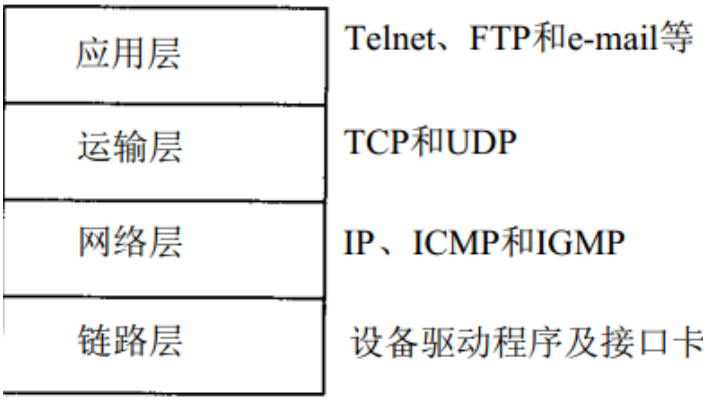
第 1 章概述

1. TCP/IP 通常被认为是一个四层协议系统，每一层负责不同的功能



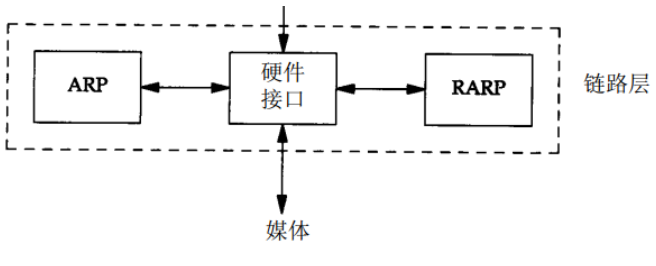
2. 在 TCP/IP 协议族中，网络层 IP 提供的是一种不可靠的服务，TCP 在不可靠的 IP 层上提供了一个可靠的运输层。
3. 连接网络的另一个途径是使用网桥。网桥是在链路层上对网络进行互连，而路由器则是在网络层上对网络进行互连。网桥使得多个局域网（LAN）组合在一起，这样对上层来说就好像是一个局域网。
4. IP 是网络层上的主要协议，同时被 TCP 和 UDP 使用。TCP 和 UDP 的每组数据都通过端系统和每个中间路由器中的 IP 层在互联网中进行传输。
5. 五类不同的 IP 地址如下：

类型	范　围
A	0.0.0.0 到 127.255.255.255
B	128.0.0.0 到 191.255.255.255
C	192.0.0.0 到 223.255.255.255
D	224.0.0.0 到 239.255.255.255
E	240.0.0.0 到 247.255.255.255

6. 在 TCP/IP 领域中，域名系统（DNS）是一个分布的数据库，由它来提供 IP 地址和主机名之间的映射信息。
7. 当目的主机收到一个以太网数据帧时，数据就开始从协议栈中由底向上升，同时去掉各层协议加上的报文首部。每层协议盒都要去检查报文首部中的协议标识，以确定接收数据的上层协议。这个过程称作分用。
8. TCP 和 UDP 采用 16 bit 的端口号来识别应用程序。知名端口号介于 1～255 之间。256～1023 之间的端口号通常都是由 Unix 系统占用，大多数 TCP /IP 实现给临时端口分配 1024～5000 之间的端口号。

第 2 章 链路层

1. 在 TCP/IP 协议族中，链路层主要有三个目的：（1）为 IP 模块发送和接收 IP 数据报；（2）为 ARP 模块发送 ARP 请求和接收 ARP 应答；（3）为 RARP 发送 RARP 请求和接收 RARP 应答。



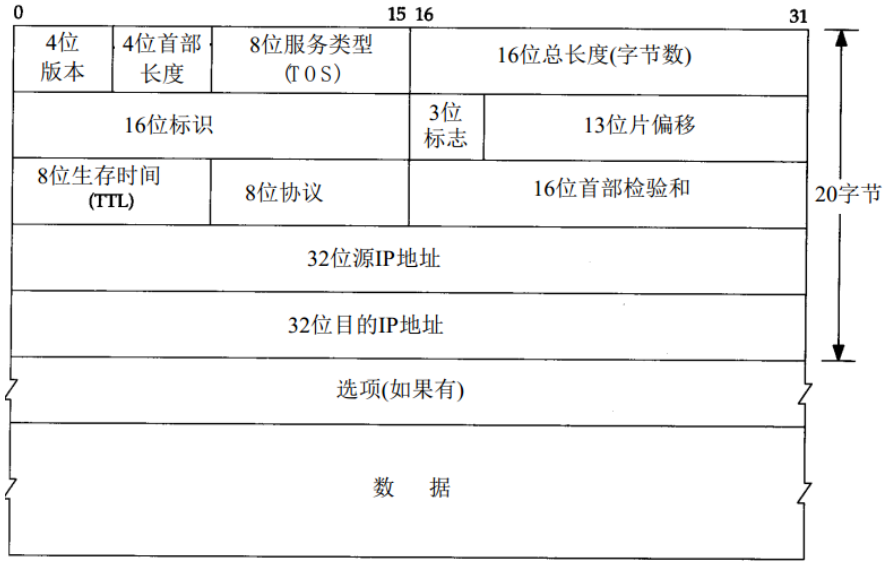
2. SLIP 的全称是 Serial Line IP。它是一种在串行线路上对 IP 数据报进行封装的简单形式。

- IP 数据报以一个称作 END（ 0xc0）的特殊字符结束，大多数实现在数据报的开始处也传一个 END 字符。
- 如果 IP 报文中某个字符为 END，那么就要连续传输两个字节 0xdb 和 0xdc 来取代它。
- 如果 IP 报文中某个字符为 SLIP 的 ESC 字符，那么就要连续传输两个字节 0xdb 和 0xdd 来取代它。

3. CSLIP：压缩的 SLIP。
4. PPP，点对点协议修改了 SLIP 协议中的所有缺陷。
5. 环回接口允许运行在同一台主机上的客户程序和服务器程序通过 TCP/IP 进行通信。A 类网络号 127 就是为环回接口预留的。根据惯例，大多数系统把 IP 地址 127.0.0.1 分配给这个接口，并命名为 localhost。
6. 以太网和 802.3 对数据帧的长度都有一个限制，其最大值分别是 1500 和 1492 字节。链路层的这个特性称作 MTU，最大传输单元。如果 IP 层有一个数据报要传，而且数据的长度比链路层的 MTU 还大，那么 IP 层就需要进行分片，把数据报分成若干片，这样每一片都小于 MTU。

第 3 章 IP：网际协议

1. IP 数据报的格式如图 3-1 所示。普通的 IP 首部长为 20 个字节，除非含有选项字段。



2. TTL（time-to-live）生存时间字段设置了数据报可以经过的最多路由器数。当该字段的值为 0 时，数据报就被丢弃，并发送 ICMP 报文通知源主机。
3. IP 路由选择是简单的，特别对于主机来说。如果目的主机与源主机直接相连（如点对点链路）或都在一个共享网络上（以太网或令牌环网），那么 IP 数据报就直接送到目的主机上。否则，主机把数据报发往一默认的路由器上，由路由器来转发该数据报。大多数的主机都是采用这种简单机制。
4. 除了 IP 地址以外，主机还需要知道有多少比特用于子网号及多少比特用于主机号。这是在引导过程中通过子网掩码来确定的。这个掩码是一个 32 bit 的值，其中值为 1 的比特留给网络号和子网号，为 0 的比特留给主机号。

