1.Developing a TCP Client-Server Application using Linux Socket Programming

Part 1: Server Side

Step 1: Writing the Server Code

1. Create a file named `server.c` in the project directory.

   c

1.
```c
// server.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>

#define PORT 8080
#define MAX_BUFFER_SIZE 1024

int main() {
    int server_fd, new_socket, valread;
    struct sockaddr_in address;
    int addrlen = sizeof(address);
    char buffer[MAX_BUFFER_SIZE] = {0};

    // Create a socket
    if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0) {
     perror("Socket creation failed");
     exit(EXIT_FAILURE);
    }

    // Set up server address struct
    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(PORT);

    // Bind the socket to the address
    if (bind(server_fd, (struct sockaddr *)&address, sizeof(address)) < 0)
{
        perror("Bind failed");
        exit(EXIT_FAILURE);
    }

    // Listen for incoming connections
    if (listen(server_fd, 3) < 0) {
        perror("Listen failed");
        exit(EXIT_FAILURE);
    }

    // Accept incoming connection
if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
(socklen_t*)&addrlen)) < 0) {
        perror("Accept failed");
        exit(EXIT_FAILURE);
    }

    // Read data from the client using TCP
    valread = read(new_socket, buffer, MAX_BUFFER_SIZE);
    printf("Received message from client: %s\n", buffer);
    // Close the connection
    close(new_socket);
    close(server_fd);

    return 0;
}
```

Step 2: Compiling and Running the Server Code

1. Compile the server code.

   bash

   ```
   gcc server.c -o server
   ```

- Run the server.

bash

```
./server
```

Part 2: Client Side

Step 1: Writing the Client Code

1. Create a file named `client.c` in the project directory.

   C

```
// client.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>

#define PORT 8080
#define MAX_BUFFER_SIZE 1024

int main() {
    int client_fd;
    struct sockaddr_in server_address; char
    message[MAX_BUFFER_SIZE];

    // Create a socket
    if ((client_fd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Configure server address server_address.sin_family
    = AF_INET; server_address.sin_port = htons(PORT);
    if (inet_pton(AF_INET, "127.0.0.1", &server_address.sin_addr) <= 0) {
        perror("Invalid address/ Address not supported");
        exit(EXIT_FAILURE);
    }

    // Connect to the server using TCP
  if (connect(client_fd, (struct sockaddr *)&server_address,
sizeof(server_address)) < 0) {
        perror("Connection Failed");
        exit(EXIT_FAILURE);
    }

    // Get user input for the message
```

```
        printf("Enter a message to send to the server: ");
        fgets(message, MAX_BUFFER_SIZE, stdin);

        // Send the message to the server using TCP
        send(client_fd, message, strlen(message), 0);

        // Close the connection
        close(client_fd);

        return 0;
    }
```

Step 2: Compiling and Running the Client Code

1. Compile the client code.

   bash

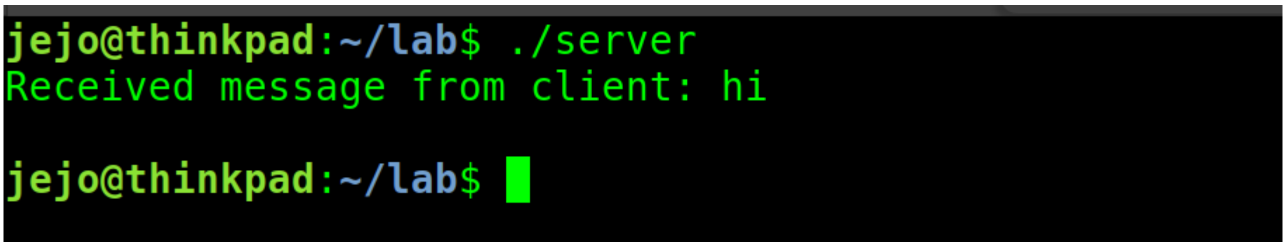
   - `gcc client.c -o client`

- Run the client.

