Assignment 1
Of
Network & Distributed System Lab (CS1052)

Masters of Technology in Computer Science And Engineering

submitted by
Arghya Bandyopadhyay
RollNo. 20CS4103

submitted to
Dr Sujoy Saha
Assistant Professor
Dept. of CSE

National Institute of Technology, Durgapur

27th March 2021

1. Write simple TCP and UDP program using socket API which will transfer simple text messages, and check TCP and UDP packets using Wireshark.

   Answer.

   The following code is the implementation for **TCP client side programming**.

```
/* tcpclient.c */

#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>


int main()
{

        int sock, bytes_recieved;
        char send_data[1024],recv_data[1024];
        struct hostent *host;
        struct sockaddr_in server_addr;

        host = gethostbyname(”127.0.0.1”);

        if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
            perror(”Socket”);
            exit(1);
        }

        server_addr.sin_family = AF_INET;
        server_addr.sin_port = htons(5000);
        server_addr.sin_addr = *((struct in_addr *)host->h_addr);
        bzero(&(server_addr.sin_zero),8);

        if (connect(sock, (struct sockaddr *)&server_addr,
                    sizeof(struct sockaddr)) == -1)
        {
            perror("Connect");
            exit(1);
        }

        while(1)
        {

          bytes_recieved=recv(sock,recv_data,1024,0);
          recv_data[bytes_recieved] = '\0';

          if (strcmp(recv_data , "q") == 0 || strcmp(recv_data , "Q") == 0)
          {
           close(sock);
           break;
          }

          else
           printf("\nRecieved data = %s " , recv_data);

           printf("\nSEND (q or Q to quit) : ");
           gets(send_data);

          if (strcmp(send_data , "q") != 0 && strcmp(send_data , "Q") != 0)
           send(sock,send_data,strlen(send_data), 0);

          else
          {
           send(sock,send_data,strlen(send_data), 0);
           close(sock);
           break;
          }

        }
return 0;
}
```

The following code is the implementation for **TCP Server side programming**.

```
/* tcpserver.c */

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>


int main()
{
        int sock, connected, bytes_recieved , true = 1;
        char send_data [1024] , recv_data[1024];

        struct sockaddr_in server_addr,client_addr;
        int sin_size;

        if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
            perror("Socket");
            exit(1);
        }

        if (setsockopt(sock,SOL_SOCKET,SO_REUSEADDR,&true,sizeof(int)) == -1) {
            perror("Setsockopt");
            exit(1);
        }

        server_addr.sin_family = AF_INET;
        server_addr.sin_port = htons(5000);
        server_addr.sin_addr.s_addr = INADDR_ANY;
        bzero(&(server_addr.sin_zero),8);

        if (bind(sock, (struct sockaddr *)&server_addr, sizeof(struct sockaddr))
                                                                       == -1) {
            perror("Unable to bind");
            exit(1);
```

```
}

if (listen(sock, 5) == -1) {
    perror("Listen");
    exit(1);
}

printf("\nTCPServer Waiting for client on port 5000");
fflush(stdout);


while(1)
{

    sin_size = sizeof(struct sockaddr_in);

    connected = accept(sock, (struct sockaddr *)&client_addr,&sin_size);

    printf("\n I got a connection from (%s , %d)",
           inet_ntoa(client_addr.sin_addr),ntohs(client_addr.sin_port));

    while (1)
    {
      printf("\n SEND (q or Q to quit) : ");
      gets(send_data);

      if (strcmp(send_data , "q") == 0 || strcmp(send_data , "Q") == 0)
      {
        send(connected, send_data,strlen(send_data), 0);
        close(connected);
        break;
      }

      else
         send(connected, send_data,strlen(send_data), 0);

      bytes_recieved = recv(connected,recv_data,1024,0);

      recv_data[bytes_recieved] = '\0';

      if (strcmp(recv_data , "q") == 0 || strcmp(recv_data , "Q") == 0)
      {
```

```
                close(connected);
                break;
              }

              else
              printf("\n RECIEVED DATA = %s " , recv_data);
              fflush(stdout);
            }
        }

      close(sock);
      return 0;
}
```

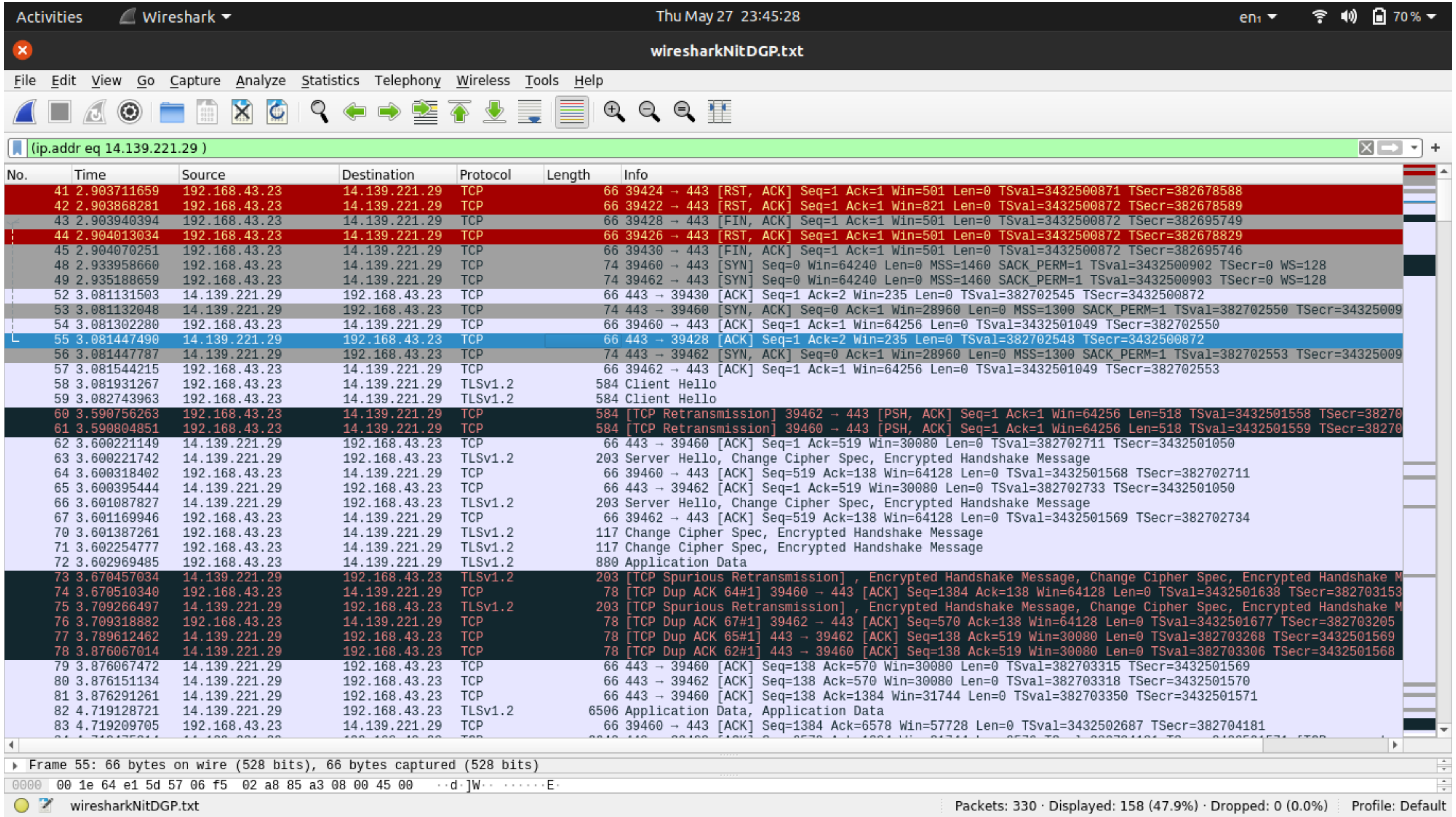
The following code is the implementation for **UDP Client side programming**.

```
/* udpclient.c */

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>

int main()
{
        int sock;
        struct sockaddr_in server_addr;
        struct hostent *host;
        char send_data[1024];

        host= (struct hostent *) gethostbyname((char *)”127.0.0.1”);


if ((sock = socket(AF_INET, SOCK_DGRAM, 0)) == -1)
{
        perror(”socket”);
        exit(1);
}

        server_addr.sin_family = AF_INET;
        server_addr.sin_port = htons(5000);
        server_addr.sin_addr = *((struct in_addr *)host->h_addr);
        bzero(&(server_addr.sin_zero),8);

   while (1)
   {

            printf(”Type Something (q or Q to quit):”);
            gets(send_data);
            if ((strcmp(send_data , ”q”) == 0) || strcmp(send_data , ”Q”) == 0)
           break;

        else
                 sendto(sock, send_data, strlen(send_data), 0,
                 (struct sockaddr *)&server_addr, sizeof(struct sockaddr));

   }

}
```

