第 5 章

传输层

传输层主要在通信子网提供的服务基础上，为源计算机和目的计算机之间提供可靠、透明的数据传输。它位于 OSI 参考模型的第四层，是网络体系结构中最重要的一层。在本章中，将详细介绍 OSI 模型中传输层的一些基本概念和理论。

本章学习要点：

- 传输层概述
- 用户数据报协议 UDP
- 传输控制协议（TCP）
- 流量控制
- TCP 的拥塞控制

5.1 传输层概述

传输层的基本功能是向端系统用户（进程—进程）提供端到端之间的可靠的数据传输，向高层用户屏蔽通信子网的细节，并提供通用的传输接口。

5.1.1 传输层服务

传输层是两台计算机经过网络进行数据通信时，第一个端到端的层次，具有缓冲作用。当网络层服务质量不能满足要求时，它将服务加以提高，以满足高层的要求。当网络层服务质量较好时，它只用很少的工作。传输层还可进行复用，即在一个网络连接上创建多个逻辑连接。

另外，传输层也称为运输层。传输层存在于端开放系统中，是位于低 3 层通信子网系统和高 3 层之间的一层，所以是非常重要的一层。因为它是源端到目的端对数据传送

进行控制从低到高的最后一层，如图 5-1 所示。

Internet 网络是通过许多小型网络组建而成的，所以在世界上各种通信子网在性能上存在着很大差异，如电话交换网、分组交换网、公用数据交换网以及局域网等通信子网都可互联，但它们提供的吞吐量、传输速率、数据延迟、通信费等都相同。

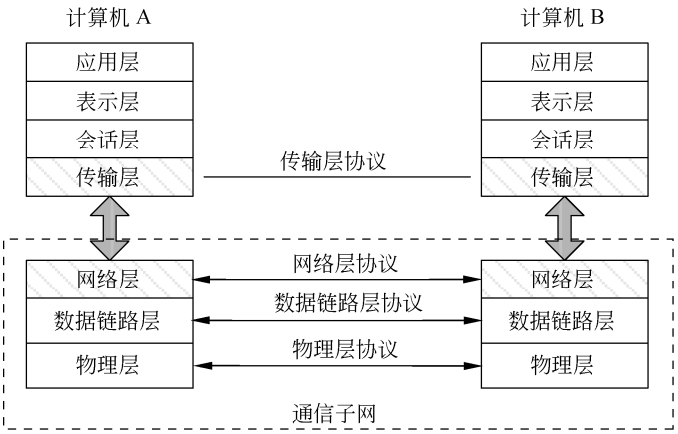

图 5-1 传输层

对于会话层来说，要求网络有一性能恒定的界面。因此，传输层就承担了这一功能，并采用分流/合流、复用等技术来调节通信子网中的差异，使会话层感受不到这些差异的存在。不仅如此，传输层还要具备差错恢复、流量控制等功能，以对会话层屏蔽通信子网在这些方面的细节与差异。

在传输层中，有两种不同类型的服务，这两种服务同网络层两种服务（前面第 4 章中学习过）一样（即面向连接和无连接服务）。传输层的面向连接服务与网络服务类似，都分为 3 个阶段：建立连接、数据传输和释放连接。在这两个层上，编址和寻址以及流控制方法也是相同的。

另外，传输层的无连接服务与网络层的无连接服务也非常相似。但是传输层的设置是必要的，因为通信子网不能保证服务质量可靠，会出现丢失分组、错序、频繁发送 N-RESET 的情况。

5.1.2 传输层端—端通信的概念

端—端通信指的是在数据传输前，经过各种各样的交换设备，在两端建立一条链路，就像它们是直接相连的一样。链路建立后，发送端就可以发送数据，直至数据发送完毕，接收端确认接收成功。

建立端到端通信链路后，发送端已知接收设备一定能收到，而且经过中间交换设备时不需要进行存储转发，因此传输延迟小。

在发送过程中，发送端的设备一直要参与传输，直到接收端收到数据为止。如果整个传输的延迟很长，那么对发送端的设备造成很大的浪费。另外，在传输过程中，如果接收设备关机或发生故障，那么端到端传输将无法实现。

端到端传输时，一旦传输端确定后，这两端之间可以同时进行多种服务数据的传输，不同的服务数据各自通过不同的服务端口传输，每一对服务端口的连接可以看作一个传输逻辑通道，它们可以共用一个网络连接。即通过一路网络连接实现端到端的多路传输连接。

1．端到端的连接管理

连接管理（Connection Management）是传输层在两个节点间建立和释放连接所必须

遵循的协议。一般可以通过三次握手协议来完成两端点的建立：计算机 A 传送一个请求一次连接的 TPDU，它的序列号是 X；计算机 B 回送一个确认该请求及其序列号的 TPDU，它的序列号为 Y；计算机 A 通过在第一个数据 TPDU 中包含序列号 X 和 Y，对计算机 B 的确认帧发回一个确认。

请求或确认的丢失可能会导致错误的发生。为此让计算机 A 和计算机 B 分别设置定时器，可以解决部分问题。如果计算机 A 的请求或计算机 B 的确认丢失了，计算机 A 将在计时结束后重新发送请求；如果计算机 A 的确认丢失了，计算机 B 将在计时结束后终止连接。

当计算机 A 与计算机 B 通信完毕后，需要两端点终止连接操作。而终止连接的操作可以通过首先计算机 A 请求终止连接，然后计算机 B 确认请求来实现。如果计算机 A 接收到计算机 B 所发送的确认帧后，再发送一个确认帧，并终止连接。在计算机 B 收到确认后，也终止连接。

在数据传输时，传输层将上层交给它的服务数据分解成多个传输层协议数据单元，将多个传输层协议数据单元分别传送到不同的网络节点，这一过程称为向下多路复用（Downward Multiplexing）。几个传输用户共享一个单一节点称为向上多路复用（Upward Multiplexing）。

2．端到端的差错控制

在传输层的通信过程中，无论是面向连接还是面向无连接的传输，都需要对传输的内容进行差错控制编码、差错检测、差错处理 3 个方面的处理。

传输层的差错控制是通过在通信子网对差错控制的基础上最后的一道差错控制措施，面对的出错率相对较低。特别是随着传输介质不断提高，这种出错概率大幅下降，传输的可靠性明显提高。因此，传输层的差错控制编码一般采用比较简单的算法。例如，在传输层协议数据单元（TPDU）内留有专门的校验和字段，用于存放校验码。

在对于差错的处理过程中，一般采用当即纠错、通知发方重传和丢弃 3 种措施。不过采用什么措施与差错控制算法以及传输服务要求有关。

3．端到端的流量控制

传输层的流量控制是对传输层协议数据单元的传送速率的控制。其中包括两个方面，分别在两端进行：在发送端控制传输层协议数据单元的发送速率和在接收端控制传输层协议数据单元的接收速率。也就是在同一对传输通信中，发送和接收的速率是各自独立的，这两端的速率可以是不一样的。传输层协议数据单元的发送与接收的速率取决于两端计算机的发送/接收能力和通信子网的传输能力两种因素。

