

计算机网络

Kajih Du

计算机网络

1 绪论

- 1.1 基本组成
- 1.2 协议和服务
- 1.3 OSI参考模型
- 1.4 TCP/IP参考模型
- 1.5 网络设备
- 1.6 度量单位

2 物理层

- 2.1 功能
- 2.2 特性
- 2.3 数据通信基本概念
- 2.4 传输方式
 - 2.4.1 基带传输和频带传输
 - 2.4.2 串行传输和并行传输
 - 2.4.3 传输模式
- 2.5 性能度量
- 2.6 复用技术
 - 2.6.1 时分复用 (Time Division Multiplexing)
 - 2.6.2 统计时分复用
 - 2.6.3 频分复用
 - 2.6.4 波分复用
 - 2.6.5 码分复用
 - 2.6.6 正交频分复用
 - 2.6.7 空分复用

3 数据链路层

- 3.1 位置
- 3.2 服务
- 3.3 成帧
 - 3.3.1 字节计数法
 - 3.3.2 带字节填充定界符法
 - 3.3.3 带比特填充定界符法

- 3.4 差错控制
 - 3.4.1 基本术语
 - 3.4.2 检错码
 - 3.4.2.1 奇偶校验
 - 3.4.2.2 校验和
 - 3.4.2.3 循环冗余校验 (CRC)
 - 3.4.3 纠错码
 - 3.4.3.1 Hamming码
- 3.5 基本协议
 - 3.5.1 乌托邦式单工协议
 - 3.5.2 无错信道的停等式协议
 - 3.5.3 有错信道的单工停等式协议
 - 3.5.4 滑动窗口协议
 - 3.5.5 回退N协议
 - 3.5.6 选择性重传协议
- 4 介质访问控制子层 (MAC子层)
 - 4.1 随机访问协议
 - 4.1.1 ALOHA协议
 - 4.1.2 分隙ALOHA协议
 - 4.1.3 载波侦听多路访问协议 (CSMA)
 - 4.1.3.1 非持续式CSMA
 - 4.1.3.2 持续式 (1-持续式) CSMA
 - 4.1.3.3 p-持续式CSMA
 - 4.1.3.4 CSMA/CD
 - 4.2 受控访问协议
 - 4.2.1 位图协议
 - 4.2.2 令牌传递
 - 4.2.3 二进制倒数计数协议
 - 4.2.4 随机访问和受控访问对比
 - 4.3 有限竞争协议
 - 4.3.1 自适应树搜索协议
 - 4.4 以太网
 - 4.5 数据链路层交换
 - 4.5.1 设备
 - 4.5.2 原理
 - 4.5.3 生成树协议
 - 4.5.4 虚拟局域网

4.6 无线局域网

1 绪论

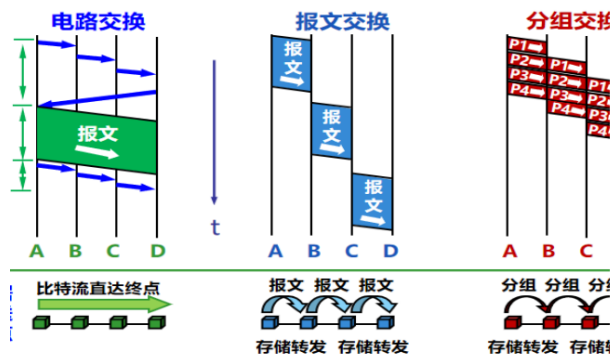
1.1 基本组成

ISP(Internet Service Provider) 网络服务提供商

网络的组成：边缘设备（接收和发送数据）和网络核心（路由和转发）。

传输单元：位（bit），在物理介质上传播的数据最小单元。数据传输使用Bit表示，K/M/G之间为 10^3 进制。

交换方式：



- 电路交换：需要建立连接并预留资源
- 报文交换：不常用
- 分组交换：适合有大量突发数据传输需求的互联网

1.2 协议和服务

协议：同一层对等实体的进行通讯的规定。包含**语法**、**语义**、**时序**三个要素。
（“水平”的）

服务：上层实体通过接口调用下层实体的服务。（“垂直”的）分为**面向连接传输服务**（按电话系统模型）和**无连接传输服务**（按邮政系统模型）。

实体可以通过协议来实现其定义的服务

面向链接服务的六个核心服务原语：连接请求、接受响应、请求数据、应答、请求断开、断开连接。

1.3 OSI参考模型

- 物理层：定义在信道上如何传输0和1
- 数据链路层：实现相邻网络实体间的数据传输；成帧：从物理层比特流中提取出完整的帧；错误检测和纠正；流量控制；物理地址（MAC Address）；共享信道的访问控制
- 网络层：将数据包跨越网络从源设备发送到目的设备（host to host）；进行路由和转发；IP地址
- 传输层：将数据从源端口发送到目的端口（进程到进程）
- 会话层：在应用程序之间建立和维持会话，并能使会话获得同步
- 表示层：关注所传递信息的语法和语义，管理数据的表示方法，传输的数据结构
- 应用层：通过应用层协议，提供应用程序便捷的网络服务调用

