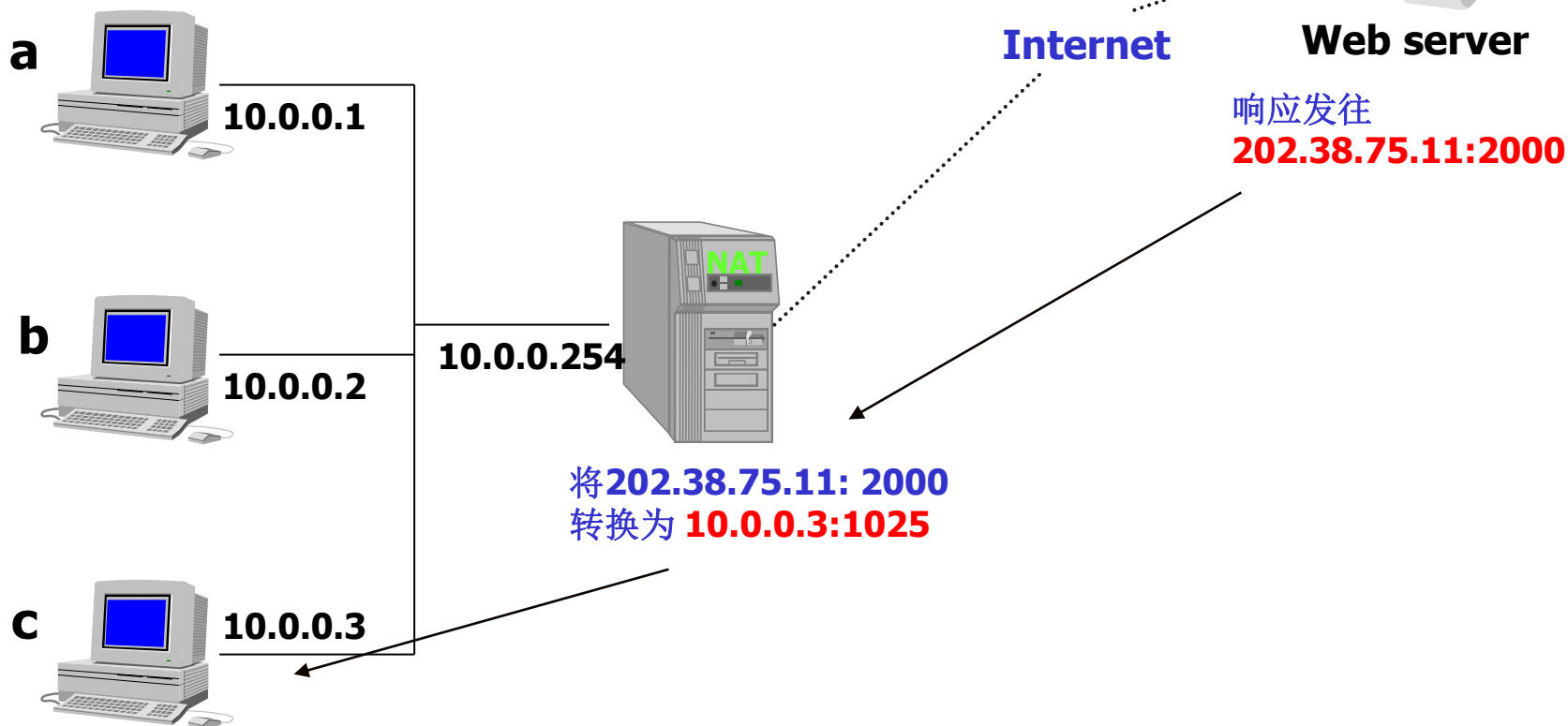
NAT: Network Address Translation

- 网络地址和端口转换**NAPT: Network Address and Port Translation**
 - 最常用的一种**NAT**方式
 - 使用**IP**地址和**TCP/UDP**端口号
 - 端口号标识了同一台主机上不同的应用，例如**FTP**服务器进程一般运行在**21**号端口，**web**服务进程一般运行在**80**号端口，**TCP/UDP**根据端口号将数据投递给相应的应用进程
- **NAPT**操作不仅仅要修改分组的**IP**头标，还要修改**TCP/UDP**头标中的端口号
 - 在**NAT**设备上，虽然内部网络所有的主机共享同一个全局**IP**地址，但是它们所使用的端口号不同

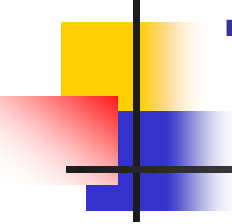
direction	Fields	Old Value	New Value
out	源IP地址: 源端口号	10.0.0.3:1025	202.38.75.11:2000
out	源IP地址: 源端口号	10.0.0.2:1023	202.38.75.11:1011
in	目的IP地址: 目的端口号	202.38.75.11:2000	10.0.0.3:1025
In	目的IP地址: 目的端口号	202.38.75.11:1011	10.0.0.2:1023



direction	Fields	Old Value	New Value
out	源IP地址: 源端口号	10.0.0.3:1025	202.38.75.11:2000
out	源IP地址: 源端口号	10.0.0.2:1023	202.38.75.11:1011
in	目的IP地址: 目的端口号	202.38.75.11:2000	10.0.0.3:1025
In	目的IP地址: 目的端口号	202.38.75.11:1011	10.0.0.2:1023

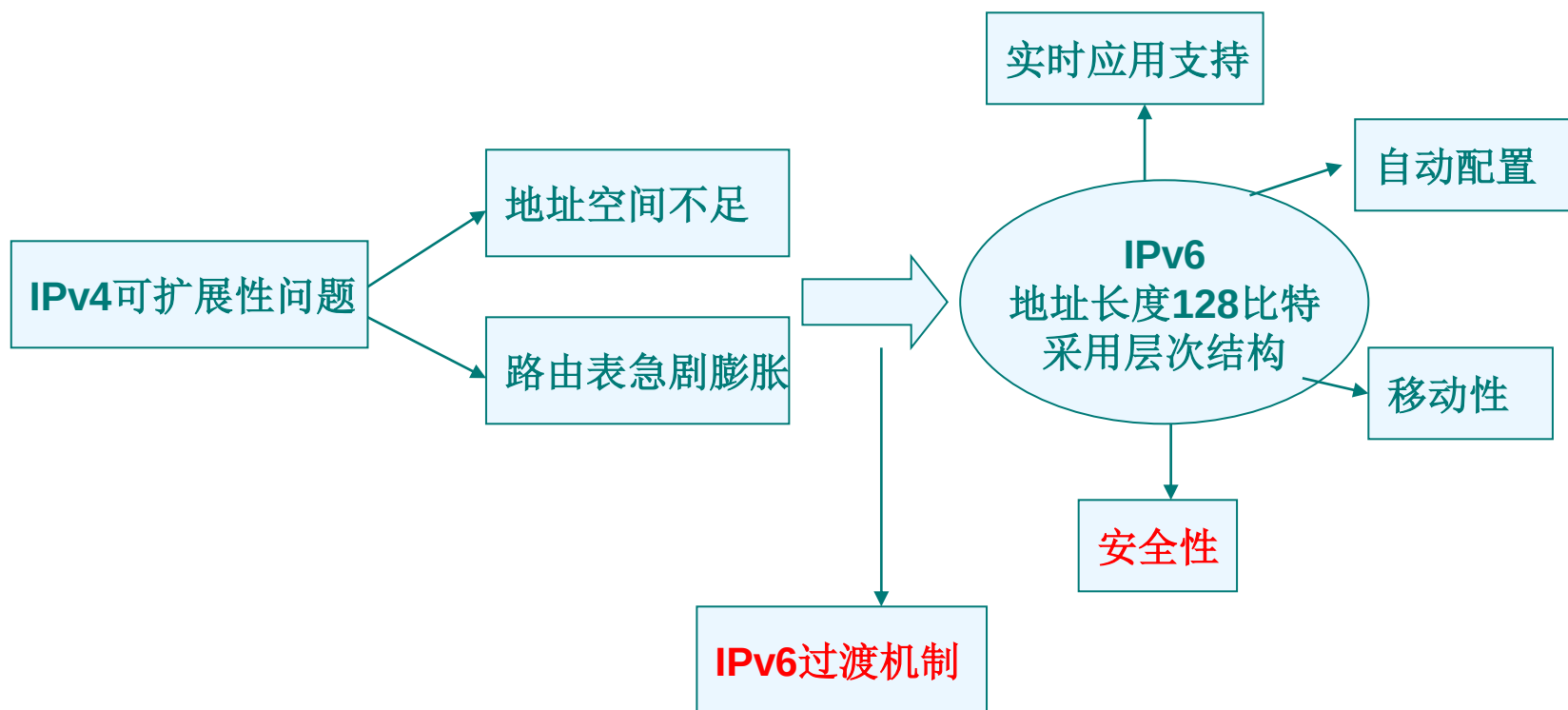


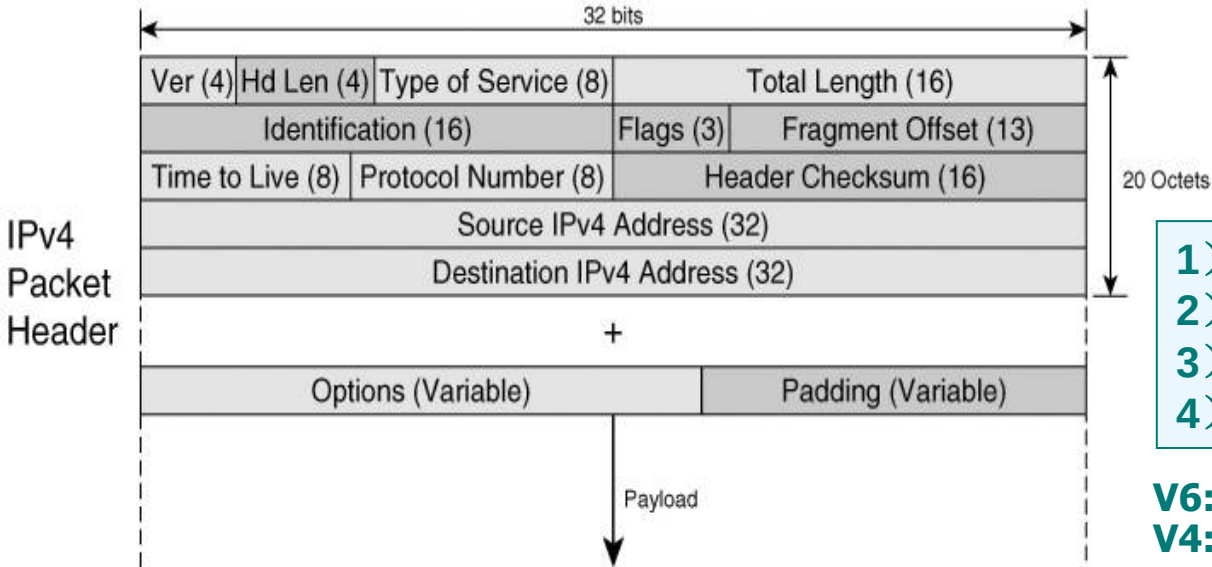
Chapter 7 Internet & TCP/IP



- **7.1 Internet基本协议栈**
- **7.2 IP地址**
- **7.3 地址解析协议**
- **7.4 IP协议**
- **7.5 IP路由和转发**
- **7.6 ICMP协议**
- **7.7 IP组播**
- **7.8 IPv4的可扩展性**
- **7.9 IPv6协议**

IPv6的引入





- 1) 扩展地址空间，128位地址长度
- 2) 简化了头标
- 3) 改善了选项功能，提供了安全功能
- 4) 修订了参数，增加了流标记域

V6: 6 fields + 2 addr

V4: 10 fields + 2 addr + options

Deleted:

Header length
type of service
identification, flags, fragment offset
Header Checksum

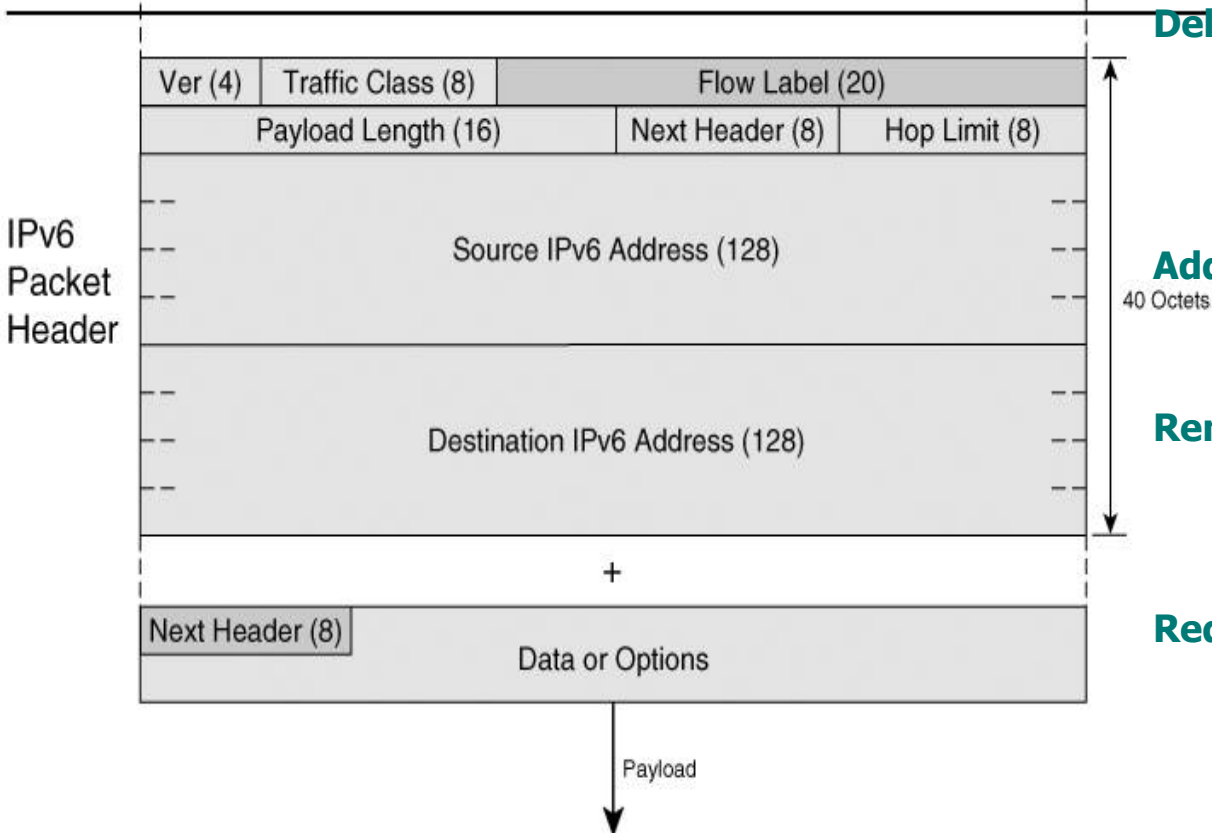
Added:

Traffic class
Flow label

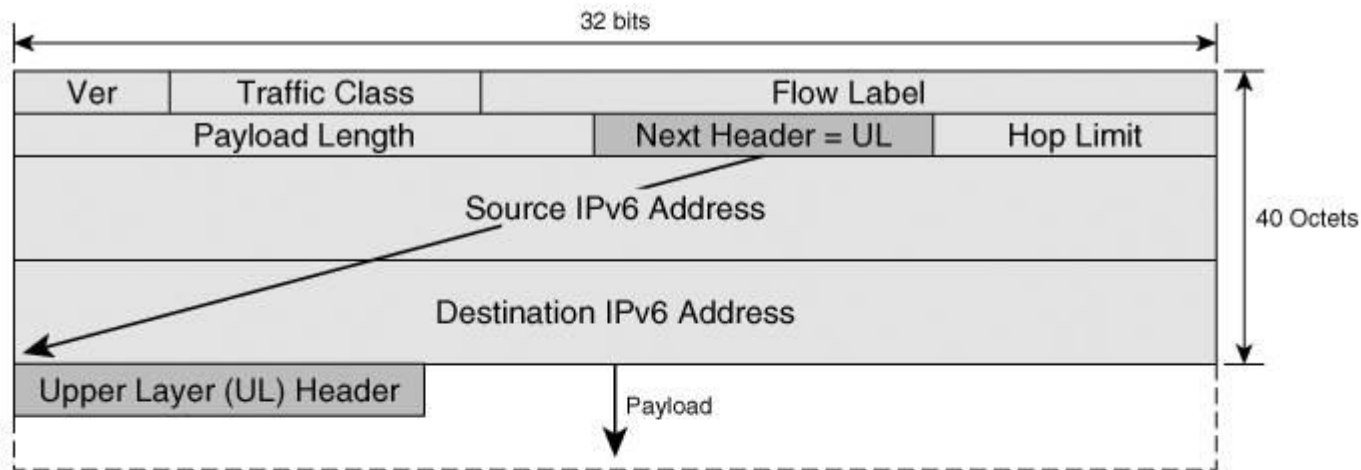
Renamed:

length -> Payload length
Protocol -> Next header
time to live -> Hop Limit

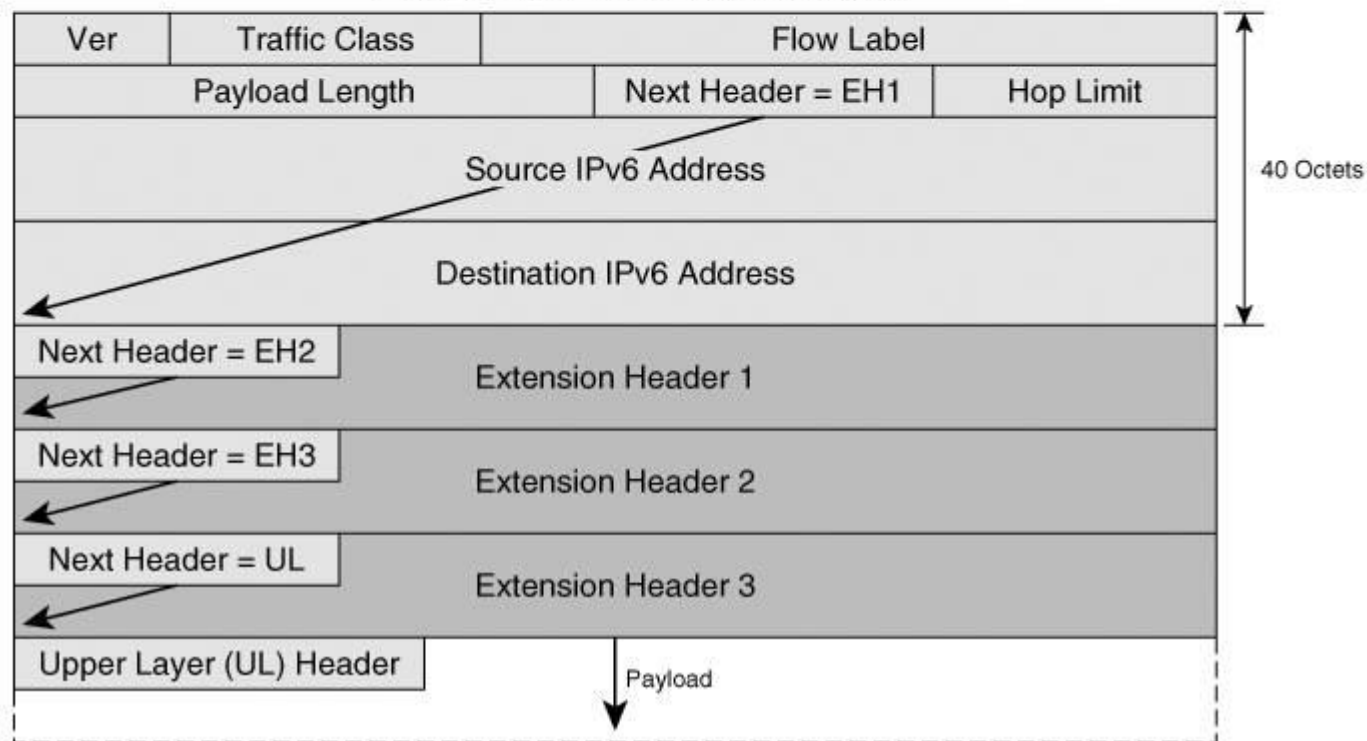
Redefined: Option mechanism



Packet without Extension Header



Packet with Extension Header

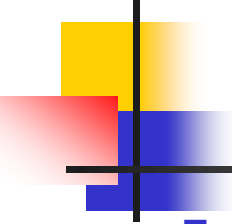


下一头标值	含义
0	中继点选项头标
4	IP
6	TCP
17	UDP
43	寻路头标
44	分片头标
45	IDRP
46	RSVP
50	封装化安全净荷
51	认证头标
58	ICMPv6
59	无下一个头标
60	信宿选项头标



IPv6扩展头标顺序

- **IPv6基本头标**
 - 中继点选项头标
 - 信宿选项头标（有寻路头标）
 - 寻路头标
 - 分片头标
 - 认证头标
 - 封装化安全净荷
 - 信宿选项头标
- 中间路由器处理
- 目的节点处理



IPv6地址

- **128位地址可产生 2^{128} 个地址**
- **理论上说，地球上每平方米有665,570,793,348,866,943,898,599个IPv6地址。**
- **保守估计，地球上每平方米可用的IPv6地址数超过1600**
- **表示方法采用十六进制冒号分割法**
 - **如1025:1ab6:0:0:0:87:a76f:1234**
 - **以上地址还可表示为1025:1ab6::87:a76f:1234**
 - **混合表示**
 - **::FFFF:129.144.52.38**
 - **地址前缀表示：IPv6地址/前缀长度**
 - **2001:da8:d800::3/64**



IPv6地址

- **IPv6地址空间分配:RFC4291**

- **单播地址(Unicast)**

- 标识单个接口,单播地址的分组被发送到该接口

- **组播地址(Multicast)**

- 标识通常属于不同节点的一组接口,组播地址的分组被发送到所有的接口

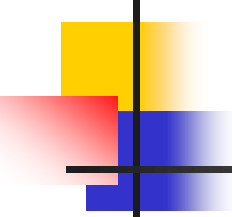
- **任播地址(Anycast)**

- 标识通常属于节点上的一组接口,任播地址的分组被发送到最近的那个接口(依据路由协议度量的最近距离)

- **IPv4地址空间分配**

- 单播地址、组播地址、广播地址

IPv6中没有广播地址,其功能被组播地址所代替



小结

- 熟练掌握**Internet**的网络层**IP**及其相关协议，掌握**ARP**协议、**IP**路由和转发、组播等机制。
- 掌握子网划分、**CIDR**、**NAT**等机制
- 实际操作**traceroute**（**tracert**）、**ping**、**ping -r**
- 思考与调研, 查找相关资料
 - 调研+抓包分析：前面我们介绍过**ICMP**协议，与**TCP/UDP**不同其中不包含端口域，那么在只有一个外部**IP**地址的情况下如何实现**NAT**？
 - 在**IPv4**中，如何实现组播地址到**MAC**地址的映射？
 - 调研两台均在**NAT**之后的主机如何进行相互通信？
- 作业
 - **23, 25, 28-33, 36,37,41(第五版)**



Chapter 8 传输层TCP & UDP

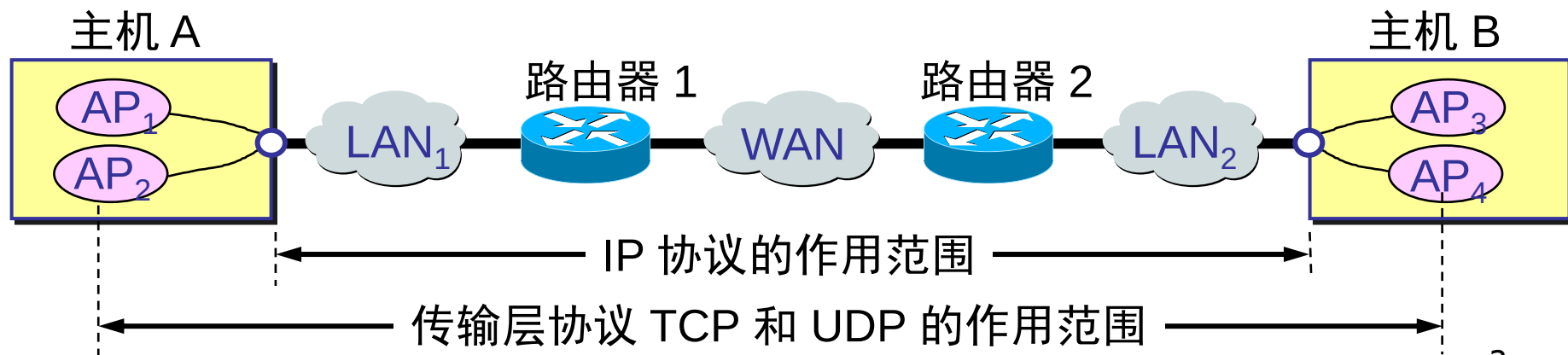
- **8.1**传输层服务
- **8.2**传输层寻址
- **8.3**建立连接
- **8.4Internet**中的传输层协议
 - 用户数据报协议**UDP**
 - 传输控制协议**TCP**
- **8.5Berkeley Socket**



Chapter 8 传输层

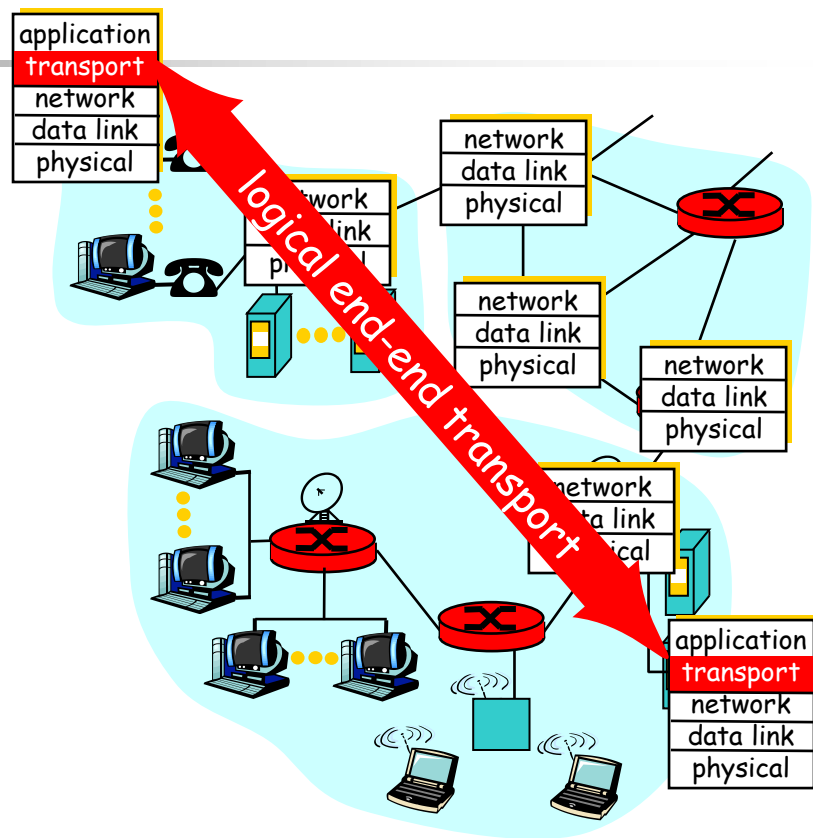
- **8.1传输层服务**
- **8.2传输层寻址**
- **8.3建立连接**
- **8.4Internet中的传输层协议**
 - 用户数据报协议**UDP**
 - 传输控制协议**TCP**
- **8.5Berkeley Socket**

传输层所处的地位

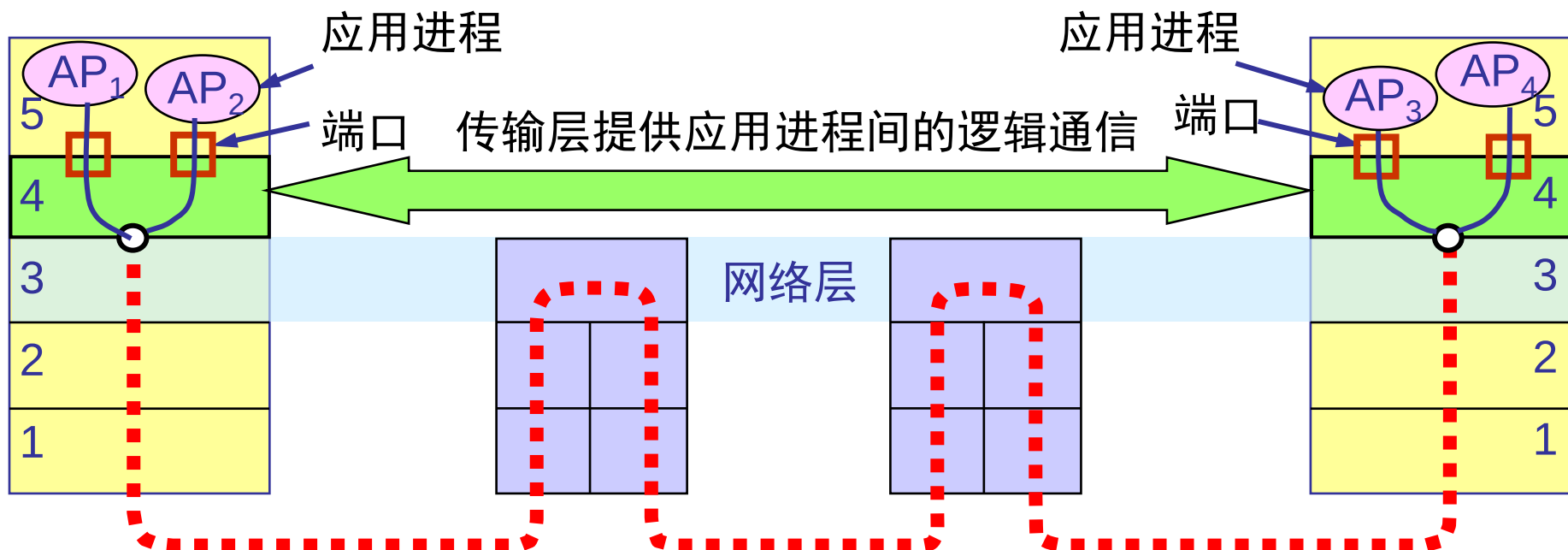


传输层服务

- 传输层是整个协议层次结构的核心，它为高层屏蔽任何与网络有关的操作，传输层是端到端协议，与使用的网络无关



为应用进程提供了逻辑通信信道





传输层导入原因

- 实现用户对数据传输的控制
 - 网络层运行在用户终端和路由器上，而传输层运行在用户主机上
 - 用户可以根据应用需求选择不同的传输层服务
- 实现运行在不同主机上的进程之间的通信
 - 每个应用进程都至少与一个传输层地址（端口）相关联
- 屏蔽下层网络的异质性
 - 对上层应用提供了一个标准的原语集合，提供传输层的虚拟管道



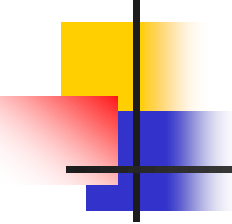
传输服务原语

- 传输服务原语允许传输用户（例如应用程序）访问传输服务

- 原语可以看作是上层访问下层服务的标准接口
- 每种传输服务都定义了各自的原语

Berkeley Socket Primitives for TCP

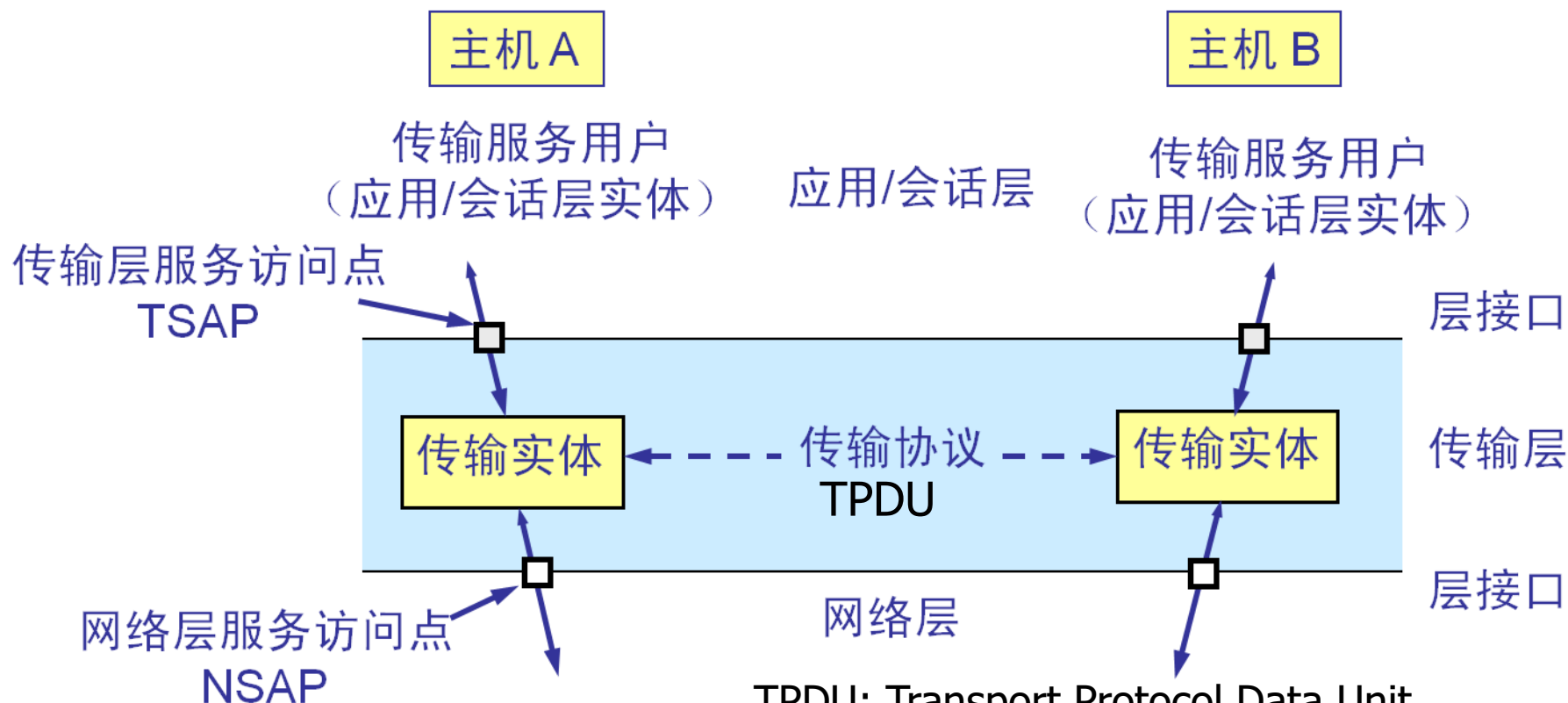
原语	含义
SOCKET	创建一个新的通信端点
BIND	将一个本地地址关联到一个套接字上
LISTEN	宣布愿意接收连接，给出队列大小
ACCEPT	阻塞调用方，直到有人企图连接上来
CONNECT	主动尝试建立一个连接
SEND	在指定的连接上发送数据
RECV	从指定的连接上接收数据
CLOSE	释放指定的连接



首先讨论的是**OSI**参考模型中的传输层的一些概念和机制，然后具体介绍**Internet**中的传输层协议

向上层提供的服务

向上层提供的服务：面向连接的传输服务和无连接的传输服务



TPDU: Transport Protocol Data Unit
TSAP: Transport Service Access Point
NSAP: Network Service Access Point



Chapter 8 传输层TCP & UDP

- **8.1**传输层服务
- **8.2**传输层寻址
- **8.3**建立连接
- **8.4Internet**中的传输层协议
 - 用户数据报协议**UDP**
 - 传输控制协议**TCP**
- **8.5Berkeley Socket**

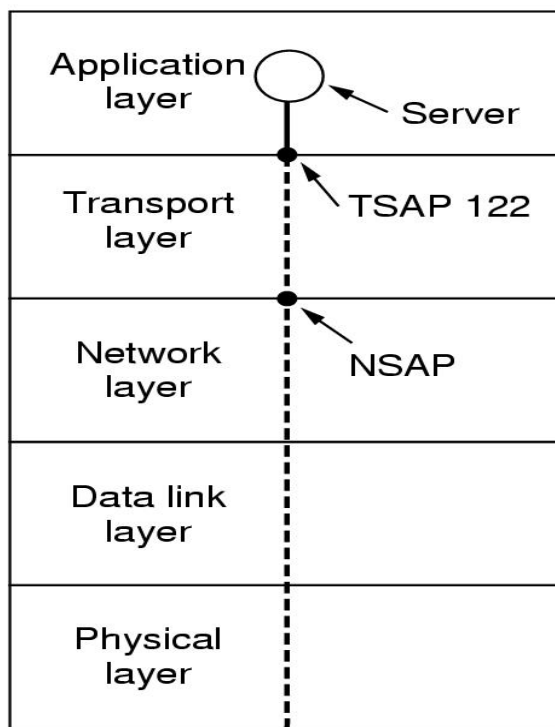
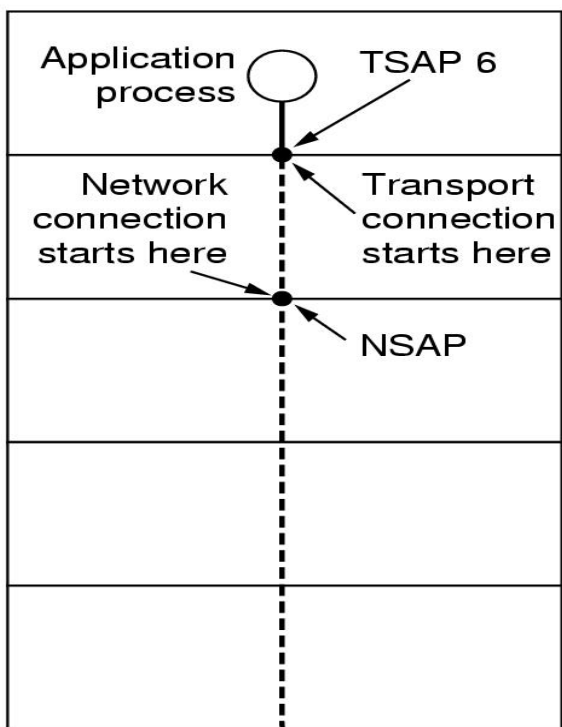
8.2 传输层寻址

TSAP: Transport Service Access Point, 对应着传输层地址
NSAP: Network Service Access Point, 对应着网络层地址

- 由于传输层为上层提供复用机制，因此，当一个传输层进程希望与另一个进程建立连接时，必须指定对方传输层所对应的**TSAP**地址

Host 1

Host 2



Host2的时间服务进程运行在122号TSAP上，等待服务请求。

Host1应用进程发出连接请求，并指明源TSAP=6，目的TSAP=122。

网络层实体向Host2建立连接（对于非连接网络，仅向Host2发送分组）

问题：如何知道对方的TSAP？



TSAP的获取方法

1. 服务进程固定在特定的**TSAP**上
只适用于少数关键的服务 周知端口(知名端口)
<http://www.iana.org/assignments/port-numbers>
2. 每个系统都有一个进程服务器，它连接在众所周知的一组**TSAP**上，用户为了与目标服务进程通信，首先必须与进程服务器通信，通过它启动目标服务进程
3. 采用名字服务器，当新的服务被创建时，必须向名字服务器注册，给出服务名和**TSAP**，用户需要相应的服务前先查找名字服务器



主机进程标识

- 网络服务访问点（**NSAP**）和传输服务访问点（**TSAP**）即{**NSAP**, **TSAP**}标识了一个主机上的进程，需要注意的是**NSAP**是**ISO**定义的概念
- 在**TCP/IP**中，**NSAP**与**IP**地址对应，**TSAP**对应着端口号（**port**），再加上所采用的协议（**TCP/UDP**），即{**IP**地址，端口号，协议类型}唯一标识一个主机上传输服务的用户进程，而{源/目的**IP**地址，源/目的端口号，协议类型}五元组标识一个传输连接的数据流。



Chapter 8 传输层TCP & UDP

- **8.1**传输层服务
- **8.2**传输层寻址
- **8.3**建立连接
- **8.4Internet**中的传输层协议
 - 用户数据报协议**UDP**
 - 传输控制协议**TCP**
- **8.5Berkeley Socket**

