

基于UDP服务设计可靠传输协议并编程实现

1811507 文静静

基于UDP服务设计可靠传输协议并编程实现

一、实验内容

二、实验要求

三、实验结果

比较停等机制与滑动窗口机制

滑动窗口机制不同窗口对比

拥塞控制和无拥塞控制的性能比较

一、实验内容

基于给定的实验测试环境，通过改变延迟时间和丢包率，完成下面3组性能对比实验：

- 停等机制与滑动窗口机制性能对比；
- 滑动窗口机制中不同窗口大小对性能的影响；
- 有拥塞控制和无拥塞控制的性能比较。

二、实验要求

- 完成给定测试文件的传输，显示传输时间和平均吞吐率。
- 性能测试指标包括吞吐率和时延，给出图形结果并进行分析。

三、实验结果

比较停等机制与滑动窗口机制

- 停等机制

```

请输入本机IP地址: 127.0.0.1
请输入本机端口号: 9988
请输入服务端IP地址: 127.0.0.1
请输入服务端端口号: 8899
Create socket successfully!
Bind successfully!
shake hand successfully
start to send 1.jpg...
send 1.jpg successfully, total 1857353 bytes
start to send 2.jpg...
send 2.jpg successfully, total 5898505 bytes
start to send 3.jpg...
send 3.jpg successfully, total 11968994 bytes
start to send helloworld.txt...
send helloworld.txt successfully, total 1655808 bytes
wave hand successfully
sendtime:6.251 s
totallength:21380660 Bytes
吞吐率:27362.8 Kbps

```

D:\vsproject\UDP_Send\Debug\UDP_Send.exe (进程 21012)已退出, 代码为 0。
按任意键关闭此窗口. . .

Microsoft Visual Studio 调试控制台

```

请输入本机IP地址: 127.0.0.1
请输入本机端口号: 8899
请输入路由器IP地址: 127.0.0.1
请输入路由器端口号: 9988
Create socket successfully!
Bind successfully!
shake hand successfully
receive the file1 successfully
receive the file2 successfully
receive the file3 successfully
receive the file4 successfully
wave hand successfully

```

D:\vsproject\UDP_Receive\Debug\UDP_Receive.exe (进程 22520)已退出, 代码为 0。
按任意键关闭此窗口. . .

时延: 6.251 s

吞吐率: 27362.8 Kbps

- 滑动窗口机制(窗口大小以15为例)

```

请输入本机IP地址: 127.0.0.1
请输入本机端口号: 9988
请输入服务端IP地址: 127.0.0.1
请输入服务端端口号: 8899
Create socket successfully!
Bind successfully!
shake hand successfully
start to send data...
please choose the size of sliding window: 15
start to send 1.jpg...
send 1.jpg successfully, total 1857353 bytes
start to send 2.jpg...
send 2.jpg successfully, total 5898505 bytes
start to send 3.jpg...
send 3.jpg successfully, total 11968994 bytes
start to send helloworld.txt...
send helloworld.txt successfully, total 1655808 bytes
sendtime:2.682 s
totallength:21380660 Bytes
吞吐率:63775.2 Kbps
wave hand successfully

```

D:\vsproject\UDP_Client\Debug\UDP_Client.exe (进程 10820)已退出, 代码为 0。
按任意键关闭此窗口. . .

