计算机网络

目录

[概述 7](#_Toc115127517)

[计算机功能 7](#_Toc115127518)

[标准化 7](#_Toc115127519)

[标准的分类 7](#_Toc115127520)

[工作阶段 7](#_Toc115127521)

[相关组织 8](#_Toc115127522)

[计算机组成 8](#_Toc115127523)

[计算机分类 8](#_Toc115127524)

[电路交换 8](#_Toc115127525)

[报文交换 9](#_Toc115127526)

[分组交换 9](#_Toc115127527)

[分组交换传输方式 10](#_Toc115127528)

[数据报方式 10](#_Toc115127529)

[虚电路方式 11](#_Toc115127530)

[性能指标 12](#_Toc115127531)

[速率 12](#_Toc115127532)

[带宽 12](#_Toc115127533)

[吞吐量 12](#_Toc115127534)

[时延 12](#_Toc115127535)

[时延带宽积 13](#_Toc115127536)

[往返时延RTT 13](#_Toc115127537)

[利用率 13](#_Toc115127538)

[体系结构 13](#_Toc115127539)

[分层原因 13](#_Toc115127540)

[分层结构概念 13](#_Toc115127541)

[OSI参考模型 14](#_Toc115127542)

[TCP/IP参考模型 15](#_Toc115127543)

[五层参考模型 15](#_Toc115127544)

[物理层 16](#_Toc115127545)

[基本概念 16](#_Toc115127546)

[通信简述 16](#_Toc115127547)

[传输单位 17](#_Toc115127548)

[传输准则 18](#_Toc115127549)

[编码 19](#_Toc115127550)

[调制 21](#_Toc115127551)

[传输媒介 22](#_Toc115127552)

[导向型传输介质 22](#_Toc115127553)

[非导向性传输介质 23](#_Toc115127554)

[物理层接口特性 23](#_Toc115127555)

[物理层设备 24](#_Toc115127556)

[中继器（RP、repeater） 24](#_Toc115127557)

[集线器（Hub） 24](#_Toc115127558)

[数据链路层 25](#_Toc115127559)

[基本概念 25](#_Toc115127560)

[数据链路层功能 25](#_Toc115127561)

[组帧 25](#_Toc115127562)

[字符计数法 25](#_Toc115127563)

[字符填充法 26](#_Toc115127564)

[零比特填充法 26](#_Toc115127565)

[违规编码法 26](#_Toc115127566)

[差错控制 27](#_Toc115127567)

[奇偶校验码 27](#_Toc115127568)

[CRC循环冗余码 28](#_Toc115127569)

[海明码 29](#_Toc115127570)

[流量控制 31](#_Toc115127571)

[停止等待协议（Stop-and-Wait） 32](#_Toc115127572)

[后退N帧协议（GBN） 37](#_Toc115127573)

[选择重传协议（SR） 40](#_Toc115127574)

[介质访问控制 42](#_Toc115127575)

[信道划分介质访问控制 43](#_Toc115127576)

[ALOHA协议 45](#_Toc115127577)

[CSMA协议 46](#_Toc115127578)

[CSMA/CD协议 47](#_Toc115127579)

[CSMA/CA协议 48](#_Toc115127580)

[轮询访问介质访问控制 49](#_Toc115127581)

[局域网 51](#_Toc115127582)

[局域网特性 51](#_Toc115127583)

[以太网 52](#_Toc115127584)

[无线局域网 55](#_Toc115127585)

[虚拟局域网 56](#_Toc115127586)

[广域网 57](#_Toc115127587)

[PPP协议 57](#_Toc115127588)

[HDLC协议 59](#_Toc115127589)

[数据链路层设备 61](#_Toc115127590)

[网桥（Bridge） 61](#_Toc115127591)

[交换机（Switch） 62](#_Toc115127592)

[交换机与网桥 63](#_Toc115127593)

[冲突域与广播域 63](#_Toc115127594)

[网络层 64](#_Toc115127595)

[基本概念 64](#_Toc115127596)

[网络层功能 64](#_Toc115127597)

[IP协议 65](#_Toc115127598)

[IP数据报 66](#_Toc115127599)

[IP地址 69](#_Toc115127600)

[专用网络通信 71](#_Toc115127601)

[子网与超网 72](#_Toc115127602)

[IPv6 75](#_Toc115127603)

[ARP协议 77](#_Toc115127604)

[ARP协议使用流程 77](#_Toc115127605)

[ARP协议的典型情况 78](#_Toc115127606)

[ICMP协议 78](#_Toc115127607)

[ICMP报文类型 78](#_Toc115127608)

[ICMP应用 80](#_Toc115127609)

[路由算法与协议 80](#_Toc115127610)

[路由算法与协议概述 80](#_Toc115127611)

[RIP协议 80](#_Toc115127612)

[OSPF协议 82](#_Toc115127613)

[BGP协议 84](#_Toc115127614)

[三种协议的比较 86](#_Toc115127615)

[IP多播 86](#_Toc115127616)

[组播实现 87](#_Toc115127617)

[IP组播地址 87](#_Toc115127618)

[硬件组播 87](#_Toc115127619)

[IGMP协议 88](#_Toc115127620)

[组播路由选择协议 88](#_Toc115127621)

[MPLS 89](#_Toc115127622)

[移动IP 89](#_Toc115127623)

[相关术语 89](#_Toc115127624)

[通信过程 89](#_Toc115127625)

[互联网连接问题 90](#_Toc115127626)

[网络层设备 91](#_Toc115127627)

[输入端口处理 91](#_Toc115127628)

[交换结构 91](#_Toc115127629)

[输出端口处理 91](#_Toc115127630)

[三层设备的对比 92](#_Toc115127631)

[路由表与路由转发 92](#_Toc115127632)

[传输层 92](#_Toc115127633)

[基本概念 92](#_Toc115127634)

[传输层的功能 92](#_Toc115127635)

[寻址与端口 92](#_Toc115127636)

[UDP协议 93](#_Toc115127637)

[主要特点 93](#_Toc115127638)

[UDP数据报格式 94](#_Toc115127639)

[UDP协议校验和 94](#_Toc115127640)

[TCP协议 96](#_Toc115127641)

[TCP协议主要特点 96](#_Toc115127642)

[TCP数据报格式 96](#_Toc115127643)

[TCP协议连接管理 98](#_Toc115127644)

[TCP协议可靠传输 101](#_Toc115127645)

[TCP协议流量控制 103](#_Toc115127646)

[TCP协议拥塞控制 104](#_Toc115127647)

[应用层 106](#_Toc115127648)

[网络应用模型 106](#_Toc115127649)

[C/S模型 106](#_Toc115127650)

[P2P模型 106](#_Toc115127651)

[DNS 107](#_Toc115127652)

[域名 107](#_Toc115127653)

[域名服务器 108](#_Toc115127654)

[域名解析过程 108](#_Toc115127655)

[文件传送协议 109](#_Toc115127656)

[FTP协议 110](#_Toc115127657)

[TFTP协议 111](#_Toc115127658)

[TELNET协议 112](#_Toc115127659)

[万维网 112](#_Toc115127660)

[WWW 112](#_Toc115127661)

[URL 112](#_Toc115127662)

[HTTP协议 112](#_Toc115127663)

[电子邮件 116](#_Toc115127664)

[电子邮件格式 116](#_Toc115127665)

[电子邮件系统组成结构 117](#_Toc115127666)

[SMTP协议 117](#_Toc115127667)

[MIME协议 119](#_Toc115127668)

[POP协议 119](#_Toc115127669)

[IMAP协议 120](#_Toc115127670)

[万维网邮件 120](#_Toc115127671)

[DHCP协议 120](#_Toc115127672)

# 概述

计算机网络是互连（互联互通）、自治（无主从关系）的计算机集合。

## 计算机功能

* 数据通信。
* 资源共享（硬件、软件、数据）。
* 分布式处理。
* 提高可靠性。
* 负载均衡。

## 标准化

### 标准的分类

* 法定标准：由权威机构制定的正式合法的标准，如。
* 事实标准：某些公司产品在竞争中占据主流，其产品的协议与技术就成为了标准，如。

### 工作阶段

（）因特网标准形式。

* 因特网草案（）。
* 建议标准（）。
* ~~草案标准（）。~~
* 因特网标准（）。

### 相关组织

* 国际标准化组织：、。
* 国际电信联盟：制定通信规则。
* 国际电气电子工程师协会：。
* 因特网工程任务组：因特网标准制定。

## 计算机组成

* 组成部分：硬件、软件、协议。
* 工作方式：边缘部分（用户直接使用，包括与方式）、核心部分（为边缘部分服务）。
* 功能组成：通信子网（数据通信）、资源子网（资源共享数据处理）。

## 计算机分类

* 分布范围：广域网（交换技术）、城域网、局域网（广播技术）、个人区域网。
* 使用者：公用网、专用网。
* 交换技术：电路交换、报文交换、分组交换。
* 拓扑结构：总线、星型、环型、网状。
* 传输技术：广播式、点对点式。

### 电路交换

1. 建立连接（电路建立）。
2. 通信。
3. 释放连接（拆除电路）。

链路支持多路复用（可参考数据链路层的等）。

优点：

1. 通信时延小，速度快。
2. 有序传输。
3. 无冲突争用。
4. 实时性强。

缺点：

1. 建立连接时间长。
2. 线路独占，使用效率低。
3. 灵活性较差。
4. 不能流量控制。
5. 无差错控制能力。

### 报文交换

报文：源应用发送的信息整体。

1. 源先根据报文加上IP地址，物理地址等，通过物理层发送。
2. 交换机收到报文之后，先进行存储，等待链路空闲之后进行转发，但是转发的路径不是固定的，是比较随机的。
3. 最后通过许多交换机最终到达目的地。

优点：

1. 无需建立连接。
2. 存储转发，动态分配线路。
3. 线路可靠性高。
4. 线路利用率较高。
5. 多目标服务。

缺点：

1. 有存储转发时延。
2. 报文大小不定，需要网络结点有较大的缓存空间。

所以报文交换不能应用于实时通讯，如电话，视频。

### 分组交换

将大的数据块分割为小的数据块。分组交换和报文交换基本相同，就是将报文切割之后再进行发送。

优点：

1. 无需建立连接。
2. 存储转发，动态分配线路。
3. 线路可靠性高。
4. 线路利用率较高。
5. 分组大小一定，存储管理更容易。
6. 分组更小，所以传输时延比报文交换的更小。

缺点：

1. 有存储转发时延。
2. 需要传输额外的信息量（编号、目的地址等）。
3. 乱序到达目的主机时，需要对分组进行排序。

![数据交换](data:image/png;base64;base64,)

数据交换

1. 报文交换和分组交换都采用存储转发。
2. 传输数据量大且传输时间远大于呼叫时间，就选择电路交换，其传输时延最小。
3. 报文交换分组交换在信道利用率上优于电路交换。

## 分组交换传输方式

数据报和分组的关系：分组是一段比较长的数据，将它进行切割成一段段之后就得到数据报。其都是网络层的传输单位。

分组交换分为数据报方式和虚电路方式两种。

对于出错率高的线路应该使用数据报方式。

### 数据报方式

* 提供无连接服务，可随时发送和接受。
* 尽最大可能交付，所以不保证可靠性。
* 不事先为分组的传输确定传输路径，每个分组独立确定传输路径，不同分组传输路径可能不同，也不一定按序到达。
* 每个分组携带源和目的地址。
* 存储转换需要排队等待。当出现拥塞时会丢弃部分分组。
* 可及时更新转发表，适应性强。
* 时延小，提高吞吐量。
* 不单独占用信道，资源利用率高。
* 路由器根据分组的目的地址转发组：基于路由协议/算法构建转发表（目的网络地址与链路接口的散列对）；检索转发表；每个分组独立选路。

### 虚电路方式

* 虚电路将数据报方式与电路交换方式结合。
* 虚电路是一条源主机到目的主机类似电路的逻辑连接，路径上所有结点都要维持该虚电路的建立，都维持一张虚电路表，每一项都记录一个打开的虚电路的信息。
* 虚电路方式提供连接服务，首先为分组的传输确定传输路径，然后沿该路径/连接传输系列分组，分组的传输路径相同，结束后拆除连接。
* 一条物理链路可能有多个逻辑电路。
* 虚拟电路包含临时和永久两种：
  + 永久性虚电路：是一种提前定义好的、基本上不需要任何建立时间的端点之间的连接。
  + 交换型虚电路：是端点之间的一种临时性连接，这些连接只持续所需的时间，并且在会话结束时就取消这种连接，B错误。

传输过程：

1. 建立虚电路连接：每个分组携带虚电路标识（），而非目的地址，源主机发送呼叫请求分组并收到呼叫应答分组后才算建立连接。
2. 数据传输：全双工通信。
3. 释放虚电路连接：源主机发送释放请求分组以拆除虚电路。

优点：

* 虚电路通信链路建立拆除需要时间，不适合交互式和小量短分组，适合长时间频繁数据交换。
* 链路建立确定传输路径。
* 可靠通信，保证有序到达。
* 提供流量控制。
* 虚电路某点或某部分破坏，则会影响其他相关电路。
* 分组首部不包含目的地址，包含虚电路标识符，开销小。

|  | 数据报服务 | 虚电路服务 |
| --- | --- | --- |
| 建立连接 | 不用 | 必须 |
| 目的地址 | 每个分组都要有完成的目的地址 | 仅在建立连接时使用，之后每个分组使用较短的虚电路号 |
| 路由选择 | 每个分组单独进行路由选择与转发 | 属于同一条虚电路的分组按照同一个路由转发 |
| 分组顺序 | 不保证有序到达 | 保证有序到达 |
| 可靠性 | 不保证，可靠性依靠用户主机保证 | 保证 |
| 网络故障适应性 | 强 | 弱 |
| 差错控制和流量控制 | 用户主机完成 | 可由网络完成也可以由用户主机完成 |

## 性能指标

### 速率

数据传输率或比特率。是主机在数字信道上传输数据位数的额定理想速度。

。

（ 存储容量方面，，因为一个字节占八个比特位，所以一般都是以字节为单位。 。 ）

### 带宽

网络中某点到另外一点所能通过的**最高数据率**，单位是b/s等。即网络设备发送的极限最高速率约束。

### 吞吐量

单位时间内通过某网络或信道接口的数据量，单位是等。这是网络实际的平均速率。吞吐量受网络带宽的限制。

### 时延

是数据从网络的一端传送到另一端锁需要的时间，单位为。时延=发送（传输）时延+传播时延+排队时延+处理时延。

* 发送时延：从发送分组的第一个比特开始到最后一个比特发送完毕的时间。发送时延=数据长度（发送速率）。
* 传播时延：数据在信道上传播的时间。传播时延=信道长度。
* 排队时延：等待输入与输出链路可用的时间。
* 处理时延：检错数据与查找转发出口的时间。

### 时延带宽积

时延带宽积=传播时延×带宽，单位为。表示是以比特为单位的链路长度，即链路当前的数据比特容量。

### 往返时延RTT

从发送方发送数据开始，到发送方收到接收方的确认（假如收到数据立刻返回确认）总共经历的时延。

越大，在收到确认前可以发送的数据越多。

=往返传播时延+末端处理时延。

### 利用率

* 信道利用率：有数据通过时间（有+无）数据通过时间。
* 网络利用率：信道利用率加权平均值。

利用率越高则时延越长，且加速度不断提高。

## 体系结构

### 分层原因

* 发起通信的计算机必须将数据通信的通路进行激活。
* 要告诉网络如何识别目的主机。
* 发起通信的计算机要查明目的主机是否开机，并且与网络连接正常。
* 发起通信的计算机要弄清楚，对方计算机中文件管理程序是否已经做好准备工作。
* 确保差错和意外可以解决。

### 分层结构概念

* 实体：第层中的活动元素称为层实体。同一层的实体就是对等实体。
* 协议：为进行网络中的对等实体数据交换而建立的规则、标准或约定。是水平方向的。
  + 语法：传输数据的格式。
  + 语义：所要完成的功能。
  + 同步：各种操作的顺序。
* 接口（访问服务点）：上层使用下层服务的入口。
  + 接口连接的是相邻层次。
* 服务：下层为相邻上层提供的功能调用。是垂直方向的。
  + 上层使用下层提供的服务。
  + 第层向层提供服务时，不仅包括第n层本身的功能，还包括下面所有层次所提供的所有功能。
  + 本层所提供服务细节对上一层完全屏蔽。
  + 服务类型：
    - 面向连接和无连接。
    - 可靠和不可靠。
    - 有应答和无应答。
* 服务数据单元：为完成用户所要求的功能而应传输的数据。即要传输的目的数据。
* 协议控制信息：控制协议操作的信息。即协议控制头。
* 协议数据单元：对等层次之间传送的数据单位。
  + 层的作为第层的，在加上本层的就变成了本层的，然后交给层。所以。
  + 物理层的为比特，数据链路层的为帧，网络层的为分组，传输层的为报文段。

### OSI参考模型

为一个法定标准，但是没有实现。为于年提出的开放系统互连模型。

| 名称 | 英文名 | 作用 |
| --- | --- | --- |
| 应用层 | Application,Layer | 用户与网络的界面；直接为用户的应用进程（例如电子邮件、文件传输和终端仿真）提供服务；HTTP、SMTP、FTP、DNS |
| 表示层 | Presentation,Layer | 规范交换信息的表示方式；数据格式转换；加密和解密；数据压缩和恢复；JPEG、ASCII |
| 会话层 | Session,Layer | 负责在数据传输中建立和维护计算机网络中两台计算机之间的通信连接（会话）；使用校验点同步（SYN）或恢复通信；ADSP、ASP |
| 传输层 | Transport,Layer | 负责端到端通讯；传输单位为报文段（TCP）或用户数据报（UDP）；可靠传输，不可靠传输 ；差错控制；流量控制；复用分用；TCP、UDP |
| 网络层 | Network,Layer | 传输分组，传输单位是数据报或分组；路由选择；流量控制；差错控制；规划IP地址(ipv4和ipv6变化只会影响网络层)；拥塞控制；网际互联；IP、ICP、IGMP、ARP、RARP、OSPF |
| 数据链路层 | Data,Link,Layer | 负责点到点通讯；将数据报组装成帧，传输单位为帧；成帧（帧的开始和结束）；透明传输；差错校验(一般丢弃，纠错由传输层解决)；流量控制；访问（接入）控制（控制信道的控制）；SDLC、HDLC、PPP、STP |
| 物理层 | Physical,Layer | 简单将数据转变为物理电信号，在物理媒体上实现比特流的透明传输，传输单位为比特；定义接口特性；定义传输模式（单工、半双工、双工）；定义传输速率；比特同步；比特编码；Rj45、802.3 |

端系统一般都是七层，而中间系统如路由器等最多只有三层。上面四层是端到端的通信，而下面三层是点到点的通信。

### TCP/IP参考模型

先有协议栈才有参考模型。

#### OSI与TCP/IP的联系

* 应用层-应用层+表示层+会话层：、、。
* 传输层-传输层：、。
* 网际层-网络层：。
* 网络接口层-数据链路层+物理层：、、。

#### 面向连接与无连接

面向连接：

* 发出请求，建立连接。
* 传输数据。
* 释放连接。

无连接直接传输数据。

两个模型之间的连接区别：

|  | 参考模型 | 模型 |
| --- | --- | --- |
| 传输层 | 面向连接 | 无连接+面向连接 |
| 网络层 | 无连接+面向连接 | 无连接 |

### 五层参考模型

应用层、传输层、网络层、数据链路层、物理层。

# 物理层

## 基本概念

物理层需要考虑如何传输比特流，而不关心到底使用何种媒体传输数据。具体传输数据的层次是第层。

### 通信简述

#### 通信流程

信源->数字信号->调制解调器->模拟型号->广域网->模拟信号->调制解调器->数字信号->信宿。

* 数字信号：取值离散。
* 模拟限号：取值连续。
* 信源：发送数据的源头。
* 信宿：接受数据的终点。
* 信道：信号的传输媒介。
  + 模拟信道、数字信道。
  + 有线信道、无线信道。

