

今日干饭背诵 (1.1)

2021年1月1日 9:53

第五章·数据链路控制及其协议（传输的是：帧）

- 【背诵】数据链路层概述
 - 目的：在有差错的线路上进行无差错传输
 - ISO定义：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（上层）
 - 基本功能（可选的：因为可以在上层得到实现）
 - 数据在数据链路层上的正常传输（建立、维护和释放）
 - 成帧：定界与同步
 - 差错控制：检错和纠错
 - （可选）顺序控制
 - （可选）流量控制（基于反馈机制）
 - 为网络层提供的服务：
 - 无确认无连接服务
 - 适用于误码率低（错误恢复高层做）/实时业务/大部分局域网
 - 有确认无连接服务
 - 适用于不可靠信道（如无线网）
 - 有确认有连接服务
- 【计算】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）
 - a. 字符计数法
 - 在帧头用一个域来表示整个帧的字符个数（个数包括了帧自己）【！】
 - 缺点：若计数出错，对本帧和后面的帧都会有影响，因此常与其他方法组合
 - b. 带字符填充的字符定界法（下面给的是ASCII码的字符定界）
 - 起始加DLE STX，结束加DLE ETX
 - DLE: Data Link Escape
 - STX: Start of Text
 - ETX: End of Text
 - 若传送的数据内容中出现了DLE，在前面填充DLE以示区分
 - 缺点：局限于8位字符和ASCII字符传送
 - 注意：第三次作业的第二题和这个不一样，作业里是按照书上的“带字节填充”（用于bit stream），做法是开头结尾加FLAG，中间碰到FLAG/ESC在前面填充ESC转义
 - c. 带位填充的标记定界法
 - 帧的起始和结束都用一个特殊的位串01111110，称为标记（flag）
 - 对传送的数据内容采用“0”比特插入删除技术：如果连续出现5个1，不管后文是什么都直接填充一个“0”比特 保证特殊位串只出现在帧的起始和结束
 - d. 物理层编码违例法
 - 只适用于物理层编码有冗余的网络
 - 例：802 LAN中曼彻斯特编码用high-low/low-high表示1/0，而high-high/low-

low不表示数据，可以用来做定界符

- 【计算】差错控制——必须在链路层实现，这是链路层的基本功能 注意差错控制指的是检错纠错
 - 纠错码：可以判断数据是否有错，并能纠正错误（适用于不可能重传的情况）
 - 码字：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：两个码字之间不同的比特位数
 - 为了检查出d个比特的错，需要使用海明距离为d+1的编码；
为了纠正d个错，需要使用海明距离为2d+1的编码
 - 设计纠错码：纠正单比特错需要 $m+r+1 \leq 2^r$ m+r种单比特错 & 1种正确情况，校验位 2^r 种组合
 - 海明码计算：回忆一下计组
 - 海明码工作：检查每个校验位k (k=1,2,4,...) 的奇偶值，若第k位不对则计数器+k，全部检查完后若计数器值为0则有效；若为m则第m位出错
 - 海明码纠正（连续）突发错误
 - 采用k个码字组成k×n矩阵（相当于一行是一个码字），按列发送，接收方恢复成k×n矩阵
 - k×r个校验位，k×m个数据位，可纠正最多为k个的突发性连续比特错
 - 检错码：可以判断数据是否有错，但不能判断哪里有错（适用在重传容易的场景）
 - 奇偶校验：添加1个奇偶位，可以检查奇数个错误
 - 循环冗余码（CRC码，多项式编码）
 - 多项式 \longleftrightarrow 01串，多项式中 x^i 的系数是0还是1对应01串从右往左数第i位是0还是1 (i从0开始)
 - 收发双方商定生成多项式G(x)
 - ◆ 最高位和最低位必须为1
 - ◆ 必须比传输信息对应的多项式短
 - 校验和计算
 - ◆ Step1: 记帧对应的多项式为M(x)，设G(x)为r阶（注意对应的位串是r+1位），则在帧的末尾加r个0，对应的多项式变为 $(x^r) \times M(x)$
 - ◆ Step2: 按模2除法计算 $(x^r) \times M(x) \div G(x)$ 对应的位串（大除法）
 - ◆ Step3: 按模2减法从对应于 $(x^r) \times M(x)$ 的位串中减去余数，结果就是要传送的带校验和的多项式T(x)
 - ◆ 注意：模2除法/减法的意思就是减法都按异或算，可以练练课件上的例子
 - 检错能力分析（见【其他】）
 - 【概念】基本的数据链路层协议（书上的Protocol 1~3）
 - 无约束单工协议（Protocol 1）进行基于反馈的流控，避免发得太快收得太慢而冲垮
 - 单工停等协议（Protocol 2，加了响应）对方
 - 新增约束条件：接收方不能无休止接收，但信道仍然完美
 - 解决方法：接收方每收到一个帧后，给发送方回送一个响应（发送方发完数据后也要等待响应然后再继续循环）
 - 书上提到“虽然这个例子中的数据流量是单工的，但帧可以在两个方向上传送，...，这里采用一个半双工的物理信道就足够了”，我觉得因为ppt上没有所以不会拿出来坑人，但大家可以意会一下

