### 实验结果分析

1、实验结果简介：

此次实验，本小组采用了无nak以及捎带确认的Go-Back-N协议以及有nak和捎带确认的Go-Back-N协议以及选择重传协议，能够较长时间地在有误码信道环境中无差错传输，同时，在传输过程中，如果出现错误，无nak的协议会丢弃错误帧，使发送方计时器超时重发，有nak的协议会发送nak给发送方，发送方收到相应nak后就会重发相应帧，因此能够可靠地实现通信。

1. 协议参数的选取：

一个数据帧共有256 + 3 + 4 = 263字节，因此发送时延为263 \* 8 / 8000 = 263ms，发送时延为270ms，则α = 263 / 270 = 0.974，对不带捎带确认的协议U = 1 / (1 + 2 \*α) = 0.339，对捎带确认的协议，U = 1 / (2 + 2 \*α) =0.253,因此理论上发送窗口数为4即可使利用率最大，在实验中我们选取MAX\_SEQ为7，而NR\_BUFS = ((MAX\_SEQ + 1) / 2) = 4。

对ack\_timer和data\_timer的选取，ack\_timer作用是当检测到发送方没有发送数据帧无法捎带确认时主动发送ack，因此比发送时延稍长即可，我们选取为276ms，data\_timer由于我们采用累计ack确认的方式，因此其值为263 \* NR\_BUFS + ACK\_TIMER + NR\_BUFS - 1

1. 理论分析：

由于需要携带帧讯息，因此最大的信息利用率为约为256/263≈97.3%。由于信道的最大比特率为8000bps，可得出每传输一个字节耗时1ms；每帧的附加信息固定为10，耗时10ms；若出现转义字符，则可能增加时间。现在假设信道上始终有数据需要传送，这样就可以简化模型。在有10-5的错误率的信道上，算下来在这100000个比特中可以传送100000/（263\*8）≈48个数据包，则可得出每48个数据包会有一个出错，假设每出错一次，在限定时间内可以重传该帧使其成为正确帧。则每传送48个数据包需要传送48+1=49次，此时信道利用率≈95.3%，而实际上由于程序设计的原因，当一个数据包超时后，往往要重复多次传输给数据包造成信道的浪费，若有k次重传，则信道利用率为（256/263）\*（48/（48+1+k））。

