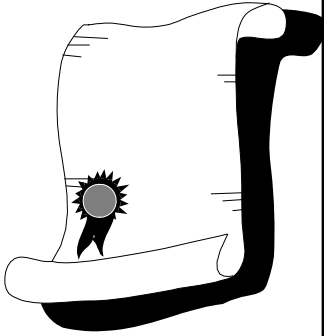


APLICAȚII



Subiecte

- 8.1. Aplicații tradiționale
 - 8.1.1. Accesarea terminalelor, protocolul TELNET
 - 8.1.1.a. Protocol de terminal virtual-VTP
 - 8.1.1.b. TELNET NVT
 - 8.1.1.c. Protocolul de transfer TELNET
 - 8.1.1.d. Mecanismul de sincronizare TELNET SYNCH
 - 8.1.1.e. Opțiunile TELNET
 - 8.1.2. Transferul de fișiere cu protocolul FTP
 - 8.1.3. Poșta electronică, e-mail
 - 8.1.3.a. Protocolul SMPT
 - 8.1.3.b. Protocolul MIME
- 8.2. Aplicații moderne
 - 8.2.1. Accesul la Web – protocolul HTTP
 - 8.2.2. Sistemul numelor de domenii DNS

Evaluare: 1. Răspunsuri la întrebări aplicații

Câteva dintre **aplicațiile tradiționale** ale Internetului sunt accesarea terminalelor (protocolul TELNET), transferul fișierelor (protocolul FTP), poșta electronică (protocolul SMPT și MIME), iar dintre **aplicațiile moderne** sunt accesarea Web (HTTP), serviciul de ghidare prin internet (DNS) și transmiterea vocii prin internet și pe suport multimedia (SIP).

8.1. Aplicații tradiționale

- Accesarea terminalelor, protocolul TELNET
- Transferul fișierelor, protocolul FTP
- Poșta electronică, protocolul SMPT și protocolul MIME

8.1.1. Accesarea terminalelor, protocolul TELNET

Este cea mai veche aplicație Internet, folosită de ARPANET din 1969.

TELNET este o facilitate de conectare la distanță bazată pe un **protocol de terminal virtual** și un **terminal virtual de rețea**.

Pentru transferul datelor, sunt translatate caracteristicile terminalului real într-un terminal virtual de rețea.

Telnet definește o procedură de negociere care permite utilizatorului și serverului să stabilească opțiunile terminalului de rețea.

Modulul 8

Aplicații

Accesul la distanță a fost prima cauză a creerii rețelelor de date. TELNET a fost proiectat într-o epocă în care utilizatorii interacționau cu calculatorul prin intermediul unor terminale simple formate dintr-o tastatură și un afișaj, plus un hardware de comunicație primitiv care permitea transmiterea fluxului de caractere în ambele sensuri. Mai multe astfel de terminale erau conectate la un calculator de comunicații, conectat direct la rețea sau la un host.

Hostul sau controlerul erau capabile să stabilească o conexiune cu hostul îndepărtat, astfel încât utilizatorul să se conecteze la distanță și să-l folosească. Problema hosturilor era cum să asocieze anumite taste (specifice terminalului) cu anumite funcții (Break sau Transmit).

Fizic, terminalele difereau prin tastatură, set de caractere, dimensiunea afișajului, lungimea liniei și viteză. Logic, interacțiunea terminal-host era guvernată de software-ul hostului, fiecare cu propria soluție pentru pornirea și oprirea proceselor, controlul fluxului, etc.

8.1.1.a. Protocol de terminal virtual-VTP

Soluția adoptată de Telnet a fost să dezvolte un **protocol de terminal virtual-VTP** (Virtual Terminal Protocol), care transformă caracteristicile terminalului real într-o formă standard, numită **terminal virtual de rețea NTV** (Network Virtual Terminal). Acesta este un dispozitiv imaginar, cu caracteristici prestabilite, și care, folosind VTP, permite stabilirea unei conexiuni între terminalul utilizatorului și hostul îndepărtat.