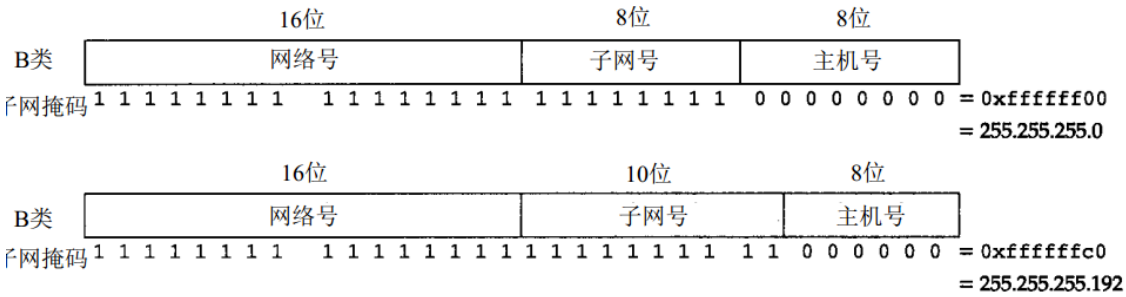


图3-7　两种不同的B类地址子网掩码的例子

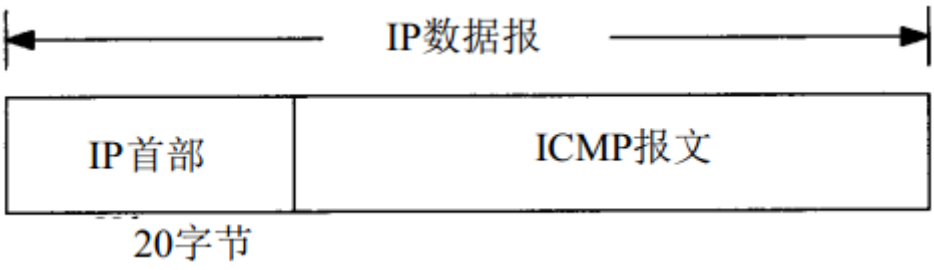
5. ifconfig 命令一般在引导时运行，以配置主机上的每个接口。netstat 命令也提供系统上的接口信息。-i 参数将打印出接口信息，-n 参数则打印出 IP 地址，而不是主机名字。

第 4 章 ARP：地址解析协议

1. 地址解析协议，即 ARP，是根据 IP 地址获取物理地址的一个 TCP/IP 协议。
2. ARP 为 IP 地址到对应的硬件地址之间提供动态映射。我们之所以用动态这个词是因为这个过程是自动完成的，一般应用程序用户或系统管理员不必关心。RARP 是被那些没有磁盘驱动器的系统使用（一般是无盘工作站或 X 终端），它需要系统管理员进行手工设置。
3. ARP 的功能是在 32bit 的 IP 地址和采用不同网络技术的硬件地址之间提供动态映射。
4. ARP 高效运行的关键是由于每个主机上都有一个 ARP 高速缓存。这个高速缓存存放了最近 Internet 地址到硬件地址之间的映射记录。高速缓存中每一项的生存时间一般为 20 分钟，起始时间从被创建时开始算起。我们可以用 arp 命令来检查 ARP 高速缓存。参数-a 的意思是显示高速缓存中所有的内容。
5. 如果 ARP 请求是从一个网络的主机发往另一个网络上的主机，那么连接这两个网络的路由器就可以回答该请求，这个过程称作委托 ARP 或 ARP 代理。
6. 免费 ARP 是指主机发送 ARP 查找自己的 IP 地址。通常，它发生在系统引导期间进行接口配置的时候。主机可以通过它来确定另一个主机是否设置了相同的 IP 地址。
7. 超级用户可以用选项-d 来删除 ARP 高速缓存中的某一项内容。

第 6 章 ICMP： Internet 控制报文协议

1. ICMP 经常被认为是 IP 层的一个组成部分。它传递差错报文以及其他需要注意的信息。ICMP 报文通常被 IP 层或更高层协议（TCP 或 UDP）使用。一些 ICMP 报文把差错报文返回给用户进程。所有报文的前 4 个字节都是一样的，但是剩下的其他字节则互不相同。