时延：2.682 s

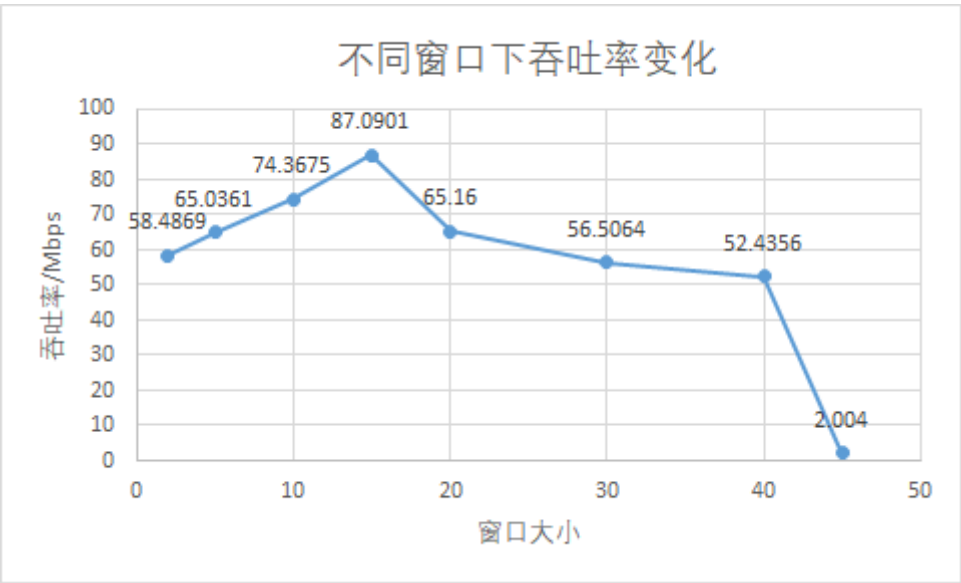
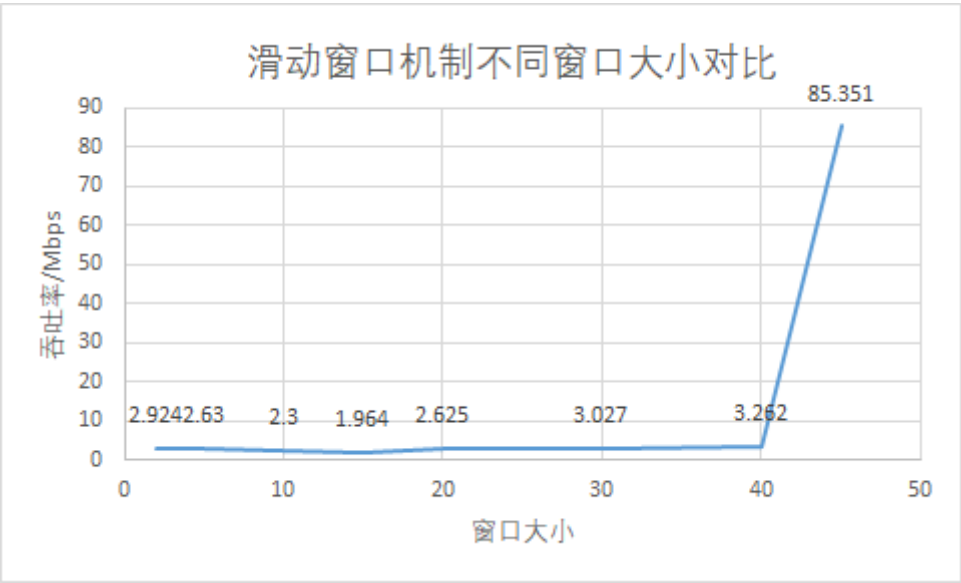
吞吐率：63775.2 Kpbs

比较分析：

由数据可知，滑动窗口一次能发送多个数据包，并且采用累积确认，有效利用网络带宽，吞吐率较停等机制，有显著提升。

滑动窗口机制不同窗口对比

窗口大小	2	5	10	15	20	30	40	45
时延(s)	2.924	2.63	2.3	1.964	2.625	3.027	3.262	85.351
吞吐率(Mpbs)	58.4869	65.0361	76.7635	87.0901	65.16	56.5064	52.4356	2.004



比较分析：

根据图表可以看出，当窗口逐渐增大时，发送端可以发送多个包，发包效率提升，吞吐率也增加，当到达窗口数15左右，吞吐率达到顶峰，之后继续增大窗口，容易造成网络堵塞，接收端来不及接收数据包，会造成丢包，延时，从而不断重复发包，吞吐率反而开始下降，窗口数45左右，吞吐率骤降。