Ambele părți, **terminalul și hostul îndepărtat** generează **semnale de control și date în limbajul lor nativ, care sunt apoi traduse în cele specifice NTV și respectiv, traficul de intrare este retranslatat în limbajul terminalului.**

Orice VTP are patru **faze de operare**:

-Managementul conexiunii, care include cererea de stabilire și de terminare a conexiunii;

-Negocierea parametrilor conexiunii, în funcție de setul limitat de posibilități ale terminalului real și constângerile NTV (de exemplu, lungimea liniei);

-Controlul (informații de control și comenzi: sfârșit de linie, întreruperi, etc.).

-Transferul datelor între cei doi corespondenți.

Datele și semnalele de control sunt transportate de același flux secvențial de caractere, deci **TELNET nu poate accepta opțiuni grafice.**

8.1.1.b. TELNET NVT

Telnet poate fi folosit de:

- **două terminale,**
- **două procese,**
- **de un terminal și un proces.**

NVT cuprinde **implicit**, doar **opțiunile simple**, care pot fi implementate pe cele mai simple terminale. Pentru alte **facilități** există faza de **negociere a opțiunilor**.

Dacă entitatea de comunicație este un proces, este necesar un **modul server de proces** pentru a converti reprezentarea NVT și a procesului.

Dacă entitatea de comunicații este un terminal este necesar un **modul TELNET utilizator** pentru asocierea caracteristicilor terminalului la cele ale NVT.

NVT Telnet este un dispozitiv orientat pe caracter, bidirecțional, cu afișaj și tastatură: afișajul răspunde la datele care intră, iar tastatura generează datele care ies, care sunt transmise pe conexiunea TELNET și, dacă se dorește ecou, și pe afișajul NVT.

8.1.1.c. Protocolul de transfer TELNET

Deși TCP este capabil de o transmisie duplex, **datele TELNET** sunt transmise **semiduplex**. Datorită caracterului duplex al TCP, semnalele de control pot fi transmise la orice moment, indiferent de sensul de transmitere a datelor.

Datele sunt transmise ca un flux de octeți, fără alte formătări. **Semnalele de control și informațiile non-data sunt transmise ca șiruri de octeți intercalate printre date.** Comenzile pot fi ale utilizatorului, sau între procesele Telnet ale protocolului de transfer și de negociere a opțiunilor.

Fiecare **comandă TELNET** este **precedată de** caracterul **IAC** (Interpret As Command, 255) pentru a-l diferenția de datele utilizatorului sau procesului. Dacă în fluxul datelor apare octetul 255 el va fi dublat, pentru asigurarea transparenței de cod.

O comandă TELNET simplă are 2 octeți, al treilea octet fiind identificatorul opțiunii. Comanda de subnegociere a opțiunilor are lungimea variabilă, dar începe întotdeauna cu o secvență de 3 octeți (IAC SB option-id) și se termină cu o secvență de 2 octeți (IAC SE).

Modulul 8

Aplicații

Protocolul TELNET de transfer al datelor minimizează încărcarea pentru transmisie deoarece nu pretinde octeți suplimentari pentru antetele mesajelor. Oricum încărcarea e mare la procesare deoarece utilizatorul TELNET și serverul TELNET trebuie să proceseze fluxul transmis caracter cu caracter, pentru translația datelor (NVT/nativ) și căutarea comenzilor.

Exemplu.

De la terminal spre proces, caracterul de linie nouă (prompter) semnifică sfârșitul intrării utilizatorului, și la fel este TELNET Go Ahead, dinspre proces spre terminal. Când apar aceste semnale, receptorul poate începe să transmită date. Acest lucru permite utilizatorului terminalului, de exemplu, să trimită un Abort Output, când procesul transmite date imprimantei.

Comenzi Telnet

IAC (Interpret as Command) (255) interpretează octetul următor ca o comandă;

Semnale de control ale utilizatorului

AYT (Are You There?) (246) cere o confirmare ca partea distantă mai e acolo;

IP (Interrupt Process) (244) intrerupe procesul de la distantă

AO (Abort Output) (245) cere terminarea rulării; se sterg si datele neimprimare încă;

EC (Erase Character) (247) cere stergerea caracterului precedent;

EL (Erase Line) (248) cere stergerea linei precedente;

BRK (Break) (243) intrerupere sau atentie

Comenzi proces la proces

GA (Go Ahead) (249) retur de linie pentru transferul semiduplex al datelor;

DM (Data Mark) (242) sincronizarea sirului, impreuna cu SYNC;

SB (Subnegotiation Begin) (250) , incepe comanda de subnegociere;

SE (Subnegotiation End), (240) terminarea parametrilor subnegocierii;

WILL, WONT, DO, DON'T, (251-254) mesaje de negociere.

