

南开大学

计算机网络实验报告

实验 3-4



专 业_____信息安全_____

学 号_____2113662_____

姓 名_____张丛_____

班 级_____信安一班_____

一、实验目的

基于给定的实验测试环境，通过改变延时和丢包率，完成下面 3

组性能对比实验：

- (1) 停等机制与滑动窗口机制性能对比；
- (2) 滑动窗口机制中不同窗口大小对性能的影响（累计确认和选择确认两种情形）；
- (3) 滑动窗口机制中相同窗口大小情况下，累计确认和选择确认的性能比较。

二、实验要求

- 控制变量法：对比时要控制单一变量（算法、窗口大小、延时、丢包率）
- Router：可能会有较大延时，传输速率不作为评分依据，也可自行设计
- 延时、丢包率对比设置：要有梯度（例如 30ms,50ms, …; 5%, 10%, …）
- 测试文件：必须使用助教发的测试文件（1.jpg、2.jpg、3.jpg、helloworld.txt）
- 性能测试指标：时延、吞吐率，要给出图、表并进行分析

三、实验内容

图表分析中用文件**传输的时间**来标志性能。

也因为对于相同的文件来说，也是由时间来计算传输速率和吞吐

率的。

统计传输时间的代码如下：

```
int time_begin = clock();
send_buffer((char*)(filename.c_str()), filename.length()); //文件名
send_buffer(buffer, len); //文件内容
int time_end = clock();

cout << filename << "文件传输完毕" << endl;

int time = (time_end - time_begin) / CLOCKS_PER_SEC;
cout << "传输时间: " << time << endl;
if (time != 0)
{
    double kbps = ((len * 8.0) / 1000) / time;
    cout << "吞吐量: " << kbps << " kbps" << endl;
}
```

使用提供的路由器来控制丢包率和时延：

A screenshot of a 'Router' configuration window. The window has a title bar with a red, yellow, and green icon and the text 'Router'. It contains several input fields for configuration: '路由器IP:' with '127 . 0 . 0 . 1', '服务器IP:' with '127 . 0 . 0 . 1', '端口:' with '4003', '服务器端口:' with '4002', '丢包率:' with '5' and a '%' symbol, and '延时:' with '5' and 'ms'. There are '确定' (OK) and '修改' (Modify) buttons. Below these is a '日志' (Log) section with a text area showing 'Router Ready!', 'Misscount :20 .', and 'Delay :5 ms .'.

Router

路由器IP: 127 . 0 . 0 . 1 服务器IP: 127 . 0 . 0 . 1

端口: 4003 服务器端口: 4002

丢包率: 5 % 延时: 5 ms

确定 修改

日志

Router Ready!
Misscount :20 .
Delay :5 ms .

