南開大學
Nankai University

计算机学院和网络空间安全学院

计算机网络课程实验报告

实验 3-4：基于可靠 UDP 的实验数据分析

姓名：魏伯繁

学号：2011395

专业：信息安全

2022 年 12 月 29 日

目录

1 实验目的与实验要求

1.1 实验目的

本次实验在已经完成 3 个版本的可靠 UDP 传输的基础上，通过调节实现代码中的参数获得不同的传输数据并对其进行分析。

通过本次实验，可以对计算机网络中的传输算法、拥塞控制算法有更为清晰的了解和认识

1.2 实验要求

在 router 中通过改变延迟时间和丢包率，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

(3) 有拥塞控制和无拥塞控制的性能比较

2 停等机制与滑动窗口的性能比较

2.1 实验数据

在该部分，我就停等机制与滑动窗口机制对传输性能的影响做了比较，在滑动窗口中，统一使用滑动窗口为 4 的情况进行对照组实验。

首先，我们固定 router 的时延不变，动态调整丢包率，观察传输的性能

在本次实验中，使用的传输文件为课程组提供的图片 1.jpg，图片大小为 1857352byte，下图中横轴表示丢包率，纵轴表示传输时延。传输时延越低传输速度越快。其中，橘黄色的折线代表滑动窗口为 4 时的传输时延，蓝色的折线则代表停等机制下的传输时延。

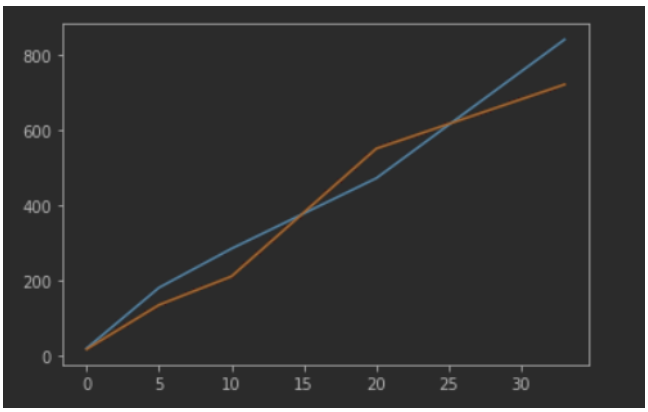

图 2.1: 丢包率-时延变化图

根据上述的实验条件，我们也可以根据计算求得其吞吐率，吞吐率数据如下图所示：

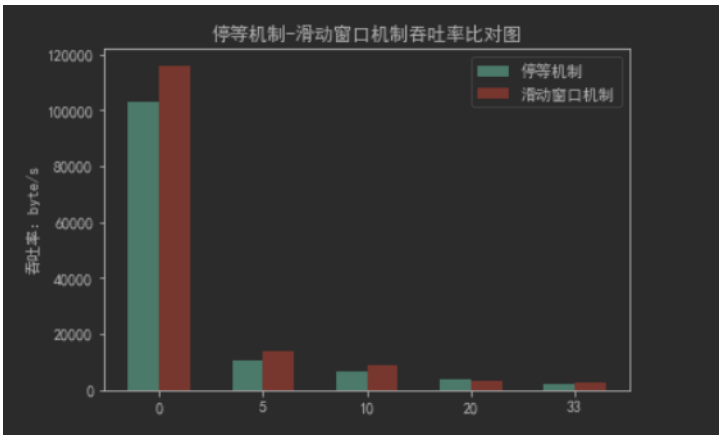

图 2.2: 丢包率-吞吐率变化图

接下来，我们固定 router 的丢包率为 5%，通过改变传输时延来检测其性能变化，通过对 router 延时为 0ms、25ms、50ms、100ms、250ms 来观察其变化对传输的影响。其中橘黄色的折线代表着滑动窗口机制的传输时延，蓝色的折线代表着停等机制的传输时延。