bash

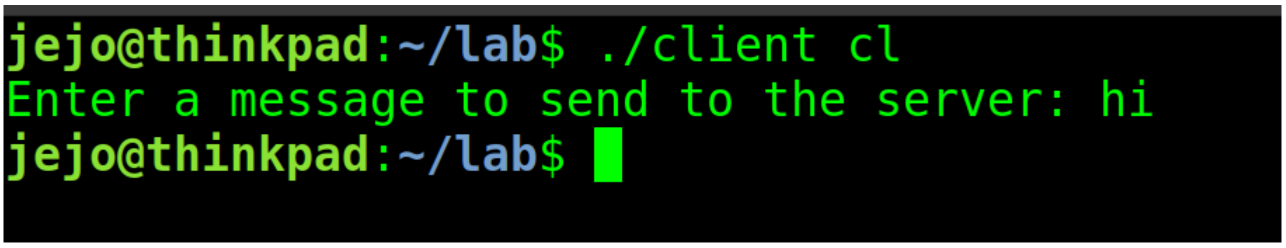
```
2. ./client
```

OUTPUT:



```
jejo@thinkpad:~/lab$ ./server
Received message from client: hi

jejo@thinkpad:~/lab$ 
```



```
jejo@thinkpad:~/lab$ ./client cl
Enter a message to send to the server: hi
jejo@thinkpad:~/lab$ 
```

2.Developing a UDP Client-Server Application using UNIX Socket Programming

Part 1: Server Side

Step 1: Writing the Server Code

```
1. // udp_server.c
   #include <stdio.h>
   #include <stdlib.h>
   #include <unistd.h>
   #include <string.h>
   #include <arpa/inet.h>

   #define PORT 8080
   #define MAX_BUFFER_SIZE 1024

   int main() {
       int server_fd;
       struct sockaddr_in server_address, client_address; socklen_t
       client_address_len = sizeof(client_address); char
       buffer[MAX_BUFFER_SIZE] = {0};

       // Create a UDP socket
       if ((server_fd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
           perror("Socket creation failed");
           exit(EXIT_FAILURE);
       }

       // Configure server address server_address.sin_family =
       AF_INET; server_address.sin_addr.s_addr = INADDR_ANY;
       server_address.sin_port = htons(PORT);

       // Bind the socket to the address
    if (bind(server_fd, (struct sockaddr *)&server_address,
sizeof(server_address)) == -1) {
           perror("Bind failed"); exit(EXIT_FAILURE);
       }

       // Server logic here (you can modify this part)

       return 0;
   }
```

Step 2: Compiling and Running the Server Code

1. Compile the server code.

```
gcc udp_server.c -o udp_server
```

- Run the server.

```
bash
```

```
2. ./udp_server
```

Part 2: Client Side

Step 1: Writing the Client Code

```
1. // udp_client.c
   #include <stdio.h>
```

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>

#define PORT 8080
#define MAX_BUFFER_SIZE 1024

int main() {
    int client_fd;
    struct sockaddr_in server_address;
    char message[MAX_BUFFER_SIZE];

    // Create a UDP socket
    if ((client_fd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        perror("Socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Configure server address
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(PORT);
    if (inet_pton(AF_INET, "127.0.0.1", &server_address.sin_addr) <= 0) {
        perror("Invalid address/ Address not supported");
        exit(EXIT_FAILURE);
    }

    // Client logic here (you can modify this part)

