

计算机网络



孙吉鹏 谷一滕
杜泽林 张晓敏
林子童 徐卫霞
袁郭苑 鲍 伟

特别鸣谢
朱方金 老师

校对：林子童
排版：张晓敏 林子童
版本号：V1.1
修订时间：2019.1



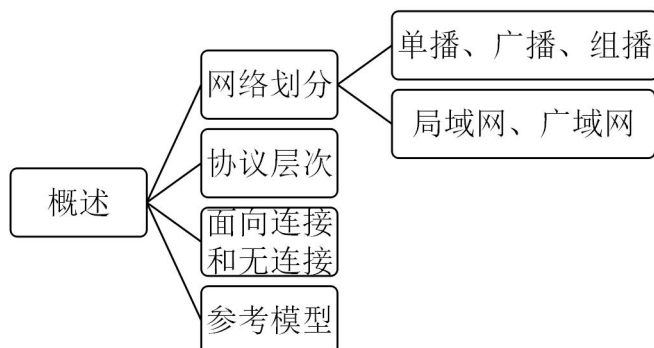
目录

第一章 引言.....	1
1.1 基本概念.....	1
1.2 网络分类.....	1
1.2.1 依据传输模式划分网络.....	1
1.2.2 依据网络尺度划分网络.....	2
1.3 服务、接口与协议.....	2
1.3.1 协议层次结构.....	2
1.3.2 服务和协议的关系.....	3
1.3.3 面向连接与无连接的服务.....	3
1.3.4 可靠和不可靠的服务.....	3
1.4 参考模型.....	4
1.4.1 OSI 模型.....	4
1.4.2 TCP/IP 模型.....	5
1.4.3 OSI 模型和 TCP/IP 模型比较.....	5
第二章 物理层.....	6
2.1 数据通信的理论基础.....	6
2.1.1 概念.....	6
2.1.2 计算信道的最大数据传输速率.....	6
2.2 传导性传输介质.....	6
2.3 公共电话交换网络.....	7
2.3.1 本地回路.....	7
2.3.2 多路复用.....	10
2.3.3 交换技术.....	10
第三章 数据链路层.....	12
3.1 数据链路层的设计问题.....	12
3.1.1 提供给网络层的服务.....	12
3.1.2 成帧.....	13
3.1.3 差错控制.....	15
3.1.4 流量控制.....	15
3.2 差错检测和纠正.....	15
3.2.1 纠错码.....	15
3.2.2 检错码.....	16
3.3 基本数据链路层协议.....	17
3.3.1 一个乌托邦式的单工协议（协议 1）.....	17
3.3.2 无错信道上的单工停-等式协议（协议 2）.....	17
3.3.3 有错信道上的单工停-等式协议（协议 3）.....	17
3.4 滑动窗口协议.....	17
3.4.1 基本概念.....	17
3.4.2 1 位滑动窗口协议（协议 4）.....	17
3.4.3 回退 N 协议（协议 5）.....	18
3.4.4 选择重传协议（协议 6）.....	18
第四章 介质访问控制子层.....	19
4.1 有线局域网协议.....	20

4.1.1 ALOHA 系统.....	20
4.1.2 CSMA.....	20
4.2 无线局域网协议.....	21
4.2.1 无线局域网与有线局域网的不同 P214.....	21
4.2.2 MACA 冲突避免多路访问.....	21
4.2.3 CSMA/CA.....	22
4.3 交换机（网桥）P257—P260.....	23
4.3.1 工作原理.....	23
4.3.2 目的地-端口哈希表（转发表）的获得.....	23
4.3.3 中继器/集线器/网桥/交换机/路由器/网关的对比 P263—P264.....	23
第五章网络层.....	24
5.1 网络层两类服务.....	24
5.2 路由算法——基于最优路径的路由算法.....	24
5.2.1 最优化原则（P281）.....	24
5.2.2 泛洪算法（P284）.....	24
5.2.3 距离矢量路由算法 DV（RIP 协议）.....	25
5.2.4 链路状态路由算法 LSP（OSPF、IS-IS 协议）.....	25
5.2.5 层次路由.....	26
5.3 拥塞控制.....	26
5.4 流量整形——平滑的流量更好管理.....	26
5.5 IPv4 协议.....	27
5.5.1 IP 头.....	27
5.5.2 分类寻址 Classful Addressing.....	28
5.5.3 子网 Subnet 与前缀 prefix.....	28
5.5.4 无类域间路由 CIDR.....	28
5.5.5 NAT 网络地址转换.....	30
5.5.6 隧道技术.....	30
5.6 Internet 控制协议.....	30
5.6.1 ICMP 控制消息协议.....	30
5.6.2 ARP.....	30
第六章 传输层.....	32
6.1 传输协议的要素.....	33
6.1.1 建立连接.....	33
6.1.2 断开连接.....	33
6.2 UDP.....	34
6.2.1 UDP 简介.....	34
6.2.2 UDP 的一些特点.....	34
6.3 TCP.....	34
6.3.1 TCP 数据段的头.....	34
6.3.2 TCP 连接建立（三次握手）.....	36
6.3.3 TCP 连接断开（三次握手）.....	36
6.3.4 拥塞控制与慢启动算法.....	36
第七章 应用层.....	38

第一章 引言

作者 鲍伟



1.1 基本概念

1. 计算机网络(computer networks)

表示一组通过单一技术相互连接起来的自主计算机集合。

2. 分布式系统 (distributed system)

分布式系统是建立在网络之上的软件系统，有高度的内聚性和透明性。

内聚性：每一个数据库分布节点高度自治，有本地数据库管理系统

透明性：每个数据库分布节点对用户应用来说是透明的，用户感觉不到数据是分布的

Internet 是最著名的计算机网络，万维网是最著名的分布式系统，万维网（软件）运行于 **Internet**（硬件）上

3. 虚拟专用网络(VPN,virtual private networks)

一种可以将不同地点的单个网络联结成一个扩展网络的技术。

4. P2P 与 CS(client-server)

CS：由高性能计算机服务器和普通计算机客户机组成，服务器负责存储数据并处理客户请求，而客户机可远程访问服务器

P2P：对等模型（又称工作组），各台计算机具有相同功能，一台计算机可作为服务器设定共享资源供网络中其他计算机使用，又可作为工作站。没有专用的服务器或工作站。

1.2 网络分类

1.2.1 依据传输模式划分网络

单播(unicasting)：只有一个发送方和一个接收方的点到点传输，也叫点到点链路(point-to-point)

广播(broadcasting)：任何一台机器发出的数据包能被其他人任何机器收到。每个数据包的地址字段指定了预期的接收方，只有预期的接收方会做出应答，其他的机器会忽略这个数据包。

多播/组播(multicasting)：将数据包发给一组机器，即所有机器的一个子集。广播

可以看成是一种特殊的组播形式。

1.2.2 依据网络尺度划分网络

1. 个域网(PAN, Personal Area Network): 允许设备围绕一个人进行通信。一个常见的例子是计算机通过无线网络与其外围设备链接。突出的技术就是蓝牙(bluetooth)。

2. 局域网(LAN, Local Area Network): 一种局部地区的私有网络, 一般在一座建筑物内或是建筑物附近, 比如家庭、办公室或工程。具体分为有线和无线两种。

局域网特点: ①距离短; ②传输速率高; ③错误率低。

1) 无线 LAN: 每台计算机都有一个无线调制解调器和一个天线, 用来和其他计算机通信。大多数情况下是和一台设备通信, 这个设备称为接入点(AP, Access Point)、无线路由器或者基站。这个设备主要负责中继无线计算机之间的数据包, 还负责中继无线计算机和 Internet 之间的数据包。代表技术就是 WIFI。

2) 有线 LAN: 大多使用铜线作为传输介质, 也有一些使用光纤。

许多有线局域网的拓扑结构是以点到点链路为基础的, 俗称以太网的 IEEE 802.3 是迄今为止最常见的一种有线局域网。

每台计算机按照以太网协议规定的方式运行, 通过一条点到点链路链接到一个盒子, 这个盒子称为交换机(switch), 一台交换机有多个端口, 每个端口连接一台计算机。交换机的工作是中继与之连接的计算机之间的数据包, 根据数据包中的地址来确定这个数据包要发送给哪台计算机。

有线局域网在性能的所有方面都超过了无线局域网, 因为通过电线或通过光纤发送信号比通过空气发送信号更容易。

3. 城域网(MAN, Metropolitan Area Network): 范围覆盖一个城市。最著名的城域网例子是许多城市都有的有线电视网。

4. 广域网(WAN, Wide Area Network): 范围很大, 它跨越很大的地理区域, 通常是一个国家、地区或者一个大陆。

通信子网(subnet): 我们按照传统的说法把机器叫做主机, 把链接这些主机的网络其余部分称为通信子网, 或简称子网。

子网的工作是把信息从一个主机携带到另一个主机

子网由两个不同部分组成: 传输线路和交换元素。

传输线路负责在机器之间移动比特, 它们可以是铜线、光纤、甚至无线链路。

交换元素或简称交换机是专用的计算机, 负责链接两条或两条以上的传输线路。现在一般称为路由器(Router)

5. 互联网: 一组相互连接起来的网络。

1.3 服务、接口与协议

1.3.1 协议层次结构

协议(protocol): 是指通信双方就如何进行通信的一种约定。

接口(interface): 定义了下层向上层提供哪些原语操作和服务。(它告诉上面的进程如何访问本层, 规定了有哪些参数以及结果是什么, 但并未说明工作过程和服务方式。)

服务是由一组原语(primitive)正式说明, 用户可以通过这些原语来访问该服务。

协议栈：一个特定的系统所使用的一组协议。

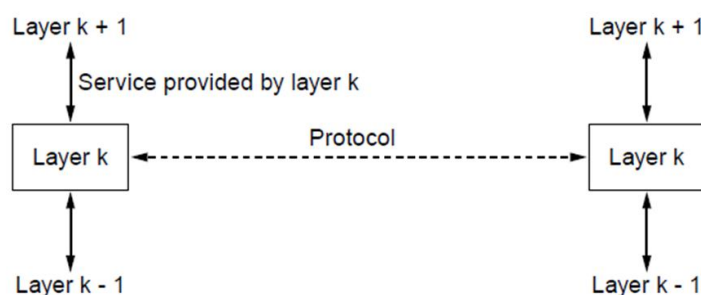
对等体(peer)：不同机器上构成相应层次的实体称为对等体。

网络体系机构(network architecture)：层和协议的集合。

1.3.2 服务和协议的关系

1. 服务是指某一层向它上一层提供的一组原语操作，服务定义了该层打算代表其用户执行哪些操作，但是他不涉及如何实现这些操作，服务也会涉及到两层之间的接口，其中底层是服务提供者，而上层是服务的用户。(上下层之间的联系)

2. 协议是一组规则，用来规定同一层上的对等体之间所交换的消息或者分组的格式和含义。这些实体利用协议来实现他们的服务定义，他们可以自由的改变协议而不影响它提供给上层的服务(对等体之间的规范)。



1.3.3 面向连接与无连接的服务

面向对象的连接的服务(connection-oriented service)：是按照电话系统建模的。服务用户首先必须建立一个连接，然后使用连接传输数据，最后释放连接。本质上像一个管道。

无连接的服务(connectionless service)：是按照邮政系统建模的。每一个报文都携带者完整的目的地地址，每个报文都由系统中的中间节点路由，并且独立于后续的报文。

区别：

- a. 面向连接的要求建立连接，因而没有传输的数据没有必要再标明传输的目的地址；而无连接的则对每个报文都由独立的目标地址。
- b. 一般来说，面向连接的可靠性较高，协议相对复杂，传输的数据按照发送顺序到达；而无连接的可靠性较差，协议相对简单，常出现乱序，重复和丢失现象。

1.3.4 可靠和不可靠的服务

1. 可靠服务：即从来不会丢失数据的一种服务。一般情况下，可靠服务都要求接收方向发送方确认收到的每个报文。

2. 不可靠的服务：不会给发送方反馈任何确认消息，不保证数据不丢失。

3. 可靠与不可靠服务同时存在的原因：

在给定的层次可靠通信并不总是可以使用的。

为了提高可靠服务而导致的固有延迟可能是不可接受的。

4. 例题：面向连接的服务是可靠的吗？

面向连接的服务只是在发送方和接收方之间建立连接，它并不能保证发送的数据流能准确无误的按序到达接收方。面向连接的服务同样分为可靠的面向连接服务和不可靠的面向连接服务。其中，前者主要包括报文序列、字节流，后者如数字化语音。

1.4 参考模型

1.4.1 OSI 模型

Open Systems Interconnection

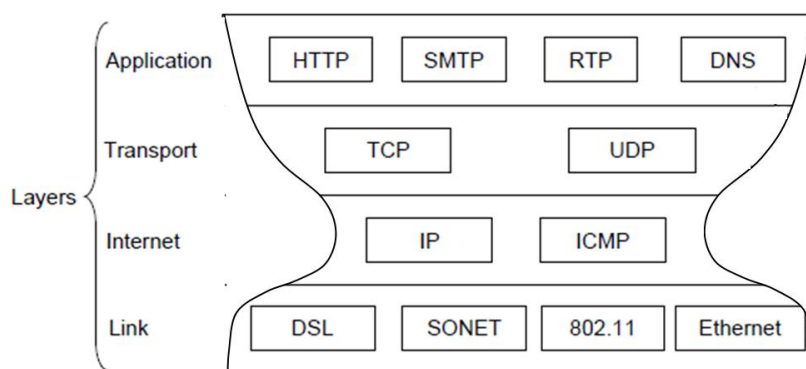
具体分为七层，由低到高分别为：物理层，数据链路层，网络层，传输层，会话层，表示层，应用层。

7	Application	– Provides functions needed by users
6	Presentation	– Converts different representations
5	Session	– Manages task dialogs
4	Transport	– Provides end-to-end delivery
3	Network	– Sends packets over multiple links
2	Data link	– Sends frames of information
1	Physical	– Sends bits as signals

名称	数据格式	功能与连接方式	典型设备
物理层	传输比特 (bit) 流	工作环境为：相邻两节点，提供点到点的传输。涉及到在通信信道上传输的原始数据位。建立、维护和取消物理连接	光纤、同轴电缆、双绞线、中继器和集线器
数据链路层	将比特信息封装成数据帧 Frame	它完成了相邻两节点之间的可靠传输。将一个原始的传输设施转变成一条逻辑的传输线路，在这条线路上，所有未检测出来的传输错误也会被反映到网络上。	网桥、交换机、网卡
网络层	分割和重新组合数据包 Packet	控制子网运行过程，确定传输路径	路由器
传输层	数据组织成数据段 Segment	为网络层找到的路径提供可靠的传输。接受来自上一层的数据，并且在必要时把这些数据分割成小的单元，然后把数据单元传递给网络层，并确保这些数据片段都能够正确到达另一端。	
会话层	数据 Data	允许不同机器上的用户之间建立回话	
表示层	数据 Data	关注所传递的信息的语法和语义，取消格式差异	
应用层	数据 Data	包含各种各样的协议	

可以看到：第 n 层的问题若第 n 层无法完全解决，那么需要依靠其上一层。如物理层实现了点到点的传输，而链路层使此传输变得可靠；网络层确定了传输路径，而传输层使此传输变得可靠。

