《计算机网络》总复习

Eslzzyl

2023 年 3 月 4 日

目录

一 计算机网络体系结构

1.1 计算机网络概述

1.1.1 计算机网路的概念

定义（周健 PPT）：将若干台具有**独立**功能的计算机系统，用某种或多种通信介质连接起来，通过完善的网络协议，在**数据交换**的基础上，实现网络**资源共享**的系统称为计算机网络。

- 独立：每台计算机都可运行各自独立的操作系统，彼此地位平等，无主从之分。
- 数据交换是网络的最基本功能。
- 资源共享是网络最终目的。

1.1.2 计算机网络的组成

- 从组成部分上看
  - ◇ 硬件：主机（端系统）+ 通信链路 + 交换设备 + 通信处理机 +……
  - ◇ 软件
  - ◇ 协议
- 从工作方式上看
  - ◇ 边缘部分：即用户使用的主机
  - ◇ 核心部分：由路由器连成的交换网络
- 从功能组成上看
  - ◇ 通信子网：由路由器和通信链路构成。
  - ◇ 资源子网：由连接到通信子网的服务器和主机构成

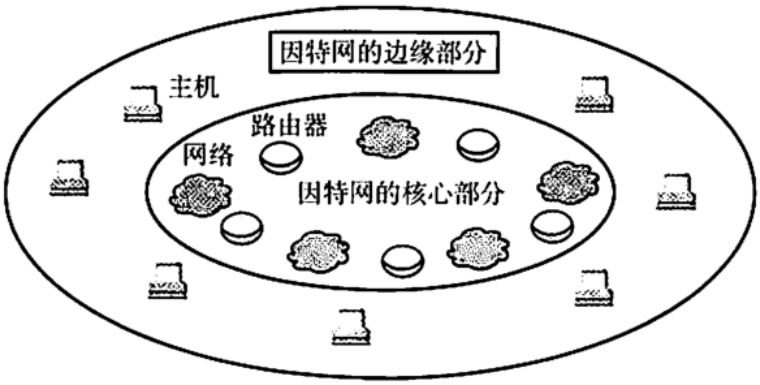

图 1.1　因特网的核心部分与边缘部分

1.1.3　计算机网络的功能

五大功能：

1. 数据通信：最基本的功能。
2. 资源共享：是软件、数据、硬件的共享，提高这些资源的利用率。
3. 分布式处理：将单一计算机的负荷分摊给其他计算机，利用空闲资源，提高系统利用率。
4. 提高可靠性：多台机器可以通过网络相互替代。
5. 负载均衡：将负载均衡地分配

1.1.4　计算机网络的分类

1. 按分布范围分类
   - 广域网（WAN）：数百公里以上。最大的广域网是 Internet
   - 城域网（MAN）：覆盖一个大城市，大约几十到上百公里。
   - 局域网（LAN）：覆盖大楼、校园或厂区，不超过数公里。
   - 个人区域网（PAN）：就是个人热点

   如果 CPU 的距离非常近（<1m），那么一般认为是多处理器系统而非网络。

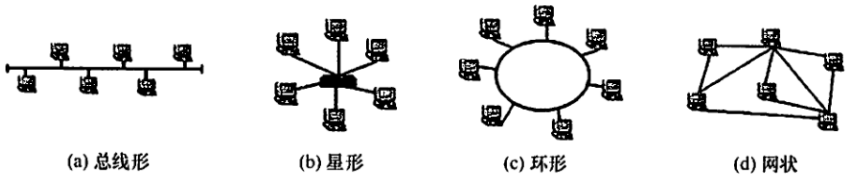

图 1.2　几种不同的网络拓扑结构

2. 按传输技术分类

   - 广播式网络：多用于局域网，也用于广域网中的无线、卫星通信。
   - 点对点网络：广域网基本上都是点对点的。
   - 二者的重要区别在于是否使用存储转发、路由选择机制。若是，则属于点对点网络。

3. 按拓扑结构分类

   - 总线形网络：建网方便，但高负载时效率低下，总线对故障敏感
   - 星形网络：依赖中央设备
   - 环形网络：典例：令牌环局域网
   - 网状网络：多用于广域网。又可分为：
     - ◇ 规则形网络
     - ◇ 不规则形网络

4. 按使用者分类

   - 公用网：电信公司出资建造的大型公用网络
   - 专用网：特定机构建立的专有网络

5. 按交换技术分类

   - 电路交换网络：包括建立连接、传输数据和断开连接三个阶段。
     - ◇ 优点：数据直接传送，时延小，实时性好。
     - ◇ 缺点：线路利用率低，不便于差错控制。
   - 报文交换网络：存储转发整个报文
     - ◇ 优点：线路利用率高

◇ 缺点：资源开销和时延都增大。

- 分组交换网络：将报文分成若干个分组后再进行存储转发。

现在的主流网络基本上都是分组交换网络。

6. 按传输介质分类
   - 有线网络
   - 无线网络

1.1.5 计算机网络的标准化工作

略

1.1.6 计算机网络的性能指标

1. **带宽（Bandwidth）**：在计网中，单位是比特/秒，表示网络的通信线路所能传输数据的能力。原指信号的频带宽度，单位是赫。
2. **时延（Delay）**：数据从链路的一端传送到另一端所需要的总时间。包括 4 部分：
   - 发送时延（传输时延）：节点将分组的所有比特发送到链路的时间。

$$\text{发送时延} = \frac{\text{分组长度}}{\text{信道带宽}}$$

   - 传播时延：比特从链路的一端传播到另一端需要的时间。

$$\text{传播时延} = \frac{\text{信道长度}}{\text{传播速率}}$$

   - 处理时延：节点进行处理消耗的时间。
   - 排队时延：数据在输出队列中等待转发的时间。

   总时延 = 以上 4 部分之和。做题时后两个一般不考虑。提速主要是减小发送时延。
3. **时延带宽积**：发送端发送的第一个比特即将到达终点时，发送端已经发出了多少个比特。

$$\text{时延带宽积} = \text{传播时延} \times \text{信道带宽}$$

4. **往返时延（RTT）**：发送端发出一个分组，到接收到确认经历的时延

5. **吞吐量（Throughput）**：单位时间内通过某个网络（或信道、接口）的数据量

6. **速率（Speed）**：主机在数字信道上传送数据的速率。指的是数据率或比特率。最高数据传输速率就是带宽。单位是 b/s，或 kb/s、Mb/s、Gb/s 等。

7. **信道利用率**：信道中有百分之多少的时间是有数据的。

$$\text{信道利用率} = \frac{\text{有数据通过时间}}{\text{总时间}}$$

网络利用率指的是整个网络的信道利用率的加权平均值。信道利用率不是越高越好。

若令 D_0 表示网络空闲时的时延，D 表示网络当前的时延，U 是信道利用率，则在适当的假定条件下，可以用下面的简单公式表示 D 和 D_0 之间的关系：

$$D = \frac{D_0}{1-U}$$

这意味着，利用率很高时，时延将急剧增大。

1.2　计算机网络体系结构与参考模型

1.2.1　计算机网络分层结构

把计算机网络的各层、层间接口及其协议的集合称为网络的**体系结构**
每层的报文分为两部分：SDU 和 PCI，它们共同组成 PDU。

- 服务数据单元（SDU）：为完成用户所要求的功能而应传送的**数据**。
- 协议控制信息（PCI）：**控制**协议操作的信息。
- 协议数据单元（PDU）：对等层次之间传送的数据单位称为该层的 PDU。

实际网络中，每层的协议数据单元都有一个通俗的名称，如：

- 物理层 PDU：比特
- 数据链路层 PDU：帧

- 网络层 PDU：分组
- 传输层 PDU：报文段

$$\text{n-SDU} + \text{n-PCI} = \text{n-PDU} = \text{(n-1)-SDU}$$

分层思想的优点：

- 耦合度低 (独立性强)
- 适应性强
- 易于实现和维护

1.2.2　计算机网络协议、接口、服务的概念

1. 协议：是规则的集合，是对等实体之间的共识。

   协议 = 语法 + 语义 + 同步（也称时序）

   - 语法：协议元素与数据的组合结构，即报文格式。
   - 语义：协议元素的含义。
   - 时序：通信双方执行的顺序和规则。

2. 接口

   - 相邻两层的实体通过服务访问点（SAP）进行交互。

3. 服务

   - 面向连接服务与无连接服务
   - 可靠服务和不可靠服务
   - 有应答服务和无应答服务

协议是水平的，服务是垂直的。

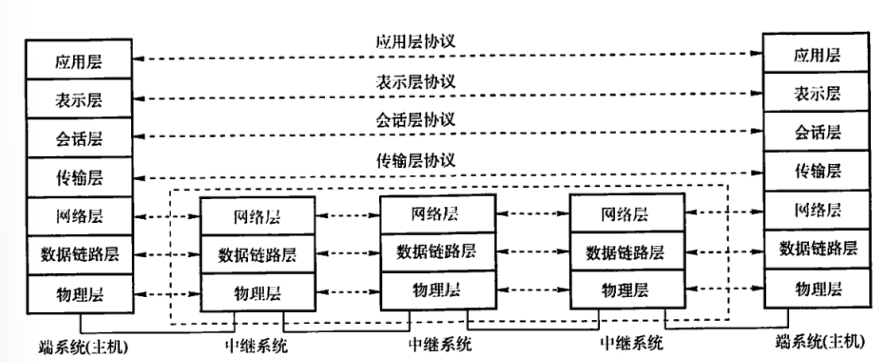

图 1: OSI 参考模型的层次结构

1.2.3　OSI 参考模型和 TCP/IP 模型

OSI 参考模型:

