

```
title: 大学课程 | 计算机网络学习笔记
tags:
  - 网络
  - 大学课程
categories:
  - 学习笔记
abbrlink: 28758
reward: true
copyright: true
date: 2019-08-28 18:54:27
cover: https://npm.elemecdn.com/justlovesmile-img/1584110553-u34116618882606377503fm11gp0.jpg
top_img: https://npm.elemecdn.com/justlovesmile-img/1584110553-u34116618882606377503fm11gp0.jpg
```

作者博客: [Justlovesmile's BLOG](#)

大二计算机网络课程笔记

第一章 概述

1.1 计算机网络在信息时代中的作用

1. 21世纪的重要特征: 数字化, 网络化, 信息化
2. 21世纪是以网络为核心的信息时代
3. 常见三类网络: 电信网络, 有线电视网络, 计算机网络
4. Internet 的中文译名:
 - (1) 因特网
 - (2) 互联网: 由数量极大的各种计算机网络互连起来
5. 互联网具有两个重要的特点: **连通性和共享**
 - (1) 连通性: 互联网使上网用户之间不管相距多远, 都可以非常方便的交换各种信息, 好像这些用户中端是直接连通的
 - (2) 共享: 就是指资源共享, 资源包括信息, 软件, 硬件
6. 互联网+

1.2 互联网概述

1.2.1 网络的网络

1. 计算机网络: 由若干**结点**和连接这些结点的**链路**组成。

- 结点包括: 计算机, 集线器, 交换机, 路由器
- 2. **互连网**: 网络之间通过路由器互连起来, 就构成了一个覆盖范围更大的计算机网络, 因此互连网是“网络中的网络”
- 3. 网络: 网络把许多计算机连接在一起, 而互连网则把许多网络通过路由器连接在一起。与网络相连的计算机称为**主机**。
- 4. 计算机网络的三种定义:
 - 广义观点: 实现远程信息处理的系统或能进一步达到资源共享的系统
 - 资源共享观点: 以能够相互共享资源的方式互联起来的自治计算机系统的集合
 - 用户透明性观点: 存在一个能为用户自动管理资源的网络操作系统, 它能够调用用户所需要的资源, 而整个网络就像一个大的计算机系统一样对用户是透明的

1.2.2 互联基础结构发展的三个阶段

1. 第一阶段: **单个网络ARPANET**向互连网发展的过程

- 互联网的雏形: 单个的分组交换网ARPANET
- 1983年TCP/IP协议成为ARPANET的标准协议
- internet (互连网): 泛指由多个计算机网络互连而成的计算机网络
- Internet (互联网): 指当前全球最大的, 开放的, 由众多网络相互连接而成的特定互连网, 它采用TCP/IP协议族作为通信的规则, 且前身为美国的ARPANET
- 2. 第二阶段: **三级结构的互联网**
- 三级计算机网络: 主干网, 地区网, 校园网 (或企业网)
- 3. 第三阶段: 形成了**多层次的ISP结构的互联网**
- 互联网服务提供者ISP
- 流程:
 - (1) ISP向互联网管理机构申请很多IP地址, 同时拥有通信线路以及路由器等连网设备
 - (2) 任何机构和个人只要向某个ISP交纳规定的费用, 就可以从该ISP获取所需IP地址的使用权, 并可通过该ISP接入到互联网
- ISP层次: 主干ISP, 地区ISP, 本地ISP
- 4. **互联网交换点IXP**:
 - 作用: **允许两个网络直接相连并交换分组, 而不再需要通过第三个网络来转发分组。**
- IXP常采用工作在数据链路层的网络交换机, 这些网络交换机都用局域网互连起来
- 5. **万维网WWW**:

万维网 (亦作“Web”、“WWW”、“W3”, 英文全称为“World Wide Web”), 是一个由许多互相链接的超文本组成的系统, 通过互联网访问。在这个系统中, 每个有用的事物, 称为一样“资源”; 并且由一个全局“统一资源标识符” (URI) 标识; 这些资源通过超文本传输协议 (Hypertext Transfer Protocol) 传送给用户, 而后者通过点击链接来获得资源。万维网联盟 (英语: World Wide Web Consortium, 简称W3C), 又称W3C理事会。1994年10月在麻省理工学院 (MIT) 计算机科学实验室成立。万维网联盟的创建者是万维网的发明者蒂姆·伯纳斯-李。

---百度百科

1.2.3 互联网的标准化工作

- 互联网协会 (ISOC)
 - 互联网体系结构委员会IAB

- 互联网工程IETF
 - 互联网工程指导小组IESG：主要针对协议的开发和标准化
- 互联网研究部IRTF
 - 互联网研究指导小组IRSG：（RFC“请求评论”）
 - 指定互联网标准的三个阶段：
 - (1) 互联网草案
 - (2) 建议标准
 - (3) 互联网标准

1.3 互联网的组成

- 从工作方式上看：
 - (1) 边缘部分：用户直接使用的部分（主机）
 - (2) 核心部分：为边缘部分提供服务的（连通性和交换）（网络和路由器）
- 从组成成分来看：
 - (1) 硬件
 - (2) 软件
 - (3) 协议
- 从功能组成上来看：
 - (1) 通信子网（数据传输，交换，控制，存储）
 - (2) 资源子网（实现硬件，软件，数据资源共享的设备及其软件的集合）

1.3.1 互联网的边缘部分

- 端系统：处在互联网边缘的部分就是连接在互联网上的所有的主机。这些主机又称为端系统
- 端系统之间的通信方式：客户-服务器方式(C/S)；对等方式 (P2P)
 - 1.客户-服务器方式：

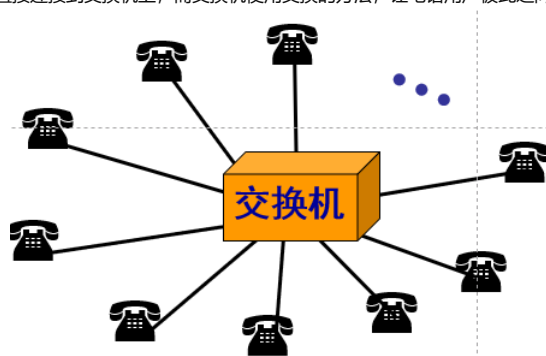
主机A运行客户程序<----->主机B运行服务器程序
- A向B发出服务请求，B向A提供服务
- 客户是服务的请求方，服务器是服务的提供方
- 对于客户程序：
 - (1) 被客户调用后运行，在通信时主动向远地服务器发起通信，因此客户程序必须知道服务器程序的地址
 - (2) 不需要特殊的硬件和很复杂的操作系统
- 对于服务器程序：
 - (1) 是一种专门来提供某种服务的程序，可同时处理多个客户请求
 - (2) 系统启动后自动调用并不断运行，被动等待接受来自各地的客户的请求，因此服务器不需要知道客户程序的地址
 - (3) 一般需要有强大的硬件和高级的操作系统支持
- 2.对等连接方式：P2P
- P2P方式：只要两台主机都运行了对等连接软件（P2P软件），就能进行平等的，对等连接通信

1.3.2 互联网的核心部分

- 在网络中心起特殊作用的是**路由器**，它是一种专用计算机（但不叫主机）
- 路由器是实现**分组交换**的关键构件，其任务是**转发收到的分组**，这是网络核心部分最重要的功能
- 互联网的核心部分是由许多网络和把它们互连起来的路由器组成，而主机处在互联网的边缘部分
- 互联网核心部分中的路由器之间一般是用高速链路连接，而在网络边缘的主机接入到核心部分则通常以相对较低速率的链路相连接

1. 电路交换的特点

- N部电话机两两直接相连，需 $N(N-1)/2$ 对电线。这种直接连接方法所需要的电线对的数量与电话机数量的平方（ N^2 ）成正比。
- 当电话机的数量增多时，就要使用**交换机**来完成全网的交换任务。
- **电路交换** (circuit switching)：每一部电话都直接连接到交换机上，而交换机使用交换的方法，让电话用户彼此之间可以很方便地通信。



- 交换：按照某种方式动态的分配传输线路的资源
- 电路交换必定是**面向连接的**
- 电路交换的三个阶段：
 - (1) **建立连接**：（占用通信资源）
 - (2) **通信**：（一直占用）
 - (3) **释放连接**：（归还通信资源）
- 在通话的全部时间内，通话的两个用户始终占用端到端的通信资源（利用率低）
- 计算机数据具有突发性
- 2. 分组交换的主要特点：
 - 分组交换网采用**存储转发技术**
 - 分组交换是面向无连接的
 - Internet网络层是分组交换
 - **报文**：要发送的整块数据
 - 在发送端，先把较长的报文划分成较短的，固定长度的**数据段**
 - 在每一个数据段前面添加上**首部**构成**分组**，分组又叫“包”，而分组的首部也可以称为“包头”
 - 主机是为用户进行信息处理的；路由器则是用来转发分组的，即进行分组交换

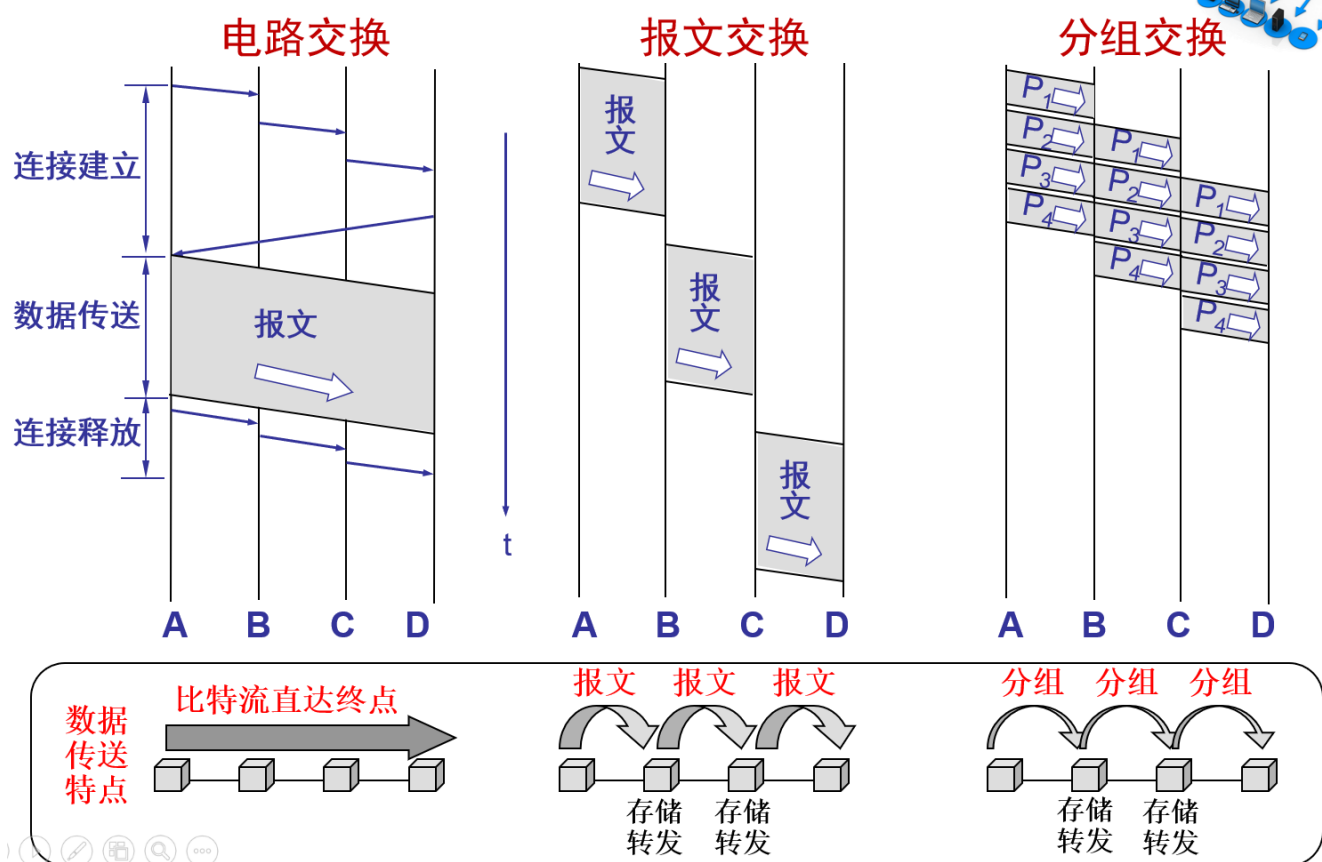
- 分组是在互联网中传送的**数据单元**。依次把各分组发送到接收端
- 分组的首部包含了目的地址，源地址等重要控制信息。
- 分组交换网中的结点交换机根据收到的分组首部的地址信息，把分组转发到下一个结点交换机
- 每一个分组在互联网中独立地选择传输路径
- 接收端收到分组后剥去首部还原成报文
- 路由器处理分组的过程：
 - (1) 把收到的**分组先放入缓存**（暂时存储）；
 - (2) **查找转发表**，找出到某个目的地址应从哪个端口转发；
 - (3) 把分组送到适当的**端口转发**出去。

优点	所采用的手段
高效	在分组传输过程中动态分配传输带宽，对通信链路是逐段占用
灵活	为每一个分组独立地选择最合适的转发路由
迅速	以分组作为传输单位，可以不先建立连接就能向其他主机发送分组
可靠	保证可靠性的网络协议；分布式多路由的分组交换网，使网络有很好的生存性

- 分组交换带来的问题：
 - (1) 分组在各结点存储转发时需要**排队**，这会造成一定的**时延**。
 - (2) 分组必须携带的首部（里面有必不可少的控制信息）也造成了一定的**开销**。

3. 三种交换的比较：

三种交换的比较



- 电路交换：整个报文的比特流连续地从源点直达终点，好像在一个管道中传送（面向连接）
- 报文交换：整个报文先传送到相邻结点，全部存储下来后查询转发表，转发到下一个结点。（面向无连接）
- 分组交换：单个分组传送到相邻结点，存储下来后查找转发表，转发到下一个结点（面向无连接）

1.4 计算机网络在我国的发展

- 规模最大的就是下面这五个：
 - (1) 中国电信互联网 CHINANET（也就是原来的中国公用计算机互联网）
 - (2) 中国联通互联网 UNINET
 - (3) 中国移动互联网 CMNET
 - (4) 中国教育和科研计算机网 CERNET
 - (5) 中国科学技术网 CSTNET

1.5 计算机网络的类别

1.5.1 计算机网络的定义

- 定义：计算机网络主要是由一些**通用的、可编程的硬件**互连而成的，而这些硬件并非专门用来实现某一特定目的（例如，传送数据或视频信号）。这些**可编程的硬件**能够用来传送多种不同类型的数据，并能支持广泛的和日益增长的应用。
- 根据定义：
 - (1) 计算机网络所连接的硬件，并不限于一般的计算机，而是包括了智能手机。

- (2) 计算机网络并非专门用来传送数据，而是能够支持很多种的应用（包括今后可能出现的各种应用）
- 可编程的硬件表明这种硬件一定包含有中央处理机CPU

1.5.2 计算机网络的类别

1. 按照网络的作用范围进行分类：
 - (1) 广域网WAN
 - (2) 城域网MAN
 - (3) 局域网LAN
 - (4) 个人区域网PAN- 无线个人区域网WPAN
- 若中央处理机之间的距离非常近（如仅1米的数量级甚至更小些），则一般就称之为**多处理机系统**，而不称它为计算机网络。
2. 按照网络使用者分类：
 - (1) 公用网
 - (2) 专用网
3. 用来把用户接入到互联网的网络
 - 接入网AN，它又称为本地接入网或居民接入网
 - 接入网是一类比较特殊计算机网络，用于讲用户接入互联网
 - 接入网本身既不属于互联网的核心部分，也不属于互联网的边缘部分。
 - 接入网是从某个**用户端系统到互联网中的第一个路由器**（也称为**边缘路由器**）之间的一种网络。

