第 3 章　计算机网络体系结构

本章主要介绍计算机网络体系结构。通过本章的学习，读者应能够:

- 了解开放系统互连参考模型中的若干重要概念
- 熟悉和掌握 OSI/RM 各层协议的功能及基本原理

3.1　网络体系结构概述

计算机网络是一个非常复杂的系统，要做到有条不紊地交换数据，每个节点必须要遵守一些事先约定好的规则才能高效协调地工作。这些为进行网络中的数据交换而建立的规则、标准或约定就称为网络协议，网络协议是计算机网络不可缺少的组成部分。早在最初的 ARPANET 设计时，对于非常复杂的网络协议就提出了分层结构处理的方法。分层处理带来的好处是：每一层可以实现一种相对独立的功能，因而可将一个难以处理的复杂问题分解为若干较容易处理的较小的问题。计算机网络协议采用层次结构，可以使各层之间相对独立，灵活性好，易于实现和维护，而且各层在结构上可以分割开，每层都可以采用最合适的技术来实现。由于每层的功能和所提供的服务都已经有了比较明确的描述，所以能够促进体系结构的标准化工作。计算机网络的体系结构（Architecture）是指这个计算机网络及其部件所应完成功能的一组抽象定义，是描述计算机网络通信方法的抽象模型结构，一般是指计算机网络的各层及其协议的集合。

协议的关键成分有：

（1）语法（Syntax）。语法包括数据格式、编码及信号电平等。

（2）语义（Semantics）。语义包括用于协调同步和差错处理的控制信息。

（3）时序（Timing）。定时包括速度匹配和排序。

1. OSI 基本参考模型

随着计算机网络技术的发展，其形式出现了多样化、复杂化，也出现了很多新问题，其中最突出的问题是不同体系结构的网络很难互连起来（即所谓的异种机连接问题）。为了更加充分地发挥计算机网络的效益，必须使不同厂家生产的计算机网络设备能够互相通信，于是越来越需要制定一个国际范围的标准，以便今后生产的网络设备尽可能遵循统一的体系结构标准。1977 年 3 月，国际标准化组织 ISO 的技术委员会 TC97 成立了一个新的技术分委会 SC16 专门研究“开放系统互连”，并于 1983 年提出了开放系统互连参考模型，即著名的 ISO 7498 国际标准（我国相应的国家标准是 GB 9387），记为 OSI/RM。开放系统互连（Open Systems Interconnection）的目的是使世界范围内的应用系统能够开放式地进行信息交换。“开放”是指只要遵循 OSI 标准，一个系统就可以和位于世界上任何地方的也遵循同一标准的其他任何系统进行通信。开放系统和开放系统互连参考模型都是抽象的概念。

在 OSI 中采用了三级抽象：参考模型（即体系结构）、服务定义和协议规范（即协议规格说明），自上而下逐步求精。OSI/RM 并不是一般的工业标准，而是一个为制定标准用的概念性框架。经过各国专家的反复研究，在 OSI/RM 中，采用了如图 3-1 所示的 7 层参考模型。表 3-1 中给出各层主要功能的简略描述，更准确的概念将在以后的有关章节中展开。

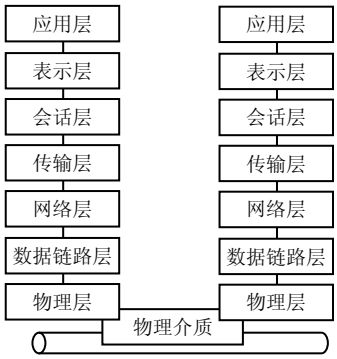

图 3-1　OSI 参考模型

表 3-1　OSI/RM 七层协议模型

层号	名称	英文名称	主要功能简介
7	应用层	Application Layer	作为与用户应用进程的接口，负责用户信息的语义表示，并在两个通信者之间进行语义匹配，它不仅要提供应用进程所需要的信息交换和远地操作，而且还要作为互相作用的应用进程的用户代理来完成一些为进行语义上有意义的信息交换所必需的功能
6	表示层	Presentation Layer	对源站点内部的数据结构进行编码，形成适合于传输的比特流，到了目的站再进行解码，转换成用户所要求的格式并保持数据的意义不变。主要用于数据格式转换
5	会话层	Session Layer	提供一个面向用户的连接服务，它给合作的会话用户之间的对话和活动提供组织和同步所必需的手段，以便对数据的传送提供控制和管理。主要用于会话的管理和数据传输的同步
4	传输层	Transport Layer	从端到端经网络透明地传送报文，完成端到端通信链路的建立、维护和管理
3	网络层	Network Layer	分组传送、路由选择和流量控制，主要用于实现端到端通信系统中中间节点的路由选择
2	数据链路层	Data Link Layer	通过一些数据链路层协议和链路控制规程，在不太可靠的物理链路上实现可靠的数据传输
1	物理层	Physical Layer	实现相邻计算机节点之间比特数据流的透明传送，尽可能屏蔽掉具体传输介质和物理设备的差异

2. *OSI 层次结构模型中的数据流动过程*

OSI 层次结构模型中数据的实际传送过程如图 3-2 所示。图中发送进程送给接收进程的数据，实际上是经过发送方各层从上到下传递到物理介质的；通过物理介质传输到接收方后，再经过从下到上各层的传递，最后到达接收进程。

在发送方从上到下逐层传递的过程中，每层都要加上适当的控制信息，即图中的 H7、

H6、...、H1，统称为报头；在数据链路层还要加上尾部数据 T2 进行校验及差错控制。到最底层成为由“0”或“1”组成的数据比特流，然后再转换为电信号在物理介质上传输至接收方。接收方在向上传递时过程正好相反，要逐层剥去发送方相应层加上的控制信息。

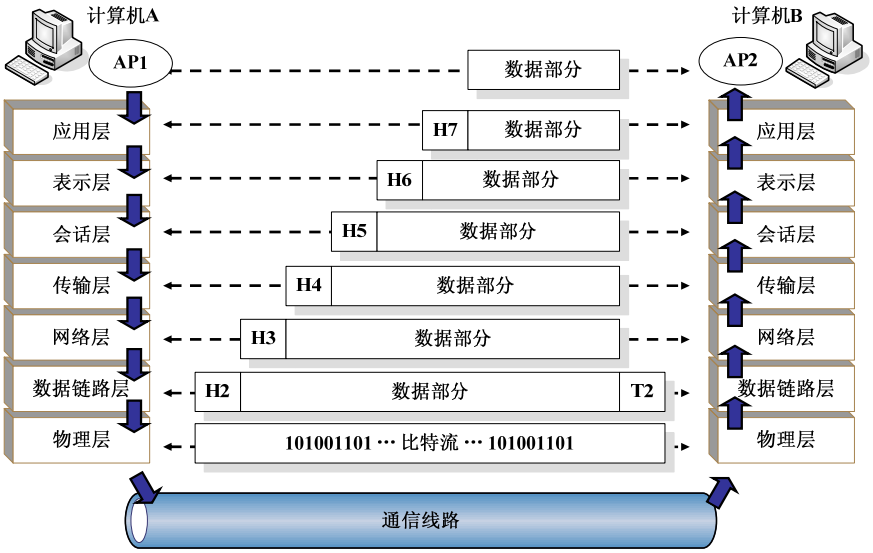

图 3-2　数据的实际传送过程

因接收方的每一层不会收到下层的控制信息，而高层的控制信息对于它来说是透明的数据，所以它只阅读和去除本层的控制信息，并进行相应的协议操作。发送方和接收方的对等实体看到的信息是相同的，就好像这些信息通过虚拟通信信道直接传给了对方一样。

3.2　物理层

3.2.1　物理层的功能

物理层协议是各种网络设备进行互连时的最低层协议。它的目的是在两个网络物理设备之间提供透明的二进制位流传输，尽可能屏蔽掉具体传输介质和物理设备的差异。需要注意的是，物理层并不是指连接计算机的具体物理设备或传输介质，如双绞线、同轴电缆、光纤等，而是要使其上面的数据链路层感觉不到这些差异。这样可使数据链路层只需要考虑如何完成本层的协议和服务，而不必考虑具体网络传输介质的差异。

国际标准化组织（ISO）在其“开放系统互连”的 7 层参考模型中对物理层有如下定义：“物理层为启动、维护和释放数据链路实体之间二进制位传输而进行的物理连接提供机械的、电气的、功能的和规程的特性。这种物理连接可以通过中间系统，每次都在物理层内进行中继的二进制位传输。这种物理连接允许进行全双工或半双工的二进制位流传输。物理服务数据单元（即二进制位）的传输可以通过同步方式或异步方式进行。”