2. 类型字段可以有 15 个不同的值，以描述特定类型的 ICMP 报文。某些 ICMP 报文还使

用代码字段的值来进一步描述不同的条件。

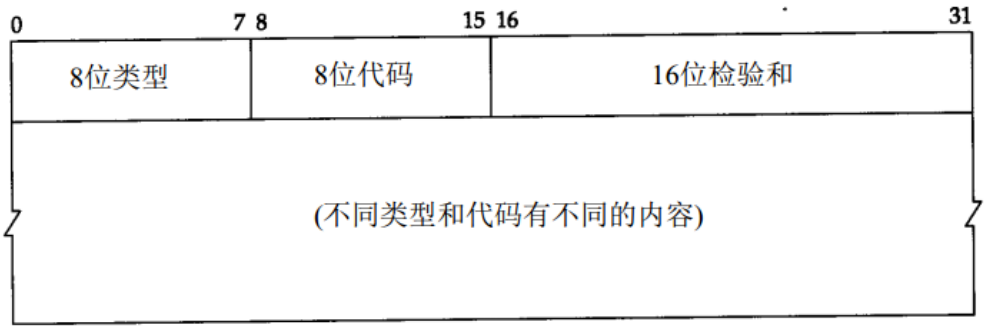


图6-2 ICMP报文

3. ICMP 时间戳请求允许系统向另一个系统查询当前的时间。返回的建议值是自午夜开始计算的毫秒数。由于返回的时间是从午夜开始计算的，因此调用者必须通过其他方法获知当时的日期，这是它的一个缺陷。

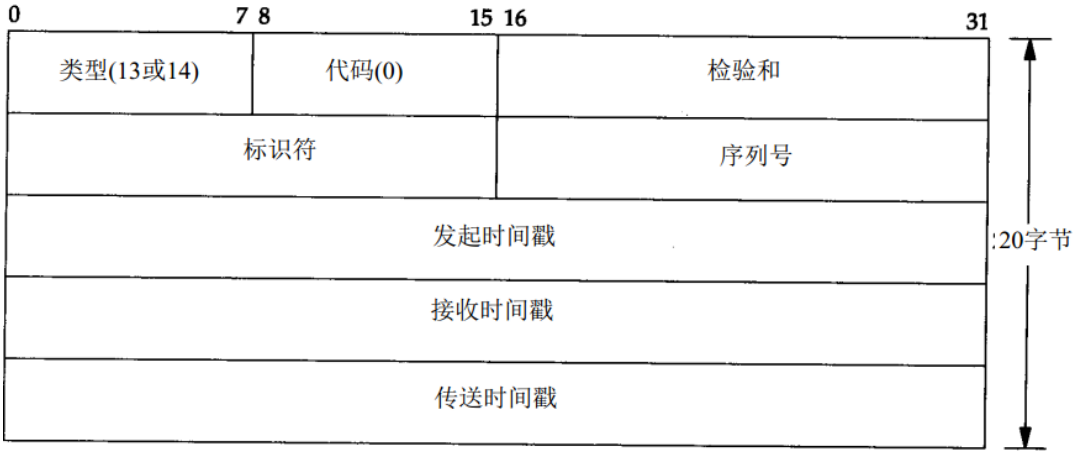


图6-6 ICMP时间戳请求和应答报文

4. UDP 的规则之一是，如果收到一份 UDP 数据报而目的端口与某个正在使用的进程不相符，那么 UDP 返回一个 ICMP 不可达报文。可以用 TFTP 来强制生成一个端口不可达报文。

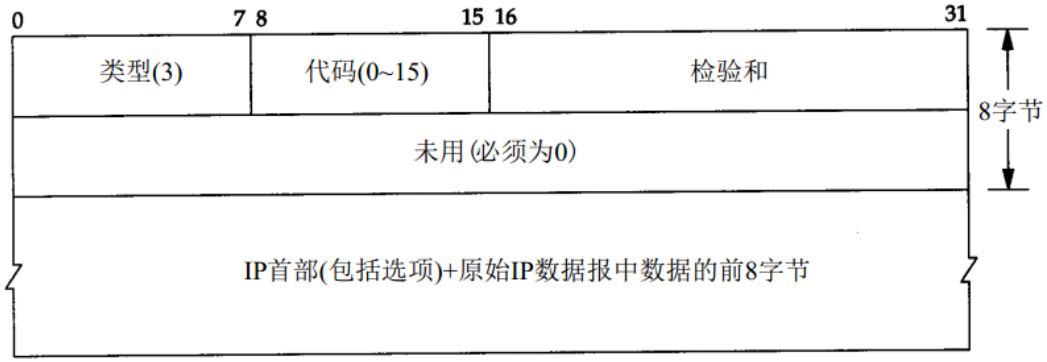


图6-10 ICMP不可达报文

第 7 章 Ping 程序

1. 一台主机的可达性可能不只取决于 IP 层是否可达，还取决于使用何种协议以及端口

号。Ping 程序的运行结果可能显示某台主机不可达，但我们可以用 Telnet 远程登录到该台主机的 25 号端口。

2. ICMP 回显请求和回显应答报文如图 7-1 所示：

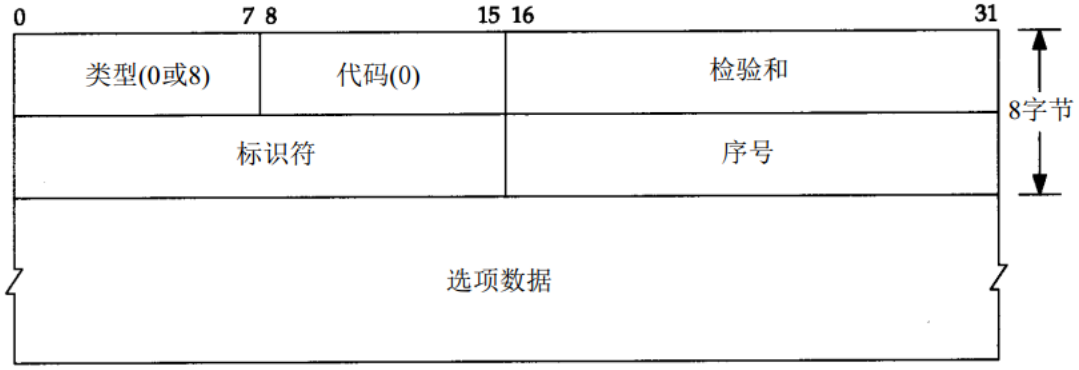


图7-1　ICMP回显请求和回显应答报文格式

第 8 章 Traceroute 程序

1. Traceroute 程序可以让我们看到 IP 数据报从一台主机传到另一台主机所经过的路由。其操作很简单：开始时发送一个 TTL 字段为 1 的 UDP 数据报，然后将 TTL 字段每次加 1，以确定路径中的每个路由器。每个路由器在丢弃 UDP 数据报时都返回一个 ICMP 超时报文 2，而最终目的主机则产生一个 ICMP 端口不可达的报文。这样就得到了该路径中的路由器的地址。

第 9 章 IP 选路

1. 路由表是指路由器或者其他互联网网络设备上存储的一张路由信息表，该表中存有到达特定网络终端的路径，在某些情况下，还有一些与这些路径相关的度量。
2. 选路是 IP 最重要的功能之一。需要进行选路的数据报可以由本地主机产生，也可以由其他主机产生。在后一种情况下，主机必须配置成一个路由器，否则通过网络接口接收到的数据报，如果目的地址不是本机就要被丢弃（例如，悄无声息地被丢弃）。
3. 我们列出了 IP 搜索路由表的几个步骤：

   1) 搜索匹配的主机地址；

   2) 搜索匹配的网络地址；

   3) 搜索默认表项（默认表项一般在路由表中被指定为一个网络表项，其网络号为 0）。

4. netstat 命令列出路由表，然后以-n 选项再次执行该命令，以数字格式打印出 IP 地址

```
svr4 % netstat -rn
Routing tables
Destination       Gateway           Flags    Refcnt Use      Interface
140.252.13.65     140.252.13.35     UGH      0      0        emd0
127.0.0.1         127.0.0.1         UH       1      0        lo0
default           140.252.13.33     UG       0      0        emd0
140.252.13.32     140.252.13.34     U        4      25043    emd0
```

对于一个给定的路由器，可以打印出五种不同的标志（flag）：

- U 该路由可以使用。
- G 该路由是到一个网关（路由器）。如果没有设置该标志，说明目的地是直接相连的。
- H 该路由是到一个主机，也就是说，目的地址是一个完整的主机地址。

- D 该路由是由重定向报文创建的。
- M 该路由已被重定向报文修改。

参考记数 Refcnt 列给出的是正在使用路由的活动进程个数。 "use"显示的是通过该路由发送的分组数。

5. 每当初始化一个接口时（通常是用 ifconfig 命令设置接口地址），就为接口自动创建一个直接路由。到达主机或网络的路由如果不是直接相连的，那么就必须加入路由表。一个常用的方法是在系统引导时显式地在初始化文件中运行 route 命令。

```
route add default sun 1
route add slip bsdi 1
```

6. 如果路由表中没有默认项，而又没有找到匹配项，如果数据报是由本地主机产生的，那么就给发送该数据报的应用程序返回一个差错，或者是“主机不可达差错”或者是“网络不可达差错”。如果是被转发的数据报，那么就给原始发送端发送一份 ICMP 主机不可达的差错报文。
7. 当路由器收到一份 IP 数据报但又不能转发时，就要发送一份 ICMP“主机不可达”差错报文。
8. 当 IP 数据报应该被发送到另一个路由器时，收到数据报的路由器就要发送 ICMP 重定向差错报文给 IP 数据报的发送端。即 IP 数据报发送到路由器 1，路由器 1 发现路由器 2 是发送该数据报的下一站，那路由器 1 则发送一个重定义报文到主机，告诉主机以后把数据报发送到 2 而不是 1。
9. 初始化路由表的一种方法，是在配置文件中指定静态路由。这种方法经常用来设置默认路由。另一种新的方法是利用 ICMP 路由器通告和请求报文。主机在引导以后要广播或多播传送一份路由器请求报文。一台或更多台路由器响应一份路由器通告报文。另外，路由器定期地广播或多播传送它们的路由器通告报文，允许每个正在监听的主机相应地更新它们的路由表。

第 10 章 动态选路协议

1. 当相邻路由器之间进行通信，以告知对方每个路由器当前所连接的网络，这时就出现了动态选路。路由器之间必须采用选路协议进行通信。路由守护程序将选路策略加入到系统中，选择路由并加入到内核的路由表中。如果守护程序发现前往同一信宿存在多条路由，那么它（以某种方法）将选择最佳路由并加入内核路由表中。如果路由守护程序发现一条链路已经断开（可能是路由器崩溃或电话线路不好），它可以删除受影响的路由或增加另一条路由以绕过该问题。
2. Unix 系统上常常运行名为 routed 路由守护程序。几乎在所有的 TCP/IP 实现中都提供该程序。该程序只使用 RIP 进行通信。这是一种用于小型到中型网络中的协议。
3. RIP：选路信息协议。RIP 报文包含中在 UDP 数据报中，如图所示：

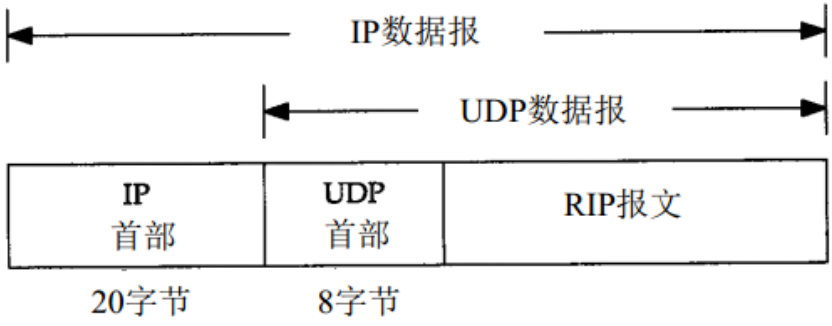


图10-2　封装在UDP数据报中的RIP报文

4. OSPF 是除 RIP 外的另一个内部网关协议。它克服了 RIP 的所有限制。与采用距离向量的 RIP 协议不同的是，OSPF 是一个链路状态协议。距离向量的意思是，RIP 发送的报文包含一个距离向量（跳数）。OSPF 与 RIP（以及其他选路协议）的不同点在于，OSPF 直接使用 IP。也就是说，它并不使用 UDP 或 TCP。
5. BGP 是一种不同自治系统的路由器之间进行通信的外部网关协议。
6. CIDR：无类型域间选路。我们指出了 B 类地址的缺乏，因此现在的多个网络站点只能采用多个 C 类网络号，而不采用单个 B 类网络号。尽管分配这些 C 类地址解决了一个问题（B 类地址的缺乏），但它却带来了另一个问题：每个 C 类网络都需要一个路由表表项。无类型域间选路（CIDR）是一个防止 Internet 路由表膨胀的方法，它也称为超网。
7. 有两种基本的选路协议，即用于同一自治系统各路由器之间的内部网关协议（IGP）和用于不同自治系统内路由器通信的外部网关协议（EGP）。最常用的 IGP 是路由信息协议（RIP），而 OSPF 是一个正在得到广泛使用的新 IGP。一种新近流行的 EGP 是边界网关协议（BGP）。

第 11 章 UDP：用户数据报协议

1. UDP 是一个简单的面向数据报的运输层协议：进程的每个输出操作都正好产生一个 UDP 数据报，并组装成一份待发送的 IP 数据报。UDP 不提供可靠性：它把应用程序传给 IP 层的数据发送出去，但是并不保证它们能到达目的地。
2. UDP 首部的各字段如图所示：

