计算机网络 Computer Networks

RandomStar

2021 年 1 月 21 日

目录

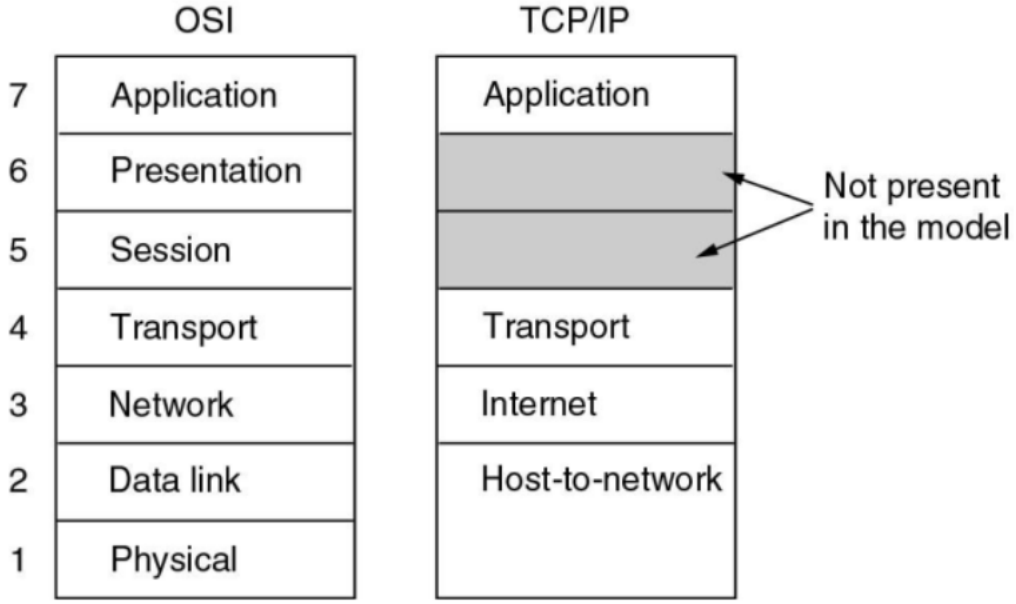
1 导论：A Walk Through Computer Network

1.1 网络模型

1.1.1 OSI 七层模型

OSI 七层模型中的三个重要概念：服务、接口和协议，其中 OSI 的七个层级分别是：

- 物理层：是真实的物理线路，在一条通信的信道上传输比特，用电子信号表示 0 和 1
- 数据链路层：将发送的数据拆分成若干的**数据帧 (frame)**，需要确认收到的帧的完整性和进行流量的调节。其中的介质访问控制子层需要控制对共享信道的访问
- 网络层：控制子网的运行，将数据包从源端发送到接收方，需要处理拥塞
  - ★ 允许异构的网络相互连接组成互联网络
- 传输层：接收会话层的数据，将数据分割成小的单元并传输道网络层
  - ★ 传输层是 7 层模型中端到端的最底层，自传输层以上都是端到端 (end-to-end)，以下的都是链式连接层
- 会话层：负责管理主机之间的会话进程，利用传输层提供的端到端服务向表示层提供增值服务，主要变现为表示层实体火用户进程建立连接并在连接上传输数据 (这一过程也叫建立同步)
- 表示层：关注传递信息的语法和语义，主要是实现数据的格式转换和压缩
- 应用层：包含了各种各样的协议，提供了**用户和网络之间的接口**，软件开发基本都是应用层的工作



在 OSI 七层模型中，不同的层传输的数据单元的名称不同，物理层是比特，数据链路层是帧，网络层是数据包，传输层是 TPDU，应用层是 APDU

Theorem 1.1 *几种常见的设备所处的 OSI 模型层级：集线器和中继器处于物理层，交换机处于数据链路层，路由器处于网络层。*

1.1.2 TCP/IP 模型和混合模型

TCP/IP 模型：原本只是 Linux/Unix 下的一个网络通信栈，比 OSI 模型简单，没有表示层和会话层，凭借实用和开放的特性成为了互联网业界的标准。TCP/IP 模型将 OSI 中的物理层和数据链路层抽象成了网络接口层，向上一次是网际层 (IP 协议)，传输层和应用层

Definition 1.1 *混合模型 Hybrid Model 是计算机学院的计算机网络主要讨论的模型，包含物理层、数据链路层、网络层、传输层和应用层。*

1.2 协议和服务

1.2.1 协议架构

几个关键名词：

- 协议 Protocol
- 层 Layer
- 对等 Peers：不同的主机之间存在着对等的协议层
- 接口 Interface：传输的过程中上层的协议会调用下层协议的接口
- 协议栈 Protocol Stack：一系列有联系的协议

1.2.2 服务类型

- 面向连接的服务：包括可信信息流，可信字节流和不可信连接，需要建立连接之后才能进行通信
- 无连接的服务：包括不可信数据报，共识数据包，请求回复，不需要建立连接

1.2.3 服务原语 Primitives

面向连接的服务有五个服务原语

- LISTEN 等待一个还没来的连接
- CONNECT 和一个等待的对等层进行连接
- RECEIVE 等待一个还没来的 message
- SEND 向对等层发送一个 message
- DISCONNECT 结束一个连接

1.3 网络 Network

1.3.1 网络的类型

- 个人局域网 PAN
- 局域网 LAN
- 城域网 MAN
- 广域网 WAN：特点是会有很多的路由器和子网
- 虚拟专用网络 VPN：计算机通过虚拟链路相连接，没有真实的线路
- ISP 网络：由网络服务商提供的网络
- 最大的网络就是因特网 Internet Work

1.3.2 网络传输方式

- 广播式传输 broadcast links：传输模型类似于总线，是总分式的结构，以太网都是基于广播的，给网络中的每台机器都发送包
- 点对点传输 P2P links：传输模型是复杂的网络拓扑结构，一次只发送给一台计算机消息
- 多播/组播 multicasting：一对一组的传播形式，传输模型是一个环，一次发给一组计算机，需要通过软件系统来实现多点广播机制
- 单播 unicasting：单个发送者和单个接收者，也就是点对点传输

1.3.3 网络的性能指标

- 数据率/比特率：数据传送的速率，单位是 bit/s 也写作 bps
- 带宽 bandwidth：原意是指某个信号具有的频带宽度，也就是某个信号所处的频率区间的长度，此时的单位是 HZ，但是在计算机网络中也可以指某个信道的最大传输速率，此时的单位是 bps。事实上两种说法一种是频域称谓，一种是时域称谓。
- 吞吐量 throughout：表示单位时间内某个网络每秒经过的实际数据量，并不是总量

2 物理层 Physical Layer

物理层存在的意义是**屏蔽不同的传输媒体和通信手段之间的差异**，使得物理层上面的数据链路层感受不到这些差异，物理层的协议也通常被称为规程 (procedure)

Theorem 2.1 一个数据通信系统分为如下几个组成部分：源系统，传输系统和目的系统；三者又叫做发送端，传输网络和接收端

2.1 通信的基本概念

2.1.1 有线通信

有线通信中的各种线路

- 双绞线 Twisted Pair：常用于电话系统，距离很远的时候信号衰减就非常厉害，需要使用中继器
- 同轴电缆 Coaxial Cable
- 光纤 Fiber Cables，利用了光的全反射原理，在特定的入射角度下不会发生折射，传输效率非常高
- 接口线 Interface Line

2.1.2 无线通信

无线通信包括无线电传输、红外传输、卫星通信和微波通信等等，主要依赖于电磁波的传播进行通信。而点淄博按照频率的不同可以分为低频中频和高频波段，其中光纤的波段是最高的。

2.1.3 通信方式

从通信双方信息的交互方式来看，可以分为三种基本的通信方式：

- 单工通信 (Simplex communication)：只能进行单个方向的通信，一方负责收，一方负责发，而不能进行反方向的交互，只要一条信道
- 半双工通信 (Half duplex communication)：通信可以双向进行，双方都可以发送和接收，但是任何一方不能同时发送和接收
- 双工通信 (Duplex communication)：通信双方可以同时发送和接收信息，也需要两条信道

2.1.4 信号

Definition 2.1 信号是数据的电气或者电磁表现，分为模拟信号和数字信号。模拟信号也叫连续信号，代表信息的参数取值是连续的；数字信号也叫离散信号，代表消息的参数取值是离散的。

Theorem 2.2 电话是将模拟信号转换成模拟信号，*Modem* 是将模拟信号转换成数字信号，*Codec* 是将数字信号转换成模拟信号。

2.1.5 信道

Definition 2.2 信道 *(Channel)* 表示向某一方向传送信息的媒体，信道和电路的概念不相同，电路往往包含了发送信道和接受信道各一条，信道的通信方式也分为单向通信、双向交替通信和双向同时通信。

2.1.6 码元

码元 (Code Element) 是指用一个固定时长的波形信号来表示一位 K 进制的数字，代表不同的离散数值的基本波形，是数字通信中数字信号的计量单位。常用的是二进制码元，一种表示 0，一种表示 1

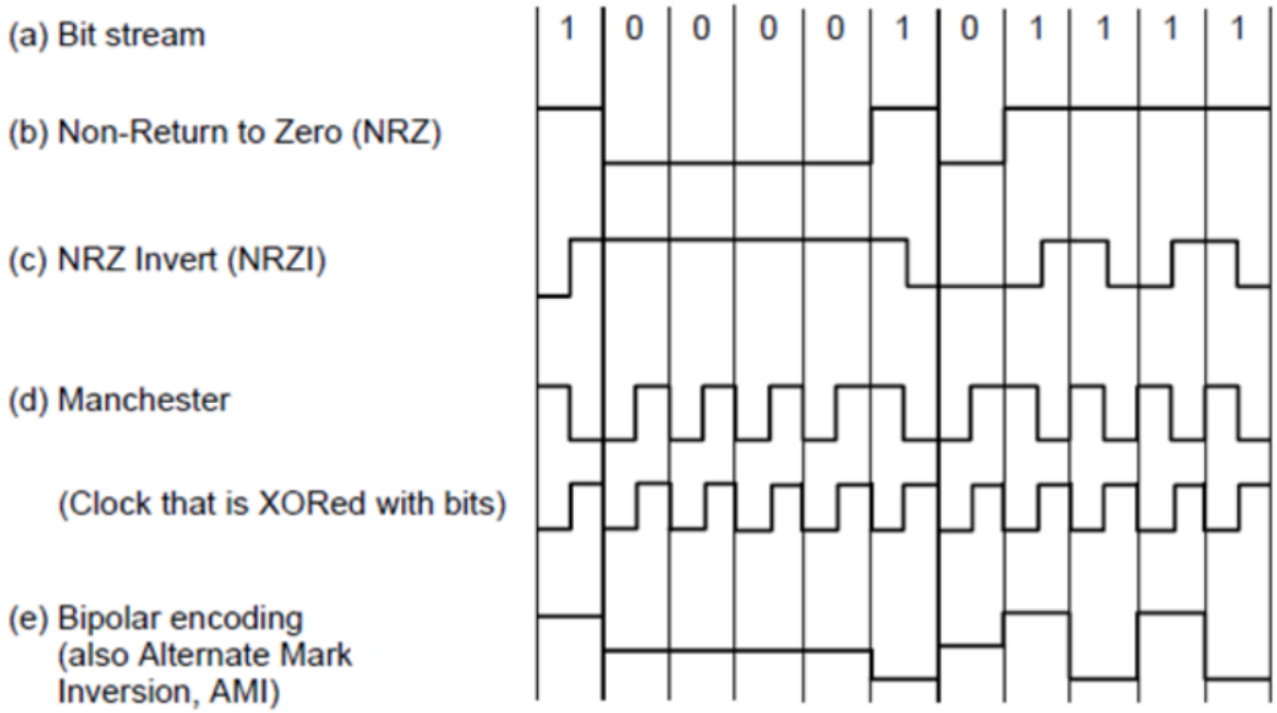
2.2 信号传输方式

2.2.1 基带传输 Baseband Transmission

基带传输的信号分为 0 和 1，占有传输介质上从 0 到最大值之间的全部频率，有如下几种不同的表示信号的方式：

- Return to zero 用高电平代表 1，低电频代表 0，每个时钟周期的中间变成低电频
- Non-return to zero 用高电平代表 1，低电平代表 0，存在着时钟恢复问题，应该从中间处采样
- NRZ Invert 信号的突变表示发生了 1 和 0 的切换，信号不变就表示一直是 1 或者一直是 0
- Manchester 曼切斯特编码，1 用半个高电平 + 半个低电平表示，0 用半个低电平和半个高电平来表示
- 4B/5B 编码每 4 位以组，按照规则扩展成 32 位，多出的 16 位作为控制位

Theorem 2.3 以太网使用的基带传输方式就是曼切斯特编码



2.2.2 通带传输 Passband Transmission

信道上允许不同的信号共存的传输方式叫做同代传输，任意一个的频率波段都可以用来传递信号。

Theorem 2.4 调制的三种方法：调幅、调频和调相，分别通过调节信号的振幅、频率和相位来起到调制的作用。

以上三种操作的英文名称都是 xxx shift keying，幅度是 Amplitude，频率是 requency，相位是 Phase，对应的学名分别叫做幅移键控 ASK、频移键控 FSK、相移键控 PSK

QPSK 是正交相移键控，有 45,135,225,315 四个偏移角度

2.3 信道的传输速率

Theorem 2.5 *Nyquist* 定理：假设带宽为 W 则信号的传输速率不会超过 $2W$，这个定理需要在一个信道并且没有噪声的情况下才适用。对于多进制的编码，数据的传输速率

$$v = 2W \log_2 M$$

其中 M 表示进制，比如二进制编码的时候 $M=2$，结果的单位是 *bps*，没有理论值的上限。

Definition 2.3 信噪比：记为 S/N，并用分贝 dB 作为度量的单位，即为信号的平均功率和噪声的平均功率之比，计算公式如下

$$dB = 10 \log_{10} \frac{S}{N}$$

所以如果以 dB 为单位给出信噪比需要用上面的公式来计算出 S/N，而如果没有单位的信噪比则往往指的就是 S/N 的值

Theorem 2.6 香农定理 *Shannon's Theorem*，信道的极限信息传输速率 C 是

$$C = W \log_2(1 + \frac{S}{N})$$

这表明信噪比越大，信道的最大传输速率也就越大

一般在计算题里面会用两种方法都算一遍最大传输速率，取其中结果最小的作为最终的估算结果

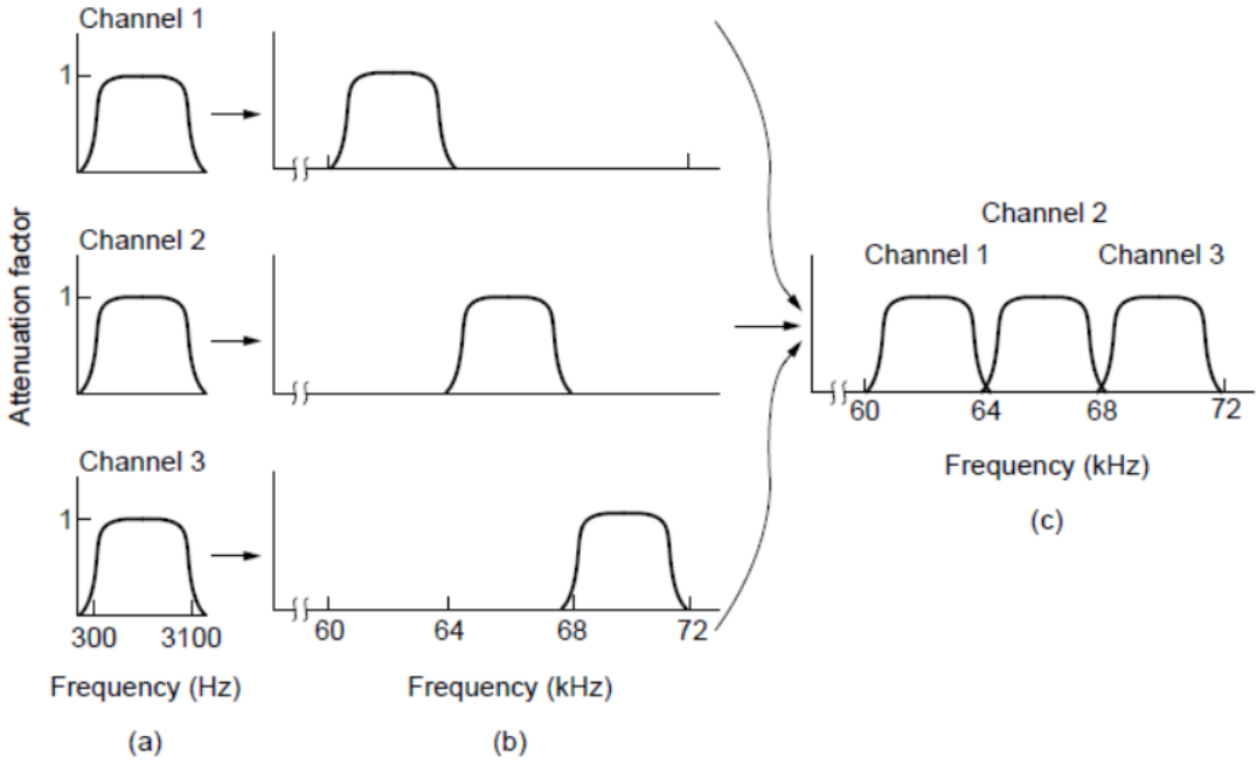
2.4 信道复用

信道复用 (Channel Division Multiplexing) 包含以下三种提高信道利用率的方法：

2.4.1 频分复用 FDM

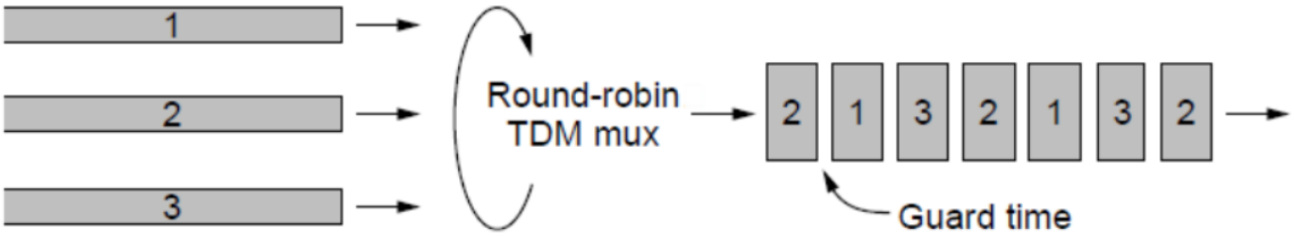
Definition 2.4 频分复用 *(FDM)*：利用通带传输的优势使得多个用户可以共享信道，每个用户拥有一个自己的频段来传输信号。正交分频复用 *(OFDM)*