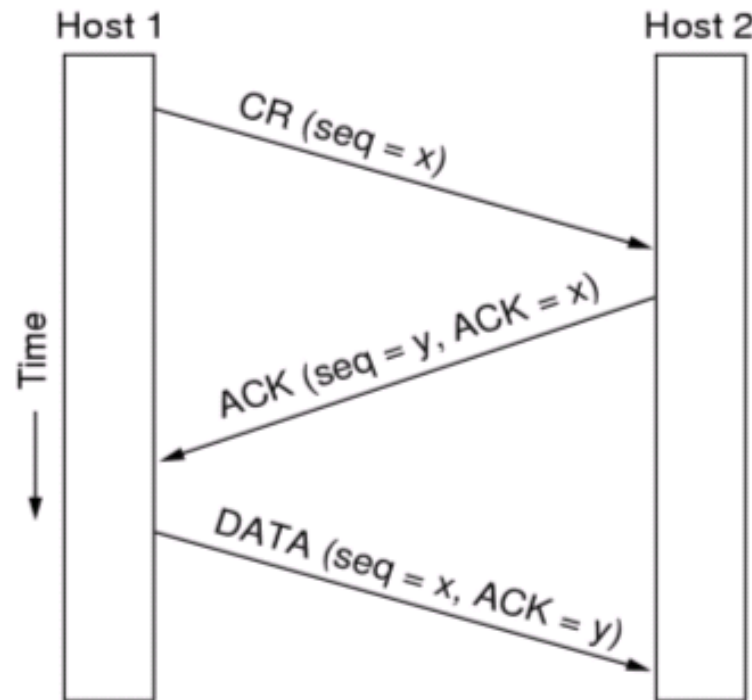


传输层连接

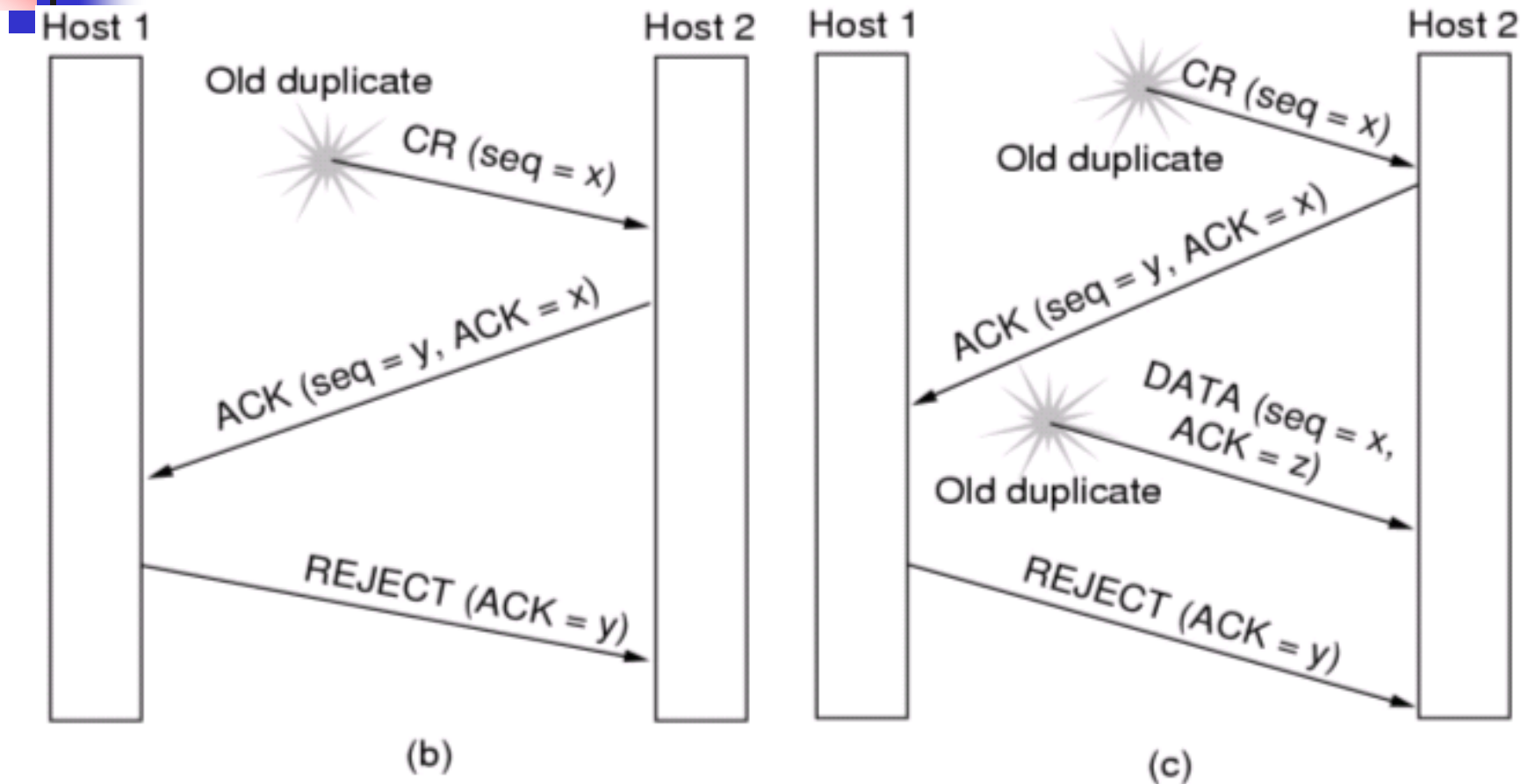
- 当传输层要向上层提供可靠传输服务时，需要在发送端和接收端协商保证这种可靠性(包括流量控制)的参数
 - 例如序列号、接收端缓存大小等
- 传输层连接本质上是在发送端和接收端上为实现可靠传输而维护的一些参数状态
- 对于面向连接的传输层，在发送数据之前要在发送端和接收端之间先进行连接建立过程

8.3 建立连接

- 一般来说，一个连接从连接请求的发出到接收到连接证实消息就可以建立一个连接，但当网络丢失、延迟导致重传而出现重复分组时，一个连接也会被建立多次
- 三步握手
 - 主机**1**选择一个序号**x**，并向主机**2**发送包含该序号的连接请求 **TPDU**
 - 主机**2**应答接受连接 **TPDU**，**TPDU**中包含确认号**x**和序列号**y**
 - 主机**1**在其发送的第一个数据中采用序列号为**x**，并确认主机**2**的序列号**y**



三步握手中的延迟重复分组



CR-连接请求; ACK-接受连接

图b为重复的CR突然出现; 图c为重复的CR和重复的ACK



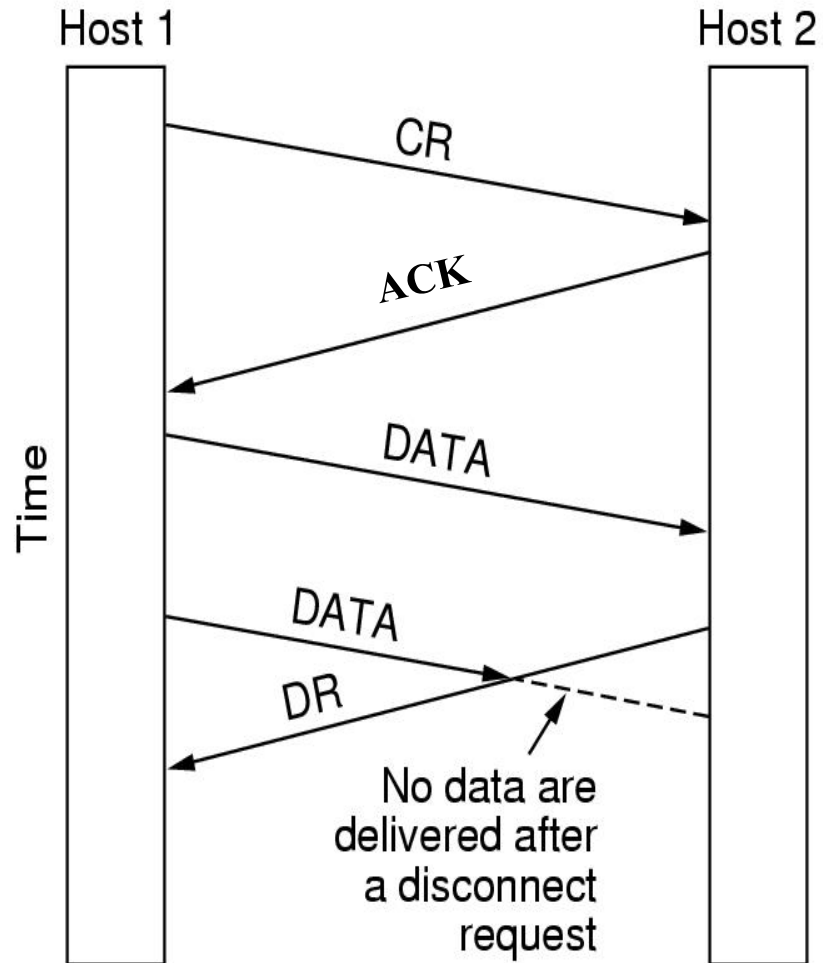
释放连接

- 非对称释放
 - 连接的任何一方都可以断开整个连接
- 对称释放
 - 把连接看作是由两个独立的单向连接，并要求单独释放每一个单向连接

非对称释放

当连接建立以后，主机1发送一个TPDU，它正确地到达了主机2，然后，主机1又发送一个TPDU，但是由于主机2在第二个TPDU到达之前就已经发送了DISCONNECT TPDU，因此，该连接虽然被释放，但是数据却丢失了。

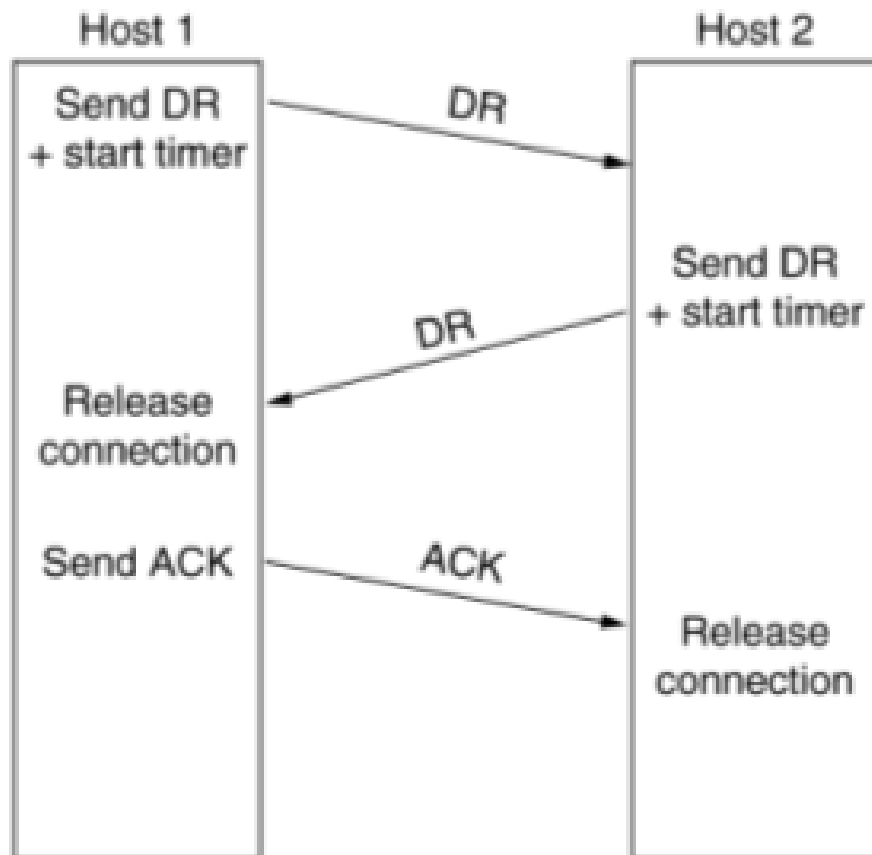
CR-连接请求
ACK-接受连接
DR-连接断开请求



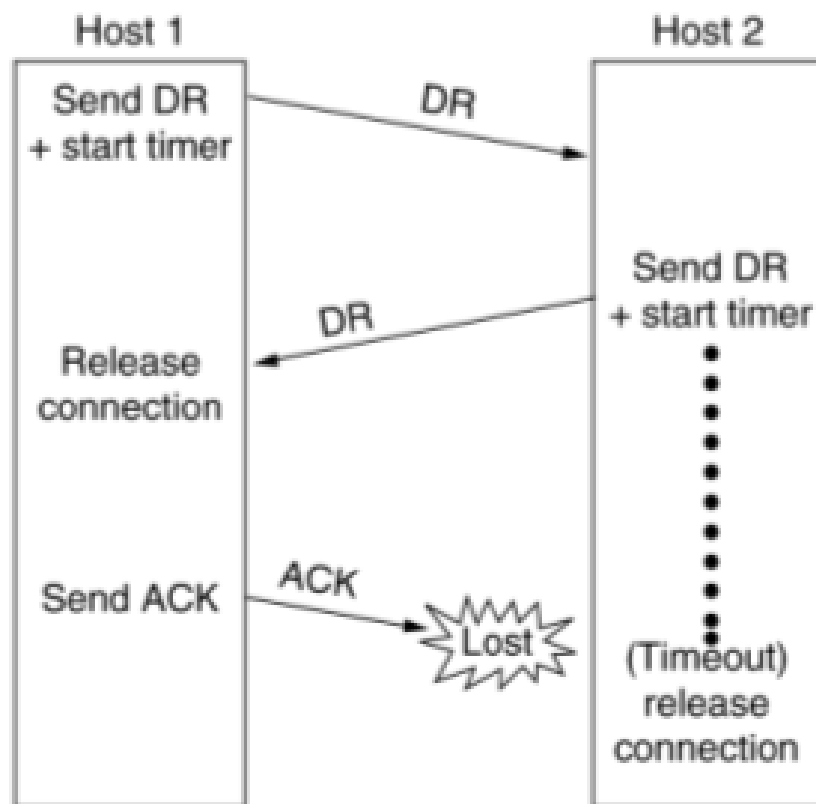
对称释放

DR-连接断开请求
ACK-接受断开

释放连接再次用到三步握手，一方发出释放连接请求后不立即拆除连接，而要等待对方确认；对方收到请求后，发送确认报文，并拆除连接，发起方收到确认后最后拆除连接。



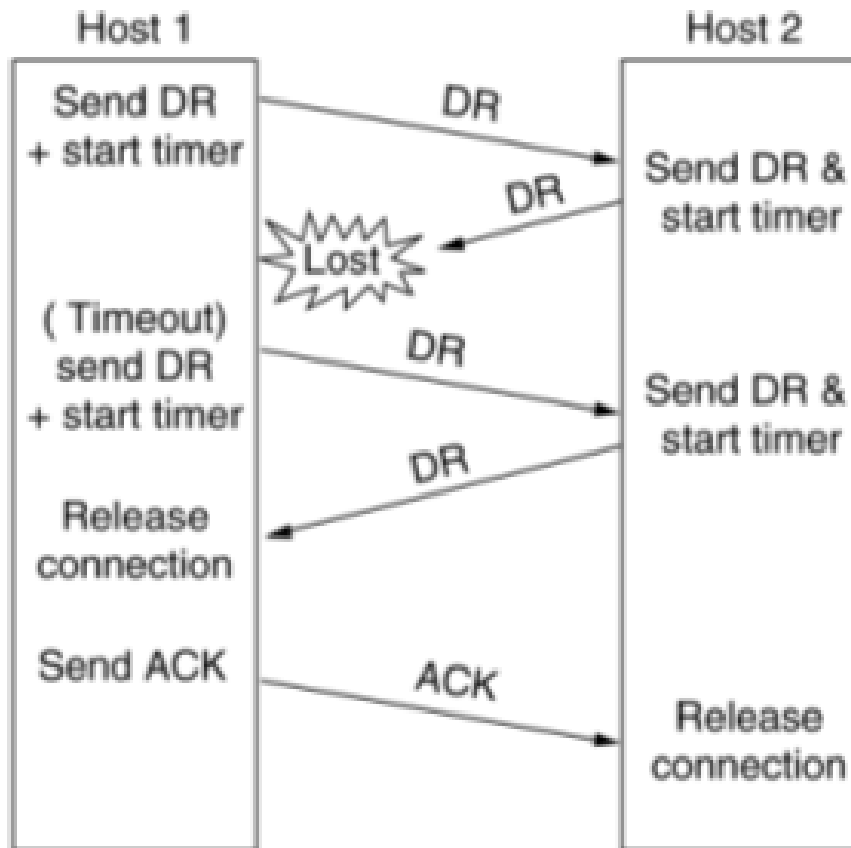
对称释放中的分组丢失



DR-连接断开请求
ACK-接受断开

(a) final ACK lost

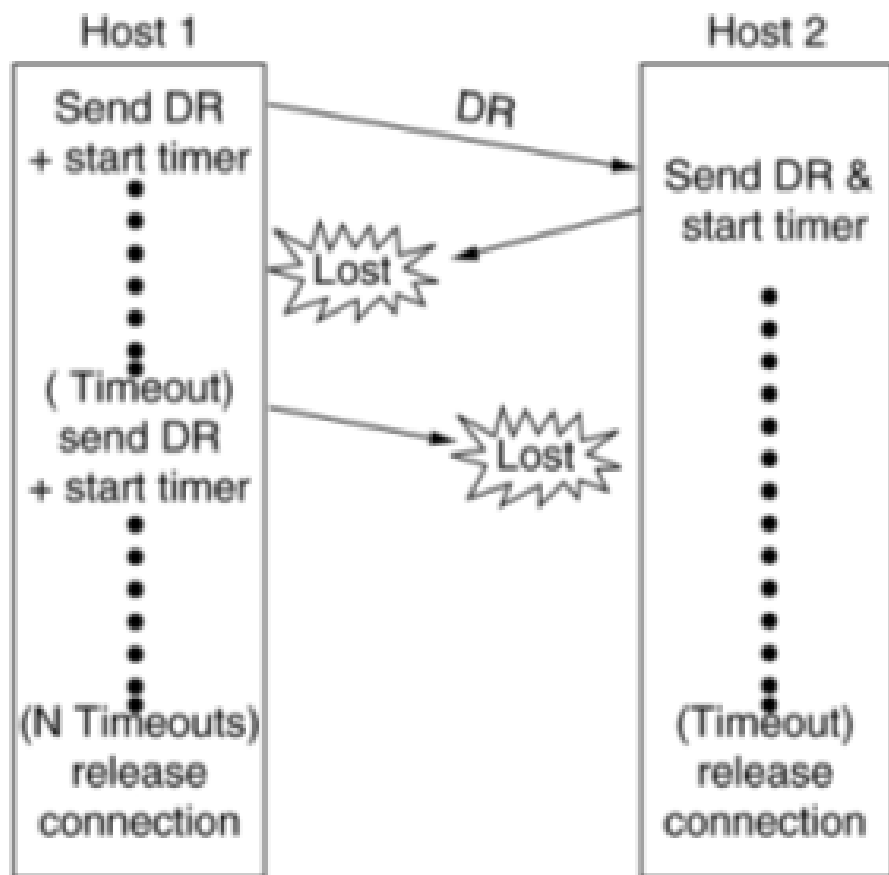
对称释放中的分组丢失



DR-连接断开请求
ACK-接受断开

(b) Response lost

对称释放中的分组丢失



DR-连接断开请求
ACK-接受断开



数据传输中的流量控制和拥塞控制

- 面向连接的传输服务应该提供流量控制和拥塞控制功能，核心是滑动窗口算法
 - 流量控制：根据接收端的缓存容量来动态地调整发送端的窗口大小，避免一个快速的发送端淹没一个慢速的接收端
 - 拥塞控制：根据网络的承载容量来动态地调整发送端的窗口大小，避免发送端发送的数据超过了网络容量
 - 拥塞发生在路由器上，但终究是由于网络中的各个节点发送太多的流量引起的。
 - 在无连接的网络层中，无法在网络层控制速率，因此，拥塞控制的唯一途径是传输层（面向连接的）放缓发送速率，拥塞控制是网络层和传输层共同的责任。

发送方的窗口大小决定了发送方一次能够发送的数据量



Chapter 8 传输层TCP & UDP

- **8.1**传输层服务
- **8.2**传输层寻址
- **8.3**建立连接
- **8.4Internet**中的传输层协议
 - 用户数据报协议**UDP**
 - 传输控制协议**TCP**
- **8.5Berkeley Socket**



8.4 Internet中的传输层协议

- **Internet**中常用的两个传输层协议
 - **用户数据报协议 (UDP: User Datagram Protocol)**，提供无连接的服务，**高效**，适合于实时传输；**UDP**中的**TPDU**称为**数据报 (datagram)**
 - **传输控制协议 (TCP: Transmission Control Protocol)**，提供**可靠**的，面向连接的服务；**TCP**中，**TPDU**称为**TCP数据段 (segment)**
- 此外，传输层还有：
 - **SCTP(Stream Control Transmission Protocol)**流控制传输协议，针对多媒体应用，是可靠的面向报文的协议。面向连接、提供全双工服务，有拥塞和流控机制，并提供多流服务。
 - **MPTCP(Multi-path TCP)**,提供多路径流服务。
 - **QUIC** (Quick UDP Internet Connection),是谷歌制定的一种基于**UDP**的低时延的互联网传输层协议（属于应用层中间件）。



UDP与TCP

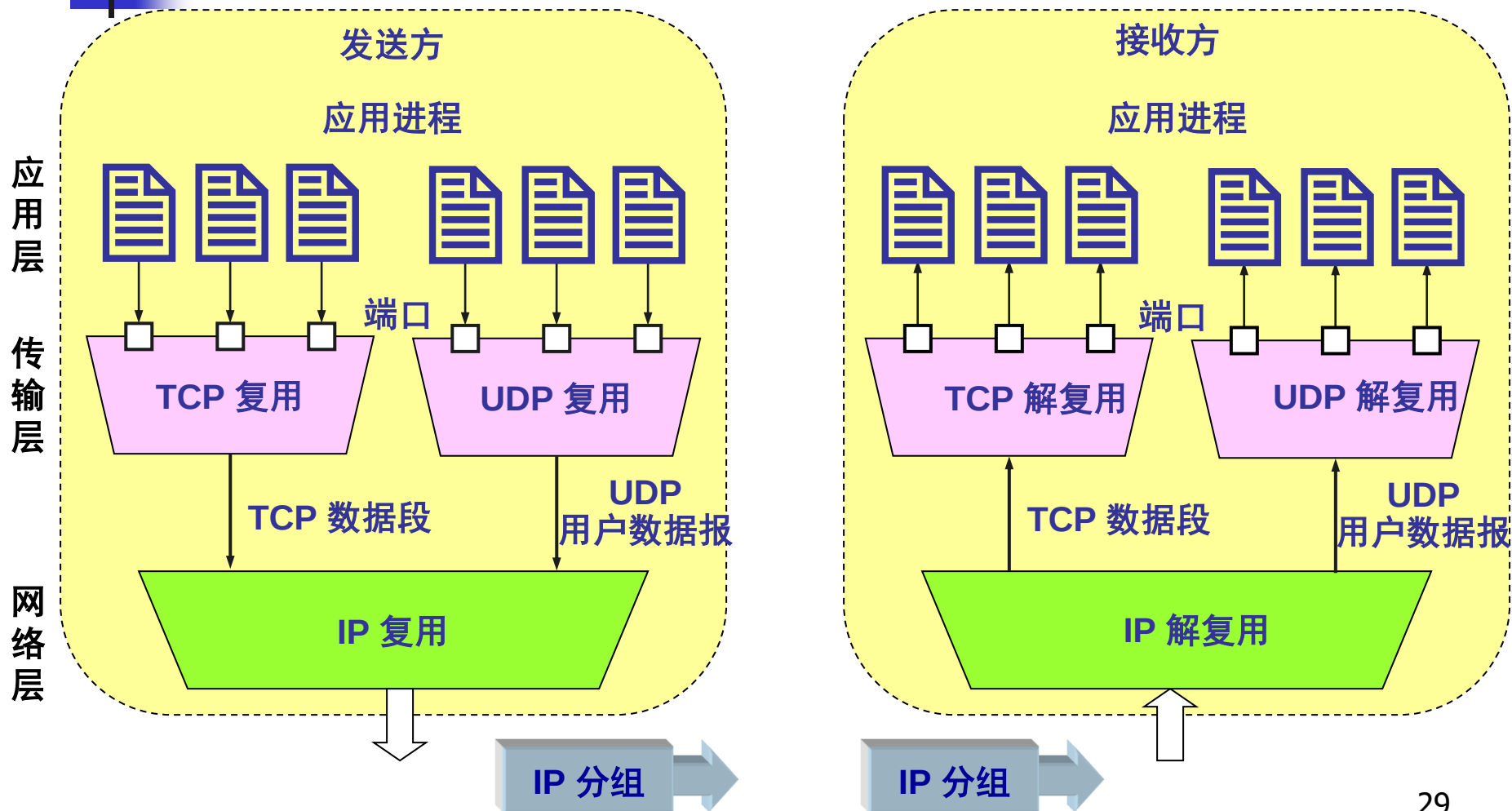
- **UDP**提供无连接的服务，在传送数据之前不需要先建立连接。对方的传输层在收到 **UDP**数据报后，不需要给出任何确认。虽然 **UDP** 不提供可靠投递，但在某些情况下 **UDP** 是一种最简单有效的工作方式。例如视频点播等实时应用常使用**UDP**
- **TCP** 提供可靠的，面向连接的服务。由于 **TCP** 要提供可靠的、面向连接的传输服务，因此不可避免地增加了许多的开销。这不仅使协议数据单元的头标增加了更多的域，还要占用许多的处理资源。



端口

- **TCP/UDP**中的端口就是传输服务访问点（**TSAP**）
- 端口是用来标识应用进程
 - 每一个网络应用进程都与端口相关联
 - 各种网络应用进程都能将其数据通过端口向下交付给传输层
 - 传输层根据接收到的**TPDU**（数据报或数据段）中包含的端口信息将其递交给与该端口关联的网络应用进程
- 部分知名端号
 - 21/tcp FTP 服务器
 - 22/tcp SSH服务器
 - 23/tcp Telnet 服务器
 - 25/tcp SMTP发送邮件服务器
 - 80/tcp WWW服务器

基于端口的进程间通信

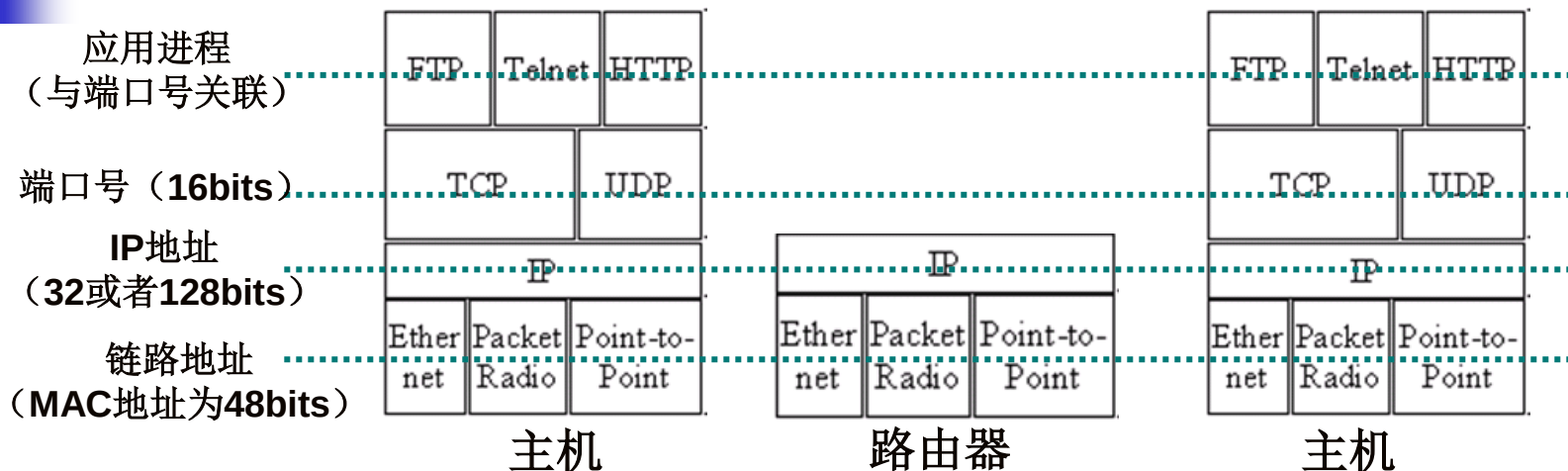




端口号

- 端口用一个 **16 bit 端口**号进行标志，共**64K**个端口号对一台主机来说是足够的。
- 端口号只具有本地意义，即端口号只是为了标识本机上的各个应用进程。在**Internet**中不同主机的相同端口号是没有联系的。
- 端口号分为两类：一类是**知名端口**，其数值一般为 **0-1023**，一般分配给一些常用的服务进程。另一类则是一**般端口**，用来随时分配给请求通信的客户进程。

Internet寻址模型



端口: 在主机上标识应用进程，数据中的端口号由程序或者系统指定

IP地址: 标识Internet上的某台主机，数据中的IP地址由程序或者系统指定

发送数据包括发送方的地址或者端口（源地址和源端口），接收方地址或者端口（目的地址和目的端口），一般来说跨越网络传输保持不变

链路地址（MAC地址）: 标识共享链路上的某个接口设备，固化在设备中，数据中的MAC地址一般由系统指定

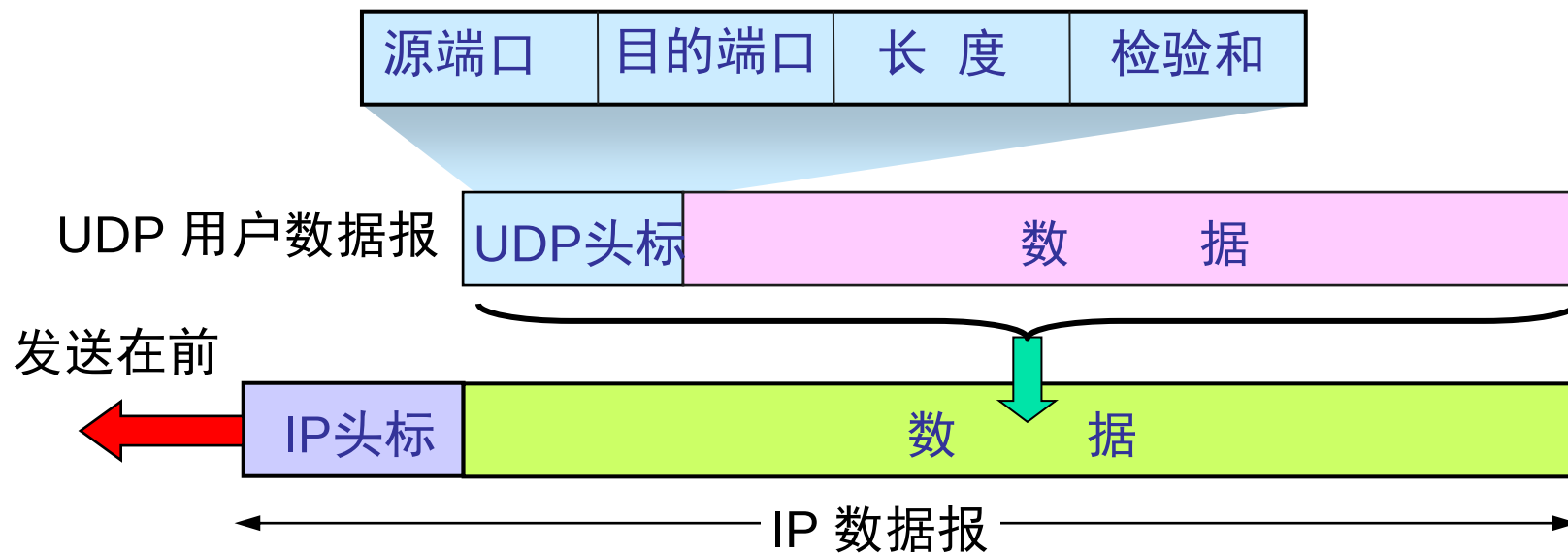
帧的源和目的MAC地址随链路变化



8.4.1 用户数据报协议-UDP

- **UDP: User datagram protocol**
 - **RFC 768**
- 传输的数据单元称为数据报
- 无连接的传输服务
 - 不可靠 (**Unreliable**)
 - 高效率

UDP头标

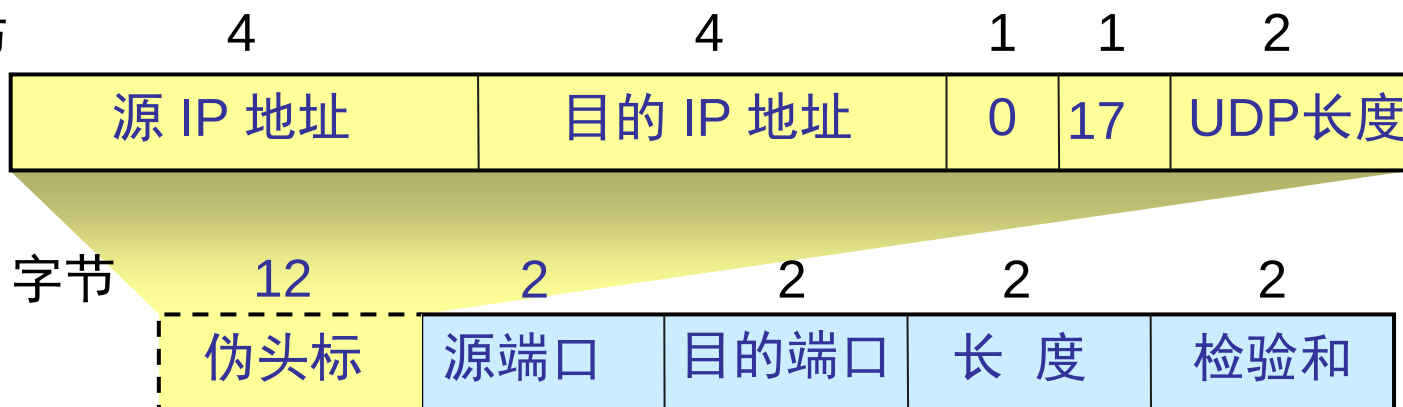


UDP头标

IP伪头标

0	8	16	31
源IP地址			
目的IP地址			
0	协议	UDP长度	

字节



1. 在**UDP**数据段前面加上**IP**伪头标
2. 当数据域的长度为奇数字节时，则数据域被填充一个额外的**0**字节
3. 将校验和域的值置为**0**
4. （伪头标+**UDP**头标+数据+可能的填充）以**16**比特为单位反码求和
5. 当结果为**0**时，将校验和置为全**1**。除此之外的情况，将结果原封不动地作为校验和



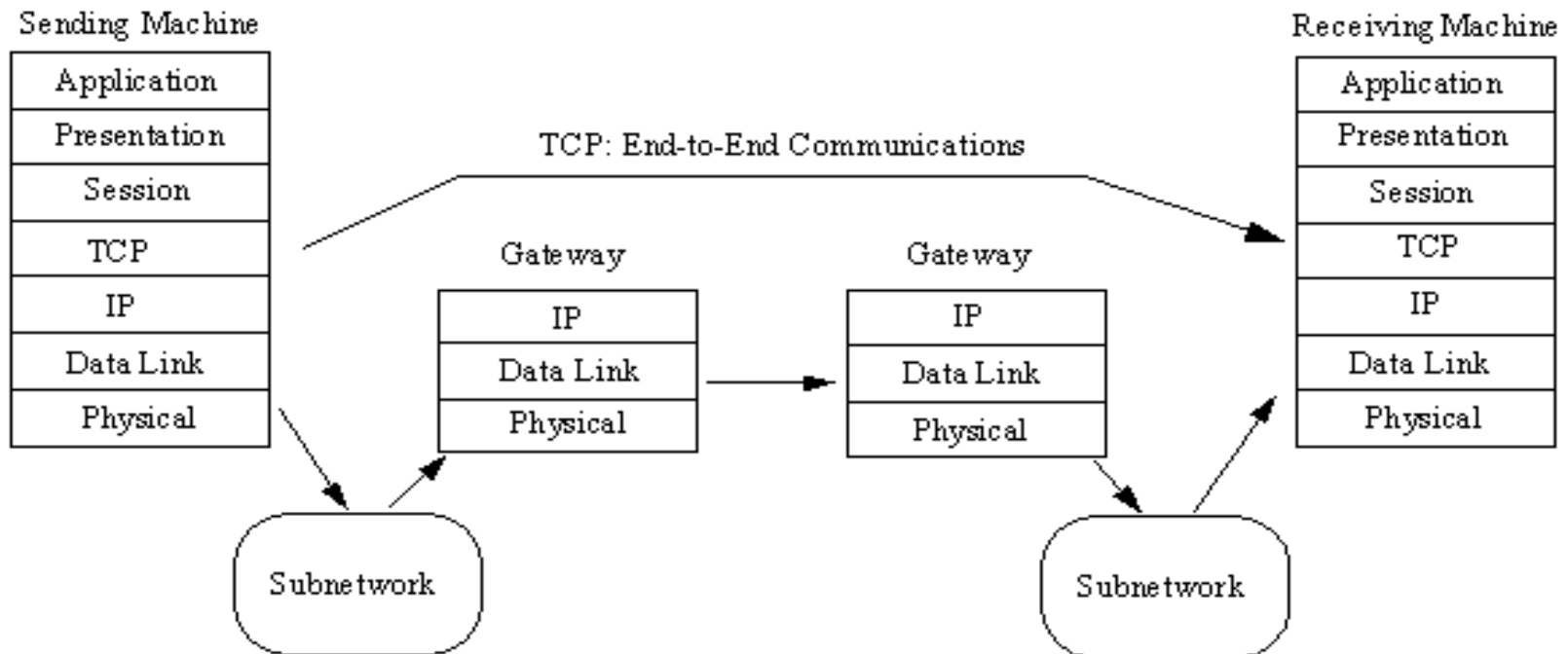
UDP应用

- 具有简单高效的特点
- 客户-服务器模式应用
 - **DNS: Domain Name System**
- 话音、视频等实时多媒体应用

问题：为什么要在**Internet**中引入**UDP**？

8.4.2 传输控制协议-TCP

■ Transmission Control Protocol-RFC 793



TCP travels in IP datagrams

Internet routers only look at IP header to forward datagrams

TCP at destination interprets TCP messages



TCP 特征

■ 字节流传输 (**Stream Transfer**)

- **TCP**把连接上的数据字节都编号，起始编号是随机的。
- 滑动窗口机制：发送端每次发送的数据量由发送窗口大小决定，而不是以应用层递交的消息为单位；接收端每次向上层递交的数据量由落在接收窗口中并且有序的数据决定。

■ 流量控制 (**Flow Control**)

■ 拥塞控制(**Congestion Control**)

■ 可靠 (**Reliability**)