1.4 TCP/IP参考模型

- 应用层
- 传输层
- 互联网层
- 网络接口层

采用IP分组交换：可在各种底层物理网络上运行；可支持各类上层应用；每个IP分组携带各自的目的地址，网络核心功能简单（通过路由表转发分组），适应爆炸性增长

1.5 网络设备

交换机：物理层、链路层（两层，无法实现路由）

路由器：物理层、链路层、网络层（三层）

只有物理层的网络设备：集线器、放大器

1.6 度量单位

比特率：数字信道上传送数据的速率，单位b/s或bps

带宽：网络内某通道传输数据能力，单位时间内网络中某信道能通过的“最高数据率”，单位bit/s

包转发率（PPS）：Packet Per Second(包/秒)，表示交换机或路由器等网络设备以包为单位的转发速率

时延：数据（一个报文或分组）从网络（或链路）的一端传送到另一端所需的时间。包含**传输时延**（数据从结点进入到传输媒体介质所需要的时间）、**传播时延**（电磁波在信道中需要传播一定距离而花费的时间）、**处理时延**（主机或路由器在收到分组时，为处理分组所花费的时间，与路由器有关）、**排队时延**（分组在路由器输入输出队列中排队等待处理所经历的时延，和路由算法有关）

往返时延：从发送方发送数据开始，到发送方收到来自接收方的确认，经历的总时间

时延带宽积：传播时延和带宽的乘积，即按比特计数的链路长度

吞吐量：单位时间内通过某个网络(或信道、接口)的数据量，单位是 b/s

有效吞吐量：单位时间内，目的地正确接收到的**有用信息的数目**（以 bit 为单位）

2 物理层

2.1 功能

位置：位于网络体系最底层（不是连接计算机或传输信号的具体的物理设备）

功能：在连接各计算机的传输媒体上**传输数据比特流**

作用：尽可能屏蔽掉不同传输媒体和通信手段的差异

2.2 特性

接口特性：物理层协议是DTE和DCE间的约定，规定了两者之间的接口特性

机械特性：定义接线器的形状和尺寸、引线数目和排列、固定和锁定装置等

电气特性：规定了DTE/DCE之间多条信号线的电气连接及有关电路特性

功能特性：描述接口执行的功能，定义接线器的每一引脚(针，Pin)的作用

过程特性：指明对于不同功能的各种可能事件的出现顺序

2.3 数据通信基本概念

通信：在源点和终点之间传递信息或消息

消息：通信过程中表达客观物质运动和主观思维活动的文字、符号、数据等

信息：消息中对通信者有意义的部分内容

数据：对某一事实的不经解释并赋予一定含义的数字、字母、文字等符号及其组合的原始表达

信号：消息的载体

信息量：一条消息所含信息的多少，与描述事件出现的概率有关，单位为比特

$$I = -\log_a(p)$$

平均信息量：每个符号包含信息量的统计平均，单位为比特/符号

$$H = \sum_i -p_i \log p_i$$

模拟数据：一段时间内具有连续的值

离散数据：一段时间内具有分散的值

2.4 传输方式

2.4.1 基带传输和频带传输

基带传输：不搬移信号频谱的传输机制

频带传输：通过调制调节器搬移信号频谱的传输机制（为了适应信道的频率特性）

2.4.2 串行传输和并行传输

串行传输：数据在一个信道上按位依次传输的方式

并行传输：数据在多个信道上同时传输的方式

2.4.3 传输模式

单工：指两个站之间只能沿一个指定的方向传送数据信号

半双工：指两个站之间可以在两个方向上传送数据信号，但不能同时进行

全双工：指两个站之间可以在两个方向上同时传送数据信号

2.5 性能度量

传输速率：单位时间内传送信息量

调制速率（波特率）：单位时间内调制信号波形变换次数，单位为波特（baud）

$$R_B = 1/T$$

数据信号速率（比特率）：单位时间内通过信道的信息量，单位bit/s

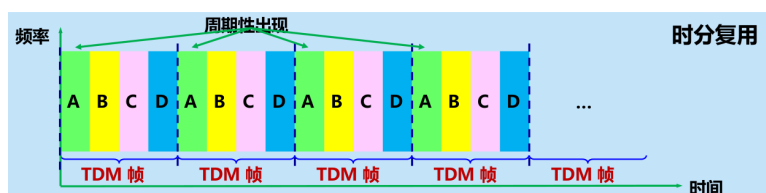
$$R_b = \sum_i \frac{1}{T_i} \log_2 M_i = R_B \log_2 M_i$$