1.4.2 TCP/IP 模型



由低到高具体包括：链路层，互联网层，传输层，应用层。

本书使用的模型：

使用了混合模型，包括五层：物理层，数据链路层，网络层，传输层，应用层。

1.4.3 OSI 模型和 TCP/IP 模型的比较

相同点：两者都是建立在协议栈概念上的，并且协议栈中的协议彼此相互独立。同时两个模型中各个层的功能也大致相似。

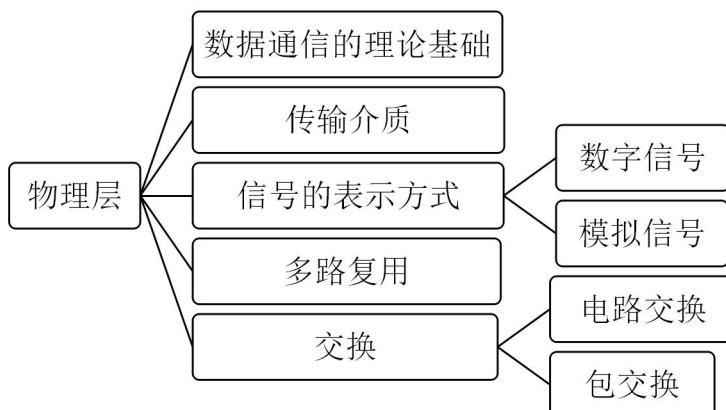
不同点：OSI 模型的实力在于模型本身，TCP/IP 模型实力在于协议。

OSI 模型分为 7 层，明确区分了服务、接口和协议。OSI 模型中的协议具有更好的隐蔽性，也更加的通用。这个模型是在协议之前产生的，它的网络层同时支持无连接和面向连接的通信，但是传输层只支持面向连接的通信。

TCP/IP 模型分为 4 层，没有明确区分服务、接口和协议。通用性差，不适合描述非 TCP/IP 网络。TCP/IP 模型是先有的协议后有的模型，协议和模型切合度高。TCP/IP 模型的网络层只支持无连接的通信，但是传输层同时支持两种。

第二章 物理层

作者 徐卫霞



2.1 数据通信的理论基础

2.1.1 概念

1. **带宽 bandwidth**: 在传输中不会明显减弱的频率的宽度，通常引用的带宽是指从 0 到使得接收能量保留一半的那个频率位置，是传输介质的一种物理属性。，通常取决于介质的构成、厚度、电线或者光纤的长度。

2. **信噪比(SNR)**: 信号功率 S 与噪声功率 N 的比值，即为信噪比 S/N 。

3. **分贝(dB)**: 通常把信噪比表示成对数的形式 $10\log_{10}S/N$ ，对数的取值单位称为分贝。信噪比为 100 可表示为 20dB。

2.1.2 计算信道的最大数据传输速率

1. 尼奎斯特定理 Nyquist

用来表示一个有限带宽的无噪声信道的最大数据传输率。

表达式: (每秒 2B 次采样)最大数据速率 $= 2B\log_2 V$ (比特/秒)

B : 带宽 V : 离散级数, 即可识别的信号个数

2. 香农定理 Shannon

用来表示一条带宽为 B Hz, 信噪比是 S/N 的有噪声信道的最大数据传输率或容量。

表达式: 最大数据传输率 $= B\log_2(1+S/N)$ (比特/秒)

B : 带宽 S/N : 信噪比

2.2 传导性传输介质

1. **磁介质**: 良好的带宽, 但是延迟高。

2. **双绞线 (twisted pair)**

原理: 两根线绞在一起, 噪音对他们的干扰是一样的, 所以他们的电压差不会改变, 通过电压差来表示信号。

类型: category 5.

概念:

①全双工链路 (full-duplex): 可以双向同时用

②半双工链路 (half-duplex): 可以双向使用但每一时刻只允许使用一个方向

③单工链路 (**simplex**): 只允许一个方向传输

3. 同轴电缆 (**coaxial cable**)

结构: (自内到外)铜芯, 绝缘材料, 编织外层导体, 保护塑料外套

优点: 很高的带宽, 很好的抗噪性。

4. 电力线

5. 光纤: 单模光纤、多模光纤

2.3 公共电话交换网络

(**PSTN, public switched telephone network**)

PSTN 是一种常用的旧式电话系统, 提供的是一个模拟的专用通道, 通道之间经由若干电环交换机连接而成, 当两台主机或路由器需通过 **PSTN** 连接时, 必须在网络接入侧使用调制解调器实现信号的模数/数模转换。

电话系统的组成: 本地回路、干线上的(多路复用)和交换局的(交换机、交换技术如虚电路交换和分组交换等)

2.3.1 本地回路

1. 概念

①数字调制 (**digital modulation**): 用模拟信号来表示比特, 比特与代表它们的信号之间的转换过程。

②基带传输 (**baseband transmission**): 信号的传输占用传输介质上从零到最大值之间的全部频率。(这是**有线**介质普遍使用的一种调制方式)。

③通带传输 (**passband transmission**): 信号占据了以载波信号频率为中心的一段频带(这是**无线和光纤**最常使用的调制方法)。

④调制解调器 (**modem**): 执行数字比特流和模拟信号流之间转换的设备。分为调制器 (**modulator**, 数字比特流转换成模拟信号)和解调器 (**demodulator**, 模拟信号转换成数字比特流)

⑤非对称数字用户线 (**ADSL, asymmetric DSL**): 一种数据传输方式, 上行和下行带宽不对称(下行速率大于上行速率, 因为大多数用户下载数据量超过上传。)。采用 **FDM** 把普通的电话线分成电话、上行和下行三个相对独立信道, 从而避免相互之间的干扰。

2. 基带传输 数字信号

1) 相关概念

①带宽效率

②时钟恢复问题: 存在一长串 **0** 或 **1** 时, 经过较长时间会导致接收方无法准确判定信号的每个比特, 比如 15 个 **0** 很像 16 个 **0**。

2) 数字信号的表示方法

下面所有例子的 **0** 和 **1** 表达方式可以交换, 即 **0** 高 **1** 低可换成 **0** 低 **1** 高, 无影响

①不归零 (**NRZ, non-return-to-zero**): 简单的将低电平表示为 **0**, 高电平表示为

1.

问题: 带宽效率低; 较多连续的 **0** 或 **1** 导致接收方无法分辨每个比特

②不归零逆转 (**NRZI, non-return-to-zero**): **0** 时信号不发生变化, **1** 时信号跳变
问题: 可以对连续的 **1** 进行区分, 但是不能对连续的 **0** 区分。



③曼彻斯特编码 (**Manchester**): 将数据信号与时钟信号通过异或方式混合在一起。



问题: 带宽效率低(需要的带宽是 **NRZ** 的两倍)

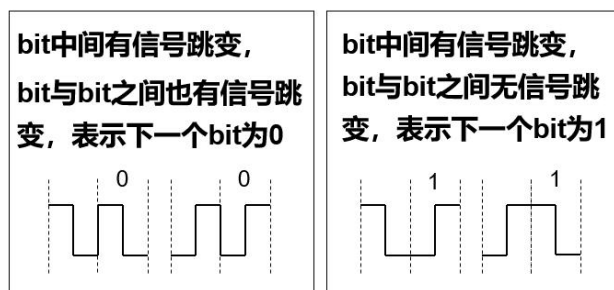
④差分曼特斯特编码: 相对调相的编码, 与时钟的表达很相近, 但不同点是: 若 **1** 则在

起始位置进行跳变，若为 0 则不进行跳变。

其中曼彻斯特和差分曼彻斯特可以这样记忆：

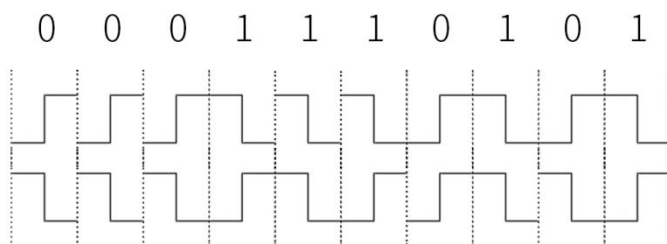
曼彻斯特中 1 用  来表示，0 用  来表示

差分曼彻斯特则是设定一个初始电平（默认为高电平）以后，0 和 1 也是用  和  来表示，只是不固定，而改为：0 则起始位置有跳变，1 则起始位置无跳变。



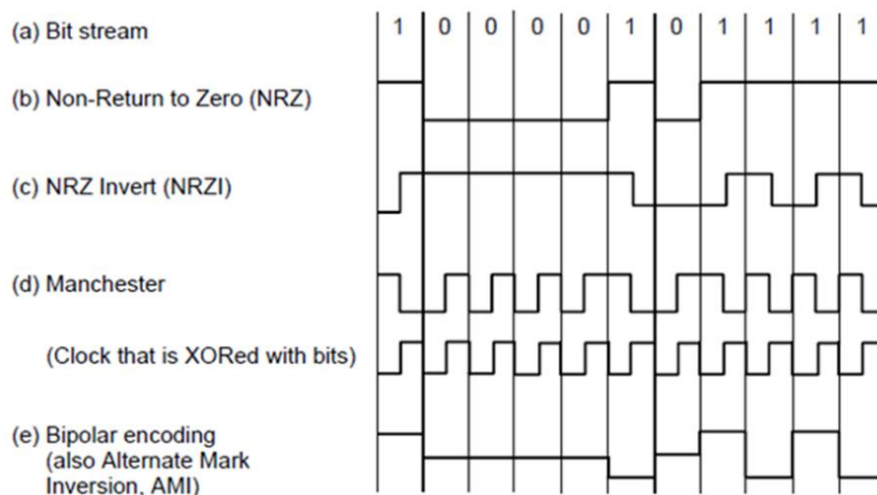
特性与曼彻斯特编码相同，但抗干扰性能强于曼彻斯特编码

下图分别是曼彻斯特编码和差分曼彻斯特编码的示意图：

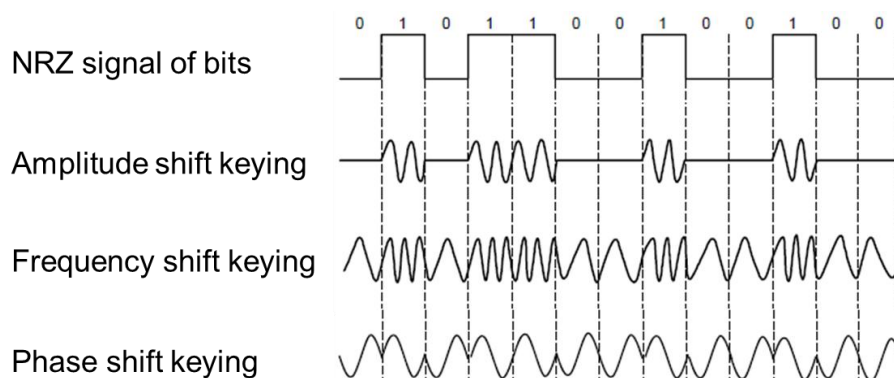


注意：两种曼彻斯特编码的 1、0 位的定义没有严格的要求，可以与上图表示的相反，只要在过程中完全一致即可

⑤N 级编码：采用 N 个信号级别，如用单个符号可以一次携带 2 个比特，只要接收方可以辨识信号的四个级别即可。如二级编码。



3 通带传输 模拟信号



$y=A\sin(fx+\phi)$ ，下面的分别表示是改变 A 、 f 和 ϕ 三个参数来区分不同信号

①幅移键控 (ASK, amplitude shift keying): 不同的振幅表示 0, 1.

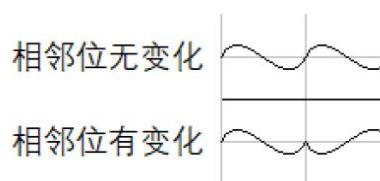
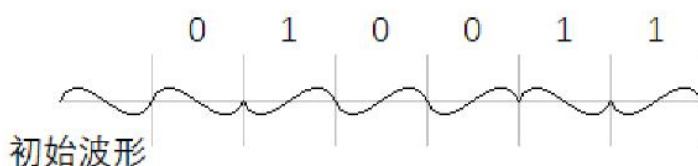
②频移键控 (FSK, frequency shift keying): 不同的频率表示不同的信号，如采用两个不同的频率分别表示 0, 1

③相移键控 (PSK, phase shift keying): 不同的相位表示不同的信号。在每个符号的周期中，把载波波形偏移 0° 或 180° 。更有效利用信道带宽的方法是使用四个偏移 (例如 45° , 90° , 135° , 180°)，这样每个符号可以表示两个比特信息，这种方式被称为正交相移键控 (QPSK, quadrature phase shift keying)。

上述①②③方法均为绝对调相 (相连两位之间互不影响)。

④相对调相 (relative phase modulation): 相邻两位之间存在相互影响。若为 1 则跳变，若为 0 则无变化，需要给出初始波形，和差分曼彻斯特类似。具体相位变化表示如右图

例子如下所示 (给出初始波形):



2.3.2 多路复用

多路复用：使多个信号可以共享同一传输路线。

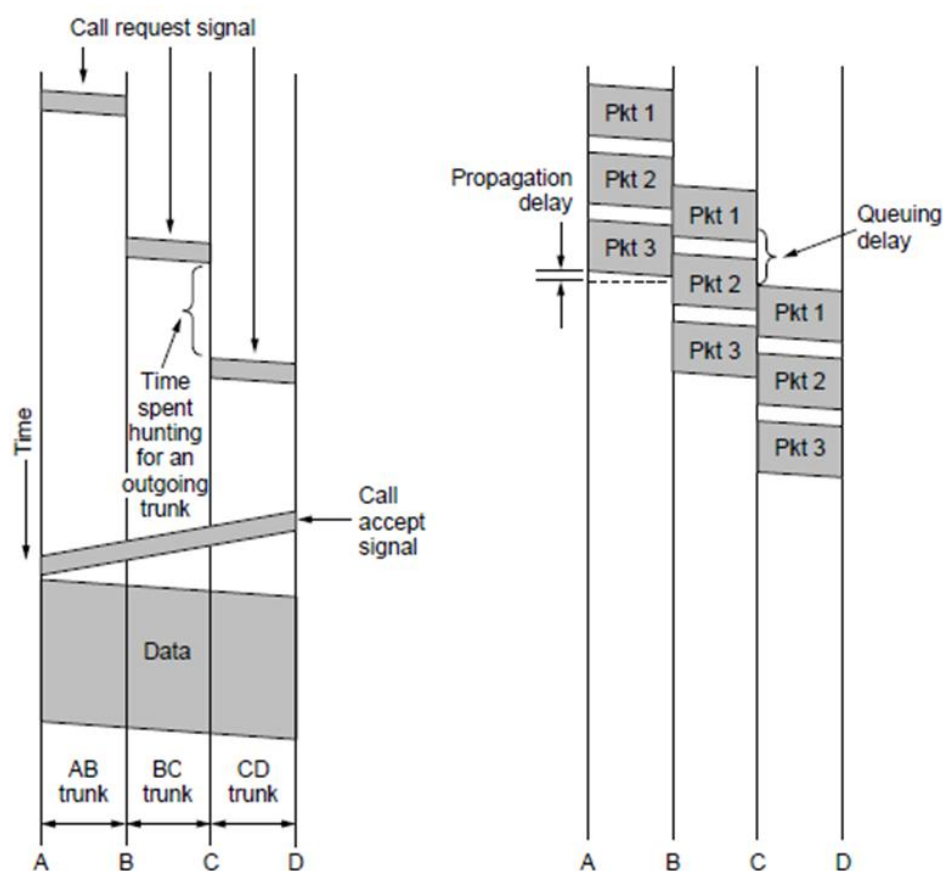
分类：

①时分（TDM, time division multiplexing）：一条物理信道按时间分成若干个时间片轮流地分配给多个信号使用。

②频分（FDM, frequency division multiplexing）：按频谱划分信道，不同频率的信号可以在同一信道内传输。

③波分（WDM, wavelength division multiplexing）：频分多路复用的一种，利用光纤信道的巨大带宽，同一光纤可以同时传输一组不同波长的光信号，并且不会互相影响。

2.3.3 交换技术



左为电路交换，右为数据包交换

1. 电路交换

（面向连接）电路交换是以电路连接为目的的交换方式，通信之前要在通信双方之间建立一条被双方独占的物理通道。一旦一个呼叫被建立起来，在两端之间的专用路径被建立就会持续到该次呼叫结束为止。

电路交换的三个阶段：(1)建立连接(2)通信(3)释放连接

2. 包交换（packet switching，也称分组交换）

（非连接）路由器采用存储-转发技术，把经过它的每个数据包（根据网络线路、包的目的地址等条件）发送到通往该包目的地的路径上，没有固定的路径每个包都可以都不同的路径，所以它们到达的顺序可能出现混乱。

出现的问题：排队延迟（queuing delay）：数据包可能会因为存在很多包要被转发而

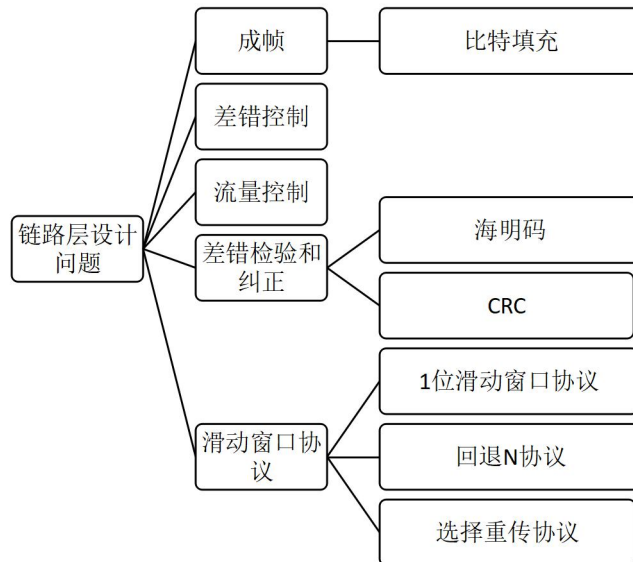
需要等待一段时间才能到它被转发，还可能引起拥塞。
包交换和电路交换的异同见课本 2.6 节的图 2-44.