说白了就是把信道分成了多个频率段。



2.4.2 时分复用 TDM

Definition 2.5 时分复用 *(TDM)*: 每个用户周期性地获取整个带宽非常短的一个时间段，每个输入流的 *bit* 从一个固定的时间槽中取出并输入到混合流中。这项技术在电话网络和蜂窝网络中的应用比较多。可能需要设置一个保护时间，不能切换的过快。



2.4.3 码分复用 CDM

Definition 2.6 码分复用 *(CDM)*: 是扩展频谱通信的一种方式，将一个窄带信号扩展到一个比较宽的频带上，允许来自不同用户的多个信号共享相同的频带，因此又叫码分多址 *(CDMA)*

在码分多址中，每个比特时间被分为 m 个更短的时间间隔，称为码片 (chip)，码片由 +1 和-1 组成。若原本要传输 b 位的码，在码分复用下就变成了传输 mb 个码片。**如果要传输一个 1 就发送分配的码片序列，**

如果要传输 0 就发送码片序列的反码，对于多种信号，我们只需要让其码片两两正交，就可以在同一个频段内同时发送多种信号，解码的时候可以通过解线性方程组来获得每种信号的组成。

Theorem 2.7 *设 S 和 T 分别是两种不同的码片，用 $\bar{S}$ 和 $\bar{T}$ 来表示其反码片，则我们有如下性质：*

$$S \cdot T = 0$$

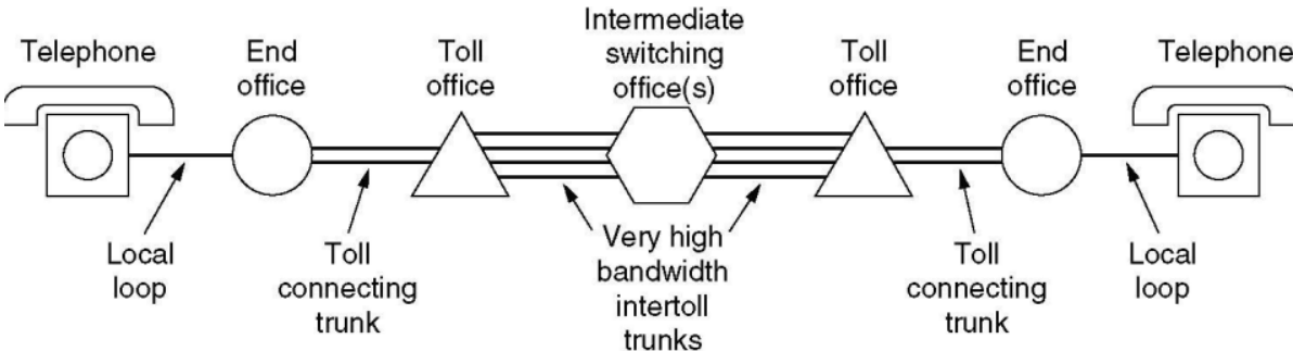
$$S \cdot S = \frac{1}{m} \sum_{i=1}^{m} 1^2 = 1$$

$$S \cdot \bar{S} = -S \cdot S = -1$$

2.5 公用电话系统 PSTN

公用电话系统用于传递人声，主要分为如下几个部分：

- 本地回路 Local loop：用来传输模拟信号的双绞线，是电话和 end office 之间的
- 干线 Trunk：数字光缆，连接了各个交换局，是两种 end office 和 switching office 之间的
- 交换局 Switching office：进行了通话的交换，从手动切换变成了计算机切换。



ADSL 是一种宽带技术，重用了电话系统的本地回路，在上面讲数字数据从客户端发送到端局，然后被虹吸到因特网中，不过一些地方本地回路已经被光纤所取代。

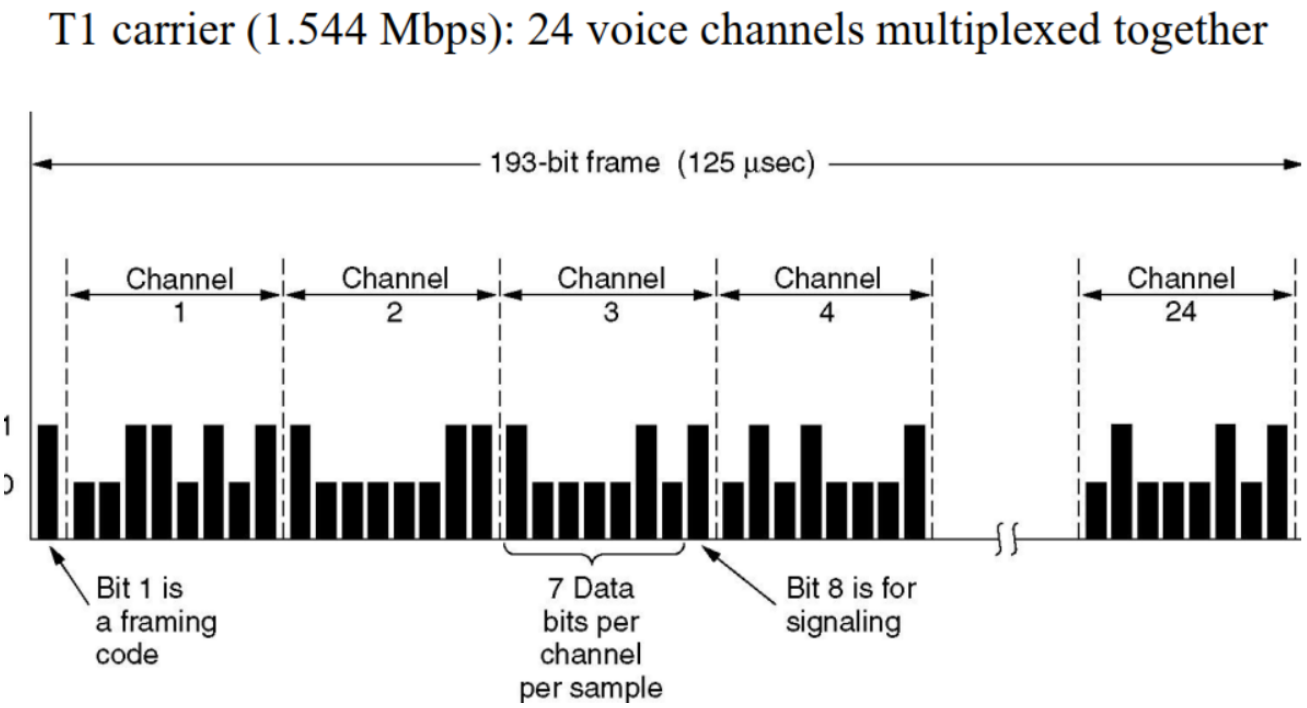
2.5.1 中继线和多路复用

中继线比本地回路要快，而且中继线支持大量的电话呼叫，高带宽的中继线的共享可以通过 TDM 和 FDM 来实现

2.5.2 载波 Carrier

PCM 脉冲编码技术：每秒采样 8000 次，每次的采样量化成 8bit 的二进制数

Definition 2.7 *T1 载波：有 24 个 PCM 信号，数据传输速率是 1.544Mbps，一个数据帧共有 193 位，一共有 24 个信道，其中每个信道有 8bit，7 位是数据，1 位是信号，第 1 位是帧码。因此 193bit 中有效的数据有 168bits，开销率约为 13%*



Definition 2.8 *E1 载波，数据传输速率是 2.048Mbps，一共有 32 个 PCM 信号，其中 30 个用来传数据，2 个传输信号，因此开销是 6.25%*

2.5.3 SONET 同步光网络

SONET 是同步光网络的简称，别的我也不知道了。

2.6 交换

常见的数据交换方式有如下几种：

2.6.1 电路交换 Circuit switching

在数据传输之前两个节点必须建立物理线路，这条物理线路可能有多个中间节点，在整个数据传输期间一直被独占，通信结束之后才被释放，因此电路交换分为三个阶段：连接建立、数据传输和连接释放。**优点是通信时延小，传输有序而没有冲突，控制简单实时性强，缺点是建立连接的时间长，线路独占，灵活性差，难以规格化。**

2.6.2 报文交换 Message switching

数据交换的单位是报文，报文携带地址等信息，报文交换**在交换节点采用的是存储转发的传输方式**，优点是不需要建立连接，动态分配线路，可靠而利用率高，提供多目标服务，缺点是会有转发延迟，对缓存空间要求较大，报文交换主要使用在早期的电报通信网中，现在使用较少。已经被分组交换所淘汰。

2.6.3 分组交换 Packet switching

分组交换和报文交换一样也采用存储转发的方式，但是解决了大报文传输的问题，分组交换限制了每次传送的数据块大小的上限，将大的数据拆分成若干个小的数据块，再加上一些控制信息，组成了若干分组(Packet)，网络就饿点根据控制信息把分组送到下一个节点，下一个节点接收到分组以后，暂时保存并等待传输，直到送到目标节点。优点是不需要专门的通信线路，信道的利用率高，缺点是存在传输的时间延迟，需要额外的空间存储控制信息，可能会引发失序、丢失、重复等现象。

分组交换还可以进一步分为**面向连接的虚电路方式和无连接的数据报方式**。数据报就是在端系统中高层协议先拆分保温，并在网络层加上控制信息后形成数据报分组 (PDU)，然后发送出去，发送分组之前不需要建立连接，可以随时发送，网络尽最大的努力来交付，不保证传输的可靠性，这实际上是后面网络层的东西。

虚电路是数据报方式和电路交换方式的结合，发送之前需要建立一段逻辑上相连的虚电路，整个过程分为虚电路建立、数据传输和虚电路的释放三个部分，虚电路的连接一旦建立之后相对应的物理路径也就确定下来了，一旦发生物理设备的故障，可能就会使得一条虚电路坏死。

3 数据链路层 Data Link Layer

3.1 基本概念

3.1.1 链路层提供的服务

物理层用来构建网络的物理链路，而数据链路层则用来构建数据链路和逻辑链路，二者本质上都是用来构建网络通信和访问通道的。

物理层都是真实存在的物理线路和设备，而数据链路主要是一些必要的硬件 (如网络适配器和交换机) 以及软件 (传输协议) 组成的。

数据链路向上提供网络层的服务，处理传输的错误，调节数据流，确保接收方不会被淹没。

- 无确认、无连接的服务：对丢失的帧不负责重发而实交给上一层处理，适用于实时通信或者误码率低的信道，比如以太网
- 有确认、无连接的服务，比如无线通信
- 有确认、面向连接的服务：真的传输需要建立数据链路、传输帧、释放数据链路，适用于长距离的不可靠链路
- 提供虚拟的交流，实际上就是通过协议和查错机制，避免通过物理层的通信而产生的差错

3.1.2 帧 Frame

帧是数据链路层发送数据的基本单位，数据包 Packet 由若干个帧 Frame 构成，帧具有一定的格式，常见的帧格式有如下几种：

- 字符计数法：帧的开头记录其长度 (数据的位数)，不常用，容易出错，因为只要一个帧出错后面的就都错了
- 字节填充法：用 STX 表示帧的开头，ETX 表示帧的结束，DLE 表示转义字符
- 比特填充法：通过在帧头和帧尾个插入一个特定的 bit 穿来标识一个数据帧的起始和结束
  - ★ 常用的是 01111110
  - ★ 为了保证传输的内容不被错误地解析，可以在传输的数据中每出现连续的 5 个 1 就添加一个 0，保证数据中不会出现连续的 6 个 1，便于区分标识字符串

3.2 差错检测和纠正

3.2.1 纠错码 Error Correcting Code

Definition 3.1 *假设一帧由 m 位数据和 r 位冗余组成，记 $n = m + r$ 则该编码方式称为 (m, n) 码*

Definition 3.2 *海明距离：两个码字 (codeword) 中不同的位的个数，如果两个码字的海明距离为 d，则需要出现 d 个 1 位的错误才能把正确的码字变成错误的码字。*

码的纠错能力

- 海明距离为 n 的编码方案只能检测出 n-1 个错误，因为如果 n 位都不同无法判断到底谁是对的
- 为了检测 d 个错误，需要 d+1 的海明距离的编码方案
- 而为了纠正 d 个错误，则需要 2d+1 个解决方案

对于每 2^m 个合法的消息，每个消息对应有 n 个非法的码字 (即海明距离为 1 的非法码字有 n 个)，此时每个合法的消息需要 n+1 位来标识，由于总共有 2^n 种位模式，因此必须有 $(n+1)2^m \leq 2^n$，即 $(n+1) \leq 2^r$

Definition 3.3 *海明编码: 海明距离为 3，可以发现 2 位的错误和纠正 1 位的错误，将码字内的位编号为 1 到 n，其中 2 的幂次位数就是校验码，其余的都是数据*

3.2.2 循环冗余校验 Cyclic Redundancy Check

基本思路是在数据帧的末尾添加若干位作为校验码，并且使得生成的新帧能被发送端和接收端所共同选定的某个特定数字整除

此时我们需要一个生成多项式 G(x)，并需要其最高位和最低位的系数都是 1，这样的一个生成多项式代表了一个二进制数，作为生成校验码的除数。假设这个除数有 K 位，则我们需要在发送的 m 位数据末尾加上 K-1 个 0 得到一个 (m+K-1) 位的数并**除以**上面选定的 K 位除数，所得到的余数就是 CRC 校验码

这个校验码也叫做帧校验序列 FCS，并且这个数只能比除数少一位，不能省略高位的 0

Theorem 3.1 *注意：上面提到的除法并不是常见的除法，而是不借位的除法 (减法是异或 XOR)*

把这个余数附加在原数据帧的末尾 (覆盖之前的 0)，构建一个新的帧发送到接收端，并在接收端除以第二步里设定的除数，如果没有余数就说明传输过程没有出错

Example 3.2.1 *这里举一个简单的例子,比如要传输的数据是 1101011011,我们选择的生成多项式是 $G(x) = x^4 + x + 1$，则对应的除数是 10011，则我们将 1101011011 后面加上 4 个 0 变成 11010110110000，运算以后得到的余数是 1110，则我们将 1110 覆盖刚才添加的 0000，得到需要发送的数据是 11010110111110，将其发送到接收端，接收端也有 10011 验证即可*

3.3 链路层传输协议

3.3.1 Naive 的单工协议

发送过程和接受过程是单独的，发送过程是一个无限的 while 循环，尽可能地发送数据，接受过程是每次到达一个未损坏的帧就接收。数据只在一个方向上传输。但是这种协议没有任何纠错能力和流量控制能力。大多数时候往往需要双工的数据传输。

3.3.2 滑动窗口协议 Sliding Window

滑动窗口 (Sliding Window) 协议就是一类双向通信的协议，所有滑动窗口协议的本质是在任何时候发送方都维持了一组序号，分别对应于允许发送的帧，我们称这些帧落在发送窗口内，类似的接受方也有接收

窗口，这些窗口可以是定长的，也可以变化。常见的滑动窗口协议分为以下三个：停止等待协议、GO-Back-N 协议、选择重传协议。

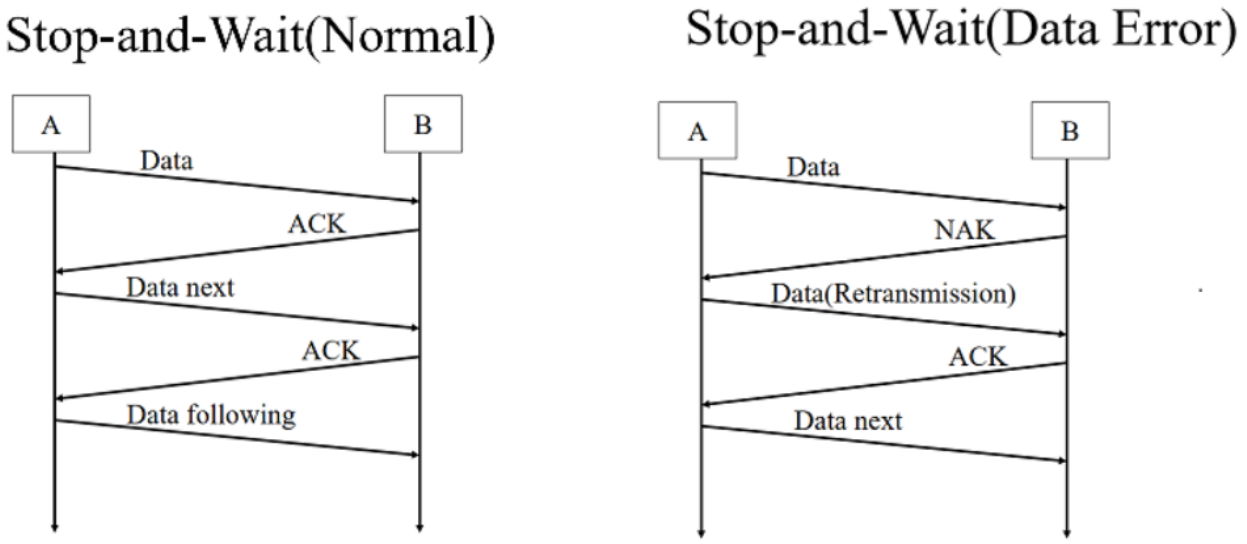
发送方维持一组发送窗口，每次收到 ACK 就把窗口移动一个位置。接收方维持一组接收窗口，每次收到一个数据帧就把窗口移动一格并发送 ACK 确认，落在接受窗口之外的数据帧一律丢弃。