相同条件下，进制越大，比特速率越大，但差错率也大。

2.6 复用技术

复用：允许用户使用一个共享信道进行通信，避免相互干扰，降低成本，提高利用率。

2.6.1 时分复用（Time Division Multiplexing）



- 每一个时分复用的用户在每一个时分复用（TDM）帧占用固定序号的时隙
- 每一个用户占用时隙周期性出现（周期为TDM帧长度）

- TDM信号为等时信号
- 所有用户在不同时间占据同样频带宽度

缺点：对信道利用率低（某用户暂时无数据发送时，在时分复用帧中分配给该用户的时隙只能处于空闲状态），不同用户之前传输时需要进行同步

2.6.2 统计时分复用

统计时分复用：动态地按需分配共用信道的时隙，只将需要传送数据的终端接入共用信道，以提高信道利用率

在传输数据前进行统计，按需**动态分配时隙**

2.6.3 频分复用



频分复用：将多路基带信号调制到不同频率载波上，再进行叠加形成一个复合信号的多路复用技术

频分复用的所有用户在同样的时间**占用不同的频率带宽资源**

2.6.4 波分复用

波分复用：利用多个激光器在单条光纤上同时发送多束不同波长激光的技术

2.6.5 码分复用

码分复用：利用码序列相关性实现的多址通信，基本思想是靠不同的地址码来区分的地址

每个站的码片序列各不相同，并且需要正交

缺点：容易受到噪声影响；传输过程中传输的数据量增加，需要的频率带宽增加

2.6.6 正交频分复用

将信道分为若干正交子通道，将高速数据信号转换成并行的低速子数据流，调制到在每个子信道上进行传输

相比于频分复用，其允许不同使用者的频带可以重叠，提高了频带的利用率

2.6.7 空分复用

空分复用：让同一个频段在不同的空间内得到重复利用

3 数据链路层

3.1 位置

向下：利用物理层提供的位流服务

向上：向网络层提供明确接口

实现功能：成帧、差错控制、流量控制

3.2 服务

根据确认（ACK）有无和连接（connection）有无分类

- 无确认无连接服务

接收方不会确认是否收到，不会记录网络传输状态

优点：传输速度快；缺点：不能确认数据是否有错误

适用：误码率低的可靠信道；实时通信

- 有确认无连接服务

接收方会确认是否收到，不会记录网络传输状态

优点：网络通信可靠

适用：不可靠的信道（无线信道）

- 有确认有连接服务

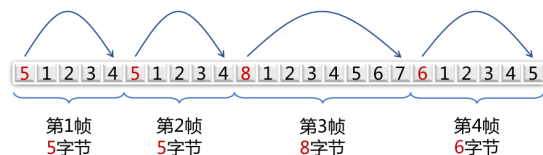
接收方会确认是否收到，并记录网络传输状态

适用：长延迟的不可靠的信道

3.3 成帧

目的：为了差错控制和流量控制

3.3.1 字节计数法



每个帧的头标表示帧包含的字节数。

缺点：只能用于无差错传输的情形，方法抗噪性低。

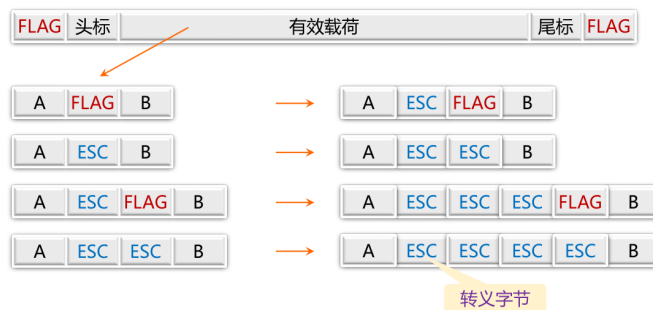
3.3.2 带字节填充定界符法



用定界符区分前后两个帧

缺点：有效载荷部分不能有与定界符相同的字节

解决方法：定义转义字符，接收到转义字符，后一字节无条件成为有效载荷

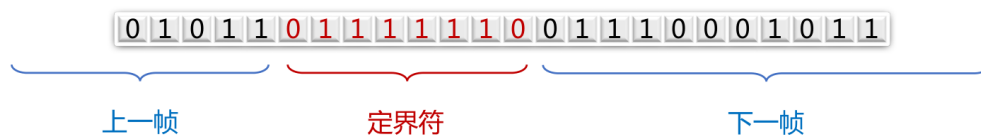


不足：转义字符和flag以字节为单位

改进方式：使用带比特填充定界符法

3.3.3 带比特填充定界符法

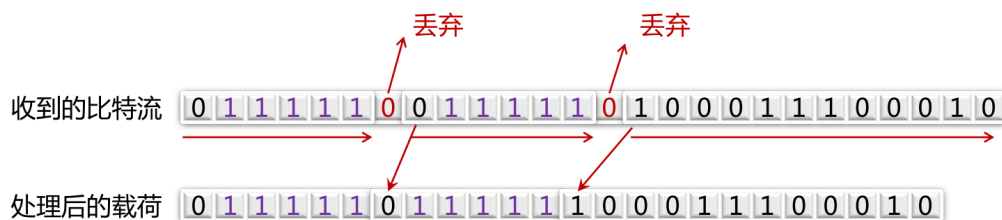
采用一特定的比特组合01111110（0x7E）来标志帧的边界，也就是一个标志字节。



