Telnet e FTP

Risorse su Telnet

Risorse su FTP

Risorse su TFTP

Risorse su SMTP

Telnet

Il programma Telnet (telecommunications network) serve a fornire un login remoto o capacità di terminale virtuale all'interno di una rete. In altre parole, un utente su una macchina A dovrebbe essere capace di collegarsi ad una macchina B in un punto qualsiasi della rete, e per quanto gli concerne egli deve avere l'impressione di trovarsi

seduto davanti alla macchina B. il servizio Telnet è fornito attraverso la porta TCP numero 23, il termine Telnet è usato per indicare sia il programma che il protocollo che forniscono questo servizio.

Telnet fu sviluppato perché a suo tempo l'unico metodo per abilitare una macchina ad accedere alle risorse di un'altra macchina (inclusi dischi e programmi) era di stabilire un link usando sistemi di comunicazione come modem e reti in porte seriali dedicate o adattatori di rete. Ciò è un tantino più complicato di quanto possa apparire a prima vista a causa dell'ampia diversità di computer e terminali . quando direttamente collegata ad un'altra macchina, la CPU deve effettuare le traduzioni di codici di terminale fra i due , il che significa un carico notevole per la CPU;: con diversi login remoti attivi, la CPU di una macchina può spendere uno smodato ammontare di tempo gestendo la traduzione. Questo è in particolare un problema con i server che possono gestire molte connessioni contemporaneamente.

Telnet allevia questi problemi incastonando le sequenze caratteristiche dei terminali all'interno del protocollo. Quando due macchine comunicano usando Telnet, Telnet stesso può determinare e settare le comunicazioni ed i parametri di terminale per la sessione durante la fase di connessione. Il protocollo telnet include la capacità di non supportare un servizio che un'estremità di una connessione non è n grato di gestire. Quando una connessione è stata stabilita mediante Telnet, entrambe le estremità si sono messe d'accordo su un metodo per scambiare informazioni fra le due macchine, togliendo il carico dalla CPU per un ammontare importante.

Usualmente, Telnet coinvolge un processo sul server che accetta richieste in ingresso per una sessione Telnet. Su sistemi Unix[1] questo processo è chiamato telnetd[2]: su sistemi windows NT ed altri sistemi operativi basati su PC , è comunemente coinvolto un programma Telnet Server. Il client (l'estremità che ha fatto la chiamata) fa girare un programma , usualmente chiamato telnet , che tenta la connessione al server. In relazione con il programma telnet vi è il programma login, comune sulle macchine Unix.

Il programma login fornisce prestazioni praticamente identiche a quelle di telnet e aggiunge supporto per l'ambiente Unix. Molte macchine, specialmente stazioni Unix, agiscono contemporaneamente da client e server, abilitando un utente a connettersi ad altre macchine sulla rete ed altri utenti a collegarsi alla propria macchina.

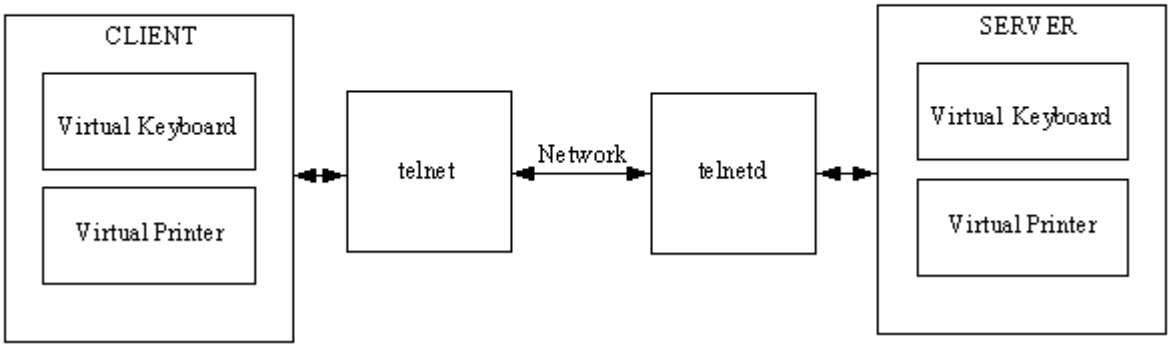
Telnet connections

Il protocollo Telnet usa il concetto di terminale virtuale di rete , o NVT (network virtual terminal), per definire entrambi gli estremi di una connessione. Ogni terminazione della connessione (ogni NVT) ha una tastiera e una stampante logica. La stampante logica po' visualizzare caratteri, e la tastiera logica può generare caratteri. La stampante logica è usualmente lo schermo di un terminale, mentre la tastiera logica è usualmente la tastiera dell'utente, anche se potrebbe essere un file o altri strema di input. Questi termini sono usati inoltre nel File Transfer Protocol

[1] Ulteriori risorse su UNix

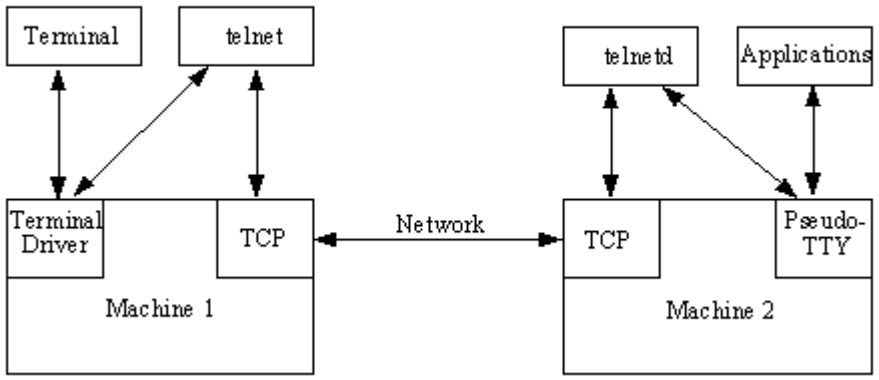
[2] Ulteriori informazioni su telnetd

(FTP) e Simple Mail Transfer Protocol(SMTP). La figura seguente mostra la tastiera e la stampante logiche



il protocollo Telnet tratta le due terminazioni della connessione come NTV. I due programmi agli estremi effettuano la traduzione dai terminali virtuali alle reali apparecchiature fisiche. Il concetto di terminale virtuale abilita telnet ad interconnettere ogni tipo di apparecchiatura, se è disponibile la mappatura fra codice virtuale e apparecchiatura fisica. Un vantaggio di questo approccio è che alcune apparecchiature fisiche non possono supportare alcune operazioni, così il terminale virtuale non ha questi codici. Quando i due terminali stanno stabilendo la connessione, si nota la mancanza di questi codici e le sequenze che li dovessero utilizzare verrebbero ignorate. Questo processo è esplicito: un terminale chiede se una funzione è supportata ,m e l'altro replica positivamente o negativamente. Se la funzione è supportata , sono inviati i codici necessari. La lista di funzioni supportate è realizzata quindi rapidamente.

Quando una connessione è stata stabilita, inizia un processo sul server per far girare le applicazioni. Ogni battitura di un tasto in una sessione Telnet deve attraversare differenti processi , come mostrato nella figura seguente



ogni battitura va attraverso Telnet, telnetd e le applicazioni che sono usate attraverso durante la sessione Telnet. Alcune applicazioni vogliono comunicare attraverso un apparecchio terminale , così il sistema remoto fa girare un driver pseudo TTY[3] che agisce come un terminale per l'applicazione. Se un'interfaccia a finestra come X[4] e Motif[5] è usata sul host e sulla macchina remota , i sistemi devono essere istruiti per permettere alle informazioni espresse attraverso finestre grafiche a passare avanti e indietro; altrimenti la macchina remota tenta di aprire le finestre sul server.

Per far partire Telnet occorre fornire o il nome o l'indirizzo IP della macchina a cui essere connesso. Il nome può essere usato solo se il sistema ha un mezzo per risolvere il nome e tradurlo nel corrispondente indirizzo IP, come con il Domain Name

[3] Ulteriori informazioni su sistemi TTY

[4] Ulteriori informazioni sul sistema X

[5] Ulteriori informazioni sul sistema Motif

System. Un nome di porta può usualmente essere utilizzato per connettere ad un servizio specifico, per esempio per connettere una macchina con l'indirizzo IP 205.150.89.1, occorre inserire il comando

telnet 205.150.89.1

se il sistema ha il nome darkstar, risolvibile nel corrispondente indirizzo IP, si può trasmettere il seguente comando

telnet darkstar

se non viene specificato nome, indirizzo, porta, Telnet entra nella modalità comando e attende specifiche istruzioni. Quando la connessione è stata stabilita, una User ID e una password vengono richieste . cui si può collegare con ogni ID e password che sono valide sul sistema remoto. Una tipica sessione con un sistema Unix assomiglia a qualcosa del genere

```
telnet 205.150.89.1

Trying...

Connected to tpci

Escape character is '^]'.

HP-UX tpci A.09.01 A 9000/720 (ttys2)

login: tparker

password: xxxxxxxx

$
```

Come si può vedere nel codice precedente, Telnet ha tentato di connettersi al sistema remoto , ha riferito che si è connessa , poi setta i parametri di comunicazione fra i due sistemi. Quando ciò è stato fatto , è stato mostrato il prompt di login (come su ogni terminale Unix) , seguito da una richiesta di password. Se la login e la password sono abilitate, il ‘prompt di shell di Unix (il segno di un dollaro) viene visualizzato per indicare che la macchina remota ora è attiva.

