# 一、名字系统及TCP/IP名字注册和名字解析

## DNS设计目标和目的

1.创建一个全球性的、可扩展的、一致性的名字空间

2.实现本地资源本地控制

3.分布式设计以避免瓶颈

4.应用的广泛性

5.支持多个底层协议

6.硬件普适性

## 域：单个对象或基于某种共性收集齐的一个对象集合

## DNS服务器支持功能

主要功能：存储和提供（通过相应来自DNS解析器的请求）名字数据

1. 与其他服务器进行交互
2. 地区管理和传输
3. 性能增强功能
4. 管理

### DNS名字服务器负载均衡

并非只能为一个DNS域名创建一条地址RR，而是可以创建多条这样的RR。这样就可以把多个IP地址与一个名字相关联，从而可以把针对一个域名的大量请求分散到多台物理IP设备上，这种做法使得DNS能够为繁忙的因特网服务器实现负载均衡

## 名字解析服务

1.标准名字解析：接受一个DNS名字作为输入并确定其对应的IP地址

2.反向名字解析：接受一个IP地址并确定与其关联的名字

3．电子邮件解析：根据报文中使用的电子邮件地址来确定应该把电子邮件报文发送到 何处。

### 名字解析器执行的功能

1．提供用户接口

2.生成并发送查询

3.处理相应

## DNS通用主文件记录格式

1. 域名
2. TTL
3. 类
4. 类型
5. 数据

### IPv6 DNS 扩展

1. 新的RR类型AAAA(IPv6地址)
2. 新的反向解析层次结构
3. 查询类型和解析过程的改动

# 二、网络文件和资源共享协议

网络文件系统（NFS）

文件传送协议（FTP）

超文本传送协议（HTTP）

## 网络文件和资源共享以及TCP/IP NFS

### 文件和资源共享协议的组件

1. 文件系统模型和体系结构
2. 资源访问方法
3. 操作集
4. 报文传送协议
5. 管理工具

### NFS专门为实现消除本地和远程文件之间差别的目标而设计

1. 研制网络文件系统（NFS）的目的是使客户机主机能够访问本地文件那样访问远程服务器上的文件。它的主要设计目标是性能、简单和跨供应商的兼容性。
2. NFS的主要组件



1. OSI模型：开放式系统互联通信参考模型
2. 通用数据交换方法(XDR):当从设备A向设备B传输关于如何访问一个文件的信息时，设备A首先将其从设备A的内部表示转换成这些数据类型的XDR表示，信息使用XDR编码跨越网络进行传输，然后设备B将其从XDR转换成自己的内部表示。
3. 使用XDR就可以在设备之间交换数据而不需要考虑每台设备使用了什么样的内部文件系统。这令NFS可以在客户机和服务器之间交换文件数据，这些客户机和服务器可能是使用完全不同的硬件和软件平台实现的。

#### 使用远程过程调用（RPC）的NFS客户机/服务器操作

##### RPC操作与运输协议用法

NFS 没有像其他大多数协议 那样采用专用报文格式。相反，客户和服务器使用RPC协议来交换文件操作请求和数据。

##### NFS中客户机和服务器的职责

NFS服务器被设计成无状态的，简言之，这意味着NFP服务器不会从一个请求到另一个请求地跟踪记录使用自己的客户机的状态。

##### 客户机和服务器高速缓存

NFS被设计成一种无状态协议，使用智能客户机和相对较哑的服务器，服务器只对请求做出相应，不维护有关那些文件正在使用的状态信息。为提高效率起见，NFS最初设计成使用UDP作为运输协议。这要求由NFS客户机负责检测丢失请求并进行重传。为能够利用TCP的可靠性和其他特色，NFSv4使用了TCP

#### NFS服务器过程和操作

#### NFS文件系统模型与装配协议

##### NFS文件系统模型

一种层次结构式的目录组织形式，目录中又包含文件和子目录。层次结构的顶端是根，根下面可以包含任意数量的文件和第一级目录，每一个目录都可以包含更多文件或其他目录，从而可以创建出任意的树状结构。

##### 装配协议

NFS版本2和版本3中没有包含用来打开或关闭某台远程服务器上的资源的过程。在这两个版本里，当NFS任务能够成功完成之前，首先必须利用专门的装配协议来装配一个文件系统，并创建一个文件句柄以便在句柄上访问中国文件，当文件系统不需要时装配协议也被用来装配一个文件系统，装配协议的实现方法与NFS自己的实现方法相似。定义了一系列使用RPC的XDR的过程。在NFSv4中装配协议已经不再需要，因为它所执行的任务已经作为NFSv4操作实现了。

# 三、主机配置与TCP/IP主机配置协议

## 主机配置概念、问题和动机

### 主机配置的目的

在TCP/IP网络中，必须为每台设备分配一个IP地址。除此之外，对一个TCP/IP网络来说，我们可能希望被告知每台主机下面的一些参数：

1. 本地网络上默认路由器的地址
2. 主机应该使用的网络掩码
3. 为主机提供特定服务的服务器的地址，例如邮件服务器或域名系统（DNS）名字服务器
4. 本地网络的最大传输单元（MTU）
5. 应该给IP数据报使用什么样的寿命（TTL，TIME TO LIVE）值

### 主机人工配置存在的问题

1. 远程配置
2. 移动设备配置
3. 哑主机配置
4. 地址共享

### 使配置过程自动化：主机配置协议

主机配置协议使管理员能够将主机设置为由主机自动确定自己的地址和其他的一些重要参数。主机配置协议之所以有用不仅因为他们节省了人工配置所需要的大量工作，而且因为它们支持远程设备、无存储设备或移动设备的自动设置。

### 主机配置协议在TCP/IP中起的作用

主机配置驻留在高层的有点：

1. 协议的操作不依赖于其上运行的硬件，从而令协议更为通用。
2. 能够在网络之间传送主机配置报文，而使用一个运行在本地网络上的低层协议则不可能做到这一点

## TCP/IP引导协议（BOOTP）

BOOTP是动态主机配置协议（DHCP）的前身。

### BOOTP概述、历史和标准

#### BOOTP：客服RARP的缺陷

反向地址解析协议（RARP）是第一个试图解决这种引导问题的尝试。

RARP能够向一台无盘设备提供其IP地址

RARP的困难在于它有太多的局限，它使用硬件广播方式运行在相当低的层次，因此他要求调整以适应不同的硬件类型。

BOOTP用来客服RARP的确定：

1. BOOTP仍基于客户机/服务器交换模式，只是作为一种高层软件协议实现，采用用户数据包协议（UDP）进行报文传输。它不像RARP那样依赖于特定的网络硬件。
2. 除了IP地址之外它还支持向客户机发送额外的配置信息，额外信息通畅可以放在一个报文中发送以提高效率。
3. 它能够处理客户机和服务器位于同一个互联网络的不同网络的情况。这使得对提供IP地址的服务器的管理能够更加集中，节省了费用同时也节省了管理时间和麻烦。

