

II.1.6 Protocolul Finger

Finger este un protocol simplu de tip serviciu Internet, bazat pe modelul client-server, care permite unui client Finger să obțină de la un server Finger informații cu caracter public despre un anumit utilizator sau despre toți cei activi în rețea, la un moment dat.

Interogarea serverului Finger se face pe portul de protocol 79 și se exprimă în formatul NVT ASCII definit de Telnet, având ca simbol de sfârșit de linie combinația de caractere CR-LF (*Carriage Return - Line Feeder*).

Serviciul Finger este orientat pe conexiunea dintre două socket-uri (prin socket, se înțelege un pointer la o structură de date memorată într-un buffer). După stabilirea conexiunii folosind TCP, se transmite cererea Finger.

Dacă apelul e realizat printr-o linie vidă (numai CR-LF), atunci serverul răspunde cu lista completă a tuturor utilizatorilor conectați la server în acel moment. Comanda folosită în acest caz este:

```
finger
```

Dacă lista completă a utilizatorilor are o dimensiune care depășește lungimea maximă a unității de transmisie din rețea (MTU), atunci TCP transmite datele în mai multe segmente de mesaj.

Comanda de interogare Finger poate să includă numele real al unui utilizator conectat la server:

```
finger {nume}
```

sau ID-ul utilizatorului și numele calculatorului-gazdă, cu sintaxa:

```
finger {nume_utilizator}@{nume_calculator_gazda}
```

Răspunsul constă în informațiile disponibile despre acesta: numele de utilizator, numele real al persoanei, contul curent al acesteia (*home directory*), ultima dată la care s-a conectat la rețea, numele stației la care s-a făcut conexiunea, informații privind folosirea serviciului de poștă electronică de către utilizator etc.

Dacă serverul Finger recepționează orice în afara unei linii vide, atunci acesta presupune că textul identifică un anumit utilizator și returnează toate informațiile publice despre utilizatorii cu numele sau ID-ul asemănător textului.

Serverul Finger transmite mesajul de răspuns stației sau sesiunii Telnet de la care s-a inițiat cererea. Clientul recepționează datele fie sub forma unui șir de octeți (*byte stream*), fie prin datagrame, și le memorează într-un buffer local. În cazul în care volumul datelor depășește capacitatea de memorare a acestuia, atunci datele se pierd și se returnează un mesaj de eroare.

Clientul recepționează un mesaj de eroare, prin intermediul ICMP, deși conexiunea a funcționat corect și transmisia este considerată reușită.

Serverul Finger închide automat conexiunea după ce a transmis toate datele solicitate.

Pe durata unei conexiuni active, este necesară aplicarea unor măsuri de securitate pentru a evita accesul nedorit al unui utilizator neautorizat la resursele unui calculator-gazdă implicat în acea conexiune. Un server Finger poate fi configurat astfel încât o cerere Finger adresată lui de la distanță, să inițieze pe server un fișier capabil să ruleze comenzi de sistem (*scripting*).

II.1.7 Protocoale de poștă electronică (SMTP, POP)

Poșta electronică (*e-mail / electronic mail*) este unul dintre cele mai utilizate servicii de comunicații din Internet.

Într-o rețea de calculatoare, sistemul de poștă electronică are următoarele componente (Fig.II.11):

1. memorie de stocare a mesajelor ce urmează a fi transmise ("coadă de ieșire");
2. procesul client de mail;
3. procesul server de mail;
4. cutiile poștale (*mailbox*) ale utilizatorilor pentru stocarea mesajelor primite.

Programul de poștă electronică de pe un calculator, denumit **agent utilizator** (UA - *User Agent*) oferă utilizatorului o interfață spre sistemul de poștă electronică din Internet, care, în general, nu este considerată parte componentă a acestuia. Scopul UA este de a facilita accesul utilizatorului la serviciul de e-mail din rețea (Fig.II.11).

Procesele (client sau server) care realizează serviciul de poștă electronică, prin transferul mesajelor în sau din Internet se numesc **agenți de transfer a mesajelor** (MTA - *Message Transfer*

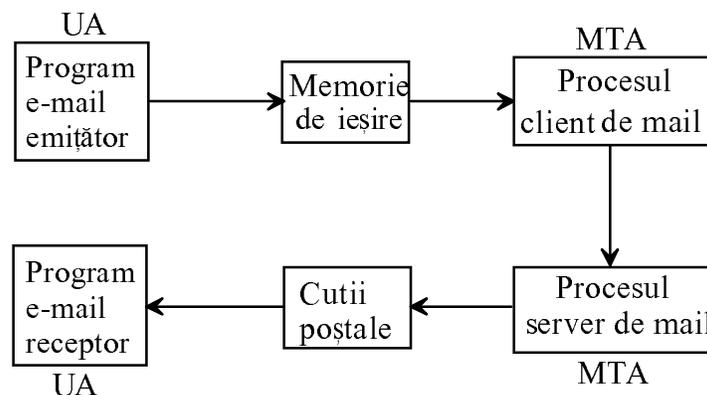


Fig. II.11 Structura sistemului de poștă electronică dintr-o rețea de calculatoare

Agent). Între doi agenți de transfer a mesajelor se stabilește o conexiune TCP, iar comunicarea se face pe baza protocoalelor de poștă electronică (de exemplu, SMTP - *Simple Message Transfer Protocol*, POP - *Post-Office Protocol*).