发送方检查有效载荷：若在有效载荷中出现连续5个1比特，则直接插入1个0比特



接收方处理：若出现连续5个1比特，若下一比特为0，则为有效载荷，直接丢弃0比特；若下一比特为1，则连同后一比特的0，构成定界符，一帧结束



带字节填充定界符法和带比特填充定界符法属于透明传输

3.4 差错控制

主要策略：检错码、纠错码

检错码：在被发送的数据块中，包含一些冗余信息，但这些信息**只能使接收方推断是否发生错误**，但不能推断哪位发生错误，接收方可以请求发送方重传数据

检错码用在高可靠、误码率较低的信道上，差错可以用重传的方法解决

纠错码：发送方在每个数据块中加入足够的冗余信息，使得接收方能够判断接收到的数据是否有错，并能纠正错误

纠错码用在错误发生比较频繁的信道上

3.4.1 基本术语

码字：一个包含 m 个数据位和 r 个校验位的 n 位单元， (n, m) 码， $n = m + r$

码率：码字中不含冗余部分所占的比例，可以用 m/n 表示

Hamming距离：两个码字之间不同对应比特的数目

检查出 d 个错，可用Hamming距离 $d + 1$ 的编码

纠正 d 个错，可用Hamming距离 $2d + 1$ 的编码（找Hamming距离最小的有效码字）

3.4.2 检错码

3.4.2.1 奇偶校验

奇偶校验：增加1位校验位，可以检查奇数位错误。

- 偶校验：保证1的个数为偶数个
- 奇校验：保证1的个数为奇数个

3.4.2.2 校验和

数据	1110011001100110	1110011001100110	数据
	1101010101010101	1101010101010101	
	① 1011101110111011	① 1011101110111011	
校验和	1011101110111100	0100010001000011	校验和
	0100010001000011	1111111111111111	

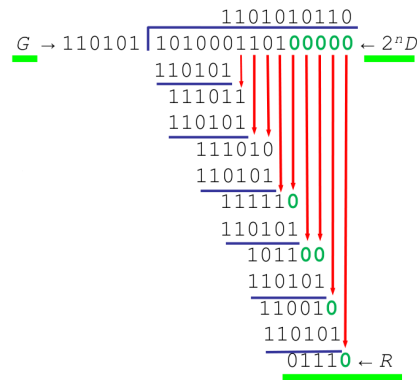
发送方：进行 16 位二进制补码求和运算，计算结果取反，随数据一同发送

接收方：进行 16 位二进制补码求和运算（包含校验和），结果非全1，则检测到错误

3.4.2.3 循环冗余校验（CRC）

- 设原始数据 D 为 k 位二进制位模式
- 如果要产生 n 位CRC校验码，事先选定一个 $n + 1$ 位二进制位模式 G (称为生成多项式，收发双方提前商定)， G 的最高位为1

- 将原始数据 D 乘以 2^n （相当于在 D 后面添加 n 个0），产生 $k + n$ 位二进制位模式，用 G 对该位模式做模2除，得到余数 R （ n 位，不足 n 位前面用0补齐）即为CRC校验码



3.4.3 纠错码

单比特纠错:

m 个信息位, r 个校验位, 对 2^m 个有效信息中任何一个, 有 n 个与其距离为1的无效码字, 于是 $(n + 1)2^m \leq 2^n$, 进而

$$m + r + 1 \leq 2^r$$

3.4.3.1 Hamming码

校验位: 2的幂次方位

...

3.5 基本协议

- 分层进程独立假设: 网络层、数据链路层、物理层为独立进程, 进程间通过传递信息通信
- 提供可靠服务假设: 提供可靠、面向连接的服务
- 只处理通信错误假设: 仅处理通信错误

声明:

- `void to_physical_layer(frame *s);`
- `void from_network_layer(packet *p);`
- `void from_physical_layer(frame *r);`

- `void to_network_layer(packet *p);`
- 事件：帧到达 `frame_arrival`、帧出错 `cksum_err`、超时 `timeout`
- 帧

```

1  typedef struct {
2      frame_kind kind; // 数据帧或确认帧
3      seq_nr seq; // 序列号
4      seq_nr ack; // 确认号
5      packet info;
6  } frame;

```

3.5.1 乌托邦式单工协议

假设：

- 完美信道：传输过程中无任何差错
- 始终就绪：发送方/接收方的网络层始终处于就绪状态
- 瞬间完成：发送方/接收方能生成/处理无限多数据

特点：不需要进行流量控制和纠错

实现：

- 发送方：循环中不断发送，从网络层获得数据封装成帧，交给物理层完成一次发送

```

1  frame s;
2  packet buffer;
3  while (true) {
4      from_network_layer(&buffer); // 从网络层获得数据
5      s.info = buffer; // 封装成帧
6      to_physical_layer(&s); // 传送到物理层
7  }