Si può usare il nome di una macchina come parte del comando Telnet solo se il sistema ha un mezzo per risolare il nome nel suo indirizzo IP. In caso contrario, non viene stabilita alcuna connessione, sebbene Telnet possa rimanere n modalità comando . per uscire, occorre usare la combinazione Ctrl+D o la sequenza di break visualizzata come parte del messaggio di start up.

Si può portare Telnet in command mode in ogni momento, usualmente usando la combinazione Ctrl+] . se si è correntemente connessi ad una sessione attiva quando si entra in modalità comando, Telnet attende che l’utente inserisca un comando , lo esegue, e poi ritorna alla sessione automaticamente. La modalità comando permette di inserire comandi relativi al client invece che al server. L’utente potrebbe averne bisogno per cambiare directory o lanciare applicazioni locali, per esempio.

Una volta che la connessione sia stata stabilita con successo, la sessione si comporta come se l’utente si trovasse sulla macchina remota , con tutti i comandi validi di quel sistema operativo. Tutte le istruzioni sono relative al server, così un comando di directory mostra la directory corrente sul server , non il cliente. Per vedere la directory del cliente , si dovrà portare Telnet in command mode . un semplice login e

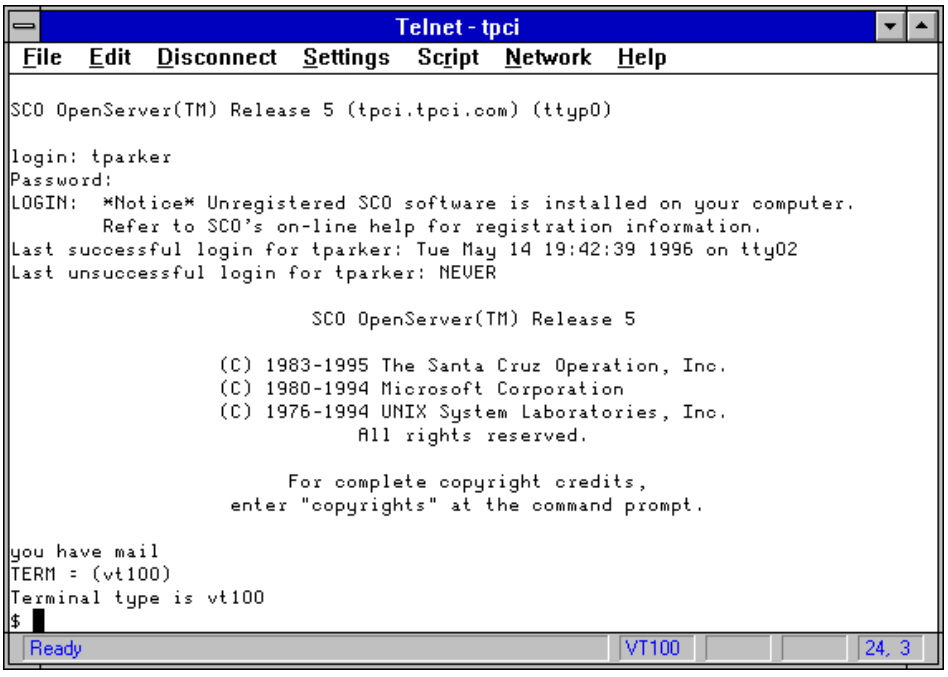
logout Telnet , chiamando da una stazione Unix (di nome merlin) ad un server (di nome tpci_hpws4) è mostrata di seguito

```
merlin> telnet tpci_hpws4

Trying...

Connected to tpci_hpws4.

Escape character is '^]'.

HP-UX tpci_hpws4 A.09.01 A 9000/720 (ttys2)

login: tparker

password: xxxxxxxx

tpci_hpws4-1> pwd

/u1/tparker

tpci_hpws4-2> cd docs

tpci_hpws4-3> pwd

/u1/tparker/docs

tpci_hpws4-2> <Ctrl+d>

Connection closed by foreign host.

merlin>
```

una volta che si è connessi alla macchina remota , la sessione si comporta esattamente come se l'utente fosse su quella macchina. Per effettuare il log out , occorre semplicemente fornire il comando logout (nel precedente esempio la combinazione

Unix Ctrl+D) . il programma telnet è utile quando si è su una macchina dalle potenzialità limitate o un terminale e si vogliono usare le capacità di elaborazione di un'altra macchina.

Le utilità Telnet sono disponibili per molti differenti sistemi operativi. La figura seguente mostra un'applicazione Telnet per Windows for Workgroups[6] (parte di una più ampia suite di applicazioni da NetManage[7] chiamata ChamaleonNFS) che sta effettuando il log in un server SCO Unix[8] . anche quando la macchina locale ha un'interfaccia grafica come Windows[9] ci si connette con molta probabilità alla macchina remota usando un'interfaccia basata sul carattere.



[6] Windows for Workgroups

[7] NetManage

[8] SCO Unix

[9] Windows

se le stazioni trasmittenti e riceventi usano un'interfaccia grafica (GUI[10]: Graphical User Interface[11]) come Motif e X , e si vogliono usare esse al posto dell'interfaccia a carattere, occorre istruire entrambi i terminali per usate il terminale locale per il windowing[12] (poiché non si 'può vedere una finestra sul sistema remoto). Localmente un programma viene attivato per istruire il sistema operativo per abilitare altre macchine a visualizzare direttamente sullo schermo , e il sistema remoto deve avere un'istruzione per redirezionare i comandi di windowing sullo schermo locale. Molti sistemi Unix effettuano ciò in questo modo:

```
tpci_server-1> xhost +

tpci_server-2> telnet tpci_hpws4

Trying...

Connected to tpci_hpws4.

Escape character is '^]'.

HP-UX tpci_hpws4 A.09.01 A 9000/720 (ttys2)

login: tparker

password: xxxxxxxx

tpci_hpws4-1> setenv DISPLAY tpci_server:0.0

tpci_hpws4-2> <Ctrl+d>
```

---

[10] GUI

[11] Graphical User Interface

[12] Windowing

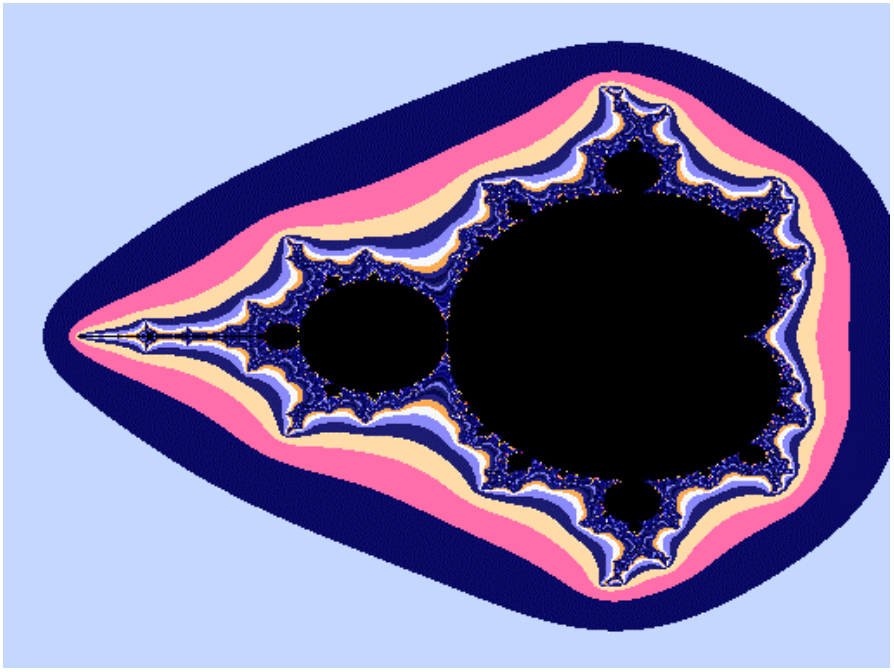
```
Connection closed by foreign host.

tpci_server-3>
```

L'istruzione UNIX xhost+ dice alla macchina locale di abilitare il sistema remoto a controllare le finestre sullo schermo locale. L'istruzione setenv DISPLAY *machine_name* eseguita sul sistema remoto Unix setta la variabile DISPLAY dell'ambiente shell di Unix sullo schermo locale. Quando una finestra deve essere aperta , il windowing appare sullo schermo locale, e il processing è condotto sul remoto. Questi esempi sono per UNIX ma una sequenza similare lavora su altre GUI.