Majoritatea MTA folosesc SMTP pentru transferul mesajelor în format NVT ASCII, prin conexiuni TCP la portul 25 (RFC 821).

SMTP permite comunicații duplex între client și server, pentru transmiterea secvențelor de comandă în format NVT ASCII (Tabel II.5).

Tabel II.5

Comenzi ale protocolului SMTP

Comandă	Semnificație
HELO	identifică emițătorul SMTP față de receptorul SMTP (<i>hello</i>)
MAIL	inițiază o tranzacție de poștă (<i>mail</i>) către o cutie poștală
RCPT	recepționează datele la adresa specificată (<i>reception</i>)
DATA	avertizează receptorul că urmează să se transmită date
RSET	abanonează tranzacția poștală curentă (<i>reset</i>)
NOOP	a nu se efectua nici o operație și returnează mesajul OK (<i>no operation</i>)
QUIT	returnează OK și închide canalul de comunicație
VRFY	verifică dacă argumentul identifică un utilizator (<i>verify</i>)
SEND	distribuie datele unui terminal
SOML	transmite poșta unei cutii poștale sau unui terminal (<i>send or mail</i>)
SAML	transmite poșta unei cutii poștale și unui terminal (<i>send and mail</i>)
EXPN	furnizează în detaliu componența unei liste de mail (<i>expand</i>)
HELP	transmite informații ajutătoare emițătorului SMTP
TURN	solicită schimbarea rolurilor între transmițătorul și receptorul SMTP

Numai primele șapte comenzi din acest tabel sunt obligatorii, celelalte fiind opționale pentru implementările SMTP (RFC 1123).

Sintaxa comenzii MAIL (similară pentru SEND, SOML, SAML) este următoarea:

MAIL FROM: <cale inversa>

Argumentul < cale inversă> este dat prin adresa de e-mail a expeditorului.

Sintaxa comenzii de recepție RCPT este:

RCPT TO: <cale directă>

Argumentul < cale directă> se înlocuiește cu adresa de e-mail a destinatarului.

La orice comandă SMTP, serverul MTA transmite un cod de răspuns SMTP, eventual cu argument, cu semnificațiile date în tabelul II.6.

Observații

1. Se poate utiliza SMTP pentru transmisia mesajelor de tip text, dar și de altă natură (imagine, audio, video) prin definirea unor extensii ale protocolului.
2. SMTP este implicat în transmisia poștei electronice spre serverul de mail și spre Internet. Preluarea mesajelor de poștă de la cutiile poștale sau din Internet se face conform protocolului POP.

Tabel II.6

Coduri de răspuns SMTP

Cod	Semnificație
211	specifică starea sistemului sau este un răspuns la o comandă HELP
214	mesaj HELP
220	<domeniu> serviciu de poștă pregătit
221	<domeniu> serviciul închide canalul de comunicație
250	acțiune încheiată cu succes
251	nu există local acel utilizator; avans la <cale directa>
354	inițiază acțiunea de mail și se încheie cu <CRLF>
421	<domeniu> serviciul nu este disponibil; se închide canalul
450	cutie poștală nedisponibilă temporar
451	acțiune abandonată din cauza unei erori locale de prelucrare a datelor
452	acțiune ne finalizată din cauza memoriei insuficiente
500	eroare de sintaxă; comandă necunoscută
501	eroare de sintaxă a parametrilor sau argumentelor
502	comandă neimplementată
503	secvență incorectă de comenzi
504	parametru meimplementat
550	cutie poștală invalidă
551	nu există local acel utilizator; încearcă la <cale directă>
552	acțiune abandonată prin depășirea spațiului de stocare alocat
553	acțiune nefinalizată: numele cutiei poștale inexistent
554	tranzacție de poștă eșuată

Aceste coduri sunt formate din trei cifre, fiecare dintre acestea având o anumită semnificație.

Prima cifră poate fi:

- 1 MTA server acceptă comanda dar solicită clientului să confirme informația de răspuns.

- 2 MTA server a încheiat cu succes acțiunea solicitată și clientul poate iniția o nouă cerere.
- 3 MTA server acceptă comanda dar solicită furnizarea unor informații suplimentare.
- 4 MTA server nu acceptă comanda din cauza unei stări de eroare temporare. Clientul poate repeta comanda ulterior.
- 5 MTA server nu acceptă în mod categoric cererea clientului.
Clientul nu trebuie să repete acea comandă decât dacă modifică argumentul sau parametrii.

A doua cifră indică:

- 0 eroare de sintaxă (comandă prea lungă sau neimplementată, argument invalid etc)
- 1 solicitarea de informații ajutătoare (*help*)
- 2 starea sistemului (pregătit sau nu, urmând să se închidă canalul)
- 3, 4 neutilizate în prezent
- 5 în mod exclusiv, acțiunea de mail (încheiată cu succes sau abandonată)

A treia cifră indică tipul textului descriptiv transmis după codul de răspuns.