- 物理层
  - ◇ 传输单位：比特
  - ◇ 功能：在物理媒体上为数据端设备透明地传输原始比特流。
  - ◇ 标准：EIA-232C、EIA/TIA RS-449、CCITT 的 X.21 等
  - ◇ 物理媒体本身不属于物理层，可以认为是第零层。
- 数据链路层
  - ◇ 传输单位：帧
  - ◇ 功能：成帧、差错控制、流量控制、传输管理等。
  - ◇ 典型协议：SDLC、HDLC、PPP、STP 和帧中继等。
- 网络层
  - ◇ 传输单位：数据报
  - ◇ 功能：路由选择、流量控制、差错控制、拥塞控制、网际互连
  - ◇ 典型协议：IP、IPX、ICMP、IGMP、ARP、RARP、OSPF 等
- 传输层
  - ◇ 传输单位：报文段（TCP）或用户数据报（UDP）

  - ◇ 功能：为端到端连接提供可靠的传输服务，是通信子网和资源子网的分界层。
  - ◇ 典型协议：TCP、UDP
- 会话层
  - ◇ 功能：负责在网络中的两节点之间建立、维持和终止通信。
- 表示层
  - ◇ 功能：对上层数据或命令进行解释，以保证一个主机应用层信息可以被另一个主机的应用程序理解。
- 应用层
  - ◇ 典型协议：FTP、SMTP、HTTP 等（协议众多）

TCP/IP 模型和 OSI 模型的比较：
相同：

- 都采用分层的结构
- 都基于独立的协议栈的概念
- 都可以解决异构网络的互联

不同：

- OSI 精确定义了服务、协议和接口三大概念，TCP/IP 则没有明确区分。
- OSI 是先有模型后有协议，TCP/IP 则反之。
- OSI 是理论上的国际标准，TCP/IP 是事实上的国际标准。
- （常考）OSI 在网络层支持无连接和面向连接的通信，但在传输层仅有面向连接的通信。TCP/IP 在网际层仅有一种无连接的通信，但传输层支持无连接和面向连接两种方式。

另有一种综合了二者优点的五层结构协议模型，纯粹用于学术研究，没有实际意义。三种模型的比较见图2。

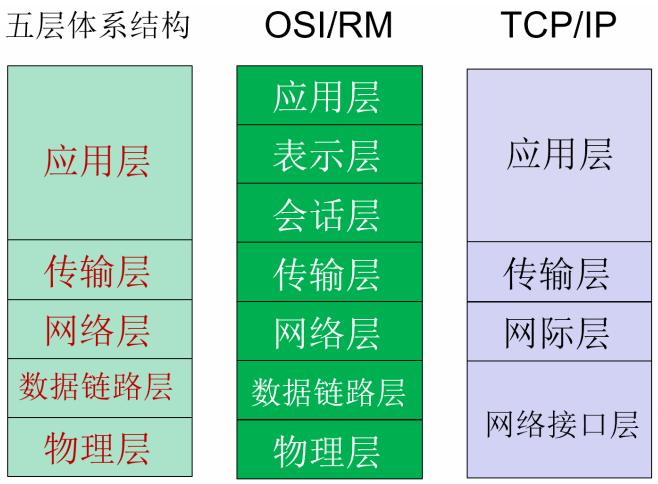

图 2: 三种模型的比较

二 物理层

2.1 通信基础

通信：将信息从一个地方传送到另外一个地方的过程。

2.1.1 基本概念

1. 数据、信号与码元

- 数据：传送信息的实体，传输方式可分为串行传输和并行传输，又分：
  - ◇ 模拟数据：可在某一区间内连续取值的数据。
  - ◇ 数字数据：可在某一区间内取有限个离散值的数据。
- 信号：数据的电气或电磁表现。又分：
  - ◇ 模拟信号：指幅度随时间作连续变化的信号。
  - ◇ 数字信号：指幅度随时间作不连续的、离散变化的信号。
- 码元：用一个固定时长的信号波形表示一位 k 进制数字。例如二进制编码时有两种码元，一种代表 0，一种代表 1。

2. 信源、信道与信宿

- 这三者是一个数据通信系统的主要组成部分。

- 信源是产生和发送数据的源头
- 信宿是接收数据的终点
- 信道是传输信号的通路。

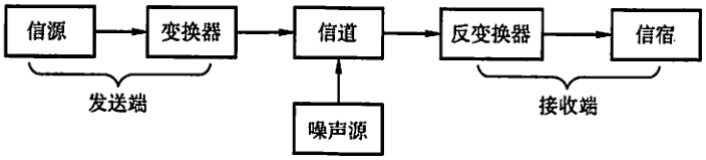

图 3: 单向通信系统模型

3. 速率、波特与带宽

- 速率指的是数据传输速率，可以用如下两种方式表示：
  - ◇ 码元传输速率（波特率）：表示通信系统传输码元的速率，单位是波特（1 波特表示每秒传输一个码元），与传输的进制无关。
  - ◇ 信息传输速率（比特率）：表示通信系统传输二**进制**码元的速率，单位是比特/秒。
  - ◇ 若一个码元携带 n 比特的信息量，则 M 波特率的码元传输速率对应的信息传输速率为 Mn 比特/秒。
- 带宽：见1.1.6节。

数据传输方式

- 从通信双方信息的交互方式看，可分为三**种**基本方式：
  - ◇ 单向通信：如无线电广播、电视广播
  - ◇ 半双工通信：可以双向发送信息，但任何一方都不能同时发送和接收信息。
  - ◇ 全双工通信：通信双方可以同时发送和接收信息。
- 从双方是否同步来看，可分为：
  - ◇ 同步方式
  - ◇ 异步方式

- 从串并行来看，可分为：
  - ◇ 串行方式
  - ◇ 并行方式

2.1.2 奈奎斯特定理

码间串扰现象：信道通过高频分量的效果往往较差，从而导致接收端接收到的信号波形失去码元之间的清晰界限。

奈奎斯特（Nyquist）定理指出：在理想低通（无噪声，带宽有限）的信道中，为了避免码间串扰，极限码元传输速率为 $2W$ 波特，其中 W 是理想低通信道的带宽。

根据奈奎斯特定理，可以得出如下结论：

1. 在任何信道中，码元传输速率是有上限的。若超过此上限，就会出现严重的码间串扰问题。
2. 频道的频带越宽，传输码元的极限速率就越高。

2.1.3 香农定理

香农定理给出了带宽受限且有高斯白噪声干扰的信道的极限数据传输速率：

$$极限速率C = W\log_2(1+\frac{S}{N})$$

其中

- W 为信道带宽
- S 为信道传输信号的平均功率
- N 为信号内部的高斯噪声功率

根据香农定理，可以得出如下结论：

1. 信道的带宽或信道中的信噪比（S/N）越大，信息的极限传输速率 C 就越高。$N \to 0$ 时，$C \to \infty$，因此无干扰信道容量为无穷大。
2. 对于一定的带宽和一定的信噪比，信息传送速率的上限是确定的。

3. 只要信息传输速率低于信道的极限速率，就能找到某种办法实现无差错的传输。

4. 设 N_0 为频谱密度，则有 $N = N_0 \times B$。当 $W \to \infty$ 时，$C \to 1.44\frac{S}{N_0}$，并不会趋于无穷大。

2.1.4 编码

1. **归零编码**。每个时钟周期的中间都要归零，提供了自同步机制，但归零要占用一点带宽。

2. **非归零编码**。不归零，但无法同步，需要额外的时钟线。

3. **反向非归零编码**。用信号的反转代表 0，信号保持不变代表 1，有同步且不会占用带宽。

4. **曼彻斯特编码**。将一个码元分成两个相等的间隔。实际上有两种截然相反的约定，较常用的一种是：前高后低为 0，1 则相反。也可同步。

   - 用于 802.3 局域网（以太网）。

5. **差分曼彻斯特编码**：若电平和上一个码元结束的电平一致，表示 1，否则表示 0。每个码元中间有一次跳变，用于同步。

   - 用于 802.5 局域网（令牌环网）。

2.1.5 调制

1. 数字信号转模拟信号

   - 幅移键控（ASK），又称调幅
   - 频移键控（FSK），又称调频
   - 相移键控（PSK），又称调相
   - 正交振幅调制（QAM）

2. 模拟信号转数字信号

   - 采样
   - 量化
   - 编码

2.1.6 电路交换、报文交换与分组交换

1. **电路交换**

   进行数据传输前，两个节点之间建立一条双方独占的物理通信路径。这条路径一直被独占到连接释放为止。

   优点：

   - 通信时延小。
   - 有序传输。
   - 没有冲突。
   - 适用范围广，适用于模拟信号和数字信号。
   - 实时性强。
   - 控制简单。

   缺点：

   - 建立连接时间长。
   - 线路独占，效率低。
   - 灵活性差。
   - 难以规格化。

2. **报文交换**

   优点：

   - 无须建立连接。
   - 动态分配线路。
   - 线路可靠性高。如果一条路径故障，还可以选择另一条路径。
   - 线路利用率高。
   - 可以提供多目标服务。