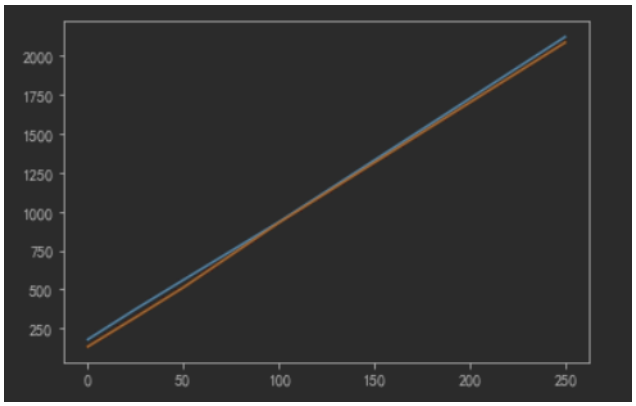

图 2.3: router 时延-传输时延变化图

同理，可以根据上述数据给出计算得到的传输吞吐率，并做比较：

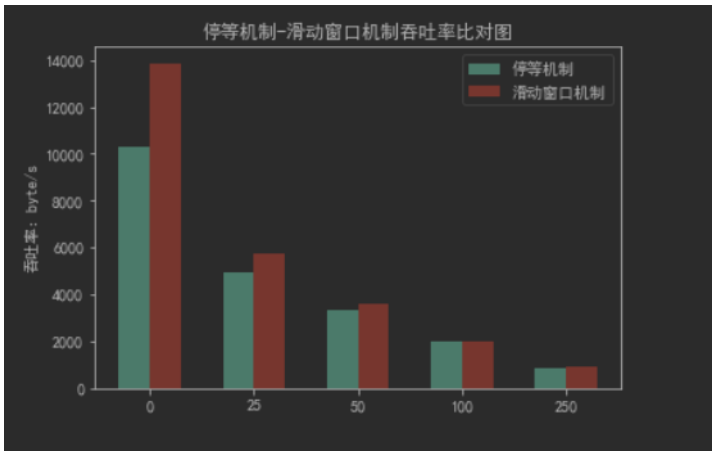

图 2.4: router 时延-吞吐率变化图

2.2 实验分析

通过上面的四张图片，我们不难分析出不管是固定 router 的丢包率改变时延还是固定 router 的时延改变丢包率都是滑动窗口在性能上更胜一筹，唯一的异常点在 router 的时延为 0，丢包率为% 时，停等机制的传输吞吐率要大于滑动窗口 GBN，其余所有情况都是 GBN 的性能更佳。

当 router 时延确定时更改丢包率，我们会发现当丢包率增大时，滑动窗口和停等机制的传输时延的吞吐率会主键变小，这是因为当丢包率非常大时，滑动窗口机制会陷入到不停的重传中从而失去了拥有多数传输序列号的优势。也就是说，在网络畅通（非丢包）时，GBN 会大量重传丢失的包从而导致其性能大幅下降，而停等机制由于其不管如何都只有一个数据包缓冲区，所以丢包率上升对其性能影响相对于 GBN 来说较少。

同样的趋势也适用于保证 router 的丢包率不变而改变其 router 时延，router 的时延过大毫无疑问会浪费大量时间在“等待”上从而浪费了本可以加速传输大量数据包的时间，导致大量数据包堆积，从而导致效率的大幅度下降

当然，有一个问题也是不容忽视的，就是 GBN 的多线程问题，由于 GBN 的实现中需要同时具备收发两个线程，所以会存在其对某些全局变量的访问冲突问题，为了解决这个问题一个比较科学的解决方案就是对全局变量加锁，而加锁就必然导致两个线程必然不是在所有时刻都能并行进行的而是存在间歇的，这样的情况也会对效率产生一些影响。

3 滑动窗口机制中不同窗口大小性能比较

3.1 实验数据

在该部分，我使用实验 3-2 编写的 GBN 传输来完成性能测试比较，通过在相同条件下调整滑动窗口的大小来做性能比较。

在本部分，具体的对照实验也分为两组，第一组为固定 router 的丢包率不变改变 router 传输时延进行性能测试；第二组为固定 router 的传输时延不变改变 router 的丢包率来进行性能测试。

首先，我们通过固定 router 的时延不变，观察在不同的 router 丢包率的情况下数据传输的性能如何，我们选取的窗口大小分别为 4、16、128、256、1024，router 的丢包率我们分别设置为 0%、5% 和 10%，此时的 router 的时延则为 0ms，下面是观察到的数据图像：