上述定义的要点是物理层主要负责在物理链路上传输二进制位流，提供为建立、维护和拆除物理链路所需要的机械的、电气的、功能的和规程的特性。

（1）机械特性：说明接口所用接线器的形状和尺寸、引线数目和排列等。

（2）电气特性：说明接口电缆线上什么样的电压表示 1 或 0。

（3）功能特性：说明某条线上出现的某一电平的电压表示何种意义。

（4）规程特性：说明对于不同功能的各种可能事件的出现顺序及各信号线的工作规则。

3.2.2 DTE 和 DCE

DTE（Data Terminal Equipment，数据终端设备）是具有一定数据处理能力及发送和接收数据能力的设备。DTE 可以是一台计算机或终端，也可以是各种 I/O 设备。大多数数据处理终端设备的数据传输能力有限，如果将相距很远的两个 DTE 设备直接连接起来，往往不能进行通信，必须在 DTE 和传输线路之间加上一个称为数据电路端接设备（Data Circuit-terminating Equipment，DCE）的中间设备。DCE 的作用就是在 DTE 和传输线路之间提供信号变换和编码的功能，并且负责建立、保持和释放数据链路的连接。典型的 DCE 是与模拟电话线路相连接的调制解调器。如图 3-3 所示为 DTE 通过 DCE 相连的典型情况。

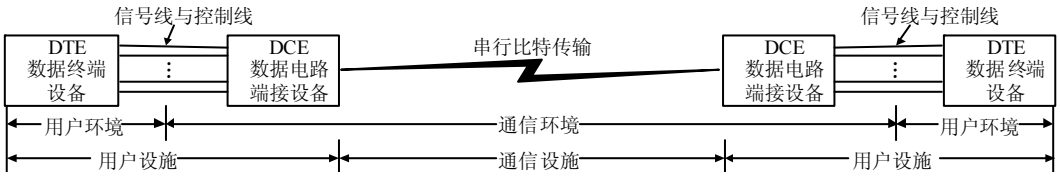

图 3-3 DTE 通过 DCE 与通信传输线路相连

DTE 和 DCE 之间的接口一般有许多条线，包括各种信号线和控制线。DCE 将 DTE 传送过来的数据，按比特逐个顺序地发往传输线路，或者从传输线路上顺序接收串行比特流，然后再交给 DTE。

3.2.3 物理层接口标准

为了提高兼容性，必须对 DTE 和 DCE 的接口进行标准化，这种接口标准就是所谓的物理层协议。下面对几个最常用的物理层标准加以介绍。

1. EIA-232-E/V.24 接口标准

EIA-232-E 是美国电子工业协会（Electronic Industries Association，EIA）制定的著名的 DTE 和 DCE 之间的物理层接口标准，它的前身是 1969 年 EIA 制定的 RS-232-C 标准接口。这里 RS 表示“推荐标准”（Recommended Standard），232 是标识号，C 是标准 RS-232 以后的第三个修订版本，它用于把 DTE 设备连接到音频范围内的调制解调器，数据传输率为 0kbps～20kbps。事实上，RS-232-C 早已成为各国厂家广泛使用的国际标准，1987 年 1 月，RS-232-C 经修改后，正式改名为 EIA-232-D，1991 又修订为 EIA-232-E。由于标准修改得并不多，因此现在很多厂商仍沿用旧的名称，有时甚至简称为“提供 232 接口”。EIA-232-E 接口标准的数据传输速率最高为 20kbps，连接电缆的最大长度不超过 15m。

物理层标准 EIA-232-E 的一些主要特点如下：

（1）机械特性。EIA-232-E 遵循 ISO 2110 关于插头座的标准，使用 25 根引脚的 DB-25 插头座，它的两个固定螺丝中心之间的距离为 47.04±0.17 mm，其他方面的尺寸也都有详细的规定，DTE 上安装带插针的公共接头连接器，DCE 上安装带插孔的母接头连接器，其引脚编号如图 3-4 所示，引脚分为上、下两排，分别有 13 根引脚和 12 根引脚，当引脚指向人的方

向时，从左到右其编号分别为 1～13 和 14～25。

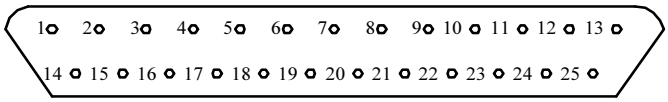

图 3-4　EIA-232-E 25 根引脚编号图

（2）电气特性。EIA-232-E 与 CCITT 的 V.28 建议书一致，采用负逻辑，此时逻辑 0 相当于对信号地线有+5～+15V 的电压，而逻辑 1 相当于对信号地线有−5～−15V 的电压。逻辑“0”相当于数据“0”（空号）或控制线的“接通”状态；逻辑“1”相当于数据“1”（传号）或控制线的“断开”状态。当连接电缆线的长度不超过 15m 时，允许数据传输速率不超过 20kbps。EIA-232-E 所规定的电压范围对过去广泛使用的晶体管电路很适合，但却远远超过了目前大部分芯片所使用的 5V 电压，这点必须加以注意。

（3）功能特性。EIA-232-E 的功能特性与 CCITT 的 V.24 建议书一致。它规定了什么电路应当连接到 25 根引脚中的哪一根以及该引脚信号线的作用。如图 3-5 所示是最常用的 10 根引脚信号线的作用，其余的一些引脚可以空着不用。在某些情况下，可以只用图 3-5 中的 9 根引脚（振铃指示 RI 信号线不用），这就是常见的 9 针 COM1 串行鼠标接口。

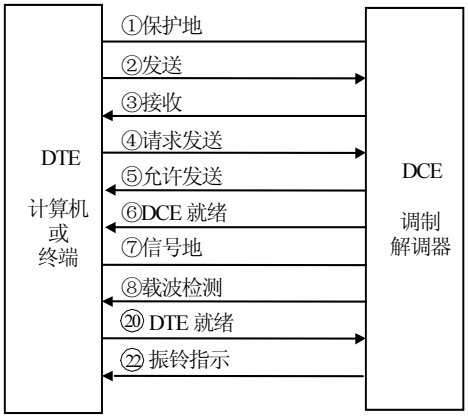

图 3-5　EIA-232-E/V.24 的主要信号线定义

（4）规程特性。EIA-232-E 的规程特性主要规定了控制信号在不同情况下有效（接通状态）和无效（断开状态）的顺序以及相互的关系。例如，只有当“DCE 就绪”和“DTE 就绪”信号都处于有效状态时，才能在 DTE 和 DCE 之间进行操作。如果 DTE 要发送数据，则先要将“请求发送”置成有效状态；当等到 DCE 将“允许发送”置成有效状态后，DTE 方能在“发送”线上发送串行数据。这种握手信号对于半双工通信是十分有用的。还有一些规程特性，这里就不一一介绍了。

2. EIA RS-449 接口标准

由于 EIA-232-E 标准信号电平过高、采用非平衡发送和接收方式，所以存在传输速率低、传输距离短、串扰信号较大等缺点。1977 年底，EIA 颁布了一个新标准 RS-449，这些标准在保持与 EIA-232-E 兼容的前提下重新定义了信号电平，并改进了电路方式，以达到较高的传输速率和较大的传输距离。

RS-499 对标准连接器做了详细的说明，由于信号线较多，使用了 37 芯和 9 芯连接器。

RS-449 的电器特性有两个标准，即平衡式的 RS-422 标准和非平衡式的 RS-423 标准。

RS-422 电气标准是平衡方式标准，它的发送器、接收器分别采用平衡发送器和差动接收器，由于采用完全独立的双线平衡传输，抗串扰能力大大增强。又由于信号电平定义为±6V（±2V 为过渡区域）的负逻辑，故当传输距离为 10m 时，速率可达 10Mbps；而距离增长至 1000m 时，速率可达到 100kbps 时，性能远远优于 EIA-232-E 标准。

RS-423 电气标准是非平衡标准，它采用单端发送器（即非平衡发送器）和差动接收器。虽然发送器与 RS-232C 标准相同，但由于接收器采用差动方式，所以传输距离和速度仍比 EIA-232-E 有较大的提高。当传输距离为 10m 时，速度可达到 100kbps；距离增至 100m 时，速度仍有 10kbps。RS-423 的信号电平定义为±6V（±4V 为过渡区域）的负逻辑。

从旧技术标准向新技术标准的过渡，需要花费巨大的代价，要经过漫长的过程。RS-423 电气特性标准可以认为是从 EIA-232-E 向 RS-449 标准全面过渡过程中的一个台阶。

3.3 数据链路层

3.3.1 数据链路层的功能

数据链路层是 OSI 参考模型中的第二层，介于物理层和网络层之间，在使用物理层的基础上向网络层提供服务。数据链路层的主要作用是：通过一些数据链路层协议和链路控制规程，在不太可靠的物理链路上实现可靠的数据传输。“线路”、“链路”和“数据链路”是不同的概念。线路中间没有任何交换节点，而链路是一条无源的端到端的物理线路段，在进行数据通信时，两台计算机之间的通信链路往往是由许多线路串接而成的。把实现控制数据传输的一些规程的硬件和软件加到链路上就构成了像数据管道一样的数据链路。有时往往将链路称为物理链路，而将数据链路称为逻辑链路，即物理链路加上必要的通信规程就是数据链路。当采用复用技术时，一条物理链路上可以有多条逻辑数据链路。数据链路层为了实现相邻节点之间数据帧的正确传输，必须包括链路管理、帧同步、流量控制、差错校验与恢复等基本功能。