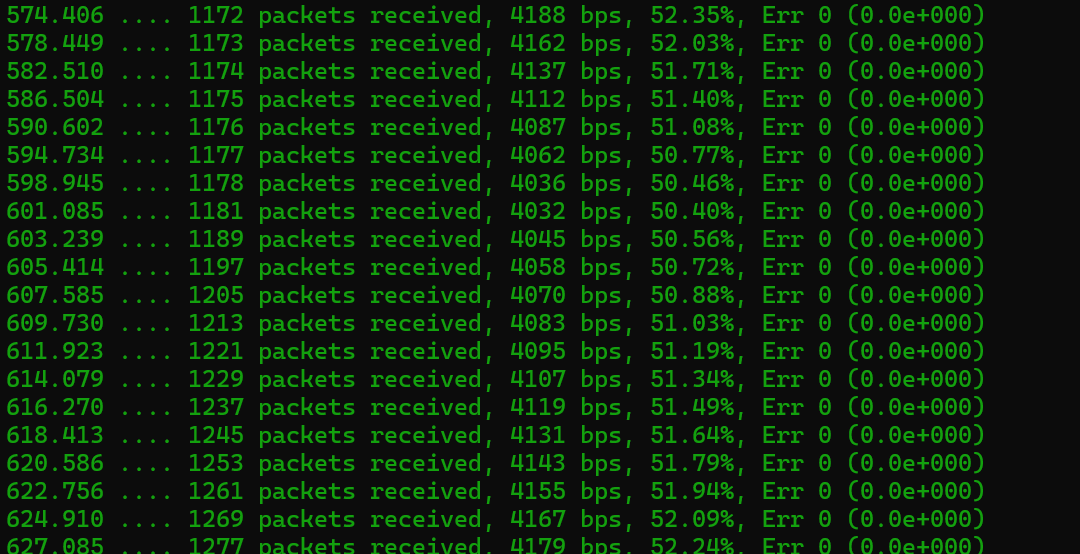
若为回退N协议，出错后需要重传后面所有窗口，则利用率会更低。

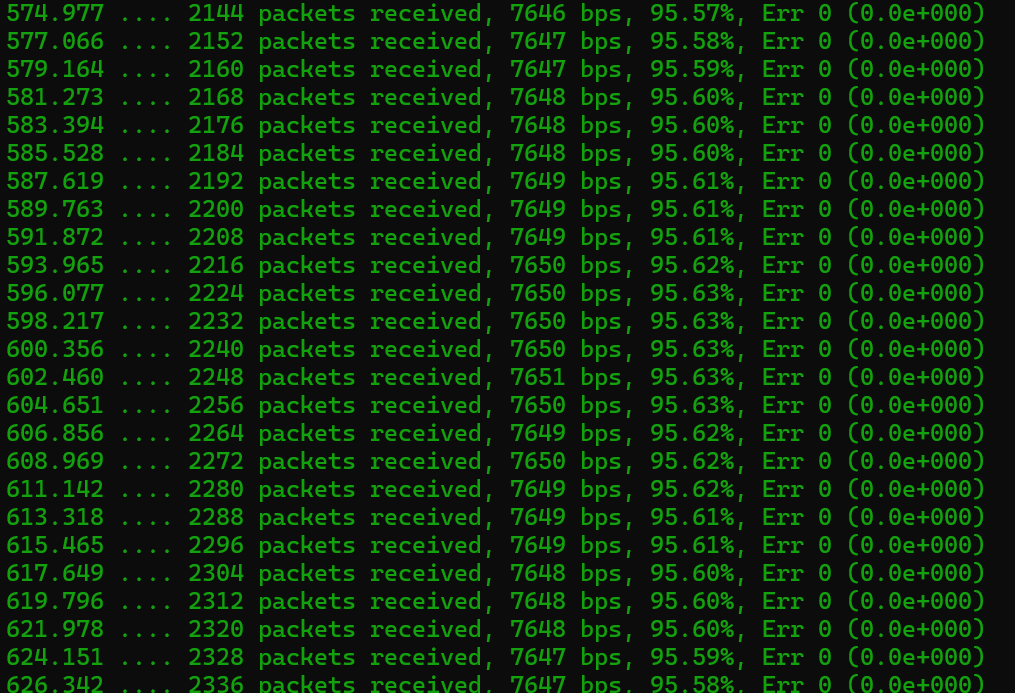
4、实验测试：

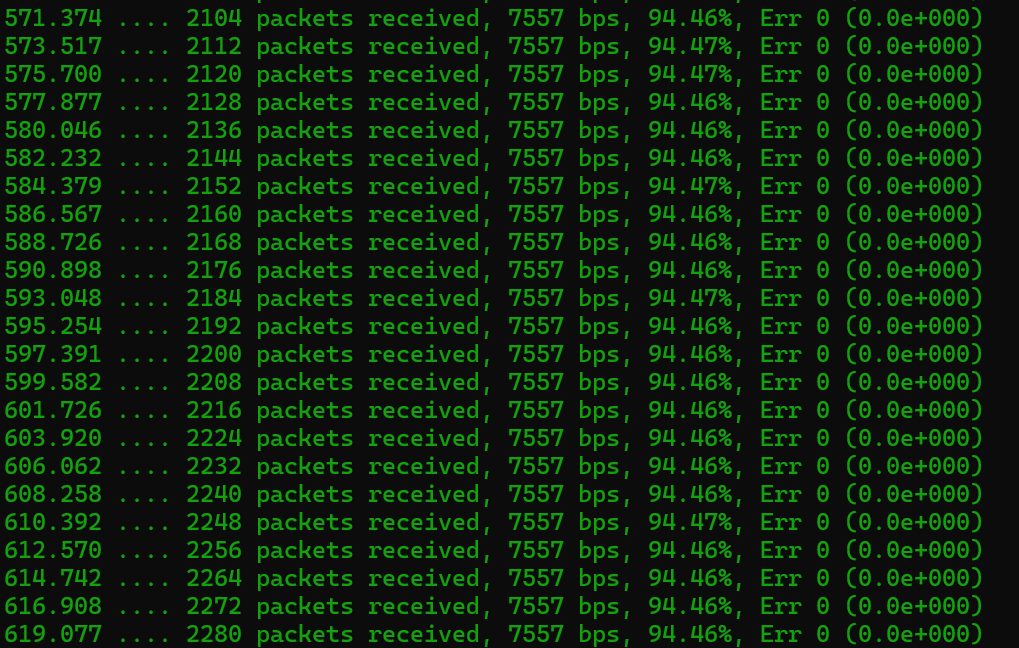
#### 性能测试记录表

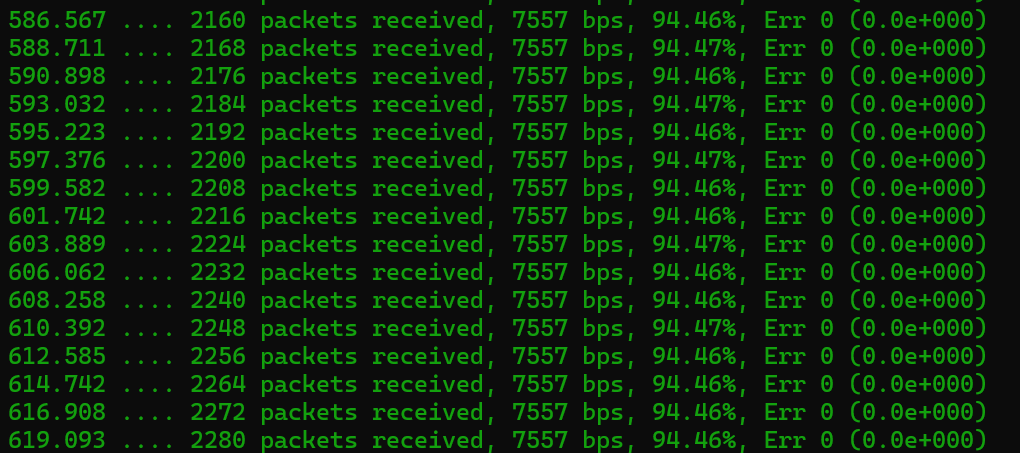
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 命令选项 | 说明 | 运行时间(秒） | GoBackN算法（无nak）  线路利用率(%) | | Selective算法  线路利用率(%) | | GoBackN算法  线路利用率(%) | |
| A | B | A | B | A | B |
| 1 | -–utopia | 无误码信道数据传输 | 626 |  |  |  |  | 95.58 | 52.24 |
| 2 | 无 | 站点A分组层平缓方式发出数据，站点B周期性交替“发送100秒，停发100秒” | 626 |  |  |  |  | 43.70 | 75.87 |
| 3 | –-flood --utopia | 无误码信道，站点A和站点B的分组层都洪水式产生分组 | 619 |  |  |  |  | 94.46 | 94.46 |
