# Report for Assignment 1

Xianghang Mi
xmi@iu.edu

## 1. Abstract

In this project, I implemented several http servers and clients with different features, such as persistent connection and multiple threads. Detailed description of those implementations is in Section 2, where you can also find out how to use those servers and clients. To test performance, availability and difference of those servers and clients, I did several experiments whose results are listed in Section 3.

## 2. Implementation and Use

### 2.1. Implementation

I implemented three different http server and two http clients, the difference of them is listed in Figure 1 as below.

| Name | Type | Description | Multi - thread | Persistent connection | Underlying Protocol |
|---|---|---|---|---|---|
| mttserver | http server | Multi-thread TCP-based server | yes | yes | TCP |
| tcpserver | http server | TCP-based server | no | yes | TCP |
| udpserver | http server | UDP-based server | no | no | UDP |
| tcpclient | http client | TCP-based server | no | yes | TCP |
| udpclient | http client | UDP-based client | no | no | UDP |

*Figure 1 list of HTTP servers and clients*

### 2.2. How to use them

1. Compile them

In the project root directory, run "./run", which will compile the source code, after finishing it,
Go to "bin" directory where you can find compiled executables: mttserver, tcpserver, udpserver, tcpclient, udpclient.

2. Run them

From the README file in the root directory, you can find instructions of how to run those executables.

## 3. Experiments

### 3.1. Compare the performance of persistent TCP connection and non-persistent TCP connection

Just as showed in Figure 2 and Figure 3, when we use TCP persistent connection to transfer multiple 1MB files, the average time cost for each file is 20,000 microseconds less than using no-persistent connection

| Client Type | Server Type | Persistent | File Size(Bytes) | File count | overall time cost | time cost per file |
|---|---|---|---|---|---|---|
| TCP | TCP Server | p | 1,094,328 | 10 | 1,143,924 | 114,392 |
| TCP | TCP Server | p | 1,094,328 | 10 | 1,156,637 | 115,664 |
| TCP | TCP Server | p | 1,094,328 | 10 | 1,186,649 | 118,665 |
| TCP | TCP Server | p | 1,094,328 | 10 | 1,163,795 | 116,380 |
| TCP | TCP Server | p | 1,094,328 | 10 | 1,165,122 | 116,512 |
| TCP | TCP Server | p | 1,094,328 | 10 | 1,159,622 | 115,962 |
| TCP | TCP Server | p | 1,094,328 | 10 | 1,166,134 | 116,613 |
| TCP | TCP Server | p | 1,094,328 | 10 | 1,132,527 | 113,253 |

| | | | | | | |
|---|---|---|---|---|---|---|
| TCP | TCP Server | p | 1,094,328 | 10 | 1,140,322 | 114,032 |
| | | | | 90 | 10,414,732 | 115,719 |

*Figure 2. persistent TCP-based file transfer*

| Client Type | Server Type | Persistent | File Size(Bytes) | File count | overall time cost | timecost per file |
|---|---|---|---|---|---|---|
| TCP | TCP Server | no | 1,094,328 | 1 | 142,493 | 142,493 |
| TCP | TCP Server | no | 1,094,328 | 1 | 130,789 | 130,789 |
| TCP | TCP Server | no | 1,094,328 | 1 | 119,389 | 119,389 |
| TCP | TCP Server | no | 1,094,328 | 1 | 140,711 | 140,711 |
| TCP | TCP Server | no | 1,094,328 | 1 | 109,982 | 109,982 |
| TCP | TCP Server | no | 1,094,328 | 1 | 139,809 | 139,809 |
| TCP | TCP Server | no | 1,094,328 | 1 | 152,750 | 152,750 |
| TCP | TCP Server | no | 1,094,328 | 1 | 138,732 | 138,732 |
| TCP | TCP Server | no | 1,094,328 | 1 | 154,595 | 154,595 |
| | | | | 9 | 1,229,250 | 136,583 |