3.3.2 差错控制

在数据链路层，差错控制主要指错误检测和重传方法。传送帧时可能出现的差错有：位出错、帧丢失、帧重复、帧乱序。位出错的分布规律及出错位的数量很难限制在预定的简单模式中，一般采用漏检率及其 CRC 检错码再加上反馈重传的方法来解决。

为了保证可靠传送，常采用的方法是向数据发送方提供有关接收方接收情况的反馈信息。一个否定性确认意味着发生了某种差错，相应的帧必须被重传。这种做法就是反馈重传。采用定时器和编号可以保证每帧最终都能正确地传给目的地——网络层。发送方确认接收方是否正确收到了由它发送的数据信息的方法，称为反馈差错控制。通常采用反馈检测和自动重发请求（ARQ）两种基本方法来实现。

1. 反馈检测法

反馈检测法也称回送校检法或“回声”法，主要用于面向字符的异步传输中，如终端与远程计算机间的通信。这是一种无须使用任何特殊代码的差错检测法。双方进行数据传输时，接收方将接收到的数据（可以是一个字符，也可以是一帧）重新发回发送方，由发送方检查是否与原始数据完全相符。若不相符，则发送方发送一个控制字符（如 DEL）通知接收方删去

出错的数据，并重新发送该数据；若相符，则发送下一个数据。

反馈检测法原理简单，实现容易，也有较高的可靠性，但每个数据均被传输两次，信道利用率很低。这种差错控制方法一般用于面向字符的异步传输中，因为在这种场合下信道利用率并不是主要矛盾。

2. 自动重发请求法（ARQ 法）

实用的差错控制方法应该是既要求传输可靠性高，又要求信道利用率高。为此可使发送方将要发送的数据帧附加一定的冗余检错码一并发送，接收方则根据检错码对数据帧进行差错检测，若发现错误，就返回请求重发的应答，发送方收到请求重发的应答后，便重新传送该数据帧。这种差错控制方法就称为自动重发请求（Automatic Repeat reQuest）法，简称 ARQ 法。

ARQ 法仅需返回少量控制信息，便可有效地确认所发数据帧是否被正确接收。ARQ 法有几种实现方案，停止等待协议和连续 ARQ 协议是其中最基本的两种方案。

（1）停止等待协议。该方案规定发送方每发送一帧后就要停下来等待接收方的确认返回，仅当接收方确认已正确接收后，发送方再继续发送下一帧。当发生帧出错或帧丢失时，接收方不会向发送方发送任何确认帧。为防止发送方无限等待接收方的确认帧，该协议设置了计时器，若到了计时器所设置的重传时间时还未收到接收方的确认帧，发送方就重传前面所发送的数据帧。同时采用对发送的帧编号的方法，即赋予每帧一个序号，从而使接收方能从该序号来区分是新发送来的帧还是已经接收但又重发来的帧。例如，帧用 0 或 1 交替编号，肯定确认帧用 ACK0 和 ACK1 表示，ACK0 表示已接收到 1 号帧，并准备接收 0 号帧。

停止等待协议方案的实现过程如下：

1）发送方每次仅将当前信息帧作为待确认帧保留在缓冲存储器中。

2）当发送方开始发送信息帧时，随即启动计时器。

3）当接收方收到无差错信息帧后，即向发送方返回一个确认帧。

4）当接收方检测到一个含有差错的信息帧时，便舍弃该帧。

5）若发送方在规定时间内收到确认帧，即将计时器清零，继而开始下一帧的发送。

6）若发送方在规定时间内未收到确认帧（即计时器超时），则应重发存于缓冲器中的待确认信息帧。

从以上过程可以看出，停止等待协议方案的收、发双方仅需设置一个帧的缓冲存储空间，便可有效地实现数据重发并确保接收方接收的数据不会重复。停止等待协议方案最主要的优点就是所需的缓冲存储空间最小，因此在链路端使用简单终端的环境中被广泛采用。

（2）连续 ARQ 协议。连续 ARQ 协议方案是指发送方可以连续发送一系列信息帧，即不用等前一帧被确认便可发送下一帧。这就需要在发送方设置一个较大的缓冲存储空间（称做重发表），用以存放若干待确认的信息帧。当发送方收到对某信息帧的确认帧后便可从重发表中将该信息帧删除。所以，连续 ARQ 方案的链路传输效率大大提高，但相应地需要更大的缓冲存储空间。

连续 ARQ 方案的实现过程描述如下：

1）发送方连续发送信息帧而不必等待确认帧的返回。

2）发送方在重发表中保存所发送的每个帧的备份。

3）重发表按先进先出队列规则操作。

4）接收方对每一个正确收到的信息帧返回一个确认帧。

5）每一个确认帧包含一个唯一的序号，随相应的确认帧返回。

6）接收方保存一个接收次序表，它包含最后正确收到的信息帧的序号。

7）当发送方收到相应信息帧的确认后，从重发表中删除该信息帧的备份。

8）当发送方检测出失序的确认帧（即第 N 号信息帧和第 N+2 号信息帧的确认帧已返回，而 N+1 号的确认帧未返回）后，便重发未被确认的信息帧。

上面连续 ARQ 过程是假定在不发生传输差错的情况下描述的，如果差错出现，如何进一步处理还可以有两种策略，即 GO-BACK-N 策略和选择重发策略。

GO-BACK-N 策略的基本原理是，当接收方检测出失序的信息帧后，要求发送方重发最后一个正确接收的信息帧之后的所有未被确认的帧；或者当发送方发送了 N 个帧后，若发现该 N 帧的前一个帧在计时器超时后仍未返回其确认信息，则该帧被判为出错或丢失，此时发送方就不得不重新发送出错帧及其后的 N 帧。这就是 GO-BACK-N（退回 N）法名称的由来。因为，对接收方来说，由于这一帧出错，就不能以正常的序号向它的高层递交数据，对其后发送来的 N 帧也可能都因为不能接收而丢弃。GO-BACK-N 法的操作过程如图 3-6 所示。图中假定发送方发送完第 7 号帧后，发现第 2 号帧的确认返回在计时器超时后还未收到，则发送方只能退回从 2 号帧开始重发。

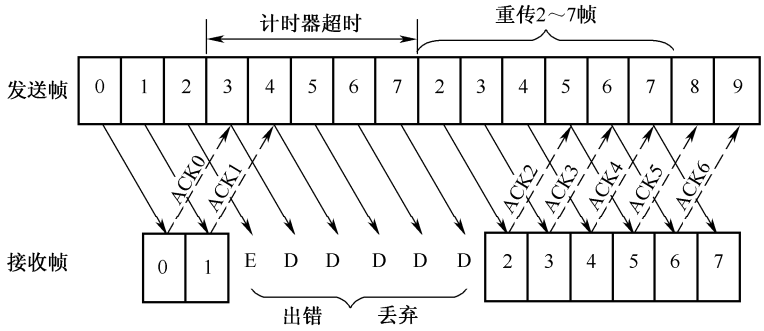

图 3-6　GO-BACK-N 法举例

GO-BACK-N 可能将已正确传送到目的方的帧再重传一遍，这显然是一种浪费。另一种效率更高的策略是当接收方发现某帧出错后，其后继续送来的正确的帧虽然不能立即递交给接收方的高层，但接收方仍可接收下来，存放在一个缓冲区中，同时要求发送方重新传送出错的那一帧。一旦收到重新传来的帧，就可以与原已存于缓冲区中的其余帧一并按正确的顺序递交高层。这种方法称为选择重发。显然，选择重发减少了浪费，但要求接收方有足够大的缓冲区空间。

3.3.3　流量控制

在数据链路层及较高层中，流量控制是一个重要的设计问题。通常，流量控制是与差错处理一起完成的，特别是在双工通信时，利用“捎带”技术使发送方知道接收方的速度能否跟得上发送方，从而能够决定是继续发送下一帧还是暂停发送，以等待收到某个反馈信息后再继续发送。

1. 停止-等待协议

停止-等待协议（Stop and Wait）是数据链路层中最基本、最简单的协议。

数据链路层从网络层接收一个分组后，加上数据链路层帧头和帧尾，再把它经物理层发送出去，同时启动一定时计数器，等待接收方回应的确认帧的到来。接收方链路层收到数据帧

后，它必须首先回应一个确认帧 ACK（认为所接收的数据正确无误）或否定性确认帧 NAK（认为接收的数据有误）给发送方链路层，再对接收的帧作出处理。正确接收的帧提交给网络层，将错误接收的帧丢弃。发送方如果在计时时间范围内得到的是 ACK，则发下一帧；如果收到的是 NAK 或者计时时间已到而没有收到 ACK，则将重发刚才送出去的帧。为了避免在无错情况下一个帧被多次重发，还需要为发出的每一个帧编号，使得接收方能够识别所接收的帧是新帧还是重发的帧，从而保证每一个帧的正确唯一。