1 位滑动窗口协议顾名思义两边的窗口大小都是 1，发送方发送了一帧之后必要要等待前一帧的确认到达之后才可以发送下一帧，所以这种协议使用的也是停止等待的方法。

3.3.3 停止等待协议 Stop-and-wait Protocol

是最简单的通信协议，每次发送完毕之后，发送方就停止，等待接收方收到数据并发送 ACK 回应包，之后再进行下一次的发送。但是可能会出现如下异常：

- 如果收到了 NAK 包则表示不确认，需要发送方重新发送
- 如果产生了丢包就会一直等不到发送方的回复，此时可以尝试主动重发
- 如果产生了 ACK 包的丢失也会导致发送方重发，但是此时接收方会丢弃接收的数据并再次发送 ACK



对于发送方的数据包可以用 1bit 的位置来进行标记，也可以用 01 交替的方式表示发送的是一个新的数据包而不是旧的数据包重新发送了

Theorem 3.2 协议效率的衡量：用 T_{frame} 表示发送方发送一个完整的帧所需要的时间，T_{prop} 表示传输到接收方需要的时间并且有如下的计算公式：

$$T_{prop} = \frac{distance}{speed}$$

$$T_{frame} = \frac{frame\_size}{bit\_rate}$$

我们令

$$\alpha = \frac{T_{prop}}{T_{frame}}$$

则链路的利用率为：

$$U = \frac{1}{2\alpha + 1}$$

Theorem 3.3 滑动窗口协议假设有 N 个窗口，则其利用率 $U = \min(\frac{N}{2\alpha+1}, 1)$，不过仅存理论可能

Theorem 3.4 对于一般的滑动窗口协议，长发送时间，短帧和高带宽会造成非常严重的浪费，一种解决的办法是管道化传输，但是这会导致数据帧传输出现错误，因此需要一定的处理办法

3.3.4 回退 N 和选择重传协议

几种不同的重新发送方式

- 回退 N(Go-Back-N)：出现错误之后就丢弃之后所有的帧，等待重新发送
  - ★ 适用于接收窗口大小为 1 的情形
  - ★ 假设存在 0-MAX_SEQ 这样 MAX_SEQ+1 个序列号，可以发送的帧最多为 MAX_SEQ 个
  - ★ 一般可以发送的序列号的个数可以由 bit 数来确定，必须要留至少一个 bit 用来收 ACK
- 选择重传 (Slective Repeat)：出错的时候先缓存后面的没有出错的帧，结束之后只需要重新发送对应出错帧即可
  - ★ 为了保证没有需要冲突，窗口的最大尺寸不应该超过 (MAX_SEQ+1)/2
  - ★ 相比回退 N 需要更大的缓冲区

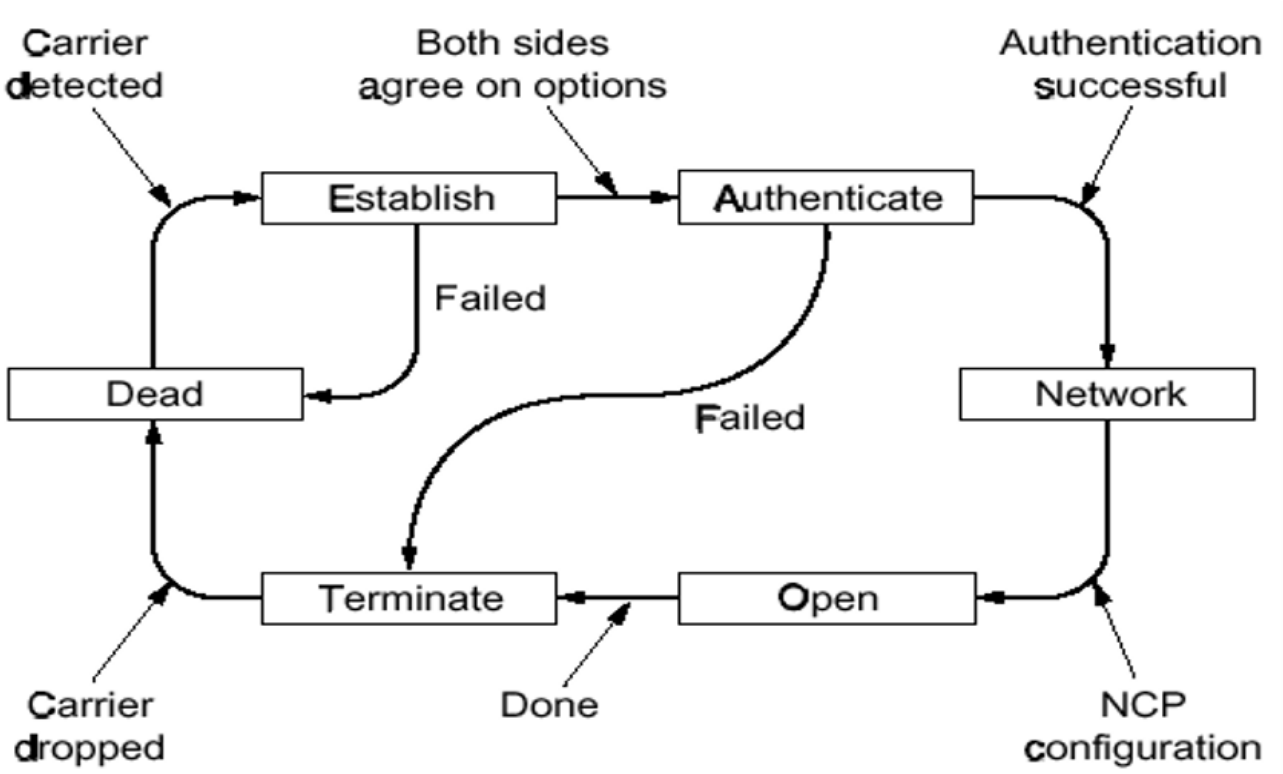
3.4 数据链路层协议

3.4.1 点对点协议 PPP

PPP 协议用于在链路中发送数据包，包含光纤链路和 ADSL，是早期协议 SLIP(串行线路 Internet 协议) 的改进，PPP 协议提供了以下的功能：

- 一种成帧的方法，可以准确地区分出一帧的结束和下一帧的开始
- 链路控制协议 LCP，可以用于启动线路、测试线路，当不再需要线路的时候关闭线路
- 网络控制协议 NCP，是一种协商网络层选项的方式，针对每一种支持的网络层都有一个不同的 NCP
- 需要提供身份验证，可以动态分配 IP 地址

PPP 协议的状态图：



3.4.2 HDLC 协议

HDLC 的全称是高级数据链路控制协议，和 PPP 的主要区别在于 PPP 面向字节而 HDLC 面向比特，**PPP 使用的是字节填充技术而 HDLC 使用的是比特填充**，HDLC 的数据帧有如下格式：

- 帧的起始和结束都是 01111110
- 8bit 的地址 +8bit 的控制位 + 若干位 data+16 位的 checksum，不同类型的帧控制位数的内容不同
- 对于信息帧，8bit 控制位 =0+3bit 序列号 +1bit 的 P/F+3bit 的下一个序列号
- 对于监督帧：10+2bit 的类型 +1bit 的 P/F+3bit 的下一个序列号
- 无符号帧：11+2bit 的类型 +P/F+3bit 的 modifier

4 介质访问控制子层 Medium Access Control Sublayer

4.1 信道的分配

网络链路可以分为两大类，主要是点到点的连接 (PPP) 和广播信道，广播信道有时候也称为**多路访问信道和随机访问信道**。广播信道会引起多方竞争，而**确定信道的下一个使用者就非常重要**，这里就是介质访问控制子层 (MAC) 的主要作用。

Theorem 4.1 *MAC 层位于数据链路层的底部，是数据链路层的一个子层。*

信道的分配分为静态分配和动态分配两种

4.1.1 静态信道分配

静态分配：主要是物理层中介绍的频分复用，波分复用和码分复用等，在用户少和信道充足的情况下简单高效。设 K 是帧的到达速率，$\frac{1}{\lambda}$ 是帧的长度，C 表示信道总数，则单个信道的帧平均延迟时间

$$T = \frac{1}{\lambda C - K}$$

在频分复用的时候时间延迟变成了 NT，其中 N 是子信道的个数

4.1.2 动态信道分配

动态信道分配，又叫多点接入，特点是信道并非在用户通信的时候固定分配给用户，常见的多路传输协议有 ALOHA 和 CSMA，动态信道分配有 5 个基本的假设：流量独立 independent traffic、单信道假设 single channel、冲突可观察 observable collision 、发送时间连续或者分槽 continuous or slootted time、载波侦听或者不侦听 carrier sense or no sense　　而动态信道的分配协议又分为竞争性的协议和非竞争性的协议，具体的协议有下面几种。

4.2 MAC 中的协议

4.2.1 ALOHA 协议

ALOHA 协议是竞争性的协议，ALOHA 协议的基本思想非常简单，就是在用户**有需要的时候就传输**，而产生冲突的帧将被损坏，因此需要检测是否产生冲突，因此 ALOHA 协议中每个站发送帧之后会把该帧重新广播给所有的站点。

如果帧被损坏了，发送方就需要随即等待一段时间 (必须随机等待，否则会一次次产生冲突)，会产生冲突的信道是一个竞争系统。

- 纯 ALOHA 协议：协议的效率计算方式是 $S = Ge^{-2G}$，其中 G 表示一个帧时内的平均帧数，即发送的帧数 (含新帧和重发帧)，但是协议的最大效率约为 18.4%，因为所有站点都是随机发送的，因此冲突非常多
- 分槽 ALOHA 协议：将时间分成离散的间隔，用户需要遵守统一的时间槽边界，此时的效率是 $S = Ge^{-G}$, 因此最高效率约为 36%，是纯 ALOHA 的两倍

4.2.2 CSMA 协议

CSMA 协议 (Carrier Sense Multiple Access Protocol，载波侦听多路访问协议)：特点是在发送数据之前一直监听信道，如果冲突了就等待一段随机的时间之后再试一次，分为如下几种类型：

- 1-persistant CSMA：先监听，有空就发送，如果信道繁忙就持续监听，**发生冲突就随机等待一段时间**，如果带宽延迟乘积大，协议的性能就差
- non-persistant CSMA：如果信道正在使用就放弃监听，改成随机等一段时间之后在尝试监听和发送，利用率高但是延迟大
- p-persistant CSMA：持续监听信道如果空闲就按照一定的概率 p 发送数据，否则推迟到下一个时间槽，信道繁忙就持续监听
- 有冲突检测的 CSMA：假设两个站点之间传播所需的最长时间是 t 那么只有一个站的传输时间大于 2t 才能被检测到冲突

几种 CSMA 的区别就在于监听信道和发送时机的选取。

4.2.3 无冲突协议 CFP

此类协议**以根本不可能产生冲突的方式解决了信道竞争的问题**，虽然这种协议并不被应用在主流系统中，目前 CFP 有如下几种类型：

- 基本位图法：每个竞争期包含 N 个槽，竞争槽中表示哪个 station 需要传输，假设数据的传输所需的时间长度为 d，栈的编号为 0 到 N-1
  - ★ 在低负载的情况下，低序号的站平均等待时间是 1.5N，高序号的站的平均等待时间是 0.5N，对所有站而言平均等待时间是 N，即每一帧的额外开销是 N 位因此低负载时的利用率为 $\frac{d}{N+d}$
  - ★ 高负载的情况下，N 个竞争位被分配到了 N 个帧上面，则此时的利用率是 $\frac{d}{1+d}$
- 令牌传递 token passing：传递一个称为令牌的短消息，在令牌环中利用网络的拓扑结构发送令牌，而令牌总线利用总线发送帧
- 二进制计数：高序号站的优先级比较高，信道的利用率是 $\frac{d}{d+\log_2 N}$

Theorem 4.2 *竞争性的协议延迟低而效率高，无冲突的协议效率低而延迟高。*

有限竞争协议：低负载的时候使用竞争协议降低延迟，高负载的时候采用无冲突的方法提高信道效率，比如自适应树遍历协议。一言蔽之就是缝合怪。

4.3 WLAN 协议

无线通信通常不能检测出正在发生的冲突，并且站点接收到的信号可能比较弱，会带来隐藏终端问题和暴露终端问题。考虑四个站点依次为 A,B,C,D

Definition 4.1 隐藏终端问题：假设 A 向 B 传送数据而 C 开启对 B 的监听，此时 C 错误地以为可以向 B 传输数据，并向 B 传输数据，造成了两个数据帧的冲突。即因为竞争者离太远而无法检测到潜在竞争者的情况称为隐藏终端问题。

Definition 4.2 暴露终端问题：如果 B 向 A 传输，而 C 也想向 D 传输数据，此时 C 开启监听发现有一个传输正在进行，导致 C 不会向 D 传输。因此无论什么时刻，一个 $WLAN$ 中只能有一个传输正在进行。

Theorem 4.3 冲突避免多路访问协议 $MACA$，基本思想是发送方刺激接收方输出一个断针，以便其附近的站点可以检测到该站点，避免在接下来数据量较大的时候产生冲突。

4.4 以太网 Ethernet

以太网是三种局域网中的一种，也是目前使用最广泛的局域网。

4.4.1 局域网 LAN

局域网是可以在一个较小的地理范围内使用的计算机网络，可以进行广播和组播，局域网的特性一般由拓扑结构、传输介质和介质访问控制方式来决定。介质访问控制的方法主要有 CSMA/CD、令牌总线和令牌环。三种常见的局域网特征如下：

- 以太网 IEEE 802.5 标准：总线形结构的逻辑拓扑、星形物理拓扑
- 令牌环 IEEE 802.5 标准，环形的逻辑拓扑结构
- FDDI 光纤分布数字接口 IEEE 802.8，环形的逻辑拓扑结构和双环物理拓扑结构

4.4.2 两类以太网

以太网采用无连接和尽最大努力交付，提供的是不可靠的服务。以太网分为经典以太网和交换式以太网，经典以太网使用前面提到的各种软硬件技术解决了多路访问的问题，而交换式以太网采用了一种称为交换机的设备来连接不同的计算机。

经典以太网坚持使用 1-persistant CSMA/CD 算法，这意味着每个站发送数据之前需要检查是否有冲突，一有空闲就立即发送，在它们发送的同时检测信道上是否有冲突，如果有冲突则立即中止，并发出一个短冲突加强信号，在等待一段时间之后随机重发。

4.4.3 网卡

以太网在逻辑上是总线形结构，所有的计算机共享一条总线。计算机和外界网络的连接通过主机中插入一块网络接口板 (也叫适配器和网络接口卡 NIC)，简称网卡，而网卡有唯一标识 MAC 地址，用于控制主机在网络上的数据通信，数据链路层的设备都使用各个网卡的 MAC 地址，同时网卡也工作在物理层，它只关注比特，而不关注任何地址信息和高层的协议信息。

MAC 地址也就是计算机的物理地址，长度为 6 字节，也就是 48 位，前 24 位是厂商编号，后面 24 位是厂商自己分配的序列号。总线上使用的是广播通信，因此网卡从网络上每次收到 MAC 帧都需要检查帧的 MAC 地址，如果是自己的就接受，否则就丢弃。

4.4.4 以太网的传输介质

以太网常用的传输介质有 4 中：粗缆、细缆、双绞线和光纤，都使用曼彻斯特编码，具有如下特征：

- 粗缆：10BASE5，拓扑结构是总线形，最长可达 500m
- 细缆：10BASE2，拓扑结构是总线形，最长可达 185m
- 非屏蔽双绞线：10BASE-T，拓扑结构是星形，最长可达 100m
- 光纤：10BASE-FL，拓扑结构是点对点，最长可达 2000m

4.4.5 以太网帧

以太网帧在 MAC 层的格式是：

- 6 字节目标地址 +6 字节源地址
- 2 字节类型，指出数据域中携带的数据应该交给哪个协议处理
- 46-1500 字节的数据，至少需要 46 字节的数据，缺少的时候会加上填充位，根据 CSMA/CD 算法得知以太网帧最少需要 64 字节，所以数据最少要 46 字节
- 4 字节校验码，可以校验地址和数据，采用 32 位的循环冗余校验

4.4.6 以太网的网速

- 10BASE-T 的以太网速度是 100MB/S
- 千兆以太网的速度是 1GB/S，万兆以太网的速度是 10GB/S

4.4.7 二进制指数后退

Definition 4.3 *二进制指数后退：以太网帧在第 i 次冲突之后，随机等待 $[0, 2^i - 1]$ 个时间槽的实践之后再重新发送，但是最大不超过 1023，即第 10 次失败时候的等待时间上限，当失败 16 次之后就会躺平放弃挣扎，并给计算机返回一个失败的报告。*