首个广泛应用于TCP/IP网络的主机配置协议是引导协议（BOOTP）。该协议专门设计用来客服PAPR的诸多缺点，支持主机配置。BOOTP计划在无存储器设备两阶段引导过程中的第一个阶段中使用，在利用BOOTP获得IP地址和其他配置参数以后，设备江勇诸如TFTP这样的协议来下载为在网络上工作所必须的软件。

#### BOOTP的修改与DHCP的研发

BOOTP的最重要的一个缺点是缺少对动态地址分配的支持。

20世纪90年代后期对动态地址分配的需求显著提高促成了动态主机配置协议（DHCP）的研发。

### BOOTP客户机/服务器报文传递和寻址

#### BOOTP报文传递和传输

BOOTP报文传递使用UDP作为自己的第四层传输协议，有以下两条原因：

1. UDP比另一个第四层运输协议即传输控制协议（TCP）要简单的多，特别适用于像BOOTP这样的简单请求/应答协议
2. 由于客户机显然不知道BOOTP服务器的地址，所以请求是在其本地网络上传播出去的。UDP支持广播，TCP则不然。

#### 丢失报文的重传

使用UDP传递BOOTP报文带来了简单性的好处，但缺点是UDP不可靠，这意味着BOOTP请求在到达服务器之前可能会丢失，或者服务器的相应可能无法返回客户机。

BOOTP使用不提供可靠特性的UDP作为传输协议，为此，BOOTP客户机必须检测自己的请求何时丢失并在必要时重传请求。

### BOOTP详细操作

#### BOOTP引导过程

1. 客户机创建请求
2. 客户机发送请求
3. 服务器收到请求对其进行处理
4. 服务器创建回答
5. 服务器发送回答
6. 客户机处理回答
7. 客户机完成引导过程

## DHCP概述与地址分配概念

DHCP包含了两种主要组件，一种是地址分配机制，另一种是允许客户机请求配置信息而服务器提供此信息的协议。DHCP以一种与BOOTP相似但又有所改进的方式实现了这两种功能。

### DHCP特色的概述

DHCP支持动态地址分配，DHCP不是使用一个将硬件地址映射到IP地址的静态表，而是使用一个IP地址池来动态分配地址。动态寻址使IP地址更有效地分配，设置可以在多个设备间共享，同时，在需要时，DHCP仍支持设备的地址的静态映射。

**要点：**DHCP是当前用于现代TCP/IP网络上的主机配置协议，它基于BOOTP，与其在许多方面相似，包括请求/回答报文交换的使用和几乎完全相同的报文格式。但是DHCP还包含了增加的功能，其中最值得注意的是动态地址分配，他允许从一个由DHCP服务器管理的共享池中给客户机分配IP地址。

### DHCP地址指派与分配机制

#### DHCP地址分配

向客户机提供IP地址是主机配置协议最基本的配置任务，DHCP标准包含了三种地址配置机制：人工、自动、和动态

#### DHCP动态分配的优点

1. 自动操作
2. 集中管理
3. 地址重用和贡献
4. 便携通用
5. 避免冲突

#### DHCP自动分配

自动分配可以看成动态分配的特例，它本质上就是客户机使用IP地址的期限（租用长度）为永远的动态分配。

**要点**：

DHCP定义了三种基本的地址分配机制。动态分配最常用的方法，它使每个客户机都从DHCP服务器租用了一个地址，并保持一段时间。服务器从共享的地址池中动态选择地址。自动分配和动态分配相似，只不过地址是永久分配而不是租用的。人工分配就和BOOTP一样，将一个地址预分配给一个特定的设备，通常只用于服务器和其他固定的重要主机。

DHCP中服务器拥有地址池中所有地址，每个客户机从服务器中租用一个地址，通常只能租用一段有限时间。

### DHCP生命周期

DHCP遵循一个租用生命周期，它通常包含以下6个阶段

1. 分配
2. 重分配
3. 正常操作
4. 更新
5. 重绑定
6. 释放

### DHCP租用地址池、范围以及地址管理

#### 地址池长度的选择

1. 客户机的数量
2. 客户机的使用情况
3. 过分配的后果

#### 租用地址范围

每个DHCP服务器维护一个地址集合，用来给客户机分配租用。它们通常是由管理员分配给服务器的连续地址块，被称为DHCP地址范围。

## DHCP配置与操作

### DHCP服务器职责

1. 地址储存与管理
2. 配置参数的存储与管理
3. 租用管理
4. 响应客户机请求
5. 管理服务

### DHCP客户机职责

1. 发起配置
2. 配置参数的管理
3. 租用管理
4. 报文重传

### DHCP租用分配、重分配和更新

#### 初始的租用分配过程

1. 客户机创建DHCPDISCOVER报文:

它自己的硬件地址，位于报文的CHAddr字段，用于标识自身。

随机的事物标识符，位于XDI字段（用于标识后继报文是同一事务的一部分）

1. 客户机发送DHCPDISCOVER报文。
2. 服务器接收并处理DHCPDISCOVER报文
3. 服务器创建DHCPOFFER报文：

分配给客户机的IP地址，位于YIAddr字段

提供的租用长度

客户机特定的配置参数

返回送给所有客户机或该客户机所在类别的通用配置参数

服务器标识符，位于DHCP服务器标识符选项

与DHCPDISCOVER报文中相同的事务标识

1. 服务器探测并/或保留提供的地址（可选）
2. 服务器发送DHCPOFFER报文
3. 客户机接收并处理DHCPOFFER报文
4. 客户机创建DHCPREQUEST报文：

被选定服务器的标识符，位于DHCP服务器标识符选项，使每个服务器都知道选中谁了。

DHCP服务器在DHCPOFFER报文中分配给客户机的IP地址，客户机将它放在请求IP地址DHCP选项中作为确认。

任何它希望的额外配置信息，位于报文的参数请求列表选项

1. 客户机发送DHCPREQUEST报文。
2. 服务器接收并处理DHCPREQUEST报文。
3. 客户机接收并处理DHCPACK或DHCPNAK报文。
4. 客户机检查该地址是否未被使用。
5. 客户机完成租用分配。