发送方每发完一帧就必须停下来，等待接收方的回应信息。因此，可以通过接收方的回应信息来进行流量控制。

2. 滑动窗口协议

停止－等待协议的主要问题是链路上只有一个帧在传输，不能连续发送多个数据帧，许多线路带宽都要浪费，滑动窗口协议可以克服这个缺点。

滑动窗口协议的主要思想是允许连续发送多个帧而无须等待应答。“窗口”是指能够连续发出或接收的帧的序号范围，它反映了正在流动的帧的个数。发送方保持的允许连续发送的帧的序号表称作发送窗口，接收方保持的允许连续接收的帧的序号表称作接收窗口。

在发送端和接收端分别设定所谓的发送窗口和接收窗口。发送窗口用来对发送端进行流量控制，而发送窗口的大小 W_S 就代表了在还没有收到对方确认的条件下发送端最多可以发送的数据帧数。发送窗口的概念最好用图形来说明，设发送序号用 3 bit 来编码，从 0 号至 7 号。在未收到对方确认信息的情况下，允许发送端最多发出 5 个数据帧，此时发送窗口大小 W_S=5。图 3-7（a）画出了刚开始发送时的情况。这时，在扇形的发送窗口内共有 5 个序号，从 0 号到 4 号，具有这些序号的数据帧就是发送端现在可以发送的帧。若发送端发完了这 5 个帧仍未收到确认信息，由于发送窗口已填满，就必须停止发送而进入等待状态。当 0 号帧的确认信息 ACK 收到后，发送窗口就沿顺时针方向旋转一个号，使窗口后沿再次与一个未被确认的帧号相邻（如图 3-7（b）所示）。由于这时 5 号帧的位置已经落入发送窗口之内，因此，发送端现在就可以发送这个 5 号帧。设以后又有 1 至 3 号帧的确认帧到达发送端，于是发送窗口再沿顺时针方向向前旋转 3 个号（如图 3-7（c）所示），相应地，发送端可以继续发送的数据帧的发送序号是 6 号、7 号和 0 号。

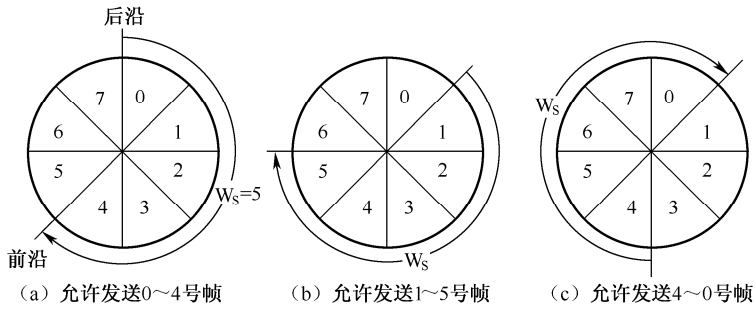

图 3-7　发送窗口 W_S 流程控制图

为了减少开销，连续 ARQ（Automatic Repeat reQuest，自动重发请求）协议还规定接收端不一定每收到一个正确的数据帧就必须发回一个确认帧，而是可以在连续收到好几个正确的数据帧以后，才对最后一个数据帧发确认信息。一旦对某一数据帧进行了确认，则表明该数据帧和这以前所有的数据帧均已正确无误地被收到了。这样做可以使接收端少发一些确认帧，从而

减少开销。

在接收端设置接收窗口的目的是为了控制哪些数据帧可以接收而哪些帧不可以接收。在接收端只有当收到的数据帧的发送序号落入接收窗口内才允许将该数据帧收下；若接收到的数据帧落在接收窗口之外，则一律将其丢弃。在连续 ARQ 协议中，接收窗口的大小 W_R=1。图 3-8（a）表示一开始接收窗口处于 0 号帧处，接收端准备接收 0 号帧。0 号帧一旦收到，接收窗口就沿顺时针方向向前旋转一个号（如图 3-8（b）所示），准备接收 1 号帧，同时向发送端发送对 0 号帧的确认信息。显然，若收到 1 号帧，则接收窗口将顺时针旋转一个号，并发出对 1 号帧的确认。但若收到的不是 1 号帧，情况就要复杂些。如果收到的帧号落在接收窗口的前面（顺时针方向），例如收到了 2 号帧，这时接收端就必须丢弃它，并发出对 2 号帧的否认信息。但若收到的帧号落在接收窗口的后面，例如收到了 0 号帧（注意：0 号帧已收到，并对它发送过确认信息），这就表明已发出的对 0 号帧的确认帧没有被发送方收到，因此现在还要再发一次对 0 号帧的确认，不过这时不能再把 0 号帧送交主机（否则就重复了）。在这两种情况下，接收窗口都不得向前旋转。当陆续收到 1 号、2 号和 3 号帧时，接收窗口的位置应如图 3-8（c）所示。

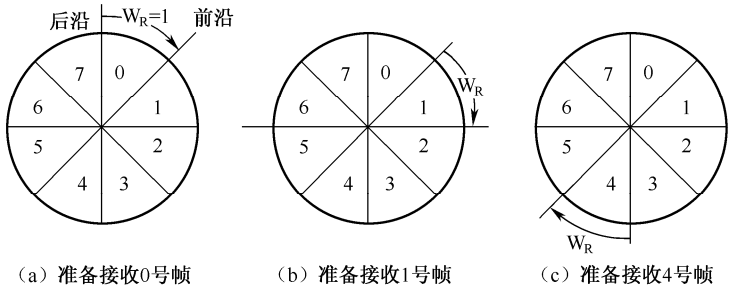

图 3-8　接收窗口 W_R 的意义

从以上的讨论可以看出，当接收窗口保持不动时，发送窗口无论如何也不会旋转，只有在接收窗口发生旋转后，发送窗口才有向前旋转的可能。正因为收发两端的窗口按照以上的规律不断地沿顺时针方向旋转滑动，因此这种协议就称为滑动窗口协议。显然当发送窗口和接收窗口的大小都等于 1 时，就是停止－等待协议；如果接收方发出某个应答信号后，不再发出新的应答信号，则两个窗口都会不断缩小，直至减小到 0，这时发送器不能再发出帧，接收器也不再接收，从而达到了流量控制的目的。

为什么说如果帧编号字段为 k 位，窗口的大小 W 不能大于 2^k-1？现在通过反例来说明。

设帧编号字段为 3 位，帧编号为 0～7，如果 W=8，发送方发送了 0～7 共 8 个帧后，停止发送并等待应答。接收方正确收到 0～7 共 8 个帧，并向发送方返回应答确认 ACK=0，如果该 ACK=0 丢失，发送方在规定的时间内接收不到应答信号，发送方重新发送了 0～7 共 8 个帧，结果接受方误认为是新的 8 个帧，协议失败。

如果 W=7，发送方发送了 0～6 共 7 个帧后，发送方返回应答确认 ACK=7 丢失，发送方重新发送了 0～6 共 7 个帧。由于接收方期望接收的帧为 7 号帧，而到达的第一个帧为 0 号帧，接收方可以判断出 0～6 号是重发帧。这就解释了为什么说如果帧编号字段为 k 位，窗口的大小 W 不能大于 2^k-1 的原因。

3.3.4　高级数据链路控制协议

HDLC 的全称是高级数据链路控制协议（Highlevel Data Link Control），它是国际标准化

组织（ISO）根据 IBM 公司的 SDLC（Synchronous Data Link Control）协议扩展开发而成的。HDLC 是面向比特的传送协议，采用“0”插入技术实现数据的透明传送，它使用滑动窗口，可以全双工传送。HDLC 用统一结构的帧进行同步传送，所有的数据链路层的传输都是以帧为单位进行的，而所有的数据和控制信息的交换也都采用帧的格式。一个帧的结构具有固定的格式，信息字段的头尾各加上 24bit 的控制信息，就构成了一个完整的 HDLC 数据帧。HDLC 的帧格式如表 3-2 所示。

表 3-2　HDLC 的帧格式

F	A	C	I	FCS	F
标志	地址	控制	信息	帧校验	标志
01111110	8bit	8bit	任意长	16bit	01111110

各字段的意义及功能如下：

（1）标志域 F（Flag）。帧标志序列 F 是一个 8 位的序列 01111110，由于帧中数据段长度可变，故用 F 来标志一帧的开始和结束。F 也可作为帧间的同步信号。当发送一些连续帧时，一个标志序列 F 可同时作为前一帧的结束标志和下一帧的开始标志。当帧与帧间不发送信息时，可连续地发送标志序列。由于帧中间也可能出现 01111110，会被当作标志，从而破坏帧的同步。为了避免这种错误的出现，要采用“0”插入技术，即发送器在发送的数据比特序列中一旦发现连续的 5 个 1，则在其后插入一个 0。这样就保证了传输的数据比特序列中不会出现和帧标志相同的 01111110。接收器则进行相反的操作：在接收的比特序列中如果发现 5 个连续 1 的序列，则检查第 6 位，若第 6 位为 0 则删除之；若第 6 位是 1 且第 7 位是 0，则认为是检测到帧尾的标志域；若第 6 位和第 7 位都是 1，则认为是发送站的停止信号。采用“0”插入技术，任意的位模式都可以出现在数据帧中，且不影响传输过程的控制，具有这种特点的数据传输叫做透明的数据传输。