#### 通信方式

* 单工通信：只能单方面通信，只需一条信道。
* 半双工通信：能发送和接受，但是不能同时发送接收，需要两条信道。
* 全双工通信：都能发送与接受，需要两条信道。

#### 传输方式

* 串行：速度慢、费用低、适合远距离。
* 并行：速度快、费用高、适合近距离。

#### 信号类型

* 基带传输：
  + 将数字信号直接用两种不同的电压表示，不用调制，直接送到数字信道上传输。
  + 是来自信源的信号，如计算机输出的表示文字或多媒体的数据信号，即发出的直接表达了要传输的信息的信号。
  + 传输媒体整个带宽都被基带信号占用。
  + 近距离，通常用于局域网。
  + 使用高低高低电平表示。
* 频带传输：
  + 将基带信号对特定频率的载波进行调制后进行的频分复用模拟信号，再传输到模拟信道上传输。
  + 将低频的基带信号进行载波调制后将信号的频率范围移动到较高频段以便在信道中传输。
  + 远距离，与无线传输。
  + 不仅解决电话系统传输数字信号问题，还可以实现多路复用。
  + 传输一个码元就相当于。
* 宽带传输：
  + 基于频带传输，使用频分复用等技术将链路容量分解为两个或多个信道，每个信道可以携带不同的信号。
  + 信道直接互不干扰，提高效率。

### 传输单位

#### 码元

定义：码元是指用一个固定时长的信号波形（数字脉冲），代表离散数值的基本波形。这个时长内的信号就是进制码元，时长就是码元宽度，当有个离散状态时，成为进制码元。

一个码元可以携带多个比特的信息。

#### 波特（Baud）

指一秒可以传输多少个码元（电平）。一波特就是一码元一秒。

比特率=波特率码元表示有效离散值个数。

#### 速率

分为信息传输速率与码元传输速率。

信息传输速率就是，就是我们平常说的网速。

码元传输速率就是指一秒可以传输多少个码元，即单位时间内脉冲个数或信号变化的次数。码元可以理解为几个比特的集合，所以信息传输速率（网速）=码元传输速率x码元所带信息量（多少比特）。单位为波特。波特传输率与进制无关。

码元所带信息量（比特数）=码元进制数（码元种数）。

#### 带宽（Band Width）

用来表示最高数据速率。

信道带宽指信道最高通过频率与最低通过频率之差，单位为赫兹。

### 传输准则

#### 奈式准则（奈奎斯特定理、Nyquist）

奈氏准则规定避免码间串扰的低通信道最高码元传输速率。

在理想低通条件下，避免码间串扰：

* 极限码元传输速率=信道带宽，单位为波特。
* 极限数据传输速率=信道带宽×码元种数，单位为。

1. 由于码元的传输速率受奈氏准则的制约，所以要提高数据的传输速率，就必须设法使每个码元能携带更多的个比特量的信息，这就需要多元制的调制方法。
2. 信道频带越宽，即能通过的信号高频分量越多，就可以用更高速率进行码元的有效传输。

#### 香农定理（Shannon）

香农定理规定带宽首先且高斯噪声条件下最高数据传输速率。

信噪比=信号的平均功率÷噪声的平均功率，单位是分贝（），或为信号的平均功率÷噪声的平均功率，这是一个单纯的比值。

采用分贝表示的原因是很多时候，信号要比噪声强得多，比如信号比噪声强亿倍，如果用数值表示的话，那么后面有个，很容易丢失一个。如果用分贝表示只为，要简单得多，而且不容易出错。分贝对于表示特别大或特别小的数值极为有利，这种表示方式在电子通信领域用途很广。

在带宽受限且有噪声的信道中，为了不产生误差，信息的数据传输速率有上限值。

极限数据传输速率=信道带宽×(信噪比=信道带宽×(信号的平均功率÷噪声的平均功率，单位为。

只要信息的传输速率低于信道的极限传输速率，就一定能找到某种方式来实现无差错传输。

#### 奈氏准则与香农定理区别

奈氏准则指出，码元传输的速率是受限的，不能任意提高，否则接收端就不能正确判定码元所携带的比特是还是（因为存在码元之间的相互干扰）。

奈氏准则是在理想条件下推导出来的。在实际条件下，最高码元传输速率要比理想条件下得出的数值小很多。电信技术人员的任务就是要在实际条件下，寻找出较好的传输码元波形，将比特转换为较为合适的传输信号。

需要注意的是，奈氏准则并未限制信息传输速率（）。

要提高信息传输速率，就必须使每个传输的码元能够代表许多比特的信息，这就需要有很好的编码技术。但码元所载的比特数确定后，信道的极限数据率也就确定了。

香农定理给出了信息传输速率的极限，即对于一定的传输带宽（单位为）和一定的信噪比，信息传输速率的上限就确定了，这个极限是不能突破的。要想提高信息传输速率，要么设法提高传输线路的带宽，要么设法提高所传信道的信噪比，此外没有其他任何办法。

香农定理告诉我们，若要得到无限大的信息传输速率，只有两个办法:要么使用无限大的传输带宽（这显然不可能)，要么使信号的信噪比无限大，即采用没有噪声的传输信道或使用无限大的发送功率（显然这也不可能）。

奈氏准则和香农定理中“带宽”的单位都是。

注意：当没有给信噪比的条件就只能使用奈氏准则，而如果当给出了信噪比与码元种数时，就需要同时使用奈氏准则与香农定理，然后取较小值作为答案。

### 编码

即将数据转换为数字信号。

#### 数字数据到数字信号

数字数据通过数字发送器转换为数字信号。

编码方式：

1. 非归零编码（）。
2. 曼彻斯特编码。
3. 差分曼彻斯特编码。
4. 归零编码（）。
5. 反向不归零编码（）。
6. 4B/5B编码。

![编码示意图](data:image/png;base64;base64,)

编码示意图

* 非归零编码：高低。非常容易实现，但是没有检错功能，却无法判断一个码元的开始与结束从而很难保持收发双方的同步。
* 曼彻斯特编码：同异，即下一个频率电平变化则代表下一位为，不变化则为。每一个码元中间都会产生电平调变，所以中间的跳变既可以作为时钟信号用于同步，也可以用于数据信号，但是其占有的频带宽度是原基带宽度的两倍。因为每一个码元被调制为两个电平，所以数据传输速率只有调制速率的一半。
* 差分曼彻斯特编码：同异，若码元为，则前半段与上一个码元的后半个码元电平相同，若为则相反。每个码元中间同样有电平跳变可用于同步，抗干扰性也强于曼彻斯特编码。

![归零编码](data:image/png;base64;base64,)

归零编码

归零编码：折平，信号电平在一个码元内都要恢复到。信道使用率低，基本上都是无使用。

![反向不归零编码](data:image/png;base64;base64,)

反向不归零编码

反向不归零编码：折平，信号电平翻转表示，不变表示。对于全的数据同步效果不好。

| 十进制数 | 4位二进制数 | 4B/5B编码 | 十进制数 | 4位二进制数 | 4B/5B编码 |
| --- | --- | --- | --- | --- | --- |
| 0 | 0000 | 11110 | 8 | 1000 | 10010 |
| 1 | 0001 | 01001 | 9 | 1001 | 10011 |
| 2 | 0010 | 10100 | 10 | 1010 | 10110 |
| 3 | 0011 | 10101 | 11 | 1011 | 10111 |
| 4 | 0100 | 01010 | 12 | 1100 | 11010 |
| 5 | 0101 | 01011 | 13 | 1101 | 11011 |
| 6 | 0110 | 01110 | 14 | 1110 | 11100 |
| 7 | 0111 | 01111 | 15 | 1111 | 11101 |

编码：位码转换为位码，比特流中插入额外的比特来打断一连串的或，编码效率为。只采用种对应种不同的位码，其他种作为控制码，控制帧开始与结束以及线路状态等或保留。

#### 模拟数据到数字信号

因为计算机内部都是二进制数据，如处理的是数字音频，所以需要将模拟声音通过采样、量化转换位有限个数字表示的离散序列，这就是音频数字化。

1. 抽样：对模拟信号周期性扫描，把时间上连续的信号变为时间上的离散信号。采样频率遵循采样定理。
2. 量化：把抽样取得的电平幅值按照一定的分级标度转换位对应的数字值并取整，于是连续的电平幅值就变为离散的数字量。
3. 编码：把量化结果转换为与之对应的二进制编码。

音频信号模拟数据通过（脉冲编码调制）编码器转换为数字信号。

采样定理：采样频率信号最高频率。

### 调制

即将数据转换为模拟信号。

同时在接收端解调模拟信号转为数字信号。

#### 数字数据到模拟信号

数字数据通过调制器转换为模拟信号。

![调制方法](data:image/png;base64;base64,)

调制方法

1. 幅移键控（调幅）。通过改变载波信号的振幅来表示数字信号和，而载波的频率和相位都不改变。比较容易实现，但抗干扰能力差。
2. 频移键控（调频）。通过改变载波信号的频率来表示数字信号和，而载波的振幅和相位都不改变。容易实现，抗干扰能力强，目前应用较为广泛。
3. 相移键控（调相）。通过改变载波信号的相位来表示数字信号和，而载波的振幅和频率都不改变。它又分为绝对调相和相对调相。
4. 正交振幅调制（）。在频率相同的前提下，将与结合起来，形成叠加信号。设波特率为，采用个相位，每个相位有种振幅，码元类型=相位数×振幅数。则该技术的数据传输速率为，即为波特率×(相位数×每相位振幅数)。

#### 模拟数据到模拟信号

模拟数据通过放大调制器转换为模拟信号。

将低频信号调制为高频信号。

## 传输媒介

是数据传输系统种在发射设备与接受设备之间的物理通路（信道是逻辑通路）。

传输媒体并不是物理层。

* 导向性传输介质：电磁波沿着固体媒介（铜线或光纤）被导向传播。
* 非导向性传输介质：自由空间，如空气，水等等。

### 导向型传输介质

#### 双绞线

* 是两根采用规则并排校合并绝缘的铜导线组成。绞合是减少电磁干扰。
* 根据有无屏蔽层分为屏蔽双绞线（）和无屏蔽双绞线（）。外加上金属网就是。
* 通信距离为几到数十公里。
* 距离太远需要使用放大器放大模拟信号，中继器整形数字信号。

#### 同轴电缆

同轴电缆由内导体、绝缘层、网状编织屏蔽层和塑料外层构成。

* ：基带同轴电缆。用于局域网。
* ：宽带同轴电缆。用于有线电视系统。

由于外导体屏蔽层的作用，同轴电缆较双绞线具有良好的抗干扰特性，被广泛用于传输较高速率的数据，其传输距离更远，但价格较双绞线贵。

#### 光纤

* 传递光脉冲，带宽大于其他传输媒体。
* 实心纤芯与包层构成。
* 分为多模光纤与单模光纤。

| 定义 | 光源 | 特点 | 外观 |
| --- | --- | --- | --- |
| 单模光纤 | 一种在横向模式直接传输光信号的管线 | 定向性好的激光二极管 | 损耗小，适合远距离传输 |
| 多模光纤 | 有多种传输光信号模式的光纤 | 普通发光二极管 | 易失真，适合近距离传输 |

光纤特点：

1. 传输损耗小，中继距离长。
2. 抗雷电电磁干扰性能好。
3. 保密性好，无串音干扰。
4. 体积小重量低。

### 非导向性传输介质

* 无线电波：
  + 全方向传播；
  + 穿透率较强；
  + 可远距离通信，主要用于电话通信。
* 微波：
  + 固定方向传播；
  + 频率高、频段广、数据率高；
  + 分为地面微波接力通信与卫星通信；
    - 卫星通信的优点：通信容量大、距离远、覆盖光、广播通信与多址通信；
    - 卫星通信的缺点：传播时延长、受天气干扰大、误码率高、成本高。
* 红外线与激光：
  + 固定方向传播；
  + 需要转换为各自的信号格式再在空间中传播。

### 物理层接口特性

* 机械特性：规格、接口形状、引线数目、引脚数量、排列情况。
* 电气特性：电压范围、阻抗匹配、传输速率、距离限制。
* 功能特性：电平意义、接口部件的用途。
* 规程特性（过程特性）：工作规程和时序的关系。

## 物理层设备

网段在物理层互联时要求数据传输速率相同，但是上层协议可以不同。

### 中继器（RP、repeater）

也称为转发器，仅作用于信号电气部分，用于对数字信号进行再生和还原，对衰减的信号进行整形、放大（不仅仅是放大）。

中继器的两端：

* 网段而非子网。使用中继器连接的几个网段仍然是一个局域网。
* 类型相同。
* 网络速度相同。
* 只负责发送而不关心是否有问题。
* 可以连不同媒体也可以连相同媒体。
* 不会存储转发所以一定要协议相同。
* 规则：采用粗同轴电缆的以太网规范，是指不能超过五个网段；是指在这些网段中的物理层网络设备（中继器，集线器）最多不超过四个；是指这些网段中最多只有三个网段挂有计算机。

注意：若网络设备具有存储转发的功能则能连接不同的协议，否则则不能。

注意：放大器和中继器都起放大作用，只不过放大器放大的是模拟信号，原理是将衰减的信号放大，而中继器放大的是数字信号，原理是将衰减的信号整形再生。

### 集线器（Hub）

集线器是多口中继器，功能类似，放大信号再转发到其他所有除输入端口外处于工作态的端口上。

* 不具备信号的定向传送能力，是一个共享式设备（广播）。
* 因为是广播所以有大的冲突域，不能分割冲突域。所以多少个计算机连入集线器，同时工作时就平分集线器拥有的带宽。
* 同时只能有两个设备进行通讯，只会传输信号。
* 只能在半双工状态下工作。
* 集线器连接的网络拓扑结构上属于星型。
* 当集线器的一个端口收到数据后立刻从除输入端口外的所有端口广播出去。

以太网上的集线器默认是，即传输速率为。

# 数据链路层

## 基本概念

* 结点：主机、路由器。
* 链路：网络中两个结点之间的物理通道。
* 数据链路：网络中两个结点之间的逻辑通道。
* 帧：链路层的协议数据单元。

### 数据链路层功能

1. 为网络层提供服务：无确认无连接服务、有确认无连接服务、有确认面向连接服务。面向连接就一定会确认。
2. 链路管理：控制对物理传输介值访问。
3. 组帧（定义数据格式）：帧定界、帧同步、透明传输。规定了帧的数据部分的长度上限——最大传送单元（）。
4. 流量控制：发送方的流量。
5. 差错控制。

## 组帧

在一段数据的前后部分添加首部和尾部，构成一个帧来定界传输（由于数据链路层以上的传输单元不用直接传输所以只用添加本层首部就可以了）。接受端在收到物理层上交的比特流后就可以根据首部和尾部的标记从而识别帧的开始和结束。

不管所传数据为什么样的比特组合都能在链路上传输。因此链路层就看不见有什么方案数据传输的东西（控制信息）就是透明传输。

那么如何组帧呢？

### 字符计数法

帧首部使用一个计数字段（第一个字节，八位）表明帧内字符数。题目计算时这个计数字段也要算成一个字符。

![字符计数](data:image/png;base64;base64,)

字符计数

缺点：如果在某一个帧内，标记位后面的某个字节的数据丢失，那么会影响后面所有的帧。

比如和，如果前面的帧丢失变成，那么后面的4就会被补到前面变成导致错误。

### 字符填充法

帧就是加头加尾分别标记开始结束，而如果数据内某段比特流数据正好与标记字段重复，就会导致误判断的情况。

字符填充法就是在误会的字符前添加转义字符。

表示帧开始，表示帧结束。

![字符填充](data:image/png;base64;base64,)

字符填充

### 零比特填充法

当帧开始符和结束符一样，且帧定界符中含有个以上连续的，就可以使用零填充法。

* 在发送端扫描整个信息字段（不包括首尾定界符），只要有连续的五个，就立即在第五个后加上一个，无论后面是还是。
* 在接收端收到一个帧时，先找到标志字段确定边界，再用硬件对比特流进行扫描，发现连续五个时就把后面的删除。

### 违规编码法

使用帧中不会用到的编码来定帧的起始和终止。

![违规编码法](data:image/png;base64;base64,)

违规编码法

由于字节计数法的脆弱性与字符填充实现的复杂性与不兼容性，所以基本上使用零比特填充与违规编码法。

## 差错控制

噪声来源：

* 全局噪声（热噪声，随机噪声）：产生随机差错。线路本身电气特征所产生的固有；可以通过改善传感器调高信噪比来减少。
* 局部噪声（冲击噪声）：产生突发差错。外界特定短暂的原因所产生，是产生差错的主要原因；可以通过编码计数来解决。

差错：

* 位错：比特位出错，变为，变为。也称为比特差错。
* 帧错：有三种情况，假如发送的是：
  1. 丢失：：计时器超时重发。
  2. 重复：：帧编号机制。
  3. 失序：：帧编号机制。

为了避免位错进行校验：

* 检错编码：
  + 奇偶校验法。
  + 循环冗余码（）。
* 纠错编码：海明码。

物理层所说的编码是针对单个比特的调制，为了达成同步。

而数据链路层所说的编码是针对一组比特，通过冗余码的计数实现对一串二进制比特率的检查。

### 奇偶校验码

首先一串数据长度为，需要在数据第一位加上一位校验码。

奇校验要选校验码或使得位的总数据里的为奇数，而偶校验要选校验码或使得位的总数据里的为偶数。

奇偶校验码对于出错的数据仍有对应的奇偶个则无法检验出是否出错。即出错奇数个位数可以检查出来，偶数个位数不能检查出来，所以其检错能力为。

如字符为原数据有位，若使用奇校验，则需要首位校验位为让整个数据有奇数个，从而整个数据变为。若接受方收到、这种数据则能校验，因为数据有偶数个，则无法校验，因为有奇数个。

### CRC循环冗余码

分为三个部分：传输数据、生成多项式、帧检验序列/冗余码。

如简单来说在接受端：

| 传输数据 |  | 生成多项式 |  | 冗余码 |
| --- | --- | --- | --- | --- |
| 5 | ÷ | 2 | =2 | 1 |

所以最终发送的数据就是传输数据＋冗余码：。

在接收端：

| 接受数据 |  | 生成多项式 |  | 冗余码 |
| --- | --- | --- | --- | --- |
| 6 | ÷ | 2 | =3 | 0 |

余数为，判断无错，就接受。

其中正式的计算方法如下，并给出例题：发送数据为，采用校验，生成多项式为，其最终发送数据是？

#### 原数据加0

假设生成多项式的阶为，则需要在原数据后面加个（多项式有位，那么阶就是位）。

如题目中生成多项式表示为多项式是，所以阶就是。

最后数据就是。

#### 模二除法

在数据加后使用模二除法除以多项式，余数就是冗余码校验码的比特序列。

模二除法就是异或，同，异，无需进位或借位或者不够减的问题。

![模二除法](data:image/png;base64;base64,)

模二除法

所以就得到了帧检验序列为。

所以要发送的数据就是要发送的数据加上帧检验序列：。

#### 检错过程

把接收到的每一个帧都除以相同的生成多项式，然后检查得到除数，如果，则接受，否则丢弃。

的生成与接收端检验都是由硬件完成，所以处理很快，因此不会延误数据的传输。

具有纠错能力，但是数据链路层只使用了其检错能力，错误就直接丢弃。

### 海明码

能发现双比特错，但是只能纠正单比特错。

#### 确定校验码位数r

海明不等式：，其中为冗余信息位，位信息位。（添加的信息位能表达所有数据状态）

假如传输的数据为位，即信息位为，就可以一个个代入得到最小的为，所以海明码的校验码为位，传输数据为位。

#### 确定校验码和数据的位置

假设4位校验码分别为、、、，数据从左到右为、。

校验码需要插入到原数据之中作为传输数据，且只能插入到的位置，即。

然后在其他位置按需填入：

| 数据位 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 代码 |  |  |  |  |  |  |  |  |  |  |
| 实际值 |  |  | 1 |  | 0 | 1 | 1 |  | 0 | 1 |