### 非DHCP地址的客户机参数配置过程

1. 客户机创建DHCPINFORM报文
2. 客户机发送DHCPINFORM报文
3. 服务器接收并处理DHCPINFORM报文
4. 服务器生成DHCPACK报文
5. 服务器发送DHCPACK报文
6. 客户机接收并处理DHCPACK报文

**要点**：不适用DHCP获取IP地址的设备仍然可以使用DHCP的其他配置能力，客户机能够广播DHCPINFOM报文。以请求任何一个可用的服务器向他发送参数，以便它可以使用该网络。DHCP服务器将请求的参数和/或默认参数放在DHCPINFORM报文的DHCP选项中返回给客户机。

## DHCP报文传递、报文类型和格式

### 报文传输

**要点**：来自BOOTP客户机的请求通常以广播方式发送，以到达任何可用的DHCP服务器。但是也有例外，比如租用更新时，请求会直接发送到已知服务器。DHCP可以以广播方式把它们的回答发送到DHCP客户机保留的特定端口号，或者以单播方式通过第2层交付。DHCP标准规定要尽可能使用第2层交付，以避免不必要的广播流量。

### 丢失报文的重传

类似于BOOTP，DHCP也使用了UDP传输，这种 不提供任何可靠性特征。当发送了请求但没有收到相应时，DHCP客户机必须察觉到，并定期重传请求。在网络环境不好的情况下，必须使用特殊的逻辑以防发送过多的请求。

## DHCP客户机/服务器的实现、特性和IPv6支持

### DHCP自配置/自动专用IP寻址（APIPA）

客户机可能无法获得租用，或者重启后不能重新得到租用，也可能无法更新已有的租用。下面几种原因可能会导致这种情况的发生：

1. DHCP服务器可能出现了故障或者离线维护
2. 客户机本地网络上的中继代理出现了故障
3. 其他硬件故障或者电源问题导致无法通信
4. 网络耗尽了可分配的地址

### APIPA操作

任何DHCP租用过程失败时，APIPA从专用地址块中随机选择一个地址，而不是直接停下来发送一条差错报文。（它的地址范围是 B 类地址块169.254.0.0~169.254.255.255）

### APIPA的限制

具有专用IP地址存在的所有限制，包括不能再英特网上使用这些地址。而且APIPA不能提供客户机需要从DHCP服务器获得的其他配置参数，最后APIPA不能喝代理APR一起工作，因为代理对所有的专用地址都会相应，这样他们看起来似乎都被占用了。

### DHCP安全性担忧

1. 未授权的DHCP服务器
2. 未授权的DHCP客户机

### IPv6中两种自配置方法

1. 无状态自配置：一种允许主机配置自身而不需要任何其他设备帮助的方法
2. 状态自配置：一种由服务器给客户机提供配置信息的技术

# 四、TCP/IP网络管理框架和协议

## TCP/IP英特网标准管理框架概述

简单网络管理协议（Simple Network Management Protocol,SNMP）

### SNMP的设计目标

简单网络管理协议（SNMP）定义了一套允许网络管理员远端监视和操作TCP/IP网络设备的技术。术语SNMP不仅指一个具体的通信协议（有时成为SNMP协议），而且指英特网管理的整个框架（SNMP框架）

### SNMP设备类型

SNMP包含的总体思想是允许网络管理需要的信息通过TCP/IP进行交换。定义了两种不同基本类型的硬件设备：

**被管节点**：网络中那些装配了软件以允许通过SNMP来管理他们的常规节点。通常来说，这些是常规的TCP/IP设备。他们有时也被称为被管节点。

**网络管理站**：（NMS）运行特殊的软件来管理上面提到的常规被管节点的指定的网络设备。网络中必须存在一个或多个NMS，因为这些是真正运行的SNMP的设备

### SNMP实体

1. 被管理节点实体：

SNMP代理：是指一个实现SNMP协议，运行被管节点为NMS提供信息并从NMS哪里获得指令的软件程序

SNMP管理信息库（MIB）：MIB定义了存储的有关节点为NMS提供信息并从NMS那里获得指令的软件程序。

1. 网络管理站实体

一个NMS的SNMP实体由以下部分组成：

SNMP管理者：实现SNMP协议的软件程序，他允许NMS从被管节点收集信息，并向它们发送指令

SNMP应用程序：允许网络管理员使用SNMP来管理网络的一个或多个管理程序。

### TCP/IP因特网标准管理框架的体系结构和协议组件

#### SNMP框架组件

1. 管理信息结构
2. 管理信息库（MIB）
3. 简单网络管理协议（SNMP）
4. 安全性和管理

## TCP/IP SMI和MIB

### TCP/IP SMI和MIB概述

#### SNMP的面向信息设计

不像大多数面向命令的协议，SNMP是面向信息的。SNMP操作。SNMP操作是通过使用被管设备维护的称为变量的对象来实现的。网络管理站通过读变量来检测设备的状态。通过改变（写）变量来控制设备的操作，而不是发出命令。

#### MIB和MIB对象

被管设备中的管理数据变量是在一个称为管理信息库（MIB）的逻辑集合中加以维护的。MIB中的对象通常称为MIB对象，它们通常被收集到一个称为MIB模块的集合中。

#### 定义MIB对象：SMI

管理信息结构（SMI）标准是负责定义MIB对象构造、描述和组织的规则。通过确保对所有管理信息使用通用数据描述，SMI允许不相似的设备通信。

### TCP/IP SNMP的概念的操作

SNMP框架的第三个主要部分是实际的简单网络管理协议（SNMP），它主要负责在设备之间传送管理信息。

SNMP报文由一组包装数据元素（称为协议数据单元或PDU）的字段组成。在有些情况下，尽管术语报文和PDU在技术上是不一样的，它们可以互换使用，SNMP的PUD根据它们的功能而被分为不同类别。

### SNMP协议操作

#### 使用GetRequest和（Get）Response报文的基本请求/响应信息轮询

1. SNMP管理者创建GetRequest-PDU
2. SNMP管理者发送GetRequest-PDU
3. SNMP代理接收并处理GetRequest-PDU
4. SNMP代理创建Response-PDU
5. SNMP代理发送Response-PDU
6. SNMP管理者处理Response-PDU

