Project 4 Report
ECE 5600

Nathan Tipton
A01207112
Partner: Erik Sargent

Dec 1, 2017

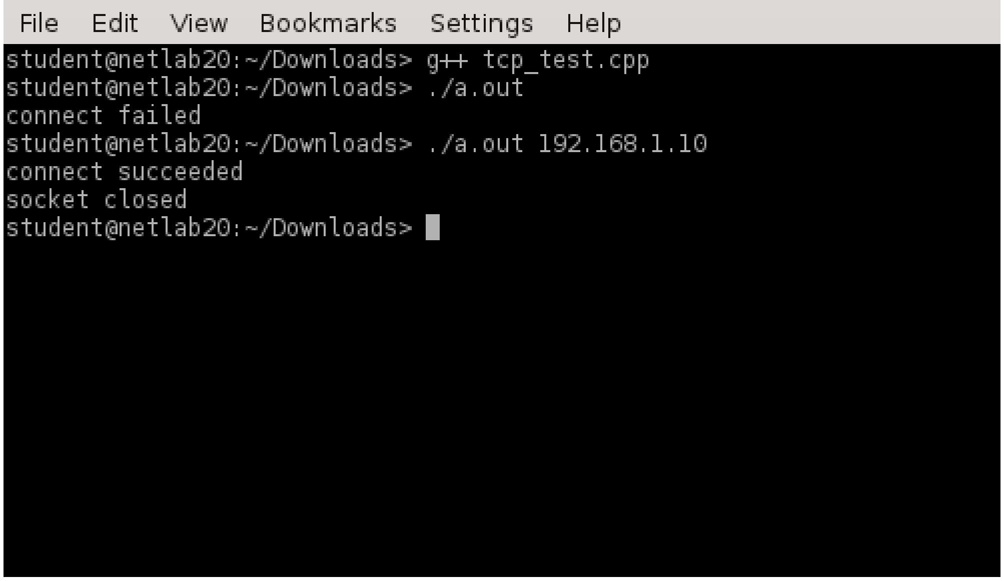
1 Objective

The purpose of this project is to familiarize ourselves with the Transport Control Protocol (TCP). We will also implement a TCP transmission.

2 Results

We implemented the server application of a TCP connection.

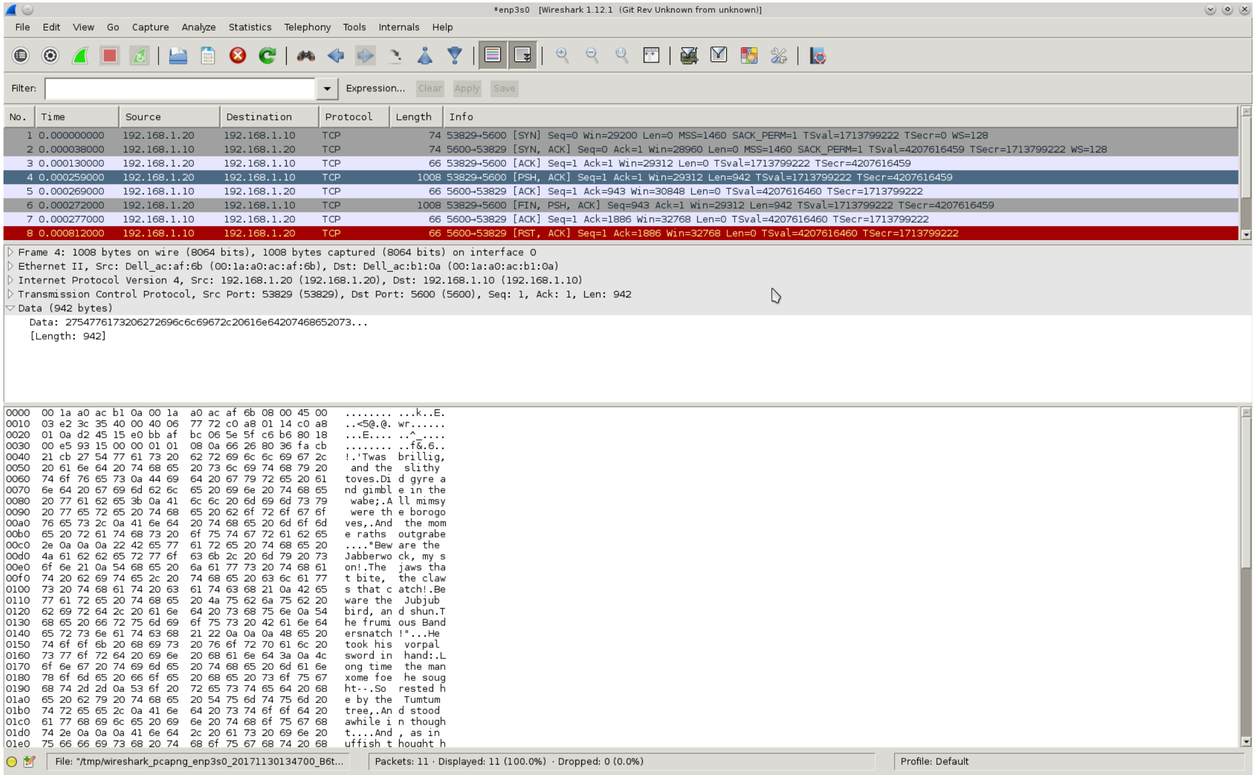
Using the provided TCP test code we send data to the our server implementation. Figure 1 shows that a connect was successfully made.



```
student@netlab20:~/Downloads> g++ tcp_test.cpp
student@netlab20:~/Downloads> ./a.out
connect failed
student@netlab20:~/Downloads> ./a.out 192.168.1.10
connect succeeded
socket closed
student@netlab20:~/Downloads> 
```