控制两端计算机收发信息数据单元速率的总策略是采用缓存的办法，即在两端计算机设置用于缓存协议数据单元的缓存器。

缓存的设置策略主要是对于低速突发数据传输，在发方建立缓存。而对于高速平稳的数据传输，为了不增加传输负荷，最大限度地利用传输带宽，在收方建立缓存。缓存的大小可以是固定的也可以是可变的，可以为每一个传输连接建立一个缓存，也可以多个传输连接循环共用一个大的缓存。

4．端到端的拥塞控制

拥塞现象是指到达通信子网中某一部分的分组数量过多，使得该部分网络来不及处理，以致引起这部分乃至整个网络性能下降的现象。

拥塞控制是通过开环控制和闭环控制两种方法来实现的。开环控制是在设计网络时，为力求在网络工作时，使其不产生拥塞。但对于变化多端的网络，使用这种控制方法代价太高，很难实现。所以网络采用比较现实的闭环控制，其实现方法如下。

- 监测网络系统在何时何处发生了拥塞。
- 将拥塞的信息传送到可以采取行动的地方。
- 根据拥塞消息，调整网络系统的运行，解决拥塞。

端到端的拥塞控制就是由网络层将拥塞的信息传送到发送端，由发送端采取措施，控制发往网络的传输数据段数。

5.1.3 网络服务与服务质量

传输层的主要功能可以看作是增加和优化网络层服务质量。如果网络层提供的服务很完备，那么传输层的工作就很容易，否则传输层的工作就较繁重。对于面向连接的服务，传输服务用户在建立连接时要说明可接受的服务质量参数值。在讨论传输层服务质量参数时需要注意以下几个问题。

- 服务质量参数是传输用户在请求建立连接时设定的，表明希望值和最小可接受的值。
- 传输层通过检查服务质量参数可以立即发现其中某些值是无法到达的。传输层可以不去与目的计算机连接，而直接通知传输用户连接请求失败与失败的原因。
- 有些情况下，传输层发现不能达到用户希望的质量参数，但可以到达稍微低一些的要求，然后再请求建立连接。
- 并非所有的传输连接都需要提供所有的参数，大多数仅仅是要求残余误码，而其他参数则是为了完善服务质量而设置的。

传输层根据网络层提供的服务种类及自身增加的服务，检查用户提出的参数，如能满足要求则建立正常连接，否则拒绝连接。服务质量参数包括用户的一些要求，如连接建立延迟、连接建立失败的概率、吞吐率、传输延迟、残余误码率、安全保护、优先级及恢复功能等。

下面将介绍服务质量中的这些参数内容。

- **连接建立延迟** 从传输服务用户要求建立连接到收到连接确认之间所经历的时间，它包括了远端传输实体的处理延迟，连接建立延迟越短，服务质量越好。
- **连接建立失败的概率** 在最大连接建立延迟时间内，连接未能建立的可能性。例如，由于网络拥塞，缺少缓冲区或其他原因造成的失败。
- **吞吐率** 在某个时间间隔内测得的每秒钟传输的用户数据的字节数。每个传输方向分别用各自的吞吐率来衡量。
- **传输延迟** 从源计算机传输用户发送报文开始到目的计算机传输用户接收到报

文为止的时间，每个方向的传输延迟是不同的。

- **残余误码率** 用于测量丢失或乱序的报文数占整个发送的报文数的百分比。理论上残余误码率应为零，实际上它可能是一较小的值。
- **安全保护** 为传输用户提供了传输层的保护，以防止未经授权的第三方读取或修改数据。
- **优先级** 为传输用户提供用以表明哪些连接更为重要的方法。当发生拥塞事件时，确保高优先级的连接先获得服务。
- **恢复功能** 当出现内部问题或拥塞情况下，传输层本身自发终止连接的可能性。

5.1.4 传输层的端口

在网络技术中，端口（Port）大致有两种意思：一是物理意义上的端口，如 ADSL Modem、集线器、交换机、路由器用于连接其他网络设备的接口等；二是逻辑意义上的端口，一般是指 TCP/IP 协议中的端口，端口号的范围从 0 到 65535。

而端口的序号小于 256 称为通用端口，如 FTP 是 21 端口、WWW 是 80 端口等。端口用来标识一个服务或应用。一台主机可以同时提供多个服务和建立多个连接。端口就是传输层的应用程序接口。

但由于 IP 地址只对应因特网中的某台计算机，而 TCP 端口号可对应计算机上的某个应用进程。因此，TCP 模块采用 IP 地址和端口号的对应来标识 TCP 连接的端点。一条 TCP 连接实质上对应了一对 TCP 端点，如图 5-2 所示。

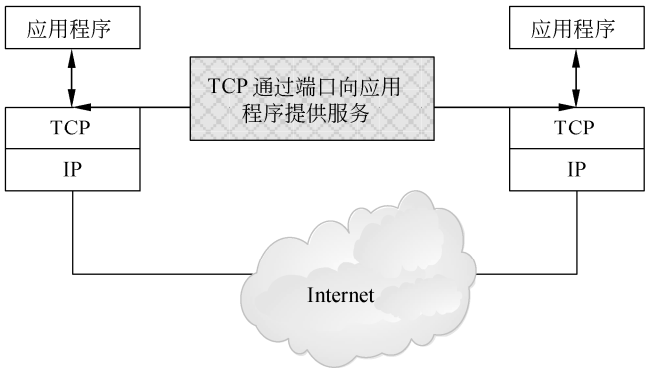

图 5-2 端口作用示意图

端口号实质上也是操作系统标识应用程序的一种方法，其取值可由用户定义或者系统分配。TCP 端口号采用了动态和静态相结合的分配方法，对于一些常用的应用服务（尤其是 TCP/IP 协议集提供的应用服务），使用固定的端口号。例如，电子邮件（SMTP）的端口号为 25，文件传输（FTP）的端口号为 21 等。

对于其他的应用服务，尤其是用户自行开发的应用服务，端口号采用动态分配方法，由用户指定操作系统分配。而在 TCP/IP 约定中，0～1023 为保留端口号，标准应用服务使用。1024 以上是自由端口号，用户应用服务使用，表 5-1 显示了重要的 TCP 端口号。

表 5-1 重要的 TCP 端口号

TCP 端口号	关 键 字	描 述
20	FTP-DATA	文件传输协议数据
21	FTP	文件传输协议控制
23	TELNET	远程登录协议
25	SMTP	简单邮件传输协议

续表

TCP 端口号	关 键 字	描 述
53	DOMAIN	域名服务器
80	HTTP	超文本传输协议
110	POP3	邮局协议
119	NNTP	新闻传送协议

5.2 用户数据报协议

用户数据报协议（User Datagram Protocal，UDP）是 ISO 参考模型中一种无连接的传输层协议，提供面向事务的简单而不可靠的信息传送服务。

5.2.1 UDP 概述

UDP 协议在网络中与 TCP 协议一样用于处理 UDP 数据包。

UDP 协议从问世至今已经被使用了很多年，虽然其最初的光彩已经被一些类似协议所掩盖，但是即使是在今天，UDP 仍然不失为一项非常实用和可行的网络传输层协议。与所熟知的 TCP 协议一样，UDP 协议直接位于 IP 协议的顶层。根据 OSI 参考模型，UDP 和 TCP 都属于传输层协议。