    return 0;
}
```

Step 2: Compiling and Running the Client Code

1. Compile the client code.

   bash

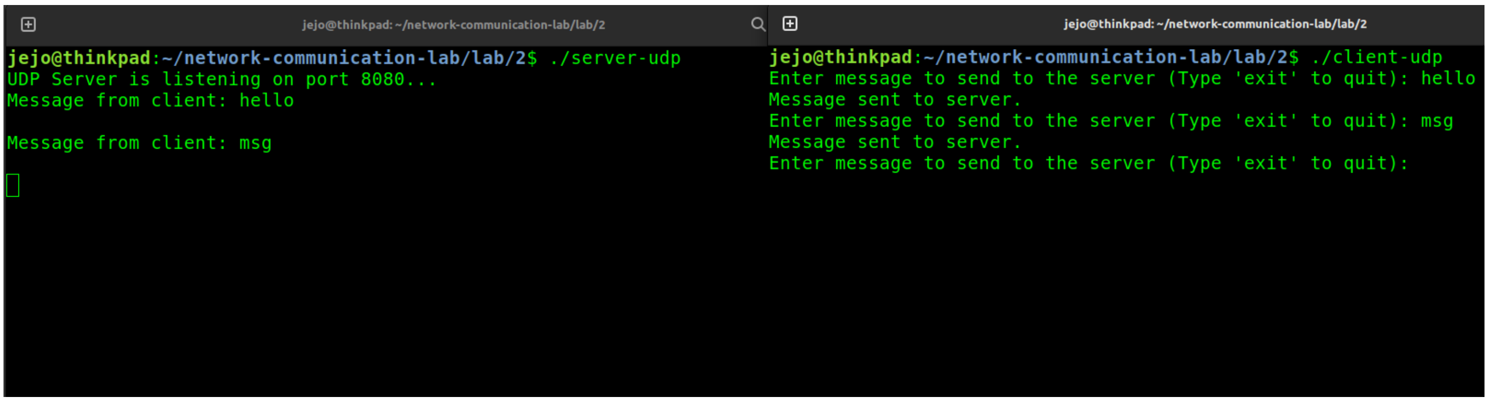
   - `gcc udp_client.c -o udp_client`

- Run the client.

Bash

2. ./udp_client

OUTPUT:



```
jejo@thinkpad:~/network-communication-lab/lab/2$ ./server-udp
UDP Server is listening on port 8080...
Message from client: hello

Message from client: msg
```

```
jejo@thinkpad:~/network-communication-lab/lab/2$ ./client-udp
Enter message to send to the server (Type 'exit' to quit): hello
Message sent to server.
Enter message to send to the server (Type 'exit' to quit): msg
Message sent to server.
Enter message to send to the server (Type 'exit' to quit):
```

3.Network Tools Demonstration

1. Ping:

Command:

```
ping www.google.com
```

---

2. TCPDump:

Command:

```
sudo tcpdump
```

---

3. Traceroute:

Command:

```
traceroute www.google.com
```

---

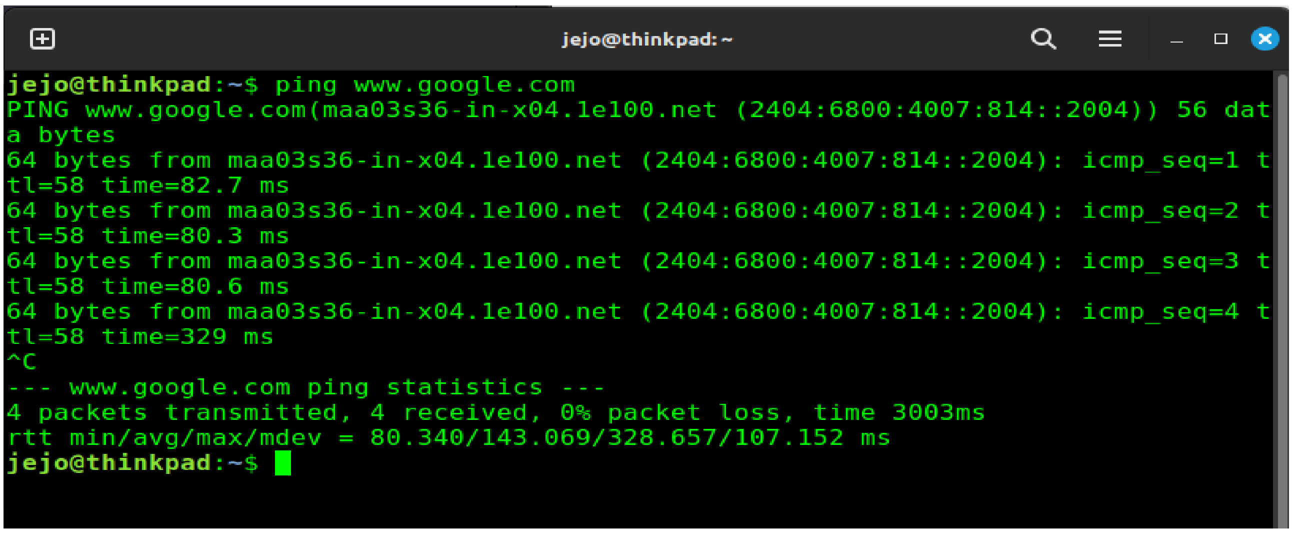
4. Netstat:

Command:

