

COMPUTER NETWORKING REPORT

Title: File Transfer Protocol

Submitted By

Prajakta Yadav(321)

Samruddhi Yadav(322)

DEPARTMENT OF COMPUTER ENGINEERING

MIT ACADEMY OF ENGINEERING

ALANDI (D), PUNE

INDEX

Abstract		i
1.	Introduction	1
2.	Literature Survey	3-5
3.	System Design	6-7
	3.1 UML Daigrams	6
	3.2 Module coding	6
4.	Screenshots	8-10
7.	Results	11
8.	References	13

INTRODUCTION

FTP powers one of the fundamental Internet functions and is the prescribed method for the transfer of files between computers. It is also the easiest and most secure way to exchange files over the Internet. The File Transfer Protocol is one of the oldest and widely used protocols on the Internet. The protocol's primary character set, bit ASCII, has served the protocol well through the early growth years of the Internet. However, as the Internet becomes more global, there is a need to support character sets beyond 7 bit ASCII. This document addresses the internationalization (I18n) of FTP, which includes supporting the multiple character sets and languages found throughout the Internet community. This is achieved by extending the FTP specification and giving recommendations for proper internationalization support.

OBJECTIVES

1. To promote sharing of files (computer programs and/or data),
2. To encourage indirect or implicit (via programs) use of remote computers,
3. To shield a user from variations in file storage systems among hosts, and
4. To transfer data reliably and efficiently.