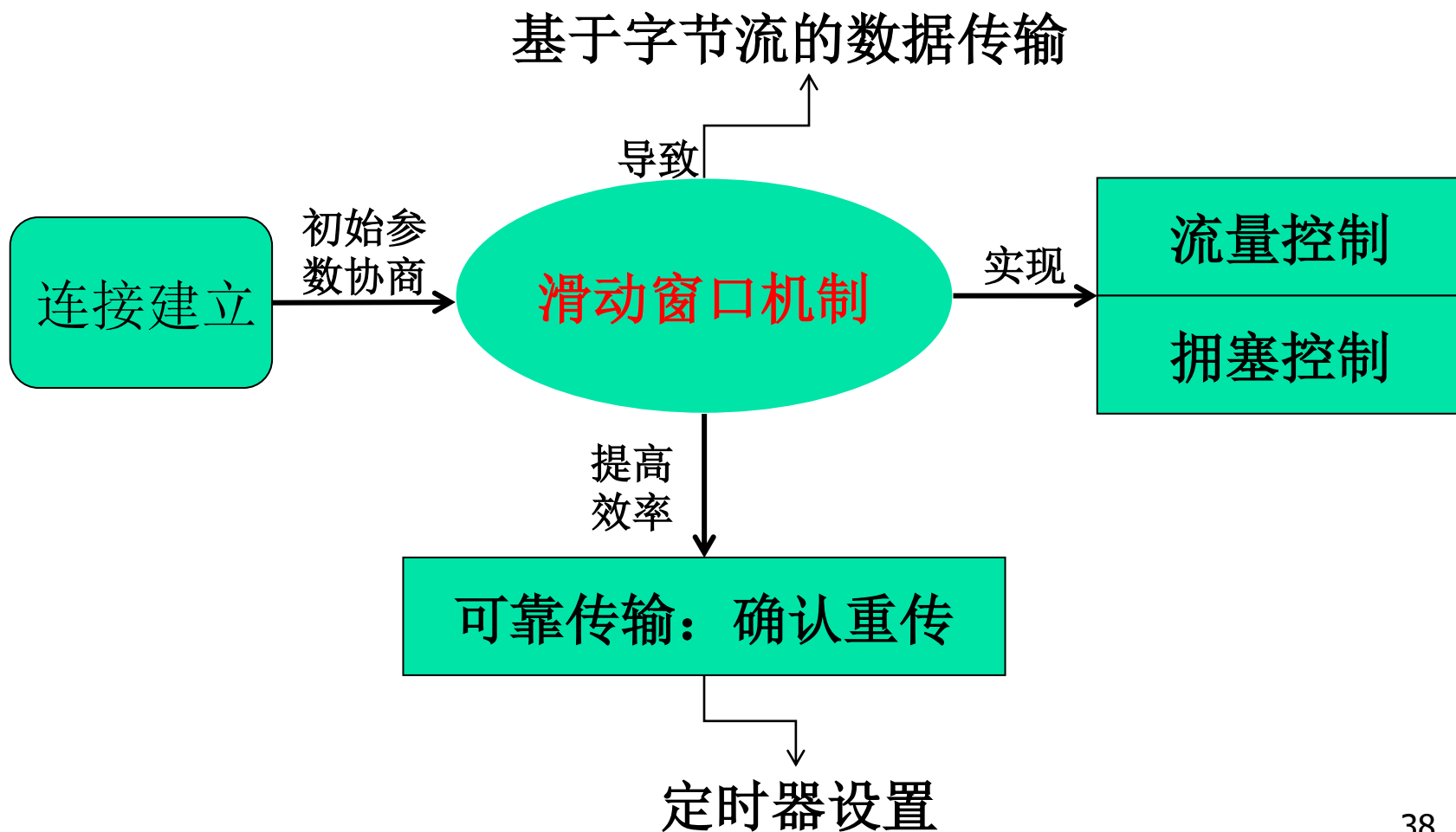
- 发送端：超时重传
- 接收端：发送确认，缓存处理乱序

■ 全双工 (**Full Duplex**)

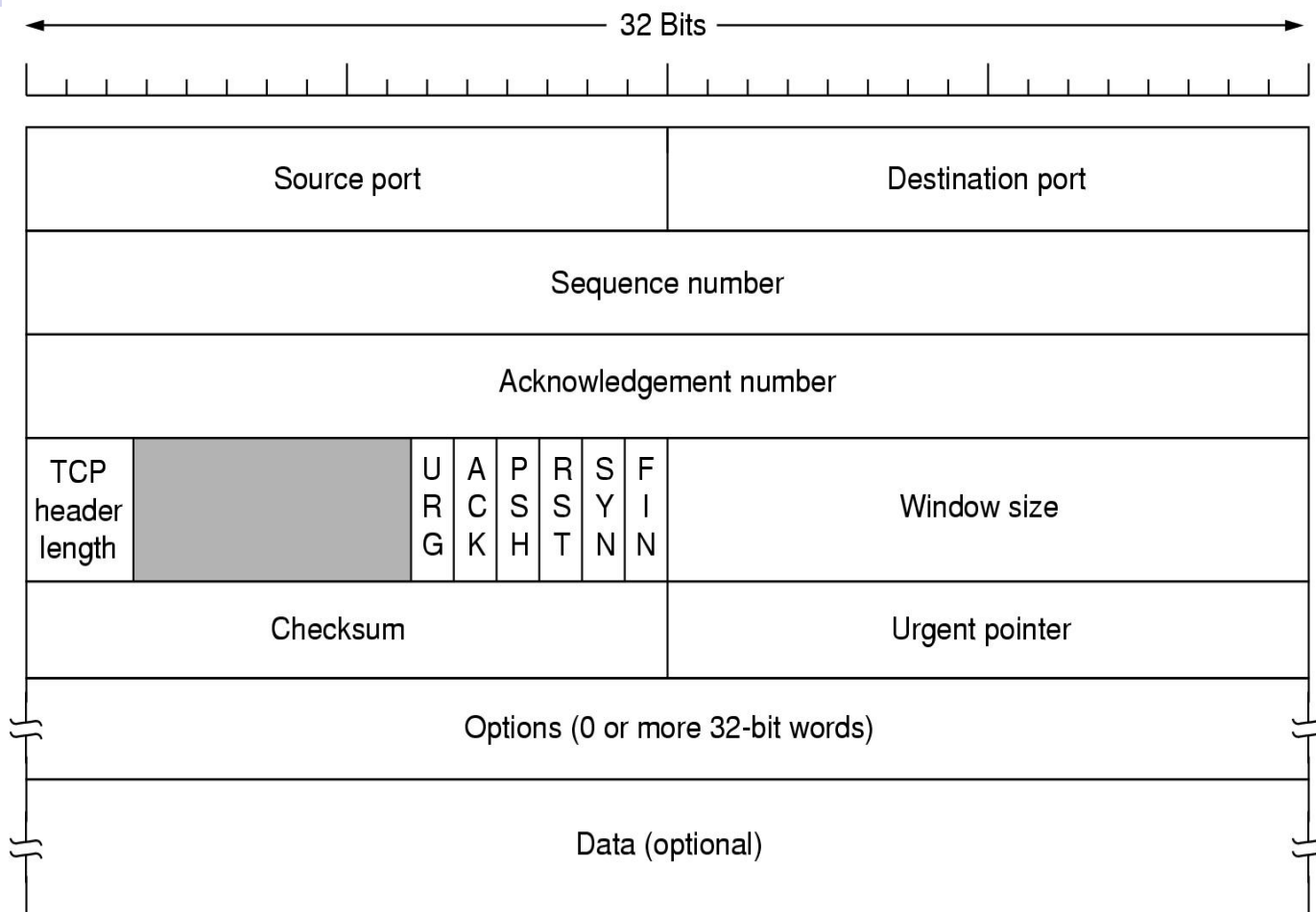
- 一个连接上传输两个方向的**TCP**数据流，各自的编号是独立的。

■ 传输的数据单元称为数据段 (**Segment**)。

TCP中的数据传输机制

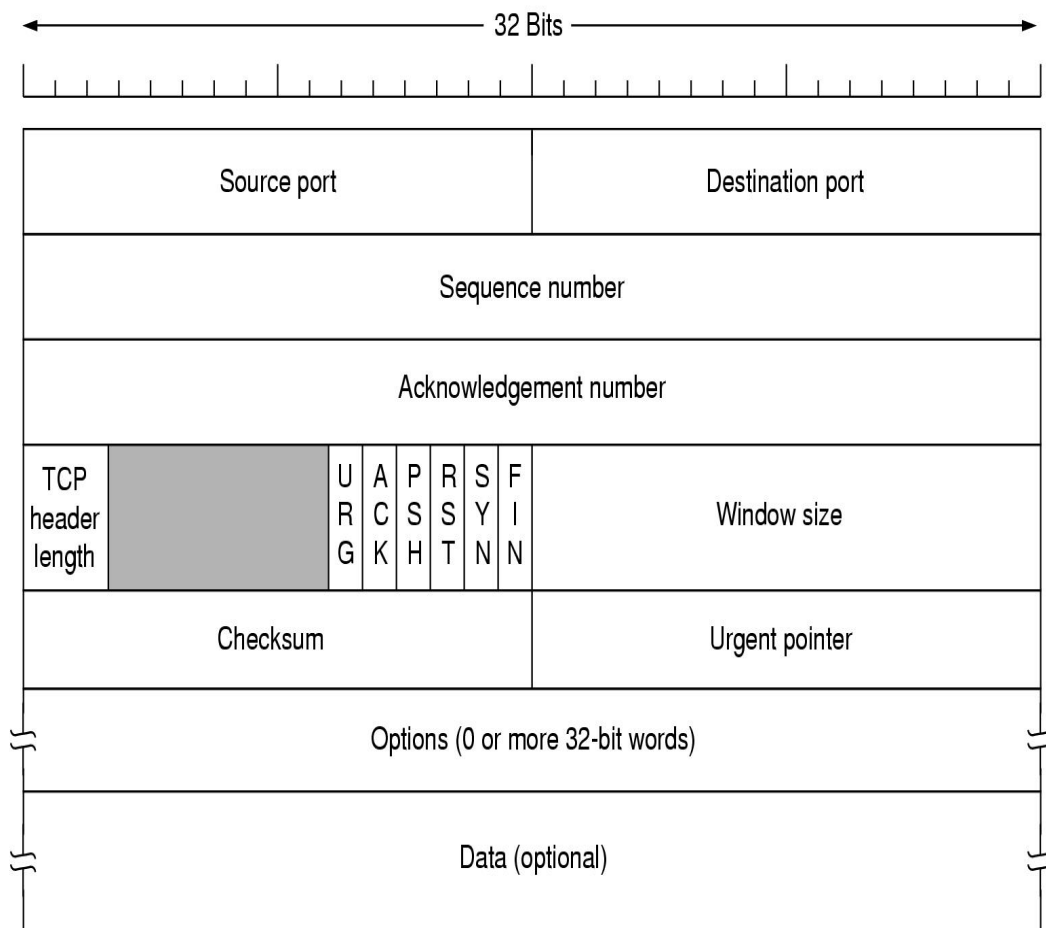


TCP头标



TCP头标

- 源端口/目的端口（**Source/Destination Port**）
 - 16位的端口号
- 序列号（**Sequence Number**）
 - 标识本数据段中第一个字节在数据流中的位置。
- 确认号（**Acknowledgment Number**）
 - 标识本数据段的接收方下一个期待接收的字节编号。
- 头标长度（**Header Length**）
 - 4bits 指明数据段的头标长度，单位是32位（4字节），随选项长度而定
- 保留（**Reserved**）：6bits，置0



TCP头标-标志位

■ URG

- 指示紧急指针有效。

■ ACK

- 指示确认段有效

■ PSH

- **Push** 操作，告诉接收主机立即将数据递交给应用程序

■ RST

- 重新连接。

■ SYN

- 同步序列号，用于建立连接

■ FIN

- 发送方已发送完数据，用于释放连接

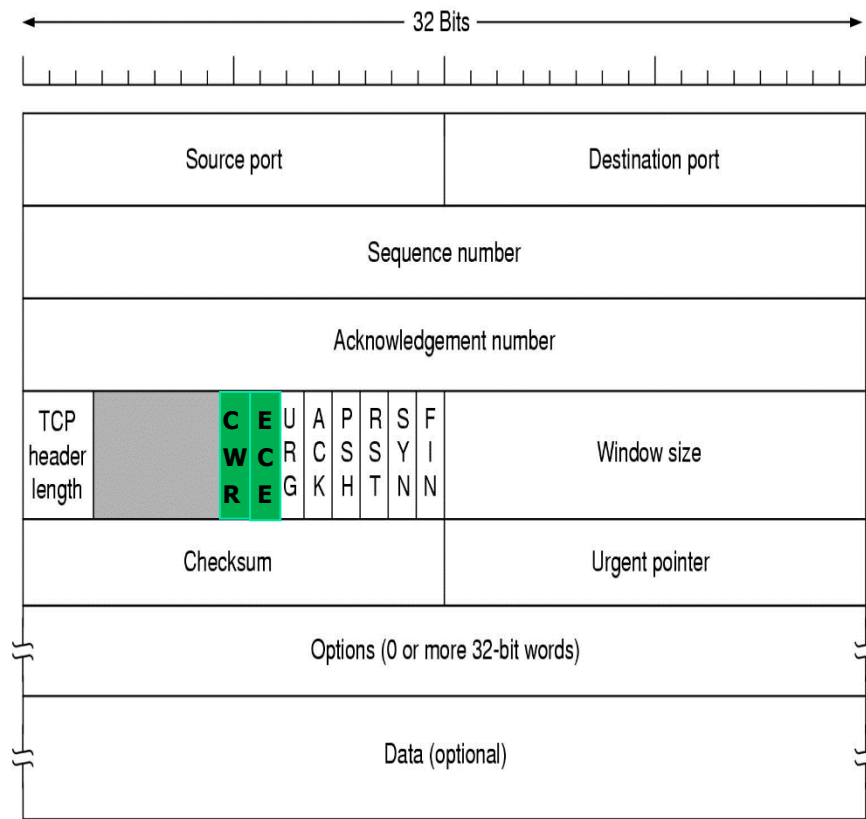
■ CWR

- Congestion Window reduced

■ ECE

- Explicit Congestion Notification

ECN {



TCP头标

- 窗口大小 (**Window size**)

- 用于流量控制，窗口指示在对方确认了的字节之后还可发送的字节数

- 检验和 (**Checksum**)

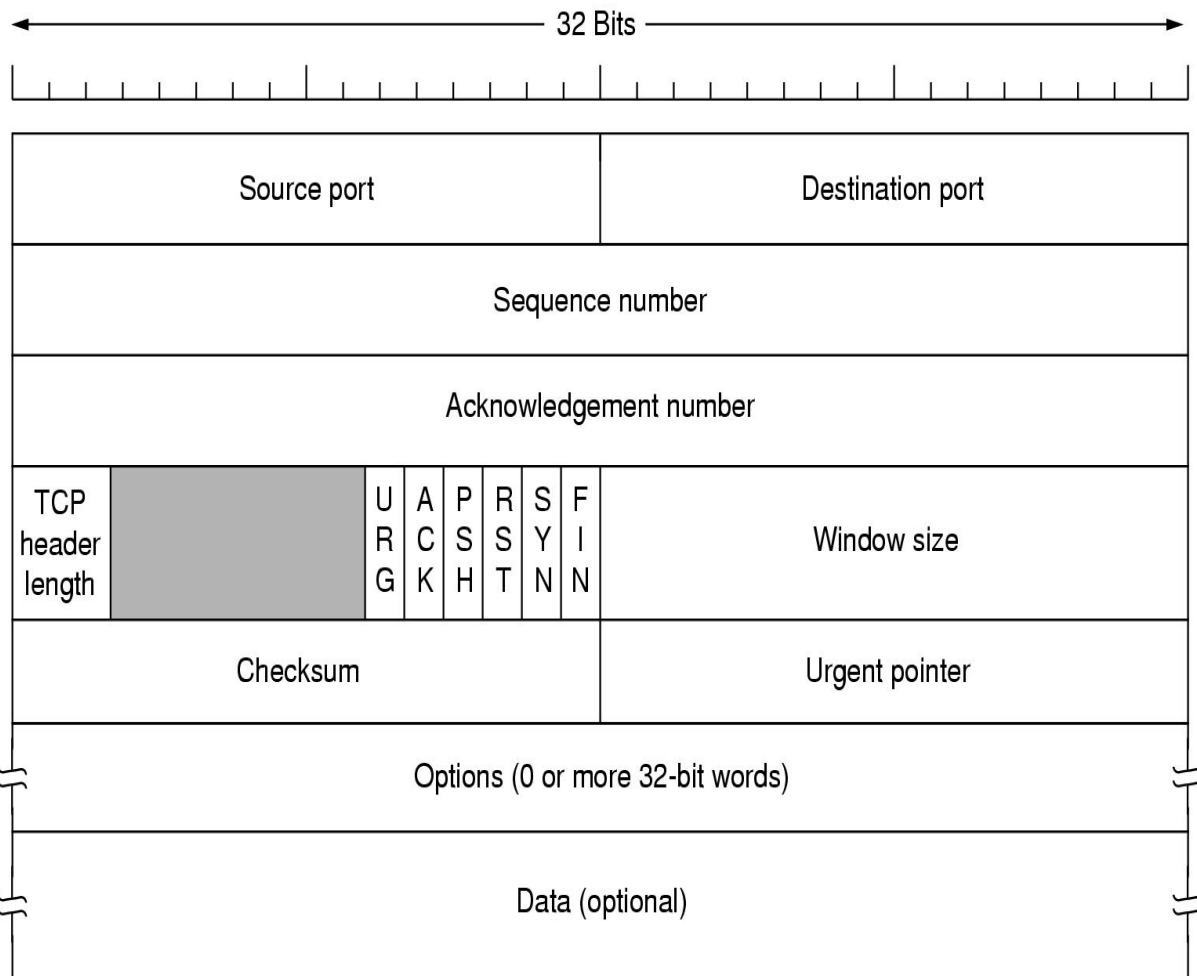
- 用来检验TCP头标和数据的完整性，检验时，要加上伪IP头标

- 紧急指针 (**Urgent Pointer**)

- 指向紧急数据的最后一个字节，该指针仅在URG置1时有效

- 选项 (**Options**)

- 最大数据段尺寸选项
 - 窗口尺度 (扩大) 选项



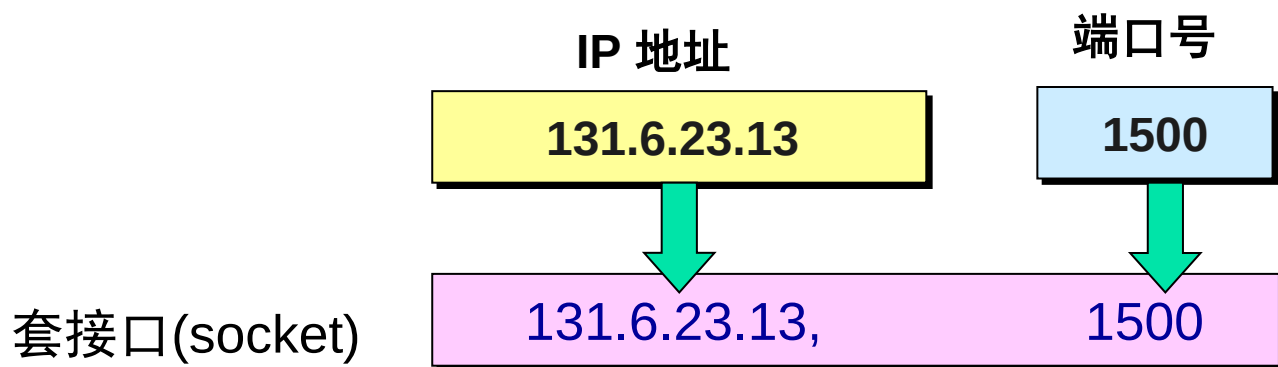


伪IP头标 (Pseudo-IP Header)

0	8	16	31
Source IP address			
Destination IP address			
zero	Protocol	TCP Length	

TCP连接

- TCP 使用“连接”（而不仅仅是“端口”）作为最基本的抽象，同时将 TCP 连接的端点称为**套接口(socket)**，或套接字、插口。
- 套接口和端口、IP 地址的关系是：



- 一条TCP连接是由发送方套接口和接收方套接口加上协议类型来唯一标识的，即TCP连接用五元组<源端IP地址、源端口号、目的IP地址、目的端口号、协议>来唯一标识。

➤ 例如：<136. 6. 23. 13, 1500, 130. 42. 85. 15, 25, TCP>

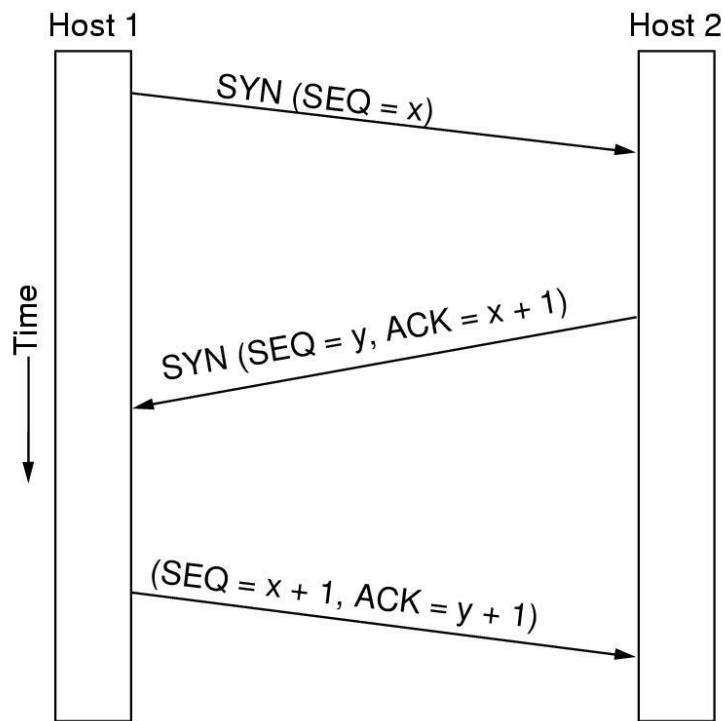


可靠的数据传输服务

- **TCP在不可靠的IP上为应用进程提供可靠的数据传输服务**
 - **连接建立：**在发送主机和接收主机之间协商初始序列号等信息
 - **连接管理：**使用滑动窗口算法对数据传输进行管理，包括确认重发、流量控制和拥塞控制等
 - **连接释放：**数据传输结束后释放连接，发送主机和接收主机释放为该连接分配的资源

TCP采用三步握手的方法建立连接

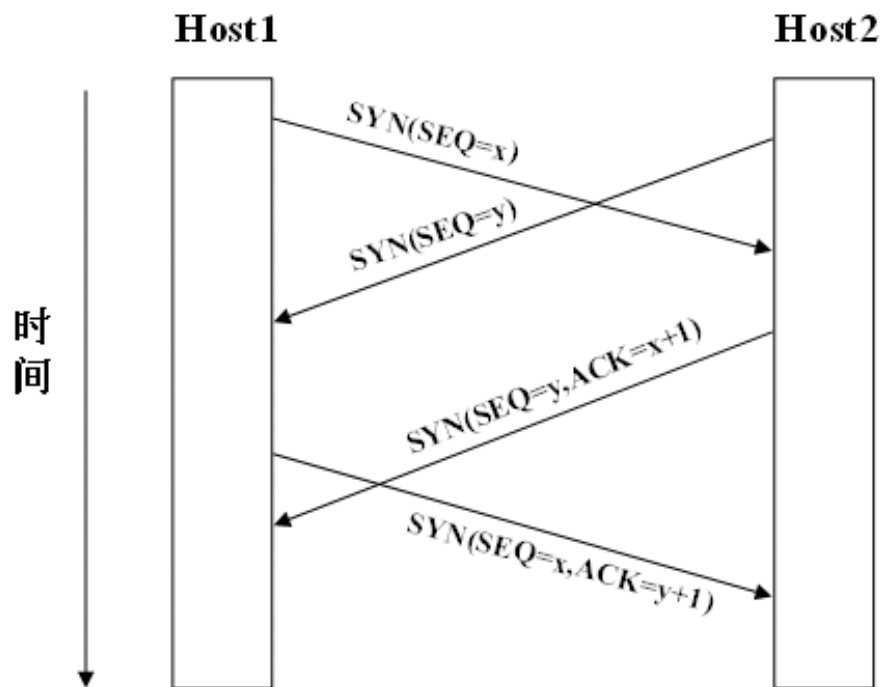
- 客户(**Host 1**)请求连接TCP数据段:
SYN标志置**1**, **ACK**标志置**0**, 选择初始序列号**SEQ=x**
- 服务器(**Host 2**)响应连接确认TCP数据段: **SYN**标志置**1**, **ACK**标志置**1**, 选择初始序列号**SEQ=y**, 并且对**x**进行确认, 设置**ACK=x+1**
 - **SYN+ACK**数据段看作是**1**字节数据
- 客户(**Host 1**)响应连接确认TCP数据段: **ACK**标志置**1**, **SEQ=x+1**, 对**y**进行确认, 设置**ACK=y+1**
 - **SYN+ACK**数据段看作是**1**字节数据



通过三步握手, **host1**和**host2**之间完成初始序列号、窗口大小、最大数据段尺寸等参数的协商, 并且分配连接所需要的资源

TCP请求连接碰撞

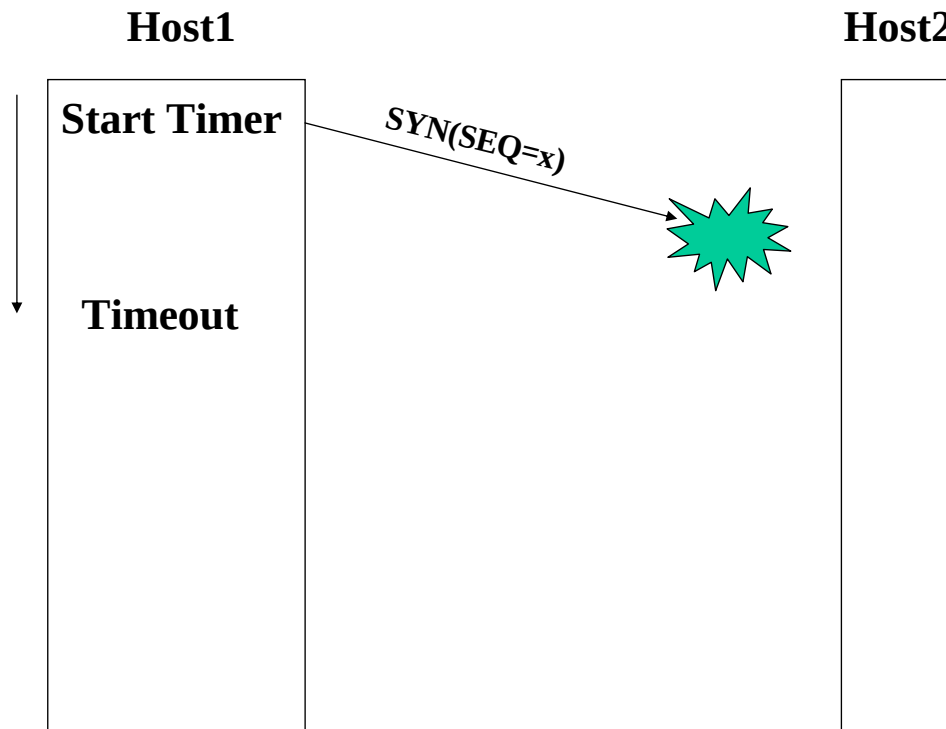
使用三步握手，即使两台主机同时想在相同的套接口之间建立一个TCP连接而发生冲突，也可以正常工作。但此时只有一个TCP连接建立起来。因此TCP连接由五元组唯一标识。



TCP连接建立定时器

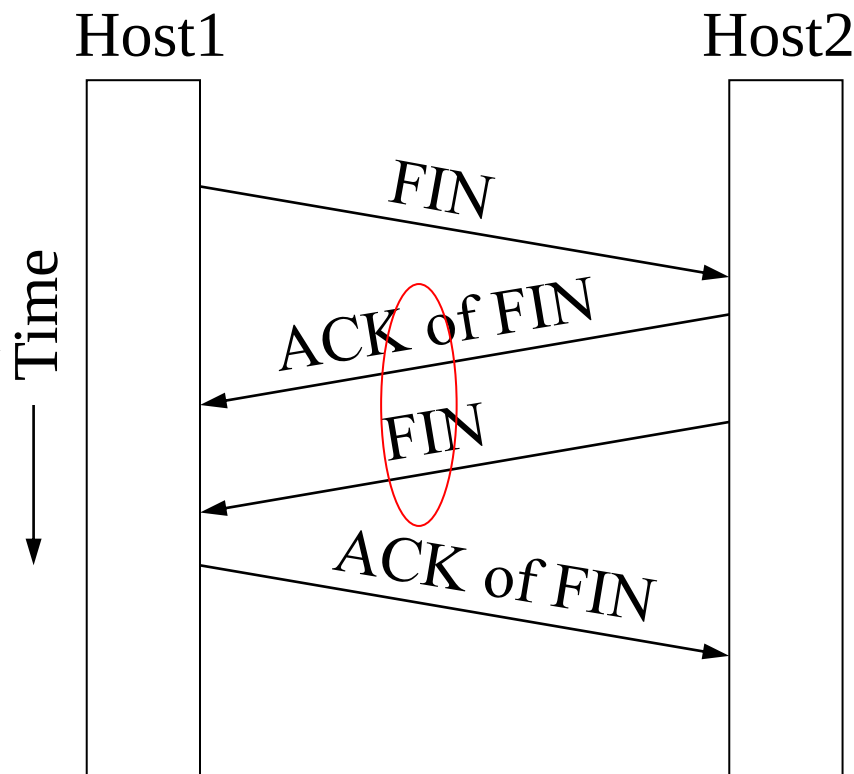
Connection Establishment Timer

当**SYN**数据段发出时，连接建立定时器就开始计时，如果在一定时间（例如**65**秒，可设置）内未收到响应，则连接建立失败。



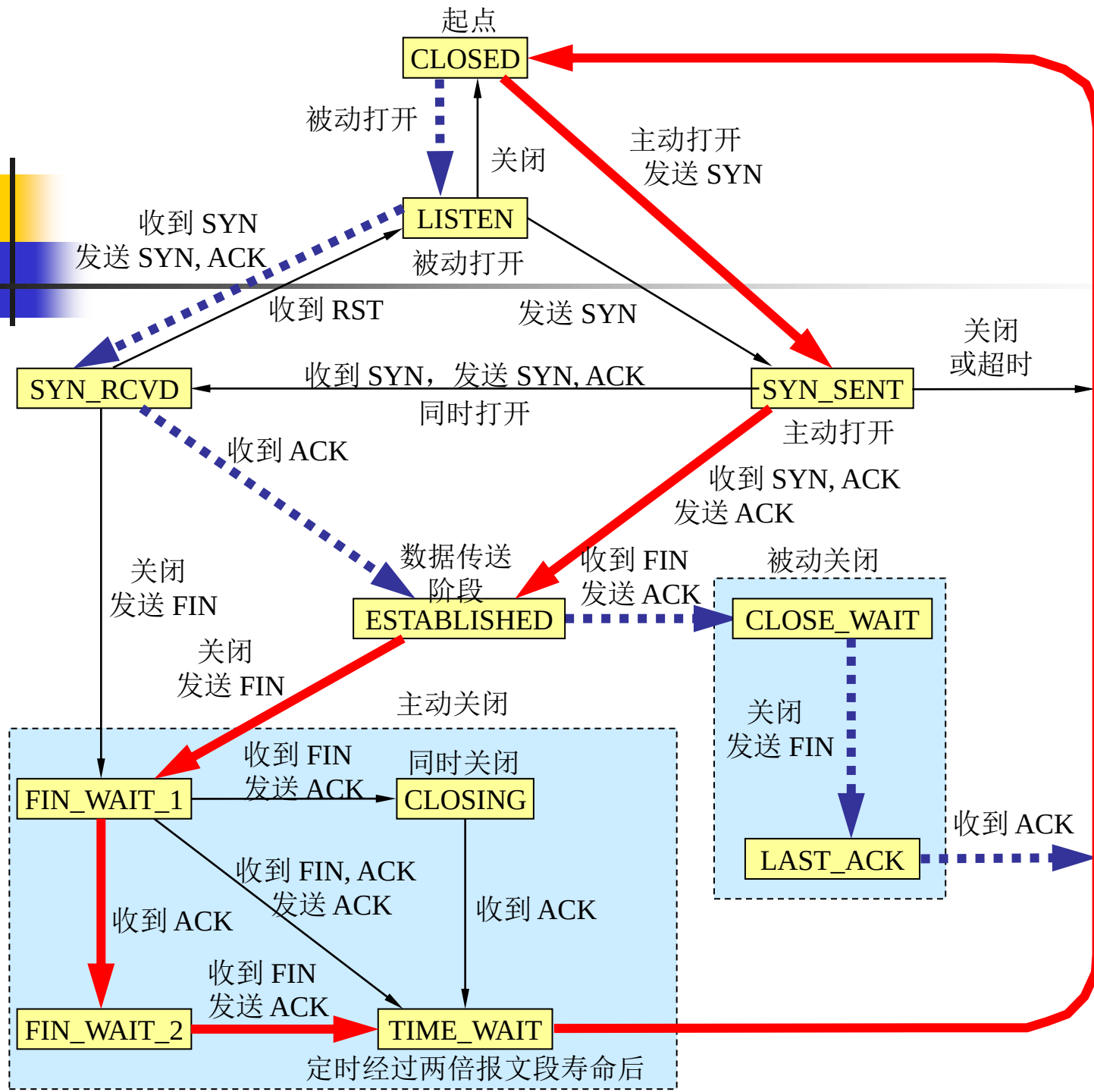
TCP采用对称释放法释放连接

- TCP的**全双工**连接可看成一个**双单工**的连接，每个单工连接都独立地释放。
- 通信双方必须都向对方发送**FIN=1**的**TCP**段并得到对方的应答，连接才能被释放，有四个阶段。
- 可以将第一个**ACK**数据段和第二个**FIN**数据段合并，从而变为三步握手。
- 为防止半连接，必须使用定时器计时，对**FIN**数据段的应答在两个最大分组生命期内未到达，就释放连接。对方也会超时释放。



主机发送**FIN**数据段后，就不再向对方发送数据

TCP 的有限状态机





TCP释放连接相关定时器

- **保持存活定时器(Keep-Alive Timer):** 当一个连接长时间闲置时，保持存活定时器会超时而使一方去检测另一方是否仍然存在，如果它未得到响应，便终止该连接。
- **闲置定时器(Quiet Timer):** 当TCP连接断开后，为防止该连接上的数据段还在网络上，并被后续打开的相同五元组的连接接收，要设置闲置定时器以防止刚刚断开连接的端口号被立即重新使用。

<136.6.23.13:1500, 130.42.85.15:25, TCP>



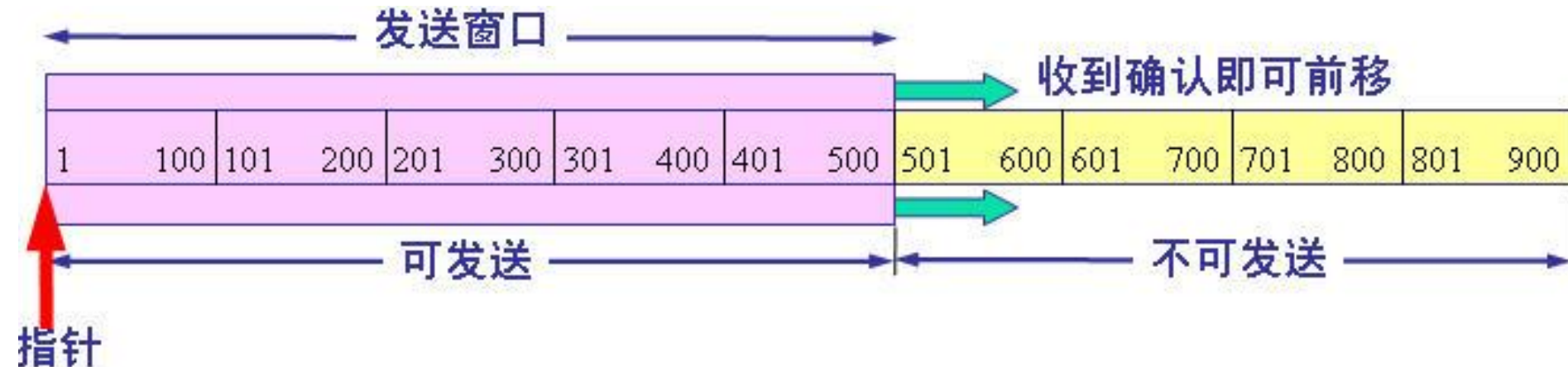
TCP 滑动窗口

Sliding Window

- 可靠有序的传输
 - 使用发送缓存和接收缓存，只有被确认，窗口才滑动
- 流量控制
 - 接收方可根据可用剩余缓冲区来指定窗口大小，并通知发送端，发送端根据窗口大小来设置发送窗口大小
 - 当缓冲区满，接收方可以发送一个窗口大小为0的数据段，指示发送方停止发送数据。但此时仍可以发送紧急数据（如用户紧急停止进程）和一字节的通知对方更新窗口大小的数据段。

TCP滑动窗口举例

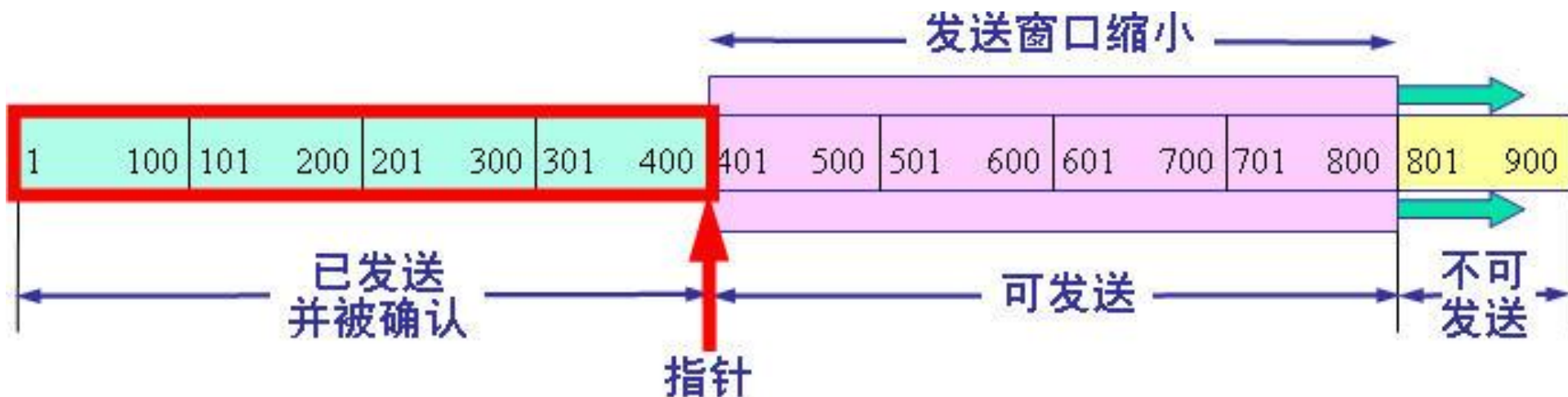
- 发送端要发送 **900** 字节长的数据，划分为 **9** 个 **100** 字节长的数据段(MSS)，而发送窗口确定为 **500** 字节
- 发送端只要收到了对方的确认，发送窗口就可前移
- 发送 **TCP** 要维护一个指针。每发送一个数据段，指针就向前移动一个数据段的距离



- 发送端已发送了 **400** 字节的数据，但只收到对前 **200** 字节数据的确认，同时窗口大小不变。
- 现在发送端还可发送 **300** 字节



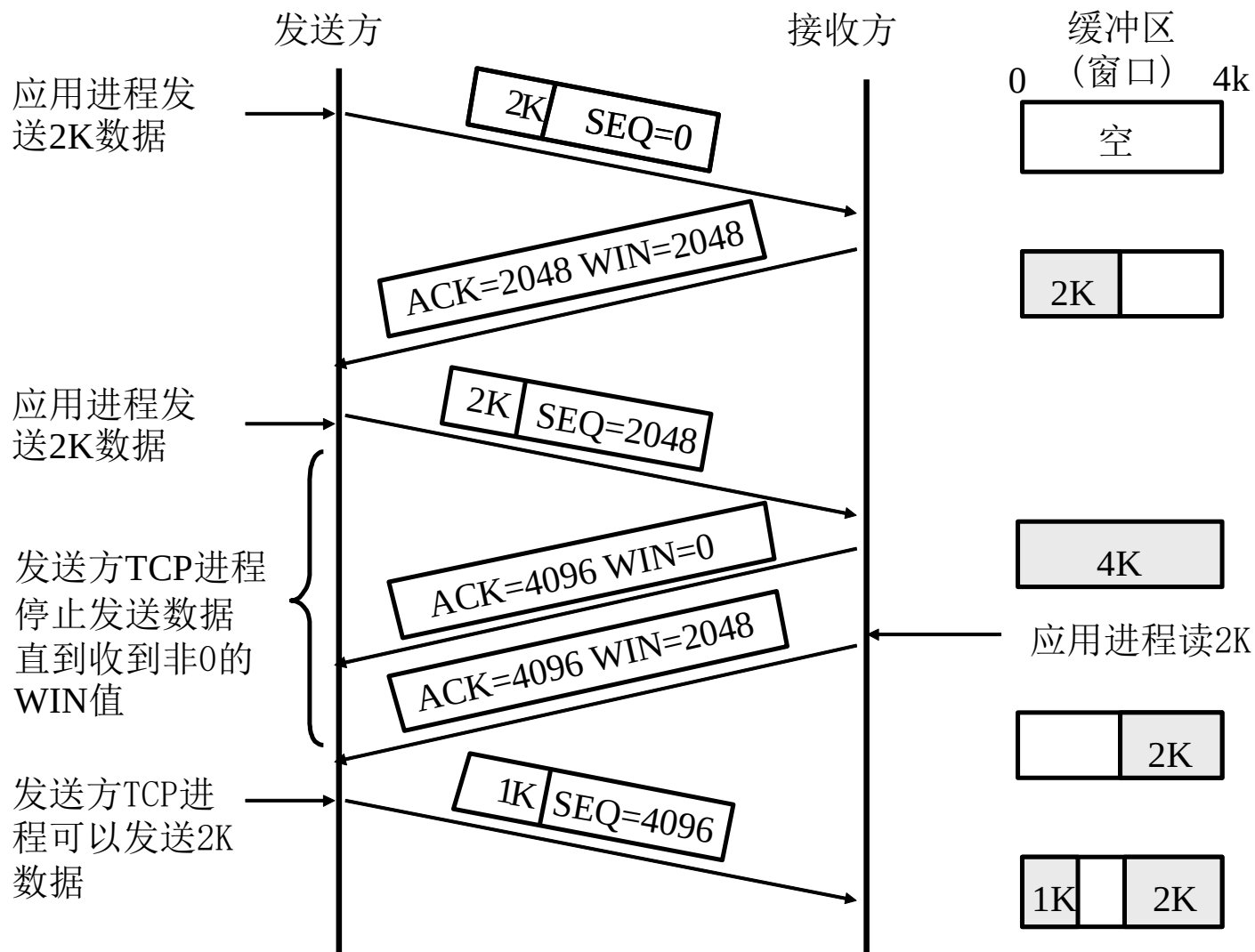
- 发送端收到了对方对前 **400** 字节数据的确认，但对方通知发送端必须把窗口减小到 **400** 字节。
- 现在发送端最多还可发送 **400** 字节的数据。