```

- 接收方：循环中持续接受，等待帧到达 `frame_arrival`，从物理层获得帧，解封装将帧的数据传到网络层，完成一次接收

```

1  frame r;
2  event_type event;
3  while (true) {
4      wait_for_event(&event);
5      from_physical_layer(&r); // 从物理层获得帧
6      to_network_layer(&r.info); // 解封装发送到网络层
7  }

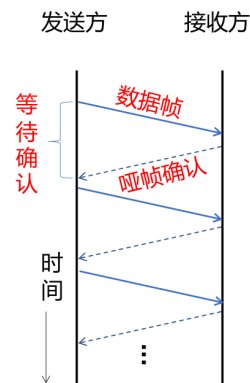
```

3.5.2 无错信道的停等式协议

假设：

- 完美信道：传输过程中无任何差错
- 始终就绪：发送方/接收方的网络层始终处于就绪状态

特点：需要考虑流量控制



实现：

- 发送方：发送一帧后，等待确认到达，确认到达后发送下一帧

```

1  frame s;
2  packet buffer;
3  event_type event;
4  while (true) {
5      from_network_layer(&buffer); // 从网络层获得数据
6      s.info = buffer; // 封装成帧
7      to_physical_layer(&s); // 传送到物理层
8      wait_for_event(&event); // 等待确认到达
9  }

```

- 接收方：完成一帧接收后交给物理层一个哑帧（帧的内容无意义），作为确认帧

```

1  frame r;
2  frame s; // 哑帧
3  event_type event;
4  while (true) {
5      wait_for_event(&event);
6      from_physical_layer(&r); // 从物理层获得帧
7      to_network_layer(&r.info); // 解封装发送到网络层
8      to_physical_layer(&s);
9  }

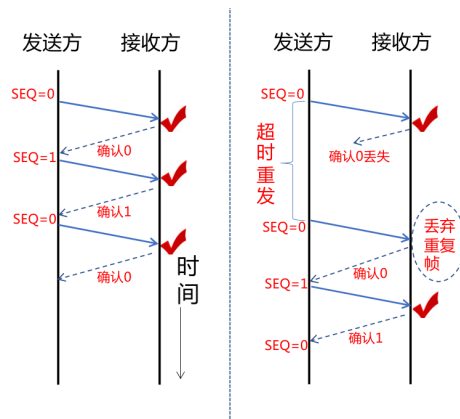
```

3.5.3 有错信道的单工停等式协议

传输过程中出现的问题：差错、丢失、重复帧

解决方法：

- 否定确认帧（NACK）：解决差错
 - 差错和丢失同时解决：设置超时（timeout）
 - 重复帧出现的原因：1. 超时设置太短；2. 确认帧丢失
- 解决重复帧：在帧中添加序列号（SEQ），1位序号（0或1）即可



实现:

- 发送方: 初始化帧序号0, 发送帧; 等待事件; 如果正确确认, 发送下一帧, 否则重发

```

1  frame s;
2  packet buffer;
3  event_type event;
4  next_frame_to_send = 0;
5  while (true) {
6      s.info = buffer; // 封装成帧
7      to_physical_layer(&s); // 传送到物理层
8      start_timer(s.seq);
9      wait_for_event(&event); // 等待事件
10     if (event == frame_arrival) {
11         from_physical_layer(&s); // 从物理层获得帧
12         if (s.ack == next_frame_to_send) { // 确认号确认
13             from_network_layer(&buffer); // 从网络层获得数据
14             inc(next_frame_to_send); // 修改序列号
15         }
16         // 如果确认号不一致, 会直接回到循环体头部重传帧
17     }
18     // 如果不是帧到达(超时、帧出错), 则回到循环体头部重传帧
19 }

```

- 接收方: 初始化期待0号帧; 等待帧到达; 如果为正确帧, 则交给网络层并发送帧确认, 否则发送上一个成功接收帧的确认
-

```

1  frame r;
2  frame s; // 确认帧
3  event_type event;
4  frame_expected = 0;
5  while (true) {
6      wait_for_event(&event);
7      if (event == frame_arrival) { // 帧到达事件
8          from_physical_layer(&r);
9          if (r.seq == frame_expected) { // 判断是否为重复帧
10             to_network_layer(&r.info);
11             inc(frame_expected); // 修改期待的序列号
12         }
13         // 如果是重复帧, 发送前一个帧的序列号, 即1 - frame_expected
14         // 如果不是重复帧, frame_expected被修改, 原来帧的序列号为1 -
15         frame_expected
16         s.ack = 1 - frame_expected;
17         to_physical_layer(&s);
18     }
19     // 帧出错等待下一个事件
20 }

