

Chapter 5

数据链路控制及其协议

zibuyu

January 7, 2006

1 定义及功能

1.1 定义

要解决的问题 :如何在有差错的线路上,提供无差错的传输服务.

OSI定义 :数据链路层的目的是为了提供功能上和规程上的方法,以便建立、维护和释放网络实体间的数据链路.

1.2 功能

1.2.1 基本定义

结点(node) :主机(host),路由器(router).

链路(link) :通信路径上连接相邻结点的通信信道。
...数据链路协议定义了结点间交换数据单元的格式,以及结点发送、接收数据单元的格式。

端到端(end to end) :源结点(source node)到目的结点(destination node)的通信。可由多个链路组成。

点到点(point to point) :相邻结点间一条链路上的通信。

1.2.2 数据链路控制规程

为使数据能够快速、正确、有效从发送点到达接收点所采取的控制方式。

1.2.3 基本功能

- 1.数据在数据链路路上的正常传输(建立、维护和释放)。
- 2.定界与同步,解决透明性问题。
- 3.差错控制。

- 4.顺序控制。
- 5.流量控制。

1.2.4 提供服务

无确认无连接服务 :适用误码率小的网络；实时网络；大部分局域网。

有确认无连接服务 :适用不可靠信道，如无线网络。

有确认有连接服务

2 成帧方法

成帧 : 将比特流分成离散的帧，并计算每个帧的校验和。

成帧方法 :字符计数法，带字符填充的首尾字符定界法，带位填充的首尾标记定界法，物理层编码违例法。...在实际多数数据链路协议中,一般采用字符技术法和其他方法结合。

2.1 字符计数法

原理 :在帧头中用一个域来表示整个帧的字符数。

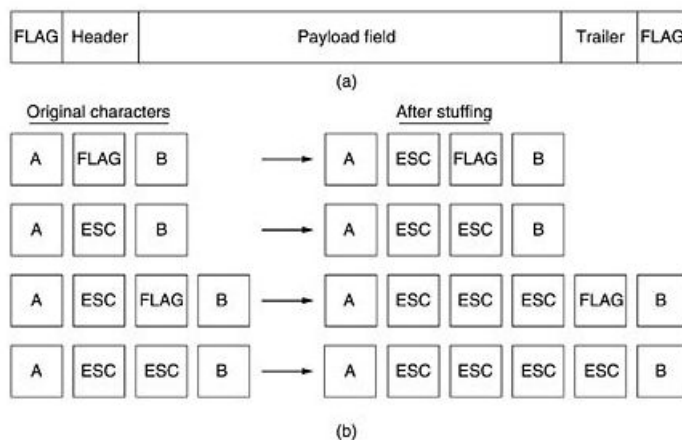
缺点 :若计数出错，则对该帧和后面的帧都有影响。



2.2 带字符填充的首尾字符定界法

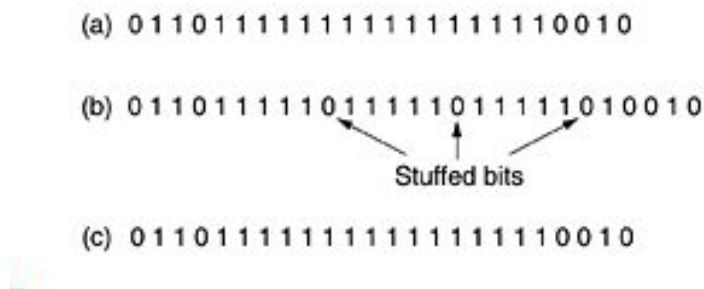
原理 :如下图.

缺点 :限于8位字符和ASCII码传送。



2.3 带位填充的首尾标记定界法

原理 :帧的起始、结束都用一特殊的位串“01111110”标记(flag). 采用“0”比特删除技术, 即在数据中遇到5个连续的“1”就自动在其后添加一位“0”, 防止与标记(flag)混淆。如下图。



2.4 物理层编码违例法

原理 :只适用于物理层编码有冗余的网络. ... 例如:曼彻斯特编码编码用high-low pair/low-high pair表示1/0, high-high/low-low不表示数据, 可以用来做定界符。

3 差错控制

差错类型 :1.帧错误; 2.帧丢失.