拥塞控制和无拥塞控制的性能比较

- 无丢包，无延时

无拥塞控制：使用滑动窗口（窗口大小为15，性能接近最优）

```
请输入本机IP地址：127.0.0.1
请输入本机端口号：9988
请输入服务端IP地址：127.0.0.1
请输入服务端端口号：8899
Create socket successfully!
Bind successfully!
shake hand successfully
start to send data...
please choose the size of sliding window: 15
start to send 1.jpg...
send 1.jpg successfully, total 1857353 bytes
start to send 2.jpg...
send 2.jpg successfully, total 5898505 bytes
start to send 3.jpg...
send 3.jpg successfully, total 11968994 bytes
start to send helloworld.txt...
send helloworld.txt successfully, total 1655808 bytes
sendtime:2.682 s
totallength:21380660 Bytes
吞吐率:63775.2 Kbps
wave hand successfully

D:\vsproject\UDP_Client\Debug\UDP_Client.exe (进程 10820)已退出，代码为 0。
按任意键关闭此窗口. . .
```

拥塞控制：阈值为16

```
请输入本机IP地址：127.0.0.1
请输入本机端口号：9988
请输入路由器IP地址：127.0.0.1
请输入路由器端口号：8899
Create socket successfully!
Bind successfully!
shake hand successfully
start to send data...
start to send 1.jpg...
send 1.jpg successfully, total 1857353 bytes
start to send 2.jpg...
send 2.jpg successfully, total 5898505 bytes
start to send 3.jpg...
send 3.jpg successfully, total 11968994 bytes
start to send helloworld.txt...
send helloworld.txt successfully, total 1655808 bytes
sendtime:3.126 s
totallength:21380660 Bytes
吞吐率:54716.9 Kbps
wave hand successfully

D:\vsproject\UDPC\Debug\UDPC.exe (进程 18832)已退出，代码为 0。
按任意键关闭此窗口. . .
```

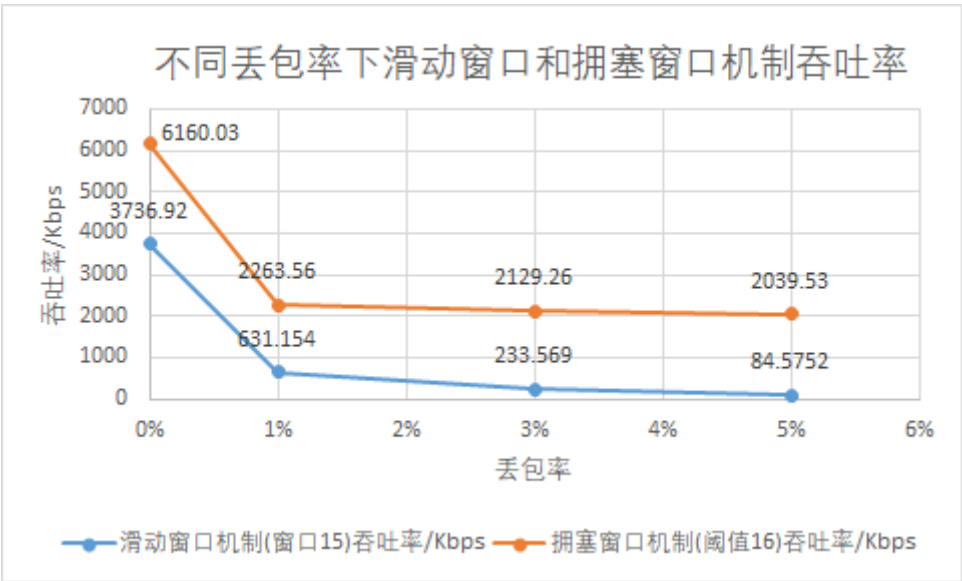
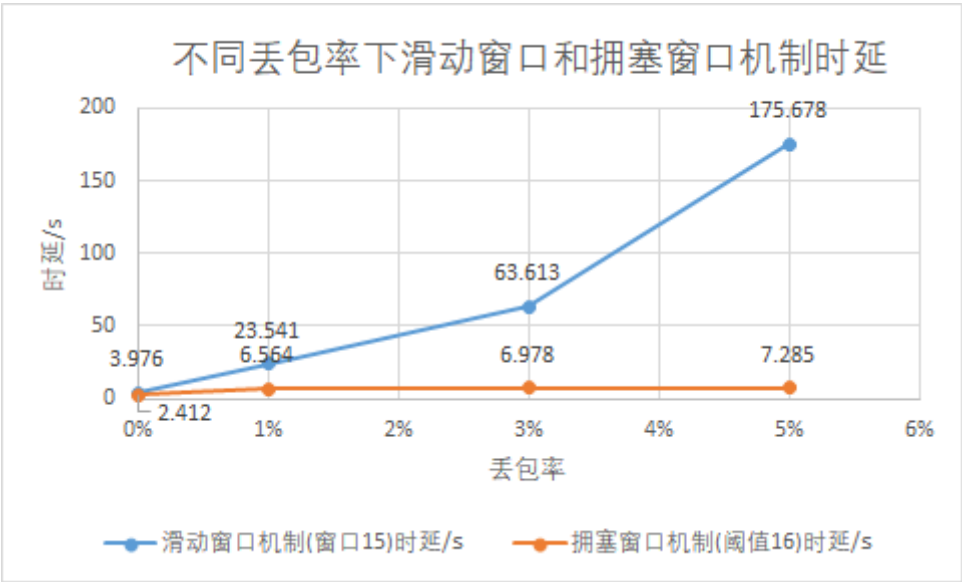
比较分析：