```

效率评估:

- 帧的大小 F , bits
- 信道带宽 R
- 传播时延和处理时延 I , 总延迟 $D = 2I$
- 信道利用率:

用于发送的有效时间为 F/R , 消息发送使用的总时间为 $F/R + D$, 利用率为

$$\frac{\frac{F}{R}}{\frac{F}{R} + D} = \frac{F}{F + RD}$$

RD 为时延带宽积

3.5.4 滑动窗口协议

长肥网络：时延带宽积很大（大于 10^5 bits）的网络，如卫星通讯网络。

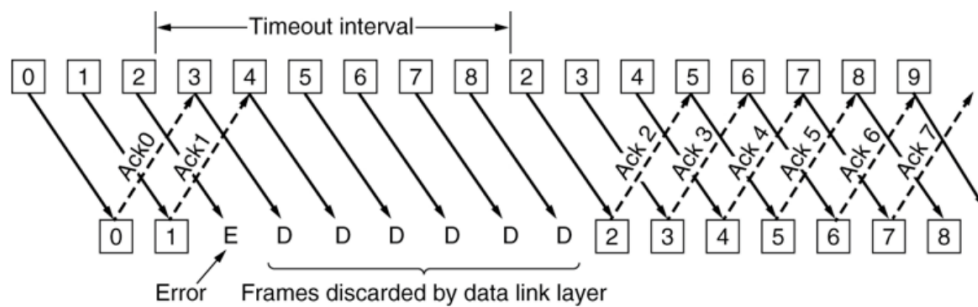
对于长肥网络，信道利用率很低。

解决方法：使用更大的帧。但帧越大，在传输中出错的概率越高，将导致更多的重传。参考分组交换的方法，允许发送方在没收到确认前发送多个帧

实现要点：

- 序号：循环利用有限的帧序号
- 累计确认：不必对收到的分组逐个发送确认，而是对按序到达的最后一个分组发送确认

3.5.5 回退N协议



- 发送方需要缓存多个分组
- 采用出错全部重发的方式：当接收端收到一个出错帧或乱序帧时，丢弃所有的后继帧（无论正确接收与否），并且不为这些帧发送确认；发送端超时后，重传所有未被确认的帧（发送方的计时器在第一个未被确认的帧上，超时后从这个帧开始重传）
- 发送方响应的三件事：
 - (1) 上层的调用：检测有没有可以使用的序号，如果有就发送
 - (2) 收到ACK：对n号帧的确认采用累积确认的方式
 - (3) 超时事件：如果出现超时，就重传所有已发送未确认的分组
- 滑动窗口长度：帧序号n位，接受窗口只需要 $W_R = 1$ ，发送窗口 $W_T \leq 2^n - 1$

适用场景：该策略对应接收窗口为1的情况，即只能按顺序接收帧

优点：连续发送提高了信道利用率

缺点：按序接收，出错后即便有正确帧到达也丢弃重传

丢失确认帧（ACK）：不影响，因为是累计确认，不会导致发送方重传

3.5.6 选择性重传协议

发送方发出连续的若干帧后，收到对其中某一帧的否认帧，或某一帧的定时器超时，则只重传该出错帧或计时器超时的数据帧

适用：该策略对应接收窗口大于1的情况，即暂存接收窗口中序号在出错帧之后的数据帧

窗口最大尺寸： 2^{n-1}

优点：避免重传已正确传送的帧

缺点：在接收端需要占用一定容量的缓存

4 介质访问控制子层（MAC子层）

4.1 随机访问协议

4.1.1 ALOHA协议

原理：想发就发

特点：

- 冲突：两个或以上的帧
- 随时可能冲突
- 冲突的帧完全破坏
- 破坏了的帧要重传

帧时 T ：发送一个标准长的帧所需的时间

满足Poisson分布：

- 一个帧时内用户产生新帧：均值 N 个
- 一个帧时内信道中产生的帧（包括重传）：均值 G 个
- $0 < N < 1$ ，轻载： N 接近0，重载： N 接
- $G \geq N$ ，轻载 $G = N$ ，重载 $G > N$

- $P(k) = G^k e^{-G} / k!$ (Poisson分布)

吞吐量 S : 发送时间 T 内发送成功的平均帧数

运载负载 G : 时间 T 内所有通信站总共发送帧的平均值, $G \geq S$

P_0 为一个帧发送成功的概率, 于是

$$S = G \times P_0$$

冲突危险期为前后 T , 即为 $2T$, 生成帧的均值为 $2G$, 不受冲突即 $2T$ 时间内没有帧传输, 即连续两个帧时 T 内没有其他帧生成的概率 P_0 为连续两个帧时都生成0帧的概率

$$P_0 = P(k=0)^2 = e^{-2G}$$

于是 $S = G e^{-2G}$, $G = 0.5$ 取到最大值0.184, 信道的利用率最高为18.4%.