```
netstat -an
```

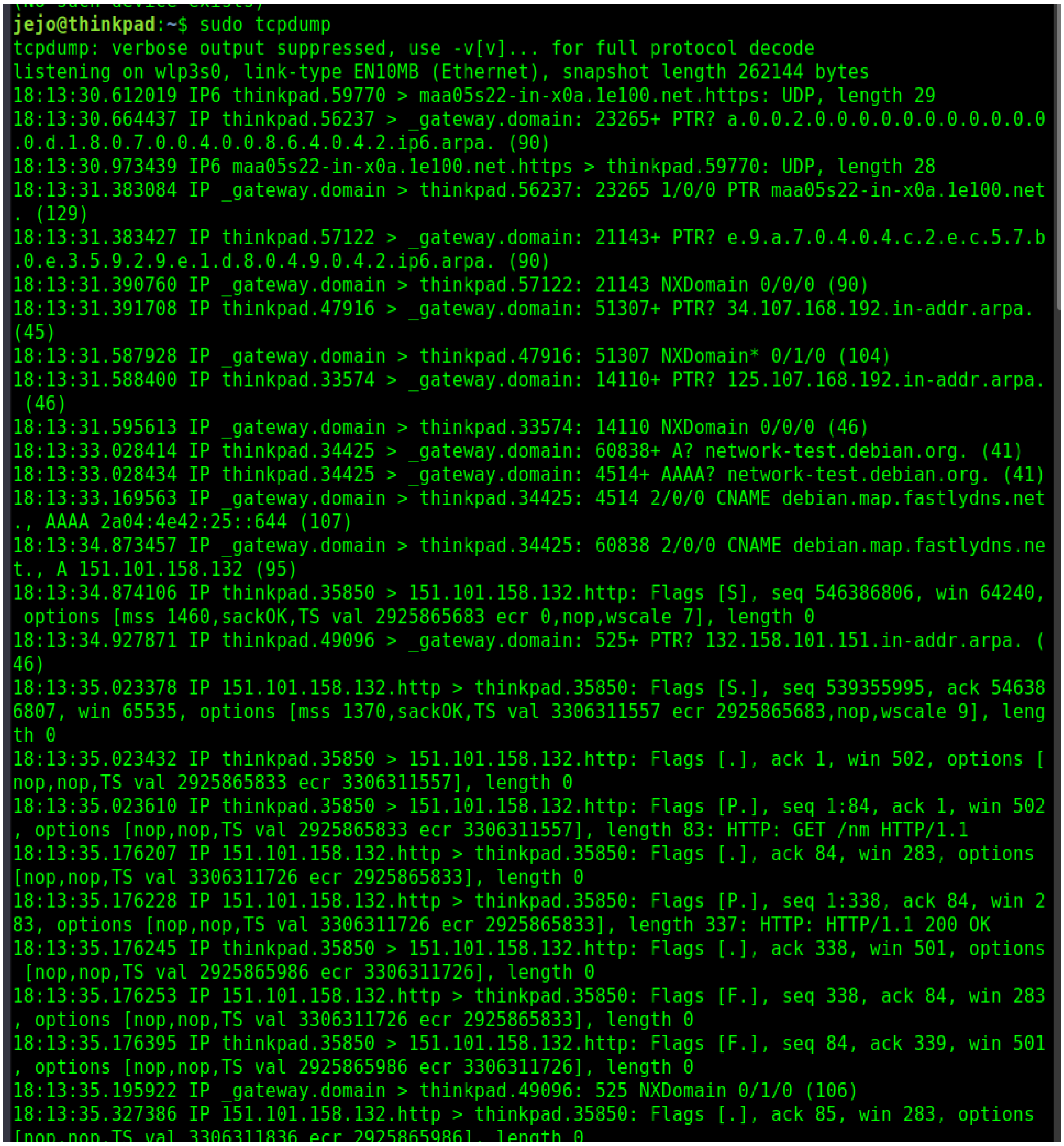
OUTPUT

PING:



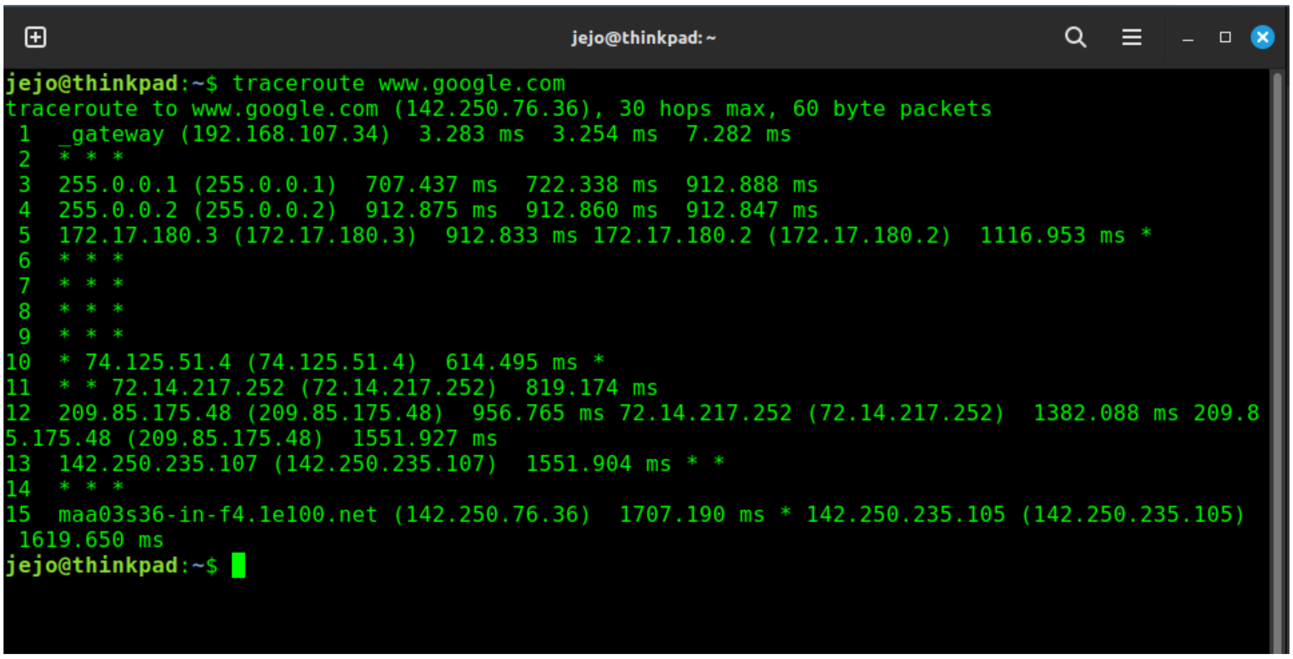
```
jejo@thinkpad:~$ ping www.google.com
PING www.google.com(maa03s36-in-x04.1e100.net (2404:6800:4007:814::2004)) 56 dat
a bytes
64 bytes from maa03s36-in-x04.1e100.net (2404:6800:4007:814::2004): icmp_seq=1 t
tl=58 time=82.7 ms
64 bytes from maa03s36-in-x04.1e100.net (2404:6800:4007:814::2004): icmp_seq=2 t
tl=58 time=80.3 ms
64 bytes from maa03s36-in-x04.1e100.net (2404:6800:4007:814::2004): icmp_seq=3 t
tl=58 time=80.6 ms
64 bytes from maa03s36-in-x04.1e100.net (2404:6800:4007:814::2004): icmp_seq=4 t
tl=58 time=329 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 80.340/143.069/328.657/107.152 ms
jejo@thinkpad:~$
```

TCPDump:



```
(No such device exists)
jejo@thinkpad:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on wlp3s0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:13:30.612019 IP6 thinkpad.59770 > maa05s22-in-x0a.1e100.net.https: UDP, length 29
18:13:30.664437 IP thinkpad.56237 > _gateway.domain: 23265+ PTR? a.0.0.2.0.0.0.0.0.0.0.0.0.0.0
.0.d.1.8.0.7.0.0.4.0.0.8.6.4.0.4.2.ip6.arpa. (90)
18:13:30.973439 IP6 maa05s22-in-x0a.1e100.net.https > thinkpad.59770: UDP, length 28
18:13:31.383084 IP _gateway.domain > thinkpad.56237: 23265 1/0/0 PTR maa05s22-in-x0a.1e100.net
. (129)
18:13:31.383427 IP thinkpad.57122 > _gateway.domain: 21143+ PTR? e.9.a.7.0.4.0.4.c.2.e.c.5.7.b
.0.e.3.5.9.2.9.e.1.d.8.0.4.9.0.4.2.ip6.arpa. (90)
18:13:31.390760 IP _gateway.domain > thinkpad.57122: 21143 NXDomain 0/0/0 (90)
18:13:31.391708 IP thinkpad.47916 > _gateway.domain: 51307+ PTR? 34.107.168.192.in-addr.arpa.
(45)
18:13:31.587928 IP _gateway.domain > thinkpad.47916: 51307 NXDomain* 0/1/0 (104)
18:13:31.588400 IP thinkpad.33574 > _gateway.domain: 14110+ PTR? 125.107.168.192.in-addr.arpa.
 (46)
18:13:31.595613 IP _gateway.domain > thinkpad.33574: 14110 NXDomain 0/0/0 (46)
18:13:33.028414 IP thinkpad.34425 > _gateway.domain: 60838+ A? network-test.debian.org. (41)
18:13:33.028434 IP thinkpad.34425 > _gateway.domain: 4514+ AAAA? network-test.debian.org. (41)
18:13:33.169563 IP _gateway.domain > thinkpad.34425: 4514 2/0/0 CNAME debian.map.fastlydns.net
., AAAA 2a04:4e42:25::644 (107)
18:13:34.873457 IP _gateway.domain > thinkpad.34425: 60838 2/0/0 CNAME debian.map.fastlydns.ne
t., A 151.101.158.132 (95)
18:13:34.874106 IP thinkpad.35850 > 151.101.158.132.http: Flags [S], seq 546386806, win 64240,
 options [mss 1460,sackOK,TS val 2925865683 ecr 0,nop,wscale 7], length 0