其中蓝色的代表丢包率为 0%，橘黄色代表丢包率为 5% 绿色代表丢包率为 10% 横轴为滑动窗口大小，纵轴为传输时延，单位为秒

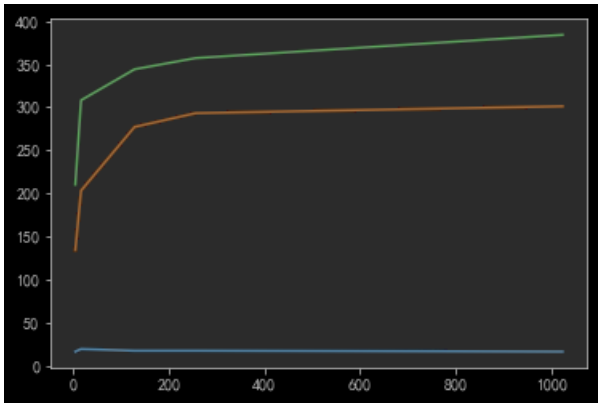

图 3.5: router 窗口大小-传输时延变化图

接下来，我们固定 router 的丢包率不变为 0%，改变 router 的传输时延为 0ms、20ms 和 50ms 观察传输的时延并绘制图像，其中蓝色代表 router 时延为 0ms，橘黄色代表传输时延为 20ms，绿色代表传输时延为 50ms，纵轴单位为秒，横轴为滑动窗口大小。