在SNMP，通信最基本类型是信息轮询，它允许NMS使用一个简单的请求/回答报文交换从被管节点读一个或多个MIB对象。

# 五、/IP应用层寻址和应用程序分类

统一资源标识符（URI）：包括统一资源定位符（URL）和统一资源名（URN）

## TCP/IP应用层寻址：URI、URL和URN

### URL概述和标准

#### URI的类别：URL和URN

统一资源定位符（URL）：URL是通过将协议或访问机制与具体的资源位置想结合来引用资源的一种URI。URL由用于访问资源的协议名开始，包含足够的信息以指出如何获得该资源。

统一资源名（URN）：URN也是一种URI，它提供了一种唯一命名资源而不需要指定访问协议或机制的方法，它不要指定一个特定的位置。

URL和URN的区别在于前者更具体，是面向如何访问资源的，而后者更抽象，被更多地用来标识资源是什么而不是如何获得它。

任何互联网都需要某种机制以允许标识和访问资源（如文件、目录和程序）。在TCP/IP中，URI用于这种“应用层寻址”。URI的两种类型是URL和URN，URL指定如何使用访问方法和位置相结合来访问一个对象，URN通过名字来标识一个对象，但不知处如何访问它。

### URL的通用语法

URL是最广泛使用的URI类型。在URL最基本的格式中包括两种元素，一种是定义访问资源的协议和其他机制的纲要，另一种是包含标识具体资源并指出如何使用其信息的纲要特定部分。一些纲要为它们的纲要特定部分使用通用语法，其他一些纲要使用它们纲要特有的语法。

语法元素具体解释：

1. <纲要> 🡪http
2. <用户和密码> 🡪joeb xx123
3. <主机> 🡪www.mysite.org
4. <端口> 🡪8080
5. <url路径> 🡪cgi-bin/pix.php
6. <参数>
7. <查询> 🡪wedding03
8. <分片>：标识URL用户感兴趣的资源中的一个特定位置 🡪Reception07

<http://joeb:xx123@www.mysite.org:8080/cgi-bin/pix.php?wedding03#Reception07>

HTTP URL 正式支持<用户>和<密码>，但是它们很少使用，同时端口号经常被省略，告诉客户端软件对这个纲要只要用默认的端口号就可以了。

#### URL 分片

<分片>元素不是URL的正式部分，原因是它只标识资源的一部分，它不是标识资源本身所需信息的一部分。

### URL纲要和纲要特定的语法

<纲要>：//<用户>:<密码>@<主机>:<端口>/<url路径>;<参数>?<查询>#<分片>

#### 万维网/超文本传送协议语法（http）

http：//<用户>:<密码>@<主机>:<端口>/<url路径>;<参数>?<查询>#<书签>

#### 文本传送协议语法（ftp）

ftp：//<用户>:<密码>@<主机>:<端口>/<url路径>;<类型>=<类型码>

#### 电子邮件语法（mailto）

[mailto:<电子邮件地址](mailto:%3c电子邮件地址)>

#### Gopher协议语法（gopher）

[gopher://<主机>:<端口>/<gopher](gopher://%3c主机%3e:%3c端口%3e/%3cgopher)路径>

#### 网络新闻/Usenet语法（news）

[news://<](news://%3c)新闻组名>

[news://<](news://%3c)报文id>

#### 网络新闻传输协议语法（nntp）

[nntp://<主机>:<端口>/<新闻组名>/<](nntp://%3c主机%3e:%3c端口%3e/%3c新闻组名%3e/%3c)文章号>

#### 远程登录语法

[telnet://<用户>:<密码>@<主机>:<端口](telnet://%3c用户%3e:%3c密码%3e@%3c主机%3e:%3c端口)>

实际使用时用户名和密码通常省略

#### 本地文件语法（file）

file://<主机>:<url路径>

本地计算机上的文件

file:///<url路径>

#### 特殊的语法规则

在微软操作系统中冒号用于盘符说明，这里用垂直管道字符（|）代替它

[file:///c|/Windows/SYSTEM32/DRIVERS/ETC/HOSTS](file:///c:/Windows/SYSTEM32/DRIVERS/ETC/HOSTS)

然而请注意，某些浏览器实际上的确允许在驱动器规约中使用冒号

### URL的相对语法和基础URL

#### 相对URL的解释规则

1. 从文档中获得基础URL
2. 从封装实体中获得基础URL
3. 从相对URL中获得基础URL

常规的URL是绝对的，意味着它们包含指定如何访问一个资源所需要的所有信息。在需要访问的许多资源几乎在同一个位置或以某种方式相关的情况下，完全指定URL效率很低。这是可以使用相对URL，它指定如何访问一个相对其他资源的资源。相对URL只能在基础URL的上下文中来解释，而这些基础URL提供了相对引用所缺失的任何信息。

#### URL编码

##### URL 编码 - 从 %00 到 %8f

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ASCII Value** | **URL-encode** | **ASCII Value** | **URL-encode** | **ASCII Value** | **URL-encode** |
| æ | %00 | 0 | %30 | ` | %60 |
|  | %01 | 1 | %31 | a | %61 |
|  | %02 | 2 | %32 | b | %62 |
|  | %03 | 3 | %33 | c | %63 |
|  | %04 | 4 | %34 | d | %64 |
|  | %05 | 5 | %35 | e | %65 |
|  | %06 | 6 | %36 | f | %66 |
|  | %07 | 7 | %37 | g | %67 |
| backspace | %08 | 8 | %38 | h | %68 |
| tab | %09 | 9 | %39 | i | %69 |
| linefeed | %0a | : | %3a | j | %6a |
|  | %0b | ; | %3b | k | %6b |
|  | %0c | < | %3c | l | %6c |
| c return | %0d | = | %3d | m | %6d |
|  | %0e | > | %3e | n | %6e |
|  | %0f | ? | %3f | o | %6f |
|  | %10 | @ | %40 | p | %70 |
|  | %11 | A | %41 | q | %71 |
|  | %12 | B | %42 | r | %72 |
|  | %13 | C | %43 | s | %73 |
|  | %14 | D | %44 | t | %74 |
|  | %15 | E | %45 | u | %75 |
|  | %16 | F | %46 | v | %76 |
|  | %17 | G | %47 | w | %77 |
|  | %18 | H | %48 | x | %78 |
|  | %19 | I | %49 | y | %79 |
|  | %1a | J | %4a | z | %7a |
|  | %1b | K | %4b | { | %7b |
|  | %1c | L | %4c | | | %7c |
|  | %1d | M | %4d | } | %7d |
|  | %1e | N | %4e | ~ | %7e |
|  | %1f | O | %4f |  | %7f |
| space | %20 | P | %50 | € | %80 |
| ! | %21 | Q | %51 |  | %81 |
| " | %22 | R | %52 | ‚ | %82 |
| # | %23 | S | %53 | ƒ | %83 |
| $ | %24 | T | %54 | „ | %84 |
| % | %25 | U | %55 | … | %85 |
| & | %26 | V | %56 | † | %86 |
| ' | %27 | W | %57 | ‡ | %87 |
| ( | %28 | X | %58 | ˆ | %88 |
| ) | %29 | Y | %59 | ‰ | %89 |
| \* | %2a | Z | %5a | Š | %8a |
| + | %2b | [ | %5b | ‹ | %8b |
| , | %2c | \ | %5c | Œ | %8c |
| - | %2d | ] | %5d |  | %8d |
| . | %2e | ^ | %5e | Ž | %8e |
| / | %2f | \_ | %5f |  | %8f |