1.6 计算机网络的性能

1.6.1 计算机网络的性能指标

1. **速率**：
 - **比特** (bit) 是计算机中数据量的单位，也是信息论中使用的信息量的单位。
 - 速率是计算机网络中最重要的一個性能指标，指的是**数据的传送速率**，它也称为**数据率** (data rate) 或 **比特率** (bit rate)。
 - 速率的单位是 bit/s, 或 kbit/s、Mbit/s、Gbit/s 等。例如 4×10^{10} bit/s 的数据率就记为 40 Gbit/s。
 - **速率往往是指额定速率或标称速率，非实际运行速率。**
2. **带宽**：
 - 两种不同意义：
 - (1) “带宽”(bandwidth) 本来是指信号具有的频带宽度，其单位是赫（或千赫、兆赫、吉赫等）。（频域）
 - (2) 在计算机网络中，带宽用来表示网络中某**通道**传送数据的能力。表示在单位时间内网络中的某信道所能通过的“**最高数据率**”。单位是 bit/s，即“比特每秒”（时域）
3. **吞吐量**：
 - 吞吐量 (throughput) 表示在**单位时间内**通过某个网络（或信道、接口）的**数据量**。
 - 吞吐量更经常地用于对现实世界中的网络的一种测量，以便知道**实际上**到底有多少数据量能够通过网络。
 - 吞吐量受网络的带宽或网络的额定速率的限制。
4. **时延**：
 - 时延是指数据（一个报文或分组，甚至比特）从网络（或链路）的**一端传送到另一端**所需的时间。
 - 有时也称为延迟或迟延。
 - 网络中的时延由以下几个不同的部分组成：
 - (1) 发送时延（传输时延）：指主机或路由器发送数据帧所需要的时间

$$\text{发送时延} = \frac{\text{数据帧长度 (bit)}}{\text{发送速率 (bit/s)}}$$

- (2) 传播时延：是电磁波在信道中传播一定的距离需要花费的时间

$$\text{传播时延} = \frac{\text{信道长度 (米)}}{\text{信号在信道上的传播速率 (米/秒)}}$$

发送时延发生在**机器内部的发送器中**，与传输信道的长度无关，**传播时延**发生在**机器外部的传输信道媒体上**，与信号的发送速率无关。信号传送的距离越远，**传播时延就越大**

- (3) 处理时延
 - 主机或路由器在**收到分组时**，为处理分组（例如分析首部、提取数据、差错检验或查找路由）所花费的时间。
- (4) 排队时延
 - 分组在路由器输入输出队列中排队**等待处理所经历的时延**。
 - 排队时延的长短往往取决于网络中当时的通信量。

总时延 = 发送时延 + 传播时延 + 处理时延 + 排队时延

- 对于**高速网络链路**，我们提高的仅仅是数据的**发送速率**而不是比特在链路上的传播速率。
- 提高链路**带宽**减小了数据的**发送时延**。

5. 时延带宽积：

- 链路的时延带宽积又称为以比特为单位的链路长度。

$$\text{时延带宽积} = \text{传播时延} \times \text{带宽}$$

- 只有在代表链路的管道都充满比特时，链路才得到了充分利用。

6. 往返时间RTT:

- 互联网上的信息不仅仅单方向传输，而是双向交互的。因此，有时很需要知道双向交互一次所需的时间。
- 往返时间表示从发送方发送数据开始，到发送方收到来自接收方的确认，总共经历的时间。
- 在互联网中，往返时间还包括各中间结点的处理时延、排队时延以及转发数据时的发送时延。
- 发送时间=数据长度/发送速率
- 有效数据率=数据长度/（发送时间+RTT）

7. 利用率:

- 利用率分为**信道利用率**和**网络利用率**
- 信道利用率**不是**越高越好，根据排队论的理论，当某信道的**利用率增大**时，该信道引起的**时延也就迅速增加**。
- 若令 D_0 表示网络空闲时的时延， D 表示网络当前的时延，则在适当的假定条件下，可以用下面的简单公式表示 D 和 D_0 之间的关系：

$$D = \frac{D_0}{1 - U}$$

- 信道或网络利用率过高会产生非常大的时延

1.6.2 计算机网络的非性能指标

1. 费用
2. 质量
3. 标准化
4. 可靠性
5. 可扩展性和可升级性
6. 易于管理和维护

1.7 计算机网络体系结构

1.7.1 计算机网络体系结构的形成

- 计算机网络是个非常复杂的系统。
- 相互通信的两个计算机系统必须**高度协调工作**才行，而这种“协调”是相当复杂的。
- “分层”可将庞大而复杂的问题，转化为若干较小的局部问题，而这些较小的局部问题就比较易于研究和处理。
- 1974 年，美国的 IBM 公司宣布了**系统网络体系结构 SNA** (System Network Architecture)。这个著名的网络标准就是按照**分层**的方法制定的。
- 由于**网络体系结构的不同，不同公司的设备很难互相连通**。
- 为了使不同体系结构的计算机网络都能互连，国际标准化组织 ISO 于 1977 年成立了专门机构研究该问题。他们提出了一个试图使各种计算机在世界范围内互连成网的**标准框架，即著名的开放系统互连基本参考模型 OSI/RM** (Open Systems Interconnection Reference Model)，简称为**OSI**。
- 只要遵循 OSI 标准，一个系统就可以和位于世界上任何地方的、也遵循这同一标准的其他任何系统进行通信。
- 开放是指非独家垄断，系统是指在现实的系统中与互连有关的各部分
- OSI 只获得了一些理论研究的结果，在市场化方面却失败了
- **法律上的国际标准 OSI** 并没有得到市场的认可。非国际标准 TCP/IP 却获得了最广泛的应用。**TCP/IP** 常被称为**事实上的国际标准**。

1.7.2 协议与划分层次

- 计算机网络中的数据交换必须遵守事先约定好的规则。**这些规则明确规定了所交换的数据的格式以及有关的同步问题（同步含有序的意思）。**
- **网络协议** (network protocol)，简称为协议，是**为进行网络中的数据交换而建立的规则、标准或约定**。
- 协议主要由以下三个要素组成：
 - (1) 语法：数据与控制信息的结构或格式
 - (2) 语义：需要发出何种控制信息，完成何种动作以及做出何种响应
 - (3) 同步：事件实现顺序的详细说明
- 协议的两种形式：
 - (1) 一种是使用便于人来阅读和理解的**文字描述**。
 - (2) 另一种是使用让计算机能够理解的**程序代码**。
 - 这两种不同形式的协议都必须能够对网络上信息交换过程做出精确的解释。
- 分层的好处：
 - (1) 各层之间是独立的。
 - (2) 灵活性好。
 - (3) 结构上可分割开。
 - (4) 易于实现和维护。
 - (5) 能促进标准化工作

缺点：

- (1) 降低效率。
 - (2) 有些功能会在不同的层次中重复出现，因而产生了额外开销。
- 通常各层要完成的功能：
 - (1) 差错控制
 - (2) 流量控制
 - (3) 分段和重装
 - (4) 复用和分用
 - (5) 连接建立和释放
 - 计算机网络的**体系结构** (architecture) 是**计算机网络的各层及其协议的集合**。
 - 体系结构就是这个计算机网络及其部件所应完成的功能的精确定义。
 - **实现** (implementation) 是遵循这种体系结构的前提下用何种硬件或软件完成这些功能的问题。
 - **体系结构是抽象的，而实现则是具体的**，是真正在运行的计算机硬件和软件。

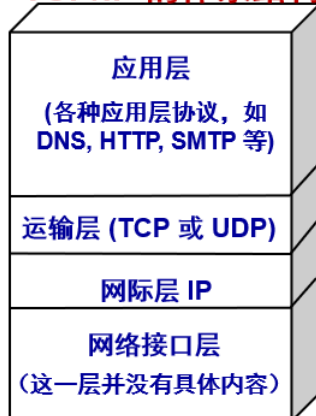
1.7.3 具有五层协议的体系结构

OSI 的体系结构



(a)

TCP/IP 的体系结构



(b)

五层协议的体系结构



(c)

计算机网络体系结构：

(a) OSI 的七层协议；(b) TCP/IP 的四层协议；(c) 五层协议

1. 应用层：

- 通过应用进程间的交互来完成特定网络应用
- 应用层协议定义的是应用程序间通信和交互的规则
- 例如：域名系统DNS，支持万维网应用的HTTP协议，支持电子邮件的SMTP协议
- 我们把应用层交互的数据单元称为报文

2. 运输层：

- 负责向两台主机中进程之间的通信提供通用的数据传输服务
- 两种主要协议：
 - 传输控制协议TCP：面向连接，其数据传输的单位是报文段
 - 用户数据报协议UDP：面向无连接，尽最大努力，数据传输的单位是用户数据报

3. 网络层：

- 负责为分组交换网上的不同主机提供通信服务
- 在发送数据时，网络层把运输层产生的报文段或用户数据报封装成分组或包进行传输
- 在TCP/IP协议中，分组称为IP数据报
- 互联网是由大量异构网络通过路由器相互连接起来的。
- 互联网使用的网络层协议是无连接的网络协议IP和许多种路由选择协议

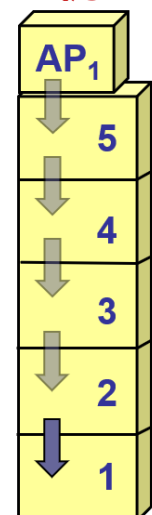
4. 数据链路层

- 数据链路层将网络层交下来的IP数据报封装成帧，在两个相邻结点间的链路上传送帧。每一帧包括数据和必要的控制信息
- 控制信息可以使接受端能检测到所接收的帧有无差错，有则丢弃。

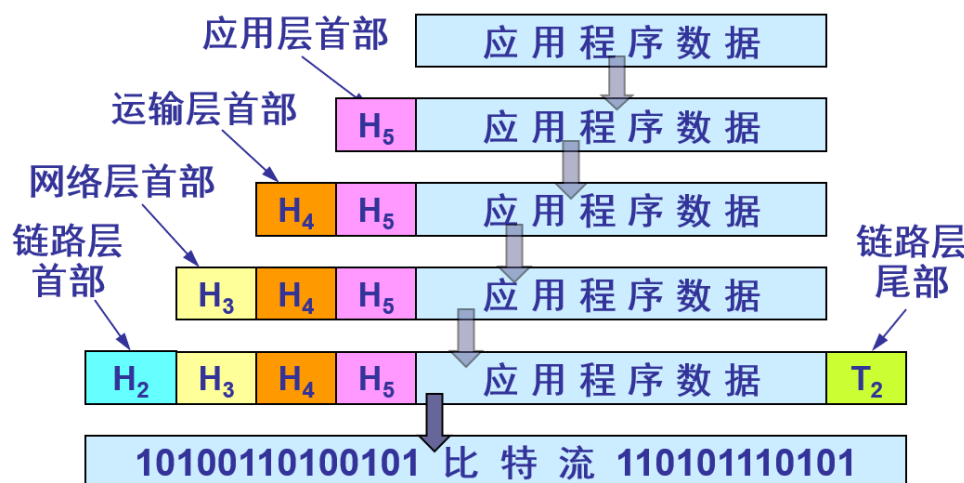
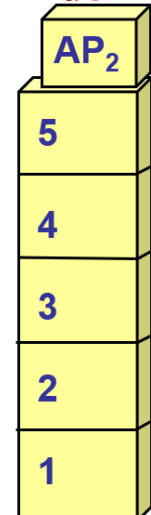
5. 物理层

- 在物理层上所传输的数据的单位是比特。

主机 1



主机 2

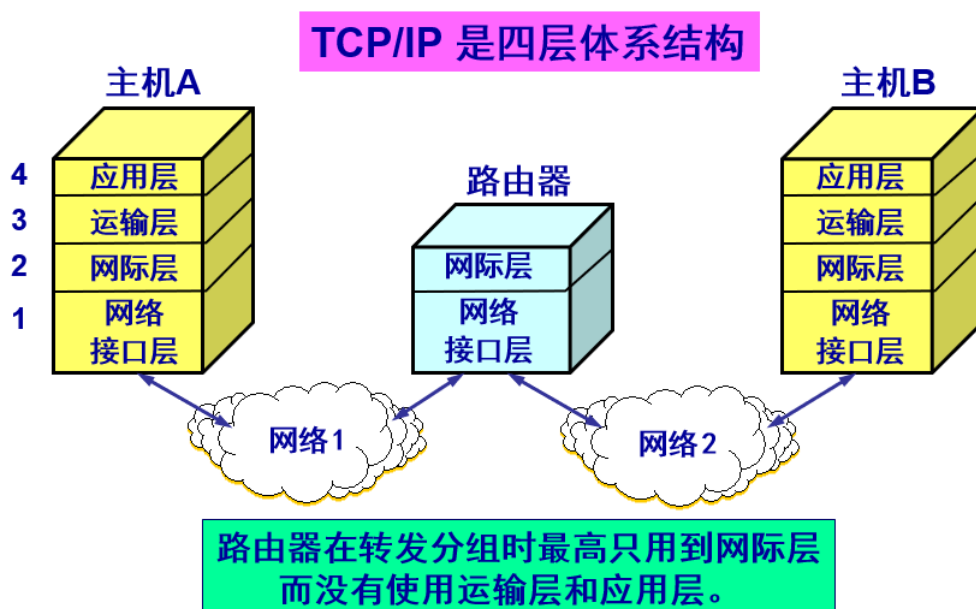


- PDU (Protocol Data Unit): 协议数据单元。
OSI 参考模型把对等层次之间传送的数据单位称为该层的协议数据单元 PDU。
- 任何两个同样的层次把数据（即数据单元加上控制信息）通过水平虚线直接传递给对方。这就是所谓的“对等层”(peer layers)之间的通信。
- 各层协议实际上就是在各个对等层之间传递数据时的各项规定。

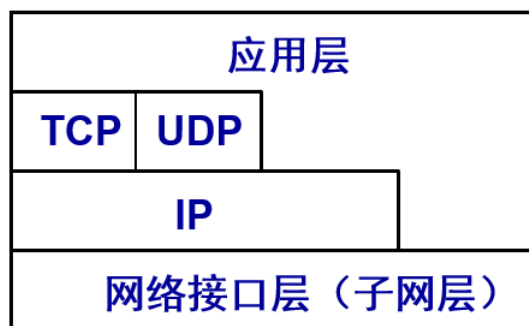
1.7.4 实体，协议，服务和访问点

- **实体** (entity) 表示任何可发送或接收信息的硬件或软件进程。
- **协议** 是控制两个对等实体进行通信的规则集合。
- 在协议的控制下，两个对等实体间的通信使得本层能够向上一层提供服务。
- 要实现本层协议，还需要使用下层所提供的服务。
- 协议的实现保证了能够向上一层提供服务。
- 本层的服务用户只能看见服务而无法看见下面的协议。即下面的协议对上面的服务用户是透明的。
- 协议是“水平的”，即协议是控制对等实体之间通信的规则。
- 服务是“垂直的”，即服务是由下层向上层通过层间接口提供的。
- 上层使用服务原语获得下层所提供的服务。
- 同一系统相邻两层的实体进行交互的地方，称为**服务访问点 SAP** (Service Access Point)。
- 服务访问点SAP是一个抽象的概念，它实际上就是一个逻辑接口。
- OSI把层与层之间交换的数据的单位称为服务数据单元 SDU (Service Data Unit)。
- SDU 可以与 PDU 不一样，例如，可以是多个 SDU 合成为一个 PDU，也可以是一个 SDU 划分为几个 PDU。
- 协议必须把所有不利的条件事先都估计到，而不能假定一切都是正常的和非常理想的。
- 看一个计算机网络协议是否正确，不能光看正常情况下是否正确，还必须非常仔细地检查这个协议能否应付各种异常情况。

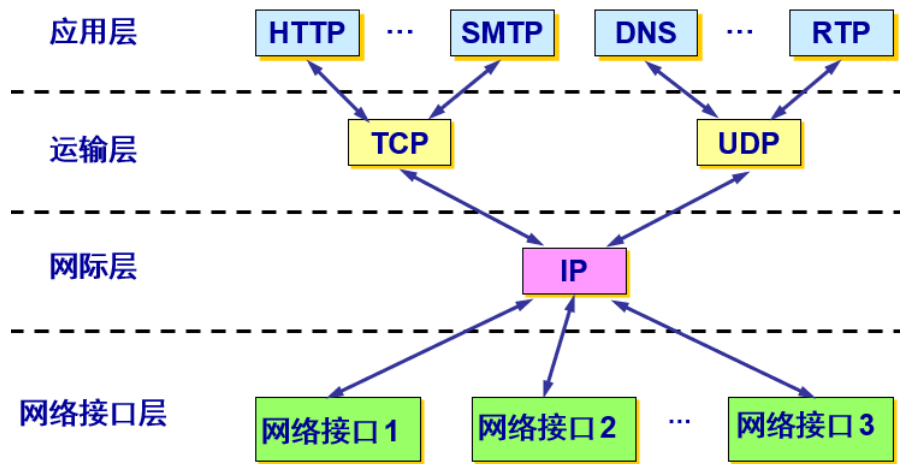
1.7.5 TCP/IP的体系结构



- 另一种表示方法:



- 还有一种表示方法:



沙漏计时器形状的 TCP/IP 协议族

- TCP/IP协议可以为各式各样的应用提供服务，允许IP协议在各式各样的网络构成的互联网上运行。

第二章 物理层

2.1 物理层的基本概念

- 物理层考虑的是怎样才能在连接各种计算机的传输媒体上传输数据比特流，而不是指具体的传输媒体。
- 物理层的作用是要尽可能地屏蔽掉不同传输媒体和通信手段的差异。用于物理层的协议也常称为物理层规程 (procedure)。
- 物理层的主要任务描述为确定与传输媒体的接口有关的一些特性：
 - 机械特性：指明接口所用接线器的形状和尺寸、引线数目和排列、固定和锁定装置等。
 - 电气特性：指明在接口电缆的各条线上出现的电压的范围。
 - 功能特性：指明某条线上出现的某一电平的电压表示何种意义。
 - 过程特性：指明对于不同功能的各种可能事件的出现顺序。
- 数据在计算机内部多采用并行传输方式，但在通信线路的传输方式一般都是串行传输