Figure 1: TCP test

With Wireshark we are able to see the handshake between the client and server. We are also able to see the data transferred. This is shown in figure 2.


*enp3s0 [Wireshark 1.12.1 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression...
Clear
Apply
Save

```
No.  Time         Source         Destination    Protocol  Length  Info
   1 0.000000000  192.168.1.20   192.168.1.10   TCP           74 53829→5600 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1713799222 TSecr=0 WS=128
   2 0.000038000  192.168.1.10   192.168.1.20   TCP           74 5600→53829 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=4207616459 TSecr=1713799222 WS=128
   3 0.000130000  192.168.1.20   192.168.1.10   TCP           66 53829→5600 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1713799222 TSecr=4207616459
   4 0.000259000  192.168.1.20   192.168.1.10   TCP         1008 53829→5600 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=942 TSval=1713799222 TSecr=4207616459
   5 0.000269000  192.168.1.10   192.168.1.20   TCP           66 5600→53829 [ACK] Seq=1 Ack=943 Win=30848 Len=0 TSval=4207616460 TSecr=1713799222
   6 0.000272000  192.168.1.20   192.168.1.10   TCP         1008 53829→5600 [FIN, PSH, ACK] Seq=943 Ack=1 Win=29312 Len=942 TSval=1713799222 TSecr=4207616459
   7 0.000277000  192.168.1.10   192.168.1.20   TCP           66 5600→53829 [ACK] Seq=1 Ack=1886 Win=32768 Len=0 TSval=4207616460 TSecr=1713799222
   8 0.000812000  192.168.1.10   192.168.1.20   TCP           66 5600→53829 [RST, ACK] Seq=1 Ack=1886 Win=32768 Len=0 TSval=4207616460 TSecr=1713799222

▷ Frame 4: 1008 bytes on wire (8064 bits), 1008 bytes captured (8064 bits) on interface 0
▷ Ethernet II, Src: Dell_ac:af:6b (00:1a:a0:ac:af:6b), Dst: Dell_ac:b1:0a (00:1a:a0:ac:b1:0a)
▷ Internet Protocol Version 4, Src: 192.168.1.20 (192.168.1.20), Dst: 192.168.1.10 (192.168.1.10)
▷ Transmission Control Protocol, Src Port: 53829 (53829), Dst Port: 5600 (5600), Seq: 1, Ack: 1, Len: 942
▽ Data (942 bytes)
    Data: 2754776173206272696c6c69672c20616e64207468652073...
    [Length: 942]

0000  00 1a a0 ac b1 0a 00 1a  a0 ac af 6b 08 00 45 00   ........ ...k..E.
0010  03 e2 3c 35 40 00 40 06  77 72 c0 a8 01 14 c0 a8   ..<5@.@. wr......
0020  01 0a d2 45 15 e0 bb af  bc 06 5e 5f c6 b6 80 18   ...E.... ..^_....
0030  00 e5 93 15 00 00 01 01  08 0a 66 26 80 36 fa cb   ........ ..f&.6..
0040  21 cb 27 54 77 61 73 20  62 72 69 6c 6c 69 67 2c   !.'Twas  brillig,
0050  20 61 6e 64 20 74 68 65  20 73 6c 69 74 68 79 20    and the  slithy 
0060  74 6f 76 65 73 0a 44 69  64 20 67 79 72 65 20 61   toves.Di d gyre a
0070  6e 64 20 67 69 6d 62 6c  65 20 69 6e 20 74 68 65   nd gimbl e in the
0080  20 77 61 62 65 3b 0a 41  6c 6c 20 6d 69 6d 73 79    wabe;.A ll mimsy
0090  20 77 65 72 65 20 74 68  65 20 62 6f 72 6f 67 6f    were th e borogo
00a0  76 65 73 2c 0a 41 6e 64  20 74 68 65 20 6d 6f 6d   ves,.And  the mom
00b0  65 20 72 61 74 68 73 20  6f 75 74 67 72 61 62 65   e raths  outgrabe
00c0  2e 0a 0a 0a 22 42 65 77  61 72 65 20 74 68 65 20   ...."Bew are the 
00d0  4a 61 62 62 65 72 77 6f  63 6b 2c 20 6d 79 20 73   Jabberwo ck, my s
00e0  6f 6e 21 0a 54 68 65 20  6a 61 77 73 20 74 68 61   on!.The  jaws tha
00f0  74 20 62 69 74 65 2c 20  74 68 65 20 63 6c 61 77   t bite,  the claw
0100  73 20 74 68 61 74 20 63  61 74 63 68 21 0a 42 65   s that c atch!.Be
0110  77 61 72 65 20 74 68 65  20 4a 75 62 6a 75 62 20   ware the  Jubjub 
0120  62 69 72 64 2c 20 61 6e  64 20 73 68 75 6e 0a 54   bird, an d shun.T
0130  68 65 20 66 72 75 6d 69  6f 75 73 20 42 61 6e 64   he frumi ous Band
0140  65 72 73 6e 61 74 63 68  21 22 0a 0a 0a 48 65 20   ersnatch !"...He 
0150  74 6f 6f 6b 20 68 69 73  20 76 6f 72 70 61 6c 20   took his  vorpal 
0160  73 77 6f 72 64 20 69 6e  20 68 61 6e 64 3a 0a 4c   sword in  hand:.L
0170  6f 6e 67 20 74 69 6d 65  20 74 68 65 20 6d 61 6e   ong time  the man
0180  78 6f 6d 65 20 66 6f 65  20 68 65 20 73 6f 75 67   xome foe  he soug
0190  68 74 2d 2d 0a 53 6f 20  72 65 73 74 65 64 20 68   ht--.So  rested h
01a0  65 20 62 79 20 74 68 65  20 54 75 6d 74 75 6d 20   e by the  Tumtum 
01b0  74 72 65 65 2c 0a 41 6e  64 20 73 74 6f 6f 64 20   tree,.An d stood 
01c0  61 77 68 69 6c 65 20 69  6e 20 74 68 6f 75 67 68   awhile i n though
01d0  74 2e 0a 0a 0a 41 6e 64  2c 20 61 73 20 69 6e 20   t....And , as in 
01e0  75 66 66 69 73 68 20 74  68 6f 75 67 68 74 20 68   uffish t hought h

File: "/tmp/wireshark_pcapng_enp3s0_20171130134700_B6t...   Packets: 11 · Displayed: 11 (100.0%) · Dropped: 0 (0.0%)   Profile: Default
```