我们可以注意到，主机B进行了三次流量控制，第一将窗口减小为300字节，第二次又较少为200字节，最后减少到0，既不允许对方发送数据。直到主机B重新发出一个新的窗口值为止。



接收方缓存区管理





傻瓜窗口症状

Silly Window Syndrome

- 以下情况被称为傻瓜窗口症状：
 - 当发送方的**TCP**每次接收到来自应用的一字节的数据后就发送
 - 当接收方的**TCP**缓冲区已满，接收方会向发送方发送窗口大小为**0**的数据，而此时接收方的应用进程以交互方式每次只读取一个字节，于是接收方又发送窗口大小为一个字节的更新数据段，发送方应邀发送一个字节的数据，于是窗口又满了，循环往复...
- 解决方法：
 - 禁止发送方发送太小的数据段，而是等到有一定数量的数据后再发送（**Nagle**算法）
 - 禁止接收方发送**1**个字节大小的窗口更新信息，而是要等到有了一定数量的可用空间后再通知对方（**Clark**算法）



持续定时器

The Persistence Timer

- 持续定时器管理的是一种较为少见的事件，即下面要介绍的死锁情况。为了让发送方暂停发送数据，接收方发送一个接收窗口为**0**的确认。后来，接收方又发送了一个更新了窗口大小的数据段，但该数据段丢失，于是，双方都处于等待。
- 为了防止上述事情发生，发送方在收到接收方发来一个窗口为**0**的数据段时，就启动持续定时器，等该定时器超时还没有收到对方修改窗口大小的数据段的话，发送方就发一个**1**字节的探测数据段，对该探测数据段的响应应包含了窗口大小，若仍为**0**，则定时器清**0**，否则则可以发送数据。

-





问题：超时时间间隔（ **timeout** ）应设为多长呢？

- 不适当的 **timeout** 会导致性能下降：
 - **timeout** 太长 - 导致发送端的等待时间太长
 - **Timeout** 太短 – 重发多余的不必要的数据
- 解决的方法是对网络的性能不断测试，采用一种不断调整超时时间间隔的动态算法

Picking a Timeout Value

- 对每条连接，**TCP**均保存变量**RTT**，用于存放到目的端的往返时间的最接近估计值。
- 当一个数据发送时，同时启动定时器，如果时间超时就重发数据段，如果在超时之前得到确认，**TCP**就测量所花费的时间，记为**M**，并根据下面公式修正**RTT**
 - $RTT = \alpha RTT + (1 - \alpha)M$
 - α 是修正因子，一般为7/8
- 1988年**Jacobson**提出了另一个公式
 - $D = \alpha D + (1 - \alpha)|RTT - M|$
这里的 α 可能不同前面的 α
 - 确定当前的超时间隔**Timeout** $= RTT + 4D$
- 对已经重发的数据段无需修改**RTT**，而是在每次传输失败时将超时时间加倍，直到该数据段被成功传输——**Karn**算法。



Timeout设置

对每条连接，**TCP**均保存变量**RTT**、偏差变量**D**

Jacobson算法

- **RTT**: 往返时间的估计值
- **M**: 往返时间的测量值
- α , α' : 修正因子
- **D**: 偏差变量

第一次测量时，**RTT**设置为**M**，**D**取为**M**的一半，
以后每测量到一个**M**，按下面公式更新

$$\mathbf{RTT_{new}} \leftarrow \alpha \mathbf{RTT} + (1 - \alpha) \mathbf{M}$$

$$\mathbf{D_{new}} \leftarrow \alpha' \mathbf{D} + (1 - \alpha') |\mathbf{RTT} - \mathbf{M}|$$

$$\mathbf{RTT} \leftarrow \mathbf{RTT_{new}}$$

$$\mathbf{D} \leftarrow \mathbf{D_{new}}$$

$$\mathbf{Timeout} \leftarrow \mathbf{RTT} + 4\mathbf{D}$$



ACK延时定时器

Delayed ACK Timer

- 当**TCP**实体收到数据段时它必须返回确认，但并不需要立即回复，它可以在（**200ms?**）内发送**ACK**数据段，如果在这段时间内它恰好有数据段要发送，它就可以在数据段内包含确认信息，因此需要**ACK**延时定时器。



选择确认

SACK: Selective ACK

- 接收端收到序列号不连续的数据段，原因是某些数据段可能被丢失
- 如果这些数据段的序列号都在接收窗口之内，那么接收端就先收下这些数据，但要把这些信息准确地告诉发送端，使发送端不要再重复发送这些已收到的数据段

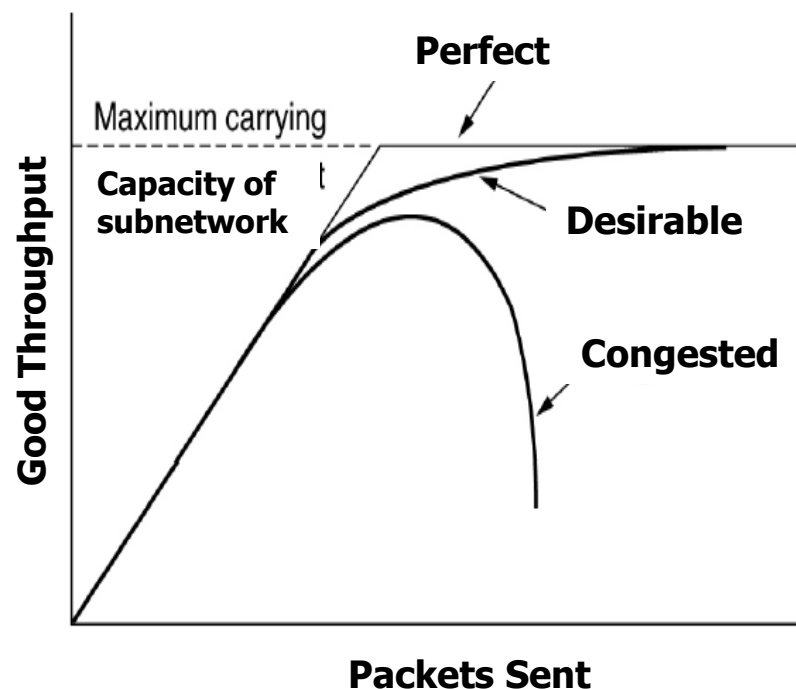


TCP流量控制与拥塞控制

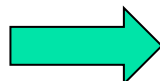
- 发送端的主机在确定发送数据段的速率时，既要根据接收端的接收能力，又要从全局考虑不要使网络发生拥塞。
- **TCP**的流量控制防止了发送方过快地传输数据使得接收方来不及处理的情形，但是网络容量有限，这时必须通过拥塞控制来防治网络过载。

网络拥塞

- 当网络中出现太多的分组，超过路由器的处理能力时，就会产生网络拥塞（**Congestion**）
 - 输出带宽、存储、处理器受限
 - 业务负载分布不均衡
- 网络拥塞会导致性能下降
 - 分组在路由器上的排队延迟和丢弃概率增大
 - 源主机大量重传分组
 - 网络有效吞吐量下降（**Good Throughput**）



TCP如何判断网络拥塞？



发送端出现数据段超时重传！



接收端窗口和拥塞窗口

- 每一个 **TCP** 连接需要维护两个窗口
 - 接收端窗口 **rwnd (receiver window)** 又称为通知窗口(**advertised window**)
 - 拥塞窗口 **cwnd (congestion window)**
- 接收端窗口 **rwnd** 是接收端根据其目前的可用接收缓存大小所许诺的最新的窗口值，是来自接收端的流量控制。接收端将此窗口值放在 **TCP** 数据段头标中的窗口大小域传送给发送端。
- 拥塞窗口 **cwnd** 是发送端根据自己估计的网络拥塞程度而设置的窗口值



发送窗口

- 发送窗口的上限值：
 - 发送端的发送窗口的上限值应当取为接收端窗口 **rwnd** 和拥塞窗口 **cwnd** 这两个变量中较小的一个，即应按以下公式确定：

发送窗口的上限值 = $\text{Min}[\text{rwnd}, \text{cwnd}]$

- 当 **rwnd** < **cwnd** 时，是接收端的接收能力限制发送窗口的最大值。
- 当 **cwnd** < **rwnd** 时，则是网络的拥塞限制发送窗口的最大值。



TCP拥塞控制算法

- 慢启动(**slow-start**)
- 拥塞避免(**congestion avoidance**)



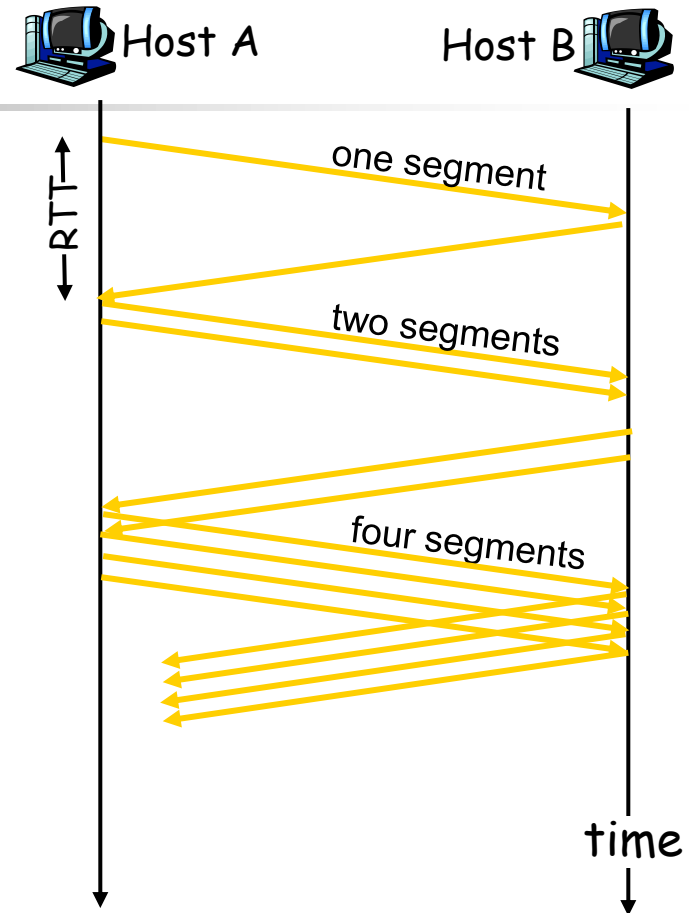
慢启动算法

- 在连接建立初期，拥塞窗口初始化为该连接最大数据段的长度
- 发送端发送一个最大数据段，得到确认后，其拥塞窗口大小加倍，依次类推，直到发送端认为出现数据段丢失或已达到接收端窗口大小（兼顾拥塞和流量控制）

Slowstart algorithm

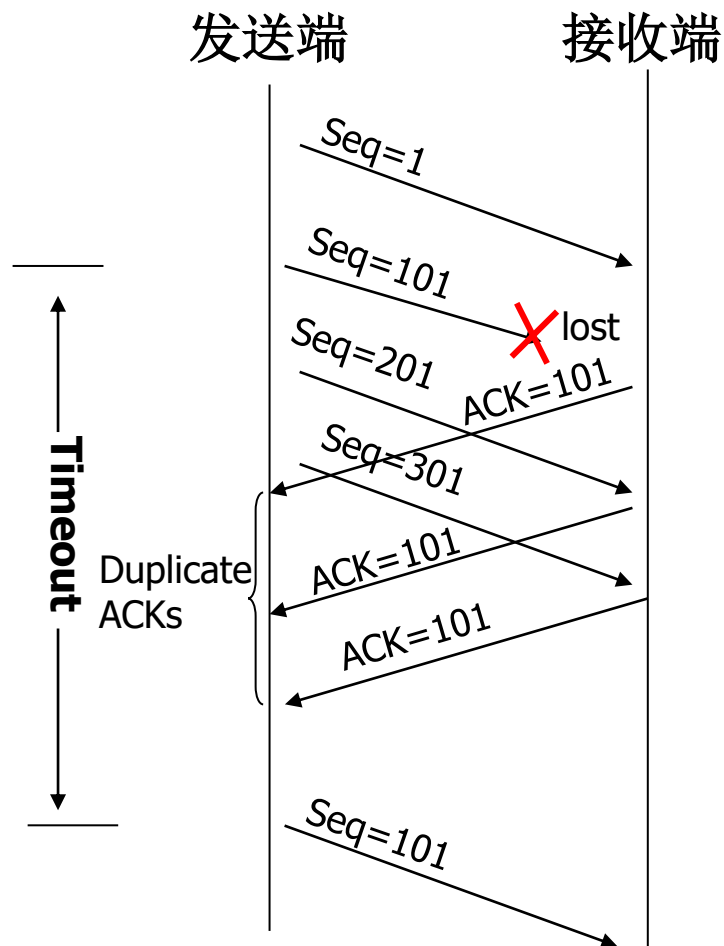
```
initialize: cwnd= 1
for (each segment ACKed)
    cwnd += cwnd
until (loss event OR
      cwnd > threshold)
```

- **exponential increase in window size (not so slow!)**
- **loss event: timeout and/or three duplicate ACKs**



Duplicate ACKs: Fast Retransmission

TCP数据段长度为
100字节，发送窗
口大小为**400**字节





TCP拥塞控制算法过程

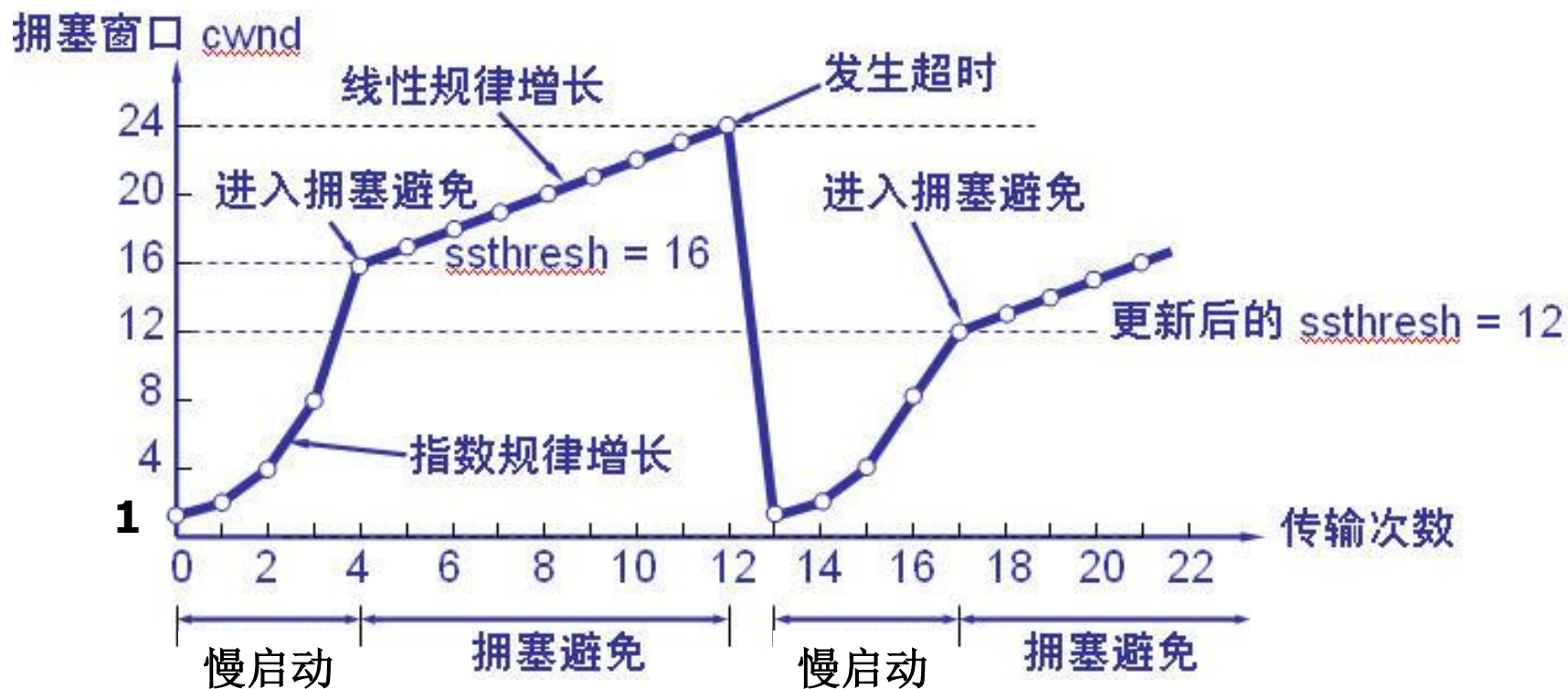
慢启动

- 当一个连接初始化时，将拥塞窗口置为一个最大数据段长度，并设置慢启动阈值**ssthresh**。
- 发送端的发送窗口不能超过拥塞窗口和接收窗口中的最小值，并假定接收端不进行流量控制。
- 发送端若收到了对所有发出的数据段的确认，就在下一次发送时将拥塞窗口加倍。可见拥塞窗口从**1**开始按指数规律增长

拥塞避免

- 拥塞窗口增长到**ssthresh**时，就每次将拥塞窗口加**1**，使拥塞窗口按线性规律增长
- 如果出现超时，就将当时拥塞窗口值减半，作为新的**ssthresh**，同时将拥塞窗口变为**1**
- 重复上述过程

TCP拥塞控制算法实例

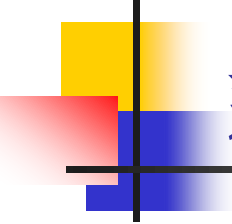


TCP Tahoe



加法增大additive increase

- “加法增大”是指执行拥塞避免算法后，在收到对所有数据段的确认后（即经过一个往返时间），就把拥塞窗口增加一个 **MSS** 大小，使拥塞窗口缓慢增大，以防止网络过早出现拥塞



乘法减小 **multiplicative decrease**

- “乘法减小”是指不论在慢启动阶段还是拥塞避免阶段，只要出现一次超时或者认为发送的数据段出现丢失时（即出现一次网络拥塞），就把慢启动阈值 **ssthresh** 设置为当前的拥塞窗口乘以 **0.5**
- 当网络频繁出现拥塞时，**ssthresh** 就下降得很快，以大大减少注入到网络中的数据量

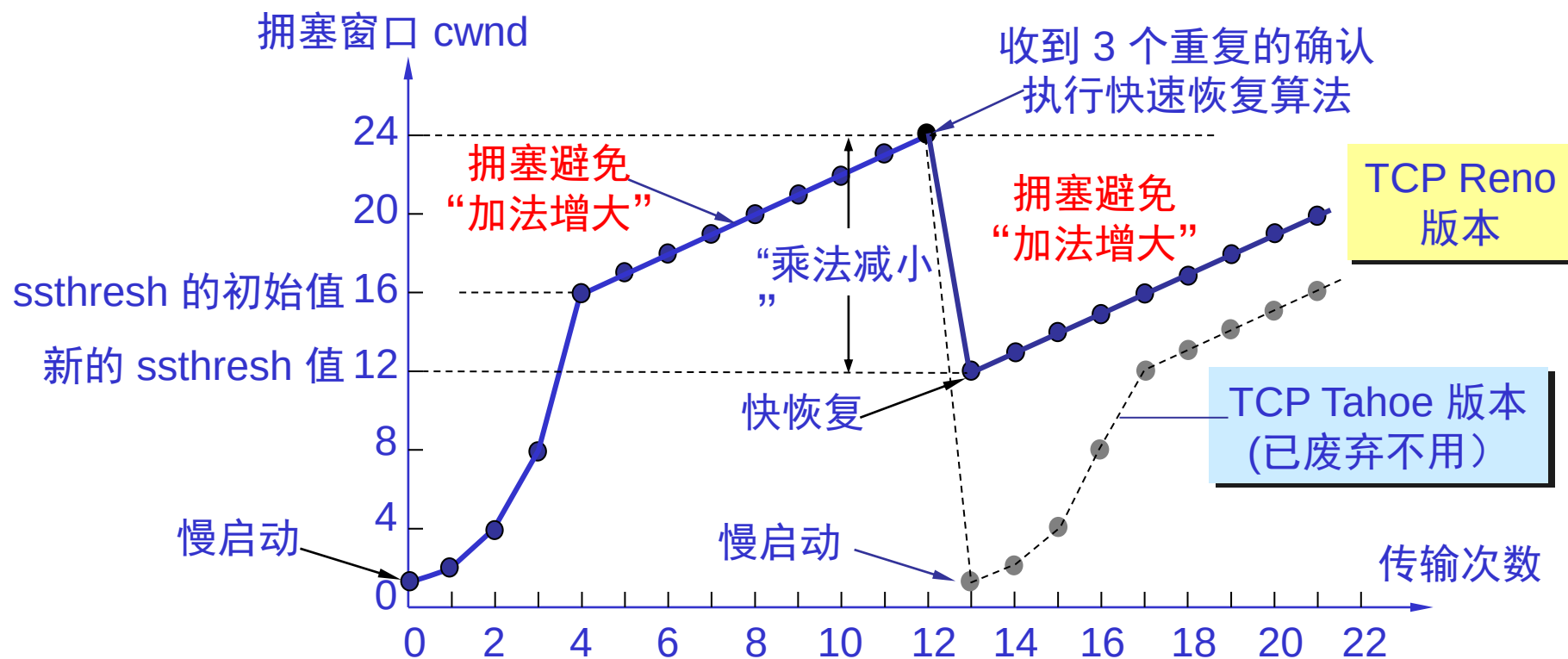
AIMD: Additive Increase, Multiplicative Decrease



快速重传和快速恢复算法

- （快速重传）当发送端收到连续三个重复的确认时，即可断定有分组丢失，就应立即重传丢失的报文段而不必等待该报文段的重传计时器超时
- （快速恢复）当判定有丢包时，执行“乘法减小”算法，把 **ssthresh** 设置为当前拥塞窗口值减半。发送端把拥塞窗口直接设置为新的慢启动阈值 **ssthresh**，然后开始执行拥塞避免算法（“加法增大”），使拥塞窗口缓慢地线性增大。

TCP Reno算法



Linux: CUBIC TCP (2008) Windows: Compound TCP (2006)



Chapter 8 传输层TCP & UDP

- **8.1**传输层服务
- **8.2**传输层寻址
- **8.3**建立连接
- **8.4Internet**中的传输层协议
 - 用户数据报协议**UDP**
 - 传输控制协议**TCP**
- **8.5Berkeley Socket**



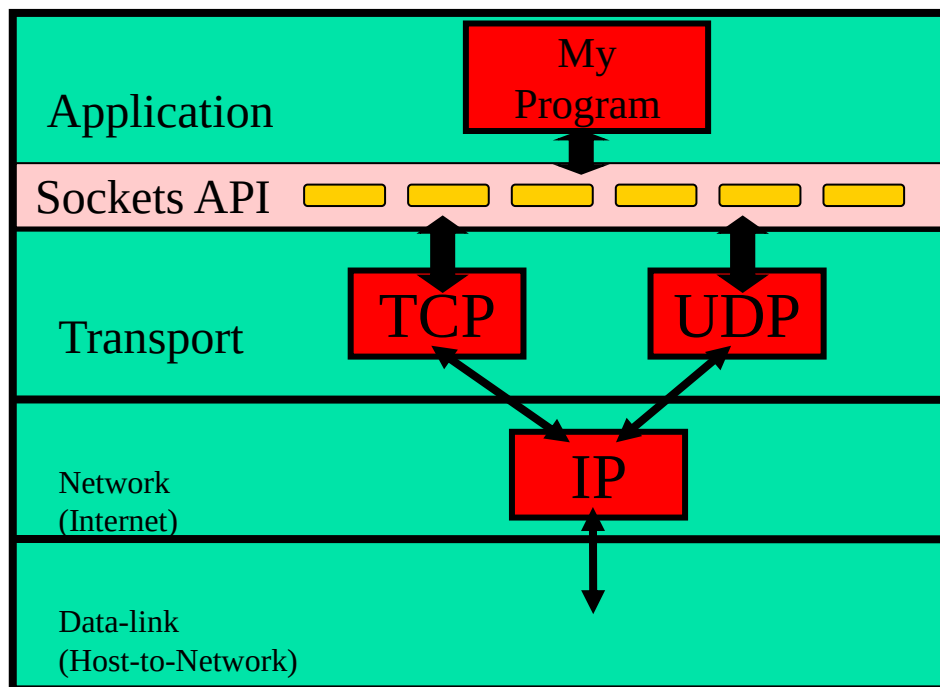
8.5 Berkeley Socket

- **Socket的产生和发展**
 - 最初在**70**年代由加州大学**Berkeley**分校开发，其目的是为**BSD(Berkeley Software Distribution)** **UNIX 4.1**版操作系统提供网络通信接口。
- **Socket在各种平台下的发展**
 - 随着**Berkeley Sockets**的广泛应用，九十年代初，**Sun**、**MS**等公司共同制定了适应**dos**和**win**平台的**windows sockets**的规范(**WinSock**)
 - **Sun Microsystems**为**Java**也制定了网络通信的**API**
 - **Linux**下的**socket**继承了**BSD sockets**的风格，但有所改动

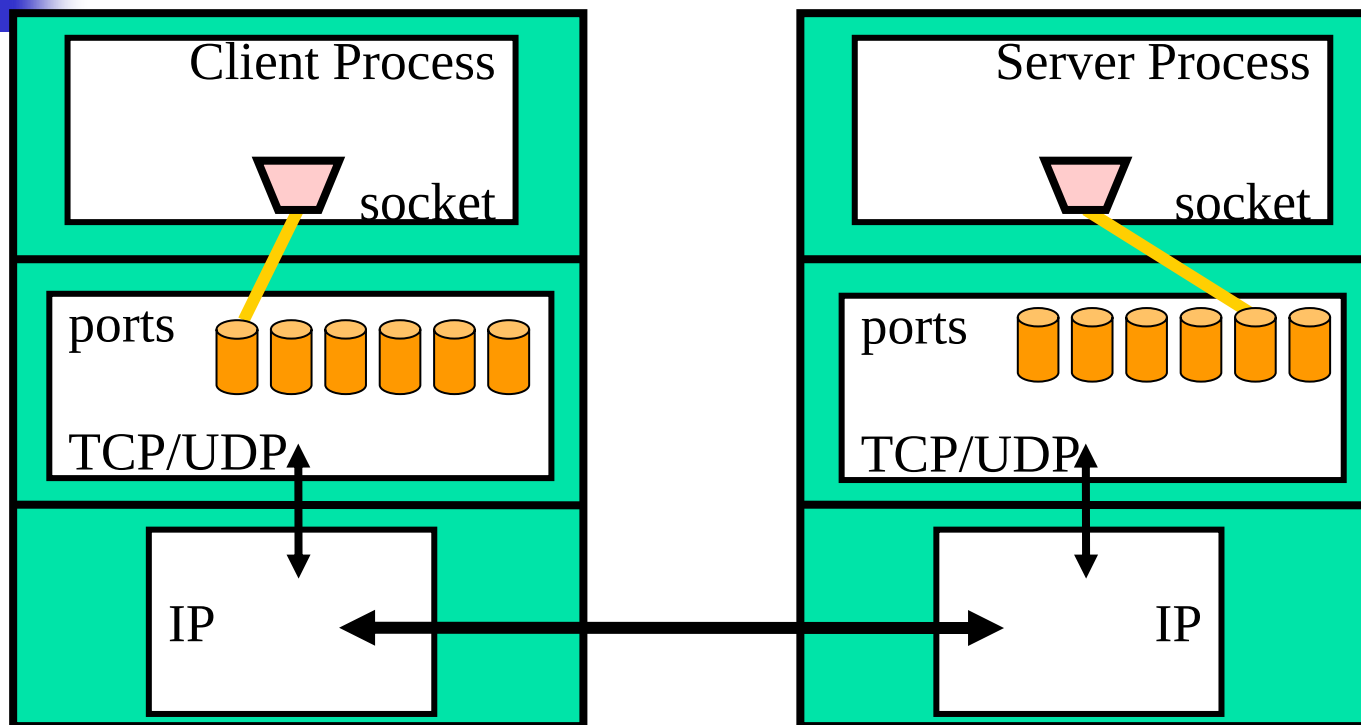
Socket原语

原语	含义
SOCKET	创建一个新的通信端点
BIND	将一个本地地址关联到一个套接字上
LISTEN	宣布愿意接收连接，给出队列大小
ACCEPT	阻塞调用方，直到有人企图连接上来
CONNECT	主动尝试建立一个连接
SEND	在指定的连接上发送数据
RECV	从指定的连接上接收数据
CLOSE	释放指定的连接

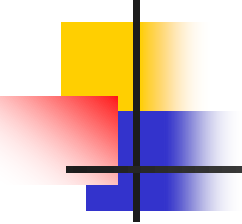
TCP套接字原语



Socket到Socket的通信

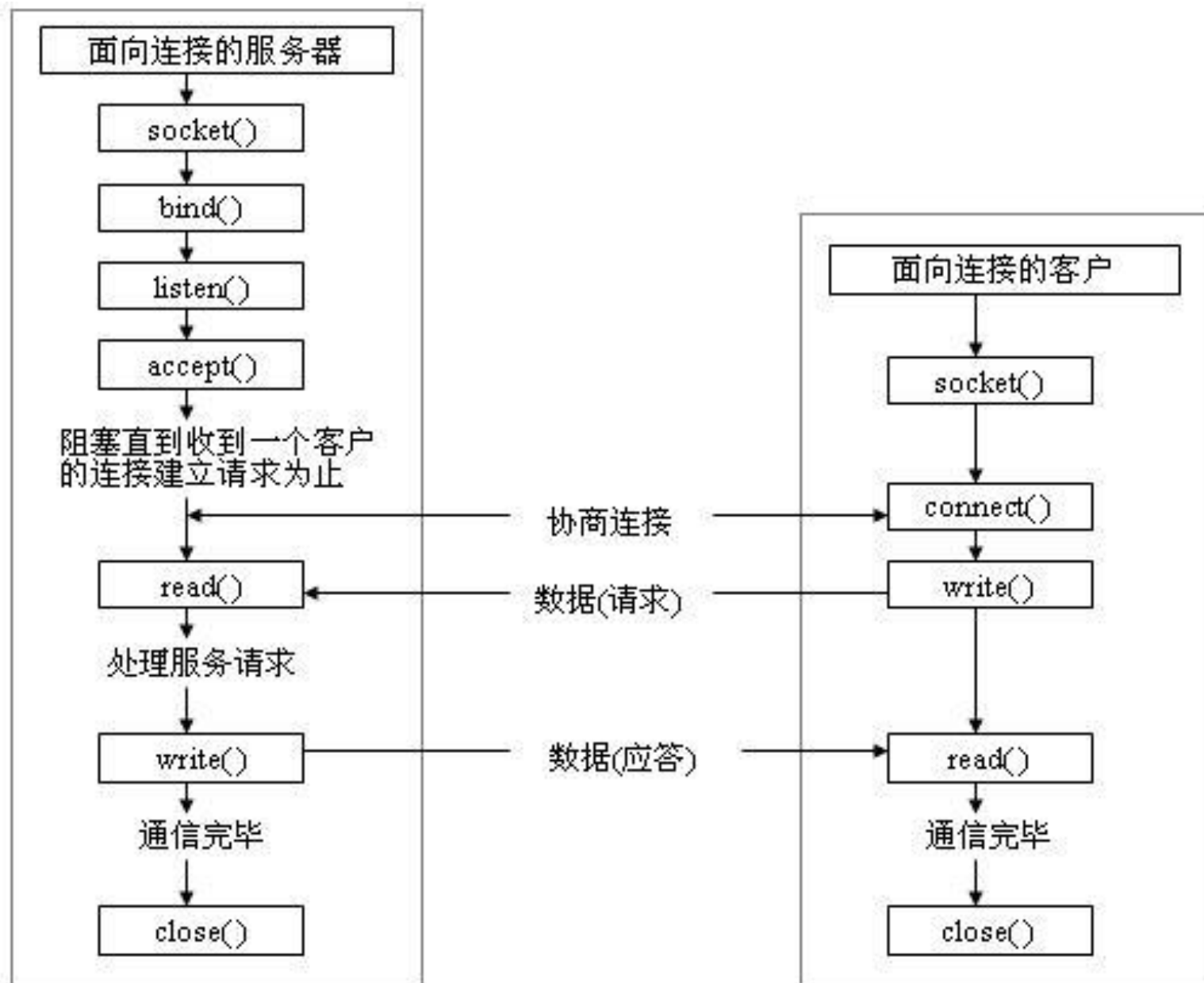


在**TCP/IP**网络应用中，两个进程间的**Socket**通信的主要采用客户/服务器（**Client/Server, C/S**）模式，即客户向服务器发出服务请求，服务器接收到请求后，提供相应的服务

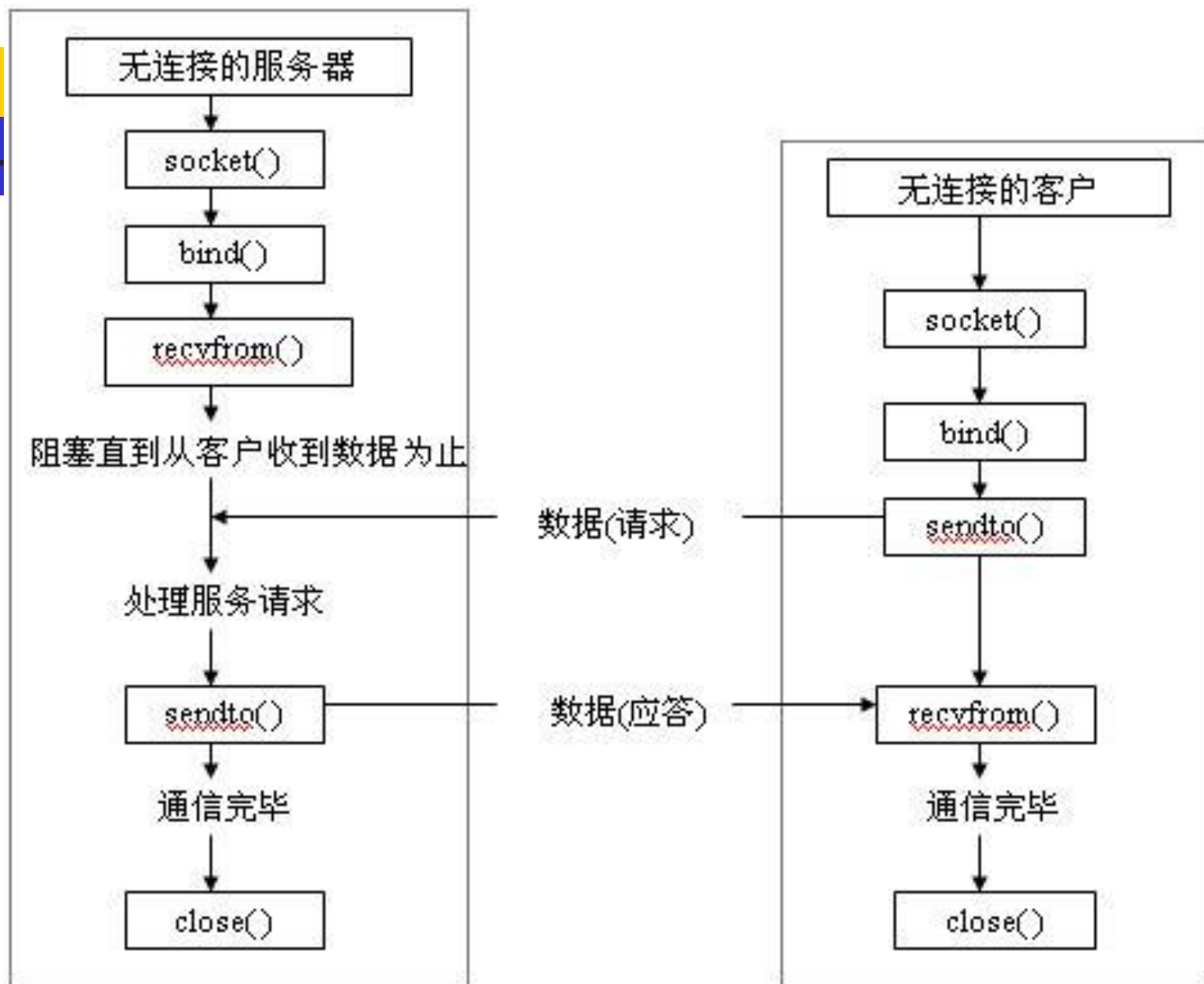
- 
- 从上面所描述过程可知：
 - 客户与服务器进程的作用是非对称的，因此编码不同。
 - 服务进程一般是先于客户请求而启动的。只要系统运行，该服务进程一直存在，直到正常或强迫终止。

客户需要知道服务器使用的**IP**地址、端口等信息

TCP Client-Server交互流程



UDP Client-Server交互流程





小结

- 掌握传输层基本功能，特别是**Internet**的传输层协议**TCP**和**UDP**，以及连接建立机制、流量控制机制、拥塞控制机制。
- 基于**Socket**的客户 / 服务器编程
- 进一步熟悉**TCP**有限状态机的迁移过程，注意其中的不常见过程
- 作业
 - 请比较数据链路层和**TCP**的滑动窗口机制有什么不同，并说明为什么会有这些不同。
 - 书（第五版）：**19、23、26、30-35**



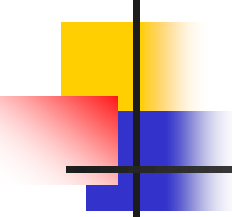
Chapter 9 应用层

- 9.1网络的计算和访问模式
- 9.2域名系统(DNS)
- 9.3文件服务(FTP)
- 9.4WWW(HTTP)
- 9.5电子邮件(SMTP)