Observații

1. În prezent, algoritmi SMTP nu acceptă ca prima cifră să fie 1, dar ea este definită în protocol pentru utilizări ulterioare.

2. Răspunsurile SMTP pot fi generate pe mai multe linii. În acest caz, este obligatoriu ca fiecare linie să înceapă cu codul 123 urmat de o cratimă și să se termine cu combinația de caractere CRLF. Numai pe ultima linie după codul de răspuns 123 nu mai apare acea cratimă, lipsa ei atenționând receptorul că este ultima linie din răspuns.

3. Pentru securizarea sistemului de e-mail, se poate configura serverul de mail să lucreze ca agent de comutație (*relay host*) sau hub de mail, astfel încât argumentele <cale directă> (*forward path*) și <cale inversă> (*reverse-path*) din comenzile SMTP adresate unei rețele să fie tocmai adresa acestui server. În comenzile MAIL și RCPT se poate impune calea ce o urmează un mesaj în rețea prin specificarea agenților de comutație, asemenea rutării pachetelor IP:

MAIL FROM: <@HOST1, @HOST2, ... @HOSTn: adresa_cutiei_postale_de_destinatie>

În procesul de transmisie, la nivelul fiecărui agent de comutație, acesta șterge numele său din calea directă și îl adaugă în calea inversă. Astfel la hostul de destinație, calea directă conține numai adresa cutiei poștale locale.

SMTP impune anumite limite referitoare la lungimea maximă a șirurilor de caractere utilizate (Tabel II.7). De asemenea, SMTP poate gestiona maximum 100 de buferi de poștă electronică.

Tabel II.7

Lungimea maximă a șirurilor de caractere admisă de SMTP

Obiect SMTP	Lungime maximă (caractere)
Nume de utilizator	64
Domeniu	64
Cale	256
Linie de comandă	512
Linie de răspuns	512
Text descriptiv	1000

Exemplu: Tranzacție SMTP între utilizatorul "expeditor" din domeniul abc.ro, către un utilizator "destinatar" din domeniul xyz.ro:

1. MTA SERVER 220 xyz.ro Simple Mail Transfer Service Ready
2. MTA CLIENT HELO abc.ro
3. MTA SERVER 250 xyz.ro
4. MTA CLIENT MAIL FROM: <expeditor@abc.ro>
5. MTA SERVER 250 OK
6. MTA CLIENT RCPT TO: <nume_eronat@xyz.ro>
7. MTA SERVER 550 No such user here
8. MTA CLIENT RCPT TO: <destinatar@xyz.ro>
9. MTA SERVER 250 OK
10. MTA CLIENT DATA
11. MTA SERVER 354 Start mail input; end with <CRLF>. <CRLF>
12. MTA CLIENT 123 - linia 1 din mesaj <CRLF>
13. MTA CLIENT 123 linia 2 din mesaj <CRLF>
14. MTA SERVER 250 OK
15. MTA CLIENT QUIT
16. MTA SERVER 221 xyz.ro Service closing transmission channel

Orice mesaj de e-mail este compus din **antet** și **corp** (RFC 822). În antetul introdus de UA și transmis către MTA, în funcție de programul de e-mail folosit, pot să apară mai multe câmpuri (*Date, From, Subject, Reply-To, cc, Atachment, Comment, Message-ID, X-Special-Action* etc). La nivelul MTA se împachetează mesajul cu informații suplimentare (eventual se specifică o cale cu agenți de comutare) în așa-numita **anvelopă** și se transmite altui MTA sau agentului de comutare local.

Se pot defini extensii ale protocolului SMTP pentru transmisia prin sistemul de e-mail a unor documente de tip multimedia (RFC 1425, RFC 1427, RFC1521).

Protocolul MIME (*Multipurpose Internet Mail Extension*) reprezintă o extensie a SMTP. Acest protocol adaugă cinci noi câmpuri în antet, care permit transmitia prin atașare la mesajele de e-mail a textelor formatate (de exemplu, RTF - *Rich Text Format*), a imaginilor cu diverse formate (GIF - *Graphic Interchange Format*, JPEG - *Joint Photographic Experts Group*), a datelor în format binar sau PS (*Postscript*), a documentelor audio în format ISDN (*Integrated Services Digital Network*) cu lege de compandare μ și codare pe 8 biți precum și a fișierelor video în format MPEG (*Movie Photographic Experts Group*).

MIME prevede pe lângă formatul standard NVT ASCII pe 7 biți, diferite scheme de codare a caracterelor, utile pentru transmisii pe 8 biți, pentru criptarea datelor sau pentru transmitia unor caractere speciale, neincluse în setul ASCII:

<i>Quoted -printable</i>	pentru transmitia în format NVT-ASCII, a unei secvențe cu un număr mic de caractere având al 8-lea bit setat, fiecare dintre acestea fiind precedat de semnul 'egal' (=) (RFC 1522)
<i>Base 64</i>	codează 3 octeți ca 4 simboluri de 6 biți (RFC 1421)
<i>8-bit</i>	transmite caracterele (ASCII și non-ASCII) pe 8-biți
<i>Binary</i>	transmisie pe 8-biți fără caracterul sfârșit de linie.