*Figure 3. no-persistent TCP-based file tranfer*

## 3.2. UDP-based Http server and client

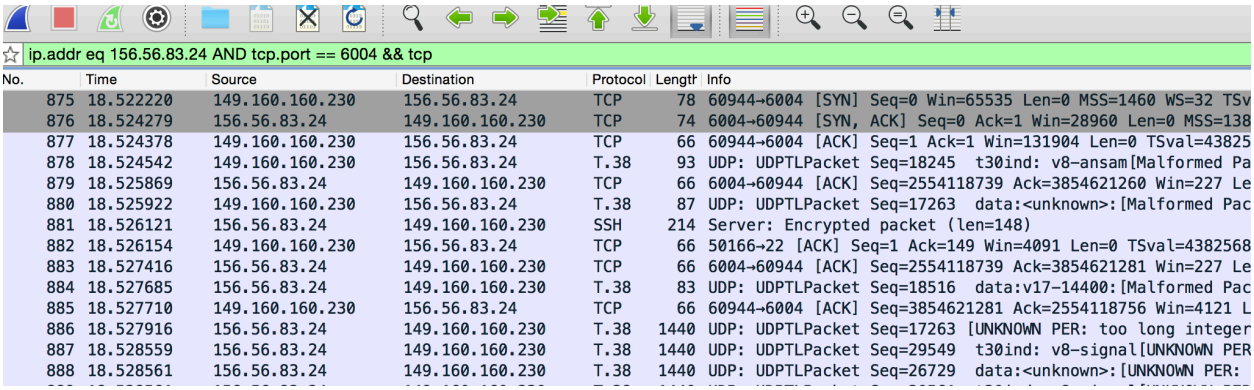
1. Compare data in Figure 2, 3, 4, we can conclude that without overhead of connection establishment, UDP can larges improve the performance of transferring files of same size.
2. However, from Figure 4, we can see data loss of UDP, So comparing with TCP, UDP can provide improved performance at overhead of data loss.

| Client Type | Server Type | Persistent | File Size(Bytes) | File count | time cost | recved bytes | loss rate |
|---|---|---|---|---|---|---|---|
| UDP | UDP | no | 1,094,328 | 1 | 75,581 | 328,281 | 0.30 |
| UDP | UDP | no | 1,094,328 | 1 | 97,299 | 924,883 | 0.85 |
| UDP | UDP | no | 1,094,328 | 1 | 90,642 | 1,040,306 | 0.95 |
| UDP | UDP | no | 1,094,328 | 1 | 102,120 | 1,094,328 | 1.00 |
| UDP | UDP | no | 1,094,328 | 1 | 100,482 | 972,909 | 0.89 |
| UDP | UDP | no | 1,094,328 | 1 | 89,653 | 1,094,328 | 1.00 |
| UDP | UDP | no | 1,094,328 | 1 | 69,602 | 415,223 | 0.38 |
| UDP | UDP | no | 1,094,328 | 1 | 92,302 | 1,076,340 | 0.98 |
| UDP | UDP | no | 1,094,328 | 1 | 83,697 | 944,428 | 0.86 |
| UDP | UDP | no | 1,094,328 | 1 | 109,917 | 881,470 | 0.81 |
| | | | 1,094,328 | | 91,130 | 877,250 | 0.80 |

*Figure 4 UDP-based file transfer*

## 3.3. Wireshark packets profile

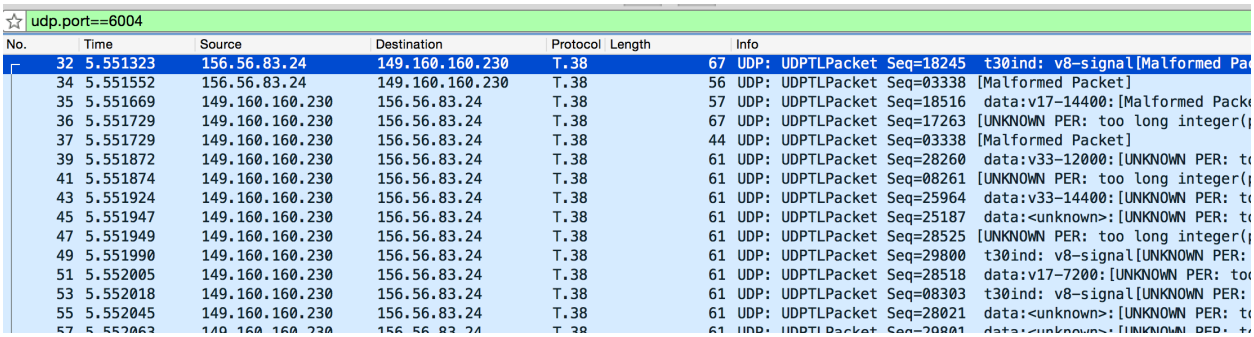
From Figure 5 and 6, we can see the TCP connection between client and server, UDP packets between UDP-based client and server.