（2）地址域 A（Address）。由主站发出的命令帧中的地址段，指明接收帧的次站地址，当地址域的内容为全 1 代码（FFH）时，表示为广播地址；由次站发出的响应帧中的地址段则表示做出应答的次站地址，即说明该响应是由哪一个次站发出的。地址段长 8bit，可指示 256 个地址。若需扩充指示的地址范围时，以 8bit 为单位进行地址扩充。每一个 8 位组的最低位指示该 8 位组是否是地址域的结尾：若为 1，表示是最后的 8 位组；若为 0，则表示不是。

（3）控制域 C（Control）。HDLC 定义了 3 种帧，可以根据控制域的格式来区分。信息帧（I 帧）装载着要传输的数据，此外还捎带着流量控制和差错控制的信号；管理帧（S 帧）用于提供实现 ARQ 的控制信息，当不使用捎带机制时用管理帧控制传输过程；无编号帧（U 帧）用于提供各种链路控制功能。

（4）信息域 I（Information）。信息域 I 位于控制段和帧校验序列之间，用来存放要传输的数据信息。可以是任意比特长组合，也未规定长度大小，但在实际应用中受到 FCS 的检验能力及站缓存大小的限制，一般规定最大信息长度不超过 256B。

（5）帧校验序列 FCS（Frame Check Sum）。帧校验序列 FCS 域中除标志域之外的所有其他域的校验序列。该字段采用 16 位 CRC 校验码进行差错控制，其生成多项式为 $X^{16}+X^{12}+X^5+1$。

3.4 网络层

3.4.1 网络层的功能

网络层是 OSI 参考模型的第三层，介于数据链路层和传输层之间。其任务是分组转发、路由选择和流量控制，最主要的功能是实现端到端通信系统中中间节点的路由选择。从 OSI/RM 的通信角度来看，网络层所提供的服务主要有两大类，即面向连接服务和无连接服务。这两种网络服务的具体实现就是所谓的虚电路服务和数据报服务。

1. 面向连接服务

连接是指两个对等实体之间为进行数据通信而进行的一种结合。面向连接服务就是在数据交换之前，必须先建立连接，当数据交换结束后，则应该终止这个连接。通常面向连接服务是一种可靠的报文序列服务，在建立连接之后，每个用户都可以发送可变长度的报文，这些报文按顺序发送给远端的用户，报文的接收也是按顺序的。有时用户可以发送一个很短（1～2 字节长）的报文，但希望这个报文可以不按序号而优先发送，这就是“加速数据”，它常用来传送中断控制命令。

由于面向连接服务和线路交换的许多特性相似，因此面向连接服务在网络层中又称为虚电路服务。“虚”表示：在两个服务用户的通信过程中虽然没有自始至终都占用一条端到端的完整物理电路，但却好像占用了一条这样的电路。面向连接服务比较适合于在一定期间内要向同一目的地连续发送许多报文的情况。若两个用户经常进行频繁通信，则可建立永久虚电路，这样可免除每次通信时连接建立和连接释放这两个过程。

2. 无连接服务

在无连接服务的情况下，两个实体之间的通信不需要先建立好一个连接，因此其下层的有关资源不需要事先进行预定保留，这些资源是在数据传输时动态地进行分配的。无连接服务不需要通信的两个实体同时处于激活状态，当发送端的实体正在进行发送时，它必须是激活的，但这时接收端的实体并不一定要激活，只有当接收端的实体正在进行接收时，它才必须是激活的。无连接服务的优点是灵活方便和比较迅速，但无连接服务不能防止报文的丢失、重复或失序。采用无连接服务时由于每个报文都必须提供完整的目的站地址，因此其开销也较大。无连接服务大致有以下三种类型：

（1）数据报。特点是发完了就行，而不需要接收端做任何响应。数据报服务简单、额外开销小，虽然数据报服务没有面向连接服务可靠，但可在此基础上由更高层构成可靠的连接服务。数据报服务适用于电子邮件，特别适合于广播或组播服务。

（2）证实交付。这是一种可靠的数据报服务。这种服务对每一个报文产生一个证实给发送方用户，不过这个证实不是来自接收端的用户而是来自提供服务的层。这种证实只能保证报文已经发给远端的目的站了，但并不能保证目的站的用户已经收到了这个报文。

（3）请求应答。这种类型的数据报服务是接收端用户每收到一个报文，就向发送端用户发送一个应答报文。但是，收发双方发送的报文都有可能丢失。如果接收端发现报文有差错，则响应一个表示有差错的报文。

3.4.2 虚电路服务与数据报服务

下面结合网络层的特点简单对比一下这两种服务。

虚电路与存储转发这一概念相联系。当我们使用座机打电话时，在通话期间，自始至终地占用一条端到端的物理线路。但当我们占用一条虚电路进行计算机通信时，由于采用的是存储转发分组交换，所以只是断续地占用一段又一段的链路，感觉好像是占用了一条端到端的物理线路。使用虚电路服务，对网络用户来说，在呼叫建立后，整个网络就好像有两条连接两个网络用户的数字管道，所有发送到网络中的分组，都按发送的先后顺序进入管道，然后按“先进先出”的原则沿着管道传送到目的站主机。在全双工通信中，每一条管道只沿一个方向传送分组，这些分组到达目的站时的顺序与发送时的顺序一样。

数据报服务则不同，由于数据报服务没有建立虚电路的过程，每一个发出的分组都须携带完整的目的站的地址信息，因而每一个分组都可以独立地选择路由。在此情况下，没有呼叫建立过程，对于网络用户来说，整个网络好像有许多条不确定的数字管道，所发送出去的每一个分组都可独立地选择一条管道来传送。这样，先发送出去的分组不一定先到达目的站主机。因此，数据报不能保证按发送顺序交付目的站。由于通常的数据传送都要求按发送顺序交付目的站主机，所以在目的站必须采取一定的措施。例如，在目的站节点开辟缓冲区，把收到的分组缓存一下，等到可以按顺序交付主机时再进行交付。

在使用数据报时，每个分组必须携带完整的地址信息。但在使用虚电路的情况下，每个分组不需要携带完整的目的地址，而仅需要有个虚电路号码的标志。这样就使分组的控制信息部分的比特数减少，因而减少了额外开销。当采用数据报服务时，端到端的流量控制由主机负责；而采用虚电路服务时，端到端的流量控制由网络负责。

对待差错处理，这两种服务也是有很大差别的。由于数据报服务不能保证按顺序交付，也不能保证不丢失和不重复，因此在使用数据报服务的情况下，主机要承担端到端的差错控制。但在使用虚电路的情况下，网络有端到端的差错控制功能，能够保证分组按顺序交付，而且不丢失、不重复。美国 ARPANET 的一些主要用户，根据他们在网络上 20 多年实际工作的经验，认为网络不管用什么方法进行设计都不可能做到绝对可靠。因此，在主机上无论如何也需要有端到端的差错控制。既然如此，网络就不要再重复地搞差错控制，只要能提供数据报服务就可以了。这就是他们极力主张使用数据报服务的理由。

数据报服务对军事通信有很重要的意义。这是因为每个分组可独立地选择路由，当某个节点发生故障时，后续的分组可另选路由，因而提高了传输可靠性。数据报服务还很适合于将一个分组发送到多个地址进行广播或组播。ARPANET 网络兼有这两种服务的特点，其在网络内部采用数据报方式传送，但在交付主机之前，由于在目的节点的缓冲区将到达的分组按照发送序号重新排序，因此交付给主机的分组顺序与发送顺序相同，这一点和虚电路服务十分相似。表 3-3 归纳了虚电路服务与数据报服务的一些主要区别。

表 3-3　虚电路与数据报的对比

	虚电路	数据报
端到端的连接	必须有	不要
目的站地址	仅在连接建立阶段使用	每个分组都有目的站的全地址
分组的顺序	总是按发送顺序到达目的站	到达目的站时可能不按发送顺序
端到端的差错处理	由通信子网负责	由主机负责
端到端的流量控制	由通信子网负责	由主机负责

3.4.3 路由选择算法

通信子网为网络源节点和目的节点提供了多条传输路径的可能性。网络节点在收到一个分组后，要确定向下一节点传送的路径，这就是路由选择。在数据报方式中，网络节点要为每个分组路由做出选择；而在虚电路方式中，只需在连接建立时确定路由。确定路由选择的策略称为路由算法。