#### 求出校验码的值

需要将数据位置的十进制先用二进制表示。

然后根据二进制中校验码位置的来判断其可以校验哪些位，因为第一步校验位的插入规定，的位置使用二进制表示一定只有一个，那么就要根据这个所在的位置找到所有同样这个位置是的位置所在的数据位，就是可以被校验码校验的原数据位。同一位可以被不同的校验码同时校验。

是第一位，即位，所以在二进制位置的第四个地方，找到个位置中所有二进制位置的第四个位为的数据位，即（）：；（）：；（）：；（）：。所以可以校验这个数据。

要使其能校验，就要令所有要校验的位异或等于，或直接等于所有校验数据位异或值。

即令。从而，根据同异得到。

同理的位置用二进制表示是，所以它可以校验二进制位置第三位是的所有数据位，即可以校验、、、。

即令。从而，根据同异得到。

同理的位置用二进制表示是，所以它可以校验二进制位置第二位是的所有数据位，即可以校验、、。

即令。从而，根据同异得到。

同理的位置用二进制表示是，所以它可以校验二进制位置第一位是的所有数据位，即可以校验、。

即令。从而，根据同异得到。

| 数据位 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 二进制 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 |
| 代码 |  |  |  |  |  |  |  |  |  |  |
| 实际值 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

从而最后海明码就是。

如果海明码需要一个最高的全校验位，则需要将其对全部数据位和其他全部校验位进行异或为得到。

#### 检错并纠错

海明码“纠错”位，需要码距为的编码方案；“检错”位，则只需码距为。

已知海明码就是。假如第五位出错，从而变为。

令所有要校验的位进行异或运算：

。

。

。

。

从进行的运算结果开始，将所有计算结果组成为，这个数字即表示出错的位置，即第五位出错。

## 流量控制

较高的发送速度和较低的接受能力不匹配时传输会出错。主要是控制发送方的流量。

数据链路层流量控制与传输层的流量控制的区别：

* 数据链路层流量控制是点到点的，而传输层的流量控制是端到端的。
* 数据链路层流量控制是相邻两个结点的，而传输层的流量控制是两个主机的。
* 数据链路层流量控制是收不下就不回复确认（确认控制帧），而传输层的流量控制是接收端发送给发送端一个窗口公告。

流量控制由自动重传请求来实现。

流量控制协议分为停止等待协议与滑动窗口协议，滑动窗口协议又分为两种后退帧协议与选择重传协议。其实停止等待协议也是一种特殊的滑动窗口协议。

后两种协议是滑动窗口技术和请求重复技术的结合，所以称为连续协议。

滑动窗口可以解决流量控制和可靠传输两个部分的功能。

数据链路层协议的滑动窗口在同一次发送与接受中都是固定的。

| 窗口大小 | 停止等待协议 | 后退N帧协议 | 选择重传协议 |
| --- | --- | --- | --- |
| 发送 | =1 | >1 | >1 |
| 接受 | =1 | =1 | >1 |

### 停止等待协议（Stop-and-Wait）

每发送完一个帧就停止发送并等待对方确认，如果收到确认就再发送下一个帧。因为是停止等待，所以只用一位以、为值来对帧编号就可以了。虽然同样是标号为的帧，但是是不同的帧。

#### 无差错情况

表示确认。

![无差错停止等待协议](data:image/png;base64;base64,)

无差错停止等待协议

#### 有差错情况

帧丢失或帧出错：

![有差错停止等待协议帧丢失出错情况](data:image/png;base64;base64,)

有差错停止等待协议帧丢失出错情况

1. 发送完一个帧后，必须保存其副本，传输完毕才能丢失该副本。
2. 数据帧与确认帧必须编号。
3. 超时计时器：每次发送一个帧就启动一个计时器。
4. 超时计时器设置的重传事件应该比帧传输的平均RTT更长。

ACK确认丢失：

![有差错停止等待协议帧确认丢失情况](data:image/png;base64;base64,)

有差错停止等待协议帧确认丢失情况

ACK确认迟到：

![有差错停止等待协议帧确认迟到情况](data:image/png;base64;base64,)

有差错停止等待协议帧确认迟到情况

信道利用率=数据帧长度÷(数据帧长度+往返时延+确认帧长度)。

信道利用率=(单位时间内发送数据的比特数÷发送发发送数据传输率)÷发送周期。

当确认帧忽略时可以用下面的公式。

信道吞吐率=信道利用率×发送方的发送速率。

#### 停止等待协议特点

* 简单。
* 信道利用率低。

### 后退N帧协议（GBN）

* 发送窗口：发送方维持一组连续的允许发送的帧的序号。
* 接受窗口：接受方维持一组连续的允许接受的帧的序号。

会一次性将在发送窗口内的个帧一个个全部发送完，然后再移动受确认的帧的个数个窗口，如果一直收不到确认信息则一直不移动发送窗口并不断依次重传。

使用累计确认模式，如果接发送方收到当前号帧的确认消息，则前面的所有帧都已经确认接收到。

#### 帧类型

假设发送窗口一次一共有个帧，且帧开始的索引值为，帧的类型分为：

1. 发完被确认的帧：发送窗口已经发送过的且已经被接收端发送确认消息且被接受的帧个数。此时从到都是已经发送且被确认的帧，发送窗口的开始索引为。已经移动发送窗口次。
2. 已经发送但仍等待确认的帧：在发送窗口中已经被发送但是因为确认信息在路上所以等待确认的，此时到都是已经发送但仍等待确认的帧。
3. 还能发送的帧：即在发送窗口中还没有被发送的帧。还能发送的帧个数为个。开始的索引为，结束的索引为。
4. 还不能发送的帧：在发送窗口后面的不能被发送的帧，开始的索引为。

#### GBN发送方必须响应的事件

1. 上层调用：上层发送数据，发送方必须检查发送窗口是否已满，未满则产生帧并发送，若已满则先缓存数据，等窗口不满时再发送。
2. 收到一个：对号帧采取累积确认，偶尔捎带确认（在计算机通信中，当一个数据帧到达的时候，接收方并不是立即发送一个单独的控制帧，而是抑制一下自己并且开始等待，直到网络层传递给它下一个分组。然后，确认信息被附在往外发送的数据帧上——帧头中的域。实际上，确认报文搭了下一个外发数据帧的便车。这种“将确认暂时延迟以便可以钩到下一个外发数据帧”的技术称为捎带确认）的方式，即标明接收方已经收到号帧和其之前的全部帧。
3. 超时事件：如果出现超时，则发送方重传所有已发送但是未被确认的帧，即超时以及**后面**的帧。接收方、发送方只按**顺序**接受帧，对于其非期待（**乱序**）的确认帧则丢弃。

#### GBN接收方要做的事件

1. 如果正确收到号帧且顺序一致，则接收方为帧发送一个，并将该帧中的数据部分上交上层。
2. 其余情况都丢弃帧，并为最近按序接收的帧重新发送，向发送方要重传帧。接收方无需缓存任何失序帧，只用维护下一个按序接收的帧序号。

#### GBN运行

假如发送窗口尺寸为：

![GBN](data:image/png;base64;base64,)

GBN

#### GBN窗口长度

采用个比特对帧编号，则接收窗口的尺寸应满足。

如果帧太大会让接收方无法区分新帧旧帧。为什么只到？因为假设最坏情况下这轮从开始的所有的帧都未收到确认，由于使用累计确认，则最大编号会空留一个编号只到，计算机就明白这里前面的还没有收到不会继续发送了，如果是全部用完编号会到，如果此时接收方收到号确认，会认为前面的全部收到这是第二轮的编号从而跳过了这轮的确认环节。

#### GBN协议特点

* 因连续发送数据帧而提高了信道利用率。
* 在重传时必须把原来已经正确传送的数据帧重传，从而令传送速率降低。

### 选择重传协议（SR）

累计确认会导致批量重传问题，所以为了实现只重传出错的帧，解决的办法就是设置单个确认而非累计确认，同时加大接收窗口，缓存乱序到达的帧。

发送窗口中分为已经发送被确认的帧、已经发送但等待确认的帧、还能发送的帧三种，其中这三种帧不一定是连续的，而只有发送窗口的第一个帧是已经发送被确认的帧发送窗口才能移动一格或多格。

接收窗口分为希望收到但是没有收到的帧、希望收到且已收到的帧、等待接收的帧三种，其中这三种帧不一定是连续的，接收窗口中收到且确认的帧位于缓存之中，而只有接收窗口的第一个帧是希望收到且已收到的帧接收窗口才能移动一格或多格。

#### SR发送方必须响应的事件

1. 上层调用：上层发送数据，发送方必须检查发送窗口是否已满，未满则产生帧并发送，若已满则先缓存数据，等窗口不满时再发送。
2. 收到一个：如果收到对应的帧序号在窗口内，则发送方将那个被确认的帧标记为已接收。如果该帧序号是窗口的下界（最左边），在窗口向前移动到具有最小序号的未确认帧处。如果窗口移动且有序号在窗口内未发送的帧，则发送这些帧。
3. 超时事件：每一个帧都具有自己的计时器，一个超时事件发生后只重传一个帧。

#### SR接收方要做的事件

1. 确认一个正确接收的帧而不管是否按序。失序的帧将被缓存，并返回发送方一个该帧的确认帧，直到所有比其序号更小的帧都被接收为止，这时才能将这一批帧交付上层。
2. 然后向前移动接收窗口。
3. 如果收到了接收窗口前的一个接收窗口长度以内的帧，就返回一个表明发送方超时重传的帧已经得到了确认。
4. 如果是其他情况就忽略该帧。

#### SR运行

假如发送窗口、接收窗口尺寸为：

![SR](data:image/png;base64;base64,)

SR

#### SR窗口长度

发送窗口最好等于接收窗口，大于会溢出，小于意义不大。

采用个比特对帧编号，则窗口的尺寸应满足。（即编号数量的一半）

如下面以到编号，表示使用两位编号，而滑动窗口大小为。

![SR窗口长度](data:image/png;base64;base64,)

SR窗口长度

所以接收方就不知道发送的号帧是新帧还是旧帧。此时滑动窗口应该大小为。

#### SR协议特点

* 对数据帧逐一确认，收到一个确认一个。
* 只用重传出错帧。
* 接收方有缓存。
* 发送窗口与接收窗口相等。
* 全部收到后一起上传。

## 介质访问控制

* 点对点链路：两个相邻结点通过一个链路链接，无第三者，常用于广域网。如协议。
* 广播式链路：所有主机共享通信介质，常用于局域网。典型拓扑结构：总线型与星型。

介质访问控制就是采取措施使得两对结点之间的通信不会互相干扰。分为：

* 静态划分信道：信道划分介质访问控制
  + 频分多路复用。
  + 时分多路复用。
  + 波分多路复用。
  + 码分多路复用。
* 动态划分信道：
  + 随机访问介质访问控制（所有用户可随机发送信息，可占用全部带宽）：
    - 协议。
    - 协议。
    - 协议。
    - 协议。
  + 轮询访问介质访问控制：
    - 轮询协议。
    - 令牌传递协议。

### 信道划分介质访问控制

使用一条共享信道，但是通过多路复用技术（多格信号组合在一条物理信道上传输使得多个终端设备共享信道资源并提高信道利用率的技术）组合进行传输，提高了信道的利用率。

#### FDM

用户在分配到一定的频带后，在通信过程中自始至终都占用这个频带。频分复用的所有用户在同样时间占用不同的频率带宽资源。

* 充分利用传输介质带宽，系统效率较高。
* 技术较成熟，实现容易。

使用较少，而是使用较多，这是因抗干扰能力强，能逐级再生整形避免干扰累计，且数字信号易于自动转换，所以用于传输模拟信号，用于传输数字信号。

#### TDM

将时间划分为一段段等长的时分复用帧（帧：在物理层传输的比特流所划分的帧，表明一个周期）。每一个时分复用的用户在每一个帧中占用固定序号的时隙，所有用户轮流占用信道。

* 当用户使用率较低时会导致信道的利用率很低
* 用户的等待时间长。

分为同步时分多路复用和异步时分多路复用（统计时分复用）.

统计时分复用：

* 添加了一个集中器，将不同用户分散的数据集中在一起，单位时间的数据组成一个帧，再发送。
* 每一个帧中的时隙数小于连接在集中器上的用户数。
* 每个用户有数据就随时发送给集中器的输缓存，然后集中器按顺序依次扫描输入缓存，把缓存中的输入数据放入帧中，一个帧满了就发出。
* 帧并不是固定分配时隙，而是按需动态分配时隙。

#### WDM

就是光的频分多路复用，根据同一根光纤中传输多种不断波长（频率）的光信号，根据不同的波长做用波长分解复用器分解出来。

#### CDM

码分复用既共享了空间也共享了时间。

码分多址（）是码分复用的一种方式。

* 一个比特分为多个码片/芯片，每一个站点被指定一个唯一的位的码片序列。
* 发送时站点发送码片序列，发送时发送码片序列反码（一般写作）。
* 如何复用：多个站点同时发送数据的时候，要求各个站点码片序列相互正交。
* 如何合并：各路数据在信道中按位线性相加。
* 如何分离：合并的数据和源站码片序列规格化内积。

假如站点的码片序列被指派为，则站发送就表示发送比特，发送就表示发送比特。

为了方便，按惯例将码片中的写为，将写为，因此站的码片序列是。

令向量表示站的码片向量，令表示站的码片向量。两个不同站的码片序列正交，即向量和的规格化内积为：。

任何一个码片向量和该码片向量自身的规格化内积都是，任何一个码片向量和该码片反码的向量的规格化内积是。即自己×自己 ，自己×别人 ，自己×反码。

令向量为，即。

当站向站发送数据时，就发送了向量。当站向站发送数据时，就发送了向量。两个向量到了公共信道上就进行叠加，实际上就是线性相加，得到。

到达站后，进行数据分离，如果要得到来自站的数据，站就必须知道站的码片序列，让与进行规格化内积。根据叠加原理，其他站点的信号都在内积的结果中被过滤掉了，内积的相关项都是，而只剩下站发送的信号。得到，所以站发出的数据是。同理，如果要得到来自站的数据，那么，因此从站发送过来的信号向量是一个反码向量，代表。

* 频谱利用率高。
* 抗干扰能力强。
* 保密性强。
* 语音质量好。
* 减少投资和降低运行成本。
* 主要用于无线通信系统，特别是移动通信系统。

### ALOHA协议

#### 纯ALOHA协议

不监听信道，不按时间槽发送，随机重发（想发就发）。

![ALOHA](data:image/png;base64;base64,)

ALOHA

* 检测冲突：如果发送冲突，接收方就会检测到差错然后不予确认，发送方在一定时间内收不到确认就判断冲突。
* 解决冲突：超时后等待随机时间重传。

假设网络负载（时间内所有站点发送成功的和未成功而重传的帧数）为，则纯网络的吞吐量利用率（时间内成功发送的平均帧数）为。当时极大，。

#### 时隙ALOHA协议

把时间划分为若干个相同的时间片（时间槽），所有用户只能在时间片的开始时刻同步接入网络信道，若发生冲突则必须等到下一个时间片开始的时刻再发送。

假设网络负载（时间内所有站点发送成功的和未成功而重传的帧数）为，则间隙网络的吞吐量利用率（时间内成功发送的平均帧数）为。当时极大，。

![时隙ALOHA](data:image/png;base64;base64,)

时隙ALOHA

### CSMA协议

载波监听多路访问协议。发送前监听信道。当信道空闲发送帧，当信道忙推迟发送。

* 载波监听：表示每一个站在发送数据之前都要检查一下总线上是否有其他计算机在发送数据。当几个站同时在总线上发送数据时，总线上的信号电压摆动值就增大，因为互相叠加。当一个站检测到的信号电压摆动值超过一定门限值时，就认为总线上至少两个站在同时发送数据，表明产生了碰撞。
* 多点接入：表示许多计算机以多点接入的方式连接在一根总线上。

#### 1-坚持CSMA

* 发送信息前监听信道。
* 当信道空闲时会直接传输，不必等待。
* 当信道忙时会一直监听，直到空闲立刻传输。
* 如果冲突（一段时间内未收到确认），则等待一个随机长的时间再监听。
* 优点：只要媒体空闲站点就立马发送，避免媒体使用率的损失。
* 缺点：若有两个或以上的站点有数据要发送，冲突就无法避免。

#### 非坚持CSMA

* 发送信息前监听信道。
* 当信道空闲时会直接传输，不必等待。
* 当信道忙时会等待一个随机时间之后再进行监听。
* 优点：采用随机的重发延迟时间可以减少冲突发生的可能性。
* 缺点：可能存在所有站点都在延迟等待中，使得媒体空闲，降低媒体使用率。

#### p-坚持CSMA

* 发送信息前监听信道。
* 当信道空闲时以概率直接传输，不必等待，概率等待到下一个时间槽再侦听，以同样的概率侦听或传输。
* 当信道忙时会等待一个随机时间之后再进行监听。
* 优点：既能像非坚持算法减少冲突，又能像-坚持算法那样减少媒体空闲时间。
* 缺点：无法在冲突时就发现，所以发生冲突时还是坚持发送完数据帧，从而造成浪费。

| 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 | 继续坚持侦听 | 放弃侦听，等待一个随机的时间后再侦听 | 持续侦听，直至信道空闲 |

### CSMA/CD协议

载波监听多点接入/碰撞检测协议。是协议的完善，不仅先监听再发送，还可以边监听边发送，当发现碰撞就立刻停止发送，就避免了协议的无法在冲突时就发现，发生冲突时还是坚持发送完数据帧造成浪费的缺点。

所以说明这个协议是适用于总线型或半双工网络上。

站点最迟总线**最远**一个端到另一个端的**往返**传播时延（**争用期/冲突窗口/碰撞窗口**）才会知道自己发送的数据没有和别人碰撞（即撞到目标站点前了），，所以只要该站在两倍的总线端到端的传播时延时间内没有检测到碰撞，那么就可以肯定本次发送没有碰撞。

#### 确定重传时机

采用截断二进制指数规避算法。

1. 确定基本退避（推迟）时间为争用期。
2. 定义参数，其表示重传次数，但是不超过。当重传次数不超过时，等于重传次数，当重传次数大于时，就一直等于。
3. 从离散的整数集合之中随机取出一个数，重传所需要的退避的时间就是倍的基本退避时间，即。
4. 当重传达次仍不能成功时，就说明网络太拥挤，认为该帧永远无法正确发出，就抛弃该帧并向高层报错。

令重传时间为。

* 第一次重传，，，。
* 第二次重传，，，。
* 第三次重传，，

#### 最小帧长

如果帧太短，那么如果发生碰撞就很容易发送完才检测到发生碰撞而无法停止发送。所以就需要规定一个最小的帧长。

最短帧长等于争用期时间内发出的比特数。

所以帧的传输时延至少要两倍于信号在总线中的传播时延。

即最小帧长=总线单向传播时延×数据传输速率×=总线长度÷传播速率×数据传输速率×=争用期×数据传输速率。

注意：以太网规定争用期为，最短帧长为，凡是小于则被判定为无效帧。

### CSMA/CA协议

载波监听多点接入/碰撞避免协议。对比协议只能检测碰撞，而协议可以避免碰撞。且协议只能用于总线类型的以太网，而可以用于无线局域网。

* 发送信息前监听信道。
* 当信道空闲时发送端发出（），包括发射端的地址、接收端的地址、下一份数据将持续发送的时间等信息。
* 当信道忙时会等待一个随机时间之后再进行监听。
* 接收端接收到后，会响应（）。
* 发送端收到后，开始发送数据帧，同时预约信道，发送方告知其他站点要传输多久的数据。
* 接收端收到数据帧后，将用来检验数据是否正确，正确则响应帧。
* 发送方收到帧后才可以进行下一个数据帧的发送，若没有则一直重传一直到规定重发次数为止（使用二进制指数退避算法确定推迟时间）。

为了尽量避免碰撞，规定，所有的站完成发送后，必须再等待一段很短的时间（继续监听）才能发送下一帧。这段时间称为帧间间隔（，）。帧间间隔的长短取决于该站要发送的帧的类型。使用了种：