第三章 数据链路层

作者 谷一滕

数据链路层是基于物理层不可靠的传输向上层提供可靠的传输，它提供的是相邻两个节点之间可靠的数据传输。

本章主要涉及网络模型中第二层（即数据链路层）的设计原则。实现通过一条通信信道连接起来的两台机器，实现可靠有效的完整信息块（称为帧）通信的一些算法。解决通信线路中出错的情况、关于有限的数据传输率、发送时间和接受时间存在的非零延迟等问题。



3.1 数据链路层的设计问题

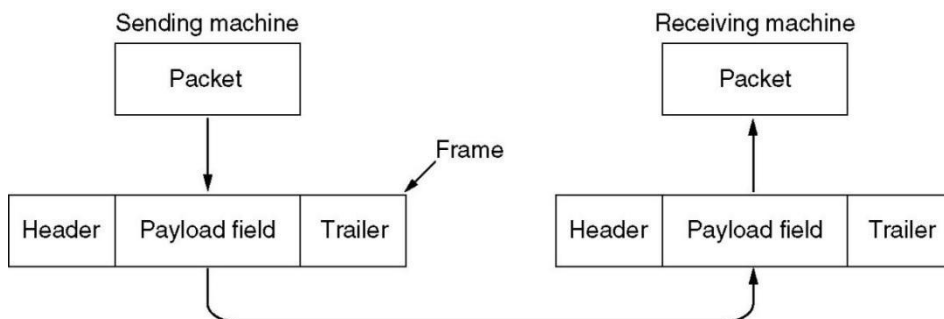
数据链路层使用物理层提供的服务在通信信道上发送和接受比特。完成一些功能：

- (1) 向网络层提供一个定义良好的服务接口。
- (2) 处理传输错误。
- (3) 调节数据流，保证慢速的接收方不会被快速的发送方淹没。

为实现这些目标：数据链路层从网络层获得数据包，并将之包装成包含：

帧头 + 有效载荷（存放数据包）+ 帧尾 的帧（frame）

数据链路层的工作核心就是帧的管理。

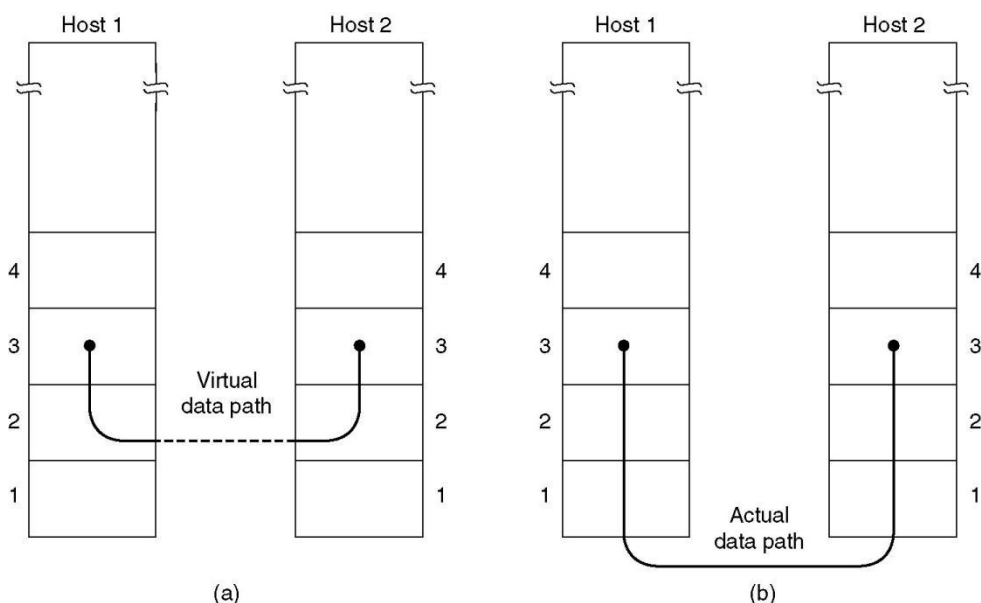


帧的通用结构：

ack 捎带确认信息	kind 帧类型	seq 帧编号	data 数据	checksum 校验和
------------	----------	---------	---------	--------------

3.1.1 提供给网络层的服务

最主要的服务是将数据从源机器的网络层传输到目标机器的网络层：



一般情况下，提供以下三种可能的服务：

(1) 无确认的无连接服务。

源机器向目标机器发送独立的帧，目标机器不对这些帧进行确认。不需建立逻辑连接。适用于错误率低或者实时通信（语音传输）的情况。

(2) 有确认的无连接服务。

源机器向目标机器发送独立的帧，目标机器会对这些帧进行确认。不需建立逻辑连接。适用于不可靠的信道（无线系统，WiFi）。

(3) 有确认的有连接服务。

源机器和目标机器在传输任何一个数据之前要建立一个连接，保证目标机器按照正确的顺序接受每一个帧。适用于长距离且不可靠的链路（卫星信道，长途电话）。

3.1.2 成帧

为检测错误和纠正错误，数据链路层将比特流拆分成多个离散的帧，为每个帧计算一个称为校验和的短令牌，并将该校验和放在帧中一起传输。为拆分比特流需要解决两个问题：

(1) 帧的边界问题：如何识别帧的边界；

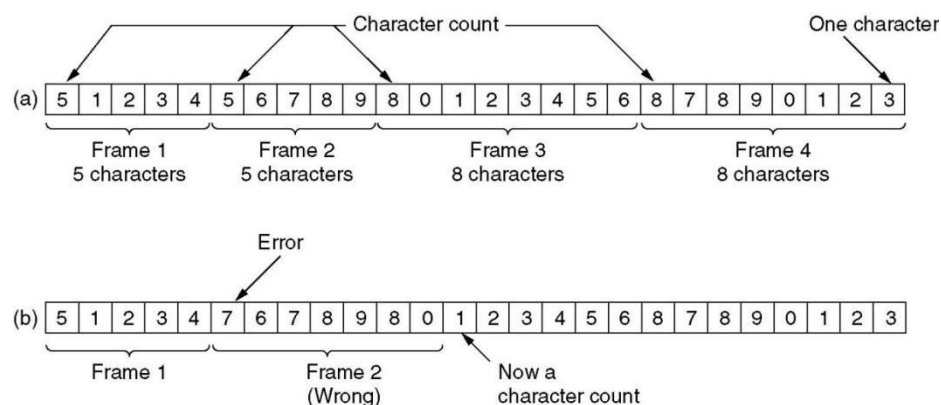
(2) 帧的透明传输（填充）问题：如果帧的数据中出现和边界一样的 flag 该如何防止被识别为边界

1. 四种成帧方法

(1) Character count（字节计数法）

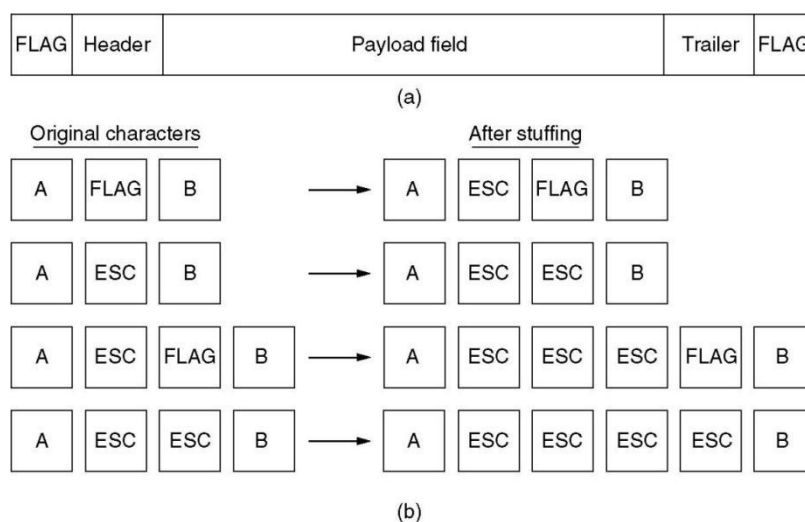
用头部的一个字段来标识该帧中的字符数：

问题：因为一个传输错误，就会全弄混。很少被使用。



(2) Flag Bytes with byte stuffing (字节填充的标志字节法)

发送方使用标志字节 (FLAG) 作为开始和结束；使用转义字节 (ESC) 表示其后的字节为数据字节而不是标志字节或转义字节。接收方将收到的数据中的转义字节删除后再传递给网络层。



(3) Starting and ending flags, with bit stuffing (比特填充的标志比特法)

使用“01111110”表示帧的开始和结束（帧的边界问题解决），并且在数据中，若遇到5个连续的比特1，就在其后填入一个比特0（帧的填充问题解决）。接收方除了将首尾的“01111110”删除外，还要将数据中所有5个连续比特1其后的比特0删除。

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

相较于方法（2），帧长度增幅更少，降低了传输数据内容。

(4) Physical layer coding violation (物理层编码违禁法)

使用“不会出现在常规数据中”的冗余比特作为边界。好处是除了开始和结束的填充外，不再需要填充额外的数据。

3.1.3 差错控制

为确保所有帧按照正确顺序传递给目标机器的网络层：

发送方发送反馈信息来确保传递可靠。

引入计时器来防止硬件故障或通信信道出错等原因丢失某一帧使发送方持续等待确认。

通过序号保证每一帧按照顺序且不会被接收方重复接收。

具体的确认方式在后面通过协议的形式来讲

3.1.4 流量控制

发送方发送帧的速度超过了接收方能够接收这些帧的速度，而导致丢帧。

解决方法：

基于反馈的流量控制（链路层）。

基于速率的流量控制（网络层）。

3.2 差错检测和纠正

3.2.1 纠错码

推断出被发送的数据是什么。适用于错误发生很频繁的信道，因为再次传输仍可能出错。

海明码。（参考书上 159 页）

1) 海明码的生成（顺序生成法）。

例3. 已知：信息码为：“1 1 0 0 1 1 0 0”（k=8）

求：海明码码字。

解：1) 把冗余码A、B、C、…，顺序插入信息码中，得海明码

码字：“A B 1 C 1 0 0 D 1 1 0 0”

码位：1 2 3 4 5 6 7 8 9 10 11 12

其中A, B, C, D分别插于 2^k 位(k=0, 1, 2, 3)。码位分别为1, 2, 4, 8。

2) 冗余码A, B, C, D的线性码位是：（相当于监督关系式）

A→1, 3, 5, 7, 9, 11;

B→2, 3, 6, 7, 10, 11;

C→4, 5, 6, 7, 12; (注 5=4+1; 6=4+2; 7=4+2+1; 12=8+4)

D→8, 9, 10, 11, 12。

3) 把线性码位的值的偶校验作为冗余码的值(设冗余码初值为0)：

$$A = \sum (0, 1, 1, 0, 1, 0) = 1$$

$$B = \sum (0, 1, 0, 0, 1, 0) = 0$$

$$C = \sum (0, 1, 0, 0, 0) = 1$$

$$D = \sum (0, 1, 1, 0, 0) = 0$$

4) 海明码为：“1 0 1 1 1 0 0 0 1 1 0 0”

2) 海明码的接收。

例4. 已知：接收的码字为：“1 0 0 1 1 0 0 0 1 1 0 0”(k=8)

求：发送端的信息码。

解：1) 设错误累加器(err)初值=0

2) 求出冗余码的偶校验和，并按码位累加到err中：

$$A = \sum (1, 0, 1, 0, 1, 0) = 1 \quad err = err + 2^0 = 1$$

$$B = \sum (0, 0, 0, 0, 1, 0) = 1 \quad err = err + 2^1 = 3$$

$$C = \sum (1, 1, 0, 0, 0) = 0 \quad err = err + 0 = 3$$

$$D = \sum (0, 1, 1, 0, 0) = 0 \quad err = err + 0 = 3$$

由 $err \neq 0$ 可知接收码字有错，

3) 码字的错误位置就是错误累加器(err)的值3。

4) 纠错——对码字的第3位值取反得正确码字：

“1 0 1 1 1 0 0 0 1 1 0 0”

5) 把位于 2^k 位的冗余码删除得信息码：“1 1 0 0 1 1 0 0”