Applicazioni complete che forniscono questa capacità per far girare finestre X e Motif su una macchina Windows, Windows 95[13], o Windows NT[14]sono disponibili da diversi venditori. Per esempio, la figura seguente mostra un'applicazione che gira su un server remoto chiamato mandel e disegna figure di Mandelbrot.

[13] Windows 95

[14] Windows NT



il server è stato istruito a visualizzare la finestra sulla macchina locale Windows for Workgroups usando un client X per macchine Windows. Il server passa tutte le informazioni circa l'ampiezza , posizione, e colore della finestra così come istruzioni per disegnare i contenuti al client locale . la finestra appare sulla macchina Windows esattamente come apparirebbe sul sistema remoto Unix.

Comandi Telnet

Diverse opzioni sono disponibili quando è stata stabilità una sessione Telnet. I loro valori possono essere cambiati durante il corso della sessione se entrambi i terminali sono d'accordo.(un terminal potrebbe essere prevenuto dall'abilitazione o disabilitazione di un servizio a causa dei settaggi dell'amministratore delle risorse). Vi sono quattro verbi usati dal protocollo Telnet per offrire, rifiutare, richiedere, e prevenire servizi. Rispettivamente will, won't, do e don't. i verbi sono pensati per essere accoppiati (will e won't, do e don't). per illustrare come funzionano

consideriamo la seguente sessione Telnet , che ha la visualizzazione di questi verbi settata usando le opzioni del comando toggle[15]:

```
tpci_server-1> telnet

telnet> toggle options

Will show option processing.

telnet> open tpci_hpws4

Trying...

Connected to tpci_hpws4.

Escape character is '^]'.

SENT do SUPPRESS GO AHEAD

SENT will TERMINAL TYPE (don't reply)

SEND will NAWS (don't reply)

RCVD do 36 (reply)

sent won't 36 (don't reply)

RECD do TERMINAL TYPE (don't reply)

RCVD will SUPPRESS GO AHEAD (don't reply)

RCVD do NAWS (don't reply)

Sent suboption NAWS 0 80 (80) 0 37 (37)

Received suboption Terminal type - request to send.
```

[15] Toggle

RCVD will ECHO (reply)

SEND do ECHO (reply)

RCVD do ECHO (reply)

SENT won't ECHO (don't reply)

HP-UX tpci_hpws4 A.09.01 A 9000/720 (ttys2)

login:

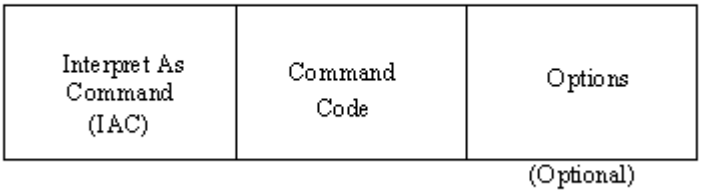
I comandi Telnet sono usati dal protocollo non dall'utente8sebbene egli li possa introdurre durante una sessione Telnet, ma ciò è usato normalmente soltanto per scopi diagnostici) . non vi sono comandi di utente in Telnet , oltre che il toggle in command mode, poiché il ruolo di Telnet è di connettere l'utente al sistema remoto e permettergli di usarlo direttamente.

Un set parziale di comandi Telnet è mostrato nella tabella seguente.

Code	Value	Description
Abort Output (AO)	245	Runs process to completion but does not send the output
Are you there (AYT)	246	Queries the other end to ensure that an application is functioning
Break (BRK)	243	Sends a break instruction
Data Mark	242	Data portion of a Sync
Do	253	Asks for the other end to perform or an acknowledgment that the other end is to perform
Don't	254	Demands that the other end stop performing or confirms that the other end is no longer performing
Erase Character	247	Erases a character in the output stream

(EC)		
Erase Line (EL)	248	Erases a line in the output stream
Go Ahead (GA)	249	Indicates permission to proceed when using half-duplex (no echo) communications
Interpret as Command (IAC)	255	Interprets the following as a command
Interrupt Process (IP)	244	Interrupts, suspends, aborts, or terminates the process
NOP	241	No operation
SB	250	Subnegotiation of an option
SE	240	End of the subnegotiation
Will	251	Instructs the other end to begin performing or confirms that this end is now performing
Won't	252	Refuses to perform or rejects the other end performing

I comandi Telnet sono inviati in un pacchetto formale chiamato comando, come mostrato nella figura seguente



Tipicamente i comandi contengono due o tre byte: l'istruzione Interpret as Command (IAC) , il codice del comando inviato, e parametri opzionali per quel comando. Le opzioni supportate da Telnet sono presenti nella tabella seguente

Code	***Description***

0	Binary transmission
1	Echo
2	Reconnection
3	Suppress Go Ahead (GA)
4	Approximate message size negotiation
5	Status
6	Timing mark
7	Remote controlled transmission and echo
8	Output line width
9	Output page length
10	Output carriage-return action
11	Output horizontal tab stop setting
12	Output horizontal tab stop action
13	Output form feed action
14	Output vertical tab stop setting
15	Output vertical tab stop action
16	Output line feed action
17	Extended ASCII characters
18	Logout
19	Bytes macro
20	Data entry terminal
21	SUPDUP
22	SUPDUP output
23	Send location
24	Terminal type
25	End of Record
26	TACACS user identification

27	Output marking
28	Terminal location number
29	3270 regime
30	X.3 PAD (Packet assembly and disassembly)
31	Window size

TN3270[16]

Molti mainframe[17] usano EBCDIC[18], mentre la maggior parte dei sistemi piccoli si affida ad ASCII[19] . questo può causare un problema quando si tenta una sessione Telnet fra macchine basate su EBCDIC e macchine basate su ASCII e viceversa, poiché i codici trasmessi non sono accurati. Per evitare ciò è stata sviluppata un'applicazione Telnet chiamata TN3270, che effettua la traduzione fra i due formati.

Quando TN3270 è usata per connettere due macchine, Telnet stesso stabilisce la connessione iniziale, e poi uno dei terminali si setta per la traduzione. Se una macchina ASCII sta chiamando una macchina EBCDIC , la traduzione fra i due formati è condotta sul server EBCDIC a meno che vi sia un gateway fra i due, nel qual caso è il gateway ad effettuare la traduzione.

Molte suite di applicazioni TCP/IP che includono un programma Telnet includono anche un programma TN3270. per esempio la figura seguente mostra una finestra

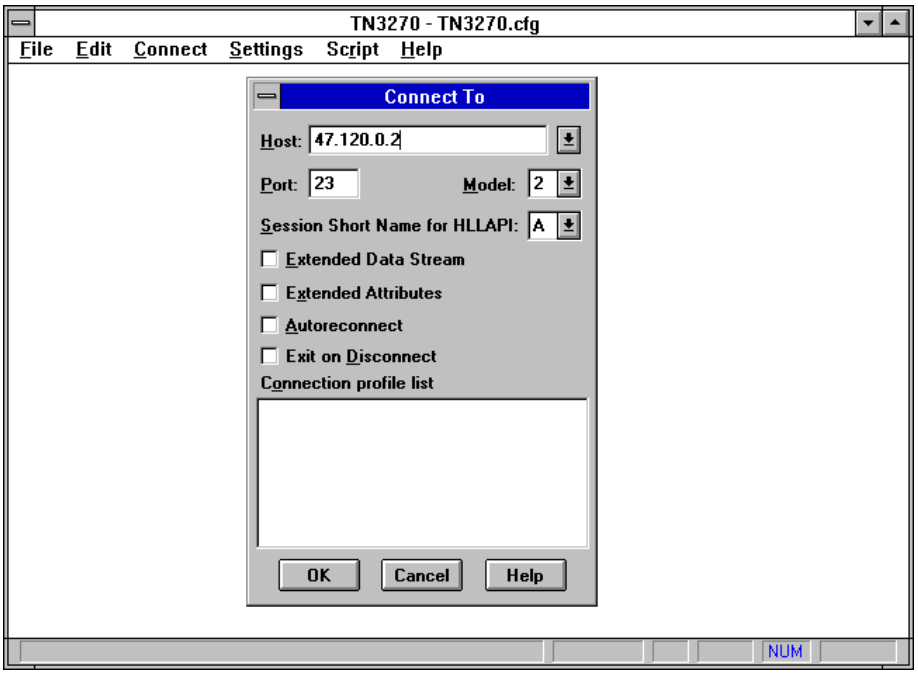
[16] TN3270

[17] Mainframe

[18] EBCDIC

[19] ASCII

3270per la suite NetManage ChamaleonNFS nel processo di connessione ad una macchina mainframe basata su EBCDIC. L'indirizzo IP del mainframe è usato per iniziare la connessione.



