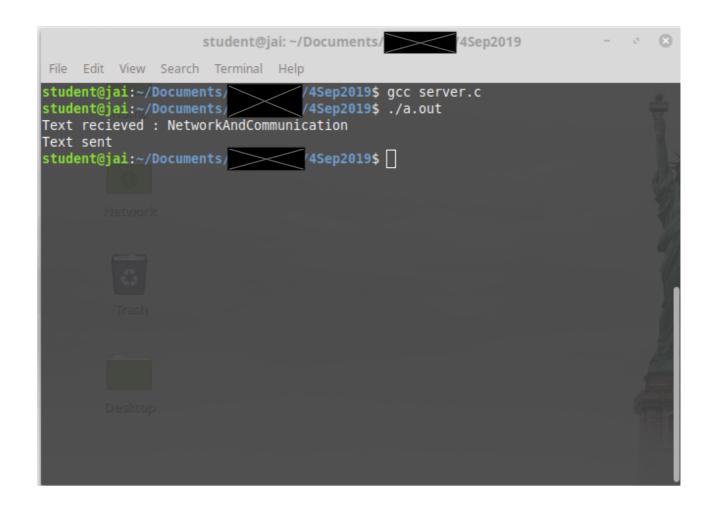
Wireshark: Using TCP:

Server:

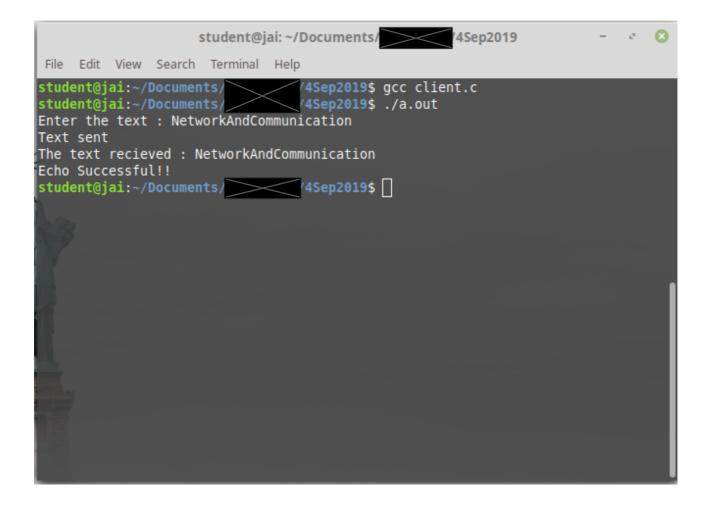
```
#include <unistd.h>
#include <stdio.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <string.h>
#define PORT 8080
int main(int argc, char const *argv[])
  int server fd, new socket, valread;
  struct sockaddr in address;
  int opt = 1;
  int addrlen = sizeof(address);
  // Creating socket file descriptor
  if ((server fd = socket(AF INET, SOCK STREAM, 0)) == 0)
     perror("socket failed");
     exit(EXIT FAILURE);
  }
  // Forcefully attaching socket to the port 8080
  if (setsockopt(server fd, SOL SOCKET, SO REUSEADDR | SO REUSEPORT,
                               &opt, sizeof(opt)))
  {
     perror("setsockopt");
     exit(EXIT_FAILURE);
  address.sin family = AF INET;
  address.sin addr.s addr = INADDR ANY;
  address.sin port = htons( PORT );
  // Forcefully attaching socket to the port 8080
  if (bind(server_fd, (struct sockaddr *)&address,
                    sizeof(address))<0)
  {
     perror("bind failed");
     exit(EXIT FAILURE);
  }
```

```
if (listen(server_fd, 3) < 0)
     perror("listen");
     exit(EXIT_FAILURE);
  if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
               (socklen t*)&addrlen))<0)
  {
     perror("accept");
     exit(EXIT_FAILURE);
  }
  char stri[1024];
  valread = read( new_socket , stri, sizeof(stri));
  printf("Text recieved : %s\n", stri);
  send(new_socket , stri , sizeof(stri), 0);
  printf("Text sent\n");
  return 0;
}
```



```
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <string.h>
#define PORT 8080
int main(int argc, char const *argv[])
  int sock = 0, valread;
  struct sockaddr in serv addr;
  if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0)
     printf("\n Socket creation error \n");
     return -1;
  }
  serv addr.sin family = AF INET;
  serv addr.sin port = htons(PORT);
  // Convert IPv4 and IPv6 addresses from text to binary form
  if(inet pton(AF INET, "127.0.0.1", &serv addr.sin addr)<=0)
     printf("\nInvalid address/ Address not supported \n");
     return -1;
  }
  if (connect(sock, (struct sockaddr *)&serv addr, sizeof(serv addr)) < 0)
     printf("\nConnection Failed \n");
     return -1;
  }
  printf("Enter the text : ");
  char stri[1024];
  scanf("%s", stri);
  send(sock, stri, sizeof(stri), 0);
  printf("Text sent\n");
  char res[1024];
  valread = read( sock , res, sizeof(res));
  printf("The text recieved : %s\n", res);
  int i=0;
  if(strlen(stri)!=strlen(res))
```

```
printf("Echo Unuccessful!!\n");
}
else
{
  int flag = 1;
  for(i=0;i<strlen(res);i++)
    {
      if(res[i]!=stri[i])
      {
          flag=0;
          break;
      }
      }
      if(flag)
      {
          printf("Echo Successful!!\n");
      }
      else
      {
          printf("Echo Unuccessful!!\n");
      }
}
return 0;</pre>
```