3.2.2 检错码

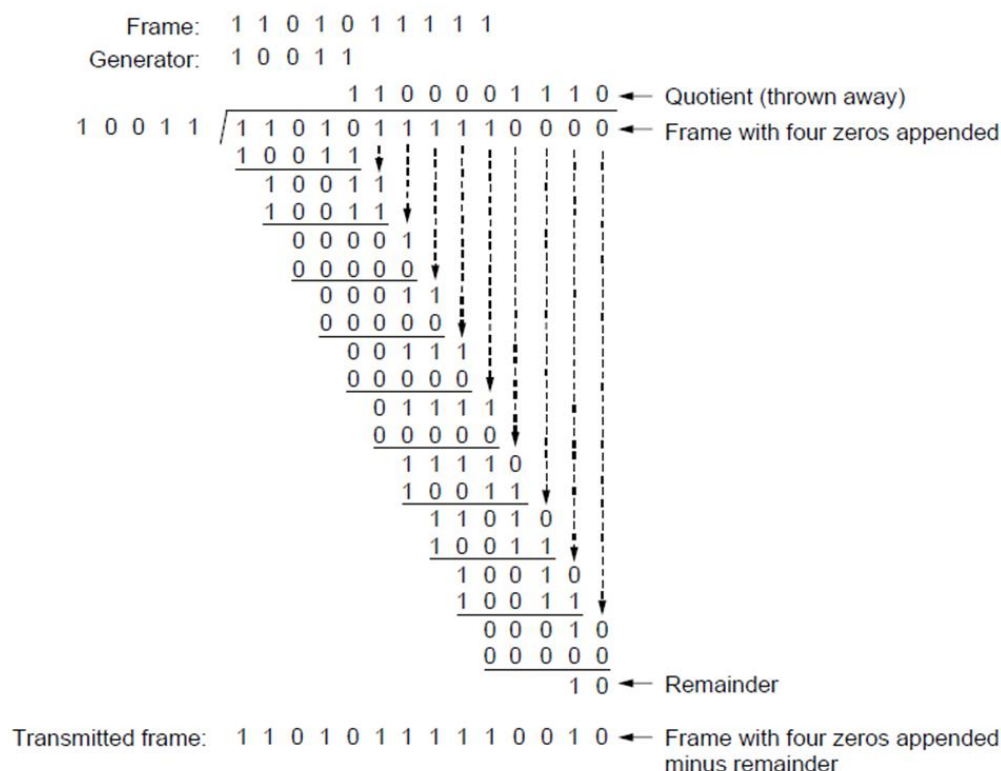
推断是否发生错误。适用于高度可靠的信道，错误偶尔发生时，只需重传整个数据块即可。循环冗余校验码(CRC, Cyclic Redundancy Check，也称作多项式编码)。注意有一定的误判率。

步骤：（以下面算式为例讲解，具体原理参见课本 P165）

① 收发双方商定一个比帧短且头尾都是 1 的 01 串叫做生成多项式，如下的 10011

② 帧的后面附上生成多项式长度减一个 0 后，作为被除数对生成多项式模 2 除，得到商和余数，如 10011 长度为 5，所以附上了 4 个 0

③ 若帧与余数合并以后在接收方被生成多项式整除，那么认为传递的过程没有出错



模 2 除原则：列竖式的方式同除法，但是模 2 除法中加法无进位，减法无借位，即加

减皆等同于异或，而商 0 还是 1 由被除数首位决定，首位是 1 商 1，否则商 0
(务必手动计算几个例子来了解计算步骤和原则)

3.3 基本数据链路层协议

组成帧的四个字段：**kind**、**seq**、**ack** 和 **info**。前三个包含控制信息，称为帧头，最后一个可能包含了要被传输的实际数据。

3.3.1 一个乌托邦式的单工协议（协议 1）

不需考虑任何错误情况：数据单向传输，双方总是就绪，数据处理时间不计，缓存空间无限大，通信信道永不丢帧。

这是一个完全不现实（理想化）的协议，“乌托邦”协议。

发送过程是一个无限的 **while** 循环，它尽可能快速地把数据放到线路上。无差错控制或者流量控制方面的限制。接受过程一直等待一个未损坏的帧到达。（发送速率和接收速率必须一样快）

3.3.2 无错信道上的单工停-等式协议（协议 2）

单向数据传输，发送方网络层一直有无限的数据要发送，信道不会出错，从不损坏或丢失帧，发送方需等待接收方确认帧返回后才发送下一帧。如果接收方不反馈应答信号，则发送方必须一直等待，然后就陷入等待接收方确认信息的过程中，因而传输效率低。

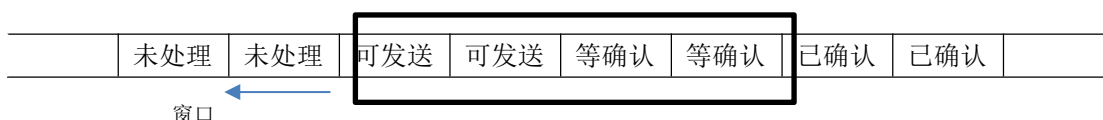
3.3.3 有错信道上的单工停-等式协议（协议 3）

信道存在噪音。需要计时器与序号配合，需要超时重传机制。

3.4 滑动窗口协议

3.4.1 基本概念

滑动窗口（**sliding window**）：为了便于理解，可以认为数据是一条传送带，而滑动窗口中的数据是当前准备处理的数据。一旦窗口的第一条数据被确认处理结束，窗口就会继续向后滑动以处理后面的数据。下面是发送方的发送窗口形象化的表示：



捎带确认（piggybacking）：暂时延缓确认以便将确认信息搭载在下一个出境数据帧上的技术。捎带确认通常与累计确认一同使用，更好的利用了信道的可用带宽。

期望确认：收到数据帧以后向发送方发送期望对方发送的下一帧的序号。

累计确认（cumulative acknowledgement）：当 n 号帧的确认到达， $n-1$ 号帧、 $n-2$ 号帧等都会自动被确认。

否定确认（NAK）：接收方检测到错误时发送的否定确认，实际是一个重传请求，在 **NAK** 中指定了要重传的帧。

发送窗口（sending window）：发送方总维持着一组序号，分别对应于允许它发送的帧，我们称这些帧落在发送窗口。

接收窗口（receiving window）：接收方维持着的一个窗口对应于一组允许它接受的帧。

3.4.2 1 位滑动窗口协议（协议 4）

（发送窗口大小=1，接收窗口大小=1）

当接收窗口大小为 1 时，可保证帧的有序接收，但效率较低。

源站发送单个帧后必须等待确认，在目的站的确认到达源站之前，源站不能发送其他数据帧。这是因为发送窗口大小仅为 1，必须用来保存当前未确认的帧以超时重传。

发送方必须在内存中保存所有的帧，因此如果最大窗口的尺寸为 n ，则发送方需要 n 个缓冲区才存放未被确认的帧。

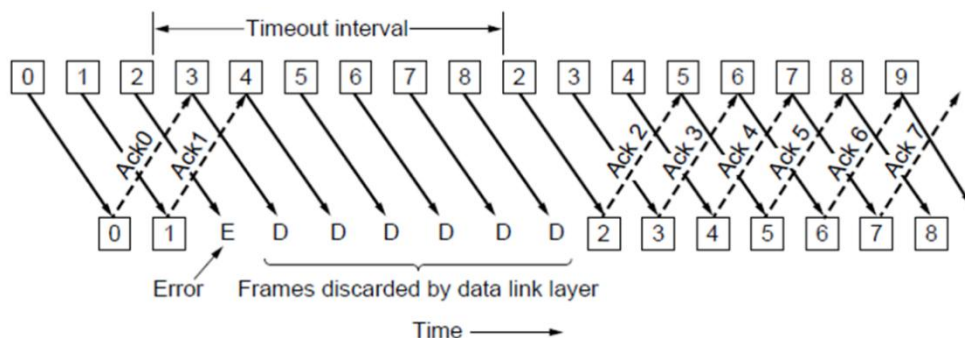
3.4.3 回退 N 协议（协议 5）

（发送窗口大小 >1 ，接收窗口大小 $=1$ ）

发送方按照顺序向对方发送帧，在收到对方的确认以后窗口向后滑动，若当前窗口中的第一个帧出现超时，那么回退到这个帧重新发送所有的帧。

而因为接收窗口大小为 1，除了数据链路层必须要递交给网络层的下一帧外，接收方拒绝接受任何帧。如果在计时器超时以前，发送方的窗口已被填满，则管道将变为空闲，最终，发送方将超时，并且按照顺序重传所有未被确认的帧，从那个受损或者丢失的帧开始。

一般使用累计确认。



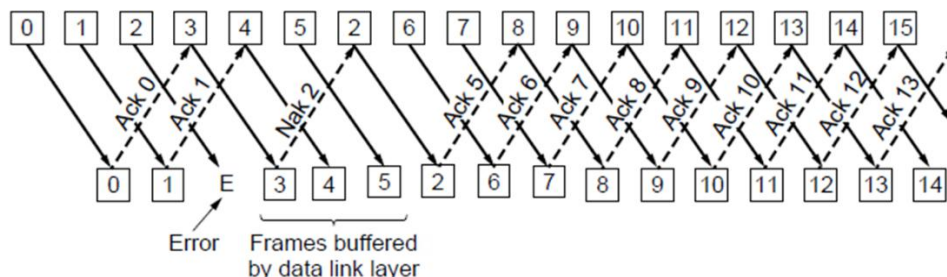
3.4.4 选择重传协议（协议 6）

（发送窗口大小 >1 ，接收窗口大小 >1 ）

发送方按照窗口的顺序依次发送帧给接收方，而接收方检查该帧是否可以落在接收窗口内，即以前没有接收过且该序号在窗口可接收的范围内，如果可以那么不管这一帧是否为网络层所期望的下一个数据包都接收该帧并暂存于缓冲区内。该帧会一直保存在数据链路层中直到所有序号比它小的帧已经按顺序递交给网络层，它才能被传递给网络层。

辅助计时器：在发送方的计时器超时之前，没有出现需要发送的反向流量，则发送一个单独的确认帧，而发送的时间间隔由辅助计时器决定，因此辅助计时器的超时间隔应该明显短于数据帧关联的计时器的间隔。

除 ACK，选择重传还有否定确认，当接收方发现坏帧丢弃时，立刻发送一个 NAK 给发送方告知该帧未收到让它重传，以防超时。

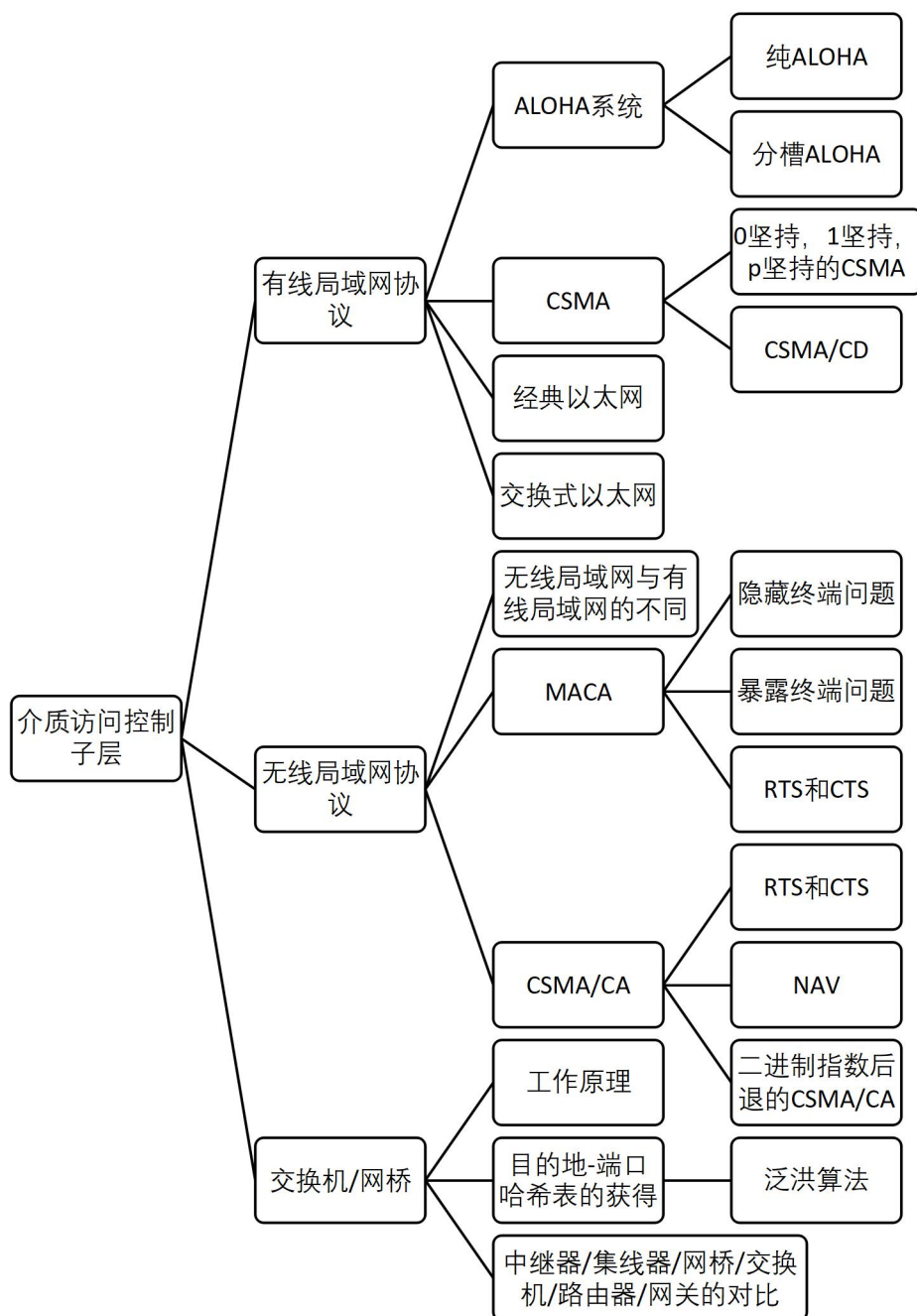


选择重传和回退 N 实际是带宽使用效率与数据链路层缓存空间之间的权衡。

第四章 介质访问控制子层

作者 孙吉鹏

Medium Access Control, MAC 子层,注意这章的标题是“子层”，也就是说这一章的内容还是属于数据链路层，是它的一个子层，虽然出现的章节比第三章要晚，但实际上却是整个数据链路层的底层。因为在同一介质中相同频率波段的传输信号会相互干扰，导致无法得到有效的信号，而采用频率，时间区别不同信号又无法最大限度地利用资源，这时候就希望有种协议可以协商三个以上的机器如何可以在没有统一调度的前提下“遵守秩序”“不打断别人”地发言，让介质顺利被利用不产生冲突。第三章的两点之间的可靠传输是后话，因为你先要确定到底是哪个点可以获得信道开始传输啊！所以说第四章内容是数据链路层的底层。



4.1 有线局域网协议

4.1.1 ALOHA 系统

1. 纯 ALOHA

用户有数据就发送，不考虑信道是否空闲，只管发。用户如果发现有冲突，则随机等待一段时间后再发。它的意义在于提出了非协调用户竞争使用单个共享信道的系统的问题，现实中已不使用，具体的讨论在中文 P203。

2. 分槽 ALOHA

为了提高发送的容量，将时间分成了离散的间隔，称为时间槽，用户只能在槽边界发数据，减少了冲突期，将吞吐量提升了一倍。详细见 P204，P205

4.1.2 CSMA

载波检测多路访问（Carrier Sense Multiple Access）

指每个站监听线路上是否有载波（传输），并据此采取相应的动作的方式。相比 ALOHA 的先发后听，这是先听后说，大大地提高了利用率。

1. 1 坚持，0 坚持，p 坚持的 CSMA

1 坚持：当一个站有数据要发送时，它首先侦听信道，确定当时是否有其他站正在传输数据；如果信道空闲，就发送数据，如果信道忙，该站持续坚持监听信道，直到空闲，一旦监听到空闲，则站立即发送一帧。如果发生冲突，该站等待一段随机的时间，然后从头开始上述过程。

0 坚持：如 1 坚持一样监听信道如果空闲则发，如果在使用中则放弃监听，等待一段随机时间后再重复上述算法，比较“随缘”。

P 坚持：以概率 P 发送数据，以概率 $q=1-P$ 将此次发送推迟到下一个时间槽发送。如果下一个时间槽也是空闲的，则它还是以概率 P 发送数据，或以概率 q 再次推迟发送。该过程一直重复，直到帧被发送出去，或者另一个站开始发送数据。属于两种方式的改良。