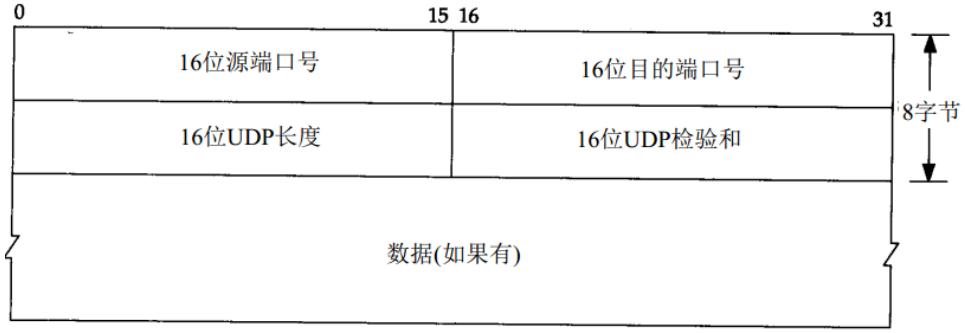


图11-2 UDP首部

3. 检验和，在数据处理和数据通信领域中，用于校验目的地一组数据项的和。校验和是指传输位数的累加，当传输结束时，接收者可以根据这个数值判断是否接到了所有的数据。如果数值匹配，那么说明传送已经完成。
4. UDP 和 TCP 在首部中都有覆盖它们首部和数据的检验和。UDP 的检验和是可选的，而 TCP 的检验和是必需的。
5. 任何时候 IP 层接收到一份要发送的 IP 数据报时，它要判断向本地哪个接口发送数据（选路），并查询该接口获得其 MTU。IP 把 MTU 与数据报长度进行比较，如果需要则进行分片。分片可以发生在原始发送端主机上，也可以发生在中间路由器上。把一份 IP 数据报分片以后，只有到达目的地才进行重新组装。
6. IP 层本身没有超时重传的机制——由更高层来负责超时和重传。
7. 发生 ICMP 不可达差错的另一种情况是，当路由器收到一份需要分片的数据报，而在 IP 首部又设置了不分片（DF）的标志比特。
8. 理论上，IP 数据报的最大长度是 65535 字节，这是由 IP 首部（图 3-1）16 比特总长度字段所限制的。去除 20 字节的 IP 首部和 8 个字节的 UDP 首部，UDP 数据报中用户数据的最长长度为 65507 字节。但是，大多数实现所提供的长度比这个最大值小。

9. 我们同样也可以使用 UDP 产生 ICMP“源站抑制(source quench)”差错。当一个系统（路由器或主机）接收数据报的速度比其处理速度快时，可能产生这个差错。注意限定词“可能”。即使一个系统已经没有缓存并丢弃数据报，也不要求它一定要发送源站抑制报文。

第 12 章 广播和多播

1. 有三种 IP 地址：单播地址、广播地址和多播地址。广播和多播仅应用于 UDP，它们对需将报文同时传往多个接收者的应用来说十分重要。
2. 四种 IP 广播地址：

- 受限的广播。在任何情况下，路由器都不转发目的地址为受限的广播地址的数据报，这样的数据报仅出现在本地网络中。受限的广播通常只在系统初始启动时才会用到。
- 指向网络的广播。指向网络的广播地址是主机号为全 1 的地址。一个路由器必须转发指向网络的广播，但它也必须有一个不进行转发的选择。
- 指向子网的广播。指向子网的广播地址为主机号为全 1 且有特定子网号的地址。作为子网直接广播地址的 IP 地址需要了解子网的掩码。
- 指向所有子网的广播。

3. 有些系统内核和路由器有一选项来控制允许或禁止转发广播数据。
4. IP 多播提供两类服务：

- 向多个目的地址传送数据。
- 客户对服务器的请求。

D 类 IP 地址被称为多播组地址。

第 13 章 IGMP：Internet 组管理协议

1. Internet 组管理协议（IGMP）用于支持主机和路由器进行多播。正如 ICMP 一样，IGMP 也被当作 IP 层的一部分。IGMP 报文通过 IP 数据报进行传输。不像我们已经见到的其他协议，IGMP 有固定的报文长度，没有可选数据。

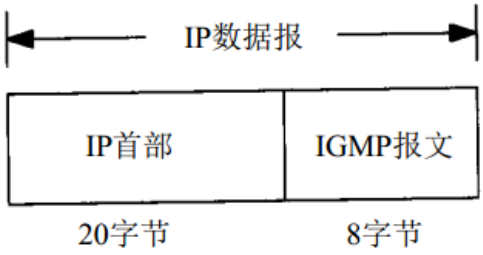


图13-1 IGMP报文封装在IP数据报中

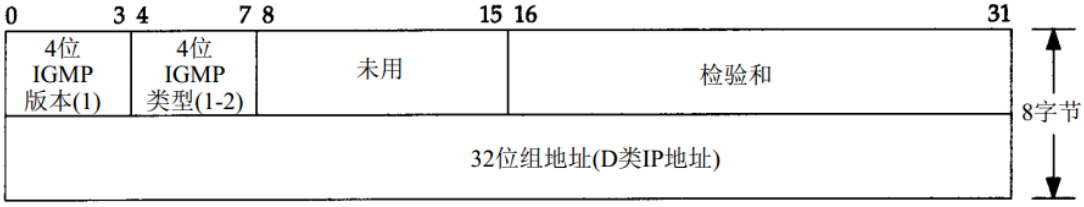


图13-2 IGMP报文的字段格式

2. 多播路由器使用 IGMP 报文来记录与该路由器相连网络中组成员的变化情况。

3. 多播是一种将报文发往多个接收者的通信方式。在一个局域网中或跨越邻近局域网的多播需要使用本章介绍的技术。广播通常局限在单个局域网中，对目前许多使用广播的应用来说，可采用多播来替代广播。

第 14 章 DNS：域名系统

1. 域名系统（DNS）是一种用于 TCP/IP 应用程序的分布式数据库，它提供主机名字和 IP 地址之间的转换及有关电子邮件的选路信息。
2. 以点“.”结尾的域名称为绝对域名或完全合格的域名，例如 sun.tuc.noao.edu.。如果一个域名不以点结尾，则认为该域名是不完全的。
3. DNS 定义了一个用于查询和响应的报文格式：