8.1.1.d. Mecanismul de sincronizare TELNET SYNCH

TCP poate urgenta datele cu pointerul URGENT DATA, receptorul fiind informat că dintr-un anumit punct sunt date urgente.

TCP transmite datele ca segmente, iar receptorul elimină antetele și le memorează. În final datele sunt pasate aplicației, imediat sau nu, astfel că în buffer pot fi date vechi așteptând să fie livrate aplicației.

Dacă segmentul care sosește are indicația urgent, TCP va alerta imediat aplicația care trece în modul „urgent” pentru ștergerea rapidă a bufferului.

Semnalul **TELNET SYNCH** permite utilizatorului să comunice procesului server **comenzi urgente**.

Exemplu

Când terminalul e conectat direct la un sistem cu divizarea timpului, utilizatorul terminalului poate apăsa tasta abort sau întrerupere, determinând procesul local să răspundă.

Dar când terminalul e conectat printr-o rețea la un calculator, apare o **întârziere**, până când semnalele de abort (AO–abort output) sau întrerupere (IP–interrupt process) ajung la proces.

Problema e parțial rezolvată cu semnalul **TELNET SYNCH** care constă dintr-o comandă **DM (Data Mark)** transmisă **într-un segment TCP** cu notificarea **urgent**. Telnet transmite comanda urmată de (IAC DM) ca date urgente.

Când TELNET destinație primește o notificare TCP urgentă, verifică fluxul de date căutând comenzile, care sunt manevrate normal, dar și descarcă toate datele. După găsirea lui DM, procesarea revine la normal.

8.1.1.e. Opțiunile TELNET

Opțiunile Telnet permit celor două părți de la capetele conexiunii să ofere **facilități suplimentare** față de NVT-ul implicit.

Opțiunile se negociază imediat la deschiderea conexiunii, pentru a oferi imediat cel mai bun serviciu posibil, dar negocierea poate avea loc și după stabilirea conexiunii.

La negociere, una din părți solicită o opțiune, iar cealaltă parte o acceptă sau nu. Dacă o acceptă, opțiunea devine imediat activă.

Modulul 8

Aplicații

Opțiunile TELNET sunt:

- de schimbarea sau îmbunătățirea caracteristicilor NVT.
- de schimbare a protocolului de transfer VTP.
- opțiuni care permit să fie definite și transmise prin conexiune informații care nu fac parte din datele utilizator sau protocolul de transfer.

0-Binary transmission	17- Entended ASCII
1-Echo	18-Logout
3-Reconnection	19-Byte macro
4-Approx Message Size Negotiation	20-Data entry terminal
5-Status	21-SUPDUP
6-Timing Mark	22-SUPDUP output
7-Remote controlled trans.and echo	23-Send location
8-Output Line width	24-Terminal type
9-Output	25- End of record
10-Output Carriage-return disposition	26-TACACS user id
11-Output horizontal tab stops	27-Output marking
12-Output horizontal	28-terminal location number
13-Output formfeed disposition	29-3270 regime
14-Output vertical tabstops	30-X.3 PAD
15-Output vertical tab disposition	31Negotiate about window size
16-Output linefeed disposition	32-Send terminal speed inform.
	33-Remote flow control

Exemplu

Opțiunile 8-16 permit definiri suplimentare ale caracteristicilor imprimantei NVT, opțiunea 20 definește un NVT nou, etc.

Opțiunea 3, Suppres Go Ahead, cere ca să nu fie folosită comanda GA, ceea ce transformă protocolul semiduplex în duplex.

Mai sunt și opțiunile care definesc comenzi noi sau caracteristici ale protocolului de transfer;

opțiunea 25, End of Record, definește o comandă Telnet pentru a indica sfârșitul intrării procesului utilizator.