The following code is the implementation for **UDP Server side programming**.

```
/* udpserver.c */

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>

int main()
{
        int sock;
        int addr_len, bytes_read;
        char recv_data[1024];
        struct sockaddr_in server_addr , client_addr;


        if ((sock = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
            perror(”Socket”);
            exit(1);
        }

        server_addr.sin_family = AF_INET;
        server_addr.sin_port = htons(5000);
        server_addr.sin_addr.s_addr = INADDR_ANY;
        bzero(&(server_addr.sin_zero),8);


        if (bind(sock,(struct sockaddr *)&server_addr,
            sizeof(struct sockaddr)) == -1)
        {
            perror(”Bind”);
            exit(1);
        }

        addr_len = sizeof(struct sockaddr);
        printf(”\nUDPServer Waiting for client on port 5000”);
        fflush(stdout);

        while (1)
        {

          bytes_read = recvfrom(sock,recv_data,1024,0,
                           (struct sockaddr *)&client_addr, &addr_len);


          recv_data[bytes_read] = '\0';

          printf(”\n(%s , %d) said : ”,inet_ntoa(client_addr.sin_addr),
                                      ntohs(client_addr.sin_port));
          printf(”%s”, recv_data);
          fflush(stdout);

        }
        return 0;
}
```

2. Using wireshark, capture the TCP headers while connecting your computer to the server of nit.dgp.ac.in.

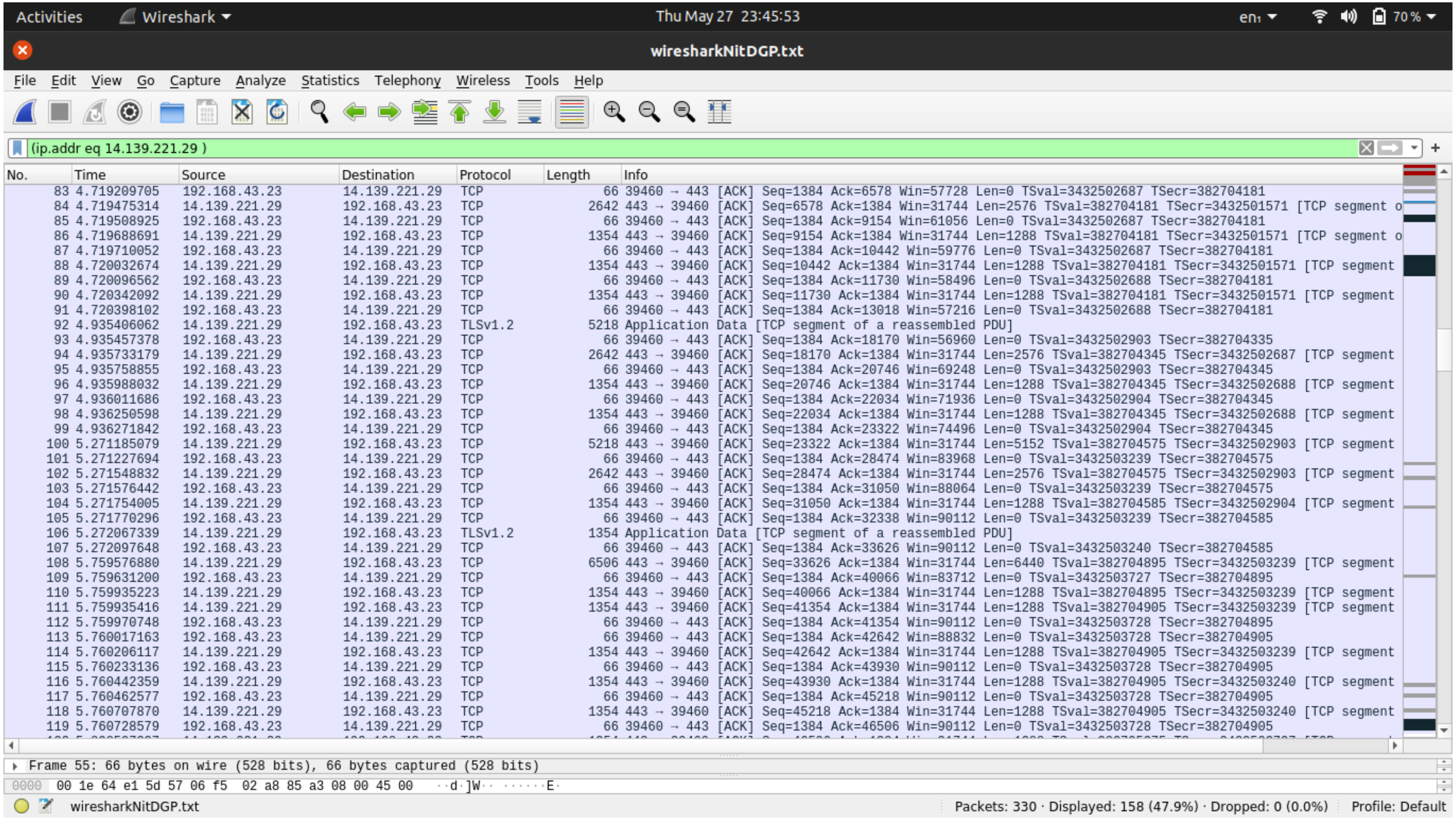

Activities
Wireshark
Thu May 27 23:45:28
en₁
70 %
wiresharkNitDGP.txt
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
(ip.addr eq 14.139.221.29 )
```