File Transfer Protocol (FTP)

Il protocollo FTP è usato per gestire file senza stabilire una connessione remota tramite Telnet. FTP permette di trasferire file in entrambi i sensi , gestire directory, ed accedere alla posta elettronica . FTP non è pensato per accedere a sistemi remoti per eseguire programmi .

FTP usa due canali TCP. La porta TCP 20 è il anale dati mentre la porta 21 è il canale comandi. FTP è differente dalla maggior parte delle altre applicazioni TCP/IP in quanto esso usa due canali, abilitando trasferimenti simultanei di comandi e dati FTP. Esso differisce anche in un altro aspetto importante : FTP conduce tutti i trasferimenti

di file in primo piano (foreground[20]) anziché in background[21]. In altre parole non usa spooler[22] o code (queue[23]) così l'utente vede il trasferimento dei file in real time[24]. Usando TCP , FTP elimina la necessità di preoccuparsi circa l'affidabilità o la gestione del collegamento, poiché FTP può fare affidamento su TCP per la realizzazione corretta di questi compiti.

Nel gergo FTP, i due canali che esistono fra le due macchine sono chiamati l'interprete di protocollo (PI: Protocol Interpreter[25]) e il processo di trasferimento dei dati (DTP: Data Transfer Process[26]) . Il PI trasferisce informazioni tra le due implementazioni usando il canale TCP 21 e DTP trasferisce dati tramite il canale 20. ciò è mostrato nella figura seguente.

---

[20] foreground

[21] background

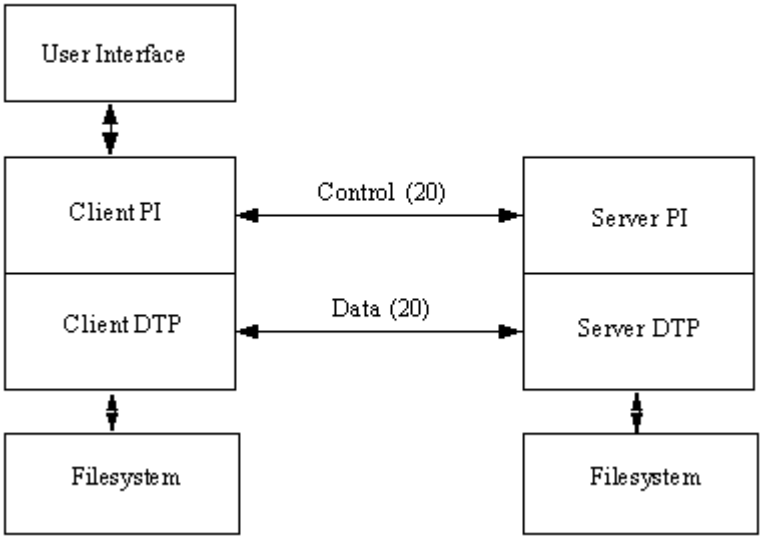
[22] spooler

[23] queue

[24] real time

[25] Protocol Interpreter

[26] DTP



FTP è simile a Telnet in quanto usa un programma server che gira continuamente e un programma separato eseguito sul client. Su sistemi Unix questi programmi sono chiamati ftpd[27] e ftp[28],rispettivamente.

Comandi FTP

Diamo un'occhiata ai comandi dietro al protocollo. Come con i comandi Telnet , essi sono per l'uso esclusivo da parte del protocollo e non dovrebbero essere usati dall'utente (sebbene l'amministratore qualche volta usa i comandi FTP per scopi di diagnostica e debugging[29]).

I comandi di protocollo interi di FTP sono sequenze di quattro caratteri ASCII terminati da un carattere di newline. Alcuni dei codici richiedono dei parametri dopo di essi. Un vantaggio primario di usare caratteri ASCII per i comandi è che un utente

[27] ftpd

[28] ftp

[29] debugging

può osservare il flusso di comandi e comprenderlo facilmente. Ciò aiuta considerevolmente nel pro esso di debugging. Inoltre esso permette all'utente cosciente di comunicare direttamente con il componente del FTP server.

I comandi FTP usati dal protocollo sono riassunti nella tabella seguente.

Command	Description
ABOR	Abort previous command
ACCT	User account ID
ALLO	Allocate storage for forthcoming operation
APPE	Append incoming data to an existing file
CDUP	Change to parent directory
CWD	Change working directory
DELE	Delete file
HELP	Retrieve information
LIST	Transfer list of directories
MKD	Make a directory
MODE	Set transfer mode
NLST	Transfer a directory listing
NOOP	No operation
PASS	User password
PASV	Request a passive open
PORT	Port address
PWD	Display current directory
QUIT	Terminate the connection
REIN	Terminate and restart a connection
REST	Restart marker (restart transfer)
RETR	Transfer copy of file

RMD	Remove a directory
RNFR	Old pathname for rename command
RNTO	New pathname for rename command
SITE	Provides service specifics
SMNT	Mount a file system
STAT	Returns status
STOR	Accept and store data
STOU	Accept data and store under different name
STRU	File structure
SYST	Query to determine operating system
TYPE	Type of data
USER	User ID

Questi comandi permettono il processo di connessione, il controllo della password, e l'effettivo trasferimento dei file. Essi non vanno confusi con i comandi disponibili per l'utente.

FTP usa inoltre semplici codici di ritorno per indicare le condizioni di trasferimento. Ogni codice di ritorno è un numero a tre cifre, il primo dei quali indica un'esecuzione avvenuta con successo (in tal caso la cifra è 1,2 o 3) o un fallimento (la prima cifra è 4 o 5) . la seconda o terza cifra specificano il codice di ritorno o la condizione di errore in maggior ritardo. I codici di ritorno FTP [30]sono mostrati nelle tabelle seguenti

First	***Description***

[30] Codici FTP

Digit	
1	Action initiated. Expect another reply before sending a new command.
2	Action completed. Can send a new command.
3	Command accepted but on hold due to lack of information.
4	Command not accepted or completed. Temporary error condition exists. Command can be reissued.
5	Command not accepted or completed. Reissuing the command will result in the same error (don't reissue).

Second Digit	Description
0	Syntax error or illegal command
1	Reply to request for information
2	Reply that refers to connection management
3	Reply for authentication command
4	Not used
5	Reply for status of server

FTP abilita trasferimenti di file in diversi formati , che sono usualmente dipendenti dal sistema. La maggior parte dei sistemi (inclusi i sistemi Unix) hanno solo due modi: testo e binario. Alcuni mainframe aggiungono il supporto dell'EBCDIC, mentre molti siti hanno un tipo locale disegnato per trasferimenti più rapidi tra macchine (il tipo locale può usare word a 32 e 64 bit).

I trasferimenti di tipo testo usano caratteri ASCII separati da caratteri carriage-return [31]e newline[32] , mentre il tipo binario abilita il trasferimento di caratteri senza conversione o formattazione. Il modo binario è più rapido e poi consente il trasferimento di tutti valori ASCII (necessari per i file non di testo) . nella maggior parte dei sistemi , FTP parte in modo testo , sebbene molti amministratori di sistema ora settano FTP al modo binario di default.

Connessioni FTP

FTP viene usualmente lanciato con il nome o indirizzo della macchina target. Come con Telnet il nome deve essere risolto in un indirizzo IP perché il comando abbia successo. La macchina target può inoltre essere specificata dalla linea di comando del FTP. Per esempio per connettersi all'indirizzo IP 205.150.89.5 , si può inserire questo comando

ftp 205.150.89.5

Quando FTP si connette con la destinazione, l'utente deve essere capace di effettuare il log nel sistema come un valido utente. Alcuni sistemi consentono un l9ogin anonimo o guest . l'estratto seguente mostra il processo di login quando un utente fornisce login e password per la macchina remota:

```
ftp tpci_hpws4
```

[31] carriage-return

[32] newline

Connected to tpci_hpws4.

220 tpci_hpws4 FTP server

Name (tpci_hpws4:tparker):

331 Password required for tparker.

Password:

230 User tparker logged in.

Remote system type is UNIX.

Using binary mode to transfer files.

Su network ampi dove è usato un sistema come Yellow Pages [33](YP[34]) o Network Information Services[35] (NIS[36]) , login FTP sono permessi sulla maggior parte delle macchine. SE YP o NIS non sono utilizzati , occorre essere nel file degli utenti validi per ottenere accesso FTP. Per trasferire file, occorre avere i permessi appropriati sul remoto.