Chapter 9 应用层

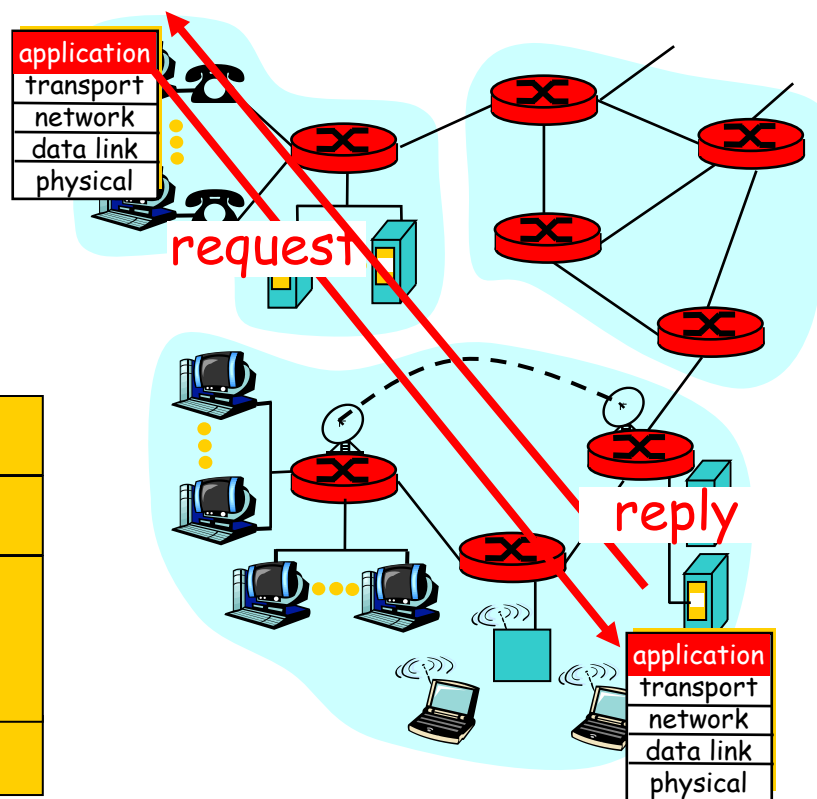
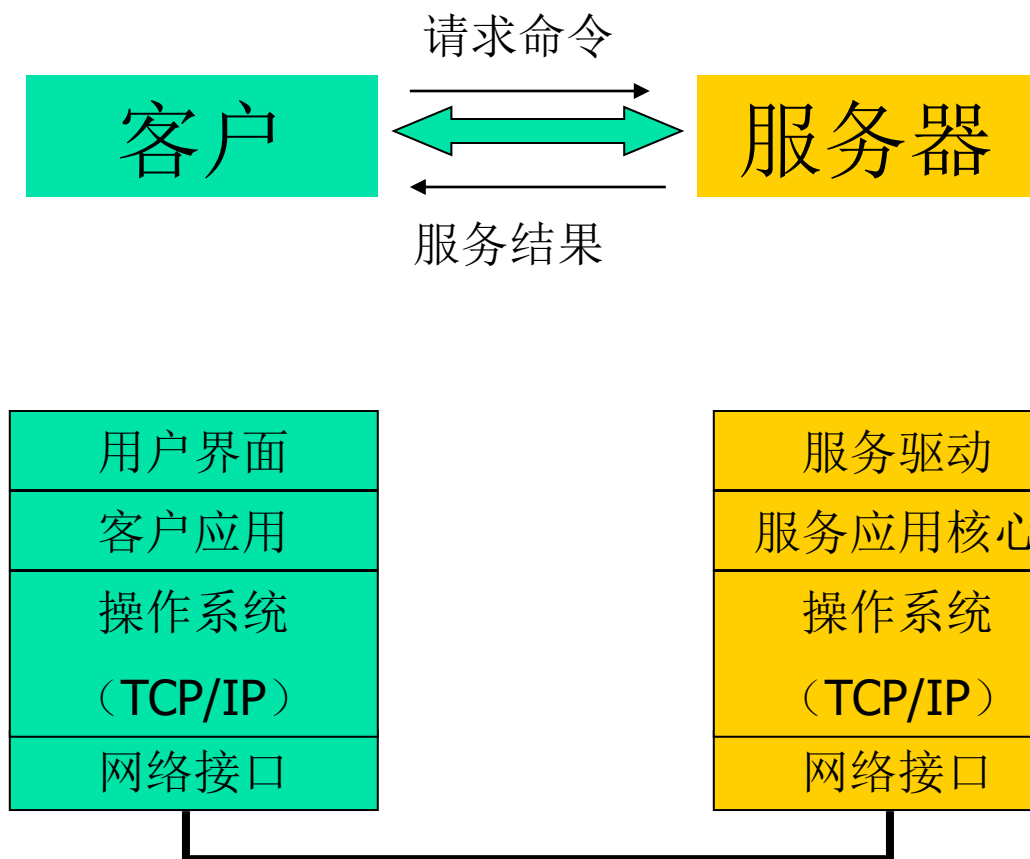
- **9.1网络的计算和访问模式**
- 9.2域名系统(DNS)
- 9.3文件服务(FTP)
- 9.4WWW(HTTP)
- 9.5电子邮件(SMTP)



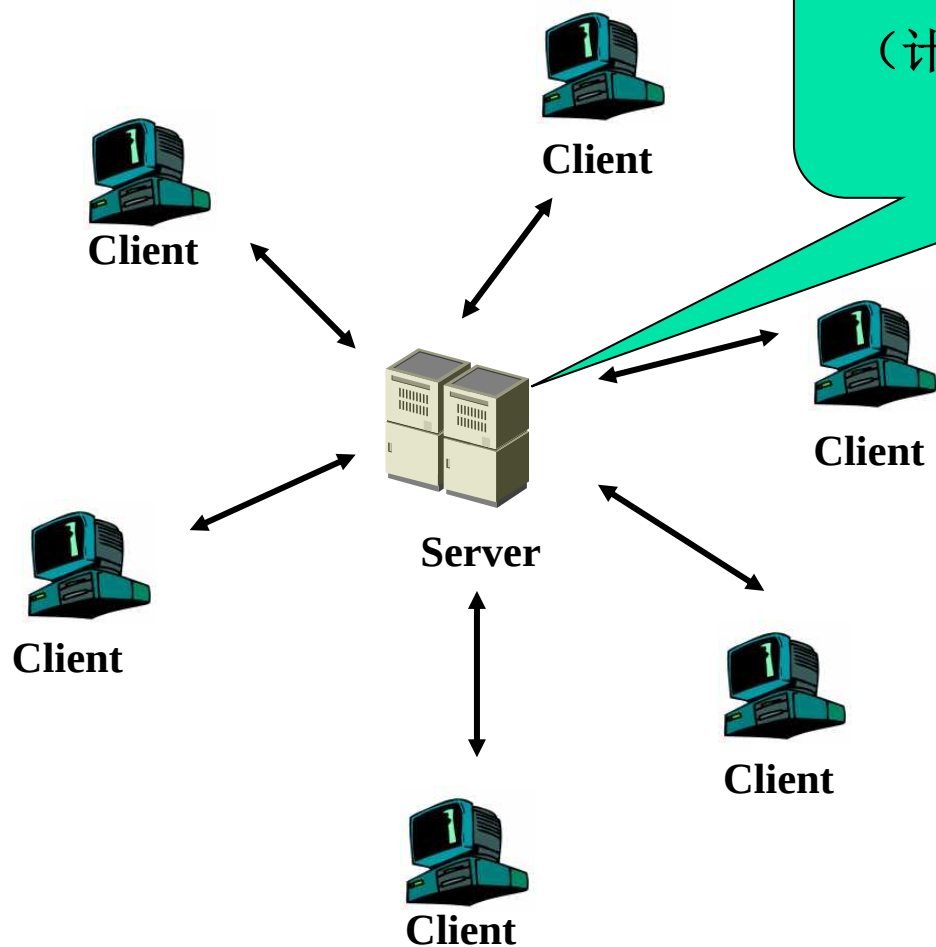
9.1 网络的计算和访问模式

- 以大型机为中心的计算模式-分时共享模式（time-sharing）
 - 特点：许多用户共享CPU资源和数据存储功能，终端只用于命令的输入和结果的显示
- 以服务器为中心的计算模式-资源共享（resource-sharing)模式
 - 特点：共享共同的应用，如文件服务器，打印服务器
- 客户端/服务器（Client/Server)模式
 - 特点：系统使用了客户和服务双方资源和计算能力来执行一个特定的任务。
 - 应用：前端（即客户部分）/后端（服务器部分）
- 对等（Peer to Peer, P2P）模式
 - 特点：用户和资源处于对等状态，分布式计算
- 云计算（cloud computing）模式
 - 以网络为中心的计算模式
 - 虚拟化的资源

客户/服务器模式 (Client/Server)



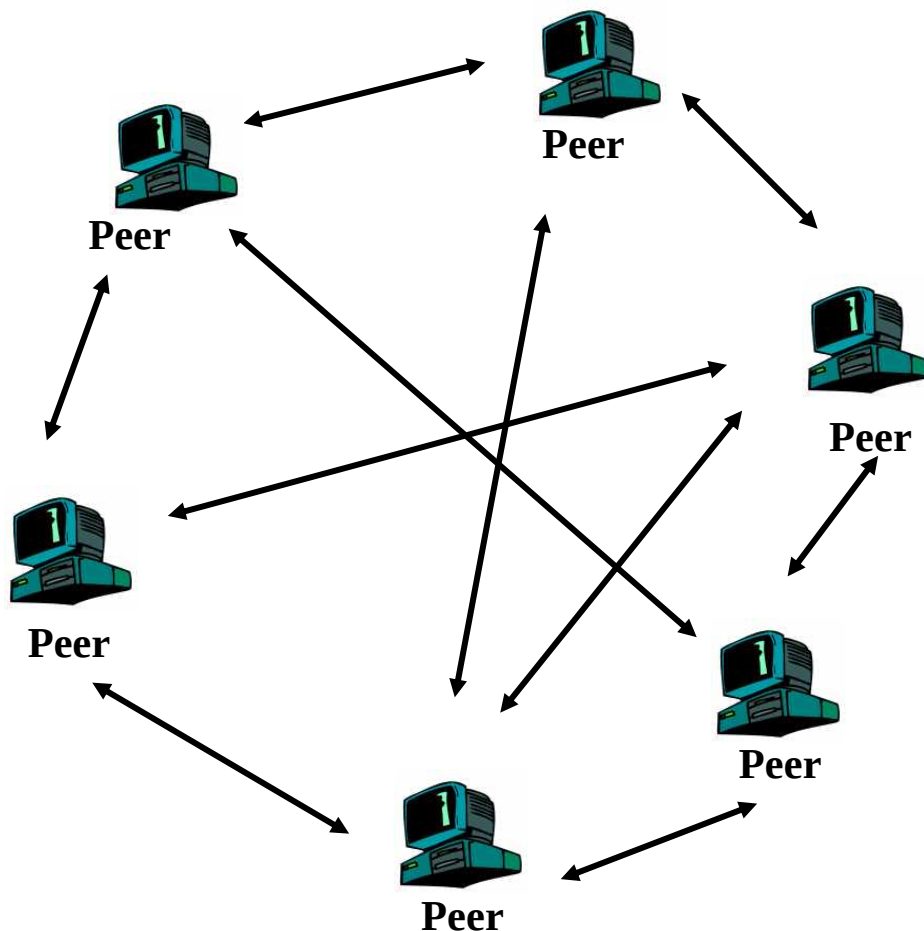
客户/服务器模式



单点失效
性能瓶颈
(计算、存储资源受限,
网络拥塞等)

对等模式

(Peer to Peer, P2P)



将服务器的功能分散到客户端，充分利用客户端的计算、存储、带宽等资源

无中心服务器
Peer既是客户端，
又是服务器



P2P的定义

- **P2P**通信模式中各方都具有相同的能力，其中任何一方都可以发起一个通信会话。在**P2P**通信过程中，每个通信节点同时具有服务器和客户端的功能。
- **P2P**网络中的节点间采用**P2P**通信模式，它是构筑在现有网络基础设施上的一个重叠网络（**Overlay Network**）



P2P连接资源的方式

目标：实现资源共享

过程：发布→定位→使用或者下载

- 基于目录服务器
 - 使用中心目录服务器用于资源发布和定位
 - SETI@home、Napster、Groove
- 完全分布式
 - 无任何中心服务器，资源的定位使用泛洪
 - Gnutella
- 层次结构
 - 将节点分为一般节点和超级节点，一般节点通过超级节点来发布和定位资源，超级节点之间采用泛洪方式来定位资源
 - KazaA、Skype
- 结构化P2P
 - 将资源和其存储位置关联起来
 - Chord、CAN、Tapestry、Pastry



P2P网络的特征

- **P2P**网络是一个应用层网络，一般由网络边缘节点构成，充分利用资源
- 网络的扩展性好
- 资源分布在各个节点中，而不是集中在一个服务器上进行管理，不存在单点瓶颈问题
- 节点之间可直接建立连接，交互共享资源

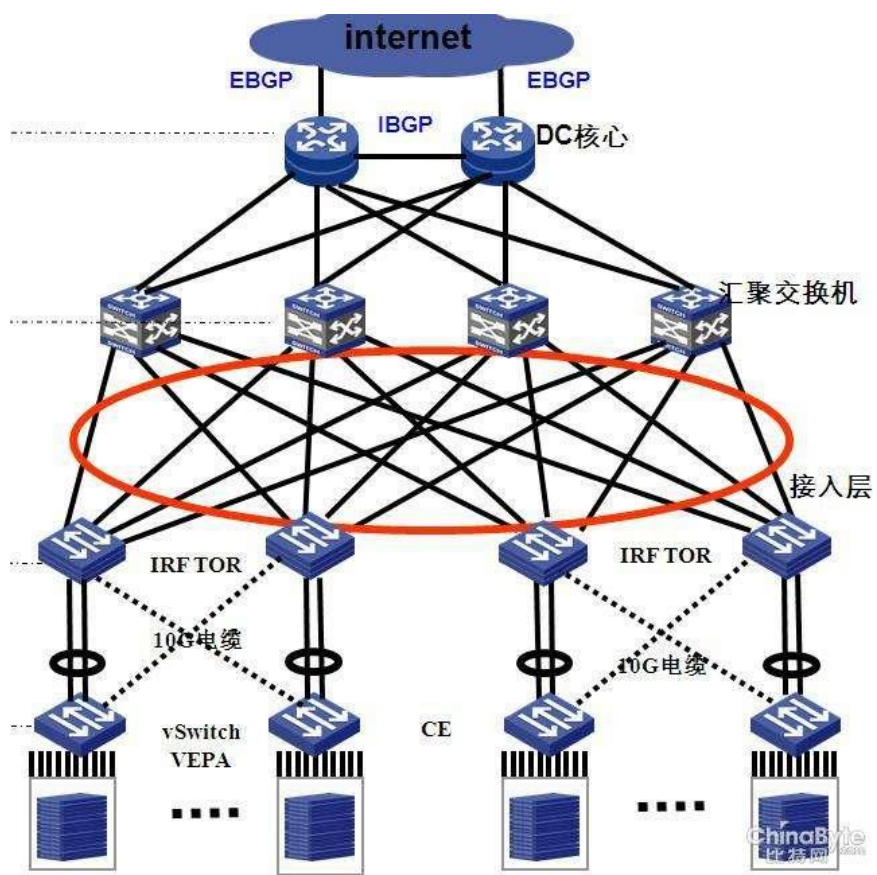
需要解决的问题：动态性、安全性、可管理性...



P2P应用技术的发展

- 从文件共享扩展到流媒体服务（实时流媒体、VoD等）-应用层组播
- 有线环境扩展到无线环境或者混合环境
- 使用P2P应用的终端复杂多样

云计算（数据中心）



与此相对应的有
——雾计算（边缘计算）



应用层需要什么样的传输层来提供服务呢？

可靠性

- 某些应用允许少量的包丢失，如话音。
- 某些应用需要**100%**的可靠传输，如**FTP**等。

带宽

- 某些 需要有一定量的带宽保证（如音、视频）
- 某些则对带宽没有特定的要求。

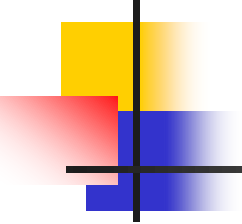
时延

- 某些应用对延时、延时抖动有严格要求。如实时多媒体业务。
- 某些则对延时要求比较宽松。如数据传输。



常用应用对传输的要求

应用	数据丢失	带宽	实时性
文件传输	不丢失	弹性	无
e-mail	不丢失	弹性	无
Web 网页	不丢失	弹性	无
实时音频/视频	允许丢失 允许丢失	音频: 5Kb-1Mb 视频:10Kb-5Mb	100's msec
存储音频/视频	允许丢失	同上	few secs
交互式游戏	允许丢失	几 Kb/s 以上	100's msec
金融应用	不丢失	弹性	yes and no



Internet 传输层提供的服务

TCP service:

- *connection-oriented*: setup required between client, server
- *reliable transport* between sending and receiving process
- *flow control*: sender won't overwhelm receiver
- *congestion control*: throttle sender when network overloaded
- *does not providing*: timing, minimum bandwidth guarantees

UDP service:

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

Q: why bother? Why is there a UDP?



Internet应用对应的传输层协议

应用	应用层协议	所依赖的传输层协议
e-mail	smtp [RFC 821]	TCP
远程登陆	telnet [RFC 854]	TCP
万维网(Web)	http [RFC 2068]	TCP
文件传输	ftp [RFC 959]	TCP
流媒体	专有协议 (e.g. RealNetworks)	TCP or UDP
远程文件服务器	NSF	TCP or UDP
IP电话	专有协议 (e.g., Vocaltec)	typically UDP



Chapter 9 应用层

- 9.1网络的计算和访问模式
- **9.2域名系统(DNS)**
- 9.3文件服务(FTP)
- 9.4WWW(HTTP)
- 9.5电子邮件(SMTP)

域名系统的重要性

全国顶级域名根服务器故障 大量网站无法访问

腾讯科技 [微博] 乐天 2014年01月21日16:35

[导读]有近三分之二的网站不同程度地出现不同地区、不同网络环境下的访问故障。



- 什么是域名？
 - 由字符和点号组成，成员名最长不超过**63**字符，全名不超过**255**字符
 - if.ustc.edu.cn, www.baidu.com, www.ustc.edu.cn, news.sina.com.cn,
- 域名系统（**DNS: Domain Name System**）负责域名的维护和管理
- 用户的绝大部分网络服务访问都需要使用**DNS**
 - **www**服务、文件传输服务、电子邮件服务等等

为什么要引入域名系统

我能高效地处理基于二进制的**IP地址**

我只能记住用字符串表示的**名字**

主机名

主机名

IP地址

hosts.txt



USTC WWW	202.38.54.246
AHU WWW	101.76.160.201
Baidu Baidu	119.75.219.42
...	...
...	...
...	...

名字 IP地址

可扩展性问题:

- 1) 数量庞大, 维护开销大, 容易成为性能瓶颈
- 2) 名字管理困难, 难以避免重名冲突

名字层次化
管理分布化

域名系统**DNS**
Domain Name
System



层次化名字空间

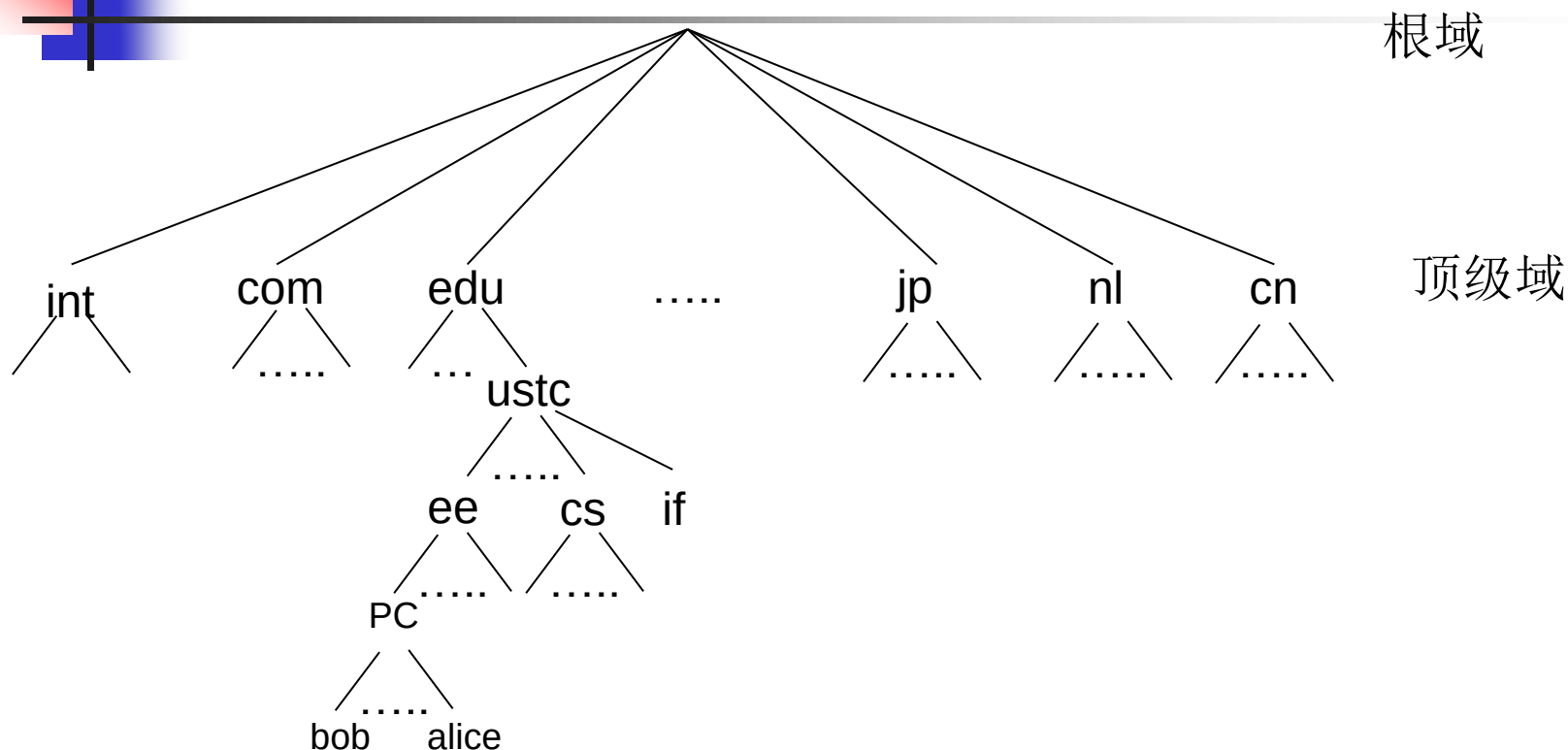
- 名字空间

- 所有可能的名字的集合

- **DNS**采用层次化的名字空间

- 每个层次都有多个名字，每个名字对应着一个域，这些名字也被称为域名
 - 域名由定义该层及其上层名字的字符串组成，不同层的字符串用 . 隔开，例如 **if.ustc.edu.cn**
 - 第1层
 - 第2层
 - 第3层
 - 第4层
 - 每个域都控制着如何分配直接在它下一层次的域，也称为该域的子域
 - 有多少个子域，以及每个子域的名字（域名）是什么

名字空间



- 1) 每个域控制着直接在它下一层次的域如何命名，确保这些域名不发生冲突
- 2) 中间节点对应着某个域，域名为从该节点向上到根节点的路径，而叶子节点的域名一般对应着某台主机的域名，例如**if.ustc.edu.cn**



常用顶级域名

com	Commercial organizations
edu	Educational institutions
gov	Government institutions
int	International organizations
mil	U.S. military institutions
net	Networking organizations
org	Non-profit organizations
Country code	A country

命名机制遵循的是组织边界，而不是物理网络的边界
例如：**www.ustc.edu.cn**

分布式域名管理

- **DNS**采用分布式的域名管理

- 以域为单位，在网络中分布多个服务器对域名进行管理，这些服务器也被称为名字服务器
- 某个域如果设置了名字服务器，则该服务器负责此域下的子域的域名的管理（有可能包括子域下层的域）

- **DNS**实现

- 采用客户/服务器通信模式，客户端执行域名查询功能的程序或者函数调用叫做解析器（**resolver**），解析器向名字服务器发送查询请求
- 传输层可采用**UDP**协议，端口号为**53**号
- **Unix/Linux**系统通过**BIND**（**Berkeley Internet Name Domain**）提供**DNS**服务



Name Server

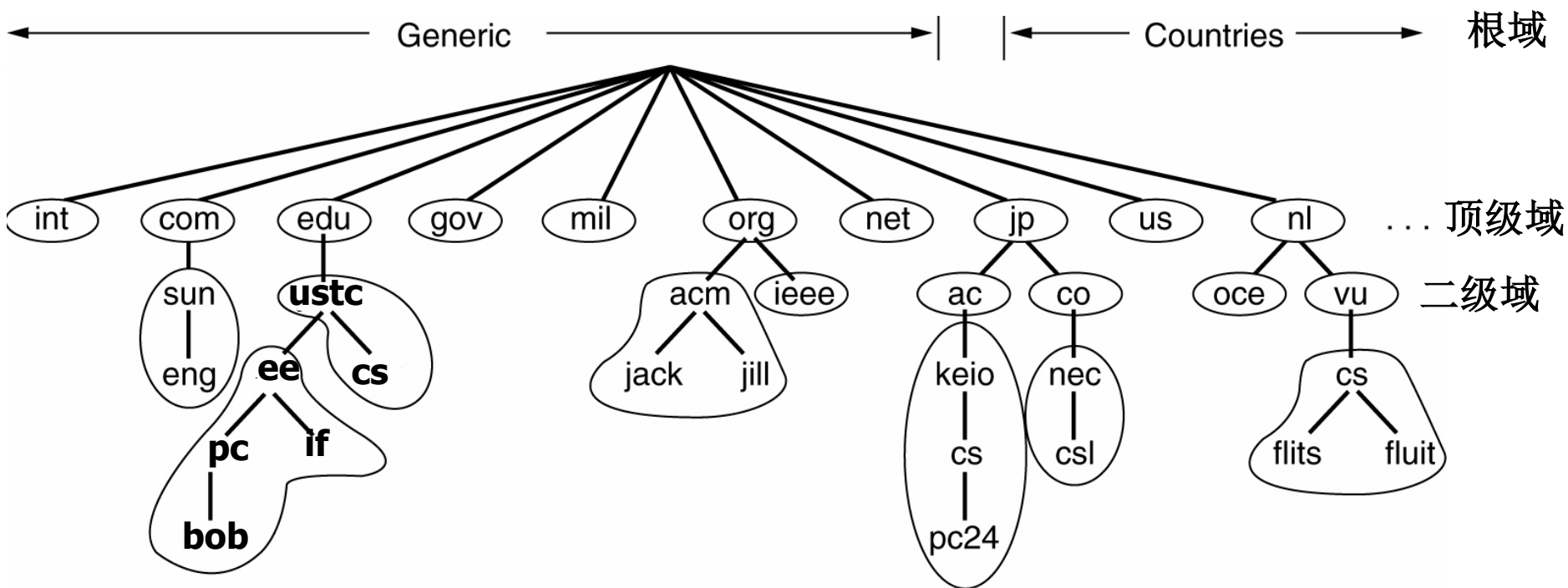
*mail.ustc.edu.cn*的IP地址是多少？

202.38.64.8



Resolver

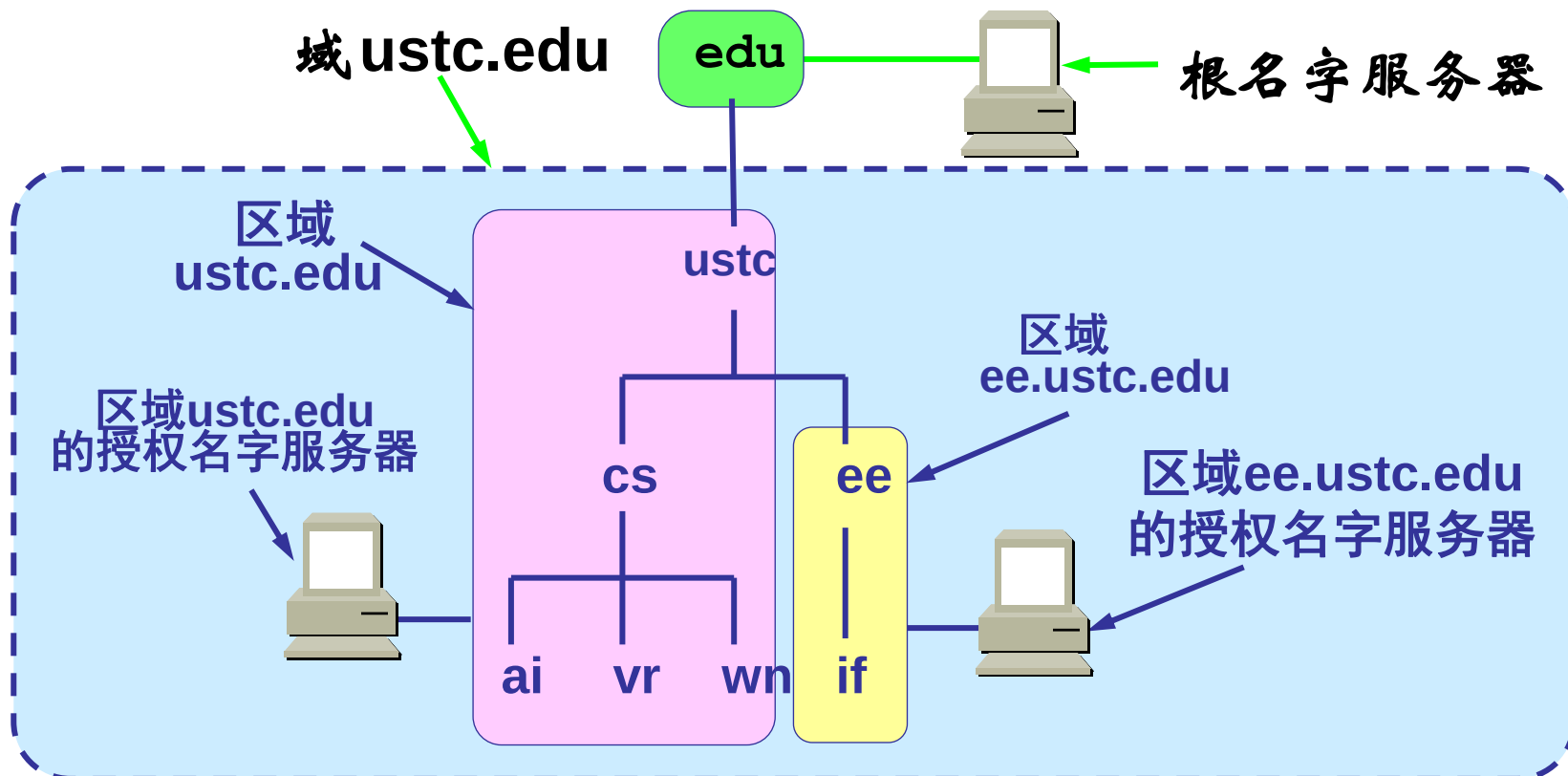
名字服务器



DNS名字空间被划分为一些不重叠的区域，每个区域的信息保存在该区域对应的名字服务器上，该服务器也被称为所在区域的授权名字服务器

授权名字服务器

- **Internet**允许各个组织根据具体情况将本组织内的域划分为若干个区域(**zone**), 并在各区域中设置相应的授权名字服务器



根名字服务

- 根名字服务保存了所有顶级域名字服务器的**IP**地址
- 全球共有**13**个根名字服务器，为确保可靠性和访问速度，每个服务器实际上是由多个服务器组成的集群

[https://www.internic.net/
domain/named.root](https://www.internic.net/domain/named.root)

```
; This file holds the information on root name servers needed to
; initialize cache of Internet domain name servers
; (e.g. reference this file in the "cache . <file>"
; configuration file of BIND domain name servers).
```

```
; This file is made available by InterNIC
; under anonymous FTP as
```

```
    file                /domain/named.cache
    on server           FTP. INTERNIC.NET
-OR-                    RS. INTERNIC.NET
```

```
last update:           November 11, 2020
related version of root zone:      2020111101
```

```
; FORMERLY NS. INTERNIC.NET
```

```
.
3600000      NS      A. ROOT-SERVERS.NET.
A. ROOT-SERVERS.NET. 3600000      A      198. 41. 0. 4
A. ROOT-SERVERS.NET. 3600000      AAAA   2001:503:ba3e::2:30
```

```
; FORMERLY NS1. ISI. EDU
```

```
.
3600000      NS      B. ROOT-SERVERS.NET.
B. ROOT-SERVERS.NET. 3600000      A      199. 9. 14. 201
B. ROOT-SERVERS.NET. 3600000      AAAA   2001:500:200::b
```

```
; FORMERLY C. PSI. NET
```

```
.
3600000      NS      C. ROOT-SERVERS.NET.
C. ROOT-SERVERS.NET. 3600000      A      192. 33. 4. 12
C. ROOT-SERVERS.NET. 3600000      AAAA   2001:500:2::c
```

```
; FORMERLY TERP. UMD. EDU
```

```
.
3600000      NS      D. ROOT-SERVERS.NET.
D. ROOT-SERVERS.NET. 3600000      A      199. 7. 91. 13
D. ROOT-SERVERS.NET. 3600000      AAAA   2001:500:2d::d
```

```
; FORMERLY NS. NASA. GOV
```

```
.
3600000      NS      E. ROOT-SERVERS.NET.
E. ROOT-SERVERS.NET. 3600000      A      192. 203. 230. 10
E. ROOT-SERVERS.NET. 3600000      AAAA   2001:500:a8::e
```

.....



资源记录格式

- 资源记录: **Resource Record**

- 名字服务器以资源记录的形式来维护本区域内的域名相关信息
- 当解析器给名字服务器一个域名, 取回的是一个与该域名相对应的资源记录。所以**DNS**系统的实际功能就是把域名映射到一条记录上

- 一条资源记录共有**5**个字段:

- **Domain_name, Time_to_live, Type, Class, Value**
- **Time_to_live:** 生存时间, 例如, 时间较长的有一天 (**86400**秒), 短的有一分钟 (**60**秒)
- **Type:** 指出记录的类型 (下一页详细解释)
- **Class:** 对**Internet**, 它总是**IN**
- **Value:** 可以是数字、域名或**ASCII**字符, 其语义基于记录类型

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

; Authoritative data for cs.vu.nl

cs.vu.nl.	86400	IN SOA	star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.	86400	IN TXT	"Faculteit Wiskunde en Informatica."
cs.vu.nl.	86400	IN TXT	"Vrije Universiteit Amsterdam."
cs.vu.nl.	86400	IN MX	1 zephyr.cs.vu.nl.
cs.vu.nl.	86400	IN MX	2 top.cs.vu.nl.

flits.cs.vu.nl.86400 IN HINFO Sun Unix

flits.cs.vu.nl.86400 IN A 130.37.16.112

flits.cs.vu.nl.86400 IN A 192.31.231.165

flits.cs.vu.nl.86400 IN MX 1 flits.cs.vu.nl.

flits.cs.vu.nl.86400 IN MX 2 zephyr.cs.vu.nl.

flits.cs.vu.nl.86400 IN MX 3 top.cs.vu.nl.

www.cs.vu.nl.86400 IN CNAMEstar.cs.vu.nl

ftp.cs.vu.nl. 86400 IN CNAMEzephyr.cs.vu.nl

rowboat IN A 130.37.56.201

IN MX 1 rowboat

IN MX 2 zephyr

IN HINFO Sun Unix

little-sister IN A 130.37.62.23

IN HINFO Mac MacOS

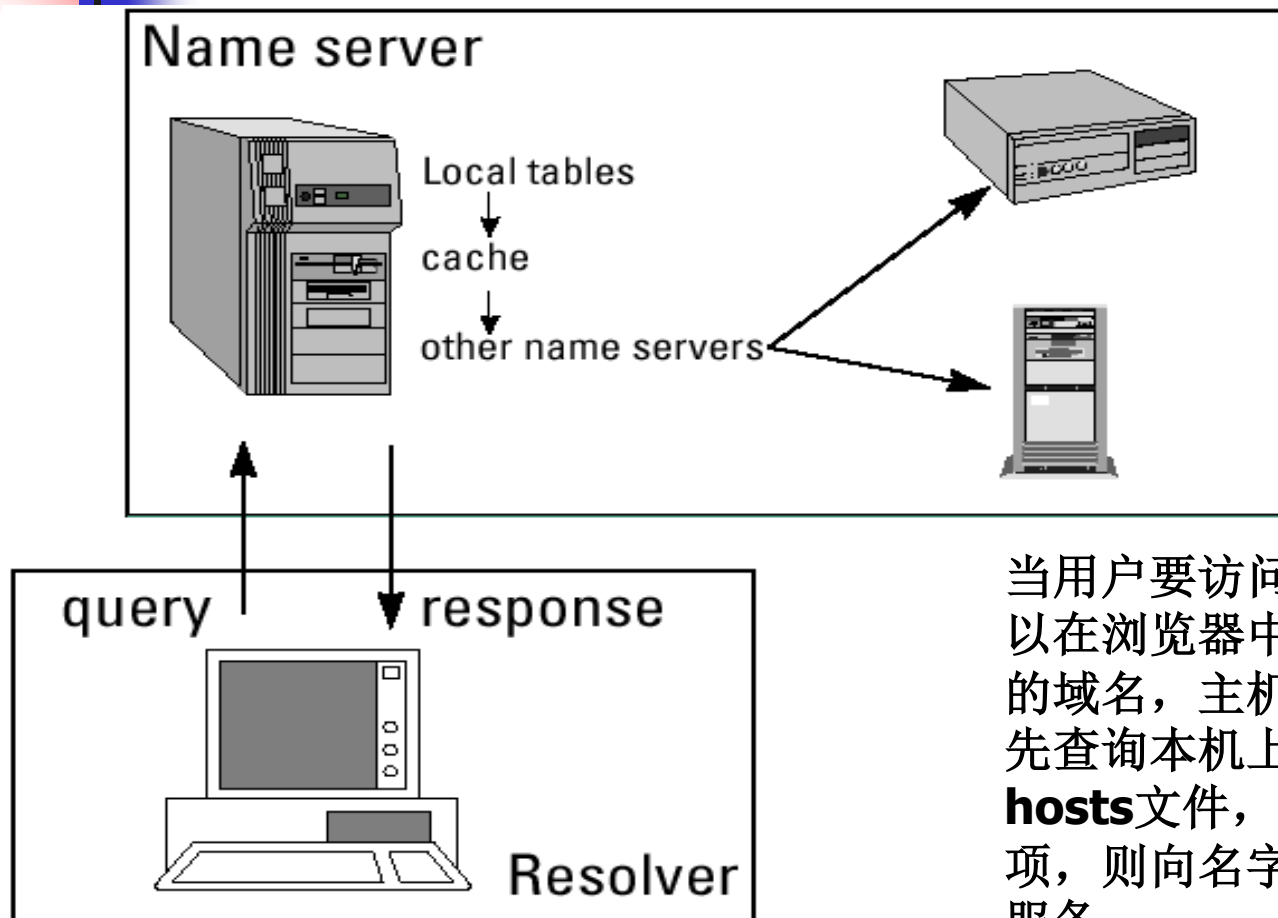
laserjet IN A 192.31.231.216

IN HINFO "HP Laserjet III Si" Proprietary



资源记录类型

- **权威记录: Authoritative Record**
 - 来自管理该记录的授权名字服务器，因此总是正确的
 - 例如**eeis.ustc.edu.cn**在**ustc.edu.cn**域所对应的名字服务器所管辖的范围内，因此该名字服务器返回的关于**eeis.ustc.edu.cn**的记录为权威记录
- **缓存记录: Cached Record**
 - 名字服务器缓存来自其它名字服务器的记录，这些有效期由**TTL**域指定，当**TTL**过期时，该记录被删除



当用户要访问**web**服务时，可以在浏览器中键入**web**服务器的域名，主机上的域名解析器先查询本机上的**cache**以及**hosts**文件，如果没有对应表项，则向名字服务器请求域名服务

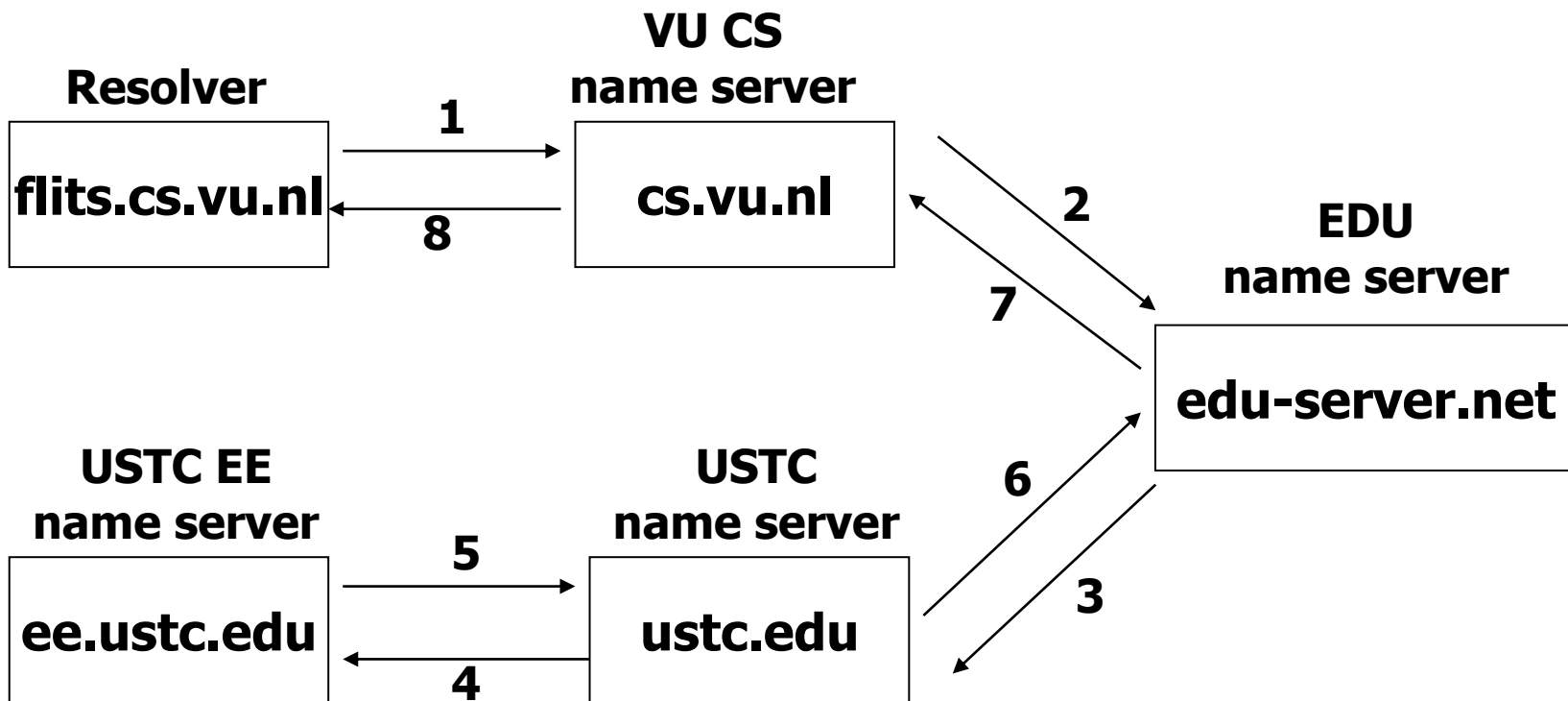


域名查询类型

- **递归查询 (recursive query):** 被请求的名字服务器如果没有域名所对应的记录, 它就会向其它域名服务器查询, 并沿着查询的路径逐个返回记录
- **迭代查询 (iterative query):** 本地名字服务器如果没有域名对应的记录, 就向更高层次的名字服务器请求, 被请求的名字服务器如果也没有该域名对应的记录就会返回一个可供查询的名字服务器地址