```
No.  Time          Source          Destination     Protocol  Length Info
  41 2.903711659   192.168.43.23   14.139.221.29   TCP         66 39424 → 443 [RST, ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3432500871 TSecr=382678588
  42 2.903868281   192.168.43.23   14.139.221.29   TCP         66 39422 → 443 [RST, ACK] Seq=1 Ack=1 Win=821 Len=0 TSval=3432500872 TSecr=382678589
  43 2.903940394   192.168.43.23   14.139.221.29   TCP         66 39428 → 443 [FIN, ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3432500872 TSecr=382695749
  44 2.904013034   192.168.43.23   14.139.221.29   TCP         66 39426 → 443 [RST, ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3432500872 TSecr=382678829
  45 2.904070251   192.168.43.23   14.139.221.29   TCP         66 39430 → 443 [FIN, ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3432500872 TSecr=382695746
  48 2.933958660   192.168.43.23   14.139.221.29   TCP         74 39460 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3432500902 TSecr=0 WS=128
  49 2.935188659   192.168.43.23   14.139.221.29   TCP         74 39462 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3432500903 TSecr=0 WS=128
  52 3.081131503   14.139.221.29   192.168.43.23   TCP         66 443 → 39430 [ACK] Seq=1 Ack=2 Win=235 Len=0 TSval=382702545 TSecr=3432500872
  53 3.081132048   14.139.221.29   192.168.43.23   TCP         74 443 → 39460 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1300 SACK_PERM=1 TSval=382702550 TSecr=34325009
  54 3.081302280   192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3432501049 TSecr=382702550
  55 3.081447490   14.139.221.29   192.168.43.23   TCP         66 443 → 39428 [ACK] Seq=1 Ack=2 Win=235 Len=0 TSval=382702548 TSecr=3432500872
  56 3.081447787   14.139.221.29   192.168.43.23   TCP         74 443 → 39462 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1300 SACK_PERM=1 TSval=382702553 TSecr=34325009
  57 3.081544215   192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3432501049 TSecr=382702553
  58 3.081931267   192.168.43.23   14.139.221.29   TLSv1.2    584 Client Hello
  59 3.082743963   192.168.43.23   14.139.221.29   TLSv1.2    584 Client Hello
  60 3.590756263   192.168.43.23   14.139.221.29   TCP        584 [TCP Retransmission] 39462 → 443 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=518 TSval=3432501558 TSecr=38270
  61 3.590804851   192.168.43.23   14.139.221.29   TCP        584 [TCP Retransmission] 39460 → 443 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=518 TSval=3432501559 TSecr=38270
  62 3.600221149   14.139.221.29   192.168.43.23   TCP         66 443 → 39460 [ACK] Seq=1 Ack=519 Win=30080 Len=0 TSval=382702711 TSecr=3432501050
  63 3.600221742   14.139.221.29   192.168.43.23   TLSv1.2    203 Server Hello, Change Cipher Spec, Encrypted Handshake Message
  64 3.600318402   192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=519 Ack=138 Win=64128 Len=0 TSval=3432501568 TSecr=382702711
  65 3.600395444   14.139.221.29   192.168.43.23   TCP         66 443 → 39462 [ACK] Seq=1 Ack=519 Win=30080 Len=0 TSval=382702733 TSecr=3432501050
  66 3.601087827   14.139.221.29   192.168.43.23   TLSv1.2    203 Server Hello, Change Cipher Spec, Encrypted Handshake Message
  67 3.601169946   192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=519 Ack=138 Win=64128 Len=0 TSval=3432501569 TSecr=382702734
  70 3.601387261   192.168.43.23   14.139.221.29   TLSv1.2    117 Change Cipher Spec, Encrypted Handshake Message
  71 3.602254777   192.168.43.23   14.139.221.29   TLSv1.2    117 Change Cipher Spec, Encrypted Handshake Message
  72 3.602969485   192.168.43.23   14.139.221.29   TLSv1.2    880 Application Data
  73 3.670457034   14.139.221.29   192.168.43.23   TLSv1.2    203 [TCP Spurious Retransmission] , Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake M
  74 3.670510340   192.168.43.23   14.139.221.29   TCP         78 [TCP Dup ACK 64#1] 39460 → 443 [ACK] Seq=1384 Ack=138 Win=64128 Len=0 TSval=3432501638 TSecr=382703153
  75 3.709266497   14.139.221.29   192.168.43.23   TLSv1.2    203 [TCP Spurious Retransmission] , Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake M
  76 3.709318882   192.168.43.23   14.139.221.29   TCP         78 [TCP Dup ACK 67#1] 39462 → 443 [ACK] Seq=570 Ack=138 Win=64128 Len=0 TSval=3432501677 TSecr=382703205
  77 3.789612462   14.139.221.29   192.168.43.23   TCP         78 [TCP Dup ACK 65#1] 443 → 39462 [ACK] Seq=138 Ack=519 Win=30080 Len=0 TSval=382703268 TSecr=3432501569
  78 3.876067014   14.139.221.29   192.168.43.23   TCP         78 [TCP Dup ACK 62#1] 443 → 39460 [ACK] Seq=138 Ack=519 Win=30080 Len=0 TSval=382703306 TSecr=3432501568
  79 3.876067472   14.139.221.29   192.168.43.23   TCP         66 443 → 39460 [ACK] Seq=138 Ack=570 Win=30080 Len=0 TSval=382703315 TSecr=3432501569
  80 3.876151134   14.139.221.29   192.168.43.23   TCP         66 443 → 39462 [ACK] Seq=138 Ack=570 Win=30080 Len=0 TSval=382703318 TSecr=3432501570
  81 3.876291261   14.139.221.29   192.168.43.23   TCP         66 443 → 39460 [ACK] Seq=138 Ack=1384 Win=31744 Len=0 TSval=382703350 TSecr=3432501571
  82 4.719128721   14.139.221.29   192.168.43.23   TLSv1.2   6506 Application Data, Application Data
  83 4.719209705   192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=1384 Ack=6578 Win=57728 Len=0 TSval=3432502687 TSecr=382704181
```

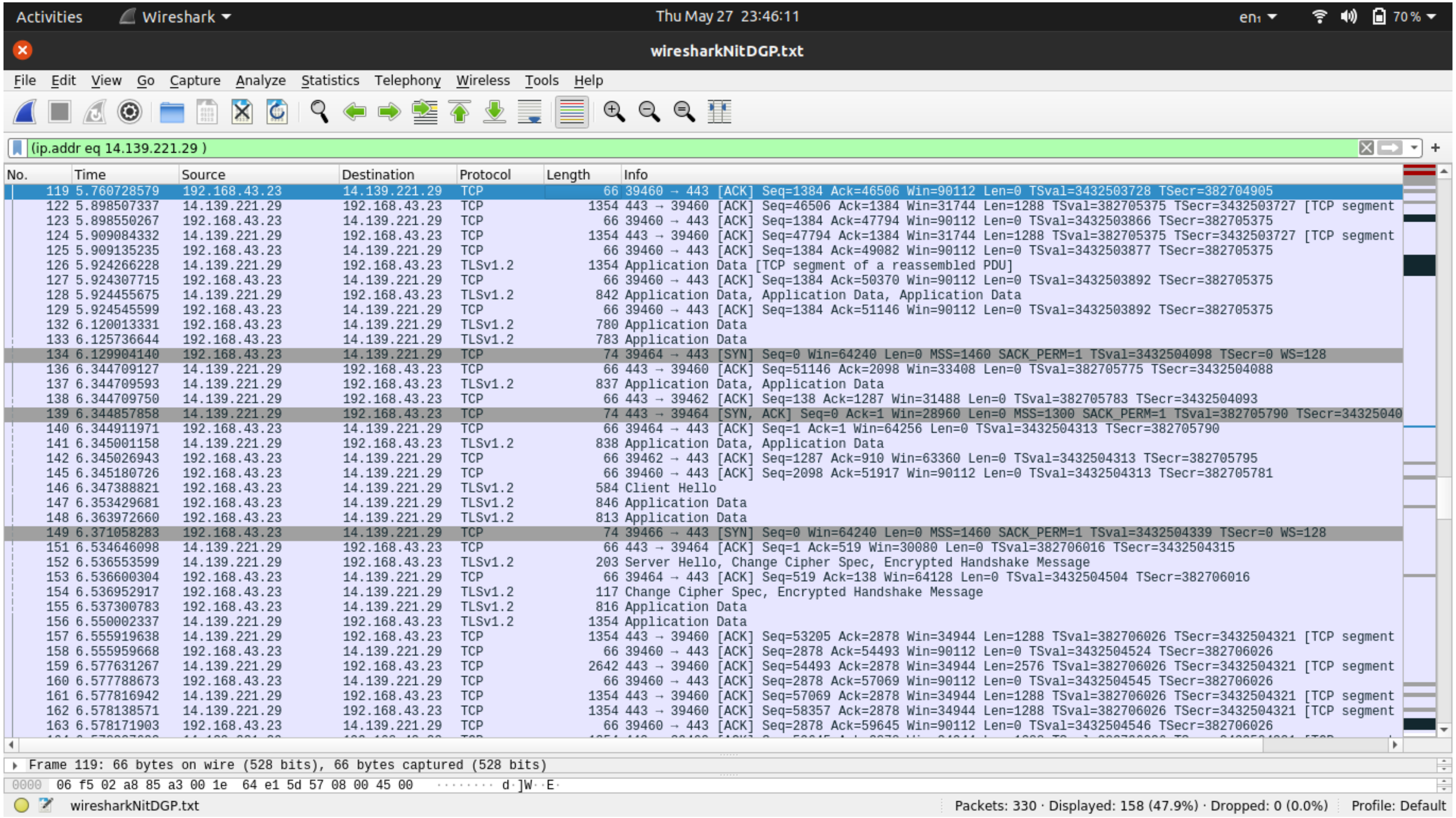
```
▸ Frame 55: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
0000  00 1e 64 e1 5d 57 06 f5  02 a8 85 a3 08 00 45 00   ··d·]W·· ······E·
```

