

EE 542 - 实验室作业 #4：快速、可靠的文件传输（团队）

指导老师Young H. Cho

提交日期：9月21日（报告）晚上11:59；9月23日（在Youtube上播放演示视频）

在本实验室中，您将探索 TCP 在不完美条件下的实际运行情况。你们的目标是开发一个使用自定义协议的文件传输程序（如 scp 或 ftp），该程序的性能要优于现有的任何其他程序。这是你们的第一个竞争性实验室。成绩最高的前 25% 小组将获得额外学分。程序应基于 Linux 操作系统，并使用 C 或 C++ 进行编码。

虽然 TCP 在互联网上是有效、可靠和相对稳健的，但它并不总是在任何情况下都能提供最佳吞吐量。在本节中，你们将设计一个基于 IP 的文件传输实用程序。该工具的设计和实现由你们小组自行决定，但必须完全满足三个要求：必须使用 IP（因此可以路由）、必须可靠地传输文件（没有错误）、必须使用类似于 scp 的命令行界面。

发送方和接收方的链路速度必须为 100Mbps，测试文件大小必须至少为 **1GB**。应使用延迟节点模拟链路的延迟和损耗率。您应在各种不同条件下测试系统。但是，您必须在作业中暴露系统的三种设置：

- 路由器的延迟（RTT）为 10 毫秒，损失率为 1%（双向），服务器、客户端和路由器的链路速度均为 100Mbps/秒。（情况 - 1）
- 路由器的延迟（RTT）为 200 毫秒，损耗率为 20%（双向），服务器、客户端和路由器的链路速度均为 100Mbps/秒。（情况 2）
- 将链路速度设为 80 Mbps/sec，延迟 (RTT) 设为 200 毫秒，路由器和服务器不掉线，客户端链路速度设为 100 Mbps/sec。（情况 3）

您应该在 MTU 为 1500 时运行上述测试，然后再在 MTU 为 9000 时运行测试。

为了进行演示和比赛，您需要创建一个实验，使用 vyos 路由器虚拟机连接两个节点（虚拟机）（增加 200ms 的 RTT 延迟和 20% 或 1% 的丢包率）。因此，根据上述情况，ping 中的往返总计应为 200 毫秒，数据包丢弃应发生在 Tx 和 Rx 中，Tx 丢弃率为 20%，Rx 丢弃率为 20%。

在 Virtualbox 中创建一个由服务器、客户端和 Vyos 虚拟机组成的简单拓扑结构，就像在 lab1 中做的那样（1 个服务器虚拟机 \Leftrightarrow 1 个 Vyos 虚拟机 \Leftrightarrow 1 个客户端虚拟机），并先在这里运行你的代码。**这样做是为了避免 AWS 费用，因为如果同时使用 3 个虚拟机，AWS 免费层会受到一些限制。在 VirtualBox 中一切正常后，在 AWS ec2 实例中编译相同的代码，并尝试文档中提到的所有情况，修复在 AWS 环境中遇到的代码错误。**

情况 1、2：（记录传输 1 千兆字节文件时的 ftp 代码性能）

在服务器和客户端虚拟机上执行以下命令，设置服务器和客户端虚拟机的速率限制，将出口速度限制为 100Mbit/s。这里假定客户端的 eth0 属于 10.200.1.0/24 子网，服务器的 eth0 属于 10.200.2.0/24 子网。

出口速率限制

```
# sudo tc qdisc add dev eth0 root tbf rate 100mbit latency 0.001ms burst 9015
```

现在执行以下命令，在两个网络接口的路由器出口路径上将速率限制设为 100Mbit/sec。这里假定 eth0 属于 10.200.1.0/24 子网，eth1 属于 10.200.2.0/24 子网。

出口速率限制

```
# sudo tc qdisc add dev eth0 root handle 1:0 tbf rate 100mbit latency 0.001ms burst 9015
# sudo tc qdisc add dev eth1 root handle 1:0 tbf rate 100mbit latency 0.001ms burst 9015
```

首先执行 iperf，确保速率限制正常。您需要获得大约 100 Mbits/sec 的 tcp 吞吐量。同时检查 udp 模式的带宽是否为 100Mbits/sec 左右（iperf -u -c <dst_ip> -b 200mbit）。

```
ubuntu@ip-10-200-2-48: ~
ubuntu@ip-10-200-2-48:~$ iperf -c 10.200.1.83
-----
Client connecting to 10.200.1.83, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[  3] local 10.200.2.48 port 33652 connected with 10.200.1.83 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3]  0.0-10.0 sec   112 MBytes  93.5 Mbits/sec
ubuntu@ip-10-200-2-48:~$

ubuntu@ip-10-200-2-48:~$ iperf -u -c 10.200.1.83 -b 200mbit
-----
Client connecting to 10.200.1.83, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
-----
[  3] local 10.200.2.48 port 44832 connected with 10.200.1.83 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3]  0.0-10.0 sec   242 MBytes  203 Mbits/sec
[  3] Sent 172412 datagrams
[  3] Server Report:
[  3]  0.0-10.3 sec   115 MBytes  94.4 Mbits/sec  15.668 ms 90103/172411 (52%)
[  3]  0.0-10.3 sec    1 datagrams received out-of-order
ubuntu@ip-10-200-2-48:~$
```