直觉上保证了“有礼貌”地在信道上不打断其他站的传输且只在空闲的时候传输就可以解决多路访问的冲突问题，但是这三个协议都没有解决在多个站同时监听到空闲后都发送时会导致的信号混杂从而冲突的情形，如果有协议保证站可以迅速监听到冲突后立即停止，则可以最大程度上节省了注定失败的传输所占用信道的时间和带宽，这种协议就是下面的带冲突检测的 CSMA，也是现在以太网实际采用的协议。

2. CSMA/CD

带冲突检测的 CSMA（CSMA with Collision Detection）P207，P208

不止在发前监听信道，在信号发送过程中发送方也要持续监听信道，如果发现线路上的信号与自己发送的信号不一致（即产生了冲突），则立刻停止传输信息，发送一段 48bit 的阻塞信号告诉与自己发生冲突的站不要漏检这次冲突，之后它等待一段时间再发送。

如何确保发送方意识到冲突是这个协议着重解决的问题，也就是发送时间需要足够长，保证即使是另一端的产生冲突的信号传过来自己也还在发，不至于发完了不再监听信道后不知道自己这次的传输已经失败了。也就是说需要至少发满信号跑一个来回的时间，也就要求发送的帧要够长，在以太网里，我们算出这个长度是 64 个字节。

3. 经典以太网 P217——P220

经典以太网 MAC 子层的帧格式 P218

Bytes	8	6	6	2	0-1500	0-46	4
Ethernet (DIX)	Preamble	Destination address	Source address	Type	Data	Pad	Check-sum
IEEE 802.3	Preamble	SOF	Destination address	Source address	Length	Data	Pad
							Check-sum

具体字段内容解释见课本 P218——P220，注意由于冲突检测要求有效数据必须大于 64 个字节，所以不算前导码的 8 位，用户 Data 段为 0 时，此时帧的最短长度只有 18 个字节，所以存在最长为 46 字节的填充位（18+46 = 64）。

4. 交换式以太网 P222——P224

以交换机为核心设备而建立起来的一种高速网络。可在高速与低速网络间转换，实现不同网络的协同。交换机分割了连接的不同网络成为不同的冲突域，不同冲突域之间信号不会彼此干扰，所以不用考虑不同自治域间的冲突问题。

4.2 无线局域网协议

4.2.1 无线局域网与有线局域网的不同 P214

①两者的传输介质有着本质区别，也正是这种区别，导致 WLAN 存在新的问题：隐藏终端问题和暴露终端问题。

②两者传输范围有区别：WLAN 中，无线电传输范围有限，一个站不能给所有其他站发送帧，也无法接收来自所有其他站的帧；在有线局域网中一个站发出一帧，所有其他站都能接收到。

③信道检测方式不同：WLAN 采用能量检测、载波检测和能量载波混合检测三种检测信道空闲的方式；以太网通过电缆中电压的变化来检测。

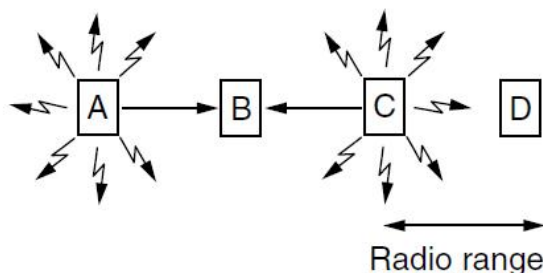
④在 WLAN 中，对某个节点来说，其刚刚发出的信号强度要远高于来自其他节点的信号强度，也就是说它自己的信号会把其他的信号给覆盖掉，但在本节点处有冲突并不意味着在接收节点处就有冲突。

4.2.2 MACA 冲突避免多路访问

（Multiple Access with Collision Avoidance）

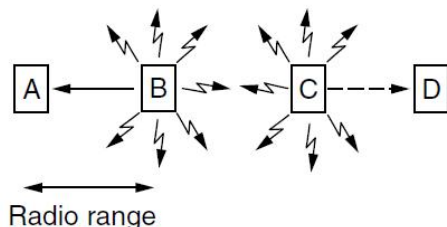
1. 隐藏终端问题 P215

A 给 B 发，但 C 的信号接收覆盖范围没法覆盖到 A，导致不知道 A 在给 B 发，误以为 B 空闲，一旦发送，则信号会冲突干扰，导致失效。可以看到，问题的关键在于 C 无法知道 B 的接收情况，因为之前的监听是在发送方 C 自己这里进行的。这个问题是致命的。



2. 暴露终端问题 P215

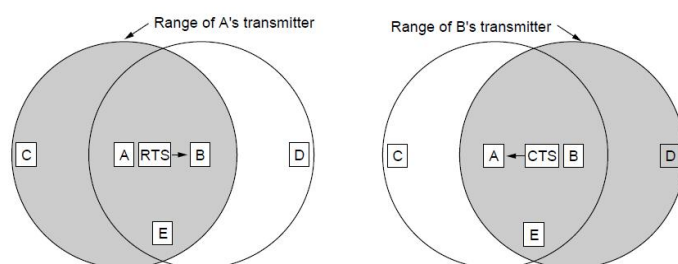
B 给 A 发，此时 C 想要发送信号给 D，但 C 监听到介质上有信号传输，则会等待 B 传输结束再进行给 D 的传输，但实际上这种等待是不必要的因为 C 即使发给 D 也不会干扰 A 的接收信号，因为干扰只在接收方端产生，所以这种问题带来了效率的降低，根源还是在于 C 不知道接收方 A 的情况，只知道自己附近发送方 B 的情况。但注意并不是致命的，因为还是没有破坏传输。



3. RTS (Request To Send) 和 CTS (Clear To Send) P215

为了解决之前说的两个问题，即之前的 CSMA 只能监听到发送方附近而不是接收方附近介质情况根本问题，希望能通过协议让接收方也“发言”，即通过这次信号让接收方附近的站点也能感受到接收方的存在，从而避免后续的继续发送造成该接收方的冲突。

解决的方案就是发送方先发 RTS，之后接收方回 CTS 信号，并从这个信号中包含这次传输的持续时长信息保证让此时长内接收方附近站点主动静默，从而不会使其受到干扰。



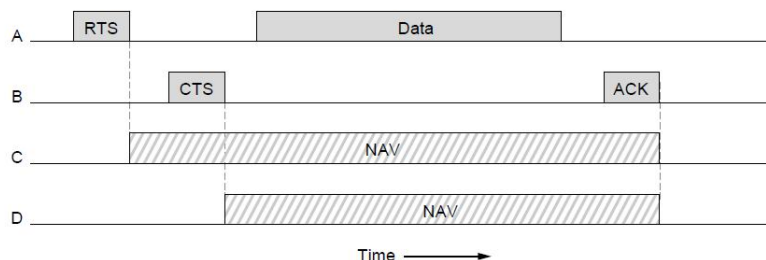
即一个点如果只听到 RTS，没有听到 CTS，说明它在发送方的发送范围内，却不在接收方的发送范围内，则它只要不干扰发送方收 CTS，即可随便发送。解决了暴露终端问题。

一个点如果只要听到 CTS 说明它就在接收方发送范围内，此时它要保持静默到持续时长，直到此次传输结束它再进行发送，否则会干扰接收方接收。解决了隐藏终端问题。

4.2.3 CSMA/CA

带有冲突避免的 CSMA (CSMA with Collision Avoidance) P234—P236

它是 802.11 MAC 子层协议的核心协议，与之前的 MACA 相比，它引入了短确认确保每一帧的发送成功，即数据发送后站启动确认计时器，如果计时器时间到但没有收到接收方回复的收到的确认，则试图重新发送。但在 RTS 和 CTS 方面并没有考虑暴露终端问题。



1. RTS 和 CTS

与 MACA 类似，但是将规则改成了听到 RTS 后也停止传输一切东西，直到此次数据传输结束后再进行传输。这样其实是无法解决暴露终端问题的，但由于考虑到暴露终端问题是效

率问题而不是致命问题，且处理这个不经常发生的问题需要耗费操作时间，所以就进行了舍弃。

2. NAV 网络分配向量 (Network Allocation Vector) P236

每个站保留的信道何时要用的逻辑记录，每个帧携带一个 NAV 字段，说明这个帧所属的一系列数据将传输多长时间。所有听到数据帧的站将在发送确认期间推迟发送，不管能否真正听到确认的发送。

3. 二进制指数后退的 CSMA/CA P220, P234

侦听很短的一段时间后发现没有信号，则随机选择 0-15 个时间槽进行倒计时倒数，当听到有帧发送时暂停倒计时，空闲时计数，到 0 时就发送，如果发送成功则目标站会发送一个短确认，如果没收到确认，则发送方加倍自己选择的时间槽数，重新试图发送。如此反复，直到成功发送帧或达到最大重传次数。

4.3 交换机（网桥）P257—P260

4.3.1 工作原理

网桥工作在数据链路层，将多个 LAN 连接起来，通过检查数据链路层地址来转发帧。

- ①当一帧到达时，网桥必须决定是将该帧转发还是丢弃
- ②如果决定转发，还必须要决定在哪个端口传输帧
- ③网桥通过在其内部配备一个大的(哈希)表来查询一帧的目的地址，该表中列出了每一个可能的目的地址以及它隶属的输出端口
- ④当网桥第一次被接入网络时，所有的哈希表都是空的，网桥使用洪泛算法完善哈希表。
- ⑤其具体转发过程为：
 - a. 目的地址的端口与源端口相同，则丢弃该帧
 - b. 目的地址的端口与源端口不同，则转发该帧到目的端口
 - c. 目标地址端口未知，则使用洪泛算法，将帧发送到所有的端口，除了他入境的那个。

4.3.2 目的地-端口哈希表（转发表）的获得

泛洪算法：不需要知道网络的拓扑结构和相关的路由计算，仅要求接收到信息的节点以广播的形式转发数据包。

对于每个发向未知目的地址的入境帧，网桥将他输送到所有的端口，除了它到来的那个端口，慢慢的网桥学习到目标地址在哪里)和向后学习法(通过检查每个窗口上发送的所有帧的源地址，网桥就可获知通过那个窗口能访问到哪些机器。

4.3.3 中继器/集线器/网桥/交换机/路由器/网关的对比 P263—P264

中继器：物理层，模拟设备，用于连接两根电缆段，放大信号。

集线器：物理层，有许多输入线路，它将这些输入线路连接起来，在任何一条线路上到达的帧都被发送到其他线路上。

交换机：数据链路层，多端口的网桥。根据帧的目的地址转发，常被用来连接独立计算机。

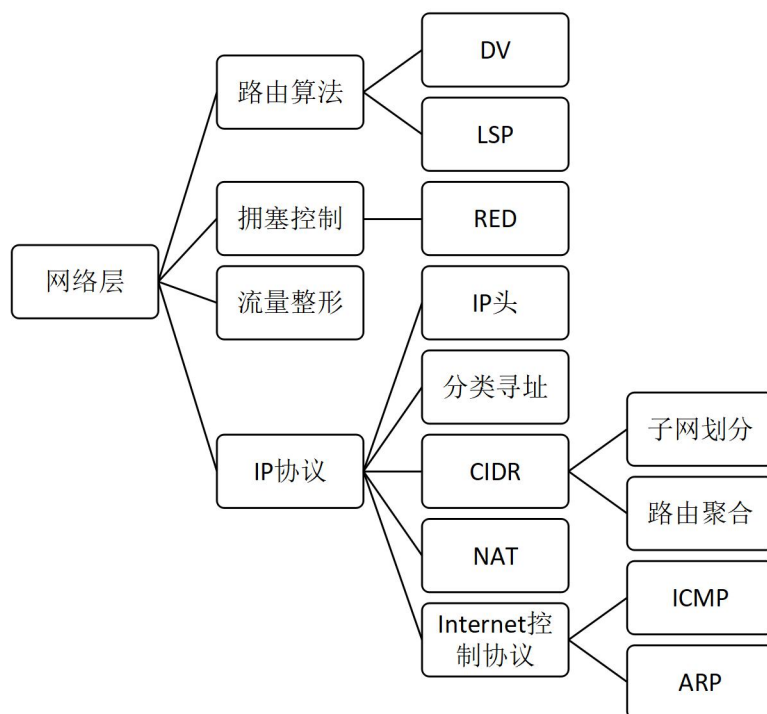
路由器：网络层，当一个分组进入到一个路由器中的时候，帧头和帧尾被剥掉，位于帧的 IP 分组被传递给路由软件，路由软件利用分组的头信息来选择一条输出线路。

网关：传输层，应用层。应用网关是将一个网络与另一个网络进行相互连通，提供特定应用的网际间设备，应用网关必须能实现相应的应用协议。应用网关可设在应用层或传输层。设在应用层的叫应用层网关，也称代理服务器。设在传输层的叫传输层网关。

第五章网络层

作者 林子童 杜泽林

数据链路层保证了数据在相邻节点的可靠传输，网络层关注的是如何将源端数据包经过网络上的节点一路送到接收方。为了实现这个目标，网络层必须知道网络拓补结构，并从中选出适当的路径。本章包含到路由算法、拥塞控制、服务质量、网络互连和 IP 协议。



5.1 网络层两类服务

（中文课本 P276—P279，注意图 5-4）

（1）无连接服务——数据报网络

特点：数据包独立路由。

（2）面向连接服务——虚电路网络

特点：发送数据报前，首先建立一条从源到目标的路径，每条报文都沿这条路径传送。

5.2 路由算法——基于最优路径的路由算法

5.2.1 最优化原则（P281）

遵循最优化原则以破除环路，方便设计路由算法。

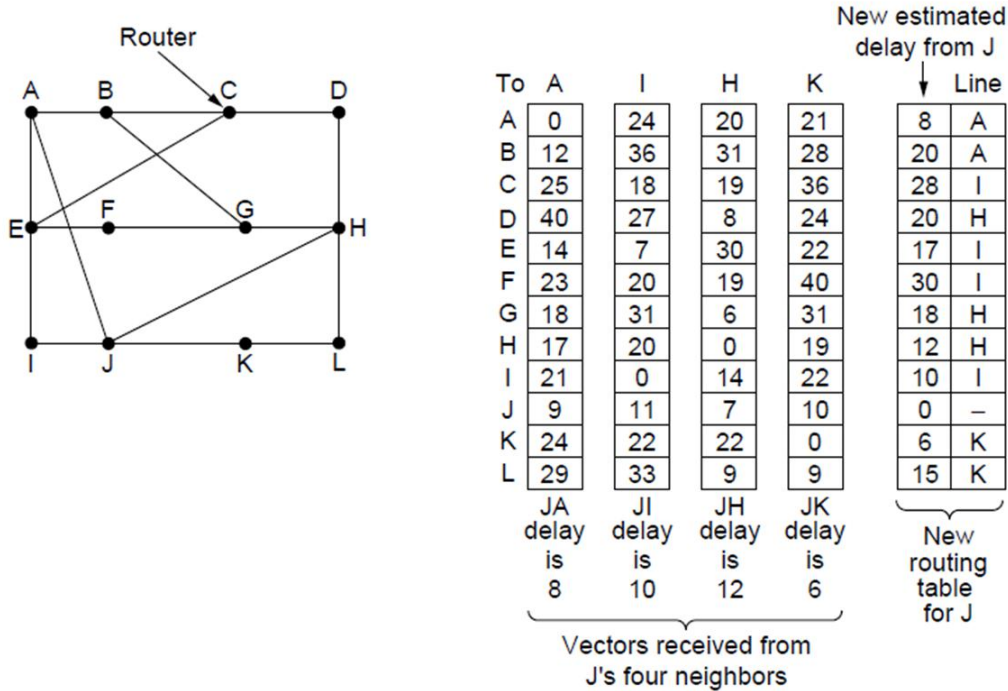
最优化原则：如果 J 在从 I 到 K 的最优路径上，那么从 J 到 K 的最优路径也必定遵循相同的路由。即最优路径的子路径还是最优路径。