```
wiresharkNitDGP.txt          Packets: 330 · Displayed: 158 (47.9%) · Dropped: 0 (0.0%)     Profile: Default
```



Activities
Wireshark
Thu May 27 23:45:53
en₁
70 %
wiresharkNitDGP.txt
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
(ip.addr eq 14.139.221.29 )

No.  Time         Source          Destination     Protocol Length Info
  83 4.719209705  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=6578 Win=57728 Len=0 TSval=3432502687 TSecr=382704181
  84 4.719475314  14.139.221.29   192.168.43.23   TCP      2642 443 → 39460 [ACK] Seq=6578 Ack=1384 Win=31744 Len=2576 TSval=382704181 TSecr=3432501571 [TCP segment o
  85 4.719508925  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=9154 Win=61056 Len=0 TSval=3432502687 TSecr=382704181
  86 4.719688691  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=9154 Ack=1384 Win=31744 Len=1288 TSval=382704181 TSecr=3432501571 [TCP segment o
  87 4.719710052  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=10442 Win=59776 Len=0 TSval=3432502687 TSecr=382704181
  88 4.720032674  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=10442 Ack=1384 Win=31744 Len=1288 TSval=382704181 TSecr=3432501571 [TCP segment
  89 4.720096562  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=11730 Win=58496 Len=0 TSval=3432502688 TSecr=382704181
  90 4.720342092  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=11730 Ack=1384 Win=31744 Len=1288 TSval=382704181 TSecr=3432501571 [TCP segment
  91 4.720398102  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=13018 Win=57216 Len=0 TSval=3432502688 TSecr=382704181
  92 4.935406062  14.139.221.29   192.168.43.23   TLSv1.2  5218 Application Data [TCP segment of a reassembled PDU]
  93 4.935457378  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=18170 Win=56960 Len=0 TSval=3432502903 TSecr=382704335
  94 4.935733179  14.139.221.29   192.168.43.23   TCP      2642 443 → 39460 [ACK] Seq=18170 Ack=1384 Win=31744 Len=2576 TSval=382704345 TSecr=3432502687 [TCP segment
  95 4.935758855  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=20746 Win=69248 Len=0 TSval=3432502903 TSecr=382704345
  96 4.935988032  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=20746 Ack=1384 Win=31744 Len=1288 TSval=382704345 TSecr=3432502688 [TCP segment
  97 4.936011686  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=22034 Win=71936 Len=0 TSval=3432502904 TSecr=382704345
  98 4.936250598  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=22034 Ack=1384 Win=31744 Len=1288 TSval=382704345 TSecr=3432502688 [TCP segment
  99 4.936271842  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=23322 Win=74496 Len=0 TSval=3432502904 TSecr=382704345
 100 5.271185079  14.139.221.29   192.168.43.23   TCP      5218 443 → 39460 [ACK] Seq=23322 Ack=1384 Win=31744 Len=5152 TSval=382704575 TSecr=3432502903 [TCP segment
 101 5.271227694  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=28474 Win=83968 Len=0 TSval=3432503239 TSecr=382704575
 102 5.271548832  14.139.221.29   192.168.43.23   TCP      2642 443 → 39460 [ACK] Seq=28474 Ack=1384 Win=31744 Len=2576 TSval=382704575 TSecr=3432502903 [TCP segment
 103 5.271576442  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=31050 Win=88064 Len=0 TSval=3432503239 TSecr=382704575
 104 5.271754005  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=31050 Ack=1384 Win=31744 Len=1288 TSval=382704585 TSecr=3432502904 [TCP segment
 105 5.271770296  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=32338 Win=90112 Len=0 TSval=3432503239 TSecr=382704585
 106 5.272067339  14.139.221.29   192.168.43.23   TLSv1.2  1354 Application Data [TCP segment of a reassembled PDU]
 107 5.272097648  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=33626 Win=90112 Len=0 TSval=3432503240 TSecr=382704585
 108 5.759576880  14.139.221.29   192.168.43.23   TCP      6506 443 → 39460 [ACK] Seq=33626 Ack=1384 Win=31744 Len=6440 TSval=382704895 TSecr=3432503239 [TCP segment
 109 5.759631200  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=40066 Win=83712 Len=0 TSval=3432503727 TSecr=382704895
 110 5.759935223  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=40066 Ack=1384 Win=31744 Len=1288 TSval=382704895 TSecr=3432503239 [TCP segment
 111 5.759935416  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=41354 Ack=1384 Win=31744 Len=1288 TSval=382704905 TSecr=3432503239 [TCP segment
 112 5.759970748  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=41354 Win=90112 Len=0 TSval=3432503728 TSecr=382704895
 113 5.760017163  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=42642 Win=88832 Len=0 TSval=3432503728 TSecr=382704905
 114 5.760206117  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=42642 Ack=1384 Win=31744 Len=1288 TSval=382704905 TSecr=3432503239 [TCP segment
 115 5.760233136  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=43930 Win=90112 Len=0 TSval=3432503728 TSecr=382704905
 116 5.760442359  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=43930 Ack=1384 Win=31744 Len=1288 TSval=382704905 TSecr=3432503240 [TCP segment
 117 5.760462577  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=45218 Win=90112 Len=0 TSval=3432503728 TSecr=382704905
 118 5.760707870  14.139.221.29   192.168.43.23   TCP      1354 443 → 39460 [ACK] Seq=45218 Ack=1384 Win=31744 Len=1288 TSval=382704905 TSecr=3432503240 [TCP segment
 119 5.760728579  192.168.43.23   14.139.221.29   TCP        66 39460 → 443 [ACK] Seq=1384 Ack=46506 Win=90112 Len=0 TSval=3432503728 TSecr=382704905

› Frame 55: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

0000  00 1e 64 e1 5d 57 06 f5  02 a8 85 a3 08 00 45 00   ··d·]W·· ······E·

wiresharkNitDGP.txt    Packets: 330 · Displayed: 158 (47.9%) · Dropped: 0 (0.0%)   Profile: Default
```

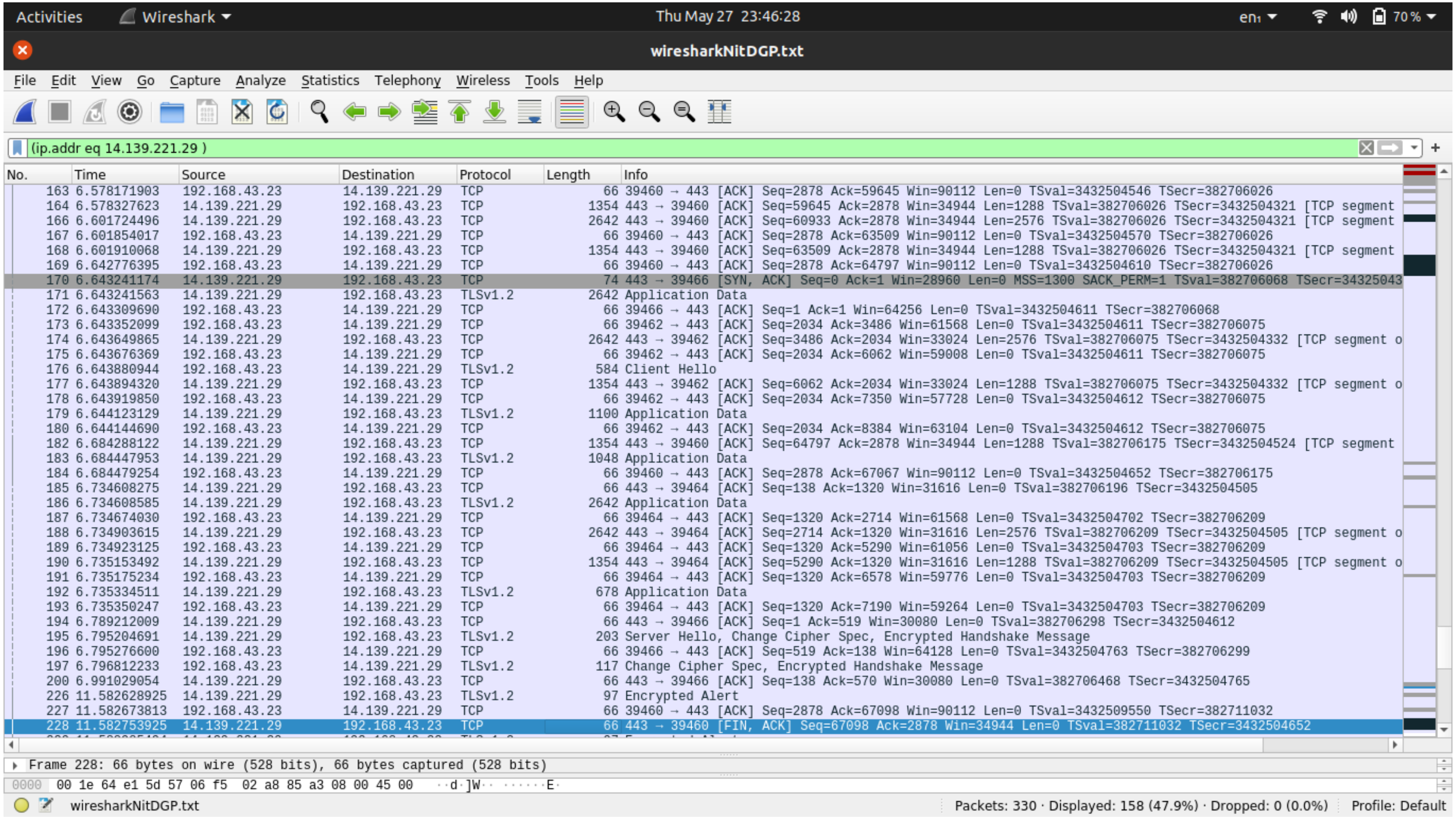

Activities
Wireshark
Thu May 27 23:46:11
en₁
70 %
wiresharkNitDGP.txt
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