   缺点：

   - 由于每个节点都要存储-转发，因此会有转发时延。

- 要求节点有较大的缓存空间。

报文交换多用于早期的电报通信网络中，现已很少用。现在常用的是分组交换。

3. **分组交换**

优点：

- 无建立时延。
- 线路利用率高。
- 相对于报文交换，简化了存储管理。
- 加速传输。分组的传输可以并行化。
- 减少了出错概率和重发数据量。因分组较短，故出错概率必然减小。

缺点：

- 存在传输时延。
- 需要处理额外的信息量。
- 分组可能失序、丢失或者重复，需要在接收端重排检错。

分组交换比报文交换的时延小，尤其适合于计算机系统的猝发式通信。

2.1.7 数据报与虚电路

它们都属于网络层，都是分组交换的方式。

2.2 传输介质

2.2.1 双绞线、同轴电缆、光纤与无线传输介质

1. **双绞线**

最常用的古老传输介质，由两根并排绞合、相互绝缘的铜线构成。价格便宜，在局域网和电话网络中普遍使用。

2. **同轴电缆**

抗干扰性能良好，传输距离远，价格稍贵。

	数据报服务	虚电路服务
连接的建立	不需要	必须有
目的地址	每个分组都有完整的目的地址	仅在建立连接阶段使用，之后每个分组使用长度较短的虚电路号
路由选择	每个分组独立地进行路由选择和转发	属于同一条虚电路的分组按照同一路由转发
分组顺序	不保证分组的有序到达	保证分组的有序到达
可靠性	不保证可靠通信，可靠性由用户主机来保证	可靠性由网络保证
对网络故障的适应性	出故障的结点丢失分组，其他分组路径选择发生变化时可以正常传输	所有经过故障结点的虚电路均不能正常工作
差错处理和流量控制	由用户主机进行流量控制，不保证数据报的可靠性	可由分组交换网负责，也可由用户主机负责

图 4: 数据报和虚电路的比较

3. **光纤**

   带宽范围极大，传输损耗小，抗大电流脉冲干扰（如雷电），保密性佳，体积小，重量轻。

4. **无线传输介质**

2.2.2 物理层接口的特性

- 机械特性
- 电气特性
- 功能特性
- 过程特性

2.3 物理层设备

2.3.1 中继器

主要作用是将信号整形并放大再转发出去，以消除信号的失真和衰减。
中继器放大数字信号，另有一种放大模拟信号的设备是放大器。

2.3.2 集线器

集线器（Hub）实际上是一个多端口的中继器。集线器在收到数据信号后，进行整形放大，然后转发到其他所有端口（不含输入端口）。

集线器只能在半双工条件下工作。

三 数据链路层

数据链路层三大基本问题：

- **封装成帧**：为了实现传输中的检错纠错，以及部分重发功能，必须将物理层的比特流封装成帧来发送。实现方法是在帧的开始和结束位置插入帧定界符。
- **透明传输**：这是指无论被传数据是什么样的比特组合，都能够在链路上传送。解决办法是使用字节填充法或比特填充法。
- **差错控制**：这里的差错控制主要针对冲击噪声。解决办法是通过特殊的编码（差错控制码），使接收端能够发现甚至自动纠正错误。

3.1 数据链路层的功能

主要作用：加强物理层传输原始比特流的功能，将物理层提供的可能出错的物理连接改造为逻辑上无差错的数据链路。

3.1.1 为网络层提供服务

数据链路层通常可为网络层提供如下服务：

1. 无确认的无连接服务：无需预先建立连接，目的机器收到数据帧后无需确认。如以太网。
2. 有确认的无连接服务。如无线通信。
3. 有确认的面向连接服务。

3.1.2 链路管理

指的是数据链路层连接的建立、维持和释放过程。

3.1.3 帧定界、帧同步与透明传输

将网络层数据的前后分别添加首部和尾部，就构成了帧。

透明传输问题：不管所传数据是什么样的比特组合，都应当能在链路上传送。

3.1.4 流量控制

实际上是限制发送方的数据流量，使之不超过接收方的能力限制。

运输层也有流量控制机制，但数据链路层的流量控制是控制相邻两个节点之间的流量，而传输层控制的是源端到目的端之间的流量。

3.1.5 差错控制

数据链路层中的错误可以分成位错和帧错两种。

- **位错**。指帧当中的某些位出现了差错，通常采用循环冗余校验（CRC）方法发现错误，通过自动重传请求（ARQ）方法重传。接收方检测到错误之后直接丢弃帧，发送方计时超时后就自动重传。
- **帧错**。指帧的丢失、重复、失序等错误。数据链路层引入定时器和编号机制，确保此类错误被正确解决。

差错出现的原因可以是热噪声或冲击噪声，差错控制机制主要处理冲击噪声造成的差错。

3.2 组帧

3.2.1 字符计数法

指的是在帧头部使用一个计数字段来标明帧内的字符数。最大的问题是一旦这个字段出错，那么双方就立即失去同步，后面的所有帧都要废弃。

3.2.2 字符填充的首尾定界符法

指的是使用特殊的字符来标志一帧的开始和结束。

3.2.3 零比特填充的首尾标志法

使用一个特定的比特模式即 01111110（0+6 个 1+0）来标志一帧的开始和结束。发送方在正式的数据中如果遇到连续 5 个 1，就在后面插入一个 0，这样不致和首尾标志混淆。接收方遇到连续 5 个 1 后，检查后面的位，如果是 1，表示是首尾标志；如果是 0，表示是数据，将删去这个 0。

3.2.4 违规编码法/违例编码法

一些编码中的某些模式是“违规”的，即正常情况下一定不会出现，可以利用这种违规序列来标志首尾。局域网的 IEEE802 标准就用了此法。

目前最常用的后两者。

3.3 差错控制

3.3.1 检错编码

检错编码都采用冗余编码技术。常见的检错编码有奇偶校验码和循环冗余（CRC）码。

CRC 码有纠错功能，但数据链路层只使用了它的检错功能。

编码效率 R：码字中信息位所占的比例。若码字中信息位为 k 位，编码时外加的冗余位为 r 位，则编码后的码字长为 $n = k + r$ 位，这时有

$$R = \frac{k}{n} = \frac{k}{k + r}$$

1. **奇偶校验码**

   是通过增加冗余位来使得码字中 1 的个数保持奇数或偶数的编码方法。通信中又可分为：

   - 垂直奇偶校验
   - 水平奇偶校验
   - 水平垂直奇偶校验

2. **循环冗余（CRC）码**

   模 2 运算：不看进位和借位，加减就是异或。

   模 2 除的上商原则（重要）：

- 部分余数的首位为 1 时，上 1
- 部分余数的首位为 0 时，上 0
- 部分余数的位数少于除数的位数时，停止除法，该部分余数就是最后的余数。
- 也就是说，上商只无脑看首位数字，不比较大小。

引入一种多项式记法，这种多项式和一个二进制位序列是一一对应的。比如，1101 对应的多项式 $M(x)$ 为：

$$M(x) = 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$$

现有待计算的信息位串 $M(x)$，又有固定的生成多项式 $G(x)$，则计算 CRC 码的过程如下：

(a) 令 $r = G(x)$位数 $- 1$，如 $G(x) = 1011$，4 位，则 $r = 3$。

(b) 将 $M(x)$ 左移 r 位。右边补零。

(c) 用模 2 除的方法计算

$$\frac{M(x) \cdot x^r}{G(x)}$$

得到余数，把这余数加到左移后的 $M(x)$，就得到 CRC 编码。

接收到 CRC 码 $T(x)$ 后，将 $T(x)$ 与同一生成多项式 $G(x)$ 作模 2 除运算，如果余数为 0，表示无误，否则表示出现错误。

例题 1. 信息位串为 1010001，生成多项式 $G(x) = x^4 + x^2 + x + 1$，求 CRC 码。

解答：

1. $M(x) = x^6 + x^4 + 1$，由 $G(x)$ 得 $r = 4$。
2. $M(x)$ 左移 4 位，得到 10100010000。
3. 计算 $\frac{M(x) \cdot x^r}{G(x)}$ 的余数，结果为 1101，过程见图5。
4. 将 1101 加到左移后的 $M(x)$，得到 CRC 码 10100011101。