Figure 2: Wireshark screenshot

2.1 Received Data

This is the text that was received by our server implementation:

’Twas brillig, and the slithy toves Did gyre and gimble in the wabe; All mimsy were the borogoves, And the mome raths outgrabe.

"Beware the Jabberwock, my son! The jaws that bite, the claws that catch! Beware the Jubjub bird, and shun The frumious Bandersnatch!"

He took his vorpal sword in hand: Long time the manxome foe he sought– So rested he by the Tumtum tree, And stood awhile in thought.

And, as in uffish thought he stood, The Jabberwock, with eyes of flame, Came whiffling through the tulgey wood, And burbled as it came!

One two! One two! And through and through The vorpal blade went snicker-snack! He left it dead, and with its head He went galumphing back.

"And hast thou slain the Jabberwock? Come to my arms, my beamish boy! O frabjous day! Callooh! Callay!" He chortled in his joy.

’Twas brillig, and the slithy toves Did gyre and gimble in the wabe; All mimsy were the borogoves, And the mome raths outgrabe.

3 Conclusion

TCP provides a connection oriented service. TCP uses connections between clients and servers. A TCP client sends data to the server and needs an acknowledgement to be sent back. If the acknowledgement is not received the data will be resent automatically.

4 Appendix

```
#include <sys/socket.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <iostream>
#include <fstream>

//
// default IP address (if not specified on the command line
//
#define IP_ADDR "0.0.0.0"


sockaddr sa;
sockaddr_in *sin = (sockaddr_in *) &sa;
int sk;

int main(int argc, char *argv[])
{
   sk = socket(PF_INET, SOCK_STREAM, 0);
   if ( sk < 0 ) return -1;

   memset(&sa,0,sizeof(sa));
   sin->sin_family = PF_INET;
   sin->sin_port = htons(5600);
   if ( argc<2 || inet_pton(PF_INET, argv[1], &(sin->sin_addr)) <= 0 )
      inet_pton(PF_INET, IP_ADDR, &(sin->sin_addr));

   if ( bind(sk, &sa, sizeof(sa)) < 0 )
   {
      printf("bind␣failed\n");
      return 0;
   }
   printf("bind␣succeeded\n");

   if ( listen(sk, 5) < 0 )
   {
      printf("listen␣failed\n");
      return 0;
   }
   printf("listen␣succeeded\n");

   sockaddr client;
   socklen_t clientlen = sizeof(client);

   int socket = accept(sk, &client, &clientlen);
   if (socket < 0)
   {
       printf("accept␣failed\n");
       return -1;
   }

   std::ofstream file("received_data.txt");
   char buffer[256];
   bzero(buffer, 256);
   int n = read(socket, buffer, 255);
   if (n < 0)
   {
       printf("Error␣reading␣from␣the␣socket\n");
       return -1;
   }
   file << buffer;
   std::cout << "Received␣data:␣" << std::endl;
```

```
    std::cout << buffer << std::endl;

    close(sk);
    printf("socket␣closed\n");
}
```