# 计算机网络

## 第1章 计算机网络体系结构

### 1.1 计算机网络概述

#### 1.1.1 计算机网络的概念、组成和功能

计算机网络最简单的定义：一些互相连接的、自治的计算机的集合。

另一种定义：按照网络协议，以共享资源为主要目的，将地理上分散且独立的计算机互相连接起来形成的集合体。

一个计算机网络应当有三个主要组成部分：若干主机，一个通信网，一系列协议。

计算机网络的三大功能：数据通信，资源共享，分布式处理。

#### 1.1.2 计算机网络的分类

按网络的覆盖范围分类，有广域网WAN(Wide Area Network)，城域网MAN(Metropolitan Area Network)，局域网LAN(Local Area Network) ，个人区域网PAN(Personal Area Network)。

按网络的使用者分类，有公用网(public network)和专用网(private network)。

按传输介质分类，可分为有线网络，无线网络和混合网。

按拓扑结构分类，有星形、总线形、环形和不规则形网。

#### 1.1.3 计算机网络与互联网的发展历史

以小写字母i开头的internet（互联网）是一个通用名字，泛指有多个计算机互联而成的网络。

以大写字母I开头的Internet（因特网）是一个专用名词，指当今全球最大的互联网，其前身是美国的ARPNET。

#### 1.1.4 计算机网络的标准化工作及相关组织

1992年由于因特网不再归美国政府管辖，因此成立了一个国际性组织叫做因特网协会(Internet Society，简称ISOC)，以便对因特网进行全面管理以及在世界范围内促进其发展和使用。ISOC下面有一个技术组织叫做因特网体系结构委员会IAB(Internet Architecture Board)，负责管理因特网有关协议的开发。IAB下面又设有两个工程部：因特网工程部IETF(Internet Engineering Task Force)和因特网研究部IRTF(Internet Research Task Force)。

所有的因特网标准都是以**RFC**(Request For Comments)的形式在因特网上发表。RFC上升为因特网标准要经过以下四个阶段：

(1)因特网草案(Internet Draft)——在这个阶段还不是RFC文档。

(2)建议标准 (Proposed Standard)——从这个阶段开始就成为RFC文档。

(3)草案标准(Draft Standard)。

(4)因特网标准(Internet Standard)。

### 1.2 计算机网络体系结构与参考模型

#### 1.2.1 计算机网络分层结构

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。

计算机网络的各层及其协议的集合，称为网络的体系结构(architecture)。

体系结构是抽象的，而实现(implementation)则是具体的。

#### 1.2.2协议、接口、服务等概念

**实体**表示任何可发送或接收信息的硬件或软件进程。在许多情况下，实体就是一个软件模块。

**协议**是控制两个对等实体进行通信的规则的集合。协议的三要素：语法、语义、同步。

在同一结点中相邻两层实体之间的通信规则称为**接口**。下层通过接口向上层提供服务。

**服务**是各层向它的直接上层提供的一组原语或操作。

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，还需要使用下面一层所提供的服务。

协议和服务在概念上是很不一样的。首先，使用本层服务的实体只能看见服务而无法看见下面的协议，下面的协议对上面的实体是透明的。其次，协议是“水平”的，而服务是“垂直”的。

OSI参考模型把对等层次之间传递的数据单位称为**协议数据单元**(Protocol Data Unit)，把相邻两层之间传递的数据单位称为**服务数据单元**(Service Data Unit)。

#### 1.2.3 OSI参考模型和TCP/IP模型

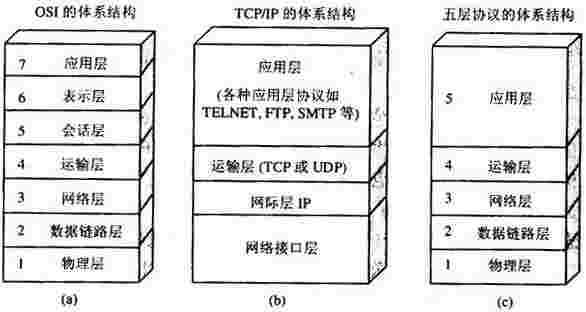


图1.1 计算机网络体系结构：(a)OSI的七层协议；

(b)TCP/IP的四层协议；(c)五层协议

物理层的数据单位为比特（bit）；

数据链路层的数据单位为帧（frame）；

网络层的数据单位为分组或数据包（packet）；

传输层的数据单位为报文或数据段（segment）。

## 第2章 物理层

### 2.1 数据通信基础知识

#### 2.1.1 数据通信系统的模型

一个数据通信系统可以划分为三大部分，即源系统，传输系统和目标系统。

源系统一般包括以下两个部分：

源点(source)：源点设备产生要传输的数据。源点又称为源站，或**信源**。

发送器：通常源点生成的数据比特流要通过发送器编码后才能够在传输系统中进行传输。典型的发送器就是调制器。

目的系统一般也包括以下两个部分：

接收器：接收传输系统传送过来的信号，并把它转化为能够被目的设备处理的信息。典型的接收器就是解调器。

终点(destination)：终点设备从接收器获取传送来的数据比特流，然后把信息输出。终点又称为目的站，或**信宿**。

通信的目的是传送**消息**(message)，**数据**(data)是运送消息的实体。**信号**(signal)则是数据的电气的或电磁的表现。

信号分为模拟信号和数字信号。

#### 2.1.2 有关信道的几个基本概念

信道和电路并不等同。**信道**(channel)一般用来表示向某一个方向传送信息的媒体。

通信模式：单工通信，半双工通信，全双工通信。

通信方式：同步通信，异步通信。

传输方式：串行传输，并行传输。

任何实际的信道所能传输的信号频率都有一定的范围，这个范围称为该信道频带的宽度，或称为**带宽**。

表征一个信道传输数字信号能力的指标为**数据速率**，有时也称为该信道的容量，是以信道每秒所能传送的比特位为单位的。

在数字传输系统中还使用**码元速率**，指的是每秒信号状态变化的次数，以**波特**为单位。一个以b波特传送信号的线路，其传送二进制数据的速率不一定是b比特/秒，因为每个信号可以运载几个比特。例如，若使用0、1、2、3、4、5、6和7共8个电平级，则每个信号值可以代表3个比特，因而这种条件下比特率将是波特数的3倍。

#### 2.1.3 信道的极限容量

奈奎斯特(H. Nyquist)证明，如果一个任意的信号通过带宽为H的低通滤波器，那么每秒采样2H次就能完整地重现通过这个滤波器的信号。如果被传信号电平分为V级，**奈奎斯特定理**表明：最大数据传输率=2Hlog2V(bps)。

由奈奎斯特定理，我们可以得到以下结论：

（1）在任何信道中，码元的传输速率是有上限的，传输速率超过此上限，就会出现严重的码间串扰问题，使接收端对码元的完全正确识别成为不可能。

（2）信道的频带越宽（也就是能够通过的信号的高频分量越多），那么就可以用更高的速率进行码元的有效传输。

（3）奈奎斯特定理给出了码元传输速率的限制，没有对信息传输速率给出限制。

**香农(Shannon)定理**：对任何带宽为H赫兹、信噪比为S/N的信道，最大数据传输率=Hlog2(1+S/N)。

信噪比常用分贝(dB)表示，在数值上=10log10(S/N)(dB)。

由香农公式可以得到以下结论：

（1）信道的带宽或信噪比越大，信息的极限传输速率就越高。

（2）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方法实现无差错传输。但香农定理并没有提供具体实现方法。

（3）对一定的传输带宽和一定的信噪比，信息传输速率的上限制就确定了。

（4）此公式得到的仅为极限信息传输速率，实际上信道的传输速率比这个结果低。

#### 2.1.4 编码与调制

模拟数据可以用模拟信号传输，也可以用数字信号传输；同样，数字数据可以用数字信号传输，也可以用模拟信号传输。这样就构成了4种编码方式，即模拟信号传输模拟数据，模拟信号传输数字数据，数字信号传输数字数据和数字信号传输模拟数据。例如，在使用调制解调器通过电话线路传输计算机数据时所采用的是模拟信号传输数字数据编码方式，在局域网内计算机与计算机之间传输数据时采用的是数字信号传输数字数据。

**1. 模拟信号传输数字数据。**

来自信源的信号常称为**基带信号**。基带信号往往包含有许多低频成分，甚至直流成分，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)。

调制可分为两大类。一类是仅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。这类调制称为**基带调制**。另一类则需要使用载波进行调制，把基带信号的频率范围搬移到较高的频段以便在信道中传输。经过载波调制后的信号称为**带通信号**（即仅在一段频率范围内能够通过信道），而使用载波的调制称为**带通调制**。

最基本的带通调制方法有：

(1) 调幅(AM)，即载波的振幅随基带数字信号而变化。

(2) 调频(FM)，即载波的频率随基带数字信号而变化。

(3) 调相(PM)，即载波的初始相位随基带数字信号而变化。

为了达到更高的数据传输速率，必须采用技术上更为复杂的多元制的振幅相位混合调制方法。例如，正交振幅调制QAM(Quadrature Amplitude Modulation)。

**例1** 在无噪声情况下，若某通信链路的带宽为3kHz，采用4个相位，每个相位具有4种振幅的QAM调制技术，则该通信链路的最大数据传输速率是\_\_\_\_。

A. 12kbps B. 24kbps C. 48kbps D. 96kbps

**[解答]**B。首先，如果带宽为H，在每个采样周期中被传信号可分为V个状态，那么奈氏定理表明：最大数据传输率=2Hlog2V位/秒。本题中，通信链路的带宽为3kHz，最大采样速率是每秒6000次。

其次，QAM是一种多元制的振幅相位混合调制方法，本题中有4种相位，每种相位各有4种振幅，共16个状态，所以是4位/波特。由于每秒6000次的最大采样速率对应6000波特的信号传输率，因此数据传输率为4×6000=24kbps。

**2. 数字信号传输数字数据**

编码方案有不归零制、曼彻斯特编码、差分曼彻斯特编码和4B/5B编码等。

（1）不归零制（NRZ）

用低电平表示0，高电平表示1；或者反过来。

（2）曼彻斯特编码

前T/2高电平，后T/2低电平表示“1”，0相反；或者反过来。

（3）差分曼彻斯特编码

比特周期开始有跳变表示0，无跳变表示1；或者反过来。

曼彻斯特编码提取每个比特中间的电平跳变作为收发双方的同步信号，无需额外的同步信号，因此曼彻斯特编码是一种“自含时钟编码”的编码方式；但是曼彻斯特编码所占的频带宽度比原始的基带信号增加了一倍，在给定的带宽的信道上，编码前后比特率相差两倍。

**3. 数字信号传输模拟数据**

最常见的例子是用于音频信号的脉码调制(Pulse Code Modulation, PCM)。它主要包括三个步骤，即抽样，量化，编码。

#### 2.1.5 电路交换、报文交换与分组交换

电路交换——整个报文的比特流连续的从源点直达终点，好像在一个管道中传送。

报文交换——整个报文先传送到相邻结点，全部存储下来后查找转发表，转发到下一个结点。

分组交换——单个分组（这只是整个报文的一部分）。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。

#### 2.1.6 数据报与虚电路

分组交换还可以进一步分成数据报和虚电路两种交换类型。

数据报分组交换为每一个分组独立地选择路由；而在虚电路分组交换中，第一个分组决定随后所有分组都要遵从的路由。

### 2.2 传输介质

#### 2.2.1 传输介质简介

**1. 导向传输介质**

**（1）双绞线**

把两根互相绝缘的铜导线并排放在一起，然后用规则的方法绞合(twist)起来就构成了双绞线。

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。距离太长时就要加放大器以便将衰减了的信号放大到合适的数值（对于模拟传输），或者加上中继器以便将失真了的数字信号进行整形（对于数字传输）。导线越粗，其通信距离就越远，但导线的价格也越高。在数字传输时，若传输速率为每秒几个兆比特，则传输距离可达几公里。由于双绞线的价格便宜且性能也不错，因此使用十分广泛。

为了提高双绞线的抗电磁干扰能力，可以在双绞线的外面再加上一层用金属丝编织成的屏蔽层。这就是屏蔽双绞线，简称为STP(Shielded Twisted Pair)。它的价格当然比无屏蔽双绞线UTP(Unshielded Twisted Pair)要贵一些。图2.2是无屏蔽双绞线和屏蔽双绞线的示意图。

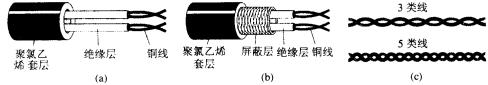


图2.2 双绞线的示意图：(a) STP; (b)UTP; (c)不同绞合程度的双绞线

1991年，美国电子工业协会EIA(Electronic Industries Association)和电信行业协会TIA(Telecommunications Industries Association)联合发布了一个标准EIA/TIA-586，它的名称是“商用建筑物电信布线标准”(Commercial Building Telecommunications Cabling Standard)。这个标准规定了用于室内传送数据的无屏蔽双绞线和屏蔽双绞线标准。

无论是那种类别的线，衰减都随频率的升高而增大（即频率越高，衰减越快）。使用更粗的导线可以降低衰减，但却增加了导线的价格和重量。

**（2）同轴电缆**

同轴电缆由内导体铜质芯线（单股实心线或多股绞合线）、绝缘层、网状编织的外导体屏蔽层（也可以是单股）以及保护塑料外层所组成（图2.3）。

Snap2.jpg

图2.3同轴电缆的结构

由于外导体屏蔽层的作用，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。

在局域网的发展初期曾广泛的使用同轴电缆作为传输媒体，但随着技术的进步，在局域网领域基本上都是采用双绞线作为传输媒体。目前同轴电缆主要用在有线电视网的居民小区中。

**（3）光缆**

光纤通信就是利用光导纤维传递光脉冲来进行通信。有光脉冲相当于1，无光脉冲相当于0.