突发错误 : 随机, 连续, 突发.

处理差错的两种方法 :1.纠错码; 2.检错码.

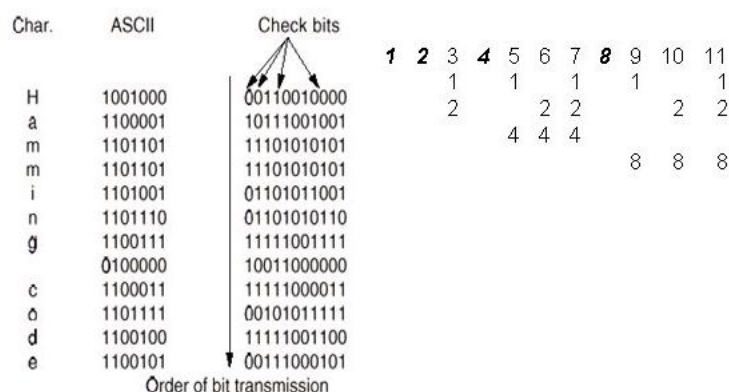
3.1 纠错码-海明码

码字(codeword) :一个帧包括 m 个数据位, r 个校验位, $n = m + r$, 则此 n 比特单元称为 n 位码字。

海明距离(Hamming Distance) :两个码字之间的不同比特位数。

海明码原理 :

- 1.码位从左边开始编号, 以“1”开始.
- 2.位号为2的幂的位是校验位,其余是信息位.
- 3.每个校验位使得包括自己在内的一些位的奇偶值为偶数 (或奇数);
- 4.为看清数据位 k 对哪些校验位有影响, 将 k 写成2的幂的和。例: $11 = 1 + 2 + 8$.



3.2 检错码-CRC码

使用纠错码传数据, 效率低, 适用于不可能重传的场合; 大多数情况采用检错码加重传。

奇偶校验码 :检错的概率只有0.5.

3.2.1 CRC码

循环冗余码(Cyclic Redundancy Check),又称多项式编码(Polynomial Code).

基本思想 :校验和 (checksum) 加在帧尾, 使带校验和的帧的多项式能被 $G(x)$ 除尽; 收方接收时, 用 $G(x)$ 去除它, 若有余数, 则传输出错。

生成多项式 $G(x)$:

- 1.发方、收方事前商定;
- 2.生成多项式的高位和低位必须为1;
- 3.生成多项式必须比传输信息对应的多项式短。

校验和计算算法 1.设 $G(x)$ 为 r 阶, 在帧的末尾加 r 个0, 使帧为 $m + r$ 位, 相应多项式为 $x_r M(x)$;

- 2.按模2除法用对应于 $G(x)$ 的位串去除对应于 $x_r M(x)$ 的位串;
- 3.按模2减法从对应于 $x_r M(x)$ 的位串中减去余数(等于或小于 r 位), 结果就是要传送的带校验和的多项式 $T(x)$ 。

注意 :所谓按模2运算, 即其中加减法, 都是异或运算, 无进位、借位。

检错能力 :课本166页.

4 数据链路层协议

4.1 无约束单工协议

An Unrestricted Simplex Protocol

工作在理想状况下 :

- 1.单工传输;
- 2.发送方无休止的工作(发送信息无限多);
- 3.接收方无休止的工作(缓冲区无限大);
- 4.通信线路不损坏和丢失帧.

工作过程 :

发送方:取数据,成帧,发送.

接收方:等待,接收帧,送数据给高层.

4.2 单工停等协议

A Simplex Stop-and-Wait Protocol

增加限制 :接收方不能无限制接收.