##### URL 编码 - 从 %90 到 %ff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ASCII Value** | **URL-encode** | **ASCII Value** | **URL-encode** | **ASCII Value** | **URL-encode** |
|  | %90 | À | %c0 | ð | %f0 |
| ‘ | %91 | Á | %c1 | ñ | %f1 |
| ’ | %92 | Â | %c2 | ò | %f2 |
| “ | %93 | Ã | %c3 | ó | %f3 |
| ” | %94 | Ä | %c4 | ô | %f4 |
| • | %95 | Å | %c5 | õ | %f5 |
| – | %96 | Æ | %c6 | ö | %f6 |
| — | %97 | Ç | %c7 | ÷ | %f7 |
| ˜ | %98 | È | %c8 | ø | %f8 |
| ™ | %99 | É | %c9 | ù | %f9 |
| š | %9a | Ê | %ca | ú | %fa |
| › | %9b | Ë | %cb | û | %fb |
| œ | %9c | Ì | %cc | ü | %fc |
|  | %9d | Í | %cd | ý | %fd |
| ž | %9e | Î | %ce | þ | %fe |
| Ÿ | %9f | Ï | %cf | ÿ | %ff |
|  | %a0 | Ð | %d0 |
| ¡ | %a1 | Ñ | %d1 |
| ¢ | %a2 | Ò | %d2 |
| £ | %a3 | Ó | %d3 |
|  | %a4 | Ô | %d4 |
| ¥ | %a5 | Õ | %d5 |
| | | %a6 | Ö | %d6 |
| § | %a7 |  | %d7 |
| ¨ | %a8 | Ø | %d8 |
| © | %a9 | Ù | %d9 |
| ª | %aa | Ú | %da |
| « | %ab | Û | %db |
| ¬ | %ac | Ü | %dc |
| ¯ | %ad | Ý | %dd |
| ® | %ae | Þ | %de |
| ¯ | %af | ß | %df |
| ° | %b0 | à | %e0 |
| ± | %b1 | á | %e1 |
| ² | %b2 | â | %e2 |
| ³ | %b3 | ã | %e3 |
| ´ | %b4 | ä | %e4 |
| µ | %b5 | å | %e5 |
| ¶ | %b6 | æ | %e6 |
| · | %b7 | ç | %e7 |
| ¸ | %b8 | è | %e8 |
| ¹ | %b9 | é | %e9 |
| º | %ba | ê | %ea |
| » | %bb | ë | %eb |
| ¼ | %bc | ì | %ec |
| ½ | %bd | í | %ed |
| ¾ | %be | î | %ee |
| ¿ | %bf | ï | %ef |

# 六、TCP/IP通用文件传送协议

## 文件传送协议（FTP）

### FTP操作概述

FTP基于客户机/服务器模型，一台用户机器上的FTP客户机创建一个FTP服务器的连接，吧文件发送到服务器上，并从服务器上获取文件

FTP在运输层采用可靠的TCP。它使用一个鉴别系统以确保只有授权的客户机才允许访问服务器。同时,FTP中存在一个称为匿名FTP的特征，它允许那些想设立通用信息服务器的组织给任何想获得文件的人提供文件。

FTP用户和协议之间的接口是以一组交互的用户命令的形式来提供的。

#### FTP的控制连接和数据连接

不像大多数协议那样，FTP不是使用单个TCP连接。当建立一个会话时，使用TCP创建一个永久的控制连接以传递命令和回答。当发送文件和其他数据时，它们在独立的TCP数据连接上传递，这个连接根据需要创建和拆除。

#### 服务器FTP进程组件

1. 服务器协议解释器（服务器PI）
2. 服务器数据传输进程（服务器DTP）

#### 用户FTP进程组件

1. 用户协议解释器（用户PI）
2. 用户数据传输进程（用户DTP）
3. 用户接口

#### 第三方文件传输（FTP代理）

服务器之间传输，没有使用用户DTP

### FTP通用数据通信和传输模式

1. 流模式：最常用，数据作为一个连续的字节序列发送
2. 块模式：数据被格式化为包含首部的块
3. 压缩模式：使用了游程编码来压缩字节

#### FTP数据类型

1. ASCII
2. EBCDIC 🡪 IBM EBCDIC字符集的文件
3. 图像
4. 本地 以逻辑字节方式存储数据的文件

#### ASCII数据类型定界问题

使用FTP在PC机和UNIX系统之间移动文件时，使用错误的数据类型集是文件损坏的主要原因（LF ，CR+LF 转化）

#### FTP命令组和命令

1. 访问控制命令
2. 传输参数命令
3. FTP服务命令

## 普通文件传送协议（TFTP）

#### FTP和TFTP的区别

TFTP是FTP大大简化的版本，只允许执行基本操作，目的是为了保持实现容易和程序长度小

FTP和TFTP至少在以下四个重要方面有很大的差别

1. 传输：FT🡪TCP TFTP🡪UDP
2. 有限的命令集：FTP允许文件发送、接收、重命名、和删除 TFTP只允许文件发送和接收
3. 有限的数据表示方法：TFTP只允许ASCII或二进制文件传输
4. 缺乏鉴别：UDP不适用登录机制或其他鉴别方法。这意味着TFTP服务器的操作者必须严格限制可以用来访问的文件