现在，在 Vyos VM（路由器）的 eth0 和 eth1 网络接口上执行以下命令，设置 RTT 延迟为 10 毫秒，tx 和 rx 损失率均为 1%。这里 eth0 为 10.200.1.0/24，eth1 为 10.200.2.0/24，反之亦然。下面的命令模拟了链路上高传播延迟时的数据包丢弃情况。

RTT 为 10 毫秒时，降幅为 1%：

```
# sudo tc qdisc add dev eth0 parent 1:1 handle 10: netem delay 5ms drop 1%
# sudo tc qdisc add dev eth1 parent 1:1 handle 10: netem delay 5ms drop 1%
```

RTT 为 200 毫秒时，下降 20%：

```
# sudo tc qdisc change dev eth0 parent 1:1 handle 10: netem delay 100ms drop 20%
# sudo tc qdisc change dev eth1 parent 1:1 handle 10: netem delay 100ms drop 20%
```

来自 ping 命令：(在服务器上执行以 ping 到客户端，在客户端上执行以 ping 到服务器)

```
Ping -i 0.2 -c 200 <dest_ip>
```

确保在 ping 输出中看到路由器配置的延迟和掉线率，掉线率在设定值的 $\pm 20\%$ 范围内，延迟值几乎略大于设定值。下面给出了一张快照作为参考。(这是案例 2)

```
64 bytes from 10.200.1.83: icmp_seq=194 ttl=63 time=201 ms
64 bytes from 10.200.1.83: icmp_seq=196 ttl=63 time=201 ms
64 bytes from 10.200.1.83: icmp_seq=197 ttl=63 time=200 ms
64 bytes from 10.200.1.83: icmp_seq=199 ttl=63 time=201 ms

--- 10.200.1.83 ping statistics ---
200 packets transmitted, 126 received, 37% packet loss, time 40450ms
rtt min/avg/max/mdev = 200.906/201.086/202.648/0.319 ms, pipe 2
ubuntu@ip-10-200-2-48:~$
```

您还需要使用 iperf 确认两个方向上 udp 模式的数据包丢包率（在客户端虚拟机上启动服务器 iperf 并运行它，然后在服务器虚拟机上启动并运行它）（`iperf -u -c <dst_ip> -b 100mbit`），允许丢包率有 $\pm 20\%$ 的变化。（这是针对情况 2）。您应该看到大约 20% 的下降率，该值的变化幅度为 $\pm 20\%$ 。

服务器 => 客户端：

```
ubuntu@ip-10-200-2-48:~$ iperf -u -c 10.200.1.83 -b 100mbit
-----
Client connecting to 10.200.1.83, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[  3] local 10.200.2.48 port 41704 connected with 10.200.1.83 port 5001
[ ID] Interval           Transfer     Bandwidth
[  3] 0.0-10.0 sec      120 MBytes  101 Mbits/sec
[  3] Sent 85471 datagrams
[  3] Server Report:
[  3] 0.0-10.3 sec      92.8 MBytes  75.9 Mbits/sec  15.723 ms 19307/85470 (23%)
[  3] 0.0-10.3 sec      1 datagrams received out-of-order
ubuntu@ip-10-200-2-48:~$
```

客户端 => 服务器：

```
ubuntu@ip-10-200-1-83:~$ iperf -u -c 10.200.2.48 -b 100mbit
-----
Client connecting to 10.200.2.48, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[  3] local 10.200.1.83 port 48437 connected with 10.200.2.48 port 5001
[ ID] Interval           Transfer     Bandwidth
[  3] 0.0-10.0 sec      120 MBytes  101 Mbits/sec
[  3] Sent 85470 datagrams
[  3] Server Report:
[  3] 0.0-10.0 sec      92.6 MBytes  77.7 Mbits/sec   0.098 ms 19414/85469 (23%)
[  3] 0.0-10.0 sec      1 datagrams received out-of-order
ubuntu@ip-10-200-1-83:~$
```

TCP iperf:

```
ubuntu@ip-10-200-2-48:~$ iperf -c 10.200.1.83
-----
Client connecting to 10.200.1.83, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[  3] local 10.200.2.48 port 33690 connected with 10.200.1.83 port 5001
[ ID] Interval           Transfer     Bandwidth
[  3] 0.0-23.9 sec      384 KBytes  132 Kbits/sec
ubuntu@ip-10-200-2-48:~$
```