协议的主要作用是将网络数据流量压缩成数据包的形式。一个典型的数据包就是一个二进制数据的传输单位。每一个数据包的前 8 个字节用来包含报头信息，剩余字节则用来包含具体的传输数据。

UDP 协议基本上是 IP 协议与上层协议的接口。UDP 协议适用端口分辨运行在同一台设备上的多个应用程序。与 TCP 不同，UDP 不提供对 IP 协议的可靠机制、流控制以及错误恢复功能等。由于 UDP 比较简单，UDP 头包含很少的字节，所以比 TCP 负载消耗少。

UDP 是无连接的，它所发送的数据之间无须建立连接，减少了开销和发送数据之前的延时；而且 UDP 尽最大努力交付，但它不保证可靠交付，无须维持复杂的链接状态表。除此之外，UDP 还具有支持一对一、一对多、多对一和多对多的交互通信，以及没有拥塞和首部开销小等特点。

5.2.2 UDP 的首部格式

数据报 UDP 具有数据字段和首部字段 2 个字段，首部字段比较简单，只有 8 个字节，由源端口、目的端口、长度和检验和 4 个字段组成，每个字段的长度为 2 个字节。

- **源端口** 表示源端口号，在需要对方回信时选用，不需要时可选用全 0。
- **目的端口** 表示目的端口号，在终点交付报文时使用。
- **长度** 表示 UDP 用户数据报的长度，其最小值为 8（仅有首部）。
- **检验和** 检测 UDP 用户数据报在传输中是否存在错误，如果存在错误则丢弃。

将 UDP 置于 IP 层之上，表示一个 UDP 报文在网络中传输时要封装到 IP 数据报中。最后，网络接口层将数据报封装到一个帧中再进行物理传输通道上的传输。封装过程如

图 5-3 所示。

在 UDP 传输过程中，IP 层的报头包含了源计算机和目的计算机，而 UDP 层的报头包含了源端口号、目的端口号、总长度和校验和，如图 5-4 所示。

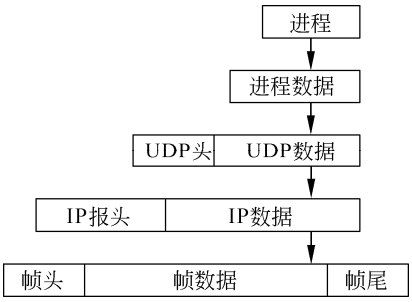

图 5-3 UDP 封装过程

图 5-4 UDP 报头

UDP 适用于不需要 TCP 可靠机制的情形，比如，当高层协议或应用程序提供错误和流控制功能的时候。UDP 是传输层协议，服务于很多知名应用层协议，包括网络文件系统（NFS）、简单网络管理协议（SNMP）、域名系统（DNS）以及简单文件传输系统（TFTP）。

5.3 传输控制协议

传输控制协议（Transmission Control Protocal，TCP）可以提供可靠的端到端的数据传输，也是重要的传输层协议。同时，它也是 TCP/IP 最具代表性的协议。TCP 协议可以提供进程通信能力和可靠性。

5.3.1 TCP 概述

TCP 是一种面向连接（连接导向）的、可靠的、基于字节流的传输层（Transport Layer）通信协议。

在因特网协议族（Internet Protocol Suite）中，TCP 层是位于 IP 层之上，应用层之下的传输层。不同主机的应用层之间经常需要可靠的、像管道一样的连接，但是 IP 层不提供这样的流机制，而是提供不可靠的包交换。

应用层向 TCP 层发送用于网间传输的、用 8 个字节表示的数据流，然后 TCP 把数据流分割成适当长度的报文段（通常受该计算机连接的网络的数据链路层的最大传送单元（MTU）的限制）。

此时，TCP 把结果包传给 IP 层，由它来通过网络将包传送给接收端实体的 TCP 层。TCP 为了保证不发生丢包，就给每个字节一个序号，同时序号也保证了传送到接收端实体的包的按序接收。

然后，接收端实体对已成功收到的字节发回一个相应的确认（ACK）；如果发送端实体在合理的往返时延（RTT）内未收到确认，那么对应的数据（假设丢失了）将会被重传。TCP 用一个校验和函数来检验数据是否有错误；在发送和接收时都要计算和校验。

尽管 IP 协议提供了一种使计算机能够发送数据和接收数据的方法，也就是将分组从信源地址传送到目的地址。但是，必须通过 TCP 协议解决数据报丢失或顺序传递。在学习 TCP 协议之前，先来了解一下 TCP/IP 协议的特点。

- **面向连接的服务** 发送方和接收方分别利用原语创建一个称为套接字的连接端点。也就是为了进行数据传输，首先必须在发方和收方之间建立连接。
- **面向数据流** 两个应用程序相互传输大量数据时，可以将数据划分为字节流。在传输过程中，接收端应用程序收到的字节流顺序同发送端送出的字节流顺序一样。
- **缓冲传输** 当应用程序将数据送给 TCP 实体时，TCP 可能将其缓存起来累加到一定量后，作为一个数据片发送出去，这样可以提高传输效率。对于那些急于发送出去的数据，例如键盘命令，协议提供了一种机制 PUSH，应用程序用 PUSH 标志通知 TCP 软件把当前在缓冲区中的数据立即发送出去。
- **提供可靠性** TCP 采用带重传的肯定确认（Positive Acknowledgement with Retransmission）来进行差错控制和流量控制。TCP 软件对于不按序到达的数据片，进行整理、组装成原报文。TCP 协议必须保证可靠性。
- **全双工连接** TCP 允许在两个方向上同时进行传送。数据流服务允许在一个方向结束数据流动；在另一个方向上，数据却在继续流动。由于是全双工，可以在一个方向的数据流上捎带对相反方向数据流的控制信息，会减轻网络负载。

5.3.2 TCP 报文格式

两台计算机上的 TCP 协议之间传输的数据单元称报文段。通过报文段的交互来建立连接、传输数据、发出确认、通告窗口大小以及关闭连接。TCP 报文格式如图 5-5 所示。

图 5-5 TCP 报文格式

下面分别介绍各段的意义。

- **源端口号（Source Port）** 本地通信端口，支持 TCP 的多路复用机制。
- **目的端口号（Destination Port）** 远地通信端口，支持 TCP 的多路复用机制。
- **序号（Sequence Number）** 数据段的第一个数据字节的序号（除含有 SYN 的段外）。在 SYN 段中，该域是 SYN 的序号，即建立本次连接的初始序号，在该连接上发送的第一个数据字节的序号为初始序号+1。
- **数据偏移（Data Offset）** 指出该段中数据的起始位置，以 4 字节为单位（TCP 头总以 32 位边界对齐）。
- **紧急指针（Urgent Pointer）** 从该段序号开始的一个正向位移，指向紧急数据的最后一个字节。

❑ **控制位（Control Bits）** 控制位共 6 个，如表 5-2 所示。

表 5-2 TCP 报头的码位字段的含义

位（从左到右）的标识	该位置 1 的含义
URG	紧急指针字段可用
ACK	确认字段可用
PSH	请求急迫操作
RST	连接复位
SYN	同步序号
FIN	发送方字节流结束

❑ **确认号（Acknowledgment Number）** 当 TCP 段头控制位中的 ACK 置位时，确认号域才有效。它表示本地希望接收的下一个数据字节的序号。对于收到有效确认号的发送者来说，其值表示接收者已经正确接收到了该序号以前的数据。

