

数据链路层实验—ARQ 协议实验指导

一、实验系统说明

实验系统包括 ARQ 协议发送端程序 (ARQ_S*.exe)，接收端程序 (ARQ_R*.exe) 和信道仿真程序 (xds.exe)，发送端发送的数据经过信道仿真程序传输给接收端。信道仿真程序可以模拟实际传输过程中信道上数据帧或应答帧丢失、差错和传输时延等各种情况，从而验证 ARQ 协议发送端、接收端程序应对传输过程中各种问题的能力。

例如：图 1 信道信息中“7:81H>80H”表示信道上第 7 个信息帧产生差错，信息值由 81H 变为 80H；“4:81H>丢失”表示信道上第 4 个信息帧产生丢失；



图 1 信道仿真过程

二、关于协议示例程序中数据帧结构的说明

1. 数据帧和应答帧以字节为单位；
2. 数据帧：低 4 位 D3~D0 为数据段(取值 0000B~1001B, 即 0~9)，最高位为校验位(D7)，发送序号段：D6~D4；
3. 应答帧：确认帧 ACK：低 4 位 D3~D0 取值 1111B (FH)，否认帧 NAK：低 4 位 D3~D0 取值 1110B (EH)，发送序号段：D6~D4；
4. 注意：发送序段为三位，但实际使用时，不同协议可以选择使用其中的一位或几位作为发送序号，或者干脆不使用，需要注意以下的程序说明。

三、示例程序说明及实验中需要回答的问题

ARQ 协议 0: 理想信道情况

程序实现：

该程序假设信道是理想信道，即数据在传输过程中不会出现差错、丢失和随机延迟，发送端直接发送数据，接收端直接接收数据。

问题：

1. 若信道是理想的，数据是否能从发送端可靠地到达接收端？
2. 若信道是不理想的，发送端程序发送的数据是否能可靠到达接收端程序？会出现什么现象？

ARQ 协议 1: 引入检错和应答帧, 解决差错问题

实现:

1. 数据帧含有一位(偶)校验位 D7: 接收端程序对数据帧进行检错, 通过应答帧告诉发送端接收是否正确(应答帧是指确认帧 ACK 或否认帧 NAK);
2. ACK 值=0FH (即二进制 00001111), NAK 值=0EH (即 00001110);
3. 发送端程序收到 ACK, 将发送下一个数据;
4. 发送端程序收到 NAK, 将重发刚才发送的数据。

问题:

1. 什么情况下发送端发出的数据可以可靠到达接收端?
2. 什么情况下发送端发出的数据不能可靠到达接收端? 会出现什么问题?

ARQ 协议 2: 引入超时计时器, 解决“死锁”问题

实现:

1. 发送端建立一个超时计时器, 每发送完一个数据帧就启动它, 如果在超时时间内没有接收到应答帧, 则重新传输数据帧;
2. 发送端示例程序的超时时间为 1000ms。

问题:

1. 如果发送端程序的超时时间设置较长或较短, 会出现什么问题? 如何选择超时时间的长短?
2. 还存在什么问题?

ARQ 协议 3: 数据帧携带发送序号 0~1, 解决“重复帧”问题

实现:

1. 每个数据帧(D6 位)携带不同的发送序号 $N_s=0\sim 1$;
2. 如果接收端收到相同 N_s 的数据帧, 则丢弃并回送一个 ACK;

问题:

1. 能彻底解决“重复帧”问题吗?
2. 还存在什么问题?

ARQ 协议 4: 引入确认帧携带发送序号 0~1

实现:

1. 每个确认帧 ACK (D6 位)携带不同的发送序号 $N_s=0\sim 1$;
2. 如果发送端收到的 ACK 中的 N_s 与刚才发送的数据帧中的 N_s 相同, 则发送下一个数据帧, 否则重发刚才的数据帧。

问题:

1. ACK 有必要携带 N_s 吗? 它能解决什么问题?
2. 还存在没有解决的问题吗?