Dopo avere effettuato il jogging in un'altra macchina usando FTP, l'utente non è realmente sulla macchina remota. Si è ancora logicamente sul client, cosicché tutte le istruzioni per trasferimenti di file e movimenti di directory devono avvenire con riferimento alla macchina locale, non alla remota.

---

[33] Yellow Pages

[34] YP

[35] Network Information Services

[36] NIS

Tutti i riferimenti a file e directory sono relativi alla macchina che ha iniziato la sessione FTP. Se non si è attenti si rischia di sovrascrivere file esistenti.

Il processo seguito d FTP quando è stabilita una connessione è il seguente:

1. Login: verifica la ID di utente e la password
2. Define directory : identifica la directory di partenza
3. Define file transfer mode: definisce il tipo di trasferimento.
4. Start data transfer: abilita I comandi d'utente
5. Stop data transfer: chiude la connessione

I passi vengono effettuati in sequenza per ogni connessione. Un utente ha diversi comandi disponibili [37]per controllare FTP; i comandi più frequentemente utilizzati sono presenti nella tabella seguente

FTP Command	Description
ascii	Switch to ASCII transfer mode
binary	Switch to binary transfer mode
cd	Change directory on the server
close	Terminate the connection
del	Delete a file on the server
dir	Display the server directory
get	Fetch a file from the server
hash	Display a pound character for each block transmitted
help	Display help
lcd	Change directory on the client

[37] FTP Command

mget	Fetch several files from the server
mput	Send several files to the server
open	Connect to a server
put	Send a file to the server
pwd	Display the current server directory
quote	Supply an FTP command directly
quit	Terminate the FTP session

Usare FTP è simile a Telnet, eccettuato il fatto che tutti i movimenti di file sono relativi al client. Così il putting di un file significa muoverlo dal client al server, mentre il gettino di un file significa il viceversa. Ecco un esempio di sessione FTP

```
tpci_hpws1-1> ftp tpci_hpws4

Connected to tpci_hpws4.

220 tpci_hpws4 FTP server (Version 1.7.109.2 Tue Jul 28 23:32:34 GMT 1992) ready.

Name (tpci_hpws4:tparker):

331 Password required for tparker.

Password:

230 User tparker logged in.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp> pwd

257 "/u1/tparker" is current directory.
```

ftp> get mandelfile1.gif

remote: mandelfile1.gif local: mandelfi.gif

200 PORT command successful

150 Opening BINARY mode data connection for mandelfile1.gif

226 File transfer complete

1192834 bytes sent in 0.89 seconds

ftp> <Ctrl+d>

tpci_hpws1-2>

In questo piccolo esempio è stato trasferito un file chiamato mandelfile1.gif da una macchina Unix (server) alla macchina locale (client). Il nome del file è stato troncato automaticamente dal server per corrispondere alle convenzioni di denominazione del filesystem[38] DOS[39] . si noti inoltre che è stato usato il modo binario (che era il modo di default del sistema) .

E' disponibile un'opzione di debugging dalla command line aggiungendo –d al comando. Questo visualizza le istruzioni del canale di comando. Le istruzioni dal client sono mostrate con una freccia come primo carattere, mentre le istruzioni dal server iniziano con tre cifre. Un PORT nella linea di comando indica l'indirizzo del canale dati su cui il client attende la risposta del server. Se non è specificato alcun

[38] filesystem

[39] DOS

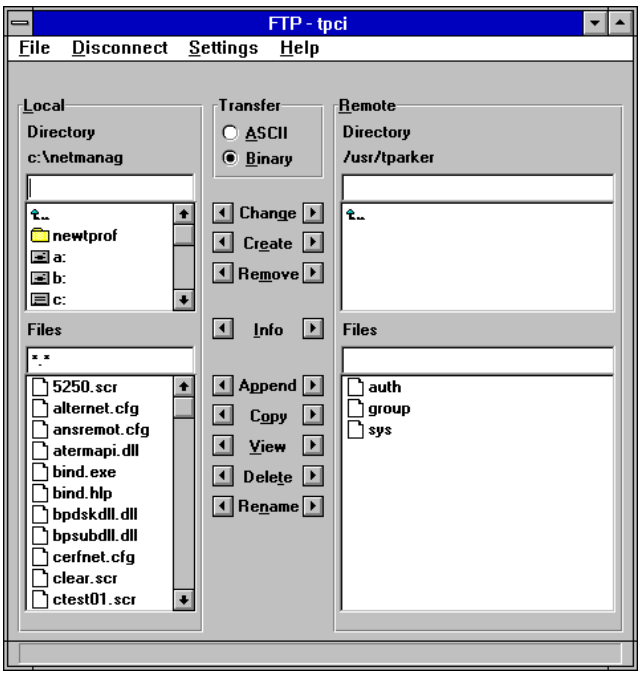
PORT, il canale di default è il 20. sfortunatamente, il progresso di trasferimenti di dati non possono essere seguiti nel modo debugging. Un esempio con l'opzione di debug settata è mostrato di seguito

```
tpci_hpws1-1> ftp -d

ftp> open tpci_hpws4

Connected to tpci_hpws4.

220 tpci_hpws4 FTP server Name (tpci_hpws4:tparker):

---> USER tparker

331 Password required for tparker.

Password:

---> PASS qwerty5

230 User tparker logged in.

---> SYST

215 UNIX Type: L8

Remote system type is UNIX.

---> Type I

200 Type set to I.

Using binary mode to transfer files.

ftp> ls

---> PORT 47,80,10,28,4,175
```

```
200 PORT command successful.
---> TYPE A
200 Type set to A.
---> LIST
150 Opening ASCII mode data connection for /bin/ls.
total 4
-rw-r-----  1 tparker  tpci    2803  Apr 29 10:46 file1
-rw-rw-r--  1 tparker  tpci    1286  Apr 14 10:46 file5_draft
-rwxr-----  2 tparker  tpci   15635  Mar 14 23:23 test_comp_1
-rw-r-----  1 tparker  tpci      52  Apr 22 12:19 xyzzy
Transfer complete.
---> TYPE I
200 Type set to I.
ftp> <Ctrl+d>
tpci_hpws1-2>
```

Si può notare come il modo sia cambiato da binario ad ASCII per inviare la lista della directory, e poi al valore binario.

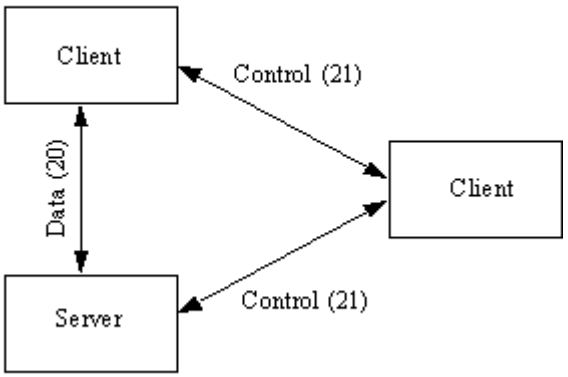
Quando FTP è usato in un ambiente grafico , si può usare un tool basato su GUI. Per esempio quella mostrata nella figura seguente



FTP Third-Party transfers[40]

FTP permette che un trasferimento avvenga attraverso una terza macchina posizionata tra il client e il server. Questa procedura è talvolta necessaria per ottenere permessi appropriati di accesso alla macchina remota. La figura seguente mostra lo schema di un trasferimento third-party , con la connessione di controllo fatta attraverso una terza macchina.

[40] FTP Third-party transfer



quando sta instaurando una connessione third-party , il client apre le connessioni di controllo tra la macchina remota e il secondo cliente che gestisce il canale di controllo. Soltanto il canale di controllo passa attraverso il secondo client, mentre i canali dati passa direttamente fra le due estremità.

Quando è sottoposta una richiesta di trasferimento , essa è trasferita attraverso il secondo client, che controlla i permessi e poi la passa al server. Il trasferimento di dati può poi aver luogo direttamente.

Anonymous FTP Access[41]

FTP richiede un ID di utente ed una password per abilitare le capacità di trasferimento, ma vi è un metodo più liberale di abilitare accesso generale a file e directory, chiamato FTP anonimo. Esso rimuove la richiesta di un account di login sulla macchina remota con una password che può essere la parola guest o il nome di login dell'utente. La sessione seguente mostra l'uso di un FTP anonimo

```
tpci_hpws4-1> ftp uofo.edu
```

[41] Anonymous FTP Access

Connected to uofo.edu.

220 uofo.edu FTP server (Version 1.7.109.2 Tue Jul 28 23:32:34 GMT 1992) ready.