- 有噪声信道的单工协议 (Protocol 3, 加了重传+序号)
 - 新增约束条件: 信道有差错, 信息帧可能损坏或丢失
 - 解决方法: 出错重传
 - 发送方定时重传
 - 发送帧头中放入1位序号 (因为是停等, 一位就够了, 只是防止响应帧损坏)
 - 确认帧也要有序号 (不然超时时间太短的情况下协议会失败)
- 【大题】滑动窗口协议 (书上的Protocol 4~6)
 - 确认序号到底是收到的帧的序号还是希望收到的下一个帧的序号要看题目说明
 - 三个滑动窗口协议都能在非理想环境下正常工作, 区别仅在于效率、复杂性和对缓冲区的要求
 - 改变条件: 全双工传输
 - 新增技术: 捎带/应答 (暂时延迟待发的ACK, 以便附加在下一个待发数据帧)
 - 充分利用信道带宽, 减少“帧到达”中断, 但是较为复杂
 - 信息帧里包含两个序号域: 发送序号和确认序号
 - 滑动窗口协议工作原理
 - 发送帧的序号变成 $0 \sim (2^n)-1$
 - 发送窗口: 大小可变, 累计确认
 - 接收窗口: 大小固定, 未必与发送窗口相同
 - 一比特滑动窗口协议 (Protocol 4, 其实还是停等, 所以对带宽利用没那么高)
 - 范围限制: 发送、接收窗口大小均=1
 - 问题1: 无差错传输的实现基于停等
 - 问题2: 若双方同时开始传输/过早超时, 则会有一半重复帧, 效率低
 - 退后n帧重传 (Protocol 5, 不再停等, 多个计时器)
 - 范围限制: $1 < \text{发送窗口} < \text{MaxSeq}+1$, 接收窗口=1 严格小于序号个数
 - 发送方有流量控制, 为重传设缓冲
 - 序号为 $0 \sim \text{MaxSeq}$, 则发送窗口尺寸 $< \text{MaxSeq}+1$ (序号个数), 不能取等
 - 计时器: 针对序号 (所以有多个), 可以使用链表, 链表里的内容记的是和上一个帧超时时间的差值
 - 事件
 - 内部事件: Network_layer_ready, timeout
 - 外部事件: Frame_arrival, Cksum_err
 - 对于出错率较高的信道, 浪费带宽
 - 选择重传 (Protocol 6, 确认计时器, NAK)
 - 范围限制: 发送窗口 > 1 , 接收窗口 > 1 (注意接收窗口的大小仍是固定的) 要保证接收窗口移动前后, 接收窗口内的帧没有重叠
 - ~~发送窗口尺寸 \geq 接收窗口尺寸时, 接收窗口 $\leq (\text{MaxSeq}+1)/2$~~
 - 发送窗口尺寸 $<$ 接收窗口尺寸时, 发送窗口+接收窗口 \leq 序号个数
 - 优化策略
 - 增加确认计时器 (解决负载不平衡带来的阻塞问题)