1. （短）：最短的，用来分隔属于一次对话的各帧，使用的帧类型有帧、帧、分片后的数据帧，以及所有回答探询的帧等。
2. （点协调）：中等长度的，在操作中使用。
3. （分布式协调：最长的，用于异步帧竞争访问的时延。

协议与协议的相同点：

1. 都是需要先侦听信道。
2. 冲突后都会进行有限的重传。

协议与协议的不同点：

1. 传输介质不同：协议无线传输；协议有线则总线型传输。
2. 载波检测方式不同：协议采用能量检测（）、载波检测（）和能量载波混合检测三种方式；协议采用电缆电压检测。
3. 协议避免冲突；协议检测冲突。

### 轮询访问介质访问控制

信道划分介质访问控制协议在网络负载重时信道效率高且公平，而在负载轻时则效率低；随机访问控制协议在网络负载重时会产生冲突开销，而负载轻时效率会比较高，单个结点可以利用信道全部带宽。

也称为轮流协议。即减少产生冲突，又尽量发送时占用全部带宽。

#### 轮询协议

主结点会轮流以发送数据帧的形式询问从属结点是否发送数据，没有被询问的从属结点无法发送数据。

问题：

1. 轮询开销。
2. 等待延迟。
3. 单点故障。

#### 令牌传递协议

一般使用令牌环网实现，逻辑上是环型的，但是物理上是星型的。

![令牌传递协议](data:image/png;base64;base64,)

令牌传递协议

用于转发令牌。而令牌是一个特殊格式的控制帧，不包含任何信息，在令牌环上循环，控制信道使用，只有有令牌才能发送数据，确保同一个时刻只有一个结点独占信道。

每一个结点都可以在**令牌持有时间**内获得发送数据的权利，而不能无限制的持有令牌，超过时间无论是否发送完成都要归还令牌。

问题：

1. 令牌开销。
2. 等待延迟。
3. 单点故障。

采用令牌传输方式的网络基本上是负载较重，通信量较大的网络。

## 局域网

使用广播信道。

### 局域网特性

特点：

1. 覆盖范围较小。
2. 采用专门的传输介质（双绞线、同轴电缆）进行联网，数据传输速率较高。
3. 通信延迟时间端，误码率低，可靠性高。
4. 各站平等共享信道。
5. 多采用分布式控制与广播式通信，可以广播与组播。

#### 拓扑结构

* 星型：中心结点时控制中心，任意两个结点之间的通信最多只用两步，传输速度快，且网络结构简单，建网容易，便于控制与管理，但是可靠性低，网络共享能力差，会单点故障。
* 总线型：网络可靠性高，网络结点之间响应速度快，共享资源能力强，设备投入量少，成本低，安装使用方便，当某结点故障时对整个网络系统影响小。但是总线损坏也会造成巨大影响。
* 环型：通信设备和线路比较节省，有单点故障问题，由于线路封闭，不易于拓展，系统响应延时长，且信息传输效率较低。
* 树型：易于拓展，易于隔离故障，也容易单点故障。

#### 局域网介质访问控制

1. ：常用于总线型局域网，也用于树型网络。
2. 令牌总线：常用于总线型局域网，也用于树型网络。把总线或树型网络中的各个工作站按一定的顺序如按接口地址大小排列形成一个逻辑环，只有令牌持有者才能控制总线发送信息。
3. 令牌环：用于环型局域网，如令牌环网。

#### 局域网的分类

1. 以太网：是应用最为广泛的局域网。包括标准以太网（）、快速以太网（）、千兆以太网（）和以太网，都符合系列标准规范。逻辑拓扑总线型，物理拓扑是星型或拓展星型，使用。
2. 令牌环网：物理星型，逻辑环型，基本上已经过时。
3. 网：物理双环结构，逻辑环型，使用光纤，造价高，基本上没有使用到。
4. 网：较新的单元交换技术，用字节固定长度的单元进行交换。
5. 无线局域网：采用标准。

#### IEEE802标准

局域网与城域网技术标准，使用的范围有以太网、令牌环、无线局域网。

* ：与物理层技术规范。
* ：令牌环网的介质访问控制协议与物理层技术规范。
* ：的光纤技术规范。
* ：无线局域网的介质范文控制协议与物理层技术规范。

局域网简称为以太网。

#### MAC子层与LLC子层

标准将局域网的数据链路层分为逻辑链路层子层与介质访问控制子层。

* 逻辑链路控制：
  + 建立和释放数据链路层连接、提供与高层接口、差错控制、帧加序号。
  + 为网络层提供服务：无确认无连接、面向连接、带确认无连接、高速传送。
* 介质访问控制：
  + 组帧拆帧（根据的序号），帧寻址和识别，竞争处理，比特差错控制等。
  + 屏蔽了不同物理链路种类的差异性。

### 以太网

* 是基带总线局域网规范，使用技术，主要负责物理层与数据链路层的规范。
* 使用协议，其特点是无连接，不可靠，不确认，不对帧编号。
* 传输介质由粗同轴电缆到细同轴电缆再到双绞线+集线器。
* 拓扑结构逻辑上总线型，物理上星型。

#### 以太网传输

当以太网发送一个数据，那么以太网将以广播形式发送，以太网上的**所有主机**包括发送端本身都能收到数据。

通过曼彻斯特编码以中间电平跳动作为同步信号，无序其他同步操作。

#### 以太网标准

* ：第一个局域网产品（以太网）规约。
* ：指定的第一个的以太网标准，帧格式有所不同。

#### 以太网类型

| 参数 | 1OBASE5 | 10BASE2 | 1OBASE-T | 10BASE-FL |
| --- | --- | --- | --- | --- |
| 传输媒体 | 基带同轴电缆（粗缆） | 基带同轴电缆（细缆) | 非屏蔽双绞线 | 光纤对（850nm） |
| 编码 | 曼彻斯特编码 | 曼彻斯特编码 | 曼彻斯特编码 | 曼彻斯特编码 |
| 拓扑结构 | 总线形 | 总线形 | 星形 | 点对点 |
| 最大段长 | 500m | 185m | 100m | 2000m |
| 最多结点数目 | 100 | 30 | 2 | 2 |

对于以太网

* 表示传输速率为，表示传输基带信号，代表使用双绞线。现在一般采用无屏蔽双绞线（）。
* 网络拓扑物理上是星型，逻辑上是总线型，每段双绞线最长为。
* 采用曼彻斯特编码。
* 使用介质访问机制。
* 超过覆盖范围拓展使用中继器。

当速率大于时就可以称为高速以太网（快速以太网）。

快速以太网仍然使用协议，它采用保持最短帧长不变而将最大电缆长度减少到的方法，使以太网的数据传输速率提高至及以上。

1. 以太网：在双绞线上传输基带信号的星型拓扑以太网，仍使用的协议。支持全双工与半双工，可以全双工方式工作下无冲突。所以全双工方式下不使用协议。
2. 吉比特以太网：在光纤（）或对双绞线（）上传送信号。支持全双工与半双工，可以全双工方式工作下无冲突。所以全双工方式下不使用协议。
3. 吉比特以太网：在光纤上传送信号。只支持全双工，无冲突，所以不使用协议。

#### 适配器与MAC地址

* 计算机与外界局域网的连接通过通信适配器完成，过去通过单独网络接口卡即网卡实现。
* 网卡主要工作在物理层和数据链路层。
* 适配器上与处理器和存储器，包括和。
* 上有计算机硬件地址，称为介质访问控制地址。
* 在局域网中，硬件地址又被称为物理地址或地址。
* 地址是全球唯一的位二进制适配器地址，即个字节，一般用连字符或冒号分隔的个十六进制数表示。前位代表厂家（规定），后位由厂家指定。

#### 以太网MAC帧

如以太网标准所说一共分为两种标准，最常用的是的格式：

![MAC帧格式](data:image/png;base64;base64,)

MAC帧格式

以太网帧附加信息，规定帧最短为，所以数据最短为，规定数据部分最长为，所以以太网帧最长为。

* 物理层会插入前导码，用于接收端与发送端时钟同步。包括前同步码（用于快速实现帧的比特同步）与帧开始定界符（表示后面的信息就是帧）两个部分。
* 地址即地址。
* 类型表示数据报的类型，表示数据应该交给哪个协议实体处理。。
* 数据最小值为是因为算法规定以太网最小帧长为字节，所以数据最小就是字节，最长为字节。数据包括高层的协议信息。
* 填充用于帧长太短来填充到字节，所以长度为字节。
* ：校验范围从目的地址到数据端末尾，采用位循环冗余码，但是不用校验帧的前导码。
* 以太网帧没有结束定界符是因为以太网采用曼彻斯特编码，如果与数据必然是电压变化的，而如果电压没有变化就代表已经结束，所以不需要结束定界符。但是帧也要加尾部。

与的区别：

1. 第三个字段是长度/类型。
2. 当长度/类型字段小于时，数据字段必须装入子层。

### 无线局域网

#### 有固定基础设施

用于有固定基础设施的无线局域网。使用星型拓扑。其中心位接入点。

无线局域网最小构建是基本服务集。一个基本服务集包括一个基站和若干移动站。

（基本服务集）不超过字节，代表基站的地址。

一个基本服务集覆盖的地理范围称为一个基本服务区。

一个可以独立也可以通过连接到一个分配系统然后连接到另一个，构建一个扩展服务集。通过门桥为无线用户提供以太网的接入。

#### 无固定基础设施

称为自组网络，没有，而是一些平等状态的移动站相互通信构成的临时网络。因此都具有路由器的功能。

#### IEEE802.11的MAC帧头格式

![IEEE802.11的MAC帧头格式](data:image/png;base64;base64,)

IEEE802.11的MAC帧头格式

对于（无线分布式系统）：

* 地址1：接收端，接收端基站地址。
* 地址2：发送端，发送端基站地址。
* 地址3：目的地址，目标主机的地址。
* 地址4：源地址，发送端的地址。

![MAC帧头格式表格](data:image/png;base64;base64,)

MAC帧头格式表格

#### 碰撞检测

无线局域网不使用协议，而使用协议，因为发送过程中不需要进行冲突检测：

1. 在无线局域网的适配器上，接收信号的强度往往远小于发送信号的强度，因此若要实现碰撞检测，那么硬件上的花费就会过大。
2. 在无线局域网中，并非所有站点都能听见对方，由此引发了隐蔽站和暴露站问题，而“所有站点都能够听见对方”正是实现协议必备的基础。

### 虚拟局域网

即。基于交换技术按逻辑进行划分局域网一个广播域。可以隔离冲突域，也可以隔离广播域。

* 有效共享网络资源。
* 简化网络管理。
* 提高网络安全性。

可以隔离冲突域也可以隔离广播域。

#### 虚拟局域网实现

通过标准定义。它在以太网帧中插入一个四字节的标识符（插入在源地址字段和类型字段之间)，称为标签，用来指明发送该帧的计算机属于哪个虚拟局域网。插入标签的帧称为帧。由于首部增加了四字节，因此以太网的最大帧长从原来的字节变为字节。

虚拟局域网通过标识符实现逻辑分组和管理，不需要额外的硬件支持。

技术可以将一个物理局域网在逻辑上划分成多个广播域，即划分为多个。技术部署在数据链路层，用于隔离二层流量。同一个内的主机共享同一个广播域，它们之间可以进行二层通信。而间的主机属于不同的广播域，不能直接实现二层互通。这样，广播报文就被限制在各个相应的内，并提高了网络的安全性。本质上虚拟局域网使用的是三层架构的交换技术（否则不能隔离广播域和冲突域）。

#### 虚拟局域网划分

虚拟局域网的划分与物理地址无关，可以处于不同的实际局域网中，本身只是一种技术。划分方式只能基于物理层、数据链路层和网络层（不能基于更上层的数据）：

* 基于交换机端口。
* 基于网卡地址。
* 基于网络层子网地址。
* 基于协议。
* 基于策略。

## 广域网

是单一的网络。的通信子网主要使用分组交换技术，达到资源共享的目的。

|  | 广域网 | 局域网 |
| --- | --- | --- |
| 覆盖范围 | 很广，通常跨区域 | 较小，通常在一个区域内 |
| 连接方式 | 结点之间都是点到点连接，但为了提高网络的可靠性，一个结点交换机往往与多个结点交换机相连 | 普遍采用多点接入技术 |
| OSI参考模型层次 | 三层：物理层，数据链路层，网络层 | 两层：物理层，数据链路层 |
| 着重点 | 资源共享 | 数据传输 |

广域网不等于互联网，因为互联网可以接入不同类型的网络，可以是局域网也可以是广域网。

### PPP协议

点对点协议是使用最广泛的面向字节的数据链路层协议，用户使用拨号电话接入因特网时一般使用协议。

#### PPP协议的特点

* 简单：对于链路层的帧，无须纠错、序号、流量控制。
* 只支持全双工链路。
* 封装成帧：具有帧定界符。
* 透明传输：面向字符。对于与帧定界符一样的比特组合的数据应如何处理：异步线路使用字节填充（因为按字节或字符传送），同步线路使用比特填充（因为按比特传送）。
* 多种网络层协议：可以采用多种协议。两端可以连接不同的网络层协议。
* 多种类型链路：串行/并行，同步/异步。
* 差错检测：错就丢弃，只检错不纠错。
* 检测连接状态：链路是否正常工作。
* 最大传送单元：数据部分最大长度。
* 网络层地址协商：需要知道通信双方的网络层地址。
* 数据压缩协商：传输数据时需要对数据进行压缩。
* 无需支持多点线路：只用满足点对点就可以了。
* 动态分配地址：因为可用于拨号连接。
* 身份验证：支持和。

|  | PAP | CHAP |
| --- | --- | --- |
| 安全性 | 低 | 高 |
| 传输密码 | 密码明文 | 密码哈希值 |
| 实现方式 | 两次握手 | 三次握手 |
| 请求方式 | 被叫方提出，主叫方响应 | 主叫方提出并发送随机哈希值，被叫方返回一个包含这个哈希值的数据报，主叫方确认后发送一个连接成功的数据报连接 |

#### PPP协议的组成

1. 一个将数据报封装到串行链路（同步或异步）的方法。
2. 链路控制协议：建立并维护数据链路连接，并进行身份验证。
3. 网络控制协议：支持多种网络层协议，每个不同的网络层协议都需要一个相应的来配置，为网络层建立和配置逻辑连接。

#### PPP协议的工作流程与帧格式

![PPP协议状态图](data:image/png;base64;base64,)

PPP协议状态图

![PPP帧格式](data:image/png;base64;base64,)

PPP帧格式

* ：表示帧定界符，用二进制表示就是。当数据内出现帧定界符就需要插入转义字符：。
* ：表示，未完善。
* ：表示，未完善。
* 协议：表示信息部分的内容，是数据报、数据或网络层控制数据等。
* ：用实现的帧检验序列。

### HDLC协议

高级数据链路控制协议是一个在同步网上传输数据、面向比特的数据链路层协议。

由根据协议扩展开发，不属于族。

#### HDLC协议的特点

* 可以透明传输。
* 面向比特。只使用比特填充法。
* 易于硬件实现。
* 只支持全双工通信。
* 所有帧采用检验。
* 对信息帧进行顺序编号，可以防止漏收和重收，传输可靠性高。

#### HDLC协议的站

1. 主站：发送命令（包括数据信息）帧、接收响应帧，并负责对整个链路的控制系统的初启、流程控制、差错检测或恢复。
2. 从站：接收由主站发来的命令帧，向其发送响应帧，并配合主站参与差错恢复等链路控制。
3. 复合站：既能发送，又能接收命令帧和响应帧，并负责整个链路的控制。

#### 数据操作方式

1. 正常响应方式：只有经过主站同意从站才能传输数据。
2. 异步平衡方式：每一个复合站都能自主传输数据。
3. 异步响应方式：无需经过主站同意从站就能传输数据。

#### HDLC协议的帧格式

![HDLC帧格式](data:image/png;base64;base64,)

HDLC帧格式

* ：表示帧定界符，用二进制表示就是。当数据内出现帧定界符就需要插入转义字符。
* ：表示，当数据操作方式是正常响应方式或异步响应方式，都是表示从站的地址，而如果是异步平衡方式，则填充应答站的地址。
* ：表示：
  1. 信息帧（）：第一位是，用于传输数据信息，或使用捎带技术对数据进行确认。
  2. 监督帧（）：，用于流量控制与差错控制，执行对信息帧的确认、请求重发和请求暂停发送等功能。
  3. 无编号帧（）：，用于提供对链路的建立、拆除等多种控制功能。

#### HDLC协议与PPP协议的联系与区别

联系：

1. 都只支持全双工链路。
2. 都可以实现透明传输。
3. 都只检测差错而不纠正差错。

区别：

|  | PPP协议 | HDLC协议 |
| --- | --- | --- |
| 面向对象 | 面向字节 | 面向比特 |
| 是否拥有协议字段 | 2B协议字段 | 没有 |
| 序号或确认机制 | 无 | 有 |
| 是否可靠 | 不可靠 | 可靠 |
| 透明传输技术 | 同步字符填充异步比特填充 | 比特填充 |
| 实现 | 软件 | 几乎硬件 |

## 数据链路层设备

主要包含网桥与交换机。

### 网桥（Bridge）

多格以太网通过网桥连接后就形成了更大的以太网，原来的每个以太网就是一个网段。

可以互联不同的物理层、不同的子层及不同速率的以太网。

网桥工作在数据链路层的子层。

当网桥收到一个帧时，不会广播此帧，而是先检测该帧的目的地址，然后确定转发到哪一个接口或丢弃（过滤）。

#### 网桥的优点

1. 过滤通信量，增大吞吐量。
2. 扩大了物理范围。
3. 提高了可靠性。
4. 可互联不同物理层、不同子层和不同速率的以太网。

#### 网桥的缺点

1. 增加时延。
2. 子层没有流量控制功能（流量控制需要编号机制，而编号机制在子层）。
3. 不同子层的网段连接需要帧格式转换。
4. 适用于少数据量的局域网，否则会出现广播风暴。

#### 网桥的种类

* 透明网桥：指以太网上的站点并不知道所发送的帧将经过哪几个网桥，是一个即插即用的设备。通过自学习的方式来寻找和更新路径。
* 源路由网桥：在发送帧时，把详细的最佳路由信息（路由最少/时间最短）放在帧的首部中。源站以广播的方式向欲通信的目的站发送多个发现帧，目的站再根据每一个发现帧的路径发送响应帧，从而以枚举的方式获得最优路径。
* 使用源路由网桥可以利用最佳路由。若在两个以太网之间使用并联的源路由网桥，则还可使通信量较平均地分配给每个网桥。采用透明网桥时，只能使用生成树，而使用生成树一般并不能保证所用的路由是最佳的，也不能在不同的链路中进行负载均衡。
* 透明网桥和源路由网桥中提到的最佳路由并不是经过路由器最少的路由，而可以是发送帧往返时间最短的路由，这样才能真正地进行负载平衡，因为往返时间长说明中间某个路由器可能超载了，所以不走这条路，换个往返时间短的路走。

### 交换机（Switch）

交换机就是多接口网桥。每一个接口都是一个冲突域。可以独占传输媒体带宽

#### 交换机的种类

直通式交换机：查完目的地址（）直接转发。延迟小，可靠性低，无法支持具有不同速率的端口的交换。

存储转发式交换机：将帧放入高速缓存，并检查是否正确，正确则转发，错误则丢弃。延迟大，可靠性高，可以支持具有不同速率的端口的交换。

#### 交换机的特点

* 以太网交换机的每个端口都直接与单台主机相连（普通网桥的端口往往连接到以太网的一个网段)，并且一般都工作在全双工方式。
* 交换机的总带宽是各端口带宽之和。
* 以太网交换机能同时连通许多对端口，使每对相互通信的主机都能像独占通信媒体那样，无碰撞地传输数据。支持多对用户同时通信。
* 以太网交换机也是一种即插即用设备（和透明网桥一样)，其内部的帧的转发表也是通过自学习算法自动地逐渐建立起来的。
* 以太网交换机由于使用了专用的交换结构芯片，因此交换速率较高。以太网交换机独占传输媒体的带宽。