(ip.addr eq 14.139.221.29)

```
No.     Time         Source          Destination     Protocol  Length Info
    119 5.760728579  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=1384 Ack=46506 Win=90112 Len=0 TSval=3432503728 TSecr=382704905
    122 5.898507337  14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=46506 Ack=1384 Win=31744 Len=1288 TSval=382705375 TSecr=3432503727 [TCP segment
    123 5.898550267  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=1384 Ack=47794 Win=90112 Len=0 TSval=3432503866 TSecr=382705375
    124 5.909084332  14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=47794 Ack=1384 Win=31744 Len=1288 TSval=382705375 TSecr=3432503727 [TCP segment
    125 5.909135235  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=1384 Ack=49082 Win=90112 Len=0 TSval=3432503877 TSecr=382705375
    126 5.924266228  14.139.221.29   192.168.43.23   TLSv1.2   1354 Application Data [TCP segment of a reassembled PDU]
    127 5.924307715  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=1384 Ack=50370 Win=90112 Len=0 TSval=3432503892 TSecr=382705375
    128 5.924455675  14.139.221.29   192.168.43.23   TLSv1.2    842 Application Data, Application Data, Application Data
    129 5.924545599  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=1384 Ack=51146 Win=90112 Len=0 TSval=3432503892 TSecr=382705375
    132 6.120013331  192.168.43.23   14.139.221.29   TLSv1.2    780 Application Data
    133 6.125736644  192.168.43.23   14.139.221.29   TLSv1.2    783 Application Data
    134 6.129904140  192.168.43.23   14.139.221.29   TCP         74 39464 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3432504098 TSecr=0 WS=128
    136 6.344709127  14.139.221.29   192.168.43.23   TCP         66 443 → 39460 [ACK] Seq=51146 Ack=2098 Win=33408 Len=0 TSval=382705775 TSecr=3432504088
    137 6.344709593  14.139.221.29   192.168.43.23   TLSv1.2    837 Application Data, Application Data
    138 6.344709750  14.139.221.29   192.168.43.23   TCP         66 443 → 39462 [ACK] Seq=138 Ack=1287 Win=31488 Len=0 TSval=382705783 TSecr=3432504093
    139 6.344857858  14.139.221.29   192.168.43.23   TCP         74 443 → 39464 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1300 SACK_PERM=1 TSval=382705790 TSecr=34325040
    140 6.344911971  192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3432504313 TSecr=382705790
    141 6.345001158  14.139.221.29   192.168.43.23   TLSv1.2    838 Application Data, Application Data
    142 6.345026943  192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=1287 Ack=910 Win=63360 Len=0 TSval=3432504313 TSecr=382705795
    145 6.345180726  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2098 Ack=51917 Win=90112 Len=0 TSval=3432504313 TSecr=382705781
    146 6.347388821  192.168.43.23   14.139.221.29   TLSv1.2    584 Client Hello
    147 6.353429681  192.168.43.23   14.139.221.29   TLSv1.2    846 Application Data
    148 6.363972660  192.168.43.23   14.139.221.29   TLSv1.2    813 Application Data
    149 6.371058283  192.168.43.23   14.139.221.29   TCP         74 39466 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3432504339 TSecr=0 WS=128
    151 6.534646098  14.139.221.29   192.168.43.23   TCP         66 443 → 39464 [ACK] Seq=1 Ack=519 Win=30080 Len=0 TSval=382706016 TSecr=3432504315
    152 6.536553599  14.139.221.29   192.168.43.23   TLSv1.2    203 Server Hello, Change Cipher Spec, Encrypted Handshake Message
    153 6.536600304  192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=519 Ack=138 Win=64128 Len=0 TSval=3432504504 TSecr=382706016
    154 6.536952917  192.168.43.23   14.139.221.29   TLSv1.2    117 Change Cipher Spec, Encrypted Handshake Message
    155 6.537300783  192.168.43.23   14.139.221.29   TLSv1.2    816 Application Data
    156 6.550002337  14.139.221.29   192.168.43.23   TLSv1.2   1354 Application Data
    157 6.555919638  14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=53205 Ack=2878 Win=34944 Len=1288 TSval=382706026 TSecr=3432504321 [TCP segment
    158 6.555959668  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=54493 Win=90112 Len=0 TSval=3432504524 TSecr=382706026
    159 6.577631267  14.139.221.29   192.168.43.23   TCP       2642 443 → 39460 [ACK] Seq=54493 Ack=2878 Win=34944 Len=2576 TSval=382706026 TSecr=3432504321 [TCP segment
    160 6.577788673  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=57069 Win=90112 Len=0 TSval=3432504545 TSecr=382706026
    161 6.577816942  14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=57069 Ack=2878 Win=34944 Len=1288 TSval=382706026 TSecr=3432504321 [TCP segment
    162 6.578138571  14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=58357 Ack=2878 Win=34944 Len=1288 TSval=382706026 TSecr=3432504321 [TCP segment
    163 6.578171903  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=59645 Win=90112 Len=0 TSval=3432504546 TSecr=382706026
```

```
▸ Frame 119: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
0000  06 f5 02 a8 85 a3 00 1e  64 e1 5d 57 08 00 45 00   ········ d·]W··E·
```

wiresharkNitDGP.txt   Packets: 330 · Displayed: 158 (47.9%) · Dropped: 0 (0.0%)   Profile: Default



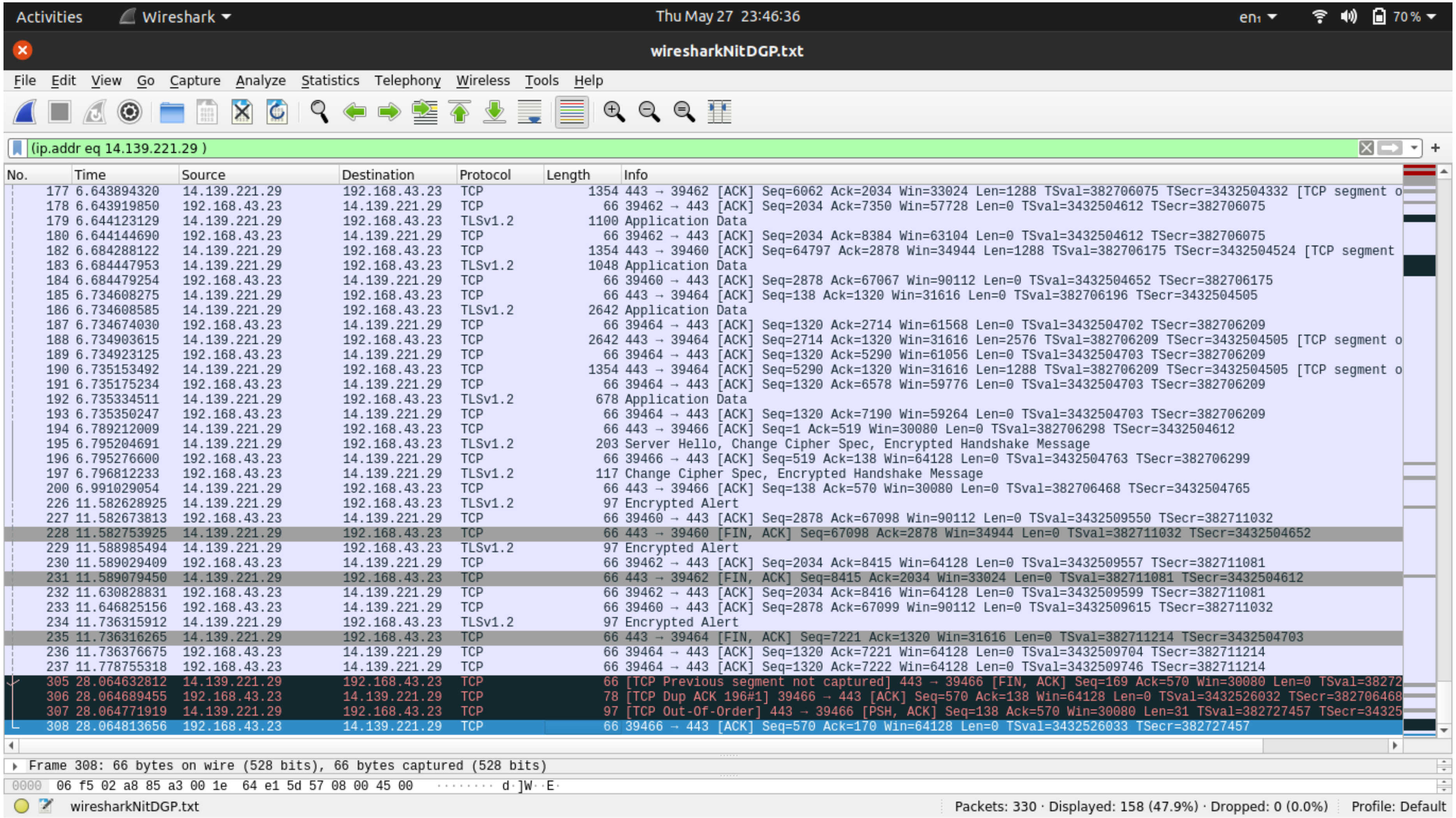
Activities
Wireshark
Thu May 27 23:46:28
en₁
70 %
wiresharkNitDGP.txt
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
(ip.addr eq 14.139.221.29 )
```

