**A5-Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.**

**SERVER:**

**#include<stdio.h>**

**#include<string.h>**

**#include<stdlib.h>**

**#include<sys/socket.h>**

**#include<sys/types.h>**

**#include<arpa/inet.h>**

**#include<netinet/in.h>**

**#include<netdb.h>**

**#include<unistd.h>**

**int main()**

**{**

**//Variables**

**FILE \*fp;**

**int sock,pt,cnt=0;**

**char filename[1024],buf\_recv[1024],buf\_send[1024];**

**struct sockaddr\_in server,client;**

**int slen = sizeof(client);**

**//Creating a Socket**

**sock = socket(AF\_INET,SOCK\_DGRAM,0);**

**if(sock == -1)**

**{**

**perror("Socket Error");**

**exit(1);**

**}**

**server.sin\_family = AF\_INET;**

**server.sin\_addr.s\_addr = INADDR\_ANY;**

**server.sin\_port = htons(8888);**

**//Binding socket to IP and Port**

**if(bind(sock, (struct sockaddr \*)&server, sizeof(server)) == -1)**

**{**

**perror("Bind Error");**

**exit(1);**

**}**

**while(1)**

**{**

**cnt=0;**

**fflush(stdout);**

**pt = recvfrom(sock, filename, sizeof(filename), 0, (struct sockaddr \*)&client, &slen);**

**printf("%s",filename);**

**//strcat(filename,"1");**

**fp=fopen(filename, "w");**

**//Initialize filedata with some random value**

**strcpy(buf\_recv,"random");**

**//Receiving file data in packets till end of file**

**while(strcmp(buf\_recv,"end") != 0)**

**{**

**//Receiving File data**

**/\*\*\*\*\*\*\*\*\*\*\*Define recvfrom in buf\_recv Here\*\*\*\*\*\*\*\*\*\*/**

**recvfrom(sock, buf\_recv, sizeof(buf\_recv), 0, (struct sockaddr \*)&client,**

**&slen);**

**//Writing received file data**

**/\*\*\*\*\*\*\*\*\*\*\*Define fwrite here Here\*\*\*\*\*\*\*\*\*\*/**

**if(strcmp(buf\_recv,"end") != 0)**

**{**

**fprintf(fp,buf\_recv);**

**}**

**printf("Received:%d\n",cnt);**

**cnt++;**

**}**

**//Closing the file**

**fclose(fp);**

**printf("File Received Successfully\n");**

**}**

**}**

**CLIENT:**

**#include<stdio.h>**

**#include<string.h>**

**#include<stdlib.h>**

**#include<sys/socket.h>**

**#include<sys/types.h>**

**#include<arpa/inet.h>**

**#include<netinet/in.h>**

**#include<netdb.h>**

**#include<unistd.h>**

**//Check if the File Exists**

**int exists(const char \*fname)**

**{**

**FILE \*file;**

**if(file = fopen(fname, "r"))**

**{**

**fclose(file);**

**return 1;**

**}**

**return 0;**

**}**

**int main(int argc, char\* argv[])**

**{**

**//Variables**

**FILE \*fp;**

**int sock,pt;**

**char filename[1024],buf\_recv[1024],buf\_send[1024];**

**struct sockaddr\_in server;**

**struct hostent \*host;**

**int slen = sizeof(server);**

**//Taking the Commanf Line Argument of IP address**

**host = gethostbyname(argv[1]);**

**//Creating a Socket**

**sock = socket(AF\_INET,SOCK\_DGRAM,0);**

**if(sock == -1)**

**{**

**perror("Socket Error");**

**exit(1);**

**}**

**server.sin\_family = AF\_INET;**

**server.sin\_port = htons(8888);**

**memcpy(&server.sin\_addr,host->h\_addr,host->h\_length);**

**//Scanning the File Name**

**printf("Enter Filename: ");**

**scanf("%s",filename);**

**//Check if the File Exists**

**if(exists(filename))**

**{**

**//Sending the FileName to the server**

**sendto(sock, filename, sizeof(filename), 0, (struct sockaddr \*)&server, slen);**

**//Opening the file in read mode**

**fp=fopen(filename, "rb");**

**usleep(100000);**

**//Reading the file in chunks**

**while (fread(buf\_send, strlen(buf\_send)+1, 1, fp) == 1)**

**{**

**//Sending the read file chunk to the server**

**/\*\*\*\*\*\*\*\*\*\*\*Define sendto Here\*\*\*\*\*\*\*\*\*\*/**

**sendto(sock, buf\_send, sizeof(buf\_send), 0, (struct sockaddr \*)&server, slen);**

**usleep(1000);**

**}**

**usleep(100000);**

**//Checking End of File**

**if (feof(fp))**

**{**

**//When end of file, sending the last chunk of data**

**/\*\*\*\*\*\*\*\*\*\*\*Define sendto Here\*\*\*\*\*\*\*\*\*\*/**

**sendto(sock, buf\_send, sizeof(buf\_send), 0, (struct sockaddr \*)&server, slen);**

**//Sending "end" as data in last packet indicating end of file data**

**strcpy(buf\_send,"end");**

**/\*\*\*\*\*\*\*\*\*\*\*Define sendto Here\*\*\*\*\*\*\*\*\*\*/**

**sendto(sock, buf\_send, sizeof(buf\_send), 0, (struct sockaddr \*)&server, slen);**

**printf("File written successfully\n");**

**}**

**else**

**{**

**printf("File not read/written successfully\n");**

**}**

**//Closing the file**

**fclose(fp);**

**}**

**else**

**{**

**printf("File Doesn't Exist\n");**

**}**

**}**

**OUTPUT:**

**SERVER:**

[oopl@localhost ~]$ vi sserv.c

[oopl@localhost ~]$ gcc -o sserv sserv.c

[oopl@localhost ~]$ ./sserv

vv.txtReceived:0

Received:1

File Received Successfully

**CLIENT:**

[oopl@localhost ~]$ vi ccli.c

[oopl@localhost ~]$ gcc -o ccli ccli.c

[oopl@localhost ~]$ ./ccli 172.168.255.89

Enter Filename: vv.txt

File written successfully

