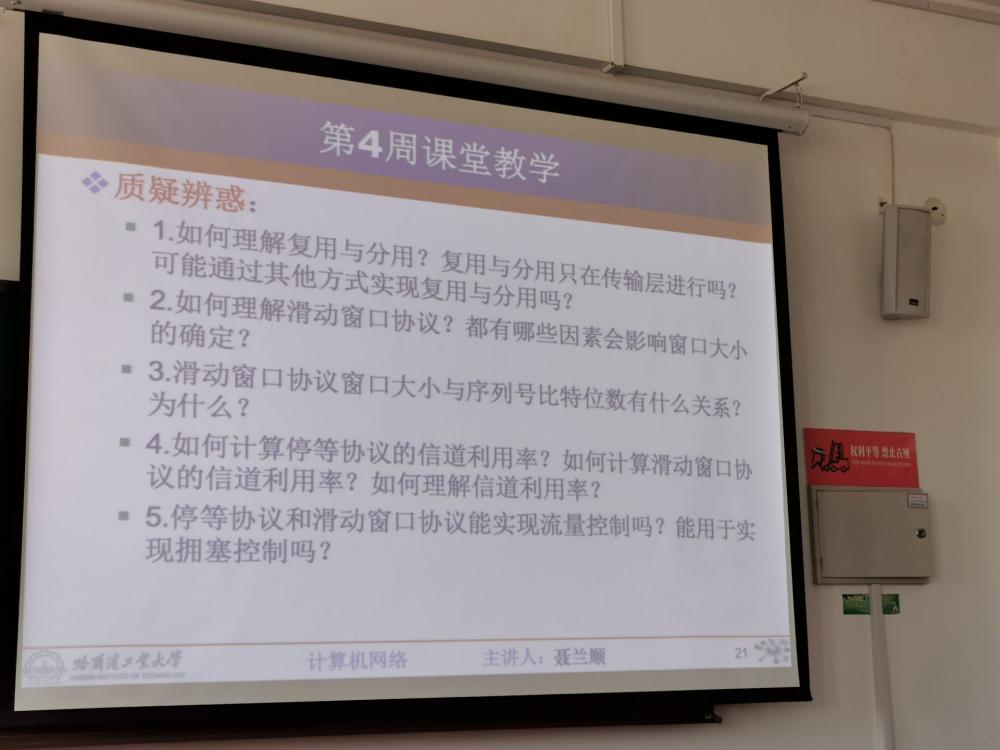
**计算机网络课堂讨论**

6班B组 ：范天泷 陈鋆 罗鹏豪 高天赋



1. 在传输层中，分用和复用更像是一对互逆的关系。多路复用是一个类似于合并的过程，针对的是发送方：不同套接字的数据块经过封装后一块传递到网络层（资源有限的情况下对这种有限资源的一种适应）；多路分用是一个类似于分割的过程，针对的是接收方：接收方能够将接收到的混在一起的数据块分开交付到正确的套接字上（匹配复用的过程）。

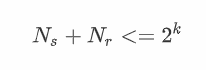
并不只在传输层进行，如之前所学过的数据交换中的电路交换（物理层），就同样有着时分复用、频分复用等多种复用与分用思想的应用。

首先，复用与分用在传输层层面实现是十分合理的，因为传输数据的复用和分用，是发生在端到端之间的，与中间系统（如路由器）关系不大，而传输层恰好位于端到端层的最底端，并且，上层的应用层又通过多个socket接口与其连接，也需要传输层实现数据的复用和分用。并且，UDP与TCP的复用实现都十分得合理，我认为暂且没有什么更好的替代方案。

1. 滑动窗口协议是一个实现流水线机制的协议——众所周知，停-等协议对信道的利用率过低，致使网络吞吐率也低，为了提高利用率，我们想到了流水线的机制，让发送方在收到ACK前连续发送多个分组。这就要求：增加序号范围，给每个分组一个唯一的序号；发送方与接收方都要能缓存分组。于是就有了能够通过窗口长度、窗口滑动分配序号实现这二者的滑动窗口协议。所以，滑动窗口协议是一个为了能在网络双方中实现流水线机制而设计的协议。

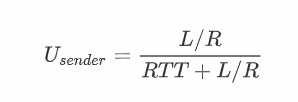
影响窗口大小确定的因素：（1）发送方能用于实现滑动窗口协议的缓存大小。（2）接收方能用于实现滑动窗口协议的缓存大小。（3）网络的拥挤程度（拥塞控制中，为了避免排队时延过长）。（4）当网络足够通畅时，若传播时延较长，可以适当增加窗口大小以提高信道利用率及吞吐率。

1. 设比特位数为k，则需要满足：

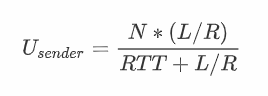


原因：若窗口过大序列号不足，可能导致发送方未收到ACK情况下重传而被接收方错误缓存的情况，原因是发送方在此轮的序号与接收方在下一轮的序号重合。而接收方和发送方窗口所包含的序列号的并集最大情况下数目为Ns + Nr（接收方窗口第一个序号等于发送方窗口最后一个序号加一） ， 故只需要满足上述条件便能保证上述错误情况不会发生。

停-等协议：



窗口滑动协议：



对信道利用率的理解：我认为信道利用率是一个描述流水线实现程度的量。虽然其定义是——发送方实际忙于将比特送进信道那部分时间与发送时间之比，这里所说的是时间的比，但从意义上，这个比的意义相当于比特流充斥信道的比例（利用率100％的话相当于第一条信息已经确认到了，刚好发送完最后一点消息，说明整个信道都充斥着比特流），因此，它的本质跟命名是一致的，并且可以通过利用率，来进一步刻画出流水线的实现程度（越高说明信道越被充分利用，流水线程度越高）

1. 流量控制需要通过限制发送方的发送速率来实现。窗口滑动协议中可以通过限制窗口的大小来改变发送速率，从而实现流量控制；停-等协议本身效率极低，很少需要控制，但在需要控制的情况下，就我们上课所学的停-等协议中没有提及流量控制的实现，一个不知道是否可行的想法是通过分组的大小来控制发送速率。

拥塞控制也同理，需要通过控制发送方速率来实现，但还需要对网络情况进行检测从而发现拥塞的情况。窗口滑动协议通过3个重复ACK事件与timeout事件来检测网络情况，并且通过慢启动、AIMD等机制来实现对发送速率的控制，从而实现拥塞控制。停-等协议本身效率极低，很少需要控制，但在需要控制的情况下，它可以通过timeout事件检测网络情况，但难以控制发送速率大小（分组大小前后不一致可能出现问题），故其难以实现拥塞控制。