2.2 数据通信的基础知识

2.2.1 数据通信系统的模型

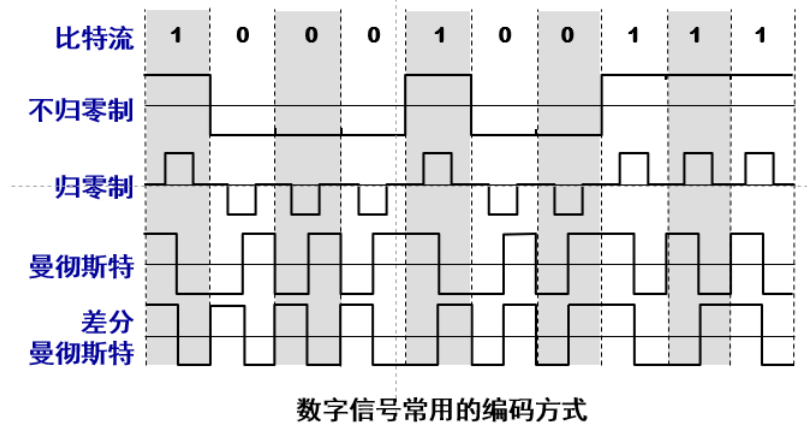
- 一个数据通信系统可以分为三个部分：
 - 源系统**（发送端，发送方）
 - 传输系统**（传输网络）
 - 目的系统**（接收端，接受方）
- 源系统包括：
 - 源点
 - 发送器
 - 接收器
 - 终点
- 常用术语：
 - 通信的目的是传送**消息**。
 - 数据**是运送消息的实体
 - 信号**则是数据的电气或电磁的表现
 - 模拟信号（**连续信号**）：消息的参数的取值是连续的
 - 数字信号（**离散信号**）：消息的参数的取值是离散的

2.2.2 有关信道的几个基本概念

- 信道：一般用来表示向某一个方向传送信息的媒体。
 - 单向通信（单工通信）：只能有一个方向的通信而没有反方向的交互
 - 双向交替通信（半双工通信）：通信的双方都可以发送信息，但不能双方同时发送与接收
 - 双向同时通信（全双工通信）：通信的双方可以同时发送和接收信息
- 基带信号**：来自信源的信号。
- 调制**：为了解决许多信道不能传输这种低频分量或直流分量
 - 基带调制**：仅对基带信号的波形进行变换，使它能够与信道特性相适应。**变换后的信号仍然是基带信号**。把这种过程称为**编码** (coding)。
 - 带通调制**：使用**载波** (carrier) 进行调制，把基带信号的频率范围搬移到较高的频段，并**转换为模拟信号**，这样就能够更好地在模拟信道中传输（即仅在一段频率范围内能够通过信道）。
 - 带通信号：经过载波调制后的信号。
- 常用编码方式：
 - 不归零制**：正电平代表 1，负电平代表 0。
 - 归零制**：正脉冲代表 1，负脉冲代表 0。

- (3) **曼彻斯特编码**：位周期中心的向上跳变代表 0，位周期中心的向下跳变代表 1。但也可反过来定义。
- (4) **差分曼彻斯特编码**：在每一位的中心处始终都有跳变。位开始边界有跳变代表 0，而位开始边界没有跳变代表 1。

(1) 常用编码方式



- 从信号波形中可以看出，曼彻斯特 (Manchester) 编码和差分曼彻斯特编码产生的信号频率比不归零制高。
- 从自同步能力来看，不归零制不能从信号波形本身中提取信号时钟频率（这叫作没有自同步能力），而曼彻斯特编码和差分曼彻斯特编码具有**自同步能力**。

5. 基本的带通调制方法

(1) 基带信号往往包含有较多的低频成分，甚至有直流成分，而许多信道并不能传输这种低频分量或直流分量。为了解决这一问题，就必须对基带信号进行**调制** (modulation)。

(2) 最基本的二元制调制方法有以下几种：

- 调幅(AM)**：载波的振幅随基带数字信号而变化。
 - 调频(FM)**：载波的频率随基带数字信号而变化。
 - 调相(PM)**：载波的初始相位随基带数字信号而变化
- (3) 正交振幅调制QAM

2.3 信道的极限容量

1. 码元传输的速率越高，或信号传输的距离越远，或传输媒体质量越差，在信道的输出端的波形的失真就越严重。

2. 限制码元在信道上的传输速率的因素：

(1) 信道能通过的频率范围：

码间串扰，奈奎准则

在任何信道中，码元传输的速率是有上限的，传输速率超过此上限，就会出现严重的码间串扰的问题，使接收端对码元的判决（识别）成为不可能

(2) 信噪比

信噪比 = 信号的平均功率/噪声的平均功率

常记为 S/N ，并用分贝 (dB) 作为度量单位。即：

信噪比(dB) = $10 \log_{10}(S/N)$ (dB)

例如，当 $S/N = 10$ 时，信噪比为 10 dB，而当 $S/N = 1000$ 时，信噪比为 30 dB。

- 1984年，香农 (Shannon) 用信息论的理论推导出了带宽受限且有高斯白噪声干扰的信道的**极限、无差错的信息传输速率（香农公式）**。
- 信道的极限信息传输速率 C 可表达为：

$$C = W \log_2(1+S/N) \quad (\text{bit/s})$$

- 其中： W 为信道的带宽 (以 Hz 为单位)；
 S 为信道内所传信号的平均功率；
 N 为信道内部的高斯噪声功率。
- 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。
- 只要信息传输速率低于信道的极限信息传输速率，就一定可以找到某种办法来实现无差错的传输。**
- 若信道带宽 W 或信噪比 S/N 没有上限（当然实际信道不可能是这样的），则信道的极限信息传输速率 C 也就没有上限。
- 实际信道上能够达到的信息传输速率要比香农的极限传输速率低不少。
- 对于频带宽度已确定的信道，如果信噪比不能再提高了，并且码元传输速率也达到了上限值，那么还有办法**提高信息的传输速率**。
- 这就是：**用编码的方法让每一个码元携带更多比特的信息量**。

2.3 物理层下面的传输媒体

- 传输媒体**也称为**传输介质**或**传输媒介**，它就是数据传输系统中在发送器和接收器之间的物理通路。
- 传输媒体可分为两大类，即**导引型传输媒体**和**非导引型传输媒体**。
- 在导引型传输媒体中，电磁波被导引沿着固体媒体（铜线或光纤）传播。
- 非导引型传输媒体就是指自由空间。在非导引型传输媒体中，电磁波的传输常称为无线传输。

2.3.1 导引型传输媒体

1. 双绞线：

- 最常用的传输媒体。

- 模拟传输和数字传输都可以使用双绞线，其通信距离一般为几到十几公里。
- **屏蔽双绞线** STP (Shielded Twisted Pair)
带金属屏蔽层
- **无屏蔽双绞线** UTP (Unshielded Twisted Pair)

室内传送数据的无屏蔽双绞线和屏蔽双绞线的标准 EIA/TIA-568。
布线标准更新为 EIA/TIA-568-A。
此标准规定了 5 个种类的 UTP 标准（从 1 类线到 5 类线）。
对传送数据来说，现在最常用的 UTP 是 5 类线（Category 5 或 CAT5）。

2. 同轴电缆：

- 同轴电缆具有很好的抗干扰特性，被广泛用于传输较高速率的数据。
- 同轴电缆的带宽取决于电缆的质量。
- 50 Ω 同轴电缆 —— LAN / 数字传输常用
- 75 Ω 同轴电缆 —— 有线电视 / 模拟传输常用

3. 光缆：

- 光纤是光纤通信的传输媒体。
- 由于可见光的频率非常高，约为 108 MHz 的量级，因此一个光纤通信系统的传输带宽远远大于目前其他各种传输媒体的带宽。
- 多模光纤
- 单模光纤
- 光线通信的优点：
 - (1) 通信容量非常大。
 - (2) 传输损耗小，中继距离长。
 - (2) 抗雷电和电磁干扰性能好。
 - (3) 无串音干扰，保密性好。
 - (4) 体积小，重量轻。

2.3.2 非引导型传输媒体

- 将自由空间称为“非引导型传输媒体”。
- 无线传输所使用的频段很广。紫外线和更高的波段目前还不能用于通信
- **短波通信**（即高频通信）主要是靠电离层的反射，但短波信道的通信质量较差，传输速率低。
- **微波**在空间主要是直线传播。
- 传统微波通信有两种方式：
 - 地面微波接力通信
 - 卫星通信：通信距离远，传播时延大

2.4 信道复用技术

2.4.1 频分复用，时分复用和统计时分复用

- **复用** (multiplexing) 是通信技术中的基本概念。它允许用户使用一个**共享信道**进行通信，降低成本，提高利用率。
- **频分复用FDM**：**频分复用的所有用户在同样的时间占用不同的带宽资源**（请注意，这里的“带宽”是频率带宽而不是数据的发送速率）。
- **时分复用**：
 - 时分复用则是将时间划分为一段段等长的时分复用帧（TDM 帧）。每一个时分复用的用户在每一个 TDM 帧中占用固定序号的时隙。
 - 每一个用户所占用的时隙是周期性地出现（其周期就是 TDM 帧的长度）。
 - TDM 信号也称为**等时**(isochronous)信号。
 - **时分复用的所有用户是在不同的时间占用同样的频带宽度**。
 - 使用时分复用系统传送计算机数据时，由于计算机数据的突发性质，用户对分配到的子信道的利用率一般是不高的。
- 在进行通信时，复用器和分用器成对的使用
- **统计时分复用STDM**：（异步时分复用）
STDM 帧不是固定分配时隙，而是按需动态地分配时隙。因此统计时分复用可以提高线路的利用率。
- **波分复用WDM**：波分复用就是光的频分复用。使用一根光纤来同时传输多个光载波信号。
- **码分复用CDM**：
 - 常用的名词是**码分多址** CDMA (Code Division Multiple Access)。
 - 各用户使用经过特殊挑选的不同码型，因此彼此不会造成干扰。
 - 这种系统发送的信号有很强的抗干扰能力，其频谱类似于白噪声，不易被敌人发现。
 - 每一个比特时间划分为 m 个短的间隔，称为**码片** (chip)。
 - 每个站被指派一个唯一的 m bit 码片序列。
 - 如发送比特 1，则发送自己的 m bit 码片序列。
 - 如发送比特 0，则发送该码片序列的二进制反码。
 - 例如，S 站的 8 bit 码片序列是 00011011。
 - 发送比特 1 时，就发送序列 00011011，
 - 发送比特 0 时，就发送序列 11100100。
 - S 站的码片序列：(-1 -1 -1 +1 +1 -1 +1 +1)
 - 假定 S 站要发送信息的数据率为 b bit/s。由于每一个比特要转换成 m 个比特的码片，因此 S 站实际上发送的数据率提高到 mb bit/s，同时 S 站所占用的频带宽度也提高到原来数值的 m 倍。

- 这种通信方式是**扩频**(spread spectrum)通信中的一种。
- 扩频通信通常有两大类：
 - 一种是**直接序列扩频DSSS** (Direct Sequence Spread Spectrum)，如上面讲的使用码片序列就是这一类。
 - 另一种是**跳频扩频FHSS** (Frequency Hopping Spread Spectrum)。
- 每个站分配的码片序列不仅**必须各不相同**，并且还**必须互相正交** (orthogonal)。
- 在实用的系统中是使用**伪随机码序列**。
- 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) 等于 0：
- 任何一个码片向量和该码片向量自己的规格化内积都是 1。
- 一个码片向量和该码片反码的向量的规格化内积值是 -1。

2.5 数字传输系统

- 与模拟通信相比，数字通信无论是在传输质量上还是经济上都有明显的优势
- 目前，长途干线大都采用PCM是数字传输方式
- 脉码调制PCM体制，最初是为了在电话局之间的中继线上传送多路的电话
- 由于历史原因PCM有两个不兼容的国际标准：T1，E1
- 当需要有更高的数据率时，可以采用复用的方法
- 旧的数字传输系统存在许多缺点：
 - (1)速率标准不统一
 - (2)不是同步传输
- **同步光纤网SONET**的各级时钟都来自一个非常精确的主时钟
- **同步数字系列SDH**，适用于微波和卫星传输

2.6 宽带接入技术

- 用户要接入互联网，必须先连接到某个ISP
- 美国联邦通信委员会FCC认为只要双向速率之和超过200 kbit/s就是**宽带**
- 从宽带接入的媒体来看，可以划分为两大类：
 - 有线宽带接入
 - 无线宽带接入

2.6.1 ADSL技术

- **非对称数字用户线ADSL**：用数字技术对现有的模拟电话用户线进行改造
- ADSL技术就是把0~4 kHz**低频频谱**给传统电话使用，而把**原来没有被利用的高端频谱**留给用户上网使用
- DSL就是**数字用户线**
- ADSL的传输距离取决于数据率和用户线的途径
- ADSL所得到的最高数据传输速率与实际的用户线上的信噪比密切相关
- ADSL的特点：
 - 上行（用户到ISP）和下行（ISP到用户）带宽不对称
 - ADSL在用户线的两端各安装一个ADSL调制解调器
 - 我国目前采用的方案是离散多音调DMT调制技术
 - 多音调：指多载波/多子信道
 - DMT技术：
 - 采用频分复用
 - 相当于在一对用户线上使用许多小的调制解调器并行的传送数据。
- ADSL的数据率：
 - ADSL采用自适应调制技术使用户线能够传送尽可能高的数据率
 - ADSL不能保证固定的数据率
- 第二代ADSL：
 - 通过提高调制效率得到更高的数据率
 - 采用无缝速率自适应技术

2.6.2 光线同轴混合网

- HFC网是在目前覆盖面很广的有线电视网CATV基础上开发的一种居民宽带接入网
- HFC网对CATV网进行了改造
- HFC网将原CATV网中的同轴电缆**主干部分该换为光纤**，并使用模拟光纤技术
- 在模拟光纤中采用光的振幅AM
- 模拟光纤从头端连接到光纤结点，即光分配结点ODN
- HFC网采用**结点体系**
- HFC网具有双向传输功能，扩展了传输频带
- 用户接口盒UIB要提供三种连接：
 - 使用同轴电缆连接到机顶盒，然后再连接到用户的电视机
 - 使用双绞线连接到用户的电话机
 - 使用电缆调制解调器来连接到用户的计算机
 - 电缆调制解调器是为HFC网而使用的调制解调器

2.6.3 FTTx技术

- FTTx是一种实现宽带居民接入网的方案，代表多种宽带光纤接入方式

第三章 数据链路层

- 数据链路层使用的信道主要有以下两种类型：
 - 点对点信道====》一对一
 - 广播信道====》一对多，采用专用的共享信道协议

- 路由器只包含网络层，链路层，物理层
- 不同链路层可能采用不同的数据链路层协议

3.1 使用点对点信道的数据链路层

3.1.1 数据链路和帧

- **链路**（物理链路）：一条无源的点到点的物理线路段，中间没有任何其他的交换结点
- 一条链路只是一条通路的一个组成部分
- **数据链路**（逻辑链路）：除了**物理线路**外，还必须有**通信协议**软件来控制这些数据的传输。若是把实现这些协议的硬件和软件加到链路上，就构成了数据链路。
 - 最常用的方法是使用**网络适配器**（即网卡）来实现这些协议的硬件和软件。
 - 一般适配器都包括了数据链路层和物理层这两方面的功能
- 早期的数据通信协议就做**通信规程**，在数据链路层中，规程和协议是同义语。
- **帧**-数据链路层的协议数据单元
- **IP数据报**：网络层的协议数据单元
- 数据链路层不必考虑物理层如何实现比特传输的细节，只考虑数据的封装等

3.1.2 三个基本问题

1. 封装成帧：

- 封装成帧就是在一段数据的前后分别添加首部和尾部；
- 首部和尾部的一个重要作用就是**帧定界**
- 每一种链路层协议都规定了所能传送的帧的**数据部分长度上限--最大传送单元MTU**
- 当数据是由可打印的 ASCII 码组成的文本文件时，帧定界可以使用特殊的**帧定界符**。
- 控制字符 SOH (Start Of Header) 放在一帧的最前面，表示帧的首部开始。另一个控制字符 EOT (End Of Transmission) 表示帧的结束。