❑ **窗口（Window）** 该段的发送者当前能够接收的从确认号开始的最大数据长度，该值主要向对方通告本地接收缓冲区的使用情况。

❑ **校验和（Checksum）** 校验对象包括协议伪头、TCP 报头和数据。

❑ **选项（Options）** 位于 TCP 头的尾端。选项有单字节和多字节两种格式。单字节格式，只有选项类型；多字节格式由一个字节的选项类型、多字节的实际选项数据和一个字节的选项长度（三部分的长度）组成。下面是 TCP 协议必须实现的选项。

选项表尾选项：KIND=0。表示 TCP 头中由全部选项组成的选项表的结束。

无操作选项：KIND=1。该选项可能出现在两个选项之间，作为一个选项分隔符，或提供一种选项字边界对齐的手段，其本身无任何意义。

最大段长选项：KIND=2，LENGTH=4。该选项主要用于通知连接的对方，本地能够接收的最大段长。它只出现在 TCP 的初始建立连接请求中（SYN 段）。如果在 TCP 的 SYN 段中没有给出该选项，就意味着 SYN 段的发送有能力接收任何长度的段。

❑ **填充（Padding）** 当 TCP 头由于含有选项而无法以 32 位边界对齐时，将会在 TCP 头的尾部出现若干字节的全 0 填充。

❑ **保留（Reserved）** 以备后用。

5.3.3 TCP 连接

TCP 是一个面向连接的协议，无论哪一方向另一方发送数据之前，都必须先在双方之间建立一条连接。下面将详细介绍 TCP 连接是如何建立的以及通信结束后是如何终止的。

1．建立连接

TCP 建立连接的过程也被称为“三次握手”过程。如图 5-6 所示，计算机 A 要与计算机 B 建立连接，A 发送第一个握手的报文段，其中 SYN 位置 1，并随机选取一个初始数序号 X，这样告诉计算机 A 自己对数据编号的信息。

计算机B在接收到计算机A发送的请求后返回一个应答报文段，也在其中指出自己的顺序号。

计算机A在接收到B的应答时发送一个确认报文，其中ACK位置1。计算机B在接收到计算机A发送的确认报文段后，连接就成功建立。

通过三次握手，计算机A与计算机B就都做好了传输数据的准备并且交换了一些信息。

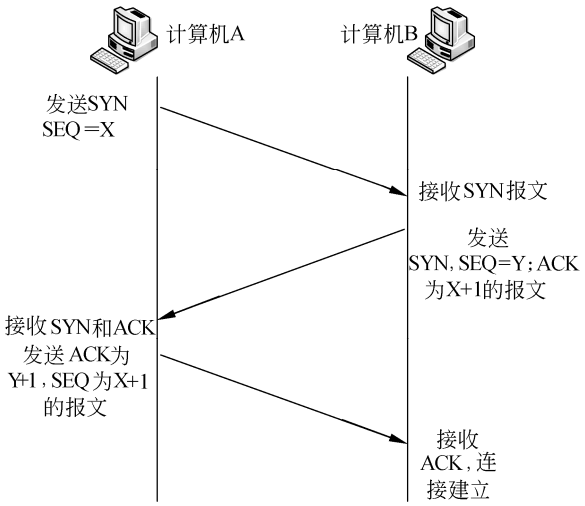

图 5-6 TCP 建立连接

2. 关闭 TCP 连接

当计算机A与计算机B的应用程序完成数据传输后，TCP将关闭连接以释放其所占用的计算机资源。通信双方都可以在数据传输接收后请求释放连接。

如图 5-7 所示，当计算机A要关闭连接时，它将发送一个FIN位置位、序列号为Y的报文段，计算机B在接收到此数据后也将马上发送一个证实信号并通知其上层的应用程序，使其直到对方已关闭连接。

此时，计算机A不再发送任何数据，但是还可以接收从计算机B传送来的数据。当计算机B要停止发送数据时，也发送一个带有FIN位置的报文段给计算机A，以告知计算机A自己要关闭连接，至此TCP连接关闭，双方通信结束。

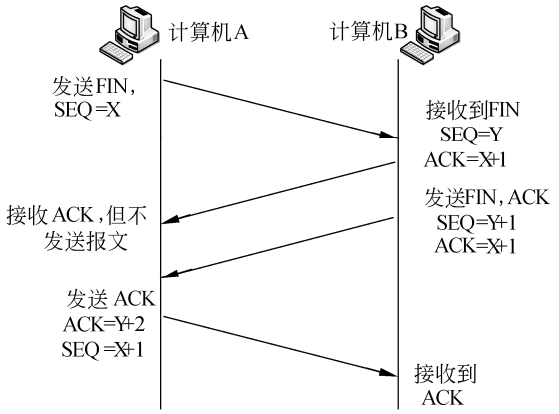

图 5-7 关闭 TCP 连接

提　示

由于TCP是全双工通信，因此只有当接收和发送双方主机都关闭连接时，连接才被真正关闭。只有一方发送关闭连接信号时，则其还能接收对方的数据，直到对方也发出关闭连接的信号为止。

5.4 流量控制

DTE与DCE速度之间存在很大差异，这样在数据的传送与接收过程当中很可能出现收方来不及接收的情况，这时就需要对发方进行控制，以免数据丢失。

5.4.1 停止等待协议

停止等待协议是最简单的流量控制算法（策略），当源主机发送一个帧后，即停止

发送，等待对方的应答。

如果收到目的地主机的肯定应答，则接着发送下一个帧。如果收到否定应答或者超过规定的时间没有收到肯定应答，则重发该帧。它是简单而重要的数据链路层协议，在物理链路层上进行流量控制和差错控制，实现可靠的数据传输。下面通过几个数据传输情况来学习停止等待协议的原理。

如图 5-8(a)所示，源主机发送一个数据帧，而目的地主机收到正确的数据帧，将此帧进行拆装后传送到网络层，然后向源主机返回一个确认帧 ACK。当源主机收到确认 ACK 后，再发送下一个数据帧。