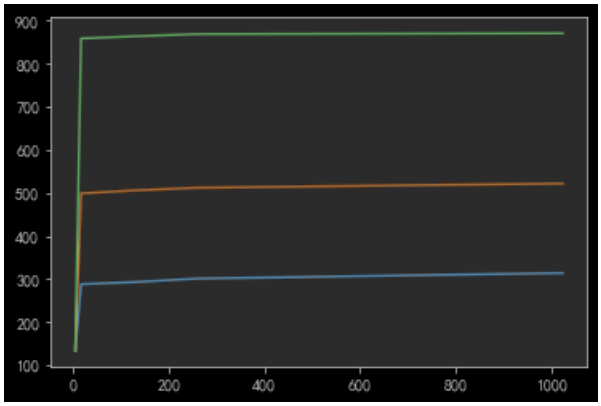

图 3.6: router 窗口大小-传输时延变化图

为了更清晰的表示其变化情况，我们分别对不同的 router 时延下，丢包率为 0% 的数据传输情况单独绘图：

下图展示了 router 时延为 0ms 情况下的性能侧视图

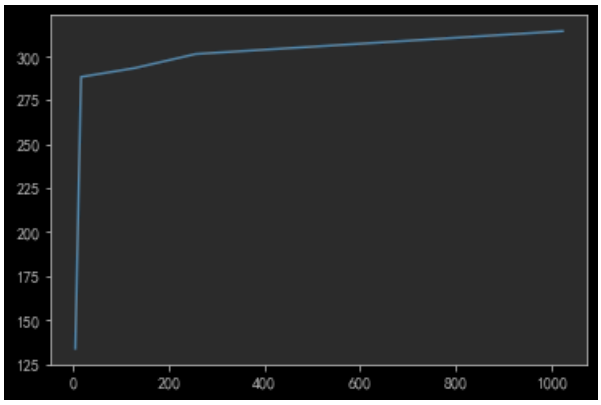

图 3.7: router 窗口大小-传输时延变化图

下图展示了 router 时延为 20ms 情况下的性能侧视图

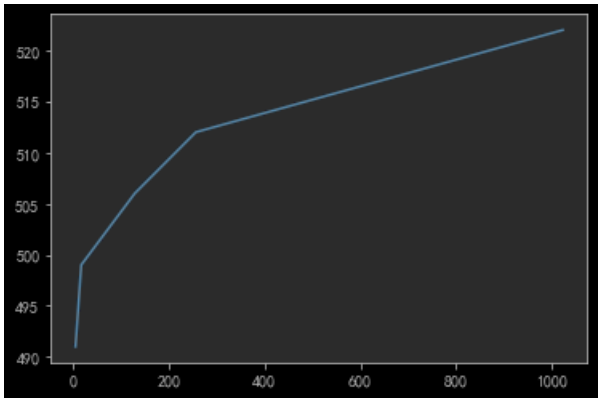

图 3.8: router 窗口大小-传输时延变化图

下图展示了 router 时延为 50ms 情况下的性能侧视图

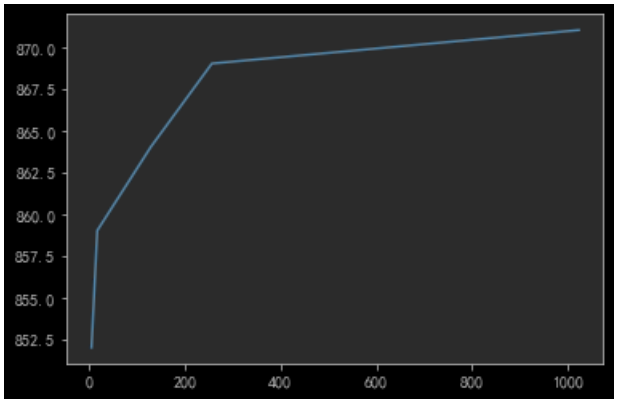

图 3.9: router 窗口大小-传输时延变化图

根据以上测得的数据，我们可以计算传输的吞吐率，我们使用的测试图片是老师统一发放的 1.jpg，其大小为 1857352byte

我们首先绘制固定 router 转发时延为 0ms 时的不通过丢包率下不同窗口大小的吞吐率情况图

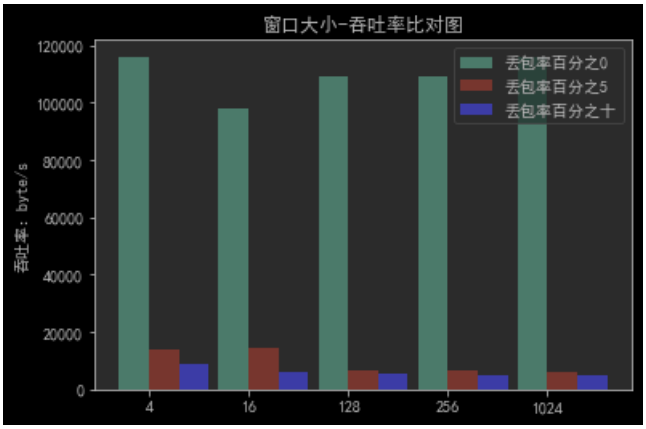

图 3.10: router 窗口大小-吞吐率变化图

接下来我们绘制固定 router 丢包率不变的情况下改变 router 转发时延时不同的窗口打下的吞吐率情况图

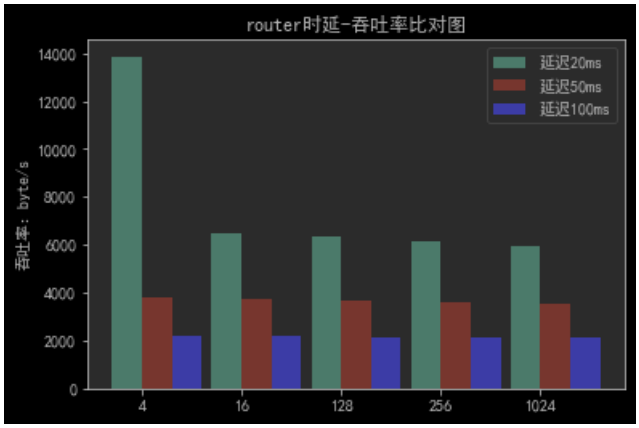

图 3.11: router 窗口大小-吞吐率变化图

3.2 实验分析

在本次实验中，我们观察到了一些和原先预想不太一致的情况，在做实验前，我的想法是随着滑动窗口的增加，传输时间会减少，并且吞吐率会增加，但实验得到的结果似乎相反，于是我针对这种情况进行了分析。

首先，我们观察到，当 router 没有转发时延和丢包率为 0 时，滑动窗口的大小对吞吐率和传输时延的影响不大，这也符合我们的预期，因为总是收到-确认不需要重传，所以窗口的大小不会很大程度上的影响传输性能

然后，我们发现，当 router 具有丢包的可能时，随着滑动窗口🄵增大转发的效率反而降低了，我的思考是这样的，根据 GBN 的特性，因为接收端每个时间都只能接受一个特定的数据包，所以随着滑动窗口的增长，接收端可能会接受很多大量的“无用的”的数据包，大量的时间被浪费在了不能被加收的数据包上，又由于 GBN 使用了多线程收发，所以会出现发线程大量抢夺资源的情况发生，从而出现大量的传输时延，但是如果使用 SR 算法，这样的情况会得到大幅度的缓解，这也是很好理解的，因为此时接收端能够接受的数据包个数增多，所以不会出现转发了一整个滑动窗口但是只能接受一个数据包的情况存在。