相比差别不大，拥塞控制情况下略慢，因为拥塞控制的阈值为16，窗口大小从1增加到16之后，窗口继续增加增速放慢，不同窗口下吞吐率不一样，而滑动窗口保持在15，吞吐率高。

• 不同丢包率，无延时

丢包率	0%	1%	3%	5%
滑动窗口机制(窗口15)时延/s	3.976	23.541	63.613	175.678
拥塞窗口机制(阈值16)时延/s	2.412	6.564	6.978	7.285

丢包率	0%	1%	3%	5%
滑动窗口机制(窗口15)吞吐率/Kbps	3736.92	631.154	233.569	84.5752
拥塞窗口机制(阈值16)吞吐率/Kbps	6160.03	2263.56	2129.26	2039.53



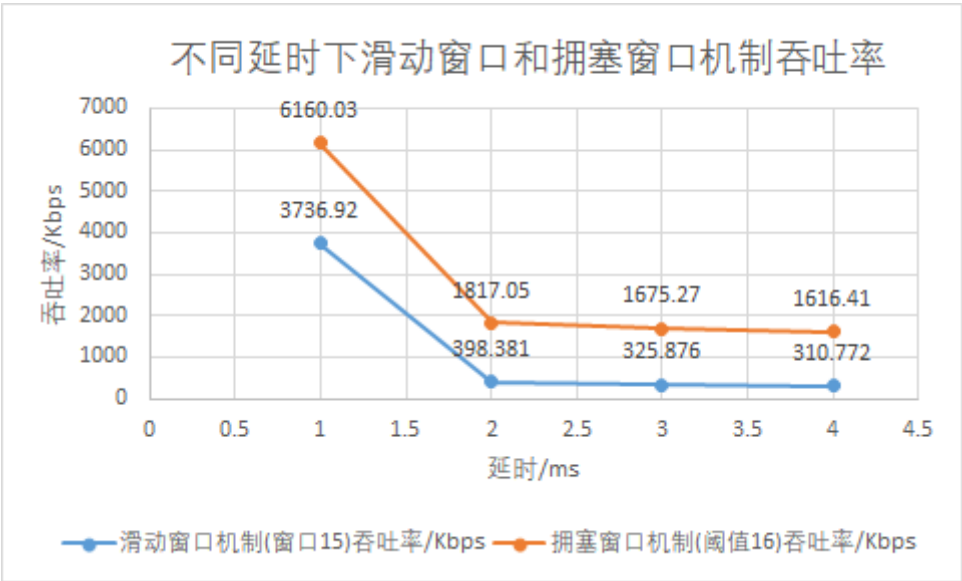
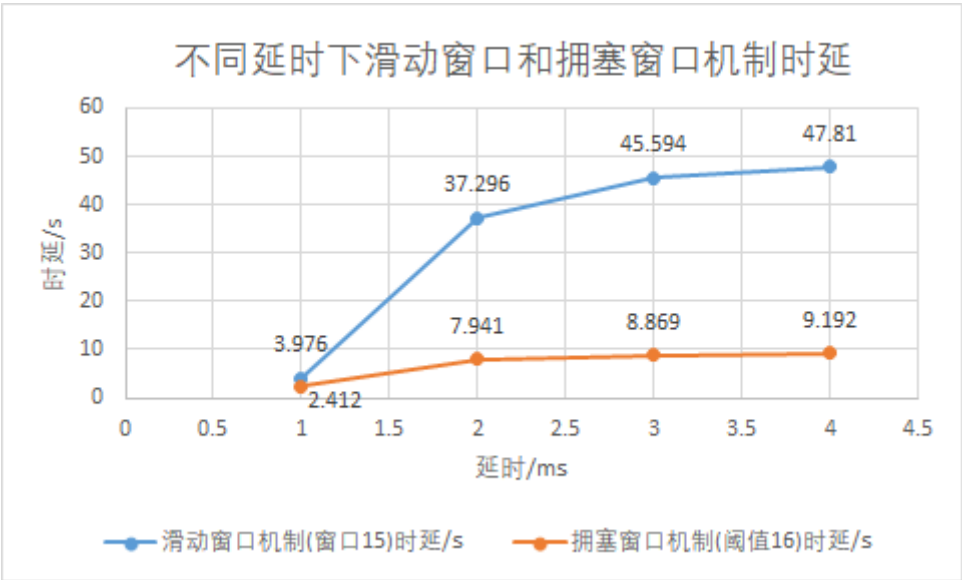
比较分析：

可以看出，当出现丢包的情况，滑动窗口只能等到延时才能重发数据包，丢包率越高，吞吐率越低；而拥塞窗口机制在收到三次重复ACK就会快速重传，或者超时的情况下重传数据包，因此丢包率变大对拥塞窗口机制的吞吐率影响不大。所以如果存在丢包的情况，拥塞窗口机制要明显优于滑动窗口机制。