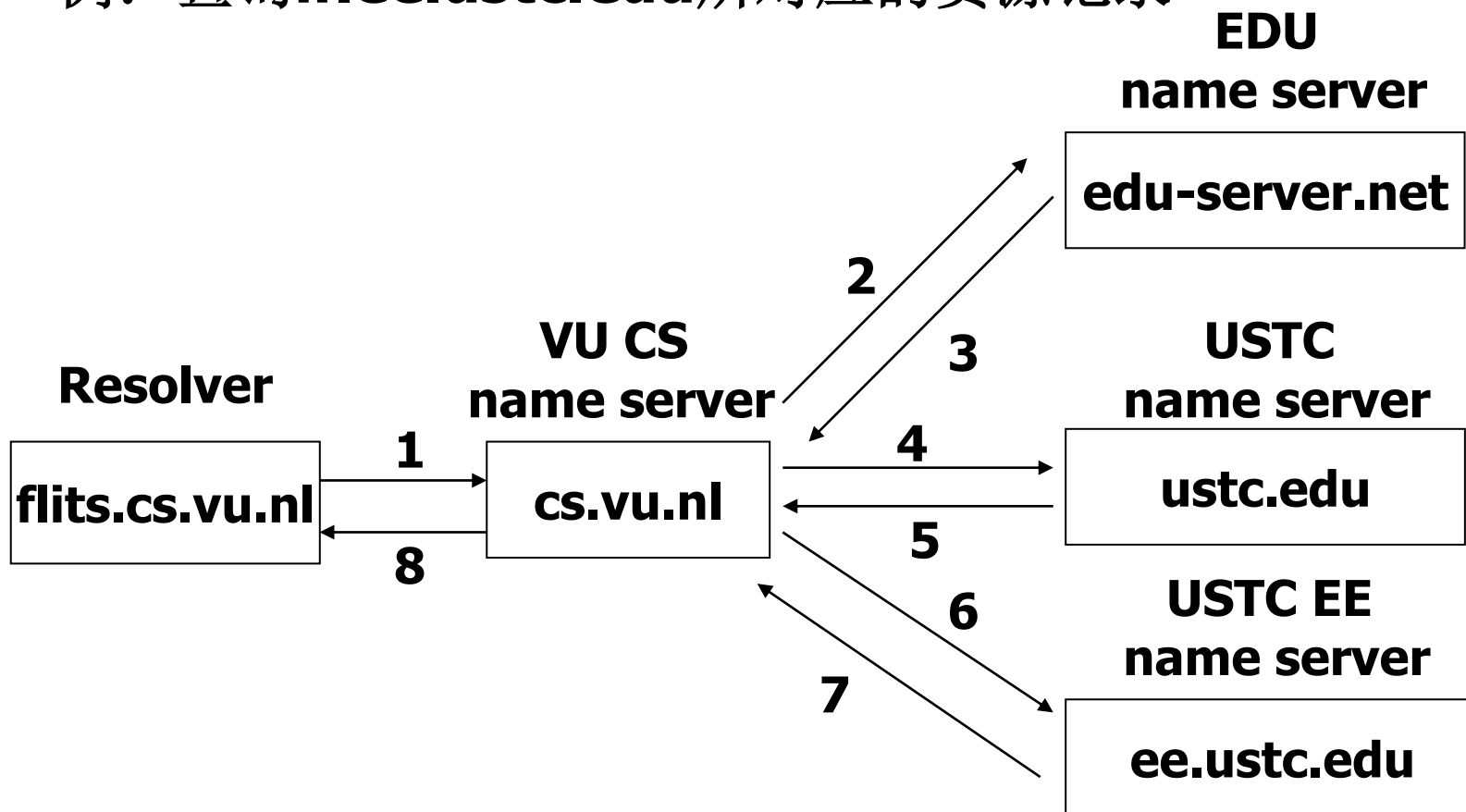
递归查询

例：查询**if.aa.ustc.edu**所对应的资源记录



迭代查询

例：查询**if.ee.ustc.edu**所对应的资源记录



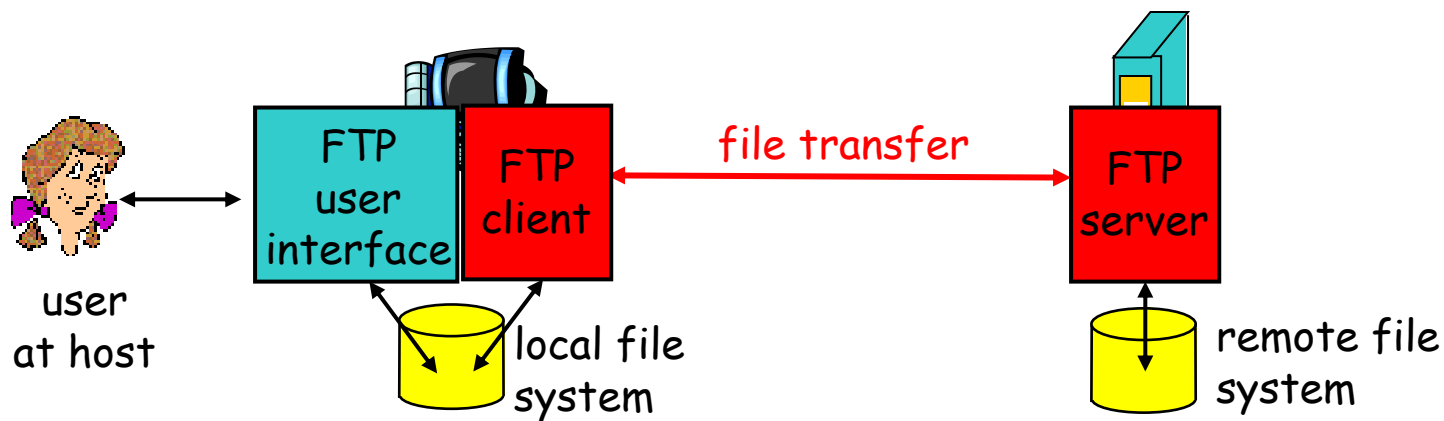


Chapter 9 应用层

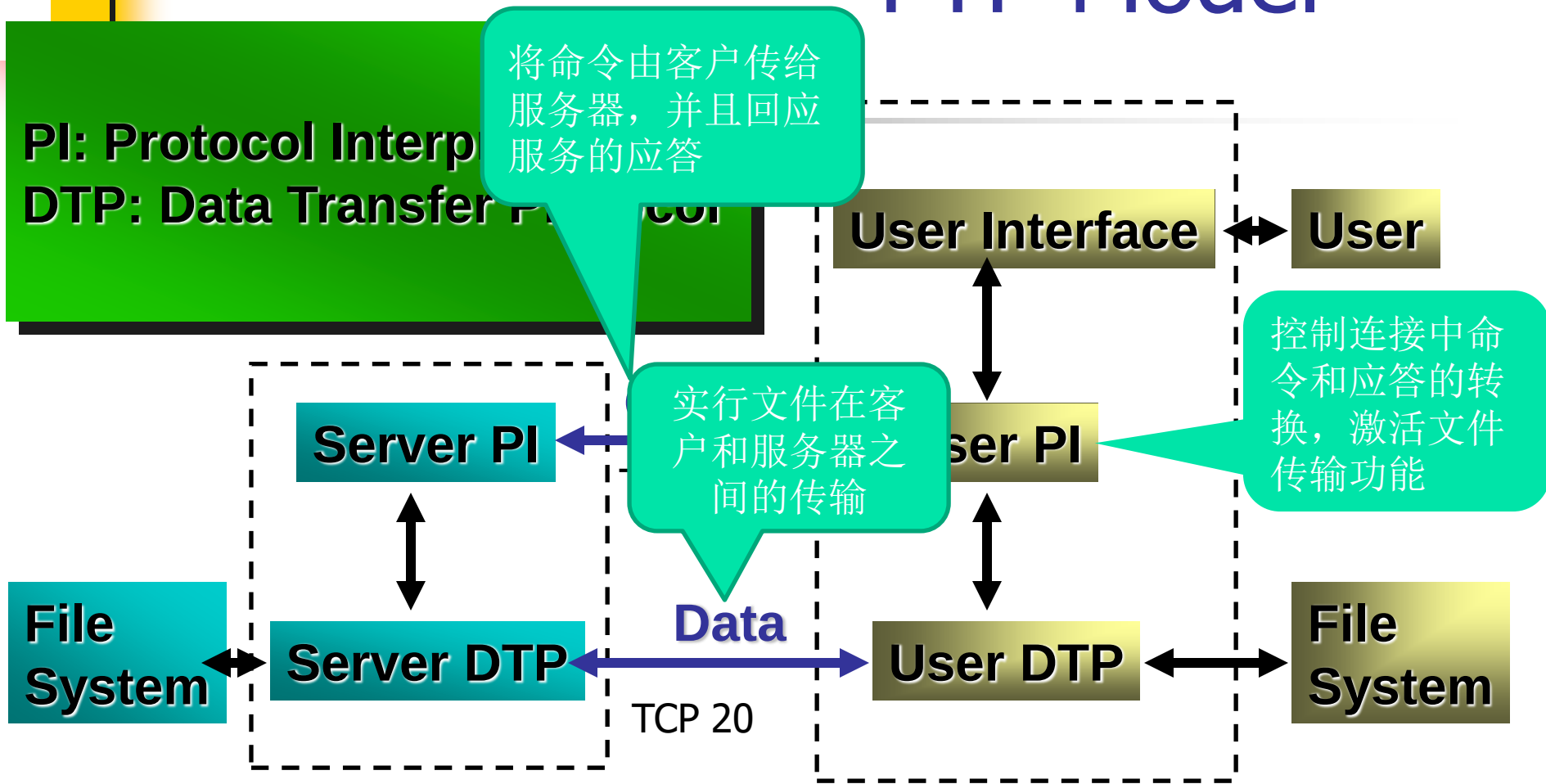
- 9.1网络的计算和访问模式
- 9.2域名系统(DNS)
- 9.3文件服务(FTP)
- 9.4WWW(HTTP)
- 9.5电子邮件(SMTP)

9.3 文件服务(FTP)

- FTP: File Transfer Protocol, RFC959
- 目的: 文件传送（上传或下载）

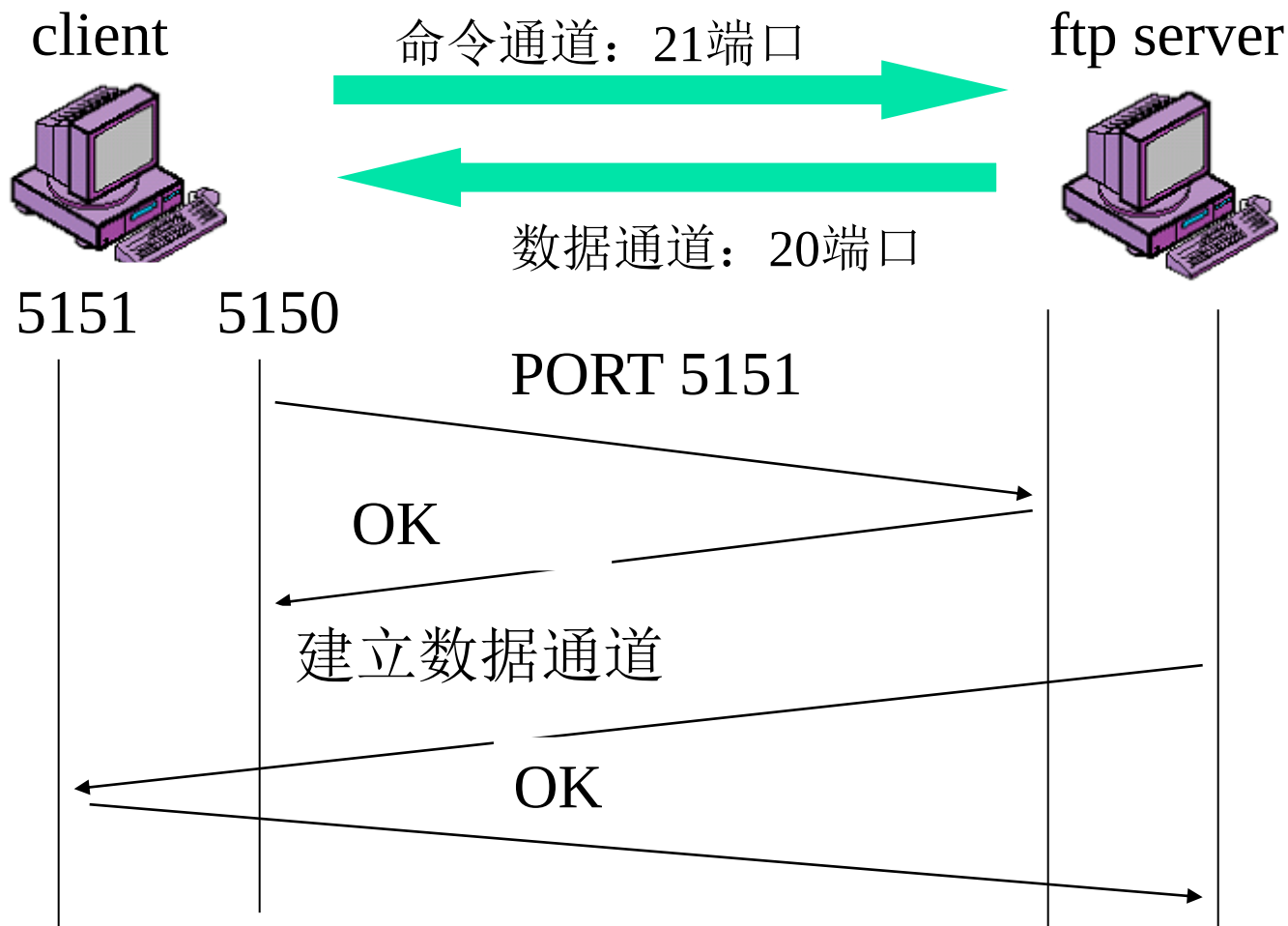


FTP Model

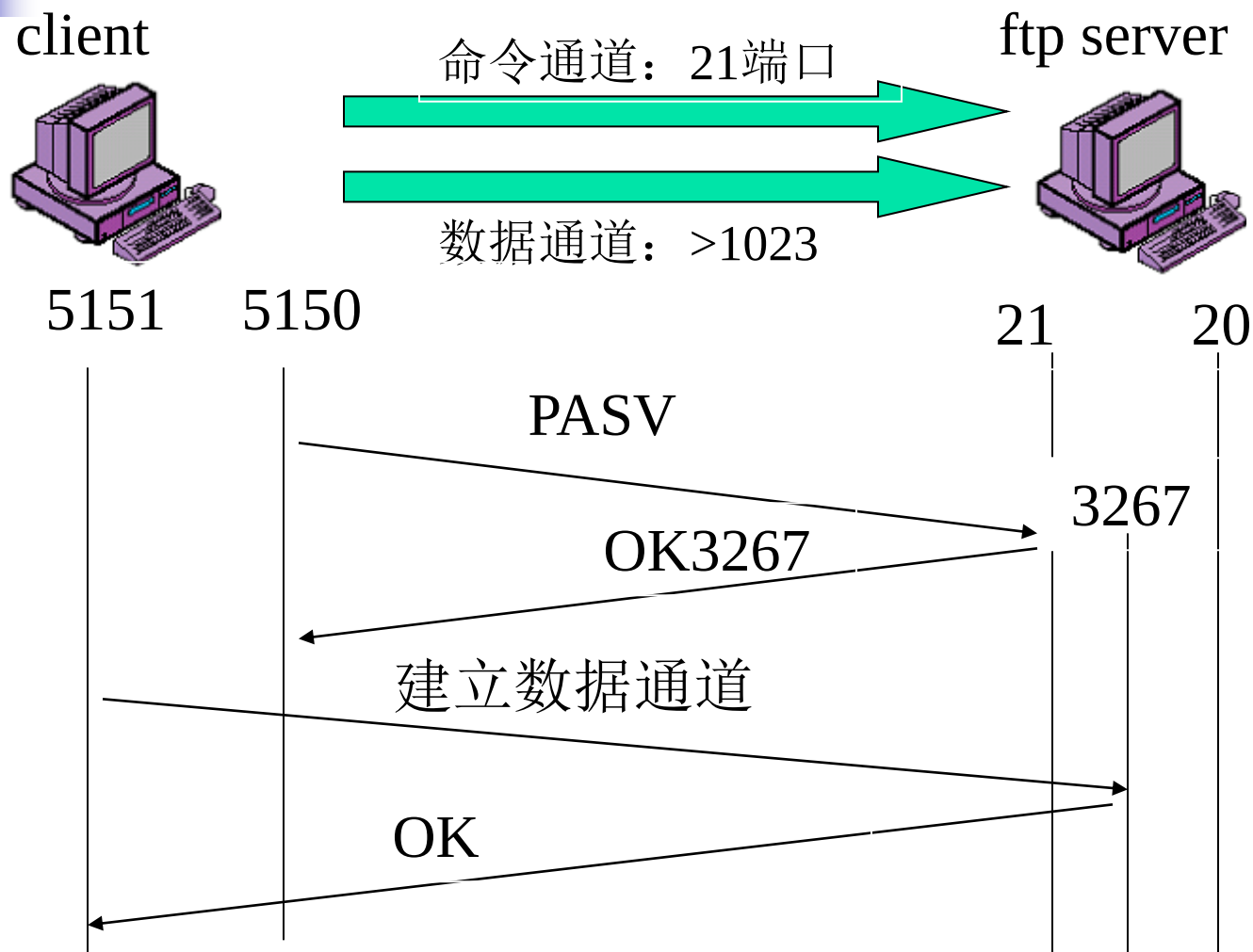


控制连接一直保持在连接的全过程，数据连接根据需要打开

ftp文件传输协议（主动模式）



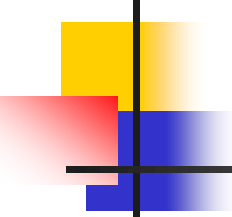
ftp文件传输协议(被动模式)





Chapter 9 应用层

- 9.1网络的计算和访问模式
- 9.2域名系统(DNS)
- 9.3文件服务(FTP)
- **9.4WWW(HTTP)**
- 9.5电子邮件(SMTP)

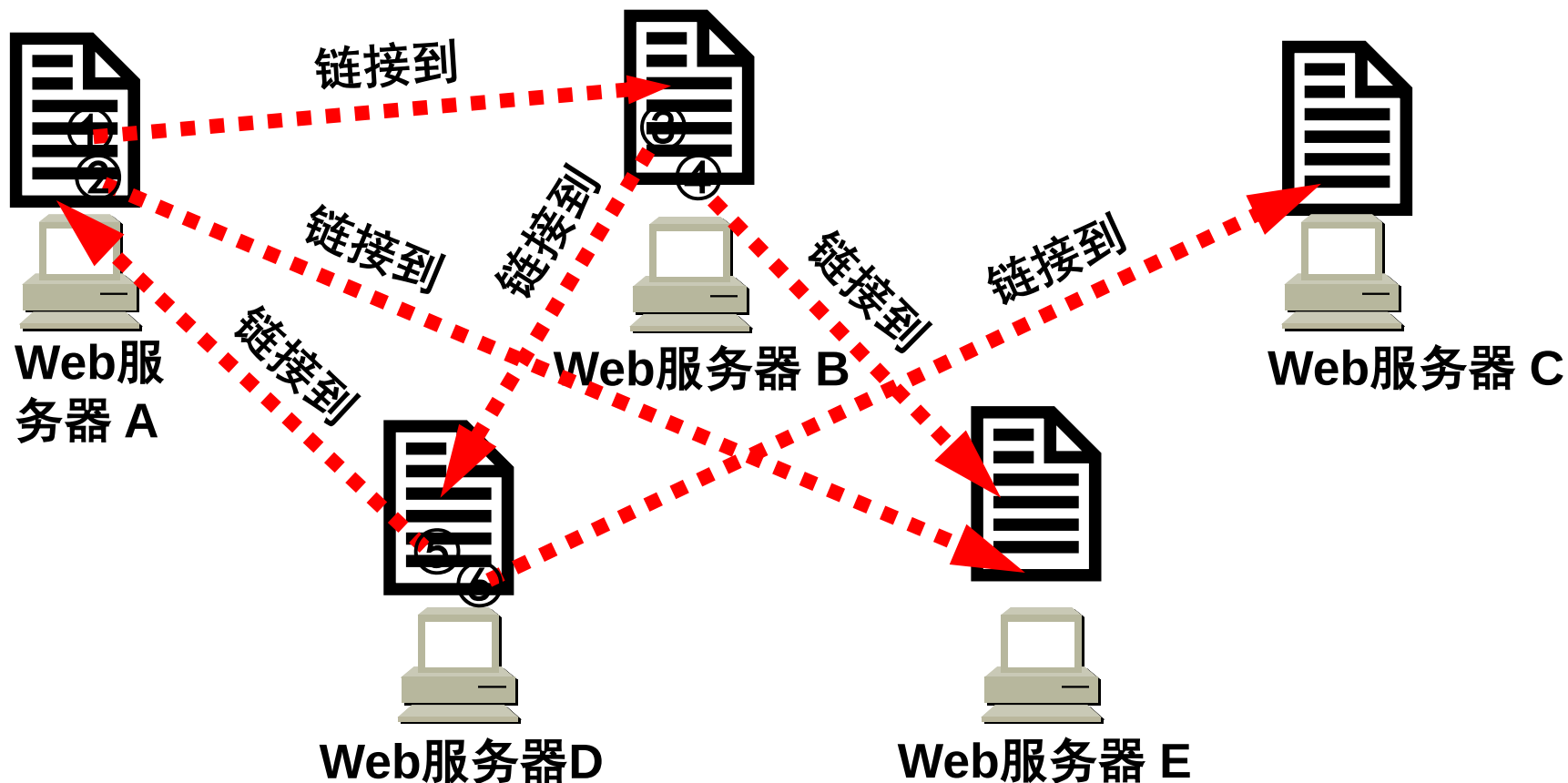


9.4 WWW(HTTP)

- 万维网(**World Wide Web**, 也称为**WWW**)起源于**1989**年欧洲粒子物理研究所, 由庞大的、分布在世界各地的**web**页面的集合组成, **web**页面的编写采用**HTML**(**HyperText Markup Language**)语言, 它采用**HTTP**(**HyperText Transfer Protocol**)协议, 页面通过浏览器(**browser**)来观看
- 万维网可以看作是分布式超媒体(**hypermedia**)系统, 它是超文本(**hypertext**)系统的扩充
 - 一个超文本由多个文本页面链接(**link**)成。利用一个链接可使用户找到另一个文本页面。超文本是万维网的基础。
 - 超媒体与超文本的区别是页面内容不同。超文本页面仅包含文本信息, 而超媒体页面还包含其他表示方式的信息, 如图形、图像、声音、动画、视频等。
 - 使用超链接(**hyperlink**)技术, 用户可以访问一个又一个页面

超链接 (Hyperlink)

超链接是链接到其它页面的文本字符串，
通过超链接，用户可以访问一个又一个的页面





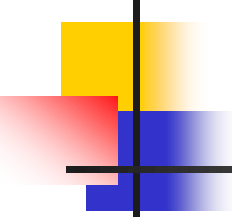
如何命名分布在整个Internet上的web页面？

- 使用**统一资源定位符 URL (Uniform Resource Locator)**来命名万维网上的各种web页面
- 每个页面在整个Internet的范围内具有惟一的标识符 **URL**



统一资源定位符 **URL**

- **URL** 是对可以从**Internet**上得到的资源的位置和访问方法的一种简洁的表示
- **URL** 给资源的位置提供一种抽象的识别方法，并用这种方法给资源定位。只要能够对资源定位，系统就可以对资源进行各种操作，如存取、更新、替换和查找其属性等
- **URL** 相当于一个文件名在网络范围的扩展。因此 **URL** 是与**Internet**相连的主机上的任何可访问对象的一个指针
- 不光使用在**http**中，还可以使用在**ftp**、**telnet**等中



URL的一般形式

- 由协议、主机域名、路径及文件名三部分组成，并且在URL 中的字符对大写或小写没有要求。

- URL 的一般形式是：

<协议>://<主机域名>:<端口>/<路径及文件名>

ftp —— 文件传输协议 FTP

http —— 超文本传送协议 HTTP

mailto —— 发送电子邮件



Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	/usr/suzanne/prog.c
news	News group	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending email	mailto:kim@acm.org
telnet	Remote login	telnet://www.w3.org:80

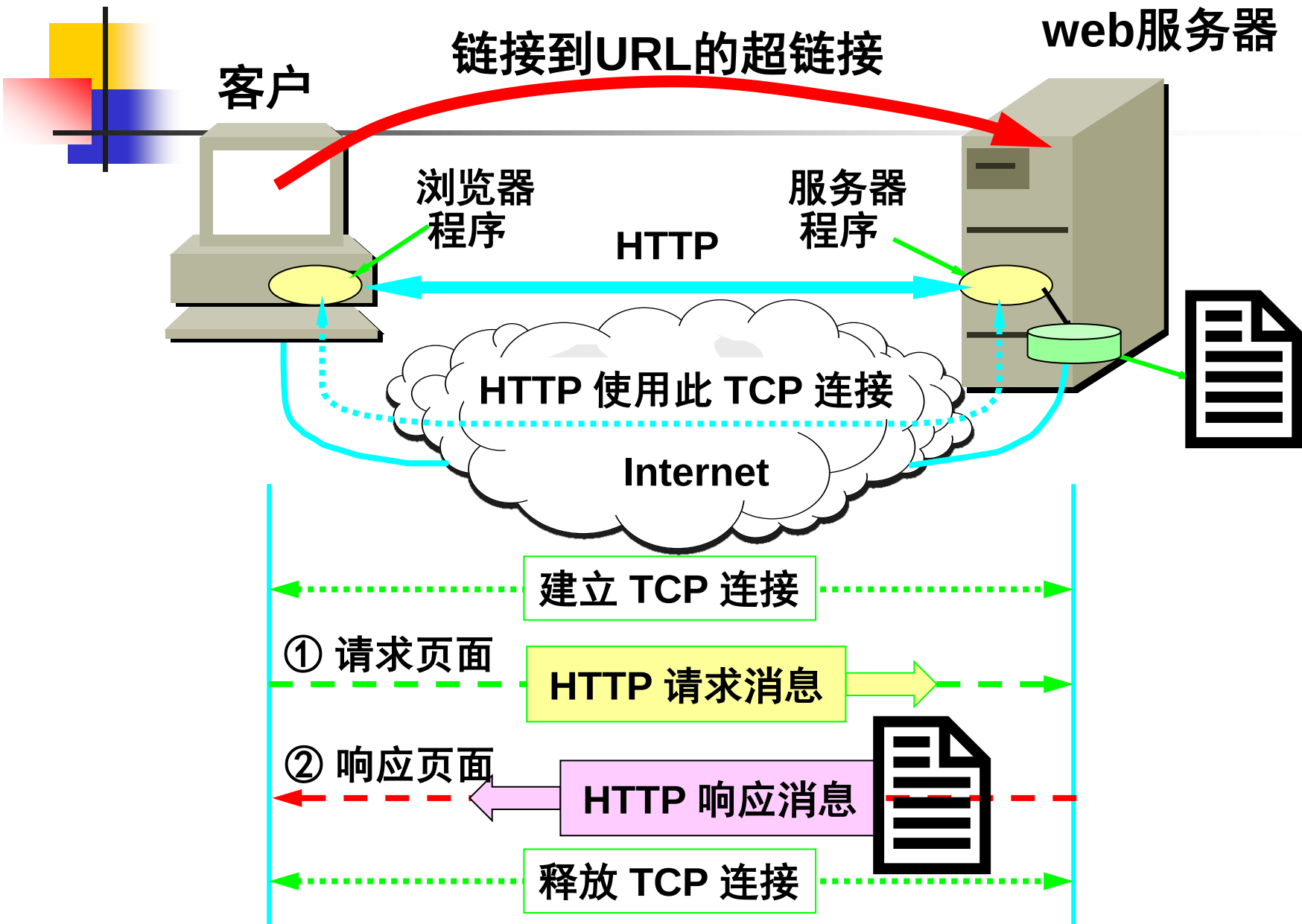
URL (Uniform Resource Locator)统一资源定位器由三部分组成：协议；网站域名；文档路径及文件名



超文本传输协议HTTP

- 采用TCP协议，周知端口为80
- HTTP 是万维网上能够可靠地交换文件（包括文本、声音、图像等各种多媒体文件）的重要基础
- 工作在客户/服务器模式，指定客户可以向服务器发送什么样的消息，并且得到什么样的响应消息
 - 由客户向服务器发出连接，服务器接受与客户的连接
 - 在客户与服务器之间交互HTTP报文
 - 关闭TCP连接

HTTP消息传输过程





万维网工作过程

- 用户在浏览器的地址栏键入**URL**，或者点击某个超链接所对应的**URL**
- 浏览器分析**URL**的主机域名，通过**DNS**系统找到资源所在的主机即**web**服务器的**IP**地址
- 浏览器与**web**服务器的**80**端口建立**TCP**连接
- 向**web**服务器发出**HTTP**请求消息，要求获得某个页面
- 服务器服务器通过**HTTP**响应消息传送页面
- **TCP**连接被释放
- 浏览器在本机上显示页面。

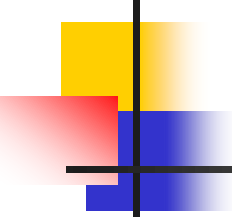


HTTP消息格式

- **HTTP** 有两类消息：

请求消息——客户端发向**web**服务器

响应消息——从**web**服务器发往客户端



请求消息的方法

方法（操作）	意义
OPTION	查询特定选项
GET	请求读取一个web页面
HEAD	请求读取由一个web页面的头部
POST	给服务器添加信息（例如，注释）
PUT	请求存储一个web页面
DELETE	删除web页面
TRACE	送回收到的请求



响应消息的状态码

- **1xx** 表示通知信息，如服务器同意处理客户请求（**100**）
- **2xx** 表示成功，如请求成功（**200**）或没有内容存在（**204**）
- **3xx** 表示重定向，如页面移动（**301**）或者缓存的页面仍然有效（**304**）
- **4xx** 表示客户错误，例如禁止页面（**403**）或者页面没有找到（**404**）
- **5xx** 表示服务器错误，如服务器内部错误（**500**）或者以后重试（**503**）



超文本标记语言HTML

- **HTML**允许在页面中包含文本、图像和指向其它**web**页面的超链接等
- **HTML** 中的 **Markup** 的意思就是“设置标记”，定义了许多用于格式化的显示命令（标记或者标签）
- 浏览器按照**HTML**页面中的各种标签，以及浏览器所使用的显示器的尺寸和分辨率，对页面重新格式化显示出来



Chapter 9 应用层

- 9.1网络的计算和访问模式
- 9.2域名系统(DNS)
- 9.3文件服务(FTP)
- 9.4WWW(HTTP)
- **9.5电子邮件(SMTP)**



相关协议

■ 消息交换的协议

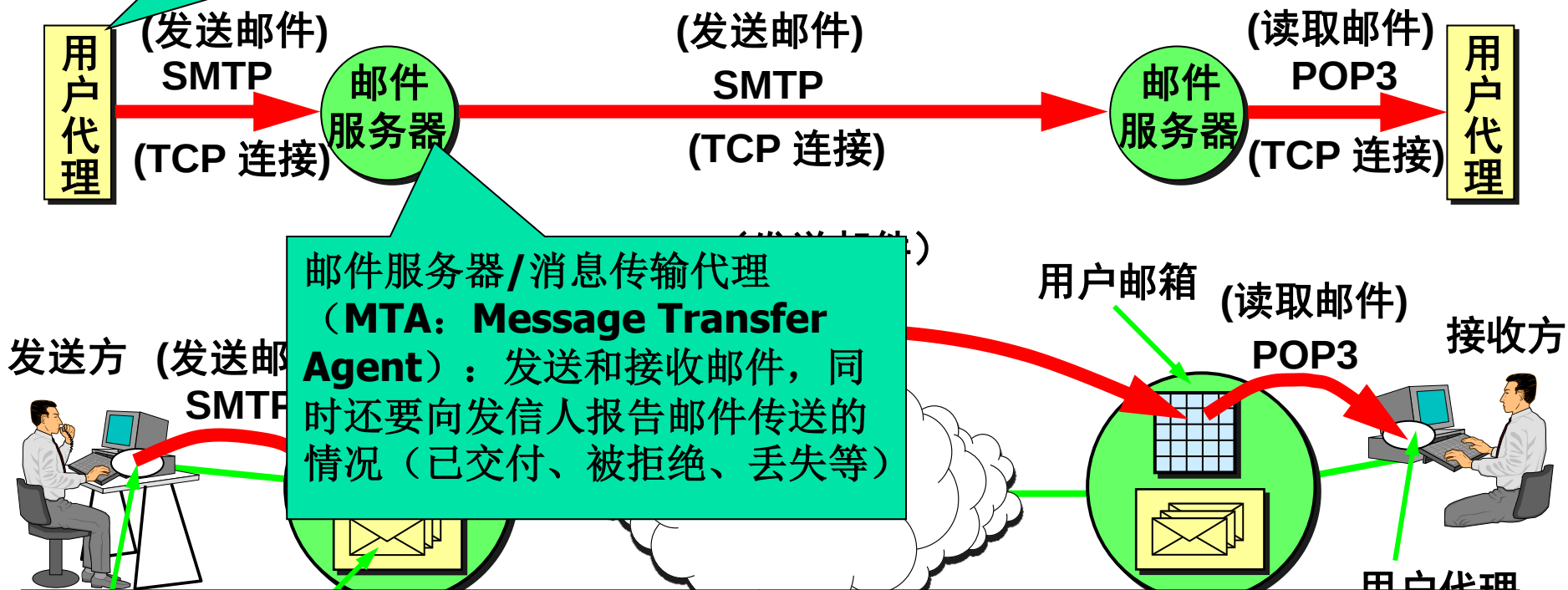
- 发送email: 简单邮件传输协议 SMTP (Simple Mail Transfer Protocol), RFC821
- 接收email: 邮局协议第3版POP3 (Post Office Protocol version 3) RFC1939; Internet消息访问协议 IMAP (Internet Message Access Protocol) RFC3501

■ 消息格式的协议

- RFC 822: 基本的ASCII的文本邮件, 1982
- 多用途Internet邮件扩展 MIME (Multipurpose Internet Mail Extensions): RFC 822的多媒体扩展, 1996

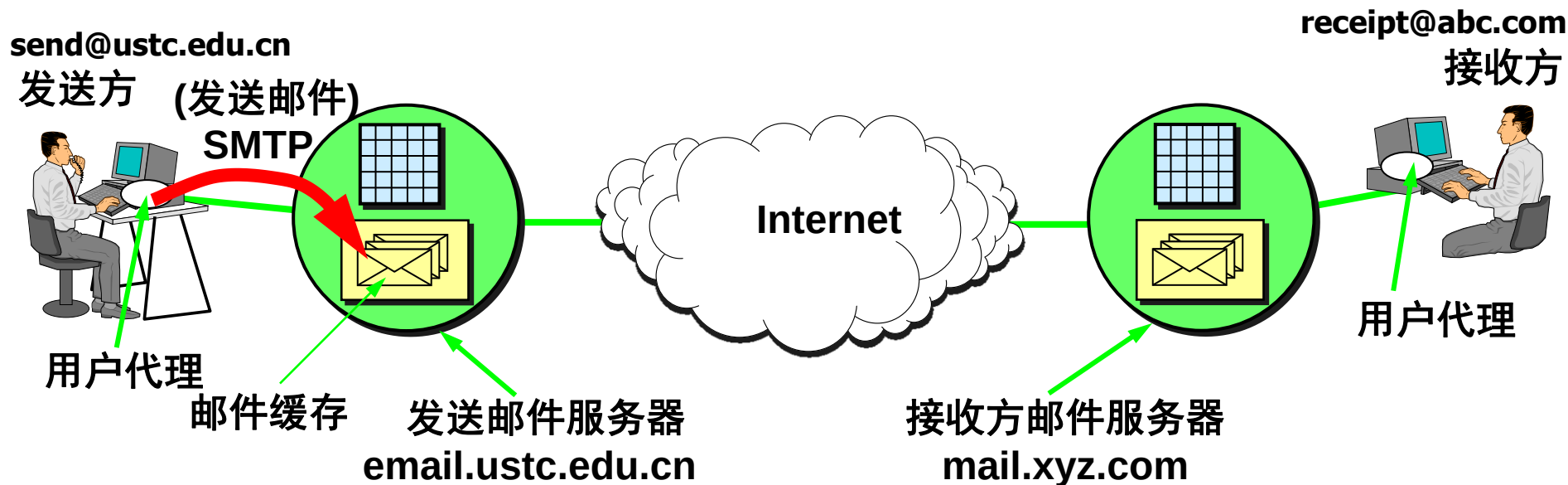
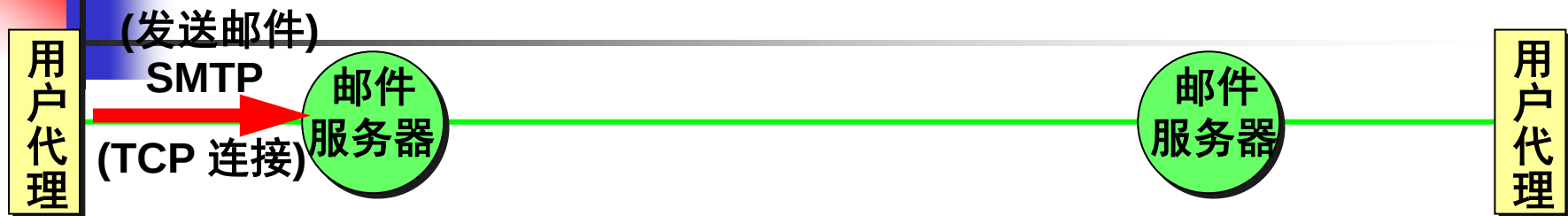
用户代理（**UA: User Agent**）：
负责邮件的生成与处理，实质上
就是一个程序用来阅读、编写、
发送邮件，例如**Outlook**
Express, **Foxmail**等。