设计路由算法时要考虑诸多技术要素。第一，考虑是选择最短路由还是选择最佳路由；第二，要考虑通信子网是采用虚电路的还是采用数据报的操作方式；第三，是采用分布式路由算法，即每节点均为到达的分组选择下一步的路由，还是采用集中式路由算法，即由中央节点或始发节点来决定整个路由；第四，要考虑关于网络拓扑、流量和延迟等网络信息的来源；第五，确定是采用静态路由选择策略还是动态路由选择策略。静态路由选择策略和动态路由选择策略，是根据路由算法能否随网络的通信量或拓扑结构自适应地进行调整变化来划分的。集中式路由算法和分布式路由算法都属于动态路由选择策略。

1. 静态路由选择策略

静态路由选择策略不用测量也无需利用网络信息，这种策略按某种固定规则进行路由选择，其中还可分为洪泛路由选择、固定路由选择和随机路由选择三种算法。

（1）洪泛路由选择。这是一种最简单的路由算法。一个网络节点从某条线路收到一个分组后，再向除该线路外的所有线路分别发送该分组。结果，最先到达目的节点的一个或若干个分组肯定经过了最短的路径，而且所有可能的路径都被尝试过。这种方法用于诸如军事网络等强壮性要求很高的场合。即使有的网络节点遭到破坏，只要源、目的之间有一条信道存在，则洪泛路由选择仍能保证数据的可靠传送。另外，这种方法也可用于将一个分组数据源传送到所有其他节点的广播式数据交换中。它还可被用来进行网络的最短路径及最短传输延迟的测试。但这种方法传输效率低，额外开销大，冗余空间多，实际应用不多。

（2）固定路由选择。这是一种使用较多的简单算法。每个网络节点存储一张表格（路由表），该表格中每一项记录着对应某个目的节点的下一跳节点或链路。当一个分组到达某节点时，该节点只要根据分组上的地址信息，便可从固定的路由表中查出对应的目的节点及所应选择的下一跳节点。一般，网络中都有一个网络控制中心，由它按照最佳路由算法求出每对源、目的节点的最佳路由，然后为每一节点构造一个固定路由表并分发给各个节点。固定路由选择法的优点是简便易行，在负载稳定，拓扑结构变化不大的网络中运行效果很好。它的缺点是灵活性差，无法应付网络中发生的阻塞和故障。

（3）随机路由选择。在这种方法中，收到分组的节点，在所有与之相邻的节点中为分组随机选择出一个节点。方法虽然简单，但实际路由不是最佳路由，这会增加不必要的负担，而且分组传输延迟也不可预测，故此法应用不广。

2. 动态路由选择策略

节点的路由选择是依靠网络当前的状态信息来决定的策略，称为动态路由选择策略。这种策略能较好地适应网络流量、拓扑结构的变化，有利于改善网络的性能。但由于算法复杂，会增加网络的负担。有三种动态路由选择策略算法：独立路由选择、集中式路由选择和分布式路由选择。

（1）独立路由选择。在这种路由算法中，节点仅根据自己搜集到的有关信息做出路由选择的决定，与其他节点不交换路由选择信息。由于每个节点只考虑本节点的运行状态，这种算

法不能正确确定距离本节点较远的路由选择，但还是能较好地适应网络流量和拓扑结构的变化，只是这种适应性比较有限。一种简单的独立路由选择算法是 Baran 在 1964 年提出的热土豆（Hot Potato）算法：当一个分组到来时，节点必须尽快脱手，将其放入输出队列最短的方向上排队，而不管该方向通向何方。

（2）集中式路由选择。集中式路由选择也像固定路由选择一样，在每个节点上存储一张路由表。不同的是，固定路由选择算法中的节点路由表由人工制作，而在集中式路由选择算法中的节点路由表由路由控制中心 RCC（Routing Control Center）定时根据网络状态计算、生成并分送到各相应节点。由于 RCC 利用了整个网络的信息，所以得到的路由选择是完美的，同时也减轻了各节点计算路由选择的负担。但它缺乏坚定性，一旦 RCC 出故障，整个网络的路由选择功能将瘫痪。

（3）分布式路由选择。在采用分布式路由选择算法的网络中，所有节点定期地与其相邻的每个节点交换路由选择信息。各节点均存储一张以网络中其他节点为索引的路由选择表，网络中每个节点占用表中一项。每一项又分为两个部分，一部分是所希望使用的到目的节点的输出线，另一部分是估计到达目的节点所需要的延迟或距离，度量标准可以是毫秒或链路段数、等待的分组数、剩余的线路和容量等。

3.4.4　拥塞控制技术

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，严重时甚至会导致网络通信业务陷入停顿，即出现死锁现象。这种现象跟公路网中常见的交通拥挤一样，当节假日公路网中车辆大量增加时，各种走向的车流相互干扰，使每辆车到达目的地的时间都相对增加（即延迟增加），甚至有时在某段公路上车辆因堵塞而无法开动（即发生局部死锁）。

网络的吞吐量与通信子网负荷（即通信子网中正在传输的分组数）有着密切的关系。当通信子网负荷比较小时，网络的吞吐量（分组数/秒）随网络负荷（每个节点中分组的平均数）的增加而线性增加。当网络负荷增加到某一值后，若网络吞吐量反而下降，则表明网络中出现了拥塞现象。在一个出现拥塞现象的网络中，到达某个节点的分组将会遇到无缓冲区可用的情况，从而使这些分组不得不由前一节点重传，或者需要由源节点或源端系统重传，从而使通信子网的有效吞吐量下降。由此引起恶性循环，使通信子网的局部甚至全部处于死锁状态，最终导致网络有效吞吐量接近为零。

引起网络拥塞的原因是多方面的，由于网络各部分的速率、带宽、容量、分组数量等的不匹配都会造成网络拥塞。

比如，当某个节点缓冲区的容量太小时，到达该节点的分组会因无空间缓存而不得不被丢弃，又不得不被多次重传，从而发生网络拥塞现象。假如现在将该节点的缓冲区容量扩展到非常大，是不是就不会出现拥塞了呢？不是的。扩大缓冲容量虽然可以使到达该节点的分组都能在这缓存的队列中排队而不受任何限制，但由于输出链路的容量和处理机的速度并未提高，那么在该队列中的绝大多数分组就会因为排队等待的时间过长而被上层软件认为超时，从而把它们重传。因此，只有所有的部分都匹配了，拥塞的问题才能解决。拥塞控制就是要控制如何有效、公平地分配网络资源。

拥塞控制方法一般有以下几种：缓冲区预分配方法、分组丢弃法和定额控制法。

（1）缓冲区预分配方法。该法用于虚电路分组交换网中。在建立虚电路时，让呼叫请求

分组的途经的节点为虚电路预先分配一个或多个数据缓冲区。若某个节点缓冲器已被占满，则呼叫请求分组另择路由，或者返回一个“忙”信号给呼叫者。这样，通过途经的各个节点为每条虚电路开设的永久性缓冲区（直到虚电路拆除），就总能有空间来接纳并转送经过的分组。此时的分组交换跟电路交换很相似。当节点收到一个分组并将它转发出去之后，该节点向发送节点返回一个确认信息。该确认一方面表示接收节点已正确收到分组；另一方面告诉发送节点，接收节点已空出缓冲区以备接收下一个分组。若节点之间的协议允许多个未处理的分组存在，则为了完全消除拥塞的可能性，每个节点要为每条虚电路保留等价于窗口大小数量的缓冲区。这种方法不管有没有通信量，都有可观的资源（线路容量或存储空间）被某个连接占有，因此网络资源的有效利用率不高。这种控制方法主要用于要求高带宽和低延迟的场合，例如传送数字化语音信息的虚电路。

（2）分组丢弃法。该法不必预先保留缓冲区，当缓冲区占满时，将到来的分组丢弃。若通信子网提供的是数据报服务，则用分组丢弃法来防止阻塞发生不会引起大的影响。但若通信子网提供的是虚电路服务，则必须在某处保存被丢弃分组的备份，以便拥塞解决后能重新传送。有两种解决被丢弃分组重发的方法，一种是让发送被丢弃分组的节点超时，并重新发送分组直至分组被收到；另一种是让发送被丢弃分组的节点在一定次数后放弃发送，并迫使数据源节点超时而重新开始发送。但是不加分辨地随意丢弃分组也不妥，因为一个包含确认信息的分组可以释放节点的缓冲区，若因节点无空余缓冲区来接收含确认信息的分组，这便使节点缓冲区失去了一次释放的机会。解决这个问题的方法是可以为每条输入链路永久地保留一块缓冲区，以用于接纳并检测所有进入的分组，对于捎带确认信息的分组，在利用了所捎带的确认释放缓冲区后，再将该分组丢弃或将该捎带好消息的分组保存在刚空出的缓冲区中。

（3）定额控制法。这种方法在通信子网中设置适当数量的称做“许可证”的特殊信息，一部分许可证在通信子网开始工作前预先以某种策略分配给各个源节点，另一部分则在子网开始工作后在网中四处环游。当源节点要发送来自源端系统的分组时，它必须首先拥有许可证，并且每发送一个分组注销一张许可证。目的节点方则每收到一个分组并将其递交给目的端系统后，便生成一张许可证。这样便可确保子网中分组数不会超过许可证的数量，从而防止了拥塞的发生。