汇集树：依照最优化原则，从所有的源到一个指定目标的最优路径的集合构成一颗以目标节点为根的树。

5.2.2 泛洪算法（P284）

泛洪路由的基本想法是源节点将消息以分组的形式发给其相邻的节点，相邻的节点再转发给它们的相邻节点，继续下去，直至分组到达网络中所有的节点。

5.2.3 距离矢量路由算法 DV (RIP 协议)



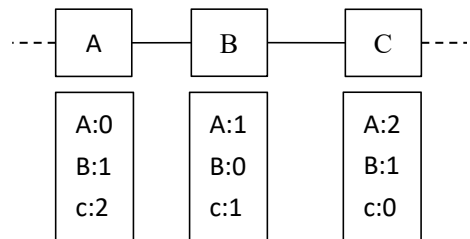
每个路由器维护一张表，表中列出了当前已知的到每个目标的最短距离，以及所使用的线路的下一跳，通过在邻居之间相互交换信息，路由器不断地更新它们的内部表，最终每个路由器都了解到达目的地的最佳链路。届时就可以根据路由表和目的地址，在任意路由上确定下一跳进行跳转。

路由表的具体维护方式：当前点与所有邻居节点交换的路由表，然后根据这些表计算该点去所有其他点的最优下一跳 $NH = \arg_{i \in Adj(A)} \{cost(A,i) + cost(i,B)\}$ 。

如上面的例子所示，当求点 J 到点 C 的最短距离和下一跳时，分别求出 $Ji+iC$ 的最小值，即 $\min\{8+25, 10+18, 12+19, 6+36\}=28$ ，最短路径是经过 I，时长 28。

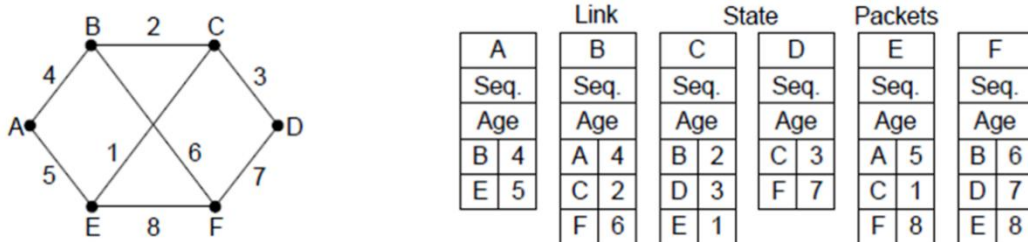
无穷计数问题：“坏消息”传播的很慢，可以从下面的例子进行理解

如图所示，ABC 各自维持自己的路由表，但如果 AB 之间的线路突然断开，那么 B 和 C 在计算各自与 A 的距离的时候，就会互为参考，每次路由表更新的时候计算出的与 C 的距离只会加 1 直到数值趋近于无穷，所以 AB 之间的断开很慢被发现。



5.2.4 链路状态路由算法 LSP (OSPF、IS-IS 协议)

每个节点都将自己的邻居信息表传给所有节点，每个节点都保存所有节点的邻居信息表，并使用 Dijkstra 计算出本节点到其他所有节点的最短路。



上面可以看到，邻居信息表中都包含两个字段：Seq 和 Age。

Seq:由于邻居信息表需要泛洪给所有节点，为了控制泛洪的规模，所以使用 Seq 序号。新的链路状态数据包到达一个节点的时候，路由器检查这个新来的数据包的序号和源路由器是否已经出现在自己的列表中。重复则丢弃，源路由器相同而序号过时则拒绝接受。

Age:在数据包发出时设定初值，然后每一秒都会减一，变为 0 则会被丢弃，①可以在泛洪时防止数据包无限生存②及时清理路由器中链路状态数据库中旧的或无效的信息，以防以下问题的影响，如：路由器崩溃后重启序号从 0 开始的话会被当做过时信息丢弃；传递过程中序号出现差错如 4 变 65540，那么只有 seq 字段就会导致后面的 5-65539 全被拒绝。

算法步骤：

- ①发现邻居节点，并知道其网络地址
- ②测量到各邻居节点的延迟或开销
- ③构造一个分组，分组中包含所有它刚知道的信息
- ④将这个分组发送给所有其他的路由器
- ⑤计算出到每一个其他路由器的最短路径

*务必重点掌握距离矢量算法和链路状态算法，详见中文课本 P285—P291

因为距离矢量算法每个节点只与邻居节点交换信息，在面临路由器失效时会出现无穷计数问题。链路状态算法用泛洪的方式分发链路状态包，所以每个节点都知道完整的网络加权拓补图，重点在链路状态包的构造和分发方法以及健壮性。

5.2.5 层次路由

随着网络规模的增长，路由器的路由表也成比例增长，结果是路由器效率下降，成为网络服务的瓶颈。解决之道是将网络分层，采用分层路由之后，路由器被划分成区域，每个路由器知道如何将数据包路由到自己所在区域内的目标地址，但是对于其他区域的内部结构毫不知情。

可以认为每个区域是一个村，普通村民知道如何与本区域所有其他人联系，而村长除此之外还知道如何与别的村的村长联系，村民与外村人联络时会先将信交给村长由村长转发，他负责本区域与其他区域的联络。后面 IP 协议的子网划分中的分类寻址就是层次化的设计。

5.3 拥塞控制

主要掌握 RED 协议，完整的流量控制应该是网络层（RED 协议）和传输层（TCP 慢启动）配合的任务。

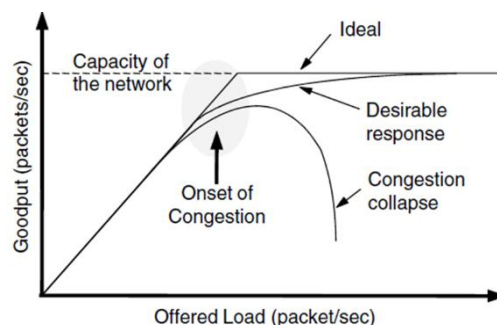
（1）拥塞：网络中存在太多数据包导致数据包被延迟和丢失，从而降低了传输性能。

当拥塞出现后，有可能遭遇拥塞崩溃。

（2）随机早期检测（RED）（P310）

当某条链路上的平均队列长度超过某个阈值时，该链路就被认为即将拥塞，因此路由器随机丢弃一小部分数据包。

当拥塞时，若路由器向发送方发抑制包，那么大量的抑制包反而会加重拥塞。所以网络层解决拥塞的思路就是防患于未然，在局面变得毫无希望前让路由器舍弃负担。后面配合传输层可以降低用户发送速率，从根本上解决拥塞。



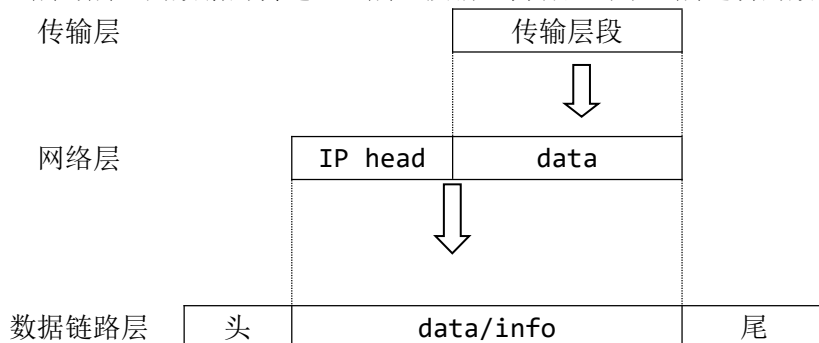
5.4 流量整形——平滑的流量更好管理

流量整形（traffic shaping）是指调节进入网络的数据流的平均速率和突发性所采用的技术，包括漏桶和令牌桶（P314）

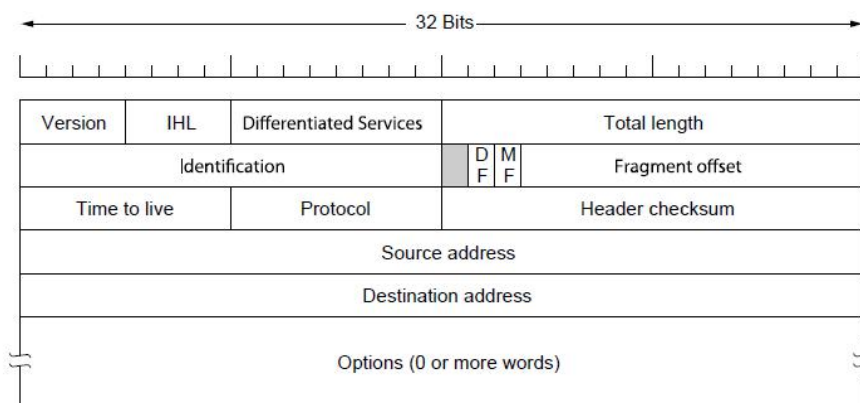
- (1) 漏桶算法：无论注入桶内的速率是大是小，出桶的速率总是恒定的。
- (2) 令牌桶算法：桶内存放发送数据的令牌，每单位时间获得一定量的令牌，发送数据时取出令牌，流量大小受限于积累的令牌数量。一般令牌桶下面会有一个漏桶用于平滑发送速率。

5.5 IPv4 协议

层与层之间数据的传递：上层整段消息内容是对下一层透明的数据



5.5.1 IP 头



IP 协议 Internet Protocol 提供尽最大努力的数据传输，不保证可靠

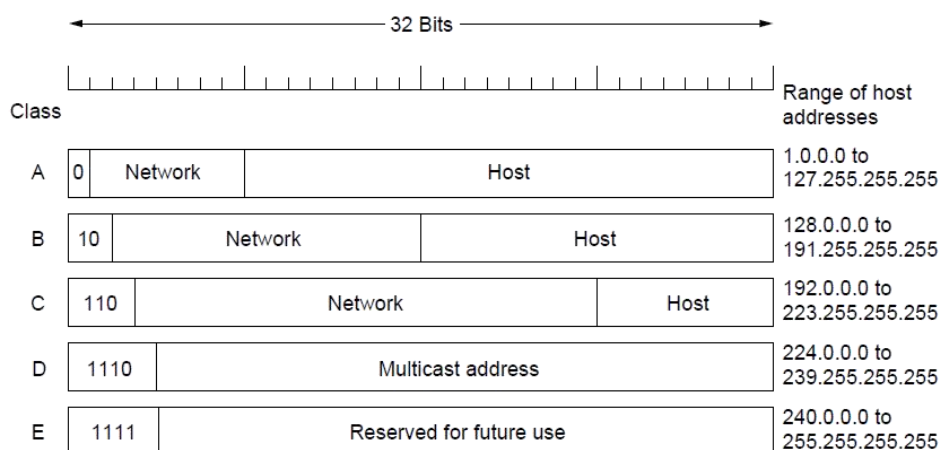
IP 数据报由头和正文组成，正文是有效净荷，携带数据；头由 20 字节定长及一个可选变长组成，携带此数据报的解释信息。下面是 IP 头的相关介绍。

- 1) 版本：记录数据报属于什么版本，Ipv4 Ipv6
- 2) IHL：IP 头长，单位为 4 字节，因为头部至少 20 字节定长，所以值域 5~15。
- 3) 区分服务：前 6 位标识数据报服务类型，后 2 位携带显示拥塞信息，了解。
- 4) 总长度：数据报总长度，单位是 1 字节 (8bit)，包括头和数据，最大长度 2^{16} 字节
- 5) 生存期 TTL：计数单位为跳数，每经过一跳减一，递减到 0 的时候数据包被丢弃并由路由器给源地址发送一个报警包。设定目的主要解决环路问题，避免数据包被永远都留在网路中。不同协议的技术单位不同，有的还设定为秒，这是为了控制拥塞，但是秒不好实现。
- 6) 协议：记录 IP 分组携带数据的协议类型，即传输层协议的编号，如 TCP、UDP 等
- 7) 头校验和：对 IP 头信息进行校验，检测数据包穿过网络时是否发生错误。但是由于只校验了头，所以 IP 数据报整体依旧不可靠
- 8) 源地址和目的地址：源地址可以用于丢包时给源地址发消息；目的地址用于寻找路径时查路由表
- 9) 选项：给出对寻址过程中要求必须经过的路由器，如今，几乎已经不再使用 IP 选项
- 10) IP 头第二行都是关于分段的处理，其用途就是：不同网络的最大帧长的要求可能不

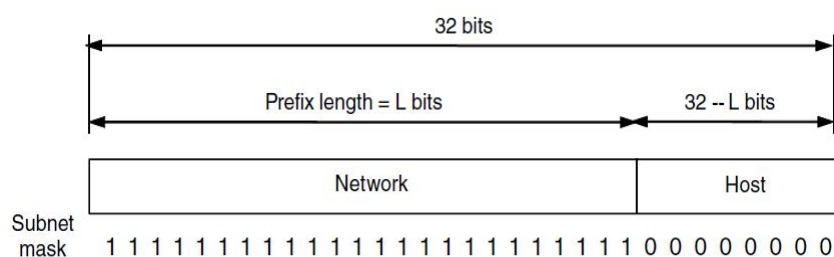
同，从而大数据进入小网络需要对大的数据进行分段。标识表示这段分段属于哪个数据报：**DF** 表示是否允许分段，**DF=1** 表示该数据报不允许分段，到达需要分段的地方就丢包返回消息；**MF** 表示所属于的数据报是否还有更多的段。**MF=0** 表示当前分段是该数据报的最后一段；分段偏移量表示该分段在整个数据包中的位置，以 8 字节为单位。具体计算此处不再详细介绍，这一行了解即可。

5.5.2 分类寻址 Classful Addressing

IP 地址被分为五类。如 A 类允许 2^7 个网络，每个网络中允许有 2^{24} 台主机，由于每类网络中允许的主机数量是固定的，所以造成了 IP 地址极大的浪费，这才引入了后面的 CIDR



5.5.3 子网 Subnet 与前缀 prefix



1) IP 地址长 32 位，由高位的可变长网络号和低位的主机号组成，同一网络上所有主机的网络号相同，一个网络对应一块连续的 IP 地址空间，这块地址空间就被称为地址的前缀

2) 写法：点分十进制，如 18.0.31.0/24，24 表示网络号位数。子网掩码是网络号长度个 1 和主机号长度个 0 组成，子网掩码和 IP 按位与可以得到网络号

3) 子网划分 **subnetting**：在内部将一个网络块分成几个部分供多个内部网络使用，但对外部世界仍然像是单个网络一样。而第一章的子网 **subnet** 是分割一个大型网络得到的一系列结果网络，是指网络中所有路由器和通信线路的集合。