1. 停等机制与滑动窗口机制性能对比

丢包率变化

控制变量：

传输 1.jpg

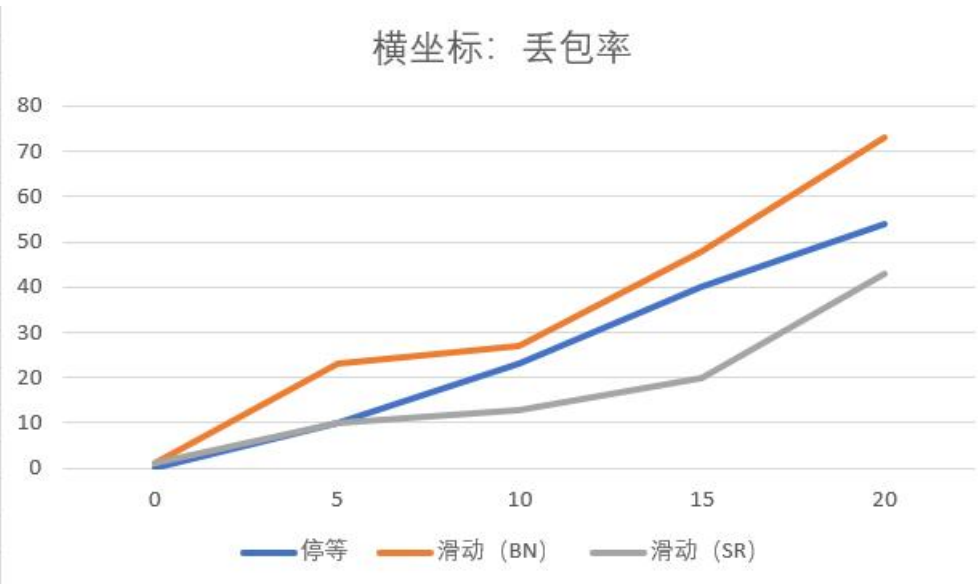
时延：5ms

滑动窗口大小：10

收集的数据：

丢包率	0	5	10	15	20
停等	0	10	23	40	54
滑动 (BN)	1	23	27	48	73
滑动 (SR)	1	10	13	20	43

用上面的数据在 Excel 绘制折线图（纵轴为传输时间）：



结论：

在时延相等的情况下，停等机制和滑动窗口机制的性能都会随丢

包率的升高而变差。

在我的实验中，**时延相同丢包率相同的情况下**，三者的性能是：

选择确认>停等机制>累积确认

这里似乎值得分析，因为正常情况下我们会认为，滑动窗口机制比停等机制更快，因为滑动窗口机制允许发送方连续发送多个数据包，提高了网络的利用率。

我先回头检查代码和日志，确认我实现的停等机制和累计重传是符合要求的。

选择重传比停等机制快很容易理解，选择重传既可以一次发送多个数据包，超时重传时也只需要重传丢失的数据包。

为什么我这里的累积确认会比停等快？个人分析认为：

1. 自己编写的累积确认程序只实现了基本的内容，实际运用中肯定会有各方面的优化。

2. 在实验中，是在环回地址(127.0.0.1)中，本地给本地传输文件，**往返时延（RTT）极短**，停等机制不需要等待很长时间来接收确认，也就相对来说提高了停等机制的网络利用率。

而对于累积确认，一旦丢包，需要重传整个窗口的数据包，使得本不怎么占据优势更加雪上加霜。且丢包率越高，性能越差。

（书上说设计流水线就是为了提高信道利用率。）

3. 在累积确认中，对发送窗口进行的维护等等，可能会消耗一定的时间。（因为在丢包率为 0 时，停等都还是快一丁点儿。）

时延变化

控制变量：

传输 1.jpg

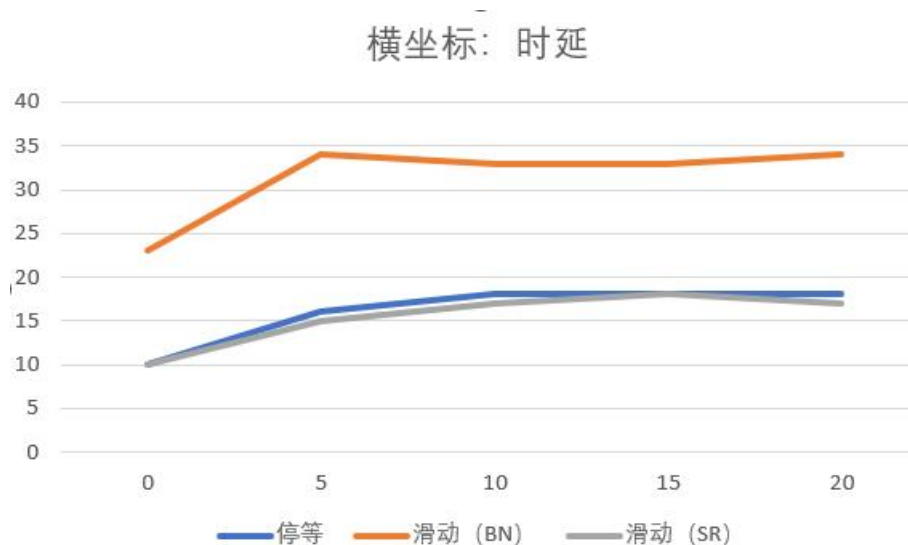
丢包率：5%

滑动窗口大小：10

收集的数据：

时延	0	5	10	15	20
停等	10	16	18	18	18
滑动 (BN)	23	34	33	33	34
滑动 (SR)	10	15	17	18	17

用上面的数据在 Excel 绘制折线图（纵轴为传输时间）：



结论：

与丢包率的变化不同，随时延的增加，传输时间**先增加**，后面就**趋于稳定**。

因为设置的延时到了一定的值，影响传输时间的主要因素可能就

不是这个延时了。

2.滑动窗口机制中不同窗口大小对性能的影响

回退 N 步

控制变量：

- 传输 1.jpg
- 丢包率： 5%
- 时延： 5ms

收集的数据：

窗口大小	5	10	15	20	25	30	35
BN传输时间	37	34	34	32	29	29	31

用上面的数据在 Excel 绘制折线图（纵轴为传输时间）：



可以看到，总体趋势是，随窗口的变大，累积确认的传输时间是

变小的（性能更好），但过大时（35 及以上），传输时间又会多一点。

一方面，窗口变大也就是增加流水线，提高网络利用率；但窗口过大，也意味着超时重传时的数据包数量会很多，导致性能的下降。

在课上我们也学过慢启动，在连接刚开始的时候，通过逐渐增大发送方的窗口大小来快速适应网络的带宽，找到适当的发送速率，以便有效地利用可用的带宽，同时避免引起网络拥塞。（虽然并不是一定要在累积确认里面。）