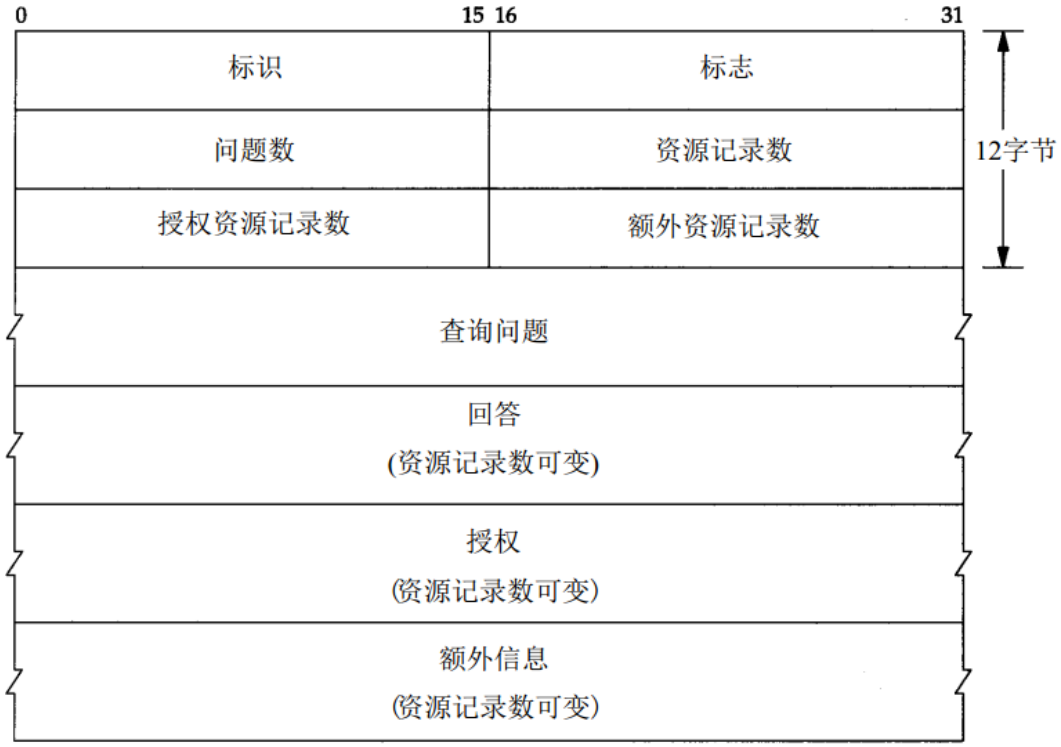


图14-3 DNS查询和响应的一般格式

4. 图显示了如何存储域名 gemini.tuc.noao.edu：

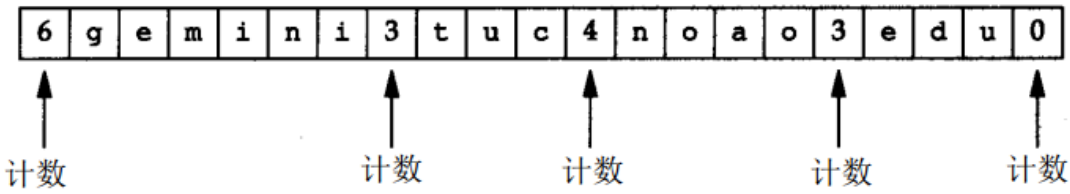


图14-6 域名gemini.tuc.noao.edu 的表示

5. 问题部分中每个问题的格式如图 14–5 所示，通常只有一个问题。

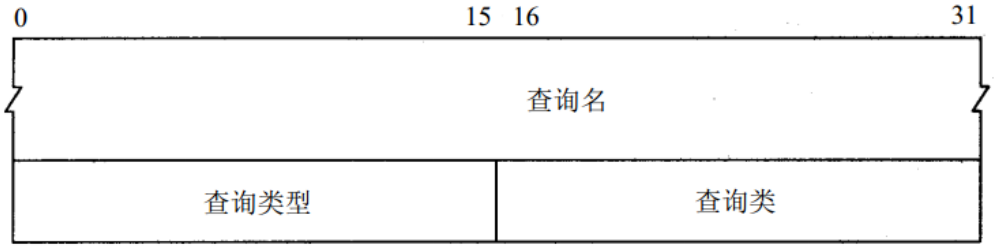


图14-5 DNS查询报文中问题部分的格式

每个问题有一个查询类型，而每个响应（也称一个资源记录，我们下面将谈到）也有一个类型。大约有 20 个不同的类型值，其中的一些目前已经过时。图 14–7 显示了其中的一些值。

名　字	数　值	描　述	类型?	查询类型
A	1	IP地址	•	•
NS	2	名字服务器	•	•
CNAME	5	规范名称	•	•
PTR	12	指针记录	•	•
HINFO	13	主机信息	•	•
MX	15	邮件交换记录	•	•
AXFR	252	对区域转换的请求		•
* 或ANY	255	对所有记录的请求		•

图14-7 DNS问题和响应的类型值和查询类型值

最常用的查询类型是 A 类型，表示期望获得查询名的 IP 地址。

6. DNS 报文中最后的三个字段，回答字段、授权字段和附加信息字段，均采用一种称为资源记录 RR（Resource Record）的相同格式。

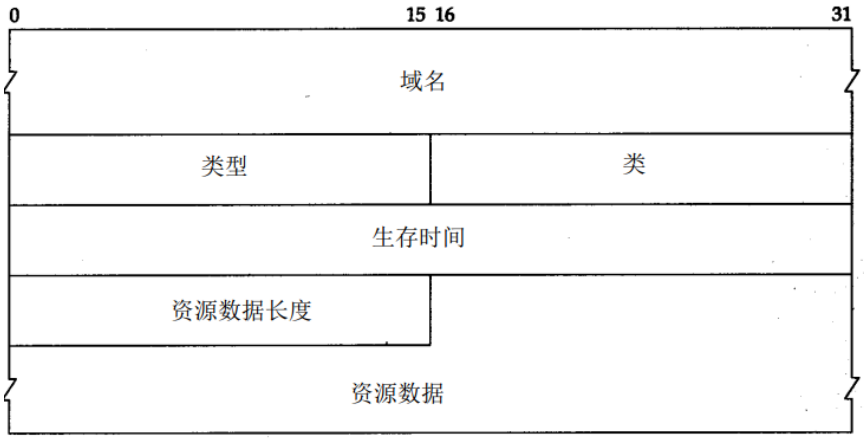


图14-8 DNS资源记录格式

域名是记录中资源数据对应的名字。它的格式和前面介绍的查询名字段格式相同。类型说明 RR 的类型码。它的值和前面介绍的查询类型值是一样的。类通常为 1，指 Internet 数据。生存时间字段是客户程序保留该资源记录的秒数。资源记录通常的生存时间值为 2 天。资源数据长度说明资源数据的数量。

7. DNS 中一直难于理解的部分就是指针查询方式，即给定一个 IP 地址，返回与该地址对应的域名。

8. /etc/resolv.conf 文件是本地 DNS 域名解析的配置文件。
9. DNS 均支持 UDP 和 TCP 访问，使用的端口号无论对 UDP 还是 TCP 都是 53。

第 15 章 TFTP：简单文件传送协议

1. TFTP 即简单文件传送协议，最初打算用于引导无盘系统（通常是工作站或 X 终端）。和使用 TCP 的文件传送协议（FTP）不同，为了保持简单和短小，TFTP 将使用 UDP。
2. TFTP 分组中并不提供用户名和口令。这是 TFTP 的一个特征（即“安全漏洞”）。由于 TFTP 是设计用于系统引导进程，它不可能提供用户名和口令。

第 16 章 BOOTP：引导程序协议

1. BOOTP 是一种用于无盘系统进行系统引导的替代方法，又称为引导程序协议。
2. BOOTP 请求和应答均被封装在 UDP 数据报中：

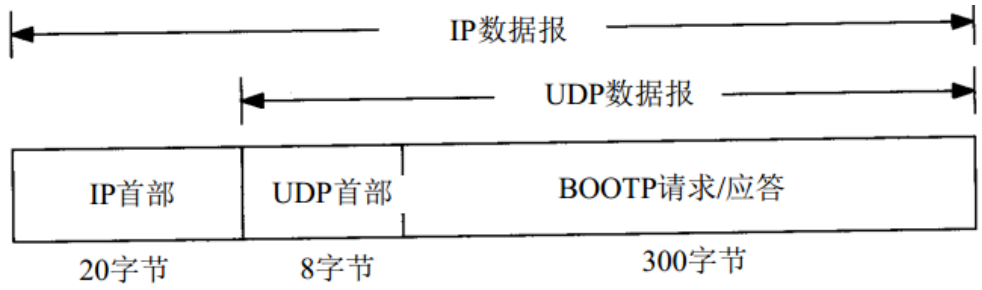


图16-1 BOOTP 请求和应答封装在一个UDP数据报内

3. BOOTP 有两个熟知端口：BOOTP 服务器为 67，BOOTP 客户为 68。选择两个端口而不是仅选择一个端口为 BOOTP 服务器用的原因是：服务器的应答可以进行广播（但通常是不用广播的）。
4. BOOTP 使用 UDP，它为引导无盘系统获得它的 IP 地址提供了除 RARP 外的另外一种选择。BOOTP 还能返回其他的信息，如路由器的 IP 地址、客户的子网掩码和名字服务器的 IP 地址。

第 17 章 TCP：传输控制协议

1. 两个应用程序通过 TCP 连接交换 8 bit 字节构成的字节流。TCP 不在字节流中插入记录标识符。我们将这称为字节流服务（byte stream service）。如果一方的应用程序先传 10 字节，又传 20 字节，再传 50 字节，连接的另一方将无法了解发方每次发送了多少字节。收方可以分 4 次接收这 80 个字节，每次接收 20 字节。一端将字节流放到 TCP 连接上，同样的字节流将出现在 TCP 连接的另一端。
2. TCP 对字节流的内容不作任何解释。TCP 不知道传输的数据字节流是二进制数据，还是 ASCII 字符、EBCDIC 字符或者其他类型数据。对字节流的解释由 TCP 连接双方的应用层解释。
3. TCP 数据被封装在一个 IP 数据报中，如图所示：