4) 转发过程：数据报到达后，路由器检查该数据包的目的地址，将该目的地址与路由表中每个子网目标的子网掩码按位与看是否匹配

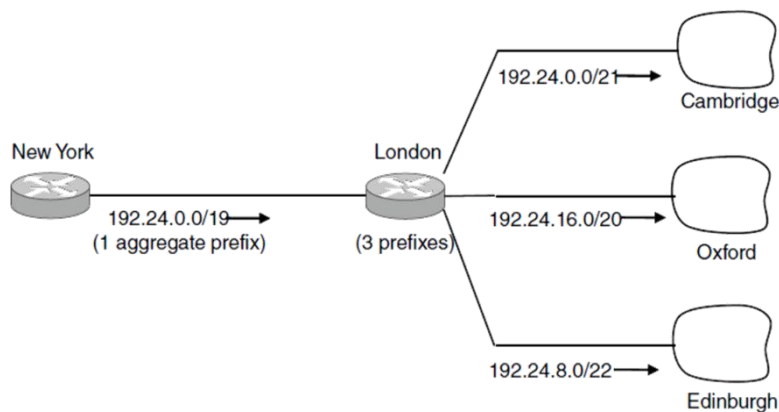
5.5.4 无类域间路由 CIDR

Classless Inter-Domain Routing，是为了解决 IP 路由表爆炸、IP 地址耗尽而提出的一种措施。在变长子网掩码的基础上提出的一种消除 ABCDE 类网络划分，并且可以在软

件的支持下实现构造超网的一种 IP 地址的划分方法。

子网划分与聚合的一个实例：以下页图为例

用一块从 194.24.0.0 开始的大小为 2^{13} 的地址分配三个大学的地址。计算过程如下。其中每个地点的两行分别代表该地的起始和终止 IP 地址。竖虚线为网络号和主机号划分。



起 始	194	24	0	0	地址数
	1 1 0 0 0 0 0 0	0 0 0 1 1 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	2^{13}
剑 桥	1 1 0 0 0 0 0 0	0 0 0 1 1 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	2^{11}
			0 0 0 0 0 1 1 1	1 1 1 1 1 1 1 1	
爱 丁 堡	1 1 0 0 0 0 0 0	0 0 0 1 1 0 0 0	0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0	2^{10}
			0 0 0 0 0 0 1 1	1 1 1 1 1 1 1 1	
空	1 1 0 0 0 0 0 0	0 0 0 1 1 0 0 0			2^{10}
牛 津	1 1 0 0 0 0 0 0	0 0 0 1 1 0 0 0	0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0	2^{12}
			0 0 0 1 1 1 1 1	1 1 1 1 1 1 1 1	

分配的原则是：一个网路只能有一个网络号，进行 IP 地址分配时尽可能连续分配。以牛津大学为例，尽管前面余下 2^{10} 个空白，但是依旧选择向后

路由聚合 route aggregation: 为了减小路由表长度，将多个小前缀地址块合并为一个前缀地址块，如上面的三个地址就可以聚合成一个地址，方法是将三者网络号最大公共部分聚合为新网络号

194.21.0.0/21

194.24.8.0/22 → 194.24.0.0/19

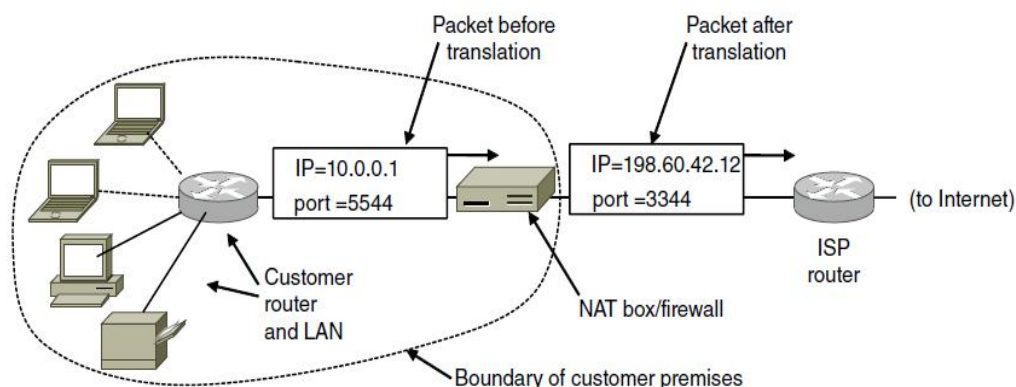
194.24.16.0/20

如此，就可以将三个路由表项合成一个，其下一跳是伦敦。但由此带来一个问题：如果上面空白的那部分被分配给非洲某学校，那么空白部分就会被错误的聚合。因此要在路由表中新增一项单独指明空白部分的下一跳。

所以，在没有聚合时，查路由只需要找到匹配的一项就可以跳到下一跳，但是有聚合以后，需要寻找路由表项中的最长匹配

路由器的转发过程完善如下：当新的消息进入路由器时，首先进入等待队列，通过一定的调度策略进行调度。调度到这个消息时，获取其目的地址，将目的地址分别与路由表中的每一项网络号的子网掩码进行比对，选取最长匹配的网络表项进行转发。当然，没有查询到匹配的时候，转发到缺省表项，也就是给上一层路由，继续寻找。

5.5.5 NAT 网络地址转换



- 1) IP 地址短缺问题解决策略：①动态分配 IP ②迁移到 IPv6 ③多台共用一个 IP
- 2) NAT(Network Address Translation)，它的思想是设定两套 IP 地址，内网相对于外网来说共用一个 **public** 地址，而在内网中，每台机器对应一个 **private** 地址
- 3) 内网之间的通信使用 **private** 地址，当想要向外网发送消息时，只需将源地址替换为内网共用的 **public** 地址
- 4) 当外网向内网发送消息时，因无法区分内网的主机，所以引入了 **port**。将私有 IP 与端口号影射成新的 **port**，当与这个端口号交互时，根据影射算法可以知道私有 IP
- 5) 缺点：①违反了最基本的协议分层原则，传输层的数据不再对网络层透明 ②私有 IP 与 **port** 对公有 **port** 的映射关系不是一一对应的③违反 IP 唯一性原则。等

5.5.6 隧道技术

隧道技术 (**tunneling**)：是一种通过使用互联网络的基础设施在不同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，到达对方以后再将包裹的数据提取出来进行传递。新的帧头提供路由信息，以便通过互联网传递被封装的负载数据。

简言之，就是处理不同网络相连接的情形，隧道技术对于两头网络相同中间不同的特例有效。它的方法就是在经过中间网络的时候加一个新的 IP 头，度过以后再拆掉

5.6 Internet 控制协议

5.6.1 ICMP 控制消息协议

为了提高 IP 数据报交付成功的机会，在网络层使用了网络控制报文协议来允许主机或者路由器来报告差错和异常情况。了解即可，详见 P358。ICMP 是通过向数据包的源地址报告有关事件使网络运行正常。

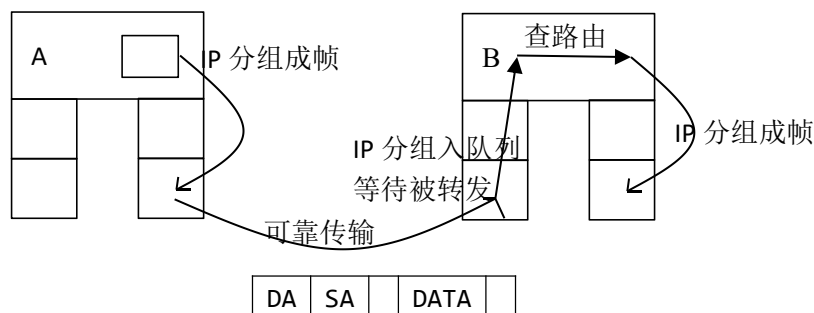
5.6.2 ARP

Address Resolution Protocol，地址解析协议

是一个连接二三层的协议，可以说是 IP 分组-帧连接协议

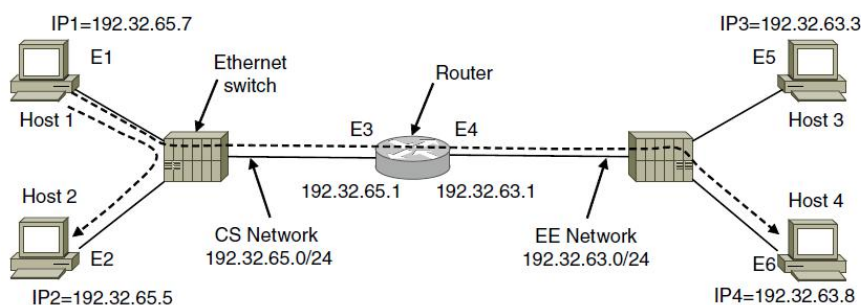
某一层的地址只在本层有效，网络层负责寻找路径，而真正实现消息传递的是链路层，所以要实现两层地址之间的映射。

- 1) 路由器之间消息的传递



现在的关键在于已知源 IP、源 MAC 和目的 IP 的情况下怎样获取目的 MAC

2) 交换机原理



在上图中的交换机中存储着一张 MAC 和 port 的对应表，交换机的一个端口和一台主机一一对应。只有发送过消息的端口的对应关系才会被交换机所知道，而一个广播能同时将一个子网的全部对应关系都传送给交换机

3) 同一子网中消息的传递

如上，主机 1 欲给主机 2 发消息，应首先经过如下两个过程：

①主机 1 ARP 广播，广播只能在一个子网内传递，无法穿透路由器。在这里，E3 将自己看做一个普通的主机

DA	SA	type	data	
48 个 1	E1	ARP 广播	IP=IP2, MAC=?	

②主机 2 ARP 应答（单播）

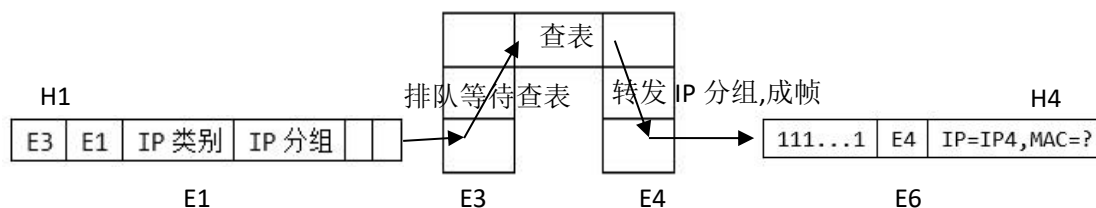
DA	SA	type	data	
E1	E2	ARP 应答	IP=IP2, MAC=E2	

4) 不同子网消息的传递

①与同一子网①的工作相同、

②E3 作为通往外网的代理分析广播的接收方是否在这个子网中。若目的 IP 与 E3 所在网络号相同那么 E3 就当做一个普通主机，否则提交到路由器的网络层，而 E3 则作为这条消息的代理与网外交互。

而在路由表中找到对方所属的子网以后，在该子网中通过 ARP 广播找到该 IP



第六章 传输层

作者：袁郭苑

写在前面：

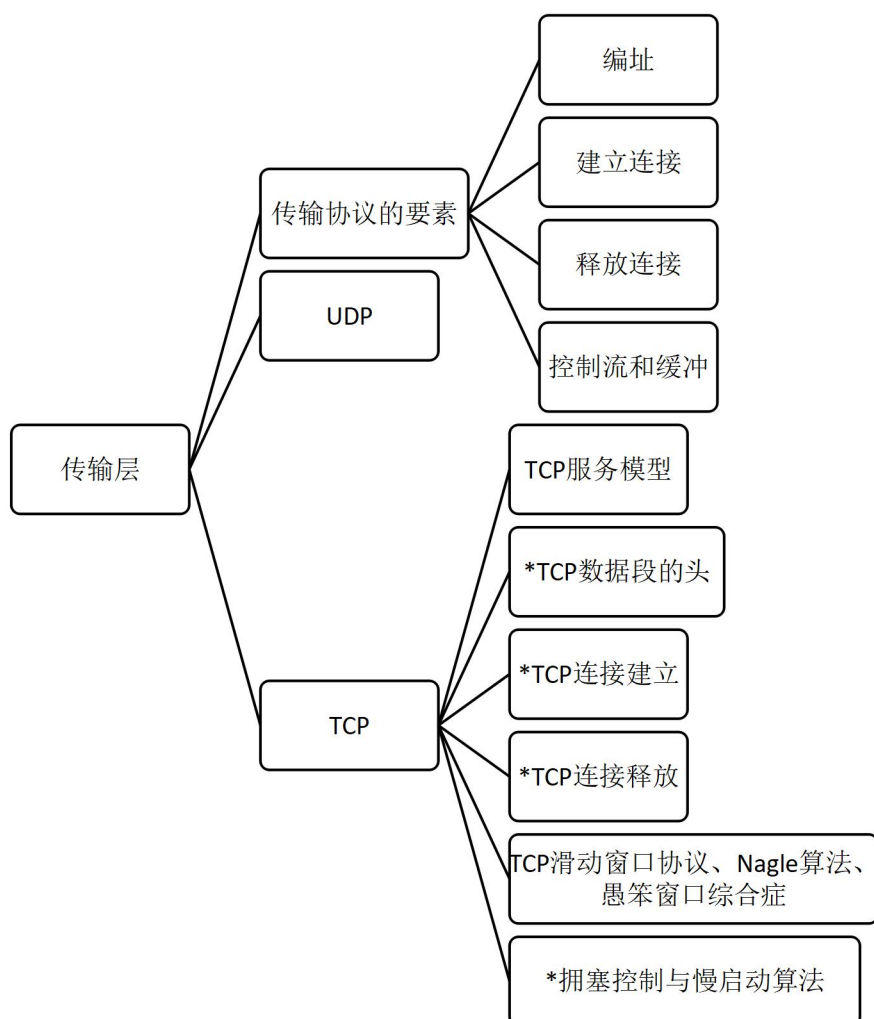
朋友们大家好，经过前面对物理层、数据链路层、网络层的学习，我们明白了如何实现点到点的可靠的传输，明白了网络中的一系列路由算法，走进传输层，我们首先要明白这一层实现的是端到端的可靠的传输，是基于不可靠的网络层之上的。

为什么之前为确保可靠而使用的超时重传和确认机制在这里会导致不可靠呢？这是因为与简单的主线（Bus）相比，我们这里所处于的是网络之中，网络里的节点是具备**缓存能力**的，由此可能导致分组的滞留，而简单的超时重传等机制产生的重复分组会在网络通信中引发很大的问题。基于这个现状，我们智慧的前辈们想出利用选号和三次握手机制建立和拆除 TCP 连接的方法克服了网络中重复分组引发的问题，在不可靠的 IP 层之上实现的可靠的数据传输协议 TCP。

本章还涉及了 UDP 协议的相关内容，与 TCP 面向连接、可靠的特点相比较，UDP 是一个无连接的、不可靠的传输层协议。

由于这次的总结整理时间仓促，所以对于一些简单知识内容，我为其标注了章节号，大家可以在课本上详细查阅，对于重点内容，我像往常一样为大家进行了详细的分析。希望大家从中有所收获。

知识框图：



6.1 传输协议的要素

6.1.1 建立连接

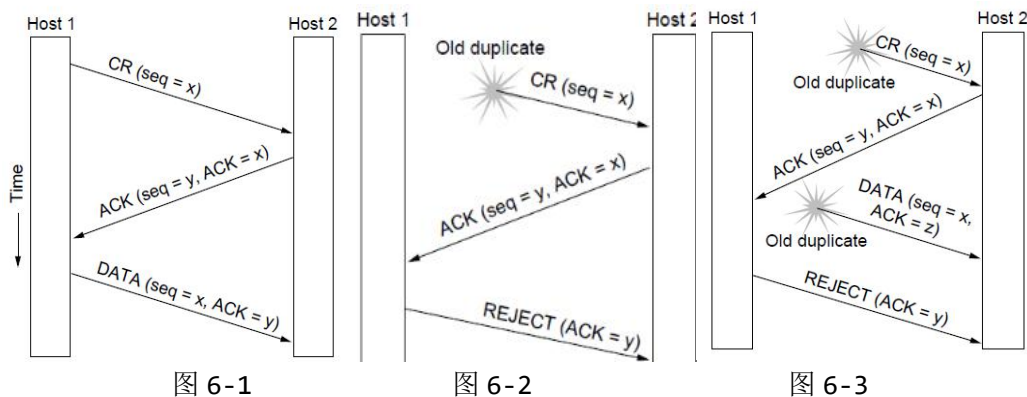


图 6-1 是正常的三次握手建立连接的过程。（此图中的 ack 值为下一次想要接收的第一个字节编号减一所得，下图亦然）。

图 6-2 这种情况是老的 CR (Connection Request) 重复分组出现了，它虽然引起了主机 2 发送相应的分组，但是主机 1 根据主机 2 发送分组的 ack 值可以发现这是异常情况，所以拒绝 (REJECT)。