- ◆ 确认计时器的超时间隔应明显短于与数据帧关联的计时间隔
- ◆ 启动条件：收到了frame_expected 或者 收到的不是frame_expected但是已经发过了NAK
 - 否定性确认帧NAK（随时发送）
 - ◆ 对于某一个帧只发一次NAK（此后如果还是不对的话发ACK（见上））
- 事件
 - 内部事件：Network_layer_ready, timeout, Ack_timeout
 - ◆ 注意timeout的时候发送窗口的上界不会变为发送窗口的下界，只是把下界重发了一遍（选择重传）
 - 外部事件：Frame_arrival, Cksum_err
 - ◆ Frame_arrival要区分收到的是数据帧还是NAK
- 接收窗口较大时，需要较大缓冲区（缓冲区数量 = 接收窗口的大小）

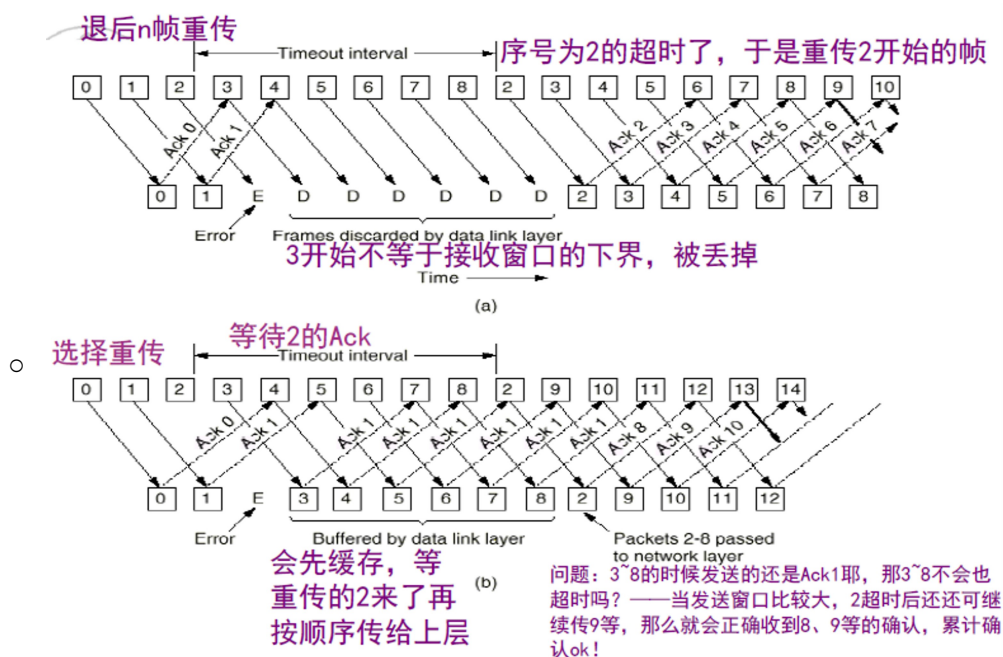


Fig. 3-15. (a) Effect of an error when the receiver window size is 1. (b) Effect of an error when the receiver window size is large.

- 【背诵】协议工程（分为以下四个部分）
 - doc a. 协议说明：提供给用户的服务+协议实体的内部操作
 - verify b. 协议验证：用于设计阶段，进行可达性分析等
 - c. 协议实现
 - d. 协议测试：一致性测试、互操作性测试、性能测试
- 【计算】形式化描述技术（考纲上的要求是会用来描述协议3，我感觉比较偏，背诵可以先不管，到时候考前把两个例子看一遍）
 - 有限状态机模型FSM
 - 基本出发点：认为通信协议主要是由响应多个“事件”（命令/信息到达/内部超时）的相对简单的处理过程组成
 - 优点：简明、精确
 - 缺点：复杂协议的事件数和状态数会剧增，状态爆炸
 - 描述协议3（有噪声信道的单工协议）
 - Petri网模型