4.1.2 分隙ALOHA协议

把时间分成时隙 (time slot), 时隙长度对应一帧的时间

帧的发送和冲突只能在时隙的起点

冲突危险期: 在同一个时隙进行传输, 因而冲突危险期为 T

吞吐量: $S = G e^{-G}$, 最大值为0.368, 为ALOHA协议的两倍

4.1.3 载波侦听多路访问协议 (CSMA)

4.1.3.1 非持续式CSMA

- 侦听, 如果介质空闲则发送
- 介质忙, 则等待随机的时间重复上一步

缺点: 信道利用率低

优点: 等待一个随机时间可以减少再次碰撞冲突的可能性

4.1.3.2 持续式 (1-持续式) CSMA

- 侦听, 如果介质空闲则发送
- 介质忙, 持续侦听, 空闲后立即发送
- 如果冲突, 等待随机时间再重复第一步

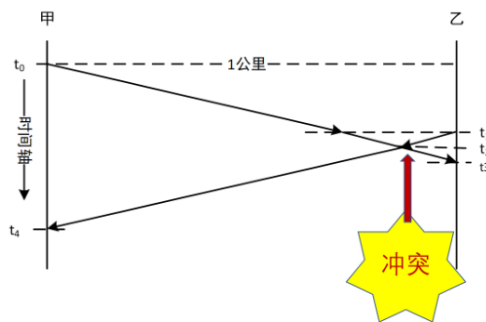
优点：信道利用率高

缺点：如果两个以上的站等待发送，一旦介质空闲就一定会发生冲突

4.1.3.3 p-持续式CSMA

- 侦听，如果介质空闲则以 p 的概率发送，以 $(1 - p)$ 的概率延迟一个单元发送
- 介质忙，持续侦听，空闲后重复上一步
- 如果延迟一个时间单元，重复第一步

CSMA冲突情形：（1）同时传送；（2）传播延迟时间



侦听到冲突：发送帧的时间不能太短，至少要 $2D$

4.1.3.4 CSMA/CD

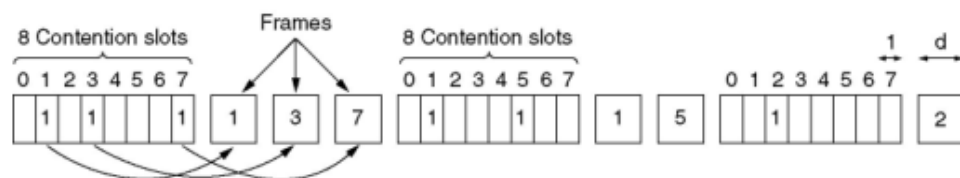
CSMA with Collision Detection

先听后发，边发边听：

- 侦听，如果介质空闲则发送
- 介质忙，持续侦听，空闲后立即发送
- 如果冲突，等待随机时间再重复第一步；并发送Jam（强化）信号（告知冲突产生）

4.2 受控访问协议

4.2.1 位图协议



- 竞争期：在自己的时槽内发送竞争比特（举手示意、资源预留）
- 传输器：按序发送

N 个用户， N 个时隙，每帧 d 比特：

- 低负荷信道利用率 $d/(d + N)$
- 高负荷信道利用率 $d/(d + 1)$

缺点：无法考虑优先级

4.2.2 令牌传递

常用于环形拓扑

只有获得令牌才能发送，一次只有一个人能获得令牌。

缺点：负载低的时候，节点需要得到令牌才能通讯，延迟较大的缺点

4.2.3 二进制倒数计数协议

站点：编序号，各序号长度相同

竞争期：有数据发送的站点从高序号到低序号排队，高的序号得到发送权（高序号站点优先）

优点：存在优先级

缺点：低序号站点可能一直无发送权

4.2.4 随机访问和受控访问对比

低负载时随机访问效率高，高负载时受控访问效率高

4.3 有限竞争协议

4.3.1 自适应树搜索协议

- 在一次成功传输后的第一个竞争时隙，所有站点同时竞争
- 如果只有一个站点申请，则获得信道
- 否则在下一竞争时隙，有一半站点参与竞争（递归），下一时隙由另一半站点参与竞争
- 即所有站点构成一棵完全二叉树

结合了随机访问和受控访问的优点

4.4 以太网

二进制指数后退的CSMA/CD：

- 基本退避时间槽长度设置为以太介质往返传播时间 2τ ，设为512比特时间
- 重传次数为 k ，且 $k \leq 10$
- 从整数集合随机 $[0, 1, \dots, 2^k - 1]$ 取一个数 r
- 重传的时延是 r 倍的时间槽 2τ
- 重传16次仍不能成功这丢弃帧，并向高层报告