#### TFTP通用操作

1. 初始连接
2. 数据传输
3. 连接中止

#### 客户机/服务器报文传递的锁步方式

允许发送方一个报文接着一个报文地连续发送 。即一个设备发送数据报文并等待确认，另一个设备发送确认并等待数据报文。

# 七、TCP/IP电子邮件系统：概念和协议

简单邮件传送协议（Simple Mail Transfer Protocol, SMTP）

邮局协议（Post Office Protocol verion 3 ,POP3）

互联网报文访问协议（Internet Message Access Protocol ,IMAP）

## TCP/IP电子邮件系统的概述和概念

### TCP/IP电子邮件系统概述和历史

最重要的TCP/IP应用程序之一是现实世界邮政交付系统的网络互联等价物。🡪email TCP/IP电子邮件不是一个简单的应用程序，而是一个包含了几种协议、软件元素和组件的完整系统。

### TCP/IP电子邮件报文的通信模型

1. 发送方的客户机主机
2. 发送方的本地SMTP服务器
3. 接收方的本地STMP服务器
4. 接收方的客户机主机

#### 协议在电子邮件通信中的作用

电子邮件系统的一个关键需求是，报文的发送方和接收方在邮件发送时不需要都在系统上。因此TCP/IP使用了一个允许发送方和接收方分离的通信模型。发送方的客户机设备缓存邮件，并在准备传输时把它移动到发送方的本地SMTP服务器，然后通过SMTP把电子邮件传输到接收方的SMTP服务器。电子邮件可以在接收方服务器上保留一段不定长的时间。当接收方准备读取邮件时，他使用一个或多个邮件访问协议和方法来获取邮件，其中最流行的两个协议是POP和IMAP

## 电子邮件的地址和寻址

### TCP/IP历史的和特殊的电子邮件寻址

1. FidoNet寻址
2. UUCP风格的寻址
3. 网关寻址

## TCP/IP电子邮件报文格式和报文处理：RFC 822和MIME

电子邮件的标准格式（RFC 822）

多用途的互联网邮件扩充协议（Multipurpose Internet Mail Extensions.MIME）

### RFC报文处理

1. 撰写
2. 发送方客户机的处理
3. SMTP服务器的处理
4. 接收方客户机的处理
5. 接收方访问

### MIME概述

#### MIME的目的

希望能够传输图表，非ASCII的文本文档、二进制程序文件和多媒体信息。

#### MIME的能力

MIME允许常规的RFC 822电子邮件报文携带下列内容：

1. 非文本信息
2. 使用非ASCII字符集的文本报文

#### 对非ASCII邮件报文首部的MIME扩展

除了对电子邮件报文体中的各种数据进行编码之外，MIME还提供了一个允许非ASCII信息放置到电子邮件部首的特征。这是通过使用引用可打印的编码或base64编码对数据进行编码，然后为指定编码类型和字符集的首部值使用一个特殊的格式来完成的，

## TCP/IP 电子邮件交付协议：SMTP

### SMTP的概述、历史和标准

TCP/IP电子邮件系统中最重要的组成部分是SMTP。SMTP起源于早期的MTP，是用于在TCP/IP系统和用户之间交付的机制。电子邮件接收方最后取回邮件的步骤是电子邮件系统中唯一没有使用SMTP的部分。

#### SMTP通信和报文传送的方法

1. 使用中继的早期电子邮件通信
2. 使用DNS和直接交付的现代电子邮件通信

#### 客户机/服务器和发送方/接收方

SMTP服务器既发送电子邮件又接收电子邮件。发送邮件的设备为该事物充当客户机。接收邮件的设备充当服务器。

### SMTP连接、会话创建和中止

1. 连接创建和中止
2. 连接创建和问候交换
3. 使用SMTP扩展来创建连接
4. 连接终止

一个SMTP会话包括三个基本阶段。首先通过创建TCP连接和使用HELO命令在SMTP发送方和接收方之间交换身份信息来创建这个会话。一旦会话创建了，可以进行邮件事物。当SMTP发送方完成这个会话时，它使用QUIT命令来中止。如果支持SMTP扩展，SMTP发送方使用EHLO命令（扩展的）来替代HELO命令，SMTP接收方用一个它允许SMTP发送方使用的扩展列表来回答。

### SMTP的特色、能力和扩展

#### SMTP的特色和能力

1. 邮件中继
2. 邮件转发
3. 邮件网关
4. 寻址调试 ：VRFY（验证，检查电子邮件地址有效性
5. 邮件列表扩展
6. 转向：可以用于SMTP服务器A给服务器B发送电子邮件，然后使服务器B在同一个会话中发送服务器A排队等待的电子邮件

#### SMTP扩展

扩展关键字 | 扩展

8BITMIMT |支持8比特MIME

AUTH |授权

DSN |交付状态通知

ENHANCEDSTATUSCODES |增强的状态码

PIPELINEG |命令流水线操作

SIZE |报文长度声明

#### SMTP命令

命令码 | 命令

HELO | hello

EHLO | hello

MAIL |发起邮件事务

RCPT |接收方

DATA |邮件报文数据

RSET |复位

VRFY | 验证

EXPN |扩充

HELP |帮助

NOOP |没有操作

QUIT |退出

#### SMTP 多行文本回答

每次SMTP发送方发出一个命令时，它从SMTP接收方收到一个回答。SMTP回答类似于FTP回答，都使用三个数字的回答码和一个描述性的文本行。也定义了一个特殊的增强状态码SMTP扩展，当启用时，它使得SMTP接收方在处理完命令后返回更详细的结果信息。

## 访问和取回TCP/IP电子邮件的协议和方法

### TCP/IP电子邮件的邮箱访问模型、方法和协议概述

#### 电子邮件的访问和取回模式

1. 在线访问模型
2. 离线访问模型
3. 分离访问模型：在线和离线的混合

### TCP/IP邮局协议（POP/POP3）

POP是当前最流行的TCP/IP电子邮件访问和取回协议。它实现了离线访问模型，允许用户从他们的SMTP服务器取回邮件，并在他们本地客户机所在计算机上使用它，它被特意设计为一个简单的协议，只有少量的命令。POP当前版本是版本3，被缩写为POP3

#### POP3的通用操作

POP3使用TCP来通信，确保了命令、相应和报文数据的可靠传输。POP3 之定义了两个基础相应

1．+OK 肯定相应，当命令或行为成功时发送。

2．-ERR 否定响应，发送它来指出出现了差错。

#### POP3会话

POP3是一个客户机/服务器协议，对它的描述使用了简单的线性状态序列，POP3会话时以POP3客户机产生一个到POP3服务器的TCP连接开始的，这时会话处于授权状态，在成功的鉴别之后梦回环转移到事物状态，在这个状态客户机可以执行邮件访问事务。当完成时，客户机结束这个会话并自动进入更新状态，在这个状态执行清理功能，POP3会话就结束了。