- 主要特性：具有较强的对并行、不确定性、异步和分布的描述能力和分析能力
- 分类
 - 条件/事件网 (C/E网)：每个位置最多一个标记，表示条件
 - 位置/变迁网 (P/T网)：每个位置中的标记可以有多个
 - 结构元素、活动元素、变迁实施规则：这个感觉还是看一下ppt，比较绕
 - 描述协议3
- 常用的数据链路层协议（了解即可，可能出干扰项，我挑了几个感觉可能作为干扰项的，剩下的看ppt吧我觉得搬运无意义）
 - HDLC：1976年ISO提出
 - 面向比特，采用滑动窗口技术，校验和用CRC
 - 3种站，2种构型，3种操作模式
 - 帧类型：信息帧/监控帧/无序号帧
 - 构造协议：选择站构型→操作模式→帧种类→12种任选功能→ok
 - X.25 LAPB (X.25的链路层协议，X.25在ch2中提到过)
 - LAP、LAPB是HDLC的子集，帧格式与HDLC完全相同
 - 检错纠错：CRC必用，超时、序号、校验指示可选
 - PPP协议 (Point-to-Point Protocol)
 - SLIP的改进，包括链路控制协议LCP和网络控制协议NCP
 - 面向字符，采用字符填充技术
 - 以帧为单位发送，而非原始IP包
- 【其他】
 - 一些基本概念
 - 结点：网络中的主机和网络设备（路由器、交换机等）
 - 链路：同心路径上连接相邻结点的通信信道
 - 点到点：一条链路上两个相邻结点间的通信
 - 端到端：从源结点到目的结点的通信，通信路径可能由多个链路组成
↑这里的点到点和端到端感觉和平时说的点到点端到端不一样，沉思
 - 实际数据通路（实际上走的网络层→数据链路层→物理层→物理层→...）
虚拟数据通路：认为链路层提供了一个逻辑上的通路，不看到具体的物理层
 - 数据链路控制规程：为使数据能迅速、正确、有效地从发送点到达接收点所采用的控制方式（差不多类似协议，just早期文件叫规程）
 - 差错出现的特点：随机、连续突发
 - 纠错码和检错码都是插入足够多的冗余信息
 - CRC检错能力分析（这里只记结论，推导太数学，可以看ppt）
 - ◆ 若发送的是 $T(x)$ ，接收的是错误的 $T(x)+E(x)$ ，则当 $(E(x)/G(x))=0$ 时不能发现错误
 - ◆ 对于具有 r 个校验位的多项式：
 - ◇ 一定能查出1比特错
 - ◇ $G(x)$ 选 $(x+1)$ 的倍数能查出奇数个比特错

- ◇ 能检查出所有长度 $\leq r$ 的突发差错
 - ◇ 有 $1/(2^{r-1})$ 的概率查不出来 $r+1$ 位突发差错
 - ◇ 有 $1/(2^r)$ 的概率查不出来长度大于 $r+1$ 的突发差错或多个较短的突发差错
- 网卡NIC可以硬件实现CRC校验
- Protocol 1~6
 - 发送方在发下一个帧之前等待一个肯定确认的协议：PAR/ARQ
 - 信道带宽 b bps, 帧长度 L bit, 往返传输延迟 R s, 则信道利用率为 $L/(L+Rb)$
 - ◇ 解决办法：流水线技术, 连发多帧后等待确认
 - ◇ 带来的问题：信道误码率高时重传很多
- 形式化描述技术FDT/形式化方法FM（应用于协议工程研究中）
 - 有啥模型
 - ◇ 有限状态机FSM（扩展：EFSM）
 - ◇ 形式化语言模型LOTOS/Estelle/SDL
 - ◇ Petri网（扩展：时间/随机/高级Petri网）
 - ◇ 进程代数（扩展：随机进程代数）
 - 模型描述能力的增强会在某种程度上增加模型分析的难度