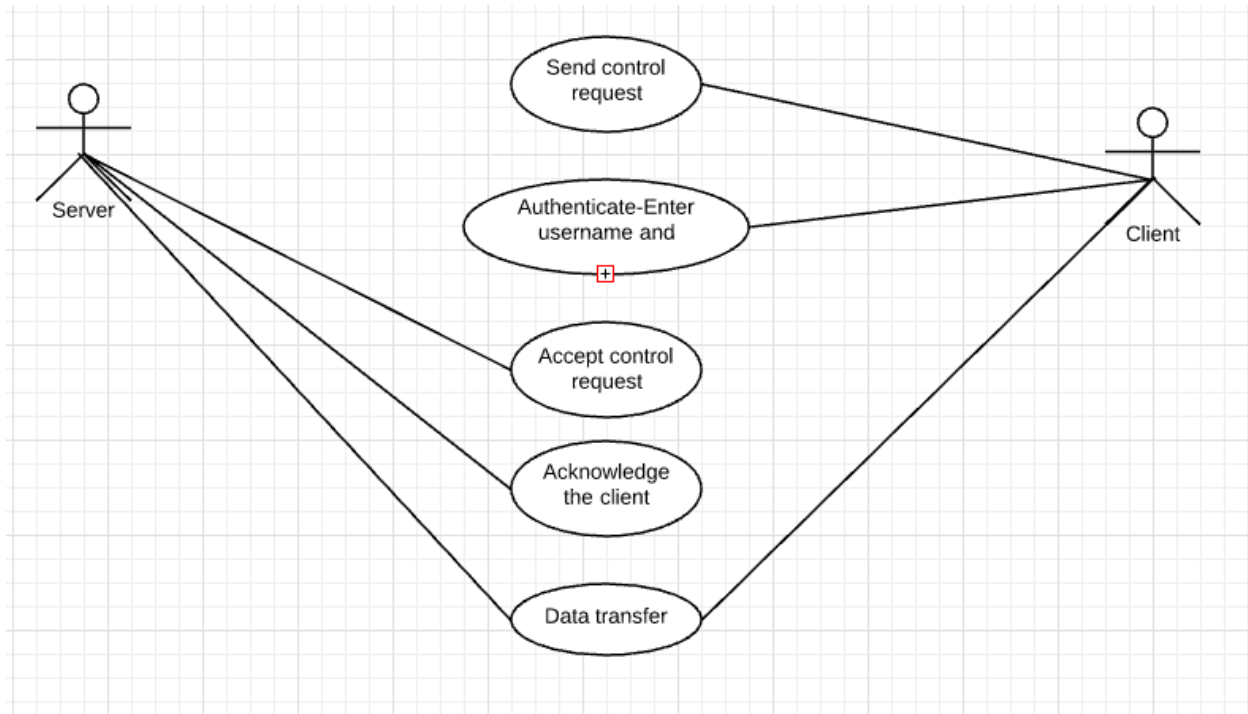
LITERATURE SURVEY

This paper presents a File Transfer Protocol (FTP) which can be used to transfer files between heterogeneous computer systems connected by a communication network. The basic mechanisms as well as the set of protocol commands and responses have been defined in the context of a general architecture. A simplified architecture, to help first implementations is also proposed. This can easily be extended to the more general one. The proposed protocol is completely compatible (even identical in some areas) with the currently proposed Virtual Terminal Protocol (VTP).

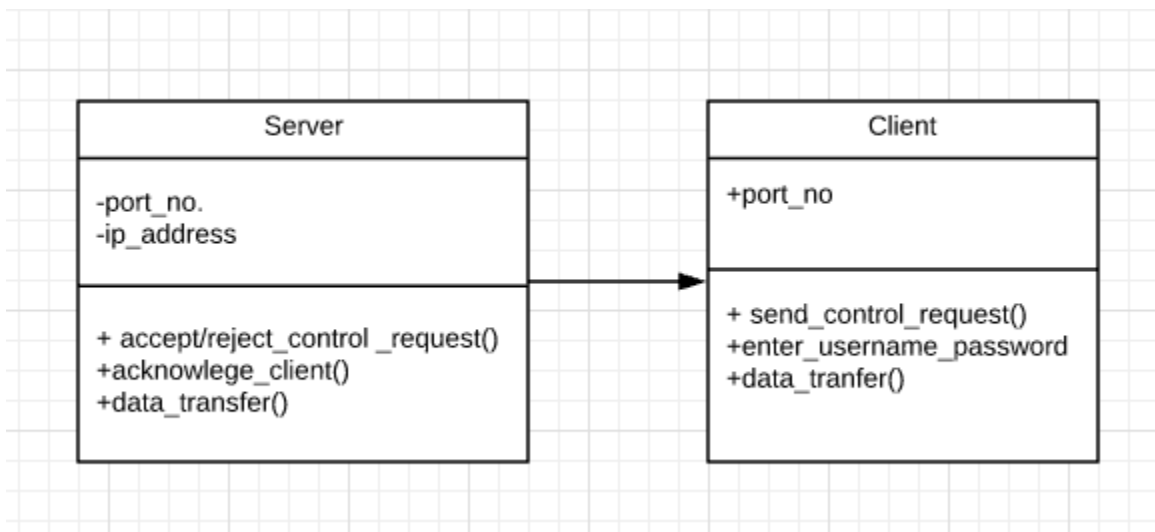
FTP is client-server architecture based protocol for transferring files over TCP/IP networks. It is used with either user-based password authentication or anonymous user access. FTP has no encryption support. All transmissions are in clear text format. User names, passwords, FTP commands and transferred files can be read by anyone who is sniffing the network. FTPS and FTP over IPsec are cost effective methods of securing FTP. These methods provide much better authentication and encryption functionalities for file transfer communication. Besides this, these methods introduce processing overhead and packet overhead for file transferring to different extends. This research paper compares and contrasts the overhead of secure file transfer methods -FTPS and FTP over IPsec

SYSTEM DESIGN

Usecase Diagram:



Class Diagram:



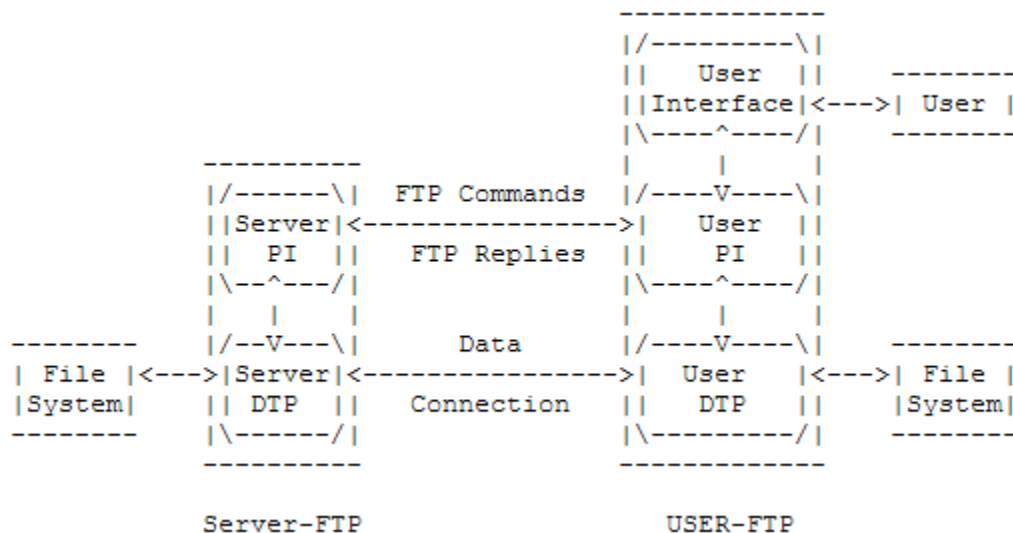
MODULE CODING

Server Side source code Programming :

```
#include<string.h>
#include<sys/ioctl.h>
#include<arpa/inet.h>
#include<stdlib.h>
#include<stdio.h>
#include<net/if_arp.h>
.int main()
{
int sd,b,cd;
struct fname[50],op[1000];
struct sockaddr_in caddr,saddr;
FILE *fp;
socklen_t clen=sizeof(caddr);
sd=socket(AF_INET,SOCK_STREAM,0);
if(sd!=-1)
printf("socket is created");
else
printf("socket is not created");
saddr.sin_family=AF_INET;
saddr.sin_port=htons(2500);
saddr.sin_addr.s_addr=htonl(INADDR_ANY);
b=bind(sd,(struct sockaddr*)&saddr,sizeof(saddr));
if(b==0)
printf("binded successfully");
else
printf("binding failed");
listen(sd,5);
cd=accept(sd,(struct sockaddr*)&caddr,&clen);
recv(cd,fname,sizeof(fname),0);
fp=open(fname,"w");
fwrite(op,strlen(op),1,fp);
printf("the file has been transferred");
close(fd);
close(cd);
fclose(fp);
return 0;
}
```

Client Side Program:

```
#include<string.h>
#include<sys/ioctl.h>
#include<arpa/inet.h>
#include<stdlib.h>
#include<stdio.h>
#include<net/if_arp.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<netdb,h>
int main()
{
int sd,c,s;
char fname[50],sip[25],op[1000];
struct sockaddr_in caddr;
struct hostent *he;
FILE *fp;
printf('enter the server ip address');
scanf("%s",sip);
he=gethostbyname(sip);
sd=socket(AF_INET,SOCK_STREAM,0);
if(sd!=1)
printf("socket created");
else
printf("socket is not created");
caddr.sin_family=AF_INET;
caddr.sin_port=htons(2500);
caddr.sin_addr=*((struct in_addr*)he->h_addr);
c=connect(sd,(struct sockaddr*)&caddr,sizeof(caddr));
if(c==0)
printf("connected to server");
else
printf("connection failed");
printf("enter the file name");
scanf("%s",fname);
send(sd,fname,sizeof(fname),0);
fp=fopen(fname,"r");
fopen(op,1000,1,fp);
send(sd,op,sizeof(op),0);
fclose(fp);
close(sd);
return 0;
}
```