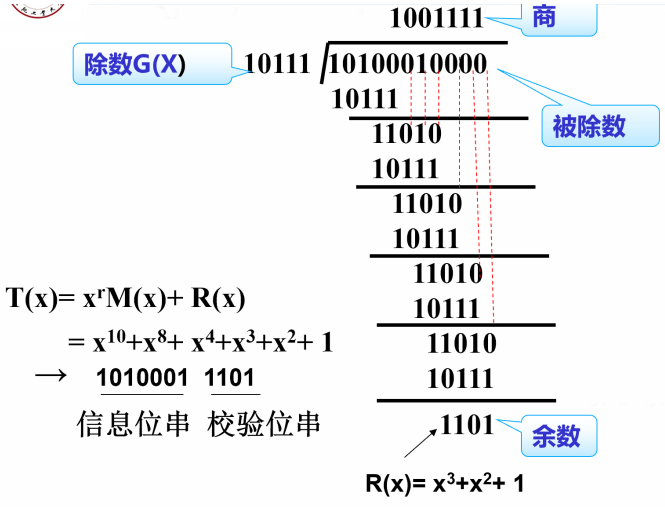

图 5: 模 2 除的计算过程

3.3.2 纠错编码

最常见的纠错编码是海明码。

3.4 流量控制与可靠传输机制

3.4.1 流量控制、可靠传输与滑动窗口机制

流量控制的基本方法是由接收方控制发送方发送数据的速率。常见的方式是停止-等待协议和滑动窗口协议。

- 停止-等待协议：发送方每发送一帧，就必须等待接收方的应答信号，才能发送下一帧。效率很低。
- 滑动窗口协议：发送端每收到一个帧的确认，发送窗口就向前滑动一个帧的位置；接收端收到数据帧后，将接收窗口向前移动一个位置，并发回确认帧。若收到的数据帧落在窗口之外，则一律丢弃。

数据链路层的滑动窗口协议中，窗口大小是固定的，这一点与 TCP 不同。如果滑动窗口的大小为 1，可保证帧的有序接收。
传统的自动重传请求（ARQ）可分为三种：

- 停止-等待 ARQ

- 后退 N 帧 ARQ
- 选择性重传 ARQ

3.4.2 单帧滑动窗口与停止-等待协议

3.4.3 多帧滑动窗口与后退 N 帧协议（GBN）

在后退 N 帧 ARQ 中，发送方可以连续发送帧，不需要等待确认。如果接收方检测出失序的帧，就要求发送方重发第一个出错帧（含）之后的所有帧。也就是说，接收方只允许顺序接收帧，对某个帧的确认就表明该帧之前的所有帧都已正确无误地收到。

若信道的误码率较大，后退 N 帧 ARQ 未必就比停止-等待 ARQ 好。

3.4.4 多帧滑动窗口与选择重传协议（SR）

3.5 介质访问控制

介质访问控制的任务是，为使用介质的每个节点隔离来自同一信道上的其他节点所传送的信号。即采取一定的措施，使得两对节点之间的通信不会发生互相干扰的情况。

常见的介质访问控制方法包括：

- 信道划分介质访问控制（静态划分信道）
- 随机访问介质访问控制（动态分配信道）
- 轮询访问介质访问控制（动态分配信道）

3.5.1 信道划分介质访问控制

信道划分的实质是：通过分时、分频、分码等方式把原来的一条广播信道，逻辑上分为几条用于两个节点之间通信的互不干扰的子信道。实际上是把广播信道转变为点对点信道。

信道划分介质访问控制可以分为以下 4 种：

1. **频分多路复用（FDM）**

是指将多路基带信号调制到不同的频率载波上，叠加形成一个复合信道的多路复用技术。为了避免不同子信道的干扰，需要在相邻信道之间加入“保护频带”。

优点是技术成熟，实现容易，系统效率高。

2. **时分多路复用（TDM）**

   是指将一条物理信道按时间分成若干时间片，轮流地分给多个信号使用，每个时间片由当前的信号独占。

   传统 TDM 是机械地按照信号数量平均分配时间片，而改进的**统计时分多路复用（STDM）**允许按照需要动态分配时间片，某个信号在其他信号都发送数据时，有可能独占全部时间。

3. **波分多路复用（WDM）**

   这实际上是光纤条件下的概念，是指光的频分多路复用。由于光的波长（频率）不同，各路光信号之间互不干扰。

4. **码分多路复用（CDM）**

   指的是采用不同的编码来区分各路原始信号的多路复用方式。CDM 同时共享了时间和频率。

   实际上更常用的术语是码分多址（CDMA）。

   CDM 频谱利用率高、抗干扰能力强、保密性好，还可以降低成本，多用于无线通信系统。

3.5.2 随机访问介质访问控制

随机访问介质访问控制中，不采用中心化的控制方式，而是所有用户根据自己的意愿随意发送信息，但发生冲突时必须重传冲突的帧，直到帧被正确接收。重传的规则称为随机访问介质访问控制协议。常用的协议有：

1. **ALOHA 协议**

   又可分为纯 ALOHA 和时隙 ALOHA 两种。

   (a) **纯 ALOHA 协议**

   站点可以不经过任何检测就发送数据。如果一段时间内没有收到确认，就等待一段随机时间后重发，直至发送成功。吞吐量很低。

(b) **时隙 ALOHA 协议**

让所有站点在时间上同步，并把时间划分成若干等长的时隙（Slot），规定只能在每个时隙开始时发送一个帧。处理冲突的策略和纯 ALOHA 一样。

2. **CSMA 协议**

载波侦听多址访问（CSMA）协议和 ALOHA 的区别是多了一个载波侦听装置，发送前需要先侦听信道是否空闲。

根据侦听方式和侦听到信道忙之后的处理方式不同，CSMA 协议可分为三种。

(a) **1-坚持 CSMA**

首先侦听信道，如果信道空闲，就立即发送数据；如果信道忙，就持续侦听直至信道空闲，然后立即发送数据。如果遇到冲突，那么等待随机一段时间后再侦听信道。

“1-坚持”的含义是：侦听到信道忙后，继续坚持侦听信道；侦听到信道空闲后，发送帧的概率为 1，即立即发送数据。

(b) **非坚持 CSMA**

和 1-坚持 CSMA 的不同是，如果信道忙，就放弃侦听，等待一段随机时间后再侦听信道。

非坚持 CSMA 降低了冲突发生的概率，但是会增加网络的延迟。

(c) ***p*-坚持 CSMA**

首先侦听信道，如果信道忙，就持续侦听到信道空闲，；如果信道空闲，就以概率 p 发送数据，以概率 $1-p$ 推迟到下一个时隙，下一个时隙依然如此，直到成功发送数据或者侦听到信道忙为止。

信道状态	1-坚持	非坚持	p-坚持
空闲	立即发送数据	立即发送数据	以概率 p 发送数据，以概率 $1-p$ 推迟到下一个时隙
忙	继续侦听	放弃，等待一个随机事件后再侦听	持续侦听，直至信道空闲

表 1: 三种不同类型的 CSMA 协议的比较

3. **CSMA/CD 协议**

这是 CSMA 的改进版本，用于有线连接的局域网，如总线形网络或半双工网络环境。发送数据前先侦听总线上有无其他站点正在发送数据，如有则

等待至空闲，若无则发送数据，但同时要持续监听总线上有无其他信号。若检测到冲突，就发送一个拥塞信号（jamming signal），让其他所有站点都知道。中止发送后，执行指数避退算法。

注意：监听时发现信道空闲，不意味着信道就真的是空闲的，可能其他站点也在发送数据，但数据还在传播过程中。

设端到端传播时延为 τ，则某个站发送帧之后至多经过 2τ 就可以得知是否发生碰撞。因此把这个 2τ 称为争用期（或冲突窗口、碰撞窗口）。

CSMA/CD 要求一个帧的传输时延必须大于 2τ，否则一个帧发送完毕后就无法再跟踪它是否出现冲突了。凡小于 2τ 的帧都是无效帧，应该直接丢弃。如果一个帧确实没有 2τ 长，就应该在后面加入填充字段。

二进制指数避退算法

- 确定基本避退时间，一般取 2τ。
- 定义参数 k，其中 $k = \min[\text{重传次数}, 10]$。
- 从整数集合 $[0, 1, \cdots, 2^k - 1]$ 中随机取出一个数 r，重传的避退时间就是 $2\tau r$。
- 重传 16 次后仍然不成功时，放弃重传，并向高层报告错误。