使用交换机后带宽总容量会变为原来的倍，为端口数。

#### 交换方式

* 直通式交换机：
  + 只检查帧的**目的地址**（），这使得帧在接收后几乎能马上被传出去。
  + 这种方式速度快，但缺乏智能性和安全性，也无法支持具有不同速率的端口的交换。
* 存储转发式交换机：
  + 先将接收到的帧缓存到高速缓存器中，并检查数据是否正确，确认无误后通过查找表转换成输出端口将该帧发送出去。
  + 如果发现帧有错，那么就将其丢弃。
  + 存储转发式的优点是可靠性高，并能支持不同速率端口间的转换。
  + 缺点是延迟较大。

### 交换机与网桥

1. 网桥的端口一般连接局域网，而交换机的端口一般直接与局域网的主机相连。
2. 交换机允许多对计算机同时通信，而网桥仅允许每个网段上的计算机同时通信。
3. 网桥采用存储转发进行转发，而以太网交换机还可以采用直通方式进行转发，且以太网交换机采用了专用的交换结构芯片，转发速度比网桥快。

### 冲突域与广播域

* 冲突域：在以太网中，如果某个网络上的两台计算机在同时通信时会发生冲突（即同一时间只能一台主机发送信息），那么这个网络就是一个冲突域（）。如果以太网中各个网段以集线器连接，因为不能避免冲突，所以它们仍然是一个冲突域。（一个数据链路层设备的接口一个冲突域）
* 广播域：网络中能接收任一设备发出的广播帧的所有设备的集合。即如果一个站点发出一个广播信号，那么所有能接收到该信号的设备范围就是一个广播域。（一个网络层设备一个广播域）
* 网段：指一个计算机网络中使用同一物理层设备（传输介质、中继器、集线器等）能直接通信的一部分。

一个网段就是一个冲突域，一个局域网就是一个广播域。

|  | 能否隔离冲突域 | 能否隔离广播域 |
| --- | --- | --- |
| 物理层设备 | 否 | 否 |
| 链路层设备 | 是 | 否 |
| 网络层设备 | 是 | 是 |

# 网络层

## 基本概念

主要任务是将分组从源端发送到目的端，为分组交换网上的不同主机提供通信服务。在任意结点间进行数据报传输，但不可靠。

### 网络层功能

#### 路由选择与分组转发

即选择最佳路径。

* 路由器的路由表中包含目的网络和到达该目的网络路径上的下一个路由器的地址。
* 分组传输时，源主机和中间路由器都不知道分组到达目的主机需要经过的完整路径。

#### SDN

网络层的主要任务是转发和路由选择。可以将网络层抽象地划分为数据层面（也称转发层面）和控制层面，转发是数据层面实现的功能，而路由选择是控制层面实现的功能。

软件定义网络是近年流行的一种创新网络架构，它采用集中式的控制层面和分布式的数据层面，两个层面相互分离，控制层面利用控制-数据接口对数据层面上的路由器进行集中式控制，方便软件来控制网络。

在传统互联网中，每个路由器既有转发表又有路由选择软件，也就是说，既有数据层面又有控制层面。

在中不需要路由选择软件，所以不需要路由器之间交换路由信息，而是由网络的控制层面的一个逻辑上的远程控制器进行控制，其掌握主机和网络状态，分析最佳路由，通过协议等将转发表（流表）下发给路由器。路由器只需要收到分组、查找转发表、转发分组。

特点：

* 控制和转发功能分离。
* 控制层面集中化。
* 接口开放可编程。

优点：

* 全局集中式控制和分布式高速转发，既利于控制层面的全局优化，又利于高性能的网络转发。
* 灵活可编程与性能的平衡，控制和转发功能分离后，使得网络可以由专有的自动化工具以编程方式配置。
* 降低成本，控制和数据层面分离后，尤其是在使用开放的接口协议后，就实现了网络设备的制造与功能软件的开发相分离，从而有效降低了成本。

缺点：

* 安全风险，集中管理容易受攻击，如果崩溃，整个网络会受到影响。
* 瓶颈问题，原本分布式的控制层面集中化后，随着网络规模扩大，控制器可能成为网络性能的瓶颈。

#### 异构网络互联

* 网络的异构性是指传输介质、数据编码方式、链路控制协议及不同的数据单元格式和转发机制，即物理层与数据链路层均不同。
* 路由器要求物理层、数据链路层、网络层协议不同，由于路由器无法处理上层数据所以要求上层的协议必须相同。

#### 拥塞控制

* 拥塞定义：
  + 如果网络负载增加，网络吞吐量明显小于正常吞吐量，则进入轻度拥塞。
  + 如果网络负载增加，网络吞吐量反而下降，则进入拥塞。
  + 如果网络负载增加，网络吞吐量变为零，则进入死锁。
* 若网络所有结点都来不及接收分组，而要大量丢弃分组的话会使网络处于拥塞状态，所以就需要控制这种拥塞。
* 开环控制：在拥塞之前就提前设计解决。
* 闭环控制：在拥塞时自动调整解决问题。

## IP协议

网际协议是协议栈中两个最主要的协议。与之配套的还有地址解析协议，网际控制报文协议、网际组管理协议，还有一个逆地址解析协议已不被使用。

![TCP/IP协议栈](data:image/jpeg;base64;base64,)

TCP/IP协议栈

协议使用协议，而与使用协议。

### IP数据报

#### IP数据报格式

![IP数据报格式](data:image/png;base64;base64,)

IP数据报格式

当单个数据报太长，就会被划分不同的数据部分加上首部，就是分组。

固定部分，按照五行来展示而不是真的这样排列，一行比特。

![IP数据报具体格式](data:image/png;base64;base64,)

IP数据报具体格式

| 名称 | 注释 | 大小 |
| --- | --- | --- |
| 版本 Version | ipv4或者ipv6 | 4位 |
| 首部长度 IHL | 单位为4字节，同时因为IP数据报固定长度（最小值）为20字节，所以此处最小值为5，即二进制的0101 | 4位 |
| 区分服务 DSCP + ECN | 希望获得哪种服务，用的比较少 | 8位 |
| 总长度 Total Length | 首部+数据的长度，单位为1字节，最大为B，但是实际上永远不会达到该长度，因为有MTU的限制 | 16位 |
| 标识 Identification | 就是一个计数器，用来表示是哪一个数据报的分片，同一个数据报的分片标识相同 | 8位 |
| 标志 Flags | 3bit，用来表示是否分片和分片是否结束；中间位DF（Don’t Fragment）为1表示禁止分片，如果是0代表允许分片；最低位MF（More Fragment）为1表示后面还有分片，如果为0表示最后一片分片 | 3位，但实际有用的只有后两位 |
| 片偏移 Fragment Offset | 用来标记分片之后，该分片在原来的数据报的位置，以8字节为单位 | 13位 |
| 生存时间 Time To Live | 即TTL，每经过一个路由器TTL-1，等于0时自动放弃，根据系统不同默认的TTL不同，为了防止无法传输的数据报在链路中无限传输 | 8位 |
| 协议 Protocol | 用来标记数据部分协议名的字段值，如ICMP：1；IGMP：2；TCP：6；EGP：8；IGP：9；UPD：17；IPv6：41；ESP：50；OSPF：89 | 8位 |
| 首部检验和 Header Checksum | 检验首部的字段是否出错，不包括数据部分，出错就丢弃此数据报 | 16位 |
| 源地址 Source IP Address | 发送方ip地址 | 32位 |
| 目的地址 Destination IP Address | 接收方ip地址 | 32位 |
| 可选字段 Options | 用来排错等安全检测 | 未知，可在0-40位之间 |
| 填充 | 将数据报对齐成4字节的整数倍，数值全部为0 | 未知，根据可选字段来定 |

重点单位：首部长度、总长度、片偏移。

由于具有标识，所以重传的数据不能与之前的数据进行重组。

注意：由于片偏移为的整数倍，所以分片后的数据报的数据部分长度必须在小于的前提下为的整数倍。

#### IP数据报分片

注意和。

* 最大传送单元，数据链路层帧可封装的数据的上限，以太网的是字节。
* 当分组超过时为了传输就必须要分片，而如果分组不允许分片则无法传输，会返回上层一个差错报告。
* 除了最后一个分片，每个分片的长度必然为八字节的整数倍。

注意：不能确定是独立的数据报，还是分片得来的，只有当且片段偏移字段时，才能确定是分片的最后一个分片。

#### 转发分组流程

路由表包括：

1. 目的网络地址。
2. 目的网络子网掩码。
3. 下一跳地址。

路由器转发分组的算法：

1. 提取目的地址。
2. 查看是否可直接交付。若网络直连路由器则直接交付目的主机。
3. 查看是否有特定主机路由（特殊地址，为了测试或安全），就直接按此路由传输。
4. 检测路由表中有无合适路径，若有到达目的地址的路由，则把数据报交付给路由器指明的下一跳路由器。
5. 该路由器找不到对应地址，若该路由器有默认路由，则发送给默认路由器，让其他路由器查看是否有该地址。
6. 超过，丢弃并报告转发分组出错。

### IP地址

默认指地址，全世界唯一的位/字节标识符，标识路由器主机的接口。地址={网络号,主机号}。

#### IP地址类别

地址一共分为五类。

![IP地址类别](data:image/png;base64;base64,)

IP地址类别

其中一些地址被予以保留做特殊用途：

![特殊IP地址](data:image/jpeg;base64;base64,)

特殊IP地址

全的广播地址也称为受限广播地址。

由于网络号不能全或全，所以基本上可用的都要减二。

| 网络类别 | 最大可指派的网络数 | 第一个可指派的网络号 | 最后一个可指派的网络号 | 每个网络中的最大主机数 |
| --- | --- | --- | --- | --- |
| A类 | 126 | 1 | 126 | 16777214 |
| B类 | 16383 | 128.1 | 191.255 | 65534 |
| C类 | 2097151 | 192.0.1 | 223.255.255 | 254 |

为什么类最大是？

因为网络号最开始第一位是，一共八位，所以二进制表示位，最大就是，但是因为是特殊地址，所以将其去掉。

为什么类最大是？

因为网络号最开始第一位是，一共八位，所以二进制表示位，最大就是，最小就是，无特殊地址。

为什么类最大是？

因为网络号最开始第一位是，一共八位，所以二进制表示位，最大就是，最小就是，无特殊IP地址。

#### IP地址特点

* 每个地址都由网络号和主机号两部分组成，因此地址是一种分等级的地址结构。分等级的好处是：
  + 地址管理机构在分配地址时只分配网络号（第一级)，而主机号（第二级）则由得到该网络的单位自行分配，方便了地址的管理。
  + 路由器仅根据目的主机所连接的网络号来转发分组（而不考虑目标主机号），从而减小了路由表所占的存储空间。
* 地址是标志一台主机（或路由器）和一条链路的接口。当一台主机同时连接到两个网络时，该主机就必须同时具有两个相应的地址，每个地址的网络号必须与所在网络的网络号相同，且这两个地址的网络号是互相**不同**的。因此网络上的一个路由器必然至少应具有两个地址（路由器每个端口必须至少分配一个地址）。不然会冲突。
* 用转发器或桥接器（网桥等）连接的若干仍然是同一个网络（同一个广播域），因此该中所有主机的地址的网络号必须相同，但主机号必须不同。
* 在地址中，所有分配到网络号的网络（无论是还是）都是平等的。
* 在同一个局域网上的主机或路由器的地址中的网络号必须是一样的。路由器总是具有两个或两个以上的地址，路由器的每个端口都有一个不同网络号的地址。

### 专用网络通信

#### NAT

网络地址转换是指通过将专用网络地址转换为公用地址，从而对外隐藏内部管理的地址。它使得整个专用网只需要一个全球地址就可以与因特网连通，由于专用网本地地址是可重用的，所以大大节省了地址的消耗。同时，它隐藏了内部网络结构，从而降低了内部网络受到攻击的风险。

在进行转发的时候必须保证内网地址与网络端口都一致，否则不转发。

此外，为了网络安全，划出了部分地址为私有地址。私有地址只用于，不用于连接（因此私有地址不能直接用于，必须通过网关利用把私有地址转换为中合法的全球地址后才能用于），并且允许私有地址被重复使用。

私有地址（可重用地址）：

![私有IP地址](data:image/png;base64;base64,)

私有IP地址

私有地址也叫做可重用地址，不同于一般的全球地址，这些地址可以在机构内部自行分配使用，这张网络就是专用互联网或是本地互联网，当然也无法直接用于与因特网的通信。

* 如果专用网的内部某些主机已经分配到了本地专用地址，又想无加密的跟互联网上的主机通信，就需要网络地址转换。
* 只用在专用网与因特网的路由器上按照软件，安装了软件的路由器就是路由器，其至少有一个有效的外部全球地址。
* 路由器就是该本地专用网的代表，负责与外部因特网联通，分发外部的数据信息，发送时一定会按转换表改变源地址或目的地址。
* 路由器维护转换表，包含端与端的散列值对。每个值都包含网络号与端口号。路由器在发送和接收时都需要根据转换表更改地址。
* 普通路由器工作在网络层，而路由器工作在网络层和传输层，因为传输时需要查看和转换端口号。

#### VPN

当目的地址是私有（专用）地址时路由器一律不转发，而如果在内网的私有地址要转发或接收本地网络其他主机的数据怎么办呢？

可以利用公用的互联网作为本地各专用网之间的通信载体，这种网络就是虚拟专用网。这种网络不同于互联网是因为它只用于本地网络的通信，但是又依靠互联网传输，所以就需要对数据进行加密。

当需要外部机构加入就是外联网；如果成员分布分散，通过某种软件建立通道，这种就是远程接入。

### 子网与超网

分级地址的缺点：

1. 地址空间利用率较低。
2. 两级地址不够灵活。

#### 子网划分

* 由两级地址（网络号+主机号）变为三级地址（网络号+子网号+主机号），将原来的主机号分割出来一部分作为子网号，子网的划分由单位内部完成，但是对外仍表现一个网络，外部无法看到本单位内子网的划分。
* 子网划分的子网号可以全或全。网络地址和主网络的网络地址是重叠的。（与区别）
* 子网的标识依靠子网掩码，其中网络号和子网号的部分全部位数为，而主机号部分全部为。
* 将子网掩码与地址逐位进行与操作，就可以得到子网网络地址。

#### 子网掩码

在使用子网掩码的条件下，路由表不仅要给出目的网络地址和下一跳地址外，还需要给出目的网络的子网掩码。

* 类地址的默认子网掩码是。
* 类地址的默认子网掩码是。
* 类地址的默认子网掩码是。

| 二进制 | 十进制 |
| --- | --- |
| 1000 0000 | 128 |
| 1100 0000 | 192 |
| 1110 0000 | 224 |
| 1111 0000 | 240 |
| 1111 1000 | 248 |
| 1111 1100 | 252 |
| 1111 1110 | 254 |
| 1111 1111 | 255 |

注意：由于子网可以全和全，所以可用子网数量不用减二。

#### IP广播

如之前的保留地址所展示，如果需要进行广播，则包含两种广播方式。

受限（有限）广播地址是位全的地址。受限的广播地址是255.255.255.255。该地址用于主机配置过程中数据报的目的地址，此时，主机可能还不知道它所在网络的网络掩码，甚至连它的地址也不知道。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。

直接广播地址包含一个有效的网络号和一个全的主机号，如果我们要求一个地址的直接广播地址，那么需要根据地址与子网掩码，将网络号部分与出来，然后将所有的主机号位全部置，从而得到的地址就是该网络地址的直接广播地址。

#### CIDR

无分类域间路由选择。

* 一般子网划分在同一个单位下的子网掩码都是相同长度的，而无分类编址就是变长的子网掩码。
* 的子网号可以全全。因为掩码与地址同时传送，所以可以。但是严格情况下不行。
* 构建的超网可以在单位内部根据不同部门的主机个数分配不同数量的子网地址。
* 通过就将子网划分的三级地址又回归为两级地址（网络前缀+主机号）。
* 记法：地址后加上/，然后写上网络前缀的位数。
* 也可用使用：二进制网络前缀\*。
* 把网络前缀都相同的连续的地址组成一个地址块。

如根据，可以得到其二进制是，所以其最小地址就是将竖线后面的位全部变为，最大地址就是将竖线后的位全部变为，即为与，可用地址就是个，地址块就是。

注意：

* 默认路由：如果是直连一个互联网，就直接是，没有指定主机，需要特别记忆。
* 主机路由：对特定目的主机地址专门制定路由，指定子网掩码为，表示子网掩码没有意义。

#### 构成超网

将多个子网构成一个较大的子网就是构成超网，即将网络前缀都相同的连续地址组成“地址块”。一个地址块可以表示很多地址，这也就是路由聚合。

方法就是将网络前缀缩短。

如有两个路由表项和，路由器会发现这两个表项到的网络地址都是同一个，所以可以合并为一个更大的地址块。

最长前缀匹配：使用时，查找路由表时可能得到多个匹配结果，应该选择具有最长网络前缀的路由，前缀越长，地址块越小，路由也越具体。

### IPv6

无论是技术还是技术都无法完全解决地址耗尽的问题，所以就出现了从根本上解决地址耗尽的问题。

解决的问题有：

1. 地址不足，增加到了位。
2. 改进首部格式，简化地址类型，且首部长度固定。
3. 快速处理/转发数据报，去掉了差错检验，不允许分片。
4. 支持（服务质量指一个网络能够利用各种基础技术，为指定的网络通信提供更好的服务能力，是网络的一种安全机制，用来解决网络延迟和阻塞等问题的一种技术）。

#### IPv6帧格式

帧包括字节即位的基本首部，再加上不超过字节的有效载荷，有效载荷中包含着多个拓展首部与数据部分。

其中报头格式如下：

![IPv6报头格式](data:image/png;base64;base64,)

IPv6报头格式

| 名称 | 作用 | 位数 |
| --- | --- | --- |
| 版本 Version | 指明协议版本，此处因为是ipv6所以总是6 | 4位 |
| 优先级 Traffic Class | 区分数据报的类型和优先级 | 8位 |
| 流标签 Flow Label | 和ipv4标识某个数据报分片不同，这是对于一整个数据报流的标记 | 19位 |
| 有效载荷长度 Payload Length | 指的是扩展首部+数据部分的大小，和ipv4的总长度和首部长度都不同，ipv6的首部长度是固定的40字节 | 16位 |
| 下一个首部（报头） Next Header | 基本首部的下一个首部指的是有效载荷里标记的的扩展首部，有效载荷里的扩展首部再指向有效载荷里标记的的扩展首部，直至最后指向数据 | 8位 |
| 跳数限制 Hop Limit | 基本相当于ipv4当中的TTL，每到一个路由器-1.减到0时丢弃 | 7位 |
| 源地址 Source Address | 发送方ipv6地址 | 128位 |
| 目标地址 Destination Address | 接收方ipv6地址 | 128位 |

#### IPv6与IPv4的区别

1. 地址空间扩大，由位变为位。
2. 将校验和字段彻底移除，减少每跳处理时间。
3. 将可选字段移出首部，首部长度固定，变成扩展首部（丢弃了首部长度字段），更加灵活。
4. 路由器一般不对扩展首部检查，提高处理效率。
5. 支持即插即用，不需要协议。
6. 首部是字节的整数倍，而首部是字节的整数倍。
7. 只能在主机处分片，而能在主机和路由器处分片，所以不允许分片。（如过大在路由器会无法分片而无法传输的话，会用返回差错报文）
8. 支持资源预分配，支持实时视像等要求，保证一定的带宽和时延的应用。
9. 取消了协议字段。
10. 取消了总长度字段，改成有效载荷字段。
11. 取消服务类型字段。