ip.addr eq 156.56.83.24 AND tcp.port == 6004 && tcp

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 875 | 18.522220 | 149.160.160.230 | 156.56.83.24 | TCP | 78 | 60944→6004 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSv |
| 876 | 18.524279 | 156.56.83.24 | 149.160.160.230 | TCP | 74 | 6004→60944 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=138 |
| 877 | 18.524378 | 149.160.160.230 | 156.56.83.24 | TCP | 66 | 60944→6004 [ACK] Seq=1 Ack=1 Win=131904 Len=0 TSval=43825 |
| 878 | 18.524542 | 149.160.160.230 | 156.56.83.24 | T.38 | 93 | UDP: UDPTLPacket Seq=18245 t30ind: v8-ansam[Malformed Pa |
| 879 | 18.525869 | 156.56.83.24 | 149.160.160.230 | TCP | 66 | 6004→60944 [ACK] Seq=2554118739 Ack=3854621260 Win=227 Le |
| 880 | 18.525922 | 149.160.160.230 | 156.56.83.24 | T.38 | 87 | UDP: UDPTLPacket Seq=17263 data:<unknown>:[Malformed Pac |
| 881 | 18.526121 | 156.56.83.24 | 149.160.160.230 | SSH | 214 | Server: Encrypted packet (len=148) |
| 882 | 18.526154 | 149.160.160.230 | 156.56.83.24 | TCP | 66 | 50166→22 [ACK] Seq=1 Ack=149 Win=4091 Len=0 TSval=4382568 |
| 883 | 18.527416 | 156.56.83.24 | 149.160.160.230 | TCP | 66 | 6004→60944 [ACK] Seq=2554118739 Ack=3854621281 Win=227 Le |
| 884 | 18.527685 | 156.56.83.24 | 149.160.160.230 | T.38 | 83 | UDP: UDPTLPacket Seq=18516 data:v17-14400:[Malformed Pac |
| 885 | 18.527710 | 149.160.160.230 | 156.56.83.24 | TCP | 66 | 60944→6004 [ACK] Seq=3854621281 Ack=2554118756 Win=4121 L |
| 886 | 18.527916 | 156.56.83.24 | 149.160.160.230 | T.38 | 1440 | UDP: UDPTLPacket Seq=17263 [UNKNOWN PER: too long integer |
| 887 | 18.528559 | 156.56.83.24 | 149.160.160.230 | T.38 | 1440 | UDP: UDPTLPacket Seq=29549 t30ind: v8-signal[UNKNOWN PER |
| 888 | 18.528561 | 156.56.83.24 | 149.160.160.230 | T.38 | 1440 | UDP: UDPTLPacket Seq=26729 data:<unknown>:[UNKNOWN PER: |

*Figure 5. TCP packets between TCP server and client*



udp.port==6004

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 32 | 5.551323 | 156.56.83.24 | 149.160.160.230 | T.38 | 67 | UDP: UDPTLPacket Seq=18245 t30ind: v8-signal[Malformed Pa |
| 34 | 5.551552 | 156.56.83.24 | 149.160.160.230 | T.38 | 56 | UDP: UDPTLPacket Seq=03338 [Malformed Packet] |
| 35 | 5.551669 | 149.160.160.230 | 156.56.83.24 | T.38 | 57 | UDP: UDPTLPacket Seq=18516 data:v17-14400:[Malformed Packe |
| 36 | 5.551729 | 149.160.160.230 | 156.56.83.24 | T.38 | 67 | UDP: UDPTLPacket Seq=17263 [UNKNOWN PER: too long integer( |
| 37 | 5.551729 | 149.160.160.230 | 156.56.83.24 | T.38 | 44 | UDP: UDPTLPacket Seq=03338 [Malformed Packet] |
| 39 | 5.551872 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=28260 data:v33-12000:[UNKNOWN PER: to |
| 41 | 5.551874 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=08261 [UNKNOWN PER: too long integer( |
| 43 | 5.551924 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=25964 data:v33-14400:[UNKNOWN PER: to |
| 45 | 5.551947 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=25187 data:<unknown>:[UNKNOWN PER: to |
| 47 | 5.551949 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=28525 [UNKNOWN PER: too long integer( |
| 49 | 5.551990 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=29800 t30ind: v8-signal[UNKNOWN PER: |
| 51 | 5.552005 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=28518 data:v17-7200:[UNKNOWN PER: too |
| 53 | 5.552018 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=08303 t30ind: v8-signal[UNKNOWN PER: |
| 55 | 5.552045 | 149.160.160.230 | 156.56.83.24 | T.38 | 61 | UDP: UDPTLPacket Seq=28021 data:<unknown>:[UNKNOWN PER: to |

*Figure 6. UDP packets between UDP server and client*