4. **CSMA/CA 协议**

适用于无线连接的局域网。与 CSMA/CD 的不同是，CSMA/CA 要尽量避免碰撞的发生。

3.5.3 轮询访问：令牌传递协议

3.6 局域网

3.7 广域网

3.8 数据链路层设备

四 网络层

4.1 网络层的功能

4.1.1 异构网络互联

中继系统分为以下 4 种：

- 物理层：转发器、集线器
- 数据链路层：网桥、交换机
- 网络层：路由器
- 网络层以上：网关

前两层只是把网络扩大了，从网络层的角度看，并不能算是网络互联。

- 路由器不转发广播包，因此能够分隔广播域，抑制网络风暴。
- 交换机可以分隔冲突域，但不能分隔广播域。
- 集线器和中继器既不能分隔冲突域又不能分隔广播域。

使用 IP 网络的好处：屏蔽互联的各网络的异构细节（如编址方案、路由选择协议等）。

例题 2. 在路由器互联的多个局域网的结构中，要求每个局域网（ ）

解答：物理层、数据链路层、网络层协议可以不同，高层协议必须相同

注记. 网络层协议不同而实现互联的例子：使用特定的路由器连接 IPv4 和 IPv6 网络

4.1.2 路由与转发

路由器的两大任务：

- 路由选择
- 分组转发

路由表是由路由选择算法得出的，转发表是从路由表得出的。

4.1.3 SDN 的基本概念

待补

4.1.4 拥塞控制

在通信子网中，因出现过量的分组而引起网络性能下降的现象称为**拥塞**。
通过观测网络吞吐量随网络负载的变化关系，可以估计网络的拥塞程度。
拥塞控制的方法：

- **开环控制**。静态的方法，在设计网络时考虑各种因素，一旦确定就不再更改。
- **闭环控制**。动态的方法，监视网络的状态，检测哪里发生了拥塞，并调整网络。

拥塞控制和流量控制的区别：

- 流量控制指的是发送端和接收端之间的点对点通信量的控制。
- 拥塞控制是全局性问题，涉及整个网络的所有主机、路由器。

4.2 路由算法

4.2.1 静态路由与动态路由

- 静态路由算法（非自适应路由算法）：需要网络管理员手动配置路由信息，不能及时适应网络状态的变化。适用于小型网络。
- 动态路由算法（自适应路由算法）：可以动态适应网络状态。又分：
  - ◇ 距离向量路由算法（D-V）：典例是 RIP 算法
  - ◇ 链路状态路由算法（L-S）：典例是 OSPF 算法

4.2.2 距离向量路由算法

所有的节点都定期地将它们的整个路由表发给**相邻**的节点。这表包含：

- 每条路径的目的地（另一节点）
- 路径的代价（距离）

D-V 协议中，好消息传得快，坏消息传得慢。当路由信息发生变化时，旧的信息可能还在网络中传递。这就是“慢收敛”。慢收敛是导致路由环路的根本原因。

可以想见，网络规模越大，交换的信息也越多。因此这种算法不适合大规模网络。

4.2.3 链路状态路由算法

定期将自己与**相邻节点**的链路状态（如距离、时延、带宽、费用等）向全网**所有节点**泛洪。之后各节点通过 Dijkstra 算法生成路由表。

L-S 算法发送的数据链不会随着网络规模的增大而增长，因而适合大规模网络。

4.2.4 层次路由

Internet 将整个互联网划分为许多较小的自治系统。每个自治系统有权决定系统内部使用何种路由选择协议。

- 一个自治系统内部使用的路由选择协议称为**内部网关协议（IGP）**，具体包括 RIP 和 OSPF 等。

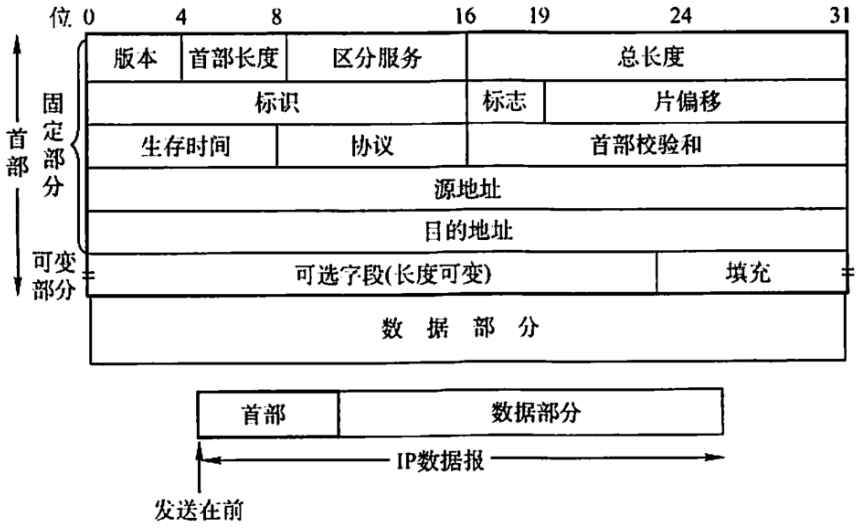

图 6: IP 数据报的格式

- 自治系统之间使用的路由选择协议称为**外部网关协议（EGP）**，如 BGP 协议。

4.3 IPv4

IP 需要 ICMP、ARP、RARP 等路由协议的支持。

4.3.1 IPv4 分组

IP 分组 = 首部 + 数据。
首部 = 固定字段（20B 长）+ 可选字段

1. **版本**。指 IP 协议的版本，可以是 4 或者 6。
2. **首部长度**。最大值是 60B，最常用的是 20B（无扩展）
3. **区分服务**，用来指明要求网络提供的服务，实际上从未使用过。
4. **总长度**，指整个 IP 分组的长度。最长为 65536B。
5. **标识**，用来标识数据报的顺序。如果数据报产生了分片，那么所有分片将共用同一个标识。

6. **标志**，占 3 位，第二位是 MF，第三位是 DF，第一位（最高位）保留。
7. **片偏移**，如果产生分片，则本字段指出当前分片相对于数据报头的偏移量。单位是 8B，所以除最后一个分片外，所有分片长度都是 8B 的整数倍。
8. **生存时间 TTL**，标识分组的寿命。路由器转发分组前先把 TTL 减一。若 TTL 减至 0 则丢弃。
9. **协议**，指分组携带的数据使用的传输层协议。值 6 表示 TCP，17 表示 UDP。
10. **首部校验和**，只校验首部，不校验数据部分。
11. **源地址**，标识发送方的 IP 地址。
12. **目的地址**，标识接收方的 IP 地址。

首部长度、总长度、片偏移是 IP 分组中有关长度的 3 个字段。它们的基本单位分别是 4B、1B 和 8B（常考加减）。

一个链路层数据报能承载的最大数据量称为最大传送单元（MTU）

对标志字段的进一步解释：

- DF（Don't Fragment）表示数据报是否允许分片，1 不允许，0 允许。
- MF（More Fragment）表示后面还有没有更多的分片。1 表示还有，0 表示没有（即当前分片是最后一个分片）。

4.3.2 IPv4 地址与 NAT

图7示出了传统分类 IP 地址的分类情况。

IP 地址::={< 网络号 >，< 主机号 >}

特殊的 IP 地址：

- 主机号全 0 表示主机所在的网络本身。
- 主机号全 1 表示主机所在的网络的广播地址。
- 127.*.*.* 保留为环回自检地址。
- 0.0.0.0 表示本机。系统启动时用到，启动后不再使用。
- 255.255.255.255 表示整个网络的广播地址，实际上是本网络的广播地址。

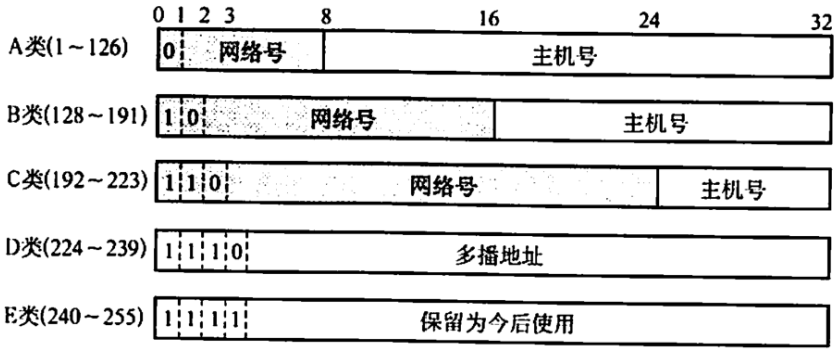

图 7: 分类的 IP 地址

网络地址转换（NAT）技术将专用地址转换为公用地址，从而大大节省了 IP 地址的消耗。

本地主机与外部网络通信时，NAT 路由器通过 NAT 转换表完成本地 IP 和全球 IP 的转换。转换表的项都是 {本地 IP: 端口}-{全球 IP: 端口} 的形式。

普通路由器在转发 IP 数据报时，不改变源 IP 地址和目的 IP 地址，但 NAT 路由器在转发时一定要改变 IP 地址。普通路由器工作在网络层，NAT 路由器则会查看和转换传输层的端口号。

4.3.3 子网划分、子网掩码、CIDR

为 IP 地址再添加一个“子网号”字段，使两级 IP 地址变成三级 IP 地址，即子网划分。

子网号其实是占用了主机号的一部分，而网络号仍然保持不变。

某个子网内，主机号全 0 的 IP 地址就是这个子网的网络号，主机号全 1 的 IP 地址是这个子网内部的广播地址。这两个地址都不能分配给主机。

使用传统 ABC 类 IP 划分的子网，子网号不能为全 0 或全 1。考虑一个主网，包含若干子网，则第一个子网的子网号全 0 时，该地址是主网的网络号；最后一个子网的子网号全 1 时，该地址是主网内部的广播地址，这样会有歧义。