Name (uofo:username): anonymous

331 Guest login ok, send userID as password.

Password: tparker

230 Guest login ok, access restrictions apply.

ftp> <Ctrl+d>

tpci_hpws4-2>

se il sistema remoto è settato per abilitare il login anonimo, viene richiesta una password all'utente e poi viene dato un avvertimento circa le limitazioni di accesso.

Server FTP[42]

La maggior parte delle macchine Unix agisce come un server FTP per default. Per fornire potenzialità FTP esse fanno girare un daemon[43] ftpd quando viene lanciato il sistema operativo. Il daemon è usualmente gestito dal processo inetd[44] di Unix. Quando l'utente comincia a utilizzare inetd, il daemon inetd controlla la pronta di comando TCP 21 per l'arrivo di una richiesta di connessione , poi lancia ftpd per

---

[42] Server FTP

[43] daemon

[44] inetd

servire quella richiesta. I sistemi Windows invece mancano di un software ftp server così occorre aggiungere un prodotto apposito.

Trivial File Access Protocol (TFTP)

Il Trivial File Access Protocol (TFTP) è uno dei più semplici protocolli di trasferimento di file in uso. Esso differisce da FTP in due punti fondamentali: esso non effettua il logging nella macchina remota , ed usa il protocollo di trasporto non orientato alla connessione User Datagram Protocol [45](UDP) invece di TCP[46]. Usando UDP, il TFTP non monitorizza il progresso nel trasferimento del file , sebbene esso debba impiegare algoritmi più complessi per assicurare appropriata integrità dei dati. Per evitare il logging sono evitati i problemi di accesso dell'utente e i permessi dei file. TFTP usa la porta TCP numero 69 , sebbene TCP non sia coinvolto.

TFTP ha pochi vantaggi su FTP . Non è usato usualmente per trasferimenti di file tra macchine dove potrebbe essere usato FTP, sebbene TFTP sia utile quando è coinvolto un terminale senza disco. Tipicamente TFTP è usato per caricare applicazioni e font in queste macchine, così come per il bootstrapping[47]. TFTP è necessario in questi casi poiché il terminale senza dischi non può eseguire FTP finché il suo sistema operativo

---

[45] User Datagram Protocol

[46] TCP

[47] bootstrapping

non sia stato completamente caricato. La piccola ampiezza dell'eseguibile TFTP e la piccola richiesta di memoria lo rendono ideale per l'inclusione in un bootstrap[48] .

TFTP gestisce permessi di accesso e di file imponendo sue proprie restrizioni. Sulla maggior parte dei sistemi Unix ad esempio, un file può essere trasferito soltanto se è accessibile a tutti gli utenti sul remoto. A causa delle regole di accesso lasche , la maggior parte degli amministratori di sistema impongono più controlli su TFTP (o bandiscono completamente il suo uso).; altrimenti sarebbe facile per un utente esperto accedere o trasferire dei file che potrebbero costituire una violazione della sicurezza.

I trasferimenti TFTP possono fallire per molte ragioni , poiché praticamente ogni tipo di errore durante un'operazione di trasferimento causa un fallimento completo. TFTP supporta alcuni messaggi di errore di base, ma non può gestire semplici errori come risorse insufficienti per un trasferimento di file o perfino il fallimento nella localizzazione del file richiesto.

Comandi TFTP[49]

Le istruzioni importanti nel set di comandi TFTP sono mostrate nella tabella seguente. Il set di comandi è simile a quello di FTP , ma differisce in alcuni importanti aspetti a causa degli aspetti non orientati alla connessione del protocollo: il più evidente è il comando connect che determina semplicemente l'indirizzo del remoto invece di iniziare la connessione.

---

[48] bootstrap

[49] TFTP Commands

TFTP Command	Description
binary	Use binary mode for transfers
connect	Determine the remote's address
get	Retrieve a file from the remote
put	Transfer a file to the remote
trace	Display protocol codes
verbose	Display all information

TFTP consente sia il trasferimento di tipo binario che quello di tipo testo. Così come per Telnet e FTP, TFTP usa un processo di sistema (tftpd[50] su un sistema Unix) ed un eseguibile, comunemente chiamato tftp. Un esempio di sessione TFTP su un host[51] Unix è mostrata qui, con l'attivazione del tracciamento e del trasferimento di tipo binario:

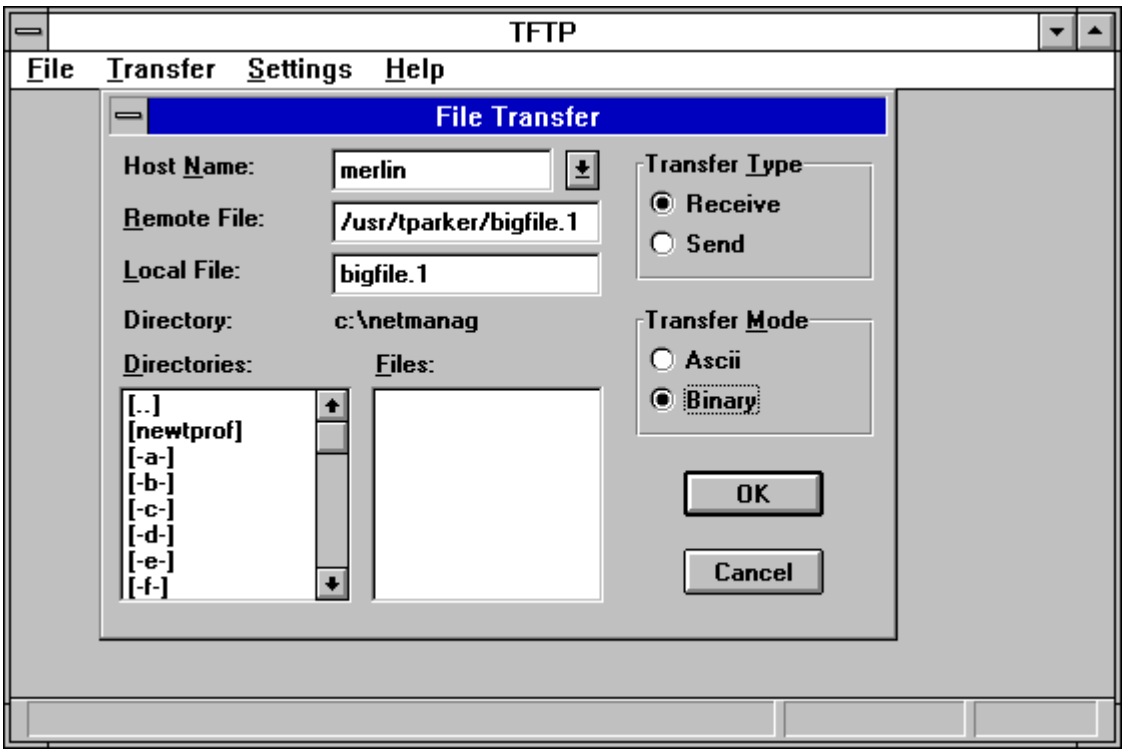
```
tpci_hpws1-1> tftp

tftp> connect tpci_hpws4

tftp> trace

Packet tracing on.

tftp> binary

Binary mode on.

tftp> verbose
```

[50] tftpd

[51] host

```
Verbose mode on.

tftp> status

Connected to tpci_hpws4.

Mode: octet Verbose: on Tracing: on

Rexmt-interval: 5 seconds, Max-timeout: 25 seconds

tftp> get /usr/rmaclean/docs/draft1

getting from tpci_hpws4:/usr/rmaclean/docs/draft1 to /tmp/draft1 [octet]

sent RRQ <file=/usr/rmaclean/docs/draft1, mode=octet>

received DATA <block1, 512 bytes>

send ACK <block=1>

received DATA <block2, 512 bytes>

send ACK <block=3>

received DATA <block4, 128 bytes>

send ACK <block=3>

Received 1152 bytes in 0.2 second 46080 bits/s]

tftp> quit

tpci_hpws1-2>
```

nella sessione si può notare che I comandi trace e verbose attivano l’eco del flusso di istruzioni tra le due macchine durante un trasferimento di file.



Pacchetti TFTP[52]

TFTP usa UDP come protocollo di trasporto , così può utilizzare l'header UDP[53] per incapsulare informazioni di protocollo TFTP . TFTP usa i campi porta sorgente e porta destinazione UDP per settare i due estremi della connessione. Effettua ciò usando gli identificatori di trasferimento TFTP (TID: TFTP Transfer Identifier) , che sono creati da TFTP e passati ad UDP, che li pone poi nell'header.

Come con Telnet ed FTP, TFTP usa il vincolamento delle porte (port binding[54]), dove la macchina mittente seleziona un TID, e la remota è settata alla porta 69 (numero di porta di TFTP). La macchina remota risponde con un riscontro della richiesta di

---

[52] Pacchetti TFTP

[53] Header UDP

[54] port binding

connessione, della porta sorgente di 69, e del TID di destinazione inviato nella richiesta.