### TCP/IP因特网报文访问协议（IMAP/IMAP4）

#### IMAP特色

IMAP主要用于在线和分离访问模型，它允许用户从多个不同的设备来访问邮件、管理多个邮箱、只选择某些报文下载等更多的功能。

#### IMAP的会话状态

1. 为鉴别状态
2. 鉴别状态
3. 选择状态
4. 退出状态

### TCP/IP直接服务器电子邮件访问

1. 直接使用STMP服务器
2. 文件共享访问
3. 拨号远程服务器访问
4. Telnet远程服务器访问

# 八、TCP/IP万维网和HTTP

超文本传送协议（Hypertext Transfer Protocol HTTP）

## 万维网和超文本的概述及概念

### 万维网的系统概念和组件

#### web的主要功能组件

1. 超文本标记语言（HTML）
2. 超文本传送协议（HTTP）
3. 统一资源标识符（URI）

#### web服务器和web浏览器

万维网是一个包含大量相关组件的完整系统，其中有3个是最重要的。HTML描述了如何构成超文本文档，它允许表示文档之间链接。HTTP是在web上移动超文本和其他文档的应用层协议。URI机制提供了一种在web上，甚至在整个因特网上标识资源的一致性手段。

### 万维网的媒体和超文本标记语言

#### HTML的元素和标记

<html>

<head>

(首部元素放在这里)

</head>

<body>

（体元素放在这里）

</body>

</html>

## HTTP的通用操作和连接

HTTP/1.1 （当前版本）

几个重要改进：

1. 支持多个主机名
2. 持久连接
3. 部分资源选择
4. 更好地支持高速缓存和代理
5. 内容协商
6. 更好的安全性

### HTTP的操作模型和客户机/服务器通信

#### 基本的HTTP客户机/服务器通信

客户机请求

服务器响应

HTTP是一个面向客户机/服务器的请求/响应协议。基本通信是由HTTP客户机给HTTP服务器发送一个HTTP请求报文和HTTP服务器给客户机返回一个HTTP响应报文组成

#### 中间设备和HTTP请求/响应链

1. 客户机请求
2. 中间设备请求
3. 服务器响应
4. 中间设备响应

### HTTP的短时间与持久连接及流水线操作

#### 持久连接

在每组请求/响应之后，TCP连接默认保持打开，以便下一个请求和响应能够立刻交换。只有当客户机对它需要的所有文档完成请求时，这个会话才会关闭。

#### 流水线操作

，对所有请求在同一个TCP会话中发送。

## HTTP的报文、方法和状态码

### HTTP的通用报文格式

<起始行>

<报文首部>

<空行>

[<报文体>]

[<报文尾>]

所有的HTTP报文都符合一个称为通用报文格式的结构。这个格式基于RFC 822和MIME电子邮件报文标准，然而HTTP没有严格遵循这些格式。每个HTTP报文都是以起始行开始的。然后包含大量报文首部，接着是一个空行和可选的报文体。报文体可以包含一个称为实体的资源，例如在客户机和服务器之间传递的文件。

#### HTTP方法

##### 通用方法

1. GET：请求服务器获取HTTP请求行中的URL所指定的资源，并吧它在响应中发回给客户机。

如果使用了某些首部，例如If-Modified-Since或If-Match，Get请求的意义可能会产生变化，记住这一点是很重要的。这些首部告诉服务器只有某些条件的情况下才发送资源。称为条件GET

1. HEAD，HEAD方法和GET方法一样，但是它告诉服务器不要发送实际的报文体。客户机在决定它是否想在服务器发送整个文件之前，经常使用整个方法来检测文件是否存在以及它的状态或长度。
2. POST： POST方法允许客户机给服务器发送一个包含任意数据的实体来处理。它通常用于使客户机能够给服务器上的程序提交信息(例如交互的HTML表单)，然后服务器根据该输入采取行动发送一个响应。

##### 其他方法

1. OPTIONS：允许客户机请求服务器给它发送关于可用通信选项的信息。
2. PUT：请求服务器把请求体中包含的实体存储到请求行指定的URL
3. DELETE：要求删除指定的资源
4. TRACE：允许客户机为了诊断目的而收回它给服务器发送的请求副本。

### HTTP的状态码和原因短语

## HTTP报文首部

### HTTP通用首部

通用首部主要用于传递关于报文自身的信息，而不是它所携带的内容。。它们提供通用信息，并控制报文如何处理和操作。

#### Cache-Control首部

Cache-Control （高速缓冲控制）

高速缓冲控制首部指定一些用于管理如何对HTTP请求或响应执行高速缓存的指示。它们取代设备执行的任何默认的高速缓存行为。

#### Warning首部

当需要告警首部来提供关于报文状态的附加信息时，使用告警首部。定义的许多告警首部类型与高速缓存相关

#### 其他HTTP通用首部

1. Connection(连接)：包含只属于这个特定连接的命令，而且不许代理保留它和用于其他连接。
2. Date（日期）：指出报文产生的日期和时间
3. Pragma（附注）：用于使实现特定的指示能够应用到请求/响应链中的所有设备。
4. Trailer（尾部）：当使用分块传送时，某些首部可以防止在发送数据的后面作为尾部。
5. Transfer-Encoding（传送编码）：指出报文体使用了什么编码，以确保它能够在设备之间正确的传送。
6. Upgrade（更新）：允许客户机设备指定它支持其他哪些协议。
7. Via（通过）：中间设备包含它来给接收方只是在传输请求或相应中使用哪个网关、代理和/隧道。