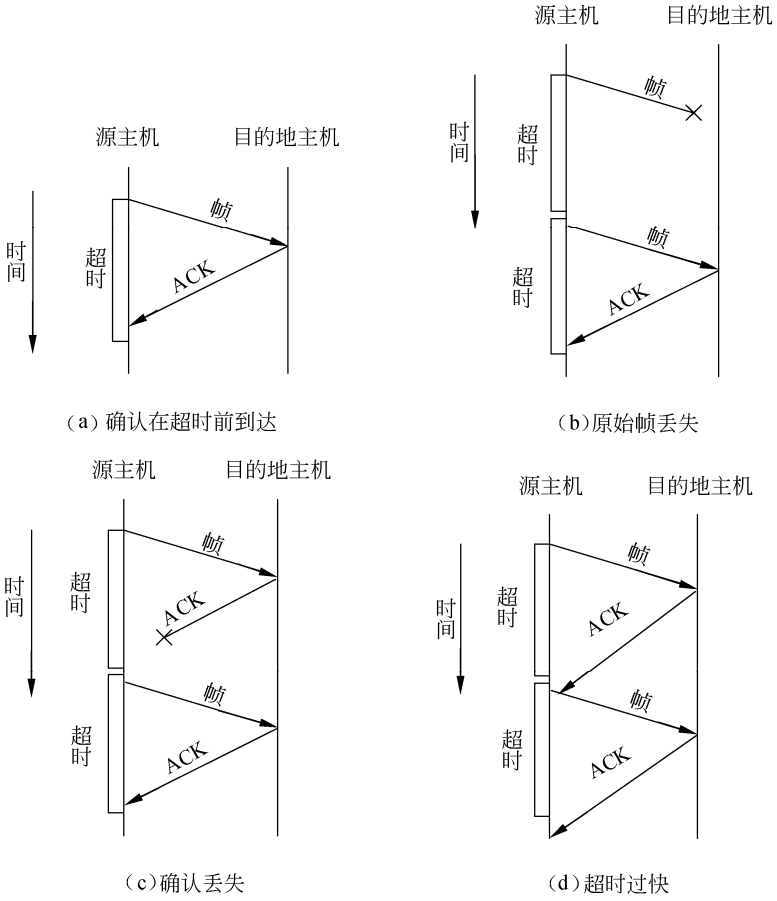

图 5-8 数据帧的几种传输情况

如果数据传输有差错，一般通过出错重发和超时重发机制来解决此问题。数据在传输过程中有差错，目的地主机收到有差错的数据帧以后可以通过检错码检查出错误，于是不向源主机发送确认帧 ACK 或者向其发送否认帧 NAK；为了避免源主机陷入一直等待，使它在发出一个数据帧后就立即启动一个定时器，如果超出了重发时间后，还没有收到目的地主机的确认帧，就重新发送该数据帧。例如，图 5-8（b）和图 5-8（c）分别表示原始帧和 ACK 丢失的情况。

技　巧

在发送数据帧后，重发时间应定得适当，一般选为略大于从发送完毕到收到确认帧所需时间的平均值。数据在发送过程中，如果连续多次重新发送都出现差错，超过一定次数，就停止发送，向上一级报告故障情况。

图 5-8（d）表示源主机正确接收了数据帧，但返回的确认帧丢失的情况。例如，源主机发送数据帧，目的地主机返回确认帧 ACK，但该确认帧在传送过程中丢失或者超时。

因此，源主机收不到确认帧，又重新发送该数据帧，于是目的地主机收到两个同样的数据帧，所以就产生错误。

提　示

“丢失”是帧在传输中出错，目的地主机用差错码检测到这类差错，接着将帧丢弃。

在停止等待算法中有一个重要的细节。假设源主机发送一个帧，并且目的地主机确认它，但这个确认丢失或迟到了，如图 5-8（c）和图 5-8（d）。

在这两种情况下，源主机超时并重发这一个数据帧，但目的地主机却认为这是下一个数据帧，因为它正确地接收并确认了上一个数据帧。这就引起重复传送帧的问题。

为解决这个问题，停止等待协议的头部通常包含 1bit 的顺序号，即顺序号可取 0 和 1，并且每一帧交替使用顺序号，如图 5-9 所示。

因此，当源主机重发帧 0 时，目的地主机可确定它是一个帧 0 的重复帧，而不是帧 1，因此可以忽略它（目的地主机仍确认它）。

停止等待算法的主要缺点是，它允许源主机每次在链路上只有一个未确认的帧，这可能远远低于链路的容量。例如，考虑一条往返时间为 45ms 的 1.5Mb/s 链路，这条链路的延迟与带宽的乘积为 67.15Kb，或近似为 8KB。由于源主机每个 RTT 仅能发送一个数据帧，假设 1 帧的大小为 1KB，则最大发送速率为

BistsPerFrame÷TimePerFrame

=1024×8÷0.045

=182Kb/s

或者大约是链路容量的 1/8。所以为完全利用链路，源主机在必须等待一个确认之前最多能够发送 8 帧。

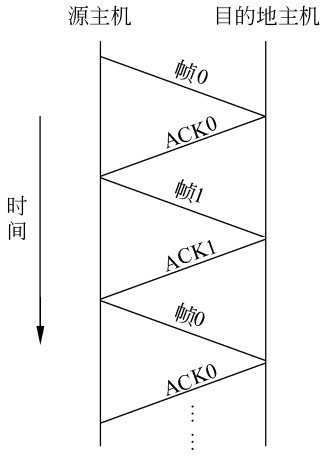

图 5-9　具有 1bit 顺序号的停止等待算法的时间线

5.4.2　滑动窗口协议

导致停—等协议信道利用率低的原因，是源主机每发送一个数据帧都需要等待目的地主机的应答，如果能允许源主机连续不断地发送数据帧，而不必每一个数据帧都等待应答，那么显然可以提高传输效率。

允许源主机连续发送多个数据帧而不需要等待目的地主机应答的算法（策略），称之为窗口机制（滑动窗口协议）。窗口机制除了提高效率以外，还满足了流量控制、差错控制等数据链路层的基本要求。

在所有的滑动窗口协议中，每一个要发出的数据帧都包含一个序列号，范围是从 0 到 2^n-1，因而序列号能恰好放入 n 位的字段中。停—等滑动窗口协议使用 n=1，限制序列号为 0 和 1，但是复杂的协议版本则使用任意值 n。例如，设 n=3，序号空间为 0 至 7（即 8 个序号）。

当数据帧发送没有得到确认信息时，需要对连续发送的数据帧的数目进行限制。导致这个问题的两个因素：一是未得到确认的数据帧太多，一旦出现错帧，就要重发已经发出的多个数据帧；二是连续发送的数据帧的数量大，编号占用的比特就多，使数据帧的额外开销增加。

下面介绍限制连续发送数据帧数据的方法。

1．发送窗口

把源主机允许连续发送而未得到确认的一组数据帧的序号集合称为发送窗口。发送端可以连续发送而未得到确认的数据帧的最大数目，称为发送窗口的尺寸。发送窗口尺寸的确定与所选用的协议有关。源主机每发送一个新数据帧，都要检查它的序号是否落在发送窗口之内。

注　意

发送窗口不是序号空间。序号空间是可以使用的序号的范围，如果用 n 比特表示帧的序号，则帧的序号范围可以从 0 到 2^n-1；而发送窗口是源主机允许连续发送的未得到确认的一组帧的序号集合，它显然应该是帧序号空间的一个子集。

源主机最早发送但还未收到应答的数据帧的序号，称之为发送窗口的后沿；发送窗口后沿加上窗口尺寸再减 1，称之为发送窗口的前沿。如果发送窗口尺寸为 m，则初始时源主机可以连续发送 m 个数据帧，这些数据帧都因出错或者丢失而需要重发，所以要设置 m 个发送缓冲区来存放 m 个数据帧的副本(例如,一个缓冲区可以存放一个数据帧)。

源主机收到发送窗口后沿所对应的帧的肯定应答后，就将发送窗口向前滑动一个序号，并从发送缓冲区中将该数据帧的副本删除。如果有新数据帧要发送，要对其按顺序进行编号，只要数据帧序号落在发送窗口之内就可发送，直至发送窗口被占满为止。

