**协议4、协议5和协议6对比**

1. **关于ACK**

协议4：只有捎带ACK, 无单独的ACK和NAK。无ACK-timer

协议5：只有捎带ACK, 无单独的ACK和NAK。无ACK-timer

协议6：有捎带ACK和单独的ACK、NAK。接收帧时需要设置ACK-timer

**2、窗口W (发送窗口WT，接收窗口WR)、帧序号fr-seq、序号比特数n定义**

协议4：n=1, fr-seq=0 or 1, WT=WR=1

协议5：n=3, fr-seq=0 to 7(MAX\_SEQ)，WT最大为7, WR=1

协议6：n=3, fr-seq=0 to 7(MAX\_SEQ), WT最大为4，WR=4, 即NR\_BUFS= (MAX\_SEQ +1)/2

**3、关于缓存buffer的管理（与代码实现相关）**

协议4：仅有发送一个缓存，用来存放next-frame-to send帧

协议5：发送有一个2n大小的缓存buffer[i]，帧序号与缓存数组序号一一对应，接收数据不需要放在缓存中，直接提交网络层

协议6：发送和接收各维护一个与窗口大小（NR\_BUFS=2n-1）一样的缓存数组out\_buf[NR\_BUFS]和in\_buf[NR\_BUFS]，帧序号fr-seq与缓存数组序号i的对应关系是：i=fr-seq%NR\_BUFS

**4、收到序号正确的一帧：**

协议4：**仅有捎带确认**，无ACK-timer, 每正确收到一帧直接回复当前buffer中的数据（无network-layer-ready事件），只修改ack的序号

协议5：**仅有捎带确认**，无ACK-timer, 正确收到一帧后等待网络层数据（network-layer-ready事件）或超时重传时才能将ack发给对方

协议6：**有捎带确认和单独的ACK、NAK控制帧，**有ACK-timer, 正确收到一帧后等待网络层数据（network-layer-ready事件），如果有则捎带确认；如果ACK-timer超时则单独发送 ACK帧。

**5、对帧序号错误和校验错误的处理**

协议4：立即**重传上一帧(捎带确认)**

协议5：**捎带确认**，等待网络层数据后与新数据一起采用捎带确认方式将上一个正确收到的帧的ACK（序号=(frame-expected+ MAX\_SEQ)% (MAX\_SEQ+1）)发送出去

协议6：如果没有发过NAK，则**立即发送NAK帧**（序号=(frame-expected+ MAX\_SEQ)% (MAX\_SEQ+1）），不使用捎带确认方式。