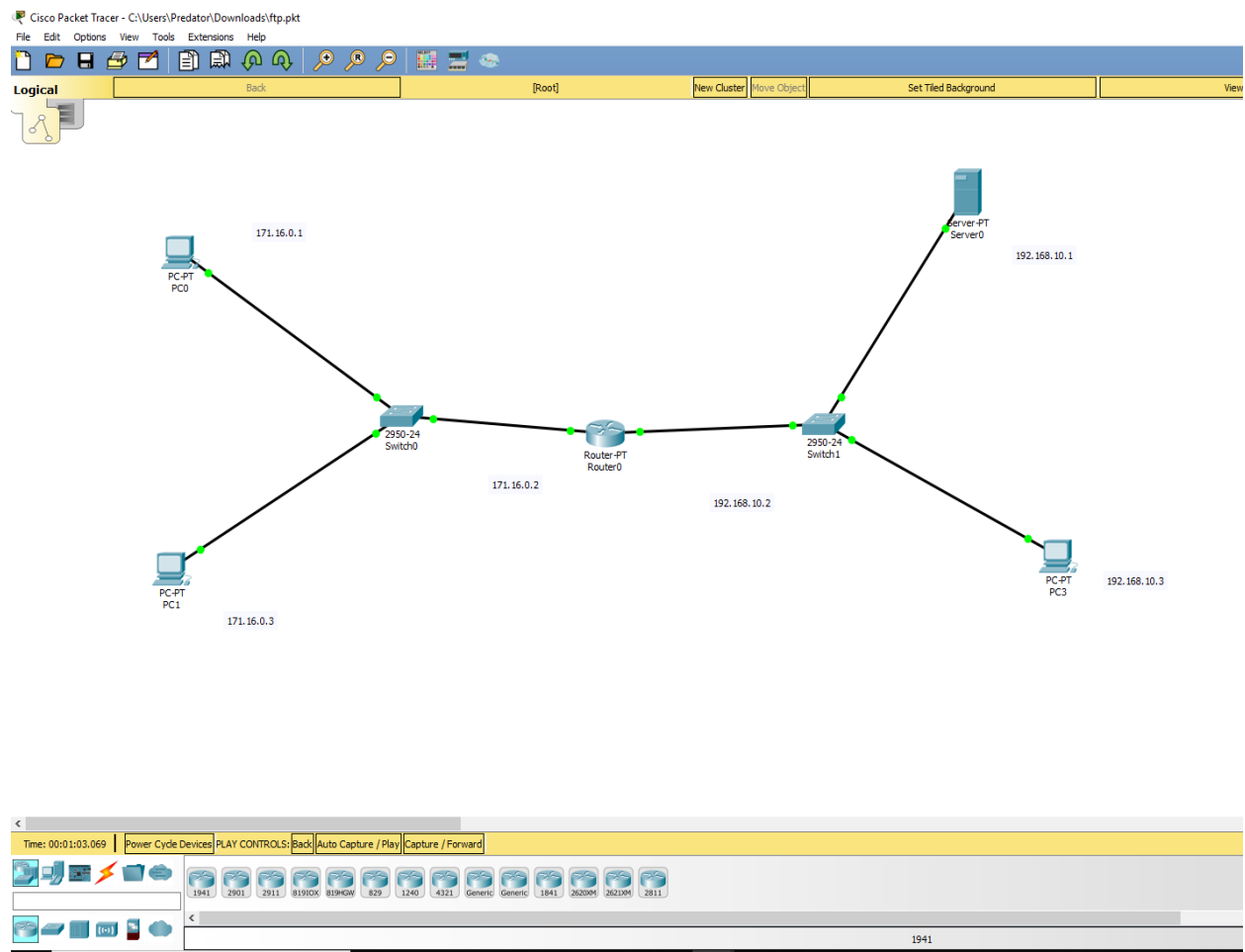


The user-protocol interpreter initiates the control connection. The control connection follows the Telnet protocol. At the initiation of the user, standard FTP commands are generated by the user-PI and transmitted to the server process via the control connection. (The user may establish a direct control connection to the server-FTP, from a TAC terminal for example, and generate standard FTP commands independently, bypassing the user-FTP process.) Standard replies are sent from the server-PI to the user-PI over the control connection in response to the commands.

The FTP commands specify the parameters for the data connection (data port, transfer mode, representation type, and structure) and the nature of file system operation (store, retrieve, append, delete, etc.). The user-DTP or its designate should "listen" on the specified data port, and the server initiate the data connection and data transfer in accordance with the specified parameters. It should be noted that the data port need not be in the same host that initiates the FTP commands via the control connection, but the user or the user-FTP process must ensure a "listen" on the specified data port. It ought to also be noted that the data connection may be used for simultaneous sending and receiving.