Codarea *Base 64* se poate aplica oricărui fișier audio, video sau de imagine pentru a fi transmis prin e-mail, în format binar.

SMTP este responsabil de livrarea poștei electronice în Internet. Pentru preluarea mesajelor de e-mail din Internet se utilizează **protocolul POP**, orientat pe conexiunea dintre client și server. (Figura II.12).

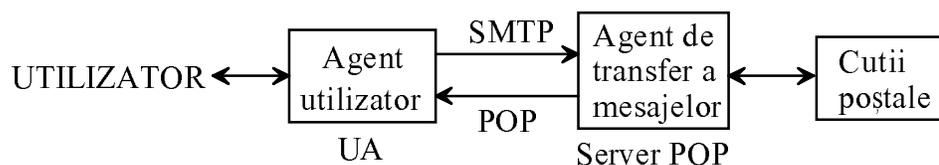


Fig. II.12 Configurația client-server POP

Deși oferă servicii relativ similare, versiunile POP2 (RFC 937) și POP3 (RFC 1225) sunt incompatibile și au asociate porturi de protocol diferite (109, 110).

POP2, corespondent fidel al SMTP, având comenzi asemănătoare, se aplică numai local.

Din motive de securitate, unii administratori de rețea dezactivează serviciul de acces de la distanță (*remote login*) în LAN.

Versiunea POP3 permite accesul de la distanță la serviciul de poștă electronică, mai precis la căsuța poștală rezervată utilizatorului.

O sesiune POP3 are trei stări: autorizare, tranzacție și actualizare (încheiere).

Pentru autorizarea accesului de la distanță la căsuța poștală a utilizatorului, aflată pe serverul POP3, protocolul prevede comenzi de identificare, cu nume de utilizator (USER) și parolă (PASS).

După autorizare, se poate efectua tranzacția, folosind comenzile:

STAT	Serverul informează clientul despre numărul de mesaje din cutia poștală și dimensiunea totală (<i>status</i>)
LIST	Afișează lista mesajelor (<i>listing</i>)
RETR	Preia un mesaj cu un anumit ID (<i>retrieve</i>)
DELE	Marchează un mesaj din cutia poștală pentru ștergere (<i>delete</i>)
RSET	Deselectează toate mesajele marcate pentru ștergere (<i>reset</i>)
LAST	Serverul returnează ID-ul ultimului mesaj accesat (<i>last message</i>)
NOOP	Serverul răspunde simplu (OK) fără a efectua vreo operație, comanda fiind utilă pentru testarea legăturii (<i>no operation</i>).

Închiderea sesiunii presupune ștergerea mesajelor și dezactivarea legăturii, cu comanda:

QUIT	Actualizează starea cutiei poștale și închide conexiunea TCP.
------	---

II.1.8 Protocoale pentru transferul fișierelor (FTP, TFTP, SFTP)

Serviciul de transfer a fișierelor în rețelele locale de calculatoare și în Internet se realizează pe baza protocoalelor de transfer a fișierelor, precum FTP (*File Transfer Protocol*), TFTP (*Trivial File Transfer Protocol*), SFTP (*Simple File Transfer Protocol*).

Protocoalele FTP (RFC 959) și SFTP (RFC 913) folosesc TCP pentru transport, ceea ce permite transferul sigur al datelor la destinație.

Protocolul TFTP (RFC 783) este implementat cu datagrame, pe baza UDP, astfel încât transferul devine nesigur, dar este simplu și rapid. TFTP este utilizat pentru transferul unor fișiere de mici dimensiuni.

Pentru transferul fișierelor în rețea, se poate utiliza și protocolul SSH (*Secure Shell Protocol*), cu cheie de criptare pentru securizarea transmisiei.

Protocolul FTP folosește coduri de comandă și de răspuns în formatul NVT ASCII, definit de Telnet, și două conexiuni TCP: **conexiunea de date** pentru transferul datelor și **conexiunea de control** pentru transmisia unor comenzi specifice.

Implementarea FTP se realizează pe baza modelului client-server în sisteme de operare diverse (UNIX, DOS, Windows), prin programe simple, de tip 'linie de comandă', sau complexe, cu

interfață grafică de utilizator (GUI - *Graphic Unit Interface*) și structură de meniuri, comode din punctul de vedere al utilizatorilor.

Serverul FTP deschide o conexiune pasivă la portul de protocol 21, după care așteaptă cererile clienților.

O sesiune FTP se activează prin cererea clientului de stabilire a unei conexiuni TCP, la portul de protocol 21 al serverului, reprezentând **conexiunea de control**, activă pe toată durata comunicării.

Dacă serverul răspunde afirmativ, cu un cod de răspuns 220, se inițiază **faza de autentificare**, prin operația de *login*, cu validarea numelui de utilizator (cod de răspuns 331) și a parolei proprii (cod de răspuns 230).