| 4 | –-flood | 站点A/B的分组层都洪水式产生分组 | 605 |  |  |  |  | 77.65 | 74.06 |
| 5 | --flood -–ber=1e-4 | 站点A/B的分组层都洪水式产生分组，线路误码率设为10-4 | 621 |  |  |  |  | 26.50 | 26.80 |

运行时间要求超过10分钟。









**五 研究探索**

1. CRC 校验能力

CRC校验码具有以下检错能力：

检查出全部单个错；检查出全部离散的二位错；检查出全部奇数个错；检查出全部长度小于或等于K位的突发错。

1. 由于本次试验过程的误码信道是一个比较固定的误码率，而在实际生活当中的误码率不是稳定的，可能会因为传输环境的不同，使得误码率波动比较大的，例如，下雨天和晴天，高噪声和低噪声的情况，传输的距离也是影响因素。对于这种动态的误码率的通信过程，可能需要其他的一些参数来控制基本参数值（窗口大小，重传时间等等）来完成。

3.对等协议实体之间的流量控制

在我们设计的协议当中，流量的控制主要通过接收窗口，发送窗口还有确认机制来实现。因为有窗口大小的限制，发送方不会一次性发送过多信息导致接收方被信息洪流所淹没，导致信息丢失。这样可能会导致信道的利用率降低，但是如果合理的设计窗口大小，依然可以达到较高的信号利用率。