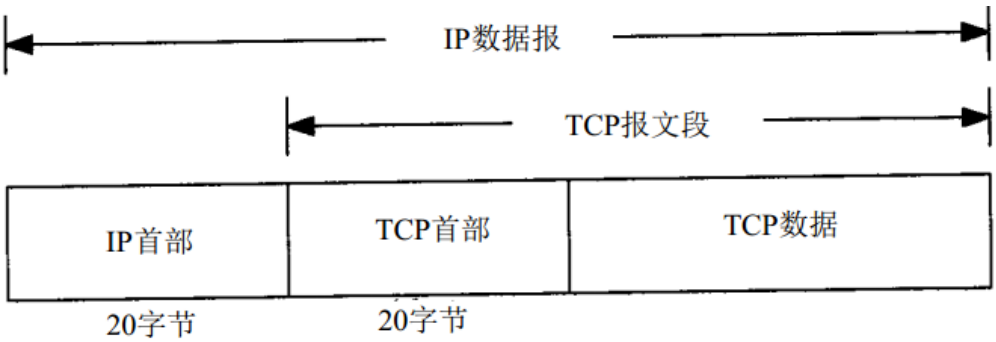


图17-1　TCP数据在IP数据报中的封装

4. TCP 为应用层提供全双工服务。这意味数据能在两个方向上独立地进行传输。
5. 最常见的可选字段是最长报文大小，又称为 MSS。

第 18 章 TCP 连接的建立与终止

1. 建立一个连接需要三次握手，而终止一个连接要经过 4 次握手。

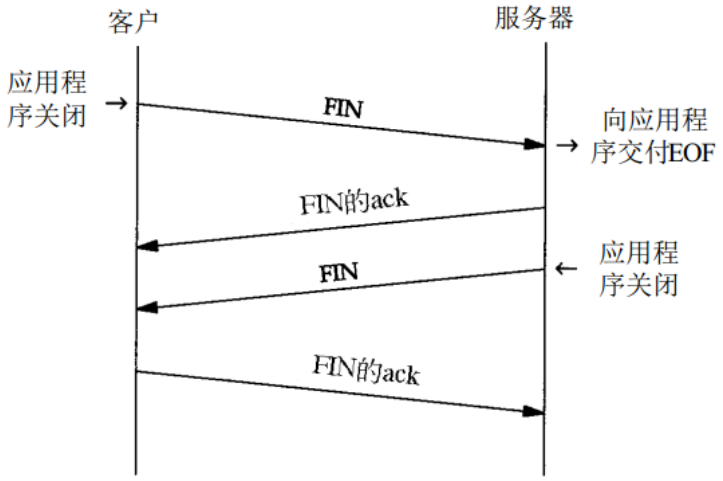


图18-4　连接终止期间报文段的正常交换

2. TCP 提供了连接的一端在结束它的发送后还能接收来自另一端数据的能力。这就是所谓的半关闭。只有很少的应用程序使用它。
3. 两个应用程序同时彼此执行主动打开的情况是可能的，尽管发生的可能性极小。每一方必须发送一个 SYN，且这些 SYN 必须传递给对方。这需要每一方使用一个对方熟知的端口作为本地端口。这又称为同时打开。双方都执行主动关闭也是可能的，TCP 协议也允许这样的同时关闭。

第 19 章 TCP 的交互数据流

1. Nagle 算法就是为了尽可能发送大块数据，避免网络中充斥着许多小数据块。Nagle 算法的基本定义是任意时刻，最多只能有一个未被确认的小段。所谓“小段”，指的是小于 MSS 尺寸的数据块，所谓“未被确认”，是指一个数据块发送出去后，没有收到对方发送的 ACK 确认该数据已收到。有时我们也需要关闭 Nagle 算法。一个典型的例子是 X 窗口系统服务器：小消息（鼠标移动）必须无时延地发送，以便为进行某种操作的交互用户提供实时的反馈。
2. 交互数据总是以小于最大报文段长度的分组发送。在较慢的广域网环境中，通常使用

Nagle 算法来减少这些小报文段的数目。这个算法限制发送者任何时候只能有一个发送的小报文段未被确认。

第 20 章 TCP 的成块数据流

1. 滑动窗口协议，该协议允许发送方在停止并等待确认前发送多个数据分组。由于发送方不必每发一个分组就停下来等待确认，因此该协议可以加速数据的传输。
2. 滑动窗口协议，理解有两点：1.“窗口”对应的是一段可以被发送者发送的字节序列，其连续的范围称之为“窗口”；2.“滑动”则是指这段“允许发送的范围”是可以随着发送的过程而变化的，方式就是按顺序“滑动”。
3. TCP 滑动窗口用来暂存两台计算机间要传送的数据分组。每台运行 TCP 协议的计算机有两个滑动窗口：一个用于数据发送，另一个用于数据接收。
4. 所谓流量控制，主要是接收方传递信息给发送方，使其不要发送数据太快，是一种端到端的控制。主要的方式就是返回的 ACK 中会包含自己的接收窗口的大小，并且利用大小来控制发送方的数据发送。
5. 网络中的链路容量和交换结点中的缓存和处理机都有着工作的极限，当网络的需求超过它们的工作极限时，就出现了拥塞。拥塞控制就是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。常用的方法就是：

- 慢启动、拥塞控制：发送的数据包大小依次递增 2 的指数级，当出现丢包时，依次递减一半。
- 快重传、快恢复

6. 滑动窗口协议的基本原理就是在任意时刻，发送方都维持了一个连续的允许发送的帧的序号，称为发送窗口；同时，接收方也维持了一个连续的允许接收的帧的序号，称为接收窗口。发送窗口和接收窗口的序号的上下界不一定要一样，甚至大小也可以不同。不同的滑动窗口协议窗口大小一般不同。发送方窗口内的序列号代表了那些已经被发送，但是还没有被确认的帧，或者是那些可以被发送的帧。
7. 我们使用三个术语来描述窗口左右边沿的运动：

- 称窗口左边沿向右边沿靠近为窗口合拢。这种现象发生在数据被发送和确认时。
- 当窗口右边沿向右移动时将允许发送更多的数据，我们称之为窗口张开。这种现象发生在另一端的接收进程读取已经确认的数据并释放了 TCP 的接收缓存时。
- 当右边沿向左移动时，我们称之为窗口收缩。

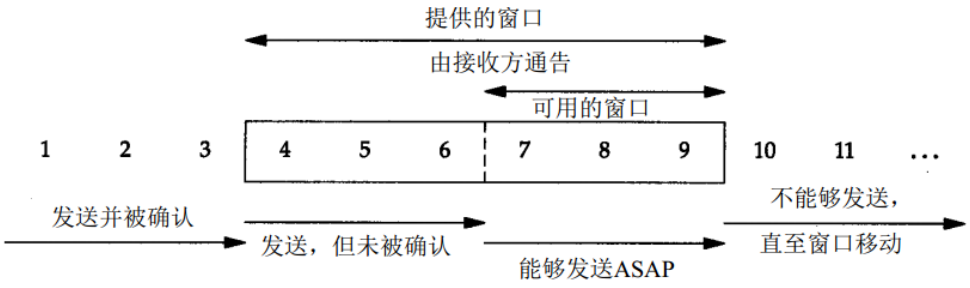


图20-4　TCP滑动窗口的可视化表示

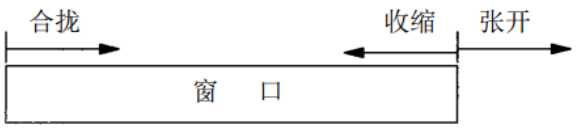


图20-5　窗口边沿的移动

8. 发送方使用 PUSH 标志通知接收方将所收到的数据全部提交给接收进程。这里的数据包括与 PUSH 一起传送的数据以及接收方 TCP 已经为接收进程收到的其他数据。

第 21 章 TCP 的超时与重传

1. 对每个连接，TCP 管理 4 个不同的定时器。

- 重传定时器使用于当希望收到另一端的确认。
- 坚持(persist)定时器使窗口大小信息保持不断流动，即使另一端关闭了其接收窗口。
- 保活(keep alive)定时器可检测到一个空闲连接的另一端何时崩溃或重启。
- 2MSL 定时器测量一个连接处于 TIME_WAIT 状态的时间。

2. RTT：往返时间。

   RTO：TCP 超时重传机制(RTO：Retransmission Timeout)，是 TCP 操作计时器的一种。

3. Karn 算法的提出是为了更好的计算 RTO。报文段每重传一次，就把超时重传时间 RTO 增大一些。典型的做法是取新的重传时间为 2 倍的就得重传时间。当不再发生报文段的重传时，才根据正常的计算方法计算 RTO。这样，只要得到 RTT，即可知道是哪次重传的 ACK 了。