#### IPv6地址表示

一般形式：冒号十六进制记法。（分为八个十六进制端）

压缩形式：一段中有连续的，可用一个代替；把在前面的全部删去；使用零压缩的方式，一串连续的可用用一对冒号代替，但是只能在一个地址中使用一次。

#### IPv6基本地址类型

| 名称 | 作用 | 要求 |
| --- | --- | --- |
| 单播 | 一对一通信 | 可做源地址，目的地址 |
| 多播 | 一对多通信，以前的广播地址当做覆盖所有主机的多播 | 可做目的地址 |
| 任播 | 一对多当中的一个通信，看似一对多，实则一对一，一般是最近的主机 | 可做目的地址 |

#### IPv4和IPv6之间的过渡

* 双栈协议：在一台设备上同时启用协议栈和协议栈。这样的话，这台设备既能和网络通信，又能和网络通信。如果这台设备是一个路由器，那么这台路由器的不同接口上，分别配置了地址和地址，并很可能分别连接了网络和网络。如果这台设备是一个计算机，那么它将同时拥有地址和地址，并具备同时处理这两个协议地址的功能。
* 隧道技术：通过使用互联网络的基础设施在网络之间传递数据的方式。使用隧道传递的数据（或负载）可以是不同协议的数据帧或包。隧道协议将其它协议的数据帧或包重新封装然后通过隧道发送。

## ARP协议

在实际网络的链路上传输数据帧时，都必须使用地址，所以协议就是完成主机或路由器地址到地址的映射，从而决定下一跳的走向，维护表。

由于局域网的数据链路层只能看见帧，所以路由器转发时会不断被解封和重新封装。数据帧的地址不变，但是源与目的地址会在不同的路由器中发生变化。（对于网桥则不改变帧的源地址）这决定了无法用地址来跨网络通信。

### ARP协议使用流程

1. 检查在主机或路由器上的高速缓存，如果有对应项就写入帧。
2. 没有对应项，就用作为目的地址，并广播请求分组，同一个局域网内的所有主机都能收到该请求。
3. 目的主机收到请求后会向源主机单播一个响应分组，源主机收到后将该映射写入缓存中。（更新）
4. 如果源主机将源地址与目的地址与操作，发现目的主机不在同一个局域网，就查询本局域网的默认网关地址并填入目的地址，传送到其他局域网。

### ARP协议的典型情况

1. 主机发送给本网络的主机：用找到主机的硬件地址。
2. 主机发送给另一网络的主机：用找到本网络上一个路由器（网关）的硬件地址。
3. 路由器发送给本网络的主机：用找到主机的硬件地址。
4. 路由器发送给另一网络的主机：用找到本网络上一个路由器（网关）的硬件地址。

## ICMP协议

网际控制报文协议支持差错报告与网络探寻。主要通过发送特定报文来实现。

协议是网络层的协议，报文是作为报文的一部分而传输的。

![ICMP帧格式](data:image/png;base64;base64,)

ICMP帧格式

### ICMP报文类型

报文类型的值占四个字节。

#### ICMP差错报文

| 类型 | 值 | 说明 |
| --- | --- | --- |
| 终点不可达 | 3 | 当路由器或主机不能交付数据报时就向源点发送终点不可达报文 |
| 源点抑制 | 4 | 当路由器或主机由于拥塞而丢弃数据报时，就向源点发送源点抑制报文，使源点知道应当把数据报的发送速率放慢 |
| 改变路由(重定向) | 5 | 路由器把改变路由报文发送给主机，让主机知道下次应将数据报发送给另外的路由器(可通过更好的路由) |
| 时间超过 | 11 | 当路由器收到生存时间TTL=0的数据报时，除丢弃该数据报外，还要向源点发送时间超过报文。当终点在预先规定的时间内不能收到一个数据报的全部数据报片时，就把已收到的数据报片都丢弃，并向源点发送时间超过报文 |
| 参数问题 | 12 | 当路由器或目的主机收到的数据报的首部中有的字段的值不正确时，就丢弃该数据报，并向源点发送参数问题报文 |

![ICMP差错报文数据字段](data:image/png;base64;base64,)

ICMP差错报文数据字段

不应发送差错报文的情况：

1. 对差错报告报文（即使差错报文本身出错也不会管）。
2. 对第一个分片的数据报片之后的所有后续数据报片。
3. 对具有组播地址的数据报。
4. 对具有特殊地址（如等）的数据报。

#### ICMP询问报文

| 类型 | 值 | 说明 |
| --- | --- | --- |
| 回送请求和回答报文 | 8或0 | 主机或路由器向特定目的主机发出的询问，收到此报文的主机必须给源主机或路由器发送ICMP回送回答报文。测试目的站是否可达以及了解其相关状态 |
| 时间戳请求和回答报文 | 13或14 | 请某个主机或路由器回答当前的日期和时间。用来进行时钟同步和测量时间 |
| 地址掩码请求和回答报文 |  |  |
| 路由器询问和通告报文 |  |  |

### ICMP应用

1. ：测试两个主机之间的连通性，使用了回送请求和回答报文。工作在应用层，直接使用网络层，而未使用或。
2. ：跟踪一个分组从源点到终点，使用了时间超过差错报告报文。工作在网络层。

## 路由算法与协议

路由算法就是让路由知道收到报文之后下一步怎么走。

路由器分组是通过路由表来转发，而路由表由路由算法给出。

* 路由选择分为直接交付和间接交付。
* 当发送站与目的站在同一网段（子网号相同）内时，就使用直接交付。直接交付不涉及到路由器。
* 反之使用间接交付，间接交付最后一个路由器肯定是直接交付。

### 路由算法与协议概述

* 静态路由算法（非自适应路由算法）：
  + 管理员手工配置路由信息。
  + 简便、可靠，在负荷稳定，拓扑变化不大的网络中较好。
  + 适用于高度安全的军事网络与较小的商业网络。
  + 路由更新慢，不适合用于大型网络。
* 动态路由算法（自适应路由算法）：
  + 路由器之间交换信息，按照路由算法优化出路由表。
  + 路由更新快，适用大型网络，及时响应链路反应与网络拓扑变化。
  + 算法复杂，加重网络负担。
  + 全局性：所有路由器掌握完整的网络拓扑和链路费用信息；如链路状态路由算法。
  + 分散性：路由器只掌握物理相连的相邻路由器与链路费用；距离向量路由算法。

### RIP协议

* 路由信息协议是一种分布式的基于距离向量的路由选择协议，最大的优点是简单。
* 要求网络中的每一个路由器都维护从它到其他每一个目的网络的唯一最佳路径（即一组距离）。
* 距离通常指跳数，即从源端口到目的端口所经过的路由器个数，经过一个路由器跳数，与路由器直接连接的网络距离为。允许一条路最多只能包含个路由器，所以表示网络不可达。
* 协议只适用于小型互联网。
* 路由表由目的网络、距离、下一跳路由器三个项目构成。
* 协议是应用层协议，使用传送数据。
* 一个报文最多只包含个路由，如果超过必须再用多的报文传送。

#### RIP协议交换信息

1. 仅和相邻路由器交换信息。
2. 路由器交换的信息是自己的路由表，其中包括每条路径的目的地与路径代价两条信息。
3. 每就交换一次，如果没收到相邻路由器的通告就默认这个路由器出现错误，并更新路由表。
4. 最开始路由器只知道直接相连的网络，若干次更新后所有路由器都会知道到达本自治系统中任何网络的信息，即收敛。

#### 距离向量算法

1. 修改相邻路由器发来的报文中的所有表项。对地址为的相邻路由器发来的报文，修改此报文的所有项目：把下一跳字段中的地址改为，并把所有的距离。
2. 对修改后的报文中的每一个项目都进行以下步骤：
   1. 本路由器路由表中没有该网络，则把这个项目直接填入路由表中。
   2. 本路由器路由表中已经有该网络，则查看下一跳路由器地址。
   3. 如果下一跳的路由器名相同，则表明距离更新了，用收到的项目替换原有项目。
   4. 如果下一跳的路由器名不同，原来距离更远则更新。
   5. 否则不处理。
3. 若超时，则把该路由器设为不可达，距离设为。
4. 返回。

#### RIP协议问题

协议存在问题：

* 限制了网络规模。
* 路由器交换完整的路由表，所以网络规模越大，开销越大。
* 当网络出现故障时，要经过较长的时间才能将此信息传送到所有的路由器，即出现路由回路问题，称为**慢收敛**。也叫做好消息传得快，坏消息传得慢。

当一个网络故障时，由于坏消息传输的可能较慢，导致故障信息晚于其相邻信息传输的原来正确信息，故障信息被其原来信息覆盖，从而让双方路由器都不断认为通过对方就能达到已经坏掉的网络，直到重复了次才会发现该网络不可达。

解决慢收敛方法：

* 限制路径最大距离：超过次就认为不可达不再传输。
* 分割水平线：路由器从某个接口接收到的更新信息不允许再从这个接口发回去。
* 抑制计时器法：等待足够多的时间（一般秒），已确保所有的机器都收到坏消息，而不会错误的连接过时的报文。需要指出的是，参与的机器都要遵循抑制策略，否则任依旧会发生环路。缺点是：如果发生环路，在抑制期内这些路由环路任然会维持下去。
* 毒性逆转：路由中毒是指在路由信息在路由表中失效时，先将度量值变为无穷大的数，而不是马上从路由表中删掉这条路由信息，然后再将其中毒的路由信息发布出去，这样相邻的路由器收到该中毒路由就可以通过其度量值就得知这条路由的度量值是，说明该路由是无效的。然后收到中毒路由信息的相邻的路由器会发送一个毒性逆转的信息，表示已经收到中毒路由信息。
* 触发刷新：正常情况下，路由器会基于计时器每将路由表发送给邻居路由器，而触发更新就是立刻发送路由更新信息。也就是说检测到网络故障的路由器会直接发送一个更新信息给邻居路由器，并依次产生触发更新通知它们的邻居路由器，此过程就叫触发更新。触发更新这种方式使整个网络上的路由器在最短的时间内收到更新信息。

### OSPF协议

* 开放最短路径优先协议的开放指协议是开放的，是指最短路径算法。
* 最主要的特征是使用分布式的链路状态协议。
* 协议是网络层（或传输层）协议，使用数据报传输。
* 对不同的链路可根据分组的不同服务类型（）而设置成不同的代价。因此，对于不同类型的业务可计算出不同的路由，十分灵活。
* 如果到同一个目的网络有多条相同代价的路径，那么可以将通信量分配给这几条路径。这称为**多路径间的负载平衡**。
* 所有在路由器之间交换的分组都具有鉴别功能，因而保证了仅在可信赖的路由器之间交换链路状态信息。
* 支持可变长度的子网划分和无分类编址。
* 每个链路状态都带上一个位的序号，序号越大，状态就越新。

#### 与RIP协议的区别

* 向本自治系统中的所有路由器发送信息，这里使用的方法是洪泛法。而仅向自己相邻的几个路由器发送信息。
* 发送的信息是与本路由器相邻的所有路由器的链路状态，但这只是路由器所知道的部分信息。“链路状态”说明本路由器和哪些路由器相邻及该链路的“度量”或代价。而在中，发送的信息是本路由器所知道的全部信息，即整个路由表。
* 只有当链路状态发生变化时，路由器才用洪泛法向所有路由器发送此信息，并且更新过程收敛得快，不会出现“坏消息传得慢”的问题。而在中，不管网络拓扑是否发生变化，路由器之间都会定期交换路由表的信息。
* 是网络层协议，它不使用或，而直接用数据报传送（其数据报首部的协议字段为）。而是应用层协议，它在传输层使用。

#### OSPF协议交换信息

1. 使用洪泛法向自治系统内所有路由器发送信息，即路由器通过输出端口向所有相邻的路由器发送信息，而每一个相邻路由器又再次将此信息发往其所有的相邻路由器。最终整个区域内所有路由器都得到了这个信息的一个副本。
2. 发送的信息就是与本路由器相邻的所有路由器的链路状态（本路由器和哪些路由器相邻，以及该链路的度量/代价——费用、距离、时延、带宽等）。
3. 路由器根据全网拓扑结构图使用最短路径算法计算自己到各目的网络的最优路径，构建新的路由表。（只存储下一跳）
4. 只有当链路状态发生变化时，路由器才向所有路由器洪泛发送此信息。
5. 最后，所有路由器都能建立一个链路状态数据库，即全网拓扑图。

#### 链路状态路由算法

涉及到的五种分组类型：

1. 每个路由器发现它的邻居结点**HELLO问候分组**（用于发现和维持邻站的可达性），并了解邻居节点的网络地址。
2. 设置到它的每个邻居的成本度量。
3. 构造**DD数据库描述分组**，向邻站给出自己的链路状态数据库中的所有链路状态项目的摘要信息。
4. 如果分组中的摘要自己都有，则邻站不做处理；如果有没有的或者是更新的，则发送**LSR链路状态请求分组**请求自己没有的和比自己更新的信息。
5. 收到邻站的分组后，发送**LSU链路状态更新分组**，使用洪泛法对全网链路进行更新。
6. 更新完毕后，邻站返回一个**LSAck链路状态确认分组**进行确认。
7. 根据算法与自己的链路状态数据库构造到其他结点之间的最短路径。
8. 用**HELLO问候分组**分组保持与邻居的连接。

当一个路由器的链路状态发生变化时，就会重复及其之后的步骤。

通常隔秒相邻两个路由器就要交换一次问候分组。且还要每隔一段时间如分粥就要刷新依次数据库中的链路状态。

#### OSPF协议的区域

为了使能够用于规模很大的网络，将一个自治系统再划分为若干个更小的范围，叫做**区域**。每一个区域都有一个位的区域标识符（用点分十进制表示）。

区域也不能太大，在一个区域内的路由器最好不超过个。

一个区域内的路由器只知道本区域内的完整网络拓扑。

处在上层的区域位主干区域，负责连通其他下层的区域和其他自治域。

路由器分为：

* 主干路由器（可以兼任）。
* 区域内部路由器。
* 区域边界路由器。
* 自治系统边界路由器。

### BGP协议

边界网关协议是外部网关协议，是应用层协议，是自治系统之间使用的，最新的版本是。

#### 层次路由

当网络规模扩大时，路由器的路由表成比例地增大。这不仅会消耗越来越多的路由器缓冲区空间，而且需要用更多时间来扫描路由表，用更多的带宽来交换路由状态信息。因此路由选择必须按照层次的方式进行。

且由于因特网规模大，而许多单位不想让外界知道自己的路由选择协议但是仍想接入因特网，所以就出现了自治系统。

自治系统就是多个使用同一种内部选择路由的协议的路由器之间构成的单独的小圈子，圈子内使用自己的协议，圈子和圈子之间也需要另一种协议。

路由选择协议分为内部网关协议（、）与外部网关协议（）。

使用层次路由时，将一个自治系统再划分为若干区域（），每个路由器都知道在本区域内如何把分组路由到目的地的细节，但不用知道其他区域的内部结构。

采用分层次划分区域的方法虽然会使交换信息的种类增多，但也会使协议更加复杂。但这样做却能使每个区域内部交换路由信息的通信量大大减小，因而使协议能够用于规模很大的自治系统中。

#### BGP协议工作原理

发言人是与之间交流信息的路由器，往往是边界路由器。

1. 每个自治系统的管理员至少选择一个路由器作为该自治系统的“发言人”。
2. 一个发言人与其他自治系统中的发言人要交换路由信息，就要先建立连接，即通过传送，然后在此连接上交换报文以建立会话（），利用会话交换路由信息。
3. 发言人之间互相交换了网络可达性的信息后，各发言人就根据所采用的策略从收到的路由信息中找出到达各的较好路由。
4. 网络可达性信息是指到达某个网络所经过的路径（一整条），且这个路径不一定是最佳的，而是选择能到达且网络较好的。

#### BGP协议特点

* 每个发言人除必须运行外，还必须运行该所用的内部网关协议，如或。
* 所交换的网络可达性信息就是要到达某个网络（用网络前缀表示）所要经过的一系列路径。
* 协议书寻找一条较好的**路由**，而非找一条最优路由。因为因特网规模太大，选择困难，且之间选择最优路由不现实，还必须考虑其他相关策略。
* 交换路由信息的结点数量级是自治系统的数量级，要比这些自治系统中的网络数少很多。
* 每个自治系统中发言人（或边界路由器）的数目是很少的。这样就使得自治系统之间的路由选择不致过分复杂。
* 支持，因此的路由表也就应当包括目的网络前缀、下一跳路由器，以及到达该目的网络所要经过的各个自治系统序列。
* 在刚刚运行时，的邻站是交换整个的路由表。但以后只需要在发生变化时更新有变化的部分。这样做对节省网络带宽和减少路由器的处理开销都有好处。

#### BGP-4报文类型

1. （打开）报文：用来与相邻的另一个发言人建立关系，并认证发送方。
2. （更新）报文：通告新路径或撤销原路径。
3. （保活）报文：在无时，周期性证实邻站的连通性；也作为的确认。
4. （通知）报文：报告先前报文的差错；关闭连接。

### 三种协议的比较

| 协议 | RIP | OSPF | BGP |
| --- | --- | --- | --- |
| 类型 | 内部 | 内部 | 外部 |
| 路由算法 | 距离-向量 | 链路状态 | 路径-向量 |
| 传递协议 | UDP | IP | TCP |
| 层次 | 应用层 | 网络层 | 应用层 |
| 路径选择 | 跳数最少 | 代价最低 | 较好，非最佳 |
| 交换结点 | 和本结点相邻的路由器 | 网络中的所有路由器 | 和本结点相邻的路由器 |
| 交换内容 | 当前本路由器知道的全部信息，即自己的路由表 | 与本路由器相邻的所有路由器的链路状态 | 首次是整个路由表，非首次是有变化的部分 |

* 是一种分布式的基于距离向量的路由选择协议，它通过广播报文来交换路由信息。
* 是一个内部网关协议，要交换的信息量较大，应使报文的长度尽量短，所以不使用传输层协议（如或），而直接采用。
* 是一个外部网关协议，在不同的自治系统之间交换路由信息，由于网络环境复杂，需要保证可靠传输，所以采用。

## IP多播

多播也称为组播。所以肯定仅用于。

当网络中某些用户需要特定数据时，组播数据发送者仅发送一次数据，避免路由环路，借助组播路由协议为组播数据包建立**组播转发树**，被传输的数据到达距离用户端尽可能近的结点后才开始复制和分发。

组播提高了数据传输效率，降低拥塞的可能性，组播组中的主机可以时同一个物理网络，如果有组播路由器也可以来自不同物理网络。

能运行组播协议的路由器就是组播路由器。

### 组播实现

因特网的组播是靠路由器来实现的，这些路由器必须增加一些能够识别组播的软件。能够运行组播协议的路由器可以是一个单独的路由器，也可以是运行组播软件的普通路由器。因特网上的组播比以太网上的组播复杂得多，因为以太网本身支持广播和组播，而因特网上当前的路由器和许多物理网络都不支持广播和组播。

多个单播可以仿真组播，此时仿真的实验基本上是相同的，但是一个组播所需要的带宽小于多个单播带宽之和，此时路由器的时延变大，而处理一个组播分组的时延是比较小的。

### IP组播地址

组播地址能让源设备能将分组发送给一组设备，属于多播组的设备将被分配一个组播组地址。

组播地址范围是到，一个类地址表示一个组播组，只能用作目的地址，源地址必然是单播地址。

1. 组播数据报由于是组播所以不是一对一，所以无法建立连接，仅能应用于，只能尽最大努力交付，不提供可靠交付，。
2. 对组播数据报不产生差错报文。
3. 并非所有类地址都可以作为组播地址。

### 硬件组播

同单播地址一样，组播地址也需要对应的组播地址在本地网络中实际传输帧。组播地址以十六进制值开头，即范围为，余下的六个十六进制位根据组播组地址的最后位转换而成，第位为。

![硬件组播](data:image/png;base64;base64,)