După eventuala schimbare a directorului de lucru curent, pentru transferul datelor se deschid **conexiuni de date** separate pentru fiecare operație de transfer de fișiere, spre server (*upload*) sau dinspre server (*download*). Clientul comunică serverului numărul de port la care să se conecteze, prin comanda FTP PORT, și deschide o conexiune pasivă. Serverul activează conexiunea pe portul specificat de client. În această fază, clientul joacă rolul de server pe conexiunea de date. Spre deosebire de serverul FTP, care pe o conexiune pasivă acceptă accesul oricărui client, clientul FTP nu va accepta pe conexiunea pasivă inițiată de el decât accesul de la adresa serverului FTP cu care a comunicat.

Procesul de preluare de fișiere (*download*) se inițiază prin comanda RETR (*Retrieve*) iar cel de transfer a unui fișier pe server (*upload*) prin comanda de stocare STOR (*Store*). Închiderea unei sesiuni FTP se realizează prin comanda QUIT. Procesul de transfer al unui fișier poate fi întrerupt în orice moment la solicitarea clientului, în modul de date urgente.

Codul 220 de răspuns la serverului marchează stabilirea conexiunii de date.

Codul 550 semnifică blocarea accesului pe server (*Access denied*).

O configurație tipică FTP este reprezentată schematic în figura II.13.

Clientul și serverul utilizează fiecare câte o interfață de interpretare a protoacoalelor (PI - *Protocol Interpreter*), între acestea fiind stabilită conexiunea de control pe toată durata transferului.

Transferul fișierului se realizează prin conexiunea de date care se stabilește între procesele FTP-client și FTP-server.

Observații

1. Dacă un client nu dispune de un cont de utilizator pe un anumit server de fișiere, atunci se poate folosi FTP în varianta anonimă (*Anonymous FTP*), cu numele de utilizator *anonymous* și parola *guest*.

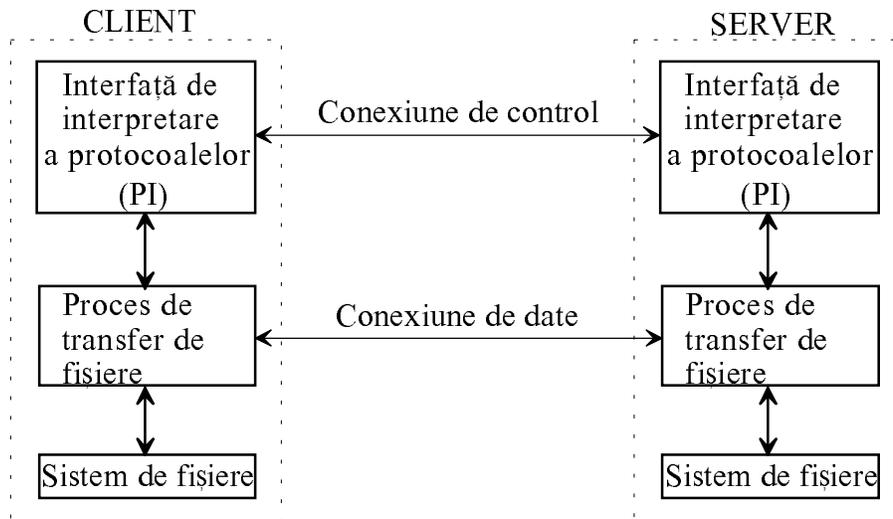


Fig. II.13 Schema de conexiuni FTP

2. Asemenea protocoalelor de poștă electronică (SMTP, POP), FTP este un protocol de tip 'pas-cu-pas' (*lock step*), ceea ce înseamnă că după fiecare comandă lansată se așteaptă un răspuns și abia după primirea acestuia se transmite următoarea comandă.

3. FTP folosește Telnet pe conexiunea de control.

4. SFTP folosește o singură conexiune TCP și permite:

- ♦ validarea utilizatorilor pe baza listelor de control a accesului (*ACL - Access Control List*);
- ♦ transferul sigur al fișierelor;
- ♦ operații de listare, redenumire, ștergere de fișiere etc.

Totuși SFTP nu s-a impus.

5. Pe conexiunea de date se pot transfera fișiere între client și server, liste de directoare sau fișiere de pe server.

Protocolul FTP prevede peste 30 de comenzi, împărțite în trei categorii:

I. de control a accesului;

II. de negociere a parametrilor (opțiunilor);

III. de serviciu.

Comenzile FTP de control a accesului sunt următoarele:

USER {nume de utilizator} - nume de utilizator (*user name*)

PASS {parola} - parolă (*password*)

ACCT {nume/numar de cont} - cont de utilizator (*account*)

CWD - schimbarea directorului curent (*change working directory*)

CDUP	- trecere la directorul-părinte (<i>change to parent directory</i>)
SMNT	- stabilirea structurii de fișiere (<i>structure mount</i>)
REIN	- reinițializează programul-client (<i>reinitialise</i>), păstrând deschisă conexiunea de control pentru alți clienți
ABORT	- abandonarea transferului de fișiere
QUIT	- închiderea conexiunii.