18:13:34.927871 IP thinkpad.49096 > _gateway.domain: 525+ PTR? 132.158.101.151.in-addr.arpa. (
46)
18:13:35.023378 IP 151.101.158.132.http > thinkpad.35850: Flags [S.], seq 539355995, ack 54638
6807, win 65535, options [mss 1370,sackOK,TS val 3306311557 ecr 2925865683,nop,wscale 9], leng
th 0
18:13:35.023432 IP thinkpad.35850 > 151.101.158.132.http: Flags [.], ack 1, win 502, options [
nop,nop,TS val 2925865833 ecr 3306311557], length 0
18:13:35.023610 IP thinkpad.35850 > 151.101.158.132.http: Flags [P.], seq 1:84, ack 1, win 502
, options [nop,nop,TS val 2925865833 ecr 3306311557], length 83: HTTP: GET /nm HTTP/1.1
18:13:35.176207 IP 151.101.158.132.http > thinkpad.35850: Flags [.], ack 84, win 283, options
[nop,nop,TS val 3306311726 ecr 2925865833], length 0
18:13:35.176228 IP 151.101.158.132.http > thinkpad.35850: Flags [P.], seq 1:338, ack 84, win 2
83, options [nop,nop,TS val 3306311726 ecr 2925865833], length 337: HTTP: HTTP/1.1 200 OK
18:13:35.176245 IP thinkpad.35850 > 151.101.158.132.http: Flags [.], ack 338, win 501, options
 [nop,nop,TS val 2925865986 ecr 3306311726], length 0
18:13:35.176253 IP 151.101.158.132.http > thinkpad.35850: Flags [F.], seq 338, ack 84, win 283
, options [nop,nop,TS val 3306311726 ecr 2925865833], length 0
18:13:35.176395 IP thinkpad.35850 > 151.101.158.132.http: Flags [F.], seq 84, ack 339, win 501
, options [nop,nop,TS val 2925865986 ecr 3306311726], length 0
18:13:35.195922 IP _gateway.domain > thinkpad.49096: 525 NXDomain 0/1/0 (106)
18:13:35.327386 IP 151.101.158.132.http > thinkpad.35850: Flags [.], ack 85, win 283, options
[nop,nop,TS val 3306311836 ecr 2925865986], length 0
```

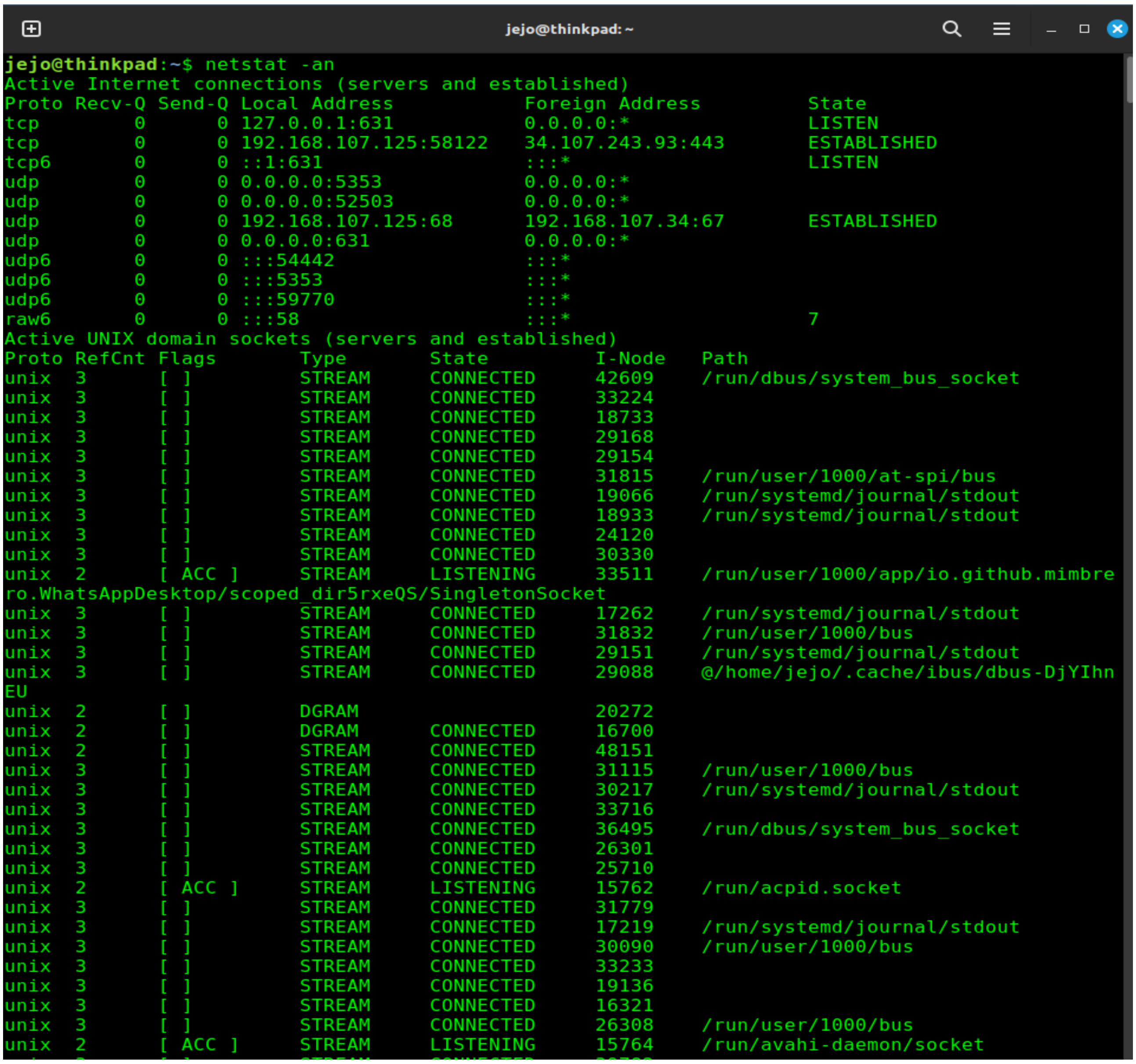
Traceroute:



```
jejo@thinkpad:~$ traceroute www.google.com
traceroute to www.google.com (142.250.76.36), 30 hops max, 60 byte packets
 1  _gateway (192.168.107.34)  3.283 ms  3.254 ms  7.282 ms
 2  * * *
 3  255.0.0.1 (255.0.0.1)  707.437 ms  722.338 ms  912.888 ms
 4  255.0.0.2 (255.0.0.2)  912.875 ms  912.860 ms  912.847 ms
 5  172.17.180.3 (172.17.180.3)  912.833 ms 172.17.180.2 (172.17.180.2)  1116.953 ms *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * 74.125.51.4 (74.125.51.4)  614.495 ms *
11  * * 72.14.217.252 (72.14.217.252)  819.174 ms
12  209.85.175.48 (209.85.175.48)  956.765 ms 72.14.217.252 (72.14.217.252)  1382.088 ms 209.8
5.175.48 (209.85.175.48)  1551.927 ms
13  142.250.235.107 (142.250.235.107)  1551.904 ms * *
14  * * *
15  maa03s36-in-f4.1e100.net (142.250.76.36)  1707.190 ms * 142.250.235.105 (142.250.235.105)
 1619.650 ms
jejo@thinkpad:~$ 
```

.

NETSTAT:



```
jejo@thinkpad:~$ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.107.125:58122   34.107.243.93:443       ESTABLISHED
tcp6       0      0 ::1:631                 :::*                    LISTEN
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp        0      0 0.0.0.0:52503           0.0.0.0:*
udp        0      0 192.168.107.125:68      192.168.107.34:67       ESTABLISHED
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp6       0      0 :::54442                :::*
udp6       0      0 :::5353                 :::*
udp6       0      0 :::59770                :::*
raw6       0      0 :::58                   :::*                    7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         STREAM     CONNECTED     42609    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     33224
unix  3      [ ]         STREAM     CONNECTED     18733
unix  3      [ ]         STREAM     CONNECTED     29168
unix  3      [ ]         STREAM     CONNECTED     29154
unix  3      [ ]         STREAM     CONNECTED     31815    /run/user/1000/at-spi/bus
unix  3      [ ]         STREAM     CONNECTED     19066    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     18933    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     24120
unix  3      [ ]         STREAM     CONNECTED     30330
unix  2      [ ACC ]     STREAM     LISTENING     33511    /run/user/1000/app/io.github.mimbre
ro.WhatsAppDesktop/scoped_dir5rxeQS/SingletonSocket
unix  3      [ ]         STREAM     CONNECTED     17262    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     31832    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     29151    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     29088    @/home/jejo/.cache/ibus/dbus-DjYIhn
EU
unix  2      [ ]         DGRAM                    20272
unix  2      [ ]         DGRAM      CONNECTED     16700
unix  2      [ ]         STREAM     CONNECTED     48151
unix  3      [ ]         STREAM     CONNECTED     31115    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     30217    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     33716
unix  3      [ ]         STREAM     CONNECTED     36495    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     26301
unix  3      [ ]         STREAM     CONNECTED     25710
unix  2      [ ACC ]     STREAM     LISTENING     15762    /run/acpid.socket
unix  3      [ ]         STREAM     CONNECTED     31779
unix  3      [ ]         STREAM     CONNECTED     17219    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     30090    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     33233
unix  3      [ ]         STREAM     CONNECTED     19136
unix  3      [ ]         STREAM     CONNECTED     16321
unix  3      [ ]         STREAM     CONNECTED     26308    /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     15764    /run/avahi-daemon/socket
```