In another situation a user might wish to transfer files between two hosts, neither of which is a local host. The user sets up control connections to the two servers and then arranges for a data connection between them. In this manner, control information is passed to the user-PI but data is transferred between the server data transfer processes. Following is a model of this server-server interaction.

SCREENSHOTS



RESULT

```
Packet Tracer PC Command Line 1.0
PC>dir

Volume in drive C has no label.
Volume Serial Number is 5E12-4AF3
Directory of C:\
1/1/1970  5:30 PM      2          praju.txt
1/1/1970  5:30 PM      5          s.txt
2/7/2106  11:58 PM     26        sampleFile.txt
                                     33 bytes      3 File(s)

PC>ftp 172.168.1.9
Trying to connect...172.168.1.9
Connected to 172.168.1.9
220- Welcome to FT Ftp server
Username:abcd
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put s.txt

Writing file s.txt to 172.168.1.9:
File transfer in progress...

[Transfer complete - 5 bytes]

5 bytes copied in 0.047 secs (106 bytes/sec)
ftp>dir

Listing /ftp directory from 172.168.1.9:
0 : c1841-advipservicesk9-mz.124-15.T1.bin      33591768
1 : c1841-ibase-mz.123-14.T7.bin                13832032
2 : c1841-ibasek9-mz.124-12.bin                 16599160
3 : c2600-advipservicesk9-mz.124-15.T1.bin      33591768
4 : c2600-i-mz.122-28.bin                       5571584
5 : c2600-ibasek9-mz.124-8.bin                  13169700
6 : c2800nm-advipservicesk9-mz.124-15.T1.bin    50938004
7 : c2800nm-advipservicesk9-mz.151-4.M4.bin     33591768
8 : c2800nm-ibase-mz.123-14.T7.bin              5571584
9 : c2800nm-ibasek9-mz.124-8.bin                15522644
10 : c2950-i6q412-mz.121-22.EA4.bin            3058048
11 : c2950-i6q412-mz.121-22.EA8.bin            3117330
```

```
Packet Tracer PC Command Line 1.0
PC>dir

Volume in drive C has no label.
Volume Serial Number is 5E12-4AF3
Directory of C:\
1/1/1970  5:30 PM      2          praju.txt
1/1/1970  5:30 PM      5          s.txt
2/7/2106  11:58 PM     26        sampleFile.txt
                                     33 bytes      3 File(s)

PC>ftp 172.168.1.9
Trying to connect...172.168.1.9
Connected to 172.168.1.9
220- Welcome to FT Ftp server
Username:abcd
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>get s.txt

Reading file s.txt from 172.168.1.9:
File transfer in progress...

[Transfer complete - 5 bytes]

5 bytes copied in 0.01 secs (500 bytes/sec)
ftp>dir

Listing /ftp directory from 172.168.1.9:
0 : c1841-advipservicesk9-mz.124-15.T1.bin      33591768
1 : c1841-ibase-mz.123-14.T7.bin                13832032
2 : c1841-ibasek9-mz.124-12.bin                 16599160
3 : c2600-advipservicesk9-mz.124-15.T1.bin      33591768
4 : c2600-i-mz.122-28.bin                       5571584
5 : c2600-ibasek9-mz.124-8.bin                  13169700
6 : c2800nm-advipservicesk9-mz.124-15.T1.bin    50938004
7 : c2800nm-advipservicesk9-mz.151-4.M4.bin     33591768
8 : c2800nm-ibase-mz.123-14.T7.bin              5571584
9 : c2800nm-ibasek9-mz.124-8.bin                15522644
10 : c2950-i6q412-mz.121-22.EA4.bin            3058048
11 : c2950-i6q412-mz.121-22.EA8.bin            3117330
```

REFERENCES

[1] Pradeep Ruwan Nawarathne, International Journal of Scientific & Engineering Research, Volume 3, Issue 11, November-2012

[2] https://www.researchgate.net/publication/220446268_A_File_Transfer_Protocol_FTP