2. 透明传输：

- 如果数据中的某个字节的二进制代码恰好和 SOH 或 EOT 一样，数据链路层就会错误地“找到帧的边界”。
- 解决方法：**字节填充** (byte stuffing) 或**字符填充** (character stuffing)。
- 发送端的数据链路层在**数据中**出现控制字符“SOH”或“EOT”的**前面插入一个转义字符“ESC”**（其十六进制编码是 1B）。
- 接收端的数据链路层在将数据送往网络层之前删除插入的转义字符。
- 如果转义字符也出现在**数据当中**，那么应在**转义字符前面插入一个转义字符 ESC**。当接收端收到连续的两个转义字符时，就删除其中前面的一个。

3. 差错检验：

- 在传输过程中可能会产生比特差错：1 可能会变成 0 而 0 也可能变成 1。
- 在一段时间内，传输错误的比特占所传输比特总数的比率称为**误码率 BER** (Bit Error Rate)。
- 误码率与信噪比有很大的关系。
- 为了保证数据传输的可靠性，在计算机网络传输数据时，必须采用各种差错检测措施。
- 循环冗余检验：
 - 在发送端，先把数据划分为组。假定每组 k 个比特。
 - 假设待传送的一组数据 M = 101001（现在 k = 6）。我们在 M 的后面再添加供差错检测用的 n 位**冗余码**一起发送。
 - 冗余码的计算：
 - 用二进制的模 2 运算进行 2^n 乘 M 的运算，这相当于在 M 后面添加 n 个 0。
 - 得到的 $(k + n)$ 位的数除以事先选定好的长度为 $(n + 1)$ 位的除数 P，得出商是 Q 而余数是 R，余数 R 比除数 P 少 1 位，即 R 是 n 位。
 - 将余数 R 作为冗余码拼接在数据 M 后面发送出去。
 - >- 冗余码的计算举例：
 - >现在 k = 6, M = 101001。
 - 设 n = 3, 除数 P = 1101,
 - 被除数是 $2^n \times M = 101001000$ 。
 - 模 2 运算的结果是：商 Q = 110101,
 - 余数 R = 001。
 - 把余数 R 作为冗余码添加在数据 M 的后面发送出去。发送的数据是： $2^n \times M + R$
 - 即：101001001，共 $(k + n)$ 位。
- 在数据后面添加上的冗余码称为**帧检验序列 FCS**
- CRC 是一种常用的检错方法，FCS 是添加在数据后面的冗余码
- FCS 可以用 CRC 这种方法得出，但是 CRC 并非用来获得 FCS 的唯一方法
- 检验结果：
 - (1) 若得出的余数 R = 0，则判定这个帧没有差错，就接受 (accept)。
 - (2) 若余数 R \neq 0，则判定这个帧有差错，就丢弃。
- 但这种检测方法并不能确定究竟是哪一个或哪几个比特出现了差错。
- 只要经过严格的挑选，并使用位数足够多的除数 P，那么出现检测不到的差错的**概率**就很小很小。
- 注意：
 - 仅用循环冗余检验 CRC 差错检测技术只能做到**无差错接受** (accept)。
 - “**无差错接受**”是指：“凡是接受的帧（即不包括丢弃的帧），我们都能以非常接近于 1 的概率认为这些帧在传输过程中没有产生差错”。
 - 也就是说：“凡是接收端数据链路层接受的帧都没有传输差错”（有差错的帧就丢弃而不接受）。
 - 要做到“**可靠传输**”（即发送什么就收到什么）就必须再加上**确认和重传机制**。
 - 应当明确，“**无比特差错**”与“**无传输差错**”是不同的概念。
- 在数据链路层使用 CRC 检验，能够实现无比特差错的传输，但这还不是可靠传输。
- 本章介绍的数据链路层协议都**不是可靠传输**的协议。

3.2 点对点协议 PPP

- 数据链路层能实现可靠传输的高级数据链路控制 HDLC
- 目前使用得最广泛的数据链路层协议：点对点协议 PPP

3.2.1 PPP 协议的特点

1. PPP协议应满足的需求：
 - (1) **简单**：首要需求，提高互操作性
 - (2) **封装成帧**：规定特殊的字符作为帧定界符
 - (3) **透明性**：必须保证数据传输的透明性
 - (4) **多种网络层协议**：在同一条物理链路上同时支持多种网络层协议
 - (5) **多种类型链路**：
 - (6) **差错检验**：检测并丢弃有差错的帧
 - (7) **检测连接状态**：
 - (8) **最大传送单元**：MTU是数据链路层的帧可以载荷的数据部分的最大长度，而不是帧的总长度
 - (9) **网络层地址协商**：
 - (10) **数据压缩协商**：
2. PPP协议不需要的功能：
 - (1) 纠错：因为可靠传输由运输层的TCP协议负责
 - (2) 流量控制
 - (3) 序号
 - (4) 多点线路：因为PPP只支持点对点的链路通信
 - (5) 半双工或单双工：因为PPP只支持全双工链路
3. PPP协议的组成：
 - (1) 一个将IP数据报封装到串行链路的方法
 - (2) 链路控制协议LCP：用来建立，配置，和测试数据链路的连接
 - (3) 网络控制协议NCP：

3.2.2 PPP协议的帧格式

1. 各字段的意义：
 - ppp帧的首部和尾部分别为4个字节和2个字节
 - 标志字段F规定为0x7E（即01111110），标志一个帧的开始和结束
 - 地址字段A规定为0xFF（即11111111）
 - 控制字段C规定为0x03（即00000011）
 - 协议字段，两个字节
2. 字节填充
 - **PPP是面向字节的，所有的PPP帧的长度都是整数字节**
 - 透明传输问题：
 - PPP用在同步传输链路时，协议规定采用硬件来完成**比特填充**
 - 当PPP用在异步传输时，就使用一种特殊的**字符填充法**
 - 把转义字符定义为0x7D（即011111101）
 - PPP协议用在SONET/SDH链路时，使用同步传输，这时PPP协议采用**零比特填充**方法实现透明传输
3. 零比特填充
 - 只要发现有5个连续的1，则立即填入一个0，因此经过这种填充后的数据不会出现六个连续的1。（因为几个特殊字段内有111111）

3.2.3 PPP协议的工作状态

- PPP协议已不是纯粹的数据链路层的协议，它还包含了物理层和网络层的内容。
- 链路静止状态
- 链路建立状态
- 链路打开状态
- 链路终止状态

3.3 使用广播信道的数据链路层

3.3.1 局域网的数据链路层

- 局域网最主要的**特点**是：
 - 网络为一个单位所拥有；
 - 地理范围和站点数目均有限。
- 局域网具有如下**主要优点**：
 - 具有广播功能，从一个站点可很方便地访问全网。局域网上的主机可共享连接在局域网上的各种硬件和软件资源。
 - 便于系统的扩展和逐渐地演变，各设备的位置可灵活调整 and 改变。
 - 提高了系统的可靠性、可用性和残存性。
- 局域网按拓扑结构分类：
 - 星型网，集线器
 - 总线网，匹配电阻
 - 环形网，干线耦合器
- 局域网中的主要传输媒体：双绞线
- 当数据率很高时，往往需要使用光纤作为传输媒体
- 媒体共享技术：
 - 静态划分信道：频分复用，时分复用，波分复用，码分复用
 - 动态媒体接入控制（多点接入）：
 - 随机接入：所有用户可随机地发送信息。但是同一时刻发送信息，会产生碰撞（冲突），故需要解决碰撞的网络协议。
 - 受控接入：用户不能随机发送信息，必须服从一定控制：如多点线路探询，轮询。

1. 以太网的标准：
 - DIX Ethernet V2 是世界上第一个局域网产品（以太网）的规约。
 - IEEE 802.3 是第一个 IEEE 的以太网标准。
 - IEEE 802 委员会就将局域网的数据链路层拆成两个子层：

- 逻辑链路控制 LLC (Logical Link Control)子层;
- 媒体接入控制 MAC (Medium Access Control)子层。
- 与接入到传输媒体有关的内容都放在 MAC子层, 而 LLC 子层则与传输媒体无关。
- **不管采用何种协议的局域网, 对 LLC 子层来说都是透明的。**

2. 适配器的作用:

- 网络接口板又称为**通信适配器** (adapter) 或**网络接口卡** NIC (Network Interface Card), 或“**网卡**”。
- 适配器的重要功能:
 - 进行串行/并行转换。
 - 对数据进行缓存。
 - 在计算机的操作系统安装设备驱动程序。
 - 实现以太网协议。
- 适配器在接受和发送各种帧时, 不适用计算机的CPU。当收到有差错的帧时, **直接丢弃**而不必通知计算机。当收到正确的帧时, 使用**中断**来通知计算机, 并交付协议栈中的网络层。
当计算机要发送IP数据报时, 就由协议栈把IP数据报向下交给适配器, 组装成帧后发送到局域网

3.3.2 CSMA/CD协议

- 总线的特点: 当一台计算机发送数据时, 总线上所有的计算机都能检测到这个数据
- 为了在总线上实现一对一的通信, 可以使适配器拥有一个独一无二的地址
- 为了通信简便, 以太网采用:
 - 采用较为灵活的无连接的工作方式
 - 不必先建立连接就可以直接发送数据。
 - 对发送的数据帧不进行编号, 也不要求对方发回确认。
 - 这样做的理由是局域网信道的质量很好, 因信道质量产生差错的概率是很小的。
 - 以太网提供的服务是**不可靠的交付, 即尽最大努力的交付。**
 - 当目的站收到有差错的数据帧时就丢弃此帧, 其他什么也不做。**差错的纠正由高层来决定。**
 - 如果高层发现丢失了一些数据而进行重传, 但**以太网并不知道这是一个重传的帧, 而是当作一个新的数据帧来发送。**
 - 以太网发送的数据都**使用曼彻斯特**编码:
 - 缺点: 占用的频带宽度比原始的基带信号增加了一倍
 - CSMA/CD协议:
 - CSMA/CD 含义: **载波监听多点接入 / 碰撞检测** (Carrier Sense Multiple Access with Collision Detection)。
 - “**多点接入**”表示许多计算机以多点接入的方式连接在一根总线上。
 - “**载波监听**”是指每一个站在发送数据之前先要检测一下总线上是否有其他计算机在发送数据, 如果有, 则暂时不要发送数据, 以免发生碰撞。
 - “**碰撞检测**”是指边发送边监听
 - 总线上并没有什么“载波”。因此, “**载波监听**”就是用电子技术检测总线上有没有其他计算机发送的数据信号。
 - 检测到碰撞后: 每一个正在发送数据的站, 一旦发现总线上出现了碰撞, 就要**立即停止发送**, 免得继续浪费网络资源, 然后**等待一段随机时间**后再次发送。
 - 由于电磁波在总线上的传播速率是有限的, 当某个站监听到总线是空闲时, 也可能总线并非真正是空闲的所以需要在发送期间进行碰撞检测, 以检测冲突。
- 在使用CSMA/CD协议时, 一个站不可能同时进行发送和接收 (但必须边发送边监听信道)
- 因此CSMA/CD协议不可能进行全双工通信只能进行双向交替通信 (**半双工通信**)
- 每个站在发送数据之后的一小段时间内, 存在着遭遇碰撞的可能性。
- 这种**发送的不确定性**使整个以太网的平均通信量远小于以太网的最高数据率。
- 最先发送数据帧的站, 在发送数据帧后**至多经过时间 2τ (两倍的端到端往返时延)** 就可知道发送的数据帧是否遭受了碰撞。
- 以太网的端到端往返时延 2τ 称为**争用期**, 或**碰撞窗口**。
- **经过争用期这段时间还没有检测到碰撞, 才能肯定这次发送不会发生碰撞。**
- 二进制指数型退避算法:
 - 发生碰撞的站在停止发送数据后, 要推迟 (退避) 一个**随机时间**才能再发送数据。
 - 基本退避时间取为争用期 2τ 。
 - 从整数集合 $\{0, 1, \dots, (2^k-1)\}$ 中**随机**地取出一个数, 记为 r 。重传所需的时延就是 r 倍的基本退避时间。
 - 参数 k 按下面的公式计算:
 $k = \text{Min}[\text{重传次数}, 10]$
 - 当 $k \leq 10$ 时, 参数 k 等于重传次数。
 - 当重传达 16 次仍不能成功时即丢弃该帧, 并向高层报告。
 - 以太网在发送数据时, 若前 64 字节没有发生冲突, 则后续的数据就不会发生冲突。**最短有效帧长为 64 字节**
- 强化碰撞: 当发送数据的站一旦发现发生了碰撞时:
 - (1) 立即停止发送数据;
 - (2) 再继续发送若干比特的人为**干扰信号**以便让所有用户都知道现在已经发生了碰撞。
- CSMA/CD协议的要点:
 - (1) 准备发送

- (2) 检测信道
- (3) 检查碰撞

3.3.3 使用集线器的星形拓扑

- 特点：
 - (1) 使用集线器的以太网在**逻辑上仍是一个总线网**，各站**共享逻辑上的总线**，各适配器使用的还是CSMA/CD协议，并且同一时刻只允许一个站发送数据
 - (2) 一个集线器很像一个多接口的转发器
 - (3) **集线器工作在物理层**，它的每一个接口仅仅简单地转发比特，不进行碰撞检测
 - (4) 采用专门的芯片

3.3.4 以太网的信道利用率

- 要提高以太网的信道利用率，就必须减少 τ 与 T_0 之比，在以太网中定义了参数 a ，它是以太网**单程端到端时延 τ 与帧的发送时间 T_0** ： $a = \tau / T_0$ ；
- $\alpha \rightarrow 0$ ，表示一旦发生碰撞就立即可以检测出来，并立即停止发送，因而信道利用率很高。
- **α 越大，表明争用期所占的比例增大**，每发生一次碰撞就浪费许多信道资源，**使得信道利用率明显降低**。
- 为提高利用率，以太网的参数 a 的值应当尽可能小些。
- 对以太网参数 α 的要求是：
 - 当数据率一定时，以太网的连线的长度受到限制，否则 τ 的数值会太大。
 - 以太网的帧长不能太短，否则 T_0 的值会太小，使 α 值太大。
 - 只有当参数 a 远小于1才能得到尽可能高的极限信道利用率。
- **理想情况下的极限信道利用率 S_{max} 为：**
 $S_{max} = T_0 / (T_0 + \tau) = 1 / (1 + a)$

3.3.5 以太网的MAC的层

1. MAC层的硬件地址：
 - 在局域网中，**硬件地址**又称为**物理地址**，或**MAC地址**。
 - IEEE 802标准为局域网规定了一种48位的全球地址，是指局域网上的每一台计算机中**固化在适配器的ROM中的地址**
 - IEEE的RA是局域网全球地址的法定管理结构，它负责分配地址字段的6个字节的前三个字节。生产局域网适配器的厂家需要向IEEE购买由这三个字节构成的组织唯一标识符OUI
 - 厂家自行指派后三个字节，称为扩展标识符，用这种方法得到的48位地址称为EUI-48
 - IEEE规定地址字段的第一字节的最低位为I/G位：I/G位为0时，地址段表示单个站地址；I/G位为1时表示组地址，用来多播
 - IEEE规定地址字段第一字节的最低第二位为G/L位：G/L位为0时是全球管理，为1时是本地管理
 - 适配器由**过滤功能**，从网络上每收到一个MAC帧就先用硬件检查MAC帧中的目的地址。
 - 发往本站的帧：单播帧，广播帧，多播帧
 - 所有的适配器都至少能够识别前两种帧，即能够识别单播地址和广播地址。
 - 有的适配器可用编程方法识别多播地址。
 - 只有目的地址才能使用广播地址和多播地址。
 - 以**混杂方式** (promiscuous mode) 工作的以太网适配器只要“听到”有帧在以太网上传输就都接收下来。
2. MAC帧的格式
 - 最常用的MAC帧是以太网V2格式
 - 最后一段是4字节的帧检验序列FCS（使用CRC检验）
 - 无效的MAC帧：
 - 长度不是整数个字节
 - 收到的FCS帧检验序列查出错误
 - 数据位置错误

3.4 扩展的以太网

3.4.1 在物理层扩展以太网

1. 使用光纤扩展
 2. 使用集线器扩展（碰撞域）
- 优点
 - (1) 使原来属于不同碰撞域的以太网上的计算机能够进行跨碰撞域的通信。
 - (2) 扩大了以太网覆盖的地理范围。
 - 缺点
 - (1) 碰撞域增大了，但总的吞吐量并未提高。
 - (2) 如果不同的碰撞域使用不同的数据率，那么就不能用集线器将它们互连起来。

3.4.2 在数据链路层扩展以太网

- 早期使用**网桥**，现在使用以太网**交换机**
1. 以太网交换机的特点
 - 实质上是一个多接口网桥
 - 工作在全双工方式
 - 相互通信的主机都是独占传输媒体，无碰撞地传输数据
 - 具有并行性，能使多对主机同时通信
 - 许多以太网交换机对收到的帧采用存储转发方式进行转发，但是也有一些交换机采用直通的交换方式
 2. 以太网交换机的自学习功能
 - 即插即用，帧交换表