现在所有的 IP 地址都必须使用子网掩码。如果网络内部没有划分子网，则使用对应类的 IP 地址的默认子网掩码：

- A 类地址：255.0.0.0
- B 类地址：255.255.0.0

- C 类地址：255.255.255.0

无分类域间路由选择 CIDR 在变长子网掩码的基础上消除了 A、B、C 类 IP 地址，实现超网构造，是目前应用最广泛的方法。

CIDR 的子网号没有“不得为全 0 或全 1”的限制。但 CIDR 子网内部的主机仍然不得使用全 0 和全 1 的主机号。

最长前缀匹配原则：使用了 CIDR 的路由表的每个项目由“网络前缀”和“下一跳地址”组成。查找路由表时可能得到不止一个结果。此时应该优先选择最长的匹配结果。

4.3.4 ARP、DHCP 和 ICMP

地址解析协议（ARP）完成了 IP 地址到 MAC 地址的映射。ARP 工作在网络层。

工作过程：每台主机都设有一个 ARP 表，用来维护本局域网内的各个主机和路由器的 IP 地址到 MAC 地址的映射关系。假设主机 A 欲向本局域网内的主机 B 发送信息，则 A 首先查看自己的 ARP 表，查找是否有匹配的 IP 地址。若有，则将数据发往对应的 MAC 地址；若无，就广播一个 ARP 请求分组。B 收到分组后，向 A 发送 ARP 响应分组，其中包含 B 的 IP 地址和 MAC 地址的映射关系。A 收到响应后，就将这个映射写入 ARP 表，并向对应 MAC 地址发送数据。

无偿 ARP：主机发送 ARP requset 报文查询自己的 IP 地址。作用：

- 确定网络中是否有其他的主机使用了该 IP 地址，如果有应答则产生错误消息。
- 无偿 ARP 可以更新 ARP 表项用，网络中其他主机收到该广播则在缓存中更新条目，收到主机强制更新，如果存在旧条目会将 MAC 更新为广播包中 MAC。

动态主机配置协议（DHCP）用于动态地给主机分配 IP 地址。

工作过程待补

DHCP 是应用层协议，基于 UDP。

网际控制报文协议（ICMP）是 IP 层协议。控制报文作为 IP 层数据，组成 IP 数据报发送出去。

ICMP 报文可分为两类

- ICMP 差错报告报文
- ICMP 询问报文

ICMP 最常见的应用是 ping 和 traceroute。ping 工作在应用层，但是跳过传输层直接使用了网络层服务；traceroute 工作在网络层。

4.4 IPv6

4.4.1 IPv6 的主要特点

4.4.2 IPv6 地址

4.5 路由协议

4.5.1 路由信息协议（RIP）

RIP 要求每条路径最多包含 15 跳，若更多（如 16）则认为不可达。
RIP 是应用层协议，它使用 UDP 传送数据（端口为 520）
RIP 的特点：

- 仅和相邻路由器交换信息。
- 路由器交换的信息是自己的路由表。
- 按照固定的时间间隔交换信息，默认值为 30 秒。

RIP 的缺点：

- 限制了网络规模，因为它规定大于 15 跳的网络都不可达。
- 路由器交换的是完整的路由表，因此网络规模越大，交换的信息量也越大。
- 路由器出现故障时，会出现慢收敛现象

4.5.2 开放最短路径优先（OSPF）协议

OSPF 是网络层协议，直接通过 IP 数据报传送。

4.5.3 边界网关协议（BGP）

BGP 是基于 TCP 的应用层协议。
其余待补

4.6 IP 组播

待补

4.7 网络层设备

4.7.1 冲突域和广播域

- 冲突域：是指连接到同一物理介质上的所有节点的集合，这些节点之间存在介质争用的现象。集线器、中继器等无脑转发信号的第 1 层设备所连接的节点都属于同一个冲突域。而第 2 层设备（网桥、交换机）、第 3 层设备（路由器）都可以划分冲突域。
- 广播域：是指接收同样广播消息的节点集合。第 1、2 层设备所连接的节点都属于同一个广播域。第 3 层的路由器则可以划分广播域。

通常所说的局域网（LAN）特指使用路由器分割的网络，也就是广播域。

4.7.2 路由器的组成和功能

若源主机和目标主机处于同一个网络，那么就**直接交付**而无需通过路由器。如果不在同一个网络，则需要路由器按照路由表将数据报转发给下一个路由器，这称为**间接交付**。

4.7.3 路由表与路由转发

路由表总是用软件实现的，转发表可以用软件实现，也可以用特殊的硬件实现。

路由表 $\neq$ 转发表。分组的实际转发是靠直接查找转发表，而不是直接查找路由表。

五 传输层

5.1 传输层提供的服务

5.1.1 传输层的功能

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。路由器只实现了下三层。

5.1.2 传输层的寻址与端口

数据链路层的 SAP 是 MAC 地址，网络层的 SAP 是 IP 地址，传输层的 SAP 是端口。

端口号可分为两类：

- 服务器端使用的端口号，又分两类：
  - ◇ 熟知端口号，范围是 0-1023
  - ◇ 登记端口号，数值为 1024-49151
- 客户端使用的端口号：数值为 49152-65535

套接字 Socket=(IP 地址：端口号)

5.1.3 无连接服务与面向连接服务

TCP 提供全双工的可靠逻辑信道。TCP 不提供广播或组播服务。

5.2 UDP 协议

5.2.1 UDP 数据报

UDP 仅在 IP 的基础上添加了两个最基本的服务：复用和分用以及差错检测。如果用户使用 UDP，那么必须由应用层提供全部的可靠性工作。

UDP 的优点：

- 无需建立连接，因此没有连接的时延。
- 无连接状态。
- 分组首部开销小。TCP 首部有 20B，UDP 首部仅有 8B。
- 没有拥塞控制，因此应用层不会经历太高的时延。
- 支持一对一、一对多、多对一和多对多的通信。

UDP 数据报 =UDP 首部 + 用户数据
UDP 首部固定为 8B 长，分为 4 个字段，每个字段长度都是 2B（16 位）。

- **源端口号**。如果不需要对方回信，可以设为全 0。

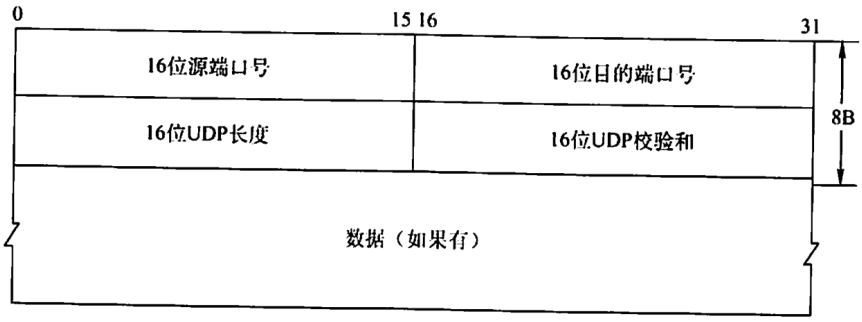

图 8: UDP 数据报的格式

- **目的端口号**，必须指定。
- **长度**。指的是首部 + 数据部分的长度，最小值是 8（即只有首部的情况）
- **校验和**。如果不希望进行校验，可以设为全 0。

如果接收方发现报文中指定的端口号找不到对应的应用进程，就丢弃 UDP 数据报，并回传一个“端口不可达”的 ICMP 报文。

5.2.2 UDP 校验

计算校验和时，需要在原本的首部前面加上 12B 的伪首部。这伪首部不参与发送，仅仅用于计算校验和。伪首部实际上是 IP 分组报头的一部分。

IP 数据报的校验和只检验 IP 数据报的首部，而 UDP 的校验和则检验整个数据报。

如果数据部分不是偶数个字节长，则需要添加 1 个全 0 字节。

5.3 TCP 协议

5.3.1 TCP 协议的特点

- 面向连接。TCP 连接是一条逻辑连接。
- TCP 是端到端的（进程到进程）
- 提供可靠交互服务，保证数据无差错、不丢失、不重复、不失序。
- 提供全双工通信，因此发送端和接收端都有发送和接收缓存。