2．接收窗口

一组目的地主机允许接收的数据帧序号的集合称为接收窗口。目的地主机最多允许接收的数据帧数目称之为接收窗口尺寸。接收窗口的上、下界分别称之为接收窗口的前、后沿。

目的地主机每收到一个数据帧，都要判断该帧是否落在接收窗口之内。如果数据帧的序号正好等于接收窗口的后沿，且经过检验正确，就将该数据帧的数据部分传送到网络层实体，并向源主机发送返回一个应答数据帧，同时使接收窗口向前滑动一个序号。

如果收到了序号不等于接收窗口后沿的数据帧，要看接收窗口尺寸的大小是否等于 1，若接收窗口的尺寸等于 1，则表示接收方只能按顺序接收，就将接收到的数据帧直接丢弃，不做任何处理；若接收窗口尺寸大于 1，则首先检验该数据帧是否正确，如果正确就暂时将它保留在接收缓冲区中，并返回应答。

然后，继续等待序号为接收窗口后沿的数据帧，直到正确地接收到接收窗口后沿的数据帧，才将其连同前面保留在接收缓冲区中的正确的数据帧按顺序送给上层，同时滑动接收窗口。

对于接收到的落在接收窗口之外的数据帧，直接丢弃即可，不需做任何处理。由此可见，无论接收窗口尺寸的大小如何，接收方交给上层的数据总是按顺序的。

发送窗口尺寸不一定等于接收窗口尺寸，窗口大小在一些协议中是固定的，但在另一些协议中是可变的。

发送窗口内的各数据帧，在传输过程中有可能丢失或者损坏，所以所发送的数据帧需要在缓冲区中保存以备重传。如果缓冲区满，就停止接收网络层的分组，直到有空闲

缓冲区。

5.5 TCP 的拥塞控制

拥塞是由于路由器超载而引起的严重延迟现象，是通信子网能力不足的表现。一旦发生拥塞，路由器便丢弃数据包，并导致发送方重传被丢弃的报文，而大量的重传报文又会进一步加剧拥塞，这种恶性循环有可能导致整个因特网无法工作。而简单地采用确认重传技术并不能解决传输层的所有问题，TCP 还必须提供适当机制以进行拥塞控制。

5.5.1 了解拥塞控制

TCP 的拥塞控制方法也是基于滑动窗口协议的。它通过限制发送方注入报文的速率而达到拥塞控制的目的。具体地说，TCP 是通过控制发送窗口的大小来对拥塞进行响应。

而决定发送窗口大小的因素有两个：第一个因素是接收方所通告的窗口大小；第二个因素是发送方的拥塞窗口限制，又叫拥塞窗口。发送窗口的大小是取二者之中的较小者。

在通信子网没有发生拥塞的情况下，发送方的拥塞窗口和接收方的通告窗口大小相等。一旦发现拥塞，发送方立即减小拥塞窗口。

TCP 协议发现拥塞的途径有两条：一条途径是因特网控制信息协议 ICMP 的源抑制报文，另一条途径是报文丢失现象。TCP 假定大多数报文丢失其原因都是通信子网拥塞。

为了迅速抑制拥塞，TCP 使用了两种技术：快速递减和慢启动。这两种技术是相关联的，实现起来也比较容易。所谓快速递减拥塞窗口的策略指的是：一旦发现报文丢失，立即将拥塞窗口大小减半；而对于保留在发送窗口中的报文，按指数增加重传定时器的定时宽度。

换句话说，当可能出现拥塞时，TCP 对传输流量和重传速率都按指数级递减。如果继续出现报文丢失，最终 TCP 将数据传输流量限制到每次只发送一个报文，即变成简单停—等协议。快速递减策略的意图是迅速而显著地减少注入通信子网的传输流量，以便路由器有足够的时间来清除在其发送队列中的数据包。

5.5.2 拥塞控制方法

拥塞结束之后，TCP 应如何恢复数据传输能力呢？TCP 协议采取一种“慢启动”技术来避免系统在流量为零和拥塞之间剧烈振荡。

所谓“慢启动”是指在新建连接或拥塞之后的流量增加，都仅以 1 个报文作为拥塞窗口的初始值，之后每收到一个确认，将拥塞窗口大小加大 1 倍。

“慢启动”技术使得因特网不会在拥塞之后或新的连接建立时，被突然增加的数据流量淹没。当建立连接时，发送方将拥寒窗口大小初始化为该连接所用的最大数据段的长度值，并随后发送一个最大长度的数据段。

如果数据段在定时器超时之前得到了确认，那么就可认为网络的传输能力大于最大的数据段长度。发送方会在原拥塞窗口的基础上再增加一个数据段的字节值，使其为两

倍最大数据段的大小，然后发送两个数据段。

当这些数据段中的每一个都被确认后，拥塞窗口大小就再增加一倍，即 4 个最大数据段的大小，然后发送 4 个数据段。当拥塞窗口是 *M* 个数据段的大小时，如果发送的所有 *M* 个数据段都被及时确认，那么将拥塞窗口大小增加 *M* 个数据段所对应的字节数目。

实际上，每次成功地得到确认都会使拥塞窗口的大小加倍，拥塞算法过程如图 5-10 所示。此处，最大的数据段长度为 1KB。假设开始时拥塞窗口为 64KB，但此时出现了超时，所以将临界值设置为 32KB，传输号 0 的拥塞窗口为 1KB。之后拥塞窗口按指数规律增大至临界值（32KB），并由此开始按线性规律增大。

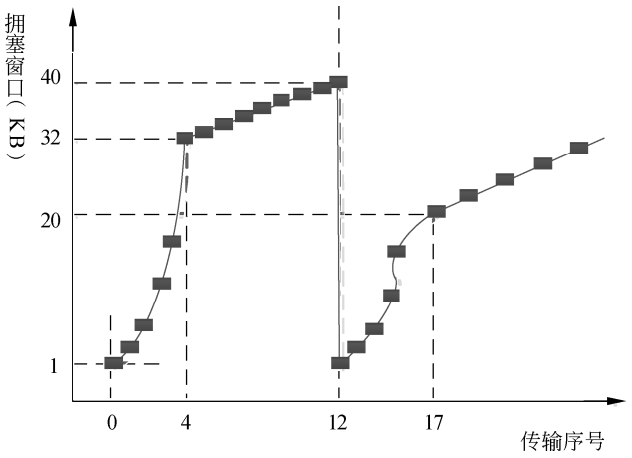

图 5-10 **Internet 拥塞算法的一个实例**

传输序号为 12 时，传输效果非常不顺利，而出现了定时器超时，临界值被设置为当前窗口的 1/2（如当前为 40KB，所以 1/2 应为 20KB）并且慢速启动又从头开始。在传输序号为 17 时，前面的 4 次传输每次均是按加倍的增量增大拥塞窗口。但这之后，窗口又将按线性增大。如果一直不出现超时现象，拥塞窗口会一直增大到接收方窗口的大小。

此时，拥塞窗口停止增大，只要不出现超时，并且接收方窗口也保持不变，则拥塞窗口保持不变。

“慢启动”技术在理想的情况下并不慢。假如 TCP 把拥塞窗口大小初始化为 1，发送一个报文后等待确认报文。当确认报文到达后，TCP 就把拥塞窗口增加为 2，并发送 2 个报文之后又等待确认。如果这两个报文的确认到达之后，拥塞窗口就增加到 4，于是就可以连续发送 4 个报文并又等待确认。收到对它们的确认后拥塞窗口就增加到 8。在 4 个往返时间之后，TCP 就可以连续发送 16 个报文。即便对更大的窗口而言，仅需 2*N* 个往返时间就可以连续发送 *N* 个报文。