选择重传

控制变量：

丢包率：5%

时延：5ms

若传输 1.jpg，得到的数据为：

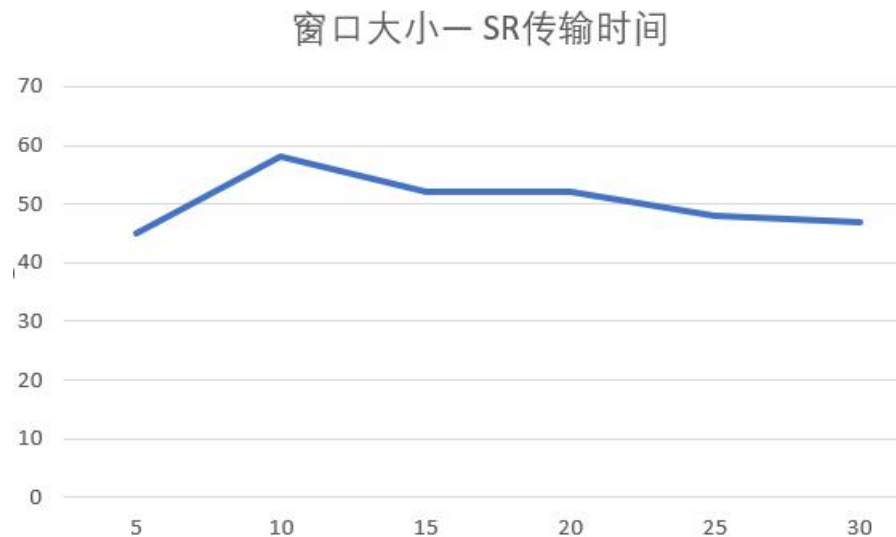
窗口大小	5	10	15	20	25	30
SR传输时间	16	15	16	15	14	13

可见传输时间基本稳定（因为传输很快）。

传输 2.jpg，收集的数据：

窗口大小	5	10	15	20	25	30
SR传输时间	45	58	52	52	48	47

用上面的数据在 Excel 绘制折线图（纵轴为传输时间）：



可见，随窗口变大，SR 的传输时间先变大，然后又减小（略趋于平缓）。

对于时间减小的阶段应该比较容易解释，流水线增加，信道利用率提高，性能好。

多次实验中，窗口从 5 到 10 这一段，传输时间都是会变大的，包括传输 3.jpg 等等。

猜测是此时引入的额外开销会大于利用率的提高。但额外的开销能是什么呢？在设计的选择确认中，似乎能列举的也只是，窗口维护、缓存区的维护、丢包重传，至于往返时延和路由器设置的时延都是固定的。

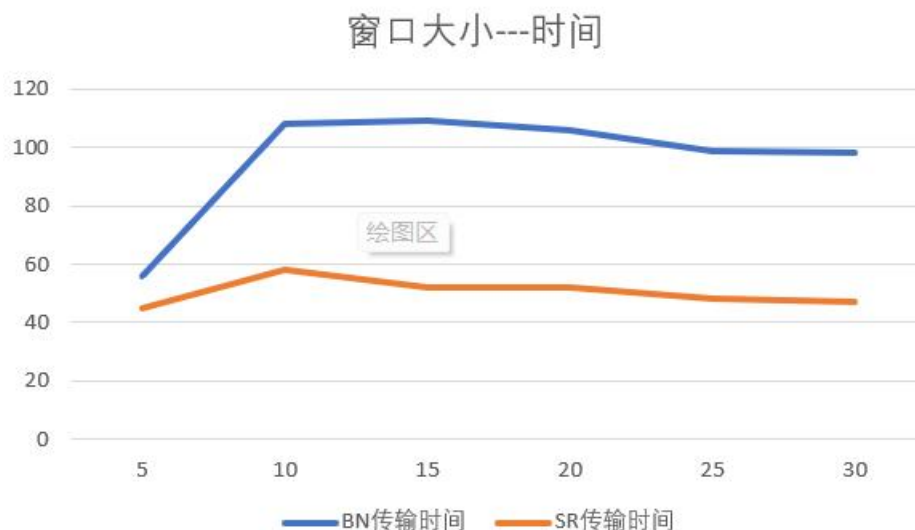
3. 累计确认和选择确认的性能比较

传输文件：2.jpg，时延 5ms,丢包率 5%

数据：

窗口大小	5	10	15	20	25	30
BN传输时间	56	108	109	106	99	98
SR传输时间	45	58	52	52	48	47

折线图：



可见，在窗口大小相同的情况下，选择确认总优于累积确认。

都利用流水线提高了网络利用率，但选择确认不需要无用的重传，所以优。

但既然如此选择确认如此优秀，生活中都用选择确认不就好了。在写实验 3-3 的代码时我发现，在发送端除了维护窗口外，在接收端还需要添加对缓存数据包的维护（对 3-2 来说），显然在实际中会更复杂。

也就是说，选择确认在传输性能方面具有优势（特别是丢包率大的网络），但在部署方面可能有更多的要求。而累积确认相对简单易实现，对于一些网络环境来说，可能更容易管理。

四、思考与总结

实验过程的数据收集：



且文件 3.jpg，helloworld.txt 也都传输过，得出的图表和结论都一致。

实验中的有些结论并不符合我一开始的预想，本着实事求是的原则，进行了多次反复实验，进行记录，最后对观察的结果尝试着分析。

不过想来，不完全符合想象倒是正常的，实际和理论知识有所区别，个人实验和实际又有所区别，且个人实验也没有反复优化。

实验本就是让我们分析验证，锻炼思维，引导我们自主思考。我也希望我确实能够做到这一点。