```
No.  Time         Source          Destination     Protocol  Length Info
 163 6.578171903  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=59645 Win=90112 Len=0 TSval=3432504546 TSecr=382706026
 164 6.578327623  14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=59645 Ack=2878 Win=34944 Len=1288 TSval=382706026 TSecr=3432504321 [TCP segment
 166 6.601724496  14.139.221.29   192.168.43.23   TCP       2642 443 → 39460 [ACK] Seq=60933 Ack=2878 Win=34944 Len=2576 TSval=382706026 TSecr=3432504321 [TCP segment
 167 6.601854017  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=63509 Win=90112 Len=0 TSval=3432504570 TSecr=382706026
 168 6.601910068  14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=63509 Ack=2878 Win=34944 Len=1288 TSval=382706026 TSecr=3432504321 [TCP segment
 169 6.642776395  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=64797 Win=90112 Len=0 TSval=3432504610 TSecr=382706026
 170 6.643241174  14.139.221.29   192.168.43.23   TCP         74 443 → 39466 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1300 SACK_PERM=1 TSval=382706068 TSecr=34325043
 171 6.643241563  14.139.221.29   192.168.43.23   TLSv1.2   2642 Application Data
 172 6.643309690  192.168.43.23   14.139.221.29   TCP         66 39466 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3432504611 TSecr=382706068
 173 6.643352099  192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=2034 Ack=3486 Win=61568 Len=0 TSval=3432504611 TSecr=382706075
 174 6.643649865  14.139.221.29   192.168.43.23   TCP       2642 443 → 39462 [ACK] Seq=3486 Ack=2034 Win=33024 Len=2576 TSval=382706075 TSecr=3432504332 [TCP segment o
 175 6.643676369  192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=2034 Ack=6062 Win=59008 Len=0 TSval=3432504611 TSecr=382706075
 176 6.643880944  192.168.43.23   14.139.221.29   TLSv1.2    584 Client Hello
 177 6.643894320  14.139.221.29   192.168.43.23   TCP       1354 443 → 39462 [ACK] Seq=6062 Ack=2034 Win=33024 Len=1288 TSval=382706075 TSecr=3432504332 [TCP segment o
 178 6.643919850  192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=2034 Ack=7350 Win=57728 Len=0 TSval=3432504612 TSecr=382706075
 179 6.644123129  14.139.221.29   192.168.43.23   TLSv1.2   1100 Application Data
 180 6.644144690  192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=2034 Ack=8384 Win=63104 Len=0 TSval=3432504612 TSecr=382706075
 182 6.684288122  14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=64797 Ack=2878 Win=34944 Len=1288 TSval=382706175 TSecr=3432504524 [TCP segment
 183 6.684447953  14.139.221.29   192.168.43.23   TLSv1.2   1048 Application Data
 184 6.684479254  192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=67067 Win=90112 Len=0 TSval=3432504652 TSecr=382706175
 185 6.734608275  14.139.221.29   192.168.43.23   TCP         66 443 → 39464 [ACK] Seq=138 Ack=1320 Win=31616 Len=0 TSval=382706196 TSecr=3432504505
 186 6.734608585  14.139.221.29   192.168.43.23   TLSv1.2   2642 Application Data
 187 6.734674030  192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=2714 Win=61568 Len=0 TSval=3432504702 TSecr=382706209
 188 6.734903615  14.139.221.29   192.168.43.23   TCP       2642 443 → 39464 [ACK] Seq=2714 Ack=1320 Win=31616 Len=2576 TSval=382706209 TSecr=3432504505 [TCP segment o
 189 6.734923125  192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=5290 Win=61056 Len=0 TSval=3432504703 TSecr=382706209
 190 6.735153492  14.139.221.29   192.168.43.23   TCP       1354 443 → 39464 [ACK] Seq=5290 Ack=1320 Win=31616 Len=1288 TSval=382706209 TSecr=3432504505 [TCP segment o
 191 6.735175234  192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=6578 Win=59776 Len=0 TSval=3432504703 TSecr=382706209
 192 6.735334511  14.139.221.29   192.168.43.23   TLSv1.2    678 Application Data
 193 6.735350247  192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=7190 Win=59264 Len=0 TSval=3432504703 TSecr=382706209
 194 6.789212009  14.139.221.29   192.168.43.23   TCP         66 443 → 39466 [ACK] Seq=1 Ack=519 Win=30080 Len=0 TSval=382706298 TSecr=3432504612
 195 6.795204691  14.139.221.29   192.168.43.23   TLSv1.2    203 Server Hello, Change Cipher Spec, Encrypted Handshake Message
 196 6.795276600  192.168.43.23   14.139.221.29   TCP         66 39466 → 443 [ACK] Seq=519 Ack=138 Win=64128 Len=0 TSval=3432504763 TSecr=382706299
 197 6.796812233  192.168.43.23   14.139.221.29   TLSv1.2    117 Change Cipher Spec, Encrypted Handshake Message
 200 6.991029054  14.139.221.29   192.168.43.23   TCP         66 443 → 39466 [ACK] Seq=138 Ack=570 Win=30080 Len=0 TSval=382706468 TSecr=3432504765
 226 11.582628925 14.139.221.29   192.168.43.23   TLSv1.2     97 Encrypted Alert
 227 11.582673813 192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=67098 Win=90112 Len=0 TSval=3432509550 TSecr=382711032
 228 11.582753925 14.139.221.29   192.168.43.23   TCP         66 443 → 39460 [FIN, ACK] Seq=67098 Ack=2878 Win=34944 Len=0 TSval=382711032 TSecr=3432504652
```

```
Frame 228: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
0000  00 1e 64 e1 5d 57 06 f5  02 a8 85 a3 08 00 45 00   ··d·]W·· ······E·
```

```
wiresharkNitDGP.txt    Packets: 330 · Displayed: 158 (47.9%) · Dropped: 0 (0.0%)    Profile: Default
```


Activities
Wireshark
Thu May 27 23:46:36
en
70 %
wiresharkNitDGP.txt
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
(ip.addr eq 14.139.221.29 )
```

```
No.     Time           Source          Destination     Protocol  Length Info
    177 6.643894320    14.139.221.29   192.168.43.23   TCP       1354 443 → 39462 [ACK] Seq=6062 Ack=2034 Win=33024 Len=1288 TSval=382706075 TSecr=3432504332 [TCP segment o
    178 6.643919850    192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=2034 Ack=7350 Win=57728 Len=0 TSval=3432504612 TSecr=382706075
    179 6.644123129    14.139.221.29   192.168.43.23   TLSv1.2   1100 Application Data
    180 6.644144690    192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=2034 Ack=8384 Win=63104 Len=0 TSval=3432504612 TSecr=382706075
    182 6.684288122    14.139.221.29   192.168.43.23   TCP       1354 443 → 39460 [ACK] Seq=64797 Ack=2878 Win=34944 Len=1288 TSval=382706175 TSecr=3432504524 [TCP segment
    183 6.684447953    14.139.221.29   192.168.43.23   TLSv1.2   1048 Application Data
    184 6.684479254    192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=67067 Win=90112 Len=0 TSval=3432504652 TSecr=382706175
    185 6.734608275    14.139.221.29   192.168.43.23   TCP         66 443 → 39464 [ACK] Seq=138 Ack=1320 Win=31616 Len=0 TSval=382706196 TSecr=3432504505
    186 6.734608585    14.139.221.29   192.168.43.23   TLSv1.2   2642 Application Data
    187 6.734674030    192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=2714 Win=61568 Len=0 TSval=3432504702 TSecr=382706209
    188 6.734903615    14.139.221.29   192.168.43.23   TCP       2642 443 → 39464 [ACK] Seq=2714 Ack=1320 Win=31616 Len=2576 TSval=382706209 TSecr=3432504505 [TCP segment o
    189 6.734923125    192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=5290 Win=61056 Len=0 TSval=3432504703 TSecr=382706209
    190 6.735153492    14.139.221.29   192.168.43.23   TCP       1354 443 → 39464 [ACK] Seq=5290 Ack=1320 Win=31616 Len=1288 TSval=382706209 TSecr=3432504505 [TCP segment o
    191 6.735175234    192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=6578 Win=59776 Len=0 TSval=3432504703 TSecr=382706209
    192 6.735334511    14.139.221.29   192.168.43.23   TLSv1.2    678 Application Data
    193 6.735350247    192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=7190 Win=59264 Len=0 TSval=3432504703 TSecr=382706209
    194 6.789212009    14.139.221.29   192.168.43.23   TCP         66 443 → 39466 [ACK] Seq=1 Ack=519 Win=30080 Len=0 TSval=382706298 TSecr=3432504612
    195 6.795204691    14.139.221.29   192.168.43.23   TLSv1.2    203 Server Hello, Change Cipher Spec, Encrypted Handshake Message
    196 6.795276600    192.168.43.23   14.139.221.29   TCP         66 39466 → 443 [ACK] Seq=519 Ack=138 Win=64128 Len=0 TSval=3432504763 TSecr=382706299
    197 6.796812233    192.168.43.23   14.139.221.29   TLSv1.2    117 Change Cipher Spec, Encrypted Handshake Message
    200 6.991029054    14.139.221.29   192.168.43.23   TCP         66 443 → 39466 [ACK] Seq=138 Ack=570 Win=30080 Len=0 TSval=382706468 TSecr=3432504765
    226 11.582628925   14.139.221.29   192.168.43.23   TLSv1.2     97 Encrypted Alert
    227 11.582673813   192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=67098 Win=90112 Len=0 TSval=3432509550 TSecr=382711032
    228 11.582753925   14.139.221.29   192.168.43.23   TCP         66 443 → 39460 [FIN, ACK] Seq=67098 Ack=2878 Win=34944 Len=0 TSval=382711032 TSecr=3432504652
    229 11.588985494   14.139.221.29   192.168.43.23   TLSv1.2     97 Encrypted Alert
    230 11.589029409   192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=2034 Ack=8415 Win=64128 Len=0 TSval=3432509557 TSecr=382711081
    231 11.589079450   14.139.221.29   192.168.43.23   TCP         66 443 → 39462 [FIN, ACK] Seq=8415 Ack=2034 Win=33024 Len=0 TSval=382711081 TSecr=3432504612
    232 11.630828831   192.168.43.23   14.139.221.29   TCP         66 39462 → 443 [ACK] Seq=2034 Ack=8416 Win=64128 Len=0 TSval=3432509599 TSecr=382711081
    233 11.646825156   192.168.43.23   14.139.221.29   TCP         66 39460 → 443 [ACK] Seq=2878 Ack=67099 Win=90112 Len=0 TSval=3432509615 TSecr=382711032
    234 11.736315912   14.139.221.29   192.168.43.23   TLSv1.2     97 Encrypted Alert
    235 11.736316265   14.139.221.29   192.168.43.23   TCP         66 443 → 39464 [FIN, ACK] Seq=7221 Ack=1320 Win=31616 Len=0 TSval=382711214 TSecr=3432504703
    236 11.736376675   192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=7221 Win=64128 Len=0 TSval=3432509704 TSecr=382711214
    237 11.778755318   192.168.43.23   14.139.221.29   TCP         66 39464 → 443 [ACK] Seq=1320 Ack=7222 Win=64128 Len=0 TSval=3432509746 TSecr=382711214
    305 28.064632812   14.139.221.29   192.168.43.23   TCP         66 [TCP Previous segment not captured] 443 → 39466 [FIN, ACK] Seq=169 Ack=570 Win=30080 Len=0 TSval=38272
    306 28.064689455   192.168.43.23   14.139.221.29   TCP         78 [TCP Dup ACK 196#1] 39466 → 443 [ACK] Seq=570 Ack=138 Win=64128 Len=0 TSval=3432526032 TSecr=382706468
    307 28.064771919   14.139.221.29   192.168.43.23   TCP         97 [TCP Out-Of-Order] 443 → 39466 [PSH, ACK] Seq=138 Ack=570 Win=30080 Len=31 TSval=382727457 TSecr=34325
    308 28.064813656   192.168.43.23   14.139.221.29   TCP         66 39466 → 443 [ACK] Seq=570 Ack=170 Win=64128 Len=0 TSval=3432526033 TSecr=382727457

▸ Frame 308: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

0000  06 f5 02 a8 85 a3 00 1e  64 e1 5d 57 08 00 45 00   ········ d·]W··E·
```

```
wiresharkNitDGP.txt        Packets: 330 · Displayed: 158 (47.9%) · Dropped: 0 (0.0%)        Profile: Default
```