拥塞的极端情况是死锁。死锁是网络中最容易发生的故障之一，即使在网络负荷不很重时也可能会发生。死锁发生时，一组节点由于没有空闲缓冲区而无法接收和转发分组，节点之间相互等待，既不能接收分组也不能转发分组，并一直保持这一僵局，严重时甚至导致整个网络的瘫痪。此时，只能靠人工干预来重新启动网络，解除死锁。但重新启动后并未消除引起死锁的隐患，所以可能再次发生死锁。死锁是由于控制技术方面的某些缺陷引起的，起因通常难以捉摸、难以发现，即使发现，也常常不能立即修复。因此，在各层协议中都必须考虑如何避免死锁问题。

最常见的死锁是发生在两个节点之间的直接存储转发死锁。例如，A 节点的所有缓冲区装满了等待输出到 B 节点的分组，而 B 节点的所有缓冲区也全部装满了等待输出到 A 节点的分组；此时，A 节点不能从 B 节点接收分组，B 节点也不能从 A 节点接收分组，从而造成两节点间的死锁。这种情况也可能发生在一组节点之间，例如，A 节点企图向 B 节点发送分组，B 节点企图向 C 节点发送分组，这种情形称做间接存储转发死锁。当一个节点处于死锁状态时，所有与之相连的链路将完全被阻塞。

一种防止存储处于死锁的方法是，为每个节点设置 M+1 个缓冲区，并以 0 到 M 编号。M

为通信子网的直径，即从任一源节点到任一目的节点间的最大链路段数。每个源节点仅当其为 0 号缓冲区时才能接收源端系统发送来的分组，而此分组仅能转发给 1 号缓冲区空闲的相邻节点，再由该节点将分组转发给它的 2 号缓冲区空闲的相邻节点……。最后，该分组或者顺利到达目的节点并被递交给目的端系统，或者到了某个节点编号为 M 的缓冲区中再也转发不下去，此时一定发生了循环，应该将该分组丢弃。由于每个分组都是按照编号递增规则分配缓冲区，所以节点之间不会相互等待空闲缓冲区而发生死锁现象。这种方法的不足之处在于，当某节点虽然有空闲缓冲区，但正巧没有所需要的特性编号的缓冲区时，分组仍要等待，从而造成了缓冲区和链路的浪费。

另一种防止存储转发死锁的方法是，使每个分组上都携带一个全局性的唯一的“时间戳”，每个节点要为每条链路保留一个特殊的接收缓冲区，而其他缓冲区均可用于存放中转分组。在每条输出链路的队列上分组按时间戳顺序排队。例如，节点 A 要将分组送到节点 B，若 B 节点没有空闲缓冲区，但正巧有要送到 A 节点的分组，此时 A、B 节点可通过特殊的接收缓冲区交换分组；若 B 节点既没有空闲缓冲区，也没有要送到 A 节点的分组，B 节点只好强行将一个出路方向大致与 A 节点方向相同的分组与 A 节点互相交换，但此时 A 节点中的分组必须比 B 节点中的分组具有更早的时间戳，这样才能保证子网中某个最早的分组不受阻挡地转发到目的地。由此可见，每个分组最终总会成为最早的分组，并总能一步一步地发送到目的节点，从而避免了死锁现象的发生。

死锁中比较严重的情况是重装死锁。假设发给一个端系统的报文很长，被源节点拆成若干个分组发送，目的节点要将所有具有相同编号的分组重新装配成报文递交给目的端系统，若目的节点用于重装报文的缓冲区空间有限，而且它无法知道正在接收的报文究竟被拆成多少个分组，此时，就可能发生严重的问题：为了接收更多的分组，该目的节点用完了它的缓冲空间，但它又不能将尚未拼装完整的报文递送给目的端系统，而邻节点仍在不断地向它传送分组，但它却无法接收。这样，经过多次尝试后，邻节点就会绕道从其他途径再向该目的节点传送分组，但该目的节点已被死锁，其周边区域也由此发生了阻塞。

避免重装死锁的发生可以有以下几种方法：①允许目的节点将不完整的报文递交给目的端系统，但该方法使端系统中的协议复杂化了。②一个不能完整重装的报文能被检测出来，并要求发送该报文的源端系统重新传送，但该方法破坏了网络层对端系统的透明性，因为端系统不该考虑诸如报文拆、装之类的工作。③每个节点配备一个缓冲空间，用以暂存不完整的报文，但该方法使每个节点增加了开销。

总之，拥塞控制是很难设计的，因为它是一个动态的问题。不过总的来说可以用开环控制和闭环控制两种方法。开环控制方法就是在设计网络时事先将有关发生拥塞的因素考虑周到，为求网络在工作时不产生拥塞，但一旦系统运行起来就不能改了。闭环控制是通过监测系统来检测拥塞在何时何处发生，然后将拥塞发生的信息传送到可采取行动的地方，从而调整网络系统的运行以解决出现的问题。

3.5　传输层

3.5.1　传输层的功能

传输层又称运输层，是介于低三层通信子网系统和高三层之间的一层。传输层的作用是

从端到端经网络透明地传送报文，完成端到端通信链路的建立、维护和管理。所谓端到端就是从进程到进程。传输层向高层用户屏蔽了高层以下通信子网的细节，使高层用户看不见实现通信功能的物理链路是什么，看不见数据链路采用什么控制规程，也看不见下面到底有几个子网以及这些子网是怎样互联起来的。传输层让高层用户看见的就好像是在两个传输层实体之间有一条端到端的可靠通信通路。通信子网中没有传输层，传输层只存在于通信子网以外的主机中。一个传输层协议通常可同时支持多个进程的连接。若通信子网所提供的服务越多，传输协议就可以做得越简单；反之，若通信子网所提供的服务越少，传输协议就必然越复杂。传输协议有时可以看成是传输层所提供的服务与网络层提供的服务之差。在极端情况下，若网络层提供的服务达到了传输层应提供的服务，则传输协议甚至就不需要了。由于有了传输层，用户（即会话实体）在进行通信时就不必知道通信网的构成及线路质量等，也不必考虑子网是局域网还是公用分组交换网，用户在传送数据时不必关心数据传送方法的细节，但传输层不对所传送的数据内容进行加工处理。

传输层协议与数据链路层协议相比较，其主要区别为：数据链路层的环境是两个分组交换节点 PSN 直接通过一条物理信道进行通信，而传输层的环境则是两个主机以整个子网为通信信道进行通信。这样就使传输层的环境比数据链路层的环境复杂得多，因而其流量控制也较为复杂。

3.5.2 传输协议的分类

网络的服务质量大致有三种类型。

A 型：网络连接具有可接受的低差错率和可接受的低故障通知率。A 型网络服务是一个完善的、理想的、可靠的网络服务，这时的传输层协议非常简单。然而实际的网络很少能达到这个水平。

B 型：网络连接具有可接受的低差错率和不可接受的高故障通知率。对于 B 型网络连接，传输协议必须提供差错恢复的功能。多数 X.25 公用分组交换网络提供的是 B 型网络服务。

C 型：网络连接对传输层服务用户来说具有不可接受的高差错率。C 型网络服务质量最差。此时要求传输层具有更强的差错恢复能力。大多数无线分组网属于这种类型。

为了能够在各种不同的网络上进行不同类型的数据传送，ISO 定义了 5 类传输协议，即第 0～4 类传输协议，它们都是面向连接的。

（1）第 0 类传输协议最简单，它的功能就是建立一个简单的端到端的传输连接，并可以在数据传送阶段将长数据报文分段传送，没有差错恢复功能，也没有将多条传输连接复用到一条网络连接上的能力，主要是面向 A 型网络服务。

（2）第 1 类传输协议也较简单，只是增加了基本的差错恢复功能，主要是面向 B 型网络服务。

（3）第 2 类传输协议具有连接复用功能，但没有对网络连接出现故障的恢复功能，这类协议还具有相应的流量控制功能，主要是面向 A 型网络服务。

（4）第 3 类传输协议包含了第 1 类和第 2 类传输协议的功能，既有差错恢复又有复用功能，主要是面向 B 型网络服务。

（5）第 4 类传输协议是最复杂的，功能较齐全，具有差错检测、控制、恢复以及复用等功能，可以在质量较差的网络上保证高可靠的数据传输，主要是面向 C 型网络服务。

ISO 关于传输协议只提供了一种连接突然释放服务，在这种释放中，处于两个传输实体之

间的数据有可能丢失，美国国家标准局联邦信息处理标准 NBS FIPS 中则有一个可以使正在传送的数据不因连接释放而丢失的服务选项，即文雅释放（Graceful Close）。