观察这一组实验结果，我们不难发现，丢包率从 0 增长到 5 时，当 router 的转发时延从 0ms 增长到 20ms 时，效率降低的最大，但是当我们逐步增大丢包率或者 router 转发时延时，效率虽然在逐步降低，但是降低的速率逐渐趋缓，这也是比较好理解的，这是因为从无到有的过程意味着一定会经历超时重传，所以没经过一个固定的时间就一定会经历超时重传，而从有增大的过程只是经历超时重传的频率增大，所以不会引起大规模的效率降低。

4 拥塞控制对传输性能的影响

4.1 实验数据

本次实验中，我们通过比较是否拥有拥塞控制对于传输性能的影响，我们在本次实验中还是通过控制变量法进行数据测量，没有拥塞控制时我们使用 GBN 实现的滑动窗口为 16 的数据进行对照组。

首先我们控制 router 的转发时延不变，改变 router 的丢包率进行数据测量。

下图展示了不同丢包率下有无拥塞控制对文件传输的影响，其中 router 的转发时延我们将它设置为 0ms，其中蓝色的折线是有拥塞控制时的传输时延变化折线，橘黄色的折线是没有拥塞控制时的时延折线

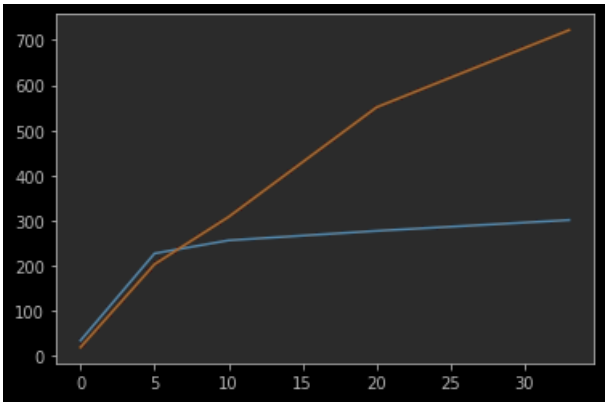

图 4.12: 丢包率-传输时延变化图

接着，我们可以根据上面的数据得到其吞吐率的变化并做出柱状对进行表示

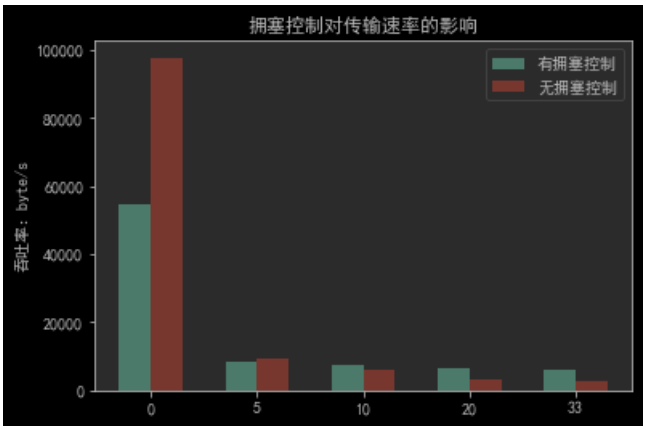

图 4.13: 丢包率-吞吐率变化图

接下来，我们通过固定 router 的丢包率为 0%，然后改变 router 的转发时延，观察文件在传输时的性能。其中蓝色的折线是有拥塞控制时的传输时延变化折线，橘黄色的折线是没有拥塞控制时的时延折线