可以存在许多条不同入射角度的光线在同一条光线中传输，这种光纤称为多模光纤。若光纤的直径减小到只有一个光波的波长，则光线不会产生反射，而是在光纤中直线传播，这种光纤称为单模光纤。

**2. 无线传输介质**

短波通信主要是靠电离层的反射。

无线电微波通信在数据通信中占有重要地位。微波的频率范围为300MHZ~300GHZ（波长1m~1cm），但主要是使用2~40GHZ的频率范围。微波在空间主要是直线传播。由于微波会穿透电离层而进入宇宙空间，因此它不像短波那样可以经电离层反射传播到地面上很远的地方。传统的微波通信主要有两种方式：地面微波接力通信和卫星通信。

卫星通信的优点是容量大，距离远，缺点是传输时延长。卫星通信非常适合于广播通信，因为它的覆盖面很广。

红外线和激光可用于近距离的数据通信。

#### 2.2.2 物理层接口的特性

（1）机械特性：主要定义物理连接的边界点，即接插装置。规定物理连接时所采用的规格、引脚的数量和排列情况。

（2）电气特性：规定传输二进制位时，线路上信号的电压高低、阻抗匹配、传输速率和距离限制。

（3）功能特性：主要定义各条物理线路的功能。

（4）过程特性：主要定义各条物理线路的工作规程和时序关系。

### 2.3 数字传输系统

#### 2.3.1 脉码调制PCM体制

在数字传输系统中的脉码调制PCM(Pulse Code Modulation)体制最初是为了在电话局之间的中继器上传送多路电话。由于历史上的原因，PCM有两个互不兼容的国际标准，即北美的24路PCM（简称为T1）和欧洲的30路PCM（简称为E1）。我国采用的是欧洲的E1标准。E1的速率是2.048Mb/s，而T1的速率是1.544Mb/s。

#### 2.3.2 同步光纤网SONET和同步数字系列SDH

PCM的两个主要缺点：（1）速率标准不统一；（2）不是同步传输。

为解决上述问题，美国在1984年首先提出了一个数字传输标准，叫做**同步光纤网SONET**(Synchronous Optical Network)。

ITU-T以美国标准SONET为基础，制定出国际标准**同步数字系列SDH**(Synchronous Digital Hierarchy)。

### 2.4 宽带接入技术

#### 2.4.1 xDSL技术

xDSL技术就是**用数字技术对现有的模拟电话用户线进行改造**，使它能够承载宽带业务。虽然标准模拟电话信号的频带被限制在300-3400Hz的范围内，但用户线本身实际可通过的信号频率仍然超过1MHZ。因此xDSL技术就把0~4kHz低端频谱留给传统电话使用，而把原来没有被利用的高端频谱留给了用户上网使用。DSL就是数字用户线(Digital Subscribe Line)的缩写。而DSL的前缀x则表示在数字用户线上实现的不同宽带方案。

下面仅对ADSL(Asymmetric Digital Subscriber Line，非对称数字用户线)进行简单介绍。

由于用户在上网时主要是从因特网下载各种文档，而向因特网发送的信息一般都不大，因此ADSL把上行和下行带宽做成不对称的。ADSL在用户线（铜线）的两端各安装一个ADSL调制解调器。这种调制解调器的实现方案有多种。我国目前采用的方案是**离散多音调DMT**(Discrete Multi-Tone)调制技术。

ADSL最大的好处就是可以利用现有的电话网的用户线，不需要重新布线。

ADSL技术也在发展。现在ITU-T已经颁布了更高速率的ADSL标准

#### 2.4.2 光纤同轴混合网(HFC网)

HFC是Hybrid Fiber Coax的缩写。HFC网是在目前覆盖面很广的有线电视网CATV的基础上开发的一种居民宽带接入网。

#### 2.4.3 FTTx技术

光纤到户FTTH(Fiber To The Home)，光纤到大楼FTTB(Fiber To The Building)，光纤到路边FTTC(Fiber To The Curb)

### 2.5 物理层设备

#### 2.5.1 中继器(repeater)

中继器是把一根线缆中的电或光信号传递给另一根线缆，不进行路由选择，也不进行分组过滤的物理层联网设备。它仅仅作用于信号的电气部分。主要功能是将信号放大并整形再转发出去，以消除信号由于经过一长段电缆而造成的失真，使信号的波形和强度达到所需要的要求，来扩大网络传输的距离，原理是信号再生（而不是将衰减的信号放大）。

#### 2.5.2 集线器(hub)

集线器就是一个多端口中继器。

## 第3章 数据链路层

### 3.1 数据链路层的功能

所谓“链路”就是一条点到点的物理线路，中间没有任何交换结点。

当在一条链路上传输数据时，还必须有一些必要的通信协议来控制数据的传输。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**(data link)。现在最常用的方法是使用网络适配器（如拨号上网使用的拨号适配器，以及通过以太网上网使用的局域网适配器）来实现这些协议的软件和硬件。一般的适配器都包含了数据链路层和物理层这两层的功能。

局域网适配器是插在主机箱内的一块网络接口板，这种接口板又称为网络接口卡NIC(Network Interface Card)，简称为网卡。

数据链路层的功能是**组帧、差错控制、流量控制与可靠传输机制（可选）、介质访问控制**。

组帧是指数据链路层必须负责帧的定界，实现一种能够识别帧的开始和结束的结构。

数据链路层可分为可靠传输和不可靠传输两类。数据链路层的传输错误可分为两大类，一类是比特差错，另一类是帧丢失、帧重复和帧失序。对于比特差错，可以在帧结构中加入检错编码或纠错编码。

对于第二类错误，可以采用可靠传输机制来解决。

OSI的观点是必须把数据链路层做成可靠的。因此在CRC检错的基础上，增加了帧编号、确认和重传机制。收到正确的帧就要向发送端发送确认。发送端在一定的期限内若没有收到对方的确认，就认为出现了差错，进行重传，直到收到对方的确认为止。这种方法在历史上曾经起到很好的效果，但现在的通信线路的质量已经大大提高了，由通信链路质量不好引起差错的概率已经大大降低。因此，因特网广泛使用的数据链路层协议都不使用确认和重传机制，即不要求数据链路层向上提供可靠传输的服务。如果在数据链路层传输数据时出现了差错并且需要进行改正，那么改正差错的任务就由上层协议来完成。实践证明，这样做**可以提高通信效率**。

数据链路层使用的信道有两种类型，点对点信道和广播信道。广播信道存在争用问题，需要有介质访问控制协议。

### 3.2 组帧

#### 3.2.1 字符计数法

如图3.1所示，字符计数法是在帧头部使用一个字段来标明帧内字符数。



图3.1 字符计数成帧法

#### 3.2.2 字节填充法

字节填充法是指在帧中采用特殊的字节来标识帧的开始和结束。例如每一帧以控制字符SOH(Start Of Header)开始，以控制字符EOT(End Of Transmission)结束。SOH和EOT都是控制字符名称，它们的十六进制编码分别是01H和04H。

当传送的是文本文件时，其数据部分显然不会出现SOH和EOT这样的控制字符。当传送的是二进制数据时，SOH和EOT可能出现在数据部分，这时，发送方的数据链路层在每一个SOH或EOT前面插入一个转义字符ESC(其十六进制编码是1BH)，接收方的数据链路层在将数据交给网络层之前删除这个转义字符。如果转义字符也出现在数据段中，则在转义字符前插入一个转义字符，对应地，接收方收到连续的两个转义字符时要删除前面一个。

还有其它字节填充法。例如，PPP协议在使用异步传输时使用自己的字节填充法，思想是一样的。

#### 3.2.3 位填充法

以HDLC(High-level Data Link Control，高级数据链路控制)协议为例。

帧开头和结尾用7E(01111110)标识，发送方在数据段中发现有5个连续1就立即插入一个0，经过这种0比特填充后的数据段，可以保证不会出现6个连续1。接收方在数据段每发现5个连续1就把后面的一个0删除。

PPP协议在使用同步传输时也使用这种位填充法。

### 3.3 差错控制

#### 3.3.1 检错编码

目前数据链路层广泛采用了CRC码进行检错。

#### 3.3.2 纠错编码

海明码，见计组2.7节。

### 3.4 流量控制与可靠传输机制

可靠传输通常使用**确认**和**超时重传**两种机制来完成（它只是一种方法，不是数据链路层专用的，其它层也可以使用）。确认是一种无数据的控制帧，这种控制帧使得接收方可以让发送方知道哪些内容被正确接收。有些情况下为了提高高传输效率，将确认捎带在一个回复帧中，称为**捎带确认**。超时重传是指发送方在发送某一个数据帧以后就开启一个计时器，在一定时间内如果没有得到发送的数据帧的ACK报文，那么就重新发送该数据帧，直到发送成功为止。

由于超时重传是自动进行的，接收方不需要通知发送方重传某个出错的帧，因此使用确认和超时重传两种机制来实现可靠传输的策略称为自动请求重传（Auto Repeat reQuest, ARQ）。有三种常用的ARQ协议，即停止-等待式（Stop-and-Wait）ARQ、后退N帧（Go-Back-N）ARQ和选择重传（Seletctive Repeat）ARQ。后两种协议是滑动窗口技术与请求重传技术的结合，由于窗口尺寸开到足够大时，帧在线路上可以连续地流动，因此又称其为连续ARQ协议。

#### 3.4.1 单帧滑动窗口与停止等待协议

在停止等待协议中，源站发送单个帧后必须等待确认，在目的站的确认帧到达源站之前，源站不能发送新的数据帧。停止等待协议可以用图3.2来说明。

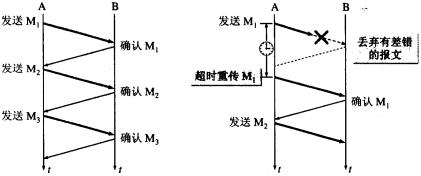


图3.2 停止等待协议

图3.3说明了两种出错情况以及对应的处理。

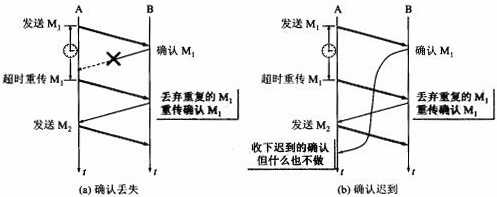


图3.3 确认丢失(a)和确认迟到(b)

对于停止等待ARQ协议，发送窗口WT=1, 接收窗口WR =1。

#### 3.4.2 多帧滑动窗口与后退N帧协议

源站在发送完一帧后，不是停下来等待确认帧，而是可以连续再发送若干帧。如果这时收到了目的站发来的确认帧，那么还可以接着发。

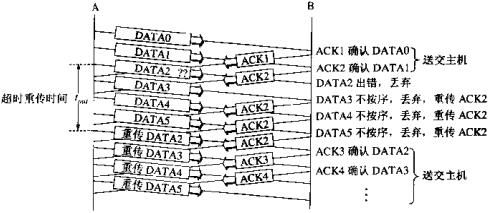


图3.4 连续ARQ协议

如图3.4所示，主机A向主机B发送数据帧。当A发完0号帧后，不是停止等待，而是继续发送后续的1号、2号帧等。A每发送完一帧就要为该帧设置超时计时器。由于连续发送了许多着帧，确认帧必须要指明是对哪一帧进行确认。在图3.4中，ACKn表示对第n-1号帧的确认。这表示第发送方说：“我已正确收到了第n-1号帧，下一次我期望收到第n号帧”。

主机B正确地收到了第0号和1号帧，现在设2号帧出了差错，于是结点B就自动将有差错的2号帧丢弃，如果此时收到了后面的3号、4号等帧，也全部丢弃，等待主机A重传。尽管主机A已经发完了5帧，但必须回退到2号帧开始重传。

为了减少开销，GBN协议一般采用**累积确认**的方式，即接收方不必每收到一个帧就发回一个确认帧，而是可以在连续收到好几个帧后，才对最后一个帧发送确认。

对于GBN ARQ协议，发送窗口WT≤2n-1, 接收窗口WR=1。

#### 3.4.3 多帧滑动窗口与选择重传协议

为进一步提高信道的利用率，可设法只重传出现差错或超时的帧。但这时必须加大接收窗口，以便收下序号不连续但仍处在接收窗口中的那些帧，等到所缺序号的帧收到后再一并递交给上一层。这就是选择重传ARQ协议。

选择重传ARQ协议可以避免重复发送那些本来已经正确到达接收方的帧，但付出的代价是在接收方要设置具有相当容量的缓存空间。

对于选择重传ARQ协议，WT>1，WR>1。

接收窗口显然不应该大于发送窗口，又因为WT+WR≤2n，所以WR≤2n-1。当接收窗口为最大值时，WT=WR=2n-1。

### 3.5 介质访问控制

当使用广播信道时，需要考虑的一个问题是如何使多用户共享通信介质资源。有三种方法。

#### 3.5.1 信道划分介质访问控制

信道服用是一种静态划分信道的技术，按照某种物理特性划分公共信道，使得每个用户与其他用户的传输信道在某个物理特性方面是能区分开的。信道复用技术有：频分多路复用（FDM）、时分多路复用（TDM）、波分多路复用（WDM）、码分多路复用（CDM）等，主要用于无线通信或光纤通信中，特别是在移动通信中广泛使用。

**1. 频分多路复用（FDM）**

频分多路复用（Frequency-Division Multiplexing, FDM），是一种将多路基带信号调制到不同频率载波上再进行叠加形成一个复合信号的多路复用技术。

**2. 时分多路复用（TDM）**

时分多路复用（又称同步时分多路复用），即将一条物理信道按时间分成若干个时间片轮流地分配给多个信号使用。

在时分多路复用中，由于计算机数据的突发性质，一个用户对已经分配到的子信道的利用率一般不高。统计时分多路复用（STDM，又称异步时分多路复用）是TDM的一种改进，它采用STDM帧，STDM帧不是固定分配时隙，而是按需动态地分配时隙，因此可以提高线路的利用率。

**3. 波分多路复用（WDM）**

即光的频分多路复用。

**4. 码分多路复用（CDM）**

码分多路复用又称码分多址（Code-Division Multiple Acess, CDMA），CDM与FDM和TDM不同，它既共享信道的频率，也共享时间，是一种真的动态复用技术。其原理是每比特时间被分成m个更短的时间槽，称为芯片（Chip），通常情况下每比特有64或128个芯片。每个站点（子信道）被指定一个唯一的mbit的代码或芯片序列。当发送1时站点就发送芯片序列，发送0时就发送芯片序列的反码。当两个或多个站点同时发送时，各路数据在信道中被线性相加。为了从信道中分离出各路信号，要求各个站点的芯片序列是相互正交的。

码分多路复用技术主要用于无线通信系统，特别是无线通信系统。

#### 3.5.2 随机访问介质访问控制

**1、ALOHA协议**

最早采用这种争用方式的网络是美国夏威夷大学在20世纪70年代建立起来的ALOHA网络。它对主机的发信时间不加任何限制，根据需要任何时刻都可以发信。源站在发信后需要侦听一段时间，侦听时间等于电波传到最远的站再返回本站所需的时间。如果在这段时间里收到目的站发来的应答信号，说明发信成功，否则，就重新发送这个帧。如果反复几次都失败就停止发送。目的站对收到的帧进行校验，如果正确无误，就立即发出应答（应答帧采用另一频率传输）。收到的数据可能不正确，比如有噪音干扰，或别的站同时也在同时发信，破坏了这个帧（即发生了冲突）。在这种情况下，目的站对该帧不予接受，也不发送应答。源站在规定时间内收不到应答就会自动重传。这种纯ALOHA方式十分简单，但是性能并不理想。随着通信负载的增加，冲突机会也急剧增加，信道的最高吞吐率大约只有18%。这里的信道吞吐率是发送成功的帧数与实际发送的帧数之比。

一种改进的方案成为分槽ALOHA。在这种网络中，信道的使用划分为等长的时间片（slot, 或称时槽）。时间片的长度等于一个站所发的帧到达目的站的最大时延。网络采用集中同步方式，所有的站都在同一时钟下工作。各站只能在时间片的起始时刻才能发信。这样改进后，如果两个站都要发信，它们将在同一瞬间开始，即它们的帧只会整个的冲突。这就避免了两个帧部分地冲突的情况，因而减少了帧冲突的概率。这种分槽ALOHA可将信道的吞吐率提高到37%。

**2、CSMA协议**

在纯ALOHA协议及其改进协议中最大的问题是发送时盲目性，即发送时不管信道当时是否空闲，这就造成了很大一部分冲突，为了解决这一问题，引入了CSMA协议。

CSMA（Carrier Sense Multiple Access, 载波监听多路访问）使用了一种检测介质是否正在被使用的机制。根据在传输介质上是否存在电压或其他形式的能量来确定通信信道是否空闲。如果一个要发信的站“听到”介质上有帧正在传输，则必须等待。

采用CSMA，需要采用一种算法来决定当发现介质忙时如何处理。常用的有3种算法，1-持续CSMA，非持续CSMA，P-持续CSMA。

第一种是1-持续CSMA。站点首先侦听信道，若信道忙，它就持续等待，直到它侦听到信道空闲时就立刻（概率1）发信。若发生冲突，站点就等待一个随机长的时间，然后重新开始侦听信道。此协议就叫1-持续CSMA，因为站点一旦发现信道空闲，其发送数据的概率是1。