3. 从总线以太网到星形以太网

- 仍然采用以太网的帧结构

3.4.3 虚拟局域网

- **虚拟局域网 VLAN** 是由一些局域网网段构成的与**物理位置无关的逻辑组**，而这些网段具有某些共同的需求。每一个 VLAN 的帧都有一个明确的标识符，指明发送这个帧的计算机是属于哪一个 VLAN。
- **虚拟局域网其实只是局域网给用户提供服务，而不是一种新型局域网。**

3.5 高速以太网

3.5.1 100BASE-T 以太网

- 速率达到或超过100Mbit/s的以太网称为高速以太网
- 100BASE-T以太网又叫快速以太网
- 可在全双工方式下工作而无冲突发生

3.5.2 吉比特以太网

- 在半双工方式下使用 CSMA/CD 协议，全双工方式不使用 CSMA/CD 协议。
- 载波延伸
- 分组突发

3.5.3 10吉比特以太网 (10GE) 和更快的以太网

- 只工作在全双工方式

3.5.4 使用以太网进行宽带接入

- 可以提供双向的宽带通信。
- 可以根据用户对带宽的需求灵活地进行带宽升级。
- 可以实现端到端的以太网传输，中间不需要再进行帧格式的转换。这就提高了数据的传输效率且降低了传输的成本。
- 但是不支持用户身份鉴别。

第四章 网络层

4.1 网络层提供的两种服务

- **面向连接还是无连接**
- 互联网采用的设计思路是这样的：**网络层向上只提供简单灵活的，无连接的，尽最大努力交付的数据报服务。**
- **网络层不提供服务质量的承诺**
- 通信之前先建立**虚电路**，以保证双方通信所需的一切网络资源。
- 虚电路表示这只是一条逻辑上的连接，分组都沿着这条逻辑连接按照存储转发方式传送

4.2 网际协议IP

- 与IP协议配套使用的三个协议：
 - 地址解析协议ARP
 - 网际控制协议ICMP
 - 网际组管理协议IGMP
- 由于网际协议IP是用来使互连起来的许多计算机网络能够进行通信的，因此TCP/IP体系中的网络层常常被称为网际层或IP层

4.2.1 虚拟互连网络

- 没有一种单一的网络能够适应所有用户的需求
- 将网络互相连接起来要使用一些中间设备：
 - 物理层中继系统：**转发器** (repeater)。
 - 数据链路层中继系统：**网桥** 或 **桥接器** (bridge)。
 - 网络层中继系统：**路由器** (router)。
 - 网桥和路由器的混合物：**桥路器** (brouter)。
 - 网络层以上的中继系统：**网关** (gateway)。
- 网络互连都是指用路由器进行网络互连和路由选择。
- 由于历史的原因，许多有关 TCP/IP 的文献将网络层使用的路由器称为网关。
- 所谓虚拟互连网络也就是逻辑互连网络，它的意思就是互连起来的各种物理网络的异构性本来是客观存在的，但是我们利用 IP 协议就可以使这些性能各异的网络从用户看起来好像是一个统一的网络。
- 使用 IP 协议的虚拟互连网络可简称为 IP 网。
- 使用虚拟互连网络的好处是：当互联网上的主机进行通信时，就好像在一个网络上通信一样，而看不见互连的各具体的网络异构细节。
- 如果在这种覆盖全球的 IP 网的上层使用 TCP 协议，那么就是现在的互联网 (Internet)。
- 互联网可以由多种异构网络互连组成

4.2.2 分类的IP地址

1. IP地址及其表示方法：

- 整个互联网就是一个**单一的，抽象的网络**。IP地址就是给互联网上的每一台主机（或路由器）的每一个接口分配一个全世界范围内是唯一的32位的标识符
- IP地址现在由**互联网名字和数字分配机构ICANN**进行分配
- IP地址的编制方法经过三个历史阶段：
 - (1) 分类的IP地址
 - (2) 子网的划分
 - (3) 构成超网
- 分类的IP地址：就是将IP地址划分位若干个固定类。

- A, B, C类地址都是由两个固定长度的字段组成，其中第一个字段是网络号，是唯一的。第二个字段是主机号，在当前网络范围内是唯一的
- 由此可见，一个IP地址在整个互联网范围内是唯一的
- A, B, C类是单播地址，D类地址用于多播，E类地址保留
- 点分十进制记法

2. 常用的三种类别的IP地址

- A类地址，**网络号全0**表示本网络，全1表示本地软件环回测试
- A类地址，**主机号全0**表示本主机连接的单个网络地址，全1表示该网络上所有的主机
- 一般不用的特殊的IP
 - IP地址的特点：
 - (1) IP是分等级的地址结构，IP地址管理机构**只分配网络号，主机号由得到网络号的单位自行分配，路由器仅根据目的主机网络号来转发分组**，减小了路由表所占的存储空间以及查找路由表的时间
 - (2) IP地址标志一台主机和一条链路的**接口**，当一台主机同时连接到两个网络上时，该主机就必须同时具有两个相应的IP地址，其网络号必须不同。这种主机也叫做**多归属主机**
 - (3) 一个网络时指具有相同网络号net-id的主机的集合，故，**用转发器或网桥连接起来的若干局域网仍是一个网络**。具有不同网络号的局域网必须用路由器进行互连
 - (4) 在IP地址中，所有分配到网络号的网络都是平等的
- 当两个路由器直接相连时，为了节省IP地址资源，常常不给这段连线分配IP地址，通常把这段特殊的网络叫做无编号网络或无名网络

4.2.3 IP地址与硬件地址

- 区分IP地址与硬件地址：从层次上来看，**物理地址是数据链路层和物理层使用的地址，而IP地址是网络层和以上各层使用的地址，是一种逻辑地址。**
- 在发送数据时，数据从高层向下到低层，然后才到通信链路上传输。使用IP地址的IP数据报一旦交给了数据链路层，就被封装成MAC帧了。MAC帧在传送时使用的源地址和目的地址都是硬件地址，这两个地址都写在MAC帧的头部。连接在通信链路上的设备在收到MAC帧时，根据MAC帧首部的硬件地址决定抛弃与否，只有在剥去MAC帧的首部和尾部后把MAC层的数据上交给网络层后，网络层才能在IP数据报的首部找到源IP地址和目的IP地址
- 注意：
 - (1) 在IP层抽象的互联网上只能看到IP数据报
 - (2) 路由器只根据目的站的IP地址的网络号进行路由选择
 - (3) 在局域网的链路层，只能看见MAC帧
 - (4) IP层抽象的互联网频闭下层这些很复杂的细节。

4.2.4 地址解析协议ARP

- ARP常被划分为网络层的协议
- ARP协议的用途：为了从网络层使用的IP地址，解析出在数据链路层使用的硬件地址。
- **地址解析协议ARP**解决这个问题的方法是在主机**ARP高速缓存**中存放一个从IP地址到硬件地址的映射表，并且这个映射表还经常动态更新
- 当主机A要向本局域网上的某台主机B发送IP数据报时，就先在其ARP高速缓存查看有无主机B的IP地址。如果有，查出硬件地址并写入MAC帧，如果没有，(1) ARP在本局域网广播发送一个ARP请求分组 (2) 在本局域网的所有主机上运行的ARP进程都收到了请求分组 (3) 主机B的IP地址与请求分组中要查询的IP地址一致，收下这个请求分组，并向主机A发送ARP响应分组，同时在这个ARP响应分组中写入自己的硬件地址 (4) 主机A收到主机B的ARP分组后，就将其ARP高速缓存中写入主机B的IP地址到硬件地址的映射
- 每一个映射地址项目都设置生存时间
- ARP是解决同一个局域网上的主机或路由器的IP地址和硬件地址的映射问题
- 从IP地址到硬件地址的解析是自动进行的，**主机的用户对这种地址解析过程是不知道的**

4.2.5 IP数据报的格式

1. IP数据报首部的固定部分中的各字段
 - (1) 版本
 - (2) 首部长度
 - (3) 区分服务
 - (4) 总长度：指首部和数据之和的长度
 - (5) 标识
 - (6) 标志：占3位，最低位MF；中间位DF
 - (7) 片偏移
 - (8) 生存时间TTL：跳数限制
 - (9) 协议
 - (10) 首部检验和：只检验数据报的首部，但不包括数据部分
 - (11) 源地址
 - (12) 目的地址

2. IP层转发分组的流程

- 在互联网上转发分组时，是从一个路由器转发到下一个路由器
- 在路由表中，对每一条路由最主要的是以下两个信息：**目的网络地址，下一跳地址。**
- 虽然互联网所有的分组转发都是**基于目的主机所在的网络**，但是存在着对特定的目的主机指明一个路由。这种路由叫做**特定主机路由**。
- 默认路由：减小路由表所占用的空间和搜索路由表所用的时间
- 分组转发算法：
 - (1) 从数据报首部提取目的主机IP地址D，得到目的网络N
 - (2) 若N就是与此路由器直接相连的某个网络地址，则进行直接交付，否则，间接交付
 - (3) 若路由表中有目的地址为D的特定主机路由，则把数据报传送给下一跳路由器，若无则 (4)
 - (4) 若路由器有到达网络N的路由，则把数据报传送给路由表中所指明的下一跳，否则 (5)
 - (5) 若路由表中有一个默认路由，则把数据报传送给路由表中所指明的默认路由器，否者 (6)
 - (6) 报告转发分组失败

4.3 划分子网和构造超网

4.3.1 划分子网

1. 从两级IP地址到三级IP地址

- IP地址设计不合理：
 - (1) IP地址空间的**利用率有时很低**
 - (2) 给每个物理网络分配一个网络号会使路由表变得太大
 - (3) 两级IP地址不够灵活
- 为了解决上诉问题，IP地址中又增加了“子网号字段”，使两级地址变成三级地址----这种做法也叫划分子网或子网寻址或子网路由选择
- 划分子网的基本思路：
 - (1) 一个拥有许多物理网络的单位，可将所属的物理网络划分为若干各子网。划分子网纯属一个单位内部的事情。本网络以外的网络看不见这个网络是由多少各子网组成，因为其对外仍然表现为一个网络
 - (2) 划分子网的方法是从网络的**主机号**借用若干位作为**子网号**，当然主机号也就减少了同样的位数，于是两级IP地址在本单位内就变成了三级IP地址，网络号，子网号和主机号
 - (3) 凡是从其他网络发送来的IP数据报，仍是根据目的网络号找到路由器，此路由器收到IP数据报后，再按照目的网络号和子网号找到目的子网，把IP数据报交付目的主机

2. 子网掩码

- 从IP数据报的首部无法看出源主机或目的主机多连接的网络是否进行了子网的划分。因此使用子网掩码。
- 将子网掩码与收到的数据报的目的IP地址**逐位相与**，得到子网的网络地址
- 再路由器的路由表中也必须要有子网掩码，如果一个网络不划分子网，那么该网络的子网掩码就使用默认子网掩码
- 子网掩码是一个网络或一个子网的重要属性
- 划分子网增加了灵活性，但却减少了能够连接在网络上的主机总机数
- 相同的IP地址和不同的子网掩码可以得到相同的网络地址

4.3.2 使用子网时分组的转发

- 路由表中必须包含：目的网络四肢，子网掩码，下一跳地址
- 划分子网的情况下，路由器转发分组的算法：
 - (1) 从收到的IP数据报的首部提取目的IP地址D
 - (2) 先判断是否为直接交付，否则执行（3）
 - (3) 若路由表中有目的地址为D的特定主机路由，则把数据报传送给路由器中所指明的下一跳路由器，否则执行（4）
 - (4) 对路由表中的每一行，用其中的子网码和D逐位相“与”，其结果为N。若N与该行的目的地址匹配，则把数据报传送给该行指明的下一跳路由器；否则执行（5）
 - (5) 若路由器中有一个默认路由，则把数据报传送给路由表中所指明的默认路由器；否则执行（6）
 - (6) 报告转发分组出错

4.3.3 无分类编址CIDR（构造超网）

1. 网络前缀

- IPv4地址空间不够，IPv6出现
- 使用变长子网掩码VLSM
- 无分类域间路由选择CIDR：
 - (1) 消除了传统的ABC类地址以及划分子网的概念，更能有效的分配地址空间。CIDR将地址空间分为前后两部分，前部分是网络前缀，用来指明网络，后面部分指明主机。CIDR使用斜线记法，在IP地址后面加一斜线，后面写网络前缀所占的位数
 - (2) CIDR把网络前缀都相同的连续的IP地址组成一个“CIDR地址块”，CIDR使用32为的地址掩码，例如：/20地址块的地址掩码是：11111111 11111111 11110000 00000000（即斜线后的数字是地址掩码中1的个数）
- 由于CIDR地址块中有很多地址，所以在路由表中就利用CIDR地址块来查找目的网络。这种地址的聚合常称为路由聚合。
- 除个别外，每一个CIDR地址块都包含了多个c类地址，这就是构成超网
- 网络前缀越短，其地址块中包含的地址数就越多

2. 最长前缀匹配

- 在查找路由表时，应当从匹配结果中选择出具有最长网络前缀的路由。

3. 使用二叉线查找路由表

4.4 网际控制报文协议ICMP

- 为了更有效的转发IP数据报和提高交付成功的机会，在网际层使用了网际控制报文协议ICMP
- ICMP允许主机或路由器报告差错情况和提供异常情况的报告

4.4.1 ICMP报文的种类

- ICMP报文有两种，即ICMP差错报告报文，ICMP询问报文
- ICMP差错报告报文的类型：
 - (1) 终点不可达
 - (2) 时间超过
 - (3) 参数问题
 - (4) 改变路由（重定向）
- ICMP询问报文的类型：
 - (1) 回送请求和回答
 - (2) 时间戳请求和回答

4.4.2 ICMP的应用举例

- ICMP的一个重要应用就是分组间探测PING，用来测试两台主机之间的连通性
- ping使用了ICMP回送请求与回送回答报文
- traceroute/tracert

4.5 互联网的路由选择协议

4.5.1 有关路由选择协议的几个基本概念

1. 理想的路由算法
 - (1) 算法必须是正确的和完整的

- (2) 算法在计算上应简单
- (3) 算法应能适应通信量和网络拓扑的变化
- (4) 算法应具有稳定性
- (5) 算法应是公平的
- (6) 算法应是最佳的
- 从路由算法能否随网络的通信量或拓扑自适应地进行调整变化来划分，则只有两大类，即静态路由选择策略（非自适应路由选择）和动态路由选择策略（自适应路由选择）。

2. 分层次的路由选择协议

- 互联网采用的路由选择协议主要是自适应，分布式路由选择协议
- 自治系统AS：在单一技术管理下的一组路由器，这些路由器使用一种自治系统内部的路由选择协议和共同的度量。
- 一个AS对其他AS表现出的是一个单一的和一致的路由选择策略
- 互联网把路由选择协议划分为两大类：
 - 内部网关协议IGP（AS内部）：包括RIP，OSPF
 - 外部网关协议EGP（AS之间）：例如：BGP

4.5.2 内部网关协议RIP

1. 工作原理：

- RIP是一种分布式的**基于距离向量的路由选择协议**
- RIP协议要求网络中的每一个路由器都要维护从它自己到其他每一各目的网络的**距离记录**：
 - 从一路由器到直接连接的网络的距离定义为1
 - 从一路由器到非直接连接的网络的距离定义为所经过的路由器数加1
 - RIP协议的“距离”也称为“跳数”，只适用于小型的互联网
- RIP协议的特点：
 - 仅和相邻路由器交换信息
 - 路由器交换的信息是当前本路由器所知道的全部信息，即自己现在的路由表
 - 按固定的时间间隔交换路由信息
- 路由器再刚刚开始工作时，它的路由表是空的，然后路由器就得出直接相连的几个网络的距离。接着每一个路由器也之和数目非常有限的相邻路由器交换，并更新路由信息
- “收敛”：在自治系统中所有的结点都得到正确的路由选择信息的过程
- 路由表更新的职责是找出到每一个目的网络的最短距离。这种更新算法又称为距离向量算法。

2. 距离向量算法

路由器收到相邻路由器（其地址为 X）的一个 RIP 报文：

(1) 先修改此 RIP 报文中的所有项目：把“下一跳”字段中的地址都改为 X，并把所有的“距离”字段的值加 1。

(2) 对修改后的 RIP 报文中的每一个项目，重复以下步骤：

若项目中的目的网络不在路由表中，则把该项目加到路由表中。

否则

若下一跳字段给出的路由器地址是同样的，则把收到的项目替换原路由表中的项目。

否则

若收到项目中的距离小于路由表中的距离，则进行更新，

否则，什么也不做。

(3) 若 3 分钟还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达路由器，即将距离置为 16（表示不可达）。