TFTP usa cinque tipi di Protocol Data units[55], cui si fa riferimento come pacchetti nel lessico TFTP. Questi pacchetti sono elencati nella tabella seguente.

Code	OpCode	Description
ACK	4	Acknowledgment
DATA	3	Send Data
Error	5	Error
RRQ	1	Read request
WRQ	2	Write request

Il loro layout è mostrato nella figura seguente.

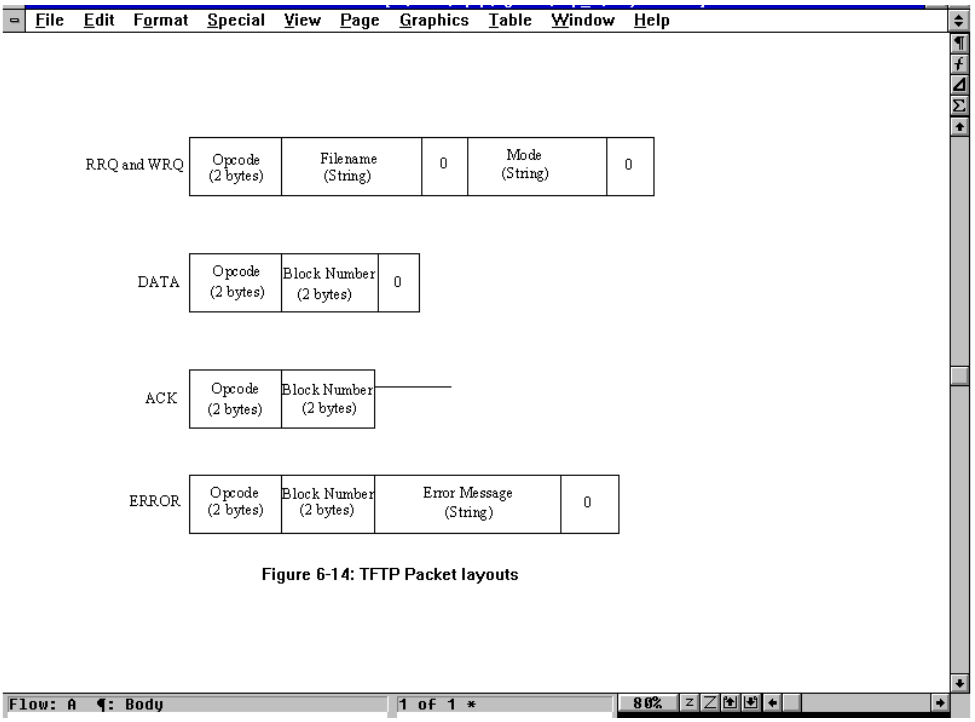


Figure 6-14: TFTP Packet layouts

I messaggi di errore supportati da TFTP sono mostrati nella tabella seguente.

[55] Protocol Data Units

Code	Description
0	Not defined
1	File not found
2	Permissions prevent access
3	Disk full or allocation limit exceeded
4	Illegal TFTP operation requested
5	Unknown transfer number

I layout sia per I pacchetti RRQ che WRQ ha un campo modo, che indica il tipo di trasferimento. Vi sono tre modi correntemente disponibili su TFTP:

- NetASCII: codici ASCII standard
- Byte: informazioni binarie e a 8 bit
- Mail: indica che la destinazione è un utente, non un file.

L'ultimo blocco in tutti i pacchetti contiene fra 0 e 511 byte di dati .

Il processo di comunicazione usato da TFTP comincia con il client che invia una richiesta RRQ o WRQ al server tramite UDP. Come parte della richiesta, un numero di transazione , il filename[56], e un codice per identificare il modo di trasmissione da usare vengono specificati. Il numero di transazione viene usato per identificare transazioni future nella sequenza.

Poiché non vi è connessione fra i due, il client setta un timer e attende una replica da parte del server. Se non arriva una replica prima che il timer espiri, viene inviata una

[56] filename

nuova richiesta. Dopo che è stato ricevuto un ACK, viene trasmesso un pacchetto di dati, per il quale viene ricevuto un ACK o un ERROR. Se vi sono diversi pacchetti da trasmettere, essi sono costruiti in modo da avere una lunghezza di 512 byte e un numero di sequenza incrementato. Il processo termina quando il server riceve un pacchetto di ampiezza inferiore a 512 byte. Per ogni pacchetto inviato, TFTP aspetta un riscontro prima di inviare il successivo. Sistema noto come protocollo flip-flop.

Simple Mail Transfer Protocol (SMTP)

Il protocollo SMTP è il metodo definito in Internet per trasferire posta elettronica [57] . SMTP è simile a FTP in molti modi, inclusa la semplicità di uso. SMTP usa la porta YTCP numero 25.

La maggior parte dei sistemi Unix usa programmi chiamati sendmail [58] o mmdf[59] per implementare SMTP (così come latri protocolli di posta[60]). Il programma sendmail, per esempio, agisce sia come client che come server, usualmente operando in background come daemon. Gli utenti non interagiscono con sendmail direttamente ma usano un programma di mail front-end[61] come mail[62], mailx[63], o Mail. Questi sistemi passano il messaggio a sendmail perché venga inoltrato.

---

[57] electronic mail o Posta elettronica

[58] sendmail

[59] mmdf

[60] protocolli di posta

[61] front-end mail program

SMTP usa spool e code. Quando un messaggio è inviato a SMTP, esso lo posiziona in una coda. SMTP tenta di inoltrarlo appena si collega alla macchina remota. Se non può inoltrare il messaggio in un limite di tempo specificato, il messaggio è restituito al mittente o rimosso.

Comandi SMTP

Le trasmissioni SMTP usano un formato semplice. Tutto il testo del messaggio è trasferito con caratteri ASCII a 7 bit. La fine del messaggio è indicata da un punto isolato in una riga. Se per qualche ragione una riga nel messaggio comincia con un punto, viene aggiunto un secondo punto per impedire confusioni.

SMTP ha un set di comandi[64] elencato nella tabella seguente

Command	Description
DATA	Message text
EXPN	Expansion of a distribution list
HELO	Use in connection establishment to exchange identifiers
HELP	Request for help
MAIL	The sender's address
NOOP	No operation
RCPT	The message destination address (more than one can be provided)
RSET	Terminate the current transaction
SAML	Send a message to the user's terminal and send mail

---

[62] mail

[63] mailx

[64] Set di comandi di SMTP

SEND	Send a message to the user's terminal
SOML	Either send a message to the user's terminal or send mail
TURN	Change the sending direction (reverse sending and receiving roles)
VRFY	Verify the user name

Quando è stabilita una connessione, i due sistemi SMTP scambiano codici di autenticazione. Poi un sistema manda un comando MAIL all'altro pere identificare il mittente e fornire informazioni sul messaggio. L'SMTP ricevente restituisce un acknowledgment , dopo di che è inviato un RCPT per identificare il ricevente. Se vengono identificati più riceventi alla locazione di ricezione, sono mandati diversi messaggi RCPT , ma il messaggio stesso è inviato una sola volta. Dopo ogni RCPT vi è un acknowledgment. Un comando DATA è seguito dalle righe di messaggio , finché un punto da solo su una riga indica la fine del messaggio. La connessione è chiusa con un comando QUIT.

I campi indirizzi del mittente e del ricevente usano formati standard Internet[65], che contengono l'user name e il dominio[66]. Il dominio può essere rimpiazzato da altre informazioni se è stabilita una connessione diretta. SMTP usa il Domain Name System per tutti gli indirizzi.

[65] Internet

[66] domain

Le utilità Berkeley[67]

L'università della California[68] a Berkeley fu di aiuto nello sviluppo di TCP/IP e fornì molti programmi di utilità al set di applicazioni. Essi sono noti come *Berkeley r-Utilities*. Sono chiamati r-utilities perché iniziano tutti con la lettera r per remoto. La maggior parte delle utility sono specifiche per Unix , sebbene esse da allora sono state portate ad altri sistemi operativi.

I file hosts.equiv[69] e .rhosts[70]

Per abilitare le macchine ha dialogare correttamente in una rete , devono essere settati i diritti di accesso per le macchine e gli utenti. Usualmente, quando sta effettuando il login in un'altra macchina, un utente deve fornire una ID utente ed una password . quando di effettua il login in molte macchine, la digitazione di queste informazioni può essere tediosa e consumare tempo. Può essere inoltre un problema di sicurezza, poiché è facile scrivere un programma che monitorizza le connessioni di rete per ottenere queste informazioni. Un modo per mettere accesso veloce senza realmente effettuare il login e prevenire l'intercettazione delle password è chiaramente utile.