Wireshark

No.	Time	Source	Destination	Protocol	Length	Info	
Г	1 0.000000000	127.0.0.1	127.0.0.1	TCP	74	1 56088 → 7777 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=3376507505 TSecr=0
	2 0.000022659	127.0.0.1	127.0.0.1	TCP	74	1 7777 → 56088 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=337650
	3 0.000045509	127.0.0.1	127.0.0.1	TCP	66	5 56088 → 7777 [.	ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=3376507505 TSecr=3376507505
	4 2.358158529	127.0.0.1	127.0.0.1	TCP	1090	9 56088 → 7777 [PSH, ACK] Seq=1 Ack=1 Win=43776 Len=1024 TSval=3376509864 TSecr=33765075(
	5 2.358186252	127.0.0.1	127.0.0.1	TCP	66	5 7777 → 56088 [.	ACK] Seq=1 Ack=1025 Win=45824 Len=0 TSval=3376509864 TSecr=3376509864
	6 2.358435131	127.0.0.1	127.0.0.1	TCP	1090	7777 → 56088 [PSH, ACK] Seq=1 Ack=1025 Win=45824 Len=1024 TSval=3376509864 TSecr=337650
	7 2.358450749	127.0.0.1	127.0.0.1	TCP	66	5 56088 → 7777 [ACK] Seq=1025 Ack=1025 Win=45824 Len=0 TSval=3376509864 TSecr=3376509864
	8 2.358647223	127.0.0.1	127.0.0.1	TCP	66	5 7777 → 56088 [FIN, ACK] Seq=1025 Ack=1025 Win=45824 Len=0 TSval=3376509864 TSecr=337650
	9 2.358806110	127.0.0.1	127.0.0.1	TCP	66	5 56088 → 7777 [FIN, ACK] Seq=1025 Ack=1026 Win=45824 Len=0 TSval=3376509864 TSecr=337650
L :	10 2.358826409	127.0.0.1	127.0.0.1	TCP	66	5 7777 → 56088 [ACK] Seq=1026 Ack=1026 Win=45824 Len=0 TSval=3376509864 TSecr=3376509864

- Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 Ethernet II, Src: 00:00:00:00:00:00:00:00:00:00:00:00), Dst: 00:00:00:00:00:00:00:00:00:00:00:00
 Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1, Dst: 127.0.0.1
 Transmission Control Protocol, Src Port: 56088, Dst Port: 7777, Seq: 0, Len: 0