(4) 返回。

3. RIP协议的报文格式

- RIP协议使用运输层的用户数据报UDP进行传送
- RIP报文由**首部**和**路由**部分组成
- 问题：**当网络出现故障时，要经过比较长度的时间才能将此信息传送到所有的路由器**
- 特点：好消息传播的快，坏消息传播的慢
- 优点：是实现简单，开销较小

4.5.3 内部网关协议OSPF

1. OSPF协议的基本特点：

- 开放最短路径优先OSPF：使用了Dijkstra提出的最短路径算法SPF
- OSPF只是一个协议的名字，它并不表示其他的路由选择协议不是“最短路径优先”
- OSPF最主要的特征是使用分布式的**链路状态协议**
 - 向本自治系统中所有路由器发送信息，洪泛法（RIP只向自己相邻的路由器发送信息）
 - 发送的信息就是与本路由器相邻的所有路由器的**链路状态**（与哪些路由器相邻+度量）
 - 只有当链路状态发生变化时，路由器才向所有路由器用洪泛法发送此信息
- 所有的路由器最终都能建立一个**链路状态数据库**，这个数据库实际上就是**全网的拓扑结构图**，这个拓扑结构图在全网范围内是一致的（链路状态数据库的同步）
- OSPF的更新过程收敛快----优点
- OSPF将一个自治系统划分为若干个**区域**
- 主干区域，区域边界路由器（R3，R4，R7），主干路由器（R3，R4，R5，R6，R7），自治系统边界路由器（R6）
- OSPF不用UDP而是直接用IP数据报传送

- OSPF首部各字段意义：
 - (1) 版本
 - (2) 类型
 - (3) 分组长度
 - (4) 路由器标识符
 - (5) 区域标识符
 - (6) 检验和
 - (7) 鉴别类型
 - (8) 鉴别
- OSPF的特点：
 - (1) OSPF对于不同类型的业务计算出不同的路由
 - (2) 负载均衡
 - (3) 鉴别
 - (4) OSPF支持CIDR
 - (5) OSPF让每一个链路状态带上序号，序号越大，状态越新

2. OSPF的五种分组类型

- (1) 类型1，问候 (Hello) 分组。
 - (2) 类型2，数据库描述 (Database Description) 分组。
 - (3) 类型3，链路状态请求 (Link State Request) 分组。
 - (4) 类型4，链路状态更新 (Link State Update) 分组，用洪泛法对全网更新链路状态。
 - (5) 类型5，链路状态确认 (Link State Acknowledgment) 分组。
- OSPF采用的是可靠的洪泛法（收到更新分组后要发送确认）
 - OSPF每隔一段时间要刷新一次数据库中的链路状态
 - OSPF协议对多点接入的局域网采用了**指定的路由器**的方法

4.5.4 外部网关协议BGP

- 内部网关协议主要是设法使数据报在**一个AS中**尽可能有效地从源站传送到目的站
- 使用BGP的原因：
 - (1) **互联网的规模太大，使得自治系统AS之间路由选择非常困难**
 - (2) **自治系统AS之间的路由选择必须考虑有关策略**
- 边界网关协议BGP只能是力求寻找一条能够到达目的网络且**比较好的**路由，而**并非要寻找一条最佳路由**
- BGP采用了**路径向量路由选择协议**
- BGP发言人与其他AS的BGP发言人要交换路由信息，就要先建立TCP连接，然后在此连接上交换BGP报文以建立**BGP会话**，利用BGP会话交换路由信息。
- 使用TCP连接交换路由信息的两个BGP发言人彼此成为对方发邻站或对应站
- BGP报文：OPEN，UPDATE，KEEPALIVE，NOTIFICATION

4.5.5 路由器的构成

1. 路由器的结构

- 路由器是一种具有多个输入端口和多个输出端口的专用计算机，其任务是转发分组，按照分组要去的目的地，从合适的端口转发给下一跳路由器。
- 路由器的转发分组正是网络层的主要工作
- 路由器结构划分为两大部分：
 - (1) **路由选择部分**：控制部分，核心构件是路由选择处理机，任务：①根据路由选择协议构造路由表②不断更新和维护路由表
 - (2) **分组转发部分**：由三部分组成：交换结构，一组输入端口，一组输出端口，
- 路由器的作用：连接不同的网络

2. 交换结构

- 也叫（交换组织），作用：根据**转发表**对分组进行处理，从合适的端口转发出去。
- 路由表是根据路由选择算法得出，而转发表是从路由表得出的
- 输入端口中的查找和转发功能在路由器的交换功能中是最重要的。
- 路由器的端口收到分组时，就用中断方式通知路由选择处理机
- 路由器中的输入或输出队列产生溢出是造成分组丢失的重要原因。
- 通过存储器
- 通过总线
- 通过纵横交换结构

4.6 IPv6

- 双栈协议：IPv4向IPv6过渡
- 隧道技术

4.6.1 IPv6的基本首部

- 变化：
 - (1) 更大的地址空间，128位
 - (2) 扩展的地址层次结构
 - (3) 灵活的首部格式
 - (4) 改进的选项
 - (5) 允许协议继续扩充

- (6) 支持即插即用
- (7) 支持资源的预分配
- (8) IPv6首部改为8字节对齐
- IPv6数据报由两大部分组成：基本首部和有效载荷（净负荷）
- IPv6 把原来 IPv4 首部中选项的功能都放在**扩展首部**中，并将扩展首部留给路径两端的源站和目的站的主机来处理。
- 数据报途中经过的路由器都不处理这些扩展首部（只有一个首部例外，即逐跳选项扩展首部）。
- 这样就大大提高了路由器的处理效率。

4.6.2 IPv6的地址

- 单播
- 多播
- 任播
- IPv6 将实现 IPv6 的主机和路由器均称为**结点**。
- 一个结点就可能有多与链路相连的接口。
IPv6 地址是分配给结点上面的接口的。
- 一个接口可以有多个单播地址。
- 其中的任何一个地址都可以当作到达该结点的目的地址。即**一个结点接口的单播地址可用来唯一地标志该结点**。
- 为了使地址再稍简洁些，IPv6 使用冒号十六进制记法
- 冒号十六进制记法可以允许零压缩 (zero compression)，即一连串连续的零可以为一对冒号所取代。
FF05::0:0:0:0:0:0:B3 可压缩为：
FF05::B3
- 注意：在任一地址中只能使用一次零压缩。

4.7 多播

4.8 虚拟专用网和网络地址转换NAT

4.9 多协议标记交换MPLS

第五章 运输层

5.1 运输层协议概述

5.1.1 进程之间的通信

- 运输层向它上面的应用层提供通信服务**
- 它属于**面向通信部分**的最高层，同时也是**用户功能的最低层**
- 当网络的边缘部分中的两个主机使用网络的核心部分的功能进行端到端的通信时，只有位于网络边缘部分的主机的协议栈才有运输层，而网络核心部分中的路由器在转发分组时都只用到下三层的功能。
- 从IP层来说，通信的两端是两台主机，但是严格来说，真正进行通信的实体是在主机中的进程（运输层）
- 网络层是为主机之间提供逻辑通信，而运输层为应用进程之间提供端到端的逻辑通信
- 在一台主机中经常有**多个应用进程**同时分别和另一台主机中的多个应用进程通信。
- 这表明运输层有一个很重要的功能——**复用** (multiplexing)和**分用** (demultiplexing)。
- 根据应用程序的不同需求，运输层需要有两种不同的运输协议，即**面向连接的 TCP** 和**无连接的 UDP**。
- 运输层向高层用户**屏蔽**了下面网络核心的细节（如网络拓扑、所采用的路由选择协议等），它使应用进程**看见的就是好像**在两个运输层实体之间有一条端到端的逻辑通信信道。
- 当运输层采用**面向连接的 TCP** 协议时，尽管**下面的网络是不可靠的**（只提供**尽最大努力服务**），但这种逻辑通信信道就相当于一条**全双工的可靠信道**。
- 当运输层采用**无连接的 UDP** 协议时，这种逻辑通信信道是一条**不可靠信道**。

5.1.2 运输层的两个主要协议

- TCP/IP 的运输层有两个主要协议：
 - (1) **用户数据报协议 UDP** (User Datagram Protocol)
 - (2) **传输控制协议 TCP** (Transmission Control Protocol)
- 两个对等运输实体在通信时传送的数据单位叫作**运输协议数据单元 TPDU**
- TCP 传送的数据单位协议是 **TCP 报文段**
- UDP 传送的数据单位协议是 **UDP 报文**或**用户数据报**。
- UDP：传输前不需要建立连接，不可靠
- TCP：传输数据前建立连接，传输后释放，其下网络是不可靠的，但是TCP提供可靠的面向连接的运输服务，TCP不提供广播，多播服务

5.1.3 运输层的端口

- 复用：应用层所有的应用进程都可以通过运输层再传送到IP层
- 分用：运输层从IP层收到发送给个应用进程的数据后，必须分别交付指明的各应用进程
- 运行在计算机中的进程是用**进程标识符**来标志的。
- 但运行在应用层的各种应用进程却不应当让**计算机操作系统指派它的进程标识符**。这是因为在互联网上使用的计算机的操作系统种类很多，而**不同的操作系统又使用不同格式的进程标识符**。
- 为了使**运行不同操作系统的计算机的应用进程能够互相通信**，就必须用统一的方法对 TCP/IP 体系的应用进程进行标志。即使用**协议端口号**
- 在协议栈层间的抽象的协议端口是**软件端口**。
路由器或交换机上的端口是**硬件端口**。
- 端口号只具有**本地意义**，即端口号只是为了标志**本计算机应用层中的各进程**
- 由此可见，两个计算机中的进程要互相通信，不仅必须知道对方的 **IP 地址**（为了找到对方的计算机），而且还要知道对方的**端口号**（为了找到对方计算机中的应用进程）。

5.2 用户数据报协议UDP

5.2.1 UDP概述

- UDP具有复用，分用，差错检验的功能

- UDP的特点：
 - (1) **UDP是无连接的**，发送数据之前不需要建立连接，因此减少了开销和发送数据之前的时延。
 - (2) **UDP使用尽最大努力交付**，即不保证可靠交付，因此主机不需要维持复杂的连接状态表。
 - (3) **UDP是面向报文的**。UDP对应用层交下来的报文，既合并，也不拆分，而是保留这些报文的边界。**UDP一次交付一个完整的报文。**
 - (4) **UDP没有拥塞控制**，因此网络出现的拥塞不会使源主机的发送速率降低。这对某些实时应用是很重要的。很适合多媒体通信的要求。选择合适大小的报文
 - (5) **UDP支持一对一、一对多、多对一和多对多的交互通信。**
 - (6) **UDP的首部开销小**，只有8个字节，比TCP的20个字节的首部要短。

5.2.2 UDP的首部格式

- 首部字段很简单，只有8个字节（4个字段，每个字段2个字节）：
 - (1) 源端口
 - (2) 目的端口
 - (3) 长度
 - (4) 检验和
- 计算检验和时，再UDP用户数据报之前增加12个字节的伪首部（仅用于计算检验和）
- IP数据报的检验和**只检验IP数据报的首部**，而UDP的检验和是**把首部和数据部分一起都检验**

5.3 传输控制协议TCP概述

5.3.1 TCP最主要的特点

- (1) **TCP是面向连接的运输层协议。**
- (2) 每一条TCP连接**只能有两个端点**，每一条TCP连接只能是**点对点的（一对一）**。
- (3) TCP提供**可靠交付**的服务。
- (4) TCP提供**全双工通信**。
- (5) **面向字节流**
 - TCP中的“流”(stream)指的是流入或流出进程的字节序列。
 - “面向字节流”的含义是：虽然应用程序和TCP的交互是一次一个数据块，但TCP把应用程序交下来的数据看成仅仅是一连串无结构的字节流。
 - TCP连接是一条虚连接而不是一条真正的物理连接
 - TCP不关心应用进程一次把多长的报文发送到TCP的缓存中
 - TCP根据对方给出的**窗口值**和**当前网络拥塞的程度**来决定一个报文段应包含多少个字节（UDP发送的报文长度是应用进程给出的）

5.3.2 TCP的连接

- TCP把连接作为**最基本的抽象**。
- 每一条TCP连接有**两个端点**。
- TCP连接的端点不是主机，不是主机的IP地址，不是应用进程，也不是运输层的协议端口。
- TCP连接的端点叫做**套接字（socket）**或插口。
- **端口号拼接到（concatenated with）IP地址即构成了套接字。**
- 套接字socket = （IP地址：端口）
- 每一条TCP连接**唯一**地被通信两端的**两个端点**（两个套接字）
- TCP连接 ::= {socket1, socket2} = {(IP1: port1), (IP2: port2)}
- TCP连接的端点是个很抽象的套接字，即（IP地址：端口号）。
- 同一个IP地址可以有多个不同的TCP连接。
- 同一个端口号也可以出现在多个不同的TCP连接中。
- 同一个名词socket可以表示多种不同的意思

可靠传输的工作原理

- 理想的传输条件有以下两个特点：
 - (1) 传输信道不产生差错。
 - (2) 不管发送方以多快的速度发送数据，接收方总是来得及处理收到的数据。
- 然而实际的网络都不具备以上两个理想条件。必须使用一些可靠传输协议，在不可靠的传输信道实现可靠传输。

5.4.1 停止等待协议

- 停止等待：每发送完一个分组就通知发送，等待对方的确认。在收到确认后再次发送下一个分组

1. 无差错情况
2. 出现差错

- 在接收方B会出现两种情况：
 - B接收M1时检测出了差错，就丢弃M1，其他什么也不做（不通知A收到有差错的分组）。
 - M1在传输过程中丢失了，这时B当然什么都不知道，也什么都不做。
 - 在这两种情况下，B都不会发送任何信息。
 - 如何保证B正确收到了M1呢？
 - 解决方法：**超时重传**
 - A为每一个已发送的分组都设置了一个**超时计时器**。
 - A只要在超时计时器到期之前收到了相应的确认，就撤销该超时计时器，继续发送下一个分组M2。
 - A需要暂时保留已发送的分组的副本，且必须有编号，并且超时计时器设置的重传时间应当比数据在分组传输的平均往返时间更长
3. 确认丢失和确认迟到

- 使用上述的确认和重传机制，我们可以在**不可靠的传输网络上实现可靠的通信**（自动重传请求ARQ）
4. 信道利用率

- 为了提高传输效率，发送方可以不使用低效率的停止等待协议，而是采用**流水线传输**。
- **流水线传输**就是发送方可连续发送多个分组，不必每发完一个分组就停顿下来等待对方的确认。这样可使信道上一直有数据不间断地传送。
- 由于信道上一直有数据不间断地传送，这种传输方式可获得很高的信道利用率。

5.4.2 连续ARQ协议

- 滑动窗口协议比较复杂，是 TCP 协议的精髓所在。
- 发送方维持的**发送窗口**，它的意义是：**位于发送窗口内的分组都可连续发送出去，而不需要等待对方的确认**。这样，信道利用率就提高了。
- 连续 ARQ 协议规定，发送方每收到一个**确认**，就把发送窗口向前**滑动一个分组**的位置。
- 接收方一般采用**累积确认**的方式。即不必对收到的分组逐个发送确认，而是**对按序到达的最后一个分组发送确认**，这样就表示：**到这个分组为止的所有分组都已正确收到了**。
- 优点：容易实现，即使确认丢失也不必重传。
- 缺点：不能向发送方反映出接收方已经正确收到的所有分组的信息。
- 如果发送方发送了前 5 个分组，而中间的第 3 个分组丢失了。这时接收方只能对前两个分组发出确认。发送方无法知道后面三个分组的下落，而只好把后面的三个分组都再重传一次。这就叫做 **Go-back-N**（回退 N），**表示需要再退回来重传已发送过的 N 个分组**。

5.5 TCP报文段的首部格式

- TCP 虽然是面向字节流的，但 TCP 传送的数据单元却是**报文段**。
- 一个 TCP 报文段分为**首部**和**数据**两部分，而 TCP 的全部功能都体现在它首部中各字段的作用。
- TCP 报文段首部的 20 个字节是固定的，后面有 4n 字节是根据需要而增加的选项（n 是整数）。因此 TCP 首部的最小长度是 20 字节。
- 首部固定部分各字段的意义：
 - (1) 源端口和目的端口
 - (2) 序号：报文段序号，每一个字节按顺序编号
 - (3) 确认号：期望收到对方下一个报文段的第一个数据字节的序号
 - (4) 数据偏移：
 - (5) 保留：
 - (6) 紧急URG：为1时，表示紧急
 - (7) 确认ACK：为1确认号字段有效
 - (8) 推送PSH：
 - (9) 复位RST
 - (10) 同步SYN
 - (11) 终止FIN
 - (12) 窗口：窗口值作为接收方让发送方设置器发送窗口的依据
 - (13) 检验和
 - (14) 紧急指针
 - (15) 选项：①**最大报文段长度MSS**，每一个TCP报文段中的**数据字段的最大长度**
②窗口扩大选项③时间戳选项④选择确认选项