• 无丢包，不同延时

延时	0ms	1ms	3ms	5ms
滑动窗口机制(窗口15)时延/s	3.976	37.296	45.594	47.81
拥塞窗口机制(阈值16)时延/s	2.412	7.941	8.869	9.192

延时	0ms	1ms	3ms	5ms
滑动窗口机制(窗口15)吞吐率/Kbps	3736.92	398.381	325.876	310.772
拥塞窗口机制(阈值16)吞吐率/Kbps	6160.03	1817.05	1675.27	1616.41



比较分析：

可以看出，当出现延时的情况，滑动窗口的窗口大小不变，每次都发相同数量的包，在延时的情况下，需要等待十分长的时间；而拥塞窗口机制在超时的情况下会调整窗口和阈值大小，将窗口重新设置为1，因此延时变长对拥塞窗口机制的吞吐率影响小于对滑动窗口机制的影响。所以如果存在延时的情况，拥塞窗口机制要明显优于滑动窗口机制。

- 丢包率5%，延时5ms

	时延/s	吞吐率/Kbps
滑动窗口机制(窗口15)	418.12	35.5353
拥塞窗口机制(阈值16)	14.47	1026.8

比较分析：

由上表数据可知，当出现丢包或者延时时，拥塞窗口机制能快速调节窗口大小和状态，重传数据包，明显由于无拥塞窗口的情况。