Opțiunea 5, Status Option, cere ca partea distantă să raporteze starea tuturor opțiunilor negociate pentru conexiune.

Majoritatea **opțiunilor TELNET** sunt valabile **la un capăt al conexiunii** sau pentru un sens de transmitere a informației.

Dacă **o opțiune** se dorește **pentru ambele sensuri**, sunt necesare **două negocieri separate**.

Modulul 8 Aplicații

Acțiunile de validare sau de invalidare a unei opțiuni depind de respectiva opțiune.

Unele opțiuni prevăd ca la **negociere** să se verifice dacă ambele părți pot suporta opțiunea, și dacă da, urmează **subnegocierea** (format, secvențiere, interpretarea mesajelor de subnegociere).

Negocierea opțiunilor

Fiecare parte, USER-TELNET sau SERVER-TELNET, poate iniția negocierea unei opțiuni care să devină validă pe partea cealaltă. Negocierea poate cere validarea unei opțiuni noi, sau invalidarea opțiunii curente. Regulile negocierii sunt:

1. Cererea de validare a unei opțiuni poate fi respinsă.
2. Cererea de invalidare a unei opțiuni trebuie acceptată.
3. Opțiunile sunt valabile doar la sfârșitul negocierii.
4. Nu se negociază (cerere sau răspuns) ceva care e deja adevărat. Deci nu se inițiază sau răspunde la o cerere de inițiere a unei opțiuni existente.

Comenzile de negociere a opțiunilor sunt: **WILL**, **WONT**, **DO**, **DONT**, a căror interpretare depinde de context.

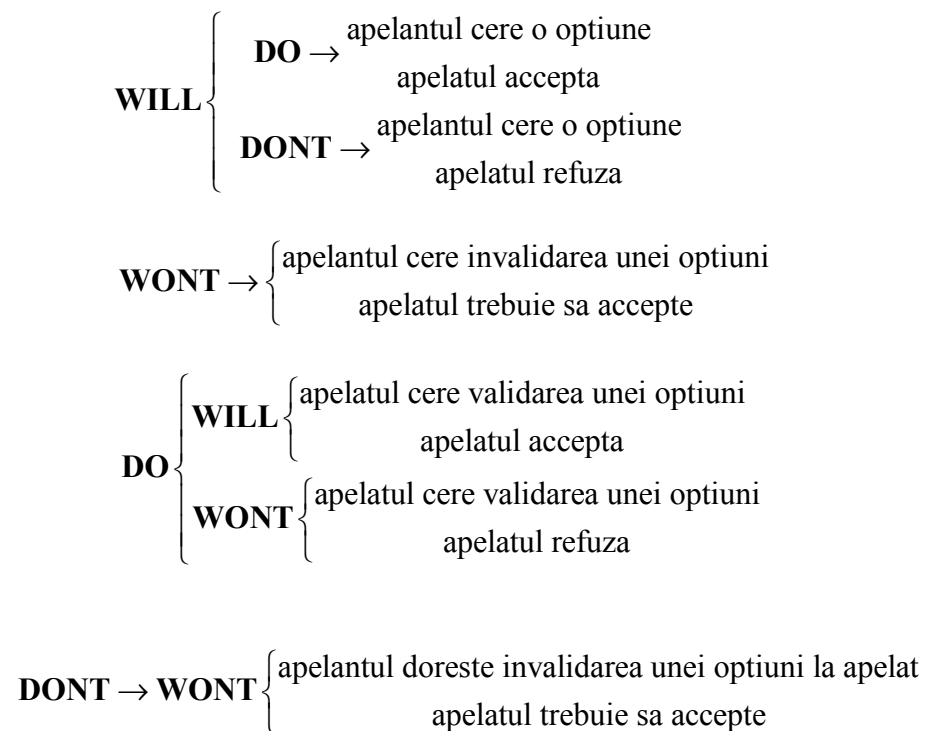


Fig. 8.1. Comenzi de negociere a opțiunilor

Exemple

-Dacă ambele părți fac aceeași cerere, ambele mesaje trec prin rețea.