5.6 TCP可靠传输的实现

5.6.1 TCP以字节为单位的滑动窗口

- 根据 B 给出的窗口值，A 构造出自己的发送窗口。
- 发送窗口表示：在没有收到 B 的确认的情况下，A 可以连续把窗口内的数据都发送出去。
- 发送窗口里面的序号表示允许发送的序号。
- 显然，窗口越大，发送方就可以在收到对方确认之前连续发送更多的数据，因而可能获得更高的传输效率。
- 描述发送窗口的状态：需要三个指针，p1，p2，p3，指针都指向序号
- **收到确认后，发送窗口向前滑动**
- A 的发送窗口内的序号都已用完，但还没有再收到确认，必须**停止发送**。
- 发送窗口中的序号全是发送未确认的或未发送的
- 发送窗口只是发送缓存中的一部分
- **发送缓存**用来暂时存放：
 - 发送应用程序传送给发送方 TCP 准备发送的数据；
 - TCP 已发送出但尚未收到确认的数据。
- **接收缓存**用来暂时存放：
 - 按序到达的、但尚未被接收应用程序读取的数据；
 - 不按序到达的数据。
- 注意：
 - (1) 第一，A 的发送窗口**并不总是**和 B 的接收窗口一样大（因为有一定的**时间滞后**）。
 - (2) 第二，TCP 标准**没有规定**对不按序到达的数据应如何处理。通常是先临时存放在接收窗口中，等到字节流中所缺少的字节收到后，再按序交付上层的应用进程。
 - (3) 第三，TCP 要求接收方必须有**累积确认**的功能，这样可以减小传输开销。
- TCP是全双工通信，每一方都有自己的发送窗口和接受窗口

5.6.2 超时重传时间的选择

- **重传机制**是 TCP 中最重要和最复杂的问题之一。TCP 每发送一个报文段，就对这个报文段设置一次计时器。只要计时器设置的重传时间到但还没有收到确认，就要重传这一报文段。重传时间的选择是 TCP 最复杂的问题之一。
- TCP 采用了一种**自适应算法**，它记录一个报文段发出的时间，以及收到相应的确认的时间。这两个时间之差就是**报文段的往返时间 RTT**。
- TCP 保留了 RTT 的一个**加权平均往返时间 RTTS**（这又称为**平滑的往返时间**）。
- 第一次测量到 RTT 样本时，RTTS 值就取为所测量到的 RTT 样本值。以后每测量到一个新的 RTT 样本，就按下式重新计算一次 RTTS：
- **超时重传时间RTO** (Retransmission Time-Out) 应略大于上面得出的加权平均往返时间 RTTS。
RFC 2988 建议使用下式计算 RTO：
- RTTD 是 RTT 的**偏差的加权平均值**。
RFC 2988 建议这样计算 RTTD。第一次测量时，RTTD 值取为测量到的 RTT 样本值的一半。在以后的测量中，则使用下式计算加权平均的 RTTD：
 β 是个小于 1 的系数，其推荐值是 1/4，即 0.25。
- Karn算法：**在计算平均往返时间 RTT 时，只要报文段重传了，就不采用其往返时间样本**，但是这又引起新的问题。当报文段的时延突然增大了很多时，在原来得出的重传时间内，不会收到确认报文段。于是就重传报文段。但根据 Karn 算法，不考虑重传的报文段的往返时间样本。这样，超时重传时间就无法更新报文段
- 修正后的Karn算法：**报文段每重传一次，就把超时重传的时间RTO增大一点。典型的做法是取新的重传时间为原来的重传时间的2倍**，当不再发生报文段的重传时，才根据报文段的往返时延更新平均往返时延 RTT 和超时重传时间 RTO 的数值。

5.6.3 选择确认SACK

- 接收方收到了和前面的字节流不连续的两个字节块。如果这些字节的序号都在接收窗口之内，那么接收方就先收下这些数据，但要把这些信息准确地告诉发送方，使发送方不要再重复发送这些已收到的数据。

5.7 TCP流量控制

5.7.1 利用滑动窗口实现流量控制

- 流量控制：**让发送方的发送速率不要太快，要让接收方来得及接收**
- 可能出现**死锁**：B向A发送了零窗口的报文段后不久，B的接收缓存又有了一些存储空间。于是B向A发送了 $rwnd = 400$ 的报文段。但这个报文段在传送过程中丢失了。A一直等待收到B发送的非零窗口的通知，而B也一直等待A发送的数据。如果没有其他措施，这种互相等待的**死锁**局面将一直延续下去。
- 为了解决这个问题，TCP为每一个连接设有一个**持续计时器**(persistence timer)。只要TCP连接的一方收到对方的零窗口通知，就启动该持续计时器。若持续计时器设置的时间到期，就发送一个**零窗口探测报文段**（仅携带1字节的数据），而对方就在确认这个探测报文段时给出了现在的窗口值。
 - 若窗口仍然是零，则收到这个报文段的一方就**重新设置**持续计时器。
 - 若窗口不是零，则死锁的僵局就可以打破了。

5.7.2 TCP的传输效率

- 控制TCP报文段的发送时机：
 - 第一种机制是TCP维持一个变量，它等于**最大报文段长度MSS**。只要缓存中存放的数据达到MSS字节时，就组装成一个TCP报文段发送出去。
 - 第二种机制是由发送方的**应用进程指明**要求发送报文段，即TCP支持的推送(push)操作。
 - 第三种机制是发送方的一个**计时器**期限到了，这时就把当前已有的缓存数据装入报文段（但长度不能超过MSS）发送出去。
- Nagle算法**：
 - 若发送应用进程要把发送的数据逐个字节地送到TCP的发送缓存，则发送方就把第一个数据字节先发送出去，把后面到达的数据字节都**缓存**起来。
 - 当发送方收到对第一个数据字符的确认后，再把发送缓存中的所有数据组装成一个报文段发送出去，同时继续对随后到达的数据进行缓存。
 - 只有在收到对前一个报文段的确认后继续发送下一个报文段。
 - 当到达的数据已达到发送窗口大小的一半或已达到报文段的最大长度时，就立即发送一个报文段。
- 糊涂窗口综合征：接收方缓存已满，应用程序一次只读取一个，导致发送方每次只发送一个字节数据，效率降低
 - 解决方法：让接收方等待一段时间，使得接收缓存已有足够空间容纳一个最长的报文段，或等待接收缓存已有一半空闲空间，只要出现这两种情况之一，接收方就发出确认报文，并向发送方通知当前的窗口大小

5.8 TCP的拥塞控制

5.8.1 拥塞控制的一般原理

- 拥塞**：在某段时间，若对网络中某一资源的需求超过了该资源所提供的**可用部分**，网络性能变坏的情况
- 问题的**实质**是整个系统的各个部分（速率，缓存大小等）不匹配
- 拥塞控制**：**防止过多的数据注入到网络中，这样可以使网络中的路由器或链路不致过载**
- 拥塞控制的前提：**网络能够承受现有的网络负荷**，这是一个**全局性**的过程
- 区分流量控制和拥塞控制：流量控制往往是指点对点通信量的控制，是端到端的问题
- 实践证明，拥塞控制是很难设计的，因为它是一个**动态的**（而不是静态的）问题。
- 当前网络正朝着高速化的方向发展，这很容易出现缓存不够大而造成分组的丢失。但分组的丢失是网络发生拥塞的**征兆**而不是原因。
- 在许多情况下，甚至正是拥塞控制本身成为引起网络性能恶化甚至发生死锁的原因。这点应特别引起重视。
- 分类：
 - 开环控制**方法就是在设计网络时事先将有关发生拥塞的因素考虑周到，力求网络在工作时不产生拥塞。
 - 闭环控制**方法是基于反馈环路的概念。属于闭环控制的有以下几种措施：
 - 监测网络系统以便检测到拥塞在何时、何处发生。
 - 将拥塞发生的信息传送到可采取行动的地方。
 - 调整网络系统的运行以解决出现的问题。

5.8.2 TCP的拥塞控制方法

- TCP采用**基于窗口**的方法进行**拥塞控制**。该方法属于**闭环控制**方法。
- TCP发送方维持一个**拥塞窗口CWND**(Congestion Window)拥塞窗口的大小取决于网络的拥塞程度，并且**动态地**在变化。
- 发送端利用拥塞窗口根据网络的拥塞情况调整发送的数据量。
- 所以，发送窗口大小不仅取决于接收方公告的接收窗口，还取决于网络的拥塞状况
- 判断网络拥塞的依据是出现**超时**
- TCP进行拥塞控制的算法：**慢开始，拥塞避免，快重传，快恢复**

1. 慢开始和拥塞避免

- 慢开始算法：当主机开始发送数据时，由于并不清楚网络的负荷情况，所以如果立即把大量数据字节注入到网络，那么就有可能引起网络发生拥塞，经验证明，较好的方法是先探测以下，即**由小到大逐渐增大发送窗口**，也就是说，**由小到大逐渐增大拥塞窗口数值**
- 慢开始规定，在每收到一个**对新的报文段的确认后**，可以把拥塞窗口增加最多一个SMSS（发送方的最大报文段）的数值。
拥塞窗口CWND每次的增加量： $\min(N, SMSS)$
- 使用慢开始算法后，每经过一个**传输轮次**(transmission round)，拥塞窗口cwnd就加倍。
- 一个**传输轮次**所经历的时间其实就是**往返时间RTT**。
- “传输轮次”更加强调：把拥塞窗口cwnd所允许发送的报文段都连续发送出去，并收到了对已发送的最后一个字节的确认。
- 例如，拥塞窗口 $cwnd = 4$ ，这时的往返时间RTT就是发送方连续发送4个报文段，并收到这4个报文段的确认，总共经历的时间。
- 慢开始门限ssthresh**的用法如下：
 - 当 $cwnd < ssthresh$ 时，使用慢开始算法。
 - 当 $cwnd > ssthresh$ 时，停止使用慢开始算法而改用拥塞避免算法。
 - 当 $cwnd = ssthresh$ 时，既可使用慢开始算法，也可使用拥塞避免算法。
- 拥塞避免算法**：让**拥塞窗口cwnd缓慢地增大**，即每经过一个往返时间RTT就把发送方的拥塞窗口cwnd加1，而不是加倍，使拥塞窗口cwnd按**线性规律缓慢增长**。
- 因此在拥塞避免阶段就有**“加法增大”**(Additive Increase)的特点。这表明在拥塞避免阶段，拥塞窗口cwnd按线性规律缓慢增长，比慢开始算法的拥塞窗口增长速率缓慢得多。

2. 快重传算法：

- 采用**快重传FR**(Fast Retransmission)算法可以让发送方早知道发生了个别报文段的丢失。

- 快重传 算法首先要求接收方不要等待自己发送数据时才进行捎带确认，而是要**立即发送确认**，即使收到了失序的报文段也要立即发出对已收到的报文段的重复确认
- 发送方只要一收到三个重复确认，就知道接收方确实没有收到报文段，因而应当立即进行重传（即“快重传”），这样就不会出现超时，发送方也不就会误认为出现了网络拥塞。
- 使用快重传可以使整个网络的吞吐量提高约20%。
- 不难看出，快重传并非取消重传计时器，而是在某些情况下可更早地重传丢失的报文段。

3. 快恢复算法

5.8.3 主动队列管理

5.9 TCP的运输连接管理

- TCP是面向连接的协议，运输连接是用来传送TCP报文的
- 运输连接有三个阶段：
 - 连接建立：采用C/S方式
 - 数据传送
 - 连接释放

5.9.1 TCP的连接建立

- TCP建立连接的过程叫做**握手**
- 握手需要在客户和服务器之间交换三个 TCP 报文段。称之为**三报文握手**。
- 采用三报文握手主要是为了防止已失效的连接请求报文段突然又传送到了，因而产生错误。
- 一开始B的TCP服务器进程先创建传输控制块TCB，A的TCP客户进程也是首先创建传输控制模块TCB
- A的 TCP 向 B 发出连接请求报文段，其首部中的**同步位 SYN = 1**，并**选择序号 seq = x**，表明传送数据时的第一个数据字节的序号是 x。TCP规定SYN报文段不能携带数据，但要消耗一个序号
- B的 TCP 收到连接请求报文段后，如同意，则发回确认。
B在**确认报文段**中应使 SYN = 1，使 ACK = 1，其确认号ack = x + 1，自己选择的序号 seq = y。
- A收到此报文段后向 B 给出确认，其 ACK = 1，确认号 ack = y + 1。A的 TCP 通知上层应用进程，连接已经建立。
- B的 TCP 收到主机 A 的确认后，也通知其上层应用进程：TCP 连接已经建立。

5.9.2 TCP的连接释放

- 数据传输结束后，通信的双方都可释放连接。
- TCP 连接释放过程是**四报文握手**。
- 连接释放报文段首部的**终止控制位FIN**
- 数据传输结束后，通信的双方都可释放连接。
- 现在 A 的应用进程先向其 TCP 发出连接释放报文段，并停止再发送数据，主动关闭 TCP连接。
- A 把连接释放报文段首部的 FIN = 1，其序号seq = u，等待 B 的确认。
- B 发出确认，确认号 ack = u + 1，而这个报文段自己的序号 seq = v。
- TCP 服务器进程通知高层应用进程。
- 从 A 到 B 这个方向的连接就释放了，TCP 连接处于半关闭状态。B 若发送数据，A 仍要接收。
- 若 B 已经没有要向 A 发送的数据，其应用进程就通知 TCP 释放连接。
- A 收到连接释放报文段后，必须发出确认。在确认报文段中 ACK = 1，确认号 ack = w + 1，自己的序号 seq = u + 1，然后进入TIME-WAIT状态。
- 为什么A在TIME-WAIT状态必须等待2MSL：
 - (1) 第一，为了保证 A 发送的最后一个 ACK 报文段能够到达 B。
 - (2) 第二，防止“已失效的连接请求报文段”出现在本连接中。A 在发送完最后一个 ACK 报文段后，再经过时间 2MSL，就可以使本连接持续的时间内所产生的所有**报文段，都从网络中消失**。这样就可以使下一个新的连接中不会出现这种旧日的连接请求报文段。
- 保活计时器

第六章 应用层

- 应用层协议是为了解决某一类应用问题，而问题的解决往往是通过位于不同主机中的多个应用进程之间的通信和协同工作来完成的。
- 应用层的具体内容就是规定应用进程在通信时所遵循的协议
- 应用层的许多协议都是基于C/S模式，客户和服务器都是通信中所涉及的两个应用进程

6.1 域名系统DNS

6.1.1 域名系统概述

- 许多应用层软件经常直接使用**域名系统 DNS** (Domain Name System)，但计算机的用户只是**间接**而不是直接使用域名系统。
- 互联网采用层次结构的命名树作为主机的名字，并使用**分布式的**域名系统 DNS。
- 名字到 IP 地址的解析是由若干个域名服务器程序完成的。域名服务器程序在专设的结点上运行，运行该程序的机器称为**域名服务器**。

- 互联网采用了**层次树状结构**的命名方法。
- 任何一个连接在互联网上的主机或路由器，都有一个**唯一**的层次结构的名字，即**域名**。
- 域名的结构由标号序列组成，各标号之间用点隔开：
... . 三级域名 . 二级域名 . 顶级域名
- 各标号分别代表不同级别的域名。
- 域名只是个**逻辑概念**，并不代表计算机所在的物理地点。
- 变长的域名和使用有助记忆的字符串，是为了便于人来使用
- 每一个域名中“点”的数目则不一定正好是三个

- 顶级域名：
 - (1) 国家顶级域名 nTLD
 .cn表示中国
 .us表示美国
 .uk表示英国
 等等
 - (2) 通用顶级域名 gTLD
 最早的顶级域名
 .com公司和企业
 .net网络服务机构
 .org非盈利性组织
 .edu美国专用教育机构
 .gov美国专用政府部门
 .mil美国专用军事部门
 .int国际组织
 新增通用顶级域名
 .aero
 .biz
 .cat
 .name
 等等
 - (3) 基础结构域名
 这种顶级域名只有一个，即**arpa**
 用于反响域名解析，因此又称为**反向域名**