3. (a) Show how the six flags (SYN, ACK, PUSH, URGENT, RST, FIN) are working in TCP protocol

   (b) What is the IP address of nitdgp.ac.in? On what port number is it sending and receiving TCP segments for this connection?

   The IP address of nitdgp.ac.in is **14.139.221.29** and the tcp port number for sending and receiving TCP segments for the connections are **39424** and **443** respectively.

   (c) Write a small socket program for the URGENT pointer and urgent flag?

   (d) What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and nitdgp.ac.in?

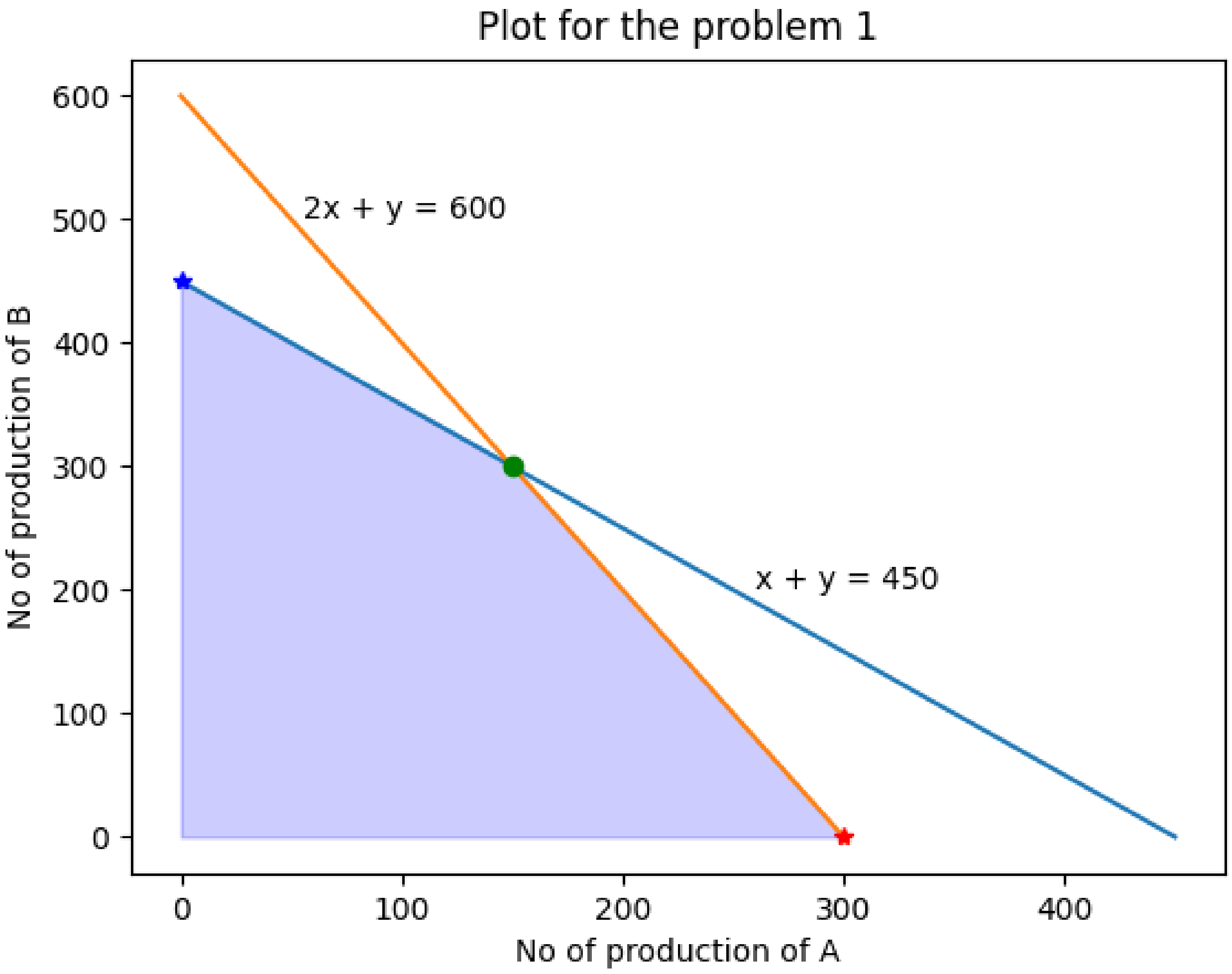
   (e) What is it in the segment that identifies the segment as a SYN segment?

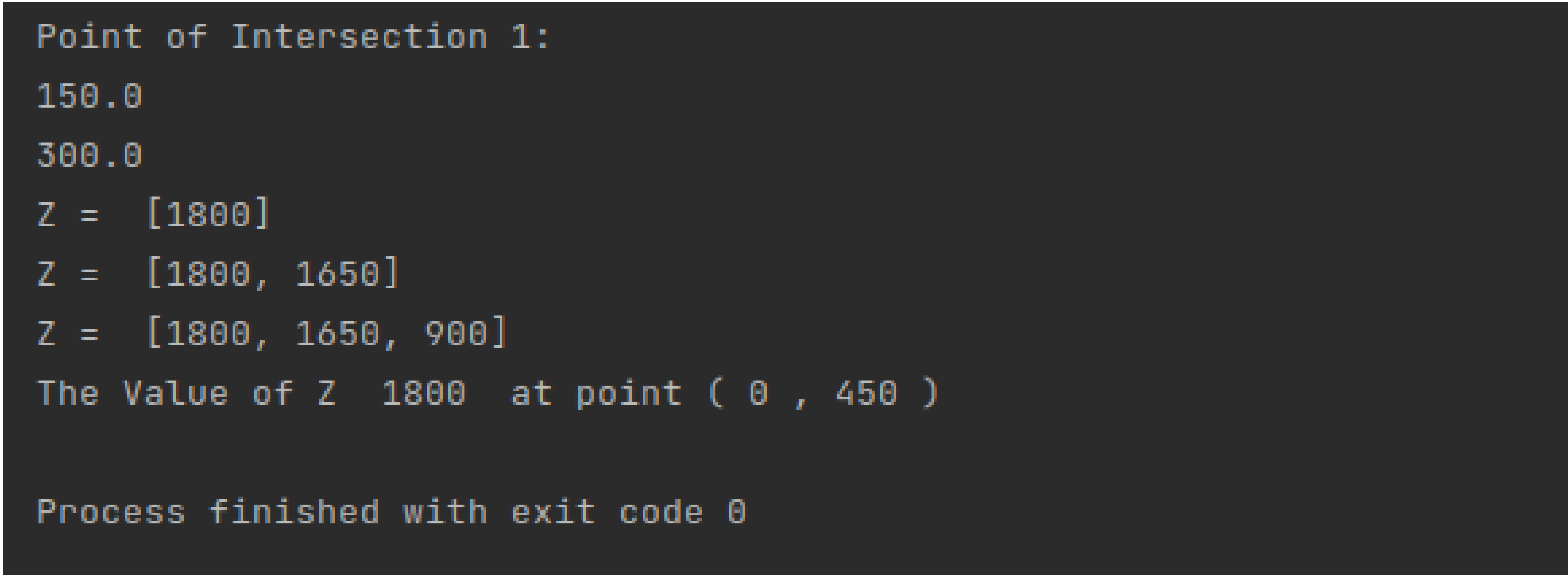
(f) What is the sequence number of the SYN-ACK segment sent by nitdgp.ac.in to the client computer in reply to the SYN?