4.5 无线局域网 802.11

IEEE802.11 是无线局域网的一系列协议标准，制定了 MAC 层的协议，可以运行在多个物理层标准上面，采用 CSMA/CA 协议 (即避免冲突的 CSMA 协议) 进行介质访问控制。

为了避免冲突，每个发送节点在发送帧之前需要监听信道，如果空闲就可以发送，而发送了一帧之后必须等一个短的时间间隔，检测是否收到了 ACK，如果收到了 ACK 就表明发送成功，否则就说明发送失败，需要重新发送。在有线局域网中遇到冲突就立马停止发送，而无线局域网中就算发生了碰撞也要发送下去。

无线局域网可以分为两大类：固定基础设施无线局域网和无固定基础设施无线局域网自组织网络

4.6 数据链路层的设备

4.6.1 网桥 Bridge

网桥可以用来**连接多个以太网，原本的每个以太网就成了一个网段**，网桥工作在数据链路层的 MAC 子层，可以将各个网段隔离开来。网桥处理数据的对象是帧，具有寻址能力和路径选择能力，以确定帧的传输方向，在 LAN 之间存储并转发数据帧，不会对修改收到的帧，可以兼容不同的物理层，但是增大了时延，不能控制流量，只能用于小规模的局域网。

Theorem 4.4 *要控制流量需要编号机制，而编号机制在数据链路层的 LLC 子层实现。*

网桥一定要有路径选择的功能，收到帧之后要决定正确的路径，将该帧传送到目标处，根据路径选择算法的不同，可以将网桥分为：

- 透明网桥：选择的不是最佳路由，往往使用生成树算法，也叫生成树网桥，可以防止广播风暴。
- 源路由网桥：选择的是最佳路由，会使用发现帧 (Discovery Frame) 进行探测，会通过学习更新自己所知道的路径，并选择最佳的路由

4.6.2 交换机 Switch

传统的集线器组成的计算机网络并不能增加容量，因为其逻辑上还是等价于单根电缆组成的经典以太网。而随着站点的加入，每个站点可以获得的容量下降。而交换式以太网，通过交换机可以解决这个问题，交换机中有一块连接所有端口的高速背板。

Theorem 4.5 *交换机相比于集线器，没有冲突，并且可以同时从不同的站发出多个数据帧，这些帧到达交换机的端口后会穿过背板并输出到合适的端口，因此性能更好。交换机中还有一个缓冲区，可以缓冲有多个数据帧输出到同一个端口的情况。*

以太网交换机的原理是检测一台端口的数据帧的源 MAC 和目的 MAC，并和系统中的动态查找表进行比较，如果数据帧的 MAC 不在表中就将其加入，然后发送数据帧给对应的端口。

目前交换方式主要有直通式交换和存储转发式交换，区别在于存储转发式交换机会先把帧缓存到高速缓冲器中，不会直接发送出去。

5 网络层 Network Layer

5.1 基本概念

5.1.1 网络层的设计

网络层里关注的是如何将源端的数据包发送到接收方，可以通过中间的路由器进行转发，数据链路层只负责将线路从一边传送到另一边，而网络层是**处理端到端数据传输的最底层**，因此网络层必须知道网络的拓扑结构，并选择适当的路径进行传输

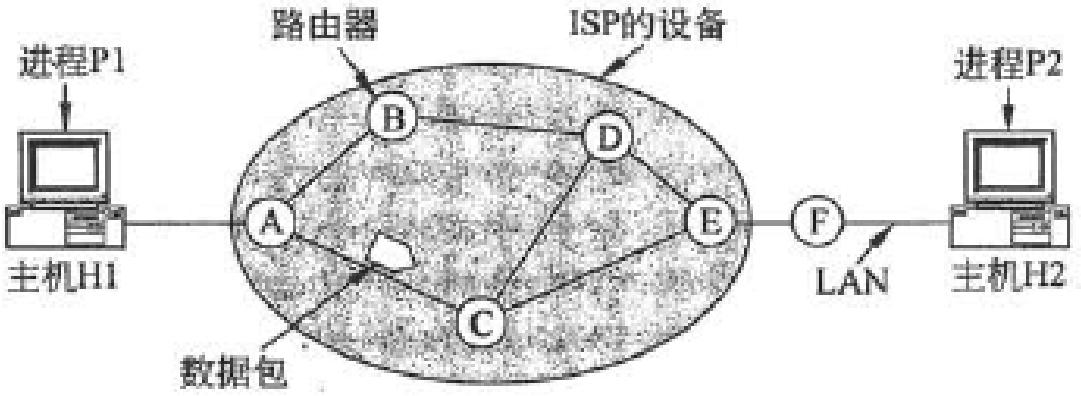


图 5-1 网络层协议的环境

Theorem 5.1 *网络层提供给传输层的服务应该独立于路由器的技术，向传输层屏蔽网络拓扑结构的具体内容，传输层可用的网络地址应该有一个统一的编址方案。*

5.1.2 网络层提供的服务

网络层提供如下服务：

- 无连接的服务：
  - ★ 每个数据包独立路由，不需要任何预先的设置
  - ★ 此时的数据包通常也称为数据报 (datagram)，对应的网络称为数据报网络
  - ★ 每个路由器中都有一个内部表，指明了针对每个可能的目标地址应该将该数据包送到哪里去
- 面向连接的服务：
  - ★ 在发送数据包之前先建立起一条虚电路，对应的网络称为虚拟点电网络
  - ★ 一个例子是多协议标签交换 MPLS

关于虚电路和报文服务的区别在物理层已经讲过了。

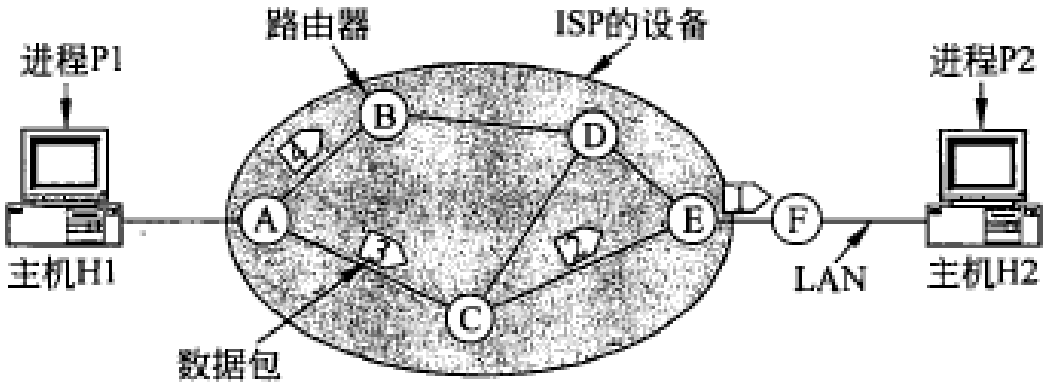
5.2 路由器和路由算法

5.2.1 路由器

路由器是网络层的核心设备，有路由和转发等功能

问题	数据报网络	虚电路网络
电路建立	不需要	需要
寻址	每个包包含全部的源和目标地址	每个包包含简短的VC号
状态信息	路由器不保留连接状态	针对每个连接，每条VC 都需要路由器保存其状态
路由方式	每个数据包被单独路由	建立VC 时选择路由，所有包都遵循该路由
路由器失效的影响	没影响，除了那些路由器崩溃期间丢失的包	穿过故障路由器的所有VC都将中断
服务质量	困难	容易，如果在预先建立每条VC 时有足够的资源可分配
拥塞控制	困难	容易，如果在预先建立每条VC 时有足够的资源可分配

Theorem 5.2 路由器内部表的每一项由两个部分组成：目标地址和通往目标地址所使用的出境线路，管理路由表更新的算法称为路由算法。



A的表(初始化)

A	—
B	B
C	C
D	B
E	C
F	C

A的表(稍后)

A	—
B	B
C	C
D	B
E	B
F	B

C的表

A	A
B	A
C	—
D	E
E	E
F	E

E的表

A	C
B	D
C	C
D	D
E	—
F	F

5.2.2 静态路由算法

路由算法是网络层软件的一部分，负责确定一个入境数据报应该被发送到哪条线路上，需要有一定的鲁棒性和稳定性，能够处理网络拓扑结构和流量的各种变化。

路由算法有两种功能，一个是路由，即更新路由表，另一个是转发，即将传递过来的数据包选择合适的路径发送出去。

路由算法可以分为自适应算法和非自适应算法，非自适应算法不会根据流量和拓扑结构来调整路由的策略，因此也叫做**静态路由**，自适应算法则会根据流量和拓扑结构来改变路由的策略，也叫做**动态路由**

Definition 5.1 最优化原则：网络拓扑结构中的最优路径的子路径一定也是一条最优子路径。从每个源端到目标的最优路径构成的集合构成了一棵以目标节点的树，称为汇集树 $SinkTree$

下面是一些常见的路由算法：

最短路径算法：就是数据结构中学过的 Dijkstra 算法，问题在于边的长度如何定义，一般都用跳数 hop 或者网线的长度来作为拓扑图中的边长

Definition 5.2 泛洪算法：每个路由器必须根据本地的知识而不是网络的全貌来做决策，泛洪算法将每个入境的数据包发送到了除了来路以外所有的出境线路。问题在于会产生大量的重复数据包，需要采取一定的措施来抑制网络拓扑结构中真的泛洪，一种方法是在数据包中添加计数器，每跳一次就减小 *1*，当计数器变成 *0* 的时候就丢弃这个数据包。

Theorem 5.3 最短路径算法和泛洪算法都是静态路由算法。

5.2.3 距离矢量路由算法 Distance Vector Routing

距离矢量算法中，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佳距离和所使用的链路，通过邻居之间相互交换信息而不断被更新，最终每个路由器都可以了解到到达目标的最佳链路。

路由表以网络中的每个路由器作为索引，并且每个路由器作为表中的一项，该表包含**到达目标路由器的首选路线和距离的估计值**，每个路由器收到了相邻的路由器发来的矢量之后就会更新自己的路由表。

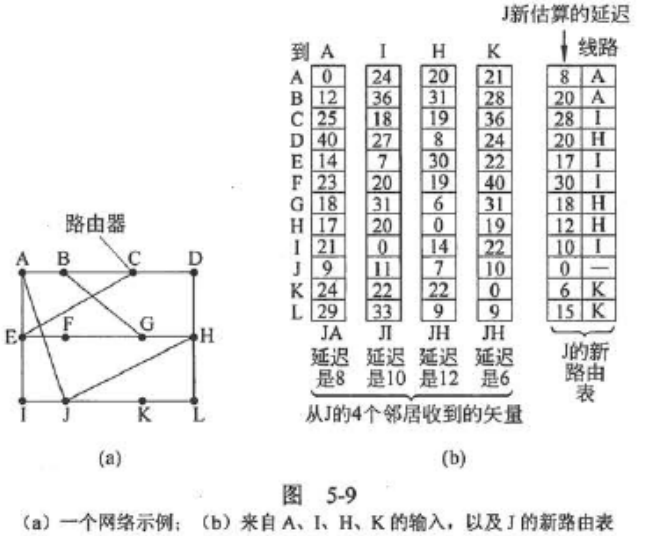


图 5-9

(a) 一个网络示例；(b) 来自 A、I、H、K 的输入，以及 J 的新路由表

整个网络最佳路径的寻找过程称为收敛，距离矢量算法可以收敛到一条最短的路径，但缺点是速度非常慢。距离矢量算法对好的结果反应特别迅速，对坏消息的反应非常迟钝。

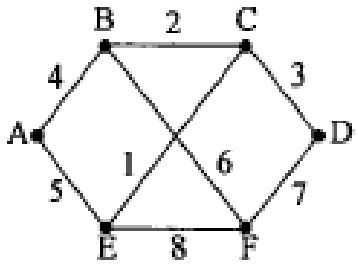
Definition 5.3 某些情况下会因为路由表的互相更新导致了路由表的计算出现无限循环的情况，这就是无穷计数问题。可以用逆毒传染（*Poisoned Reverse*）方法解决

Definition 5.4 逆毒传染：在基于路由信息协议的网络中，当一条路径信息无效之后，路由器并不马上从路由表中将其删除，而是用无穷大作为其路径长度并将信息广播出去，但是该方法不能完全解决无穷计数问题。

5.2.4 链路状态路由算法 Link State Routing

该算法需要每个路由器需要完成如下五个步骤：

- 发现邻居节点，并了解其网络地址
- 设置邻居节点的距离或者成本度量值
- 构造一个包含刚才所得信息的**链路信息包**
- 将包发送给所有的路由器，并接受来自所有其他路由器的信息包
- 计算出到达每个路由器的最短路径



(a)

链路状态包

A	
序号	
年龄	
B	4
E	5

B	
序号	
年龄	
A	4
C	2
F	6

C	
序号	
年龄	
B	2
D	3
E	1

D	
序号	
年龄	
C	3
F	7

E	
序号	
年龄	
A	5
C	1
F	8

F	
序号	
年龄	
B	6
D	7
E	8

(b)

图 5-12

（a）一个网络示例；（b）该网络的链路状态包

5.2.5 层次路由

路由器被划分成了不同的区域，每个路由器知道如何将数据包路由到自己所在的区域内的布标地址，但是对于其他区域的内部结构不知情当网络被互相连结到一起的时候，每个网络就会作为一个独立的区域，一个网络中的路由器不必知道其他网络的拓扑结构。

相当于路由器被划分成了若干个小的网络，互相之间屏蔽了单个路由器的具体信息，就像墙一样。

5.2.6 广播路由

是一种浪费带宽的行为，并且需要源头拥有所有目标机器的完整地址列表，不够理想但是被广泛使用。

多目标路由：每个数据包包含了一组目标地址或者位图。

5.3 网络层的服务质量

从源端发送到一个接受方的数据包称为一个流 (Flow)，在面向连接的网络中，每个流需要用带宽、延迟、抖动和丢失四个参数来决定其服务质量 (Quality of Service, QoS)

5.3.1 准入控制

服务质量的保证通过准入控制的过程来建立。用户提供一个有 QoS 的流，然后网络根据自己的容量以及向其他流做出的承诺来决定是否接受这个流，如果接收就需要预留路由器的容量，用于保证服务质量。

如果路由算法寻找到的最佳路径上面没有足够的容量，那么这种路由算法就可能导致某些流被拒绝，如果新的流所需要的带宽超过了剩余容量，那么通过选择另外一条可以使用的路径仍可以保证服务质量，这种路由就叫做 QoS 路由。

5.3.2 综合服务与 RSVP

RSVP 资源预留协议，是综合服务体系结构中最主要的协议，该协议的主要功能是预留资源，发送数据则需要使用其他协议，允许多个发送方给多个接受组传递数据，也允许接受方自由切换频道，在消除拥塞的同时优化宽带的使用。最简单的情况下 RSVP 使用了基于生成树的组播路由。

5.3.3 分区服务

事实上综合服务在大量的流面前是不实际的，因此 IETF 设计了一种更简单的 QoS 方法，该方法很大程度上由路由器在本地实现，无需提前设置好流，也不牵涉每条路径，这种方式就叫做分区服务 Differentiated Service，可以由一组路由器提供，这些路由器构成了一个管理域，有加速转发和确保转发的服务类别。

5.4 IP 协议

网际协议 IP 是 TCP/IP 体系中最重要的协议之一，用来连接计算机网络进行通信，被 IP 协议所连接起来的计算机网络可以看成是一个 **虚拟互联网络**，因为不同的计算机网络之间的物理介质的异构性是客观存在的，而 IP 协议使得性能各异的网络在网络层被统一为一个统一的网络

IP 协议分为 IPv4 和 IPv6 两种，目前依然是 IPv4 协议为主，为了连接不同的网络，经常需要用到一些中间设备：

- 物理层：转发器 Repeater
- 数据链路层：网桥 Bridge，也叫桥接器
- 网络层：路由 Router

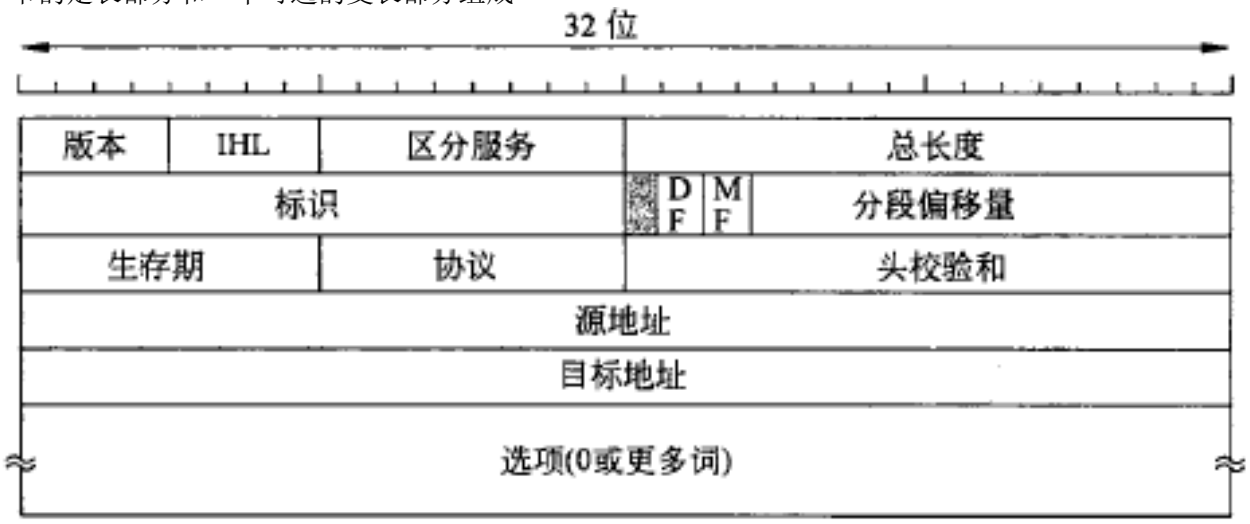
- 网络层以上使用的中间设备是网关 (Gateway), 可以用来连接两个不兼容的系统，但是需要在高层进行协议的转换