组成



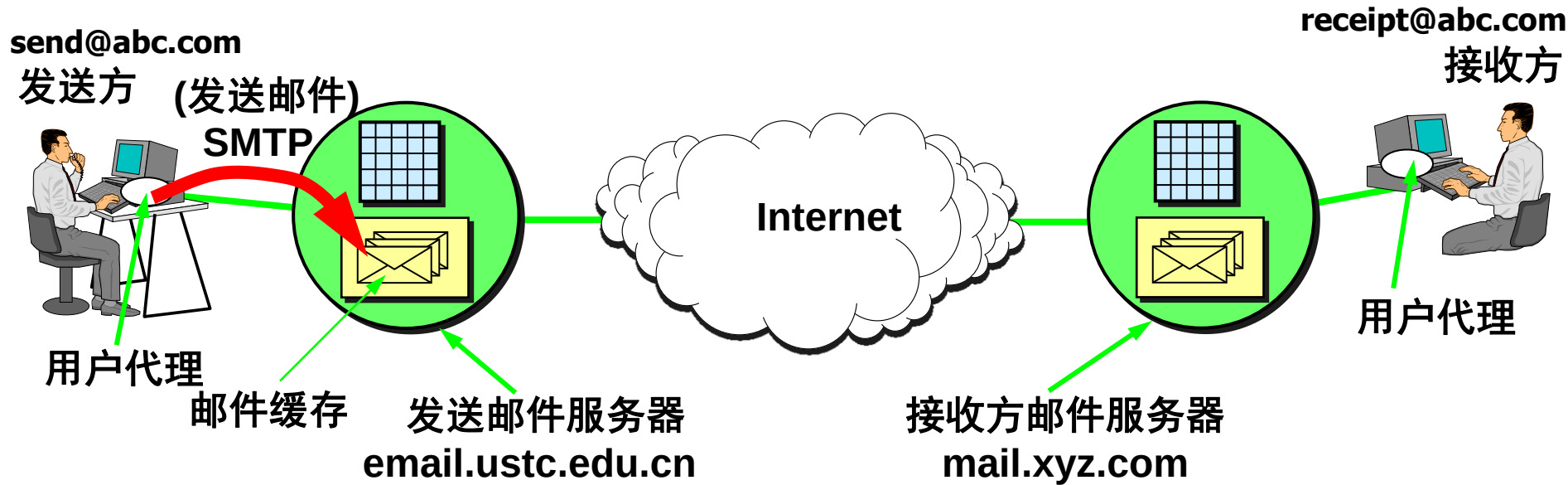
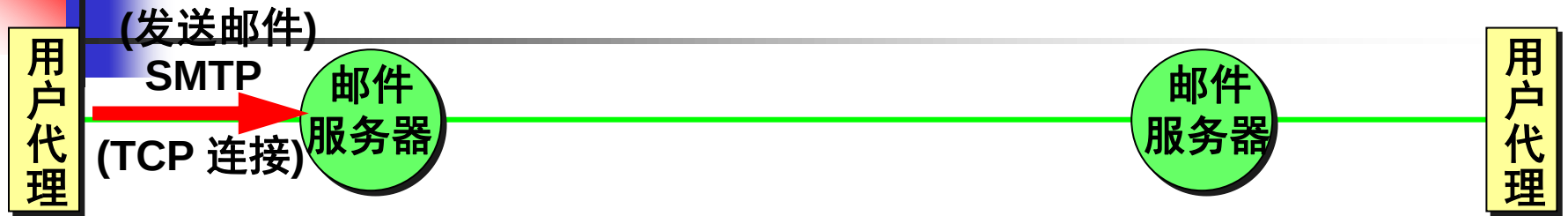
注意：一个邮件服务器既可以作为客户端，也可以作为服务器。
例如，当邮件服务器 **A** 向另一个邮件服务器 **B** 发送邮件时，
邮件服务器 **A** 就作为 **SMTP** 客户，而 **B** 是 **SMTP** 服务器。

电子邮件发送和接收



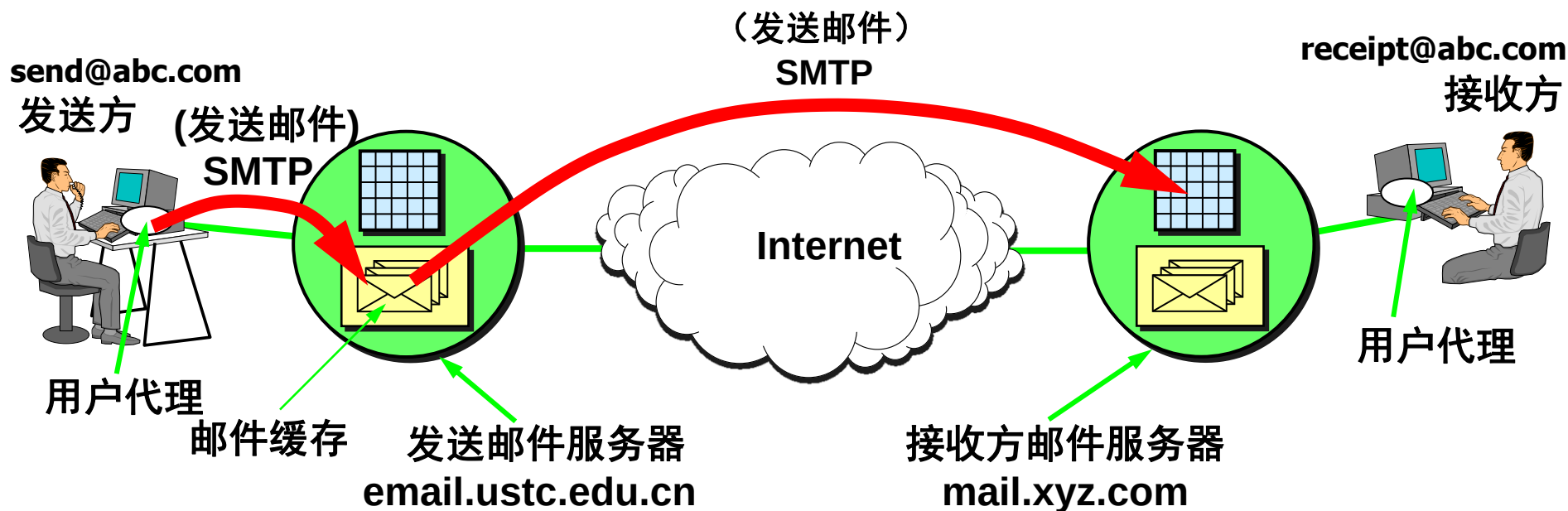
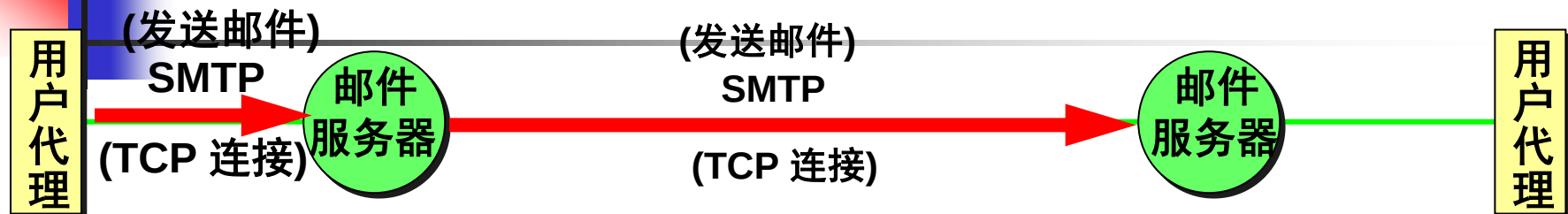
(1) 发信人调用用户代理来编辑要发送到receipt@abc.com的邮件。
用户代理用 SMTP 把邮件传送给预先设置的发送邮件服务器

电子邮件发送和接收



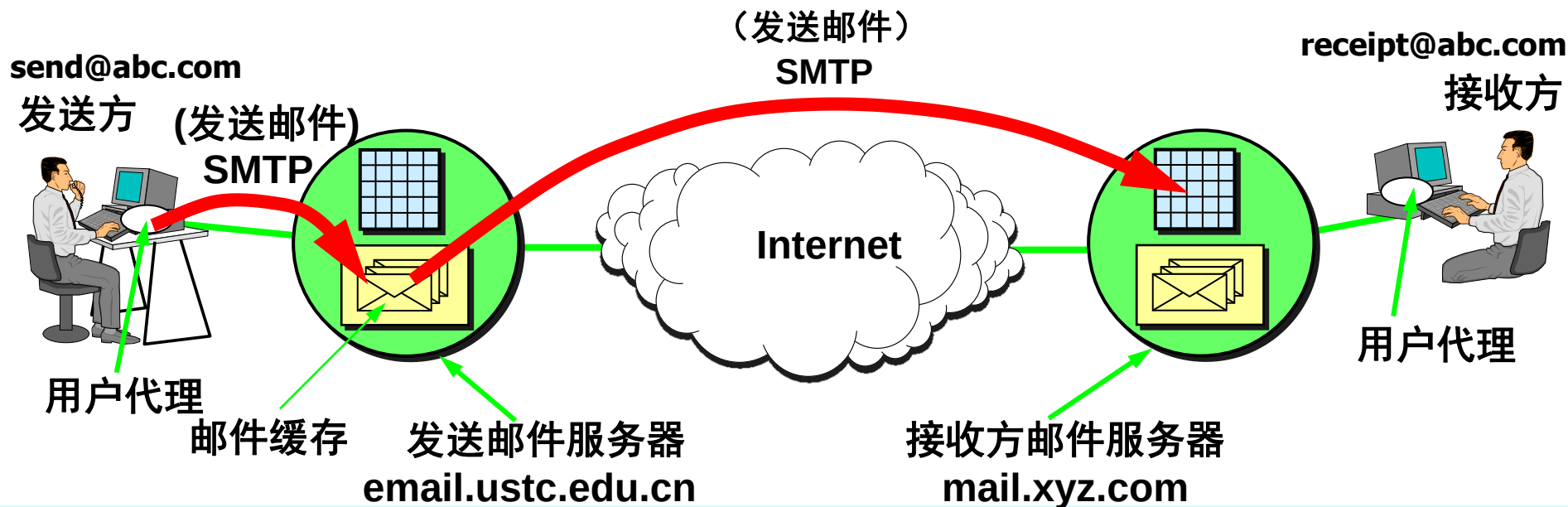
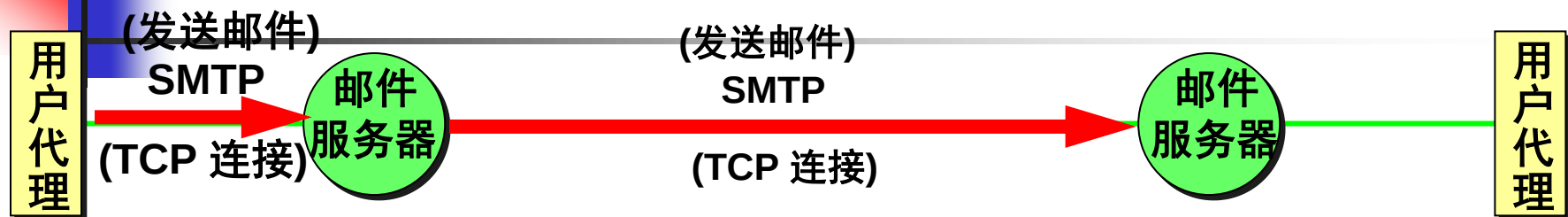
(2) 发送邮件服务器将邮件放入邮件缓存队列中，等待发送。

电子邮件发送和接收



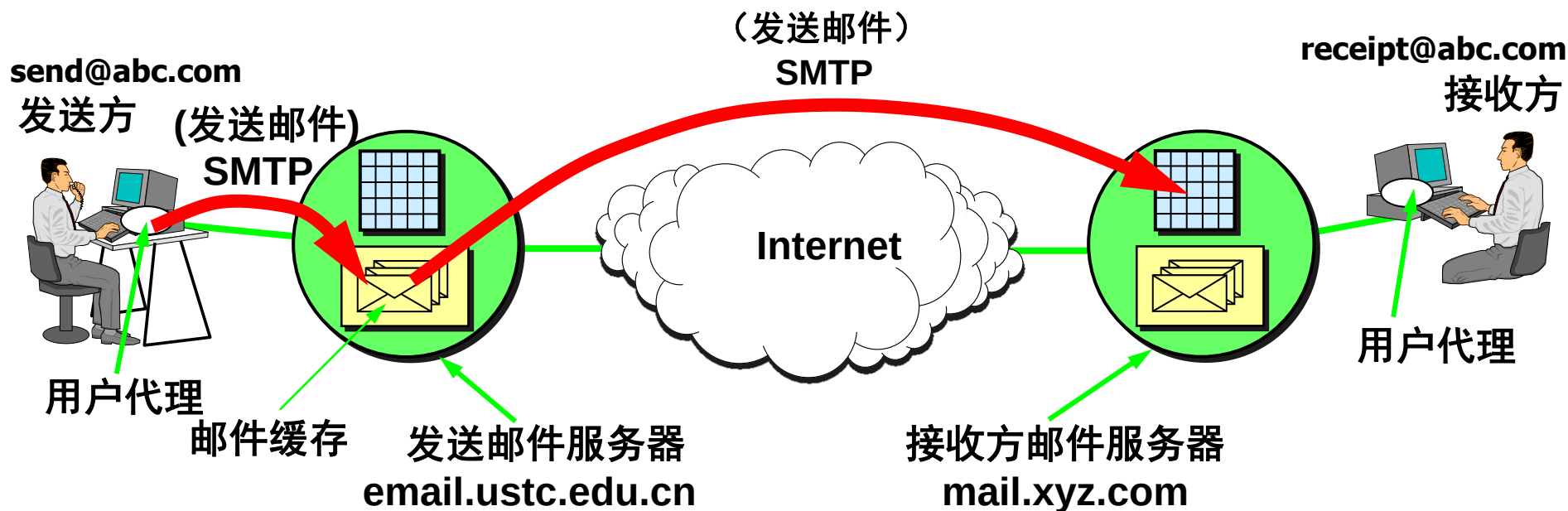
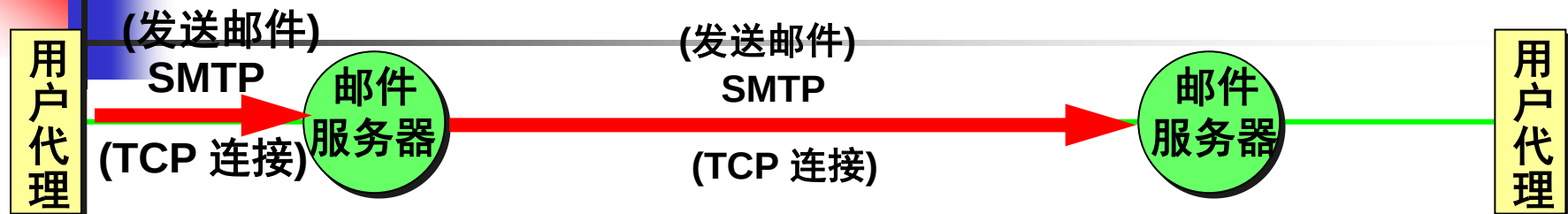
(3) 发送邮件服务器的 SMTP 客户进程，发现在有待发送的邮件，向接收方邮件服务器的SMTP 服务器进程发起 TCP 连接

电子邮件发送和接收



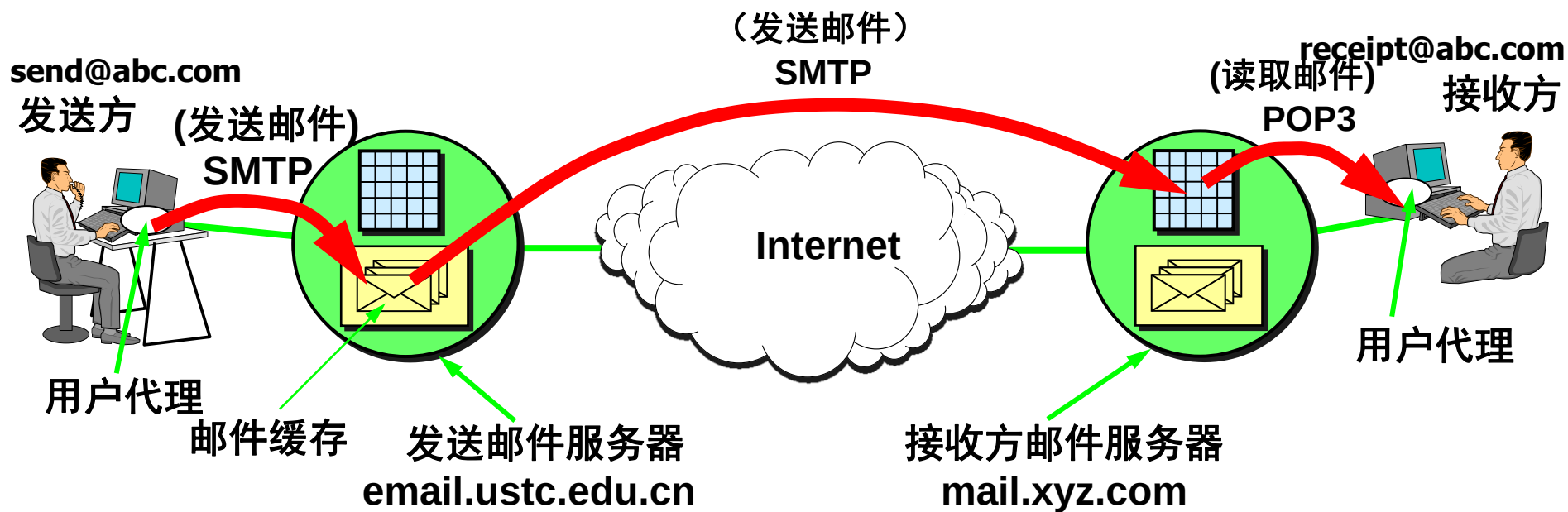
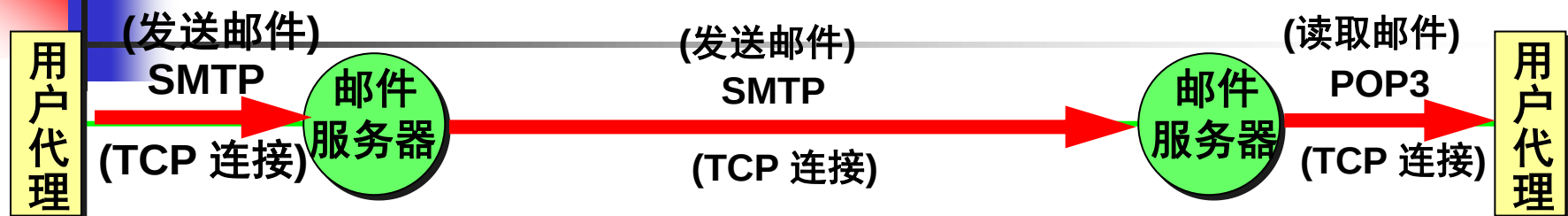
(4) TCP 连接建立后, SMTP 客户进程开始向远程的 SMTP 服务器进程发送邮件。当待发送邮件发送完, SMTP 就关闭建立的连接。

电子邮件发送和接收



(5) 运行在接收方邮件服务器中的 SMTP 服务器进程收到邮件后，将邮件放入接收方的用户邮箱中，等待接收方在方便时进行读取。

电子邮件发送和接收



(6) 接收方在打算收信时，调用用户代理，使用 POP3（或 IMAP）协议将自己的邮件从接收方邮件服务器的用户邮箱中的取回。

Push和Pull

MTA: Message Transfer Agent

MAA: Message Access Agent

MTA
client

SMTP协议

MTA
server

Client pushes messages

邮件发送
基于**Push**
操作

MAA
client

POP3、IMAP协议

MAA
server

Client pulls messages

邮件接收
基于**Pull**
操作



电子邮件格式

Behrouz Forouzan
De Anza College
Cupertino, CA 96014

Sophia Fegan
Com-Net
Cupertino, CA 95014

Sophia Fegan
Com-Net
Cupertino, CA 95014
Jan. 5, 2005

Subject: Network

Dear Ms. Fegan:
We want to inform you that
our network is working properly
after the last repair.

Yours truly,
Behrouz Forouzan

Mail From: forouzan@deanza.edu
RCPT To: fegan@comnet.com

From: Behrouz Forouzan
To: Sophia Fegan
Date: 1/5/05
Subject: Network

Dear Ms. Fegan:
We want to inform you that
our network is working properly
after the last repair.

Yours truly,
Behrouz Forouzan

Envelope

Header

Body
Message



电子邮件地址格式

用户名@邮件服务器的域名

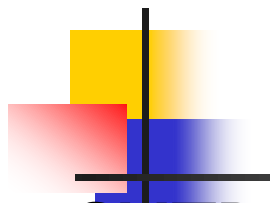
符号“@”读作“**at**”，表示“在”的意思。

例如，电子邮件地址

student@mail.ustc.edu.cn

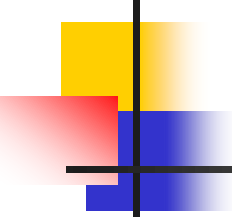
用户名在该域的
范围内是惟一

邮箱所在邮件服
务器的域



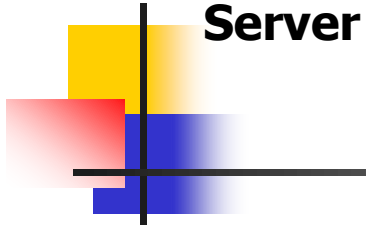
简单邮件传送协议 SMTP

- SMTP 规范两个相互通信的 SMTP 进程之间应如何交换信息，传输层采用TCP协议，25号端口。
- SMTP 使用客户/服务器方式，发送邮件进程就是 SMTP 客户，接收邮件进程就是 SMTP 服务器。
- SMTP 采用简单的ASCII消息，在多媒体、加密传输中存在问题。
 - 多用途Internet邮件扩展（MIME）
- SMTP以明文方式传输，缺少认证，安全上存在不足。
 - 带扩展功能的SMTP(ESMTP)



POP

- 邮局协议 **POP (Post Office Protocol)** 是一个非常简单、但功能有限的邮件读取协议，现在使用的是它的第三个版本 **POP3**
- **POP3**使用客户/服务器的工作方式
 - 在接收邮件的用户主机中必须运行 **POP** 客户端进程，而在用户所连接的邮件服务器中则运行 **POP** 服务器进程
- **POP3**协议一般假设用户从服务器上把邮件下载到本地主机上，用户对邮件的操作在本机进行
 - **IMAP**：邮件存储在邮件服务器上，用户本机**IMAP**客户端操作与邮件服务器同步



**MAA
Server**

**MAA: Message
Access Agent**

Mail Server



POP3
server

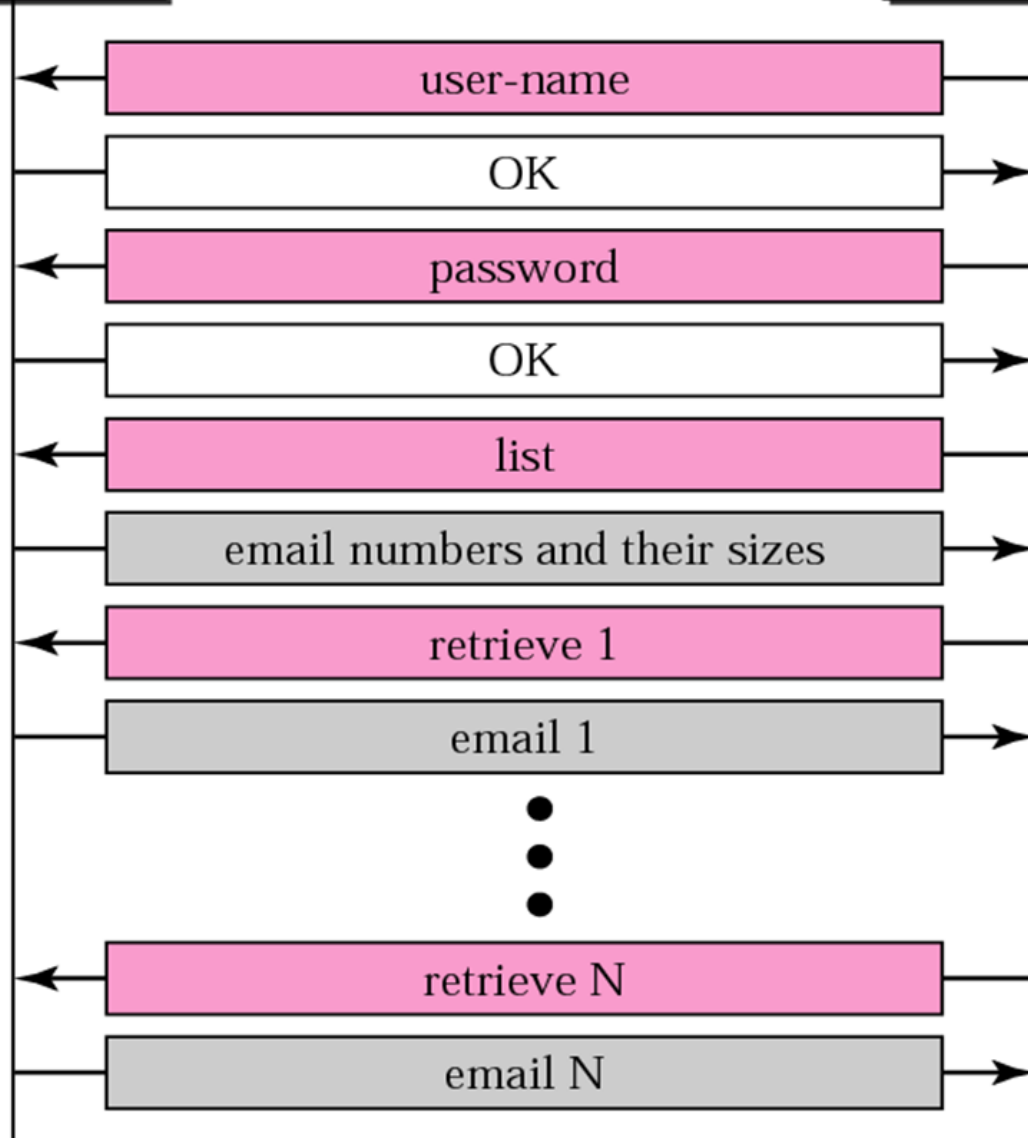
POP3

User Computer



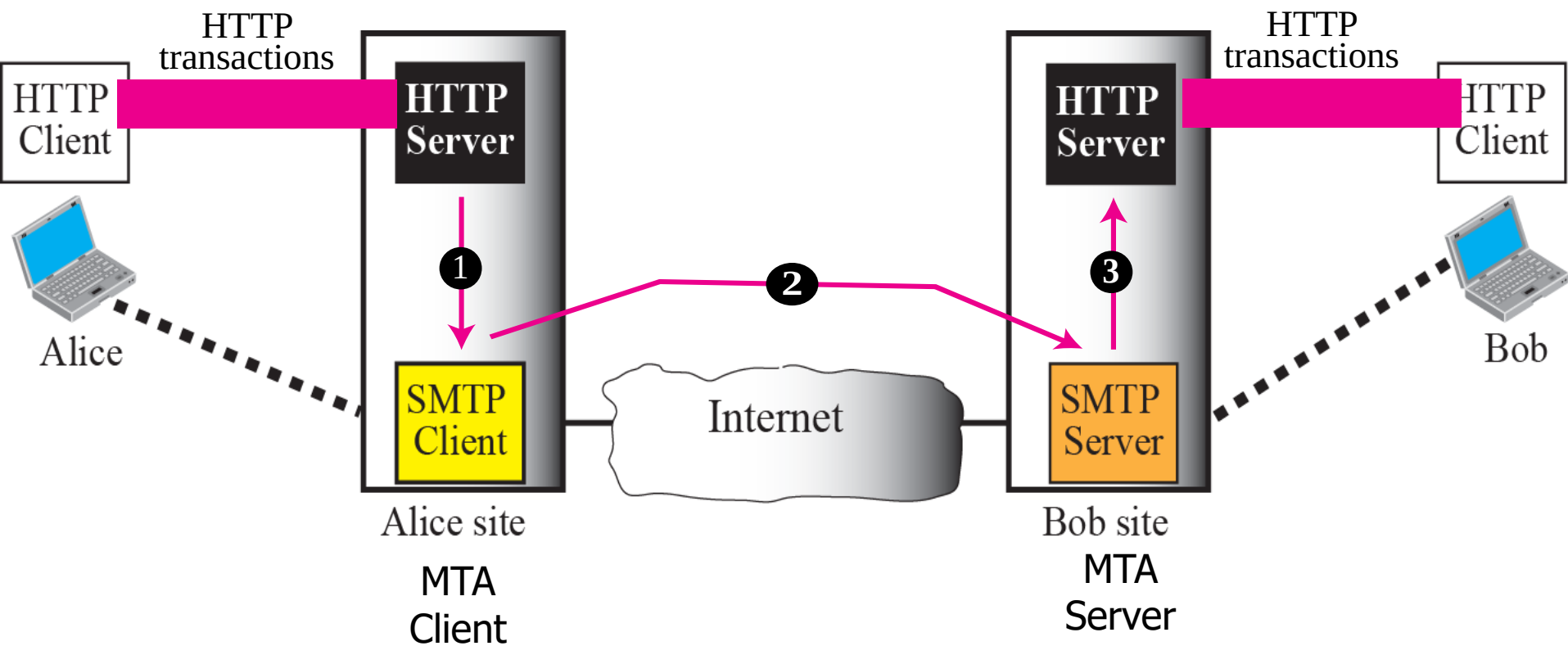
POP3
client

**MAA
Client**



基于web的mail

- Gmail, Hotmail, Coremail,





小结

- 客户/服务器模式与**P2P**模式
- **DNS**系统中域名、域、区域及资源记录的概念，迭代查询和递归查询
- **FTP**的主动模式和被动模式
- **URL**的概念、**HTTP**访问的基本流程
- 电子邮件应用中发送和接收邮件方式



Chapter 10网络安全概要

- 10.1概述
- 10.2密码学基础知识
- 10.3数字签名与认证
- 10.4典型的网络安全威胁
- 10.5网络安全协议课程体系介绍



Chapter 10网络安全概要

- 10.1概述
- 10.2密码学基础知识
- 10.3数字签名与认证
- 10.4典型的网络安全威胁
- 10.5网络安全协议课程体系介绍



10.1概述

■ 1.常见的不安全因素

- 物理因素：物理设备的不安全，电磁波泄漏等
- 系统因素：系统软、硬件漏洞，病毒感染，入侵
- 网络因素：网络协议漏洞，会话劫持、数据篡改，网络拥塞，拒绝服务
- 管理因素：管理员安全意识淡漠，误操作



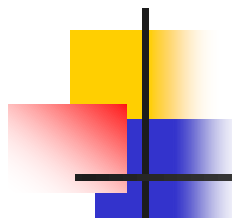
2. 不安全的原因

- **自身的缺陷：**系统软硬件缺陷，网络协议的缺陷
- **开放性**
 - 系统开放：计算机及计算机通信系统是根据行业标准规定的接口建立起来的。
 - 标准开放：网络运行的各层协议是开放的，并且标准的制定也是开放的。
 - 业务开放：用户可以根据需要开发新的业务。
- **黑客攻击**
 - 基于兴趣的入侵
 - 基于利益的入侵
 - 信息战



3. 网络的潜在威胁

- 非授权访问（**unauthorized access**）：非授权用户的入侵。
- 信息泄露（**disclosure of information**）：造成将有价值的和高度机密的信息暴露给无权访问该信息的人。
- 拒绝服务（**denial of service**）：使得系统难以或不可能继续执行任务。

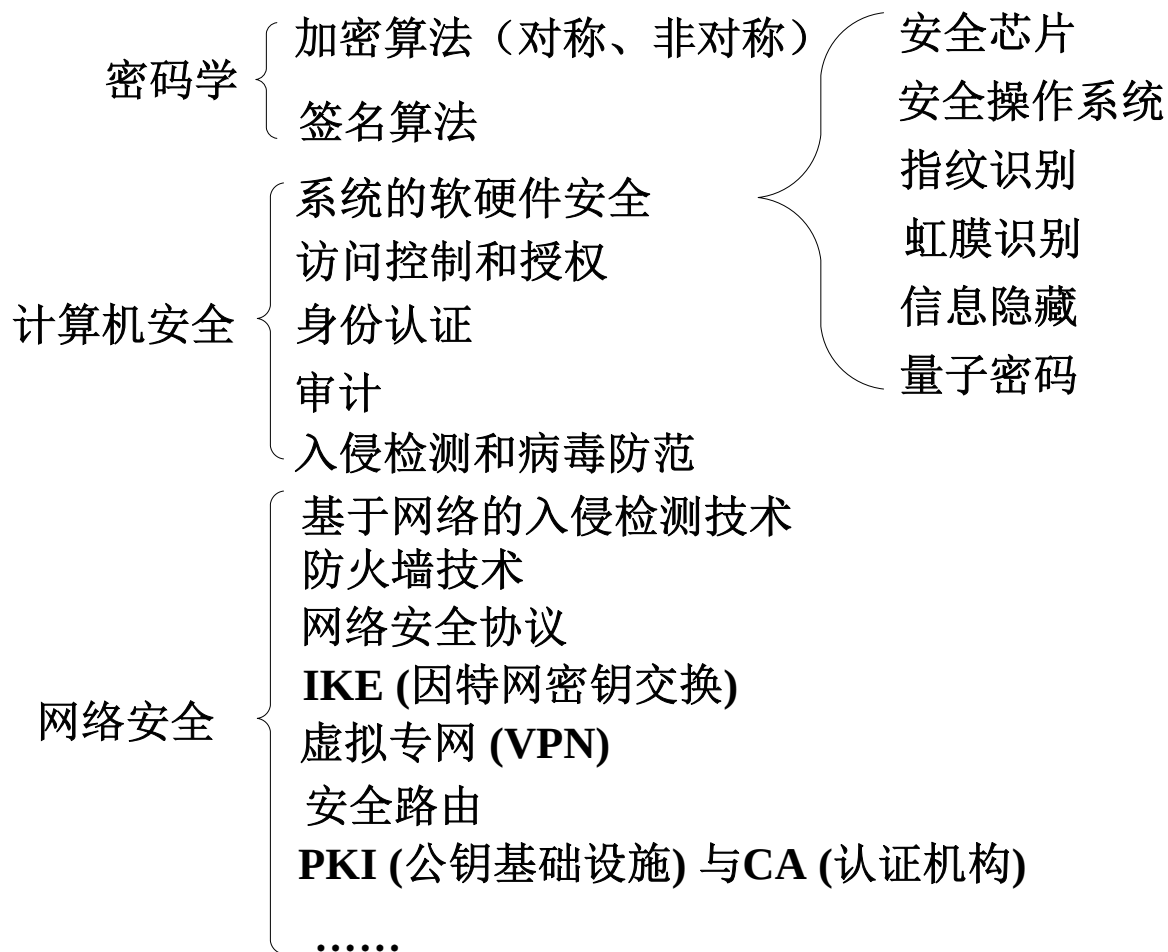


4.网络安全服务

- **认证**（Authentication）:提供某个实体的身份保证
- **访问控制**（Access control）:保护资源，防止对它的非法使用和操纵
- **数据加密**（Data encryption）:保护信息不被泄露
- **数据完整性**（Integrity）:保护信息以防止非法篡改
- **不可否认性**（No-repudiation）:防止参与通信的一方事后否认



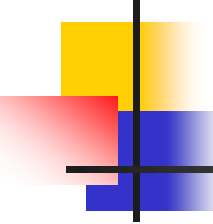
5. 信息安全分类





6. 安全理论与技术

- 密码理论与技术
- 认证识别理论与技术
- 授权与访问控制理论与技术
- 审计追踪技术
- 防火墙与访问代理技术
- 入侵检测技术
- 反病毒技术



7. 网络安全学科

- 网络安全的重要性
 - 政务、商务、金融、军事、社会稳定等
- 网络安全是一个跨多门学科的综合性的科学
 - 包括：通信技术、网络技术、计算机软件、硬件设计技术、密码学等
 - 在理论上，网络安全是建立在密码学以及网络安全协议的基础上的
 - 从技术上，网络安全取决于两个方面：网络设备的硬件和软件的安全，以及设备的访问控制



8.网络安全的主要任务

- 保障网络与系统
 - 安全、可靠、高效、可控、持续地运行
- 保障信息
 - 机密、完整、不可否认地传输和使用



9.需要保护的对象

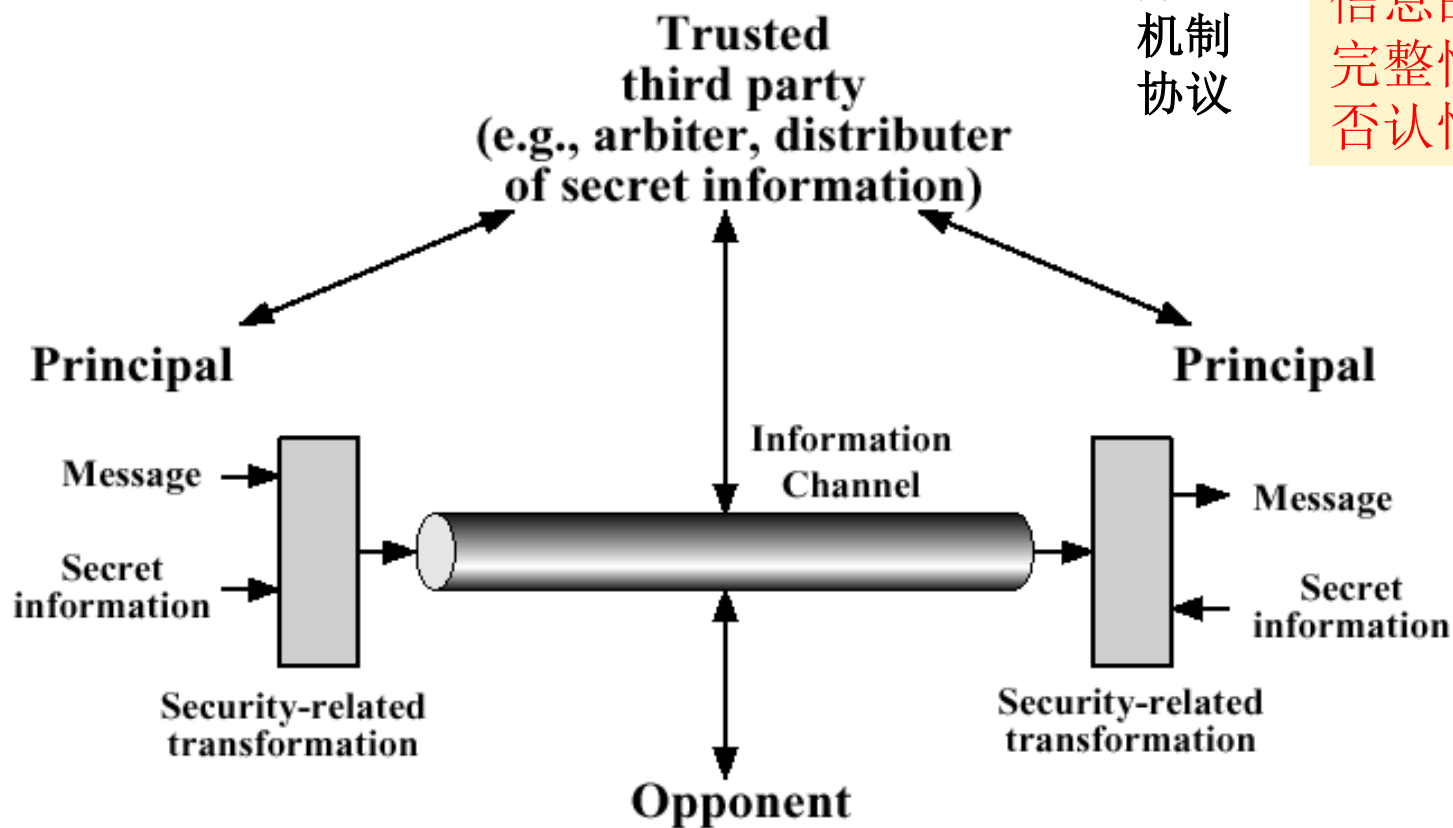
- 硬件
 - 服务器、路由器、主机（PC&工作站）等
- 软件
 - 操作系统、应用软件等
- 数据
 - 电子邮件、电子商务、电子政务、信息发布等

网络安全模型

身份认证

信息的机密性、
完整性、不可
否认性

算法
机制
协议



网络访问安全模型

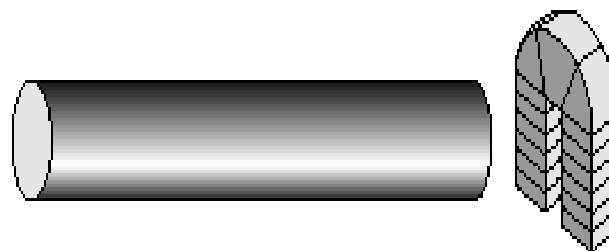
保证网络的
可用性、可控性

Firewalls and Security Gateways

IDS

Information System

Opponent
—human (e.g., cracker)
—software
(e.g., virus, worm)



Access Channel

Gatekeeper
function

Computing resources
(processor, memory, I/O)