- 面向字节流。

UDP 报文的长度由应用进程决定，TCP 报文的长度取决于接收方的接收窗口大小和网络的拥塞程度。

5.3.2 TCP 报文段

TCP 报文段 = 首部 + 数据
首部的前 20B 是固定的，如果需要，可以附加 4N 个字节的选项。

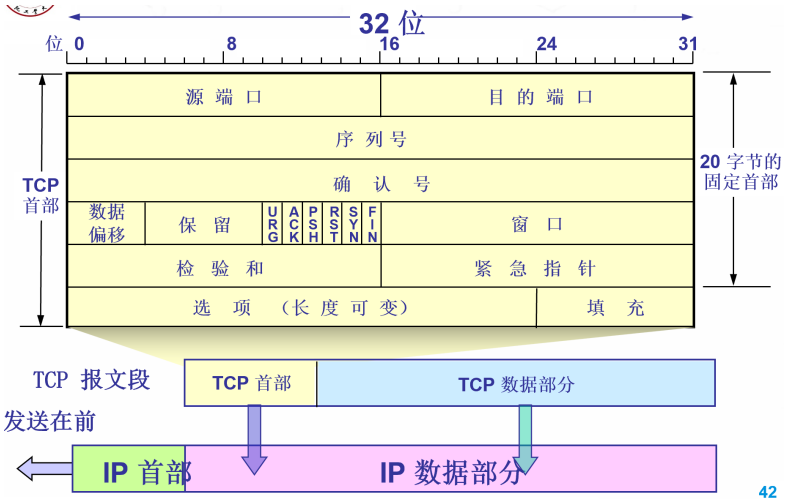

图 9: TCP 报文段的格式

- **源端口和目的端口**。各占 2B。
- **序号**，占 4B。指的是本报文段发送的数据的第一个字节的序号。
- **确认号**，占 4B。指的是**期望**收到下一个报文段的第一个数据字节的序号。若确认号为 N，则表示到序号 $N-1$ 为止的所有数据都已正确收到。
- **数据偏移**，占 4 位。表示 TCP 首部的长度。这长度的单位是 4B，由于 4 位二进制表示的最大值是 15，因此 TCP 首部最长可以达到 60B。
- **保留**，占 6 位。目前恒为全 0。
- **紧急位 URG**，和紧急指针字段配合使用。URG=1 时，表示报文段中有紧急数据，需要尽快发送。

- **确认位 ACK**，仅当 ACK=1 时确认号才有效，否则确认号无效。建立 TCP 连接后，所有的报文段的 ACK 都等于 1。
- **推送位 PSH**，接收方接收到 PSH=1 的报文段时，应当立即提交到应用层，而不是放在缓存里等待。
- **复位位 RST**，RST=1 时，表示出现严重差错，必须断开 TCP 连接。
- **同步位 SYN**，SYN=1 时表示这是一个连接请求或连接接受报文。
- **终止位 FIN**，用来释放 TCP 连接。FIN=1 时，表明发送方要求释放连接。
- **窗口**，占 2B。表示当前允许对方发送的数据量。
- **校验和**，占 2B。和 UDP 一样，校验和校验的是整个报文段，且也要加上 12B 的伪首部（除协议字段从 17 改成 6 之外，其他和 UDP 伪首部完全一致）。
- **紧急指针**，占 2B。指出紧急数据有多少字节（紧急数据位于数据部分的最前面）。
- **选项**，长度可变。
- **填充**，仅仅是为了使首部长度是 4B 的整数倍。

5.3.3 TCP 连接管理

每个 TCP 连接都有三个阶段：连接建立、数据传送和连接释放。

TCP 连接采用 C/S 模式，主动发起连接的进程是 Client，被动等待连接的进程是 Server。

- **TCP 连接的建立：三次握手**
  - ◇ 第一步：客户机向服务器发送连接请求报文段，字段设置如图10所示。SYN 报文段不能携带数据，但要消耗一个序号。此时 TCP 客户进程进入 SYN-SENT 状态。
  - ◇ 第二步：服务器接收到请求后，如同意，则发回确认，并为连接分配响应的资源。字段设置如图10所示。确认报文段也不能携带数据、要消耗一个序号。此时 TCP 服务器进程进入 SYN-RCVD 状态。

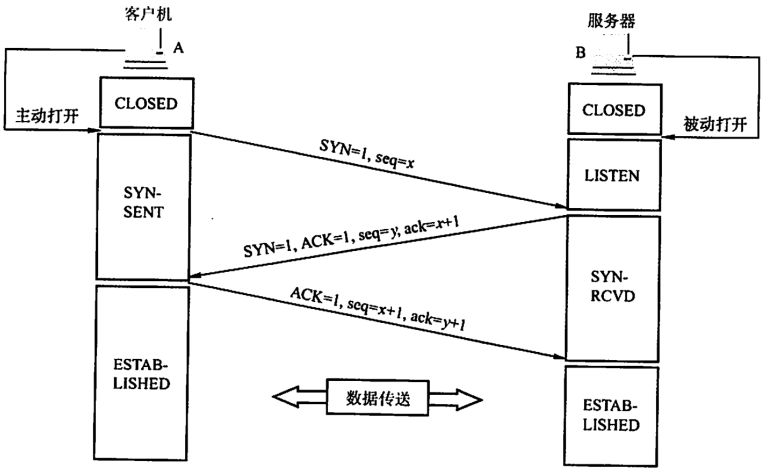

图 10: 三次握手过程

◇ 第三步：客户机收到确认后，也要回传确认，并为连接分配响应的资源。字段设置如图10所示。此时，客户端进入 ESTABLISHED 状态。此报文段可以携带数据。

由于服务器先分配资源，客户端后分配资源，因此服务器容易受到 SYN 洪泛攻击。

TCP 连接一旦建立，再次尝试建立连接将会失败，原先的连接不受影响。

- **TCP 连接的释放：四次挥手**

◇ 第一步，客户机希望关闭连接时，发送一个 FIN=1 的报文段，进入 FIN-WAIT-1 状态。由于 TCP 是全双工的，此时对方仍然可以发送数据。

◇ 第二步，服务器收到连接释放报文后发回确认，进入 CLOSE-WAIT 状态。此时，从客户端到服务器这个方向的连接就已经释放，但服务器仍然可以继续发送数据。

◇ 第三步，服务器不需要再发送数据时，发出 FIN=1 的报文段，进入 LAST-ACK 状态。

◇ 第四步，客户机收到连接释放报文段后，发回确认，服务器收到确认后进入 CLOSED 状态。此时 TCP 连接还未释放，必须等待计时器超过 2MSL 后（此时认为网络中所有的报文都已到达目的地），客户机也进入 CLOSED 状态。

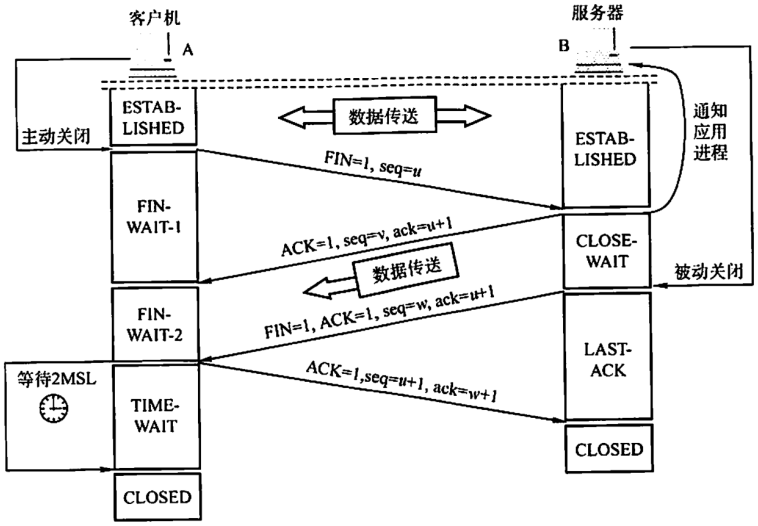

图 11: 四次挥手过程

5.3.4 TCP 可靠传输

TCP 使用校验、序号、确认和重传等机制来保证可靠连接。

TCP 默认使用累积确认，即 TCP 只确认数据流中至第一个丢失字节为止的字节，即使丢失字节后面有正常接收的字节。

有两种事件会导致 TCP 对报文段进行重传：

- 超时。TCP 每发送一个报文段，就会重置一次计时器。如果计时器超时而仍未收到确认，就重传报文段。为了计算超时重传时间，TCP 采用一种自适应算法（待补）：
- 冗余 ACK。TCP 规定每当接收端收到比期望序号更大的报文段到达时，就发送一个冗余 ACK 信号，指明下一个期待字节的序号。如果接收端累计收到对同一个报文段的三个冗余 ACK 时，可以认为报文已经丢失，于是立即重传。这种技术通常称为**快速重传**。

5.3.5 TCP 流量控制

TCP 提供一种基于滑动窗口协议的的流量控制机制。在通信过程中，接收方根据自己接收缓存的大小，动态地调整发送方的发送窗口大小，这称为**接收窗口 rwnd**；同时，发送方根据其对网络拥塞程度的估计而切丁**拥塞窗口 cwnd** 的大小。