1-持续和P-持续CSMA的区别是，当侦听到信道空闲时，前者会立刻发信，后者以概率P发信，以概率1-P将该此发送推迟到下一时隙。三者的共同点是，当发生冲突后，会等待一个随机长的时间再开始侦听。

1-持续和非持续CSMA的区别是，如果信道正忙，前者会一直侦听，后者会等待一个随机长的时间后再侦听。

**3、CSMA/CD协议**

使用冲突检测机制的CSMA称作CSMA/CD，CD指的是冲突检测，详细介绍见3.5.2节。

**4、CSMA/CA协议**

CA(Collision Avoidance)是指碰撞避免，详细介绍见3.5.3节。

#### 3.5.3 轮询访问介质访问控制

令牌传递协议

目前令牌环网早已退出了市场。

### 3.6 局域网

#### 3.6.1 局域网的基本概念与体系结构

局域网(LAN)有3个主要特征：

(1) 地理范围和结点数目有限；

(2) 有较高的数据率、较低的时延和较低的误码率；

(3) 为一个单位所拥有。

1982年DEC、Intel和施乐(Xerox)公司联合颁布了DIX Ethernet V2标准。1983年IEEE 802委员会的802.3工作组制定了IEEE 802.3标准。

然而到了20世纪90年代，激烈竞争的局域网市场逐渐明朗。以太网在局域网市场中已取得了垄断地位，并且几乎成为了局域网的代名词。由于因特网发展很快而TCP/IP体系经常使用的局域网只剩下DIX Ethernet V2而不是IEEE 802.3标准中的局域网，因此现在IEEE 802委员会制定的逻辑链路层LLC（即802.3标准）的作用已经消失了，很多厂商生产的适配器上就仅装有MAC协议而没有LLC协议。

从本节开始介绍的都是以太网技术。

#### 3.6.2 以太网

以太网是使用1-持续CSMA/CD技术的总线型网络。

以太网的逻辑结构是总线型结构，物理结构是星型或拓展星型结构。

以太网为了通信简便，采取了(1)无连接的工作方式；(2)曼彻斯特编码。

##### 3.6.2.1 CSMA/CD协议

在大多数CSMA/CD系统中，都有一条重要的规则，就是任何帧都应该有足够的长度以使冲突能在帧传输完毕前被检测到。

CSMA/CD是载波监听多点接入/冲突检测(Carrier Sense Multiple Access with Collision Detection)的缩写。下面是CSMA/CD协议的要点。

“**多点接入**”就是说明这是总线型网络，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**”就是“发送前先监听”，即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站在发送数据，如果有，暂时不要发送数据，要等待信道变为空闲时再发送。

“**碰撞检测**”就是“边发送边监听”，即适配器边发送数据边检测信道上的信号电压的变化情况，一边判断自己在发送数据时其他站是否也在发送数据。当几个站同时在总线上发送数据时，总线上的信号电压变化幅度将会增大（互相叠加）。当适配器检测到的信号电压变化幅度超过一定的阀值时，就认为总线上至少有两个站同时在发送数据，表明产生了冲突。

可简单概括为“**先听后发、边听边发、冲突停发、随机重发**”。

以太网使用**截断二进制指数退避** (truncated binary exponential backoff)算法来解决碰撞问题。这种算法让发生碰撞的站在停止发送数据后，不是等待信道变为空闲后立即再发送数据，而是推迟（这叫做）一个随机的时间。这样是为了使重传时再次发生冲突的概率减小。具体的退避算法如下：

(1) 确定基本退避时间。它就是争用期2τ。**以太网把争用期定为512比特时间**，即即规定最短帧长为64B。凡长度小于64B的都是由于冲突而异常终止的无效帧，如果前64B没有发生冲突，那么后续的数据也不会发生冲突（表示已经成功抢占信道）。

(2) 从离散的整数集合[0, 1, ..., (2k-1)]中随机取出一个数，记为r，重传应推后的时间是r倍的争用期。上面的参数k按下面的公式计算：

k=Min{重传次数，10}

可见当重传次数不超过10时，参数k等于重传次数；但当重传次数超过10时，k就不再增大而一直等于10。

(3) 当重传达16次仍不能成功时(这表明同时打算发送数据的站太多，一直连续发生冲突)，则丢弃该帧，并向高层报告。

以太网还采取一种叫做**强化碰撞**的措施。这就是当发送数据的站一旦发现发生了碰撞，除了立即停止发送数据外，还要再继续发送32比特或48比特的人为干扰信号(jamming signal)，以便让所有用户都知道现在已经发生了碰撞。

以太网还规定了**帧间最小间隔**为96比特时间。这样做是为了使刚刚收到帧的站的接收缓存来得及清理，做好接收下一帧的准备。

根据以上所讨论的，CSMA/CD协议的**要点归纳如下**：

(1) 适配器从网络层获得一个分组，加上以太网的首部和尾部，组成以太网帧，放入适配器的缓冲中，准备发送。

(2) 若检测到信道空闲（即在96比特时间内没有检测到信道上有信号），就发送这个帧。若检测到信道忙，则继续检测并等待信道转为空闲，然后发送这个帧。

(3) 在发送过程中持续监测信道，若一直未检测到碰撞，就顺利把这个帧成功发送完毕。若检测到碰撞，则终止数据的发送，并发送一个人为干扰信号。

(4) 在终止发送后，适配器就执行指数退避算法，等待r倍512比特时间后，返回到步骤(2)。

##### 3.6.2.2 MAC层

**1、MAC地址**

**名字**指出我们所要寻找的那个资源，**地址**指出那个资源在何处，**路由**告诉我们如何到达该处。

严格的讲，MAC地址是一种名字，不过人们还是习惯于把这种48位的“名字”称为地址。

MAC地址一般固化在适配器的ROM中。

IEEE 802标准规定MAC地址字段可采用6字节或2字节这两种中的一种。6字节地址对局部范围内使用的局域网的确是太长了，但是由于6字节的地址字段可使全世界所有的局域网适配器都具有不相同的地址，因此现在的局域网适配器实际是上用的都是6字节的MAC地址。

现在IEEE的注册管理机构RA(Registration Authority)是局域网全球地址的法定管理机构，它负责分配地址字段的6个字节中的前三个字节（即高位24位）。世界上凡要生产局域网适配器的厂家都必须向IEEE购买由这三个字节构成的这个号（即地址块），这个号的正式名称是组织唯一标志符OUI(Organizationally Unique Identifier)，通常也叫做公司标志符(company\_id)。

**2、MAC帧的格式**

常用的以太网MAC帧格式有两种标准，一种是DIX Ethernet V2标准（即以太网V2标准），另一种是IEEE的802.3 标准。这里只介绍时用的最多的以太网V2的MAC帧格式，如图3.5所示。图中假定网络层使用的是IP协议。实际上使用其他的协议也是可以的。

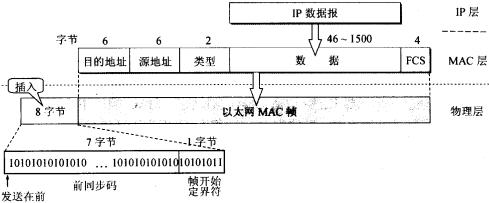


图3.5 以太网V2的MAC帧格式

##### 3.6.2.3 快速以太网

速率达到或超过100Mbps的以太网称为高速以太网。下面简单介绍3种高速以太网技术。

**1、百兆以太网**

100BASE-T是在双绞线上传送100Mbps基带信号的星形拓扑以太网，仍使用IEEE 802.3的CSMA/CD协议，它又称为**快速以太网**(Fast Ethernet)。

1995年IEEE把100BASE-T的快速以太网定为正式标准，其代号为IEEE 802.3u。

快速以太网的基本思想很简单，保持帧格式不变，只是把比特时间从100ns减少到10ns，即以10倍速度使用传统的以太网MAC。

100BASE-T MAC和10Mbps经典以太网MAC几乎完全一样，二者都具有下列参数值：时隙=512比特时间，重试次数极限=16，退避次数极限=10，冲突加强（Jam）信号长度=32比特，最大帧长=1518字节，最小帧长=64字节。唯一不同的参数就是帧际间隙时间，10Mbps是9.6μs，快速以太网（100Mbps）是0.96μs。

快速以太网使用4B/5B编码。

**2、千兆以太网**

千兆以太网有两个标准，主要区别在于传输介质不同。IEEE 802.3z工作组负责制定管线和同轴电缆的全双工链路标准，产生了IEEE 802.3z标准(1000BASE-X)。IEEE 802.3ab工作组负责制定UTP电缆的半双工链路标准，产生了IEEE 802.3ab标准(1000BASE-T)。

千兆以太网有以下几个特点：

(1) 允许在1Gb/s下全双工和半双工两种方式工作。

(2) 使用IEEE 802.3协议规定的帧格式。

(3) 在半双工下使用CSMA/CD协议（全双工方式下不需要）。

(4) 与10BASE-T和100BASE-T技术向后兼容。

千兆以太网工作在半双工方式时，使用载波延伸(carrier extension)和分组突发(packet bursing)。

千兆以太网使用8B/10B编码。

**3、万兆以太网**

2002年IEEE颁布了万兆以太网标准IEEE 802.3ae。

万兆以太网(10GE)只适用光纤作为传输媒体。

10GE只工作在全双工方式，因此不存在争用问题，也不使用CSMA/CD协议。

#### 3.6.3 无线局域网

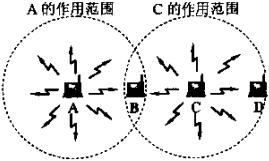
1997年IEEE制定出无线局域网的协议标准802.11，ISO/IEC也批准了这一标准。

无线局域网不能简单地搬用CSMA/CD协议，主要有两个原因：

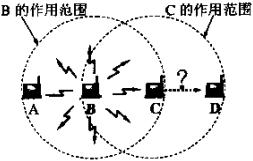
(1) 无线局域网上不能够检测碰撞，因为发送设备的功率要比接收设备的功率强得多，在这种情况下，碰撞检测是不可行的。

(2) 更重要的是，即使能够实现碰撞检测，当源站在发送数据时检测到新倒是空闲的，在目的站仍然有可能发生碰撞。

图3.6展示了无线局域网的特殊问题。图中画有4个无线移动站，并假定无线电信号传播的范围是以发送站为圆心的一个圆形面积。图3.6(a)中的问题叫做**隐蔽站问题**(hidden station problem)，图3.6 (b)中的问题叫做**暴露站问题**(exposed station problem)。



(a)A和C同时向B发送信号，发生碰撞；



(b)B向A发送信号，影响C向D发送数据

图3.6 无线局域网的问题

因此一个能够避免冲突的协议更有意义。这就是CSMA/CA协议，802.11就是使用了CSMA/CA协议。

为了尽量避免碰撞，802.11使用了两种方法：

(1) 采用一种P-持续机制，加上空闲时间管理。当一个站检测到信道空闲时，必须等待一个指定的**帧间间隔IFS**(InterFrame Space)才能发送数据。高优先级帧的帧间间隔较短。

(2) 对信道进行预约。源站在发送数据帧之前先发送一个短的控制帧，叫做**请求发送RTS**(Request To Send)，它包括源地址、目的地址和这次通信所需的时间。

#### 3.6.4 令牌环网

令牌沿着环形总线依次传递，拿到令牌的结点才能发送数据由于令牌在环上是按顺序依次传递的，因此对所有结点而言，访问权是公平的。

目前令牌环网早已退出了市场。

### 3.7 广域网

#### 3.7.1 广域网的基本概念

广域网由结点交换机和链路组成。结点之间是点到点的连接。

广域网和互联网的区别：广域网是单个网络，使用结点交换机连接各主机；互联网由多个网连接而成，使用路由器连接各网络。

结点交换关机工作在第二层，路由器工作在第三层。

结点交换机与以太网交换机都是数据链路层设备，区别在于，前者使用点对点信道，后者使用广播信道。

#### 3.7.2 PPP协议

PPP(Point to Point Protocol)是使用串行线路通信的面向字节的协议，主要用于用户连接到ISP。

PPP协议有三个组成部分：组帧方法，链路控制协议LCP(Link Control Protocol)和网络控制协议NCP(Network Control Protocol)。

PPP协议不需要的功能：

(1) 纠错(error correction)。PPP协议只进行检错。

(2) 流量控制。

(3) 序号。PPP协议是不可靠传输协议，因此不需要给帧编号。

(4) 多点线路。PPP协议只支持点对点的链路通信。

(5) 半双工或单工链路。PPP协议只支持全双工链路。

#### 3.7.3 HDLC协议

高级链路控制HDLC(High-level Data Link Control)是面向比特的能实现可靠传输的数据链路层协议。现在很少使用了。对于点对点链路，简单得多的PPP协议目前使用的最广泛。

全双工通信；所有帧采用CRC校验；采用0比特填充法。

### 3.8 数据链路层设备

#### 3.8.1 网桥(bridge)

网桥和中继器（或集线器）的区别是：网桥是按存储转发的方式工作的，一定是先把整个帧收下来再进行处理；中继器（或集线器）是逐比特转发。

网桥转发帧时有三种操作：转发，过滤和洪泛帧。网桥在转发帧时不改变帧的源地址和目的地址。

网桥分为透明网桥和源路由网桥。**透明网桥**是一种即插即用设备(plug-and-play device)。当网桥刚刚连接到以太网时，其转发表是空的，网桥会按照以下自学习算法逐步建立起转发表：若从某个站A发出的帧从接口x进入了某网桥，那么从这个接口出发沿相反方向一定可以把一个帧传送到A。所以网桥每收到一个帧，就记下其源地址和进入网桥的接口，作为转发表中的一个项目。

**源路由网桥**是一种由源站负责路由选择的网桥。源路由网桥是在发送帧时，把详细的路由信息放在帧的首部中。

为了发现合适的路由，源站以广播方式向欲通信的目的站发送一个发现帧(discovery fram)作为探测之用。发现帧将在整个扩展的以太网中沿着所有可能的路由传送。在传送过程中，每个发现帧都记录所经过的路由。当这些发现帧达到目的站时，就沿着各自的路由返回源站。源站在得知这些路由后，从所有可能的路由中选择出一个最佳路由。以后，凡从这个源站向该目的站发送的帧的首部，都必须携带源站所确定的这一路由信息。

源路由网桥对主机是不透明的，主机必须知道网桥的标识以及连接到哪一个网段上。

#### 3.8.2 以太网交换机(switch)

以太网交换机实质上就是一个多端口的网桥。

利用以太网交换机可以很方便地实现**虚拟局域网VLAN**(Virtual LAN)。

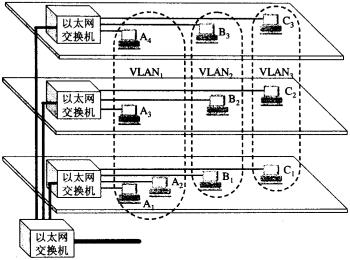


图3.7 3个虚拟局域网VLAN1、VLAN2和VLAN3

1988年IEEE指定了802.3ac标准，这个标准定义了以太网的帧格式的扩展，以便支持虚拟局域网。虚拟局域网协议允许在以太网的帧格式中插入一个4字节的标识符（见图3.8），称为VLAN标记(tag)，用来指明发送该帧的工作站属于哪一个虚拟局域网。