3.5.3　传输层协议的要素

传输层协议的实现取决于它赖以运行的网络环境以及它提供的服务类型。下面假定传输层必须满足不可靠的网络服务，传输层协议必须要解决如下 5 个问题：

1. 寻址

寻址功能关系到用户如何在网络中标识自己或得到其他用户的名字地址，这是传输层协议必须具备的功能。对地址的编排多采用层次型地址，例如：

地址=<国家><网络><主机><端口>

<国家>和<网络>字段在整个网络中有效，而<主机>和<端口>只在它所属的系统中有局部意义。

2. 建立连接、数据传送和拆除连接

（1）建立连接。传送层连接的建立要保证双方建立起连接，使通信双方确信对方存在，协商任选参数（传输协议数据单元 TPDU 长度、窗口大小以及服务质量等）和分配传输实体资源（存储缓冲区、连接入口表项等）。

（2）数据传送。用户进程建立起连接后，就进入数据传送阶段。数据传送按 TPDU 的大小和格式组织，数据传送包括一般数据传送和加速数据传送。加速数据传送比一般数据传送有更高的优先权。传输层要向用户提供可靠的、透明的数据传送，以保证传输层协议数据单元 TPDU 不出错、不丢失、不重复和按次序向目的地提交数据，还要进行流量控制。

（3）拆除连接。如传输实体从用户收到一个拆除连接的通知，就除去未送完的数据，并发出一个拆除连接请求 TPDU 给对方。当对方传输层收到拆除请求后，就发回一确认 TPDU，除去未接收完的数据，并通知用户。为增加拆除连接的可靠性，常用三次握手法拆除连接。

3. 流量控制

传输层的流量控制，在很多方面与数据链路层相似，都是为防止发送过快而超过接收者的能力，采用的方法都是基于滑动窗口的原理。数据链路层由于连线少，通信量大，常采用固定窗口大小，而传输层则采用动态窗口管理和动态缓冲分配策略。

4. 多路复用

当传输服务用户进程产生的信息流较少时，可将多个传输连接映射到一个网络连接上，以便充分利用网络连接的传输效率，即所谓向上多路复用。相反，当一对进程间传送的信息量大于网络连接（即一条虚电路）所能传送的信息量时，该传输连接可打开多个网络连接（即多条虚电路），以便多条网络连接共同传送同一个传输连接的信息，实现对传输服务用户进程信息分流传输，以保证传输层信息吞吐量的要求，即所谓向下多路复用。

5. 崩溃恢复

当传输实体所在的主机系统崩溃后，所有连接状态信息都丢失了。如果主机系统重新启动，受崩溃影响的连接就变成了半开通的连接，因为另外一方没有经历崩溃的灾难，并不知道对方出了问题。传输实体应该保留一个“放弃定时器”，这个定时器测量等待一个多次重传的 TPDU 的应答信号的时间。定时器超时后，传输实体就认为另外一边的传输实体或中间的网络已经失效，自动关闭连接，并把异常情况通知上层的传输用户。未崩溃方等待对方重新启动后发来的信号，双方进行协调后，再从崩溃处开始新的工作。

3.6 高层

3.6.1 会话层

会话层建立在传输层提供的完整提交平台上，因而它不必担心协议数据单元的损坏和丢失，差错恢复的工作都由传输层完成了。会话层的任务主要是在传输连接的基础上提供增值服务，对端用户之间的对话进行协调和管理。所谓一次会话，就是两个用户进程之间为完成一次完整的通信而建立会话连接。应用进程之间为完成某项处理任务而需进行一系列内容相关的信息交换，会话层就是为有序地、方便地控制这种信息交换提供控制机制。

会话层完成的主要功能有：

1. 会话连接到传输连接的映射

会话连接要通过传输连接来实现，会话连接和传输连接有 3 种对应关系：一个会话连接对应一个传输连接；多个会话连接对应一个传输连接，它表示相继建立的几个会话使用同一个传输连接，但不能把多个会话连接同时对应一个传输连接，即会话层不支持多路复用；一个会话连接对应多个传输连接，它表示一个会话跨越了几个传输连接。

2. 数据传送

会话用户进程间的数据通信大多数是交互式的半双工通信方式，对半双工交互式的会话服务用户之间的通信用数据令牌来控制，有数据令牌的会话服务用户才可以发送数据，另一方只能接收数据。当数据发完后，就将数据令牌转让给对方，对方也可以请求令牌。持有释放令牌的会话服务用户可以释放连接，任一方释放连接都要得到对方的认可才可执行释放连接的动作；否则要继续维持数据交换，这称为有序释放或协商释放，以避免随意释放引起的数据丢失。这样解决了谁发送以及发送完整的数据，使会话有顺序地进行。数据分为常规数据、加速数据、特权数据和能力数据。

3. 同步

用户的会话可由对话单元组成，一个对话单元是基本的交换单元且每个对话单元都是单向的、连续的。会话用户可按对话单元交互传送。因此，不同的对话单元可以不是一个方向的，主同步点就是在数据流中标出对话单元。一个主同步点表示前一个对话单元的结束，下一个对话单元的开始。在一个对话单元内部即两个主同步点之间可以设置次同步点，用于对话单元数据的结构化。同步点的设置是为了便于实现同步操作，即重新同步。如将一个文件连续发送给对方，一旦出现错误，从双方同意的同步点处重新开始继续传送，而不必从文件的开头恢复会话，以提高传送的效率。

4. 活动管理

会话服务用户之间的交互对话可以划分为不同的逻辑单元，每个逻辑单元称为活动。每个活动完全独立于它前后的其他活动，并且具有完整的逻辑功能。一个活动可以包含多个对话单元，活动包含的信息是双向的，一个对话单元只能是单方向的，而一个会话包含多个活动，一个活动又可以被中断，当它恢复执行时不会丢失信息。活动管理是构成一个会话的主要方法。

3.6.2 表示层

表示层向应用提供资料的表示，要解决不同系统的数据表示问题，解释所交换数据的意

义，进行正文压缩及各种变换，如代码转换、格式变换等，以便用户使用，使采用不同表示方法的各开放系统之间能相互通信。此外，利用密码对正文进行加密、解密也是该层的任务。

1．数据转换

不同的计算机有不同的数据内部表示，如不同的机器字长、不同的浮点数格式以及不同的字符编码等。在采用 ASCII 码的计算机中，使用的编码也不尽相同。显然这就使得开放的系统之间不能简单地交换数据。数据转换的主要任务是把要传送的数据按协议格式表示成收发双方都能理解的形式。因此，为使不同计算机之间进行通信时不曲解原数据含义，必须建立一种统一的数据格式转换机制。为此国际标准化组织 ISO 综合和吸收了许多高级程序设计语言对数据类型的描述方法，制定了称之为抽象语法表示法 ASNI.1（Abstract Syntax Notation.1）标准（即 ISO8824）和相配套的基本编码规则 BER（Basic Encoding Rule）。通过抽象语法和传送语法，把信息传送到远端系统，远端系统按照同样方法对信息进行还原。

2．数据压缩

在网络环境下，为了节省通信带宽，减少传输费用和加快传输速度，常常采用数据压缩办法来减少发送的数据量。数据压缩的方法有许多种：

（1）缩写法，如“中华人民共和国”缩写为“中国”；Operating System 缩写为 OS。

（2）霍夫曼（Huffman）编码，如我国的长途自动电话区号编码即为 Huffman 编码。

（3）上下文相关编码，如空格、换行及先导零串都可以用该法处理。

3．数据加密

为了保证网络上传输的数据不被恶意地复制、窃听或加入非法信息，必须对传输的信息加密，收方收到加密的信息后再进行解密。

3.6.3　应用层

应用层是 OSI 的最高层，也是用户访问网络的接口层，是直接面向用户的。在 OSI 环境下，为用户提供各种网络服务，例如电子邮件、文件传输、虚拟终端、远程登录等。

应用层包含了若干独立的、用户通用的服务模型，其主要目的是为用户提供一个窗口，用户通过这个窗口互相交换信息。应用层的内容完全取决于用户，各用户可以各自决定要完成什么功能和使用什么协议，该层包括的网络应用程序有的由网络公司提供，有的是用户自己开发。对于一些常用的服务功能，人们制定了标准，避免了一些重复的工作。

一、选择题

1．在组成网络协议的三素中，（　　）是指用户数据与控制信息的结构与格式。

A．语法　　B．语义　　C．时序　　D．接口

2．在 OSI 参考模型中，（　　）是参考模型的底层。

A．物理层　　B．数据链路层

C．网络层　　D．传输层

3．在 OSI 参考模型中数据链路层的数据服务单元是（　　）。

A．比特序列　　B．分组　　C．报文　　D．帧

二、填空题

1．网络层的主要功能是通过_____算法分组通过通信子网选择最适当的路径。

2．高层互联中的高层是指从_____开始到_____的多个层次，高层互联的核心是在高层之间进行_____转换。

3．_____是实现高层网络互联的设备。

三、简答题

1．什么是网络体系结构？为什么要定义网络体系结构？

2．比较虚电路服务与数据报服务的异同。

3．流量控制与路由选择有何异同？

4．简述防止拥塞的几种方法。

5．简述计算机通信中异步传输和同步传输的区别。