为避免拥塞窗口增大速度过快导致可能的拥塞，TCP 还附加一个限制。即当拥塞窗口增大到拥塞时窗口大小的一半时，TCP 进入拥塞避免状态，降低窗口增大的速度。在拥塞避免状态，即使发送窗口中所有的报文都被确认之后，拥塞窗口的大小也只能加 1。

把快速递减、慢启动、拥塞避免、对往返时间变化的测量以及按指数规律对重传定时器进行补偿等技术结合在一起，就能在不明显增加 TCP 协议栈运行开销的情况下显著地提高 TCP 的性能。

5.6 课堂练习：使用网络共享软件

iSHARE 是一款文件共享软件，也是新一代通用下载工具。它不仅可以实现局域网之间的文件共享，也可以下载一些影视、音乐、游戏等多媒体资源。在本练习中，将详细介绍安装和使用 iSHARE 网络共享软件的操作方法和步骤。

操作步骤：

1 下载并解压 iSHARE 软件，双击运行程序，会自动弹出【参数配置】对话框，输入相应的参数，如图 5-11 所示。

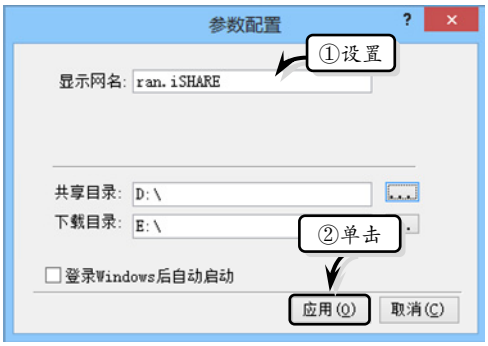

图 5-11 设置配置参数

2 此时，在 iSHARE 窗口中双击左侧的网名，将在右侧的列表框中显示共享文件，如图 5-12 所示。

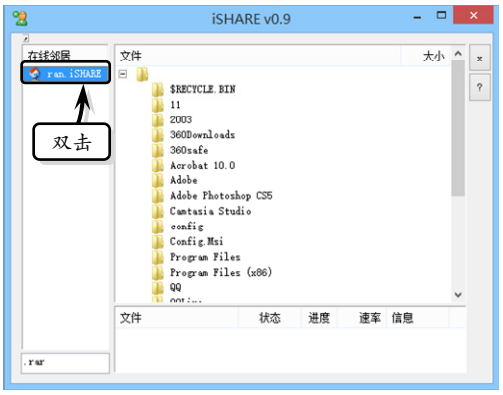

图 5-12 显示共享文件

3 选择右侧列表框中的一个文件夹，右击执行【下载】按钮，即可下载该文件，如图 5-13 所示。

4 右击【任务底栏】中下载完毕的文件，执行【打开文件】命令。在新窗口中，查看文件，如图 5-14 所示。

图 5-13 下载文件

图 5-14 查看下载文件

提 示

如何精简图片的大小呢，一般来说，图片颜色较少、色调平板均匀以及颜色在 256 色以内的最好把它处理成 GIF 图像格式，如果是一些色彩比较丰富的图片，如扫描的照片，最好把它处理成 JPG 图像格式，因为 GIF 和 JPG 各有各的压缩优势，应根据具体的图片来选择压缩比。

5.7 课堂练习：对等网聊天

对等网是网络中最简单的局域网，除了可以共享文件或打印机等网络资源之外，还可以使用专门的软件，实现对等网中的聊天功能。在本练习中，将通过聊天软件 WinPopupX，详细介绍在对等网中实现聊天功能的操作方法和实用技巧。