6.1.3 域名服务器

- 一个服务器所管辖的范围叫做**区**
- 各单位根据具体情况来**划分**自己管辖范围的区。但在一个区中的所有节点必须是能够连通的。
- 每一个区设置相应的**权限域名服务器**，用来保存该区中的所有主机的域名到 IP 地址的映射。
- DNS 服务器的管辖范围不是以“域”为单位，而是以“**区**”为单位。
- 域名服务器有以下四种类型：
 - **根域名服务器**
 - 最高层次的域名服务器，也是最重要的域名服务器。所有的根域名服务器都知道所有的顶级域名服务器的域名和 IP 地址。
 - 不管是哪一个本地域名服务器，若要对互联网上任何一个域名进行解析，只要自己无法解析，就首先求助于根域名服务器。
 - 根域名服务器并不直接把域名直接转换成 IP 地址。
 - 在使用迭代查询时，根域名服务器把下一步应当找的顶级域名服务器的 IP 地址告诉本地域名服务器。
 - **根域名服务器共有 13 套装置，不是 13 个机器** (a~m)
 - **顶级域名服务器**
 - 顶级域名服务器（即 TLD 服务器）负责管理在该顶级域名服务器注册的所有**二级域名**。
 - **权限域名服务器**
 - 负责一个区的域名服务器
 - **本地域名服务器**
 - 当一个主机发出 DNS 查询请求时，这个查询请求报文就发送给本地域名服务器
 - 这种域名服务器有时也称为**默认域名服务器**；
- 提高域名服务器的可靠性：
 - DNS 域名服务器都把**数据复制**到几个域名服务器来保存，其中的一个是**主域名服务器**，其他的是**辅助域名服务器**。
 - 当主域名服务器出故障时，辅助域名服务器可以保证 DNS 的查询工作不会中断。
 - 主域名服务器**定期**把数据复制到辅助域名服务器中，而更改数据**只能在主域名服务器**中进行。这样就保证了数据的一致性。
- 域名解析过程：
 - 主机向**本地域名服务器**的查询一般都是采用**递归查询**。如果主机所询问的本地域名服务器不知道被查询域名的 IP 地址，那么本地域名服务器就以 DNS 客户的身份，向其他根域名服务器继续发出查询请求报文。
 - 本地域名服务器向**根域名服务器**的查询通常是采用**迭代查询**。当根域名服务器收到本地域名服务器的迭代查询请求报文时，要么给出所要查询的 IP 地址，要么告诉本地域名服务器：“你下一步应当向哪一个域名服务器进行查询”。然后让本地域名服务器进行后续的查询。
- **每个域名服务器都维护一个高速缓存**，存放**最近用过的名字**以及从何处获得名字映射信息的记录

6.2 文件传送协议

6.2.1 FTP概述

- **文件传送协议 FTP** (File Transfer Protocol) 是互联网上使用得最广泛的文件传送协议。
- FTP 提供**交互式**的访问，允许客户指明文件的类型与格式，并允许文件具有存取权限。
- FTP 屏蔽了各计算机系统的细节，因而适合于在异构网络中任意计算机之间传送文件。

- 文件传送非常困难。原因是众多的计算机厂商研制出的文件系统多达数百种，且差别很大

6.2.2 FTP的基本工作原理

- 网络环境下复制文件的复杂性：
 - (1) 计算机存储数据的格式不同。
 - (2) 文件的目录结构和文件命名的规定不同。
 - (3) 对于相同的文件存取功能，操作系统使用的命令不同。
 - (4) 访问控制方法不同。
- FTP的特点：
 - (1) 只提供文件传送的一些基本的服务
 - (2) 使用TCP可靠的运输服务
 - (3) 减少消除在不同操作系统下处理文件的不兼容性
 - (4) FTP使用C/S模式。一个FTP服务器进程可以同时为多个客户进程提供服务。
 - (5) FTP的服务器进程有两大部分：
一个主进程，负责接受新的请求；
若干个从属进程，负责处理单个请求
- 两个连接：
 - 控制连接：控制进程
 - 数据连接：数据传送进程
- 当客户进程向服务器进程发出建立连接请求时，要寻找连接服务器进程的**熟知端口**（21），同时还要告诉服务器进程自己的**另一个端口号码**，用于建立数据传送连接。接着，服务器进程用自己传送数据的熟知端口（20）与客户进程所提供的端口号码建立数据传送连接。由于FTP使用了**两个不同的端口号**，所以数据连接与控制连接不会发生混乱。
- 使用两个不同端口号：
 - 使协议更加简单和更容易实现。
 - 在传输文件时还可以利用控制连接（例如，客户发送请求终止传输）。
- NFS：
 - NFS 允许应用进程打开一个远地文件，并能在该文件的某一个特定的位置上开始读写数据。
 - NFS 在网络上传送的只是**少量的修改数据**

6.2.3 简单文件传送协议TFTP

- TFTP (Trivial File Transfer Protocol) 是一个很小且易于实现的文件传送协议。
- TFTP 使用**客户服务器方式**和使用 **UDP** 数据报，因此 TFTP 需要有自己的差错改正措施。
- TFTP 只支持文件传输而不支持交互。
- TFTP 没有一个庞大的命令集，没有列目录的功能，也不能对用户进行身份鉴别。
- 特点：
 - (1) 每次传送的数据 PDU 中有 512 字节的数据，但最后一次可不足 512 字节。
 - (2) 数据 **PDU** 也称为**文件块** (block)，每个块按序编号，从 1 开始。
 - (3) 支持 ASCII 码或二进制传送。
 - (4) 可对文件进行读或写。
 - (5) 使用很简单的首部。
- TFTP 的工作很像停止等待协议

6.3 远程终端协议TELNET

- **TELNET** 是一个简单的**远程终端协议**，也是互联网的正式标准。
- 用户用 TELNET 就可在其所在地通过 TCP 连接注册（即登录）到远地的另一个主机上（使用主机名或 IP 地址）。
- TELNET 也使用**客户-服务器方式**。在本地系统运行 TELNET 客户进程，而在远地主机则运行 TELNET 服务器进程。
- 和 FTP 的情况相似，服务器中的**主进程**等待新的请求，并产生**从属进程**来处理每一个连接
- TELNET 使用网络虚拟终端 NVT 格式

6.4 万维网WWW

6.4.1 万维网概述

- **万维网** WWW (World Wide Web) 并非某种特殊的计算机网络。
- 万维网是一个大规模的、联机式的**信息储藏所**。
- 万维网用**链接**的方法能非常方便地从互联网上的一个站点访问另一个站点，从而主动地按需获取丰富的信息。这种访问方式称为“**链接**”。
- 万维网提供**分布式服务**：
- 万维网是**分布式超媒体** (hypermedia) 系统，它是**超文本**系统的扩充。
- 一个**超文本**由多个**信息源链接**成。利用一个链接可使用户找到另一个文档。这些文档可以位于世界上任何一个接在互联网上的超文本系统中。超文本是万维网的基础。
- **超媒体与超文本的区别是文档内容不同**。超文本文档仅包含文本信息，而超媒体文档还包含其他表示方式的信息，如图形、图像、声音、动画，甚至活动视频图像。
- 万维网的工作方式：
 - (1) 万维网以**客户-服务器方式**工作。
 - (2) **浏览器**就是在用户计算机上的万维网**客户程序**。万维网**文档**所驻留的计算机则运行**服务器程序**，因此这个计算机也称为**万维网服务器**。
 - (3) 客户程序向服务器程序发出请求，服务器程序向客户程序送回客户所要的**万维网文档**。
 - (4) 在一个客户程序主窗口上显示出的万维网文档称为**页面** (page)。
- 万维网解决的问题：
 - 怎样标志分布在互联网上的万维网文档？
 - 使用**统一资源定位符 URL** 来标志万维网上的各种文档。
 - 使每一个文档在整个互联网的范围内具有**唯一**的标识符 URL。
 - 用何协议实现万维网上各种超链的链接？
 - 在万维网客户程序与万维网服务器程序之间进行交互所使用的协议，是**超文本传送协议 HTTP**。

- HTTP (HyperText Transfer Protocol) 是一个**应用层协议**，它使用 **TCP 连接**进行可靠的传送。
- 怎样使各种万维网文档都能在互联网上的各种计算机上显示出来，同时使用户清楚地知道在什么地方存在着超链？
 - **超文本标记语言 HTML** 使得万维网页面的设计者可以很方便地用一个超链从本页面的某处链接到互联网上的任何一个万维网页面，并且能够在自己的计算机屏幕上将这些页面显示出来。
- 怎样使用户能够很方便地找到所需的信息？
 - 为了在万维网上方便地查找信息，用户可使用各种的**搜索工具**（即**搜索引擎**）。

6.4.2 同一资源定位符URL

1.URL的格式：URL 相当于一个文件名在网络范围的扩展

- 字符对大小写无要求
- `://`是格式规定
- `<主机>`是存放资源的主机在互联网中的域名
- `<端口>/<路径>` 有时可以省略（http的默认端口80）

6.4.3 超文本传送协议HTTP

1. HTTP的操作过程

- HTTP是**面向事务**的应用层协议

2. HTTP的主要特点

- HTTP是**面向事务的客户-服务器协议**
- HTTP协议是**无状态，持续连接**的
 - 持续连接有两种工作方式：非流水线方式；流水线方式
- HTTP协议本身也是**无连接**的，他使用了面向连接的TCP向上提供的服务

3. 代理服务器：

- **代理服务器** (proxy server) 又称为**万维网高速缓存** (Web cache)，它代表浏览器发出 HTTP 请求。
- 万维网高速缓存把**最近**的一些请求和响应暂存在本地磁盘中。
- 当与暂时存放的请求**相同**的新请求到达时，万维网高速缓存就**把暂存的响应发送出去**，而不需要按 URL 的地址再去互联网访问该资源
- 使用高速缓存可以**减少**访问互联网服务器的时延

4. HTTP的报文结构：

- HTTP有两种报文：
 - 请求报文
 - 响应报文
- HTTP是面向正文的，因此在报文中的每一个字段都是一些ASCII码串
- 报文由三个部分组成，即**开始行、首部行和实体主体**。
- 在**请求报文**中，开始行就是请求行。
- “**方法**”是面向对象技术中使用的专门名词。所谓“方法”就是**对所请求的对象进行的操作**，因此这些方法实际上也就是一些**命令**。因此，请求报文的类型是由它所采用的方法决定的。
- 在**响应报文**中，开始行就是状态行
- 状态行包括三项内容，即 **HTTP 的版本，状态码**，以及解释状态码的**简单短语**。
- 状态码都是三位数字：
 - 1xx 表示通知信息**的，如请求收到了或正在进行处理。
 - 2xx 表示成功**，如接受或知道了。
 - 3xx 表示重定向**，表示要完成请求还必须采取进一步的行动。
 - 4xx 表示客户的差错**，如请求中有错误的语法或不能完成。
 - 5xx 表示服务器的差错**，如服务器失效无法完成请求。

4. 在服务器上存放用户的信息

- 万维网站点使用 **Cookie** 来跟踪用户。
- Cookie 表示在 HTTP 服务器和客户之间传递的**状态信息**。
- 使用 Cookie 的网站服务器为用户产生一个**唯一的识别码**。利用此识别码，网站就能够跟踪该用户在该网站的活动。

6.4.4 万维网的文档

1. 超文本标记语言HTML

- 超文本标记语言 HTML 中的 Markup 的意思就是“设置标记”。
- HTML 定义了许多用于**排版的命令**（即标签）。
- HTML 把各种标签嵌入到万维网的页面中。这样就构成了所谓的 HTML 文档。HTML 文档是一种可以用任何文本编辑器创建的 ASCII 码文件。
- 仅当 HTML 文档是以 .html 或 .htm 为后缀时，浏览器才对此文档的各种标签进行解释
- HTML还规定了链接的设置方法。每个链接都有一个起点和终点。
- **远程链接**：超链的终点是其他网点上的页面。
- **本地链接**：超链指向本计算机中的某个文件
- **XML**：
- XML (Extensible Markup Language)是**可扩展标记语言**，它和HTML很相似。
- 但XML的设计宗旨是**传输数据**，而不是显示数据（HTML是为了在浏览器上显示数据）。
- XML 不是要替换 HTML，而是对 HTML 的补充
- XHTML：是可扩展超文本标记语言，是更为严格的HTML版本
- CSS是**层叠样式表**，用于为 HTML 文档定义布局。
 - CSS 与 HTML 的区别就是：HTML 用于结构化内容，而 CSS 则用于格式化结构化的内容

2. 动态万维网文档：

- **静态文档**是指该文档创作完毕后就存放在万维网服务器中，在被用户浏览的过程中，内容不会改变。
- **动态文档**是指文档的内容是在浏览器访问万维网服务器时才由应用程序动态创建。
- 动态文档和静态文档之间的主要差别体现在**服务器**一端。这主要是**文档内容的生成方法不同**。而从浏览器的角度看，这两种文档并没有区别。
- **网关接口CGI**：定义了动态文档应如何创建，输入数据应如何提供给应用程序，以及输出结果应如何使用
- CGI 程序的正式名字是 **CGI 脚本** (script)或cgi-bin脚本。
- “脚本”指的是一个程序，它被另一个程序（解释程序）而不是计算机的处理机来解释或执行。
- 脚本运行起来要比一般的编译程序要慢，因为它的每一条指令先要被另一个程序来处理（这就要一些附加的指令），而不是直接被指令处理器来处理

3. 活动万维网文档

- **活动文档** (active document) 技术把所有的工作都**转移给浏览器端**，提供屏幕连续更新技术。

6.5 电子邮件

6.5.1 电子邮件概述

- **电子邮件** (e-mail) 是互联网上使用得最多的和最受用户欢迎的一种应用
- 电子邮件把邮件发送到收件人使用的邮件服务器，并放在其中的收件人**邮箱**中，收件人可随时上网到自己使用的邮件服务器进行读取
- 传递迅速，费用低廉
- 发送邮件的协议：**简单邮件传送协议SMTP**（7位ASCII码邮件）
读取邮件的协议：**邮局协议POP3** 和 **IMAP**
- **通用互联网邮件扩充MIME** 在其邮件首部中说明了邮件的数据类型(如文本、声音、图像、视像等)，使用 MIME 可在邮件中同时传送多种类型的数据。
- 一个电子邮件系统应该具有三个主要组成构件：
 - **用户代理**
 - **邮件服务器**
 - **邮件发送协议，邮件读取协议**
- 用户代理UA（电子邮件客户端软件）：
 - 用户代理的功能：**撰写，显示，处理，通信**
 - 邮件服务器按照**客户-服务器方式**工作
- 电子邮件的组成：
 - **信封**
 - **内容**
- 电子邮件的传输程序根据邮件信封上的信息来传送邮件。用户在从自己的邮箱中读取邮件时才能见到邮件的内容
- TCP/IP 体系的电子邮件系统规定**电子邮件地址**的格式如下：
收件人邮箱名@邮箱所在主机的域名
- 符号“@”读作“at”，表示“在”的意思。

6.5.2 简单邮件传送协议SMTP

- SMTP 所规定的就是在**两个相互通信的 SMTP 进程之间应如何交换信息**。
- 由于 SMTP 使用**客户服务器方式**，因此负责发送邮件的 SMTP 进程就是 SMTP 客户，而负责接收邮件的 SMTP 进程就是 SMTP 服务器。
- SMTP 规定了 14 条命令和 21 种应答信息。每条命令用 4 个字母组成，而每一种应答信息一般只有一行信息，由一个 3 位数字的代码开始，后面附上（也可不附上）很简单的文字说明
- SMTP通信的三个阶段：
 1. **连接建立**：**SMTP不使用中间的邮件服务器**
 2. **邮件传送**
 3. **连接释放**：邮件发送完毕，SMTP应释放TCP连接

6.5.3 电子邮件的信息格式

- 一个电子邮件分为**信封**和**内容**两大部分。
- RFC 822 只规定了邮件**内容**中的**首部(header)**格式，而对邮件的**主体(body)**部分则让用户自由撰写。
- 用户写好首部后，邮件系统将自动地将信封所需的信息提取出来并写在信封上。所以用户不需要填写电子邮件信封上的信息。
- 邮件内容首部包括一些关键字，后面加上冒号。最重要的关键字是：To（写到地址簿）和 Subject（主题）。
- 其他还有：
 - “Cc”：抄送，复写副本；
 - “Bcc”：盲复写副本，暗送；
 - “From”：发送人的电子邮件地址
 - “Date”：发信日期

6.5.4 邮件读取协议POP3和IMAP

- 邮局协议POP使用C/S方式
- 网际报文存取协议IMAP：C/S方式，联机协议

6.5.5 基于万维网的电子邮件

- 浏览器和互联网上的邮件服务器之间传送邮件时，使用HTTP协议
- 个邮件服务器之间传送邮件时，使用SMTP协议

6.6 动态主机配置协议DHCP

- 为了将软件协议做成通用的和便于移植，协议软件的编写者把协议软件参数化，一台计算机和另一台计算机的区别，都可通过一些不同的参数来体现，在协议软件中给这些参数赋值的动作叫做**协议配置**
- 互联网广泛使用的**动态主机配置协议 DHCP**，提供了**即插即用连网**的机制。
- 需要 IP 地址的主机在启动时就向 DHCP 服务器**广播发送发现报文**
- 本地网络上所有主机都能收到此广播报文，但只有 DHCP 服务器才回答此广播报文
- DHCP 服务器的回答报文叫做**提供报文**
- DHCP **中继代理**，它配置了 DHCP 服务器的 IP 地址信息