硬件组播

有时候可能会地址映射成同样的地址，从而数据组播时可能会出错，所以在收到多播数据报的主机，还要在层利用软件进行过滤，将不是本主机要接受的数据丢弃。

### IGMP协议

网际组管理协议让连接在本地局域网上的多播路由器知道本局域网上是否还有主机的某个进程参加或退出了某个多播组。

首部中协议字段值为表示使用协议。

#### 工作阶段

1. 某主机要加入组播组时，该主机向组播组的组播地址发送一个报文，声明自己要称为该组的成员。本地组播路由器收到报文后，要利用组播路由选择协议把这组成员关系发给因特网上的其他组播路由器。
2. 本地组播路由器周期性探询本地局域网上的主机，以便知道这些主机是否还是组播组的成员。
3. 只要有一个主机对某个组响应，那么组播路由器就认为这个组是活跃的。
4. 如果经过几次探询后没有一个主机响应，组播路由器就认为本网络上的没有此组播组的主机，因此就不再把这组的成员关系发给其他的组播路由器。组播路由器知道的成员关系只是所连接的局域网中有无组播组的成员。

### 组播路由选择协议

组播路由协议目的是找出以源主机为根节点的组播转发树。

对不同的多播组对应不同的多播转发树；同一个多播组，对不同的源点也会也不同的多播转发树。

#### 组播数据报方法

1. 洪泛与剪除。
2. 隧道技术。
3. 基于核心的发现技术。

#### 建议组播路由选择协议

1. 距离向量多播路由选择协议。
2. 基于核心的转发树。
3. 开发最短通路优先的多播扩展。
4. 协议无关多播-稀疏方式。
5. 协议无关多播-密集方式。

## MPLS

多协议标记交换利用面向连接的技术，使每个分组都携带一个叫做标记的小整数，当分组到达交换机时，交换机读取分组的标记，并用标记值来检索分组转发表。

具有以下特点：

1. 支持面向连接的服务质量。
2. 支持流量工程，平衡网络负载。
3. 有效地支持虚拟专用网。

## 移动IP

移动技术是移动结点（计算机/服务器等）以固定的网络地址，实现跨越不同网段的漫游功能，并保证了基于网络的网络权限在漫游过程中不发生任何改变。移动结点具有永久地址的移动设备。

### 相关术语

* 归属代理（本地代理）：一个移动结点拥有的就“居所”称为归属网络，在归属网络中代表移动节点执行，移动管理功能的实体叫做归属代理。
* 外部代理（外地代理）：在外部网络中帮助移动节点完成移动管理功能的实体称为外部代理。
* 永久地址（归属地址/主地址）：移动站点在归属网络中的原始地址。
* 转交地址（辅地址）：移动站点在外部网络使用的临时地址。

### 通信过程

移动的基本工作过程可以分为代理发现、注册、分组路由与注销四个阶段

刚进入外部网络：

1. 在外部代理登记获得一个转交地址，离开时注销。
2. 外地代理向本地代理登记转交地址。

给发送数据报：

1. 本地代理截获数据报。
2. 本地代理再封装数据报，新的数据报目的地址是转交地址，发给外部代理（隧道）。
3. 外部代理拆封数据报并发给。

给发送数据报：

用自己的主地址作为数据报源地址，用的地址作为数据报的目的地址。

移动到了下一个网络：

1. 在新外部代理登记注册一个转交地址。
2. 新外部代理给本地代理发送新的转交地址（覆盖旧的）。
3. 通信。

回到了归属网络:

1. 向本地代理注销转交地址。
2. 按原始方式通信。

移动为移动主机设置了两个地址，即主地址和辅地址（转交地址）。移动主机在本地网时，使用的是主地址。当移动到另一个网络时，需要获得一个临时的辅地址，但此时主地址仍然不变。从外网移回本地网时，辅地址改变或撤销，而主地址仍然保持不变。

## 互联网连接问题

计算机网络中主机不仅需要跟其他主机相联，还需要跟互联网相联。

* 当询问该网段的主机是否可以访问互联网时，要查看对应网段的默认网关地址，此时默认网关的地址必须跟连接的路由器的接口地址一致，那么可以访问互联网。
* 当主机处于不同的网段（子网号不同）则需要路由器进行连接（就是互联网连接问题，因为不同网段主机访问即使再近都要访问互联网再找到该地址）。此时默认网关的地址必须跟连接的路由器的接口地址一致，否则不能通过路由器转发，从而无法连接。
* 当一个路由表中要填写访问互联网的路径时，那么这个路径就是固定的，掩码也是。这样的话只要这个地址不与路由表中的其他所有地址匹配，那么他们都可以选择这个最终的方案通过互联网查找地址。
* 如果是要到固定的设备，如域名服务器，则必须地址全部匹配，此时子网掩码就是。

## 网络层设备

一般为路由器，是一种具有多个输入端口和多个输出端口的专用计算机，其任务是根据转发表分组转发。

若收到分组等，则把分组送往路由选择处理机；若收到数据分组，则查找转发表并输出。

功能为分组转发和路由计算，不进行差错检测。

路由器作用于物理层、数据链路层、网络层。

### 输入端口处理

1. 从线路接受分组。
2. 物理层进行处理。
3. 数据链路层进行处理。
4. 网络层进行处理，首先对数据进行分组与排队，再进行查表与转发。
5. 输出到交换结构中。

输入端口的查找和转发功能在路由器的交换功能中最重要。

### 交换结构

根据转发表对分组处理。

* 通过存储器进行交换。
* 通过总线进行交换。
* 通过互联网进行交换。

交换结构由路由选择处理机、路由选择协议、路由表组成。

### 输出端口处理

1. 交换结构中收到分组。
2. 网络层进行处理，首先对数据进行分组与排队，速度太快需要放在缓存中并进行缓存处理。
3. 数据链路层进行处理。
4. 物理层进行处理。
5. 向线路发送分组。

若路由器处理分组的速率赶不上分组进入队列的速率，则队列的存储空间必然最终降为，使后面再进入队列的分组由于没有存储而被丢弃。

路由器中的输入或输出队列产生溢出是造成分组丢失的主要原因。

### 三层设备的对比

* 路由器可以互联两个不同网络层协议的网段。面向协议，根据不同协议进行不同操作。
* 网桥可以互联两个物理层和链路层不同的网段。与高层协议无关
* 集线器不能互联两个物理层不同的网段。

### 路由表与路由转发

路由表由路由选择算法得到，主要用于路由选择，总由软件实现。包括四个部分：目的网络地址、子网掩码、下一跳地址、接口。

其中路由表都有一个默认路由：，其子网掩码为，当不知道发送给谁时就发送这个默认路由，让别的路由器帮忙处理。

转发表由路由表而来，可以使用软件实现也可以使用特殊的硬件实现。转发表必须包含完成转发功能所必须的信息，在每一行都包含要达到的目的网络到输出端口和某些地址信息的映射。

一般默认网关地址就是路由器的端口地址。

# 传输层

## 基本概念

传输层只有主机才有，而路由器这种中间设备至多只有物理层、数据链路层和网络层三层架构。

是面向通信的最高层，也是用户功能的最底层。

### 传输层的功能

1. 传输层提供进程与进程之间的逻辑通信。
2. 复用与分用。（复用：应用层所有的应用进程都可以通过传输层再传输到网络层；分用：传输层从网络层收到数据后交付指明的应用进程）
3. 差错检测。

### 寻址与端口

* 端口（逻辑端口/软件端口）：是传输层的，标识主机中的应用进程。
* 每一个端口有用于区分的端口号，只有本地意义，因特网中不同主机的相同端口号无联系。
* 端口号长度为，能标识个不同的端口号。
* 端口号按范围划分可以分为：
  + 服务端使用的端口号：
    - 熟知端口号（）：给最重要的一些应用程序，让所有用户都知道。
    - 登记端口号（）：为没有熟知端口号的应用程序使用。
  + 客户端使用的端口号：仅在客户进程运行时才系统动态分配。
* 套接字=(主机地址,端口号)。唯一标识了网络中的一个主机和它上面的一个进程。如果有新的同样套接字的连接请求建立，则建立失败，不影响原有连接。

常用的端口号：

| 应用程序 | 熟知端口号 |
| --- | --- |
| FTP数据 | 20 |
| FTP控制 | 21 |
| TELNET | 23 |
| SMTP | 25 |
| DNS | 53 |
| TFTP | 69 |
| HTTP | 80 |
| POP3 | 110 |
| SNMP | 161 |
| HTTPS | 443 |

## UDP协议

用户数据报协议只在数据报服务之上添加了两个功能，即只有复用分用（接收方的传输层剥去报文首部后，能把这些数据正确交付到目的进程。通过目的端口实现）与差错检测功能。

### 主要特点

1. 无连接，减少开销和发送数据之前的时延。
2. 使用最大努力交付，而非保证可靠交付。所以不会对报文编号。
3. 面向报文（不对报文拆分，应用层给多长报文，就会照样一次发送一个完整报文），适合一次性传输少量数据的网络应用。
4. 无拥塞控制，适合很多实时应用，实时应用延迟要求高，需要立即响应。
5. 支持一对一、一对多、多对一、多对多的交互通信。
6. 首部开销小，，而需要。

因为不保证可靠性，所以其可靠性由应用层完成。

### UDP数据报格式

![UDP数据报格式](data:image/png;base64;base64,)

UDP数据报格式

* 源端口号：需要对方回应时选用，如果不需要回应，可以不填，即是全的。
* 目的端口号：是必填的。分用时，如果找不到对应的目的端口号就丢弃该报文，并向发送方发送端口不可达差错报告报文。
* 长度：整个用户数据报的长度，首部加上数据部分。最小值为。以字节为单位。不包括伪首部。
* 检验和：检测整个数据报是否有错（伪首部+首部+数据，而只检测首部），错就丢弃。若不想校验则是全。若计算结果为全则置为全。

### UDP协议校验和

具体的数据报格式如下：

![UDP具体格式](data:image/png;base64;base64,)

UDP具体格式

* 伪首部只有在计算检验和时才出现，不向下传达也不向上提交。
* 其中的代表封装报文的数据报首部协议字段是。
* 长度是首部加上数据长度，不包括伪首部。

发送端：

如果不使用校验和字段，则字段全部填充。

1. 填上的伪首部。
2. 全填充检验和字段。
3. 数据报要看成许多位的字符串连接一起，全填充数据部分末尾，使数据报成为偶数个字节。
4. 伪首部+首部+数据部分采用二进制反码求和。
5. 二进制反码运算求和再取反填入校验和字段。
6. 去掉伪首部进行发送。

接收端：

如果校验和字段计算结果恰好为，则表示错误，字段全部填充。

1. 填上伪首部，若不是偶数个字节还要在末尾加。
2. 伪首部+首部+数据部分采用二进制反码求和。
3. 如果结果全为则无差错，否则出错，丢弃或交给应用层并附上出错的警告。

![UDP校验](data:image/png;base64;base64,)

UDP校验

## TCP协议

传输控制协议对比用户数据报协议更复杂。

### TCP协议主要特点

1. 是面向连接（虚连接）的传输层协议。
2. 每一条连接只能有两个端点，所以连接是一对一的。
3. 提供可靠有序的服务，无差错不重复（使用确认机制）。
4. 提供全双工通信，包含发送缓存（准备发送的数据与已放送但是未确认的数据）与接受缓存（按序到达但是未被读取的数据与不按序到达的数据）。
5. 面向字节流。把应用程序交下来的数据看成一连串的无结构字节流。

### TCP数据报格式

传输的数据单位称为报文段。可以用来传输数据，也可以用来建立连接、释放连接、应答。长度为整数倍，默认最短为，报头最长为。

![TCP报文段格式](data:image/png;base64;base64,)

TCP报文段格式

* 源端口和目的端口：各。
* 序号：在一个连接中传送的字节流中的每一个字节都按顺序编号，本字段表示本报文段所发送数据的第一个字节的序号。范围为。
* 确认号：期望收到对方下一个报文段的第一个数据字节的序号。若确认号为，则证明到序号为止的所有数据都已正确收到。
* 数据偏移（首部长度）：报文段的数据起始处距离报文段的起始处有多远，即报头的长度。以位单位，即个数值是，最大值为，达到首部的最大值。
* 保留：目前为。

还有六个控制位，除了和位都较重要：

1. 紧急位：时， 标明此报文段中有紧急数据，是高优先级的数据，应尽快传送，不用在缓存里排队。紧急数据都在数据报最前面，配合紧急指针字段使用，即数据从第一个字节到紧急指针所指字节之间的数据就是紧急数据。
2. 确认位：时确认号有效，在连接建立后所有传送的报文段都必须把置为。
3. 推送位：时，接收方尽快交付接收应用进程，不再等到缓存填满再向上交付。如果没有，一般都是接收方缓存满了之后再将数据发送到主机。
4. 复位：时， 表明连接中出现严重差错，必须释放连接，然后再重新建立传输链接。
5. 同步位：时，表明是一个连接请求/连接接受报文。此时若代表这是一个连接请求报文，若代表这是一个连接接收报文。
6. 终止位：时，表明此报文段文送方数据已发完，要求释放连接。

* 窗口：指的是发送本报文段的一方的接收窗口，即现在允许对方还可以发送的数据量，防止对方发送过多数据导致自己无法接受。占，范围。
* 检验和：检验首部+数据，检验时要加上伪首部，第四个的协议字段由改为。
* 紧急指针： 时才有意义，指出本报文段中数据部分的紧急数据的字节数。
* 选项：最大报文段长度、窗口扩大、时间戳、选择确认……
* 填充：当首部长度不为的整数倍就由填充部分填充，到字节的整数倍。

### TCP协议连接管理

TCP建立连接采用客户服务器方式。但是实际上任何一台计算机都可能做服务器也可能做客户端。

#### TCP三次握手（建立连接）

![TCP建立连接](data:image/png;base64;base64,)

TCP建立连接

1. 最开始客户端与服务端都是关闭状态。
2. 服务器端创建传输控制块，进入收听状态准备接受连接请求。
3. 客户端创建，发送请求连接报文段，无应用层数据。然后客户端进入同步已发送状态。
4. 服务端接受报文段后进入同步收到状态，服务器端为该连接分配缓存和变量，并向客户端返回确认报文段，允许连接，无应用层数据。
5. 客户端接受报文后变成已建立连接状态，为该连接分配缓存和变量，并向服务器端返回确认的确认，可以携带数据。
6. 服务端接受到报文段后变成已建立连接状态。

注释：其中表示序号，指本报文的随机编号；表示确认号，指期待对方发送的报文的第一个序号。

若不指出为，则代表其值为。

1. 第一部分：
   * ：主机要建立连接了。
   * （随机）：后面没有数据。
2. 第二部分：
   * ：主机同意主机建立连接。
   * ：连接确认建立了，之后的必须都置为，表示开始同步。
   * （随机）：设置初始序号，后面没有数据。
   * ：表示期待对方放松的报文段的第一个字节，之前发送方说发送的是第位数据（虽然发送方是任意给出的），所以主机要的是位数据。
3. 第三部分：
   * ：连接建立了，之后的必须都置为。
   * ：主机发送的报文段的第一个字节就是。
   * ：之前接收方发送的是第位数据（虽然接收方是任意给出的），所以主机要的是位数，对其确认。

值得注意的是的值是随机的，所以客户端的和服务器端的序列值可能相同

#### SYN洪泛攻击

由于三次握手时服务器的资源在第二次握手时分配，客户端自愿者第三次握手时分配，可能导致反复确认与占用，产生洪泛攻击。洪泛攻击发生在第四层。

这种方式利用协议的特性，就是三次握手。攻击者发送的包，是三次握手中的第一个数据包，即第一步，而当服务器返回后，该攻击者不对其进行再确认，那这个连接就处于挂起状态，也就是所谓的半连接状态，服务器收不到再确认的话，还会重复发送给攻击者。这样更加会浪费服务器的资源。攻击者对服务器发送非常大量的这种连接，由于每一个都没法完成三次握手，所以在服务器上，这些连接会因为挂起状态而消耗和内存，最后服务器可能死机，就无法为正常用户提供服务了。可以通过设置来解决。

#### TCP四次挥手（连接释放）

每一条连接的两个进程中的任何一个都能终止连接，连接结束后主机的资源将被释放。

![TCP释放连接](data:image/png;base64;base64,)

TCP释放连接

1. 最开始客户端与服务端都是已建立连接状态。
2. 客户端发送连接释放报文段，停止发送数据，主动关闭连接，进入终止等待状态。
3. 服务端会回送一个确认报文段，此时服务器端进入关闭等待状态，客户到服务器这个方向的连接就释放了——半关闭的状态，不允许客户端再发送数据给服务器。
4. 客户端接受报文段后进入终止等待状态。
5. 服务器发完数据，如果没有要向服务器发送的数据，就发出释放连接报文段，主动关闭连接，进入最后确认阶段。
6. 客户端回送一个确认报文字段，服务器端接收后进入关闭状态。客户端等到时间等待计时器设置的（最长报文段寿命）后彻底关闭连接，关闭服务器到客户这个方向，进入关闭状态。

注释：

1. 第一部分：
   * ：主机要释放连接。
   * （随机）：后面可以有数据也可以没有数据。
2. 第二部分：
   * ：连接建立了，之后的必须都置为。
   * （随机）：第一部分数据长度，如果第一部分的确认报文没有数据就是。
   * ：之前发送方发送的是第位数据，所以主机要的是位数据（尽管此时已经决定释放连接了）。
3. 第三部分：
   * ：主机要释放连接。
   * ：连接建立了，之后的必须都置为。
   * （随机）：第二部分数据长度，如果第二部分的确认报文没有数据就是。
   * ：之前发送方说发送的是第位数据，所以主机要的是位数据（因为A直接不发数据了，所以第二段第三段的都是）。
4. 第四部分：
   * ：连接建立了，之后的必须都置为。
   * ：之前发的数据时第位数据，也要第位数据，所以我发第位数据。
   * ：之前发送方说发送的是第位数据，所以主机要的是位数据。

为什么要等待时间？

1. 保证发送的最后一个报文段能发送到，否则服务器会接收不到确认的信息，而已经关闭无法重发确认报文段，从而无法正常关闭。
2. 防止已失效的连接请求报文段传输到下一次的连接请求，干扰下一次的连接服务。

### TCP协议可靠传输

传输层使用的是与的混合。

#### 校验

通过校验的方式来保证数据一致，其方式也是如校验一样增加伪首部与校验和。

#### 序号

报文传输时每个字节都会编上序号，一个字节占用一个序号，并按报文段的形式一起发送，报文段长度不定，根据来定。

序号字段指一个报文段第一个字节的序号。

序号建立在传送的字节流上，而不是报文段。

虽然面向字节，但是不是每个字节都要发回确认，而是在发送一个报文段后才发回一个确认，确认号为报文段第一个字节的序号，所以是对**报文段**的确认机制。

#### 确认

确认号是期望收到的下一个报文段数据的第一个字节的序号。

缓存中的字节流按序传输后不会立刻在缓存中清除，而会等待接收方的确认字段，可以是单独确认也可以携带确认。

一般采用的是累计确认的方式。收到确认字段后就可以从缓存中清除对应报文段。

#### 重传

一般有两种情况会产生重传：

超时：

* 每发送一个报文段就会设置一次计时器，在重传时间内未收到确认就需要重传已发送的报文段。
* 由于下层互联网环境复杂，每次路由选择可能变化从而带来时延方差也很大，所以采用自适应算法：
  + 记录报文段发出时间和收到响应确认时间，称其差为报文段的往返时间。
  + 根据的测量值动态改变重传时间（加权平均往返时间）。
  + 从而超时计时器设计的超时重传时间应该略大于。

新估计旧新样本。

冗余：

* 为了加快发现需要重传的报文段，可以采用冗余（冗余确认/快重传），每当比期望序号大的失序报文段到达时，发送一个冗余，指明下一个期待字节的序号。
* 如发送方已发送，，，，报文段，接收方收到会返回对的确认（确认号为的第一个字节），如果接收方收到、，都会返回对的确认，发送方收到个对于报文段的冗余就会认为报文段丢失，从而快速重传报文段。