互联网络的通信过程：

- 从传输层获取数据流，并拆分成若干段作为 IP 数据包发送
- 每个数据报理论上最大容量是 64KB，但往往不会超过 1500bytes
- IP 路由器转发每个数据包穿过互联网络，沿着一条路径把数据包从一个路由器转发到下一个路由器，直到到达目的地
- 在接收端，网络层将数据交给传输层，传输层再交给接收进程

5.4.1 IPv4 协议的协议头

每个 IP 协议的数据报包含两个部分，即头和正文，正文部分也叫做有效净荷，而协议头一般有 20 个字节的定长部分和一个可选的变长部分组成



- 版本字段记录了协议属于哪个版本，占 4bit，目前比较多的依然是 IPv4
- IHL 字段表明了**协议头的长度**，以 32bit 为单位，最小值是 5 表示没有可选项，最大值是 15，因此可选项最多有 40bytes
- 总长度包含了**数据报中所有内容的总长度：头 + 数据**
- 标识用来让目标主机确定一个新到的分段属于哪个数据报，**同一个数据报的标识相同**
- 有一个 1bit 的空位
- 1bit 的 DF 表示数据报是否分段，1bit 的 MF 表示更多的段，除了最后一段之外所有段都必须要设置 MF

- 分段偏移量表示当前段在数据报中的位置，该字段有 13 位，所有段的长度必须要是 8 的倍数，因此最多有 8192 个段
- 生存期是一个计数器，每次一跳之后计数器都会减小，减小到 0 的时候就会被丢弃
- 协议代表数据报需要被交给哪个传输进程，可能是 TCP 和 UDP
- 头校验和用来校验和保护地址，**只用于 IP 包的头部的检验**
- 源地址和目标地址都是 32 位的 IP 地址

5.4.2 IP 地址

IPv4 协议中的 IP 地址是一个长为 32bits 的地址，网络中的每一台主机和路由器都有 IP 地址。

Theorem 5.4 *IP 地址是一种逻辑地址，是通过软件实现的，而物理层和数据链路层都是物理地址。IP 数据报一旦交给了数据链路层就会被封装成 MAC 帧，MAC 帧传输的时候用的是硬件地址。*

Theorem 5.5 *IP 地址并不是指向一台主机而是一个网络接口，所以如果一台主机在两个网络上，它就必须拥有两个 IP 地址，而路由器有多个网络接口，所以一个路由器有多个 IP 地址。*

IP 地址具有一定的层次性，每个 32 位地址由高位的可变长网络号和主机的主机号，同一个网络上所有主机的网络号都是相同的，这一段就是前缀。

Definition 5.5 *子网掩码（subnet mask）：IP 地址中的前缀就是子网掩码，常见的有 8 位、16 位和 24 位等类型，比如 128.208.101.21/24 表示 32 位的 IP 地址中的子网掩码是 24 位，后面 8 位就是主机地址。*

层次化的地址使得路由规模化了，但是也造成了地址的浪费，IP 地址在这种模式下可以分为五种类型：

- A 类 IP 的子网掩码有 8 位并且第一位是 0，范围是 1.0.0.0-127.255.255.255
- B 类有 16 位且前两位是 10，范围书 128.0.0.0 到 191.255.255.255
- C 类地址的子网掩码有 24 位且前两位是 110，范围是 192.0.0.0 到 223.255.255.255
- D 类 IP 是 1110 开头的多播地址，范围是 224.0.0.0 到 239.255.255.255
- E 类地址的前四位 1111，范围是 240.0.0.0 到 255.255.255

Definition 5.6 *子网 Subnet: 在一个网络的内部将网络划分成几个内部网络，但是对外仍然表现得像一个网络一样，这样划分出来的就是子网*

5.4.3 无类域间路由 CIDR

上面说的按块分配子网依然解决不了路由表爆炸的问题，尤其是位于默认自由区的骨干路由器，其路由表的规模依然非常大，因此我们找到了一种新的解决办法：

Definition 5.7 路由聚合 *RouteAggregation*：可以运用和子网划分相同的方法，不同地点的路由器可以知道一个给定 *IP* 地址的不同大小的前缀，并将多个小前缀的地址合并成一个大的前缀，此时产生的叫做超网。

事实上这就是划分子网的逆过程，这一过程也叫做地址聚合，通过地址聚合，IP 地址可以包含大小不等的多个前缀，同一个 IP 地址可以作为不同长度的前缀被多个路由器使用，这个设计就叫做无类域间路由 (Classless Inter-Domain Routing)

5.4.4 网络地址转换 NAT

因为 IP 地址非常有限，因此我们可以采取动态分配 IP 地址的策略，在主机不活跃的时候回收 IP 地址然后分配给别人，但是这个办法依然有其局限性，因为不同的计算机对联网时间的长度有不同的需求。

NAT 的基本思想就是给每个客户网络 (比如家庭或者公司) 分配一个 IP 地址用来传输流量，但是在客户网络的内部，每台计算机都有唯一的 IP，该 IP 主要用于路由客户网络内部的流量，当一个数据报离开客户网络发到外面去的时候，就必须进行地址转换，**把内部 IP 转换成公共的 IP 地址**，目前已经有三个保留的地址范围，任何网络内部可以自由使用这些地址，并且这些地址不允许出现在因特网上，比如 192.168.0.0-192.168.255.255/16 就是一个保留的 IP 段，可以支持 65536 台主机。

NAT 也带来不少问题，不少 IP 的原教旨主义者认为 NAT 违反了 IP 最基本的结构模型，即每一台机器对应一个 IP 地址，它打破了端到端的连接模型，并且将因特网从无连接的网络变成了面向连接的网络，违反了最基本的协议分层规则。并且 NAT 只支持 TCP/UDP 的传输层协议。

5.4.5 IPv6

IPv6 解决了从根本上解决了 IPv4 地址耗尽的问题，将地址扩大到了 128 位，依然采用扩展的层次结构和头部格式，IPv6 的数据报的目的可以支持单播、多播和任播，任播是新增加的类型，任播面向一组计算机，但是数据报只交付给其中一台计算机。

IPv6 的地址采用十六进制表示，每四位用一个十六进制数来表示，每四个十六进制数用冒号分隔，一共有八组，但是也可以用两个冒号代替连续的若干个 0

5.5 网络层的协议

5.5.1 控制消息协议 ICMP

ICMP 是路由器监视 Internet 的协议，当路由器处理数据包的时候发生了以外，就可以用 ICMP 向数据包的源端报告有关的事件，所有的 ICMP 都被封装在一个 IP 数据包里，ICMP 分为以下几个消息类型：

消息类型	描述
目的地不可达	数据包无法传递
超时	TTL字段减为0
参数问题	无效的头字段
源抑制	抑制包
重定向	告知路由器有关地理信息
回显和回显应答	检查一台机器是否活着
请求/应答时间戳	与回显一样，但还要求时间戳
路由器通告/恳求	发现附近的路由器

5.5.2 地址解析协议 ARP

以太网的网卡并不能理解 IP 地址，每一块 NIC 根据 48 位的以太网地址来发送和接收帧，ARP 协议的作用就是将 IP 地址映射成以太网的地址，优点在于简单，既可以算作网络层的协议，也可以算作数据链路层的协议。

ARP 协议解决转换问题的方法是在主机的 ARP 高速缓存中存放一个 IP 地址到硬件地址的映射表，并进行动态的更新，每一台主机上都设有一个 SRP 高速缓存，里面有本局域网上的各台主机和路由器的 IP 到硬件地址的映射表。

当一台主机 A 要想本局域网上的主机 B 发送 IP 数据报的时候就会先在缓存中查看查出对应的硬件地址然后写入 MAC 帧中进行发送，否则需要进行一系列的复杂操作。

5.5.3 反向地址解析协议 RARP

其实就是让物理机器从 ARP 表中请求其 IP 地址，维护一个 MAC 地址到 IP 的映射表，将 MAC 地址转换成对应的 IP 地址，可以用于以太网、光纤分布式数据接口以及令牌环。

5.5.4 动态主机配置协议 DHCP

每个网络必须有一个 DHCP 服务器负责地址的配置，为发送请求的主机分配一个空闲的 IP 地址，并通过 DHCP 的 OFFER 包返回给主机。但是动态分配的 IP 地址只能持续一段时间，到期之前主机必须请求 DHCP 续订，否则时间到了之后主机的 IP 地址就会被取消。

5.5.5 标签交换和 MPLS

多协议标签交换 (MPLS) 是指在每个数据包前面增加一个标签，路由器根据数据包标签进行转发，用标签作为内部表的一个索引，快速查找出正确的输出路线。通用的 MPLS 有 4 个字节，包含 4 个字段，标

签存放的是索引，QoS 表明服务类别，S 表示在层次网络中叠加多个标签的做法，TTL 字段指出该数据包还可以被转发多少次，每经过一个路由器 TTL 减小 1，变成 0 的时候数据包就会被丢弃

5.6 路由协议

5.6.1 自治系统 Autonomous System

自治系统是单一技术管理下的一组路由器，这些路由器使用一种内部的路由选择协议和共同的度量来确定 AS 内部的路由。AS 需要内部网关协议 IGP 和外部网关协议 EGP

5.6.2 路由信息协议 RIP

路由信息协议 (Routing Information Protocol) 是一种分布式的基于距离向量的路由选择协议，比较简单，规定每个路由器都要维护距离向量，而距离也称为跳数，规定从一个路由器到直接连接网络的跳数为 1，RIP 认为最佳路由就是通过的路由器的数目少，即 RIP 会优先选择跳数便较少的路径，一条路径最多只能包含 15 个路由器，默认在每 30s 广播一次更新信息，并且动态维护路由表。

相比于 OSPF，RIP 只和相邻的路由器交换信息，这个信息也就是路由表，并且有固定的时间间隔。

5.6.3 内部网关路由协议 OSPF

OSPF 的全称是开放最短路径协议 (Open Shortest Path First)，借鉴了 IS-IS(中间系统到中间系统) 协议，已经成为了 ISO 的标准，它具有以下特点：

- 使用了分布式的**链路状态协议和最短路径算法**
- 是一种动态的算法，支持多种距离度量
- 实现了均衡负载，使用层次化的网络系统
- 同时支持点到点的链路和广播网络

OSPF 将自治系统划分成了若干个 area，每一个都是一个单独的网络，每个自治系统有一个骨干区域。路由器也分为区域边界路由器和内部路由器、自治系统边界路由器和骨干路由器等等，其中自治系统边界路由器可以把通往其他自治系统的外部路由注入到本区域中。

OSPF 的工作方式本质上是对一张图进行操作，将一组实际网络、路由器和线路抽象到一个有向图中，路由器之间的连接可以用两条有向的弧来表示，**OSPF 会记住最短的路径集合，并在报文转发阶段把流量分摊到多条路径上面，实现负载均衡**，这种方法称为等价成本多路径 (ECMP)

OSPF 的消息类型有五种，在邻接的路由器之间进行传递，分别是：

- 问候分组，用来发现和维持邻居的可达性
- 数据库描述分组，给出自己的链路状态数据库中所有信息
- 链路状态请求分组，向对方请求阿松某些链路状态项目的详细信息
- 链路状态更新分组，用 flood 更新全网的链路状态
- 链路状态确认分组，对链路更新分组的确认

5.6.4 外部网关路由协议 BGP

BGP 的全称是边界网关协议 Border Gateway Protocol，在一个自治系统的内部推荐使用 OSPF 和 ISIS，而在自治系统之间用 BGP 比较好，这是因为域内协议和域间协议的目标不同，域内协议所需要做的只是尽可能有效地将数据包从源端发送到接收方，而域间的路由协议则必须要考虑大量的选择策略，比如是否接收某个特定自治系统的消息等等。

Theorem 5.6 *BGP 协议只能力求寻找到一条可以达到目的网络并且比较好的路由，采用了路径向量路由选择协议。*

BGP 使用四种报文，分别是打开，更新，保活和通知

5.6.5 三种路由协议的比较

RIP、OSPF、BGP 三种协议的区别有：

- RIP 使用距离向量算法，传递 UDP 协议，选择跳数最少的路径
- OSPF 使用链路状态算法，传递 IP 协议，选择代价最低的路径
- BGP 使用路径向量算法，传递 TCP 协议，选择比较好的路径

5.7 IP 组播和移动 IP

5.7.1 Internet 组播

IP 的组播一定只用在 UDP 协议上，普通的 IP 通信只发生在一个发送方和一个接收方之间，而 IP 用**D 类 IP 地址来支持一对多的通信或者组播**，每个 D 类地址标识了一组主机。Internet 组管理协议 (IGMP) 用来解决广播成员分布在不同网络上面的情况，在自治系统中主要使用协议独立组播协议 PIM，反正这一大堆莫名其妙的协议讲了什么说实话我也没看懂，暂且放到一边算了。

值得注意的是，组播一次只发送一份数据。

5.7.2 移动 IP

移动 IP 满足了移动节点保持连接性的需求，一个移动节点可以在不改变 IP 地址的情况下改变其物理位置，基于 IPv4 的移动 IP 定义了三种功能实体，移动节点、本地　理和外部代理，本地和外部的代理又称移动代理。

- 移动节点拥有永久 IP 地址
- 本地代理：代表移动节点在归属网络中执行移动管理功能，采用隧道技术
- 外部代理：在外部网络中帮助移动节点完成移动管理功能

在移动 IP 中，每个移动节点都有一个唯一的本地地址，时不变的，在本地网络链路上每个本地节点还必须有一个本地代理维护移动节点的位置信息，这需要用到转交地址技术，当移动节点连接到外地网络时，转交地址就用来标识移动节点的位置进行路由选择。移动节点的本地地址和当前转交地址的联合称为移动绑定，

当移动节点得到一个新的转交地址的时候，通过绑定向本地代理进行注册，以便让本地代理了解位置。也就是说移动设备有两个地址，主地址在本地网络使用，辅助地址在别的网络中使用，通过代理来维护位置信息。

6 运输层 Transport Layer

6.1 基本概念

6.1.1 运输层的作用

运输层向位于其上的应用层提供通信服务，是**用户功能中的最底层**，也是属于面向通信部分的最高层。计算机通信的真正端点并不是主机，而是**主机之中的进程**，也就是说端到端的通信其实是两个应用进程之间的通信。

运输层提供了进程之间的逻辑通信，而网络层则提供了主机之间的逻辑通信，基于这种功能，运输层最重要的功能是复用 (multiplexing) 和分用 (demultiplexing),

- 复用：发送方不同的进程可以用同一个传输层的协议来传输数据
- 分用：接收方的运输层在处理完报文之后可以把数据交付给正确的接收方进程

6.1.2 运输层的重要协议 TCP 和 UDP

TCP 是传输控制协议，UDP 是用户数据报协议，两个 peer 在进行通信的时候传输的数据单位叫做运输协议数据单元 (TPDU)，但是在 TCP/IP 体系中，TCP 传输的数据单元被称为 TCP 报文段，而 UDP 传输的则被称为 UDP 用户数据报

Theorem 6.1 *TCP 提供的是面向连接的服务，在传输之前必须先建立连接，传输结束之后则需要释放连接，并且没有多播和广播的服务，UDP 是无连接的协议，远程的主机的运输层在收到 UDP 之后不需要给出任何的确认*

6.1.3 通信端口 Port

在物理层等比较底层的结构中端口指的是物理端口，而传输层的端口指的是软件端口，是应用层的各种协议进程和运输层之间进行交互的一种地址。TCP/IP 体系中的运输层一般用 16 位的端口号，最多支持 65535 个不同的端口。

端口也叫做传输服务访问点 (TSAP)，常用的端口中，已经有一部分端口被分配为固定的用途，比如：

端口	协议	用途
20, 21	FTP	文件传输
22	SSH	远程登录，Telnet的替代品
25	SMTP	电子邮件
80	HTTP	万维网
110	POP-3	访问远程邮件
143	IMAP	访问远程邮件
443	HTTPS	安全的Web（SSL/TLS之上的HTTP）
543	RTSP	媒体播放控制
631	IPP	打印共享

其中 0-1023 是常用

端口号，基本都有固定的用途

6.1.4 传输协议的基本要素

- 寻址:建立通信的时候指定数据要连接到哪个应用进程上面,往往是使用一个端口映射器 (port mapper) 进程来处理端口和服务的映射关系
- 建立连接：需要拥有一套建立连接的算法，常见的有 TCP 的三次握手
- 释放连接：分为非对称的释放和对称的释放，非对称释放是一段释放即可，而对称释放需要两端都进行释放
- 差错控制和流量控制，对于低带宽的突发流量可以不使用缓冲区，但是对于文件传输和高带宽流量就需要使用缓冲区
- 多路复用和逆多路复用，到达的数据段需要用某种方式告知它需要交给哪个进程处理，逆多路复用则是将应用层发出来的数据分散成多段发送
- 崩溃恢复，在路由器或者主机崩溃的时候重新发送

6.2 用户数据报协议 UDP

6.2.1 UDP 的特点

用户数据报协议 UDP 只在 IP 协议上增加了复用和分用的基本功能，主要的特点有：

- 无连接，发送数据不需要建立连接，减小了开销和延迟，协议头只有 8 字节，比 TCP 轻便
- 面向报文，一次交付一个报文，对应用层提供的报文**既不拆分也不合并**，直接添加头部之后交给 IP
- 尽最大努力交付，也就是不保证可靠的交付，因此主机不需要维持复杂的连接状态
- 没有拥塞控制，网络中出现拥塞时不会降低发送速率，允许拥塞的时候丢数据但不允许高延迟，因此可能会引发网络拥堵
- 支持一对一、一对多、多对多的交互通信