   相关背景：发送端发出一个报文段，并且设定的超时时间到了，还没有收到确认，于是重传报文段。经过了一段时间后，收到了报文段。现在的问题是：如何判定此确认报文段是对先发送的报文段的确认，还是对后来重传的报文段的确认？由于重传的报文段与原先的报文段完全一样，因此源主机在收到确认后就无法做出正确的判断，而正确的判断对确定平滑的往返时间（SRTT）的关系很大。

4. 拥塞避免算法是一种处理丢失分组的方法，它和慢启动算法是两个目的不同、独立的算法。但是当拥塞发生时，我们希望降低分组进入网络的传输速率，于是可以调用慢启动来作到这一点。在实际中这两个算法通常在一起实现。拥塞避免是发送方使用的流量控制，而通告窗口则是接收方使用的流量控制。

5. TCP 能够遇到的最常见的 ICMP 差错就是源站抑制、主机不可达和网络不可达。

6. 当 TCP 超时并重传时，它不一定要重传同样的报文段。相反，TCP 允许进行重新分组而发送一个较大的报文段，这将有助于提高性能（当然，这个较大的报文段不能够超过接收方声明的 MSS）。

第 22 章 TCP 的坚持定时器

1. 如果一个确认丢失了，则双方就有可能因为等待对方而使连接终止：接收方等待接收数据（因为它已经向发送方通告了一个非 0 的窗口），而发送方在等待允许它继续发送数据的窗口更新。为防止这种死锁情况的发生，发送方使用一个坚持定时器(persist timer)来周期性地向接收方查询，以便发现窗口是否已增大。

2. 在连接的一方需要发送数据但对方已通告窗口大小为 0 时，就需要设置 TCP 的坚持定时器。

第 23 章 TCP 的保活定时器

1. 保活定时器的目的在于看看对方有没有发生异常，如果有异常就及时关闭连接。当传输双方不主动关闭连接时，就算双方没有交换任何数据，连接也是一直有效的。

2. 当服务器发送探测报文时，客户端可能处于 4 种不同的情况：仍然正常运行、已经崩溃、已经崩溃并重启了、由于中间链路问题不可达。在不同的情况下，服务器会得到不一样的反馈。

- 客户主机依然正常运行，并且从服务器端可达
  - 客户端的 TCP 响应正常，从而服务器端知道对方是正常的。保活定时器会在两小时以后继续触发。
- 客户主机已经崩溃，并且关闭或者正在重新启动
  - 客户端的 TCP 没有响应，服务器没有收到对探测包的响应，此后每隔 75s 发送探测报文，一共发送 9 次。socket 函数会返回-1，errno 设置为 ETIMEDOUT，表示连接超时。
- 客户主机已经崩溃，并且重新启动了
  - 客户端的 TCP 发送 RST，服务器端收到后关闭此连接。socket 函数会返回-1，errno 设置为 ECONNRESET，表示连接被对端复位了。
- 客户主机依然正常运行，但是从服务器不可达
  - 双方的反应和第二种是一样的，因为服务器不能区分对端异常与中间链路异常。socket 函数会返回-1，errno 设置为 EHOSTUNREACH，表示对端不可达。

第 24 章 TCP 的未来和性能

1. 具有大的带宽时延乘积的网络被称为长肥网络（即 LFN），而一个运行在 LFN 上的 TCP 连接被称为长肥管道。
2. 路径 MTU 发现在 MTU 较大时，对于非本地连接，允许 TCP 使用比默认的 536 大的窗口。这样可以提高性能。
3. 窗口扩大选项使最大的 TCP 窗口从 65535 增加到 1 千兆字节以上。时间戳选项允许多个报文段被精确计时，并允许接收方提供序号回绕保护（PAWS）。这对于高速连接是必须的。这些新的 TCP 选项在连接时进行协商，并被不理解它们的旧系统忽略，从而允许较新的系统与旧的系统进行交互。
4. 为事务用的 TCP 扩展，即 T/TCP，允许一个客户/服务器的请求-应答序列在通常的情况下只使用三个报文段来完成。它避免使用三次握手，并缩短了 TIME_WAIT 状态，其方法是为每个主机高速缓存少量的信息，这些信息曾用来建立过一个连接。

第 25 章 SNMP: 简单网络管理协议

1. 基于 TCP/IP 的网络管理包含两个部分：网络管理站（也叫管理进程，manager）和被管的网络单元（也叫被管设备）。被管设备种类繁多，例如：路由器、X 终端、终端服务器和打印机等。被管设备端和管理相关的软件叫做代理程序(agent)或代理进程。管理站一般都是带有彩色监视器的工作站，可以显示所有被管设备的状态(例如连接是否掉线、各种连接上的流量状况等)。
2. 管理进程和代理进程之间的通信可以有两种方式。一种是管理进程向代理进程发出请求，询问一个具体的参数值（例如：你产生了多少个不可达的 ICMP 端口？）。另外一种方式是代理进程主动向管理进程报告有某些重要的事件发生（例如：一个连接口掉线了）。
3. 基于 TCP/IP 的网络管理包含 3 个组成部分：

- 一个管理信息库 MIB。管理信息库包含所有代理进程的所有可被查询和修改的参数。
- 关于 MIB 的一套公用的结构和表示符号。叫做管理信息结构 SMI。
- 管理进程和代理进程之间的通信协议，叫做简单网络管理协议 SNMP。在 SNMP 中，用得最多的协议还是 UDP。

4. 关于管理进程和代理进程之间的交互信息，SNMP 定义了 5 种报文：

- get-request 操作：从代理进程处提取一个或多个参数值。
- get-next-request 操作：从代理进程处提取一个或多个参数的下一个参数值（关于“下一个（next）”的含义将在后面的章节中介绍）。
- set-request 操作：设置代理进程的一个或多个参数值。
- get-response 操作：返回的一个或多个参数值。这个操作是由代理进程发出的。它是前面 3 中操作的响应操作。
- trap 操作：代理进程主动发出的报文，通知管理进程有某些事情发生。

前面的 3 个操作是由管理进程向代理进程发出的。后面两个是代理进程发给管理进程的（为简化起见，前面 3 个操作今后叫做 get、get-next 和 set 操作）。管理进程发出的前面 3 种操作采用 UDP 的 161 端口。代理进程发出的 Trap 操作采用 UDP 的 162 端口。由于收发采用了不同的端口号，所以一个系统可以同时为管理进程和代理进程

5. 对象标识是一种数据类型，它指明一种“授权”命名的对象。“授权”的意思就是这些标识不是随便分配的，它是由一些权威机构进行管理和分配的。对象标识是一个整数序列，以点（“.”）分隔。
6. 所谓管理信息库，或者 MIB，就是所有代理进程包含的、并且能够被管理进程进行查询和设置的信息的集合。MIB 被划分为若干个组，如 system、interfaces、at（地址转换）和 ip 组等。
7. 在正式的 SNMP 规范中都是采用 ASN.1 语法，并且在 SNMP 报文中比特的编码采用 BER。

第 26 章 Telnet 和 Rlogin：远程登录

1. 在 TCP/IP 网络上，有两种应用提供远程登录功能：

- Telnet 是标准的提供远程登录功能的应用，几乎每个 TCP/IP 的实现都提供这个功能。它能够运行在不同操作系统的主机之间。Telnet 通过客户进程和服务器进程之间的选项协商机制，从而确定通信双方可以提供的功能特性。
- Rlogin 起源于伯克利 Unix，开始它只能工作在 Unix 系统之间，现在已经可以在其他操作系统上运行