5.3.6 TCP 拥塞控制

Internet 建议标准定义了进行拥塞控制的 4 种算法：慢开始、拥塞避免、快重传和快恢复。

发送窗口的上限值是上节提到的接收窗口和拥塞窗口中较小的一个，即

$$\text{发送窗口的上限值} = \min[\text{rwnd}, \text{cwnd}]$$

1. 慢开始算法

   TCP 连接刚刚建立时，先令拥塞窗口 cwnd=1，即一个最大报文段长度 MSS。每收到一个新的确认后，将 cwnd 加 1，即增大一个 MSS。

   需要注意当 cwnd 增加时，发送方一次可以发出更多报文段，则接收方的确认报文段也会增多，因此 cwnd 的增长速度会加快，从而呈现一种指数的趋势。所谓的“慢”指的不是 cwnd 增长慢，而是一开始限制 cwnd 到一个很小的值，从而使发送速率变得很慢。

   cwnd 增大到一个规定的慢开始门限值 ssthresh 时，改用拥塞避免算法。具体来说，如果当前 cwnd 再次翻倍后超过了 ssthresh，就将 cwnd 设为 ssthresh，并改用拥塞避免算法。注意此时不能再翻倍 cwnd 了。

2. 拥塞避免算法

   拥塞避免算法的思路是每经过一个 RTT 就让 cwnd 加 1，而不是翻倍，这样 cwnd 将会按照线性增长。

   - cwnd < ssthresh 时，使用慢开始算法。
   - cwnd > ssthresh 时，使用拥塞避免算法。
   - cwnd = ssthresh 时，两种都可用，通常用拥塞避免算法。

   无论在慢开始阶段还是拥塞避免阶段，只要检测到网络中出现拥塞，就立即把 ssthresh 设置为当前 cwnd 的一半（但不得小于 2），同时将 cwnd 设置为 1，重新执行慢开始。这样可以迅速减少主机发送的分组数量，给路由器争取出恢复的时间。

3. 快重传

   快重传和快恢复算法都是对慢开始和拥塞避免算法的改进。

   冗余 ACK 也可用于网络拥塞的检测。当发送方连续收到三个重复的 ACK 报文时，直接重传对方尚未收到的报文段。

4. 快恢复

   当发送方连续收到三个冗余 ACK 时，把 ssthresh 设置为此时发送方 cwnd 的一半，同时 cwnd 也减半（保持和 ssthresh 一致），然后执行**拥塞避免算法**，而不是慢开始算法。这样发送速率恢复得较快，因此叫“快恢复”。

注意本节为了简便，忽略了接收窗口 rwnd，认为发送窗口只取决于拥塞窗口（cwnd）。实际上发送窗口是由 rwnd 和 cwnd 共同决定的。

超时重传时间（RTO）的确定对 TCP 的性能有重大影响。

- 若 RTO<RTT，则会造成很多不必要的重传；
- 若 RTO≫RTT，则会浪费资源，降低网络利用率。

RTT 是一个动态更新的值，每个连接都有独立的 RTT，且同一连接在不同时刻的 RTT 也可能不同。

TCP 使用一个名为**加权平均往返时间 RTT**$_S$ 的参数反映往返时间。一般来说，RTO 应略大于 RTT$_S$。

若是第一次测量到 RTT 样本，则 RTT$_S$ 值就取为所测量到的 RTT 样本值；否则按照下式计算：

$$\text{新 RTT}_S = (1-\alpha) \times \text{旧 RTT}_S + \alpha \times \text{新 RTT}_S\text{样本}$$

其中，$0 \leq \alpha < 1$。RFC 2988 推荐的 α 值为 0.125。

RFC 2988 建议使用下式计算 RTO：

$$\text{RTO} = 4\text{RTT}_S + 4 \times \text{RTT}_D$$

由此又引入 RTT$_D$，若是第一次测量，RTT$_D$ 值取为测量到的 RTT 样本值的一半；否则，使用下式计算：

$$\text{新 RTT}_D = (1-\beta) \times \text{旧 RTT}_D + \beta \times |\text{RTT}_S - \text{新 RTT 样本}|$$

β 的推荐值为 0.25。

六 应用层

6.1 网络应用模型

6.1.1 客户/服务器模型

客户机是面向用户的，服务器是面向任务的。

6.1.2 P2P 模型

与 C/S 模型相比，P2P 模型的优点是：

- 减轻了服务器压力，消除了对某个服务器的完全依赖，大大提高系统效率和资源利用率。
- 多个客户机之间可以直接共享文档。
- 可扩展性好。
- 网络健壮性强，单个节点故障不会对整体造成太大影响。

6.2 域名系统（DNS）

DNS 系统采用 C/S 模型，运行在 UDP 之上，使用 53 号端口。

域名和 IP 地址并不是一一对应的关系。可以是一对一，也可以是一对多、多对一。

注记. 如果一台主机通过两张网卡接入两个网络，那么就具有两个 IP 地址。而一个域名可以映射到多台主机（负载均衡），一台主机也可以映射到多个域名（虚拟主机）。

6.2.1 域名服务器

DNS 使用了大量的域名服务器，它们以层次方式组织起来。这是典型的分布式系统。域名服务器有如下 4 种：

1. **根域名服务器**。层次最高。根域名服务器知道所有顶级域名服务器的 IP 地址。本地域名服务器如果无法解析域名，就要咨询根域名服务器。世界上有 13 个互为备份的根域名服务器。

2. **顶级域名服务器**。管辖在该顶级域名（如.com）下注册的所有二级域名。
3. **授权域名服务器**，又叫权限域名服务器。每台接入互联网的主机都必须在授权域名服务器处登记，可以登记不止一台。
4. **本地域名服务器**。一台主机发出 DNS 查询请求时，就是本地域名服务器响应了请求。

   实际上，很多服务器都同时充当了后两者。

6.2.2 域名解析过程

域名解析可以是正向解析（域名->IP 地址），也可以是反向解析（IP 地址->域名）。

域名解析有两种方式：递归查询和递归与迭代相结合的查询。

1. 主机向本地域名服务器的查询采用递归查询。
2. 本地域名服务器向根域名服务器的查询采用迭代查询。

例题 3. 假设某主机访问访问连接 http://www.abc.com/index.html，局域网内的本地域名服务器为递归查询，其他所有域名服务器为迭代查询，局域网访问外网的时延不可忽略，其他各种时延均可忽略，则从点击超链接到浏览器接收到 index.html 为止，可能的最短 RTT 和最长 RTT 分别是（ ）和（ ）。

解答： 2, 5

- 最理想情况：主机有 www.abc.com 到其 IP 地址的映射缓存，无需 DNS 查询，直接请求文件，则 TCP 连接建立需要一个 RTT，请求文件需要一个 RTT。
- 最坏情况：主机向本地域名服务器查询 www.abc.com 的 IP 地址（延时忽略），本地域名服务器迭代查询，访问根域名服务器、.com 顶级域名服务器、abc.com 域名服务器，分别花费 1 个 RTT，共 3 个 RTT，再加上上述的 2 个 RTT，为 5RTT。

6.3 文件传输协议（FTP）

FTP 允许客户指明文件的类型与格式，并允许文件具有存取权限。
FTP 服务器分为两部分：

- 一个主进程，负责接收新的请求。
- 若干个从属进程，负责处理单个请求。

FTP 的连接也分两种：

- 控制连接，监听 21 端口
- 数据连接，监听 20 端口

文件列表是通过数据连接传送的。

6.4 电子邮件

6.4.1 电子邮件系统的组成结构

一个电子邮件系统应具有三个主要组成构件：

- **用户代理（UA）**。用户和电子邮件系统的接口。通常情况下就是电子邮件客户端软件。
- **邮件服务器**。负责发送和接收文件，采用 C/S 模式工作，但一个邮件服务器必须能够同时充当客户端和服务器。
- **邮件发送和读取协议**。如 SMTP 和 POP3（或 IMAP）。SMTP 是“推”的方式，POP3 是“拉”的方式。

POP3 在传输层使用明文传送密码。

6.5 万维网（WWW）

6.5.1 超文本传输协议 HTTP

HTTP 包含两类报文：

- 请求报文

- 响应报文

HTTP 本身是无连接的，即虽然 HTTP 基于 TCP，但通信双方在通信之前无需建立 HTTP 连接。

HTTP 是无状态的。服务器不记得是否为某一特定主机服务过，也不记得服务过多少次。(但是可以通过 Cookie+ 数据库的方式跟踪用户的活动)

HTTP 可以使用非持久连接，也可以使用持久连接（HTTP/1.1 支持）

- 非持久连接下，每个网页元素对象都需要建立并释放一次 TCP 连接，请求一个 WWW 文档需要的时间 = 该文档的传输时间 + 两倍 RTT（TCP 建立一倍，请求和接收文档一倍）
- 持久连接下，一个 TCP 连接上可以多次传送 HTTP 报文。又分流水线和非流水线两种方式：
  - ◇ 非流水线方式：客户在收到前一个响应后才能发出下一个请求，浪费资源。
  - ◇ 流水线方式：可以连续发请求。

HTTP/1.1 默认使用流水线的持久连接。

6.6 应用层协议总结

应用程序	FTP 数据连接	FTP 控制连接	TELNET	SMTP	DNS	TFTP	HTTP	POP3	SNMP
使用协议	TCP	TCP	TCP	TCP	UDP	UDP	TCP	TCP	UDP
熟知端口号	20	21	23	25	53	69	80	110	161

表 2: 常见应用层协议小结