Data

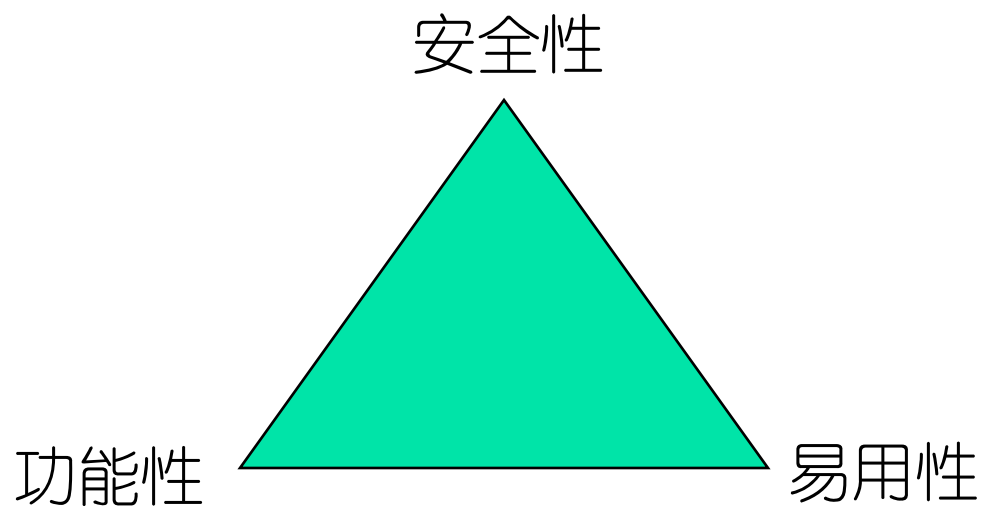
Processes

Software

Internal security controls

安全性、功能性和易用性

- 真正“安全”的机器是没有联网并且深埋在地下的机器。





Chapter 10网络安全概要

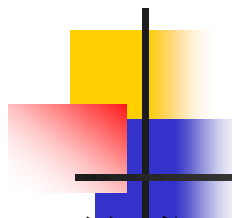
- 10.1概述
- **10.2密码学基础知识**
- 10.3数字签名与认证
- 10.4典型的网络安全威胁
- 10.5网络安全协议课程体系介绍



10.2密码学基础知识

■ 1.基本概念

- 明文 (plaintext) : 作为加密输入的原始信息
- 密文 (ciphertext): 明文加密后的结果
- 加密 (encryption) : 是一组含有参数的变换, 将明文变为密文的过程
- 加密算法: 对明文进行加密时采用的规则
- 解密 (decryption) : 由密文恢复出明文的过程
- 解密算法: 对密文进行解密时采用的规则
- 密钥 (key): 参与变换的参数, 分别有加密密钥和解密密钥



2.密码算法分类

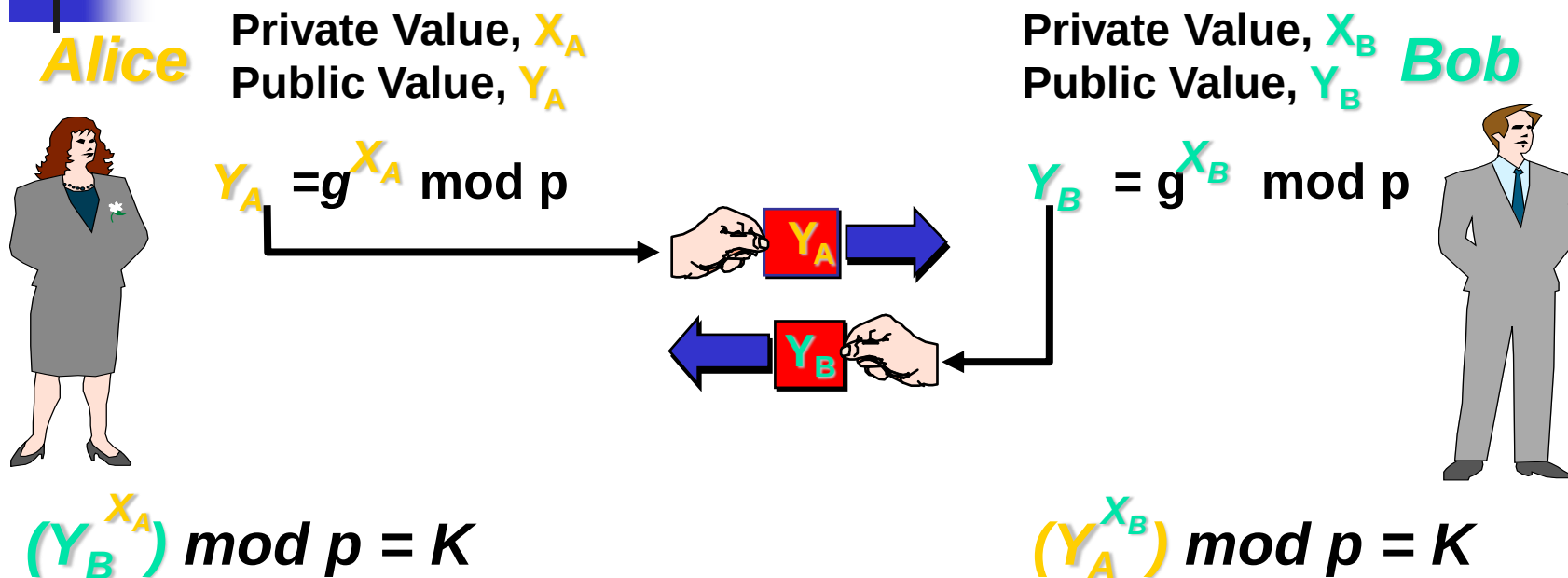
■ 按发展进程或体制分

- 古典密码:基于字符替换的密码，现在已很少使用了，但是它代表了密码的起源
- 对称密钥体制（ Symmetric System ）:加密密钥和解密密钥相同，这些算法也叫作单钥密码体制（one-key system），如DES、3DES、AES、IDEA等
- 非对称密钥体制(Asymmetric System)：加密密钥和解密密钥不同，也叫公钥密码体制（public key system)或双钥密码体制（two-key system），如RSA、ELGamal、椭圆曲线算法

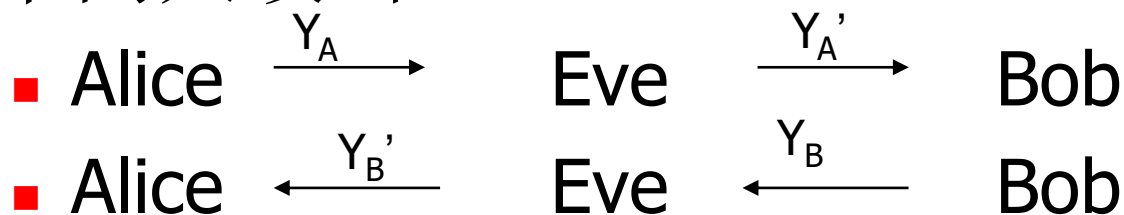
■ 按加密模式分

- 序列密码（stream cipher): 序列密码按位或字节加密，也可以称为流密码，序列密码是手工和机械密码时代的主流。
- 分组密码(block cipher): 分组密码将明文分成固定长度的组，用同一密钥和算法对每一块加密，输出也是固定长度的密文。

3. Diffie-Hellman 密钥交换



■ 中间人攻击





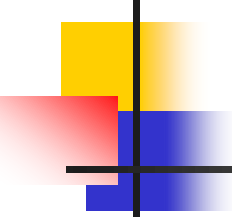
Chapter 10网络安全概要

- 10.1概述
- 10.2密码学基础知识
- **10.3数字签名与认证**
- 10.4典型的网络安全威胁
- 10.5网络安全协议课程体系介绍



10.3 数字签名与认证

- 通信中潜在的威胁
 - 消息伪造
 - 内容篡改
 - 延迟或重播
 - 否认
- 数字签名-消息的不可否认性
- 认证-消息的完整可靠



1. 数字签名

- 公钥密码学的一个重要应用就是数字签名，数字签名就是利用私钥生成签名，而用公钥验证签名。
- 一个数字签名方案时是由**签名算法和验证算法**组成
 - 签名算法利用私钥生成签名，称消息 m 的签名为 $\text{sig}(m)$ ，然后将 $(m, \text{sig}(m))$ 发给接收方
 - 验证算法利用签名者的公钥对 $\text{sig}(m)$ 进行解密，如果解密输出与 m 一致，则为合法数据。
- 由于无法识别数字签名与其拷贝之间的差异，所以，在数字签名前应加上**时间戳**。
- 数字签名标准（DSS）
 - DSA（数字签名算法，是Elgamal公钥算法的一种变体）
 - RSA



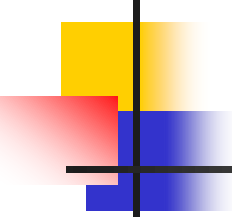
2. 认证

- 可用来做认证的函数有三类：
 - **(1) 加密函数**
 - 用对称密钥加密，信息的完整作为对信息的认证
 - 用公钥密码中的私钥加密，但加密速度太慢，并且很多情况下，消息并不需要加密。
 - **(2) 消息认证码MAC (Message Authentication Code)**
是对信源消息的一个编码函数
 - **(3) 散列函数 (Hash Function) (需要结合签名)**
它将任意长的信息映射成一个固定长度的信息



消息认证码MAC

- 利用函数 f 和密钥 k ，对要发送的明文 x 或密文 y 变换成 r bit的消息认证码 $f(k,x)$ (或 $f(k,y)$)，将其称为认证符附加在 x (或 y)之后发出，接收者收到发送的消息序列后，按发方同样的方法对接收的数据(或解密后)进行计算，应得到相应的 r bit数据

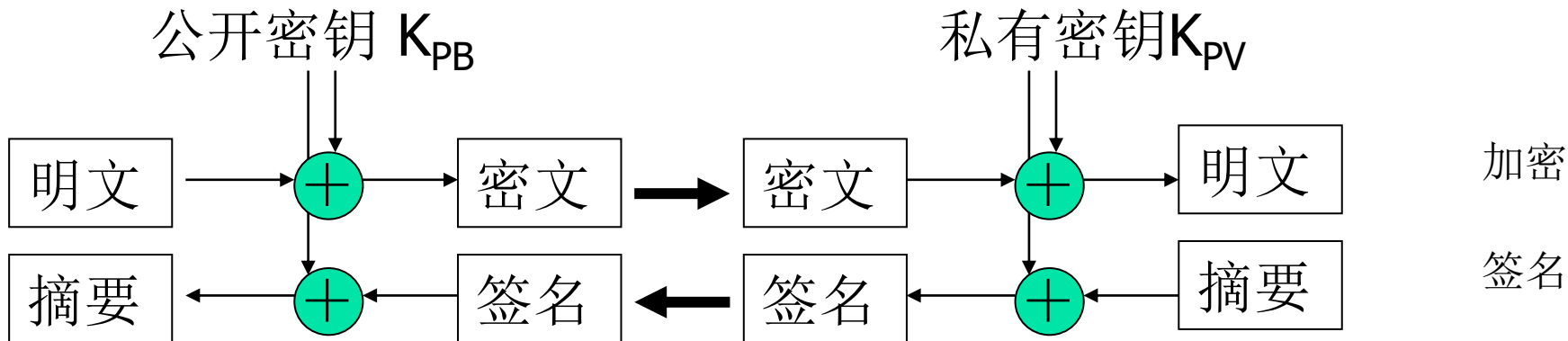
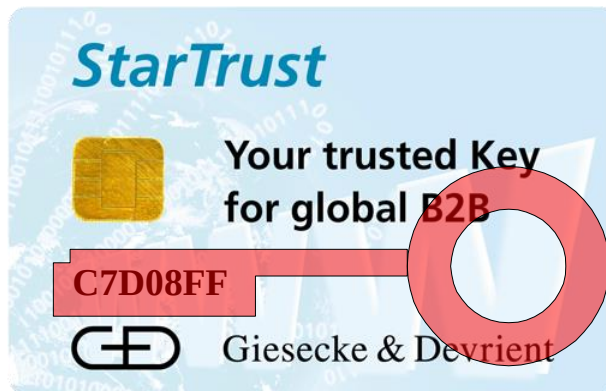


散列函数

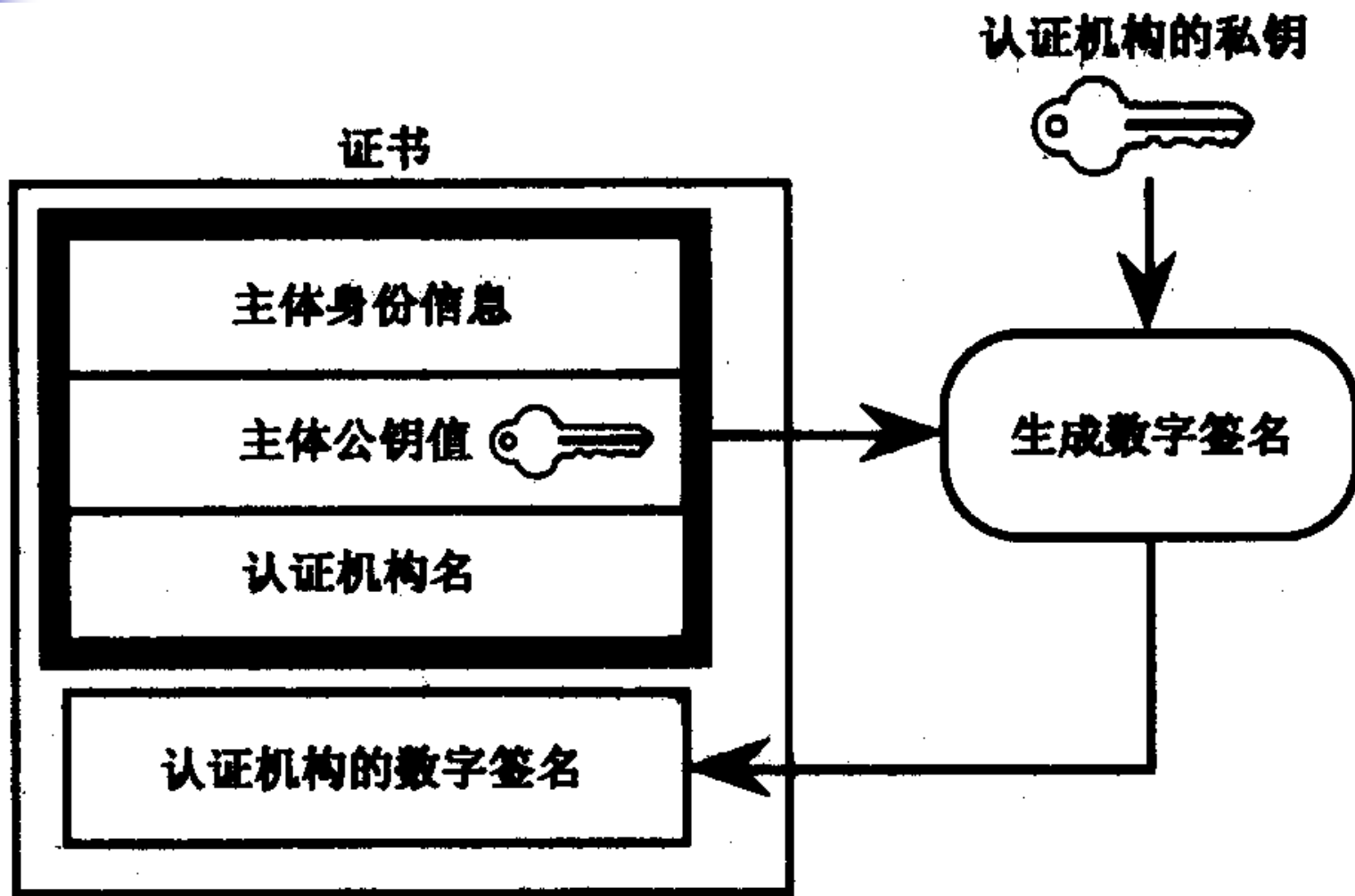
- 单向散列函数（**hash**，杂凑函数）**MAC**的一种变形，可以从一段很长的报文中计算出一个固定长度的比特串，这种散列函数通常称为报文摘要（**message digest**），用于消息的完整性检验。
- 单向散列函数有以下特性：
 - 给定 **P**，易于计算出 **MD (P)**
 - 只给出 **MD (P)**，几乎无法找出 **P**
 - 无法生成两条具有同样报文摘要的报文
- 认证方法：

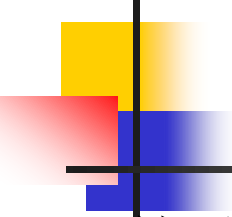
■ 消息	X	任意比特
■ 消息摘要	$Z=H(X)$	160比特
■ 签名摘要	$Y=\text{sig}_k(Z)$	320比特
- 标准：**MD5**（128比特），**SHA-1**（160比特）等

The screenshot shows a Windows XP desktop with a certificate export wizard. The 'Exportable' checkbox is checked. The 'Exportable Contents' list shows 'Certificate Policies' selected. The 'Exportable Policies' list shows 'Certificate Policy: PolicyIdent...' selected. The 'Exportable Policies' list is highlighted with a yellow circle.



数字证书





公钥基础设施 (Public Key Infrastructure)

- 什么是PKI呢？
 - PKI是一个用非对称密码算法原理和技术来实现并提供安全服务的具有通用性的安全基础设施。能够为所有网络应用提供采用加密和数字签名等密码服务所需要的密钥和证书管理。
- 为什么需要PKI？
 - 电子政务、电子商务对信息传输的安全需求
 - 在收发双方建立信任关系，提供身份认证、数字签名、加密等安全服务
 - 收发双方不需要共享密钥，通过公钥加密传输会话密钥
- 核心组成部分-CA (Certificate Authority)
 - 值得信赖的，将公开密钥与其对应的实体进行绑定



Chapter 10网络安全概要

- 10.1概述
- 10.2密码学基础知识
- 10.3数字签名与认证
- **10.4典型的网络安全威胁**
- 10.5网络安全协议课程体系介绍



10.4典型的网络安全威胁

- 1.主动攻击与被动攻击
- 2.常见攻击分类

1. 主动攻击和被动攻击

- 主动攻击

- 伪装

- 重放

- 篡改

- 拒绝

- 被动攻击

- 报文分析

- 流量分析

即冒名顶替。一般而言，伪装攻击的同时先被动地窃取通信数据，然后主动攻击。

即修改报文的内容。或者对截

阻止或占据对通信设施的正常使用或管理。针对特定目标或是某个网络区域

窃听和分析所传输的报文内容

分析通信主机的位置、通信的频率、报文长度等信息

得传输信息。



2. 常见攻击分类

- **口令破解**：攻击者可通过获取口令文件，然后运用口令破解工具获得口令，也可通过猜测或窃听等方式获取口令
- **连接盗用**：在合法的通信连接建立后，攻击者可通过阻塞或摧毁通信的一方来接管已经过认证建立起来的连接，从而假冒被接管方与对方通信
- **服务拒绝**：攻击者可直接发动攻击，也可通过控制其它主机发起攻击，使目标瘫痪，如发送大量的数据洪流阻塞目标
- **网络窃听**：网络的开放性使攻击者可通过直接或间接窃听获取所需信息
- **数据篡改**：攻击者可通过截获并修改数据或重放数据等方式破坏数据的完整性



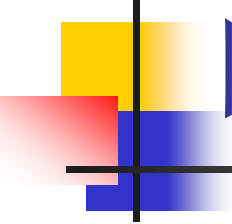
常见攻击分类（续）

- **地址欺骗：**攻击者可通过伪装成被信任的IP 地址、邮件地址等方式来骗取目标的信任
- **社会工程：**攻击者可通过各种社交渠道获得有关目标的结构、使用情况、安全防范措施等有用信息从而提高攻击成功率
- **恶意扫描：**攻击者可编制或使用现有扫描工具发现目标的漏洞，进而发起攻击
- **基础设施破坏：**攻击者可通过破坏DNS 或路由信息等基础设施，使目标陷于孤立
- **数据驱动攻击：**攻击者可通过施放病毒、特洛伊木马、数据炸弹等方式破坏或遥控目标



Chapter 10网络安全概要

- 10.1概述
- 10.2密码学基础知识
- 10.3数字签名与认证
- 10.4典型的网络安全威胁
- **10.5网络安全协议课程体系介绍**



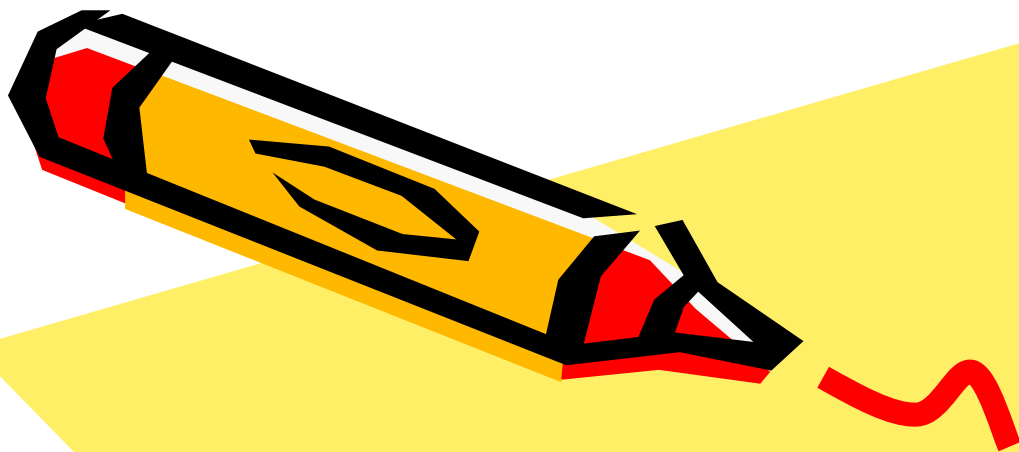
网络安全协议课程体系介绍

- 第一章：网络安全综述
- 第二章：公钥基础设施PKI
- 第三章：IPSec-AH和ESP
- 第四章：IPSec-IKE
- 第五章：SSL/TLS基本协议
- 第六章：防火墙
- 第七章：虚拟专用网VPN
- 第八章：应用层安全协议
- 第九章：WLAN的安全



小结

- 掌握信息、网络安全的基本概念。



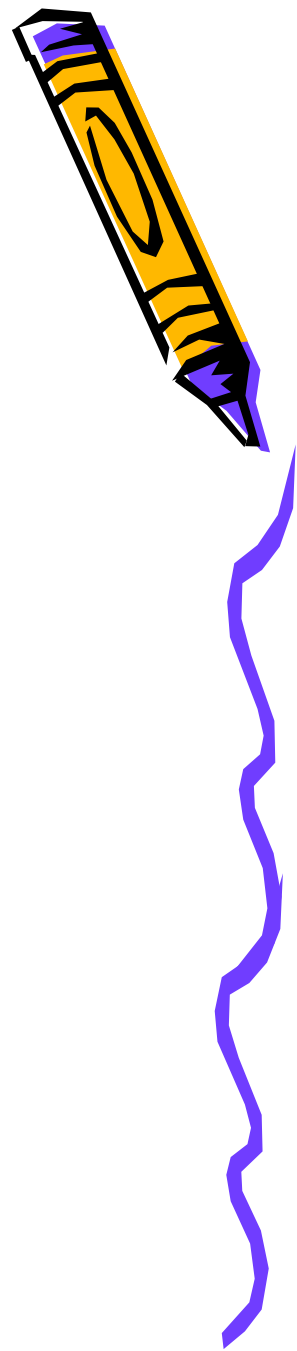
计算机网络复习提纲

14 February 2021



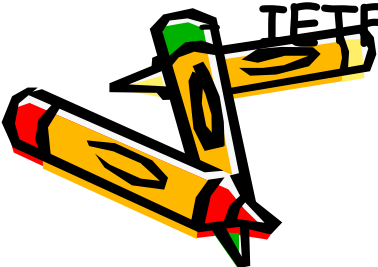
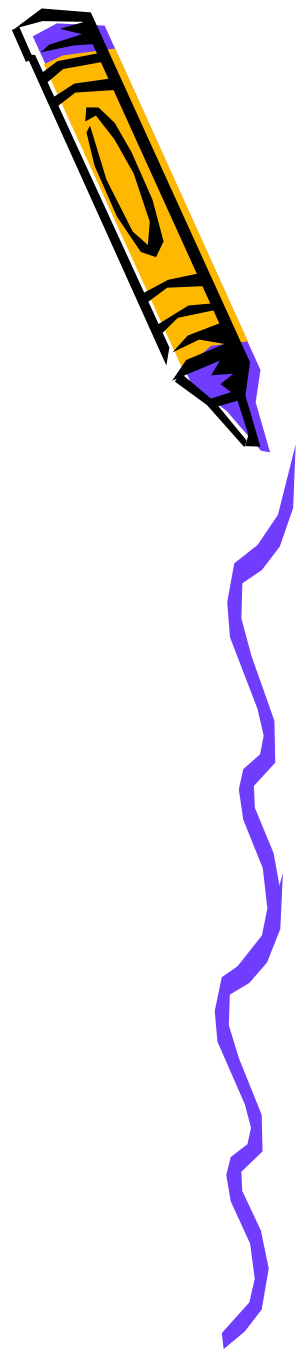
课程提纲

1. 概述
2. 网络的体系结构与参考模型
3. 物理层
4. 数据链路层
5. 局域网-以太网和WIFI
6. 网络层
7. Internet & TCP/IP
8. 传输层
9. 应用层
10. 网络安全



Chapter 1 概述

- 1.1 国际**Internet** 的发展史
- 1.2 中国**Internet**的发展史
- 1.3 **IPv6**发展史
 - **IP**地址是做什么用的
 - 为什么需要有**IPv6**
- 1.4 **Internet**的应用
 - 常用业务：文件、视频、交互消息
 - 移动互联网
- 1.5 网络的标准和标准化组织
 - **ITU**、**ISO**、**IEEE**
IETF、**IRTF**



Chapter 2 网络的体系结构与参考模型

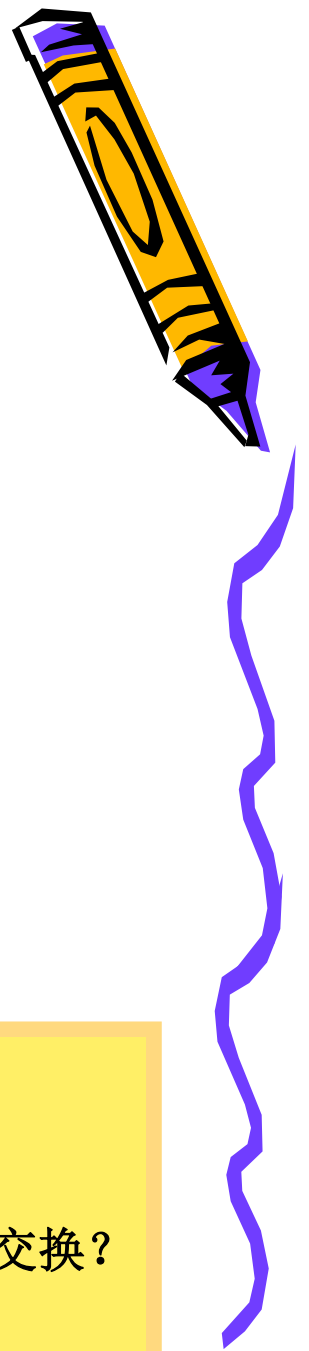


- 2.1 网络的构成
 - 端系统、通信链路、交换系统
 - **Internet**: 接入网、主干网
- 2.2 网络的体系结构
 - 分层思想
 - 层、协议栈、接口
- 2.3 **OSI**模型: 数据链路层、网络层、传输层功能
- 2.4 **TCP/IP**模型: 各种常见协议和功能
- 2.5 **OSI**与**TCP/IP**模型比较
- 2.6 网络的分类

1. 体系结构, 层、协议、服务、原语
2. 两种参考模型及其特点比较
3. 各层主要功能
4. 非连接与面向连接的服务



Chapter 3 物理层



- 3.1 数据通信的理论基础
 - 傅里叶级数
 - 数字信号、模拟信号
 - 最大传输速率（**Nyquist**定理、**Shannon**定理）
- 3.2 常见的传输介质
- 3.3 数字解调和多路复用
 - 调制与解调
 - 复用技术
- 3.4 交换技术
 - 电路交换、报文交换、分组交换
 - 四种时延：传输、传播、处理、排队



1. **Nyquist**定理， **Shannon**定理

2. 传输介质

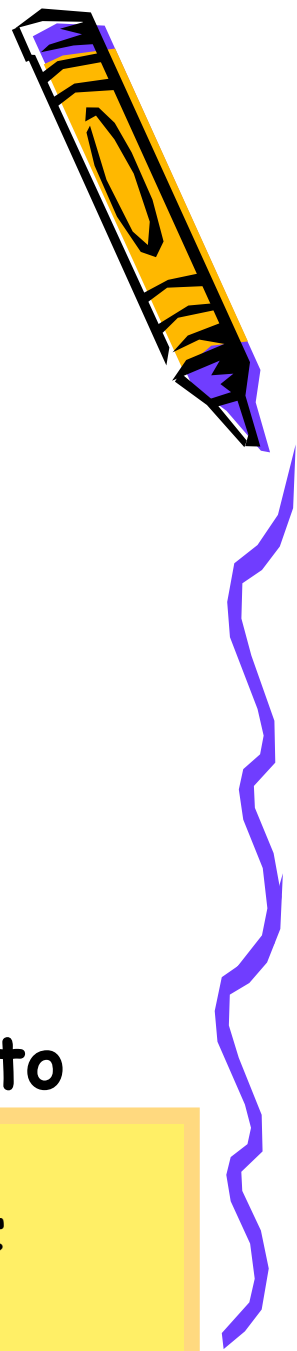
3. 什么是电路交换、报文交换、分组交换？

4. 什么是虚电路与数据报？

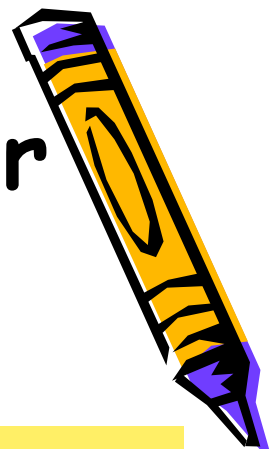
Chapter 4 数据链路层

- 4.1 数据链路层的功能
 - 给网络层提供服务
 - 面向连接，无连接有确认，无连接无确认
 - 成帧、错误控制、流控
- 4.2 差错检测与校正
 - 检错码与纠错码，**CRC**
- 4.3 基本数据链路协议
- 4.4 滑动窗口（Slide Windows）协议
- 4.5 面向位的协议**HDL**C
- 4.6 面向字节的数据链路层协议-**PPP** (Point to Point Protocol)

1. 数据链路层基本功能
2. 检错码与纠错码，**CRC**
3. 超时重传
4. 流量控制与滑动窗口



Chapter 5 LAN & MAC Sub layer



- 5.1 传统LAN的基本概念

- 传输媒体、拓扑结构

1. 传统局域网特征，拓扑结构

- 5.2 多路访问协议

- CSMA/CD

2. 媒体访问控制方法

3. 以太网（共享媒体/CSMA/CD，交换型）、最大帧长、最小帧长的基本概念以及设置帧长的理由

- 5.3 以太网的MAC协议

- 5.4 无线LAN

- DCF和PCF、掩藏终端和暴露站问题、CSMA/CA

- 5.5 网桥

- 广播域、碰撞域(冲突域)

4. 无线LAN基本概念（隐藏/暴露终端问题，CSMA/CA），802.11规范（DCF/PCF，间隔，地址）

- 5.6 交换式局域网

5. 网桥、交换机工作原理



Chapter 6 Network layer



- 6.1 网络层向传输层提供的服务
- 6.2 虚电路与数据报
- 6.3 路由算法
 - 距离矢量（DV）、链路状态（LS）
 - 分级路由
- 6.4 拥塞控制
 - 流量整形算法：漏桶、令牌桶
- 6.5 网络互连
 - 互联设备：交换机、路由器

1. 网络层向传输层提供的服务

- 虚电路（面向连接）
- 数据报（无连接）

2. 路由算法

- 静态路由、动态路由
- 链路状态、距离矢量
- 层次化路由（又称分层路由）

3. 拥塞控制

- 拥塞控制和流量控制的区别
- 漏桶、令牌桶

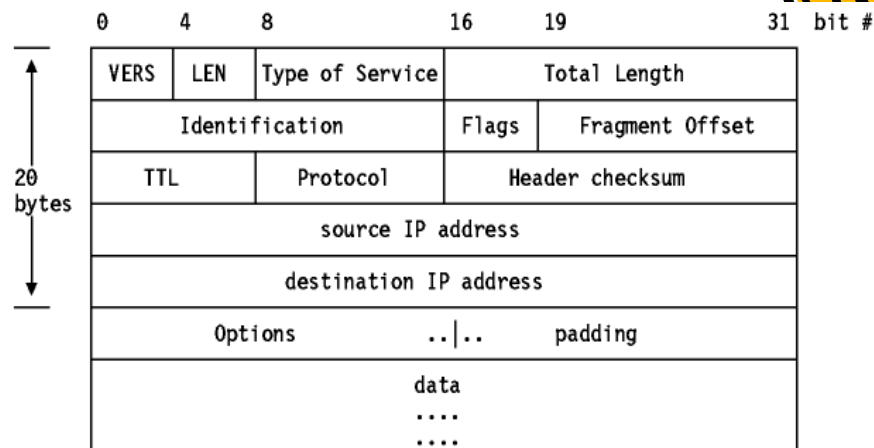
4. 互连设备、作用

- 从物理层到应用层都有哪些经典的互连设备，它们的转发机制？
- 哪些设备能隔离碰撞域、广播域



Chapter 7 Internet & TCP/IP

- 7.1 Internet基本协议栈
 - IP/TCP/UDP
- 7.2 IP地址
- 7.3 地址解析协议
 - ARP/RARP
- 7.4 IP协议
- 7.5 IP转发和路由
- 7.6 ICMP协议
 - Ping、Traceroute、ping -r基本原理
- 7.7 IP组播（简单了解）
- 7.8 IPv4的可扩展性
 - 三种过程中的解决方案:划分子网、CIDR及NAT
- 7.9 IPv6协议（简单了解）
 - 与IPv4协议的区别

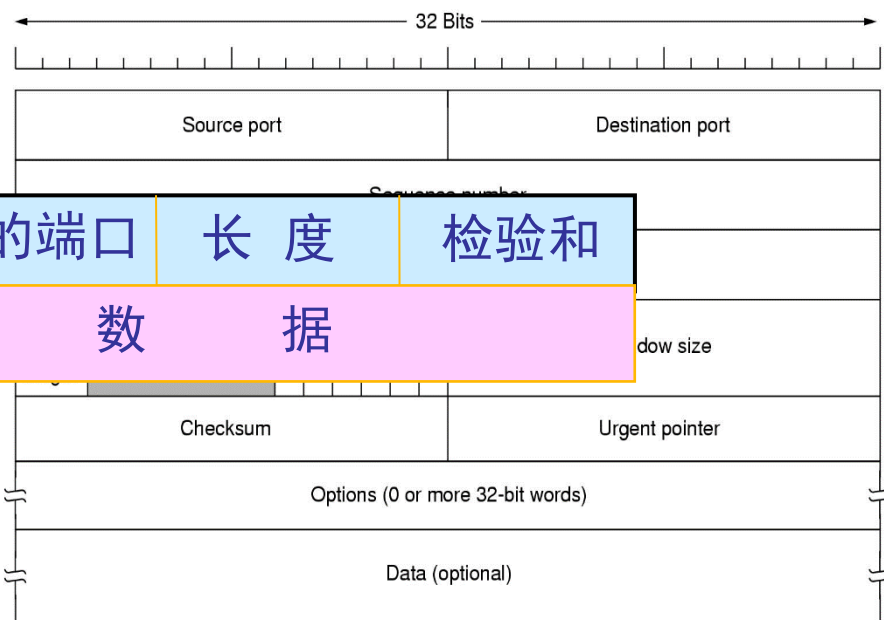


1. TCP/IP基本协议栈结构
2. IP地址和MAC地址的区别、ARP原理
3. IP为什么需要分段及怎样分段？
4. IPv4和IPv6地址类型，IPv6与IPv4的比较、ICMP协议功能
5. 划分子网、CIDR及NAT的原理
 1. 地址分配与地址聚类
 2. 路由器对IP头的处理
6. 路由发现和分组转发、距离矢量和链路状态路由协议
7. IPv6地址

Chapter 8 传输层



- 8.1 传输层服务
 - TCP和UDP, 为什么更存在正
- 8.2 传输层寻址
- 8.3 建立连接
- 8.4 Internet中的传输层协议
 - 用户数据报协议UDP
 - 传输控制协议TCP
 - 建立连接、释放连接
 - 流量控制、拥塞控制
- 8.5 Berkeley Socket
 - 实验



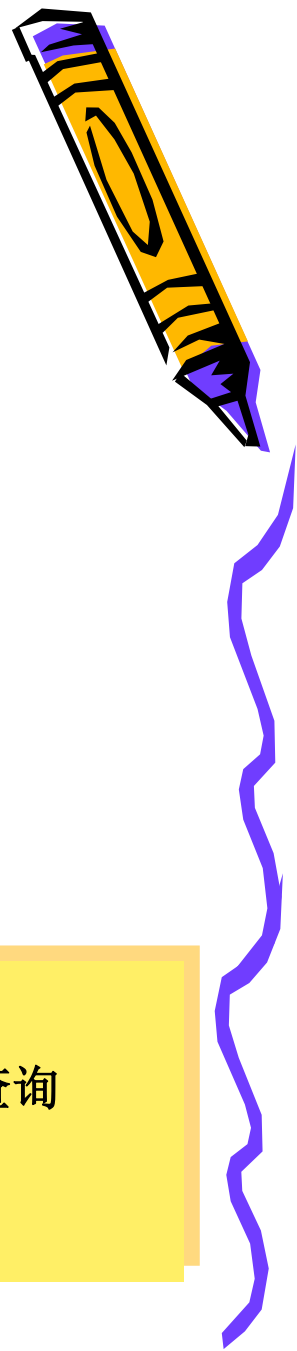
1. 传输层实现不同主机上的应用进程之间的通信，通过端口寻址
2. 为什么要在IP的基础上引入UDP?
3. TCP中数据流的概念
 - 面向字节流的可靠服务
4. TCP中的流量控制及拥塞控制机制
5. 套接口的基本概念及套接口编程



Chapter 9 应用层

- 9.1 网络的计算和访问模式
- 9.2 域名系统(DNS)
 - 迭代查询和递归查询
- 9.3 WWW(HTTP)
 - http和TCP连接的关系、URL、HTML
- 9.4 电子邮件
 - 相关协议和基本传送过程

1. 客户服/务器模式
2. DNS中的迭代查询和递归查询
3. 电子邮件相关协议
 - MIME, PoP3, SMTP, IMAP



Chapter 10 网络安全



- 10.1 概述
 - 网络安全模型和网络访问安全模型
- 10.2 密码学基础知识
 - 基本对称和非对称算法，算法举例
 - 流密码和分组密码（了解基本概念即可）
- 10.3 数字签名与认证
 - 源认证和数据认证
 1. 网络安全模型和网络访问安全模型
 2. 主动攻击和被动攻击的举例
- 10.4 典型的网络安全威胁
 - 主动攻击和被动攻击
 3. 三类常用的密码学算法：
 - 对称加密、分对称加密和哈希函数
 4. 签名的流程
 5. 加密算法用于认证：基于挑战-应答
 6. 密钥协商的常用方法：DH和数字信封
 7. PKI和证书的基本构成



考试安排



- 考试类型
 - 填空（**20分**）、选择（单选和不定项选择，**20分**）、计算和问答（**60分**)
 - 填空、选择部分会涉及与实验相关内容，约**4-8分**
 - 常用名词和概念的中英文及缩写
- 考试时间、地点
 - 2021年1月21日，14:30-16:30
 - 3C103
- 答疑时间、地点
 - 1月11日上午9:30-11:30，安排一个教室
 - 1月18日上午9:30-11:30，科技楼809、905





谢谢大家对课程的支持！

欢迎有志于从事网络通信、网络安全领域研究的同学

加入信息网络实验室（未来网络实验室的一部分）！

要求：1、计算机网络考试成绩>85（基本要求）， or

自信具有较好的编程能力、写作能力、逻辑思维能力或数学基础

2、踏实、肯干，学有余力

3、打算校内保研or出国

（后续只给在实验室有实际科研经历的同学提供推荐信）

联系方式：kpxue@ustc.edu.cn

实验室主页：<http://if.ustc.edu.cn>

个人主页：<http://staff.ustc.edu.cn/~kpxue>