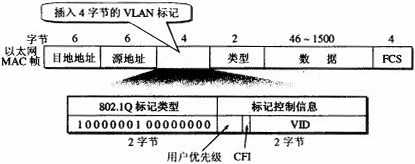


图3.8 在以太网的帧格式中插入VLAN标记

由于插入了一个VLAN标记，以太网帧的首部增加了4个字节，因此以太网帧的最大长度从原来的1518字节变为1522字节。

## 第4章 网络层

### 4.1 网络层的功能

异构网络互联，路由与转发，拥塞控制

### 4.2 IP协议

网际协议IP是TCP/IP体系中两个最主要的协议之一，也是最重要的因特网标准协议之一。与IP协议配到使用的还有四个协议：

地址解析协议ARP(Address Resolution Protocol)

逆地址解析协议RARP(Reverse Address Resolution Protocol)

网际控制报文协议ICMP(Internet Control Message Protocol)

网际组管理协议IGMP (Internet Group Management Protocol)

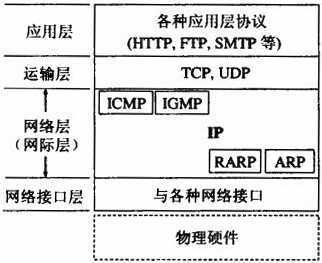


图4.1 网际协议IP及其配套协议

图4.1画出了这四个协议和网际协议IP的关系。在这一层中，ARP和RARP画在最下面，因为IP经常要使用这两个协议。ICMP和IGMP画在这一层的上部，因为它们要使用IP协议。这四个协议将在后面陆续介绍。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，因此TCP/IP体系中的网络层常常称为**网际层**(internet layer)，或**IP层**。

#### 4.2.1 分类的IP地址

IP地址现在由因特网名字与号码指派公司ICANN(Internet Corporation for Assigned Names and Numbers)进行分配。

IP地址的编址方法共经过了三个历史阶段。这三个阶段是：

(1) 分类的IP地址。这是最基本的编址方法，在1981年就通过了相应的标准协议。

(2) 子网的划分。这是对最基本的编址方法的改进，其标准RFC950在1985年通过。

(3) 构成超网。这是比较新的无分类编址方法。1993年提出来后很快就得到推广应用。

本节只讨论最基本的分类IP地址。

分类的IP地址中，IP地址::={<网络号>, <主机号>}。如图4.2所示。

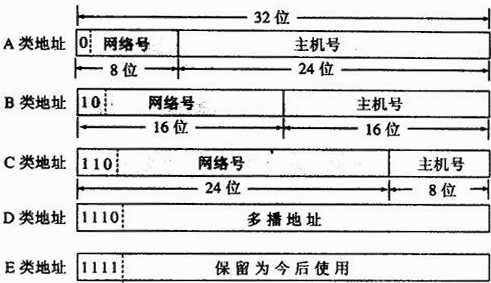


图4.2 分类的IP地址

IP地址都是32位的二进制代码。为了提高可读性，我们常常把32位的IP地址中的每8位用其等效的十进制数字表示，并且在这些数字之间加上一个点。这就叫做**点分十进制记法**（dotted decimal notation）。

表4.1给出了IP地址的指派范围。

表4.1 IP地址的指派范围

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 网络类别 | 最大可指派的网络数 | 第一个可指派的网络号 | 最后一个可指派的网络号 | 每个网络中的最大主机数 |
| A | 126(27-2) | 1 | 126 | 16777214(224-2) |
| B | 16383(214-1) | 128.1 | 191.255 | 65534(216-2) |
| C | 2097151(231-1) | 192.0.1 | 223.255.255 | 254(28-2) |

表4.2给出了一般不使用的IP地址，这些地址只在特定的情况下使用。

表4.2 一般不使用的特殊IP地址

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 网络号 | 主机号 | 源地址使用 | 目的地址使用 | 代表的意思 |
| 0 | 0 | 可以 | 不可 | 在本网络上的本主机（见DHCP协议） |
| 0 | host-id | 可以 | 不可 | 在本网络上的某个主机host-id |
| 全1 | 全1 | 不可 | 可以 | 只在本网络上进行广播（各路由器均不转发） |
| net-id | 全1 | 不可 | 可以 | 对net-id上的所有主机进行广播 |
| 127 | 非全0或全1的任何数 | 可以 | 可以 | 用作本地软件环回测试之用 |

当两个路由器直接相连时，在连线两端的接口处，可以分配也可以不分配IP地址。但为了节约IP资源，现在常常不分配IP地址。

#### 4.2.2 ARP协议和RARP协议

知道IP地址找出对应的物理地址，这是ARP协议，知道物理地址找出对应的IP地址，这是RARP协议。

RARP协议目前已被DHCP协议取代，现在很少使用了，因此这里不再进一步介绍RARP。DHCP协议见6.6节。

ARP协议的要点：ARP高速缓存，生存时间，广播，响应。

下面我们归纳出使用ARP的四种典型情况。

(1) 发送方是主机，要把IP数据包发送到本网络上的另一个主机。这时用ARP找到目的主机的硬件地址。

(2) 发送方是主机，要把IP数据包发送到另一个网络上的一个主机。这时用ARP找到本网络上的一个路由器的硬件地址。剩下的工作由这个路由器来完成。

(3) 发送方是路由器，要把IP数据包转发送到本网络上的另一个主机。这时用ARP找到目的主机的硬件地址。

(4) 发送方是路由器，要把IP数据包转发送到另一个网络上的一个主机。这时用ARP找到本网络上的一个路由器的硬件地址。剩下的工作由这个路由器来完成。

在许多情况下需要多次使用ARP，但都只是以上几种情况的反复使用而已。

#### 4.2.3 IP数据报的格式

在TCP/IP标准中，各种数据格式常常以32位（4字节）位为单位来描述。图4.3是IP数据包的完整格式。

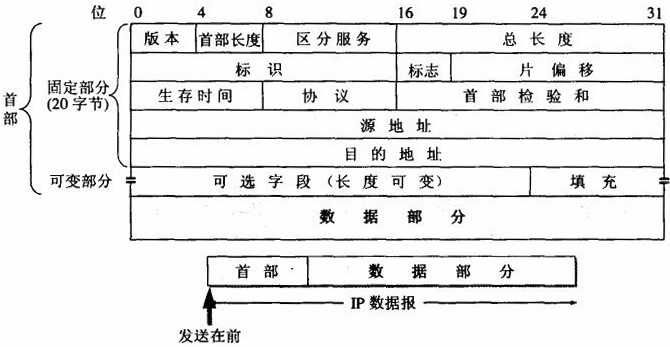


图4.3 IP数据包的格式

部分重要字段含义如下：

（1）首部长度——占4位，可表示的最大十进制数是15，单位是32比特（4字节），故首部长度最大值是60B。最常用的首部长度是20B，这时不使用任何选项。

（2）总长度——占16位，指首部和数据之和的长度，单位为字节。

（3）标识——占16位，当数据报长度超过MTU时，必须进行分片，这时标识字段的值就被复制到所有的数据报片的标识字段中。相同标识的数据报片最后能正确的重装为原来的数据报。

（4）标志——占3位，目前只有前两位有意义。最低位记为MF(More Fragment)，MF=1表示后面还有分片，MF=0表示这是最后一个分片。中间一位记为DF(Don’t Fragment)，只有当DF=0时才允许分片。

（5）片偏移——占13位，指某片在原分组中的相对位置，片偏移以8B为单位。这就是说，每个分片的长度一定是8字节的整数倍。

（6）生存时间——占8位，英文缩写是TTL(Time To Live)，表明数据报在网络中的寿命。目前TTL的含义已改为“跳数限制”。

（7）协议——占8位，指出本数据报携带的数据是使用何种协议。

（8）首部校验和——占16位，计算结果为0表示无差错。

（9）源地址——占32位。

（10）目的地址——占32位。

#### 4.2.4 IP层转发分组的流程

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）

路由器和普通主机中都有一张路由表，路由表中都有一条**默认路由**(default route)。

虽然分组转发都是基于目的主机所在的网络地址，但允许有这样的特例，即对特定的目的主机指定一个路由，这种路由叫做**特定主机路由**。它可使网络管理员能更方便地控制和测试网络。

分组转发的流程如下：

(1) 从数据报的首部提取目的主机的IP地址D，得出目的网络地址为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**，不需要在经过其他的路由器，直接把数据报交付给目的主机（这里包括把目的主机的IP地址D转换为具体的硬件地址和，把数据报封装为MAC帧，再发送此帧）；否则就是简介交付，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表中所指明的下一跳路由；否则，执行(4)。

(4) 若路由表中有到达网络N的路由，则把数据报传送给路由表中所指明的下一条路由器；否则，执行(5)。

(5) 若路由表中有一个默认路由，则把数据报传送给默认路由器；否则，执行(6)。

(6) 报告转发分组出错。

### 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**

分类的IP编址方法粒度太大，不够灵活。从1985年起在IP地址中又增加了一个“子网号”字段，使两级IP地址变成三级IP地址，这种做法叫做**划分子网(**subnetting)[RFC950]。划分子网已成为因特网正式标准。

划分子网后，IP地址::={<网络号>, <子网号>, <主机号>}。

**2、子网掩码**

由于IP地址本身以及IP数据报的首部都没有包含任何有关子网划分的信息，因此必须另外想办法，这就是使用**子网掩码**(subnet mask)。

现在的因特网标准规定：所有的网络都必须使用子网掩码，同时在路由表中也必须有子网掩码这一栏。

如果一个网络部划分子网，那么该网络的子网掩码就使用默认子网掩码。显然

A类地址的默认子网掩码是255.0.0.0，或0xFF000000，

B类地址的默认子网掩码是255. 255.0.0，或0xFFFF0000，

C类地址的默认子网掩码是255. 255. 255.0，或0xFFFFFF00。

虽然RFC文档并没有规定子网中的一串1必须是连续的，但却极力推荐在子网掩码中使用连续的1。

**RFC950文档规定，子网号不能为全0或全1。**

**CIDR中的子网号可以为全0或全1。**

**无论是两级IP地址、三级IP地址还是CIDR，主机号都不能为全0或全1。**

**3、划分子网后分组转发的流程**

划分子网后，路由表中的每一条路必须包含以下三项信息：

（目的网络地址，子网掩码，下一跳地址）

在划分子网的情况下，路由器转发分组的流程如下：

(1) 从收到的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。对路由器直接相连的网络逐个进行检查：用各网络的子网掩码和D逐位相“与”（AND操作），看结果是否和这个网络地址匹配。若匹配，则把分组进行直接交付，转发任务结束。否则就是间接交付，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表中所指明的下一条路由器；否则，执行(4)。

(4) 对路由表中的每一行（目的网络地址，子网掩码，下一跳地址），用其中的子网掩码和D逐位相“与”，其结果为N。若N与该行的目的网络地址匹配，则把数据报传送给该行指明的下一条路由；否则，执行(5)。

(5) 若路由表中有一个默认路由，则把数据报传送给默认路由器；否则，执行(6)。

(6) 报告转发分组出错。

#### 4.3.2 构造超网

无分类编址的正式名字是无分类域间路由选择CIDR(Classless Inter-Domain Routing，CIDR的读音是“sider”)。现在CIDR已成为了因特网建议标准[RFC1517~1520]。

CIDR最主要的特点有两个：

(1) CIDR消除了传统的A类、B类、C类地址以及划分子网的概念。CIDR是无分类的两级编址，它的记法是

IP地址::={<网络前缀>, <主机号>}

CIDR使用“斜线记法”，即在IP地址后面加上斜线“/”，然后写上网络前缀所占的位数。

(2) CIDR把网络前缀相同的连续的IP地址组成一个“CIDR地址块”。

一个CIDR地址块可以包含多个A类、B类或C类网络，这就是**构造超网**(supernetting)这一名词的来源。

网络前缀小于13位或大于27位的都很少使用。

为了进行路由选择，CIDR使用32位的**地址掩码**(address mask)。地址掩码由一串1和一串0组成，而1的个数就是网络前缀的长度。

一个CIDR地址块可以继续划分子网，这些子网也都只有一个网络前缀和一个主机号字段，但子网的网络前缀比整个地址块的网络前缀要长些。

路由器中的每一条路由主要包含两项信息：

（目的网络前缀，下一跳地址）

在查找路由表时可能会得到不止一个匹配结果，这时路由器采用最长前缀匹配(longest-prefix matching)。

路由器常使用二叉树将路由表中的每一条网络前缀组织成层次结构，然后自上而下按层次查找，这样可以加快速度。

### 4.4 ICMP协议

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报文控制协议ICMP(Internet Control Message Protocol)[RFC792]。ICMP协议是一种差错报告机制，它允许路由器或目的主机在遇到差错时，向源主机发送ICMP报文报告错误。ICMP是因特网标准协议。但ICMP不是高层协议，而是IP层的协议。ICMP报文作为IP层数据报的数据，加上数据报的首部，组成IP数据报发送出去。ICMP报文格式如图4.4所示。

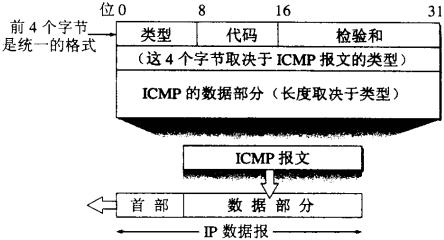


图4.4 ICMP报文的格式

#### 4.4.1 ICMP报文的种类

ICMP报文的种类有两种，即ICMP差错报告报文和ICMP询问报文。

ICMP报文的前4个字节是统一的格式：即类型、代码和校验和。接着的4个字节的内容与ICMP的类型有关。最后面是数据字段，其长度取决于ICMP的类型。表4.3给出了几种常用的ICMP报文类型。

表4.3 几种常用的ICMP报文类型

|  |  |  |
| --- | --- | --- |
| ICMP报文种类 | 类型的值 | ICMP报文的类型 |
| 差错报告报文 | 3 | 终点不可达 |
| 4 | 源点抑制(Source quench) |
| 11 | 时间超过 |
| 12 | 参数问题 |
| 5 | 改变路由(Redirect) |
| 询问报文 | 8或0 | 回送(Echo)请求或回答 |
| 13或14 | 时间戳(Timestamp)请求或回答 |

所有的ICMP差错报告报文中的数据字段都具有同样的格式，如图4.5所示。把收到的需要进行差错报告的IP数据报的首部和数据字段的前8个字节提取出来，作为ICMP报文的数据字段，再加上相应的ICMP差错报告报文的前8个字节，就构成了ICMP差错报告报文。提取收到的数据报的数据字段的前8个字节是为了得到运输层的端口号（对于TCP和UDP）以及运输层报文的发送序号（对于TCP）。这些信息对于源点通知高层协议是有用的。整个ICMP报文作为IP数据报的数据字段发送给源点。

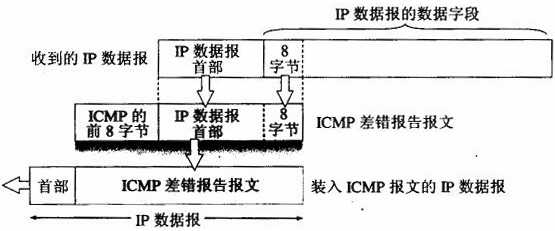


图4.5 ICMP差错报告报文的数据字段的内容

下面是不应该发送ICMP差错报告报文的几种情况。

(1) 对ICMP差错报告报文不再发送ICMP差错报告报文。

(2) 对第一个分片的数据报片的所有后续数据报片都不发送ICMP差错报告报文。

(3) 对具有多播地址的数据报都不发送ICMP差错报告报文。

(4) 对具有特殊地址（例如127.0.0.0或0.0.0.0）的数据报不发送ICMP差错报告报文。

#### 4.4.2 ICMP应用举例

1、ping

分组网间探测PING(Packet InterNet Grouper)，用来测试两个主机之间的连通性。PING使用了ICMP回送请求与回送回答报文。

Windows用户可以在cmd中输入“ping hostname”（hostname可以是域名或IP地址），回车就可以看到结果。PC机一连发出四个回送请求报文，如果目标主机正常工作且响应这个ICMP回送请求报文（有的主机为了防止恶意攻击就不理睬这种报文），那么它就发回ICMP回送回答报文。由于往返的ICMP报文上都有时间戳，因此很容易得出往返时间