### HTTP请求首部

1. Accept（接受）：允许客户机告诉
2. Accept-charset（接受字符集）：指定客户机想在相应中接收哪种字符集，而不是哪种媒体类型。
3. Accept-Encoding（接受编码）：指定客户机想要接受哪一种内容编码
4. Accept-Language（接受语言）：提供一个语言标记列表，指出客户机支持或期待服务器在响应中使用哪种语言。
5. Authorization（授权）：客户机使用它给服务器呈现鉴别信息（称为证书），以允许鉴别客户机。
6. Expect（期待）：指出客户机期待服务器执行的某种行为类型。通常服务器接受这些指出的参数。如果不接受没他将范围一个417（期待失败）响应。
7. From（来自）：包含产生请求的用户的电子邮件地址，应该只用于传递信息的目的，不能用于任何类型的访问权判断或鉴别。
8. Host（主机）：允许在一个特定IP主机上的同一个Web服务器服务于多个域。
9. if-Match（如果匹配）：通过指定对应于客户机希望访问的特定实体的实体标记来产生一个有条件的方法。
10. if-Modified-Since（如果自……以来被修改）：产生一个由条件的方法，告诉服务器只有自从在这个首部中指定的时间以来被请求的实体已经被修改过了才返回它。
11. if-None-Match（如果不匹配）
12. if-Range（如果范围）：在一个请求中允许客户机既有效地检测实体是否已经改变，又请求只发送实体的一部分。
13. if-Unmodified-Since（如果自……以来未被修改）
14. Max-Forwards（最多转发）：指定请求可以转发给请求链中下一个设备的次数限制。
15. Proxy-Authorization（代理授权）：用于为了鉴别而给代理服务器（不是端服务器）显示证书。
16. Range（范围）：通过指定实体中要获取的字节范围，允许客户机请求服务器只给它发送实体的一部分。
17. Referrer（引用）（正确拼法referrer，但是在纠正之前很多软件已经用了这个拼写方法，以至于IETF就选择了在HTTP/1.1中不纠正这个拼法）：告诉服务器资源的URL，当前请求的URL就是从此处获得的。
18. TE（传送编码）给服务器提供关于客户机希望如何为服务器发送的实体处理传送编码的信息。
19. User-Agent(用户代理)：提供关于客户机软件的信息。

### HTTP响应首部

1. Accept-Ranges（接受范围）：告诉客户机服务器是否接受使用Range请求首部的部分内容请求，如果接受，是什么类型的。
2. Age（寿命）：告诉客户机资源的大致期限
3. ETag（实体标记）：为包含在响应中的实体指定实体标记。
4. Location（位置）：指出一个新的URL，服务器指示客户机使用它代替客户机最初请求的URL。
5. Proxy-Authenticate（代理鉴别）：指出代理需要客户机如何执行鉴别
6. Retry-After（重试时间）：告诉客户机它应该什么时候再尝试请求
7. Server（服务器）：User-Agent请求首部的服务器版本。它标识产生这个响应的服务器软件的类型和版本
8. Vary（变更）：指定哪个请求首部字段能够完全决定是否允许告诉缓冲使用这个响应来回答对同一资源的后序请求而不需要重新生效。
9. WWW-Authenticate（WWW鉴别）：指出服务器希望客户机如何鉴别

### HTTP实体首部

1. Allow（允许）：列出一个特定资源支持的所有方法
2. Content-Encoding（内容编码）：描述可以用于对实体进行编码的任何可选方法。
3. Content-Language（内容语言）：为使用实体而制定自然（人类）语言
4. Content-Length（内容长度）：以八位组来指示实体的长度
5. Content-Location（内容位置）：以绝对或相对URL的方式指定实体的资源位置。
6. Content-MD5（内容MD5）：为实体包含一个报文摘要5（MD5）的摘要
7. Content-Range（内容范围）：当报文包含一个只有完整资源的一部分的实体时发送这个首部，指出该报文包含整个文件的哪部分以及资源的整体长度。
8. Content-Type（内容类型）：用于指定实体的媒体类型和子类型
9. Expires（截止期）：指定日期和时间
10. Last-modified（最后被修改）：指出服务器认为实体最后一次改变的日期和时间

## HTTP实体、传送、编码方法和内容管理

#### HTTP的两级编码方案

1. 内容编码
2. 传送编码

#### 内容和传送编码的使用

HTTP为数据传送支持两级编码。第一级是内容编码，它用于在某些环境下转换HTTP报文中携带的实体。第二级是传送编码，它用于对整个HTTP报文进行编码来确保报文的安全传送。当对实体进行压缩来提高通信效率时，通常使用内容编码。传送编码主要用于处理标识

## HTTP的特性、能力和问题

### HTTP高速缓存特性和问题

#### 高速缓存的优点

#### 高速缓存的位置

1. 在Web客户机上的高速缓存
2. 在中间设备上的高速缓存
3. 在Web服务器上的高速缓存

### 高速缓存的控制

#### 高速缓存的一些重要问题

1. 高速缓存的老化和无状态
2. 高速缓存的过期和确认
3. 给用户传递高速缓存的状态
4. 首部的高速缓存
5. 资源更新的影响
6. 隐秘性担忧

### HTTP代理服务器和代理

#### 代理的优点

1. 安全性：可以设立代理来检查输出的请求和输入的相应。
2. 高速缓存：在中间设备上设立一个共享高速缓存，使得一个客户机请求的资源对另一个客户机也是可用的。
3. 性能：在某些环境下使用代理服务器可以大大提高性能，尤其减少了通信时延。

#### 一些重要的代理问题

1. 能力不一致
2. 鉴别需求
3. 与高速缓存的交互
4. 编码
5. 代理跟踪的处理

### HTTP的安全性和隐秘性

#### HTTP的鉴别方法

1. 基本鉴别：常规的用户名/密码鉴别类型，基本鉴别被认为没有很强的安全性，应为它不加密地发送证书，这意味着他们可能被截获。
2. 摘要鉴别：使用了更复杂的技术，包括加密，以防止怀有恶意的人探听证书信息。

### 使用cookie进行HTTP状态管理

HTTP完全是无状态的，每次服务器从客户机收到一个请求时，它处理这个请求，并发送一个响应，然后就遗忘这个请求。

当服务器实现一个需要通过一组事务来维持状态的功能时，它给Web客户机发送少量称为cookie的数据，cookie包含与这个特殊Web应用程序有关的重要信息，例如顾客的名字、购物车中的无盘，或者用户名和密码。客户机在cookie中存储这个信息，当对设置cookie的服务器发出后续请求时使用它，服务器可以根据新请求中的信息更新cookie并把它发回客户机。

#### cookie存在的问题

1. 敏感信息的传输：cookie的报文可能被拦截，敏感信息存储在cookie的话会被利用
2. cookie不受欢迎的用法：服务器可以为了追踪用户访问的Web站点而设置cookie，有些人认为中国侵犯了他们的隐私。
3. 第三方或无意识的cookie

## USENET（网络新闻）和TCP/IP NNTP

## Gopher协议

# 十、交互式和管理型的使用程序及协议

## TCP/IP交互的和远程应用协议

## TCP/IP管理和故障定位并解决使用程序及协议