(g) What is the value of the Acknowledgement field in the SYN-ACK segment?

(h) How did nitdgp.ac.in determine that value?

(i) What is it in the segment that identifies the segment as a SYN-ACK segment?





```
Point of Intersection 1:
150.0
300.0
Z =  [1800]
Z =  [1800, 1650]
Z =  [1800, 1650, 900]
The Value of Z  1800  at point ( 0 , 450 )

Process finished with exit code 0
```

Problem statement 2

Egg contains 6 units of vitamin A per gram and 7 units of vitamin B per gram and costs 12 paisa per gram Milk contains 8 units of vitamins A per gram and 12 units of vitamin B per gram and costs 20 paisa per gram. The daity minimum A&B is 100 unit & 120 units. Formulate the LPP and find the optimal solution by graphical method.

Problem formulation

amount of vitamin $A \geq 100$ unit

amount of vitamin $B \geq 120$ unit

Lets take. x gm of Egg

and y gm of milk.

vitamin A content $= 6x + 8y \geq 100$ —

vitamin B content $= 7x + 12y \geq 120$ —

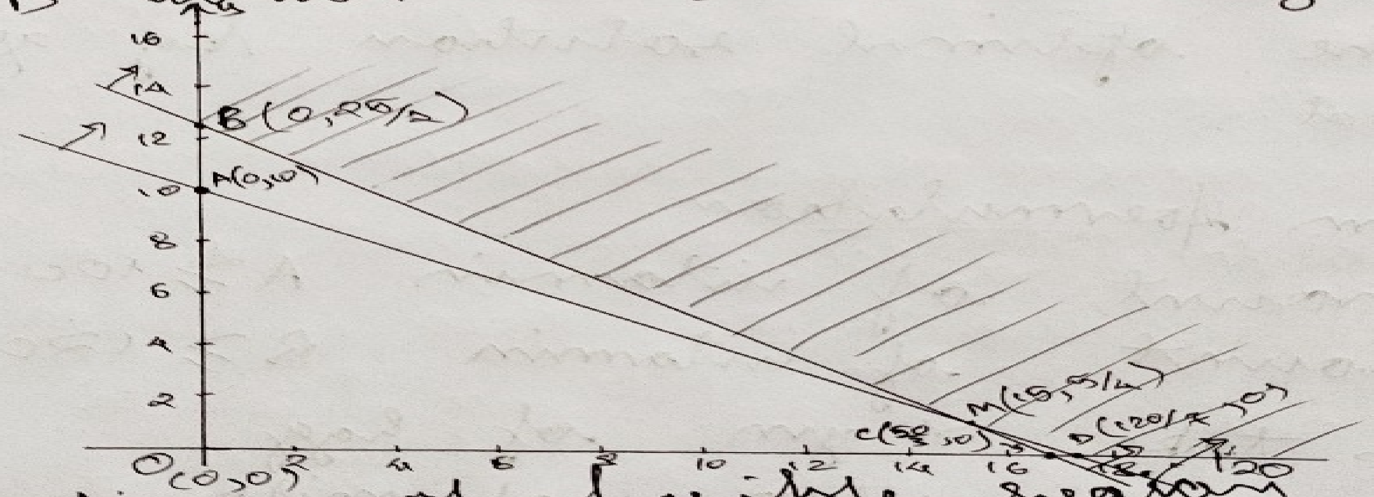
cost of food $(c) = 12x + 20y$

$x \geq 0$, $y \geq 0$

from eqn ① →
when $x=0$, $y=100/8=25/2$ $(0,25/2)$
for $y=0$, $x=100/6=50/3$ $(50/3,0)$

from eqn ② →
when $x=0$, $y=120/12=10$ $(0,10)$
when $y=0$, $x=120/7$ $(120/7,0)$

Feasible unbound region is BMD which is shaded in graph.



vertices of feasible region
$B(0,25/2)$, $D(120/7,0)$
M is point of intersection of the lines

$6x+8y \geq 100$ $\times 3$ $18x+24y=300$

hence $x = 15 \Rightarrow 6 \times 15 + 8y = 100$

$$y = \frac{10}{8} = \frac{5}{4}$$

$$M\left(15, \frac{5}{4}\right)$$

objective function $z = 12x + 20y$

cost at $B\left(0, \frac{25}{2}\right) = 12 \times 0 + \frac{25}{2} \times 20$

$= 250$ paise

cost at $D\left(\frac{120}{7}, 0\right) = \frac{120}{7} \times 12 + 20(0)$

$= 205.7$ paise

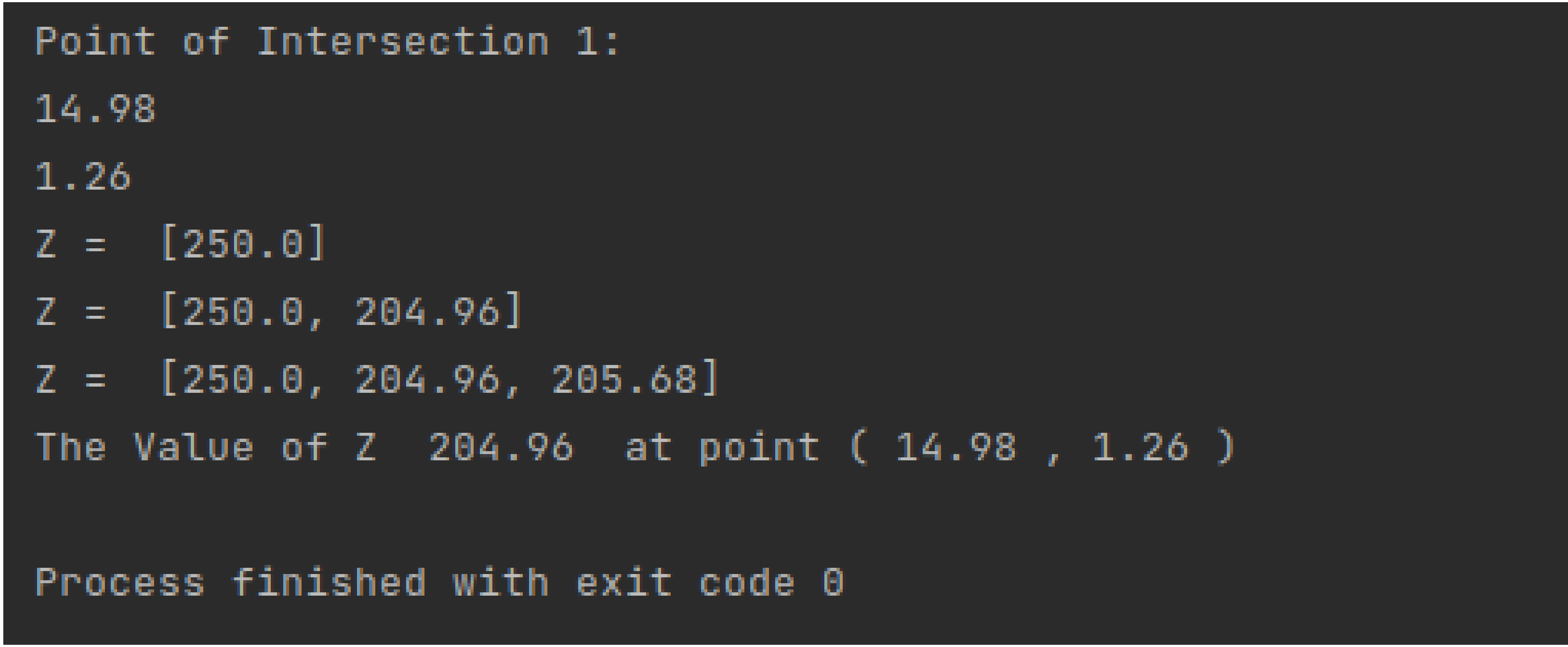
cost at $M\left(15, \frac{5}{4}\right) = 12(15) + \frac{20 \times 5}{4}$

$= 205$ paise

Minimum cost for the diet = 205 pa

in which 15 are eggs and

1.25 of milk.



```
Point of Intersection 1:
14.98
1.26
Z =  [250.0]
Z =  [250.0, 204.96]
Z =  [250.0, 204.96, 205.68]
The Value of Z  204.96  at point ( 14.98 , 1.26 )

Process finished with exit code 0
```