2、traceroute

另一个非常有用的应用是traceroute（这是UNIX中的名字，在Windows中这个命令是tracert），它用来跟踪一个分组从源点到终点的路径。下面简单介绍traceroute的工作原理。

Traceroute从源主机向目的主机发送一连串的IP数据报，数据报中封装的是无法交付的UDP数据报（使用了非法端口）。第一个数据报P1的生存时间TTL设置为1。当P1到达路径上的第一个路由器R1时，路由器R1先收下它，接着把TTL的值减1。由于TTL等于0了，R1就把P1丢弃了，并向源主机发送一个**ICMP时间超过**差错报告报文。

源主机接着发送第二个数据报P2，并把TTL设置为2。P2先到达路由器R1，R1收下后把TTL减1再转发给路由器R2。R2收到P2时TTL为1，再减1后TTL就变为0了。R2就丢弃P2，并向源主机发送一个**ICMP时间超过**差错报告报文。这样一直继续下去。当最后一个数据报刚刚达到目的主机时，数据报的TTL是1，主机不转发数据，也不把TTL值减1。但因IP数据报中封装的是无法交付的UDP数据报，因此目的主机要向源主机发送**ICMP终点不可达**差错报告报文。

这样，源主机达到了自己的目的，因为这些路由器和最后目的主机发来的ICMP报文正好给出了源主机想知道的路由信息——到达目的主机所经过的路由器的IP地址，以及到达其中每一个路由器的往返时间。

**Ping工作在应用层，traceroute/tracert 工作在网络层。**

### 4.5 路由协议

#### 4.5.1 几个基本概念

1、路由算法

路由协议的核心就是路由算法，即如何建立路由表。

2、分层次的路由协议

由于以下两个原因：

(1) 因特网的规模非常大，如果让所有路由器都知道所有的网络应该怎样到达，那么路由表将非常大，不现实。

(2) 许多单位不愿意外界了解自己网络的布局细节。

因此，因特网采用分层次的路由协议，将整个因特网划分为许多较小的自治系统(autonomous system)。

这样，因特网的路由协议可分为两大类：

(1) 内部网关协议IGP(Interior Gateway Protocol)，即自治系统内部使用的路由协议。目前使用的最多的是RIP和OSPF协议。

(2) 外部网关协议EGP(External Gateway Protocol)，即自治系统之间使用的路由协议。目前使用的最多的是BGP协议。

自治系统之间的路由也叫域间路由(interdomain routing)，自治系统内部的路由也叫域内路由(intradomain routing)。

#### 4.5.2 RIP协议

**1、工作原理**

**路由信息协议RIP**(Routing Information Protocol)是内部网关协议IGP中最先得到广泛使用的协议[RFC 1058]。RIP是一种分布式的基于距离向量的路由选择协议，是因特网的标准协议，起最大的优点就是简单。

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个目的网络的距离记录。RIP协议将“距离”定义如下：从一路由器到直接连接的网络的距离是1，从一路由器到非直接连接的网络的距离为所经过的路由器数加1。

RIP协议的“距离”也称为“**跳数**”(hop count)，因为每经过一个路由器，跳数就加1。RIP认为好的路由就是它通过的路由器的数目少，即“距离短”。RIP允许一条路径最多只能包含15个路由器（即最多允许15跳）。因此“距离”的最大值为16，表示网络不可达。可见**RIP只是用于小型网络**。

本节讨论的RIP协议和下一节要讨论的OSPF协议，都是分布式路由选择协议。它们的共同特点就是每一个路由器都要不断地和其他一些路由器交换路由信息。我们一定要弄清以下三个要点，即**和哪些路由器交换信息？交换什么信息？在什么时候交换信息？**

RIP协议的特点是：

1. 仅和相邻路由器交换信息。
2. 路由器交换的信息是当前本路由器所知道的全部信息，即自己的路由表。
3. 按固定时间间隔交换路由信息。

**2、距离向量算法**

对每一个相邻路由器发送过来的RIP报文，进行以下步骤：

(1) 对地址为X的相邻路由器发来的RIP报文，先修改此报文中的所有项目：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。每一个项目都有三个关键数据，即：到目的网络N，距离是d，下一跳路由器是X。

(2) 对修改后的RIP报文中的每一个项目，进行以下步骤：

若原来的路由表中没有目的网络N，则把该项目添加到路由表中。否则，

若下一跳路由器地址是X，则把收到的项目替换原路由表中的项目。否则，

若收到的项目中距离d小于路由表中的距离，则进行更新

否则什么也不做

(3) 若3分钟内还没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的路由器，即把距离设置为16。

例1 已知路由器R6有表4.4 (a)所示的路由表。现在收到相邻路由器R4发来的路由更新信息，如表4.4 (b)所示。试更新路由器R6的路由表。

表4.4(a) 路由器R6的路由表

|  |  |  |
| --- | --- | --- |
| 目的网络 | 距离 | 下一跳路由器 |
| Net2 | 3 | R4 |
| Net3 | 4 | R5 |
| ... | ... | ... |

表4.4(b) R4发来的路由更新信息

|  |  |  |
| --- | --- | --- |
| 目的网络 | 距离 | 下一跳路由器 |
| Net1 | 3 | R1 |
| Net2 | 4 | R2 |
| Net3 | 1 | 直接交付 |

【解】如同路由器一样，我们不需要知道该网络的拓扑结构。

先把表4.4(b)中的距离都加1，并把下一跳路由器都改为R4。得出表4.4(c)。

表4.4(c) 修改后的表4.4(b)

|  |  |  |
| --- | --- | --- |
| 目的网络 | 距离 | 下一跳路由器 |
| Net1 | 4 | R4 |
| Net2 | 5 | R4 |
| Net3 | 2 | R4 |

把这个表的每一行和表4.4(a)进行比较

第一行在表4.4(a)中没有，因此要把这一行添加到表4.4(a)中。

第二行的Net2在表4.4(a)中有，且下一跳路由器也是R4。因此要更新（距离增大了）。

第三行的Net3在表4.4(a)中有，但下一条路由表不同。于是就要比较距离。新的距离是2，小于原来表中的4，因此要更新。

这样，得出更新后的R6路由表如表4.4(d)所示。

表4.4(d) 路由器R6更新后的路由表

|  |  |  |
| --- | --- | --- |
| 目的网络 | 距离 | 下一跳路由器 |
| Net1 | 4 | R4 |
| Net2 | 5 | R4 |
| Net3 | 2 | R4 |
| ... | ... | ... |

**3、RIP报文的格式**

现在较新的RIP版本是1998年11月公布的RIP2[RFC2453]，已成为因特网标准协议。新版本协议本身并无多大变化，但性能上有些改进你。RIP2可以支持边长子网掩码和CIDR。此外，RIP2还提供简单的鉴别过程支持多播。

图4.6是RIP2的报文格式。它和RIP1的首部相同，但后面的路由部分不一样。从图4.6还可看出，RIP协议使用运输层的用户数据报UDP进行传送。

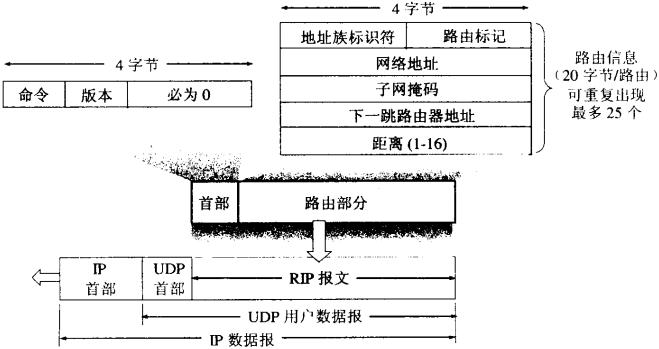


图4.6 RIP2的报文格式

各字段含义请参考教材。

除了前面的三个基本要点外，OSPF还有下列一些特点：

（1）OSPF对不同的链路可根据IP分组的不同服务类型（TOS）而设置不同的代价。因此，OSPF对不同类型的业务可计算出不同的路由这种灵活性是RIP所没有的。

（2）如果到同一个目标网络有多条相同代价的路径，那么可以将通信量分配给这几条路径。这叫做多路径间的负载均衡（load balancing）。

（3）所有在OSPF路由器之间交换的分组（如两路状态更新分组）都具有鉴别的功能，因而保证了仅在可信赖的路由器之间交换链路状态信息。

（4）OSPF支持可变长度的子网划分和无分类的编址CIDR。

（5）由于网络中的链路状态可能经常发生变化，因此OSPF让每一个链路状态都带上一个32位的序号，序号越大状态就越新。

**3. OSPF的五种分组类型**

OSPF共有以下五种分组类型：

（1）类型1，问候（Hello）分组，用来发现和维持邻站的可达性。

（2）类型2，数据库描述（Database Description）分组，向邻站给出自己的链路状态数据库中所有链路状态项目的摘要信息。

（3）类型3，链路状态请求（Link State Request）分组，向对方请求发送某些链路状态项目的详细信息。

（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。这种分组是最复杂的，也是OSPF协议最核心的部分。

（5）类型5，链路状态确认（Link State Acknowledgment）分组，对链路更新分组的确认。

图4.7给出了OSPF的基本操作，说明了两个路由器需要交换各种类型的分组。

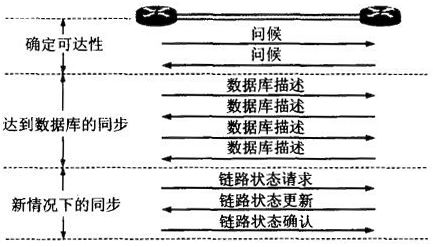


图4.6 OSPF的基本操作

OSPF规定，每两个相邻路由器每隔10秒钟要交换一次问候分组。若有40秒没有收到某个相邻路由器发来的问候分组，则可认为该相邻路由器是不可达的。

OSPF使用的是可靠的洪泛法，即在收到更新分组后要发送确认。

为了确保链路状态数据库与全网的状态保持一致，OSPF还规定每隔一段时间，如30分钟，要刷新一次数据库中的链路状态。

#### 4.5.3 OSPF协议

**1、工作原理**

**开放最短路径优先OSPF**(Open Shortest Path First)市委客服RIP的缺点在1989年开发出来的。OSPF的第二个版本OSPF2已成为因特网正式标准[RFC2328]。

OSPF使用分布式的链路状态协议(link state protocol)，它的三个要点和RIP都不一样：

(1) 向本自治系统中的所有路由器发送信息。使用洪泛法(flooding)。RIP仅仅向相邻的几个路由器发送信息。

(2) 发送的信息是自己与所有相邻路由器之间的链路状态。所谓“**链路状态**”就是说明本路由器都和那些路由器相邻，以及该链路的“**度量**”(metric)。OSPF用这个“度量”来表示费用、距离、时延等等。有时为了方便就称这个度量为“**代价**”。

(3) 只有当链路状态发生变化时，路由器才使用洪泛法发送此信息。RIP是定期交换信息。

由于各路由器向其他路由器发送自己的邻接链路状态，因此所有的路由器最终都能建立起一个链路状态数据库(link-state database)，这个数据库实际上就是全网的拓扑结构图，它在每一个路由器中都是一模一样的。各个路由器使用同样一份数据，计算自己的单源最短路径，从而构造出路由表。

RIP与OSPF的**主要区别**是：前者交换的信息，包含从其他路由器获取的信息，即“我的信息依赖你的，而你的信息又依赖我的”，因而这个信息是不可靠的。后者交换的信息是自己与相邻路由器的链路状态，这个信息是可靠的，因为发送者本身可以验证它。因此，OSPF不存在“坏消息传播的慢”的问题。

OSPF的主要优点是：(1) 收敛快，当一个路由器有了全网的链路状态数据库，可以在本地立即计算出自己到各点的最短路径，保证一次收敛；(2) 规模大，由于每个路由器只关心自己与相邻路由器的链路状态，与整个网络的路由器数目无关，因此OSPF可以适应很大规模的互联网。

OSPF还可以将本自治系统再划分为若干个更小的范围，叫做**区域**(area)。

**2、OSPF报文的格式**

OSPF不使用UDP而直接使用IP数据报传送信息，这样可以减少通信量。

OSPF使用24字节的固定长度首部，如图4.7所示。

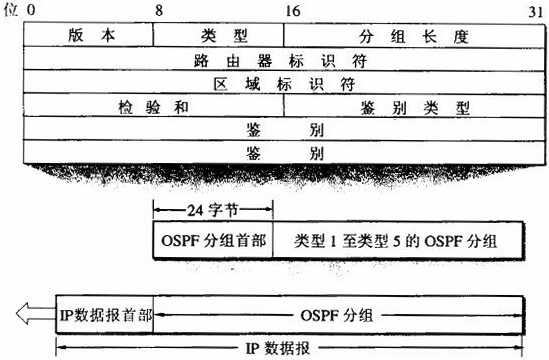


图4.7 OSPF分组用IP数据报传送

各字段含义请参考教材。

除了前面的三个基本特征外，OSPF还有下列一些特点。

(1) OSPF对不同的链路可根据IP分组的不同服务类型(TOS)而设置成不同代价，因此OSPF对不同类型的业务可计算出不同的路由，这种灵活性是RIP所没有的。

(2) 如果到同一个目的网络有多条相同代价的路径，那么可以将通信量分配给这几条路径，这叫做多路径负载均衡(load balancing)。

(3) 所有在OSPF路由器之间交换的分组（例如，链路状态更新分组）都具有鉴别的功能，因而保证了仅在可信赖的路由器之间交换链路状态信息。

(4) OSPF支持可变长度的子网划分和无分类的编制CIDR。

(5) 由于网络中链路状态可能经常发生变化，因此OSPF让每一个链路状态都带上一个32位的序号，序号越大状态就越新。

**3. OSPF的五种分组类型**

OSPF共有以下五种分组类型：

（1）类型1，问候(Hello)分组，用来发现和维持邻站的可达性。

（2）类型2，数据库描述(Database Description)分组，向邻站给出自己的链路状态数据库中的所有链路状态项目的摘要信息。

（3）类型3，链路状态请求(Link State Request)分组，向对方请求发送某些链路状态项目的详细信息。

（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。这种分组是最复杂的，也是OSPF协议最核心的部分。

（5）类型5，链路状态确认(Link State Acknowledgment)分组，对链路更新分组的确认。

图4.7 给出了OSPF的基本操作，说明了两个路由器需要交换各种类型的分组。

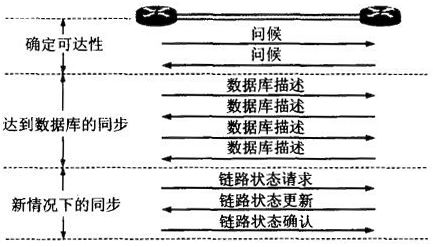


图4.7 OSPF的基本操作

OSPF规定，每两个相邻路由器每隔10秒钟要交换一次问候分组。若有40秒没有收到某个相邻路由器发来的问候分组，则可认为该路由器是不可达的，应立即修改链路状态数据库。