2. 远程登录采用客户-服务器模式。图 26-1 显示的是一个 Telnet 客户和服务器的典型连接图：

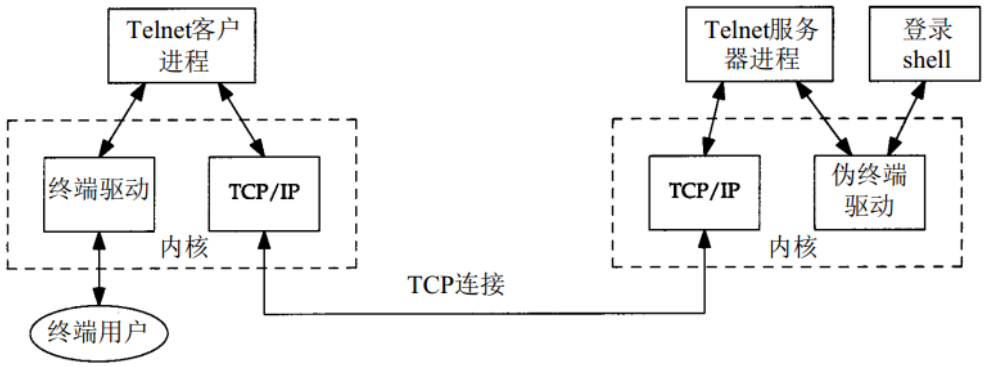


图26-1　客户-服务器模式的Telnet简图

第 27 章 FTP：文件传送协议

1. FTP 采用两个 TCP 连接来传输一个文件。

- 控制连接以通常的客户服务器方式建立。服务器以被动方式打开众所周知的用于 FTP 的端口（21），等待客户的连接。客户则以主动方式打开 TCP 端口 21，来建立连接。控制连接始终等待客户与服务器之间的通信。该连接将命令从客户传给服务器，并传回服务器的应答。
- 每当一个文件在客户与服务器之间传输时，就创建一个数据连接。

  控制连接一直保持到客户-服务器连接的全过程，但数据连接可以根据需要随时来，随时走。

2. FTP 协议规范提供了控制文件传送与存储的多种选择。在以下四个方面中每一个方面都必须作出一个选择：

- 文件类型
  - ASCII 码文件类型（默认）。
  - EBCDIC 文件类型　该文本文件传输方式要求两端都是 EBCDIC 系统。
  - 图像文件类型　　也称为二进制文件类型）。
  - 本地文件类型
- 格式控制
  - 非打印（默认）　　文件中不含有垂直格式信息。
  - 远程登录格式控制　　文件含有向打印机解释的远程登录垂直格式控制。
  - Fortran 回车控制　　每行首字符是 Fortran 格式控制符。
- 结构
  - 文件结构（默认）　文件被认为是一个连续的字节流。不存在内部的文件结构。
  - 记录结构　　该结构只用于文本文件（ASCII 或 EBCDIC）。
  - 页结构　　每页都带有页号发送，以便收方能随机地存储各页。该结构由 TOPS-20 操
  - 作系统提供（主机需求 RFC 不提倡采用该结构）。
- 传输方式
  - 流方式（默认）　　文件以字节流的形式传输。对于文件结构，发方在文件尾提示关闭数据连接。对于记录结构，有专用的两字节序列码标志记录结束和文件结束。
  - 块方式　　文件以一系列块来传输，每块前面都带有一个或多个首部字节。
  - 压缩方式　　一个简单的全长编码压缩方法，压缩连续出现的相同字节。

3. 数据连接有以下三大用途：

- 从客户向服务器发送一个文件。
- 从服务器向客户发送一个文件。
- 从服务器向客户发送文件或目录列表。

第 28 章 SMTP: 简单邮件传送协议

1. 图显示了一个用 TCP/IP 交换电子邮件的示意图。

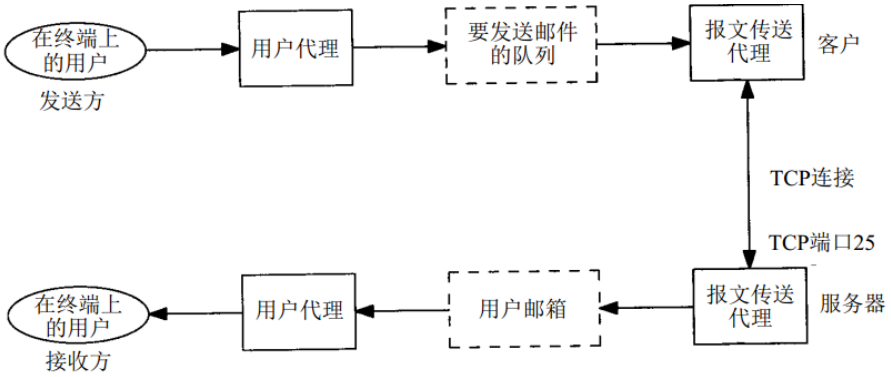


图28-1 Internet电子邮件示意图

2. 用 TCP 进行的邮件交换是由报文传送代理 MTA（Message Transfer Agent）完成的。最普通的 Unix 系统中的 MTA 是 Sendmail。用户通常不和 MTA 打交道，由系统管理员负责设置本地的 MTA。通常，用户可以选择它们自己的用户代理。

3. MUA、MTA 和 MDA：

- MUA（MAIL USER AGENT）：电子邮件系统的构成之一，接受用户输入的各种指令，将用户的邮件发送至 MTA 或者通过 POP3、IMAP 协议将邮件从 MTA 取到本机。MUA 不接受消息，它们只显示已经在邮箱中的消息。
- MTA：MUA 是用在 Client 端的软件，而 MTA 是用在邮件主机上的软件，它也是主要的邮件服务器。MTA 就是“邮件传送代理”的意思，既然是“传送代理”，那么用户寄信与收信时，都找 MTA 就对了！因为它负责帮用户传送。相当于邮政转运车。
- MDA：“邮件投递代理”主要的功能就是将 MTA 接收的信件依照信件的流向（送到哪里）将该信件放置到本机账户下的邮件文件中（收件箱），或者再经由 MTA 将信件送到下个 MTA。如果信件的流向是到本机，这个邮件代理的功能就不只是将由 MTA 传来的邮件放置到每个用户的收件箱，它还可以具有邮件过滤（filtering）与其他相关功能。相当于邮政投递员。

4. 最小 SMTP 实现支持 8 种命令：

- RSET 命令异常中止当前的邮件事务并使两端复位。丢掉所有有关发送方、接收方或邮件的存储信息。
- VRFY 命令使客户能够询问发送方以验证接收方地址，而无需向接收方发送邮件。
- NOOP 命令除了强迫服务器响应一个 OK 应答码（200）外，不做任何事情。
- TURN 命令使客户和服务器交换角色，无需拆除 TCP 连接并建立新的连接就能以相反方向发送邮件（Sendmail 不支持这个命令）。
- 其他还有三个很少被实现的命令（SEND、SOML 和 SAML）取代 MAIL 命令。这三个命令允许邮件直接发送到客户终端（如果已注册）或发送到接收方的邮箱。

第 29 章 网络文件系统

1. 使用 NFS，客户可以透明地访问服务器上的文件和文件系统。NFS 是一个使用 SunRPC 构造的客户服务器应用程序。NFS 客户通过向一个 NFS 服务器发送 RPC 请求来访问其上的文件。

2. NFS 中一个基本概念是文件句柄(file handle)。它是一个不透明(opaque)的对象，用来引用服务器上的一个文件或目录。不透明指的是服务器创建文件句柄，把它传递给客户，然后客户访问文件时，使用对应的文件句柄。每次一个客户进程打开一个实际上

位于一个 NFS 服务器上的文件时，NFS 客户就会从 NFS 服务器那里获得该文件的一个文件句柄。每次 NFS 客户为用户进程读或写文件时，文件句柄就会传给服务器以指定被访问的文件

3. 客户必须在访问服务器上一个文件系统中的文件之前，使用安装协议安装那个文件系统。一般情况下，这是在客户主机引导时完成的。最后的结果就是客户获得服务器文件系统的一个文件句柄

第 **30** 章 其他的 **TCP/IP** 应用程序

1. Finger 协议返回一个指定主机上一个或多个用户的信息。它常被用来检查某个人是否登录了，或者搞清一个人的登录名以便给他发送邮件。Finger 服务器有一个知名的端口 79。
2. Whois 协议是另一种信息服务。尽管任何站点都可以提供一个 Whois 服务器，这个服务器维护着所有的 DNS 域和很多连接在 Internet 上的系统的系统管理员的信息。不幸的是信息有可能是过期的或不完整的。Whois 服务器有一个知名的 TCP 端口 43。
3. X 窗口系统，或简称为 X，是一种客户-服务器应用程序。它可以使得多个客户（应用）使用由一个服务器管理的位映射显示器。服务器是一个软件，用来管理显示器、键盘和鼠标。客户是一个应用程序，它与服务器在同一台主机上或者在不同的主机上。在后一种情况下，客户与服务器之间通信的通用形式是 TCP，尽管也可以使用诸如 DECNET 的其他协议。