您还需要在发送方计算机中创建一个 1GB 的文件。您可以按照 scp 部分 Lab3 中提到的步骤创建

。

```
# dd if=/dev/urandom of=data.bin bs=1M count=1024
```

案例 3：（记录传输 1 千兆字节文件时的 ftp 代码性能）

删除所有虚拟机 qdisc 上先前设置的配置（使用 `sudo tc qdisc show` 进行验证）。现在将服务器和客户端虚拟机的速率限制设为 100Mbps/sec。在路由器虚拟机上将速率限制设为 80Mbps/sec，RTT 为 200ms。在此步骤中没有配置丢弃。确保在 iperf 中进行 udp 测试时，ping 显示延迟为 200 毫秒，速率限制接近 80Mbps/sec，ping 中看不到丢包。

路由器 VM 命令：

出口速率限制

```
# sudo tc qdisc add dev eth0 root handle 1:0 tbf rate 80mbit latency 0.001ms burst 9015
# sudo tc qdisc add dev eth1 root handle 1:0 tbf rate 80mbit latency 0.001ms burst 9015
```

在 tx、rx 中添加出口延迟：

```
# sudo tc qdisc add dev eth0 parent 1:1 handle 10: netem delay 100ms
# sudo tc qdisc add dev eth1 parent 1:1 handle 10: netem delay 100ms
```

服务器、客户端虚拟机出口速率限制命令：

```
# sudo tc qdisc add dev eth0 root tbf rate 100mbit latency 0.001ms burst 9015
```

```
ubuntu@ip-10-200-2-48:~$ iperf -u -c 10.200.1.83 -b 100mbit
-----
Client connecting to 10.200.1.83, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[  3] local 10.200.2.48 port 46053 connected with 10.200.1.83 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-10.0 sec  120 MBytes  101 Mbits/sec
[  3] Sent 85471 datagrams
[  3] Server Report:
[  3] 0.0-10.1 sec  93.2 MBytes  77.8 Mbits/sec   0.111 ms 18984/85469 (22%)
[  3] 0.0-10.1 sec  1 datagrams received out-of-order
ubuntu@ip-10-200-2-48:~$
```

```
ubuntu@ip-10-200-2-48: ~
ubuntu@ip-10-200-2-48:~$ iperf -c 10.200.1.83
-----
Client connecting to 10.200.1.83, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[  3] local 10.200.2.48 port 33688 connected with 10.200.1.83 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-10.9 sec   3.38 MBytes  2.60 Mbits/sec
ubuntu@ip-10-200-2-48:~$
```

批判性思维

正如您在案例 2 和案例 3 中观察到的，iperf UDP 的吞吐量相似（约 78Mbps），但 TCP 的吞吐量最多相差 20 倍。请考虑您的配置，并在报告中准确描述案例 2 和案例 3 在网络缓冲、数据包丢失

、延迟和窗口大小方面可能发生的情况。

定制快速可靠的协议设计

您的程序需要在**情况 1、2、3**下，以尽可能高的性能通过这个**不良网络链路**将整个文件从发送方可靠地传输到接收方。还要确保您的 ftp 应用程序能在不同 mtu 的接口上运行。**先在 mtu 为 1500 时进行测试，然后在 mtu 为 9001 时进行测试。**属于子网 10.200.1.0/24、10.200.2.0/24 的虚拟机的所有网络接口的 mtu 必须设置为相同值。(即服务器、客户端和 vyos 虚拟机)。客户端 ftp 应用程序应将文件保存到命令行中指定的位置。同样，ftp 应用程序的服务器部分应发送命令行中指定的文件。

您必须在数据的第一比特离开发送方和最后比特被接收方接收时打上时间戳。文件单程传输的总时间将用于比较您的程序与其他团队程序的性能。您还需要在**原始文件和接收文件上运行 MD5**，以证明系统的可靠性。您需要在上述三种情况下，在两个指定的掉线率、延迟和速率限制条件下，运行您的 mtu 分别为 1500 和 9001 的 ftp 应用程序。(重复同样的程序，先运行 mtu 为 1500 的程序，再运行 mtu 为 9001 的程序，观察文件传输时间)。

该系统的执行应简单明了，任何作弊行为都将导致在比赛中自动落败，并可能导致实验室得分为零。这项任务的目的是鼓励为每个人营造一个健康的竞争发展环境。我们鼓励大家互相帮助，以取得最佳成绩。不过，最后吞吐量最高的两个小组将获得额外奖励。

- 1) 决赛 FTP 的最低传输速率为 20 Mbps，以便与其他团队竞争。
- 2) 在报告文件中详细描述文件传输工具背后的概念、结果、流程图、数据结构和所用算法以及分析，报告文件必须与源文件一起在黑板指定的到期日晚上 11:59 之前提交。
- 3) 评分将以提交的报告幻灯片和演示为依据

参考资料

<https://computing.llnl.gov/tutorials/pthreads/> (多线程)

<https://www.geeksforgeeks.org/data-structures/> (数据结构)

<https://www.geeksforgeeks.org/fundamentals-of-algorithms/> (算法)