图 6-3 这种情况是老的 CR (Connection Request) 重复分组和老的数据重复分组出现的情况。虽然老的 CR 重复分组引起了主机 2 发送相应的分组，但是主机 1 根据主机 2 发送分组的 ack 值可以发现这是异常情况，产生拒绝 (REJECT)；对于数据重复分组，主机 2 根据分组的 ack 值可以发现这是异常情况。

6.1.2 断开连接

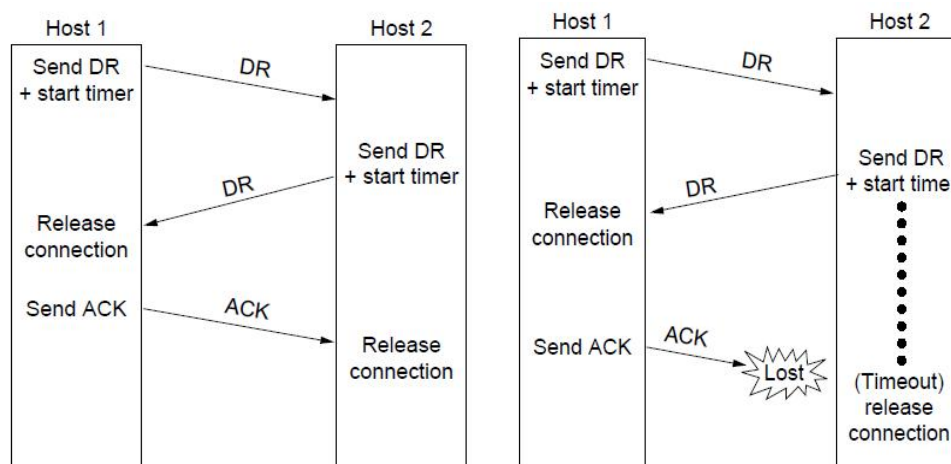


图 6-4 是正常的三次握手断开连接的过程。

图 6-5 这种情况是最后主机 1 发出的 ack 丢失的情况，这时候，当主机 2 的计时器超时后，主机 2 就会释放连接。

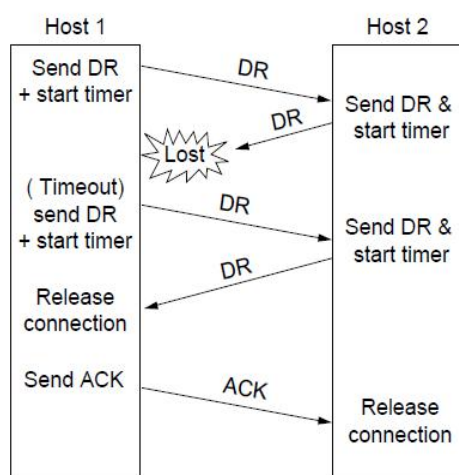


图 6-6

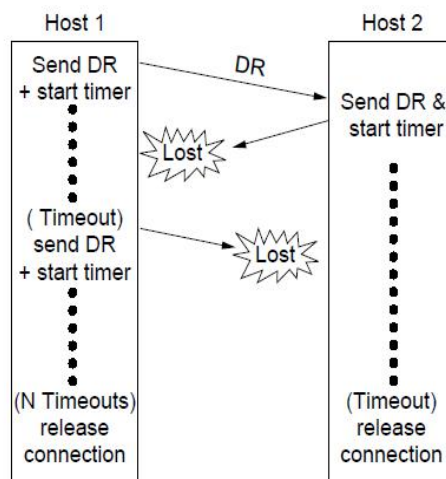


图 6-7

图 6-6 这种情况是主机 2 给主机 1 的 DR (Disconnection Request) 的应答丢失了, 这种情况下, 当主机 1 的计时器超时后, 主机 1 会重新发送 DR。

图 6-7 这种情况是主机 2 给主机 1 的 DR (Disconnection Request) 的应答和主机 1 后续的 DR 都丢失了, 这种情况下, 主机 1 经过 N 次重传之后, 就会放弃, 并且释放连接; 而主机 2 在计时器超时之后也会释放连接。这部分大家可以再详细参看课本 6.2 节。

6.2 UDP

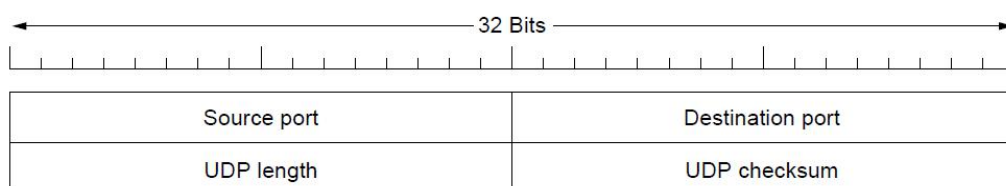
6.2.1 UDP 简介

UDP (User Datagram Protocol), 用户数据报协议。它的协议号是 17。

6.2.2 UDP 的一些特点

第一, UDP 是无连接的, 不可靠的。

第二, 分组头部开销小。TCP 有 20 字节的头部开销, UDP 只有 8 字节。UDP 头的布局结构如下所示:



第三, UDP 尤其适用的一个领域是在客户-服务器的情形下。

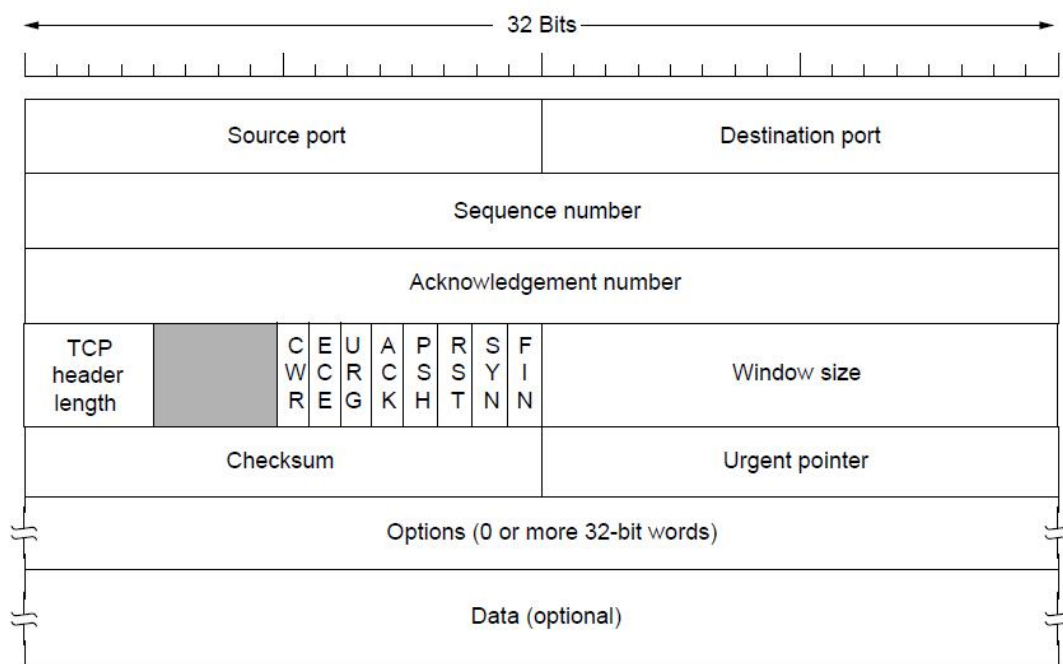
第四, UDP 的一个应用是 DNS (Domain Name System) <程序与 DNS 服务器之间>。

第五, UDP 不考虑流控制、错误控制, 在收到一个坏的数据段之后它也不重传。所有这些工作都留给用户进程。

其他 UDP 的相关知识大家请见 6.4 节。

6.3 TCP

6.3.1 TCP 数据段的头



上图显示了 TCP 数据段的布局结构，每一行为 32 位，即 4 个字节。

第一行是源端口（Source port）和目标端口（Destination port）信息。

第二行是序列号（Sequence number），表示此次发送数据的第一个字节的编号

第三行是确认号（Acknowledge number），表示下次想要接收数据的第一个字节的编号。

第四行由几个部分组成，第一部分是 TCP 头长度（TCP header length），它占 4 位，单位是“4 字节”，所以我们可以简单计算一下： $2^4 \times 4B = 64B$ （其中有 20B 是 TCP 数据段的头的固定长度，另外 44B 是可选项 Options）；第二部分是未使用的 4 位域；第三部分是 8 个 1 位标志：CWR 和 ECE 用作拥塞控制的信号、URG 置 1 表示使用了紧急指针、ACK 置 1 表示确认号字段是有效的、PSH 位表示这是带有 PUSH 标志的数据、RST 位被用于重置一个已经混乱的连接（一般而言，如果得到的数据段被设置了 RST 位，那说明你这一端有了问题）、SYN 被用于建立连接的过程、FIN 被用于释放一个连接；第四部分是窗口大小（Window size），它表示这个 TCP 数据段发送方当前可用的缓冲区大小，表示的是这一方的接收能力。

第五行由两部分组成：第一部分是校验和（Checksum），它校验的范围包括 TC2 区 P 数据段的头部、数据以及伪 TCP 头，下面我们来看看什么是伪 TCP



图 6-8 在 TCP 校验和计算中的伪头部

伪 TCP 头的第一行是源地址，第二行是目标地址，第三行由三部分组成：8 位 0，TCP 的协议号（6）以及 TCP 数据段（包括 TCP 头）的字节计数。第五行的第二部分是紧急指针（Urgent pointer），它指向一段程序，调用后清除内存里的相关内容。

第六行可选项 (Options) 是可选的, 前面我们计算过了, 它的长度是 0B~44B。

6.3.2 TCP 连接建立 (三次握手)

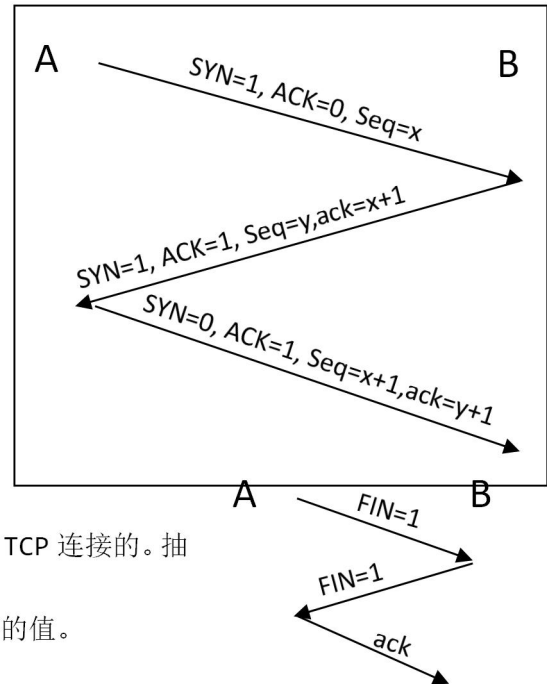
TCP 是面向连接的, 所以有三个重要阶段: 建立连接、使用连接和拆除连接, 这一部分我们讨论三次握手法建立 TCP 连接。

由于网络中节点的缓存能力的影响, 使得分组可能滞留在网络之中, 由于超时重传机制等原因产生的重复分组可能会引起严重的问题, 于是我们利用选号和三次握手来保证可靠的传输。

抽象图示如右图:

右图是三次握手建立连接的示意图, 通常在第三次发送消息的时候, TCP 数据段里已经携带了数据, 故将 SYN 位置为了 0。

在这里我仅是列出了最为重要的 SYN、ACK 标志位的值



6.3.3 TCP 连接断开 (三次握手)

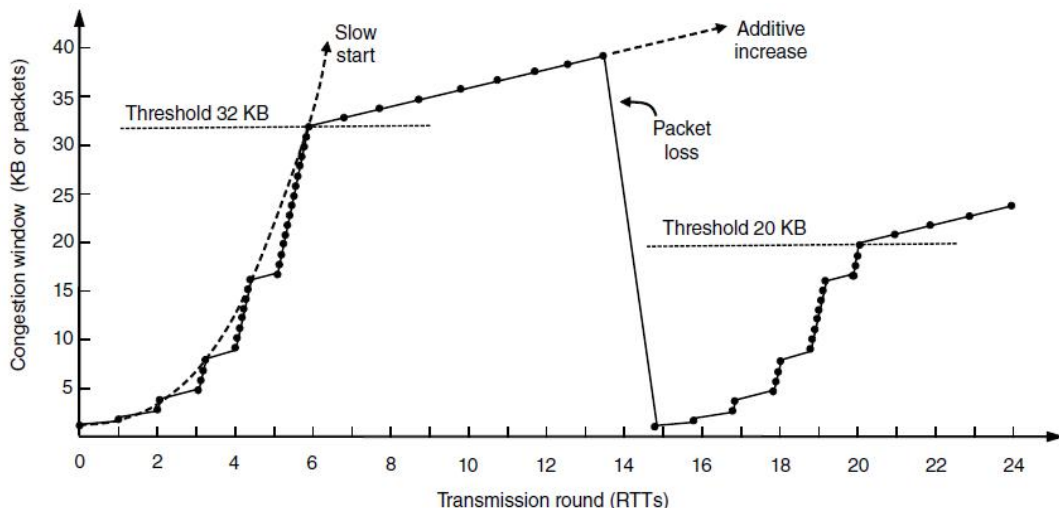
下面我们来看一看是如何用三次握手法断开 TCP 连接的。抽象图示如右:

在这里我仅是列出了最为重要的 FIN 标志位的值。

6.3.4 拥塞控制与慢启动算法

网络的拥塞里存在两个方面的问题, 对应着每个发送方维护的两个窗口: 网络容量 (对应拥塞窗口) 和接收方的容量 (对应接收方准许的窗口)。拥塞控制的本质是降低发送方的发送速率, 所以发送方的速率应该取以上两者里的小值。

拥塞控制实际上是由网络层 (RED) 和传输层 (TCP 慢启动) 共同完成的。接下来让我们一起看看什么是 TCP 慢启动。



以上图示即展示了 TCP 慢启动的过程, 一开始通过成倍增加 (指数级) 拥塞窗口的大小不断试探网络连接情况, 当到达阈值的时候, 开始线性地增长拥塞窗口的大小。当一次超时发生的时候, 将阈值设置成为当前拥塞窗口的一半, 而拥塞窗口被重置为初始的值。

拥塞窗口一直增长, 直至发生超时或达到接收方准许的窗口大小。

让我们再来看看网络层和传输层是如何合作完成拥塞控制, 降低发送方的发送速率的。

当网络中路由器的被使用缓冲区的大小到达路由器的阈值的时候，路由器开始执行 RED 协议，随意丢弃某些分组，被丢弃的分组的发送方因此会超时，这时通过 TCP 慢启动会降低发送方速率。

第七章 应用层

域名系统(Domain Name System): 一种层次的, 基于域的将主机名(域名)映射成 IP 的命名方案。

顶级域名主要包括通用的(edu,gov,com)和国家或地区的(cn,uk)。