6.2.2 UDP 的 header

UDP 的头部一共有 8 个字节，其中源端口、目标端口、数据报长度和检验和各占两个字节，数据报长度的最小值是 8(只有一个头)，检验和可以用来检验数据报在传输过程中是否出错。

当运输层收到一个 UDP 数据报的时候，根据头部的目标端口将 UDP 数据报通过相应的端口上交给应用进程，如果发现端口号不存在，就会丢弃这个报文并使用 ICMP 发送“端口不可达”的信息给发送方

6.2.3 UDP 的应用

UDP 的应用主要有：实时传输协议 RTP，实时传输控制协议 RTCP，域名系统 DNS，在防抖动和缓冲的音视频播放中也有一定的应用。

6.3 传输控制协议 TCP

TCP 协议比 UDP 协议要复杂，有如下特点：

- 面向连接：使用之前必须建立连接，使用完之后必须释放
- 面向字节流，将应用层提供的内容都看成是无结构的数据流
- 点对点连接，每一跳 TCP 连接有且仅有两个端点
- 全双工通信，一台主机可以同时作为发送方和接收方
- 提供可靠交付，无差错，不丢失，不重复并且按既定顺序到达

6.3.1 TCP 中的连接和 socket

连接是 TCP 中最基本也是最重要的一个抽象，每条 TCP 连接都有两个端点，被称为套接字 socket，IP 地址 + 端口号 port 就可以构成一个 socket，每一条 TCP 连接由两个 socket 来确定，一个 IP 地址可以有多个 socket

Berkerley Socket 就是在操作系统内核实现的一个 socket 封装库，提供了如下操作：

- SOCKET：创建一个 socket
- BIND：绑定一个 IP 地址
- LISTEN：开启一个监听队列对收到的信息进行监听
- ACCEPT：向发送方发送成功接收的通知
- CONNECT：尝试和另一个 socket 进行连接
- SEND：发送数据
- RECEIVE：接收数据
- CLOSE：释放 socket 连接

6.3.2 TCP 头部

一个 TCP 报文段由头部和数据两部分组成，整个 TCP 报文段**作为 IP 数据报的数据部分封装在 IP 数据报中**，而 IP 数据报也有 20 字节的头部，因此一个数据段的最大长度就变成了 65535-20-20=65495，TCP 的头部有至少 20 个字节，其主要结构和不同部分的功能如下：

- 源端口和目标端口表示连接的本地端点，各占 2 字节，TCP 地址 +IP 头中的 IP 地址可以组成一个 48 位的唯一表示
- 序号和确认号都是 4 字节，**序号指的是本报文段所发送的数据的第一个字节在全文中的序号，确认号是期望收到对方的下一个报文段的数据的第一个字节的序号**，而不是当前报文段最后一个字节的序号

- TCP 的头部长度占 4 字节，表示头部的长度，因为有 option 的存在，头部长度是不固定的，option 一定是 4 字节的倍数，因此 TCP 的头部最大只有 60 字节
- 保留字段占 4 字节，后面还有 8 位标志位：
  - ★ CWR 和 ECE 用作拥塞控制
  - ★ URG 是紧急位，URG=1 表明紧急指针字段有效
  - ★ ACK 是确认位，ACK=1 表示确认号字段有效，否则无效
  - ★ PSH 是推送位，收到 PSH=1 的报文段就应该尽快交付应用进程而不等待整个缓存填满再交付
  - ★ RST 是复位位，RST=1 表示连接出错必须释放然后重新建立连接
  - ★ SYN 是同步位，SYN=1 表示这是一个连接请求或者接收报文
  - ★ FIN 是终止位，FIN=1 表示可以释放连接
- 窗口大小占 2 字节，它指出 TCP 的**滑动窗口的剩余大小，表明接收方还能接受多少字节**
- 校验和占 2 字节，可以用来检验头部和数据两个部分
- 紧急指针占 2 字节，表示本报文中紧急数据共有多少字节，这些内容都放在数据段的最前面
- 选项占 4 字节的倍数 (可以为 0)，提供了添加 TCP 额外设施的途径

6.3.3 TCP 建立连接与三次握手

TCP 建立连接需要每一方都能知道对方的存在，要允许双方协商一些参数，能够对运输实体资源进行分配，TCP 使用三次握手的方式建立连接，三次握手分为如下三个步骤：

- 第一步：A 向 B 发送 SYN=1 和起始序列号 seq=x
- 第二步：B 收到之后如果同意建立连接就发送确认，设置 SYN=ACK=1 和 B 段产生的 seq=y，确认号字段的值是 ack=x+1
- 第三步：A 收到 B 的同意请求，并告诉 B 收到了来自 B 的同意请求，ACK=1,seq=x+1，ack=y+1

6.3.4 TCP 连接释放与四次握手

而在 TCP 释放连接的时候更麻烦，TCP 需要进行四次握手才能完成连接的释放，具体步骤是：

- A 要关闭连接的时候通知 B，发送一个释放报文段，FIN=1，seq=u
- B 收到这个报文之后发出确认，确认号是 ack=u+1，而自己的序号 seq 是 v
- 若 B 已经没有需要向 A 发送的数据就通知 A 进行释放，此时 FIN=1，ACK=1，seq=w，ack=u+1
- A 收到通知之后需要发出确认，ack=w+1，seq=u+1，等待计时器设置的时间 2MSL 之后,A 才能关闭连接

6.3.5 TCP 的可靠性保证

事实上 TCP 的数据报发给网络层之后由网络层发送到网络中去，但是 IP 协议是“尽全力发送的”，因此并不可靠，TCP 采取了一些措施来保证传输的可靠性。主要包括以下几种：

- 序号：TCP 头部的序号字段用来标识数据段的顺序，这些序号建立在无结构的数据流而不是报文段上，每个字节都有一个序号，而头部的序号则是本数据报中的第一个字节的序号，接收方收到数据报之后需要根据顺序重新组合出正确顺序的数据
- 确认号：TCP 通过确认号 ACK 来标识希望收到的下一个报文段的数据的第一个字节的序号，据此来组合收到的若干数据段
- 重传：TCP 协议会在超时和收到冗余 ACK 的时候进行重传。超时通过设置计时器来确定，但是效率非常低，而冗余 ACK 会在接受方没有收到期望的数据报的时候进行发送，比如发送了 1，2，3，4，5 五个数据报，而 2 丢失了，那么接收方在收到 345 的时候因为不是期望的数据报，就会给发送方发送三个 111 的 ACK，这种技术叫做快速重传。
- 滑动窗口：以字节为单位

6.3.6 TCP 的流量控制

流量控制，即使得发送方的发送速率和接收方的接受速率相匹配，TCP 使用滑动窗口协议来进行流量控制，接收方根据自己剩余缓冲区的大小调整发送方的发送窗口的大小，叫做接收窗口 rwnd，通过设置 TCP 头部的窗口大小来实现，发送方根据其对当前网络拥塞程序的估计而确定的窗口值，称为拥塞窗口 cwnd。

Theorem 6.2 *传输层和数据链路层的流量控制的区别在于，传输层是端到端的流量控制，而数据链路层是两个相邻节点的流量控制，此外数据链路层的滑动窗口大小不能改变。*

6.3.7 TCP 计时器管理

TCP 协议使用了多种计时器来进行时间的管理：

- 重传计时器 RTO：每次 TCP 发出一个段就启动一个重传计时器，如果超时了就要重传
- 持续计时器：为了避免出现死锁现象
- 保活计时器：当连接空闲了较长时间之后，保活计时器可能会超时，从而促使一端来查看连接是否还存在，如果另一端没有响应就断开 TCP 连接

6.3.8 TCP 拥塞控制

拥塞控制就是防止网络中出现因为数据过多而产生拥塞和过载，流量控制中发送方拥有接收窗口和拥塞窗口两种窗口，而发送窗口的大小的上限值就是两个里面的较小值。而发送窗口的大小主要取决于网络的拥塞情况，因此可以将发送窗口等同于拥塞窗口

Definition 6.1 *缓慢启动算法：TCP 刚创建连接并发送报文段的时候，先令拥塞窗口 =1，即一个最大报文段的长度 MSS，每次收到对发送出去的报文的确认的时候就将拥塞窗口增大 1，逐渐增大到一个阈值 ssthresh，然后使用拥塞控制算法*

Definition 6.2 *拥塞控制算法：在缓慢启动算法达到阈值的时候就使得拥塞窗口的增长速度变成线性，每次遇到网络拥塞就将阈值变成当前拥塞窗口的一半，然后从头开始继续缓慢启动算法，如此循环往复*

Definition 6.3 *快速重传算法：使用冗余 ACK 来检测网络拥塞，当连续收到三个冗余 ACK 的时候直接对未收到的报文段进行重传而不等待超时。*

Definition 6.4 *快速恢复算法：类似于拥塞控制算法，在连续收到三个冗余 ACK 的时候就将阈值设置为当前拥塞窗口的一半，然后将拥塞窗口的值也设置为阈值，然后缓慢地线性增加。*

快速恢复算法和拥塞控制算法的区别就在于拥塞窗口不是从 1 开始重新缓慢启动

6.4 DTN 体系结构

DTN 的全称是

7 应用层 Application Layer

7.1 域名系统 Domain Name System

7.1.1 域名

域名就是一台计算机主机的名字，具有层次树状结构，比如 cspo.zju.edu.cn 实际上就有四层结构，并且从左到右的域名级别逐渐提高，不同级别之间用小数点隔开。域名都由英文字母和数字组成。当然域名只是一个逻辑概念，并不代表真实的物理位置。

域名有多种级别，常见的顶级域名有国家顶级域名比如 cn，通用顶级域名 com，基础结构域名，而二级域名、三级域名之类的就更多了。

要注意的是，**域名和 IP 地址、MAC、主机都不是一一对应关系，比如一个 IP 可以对应好几个域名**

7.1.2 工作原理

域名系统 (DNS) 可以将计算机网络中的主机名字解析成对应的 IP 地址，是一个**联机的分布式数据库系统，采用 C/S 的模式，**将大多数域名在本地进行解析而少量域名需要进行互联网的通信来解析。而完成这一解析的过程需要依赖许多的域名服务程序，域名服务器在专门设置的节点上运行。

解析的时候需要把解析的域名放在 DNS 的请求报文中，以 UDP 用户数据报的形式发给本地域名服务器，本地域名服务器把对应的 IP 地址放在回答报文中返回，而如果在本地的域名数据库中没有找到，就需要向别的域名服务器发出请求。

7.1.3 域名服务器

Definition 7.1 区：一个服务器所管辖的范围叫做区，一个区中的所有节点都是联通的，每一个区设置相应的权限域名服务器，用来保存该区中所有主机的域名，区是域的子集。

域名服务器也分为若干个区，包括以下几种：

- 根域名服务器：最高层次、最重要的域名服务器，本地域名服务器无法解析的时候会首先请求根域名服务器，采用任播的技术
- 顶级域名服务器：负责管理所有该顶级域名下面的二级域名
- 权限域名服务器：负责管理一个区的域名服务器，可以将管辖的主机名转化成对应的 IP 地址
- 本地域名服务器：也叫做默认域名服务器，位于主机本地

为了提高安全性，可以设置一个主域名服务器和若干个辅助域名服务器

7.1.4 域名的查询

主机向本地域名服务器的查询一般都是**递归查询**，即如果查不到，就由本地域名服务器向根域名服务器发送查询请求而不是让主机自己来进行下一步的查询。本地域名服务器向根域名服务器的查询是**迭代查询**，即由本地域名服务器一个个查询过去，为了提高查询的效率，域名服务器中广泛使用了高速缓存，因此查询的时候最多向外发送 4 次 DNS 请求，最少可能是 0 次。

7.2 电子邮件和相关协议

7.2.1 电子邮件系统

一个电子邮件系统由三个部分组成，分别是

- 用户代理：一般就是一个电子邮件的客户端软件，至少需要有撰写邮件、显示、处理、通信等基本功能
- 邮件服务器：必须要同时可以充当客户机和服务器
- 邮件发送协议 SMTP 和邮件读取协议 POP3，二者都使用 TCP 来传送邮件

7.2.2 邮件收发的过程

- 用户代理使用 SMTP 向发送方邮件服务器的 SMTP 服务器建立 TCP 连接并发送邮件
- 发送方的邮件服务器的 SMTP 客户向接收方的邮件服务器建立 TCP 连接，并发送给 SMTP 服务器
- 接收方的 POP3 服务器读取邮件，向收件人的用户代理 POP3 客户发送邮件的内容

7.2.3 SMTP 和 MIME

SMTP 协议有局限性，只能传输 ASCII 编码的文本文件，而不能传送可执行文件或者二进制对象，并且限于传送 7 位的 ASCII 码，会拒绝长度超过一定范围的邮件。在此基础上发展出了通用互联网邮件扩充 MIME，可以在继续使用原有的邮件格式的基础上增加了邮件主体的结构，支持可执行文件和二进制对象的传送。

7.2.4 邮局协议 POP3 和网际报文存取协议 IMAP

POP3 是一个非常简单、功能有限的邮件读取协议的第三代，使用客户服务器的工作方式，需要输入鉴别信息之后才允许对邮箱进行读取，只要用户从 POP3 服务器中读取了邮件，POP3 就会将该邮件删除。

IMAP 比 POP3 复杂得多，也使用客户服务器的方式工作，是一个联机协议，用户计算机上可以运行 IMAP 的客户程序，然后和接收方的邮件服务器上的 IMAP 服务器建议 TCP 连接，可以在不同的地方使用不同的计算机随时读取邮件。

7.3 万维网 World Wide Web 和 HTTP

7.3.1 万维网 WWW

万维网是一个大规模的联机信息储藏所（不知道是谁写的这么变扭的定义），是一个分布式的超媒体系统，是超文本系统的扩充，以客户服务器（浏览器）的方式进行工作，客户程序向服务程序发出请求，服务程序再发送回所需要的万维网文档，使用链接的方法可以非常方便地访问互联网上的站点，运用统一资源定位符和超文本传输协议来标志和发送各类文档。

7.3.2 统一资源定位符 URL

统一资源定位符可以表示从互联网中得到的资源位置和访问这些资源的方法，基本的格式是协议://主机: 端口/路径，并且不区分大小写。万维网中主要使用的就是 HTTP 协议。

7.3.3 HTTP 协议

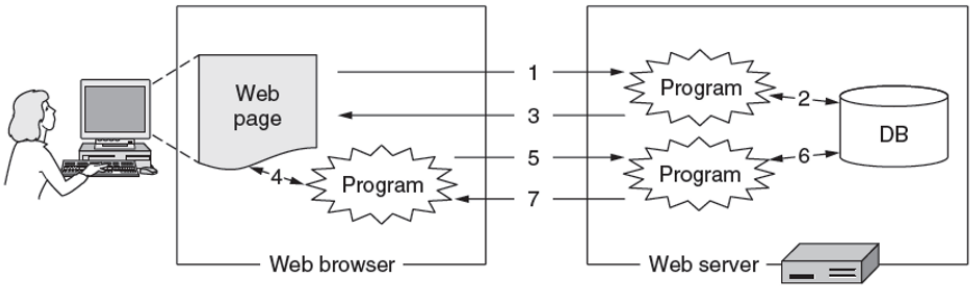
HTTP 协议定义了浏览器如何向万维网服务器请求万维网文档，以及服务器如何将文档传给浏览器，具有如下特点：

- HTTP 是**面向事务**的，可以传输文本、超文本、声音、图像等格式的信息
- **HTTP 协议使用 TCP 作为运输层的协议，需要在建立 TCP 连接之后再进行发送，保证了传输的可靠性**
- **HTTP 协议是无状态的**，即多次访问同一个服务器上的同一个页面，服务器的响应是相同的
- HTTP 协议使用 ASCII 编码

HTTP/1.1 具有持续连接的特性，有流水线式和非流水线式两种工作方式，流水线工作模式的时候，浏览器可以发送连续的 HTTP 请求响应，即不需要收到新的请求就可以发送一条新的请求出去，这样一来客户的访问时间只有一个 RTT，流水线式的工作模式使得 TCP 连接中的空闲时间减少。 HTTP 协议使用一些状态码来标注 HTTP 请求的执行情况，常见的有如下几种：

- 1xx 代表一些信息，比如 100 表示服务器同意处理客户请求
- 2xx 表示成功，比如 200 表示请求成功，204 表示没有内容
- 3xx 表示重定向，比如 301 表示页面转移，304 表示**高速缓存的页面**依然有效
- 4xx 表示客户端错误，403 表示禁止访问的页面，404 表示找不到页面
- 5xx 表示服务端错误，500 表示内部服务器错误，503 表示再试一次

上面提到了页面的高速缓存，也就是 HTTP 请求的 cache，也叫做代理服务器（proxy server）是一种网络实体，是将一些最近的 HTTP 请求和响应存储在本地磁盘中，如果发现新的请求和 cache 中的相同就不需要去服务器请求新的页面而是直接返回已经存储的响应结果。