OSPF使用的是可靠的洪泛法，即在收到更新分组后要发送确认。

为了确保链路状态数据库与全网的状态保持一致，OSPF还规定每隔一段时间，如30分钟，要刷新一次数据库中的链路状态。

#### 4.5.4 BGP协议

不同AS之间的路由为什么不能使用前面讨论过内部网关协议，如RIP或OSPF？我们知道，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。在一个AS内部也不需要考虑其他方面的策略。然而BGP使用的环境不同。这主要是因为以下两个原因：

(1) 因特网的规模太大。如果使用OSPF，每个路由器要维护一个很大的链路状态数据库，计算最短路径花费的时间也太长。另外，不同AS对代价的定义是不同的，当一条路径通过不同AS时，要计算出有意义的代价是不可能的。比较合理的做法是在AS之间交换“可连性”信息，例如，告诉相邻路由器：“到达网络N可经过ASx”。

(2) AS之间的路由选择必须考虑有关政策。主要是政治、经济和安全方面的策略。

由于上述情况，边界网关协议BGP只是力求寻找较好路由而不是最佳路由。

BGP使用了路径向量协议(path vector)协议。BGP的路由表中的每一条路由主要包含3项信息：

（目的网络前缀，下一跳地址，所要经过的AS序列）

#### 4.5.5 路由器的结构

图4.8给出了一种典型的路由器的结构框图。

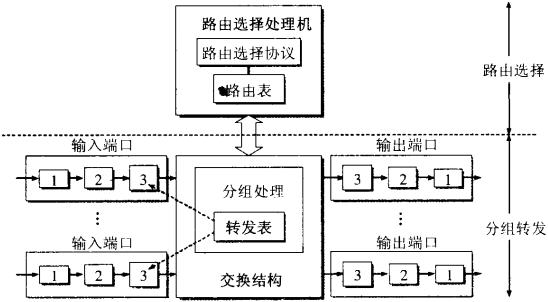


图4.8 典型的路由器的结构

从图中可以看出，路由器结构可划分为两大部分；路由选择部分和分组转发部分。

路由选择部分也叫控制部分，其核心构件是路由选择处理机，它的任务是根据所选定的路由协议构造出路由表。

分组转发部分由三部分组成：一组输入端口，交换结构，一组输出端口。

交换结构(switching fabric)的作用是，根据转发表将从某个输入端口进入的分组转移到某个合适的输出端口。常用的交换结构有三种；存储器，总线，互联网络(innerconnection network)。

转发和路由选择是有区别的。转发仅仅涉及到一个路由器，而路由选择要涉及到多个路由器。

### 4.6 IP多播

#### 4.6.1 IP多播的基本概念

IP多播分为两种，一种是只在局域网上进行硬件多播，另一种是在因特网上进行多播。

在因特网上进行多播叫做IP多播。

显然，多播数据报的首部只能有一个目的IP地址，不可能写入多个IP地址，因此多播数据报的目的地址写入的是多播组的标识符，然后让加入到这个多播组的主机的IP地址与这个多播组关联起来。

多播组的标识符就是IP地址中的D类地址，一个D类地址标识一个多播组。

多播数据报与普通IP数据报的区别是，它使用D类地址作为目的地址，并且首部的协议字段值是2，表示使用IGMP协议。

IP多播主要靠路由器来实现。能够运行多播协议的路由器称为多播路由器(multicast router)。

注意：

（1）多播数据报“尽最大努力交付”，不提供可靠交付。

（2）多播地址只能用于目的地址，不能用于源地址。

（3）对多播数据报不产生ICMP差错报文，因此若ping一个多播地址将永远不会收到响应。

（4）并非所有的D类地址都可以作为多播地址。

#### 4.6.2 在局域网上进行硬件多播

因特网号码指派管理局IANA拥有的以太网地址块的高24位是00-00-5E，以太网硬件地址**最高字节的最低位**为1时是多播地址，因此多播地址的范围是01-00-5E-00-00-00到01-00-5E-FF-FF-FF。IANA只用其中一半做为多播地址，因此实际上以太网的多播地址的范围是01-00-5E-00-00-00到01-00-5E-7F-FF-FF。

为了把IP多播地址映射到以太网的多播地址，采用了一种折中方案，即把IP多播地址的低23位放入01-00-5E-00-00-00的低23位，组成一个硬件多播地址。D类地址可供分配的有28位，有5位被丢弃了，多个IP多播地址可能映射到同一个硬件多播地址，因此，收到多播数据报的主机，还要在IP层进行过滤。若该主机加入的多播组的标识与数据报中的目的地址不相同，则丢弃该数据报。

#### 4.6.3 IGMP协议

IGMP协议是让连接在本地局域网上的多播路由器知道本局域网上是否有主机参加或退出了某个多播组。IGMP不知道IP多播组包含的成员数，也不知道这些成员都分布在哪些网络上。

显然，仅有IGMP协议是不能完成多播任务的。连接在局域网上的多播路由器还必须和因特网上的其他路由器协同工作，以便把多播数据报用最小代价传送给所有的组成员。这就需要使用多播路由选择协议。

#### 4.6.4 多播路由选择协议

多播路由选择协议比单播路由选择协议复杂得多。我们可以通过一个简单的例子来说明，如图4.9。

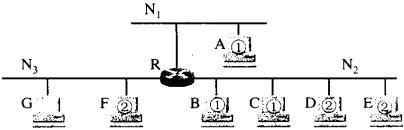


图4.9 用来说明多播路由选择的例子

(1) 每一个主机可以随时加入或离开一个多播组，因此多播转发必须动态地适应多播组成员的变化（（这时网络拓扑并未发生变化）。

(2) 多播路由器在转发多播数据报时，不能仅仅根据多播数据报中的目的地址，而是还要考虑这个多播数据报从什么地方来和要到什么地方去。

(3) 多播数据报可以由没有加入多播组的主机发出，也可以通过没有组成员接入的网络。

多播路由选择协议一般使用了以下三种方法。

(1) 洪泛与剪除。这种方法适合于较小的多播组，而所有的组成员接入的局域网也是相邻接的。一开始，路由器转发多播数据报使用洪泛的方法。为了避免兜圈子，采用了叫做**反向路径广播RPB**(Reverse Path Broadcasting)的策略。

(2) 隧道技术(tunneling)。隧道技术适用于多播组的位置在地理上很分散的情况。

(3) 基于核心的发现技术。

虽然在TCP/IP中IP多播协议已成为建议标准，但多播路由选择协议（用来在多播路由器之间传播路由信息）则尚未标准化。

注意：

（1）多播数据报“尽最大努力交付”，不提供可靠交付。

（2）多播地址只能用于目的地址，不能用于源地址。

（3）对多播数据报不产生ICMP差错报文，因此若ping一个多播地址将永远不会收到响应。

（4）并非所有的D类地址都可以作为多播地址。

### 4.7 VPN和NAT

#### 4.7.1 虚拟专用网VPN

RFC1918指明的专用地址(private address)是：

1. 10.0.0.0到10.255.255.255，即一个A类网络。
2. 172.16.0.0到172.31.255.255，即16个B类网络。
3. 192.168.0.0到192.168.255.255，即256个C类网络。

如果分布在不同地点的专用网需要进行通信，这是有两种方法。第一种是租用电信公司的通信线路为本机构专用。第二种方法是采用VPN（Virtual Private Network，虚拟专用网）技术。

一个机构要构建自己的VPN就必须为它的每一个场所购买支持VPN的路由器。VPN路由器使用隧道技术实现虚拟专用网。

#### 4.7.2 网络地址转换NAT

如果专用网内部的主机箱要和因特网上的主机通信，目前使用的最多的方法是**网络地址转换NAT**(Network Address Translation)。

这种方法需要专用网连接到因特网路由器支持NAT。支持NAT的路由器叫做NAT路由器。

显然，通过NAT路由器的通信必须由专用网内部的主机发起，因此，专用网内部的主机是不可能充当服务器的。

为了更加有效地利用NAT路由器上的全球IP地址，现在常用的NAT转换表把运输层的端口号也利用上。这样，一个主机可以同时和因特网上的多个主机进行通信。

使用端口号的NAT叫做网络地址与端口号转换NAPT(Network Address and Port Translation)。但在许多文献中，还是使用NAT这个更加简单的缩写。表4.5说明了NAPT的地址转换机制。

表4.5 NAPT地址转换表举例

|  |  |  |  |
| --- | --- | --- | --- |
| 方向 | 字段 | 旧的IP地址和端口号 | 新的IP地址和端口号 |
| 出 | 源IP地址：TCP源端口 | 192.168.0.3:30000 | 172.38.1.5:40001 |
| 出 | 源IP地址：TCP源端口 | 192.168.0.4:30000 | 172.38.1.5:40002 |
| 入 | 目的IP地址：TCP目的端口 | 172.38.1.5:40001 | 192.168.0.3:30000 |
| 入 | 目的IP地址：TCP目的端口 | 172.38.1.5:40002 | 192.168.0.4:30000 |

### 4.8 IPv6

#### 4.8.1 IPv6的基本首部

IPv6所引进的主要变化如下：

（1）更大的地址空间。IPv6把IP地址从32位增大到128位。

（2）扩展的地址层次结构。IPv6由于地址空间很大，因此可以划分为更多的层次。

（3）灵活的首部格式。IPv6数据包的首部和IPv4的并不兼容。IPv6定义了许多可选的扩展首部，不仅可提供比IPv4更多的功能，而且还可提高路由器的处理效率，这是因为路由器对扩展首部不进行处理（除逐跳首部外）。

（4）改进的选项。

（5）允许协议继续扩展。

（6）支持即插即用（即自动配置）。

（7）支持资源的预分配。

IPv6数据报在**基本首部**（base header）的后面允许有零个或多个扩展首部（extension header），再后面是数据（图4.10）。但请注意，所有的扩展首部都不属于IPv6数据报的首部。所有的扩展首部和数据合起来叫做数据报的**有效载荷**(payload)或净负荷。

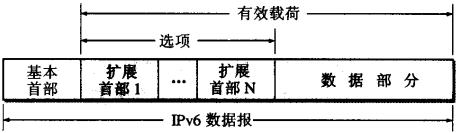


图4.10 具有多个扩展首部的IPv6数据报的一般形式

图4.11是IPv6数据报的基本首部。

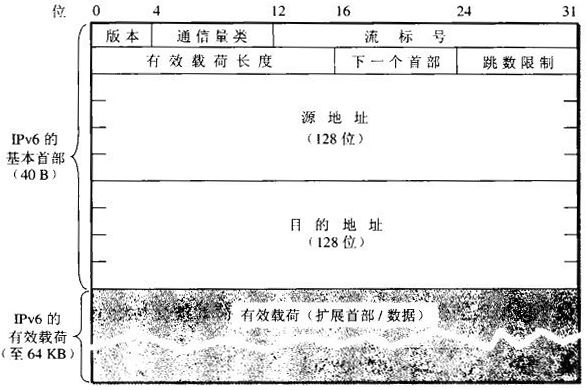


图4.11 40字节长的IPv6基本首部

与IPv4相比，IPv6对首部中的某些字段进行了如下更改：

取消了首部长度字段，因为它的首部长度是固定的（40B）。

取消了服务类型字段，因为优先级和流标号合起来实现了服务类型字段的功能。

取消了总长度字段，改用有效载荷字段。

取消了标识、标志和片偏移字段，因为这些功能已包含在分片扩展首部中。

把TTL字段改称为跳数限制字段，名称与作用更加一致。

取消了协议字段，改用下一个首部字段。

取消了校验和字段，这样就加快了路由器处理数据报的速度。我们知道，在数据链路层对检测出有差错的帧就丢弃。在运输层，当使用UDP时，若检测出有差错的用户数据报就丢弃，当使用TCP时，若检测出有差错的报文段就重传。因此在网络层的差错检测功能可以精简掉。

取消了选项字段，而用扩展首部来实现选项功能。

下面解释IPv6基本首部中各字段的作用。

（1）版本(version)——占4位。指明IP协议的版本，对IPv6该字段是6。

（2）通信量类(traffic lass)——占8位。这是为了区分不同的IPv6数据报的类别或优先级。

（3）流标号(flow label)——占20位。对于非实时数据，流标号则没有用处。

（4）有效载荷长度(payload length)——占16位，单位为字节。

（5）下一个首部(next header)——占8位。它相当于IPv4的协议字段或可选字段。

（6）跳数限制(hop limit)——占8位。

（7）源地址——占128位。

（8）目的地址——占128位。

#### 4.8.2 IPv6地址

IPv6的地址有128位，即16个字节。

IPv6地址的3种类型：单播(unicast)、多播(multicast)、任播(anycast)。

用点分十进制记法太长，IPv6使用冒号十六进制记法（colon hexadecimal notation，简写为hex colon），它把每个16bit的值用十六进制表示，各值之间用冒号分隔。

冒号十六进制记法还包含两个技术使它尤其有用。首先，可以允许零压缩(zero compression)，即一连串的0可以为一对冒号所取代。例如FF05:0:0:0:0:0:0:B3可以写成FF05::B3。

其次，可以结合使用点分十进制记法的后缀。例如下面的串是一个合法的冒号十六进制记法：0:0:0:0:0:0:128.10.2.1，再使用零压缩即可得出:: 128.10.2.1。

### 4.9 移动IP

#### 4.9.1 移动IP的概念

使用移动IP，一个主机可以在不改变IP地址的情况下改变其地理位置。

一个移动结点跟一个称作家乡IP地址的固定IP地址相关联。当移动结点在外部网络（即离开家乡网络）的时候，在移动结点的家乡网络上的一个路由器把IP数据报投递给移动结点。该路由器被称作**家乡代理**。

当一个移动结点出现在外部网络时，它跟那里的**外部代理**（通常是外部网络上的一个路由器）联系，并进行注册，然后外部代理跟家乡代理联系，给它一个关照地址（通常就是外部代理自己的IP地址）。

#### 4.9.2 移动IP通信过程

当发给移动用户的一个IP分组到达用户的家乡LAN时，它在连接到LAN的某个路由器上到达，该路由器用常规的方法确定目的主机的物理地址，即广播一个ARP分组进行询问。家乡代理通过给出自己的MAC地址应答这个ARP请求分组，然后路由器把IP分组发送到家乡代理。

家乡代理把IP分组隧道传送给关照地址，即把该IP分组封装在另一个IP分组的数据部分，后者的目的地址指向外部代理。外部代理收到IP分组后，将其解除封装后把原IP分组包装在数据链路帧中投递给移动结点。此外，家乡代理还把关照地址发送给IP分组的发送方，随后，发送方可以把后续的IP分组直接隧道传送给外部代理。

### 4.10 网络层设备

网络层设备即路由器。

本章我们所碰到过的路由器有，普通路由器，多播路由器，VPN路由器，NAT路由器。

## 第5章 传输层

### 5.1 传输层提供的服务

#### 5.1.1 传输层的功能

(1) 复用和分用。

(2) 传输层提供应用进程之间的逻辑通信（即端到端的通信）。与网络层的区别是，网络层提供的是主机之间的逻辑通信。

(3) 对收到的报文进行差错检测。而网络层只检查IP数据包的首部。

(4) 提供两种不同的传输协议，即面向连接的TCP和无连接的UDP。而网络层无法同时实现两种协议（即网络层要么只提供面线连接的服务，如虚电路，要么只提供无连接服务，如数据报，而不可能再网络层同时存在着两种方式）。