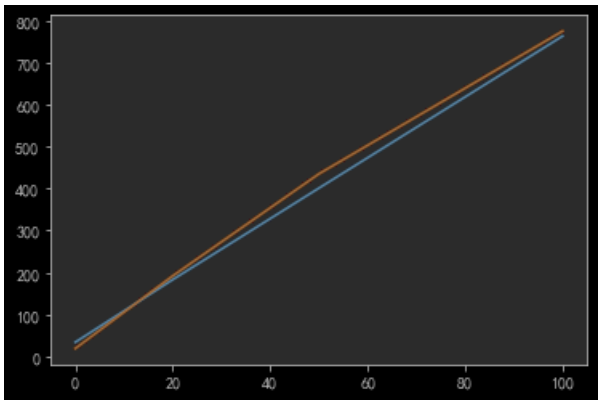

图 4.14: router 转发时延-传输时延变化图

然后通过上面的数据，我们可以计算得到最终的吞吐率，并绘制图像

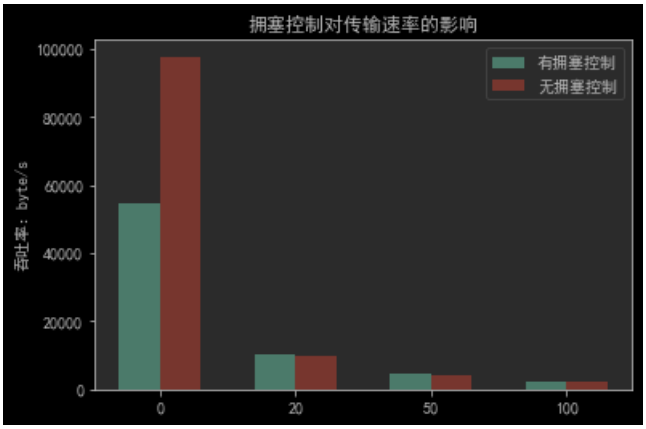

图 4.15: router 转发时延-吞吐率变化图

4.2 实验分析

有无拥塞控制的实验大体上是符合我们对实验结果的预期的，因为拥塞控制随着网络情况不断的调整滑动窗口的大小，让窗口能较长时间维持在一个合适的大小，这样的动态稳定会增加文件的传输速率，而当没有拥塞控制时固定的窗口大小很可能和网络的拥堵情况不适配导致传输效率降低。

5 总结

通过本次实验，我对本学期在计算机网络课程中所实现过的可靠 UDP 传输程序进行了完整复盘，并对其性能做了综合考量。让我加深了对停等机制、GBN、SR、滑动窗口、拥塞控制等概念的理解，为后续的学习奠定了良好的基础。

6 附录

部分实验数据：

	A	B	C	D	E	F	G	H	I	J
1	所用程序	丢包率	路由时延	重传时间	滑动窗口		文件大小	传送时间	吞吐率	
2	拥塞控制	0	0	100ms			1867352	34s	54628	
3	拥塞控制	5	0	100ms			1857352	227s	8182.17	
4	拥塞控制	10	0	100ms			1857352	256s	7255.28	
5	拥塞控制	20	0	100ms			1857352	277s	6705.24	
6	拥塞控制	33	0	100ms			1857352	301s	6170.6	
7										
8	GBN	0	0	100ms	4		1857352	16s	116085	
9	GBN	0	0	100ms	16		1857352	19s	97755.4	
10	GBN	0	0	100ms	128		1857352	17s	109256	
11	GBN	0	0	100ms	256		1857352	17s	109256	
12	GBN	0	0	100ms	1024		1857352	16s	116085	
13	GBN	5	0	100ms	4		1857352	134s	13860.8	
14	GBN	5	0	100ms	16		1857352	203s	14510.56	
15	GBN	5	0	100ms	128		1857352	277s	6705.24	
16	GBN	5	0	100ms	256		1857352	293s	6339.08	
17	GBN	5	0	100ms	1024		1857352	301s	6170.605	
18	GBN	10	0	100ms	4		1857352	210s	8844.54	
19	GBN	10	0	100ms	16		1857352	308s	6030.36	
20	GBN	10	0	100ms	128		1857352	344s	5399.27	
21	GBN	10	0	100ms	256		1857352	357s	5202.67	
22	GBN	10	0	100ms	1024		1857352	384s	4836.85	
23	GBN	20	0	100ms	16		1857352	551s	3370.88	
24	GBN	33	0	100ms	16		1857352	722s	2572.51	
25										
26	停等	0	0	100ms	2		1857253	18s	103186	
27	停等	5	0	100ms	2		1857253	180s	10318.6	
28	停等	10	0	100ms	2		1857253	284s	6539.98	
29	停等	20	0	100ms	2		1857253	472s	3935.07	
30	停等	33	0	100ms	2		1857253	842s	2205.88	

图 6.16: 部分实验数据

全部实验数据及绘图代码：https://github.com/wbf1015/computer_network/tree/main/