7.3.4 HTTP 请求的格式

HTTP 的报文分为请求报文和响应报文两类，HTTP 是面向文本的，因此使用 ASCII 编码，各个字段的长度都是不确定的，但是请求和响应报文都由三个部分组成：

- 开始行：用于区分是请求报文还是响应报文，请求报文中叫做请求行，响应报文中叫做状态行，以回车换行结尾
  - ★ 请求行有方法、URL 和 HTTP 协议版本，方法一般都是 GET、POST 等等
  - ★ 响应报文中的状态行有版本、状态码和短语
- 首部行：可以有若干行，用于传输一些元信息，比如主机地址，用户代理等等，用字段名 + 冒号 +value+ 回车换行的形式存储
- 报文的实体：请求中一般没有，响应中可能会有，是 HTTP 报文的正文

首部行和报文的实体之间还有一个回车换行。

8 网络安全 Web Security

这一章主要讲网络安全相关的内容，但好像考试考的比较少，但还是应该仔细看一看。

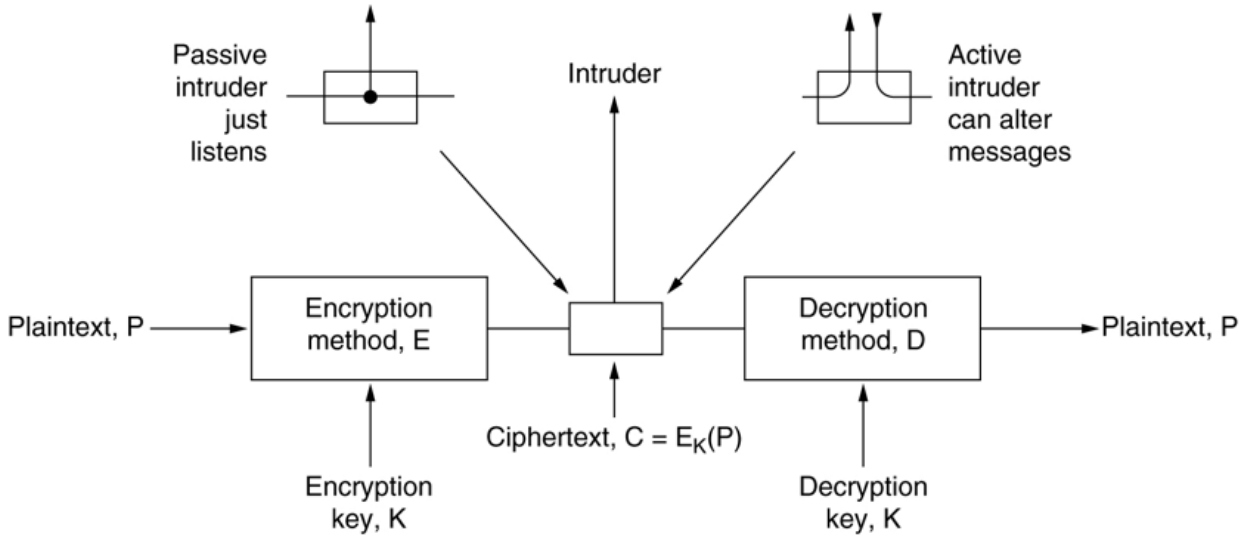
8.1 密码学相关知识

8.1.1 加密算法

古典密码学的加密算法主要有置换密码和替代密码，置换密码将每个或者每组字母用另一个或者另一组字母来代替，最古老的密码之一就是凯撒密码，采用了单个字母的置换。替代密码将明文重新排序，需要密钥进行解密

8.1.2 加密模型

任何加密模型的安全性都取决于密钥的长度以及攻破密码所需要的计算量，加密模型可以用下面的图来表示：



8.1.3 两个密码学基本原则

- 原则 1：被加密后的信息一定包含了冗余信息
- 原则 2：需要采取方法来对抗重放攻击

8.1.4 两类密码体制

密码体制主要分为对称密钥体制和公钥密码体制，对称密钥使用相同的加密密钥和解密密钥，而公钥密码使用不同的加密密钥和解密密钥，分别叫公钥和私钥，常见的加密算法有：

- DES(数据加密标准) 是对称加密，是一种分组密码，由 IBM 公司研制，有安全性更强的三重 DES

- AES(高级加密标准) 是对称加密，由 Jaon Daemen 和 Vincent Rijmen 提出
- RSA 是三个名字首字母分别是 RSA 的人提出来的加密算法，是公钥密码，以 Galois 的域理论为基础

DES 和 AES 都是块密码，因为是以一定大小的块作为单位来加密的。而公开密钥算法都是不能被选择明文攻击破解的。

为什么这里要提这么多人名就是因为听学长学姐说这几年期末考试的选择题里出现了大量莫名其妙的考察人名地名时间的题目。

8.1.5 密码攻击

Definition 8.1 重放攻击（*replay-attack*）：直接截取密码报文，不需要进行破译，而是伪装成发送方发给接受方，然后获取其回复消息，这种攻击方式可以使用不重数（*nonce*）来化解。

Definition 8.2 中间人攻击（*man-in-the-middle*）：我也没看懂是啥意思，大概就是中间人把不重数用自己的私钥加密之后，分别向发送方和接受方发送获取密钥的请求，然后获得其密钥破译密码。

8.1.6 密码散列函数

散列函数也叫做哈希函数 (hash function)，具有单向加密的特点，输入长度不固定但是输出的长度是固定的，要找到两个输出的报文在计算上是不可行的，常见的密码散列函数有：

- MD5：报文摘要算法，算法需要将报文按照规则填充成 512 的倍数，然后每个 512 位的块分成 128 位的块，128 位的再分成 32 的小块进行 hash
- SHA 是美国 NIST 机构提出的散列算法，但是码长是 160 位，比 MD5 更安全

8.1.7 密钥分配

由于密码算法是公开的，网络的安全性就完全基于密钥的保护，不同的密码体系的分配方式不同，对于对称密钥：

- 设立密钥分配中心 (KDC,Key Distribution Center)，常用的密钥分配协议是 Kerberos V5，使用界别服务器 AS 和证书授予服务器 TGS
- 证书具有一定的有效期，过期就会失效，不能被用于多次重放攻击

而对于公钥的分配，可以使用认证中心 (CA) 把公钥和对应的实体进行绑定，ITU-T 制定了 X.509 标准，并在 RFC5280 中给出了**互联网公钥基础设施 PKI**，

8.2 互联网安全协议

8.2.1 IPsec 协议族

IPsec 是可以在 IP 层提供互联网通信安全的协议族，分为三个部分：

- IP 安全数据报格式的两个协议：鉴别首部协议 AHP 和封装安全有效载荷协议 ESPP

- 有关加密算法的三个协议
- 互联网密钥交换协议 IKEP

IP 安全数据报有两种不同的方式，分别是运输方式和隧道方式。运输方式是在运输层的报文段的前后分别添加若干控制信息再加上 IP 头部，隧道在原始的 IP 数据报的前后添加控制信息，再加上新的 IP 首部构成一个 IP 安全数据报。

安全关联 SA 是发送 IP 安全数据报之前在源实体和目的实体之间创建一条网络层的逻辑连接，**将无连接的网络层变成了具有逻辑连接的网络层**，并且这种链接是一个单向连接。

8.2.2 安全套接字层 SSL

是 Netscape 提出的，运输层的安全协议，作用在 HTTP 和运输层之间，在 TCP 之上建立一个安全通道，可以提供如下服务：

- SSL 服务器鉴别，允许用户证实服务器的身份
- SSL 客户鉴别，允许服务器证实客服身份
- 加密 SSL 会话，对客户和服务器之间的报文进行加密，并且检测报文是否被篡改

工作过程：协商加密算法、服务器鉴别、会话密钥计算、安全数据传输。HTTPS 是提供安全服务的 HTTP 协议，调用 SSL 对整个网页进行加密。运输层安全协议 TLSP 是基于 SSL 的标准化协议，原本还有安全电子交易协议 SET 但是已经被淘汰了。

8.2.3 应用层安全协议

PGP 是 Zimmerman 于 1995 年开发的电子邮件的标准，用于保护邮件的隐私，

8.3 防火墙和入侵检测

8.3.1 防火墙 Firewall

防火墙是一种访问控制技术，是一种特殊的路由器，可以禁止任何不必要的通信，可以实施一定的访问控制策略，防火墙内的是可信的网络，而防火墙外是不可信的网络，实现防火墙的主要技术有：

- 分组过滤路由器：按照一定的规则进行分组过滤，对进出内部网络的分组执行转发或者丢弃
- 代理服务器：在应用层通信中起到报文中继的作用，一种网络应用需要一个应用网关，所有的进出网络的应用程序都必须通过应用网关

真正的防火墙一半两种技术混合使用。

8.3.2 入侵检测系统 IDS

对网络的分组执行深度分组检查，当观察奥可以分组的时候就向网络管理员发出警报，可以检测多种网络攻击，比如网络映射、端口扫描、DoS 攻击 (拒绝服务攻击，DDoS 是分布式的拒绝服务攻击)、蠕虫和病毒、系统修改、漏洞攻击等等。一般入侵检测可以分为基于特征的入侵检测和基于异常的入侵检测。

9 计算机网络实验:GNS3

计算机网络实验中有几个实验要用 GNS3 模拟器进行。听说这一部分实验期末考试也会考指令的填空题，所以还是需要复习一下。

9.1 三层交换机实验

9.1.1 实验的基本原理和步骤

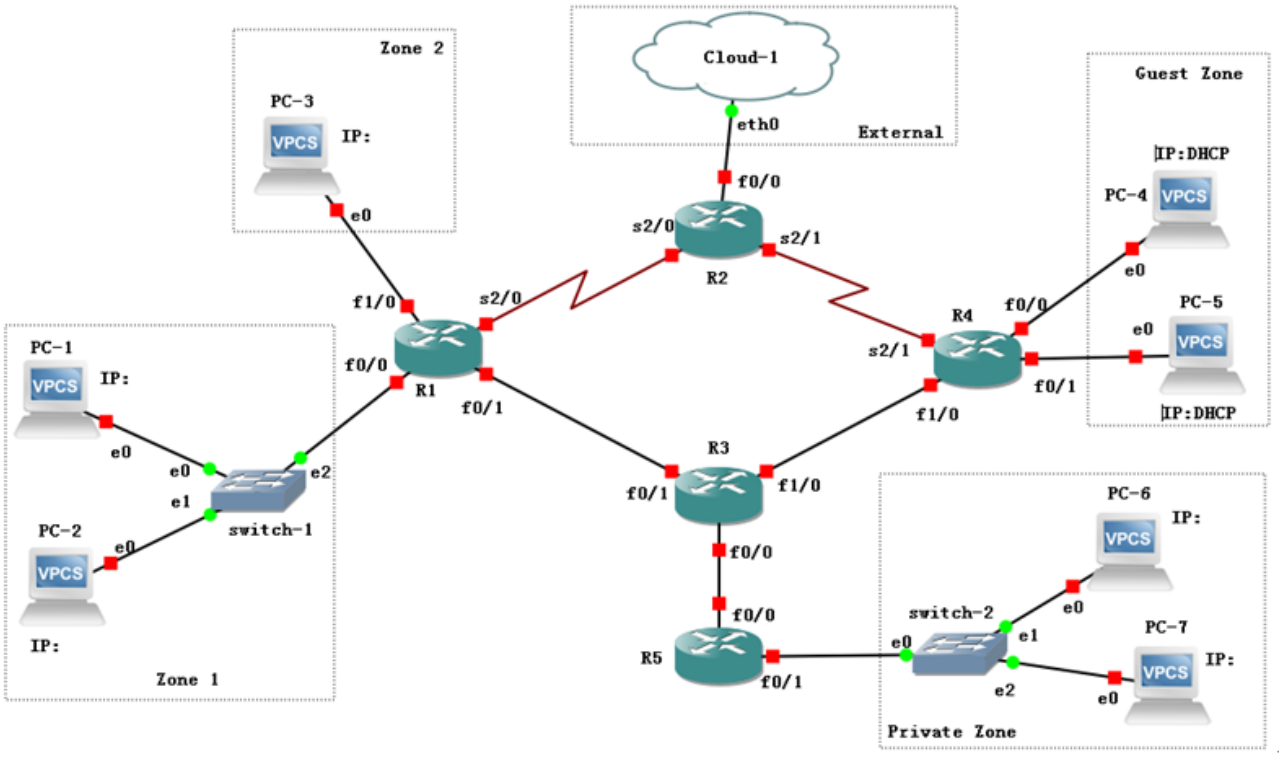
9.1.2 交换机相关指令

9.2 静态路由协议的配置

计网的第四个实验，主要内容是配置静态路由协议，掌握交换和路由的区别 (其实我还是讲不清楚交换和路由的区别)

9.2.1 实验的基本原理和步骤

该实验需要使用多个路由器来连接局域网，分别采用以太网和串口的方式连接路由器，分别采用竞态地址分配，动态地址分配构建多种类型的局域网，并配置 NAT 实现私有网络和公共网络的互联。实验的拓扑图如下：



9.2.2 PC 相关配置指令

对于 PC 可以使用这样几条命令：

- ip IP 地址/子网掩码表示配置 IP 地址
- show 表示查看 PC 的配置情况
- trace IP 地址路由跟踪
- ping 可以用于测试两个端口之间的连通性，ping 目标 IP 地址 source 源 IP 地址可以指定源 IP 地址

9.2.3 路由器相关配置指令

对于路由器可以使用这样一些命令：

- interface 接口名进入路由器的逻辑子接口进行配置
- ip address IP 地址子网掩码给接口配置 IP 地址
- no shutdown 打开路由器的逻辑子接口
- ip route 查看路由表

9.2.4 DHCP 服务的配置

给一个路由器配置 DHCP 服务的基本步骤如下，首先需要配置好路由器当前子接口的 IP，然后：

- ip dhcp pool 编号 (实验里就一个，所以是 1) 定义子网的 DHCP 地址池
- network IP 地址/掩码长度定义 DHCP 的网络地址
- default-router IP 地址定义 DHCO 的默认网关对应的子接口的地址，根据需要可以再定义第二个地址池
- service dhcp 启动 DHCP 服务

之后可以在与这个子接口相连的 PC 上运行命令 ip dhcp 来获取动态分配的 IP 地址，并可以用 ip dhcp binding 命令来查看已分配 DHCP 的主机信息

9.2.5 HDLC 协议的配置

HDLC 协议主要在路由器的串口之间进行配置，实验中需要先对路由器的子接口配置 IP 地址，然后:

- encapsulation hdlc 设置数据链路层的协议是 HDLC
- clock rate 128000 设置时钟速率
- no shutdown 激活接口

9.2.6 PPP 协议的配置

PPP 协议是另一种数据链路层的协议，在路由器的串口中配置的过程如下：

- 还是要先配置好路由器子接口的 IP 地址
- encapsulation ppp
- ppp authentication chap 设置 PPP 认证模式为 CHAP
- no shutdown
- username R4 password 1234 为对方设置用户名和密码

之后可以用 show int s0/1 类似的指令来查看串口的配置情况

9.2.7 静态路由相关指令

涉及到静态路由配置的指令有：

- ip route 查看路由表
- ip route 目标网络子网掩码下一跳的地址添加静态路由，这种添加上去的路由的状态是 S，表示静态的

路由表中有很多常见的状态码，比如 C 表示 connected，S 表示 static，R 表示 RIP，M 表示 mobile，B 表示 BGP，O 表示 OSPF，标识了这一条路由的属性。

9.2.8 NAT 服务的配置

在进入子接口之后，进行如下配置进行 NAT 服务的配置：

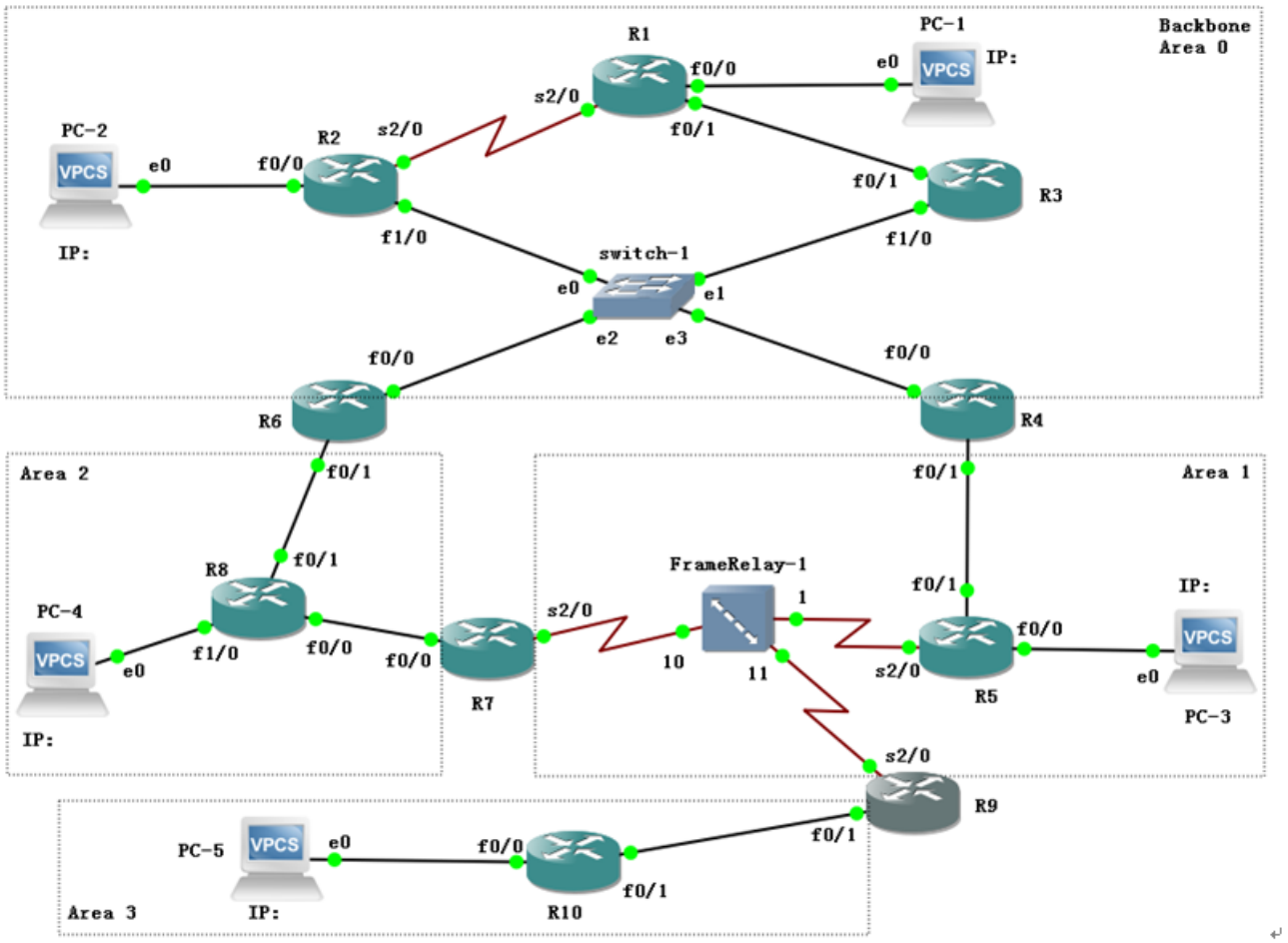
- ip nat inside 定义内部接口
- ip nat outside 定义外部接口
- access-list permit 子网地址子网掩码的反码设置访问控制列表，允许这个子网下的 IP 向外访问
- ip nat inside source list 1 interface f0/0 overload 定义从内到外的访问需要进行的源地址转换，使用路由器的外部接口地址作为转换后的外部地址