Comenzile de negociere a parametrilor stabilesc opțiunile clientului. Protocolul FTP prevede patru **categorii de opțiuni** negociabile între client și server:

a. **tipuri de fișiere:**

local - transfer între calculatoare care folosesc cuvinte cu număr diferit de biți (L);

binar (image) - realizează transferul datelor din fișier ca flux de octeți (I);

EBCDIC (*Extended Binary Coded Decimal Interchange Code*) (E);

ASCII - tipul implicit pentru FTP, definit de Telnet pentru NVT (A).

b. **formate de fișiere:**

neprintabil (*nonprint*) - format implicit pentru FTP, care nu conține elemente de formatare verticală a textelor (N);

TELNET - folosește informația de control pentru formatare verticală;

FORTTRAN - prevăzut cu caractere speciale de delimitare (CR - *Carriage Return*) în text (C).

c. **structuri de fișiere:**

structură tip "înregistrare" (R - *Record*) - sfârșitul înregistrării și al fișierului este marcat cu o combinație de două caractere;

structură tip "fișier" (F - *File*) - sfârșitul fișierului coincide cu închiderea conexiunii TCP;

structură tip "pagină" (P - *Page*).

d. **moduri de transmisie**

bloc - fișierele sunt transferate ca succesiuni de blocuri de date, cu antete proprii (B);

comprimat - fișierele sunt comprimate prin precodare RLL (*Run-Length Limited*) în vederea transferului (C);

flux de octeți (*stream*) - transferul fișierelor se face continuu, ca șir de octeți (S).

În general, FTP folosește fișiere de tip binar sau ASCII, formatul neprintabil, structura de tip "fișier" și modul de transmisie ca flux de octeți.

Următoarele **comenzi de stabilire a opțiunilor** de transfer semnifică:

TYPE	- tipul fișierului;
STRU	- structura de fișiere (<i>structure</i>);
MODE	- modul de transfer;
PORT	- portul de date al clientului;
PASV	- clientul deschide o conexiune pasivă pe portul specificat.

Comenzi FTP de serviciu:

RETR	- preluare de fișier de pe alt calculator (<i>retrieve</i>);
STOR	- stocare de fișier pe alt calculator (<i>store</i>);
STOU	- stocare de fișier cu nume unic în director (<i>store unique</i>);
APPE	- adaugă date la un fișier aflat pe alt calculator (<i>append with create</i>);
ALLO	- alocă spațiu de memorie pe discul altui calculator (<i>allocate</i>);
REST	- reia procesul de transfer de date dintr-un fișier fără a retransmite datele deja transferate (<i>restart</i>);
ABOR	- abandonează procesul de transfer al unui fișier (<i>abort</i>);
DELE	- șterge un fișier de pe un alt calculator (<i>delete</i>);
LIST	- cere serverului lista fișierelor dintr-un director (<i>list</i>);
MKD	- creează un nou director pe alt calculator (<i>make directory</i>);
RMD	- șterge un director de pe alt calculator (<i>remove directory</i>);
SYST	- permite determinarea tipului sistemului de operare folosit pe alt calculator (<i>system</i>);
HELP	- clientul cere serverului informații ajutătoare referitoare la un argument (<i>help</i>);
NOOP	- testează conexiunea de control dintre client și server (<i>no operation</i>).

FTP folosește **coduri de răspuns** de trei cifre, asemenea SMTP, fiecare cifră având o anumită semnificație.

Prima cifră din codul de răspuns FTP semnifică tipul răspunsului serverului:

- 1 - preliminar pozitiv;
- 2 - definitiv pozitiv;
- 3 - intermediar pozitiv;

4 - temporar negativ;

5 - definitiv negativ.

Cea de a doua cifră reprezintă tipul textului care însoțește acel cod numeric de răspuns:

0 - eroare de sintaxă;

1 - informație;

2 - conexiuni;

3 - autentificare;

4 - nedefinit;

5 - sistem de fișiere.

A treia cifră din codul de răspuns FTP selectează textul descriptiv atașat acestuia, dintr-o listă de texte posibile.

Exemple de coduri de răspuns FTP:

125	conexiune de date deschisă; urmează transferul unui fișier
200	mesaj OK
220	serviciu pregătit pentru un nou client
230	numele și parola utilizatorului au fost validate; continuă
331	nume de utilizator validat; introdu parola
426	conexiune închisă; transfer abandonat
500	eroare de sintaxă; comandă necunoscută
502	comandă neimplementată
553	acțiune neexecutată; nume de fișier nepermis.

Comparativ cu FTP, **protocolul TFTP** utilizează un set restrâns de comenzi doar pentru scrierea sau citirea unui fișier, este mult mai simplu și poate fi folosit pentru transferul unor fișiere de dimensiuni mici, între echipamente interconectate direct (de exemplu, între două calculatoare sau între un calculator și un router).

Fișierele sunt transferate fragmentat, în datagrame cu lungime de 512 octeți, cu excepția ultimei care prin dimensiunea sa redusă marchează sfârșitul transmisiei.

Deoarece transportul datelor se face prin UDP și este nesigur, se impune aplicarea unor coduri de corecție a erorilor la nivelul antetului datagramei pentru a se găsi destinația corectă.

TFTP este utilizat și pentru transferul fișierelor de configurare (*boot*) din sistemul de operare, pe calculatoarele dintr-o rețea, la pornirea acestora (RFC 906). TFTP a devenit protocolul standard Internet pentru copierea programelor de boot pe stațiile de lucru fără disc, de pe un server de fișiere TFTP.