#### 5.1.2 传输层的寻址与端口

传输层从IP层收到数据后必须交付给指定的应用进程。这就是分用。显然，给应用层的每一个进程赋予一个非常明确的标识是至关重要的。解决方法是在传输层使用协议端口号(protocol port number) ，通常简称为**端口**(port)。

为什么不使用进程ID标识报文投递的目的实体呢？（1）进程ID是操作系统特有的，不同的操作系统使用不同格式的进程ID；（2）单个进程有可能建立多个通信，用进程ID达不到这个目的。

端口号有16位，共分为两大类：

(1) 服务器端使用的端口号。这里又分为两类，最重要的一类叫做**熟知端口号**(well-known port number)或**系统端口号**，数值为0~1023。这些数值可以在网址[www.iana.org](http://www.iana.org) 查到。IANA把这些端口号指派给了TCP/IP最重要的一些应用程序。

下面给出一些常用的熟知端口。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 应用程序 | FTP | TELNET | SMTP | DNS | TFTP | HTTP |
| 熟知端口号 | 21 | 23 | 25 | 53 | 69 | 80 |

另一类叫做**登记端口号**，数值为1024~49151(3×214-1)。这类端口号是为没有数值端口号的应用程序使用的。使用这类端口号必须在IANA按照规定的手续登记，以防止重复。

(2) 客户端使用的端口号。数值为49152~65535。由于这类端口仅在客户进程运行时才动态选择，因此又叫作**短暂端口号**(ephemeral port)。

#### 5.1.3 无连接服务与面向连接服务

(1) UDP是无连接的，不可靠的，面向报文的，无拥塞控制，支持一对一，一对多，多对多的交互通信。TCP是面向连接的，可靠的，面向字节流的，有拥塞控制，支持一对一的全双工通信。

(2) TCP以报文段(segment)为单位，UDP以报文(message)为单位。

### 5.2 UDP协议

#### 5.2.1 UDP数据报

UDP首部只有8字节，由4个字段组成，每个字段的长度都是两个字节，如图5.1所示。

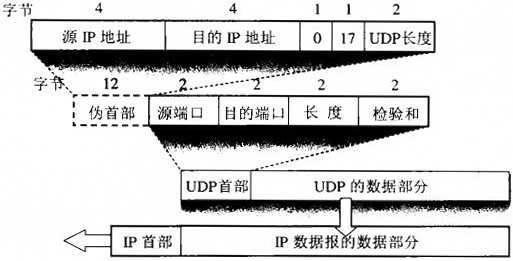


图5.1 UDP报文的格式

各字段含义如下：

(1) 源端口 源端口号。在需要对方回信时选用，不需要时可用全0。

(2) 目的端口 目的端口号。这在目的站交付报文时必须要使用到。

(3) 长度 UDP报文的长度，其最小值是8（仅有首部）。

(4) 校验和 检测UDP报文在传输中是否有错，有错就丢弃

#### 5.2.2 UDP校验

UDP在计算校验和时有些特殊。在计算校验和时候，要在UDP报文之前增加12字节的伪首部。所谓“伪首部”是因为这种首部并不真正存在，仅仅是为了计算校验和。

IP协议只校验IP报文的首部，而UDP则是首部和数据部分一起都校验。当无错时结果应为全1。

### 5.3 TCP协议

#### 5.3.2 TCP的连接

#### 5.3.1 TCP报文段

TCP报文段的格式如图5.2所示。

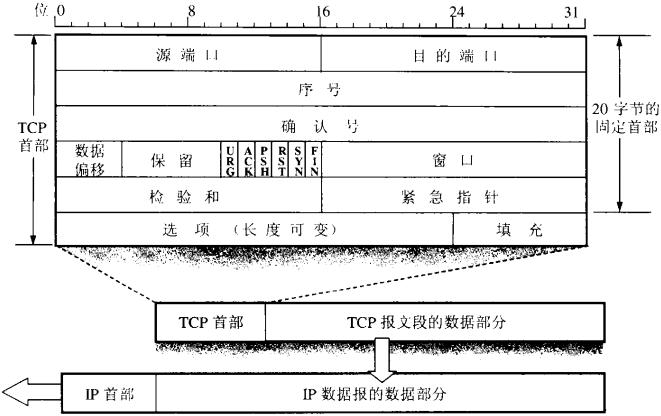


图5.2 TCP报文的格式

各字段含义请参考教材。

#### 5.3.2 TCP的连接管理

TCP连接的端点叫做**套接字**(socket)。根据RFC793的定义：端口号拼接到IP地址即构成了套接字。因此套接字的表示方法是在点分十进制的IP地址后面写上端口号，中间用冒号隔开，即

套接字socket::=（IP地址：端口号）

每一条TCP连接唯一地被通信两端的两个端点（即两个套接字）所确定，即

TCP连接::={socket1, socket2} = {IP1:port1, IP2:port2}

TCP连接有3个阶段；连接建立，数据传输，连接释放。

**1. 连接建立**

TCP连接的建立采用客户服务器方式。主动发起连接建立的应用进程叫做客户(client)，而被动等待连接建立的应用进程叫做服务器(server)。

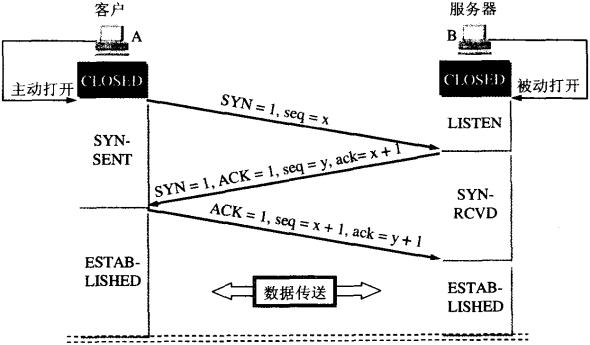


图5.7 三次握手建立连接

为什么A还要发送一次确认呢？这主要是为了防止已失效的连接请求报文突然又传送到了B，从而产生错误。

**2. 连接释放**

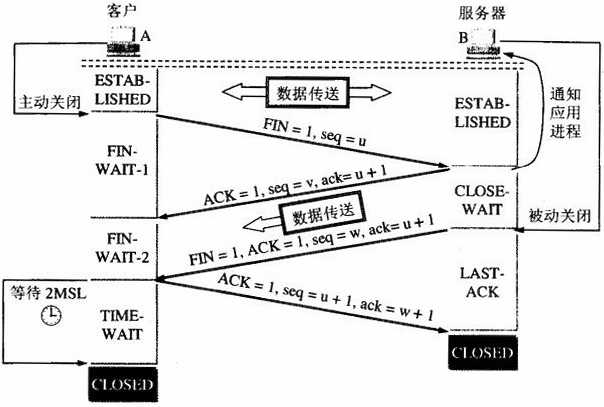


图5.8 四次握手释放连接

#### 5.3.3 TCP的可靠传输机制

TCP的可靠传输机制实质上就是一种选择重传ARQ协议，只不过更加精细。

**1. 以字节为单位的滑动窗口**

接收方采用累积确认，但不应过分推迟发送确认，否则会导致发送方不必要的重传。TCP标准规定，确认推迟的时间不应超过0.5秒。

**2. 超时重传时间的选择**

TCP采用了一种自适应算法。

TCP保留了RTT的一个加权平均往返时间RTTs（s表示smooth），当第一次测量RTT样本时，RTTs值就为所测量到的RTT样本值，但以后没测量一个新的RTT样本，就按下时重新计算一次RTTs：

新的RTTs=(1-α) ×旧的RTTs+α×新的RTT样本

在上式中，0≤α≤1。若α很接近于零，表示新的RTTs值和旧的RTTs值相比变化不大，受新的RTT样本影响不大（RTTs值更新较慢）。若α接近于1，表示新的RTTs值受新的RTT样本影响较大（RTTs值更新较快）。RFC2988推荐的α值为1/8，即0.125。

超时重传时间RTO(Retransmission Time-Out)应略大于RTTs。RFC2988建议使用下式计算RTO：

RTO=RTTs+4×RTTD

其中RTTD是RTT的偏差的加权平均值。RFC2988建议这样计算RTTD。当第一次测量时，RTTD值取为测量到的RTT样本值的一半。在以后的测量中，使用下式计算RTTD：

新的RTTD=(1-β) ×旧的RTTD+β×|RTTs-新的RTT样本|

这里β是个小于1的系数，它的推荐值是1/4，即0.25。

由于对于一个重传报文段的确认来说，很难分辨它是原报文段的确认还是重传报文段的确认，因此使用修正的Karn算法：在计算RTTs时，只要报文段重传了，就不采用其往返时间样本，且报文段每重传一次，就把RTO增大一些。典型的做法是取新的RTO为旧的RTO的2倍。

**3. 选择确认SACK**

对于不按序到达的字节应如何处理，TCP标准并无明确规定。接收方可以直接丢弃，等待重传，也可以暂存下来，设法只重传缺少的数据。选择确认(Selective ACK)就是这样一种机制。

RFC2018规定，如果要使用选择确认，就要在TCP首部的选项中增加“SACK”选项，以便报告收到的不连续的字节块的边界。

#### 5.3.4 TCP的流量控制

**1. 利用滑动窗口时下流量控制**

所谓**流量控制**(flow control)就是让发送方的发送速率不要太快，让接收方来得及接收。

利用滑动窗口可以很方便地在TCP连接上实现流量控制。

下面通过图5.3的例子说明如何利用滑动窗口机制进行流量控制。

设A向B发送数据。在连接建立时，B告诉了A：“我的接收窗口rwnd=400”（协商过程在图中没有显示出来）。因此，发送方的发送窗口不能超过接收方给出的接收窗口的数值。请注意，TCP的窗口单位是字节。设图中每次发送的数据为100字节长，字节序号从1开始（见图中第一个箭头上面的序号seq=1）。图中箭头上面大写的ACK表示首部中的确认为ACK，小写ack表示确认字段的值。

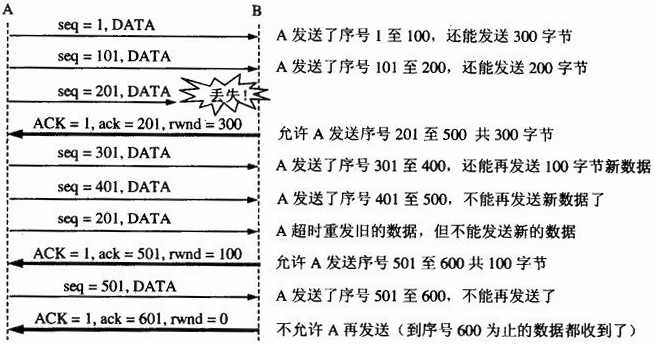


图5.3 利用滑动窗口进行流量控制举例

从图中可以看到，接收方B进行了3次流量控制。

现在我们考虑另一种情况。在图5.3中，B向A发送了一个零窗口的报文后不久，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，然而这个报文在传送过程中丢失了。A一直等待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持续下去。

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（仅携带1字节的数据），而对方就在确认这个探测报文时给出现在的窗口值。如果窗口仍然是零，那么收到这个报文的一方就重新设置持续计时器。如果不是零，那么死锁的僵局就可以打破了。

#### 5.3.5 TCP的拥塞控制

**1. 拥塞控制的概念**

所谓**拥塞控制(**congestion control**)**就是防止过多的数据注入到网络中，这样可以是网络中的路由器或链路不致过载。拥塞控制是一个全局性过程，流量控制只是一个端到端的问题。

**2. 几种拥塞控制方法**

1999年公布的因特网建议标准RFC2581定义了进行拥塞控制的四种算法，即**慢开始**(slow-start)、**拥塞避免**(congestion avoidance)、**快重传**(fast retransmit)和**快恢复**(fast recovery)。以后RFC2582和RFC3390又对这些算法进行了一些改进。下面就介绍这些算法的原理。

**（1）慢开始和拥塞避免**

while(true) {

cwnd = 1;

// sstrresh = ...，第一次的初值，书中并未涉及

while(按时收到一个确认) {

if(cwnd < ssthresh) { // 慢开始阶段

cwnd \* =2;

} else if(cwnd > ssthresh) { //拥塞避免阶段

cwnd += 1;

} else { // 二者皆可

cwnd \*= 2;

// cwnd += 1;

}

}

// 没有按时收到确认，则认为网络出现了拥塞

// 发送窗口的一半，twnd=min{rwnd,cwnd}

ssthresh = twnd/2;

}

慢开始的“慢”并不是指拥塞窗口cwnd的增长速率慢，而是指TCP开始发送时先设置cwnd=1，起点很慢。

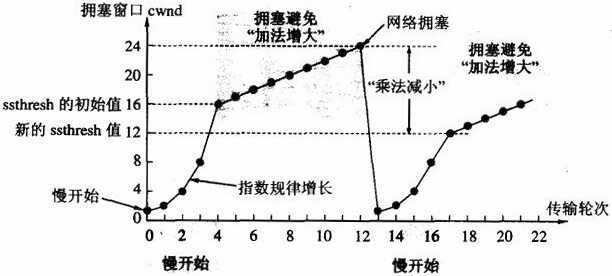


图5.4 慢开始和拥塞避免

**（2）快重传和快恢复**

快重传算法：接收方每收到一个是失序报文段后就立即发出重复确认。这样做可以让发送方即早知道有报文段没有到达接收方。发送方只要一连收到三个重复确认就应当立即重传对方的未收到的报文段，而不必等待重传计时器到期，节省了时间。

快恢复算法：(1) 当发送方收到连续三个重复的确认是，就执行“乘法减小”算法，把慢开始门限ssthresh减半；(2) 与慢开始（慢开始算法将拥塞窗口cwnd设置为1）不同之处是把cwnd的值设置为慢开始门限ssthresh减半后的数值，然后开始执行拥塞避免算法（“加法增大”）。

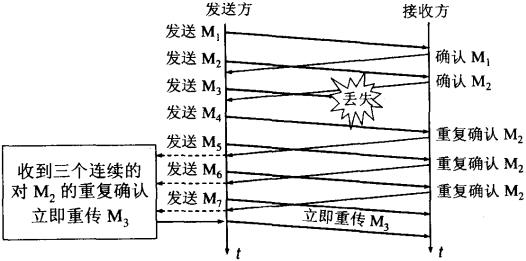


图5.5 快重传的示意图

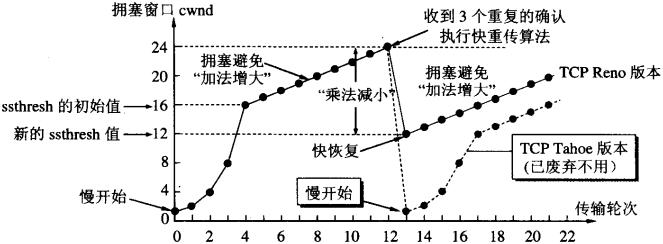


图5.6 连续收到三个重复的确认转入拥塞避免

如果把本节所讨论的拥塞控制和接收方对发送方的流量控制一起考虑，那么很显然，发送方的窗口上限值应当取接收窗口rwnd和拥塞窗口cwnd这两个变量中较小的一个。

## 第6章 应用层

### 6.1 网络应用模型

客户-服务器模型和P2P模型。

### 6.2 DNS系统

域名系统DNS(Domain Name System)用于把域名转换为IP地址，它是一个联机分布式数据库。

#### 6.2.1 层次域名结构

从语法上讲，每一个域名都是由标号序列组成，而个标号之间用点隔开。

DNS规定，域名中的标号都由英文字符和数字组成，每一个标号不超过63个字符，也不区分大小写字母。标号中除连字符“-”外不能使用其他的标点符号。级别最低的标号写在最左边，而级别最高的标号则写在最右边。有多个标号组成的完整域名总共不超过255个字符。DNS既不规定一个域名需要包含多少个下级域名，也不规定每一级的域名代表什么意思。各级域名由其上一级的域名管理机构管理，而最高的顶级域名则由ICANN进行管理。用这种方法可使每一个域名在整个因特网的范围内是唯一的，并且也容易设计出一种查找域名的机制。