9.3 动态路由协议 OSPF 的配置

配置 OSPF 工作协议，并理解 OSPF 的基本概念

9.3.1 实验的基本原理和步骤

该实验需要先用网线连接多个路由器，并配置 IP，将网络拓扑结构分成 0123 四个区域，在 Area0 和 Area1 设置 OSPF 协议，使得路由器可以学习到新的路由信息，在 Area2 启动 OSPF 之后再非广播多路访问网络 (NBMA) 中配置 OSPF 协议，再 Area3 设置 OSPF 协议并建立虚链路，观察各类 PC 和路由器之间能否 ping 通。不过实验会先配置 RIP 协议作为 OSPF 的对照，该实验使用的网络拓扑图如下：



9.3.2 RIP 协议配置相关命令

RIP 协议相关配置命令如下：

- router rip 在路由器上启用 RIP 协议
- version 2 设置 RIP 协议的版本，一般设置成版本 2
- network ip_net 将子网接入协议中

9.3.3 OSPF 相关配置指令

- router ospf id 启用 OSPF 协议

- network 子网地址掩码的反码 area area-id 表示配置路由器接口 (子网) 所属的 area id
- interface loopback 0 给路由器的回环接口配置地址，然后需要使用 ip address 命令
- router-id IP 地址可以手工指定路由器的 IP 地址，需要 reload 之后才会生效
- clear ip ospf process 清除 OSPF 的状态
- debug ip ospf events 打开 OSPF 的事件调试
- no debug ip ospf events 关闭调试信息

查看 OSPF 状态相关的命令

- show ip ospf database 查看 OSPF 的数据库
- show ip ospf neighbor detail 观察 OSPF 的邻居关系
- show ip ospf interface 查看路由器的 OSPF 接口状态

9.3.4 Frame Relay 协议的配置

- encapsulation frame-relay 设置链路层的协议为 frame-relay
- frame-relay lmi-type ANSI 支持 ANSI 模式，LMI
- interface s0/0.1 multiple 创建子接口
- frame-relay interface-dlci 101 配置接口 DLCI
- show frame-relay map 查看 frame relay 的映射
- ip ospf network point-to-mulitpoint 配置 s0/0 的接口为点对多点的网络类型

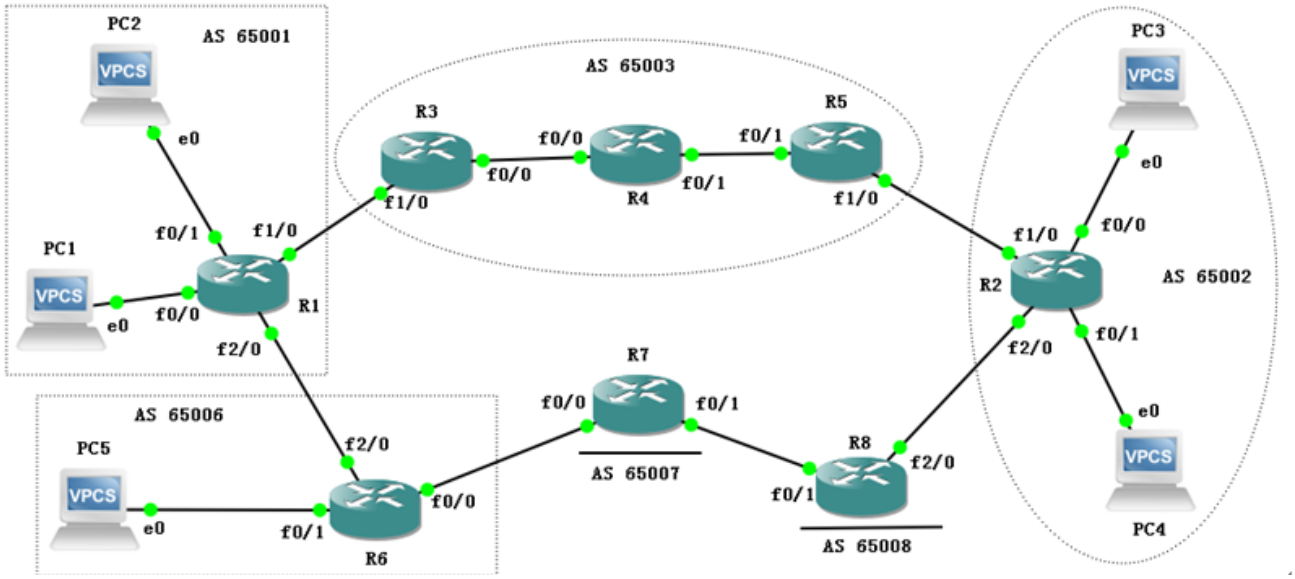
9.3.5 虚链路相关的配置

- area area-id virtual-link router-id 在两个边界路由器之间建立虚链路，area-id 是传递数据的区域 id，router-id 设置成对方的
- area area-id range 子网子网掩码

9.4 动态路由协议 BGP 的配置

9.4.1 实验的基本原理和步骤

创建多种类型的网络，各自成为一个独立的 AS，在内部启用 OSPF，而在路由器的边界启用 BGP 协议，配置路由聚合和多路径的负载均衡，该实验使用的拓扑图如下：



9.4.2 BGP 相关配置指令

基本的 BGP 配置

- router bgp AS-number 启用 BGP 协议，并设置 AS 号
- network x.x.x.x mask x.x.x.x 宣告直连网络
- neighbor IP-Address remote-as AS-Number 将对方加入 AS 内部成为邻居时用相同的 AS，将对方增加为 AS 之间的邻居时用对方的 AS
- show ip bgp router 查看 BGP 的邻居关系
- debug ip bgp 打开 BGP 调试
- no debug ip bgp 关闭 BGP 调试
- show ip bgp 打开 BGP 数据库
- neighbor IP-Addr update-source loopback 0 设置 BGP 更新源为回环接口

重分发功能的配置：

- synchronization 打开同步功能，需要在 router bgp 之后使用
- redistribute bgp AS-number subnets 在 OSPF 中启用 BGP 的重分发，需要先 router ospf
- redistribute ospf id 打开 BGP 的 OSPF 重分发，需要先 router bgp
- clear ip bgp * 清除 BGP 信息

总结：可以在 OSPF 中打开 BGP 的重分发，也可以在 BGP 中打开 OSPF 的重分发，但是配置之前都需要先进入对应的协议中，不能直接配置，命令行中会从 config 变成 config-router

路由过滤的配置：

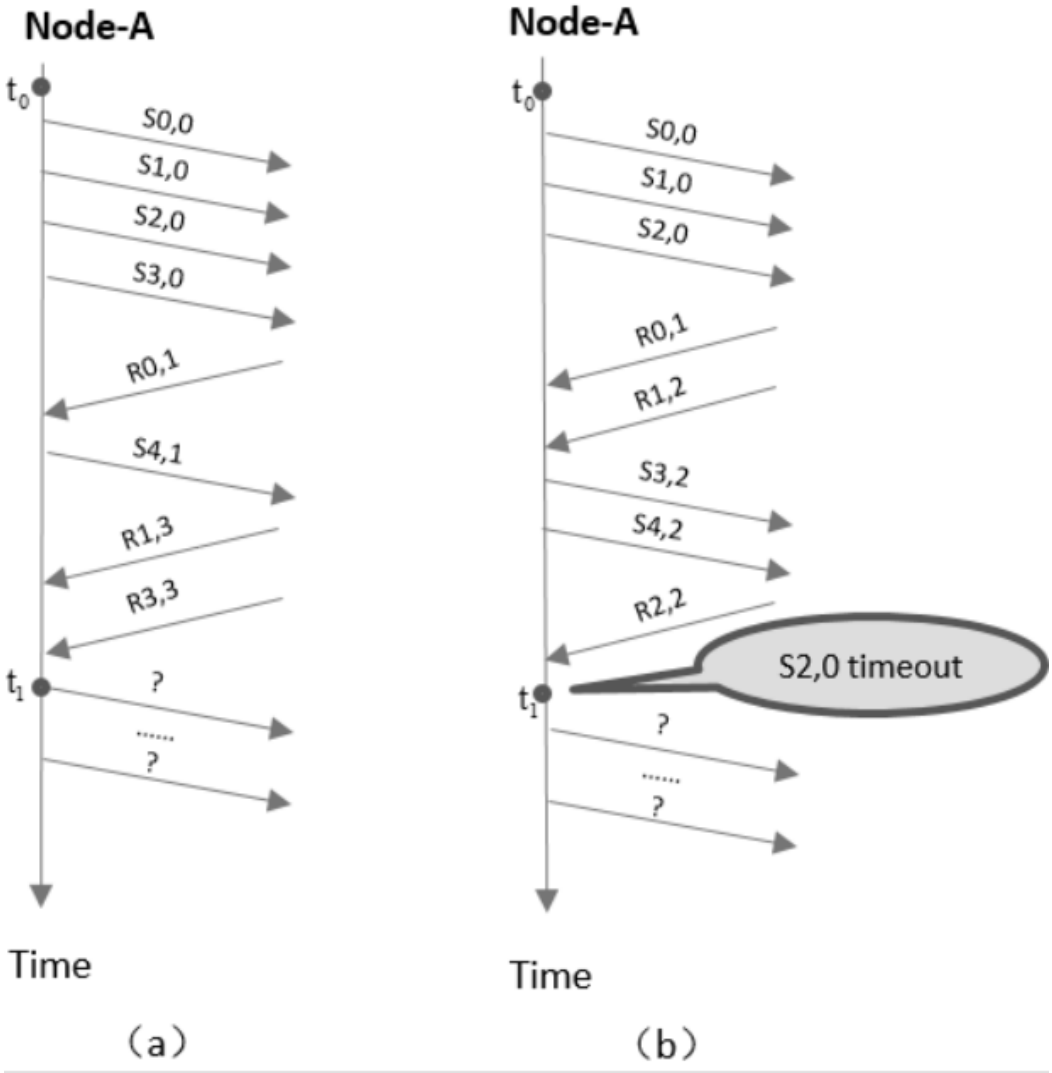
- access-list <id> deny <subnet> <mask> 先创建访问列表
- neighbor <router id> distribute-list <access-list-id> out 然后再配置过滤路由
- aggregate-address <ip-network> 子网掩码 summary-only as-set 设置聚合路由
- maximum-paths 2 允许设置多条路径

9.4.3 IPv6 相关指令

- show ipv6 interface IPv6 的接口查看
- ipv6 unicast-routing 启用单播路由，之后可以用 network 宣告直连网络
- show ip bgp ipv6 unicast neighbors 查看单播路由的邻居信息
- show ipv6 查看 IPv6 的信息，show ipv6 route 可以查看路由表
- interface Tunnel <id> 创建 IPv6 隧道
- ipv6 address <address>/掩码长度设置 IPv6 隧道的地址
- tunnel source <interface number> 设置隧道的源接口
- tunnel destination <ipv4 address> 设置隧道的目标 IPv4 地址
- tunnel mode ipv6ip 设置隧道为手工配置
- ipv6 route <ipv6 network> Tunnel <id> 设置 IPv6 的静态路由
- redistribute static 重分发静态路由

10 面向考试的题型总结

10.1 链路层滑动窗口协议



这类题目似乎经常作为大题出现，这里发送的帧有两个参数 x 和 y，其中 x 表示发送的序号，而 **y 是确认的序号 (表明希望接受的对方的下一帧的序号)**，下面来具体分析一下这幅图。

在图 a 中，我们发现，A 一开始连续发出去了 4 帧之后受到了第一个回应 R(0,1)，这个回应被认为是有效的，因此后面 A 发出的帧的确认号就变成了 1，而 R(0,1) 表明 0 号帧被 B 成功接收，接着发出去了一个 S(4,1)，收到了 R(1,3) 和 R(3,3) 表明 B 下一个希望接受的是 A 发出的 3 号帧，因此可以断定，在 t1 的时刻，A 发出的前三个帧已经被成功接收。而题目中序列号有 3bits，因此发送窗口最大为 7，而此时 A 已经发出去了 S(3.0) 和 S(4,1)，因此最多还能再发送 5 个帧，因为已经收到了 R1，所以下一个希望收到 2，因此在收到 B 的新帧之前，A 发出的第一个帧应该是 S(5,2)，而最后一个发送出去的数据帧是 S(1,2)，因为要发 5 个帧，只有三位序号，567 之后就是 01，所以最后一个帧的序号是 1

而在图 b 中，同样可以判断 t1 时刻，A 已经成功被 B 接收的帧是前两个 (0,0)(1,0)，而因为收到了

R(2,2) 表明 B 还在等 A 发出的 2 号帧，因此发出的 234 号帧都需要重新发送，而因为已经收到了 R(2,2)，因此确认号是 3，大概就是这样。

10.2 IP 与 TCP 包的解析

No.	The First 40 bytes of IP packet (HEX)				
1	45 00 00 32	01 9b 40 00	80 06 1d e8	c0 a8 00 08	d3 44 47 50
	07 65 13 88	84 6b 41 c5	00 00 00 00	70 02 43 80	5d b0 00 00
2	45 00 00 32	00 00 40 00	30 06 6e 83	d3 44 47 50	c0 a8 00 08
	13 88 07 65	00 00 00 0a	84 6b 41 c6	70 12 16 d0	37 e1 00 00
3	45 00 00 28	01 9c 40 00	80 06 1d ef	c0 a8 00 08	d3 44 47 50
	07 65 13 88	84 6b 41 c6	00 00 00 0b	50 10 43 80	2b 32 00 00
4	45 00 00 38	01 9d 40 00	80 06 1d de	c0 a8 00 08	d3 44 47 50
	07 65 13 88	84 6b 41 c6	00 00 00 0b	50 18 43 80	c6 55 00 00
5	45 00 00 28	68 11 40 00	30 06 06 7a	d3 44 47 50	c0 a8 00 08
	13 88 07 65	00 00 00 0b	84 6b 41 e6	50 10 16 d0	57 d2 00 00

这个题目我第一次在小测题目里面看到，感觉做起来比较难，但是可能熟练了之后就会简单一点，IP 包和 TCP 包的头部的格式是固定的，因此 IP 包的解析也是有固定模式的。**一般来说题目中捕获的数据包都是按照时间顺序的，可以从中推断出一个 TCP 连接建立的过程!!!**

首先，这类题目给的格式一般是若干个十六进制数，而一个十六进制数是 4bits，一个字节是 8bits，因此连续的两位才能组成一个完整的字节，我们把两个连续的十六进制数看成一组，那么题目中给的一个 IP 包，前 20 组就是 IP 数据报的头部，然后可以根据 IP 数据报头部的格式来进行解析，后面的 20 组就是 TCP 数据报的头部。

我总结了一下经常会考的几个点：

- 通过 IP 地址来判断收发方的情况：找 IP 头部的最后两个四连组，分别代表源地址和目标地址
- 判断数据包的长度：看 IP 头部的第三和第四组，这 16bits 表示数据报的总长度
- MAC 帧相关：暂时不懂
- 路由相关：
  - ★ 计算数据包的 TTL：看三个四连的第一个，这里记录了数据包的 TTL
  - ★ 计算数据包经过了多少个路由器：找到最早发出的包和接收端收到的报相减就是经过的路由器数目
- 协议类型可以看第十组，一般都是 TCP 或者 UDP，其中 06 表示 TCP 包
- 判断包和包的对应关系：结合多个位置进行确定，比如长度、校验和
- 求序列号：直接看 TCP 部分 (也就是第二排) 的第二大组的内容
- 接受方已经收到的字节数：用确认号相减就可以获得，比如上面这题中，2 和 5 是从 B 发出的相邻数据包，查看确认号发现差值是 32，这就表明这一期间 B 收到了 32 字节的数据，我们可以从有顺序的包中推断出 TCP 连接建立的过程。

- 端口号相关：找哪个端口是被监听的发送端口要找从发送方送过来的包里面的端口号

10.3 TCP 拥塞控制相关内容