Protocolul SSH (*Secure Shell Protocol*) este un protocol utilizat pentru servicii de poștă electronică, transfer de fișiere și acces de la distanță, pe portul de aplicații 22, în baza modelului client-server, cu autentificarea utilizatorului și criptarea mesajelor. Acest protocol folosește algoritmi de criptare cu chei publice pentru clienți și server de 512 - 2048 biți, algoritmi DES (*Data Encryption System*) și 3DES (*Triple DES*). Este utilizat cu succes pentru transferul mesajelor între procesele de management din rețea.

II.1.9 Protocoale de timp pentru Internet

Pentru sincronizarea ceasurilor interne ale calculatoarelor conectate în rețea, se pot utiliza diferite protocoale de timp:

- ◆ NTP - *Network Time Protocol* (RFC 1305);
- ◆ SNTP - *Simple Network Time Protocol* (RFC 1361);
- ◆ *Daytime Protocol* (RFC 867);
- ◆ *Time Protocol* (RFC 868).

Aceste protocoale sunt rulate pe serverele de timp care, la cererea clientului, furnizează informații de timp, cu anumite precizii, folosind diferite referințe de timp.

De exemplu, PC-urile calculează timpul curent ca numărul de secunde scurse de la 1 ianuarie 1970, ora 00:00 GMT. În Internet, se ia ca referință data de 1 ianuarie 1990, ora 00:00 GMT. Momentele luate ca referințe de timp pot să creeze incompatibilități în procesele de comunicație din Internet.

Cunoașterea timpului în rețea este deosebit de utilă pentru majoritatea programelor rulate pe calculatoare, pentru determinarea duratelor unor procese de comunicație, datarea documentelor, calculul termenelor de expirare a licențelor, analiza statistică a transferurilor de date etc.

Întrucât protocoalele de timp generează date numerice în format întreg, apar probleme legate de respectarea **ordinii octeților** în rețea și la nivelul calculatoarelor-gază.

În Internet, se transmite octetul mai semnificativ înaintea celui mai puțin semnificativ (ordinea *big indian*). Unele procesoare (Motorola, IBM, Macintosh) recunosc această ordine.

Alte procesoare, precum cele produse de firmele Intel și DEC (*Digital Equipment Corporation*), folosesc ordinea inversă (*little indian*), fiind necesară conversia formatului datelor la preluarea lor din Internet.

Protocolul DAYTIME generează informația de timp într-un format accesibil utilizatorului uman, precum:

<numele zilei><luna><ziua (numeric)><ora:minutul:secunda><anul>

Exemplu: Saturday, July 05, 10:23:30, 2003.

Șirul de caractere se încheie cu combinația CRLF specifică formatului NVT ASCII.

Protocolul TIME este unul dintre cele mai simple protocoale de timp folosite în Internet. Serverul *TIME* furnizează informația de timp curent ca număr de 4 octeți (*long integer*), reprezentând numărul de secunde scurse de la 1 ianuarie 1900, ora 12:00:01 AM până în prezent. Prin urmare, apare o diferență de 70 de ani față de referința de timp utilizată de un server Internet. Aceasta corespunde valorii hexazecimale exprimate ca întreg fără semn 0x.83.AA.7E.80.

Observație

În literatura de specialitate, există convenția ca toate valorile hexazecimale să fie precedate de prefixul '0x', care nu se ia în considerare în calculul lungimii întregului ca număr de octeți. De exemplu, 0x.23 reprezintă un singur octet (23) exprimat în format hexazecimal.

Comunicația dintre clientul și serverul *TIME* se realizează în mai mulți pași:

Pas 1. Serverul de timp 'ascultă' portul de aplicații 37 în așteptarea cererii unui client, transmisă fie prin TCP, fie prin UDP.

Pas 2. Dacă un client deschide o conexiune la portul 37 sau serverul recepționează o datagramă vidă pe acest port, atunci serverul de timp trimite clientului informația privind data și ora curentă, sub forma unui număr de 32 de biți, fie prin TCP după care închide conexiunea, fie printr-o datagramă UDP.

Pas 3. Clientul *TIME* recepționează datele și, în cazul utilizării TCP, închide conexiunea.

Protocoalele *DAYTIME* (portul 13) și *TIME* (portul 37) pot lucra atât cu TCP, cât și cu UDP.

NTP, dezvoltat din protocolul *TIME*, în mai multe versiuni (RFC 958, RFC 1305), corespunde portului de protocol 123 și este cel mai complex dintre protocoalele de timp utilizate în Internet, având o precizie de 1 - 50 milisecunde față de timpul standard oficial. Folosește UDP ca protocol de transport, IP cu sume de control Internet și ICMP (RFC 792) pentru transmisia mesajelor de interogare privind informația de timp (*timestamp message*).

SNTP utilizează algoritmi simplificați față de NTP, care nu asigură aceeași precizie de generare a informațiilor de timp.

În Internet, se utilizează o structură ierarhică de servere NTP. Serverele primare generează informația de timp cu precizie maximă și se interoghează reciproc, periodic, pentru sincronizare. Serverele NTP secundare își sincronizează ceasul prin cereri periodice adresate serverelor primare de timp. Serverele secundare furnizează clienților informații de timp mai puțin precise din cauza erorilor care apar în procesul de comunicație.