解决方案 :接收方收到一帧后,给发送方送回一个响应.

工作过程 :

发送程序: 取数据, 成帧, 发送帧, 等待响应帧;

接收程序: 等待, 接收帧, 送数据给高层, 回送响应帧.

4.3 有噪声信道的单工协议

A Simplex Protocol for A Noisy Channel

增加限制 :信道有差错,信息帧可能损坏或丢失.

解决方案 :出错重传.

问题解决 :

- 1.重传时机:定时(超时)重传.
- 2.响应帧损坏:发送的数据帧头加入序列号.防止接收方将同一帧重复提交给网络层.
- 3.精简帧头:序列号1位.

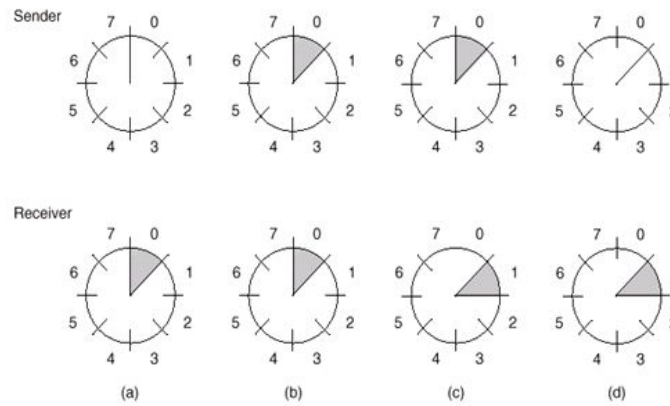
PAR/ARQ :发送方在准备下一数据项目之前,等待一个肯定确认的协议称为:PAR(Positive Acknowledgement with Retransmission)支持重传的肯定确认协议,或者ARQ(Automatic Repeat reQuest)自动重复请求协议.

4.4 滑动窗口协议(Sliding Window Protocol)

捎带技术 :piggybacking,实现双工,充分利用带宽.

滑动窗口协议工作原理 :

- 1.发送的信息帧都有一个序号,从0到某个最大值,0到 $2n - 1$,一般用 n 个二进制位表示;
- 2.发送端始终保持一个已发送但尚未确认的帧的序号表,称为发送窗口。发送窗口的上界表示要发送的下一个帧的序号,下界表示未得到确认的帧的最小编号。
发送窗口大小= 上界- 下界,大小可变;
- 3.发送端每发送一个帧,序号取上界值,上界加1;每接收到一个正确响应帧,下界加1;
- 4.接收端有一个接收窗口,大小固定,但不一定与发送窗口相同。接收窗口的上界表示允许接收的序号最大的帧,下界表示希望接收的帧;
- 5.接收窗口容纳允许接收的信息帧,落在窗口外的帧均被丢弃。序号等于下界的帧被正确接收,并产生一个响应帧,上界、下界都加1。接收窗口大小不变。



4.5 一比特滑动窗口协议

A One Bit Sliding Window Protocol

协议特点：

1. 窗口大小： $N = 1$ ，发送序号和接收序号的取值范围：0, 1；
 2. 可进行数据双向传输，信息帧中可含有确认信息（piggybacking技术）
 3. 信息帧中包括两个序号域：发送序号和接收序号（已经正确收到的帧的序号）
- 注意: 此处的ack是已经正确接收帧的序号, 而不是希望接收的帧序号.