4.5 数据链路层交换

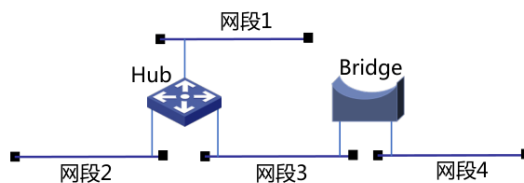
4.5.1 设备

物理层设备（如集线器Hub）无法隔绝冲突域，只有数据链路层以上的设备可以隔绝冲突域

网桥或交换机：分隔冲突域

网桥的功能：转发和过滤

图中冲突域的个数为？



2个冲突域，网桥接口的个数即为冲突域的个数，**集线器无法隔绝冲突域**

理想网桥是透明的：即插即用，无需配置；网络站点无需感知网桥的存在与否

4.5.2 原理

建立MAC地址表：逆向学习发送帧的站的MAC地址，帧到达检查帧的源地址是否在MAC地址表上，如果不在这增加表项，如果在地址表上则更新老化时间

- 增加表项：帧的源地址对应的项不在表中
- 删除表项：老化时间到期
- 更新表项：帧的源地址在表中，更新时间戳

转发过程：逆向学习源地址，根据目的地址，查询MAC地址表

- 如果来源端口和目的端口一样，则丢弃数据帧（在同一冲突域，无需转发）
- 如果来源端口和目的端口不同，根据目的地址查询MAC地址表，找到相应端口转发
- **泛洪**：如果目的地址在MAC地址表上不存在，则将数据帧从所有端口（除了入境口）发送出去

两种目的地址的帧需要泛洪：

- 广播帧：MAC地址为FF-FF-FF-FF-FF-FF的帧
- 未知单播帧

交换模式：

- 存储转发：转发前必须接收整个帧、执行CRC校验
缺点：延迟大
优点：不转发出错帧、支持非对称交换
- 直通交换：一旦接收到帧的目的地址，就开始转发

缺点：可能转发错误帧、不支持非对称交换

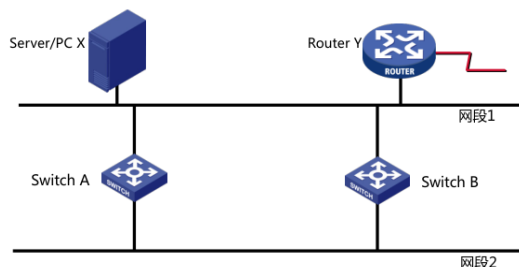
优点：延迟非常小，可以边入边出

- 无碎片交换：接收到帧的前64字节，即开始转发

缺点：仍可能转发错误帧，不支持非对称交换

优点：过滤了冲突碎片，延迟和转发错帧介于存储转发和直通交换之间

4.5.3 生成树协议



冗余拓扑：为了保证可靠传输

代价：出现物理环路

引发的问题：

- 广播风暴：物理环路上无休止泛洪广播流量，无限循环，迅速消耗网络资源
- 重复帧：发送到同一网段设备的帧，发到环路的单播帧（假设交换机MAC地址表没有目的路由器的MAC地址）会使得目的设备收到重复帧
- MAC地址表不稳定：
- 当一个帧的多个副本到达不同端口时，交换机会不断修改同一MAC地址对应的端口

生成树选举过程：

- 选举根桥
 - 首先比较优先级，优先级数值最小的交换机胜出成为根桥
 - 如果优先级数值相等，MAC地址最小的交换机成为根桥
- 选举根端口
 - 每个非根桥，通过比较其每个端口到根桥的根路径开销，选出根端口

- 具有最小根路径开销的端口被选作根端口
 - 如果多个端口的根路径开销相同，则端口ID最小的端口被选作根端口
 - 非根桥只能有一个根端口，根端口处于转发状态
- 为每个网段确定一个指定端口

4.5.4 虚拟局域网

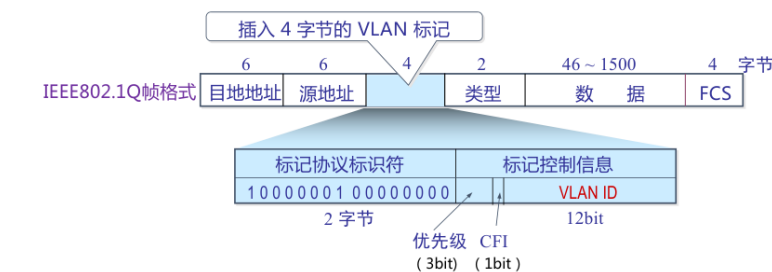
广播域：广播帧能到达的范围

交换机所有端口同属于一个广播域，无法隔离广播域。但支持VLAN的交换机可以隔离广播域，不同VLAN成员之间通讯需要三层设备

VLAN表：记录VLAN ID和端口号/MAC地址/协议/子网的对应关系

区分不同VLAN的数据帧：在数据帧中携带VLAN标记，该标记由支持VLAN的交换机增加和删除，对终端站点透明

帧标记标准：IEEE802.1Q



4.6 无线局域网