Este necesar ca între oricare două calculatoare (servere sau hosturi) care comunică în rețea prin protocoale de timp, să se procedeze la sincronizarea periodică a ceasurilor interne, în pereche.

Observație: Este dificilă estimarea cu precizie a duratei de transfer în rețea a pachetelor IP de la sursă la destinație, chiar dacă se folosește NTP. Operația este necesară pentru analiza statistică a proceselor de comunicație care se desfășoară în rețea.

II.1.10 Protocolul ICMP

Protocolul ICMP (*Internet Control Message Protocol*) corespunde nivelului Internet din modelul TCP/IP, fiind responsabil de transmisia mesajelor de eroare sau de interogare, utile pentru procesele de comunicație din rețea, dar și pentru testarea și depanarea acesteia.

Aplicațiile comunică direct cu ICMP și nu prin intermediul protocoalelor de transport.

Pentru încapsularea mesajelor și transmisia lor în rețea sub formă de pachete, ICMP folosește protocolul Internet.

Protocolul ICMP are unele limitări:

1. Raportează eroarea de transmisie a unui pachet numai sursei.
2. Nu transmite mesaje de eroare asociate altor mesaje ICMP.
3. Nu generează mesaje pe adrese de destinație de tip 'broadcast' sau 'multicast', pentru a evita încărcarea rețelei prin apariția fenomenelor de "furtună-în-rețea" (*broadcast storm*).
4. Nu furnizează mesaje de eroare pentru toate fragmentele unei datagrame eronate, ci numai pentru primul.

ICMP folosește 15 tipuri de mesaje. În general, modulele software care implementează ICMP nu includ decât anumite mesaje (Tabel II.8).

Împachetarea mesajelor ICMP se face cu un antet de 4 octeți, în care se specifică tipul mesajului, codul de răspuns și suma de control Internet (Fig. II.14).

Lungimea și conținutul câmpului de date ICMP depind de tipul mesajului.

Pentru mesaje de eroare, în acest câmp se transmit primii 64 de octeți din pachetul IP care a provocat eroarea.

Pentru mesaje de interogare de tip 'ecou' (0), folosite pentru testarea conexiunilor de către aplicația PING (*Packet INternet Groper*), câmpul de date ICMP include un octet de identificare, un octet 'număr de secvență' și alți doi octeți reprezentând o secvență opțională de date.

0	8	16	31
Tipul mesajului	Codul de eroare	Suma de control	Date ICMP

Fig. II.14 Formatul mesajului ICMP

Tabel II.8 Mesaje ICMP

Tip	Semnificație
0	Răspuns la ecou (<i>Echo Reply</i>)
3	Destinație inaccesibilă (<i>Destination Unreachable</i>)
5	Redirecționare (<i>Redirect</i>)
8	Cerere de ecou (<i>Echo Request</i>)
9	Informație de rutare (<i>Router Reply</i>)
10	Cerere de rutare (<i>Router Request</i>)
11	Timpul de viață a expirat (<i>Time-to-Live Expired</i>)
13	Cerere de sincronizare a ceasului (<i>Timestamp Request</i>)
14	Răspuns de sincronizare a ceasului (<i>Timestamp Reply</i>)
17	Cerere pentru masca de rețea (<i>Network Mask Request</i>)
18	Furnizarea măștii de rețea (<i>Network Mask Reply</i>)

Câmpul de date din mesaje ICMP de tip 3, "destinație inaccesibilă", conține 68 de octeți, dintre care primii patru corespund unui cuvânt de 32 de biți complet nul iar următorii 64 de octeți provin din pachetul original transmis.

Similar, pentru mesaje de redirecționare, se includ în câmpul datelor adresa IP a routerului (*next hop*) și primii 64 de octeți din pachet.

Stațiile de lucru fără disc folosesc ICMP pentru aflarea propriei măști de rețea. Mesajele ICMP de cerere și răspuns referitoare la aceasta, includ în câmpul datelor ICMP de 8 octeți, un identificator și un număr de secvență, urmate de masca de rețea propriu-zisă.

Observații

1. Suma de control Internet calculată pe 16 biți (RFC 1071) se aplică atât antetului ICMP, cât și câmpului de date.

2. La același tip de mesaj, se pot asocia coduri de răspuns diferite în funcție de cauza problemei. De exemplu, pentru destinație inaccesibilă sunt posibile 16 coduri de răspuns (rețea inaccesibilă, protocol inaccesibil, rețea de destinație necunoscută etc).

3. Mesajul ICMP este încapsulat în vederea transmisiei cu antetul IP.

4. Operația PING se realizează în două faze: prima de transmisie a cererii de ecou ICMP, a doua de recepționare a răspunsului la ecou prin intermediul ICMP. Un răspuns afirmativ, de găsire a destinației, include durata de transfer dus-întors a pachetului între sursă și destinație, exprimată în milisecunde, rata de erori etc.

5. Mesajele ICMP de interogare a routerelor sunt utilizate pentru actualizarea dinamică a tabelor de rutare.