### TCP协议流量控制

使用滑动窗口机制来完成流量控制，与数据链路层的滑动窗口类似。单位为字节。

在通信过程中，接收方会根据接收缓存的大小，动态调整发送方发送窗口的大小，即接收窗口（接受方设置确认报文段的窗口字段，将通知给发送方），发送方的发送窗口取接收窗口和拥塞窗口（根据当前网络拥塞程度而由发送发确定的窗口值，与网络带宽与时延相关）的最小值。

向发送数据，连接建立时，告诉：的，设每一个报文段，报文段序号初始值为。

![TCP流量控制](data:image/png;base64;base64,)

TCP流量控制

只有处理完接收窗口中的数据才能继续接收的数据，发送一个不为的报文。

而如果这个告诉接收窗口不为的报文丢失了，就一直会等待发送，就会一直等待接收，从而产生死锁般的情况。

为每一个连接设有一个持续计时器，只要连接的一方收到对方的零窗口通知（即的通知）就启动持续计时器。

如果计时器设置的时间到期，就会发送一个零窗口探测报文段，接收方收到探测报文段就会给出现在的窗口值。

如果窗口仍然是，那么发送方就重新设置持续计时器。

### TCP协议拥塞控制

出现拥塞条件：需求>可用资源。

当网络中有许多资源同时呈现供应不足时网络性能变坏，网络吞吐量将随输入负荷增大而下降。

拥塞控制：防止过多的数据注入网络中。与流量控制不同的是它是面向全局的，是因为网络堵塞。形象来说拥塞控制就是为了控制路上堵车，而流量控制就是降低发车率。单位为。

* 接收窗口指接收方能接收缓存设置的值，并告知给发送方，反映接收方容量。
* 拥塞窗口指发送方根据自己估算的网络拥塞程度而设置的窗口值，反映网络当前容量。

拥塞控制的假定：

1. 数据单方向发送，而另一个方向只发送确认，而不会捎带确认。
2. 接收方总是有足够大的缓存空间，因而发送窗口大小取决于拥塞程度。发送窗口=(接收窗口,拥塞窗口)。

#### 慢开始与拥塞避免

![慢开始与拥塞避免](data:image/png;base64;base64,)

慢开始与拥塞避免

初始值是，指一个最大报文段长度。

传输轮次指发送了一批报文段并收到其确认的时间，一般指一个往返时延。可能一次性传输多个报文。

1. 最开始是慢开始算法，一步步试探网络拥塞，开始时以的指数形式增长。
2. 的意思是慢开始门限，代表从这个地方注入的报文段就比较多了，需要开始慢速增加了。
3. 拥塞窗口超过慢开始门限后进行拥塞避免算法，之后一段都是线性增长，每次增加，直至达到网络拥塞状态。
4. 当网络开始拥塞时，进行乘法减小，瞬间将设置为，同时调整原来的的值到之前达到网络拥塞状态前值的，（这里是降到），但是不能小于。这样就能让拥塞的路由器能快速把队列中积压的分组处理完。
5. 重复以上步骤，但是注意此时变了之后线性增长的转折点也变了。所以最后拥塞窗口会波动逼近适合当前网络拥塞状态的窗口值。

注意：当慢开始进行指数增长时，当时，则一个后，不会让慢开始的拥塞窗口超过阈值。

#### 快重传与快恢复

快重传（冗余）在协议可靠传输中已经提到过。

![快重传与快恢复](data:image/png;base64;base64,)

快重传与快恢复

这里和上面的慢开始和拥塞避免的一开始步骤差不多，都是先指数增长再转变为线性增长。

不同的点是快重传和快恢复算法是在收到连续的确认之后执行，这里的就是冗余，冗余的特点是如果多次对某一段请求的数据没有被收到，达到一定数目，一般为三个冗余之后就会立即执行重传。

但是此时只是降到现在的一半，再重新线性增长。而不是像慢开始和拥塞避免的从头开始，这就是快恢复。

一般而言建立连接和网络超时时使用慢开始和拥塞避免算法；当发送方接收到冗余时使用快重传和快恢复算法。

# 应用层

应用层对应用程序的通信提供服务。

传输层功能：

1. 文件传输、访问和管理。
2. 电子邮件。
3. 虚拟终端。
4. 查询服务与远程作业登陆。

## 网络应用模型

### C/S模型

客户/服务器模型中服务器提供计算服务的设备，而客户指请求计算服务的主机。

服务器的特点：

1. 永久提供服务。
2. 永久访问地址/域名。

客户机的特点：

1. 与服务器通信，使用服务器提供的服务。
2. 间歇接入网络。
3. 可能使用动态地址。
4. 不能直接与其他客户机通信。

应用：、、电子邮件等。

### P2P模型

对等模型也可以看作模型的变形

对等模型的特点：

1. 不存在永远在线的服务器。
2. 每个主机既提供服务也可以提供服务。
3. 任意端系统或结点之间可以直接通信。
4. 结点间歇接入网络。
5. 结点可能改变地址。
6. 扩展性好。
7. 网络健壮性强。
8. 占用较多内存，影响速度。

## DNS

域名系统将域名转换为地址。域名方便记忆。基于客户/服务器模式，运行在上，使用号端口。

域名系统是一个联机分布式的数据库系统，并采用客户/服务器模式。

### 域名

域名从右到左级别降低，最右边的是顶级域名，如，就是顶级域名，就是二级域名，就是三级域名。其实后面应该有一个点，这个点就是根。

* 标号中的英文不区分大小写。
* 标号中除连字符-外不能使用其他的标点符号。
* 每个标号不超过个字符，多标号组成的完整域名最长不超过个字符。
* 级别最低的域名写在最左边，级别最高的顶级域名写在最右边。

顶级域名分为：

* 国家顶级域名：，等。
* 通用顶级域名：（商业组织），（信息服务提供机构），（非盈利机构），（政府），（国际组织），（航空企业），（博物馆），（旅游业）等。
* 基础结构域名/反向域名：，用于反向域名解析。

二级域名分为：

* 类别域名：（科研机构），（盈利机构），（教育机构），（信息服务提供机构），（非盈利机构），（政府），（国防机构）。
* 行政区域名：用于省市如等。
* 自主注册域名，要求全球唯一。

### 域名服务器

域名到地址的解析由运行在域名服务器上的程序完成。一个服务器管辖的范围称为区。

* 根域名服务器：
  + 知道所有顶级域名服务器域名与地址。
  + 若本地域名服务器不能解析首先交给根域名服务器。
  + 一共个根域名服务器，到。
  + 用来管辖顶级域，不直接进行地址转换，而是告诉本地域名服务器下一步应该找哪个顶级域名服务器进行查询。
* 顶级域名服务器：管理该顶级域名服务器注册的所有二级域名。
* 权限（授权）域名服务器：
  + 负责一个区（域）的域名服务器。
  + 每台主机都必须在权限域名服务器上登记。
  + 权限域名服务器也可以是本地域名服务器。
  + 可以将其管辖的主机名转换为地址。
* 本地域名服务器：当一个主机发出查询请求时，这个请求报文就发送给本地域名服务器，是最近的服务器。

### 域名解析过程

将域名映射为地址称为正向解析，把地址映射为域名称为反向解析。

需要请求时通过本机客户端构造一个请求报文，以数据报方式发送给本地域名服务器。

* 递归查询：本地域名服务器查找根域名服务器，根域名服务器查找顶级域名服务器，顶级域名服务器查找权限域名服务器，最后得到的结果再由权限域名服务器到顶级域名服务器到根域名服务器到本地域名服务器最后到主机。（靠别人）
* 迭代查询：本地域名服务器查找根域名服务器，本地域名服务器查找顶级域名服务器，本地域名服务器查找权限域名服务器。（靠自己）

![图示

描述已自动生成](data:image/png;base64;base64,)

一般使用递归查询结合迭代查询结合的方式。

1. 客户机向其本地域名服务器发出请求报文（递归查询）。
2. 本地域名服务器收到请求后，查询本地缓存，若有则直接得到，若没有该记录，则以客户的身份向根域名服务器发出解析请求报文（迭代查询）。
3. 根域名服务器收到请求后，判断该域名属于域，将对应的顶级域名服务器的地址返回给本地域名服务器。
4. 本地域名服务器向顶级域名服务器发出解析请求报文（迭代查询）。
5. 顶级域名服务器收到请求后，判断该域名属于域，因此将对应的授权域名服务器的地址返回给本地域名服务器。
6. 本地域名服务器向授权域名服务器发起解析请求报文（迭代查询)。
7. 授权域名服务器收到请求后，将查询结果返回给本地域名服务器。本地域名服务器将查询结果保存到本地缓存，同时返回给客户机。

为了解决查询速度过慢，高速缓存可以存储域名与地址值。

## 文件传送协议

分为文件传送协议和简单文件传送协议。

### FTP协议

默认数据传输端口，控制端口。这些端口都是服务器上的，客户端的端口由程序自己分配。

可以指明文件类型与格式，并可以给予存取权限。

* 提供不同种类主机系统之间的文件传输能力。
* 基于模型。用户通过一个客户机程序连接至远程计算机上运行的服务器程序。依照协议提供服务，进行文件传输的计算机就是服务器，连接服务器，遵循协议与服务器传送文件的电脑就是服务端。

#### FTP协议工作原理

* 登录（用户名+密码或匿名登录）。
* 使用协议提供可靠传输。
* 服务器进程：个主进程与个从属进程提供服务。

为什么有匿名登陆：对于一些公共服务器来说，增加验证阶段就是增加资源开销，减少验证阶段就可以节省资源来更好地服务，使用特殊用户名。

主进程和从属进程的区别：主进程是打开端口，让外部发送的数据可以进来，并且将这些数据逐个分配各从属进程。从属进程则是单独为这些数据服务。

![图示

描述已自动生成](data:image/png;base64;base64,)

#### FTP协议连接流程

* 客户端和服务器端先建立连接，端口是，称为控制连接。用来传输控制信息，以七位格式传输。
* 然后看情况是主动建立连接还是被动建立连接。
* 主动连接是指服务器端主动发送请求和客户端进行连接，此时端口号固定是。
* 被动连接是指客户端发送请求和服务器端建立数据传送连接，此时端口号是不确定，有两者协商得到，一般大于。
* 数据传输完成之后，数据连接断开，控制连接继续保持，直至两边发送断开请求。

数据连接与控制连接都属于从属进行，控制连接始终保持，数据连接只保持一会。

正是因为的控制信息是单独使用一个端口，所以称的控制信息是带外传送的。

#### FTP协议传输模式

* 文本模式：模式，以文本序列传输数据。
* 二进制模式：模式，以二进制序列传输数据。

### TFTP协议

协议了解即可。

#### 特点

* 简单文件传送协议端口号为。
* 每次传输的数据报文中包含字节，最后一次可不足字节。
* 数据报按序编号，从开始。
* 支持码或二进制传输。
* 可对文件进行读写。
* 首部简单。
* 使用模型。
* 使用数据报。

#### 优点

* 适用于只支持协议的环境，可以同时向许多机器下载。
* 占用的内存很小且很容易实现，对于小型计算机或特殊设备非常有利。

#### 缺点

* 需要自己实现差错改正。
* 只支持文件传输而不支持交互。
* 没有很大的命令集，无法列目录，也无法用户身份鉴定。

#### TFTP协议工作流程

* 客户进程发送一个读/写请求报文给服务器进程，端口号为。
* 服务器选择另一个端口和客户进程通信。
* 传输文件长度若正好是的整数倍，则传输完毕后还要发送一个只含有首部而无数据的报文。
* 传输文件长度若不是512的整数倍，则最后一个报文数据部分不满字节，正好作为文件结束标志。

## TELNET协议

远程终端协议（终端仿真协议）让用户使用连接登录到远地主机。其定义了数据和命令应该如何通过互联网，即网络虚拟终端。

现在这个协议很少使用了。

## 万维网

### WWW

* 万维网是一个大规模联机式的信息储藏所，是无数个网络结点与网页的集合。
* 是以方式工作，浏览器就是客户端，文档所在的主机允许服务器程序。
* 用户可以点击超链接获取资源，是通过超文本传输协议传输。
* 万维网使用超文本标记语言使得网页设计者可以从一个页面跳转到另一个页面并能显示页面。

### URL

统一资源定位符唯一标识资源，一般形式是：<协议>://<主机>:<端口>/<路径>。

不区分大小写。

可以使用也可以使用。

### HTTP协议

* 超文本传输协议定义了浏览器怎样向万维网服务器请求文档，以及服务器如何传输文档。
* 服务器监听运行于端口。
* 可以不一次性下载完页面的所有资源，可以只下载文本部分，其他音频视频等待用户下一步请求之后再传输。

#### HTTP协议工作流程

1. 浏览器分析。
2. 浏览器向请求解析地址。
3. 解析出地址。
4. 浏览器与服务器建立连接。
5. 浏览器发出取文件命令。
6. 服务器响应。
7. 释放连接。
8. 浏览器显示。

#### HTTP协议的特点

1. 无状态（即无法记忆用户），如果需要记忆就需要使用，是存储在用户主机的文本协议。
2. 协议采用连接，但是协议本身无连接，即交换报文之前不需要先建立连接。

#### HTTP协议的连接

连接包括：

* 持久连接：
  + 非流水线。
  + 流水线。
* 非持久连接。

![图示

描述已自动生成](data:image/png;base64;base64,)

时间连接较长，每次需要文档接收时间。

![图示

描述已自动生成](data:image/png;base64;base64,)

这个图片是非流水线式，如果是流水线则一次性可以发送多个请求。

#### HTTP协议报文

报文分为请求报文和响应报文，因为其面向文本，所以报文中的每一个字段都是码串。

![HTTPformat]

* 请求报文与响应报文的第一行叫做开始行，用于互相区分。分为请求行和状态行。
  + 请求报文的方法是指命令，就是对所请求的对象进行什么操作，如获取/删除等等。
  + 就是资源标识符。
  + 状态码，表明服务器当前的状态:
    - 表示通知信息的，如请求收到了或正在处理。
    - 表示成功，如接受或知道了。
    - 表示重定向，如要完成请求还必须采取进一步的行动。
    - 表示客户的差错，如请求中有错误的语法或不能完成。
    - 表示服务器的差错，如服务器失效无法完成请求。
  + 版本是指使用的是什么版本的协议。
  + 标识一行的结束，分别为回车和换行。同时，在整个首部行结束时，为了区别首部行和实体主体还会有一行单独的。
* 首部行用于说明浏览器、服务器和报文主体的一些信息。每一行都需要回车和换行，最后还需要一个空行。
* 实体柱体：请求报文一般不用，响应报文也可能没有。

| 方法（操作） | 意义 | 作用 |
| --- | --- | --- |
| HEAD | 请求读取由URL标识的信息的首部 | 服务器可对HTTP报文响应但不返回请求对象，用于调试 |
| GET | 请求读取由URL标识的信息 | 从服务器上获取数据 |
| POST | 给服务器添加信息（如注释） | 向服务器传递数据 |
| CONNECT | 用于代理服务器 | 将服务器作为代理，让服务器代替用户访问其他网页返回返回给用户 |

## 电子邮件

电子邮件是一种异步通信方式。

### 电子邮件格式

* 信封：收件人的邮箱（是必须的，关键词）等。
* 内容：
  + 首部：发送人的邮箱；主题（可选，关键词）等。
  + 主体：邮件内容。

电子邮件地址：收件人邮箱名@邮箱所在主机的域名。如。

### 电子邮件系统组成结构

应该包括三个部分：用户代理、邮件服务器、电子邮件协议。

![emailstructure]

用户代理的四个功能：

* 撰写就是给用户编辑信件的环境。
* 显示就是可以看到自己写的和自己收的信件内容。
* 处理就是对信件进行操作，包括删除，打印，转发等等。
* 通信就是可以将邮件发送到邮件服务器当中，同时可以从邮件服务器当中读取邮件。

邮件服务器的功能：

* 邮件服务器端的发送和接受是指从自己的用户代理处接收邮件，之后向对面的邮件服务器发送邮件。
* 邮件服务器的报告邮件发送结果就是投递是否成功这种情况。
* 邮件服务器既可以作为客户端又可以作为服务器端，使用的是方式。

协议：

* 发邮件用的是。
* 收邮件的是或者。

### SMTP协议

简单邮件传送协议规定了两个互相通信的进程之间应该如何交换信息。

建立在连接上，使用的端口号为。

使用模型，负责发送邮件的进程就是客户，负责接收邮件的进程就是服务器。

这里客户和服务器不是固定死的，可以也可以成为服务器，服务器也可以成为用户，由发送方和接收方决定，发送方就是客户，接收方就是服务器。

只支持传输比特码内容，不支持二进制文本。

#### SMTP协议通信

规定了条命令（几个字母）和种应答信息（三位数字代码+简单文字说明）。具体的过程了解就可以。

令发送方邮件服务器为，接收方邮件服务器为。

1. 连接建立：
   1. 发送方将邮件发送给（客户）的邮件缓存中。
   2. 每隔一段时间就会扫描缓存，如果有邮件就准备使用协议与号发送，与建立连接。
   3. 发出应答信息：，表明可以发送。
   4. 然后向发送一个命令并附上发送方的主机名。
   5. 如果有能力接收邮件，就应答：。否则就应答：。
2. 邮件发送：
   1. 会向发送一个命令：<邮件发送者地址>。
   2. 如果能接收，就返回：，如果不能接收，就返回不能返回的原因的数字代码和英文说明。
   3. 收到允许发送的命令后，会发送多个命令：<邮件接收者地址>，表示发送给谁。
   4. 如果接收到并确认，就返回：，如果不能接收，就返回：。
   5. 发送一个命令，表示要开始传输数据了。
   6. 返回：<><>.<><>。表示同意传输。
   7. 开始传输数据。
   8. 发送命令：<><>.<><>表示数据传输已经结束。
   9. 返回：表示明白传输已经结束。
3. 连接释放：
   1. 邮件发完，发送命令表示要释放连接。
   2. 返回：表示同意释放连接。

### MIME协议

的缺点：

1. 不能传送可执行文件或其他二进制对象。
2. 仅限于传送位码，不能传送其他非英语国家的文字。
3. 服务器会拒绝超过一定长度的邮件。

通用因特网邮件扩充协议改善发送数据的缺点，是的功能性扩展。

![手机屏幕的截图

描述已自动生成](data:image/png;base64;base64,)

协议已经逐渐开始应用到浏览器当中，通过对不同文件类型用不同的标识符标识，来让浏览器读取通过的相关文件。

### POP协议

邮局协议现在一般是。建立于连接上，端口号是，使用模型。

工作方式：

* 下载并保存在服务器。
* 下载并删除。

使用明文在传输层上传输密码，不进行加密。

与协议一样基于码，只能传输码，如果要传输非码则必须使用。

同一个账户可以有多个邮件接收目录。

### IMAP协议

网际报文存取协议比协议更复杂。

提供创建文件夹，移动邮件、查询邮件等连接命令，从而维护了会话用户的状态信息。

当客户端打开服务器邮箱时，用户可以看到邮件的首部，只有打开某个邮件时才上传到用户本地计算机上。所以适用于低带宽的情况，避免取回不想取回的大数据。

可以在不同地方不同的计算机上随时处理邮件，还可以只读取邮件的一部分。

### 万维网邮件

此外，随着万维网的流行，目前出现了很多基于万维网的电子邮件，如、等。这种电子邮件的特点是，用户浏览器与或的邮件服务器之间的邮件发送或接收使用的是，而仅在不同邮件服务器之间传送邮件时才使用。

## DHCP协议

动态主机配置协议用于动态分配地址、子网掩码与默认网关等，使用客户/服务器方式，客户端与服务端通过广播方式进行交互，基于协议（因为没有地址所以无法建立连接）。

提供即插即用机制，允许地址重用、移动用户加入网络、在用地址续租。

工作流程：

1. 主机广播“发现报文”，寻找网络中的服务器，从其中获得一个地址。
2. 服务器收到“发现报文”后，广播“提供报文”，其中包括提供客户机的地址与配置信息。
3. 客户机收到“提供报文”，接收服务器所提供的相关参数，客户机广播“请求报文”，向服务器请求提供地址。
4. 服务器广播“确认报文”，将地址分配给客户机。