L'amministratore di sistema può decidere che tutti i nomi di login usati su altre macchine i cui nomi sono nel file hosts.equiv hanno accesso consentito sulla macchina locale. Questo abilita un protocollo che interroga una macchina per

[67] The Berkeley Utilities

[68] Università della California Berkeley

[69] hosts.equiv

[70] .rhosts

ottenere accesso a controllare il file per individuare il nome della macchina che richiede accesso. , e se lo trova, a garantire accesso all'utente che si trova su quella macchina. L'utente ha gli stessi diritti di accesso che ha sull'altra macchina.

Se il protocollo non trova una registrazione sul file , esso può controllare un altro file gestito nella home directory[71] dell'utente, chiamato .rhosts . un utente può controllare che ha accesso al suo nome di login con il file .rhosts nella sua home directory, abilitando altri utenti ad effettuare il login come se essi fossero quell'utente. Il file deve essere posseduto dall'utente che non deve consentire accesso in scrittura a tutti gli utenti. Un file .rhosts consiste di una riga per ogni utente accolto nella home directory . la riga consiste di un nome di macchina e di un nome di login. Un esempio è mostrato qui

```
tpci_hpws1 rmaclean

tpci_hpws1 bsmallwood

tpci_hpws3 ychow

tpci_hpws3 bsmallwood

tpci_hpws4 glessard

tpci_hpws4 bsmallwood

tpci_sunws1 chatton

merlin tparker
```

[71] home directory

merlin ahoyt

merlin lrainsford

rlogin[72]

Il comando rlogin (remote login) abilita un utente ad effettuare il login in un'altra macchina. E' molto simile al Telnet anche se molto più semplice. Vi è un programma in background che gira sul server e chiamato rlogind[73] , mentre il programma rlogin risiede sul client.

Il protocollo rlogin inizia una sessione inviando stringhe di tre caratteri separate da degli zeri. La prima stringa è l'ID di login dell'utente (sul client) , la seconda stringa è il nome di login per il server (usualmente ma non sempre lo stesso) e la terza stringa è il nome di login e la velocità di trasmissione del terminale[74] di utente. Quando viene ricevuta sul server, la stringa può essere convertita in variabili di ambiente[75] . non si può effettuare il login con un user ID differente , poiché il sistema non effettua il prompt[76] per il nome di login . comunque esso effettua il prompt per un password.

Dopo che è completato il processo di login , rlogin non usa alcun protocollo. Ogni carattere digitato sulla macchina cliente è inviato al server , mentre ogni carattere

---

[72] rlogin

[73] rlogind

[74] terminal

[75] variabili di ambiente

[76] prompt

generato sul server è mostrato sulla console[77] del client. La sola uscita verso il sistema locale è la chiusura della connessione con CTRL+D o l'immissione del carattere di escape[78] da solo su una riga. Per default il carattere di escare è una tilde (~).

Alcune versioni di rlogin abilitano una shell escape[79] , una temporanea sospensione della sessione rlogin e il ritorno al sistema operativo , usando ~!

rsh[80]

l'utilità rsh (remote shell[81]) consente di eseguire comandi su una macchina remota. È coinvolto un processo in background chiamato rshd[82] . eseguire un comando su una macchina remota significa aggiungere rsh e il nome della macchina all'inizio della riga di comando . l'utilità rsh dipende dalla presenza di host.equiv o .rhosts per abilitare il login , altrimenti l'accesso non è garantito.

L'utilità rsh non è una shell nel senso che essa non interpreta i comandi come la shell UNIX C[83] o la shell Bourne[84]. Invece un comando inserito viene inviato all'input e

---

[77] console

[78] escape character

[79] shell escape

[80] rsh

[81] remote shell

[82] rshd

[83] Unix C shell

[84] Bourne shell

output standard[85] del server , eseguendo il comando come un processo locale attraverso la connessione TCP. Il vantaggio principale è che uno script di shell [86]eseguito sulla macchina locale può esedre sottoposto alla macchina remota senza modifiche , dove esso gira come se fosse locale (fatta eccezione per l'uso del file system remoto).

Sfortunatamente ogni codice di ritorno generato dal sistema remoto non viene rinviato alla macchina locale. Inoltre, la maggior parte delle applicazioni orientate allo schermo non funziona in maniera appropriata, poiché esse non hanno un output di terminale su cui scrivere.

rcp[87]

il comando rcp (remote copy) è simile Al comando Unix cp[88] , fatta eccezione per il fatto che esso lavora attraverso la rete. La sintassi e la lista delle opzioni sono le stesse di cp, sebbene un nome di macchina sia usualmente specificato come parte del filename aggiungendo il nome della macchina seguito da una virgola . è supportata anche la copia ricorsiva delle directory 8un aspetto utile ed attraente non supportato da Telnet e FTP). Il programma rcp agisce sia da client che da server, ed inizializza quando arriva una richiesta.

rcp tpci_hpws4:/user/tparker/doc/draft1 .

---

[85] standard input and output

[86] shell script

[87] rcp

[88] Unix cp command

rcp file2 merlin:/u1/bsmallwood/temp/file2

rcp -r merlin:/u2/tparker/tcp_book tpci_server/tcp_book

rcp merlin:/u1/ychow/iso9000_doc tpci_server:/u1/iso/doc1/iso_doc_from_ychow
rcp file4 tparker@tpci.com:new_info

come indica l'esempio , sono specificati i filename sia alla macchina locale che a quella remota. , con convenzioni standard Unix. Il terzo esempio mostra un file che viene trasferito da una macchina all'altra , nessuna delle quali è la macchina da cui è stato inizializzato il comando. L'ultimo esempio mostra l'uso di un nome in stie DSN per l'indirizzo di destinazione.

L'utilità rcp è un metodo più veloce per trasferire dati rispetto a FTP, sebbene richieda un permesso di accesso tramite un file .rhots.senza una registrazione in questo file è rifiutato l'accesso e devono essere usati FTP o TFTP.

rwho[89]

il comando rho (remote who) usa il daemon rwhod per visualizzare una lista di utenti sulla rete. Esso mostra tutti gli utenti della rete, compilando la lista da un pacchetto di

[89] rwho

informazioni inviato regolarmente da tutti i programmi rwhod[90] che girano. Al frequenza di questo broadcast dipende dal sistema ma è usualmente nell'ordine di un invio in un intervallo che va da 1 a 3 minuti. Quando un programma rwhod riceve un broadcast[91] da un'altra macchina , lo piazza in un file di sistema per usi futuri.

Quando una macchina non ha inviato un messaggio broadcast all'interno di un limite di tempo (usualmente 11 minuti) , si presume che essa si sia disconnessa dalla rete, e tutti gli utenti elencati come attivi su quella macchina nel file di sistema sono ignorati. Un utente viene cancellato dal messaggio broadcast se dal suo terminale non si è avuto alcun segnale nell'arco di un'ora. Il risultato di una inchiesta rwho è mostrato nell'esempio seguente . per ogni utente esso mostra il nome di login, il nome di macchina e il nome di terminale, e l'ora e la data di login.

bsmallwood merlin:tty2p Feb 29 09:01

etreijs tpci_hpws2:tty01 Feb 29 12:12

rmaclean goofus:tty02 Feb 28 23:52

tparker merlin:tty01 Feb 29 11:43

ychow prudie:tty2a Feb 28 11:37

il programma ha un problema fondamentale su ampie reti: l'invio continuo di pacchetti di aggiornamento da ogni macchina crea un notevole aumento del traffico

[90] rwhod

[91] broadcast

in rete. Per questa ragione, alcune implementazioni richiedono direttamente alle macchine gli utenti soltanto quando vi è una rwho request.

ruptime[92]

l'utilità ruptime mostra una lista di tutte le macchine in rete, il loro stato, il numero dei loro utenti attivi, il carico corrente, e il tempo passato dall'avvio ella macchina. Il programma usa le stesse informazioni del programma rwho.

Ecco un esempio di output del comando ruptime

```
merlin     up    3:15,12 users, load 0.90, 0.50, 0.09

prudie     down  9:12

tpci_hpws1 up    11:05, 3 users, load 0.10, 0.10, 0.00

tpci_hpws2 up    23:59, 5 users, load 0.30, 0.25, 0.08

tpci_hpws3 down  6:45

tpci_hpws4 up    9:05, 1 user,  load 0.12, 0.05, 0.01
```

rexec[93]

il commando rexec (remote execution) è una vestigia delle prime versioni di Unix. Fu disegnato per permettere l'esecuzione remota di programmi attraverso un processo server chiamato rexecd. L'utilità usa la porta TCP numero 512.

---

[92] ruptime

[93] rexec

Il protocollo utilizzato da rexec è molto simile a rsh , eccettuato il fatto che una password criptata è inviata con al richiesta e vi è un processo di login completo. L’utilità è usata raramente poiché rsh è un metodo molto più veloce per eseguire un comando in remoto.