根据2006年12的统计，现在顶级域名TLD(Top Level Domain)已有256个，分为三大类：

(1) 国际顶级域名nTLD：采用ISO 3166的规定。如：cn表示中国，us表示美国，uk表示英国，等等。国家顶级域名又常记为ccTLD（cc表示国家代码，country-code）。

(2) 通用顶级域名gTLD：最常用的通用顶级域名有3个，com（公司企业），net（网络服务机构），org（非营利性组织）。

(3) 基础结构域名(infrastructure domain)：这种顶级域名只有一个，即arpa，用于反向域名解析，因此又称为反向域名。

#### 6.2.2 域名服务器

因特网上的DNS域名服务器也是按层次安排的，如图6.1所示。

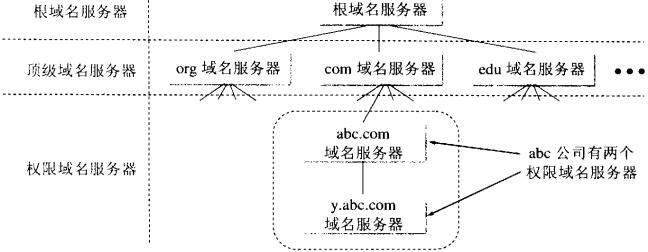


图6.1 树状结构的DNS域名服务器

每一个域名服务器都只对域名体系中的一部分进行管辖。根据域名服务器所起的作用，可以把域名服务器划分为四种不同的类型：

(1) 根域名服务器(root name sever)：根域名服务器是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。根域名服务器是最重要的域名服务器，因为不管是哪一个本地域名服务器，若要对因特网上的任何一个域名进行解析，只要自己无法解析，就首先要求助于根域名服务器。在因特网上共有13个不同IP地址的根域名服务器，他们的名字使用一个英文字母命名，从a一直到m，这些根域名服务器相应的域名分别是a.rootservers.net, ..., m.rootservers.net。但请注意，根域名服务器的数目并不是13个机器。根域名服务器采用了任播技术。

(2) 顶级域名服务器(root name sever)：这些域名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。

(3) 权限域名服务器(authoritative name sever)：一个域名服务器所管理的范围叫做区(zone)。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。当一个权限域名服务器还不能给出最后的查询回答时，就会告诉发出查询请求的DNS客户，下一步应当找哪一个权限域名服务器。

(4) 本地域名服务器(local name sever)：每一个ISP，或一个大学，甚至一个大学里的系，都可以拥有一个本地域名服务器，这种域名服务器有时也称为默认域名服务器。在windows操作系统中设置DNS时设置的就是本地域名服务器。

#### 6.2.3 域名解析过程

域名解析过程有两个要点：

1. 主机向本地域名服务器的查询一般都是采用递归查询(recursive query)。
2. 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(iterative query)。

图6.2说明了这两种查询的差别。

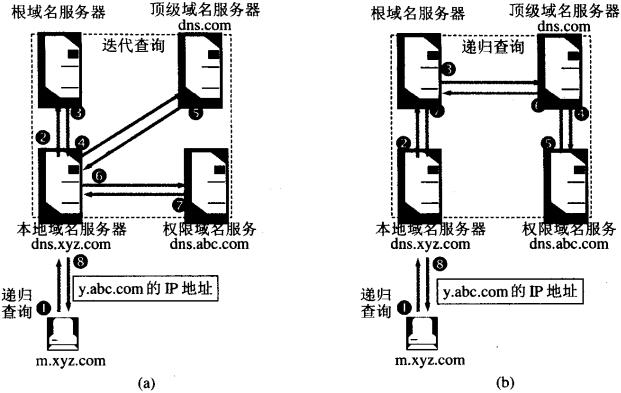


图6.2 DNS查询举例：(a)本地域名服务器采用迭代查询；

(b) 本地域名服务器采用递归查询

域名服务器广泛地使用了高速缓存。普通主机也有高速缓存。很多主机会从本地域名服务器下载全部的数据库。只有在本机缓存找不到时才使用域名服务器。

### 6.3 文件传输协议(FTP)

文件传输协议有两种，基于TCP的FTP和基于UDP的TFTP。

#### 6.3.1 FTP

文件传输协议FTP(File Transfer Protocol)[RFC 959]是因特网上使用的最广泛的文件传输协议。FTP提供交互式访问，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使用客户服务器方式。FTP的服务器进程由两大部分组成：一个主进程，负责接受新的请求；另外有若干个从属进程，负责处理单个请求。

主进程的工作步骤如下：

(1) 打开熟知端口（端口号为21），使客户进程能够连接上。

(2) 等待客户进程发出连接请求。

(3) 启动从属进程来处理客户进程发来的请求。从属进程对客户进程的请求处理完毕后即终止，但从属进程在运行期间根据需要还可能创建其他一些子进程。

(4) 回到等待状态，继续接受其他客户进程发来的请求。子进程与从属进程的处理是并发地进行。

FTP的工作情况如图6.3所示。

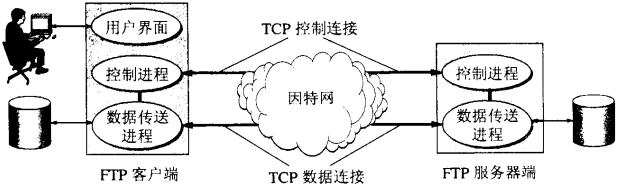


图6.3 FTP的工作原理

图中的服务器端有两个从属进程：控制进程和数据传送进程。为简单起见，主进程没有画上。在进行文件传输时，FTP的客户和服务器之间要建立两个并行的TCP连接：控制连接（端口21）和数据连接（端口20）。

#### 6.3.2 TFTP

简单文件传输协议TFTP(Trivial File Transfer Protocol)是一个很小且易于实现的文件传输协议。TFTP的版本2是因特网正式标准[RFC1350]。TFTP使用UDP数据包，因此TFTP需要有自己的纠错机制。TFTP只支持文件传输而不支持交互。

### 6.4 电子邮件

#### 6.4.1 电子邮件系统的组成结构

电子邮件系统的三个组成部分：用户代理，邮件服务器以及邮件发送协议（如SMTP）和邮件读取协议（如POP3）。

用户代理UA(User Agent)就是运行在用户PC机上的一个程序，因此用户代理又称为**电子邮件客户端软件**。

#### 6.4.2 电子邮件格式与MIME

一个电子邮件分为信封和内容两大部分。在RFC2822中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, Subject, Cc, Reply-To。

SMTP只能传送7位的ASCII码，在这种情况下就出现了通用因特网邮件扩充**MIME**(Multipurpose Internet Mail Extension)[RFC2045~2049]。MIME并没有改动或取代SMTP，MIME的意图是继续使用目前的RFC822格式，但增加了邮件主体的结构，并定义了传送非ASCII码的编码规则。

三种常见的内容传送编码(Content Transferring Encoding)：7位ASCII码，quoted-printable编码，base64编码。

#### 6.4.3 简单邮件传送协议SMTP

SMTP是一种基于文本的电子邮件传输协议，用于在邮件服务器之间交换邮件。

SMTP运行在TCP连接之上，使用25号端口。

SMTP规定了14条命令和21中应答消息。

每一条命令由4个字母组成，如HELO, MAIL, RCPT, DATA和QUIT等。

每一种应答一般只有一行信息，由一个3位数字的代码开头。以2开头的应答意味着成功，3开头的表示有进一步的动作，4和5开头的表示错误，4开头的是暂时性错误，5开头的是永久性错误。

SMTP通信的三个阶段：连接建立，邮件传送，连接释放。

SMTP不使用中间的邮件服务器。

#### 6.4.4 邮件读取协议POP3

现在常用的邮件读取协议有两个，即邮局协议第3个版本POP3和网际报文存取协议IMAP(Internet Message Access Protocol)。

邮局协议POP是一个非常简单、功能有限的邮件读取协议。邮局协议POP最初公布于1984年[RFC918]。经过几次更新，现在使用的是1996年的版本POP3[RFC1939]，它已成为因特网正式标准。大多数的ISP都支持POP。POP3可简称为POP。

POP3协议的一个特点就是，只要用户从POP服务器读取了邮件，POP服务器就把该邮件删除。这在某些情况下不够方便。POP3进行了一些扩充，允许用户设置邮件读取后仍然在服务器中存放的时间[RFC2449]。目前RFC2449还只是因特网建议标准。

#### 6.4.5 基于万维网的电子邮件

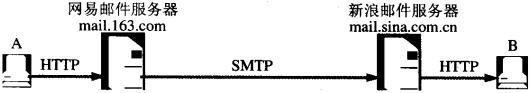


图6.5 基于万维网的电子邮件的工作过程

### 6.5 WWW

#### 6.5.1 万维网概述

**万维网WWW**(World Wide Web)是一个分布式的**超媒体**系统，它是超文本的扩展。所谓**超文本**是包含指向其他文档的连接的文本。超媒体与超文本的区别是，超文本文档仅包含文本信息，而超媒体文档还可以包含其他形式的信息，如图形、图像、声音等。

万维网使用CS方式工作。客户端向服务器发出请求，服务器返回客户端所请求的万维网文档。因此，万维网必须解决以下问题：

1. 怎样标志分布在整个因特网上的万维网文档？
2. 用什么样的协议来实现万维网上的各种链接？
3. 怎样是不同作者创作的不同风格的万维网文档都能在各种主机上正确显示出来，同时使用户清楚地知道在什么地方存在着链接？
4. 怎样使用户能够很方便地找到所需的信息？

为了解决第1个问题，万维网使用统一资源定位符URL(Uniform Resource Locator)；第2个问题，超文本传输协议HTTP(HyperText Transfer Protocol)；第3个问题，超文本标记语言HTML(HyperText Markup Language)；第4个问题，搜索引擎。

#### 6.5.2 统一资源定位符URL

URL的格式：

<协议>://<主机域名>:<端口>/<文件路径>

#### 6.5.3 超文本传输协议HTTP

**1、HTTP的操作过程**

假定用户点击了一个超链接，其URL是<http://www.tsinghua.edu.cn/index.htm> ，指向的是清华大学的首页。下面我们用HTTP/1.0更具体的说明在用户点击鼠标后所发生的一系列事件：

(1) 浏览器向DNS请求解析[www.tsinghua.edu.cn](http://www.tsinghua.edu.cn) 的IP地址。

(2) DNS解析出清华大学服务器的IP地址为166.111.4.100。

(3) 浏览器与服务器建立TCP链接（端口是80）。

(4) 浏览器发出取文件命令：GET /index.html。

(5) 服务器把文件indexhtml发送给浏览器。

(6) 释放TCP连接。

(7) 浏览器显示清华大学首页页面。

虽然HTTP使用了TCP链接，但HTTP是无状态的，也就是说，服务器不记得曾经访问过的这个客户，每一次响应和第一次相应都是相同的。

**HTTP/1.1协议**较好地解决了这个问题，它使用了**持续连接**(persistent connection)。所谓持续连接就是万维网服务器在发送响应后仍然保持这条连接，使同一个客户和该服务器可以继续在这条连接上传送后续的HTTP请求报文和响应报文。这并不局限用于传送同一个页面上链接的文档，而是只要这些文档都在同一个服务器上就行。对于非持续连接，每个网页元素的传输都需要单独建立一个TCP连接，而持续连接只需要建立一个TCP连接就能传输整个网页。

HTTP/1.1协议的持续连接有两种工作方式，即非流水线方式(without pipeline)和流水线方式(with pipeline)。

非流水线方式的特点是，客户在收到前一个响应后才能发出下一个请求。流水线方式的特点是，客户在收到响应报文之前就能够接着发送新的请求报文。

**2、HTTP报文的格式**

HTTP报文是文本格式的，即一个ASCII字符串。

HTTP报文有两类，请求报文和响应报文。如图6.4所示。

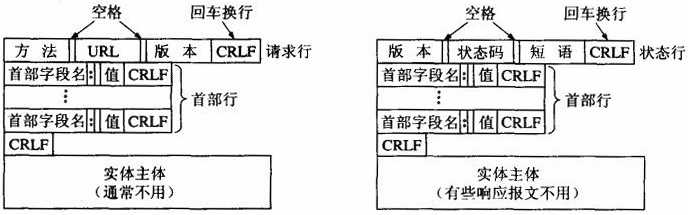


图6.4 HTTP报文的格式：(a) 请求报文；(b) 响应报文

**3、Cookie**

HTTP是无状态的，但一些网站却常常希望能够识别用户。在RFC2109中对Cookie进行了定义，规定万维网站点可以使用cookie来跟踪用户。

Cookie是这样工作的。当用户张三浏览某个使用Cookie的网站时，该网站的服务器就为张三产生一个唯一的识别码，并以此作为索引在服务器的后端数据库中产生一个项目。接着再给张三的HTTP响应报文中添加一个叫做Set-cookie的首部行。这里的“首部字段名”就是“Set-cookie”，而后面的“值”就是赋予该用户的“识别码”。例如这个首部行是这样的：

Set-cookie: 12345678

当张三收到这个响应时，浏览器就在磁盘上的某个Cookie文件中添加一行，其中包含这个服务器的主机名和Set-cookie后面的识别码。当张三继续浏览这个网站时，每发送一个HTTP请求报文时，浏览器就会从这个Cookie文件中取出该识别码，并放到HTTP请求报文的首部行中：

Cookie: 12345678

于是，这个网站就能够跟踪用户12345678（张三）在该网站的活动。

#### 6.5.4 万维网的文档

超文本标记语言HTML，动态文档，活动文档。

### 6.6 DHCP

连接到因特网的计算机的协议软件需要配置的项目包括：IP地址，子网掩码，默认路由器，默认域名服务器。

这些信息通常存储在一个配置文件中。但有时(1)无盘工作站只有ROM没有硬盘；(2)许多便携式计算机经常更换网络，每次人工进行配置既不方便又容易出错。因此，需要采用自动协议配置的方法。

目前广泛使用的是**动态主机配置协议DHCP**(Dynamic Host Configuration Protocol)。

DHCP使用客户服务器方式。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(DHCPDISCOVER)（将目的IP地址置为全1，即255.255.255.255），这时该主机就成为DHCP客户。DHCP服务器先在其数据库中查找该计算机的配置信息，若找到，则返回找到的信息，若找不到，则从服务器的IP地址池(address pool)中取一个地址分配给该计算机。DHCP服务器的回答报文叫做提供报文(DHCPOFFER)，表示“提供”了IP地址等配置信息。

## 参考资料

[1] 计算机网络（第5版），谢希仁，电子工业出版社，2009

[2] 计算机学科专业基础综合考试大纲解析（2010年版），高等教育出版社，2009

由于时间仓促，错误和疏漏之处在所难免，欢迎414的同学指出错误。若

(1) 指出10个以上错误；或

(2) 添加5个以上知识点；或

(3) 添加3个以上经典例题。

将会被载入~~史册~~感谢名单。

版本：2.1，最后更新日期：2010/12/02

官方地址：<http://www.gotothu.com> （未来清华人）

作者：戴方勤

个人博客：<http://www.yanjiuyanjiu.com/> （研究研究）

本手册遵循创作共享协议2.0，禁止一切商业用途。