操作步骤：

1 运行 WinPopupX 安装软件，在弹出的【选择安装语言】对话框中，选择【中文（简体）】语言，并单击【确定】按钮，如图 5-15 所示。

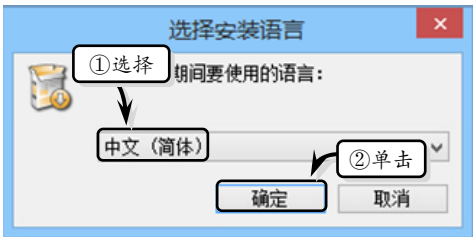

图 5-15 选择安装语言

2 在弹出的【安装向导-WinPopupX】对话框中，查看安装建议，并单击【下一步】按钮，如图 5-16 所示。

图 5-16 查看安装建议

3 在向导中的【许可协议】列表中，选中【我接受协议】选项，并单击【下一步】按钮，如图 5-17 所示。

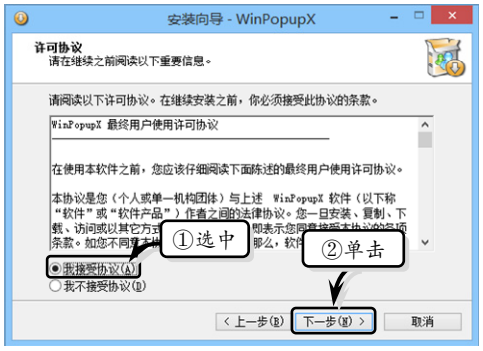

图 5-17 接受许可协议

4 在向导对话框中的【选择目标位置】列表中，更改安装位置，并单击【下一步】按钮，如图 5-18 所示。

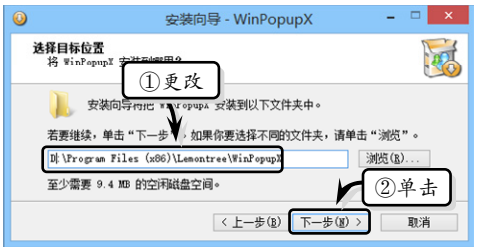

图 5-18 设置安装位置

5 在向导对话框中的【选择开始菜单文件夹】列表中，设置开始菜单文件夹，并单击【下一步】按钮，如图 5-19 所示。

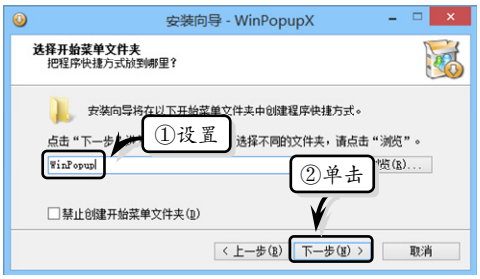

图 5-19 设置菜单文件夹

6 在向导对话框中的【选择附加任务】列表中，启用【创建桌面图标】复选框，并单击【下一步】按钮，如图 5-20 所示。

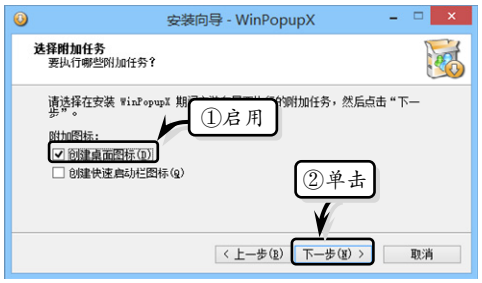

图 5-20 选择附加任务

7 在【准备安装】列表中查看安装信息，单击【安装】按钮，开始安装软件，如图 5-21 所示。

8 运行 WinPopupX 软件，在主界面中将显示本机信息和在线用户列表，如图 5-22 所示。

9 展开【我的好友】列表，双击列表中的计算机名称，在弹出的聊天窗口中输入聊天信

息，如图 5-23 所示。

图 5-21 安装软件

图 5-22 WinPopup 主界面

图 5-23 输入文字信息

10 单击聊天窗口中的【表情】下拉按钮，在其下拉列表中选择一种表情符号，添加表情图案，如图 5-24 所示，并单击【发送】按钮，发送信息。

11 在聊天窗口中，单击【插入图片】按钮，在弹出的【打开】对话框中选择图片文件，单击【打开】按钮，并单击【发送】按钮，如图 5-25 所示。

图 5-24 添加表情图案

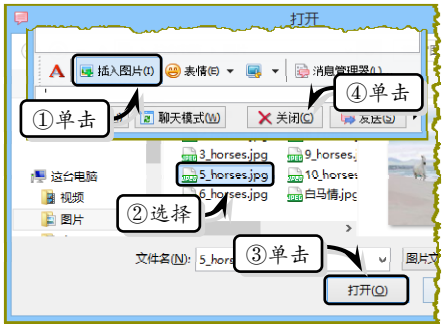

图 5-25 插入图片

12 单击聊天对话框中的【消息管理器】按钮，可在弹出的【消息管理器】对话框中查看、刷新和压缩消息，如图 5-26 所示。

图 5-26 查看消息

5.8 思考与练习

一、填空题

1. 在 IP 互联网中，_____和_____是传输层最重要的两种协议，它们为上层用户提供不同级别的通信可靠性。

2. TCP 可以提供_____服务，UDP 可

以提供_____服务。

3．在 TCP/IP 体系中，根据所使用的协议是 TCP 或 UDP，分别称之为_____或_____。

4．TCP 连接包括_____、_____和_____3 个过程。

5．TCP 的拥塞控制有_____、_____、_____和_____4 个阶段。

6．TCP 通过_____提供连接服务，最后通过连接服务来接收和发送数据。

7．TCP 是因特网中的传输层协议，使用_____协议建立连接。

8．传输层是 OSI 中最重要、最关键的一层，是唯一负责总体的_____和_____的一层。

9．传输层的任务是根据通信子网的特征，最佳地利用_____，为两个端系统的会话层之间，提供建立、维护和取消传输连接的功能，负责端到端的可靠数据传输。

10．传输层利用网络层提供的服务，并通过传输层_____提供给高层用户传输数据的通信端口，使系统间高层资源的共享不须考虑数据通信方面和不可靠的数据传输方向的问题。

二、选择题

1．在 TCP/IP 参考模型中，传输层的主要作用是在互联网络的源主机与目的主机对等实体之间建立用于会话的_____。

A．点到点连接
B．操作连接
C．端到端连接
D．控制连接

2．下列协议中属于面向连接的是_____。

A．IP
B．UDP
C．DHCP
D．TCP

3．关于 TCP 和 UDP 端口，下列说法中正确的是_____。

A．TCP 和 UDP 分别拥有自己的端口号，二者互不干扰，可以共存于同一台主机
B．TCP 和 UDP 分别拥有自己的端口号，但二者不能共存于同一台主机
C．TCP 和 UDP 的端口号没有本质区别，二者互不干扰，可以共存于同一台主机
D．TCP 和 UDP 的端口号没有本质区别，但二者不能共存于同一台主机

4．下列说法错误的是_____。

A．用户数据报协议 UDP 提供了面向非连接的、不可靠的传输服务
B．由于 UDP 是面向非连接的，因此它可以将数据直接封装在 IP 数据报中进行发送
C．在应用程序利用 UDP 协议传输数据之前，首先需要建立一条到达主机的 UDP 连接
D．当一个连接建立时，为连接的每一端分配一块缓冲区来存储接收到的数据，并将缓冲区的尺寸发送给另一端

5．TCP 使用_____进行流量控制。

A．三次握手法
B．窗口控制机制
C．自动重发机制
D．端口机制

6．下面协议被认为是面向非连接的传输层协议的是_____。

A．IP
B．UDP
C．TCP
D．RIP

7．为了保证连接的可靠性，TCP 通常采用_____。

A．三次握手法
B．窗口控制机制
C．自动重发机制
D．端口机制

8．在 TCP/IP 协议中，UDP 协议工作在_____。

A．应用层
B．传输层
C．网络互联层
D．网络接口层

9．一条 TCP 连接的建立过程包括_____个步骤。

A．2
B．3
C．4
D．5

10．一条 TCP 连接的释放过程包括_____

个步骤。

A．2

B．3

C．4

D．5

11．TCP/IP 的传输层协议使用_____形式将数据传送给上层应用程序。

A．IP 地址

B．MAC 地址

C．端口号

D．套接字地址

12．传输层提供的服务使高层的用户可以完全不考虑信息在物理层、_____通信的具体细节，方便用户使用。

A．数据链路层

B．数据链路层的两个子层

C．数据链路层和网络层

D．网络层

三、问答题

1．什么是三次握手法？

2．如何理解端到端通信？

3．TCP 和 UDP 之间的区别是什么？

4．为什么 TCP 协议对每个 TCP 数据字节都要进行编号？

四、上机练习

1．查看当前计算机的名称

hostname 命令用于显示或设置系统的主机名。用户若需要查看当前计算机的名称，可以单击【开始】按钮，执行【运行】命令，在弹出的【运行】对话框中，输入 cmd 命令。然后，在弹出的 MS-DOS 窗口中，输入 hostname 命令，即可查看到当前计算机的名称，如图 5-27 所示。

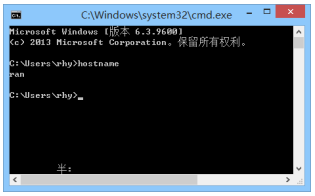

图 5-27　查看当前计算机的名称

2．清空高速缓存中的网页

当访问一个网站时系统将从 DNS 缓存中读取该域名所对应的 IP 地址，为了提高网速，可以将网页中的缓冲网页删除。

在 IE 浏览器中，执行【工具】|【Internet 选项】命令。在弹出的对话框中，激活【常规】选项卡，并单击【删除】按钮。在弹出的【删除浏览历史记录】对话框中，单击【删除】按钮即可，如图 5-28 所示。

图 5-28　删除网页

然后，右击【开始】按钮，执行【命令提示符】命令，在弹出的【命令提示符】窗口中，输入命令 ipconfig/flushdns，按下 Enter 键即可，如图 5-29 所示。

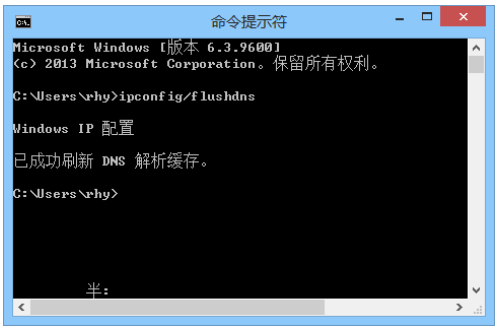

图 5-29 删除高速缓存中的网页