存在问题：

1. 能保证无差错传输，但是基于停等方式；
2. 若双方同时开始发送，则会有一半重复帧；
3. 效率低，传输时间长。

4.6 退后n帧协议

A Protocol Using Go Back n

流水线(pipelining)技术：连续发送多帧后再等待确认。

4.6.1 处理坏帧、重复帧的方法

1. 退后n帧(Go Back n):
 - 接收方从出错帧起丢弃所有后继帧；
 - 接收窗口为1；
 - 对于出错率较高的信道，浪费带宽。

2. 选择重传(Selective Repeat):

- 接收窗口大于1, 先暂存出错帧的后继帧;
- 只重传坏帧;
- 对最高序号的帧进行确认;
- 接收窗口较大时, 需较大缓冲区。

4.6.2 退后n帧协议

特点 :

- 1.发送方有流量控制,为重传设缓冲:(1)发送窗口未滿, EnableNetworkLayer;
(2)送窗口滿, DisableNetworkLayer
- 2.发送窗口大小 \leq 序号个数 $(MaxSeq + 1)$;
- 3.由于有多个未确认帧, 设多个计时器.

4.6.3 选择性重传协议

特点 :

- 1.发送窗口大小: $MaxSeq$, 接收窗口大小 $(MaxSeq + 1)/2$;
- 2.窗口参数:
 - ... 发送窗口下界: AckExpected
 - ... 发送窗口上界: NextFrameToSend
 - ... 接收窗口下界: FrameExpected
 - ... 接收窗口上界: TooFar
- 3.增加确认计时器, 解决两个方向负载不平衡带来的阻塞问题;
- 4.可随时发送否定性确认帧NAK(Negative Acknowledgement).

5 协议工程

5.1 协议说明 (Protocol Specification)

必须既定义一个协议实体提供给它的用户的服务, 又定义该协议实体的内部操作。

5.2 协议验证 (Protocol Verification)

验证协议说明是否完整、正确。

5.3 协议实现 (Protocol Implementation)

用硬件和/或软件实现协议说明中规定的功能。

5.4 协议测试 (Protocol Testing)

用测试的方法来检查协议实现是否满足要求, 包括: 协议实现是否与协议说明一致 (一致性测试)、协议实现之间的互操作能力 (互操作性测试) 和协议实现的性能 (性能测试) 等。

6 形式化描述技术FDT

Finite Description Technology
形式化方法FM(Formal Method).

常用形式化方法 :

- 1.FSM(Finite State Machine)
- 2.形式化语言模型
- 3.Petri网
- 4.过程代数(Process Algebra)

7 常用数据链路层协议

1. IBM的SNA使用的数据链路协议SDLC (Synchronous Data Link Control protocol) ;
2. ANSI修改SDLC, 提出ADCCP (Advanced Data Communication Control Procedure) ;
3. ISO修改SDLC, 提出高级数据链路控制规程HDLC (High-level Data Link Control) ;
4. CCITT修改HDLC, 提出LAP (Link Access Procedure) 作为X.25网络接口标准的一部分, 后来改为LAPB。

7.1 HDLC (High-level Data Link Control)

三种类型站 :

- 1.主站 (primary station) ;
- 2.次站 (secondary station) ;
- 3.组合站 (combined station) .

链路构型 :

- 1.非平衡型;
- 2.平衡型.

基本操作模式 :

- 1.正规响应模式NRM (Normal Response Mode) ;
- 2.异步响应模式ARM (Asynchronous Response Mode) ;
- 3.异步平衡模式ABM(Asynchronous Balanced Mode).

帧结构 :
定界符01111110
既检错也纠错;
underline面向比特.

帧类型 :
信息帧 (Information) 监控帧 (Supervisory) 无序号帧 (Unnumbered) .

7.2 LAPB(Link Access Procedure B)

1. X.25的LAPB帧格式与HDLC完全相同.
2. 采用CRC校验, 只检错, 不纠错, 丢弃出错帧.

7.3 PPP(Point-to-Point Protocol)

Internet数据链路层协议,点到点协议.

1. 以帧为单位发送, 而不是原始IP包.
2. 包括:
链路控制协议LCP (Link Control Protocol) ;
网络控制协议NCP (Network Control Protocol) .
3. 帧格式与HDLC相似, 区别在于PPP是面向字符的, 采用字符填充技术.
4. 提供差错校验、支持多种协议、允许动态分配IP地址、支持认证;
5. PPP帧没有序号域, 不使用滑动窗口技术.