-Dacă A vrea ca B să implementeze o comandă, A generează comanda DO, și dacă B e de acord răspunde cu comanda WILL.

-Dacă B vrea ca el B, să implementeze o opțiune generează o comandă WILL și dacă A e de acord răspunde cu comanda DO.

-Dacă A și B fac o cerere lui B să implementeze o opțiune, în același timp, atunci A generează o comandă DO și B o comandă WILL; aceste comenzi traversează rețeaua și ambele părți primesc un răspuns la comenzile lor .

8.1.2. Transferul de fișiere cu protocolul FTP

Ca și Telnet, FTP (File Transfer Protocol) a apărut într-o epocă cu sisteme foarte diferite, deci lucrează cu o **mare varietate de comenzi, moduri de transfer și reprezentări de date**, multe dintre ele învechite.

8.1.2.a.Obiectivele FTP sunt:

Obiectivele FTP sunt:

- promovarea partajării fișierelor (programe și/sau date);
- încurajarea indirectă, prin programe, a folosirii calculatoarelor la distanță;
- protejarea utilizatorului de variațiile sistemelor de memorare a fișierelor de pe hosturi diferite;
- transferul sigur și eficient al datelor.

Implicit, se lucrează mai mult cu **sisteme de fișiere** decât cu fișiere simple.

Un **fișier** este un simplu set de biți cu un nume, în viziunea **TFTP** (Trivial File Transfer Protocol). La acest protocol, transferul fișierului e o sarcină simplă: trimite antetul cerut pentru scrierea sau citirea unui fișier cu un nume oarecare, după care înșiră biții în rețea.

FTP consideră **sisteme de fișiere** și astfel lucrează cu metadate ca: numele de cale a fișierelor, controlul accesului și reprezentarea datelor.

8.1.2.b. Modelul FTP

Ca și TELNET-ul, FTP implică entitatea USER-FTP și entitatea SERVER-FTP.

Utilizatorul este hostul care **inițiază transferul**, el alege numele fișierului și opțiunile pentru transfer. **Serverul acceptă sau respinge cererea de transfer**, conform cu criteriul propriu de protecție a sistemului său de fișiere și cu opțiunile solicitate. Dacă cererea de transfer e acceptată, **serverul e responsabil cu stabilirea și manevrarea transferului.**

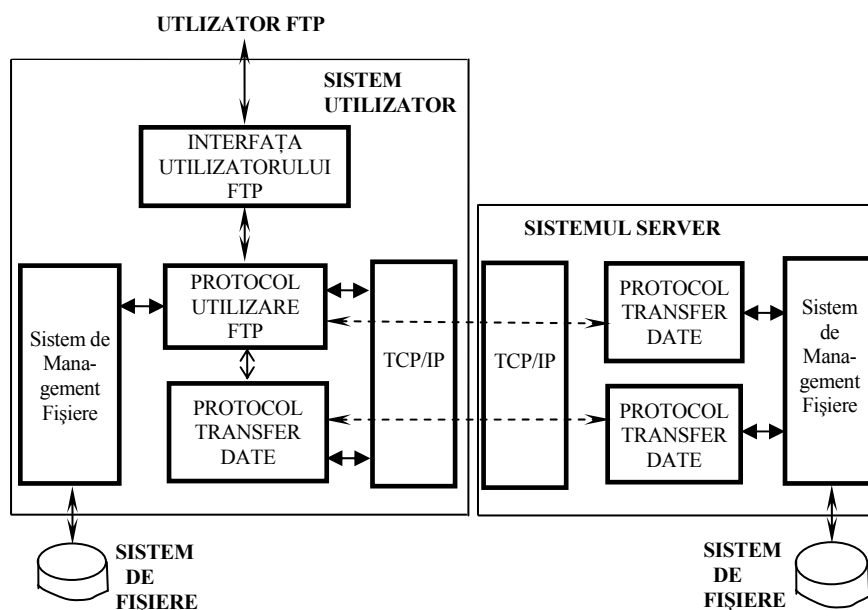


Fig.8.2. Modelul FTP

FTP operează pe **două niveluri**:

-la început, modulul de protocol User-FTP **stabilește o conexiune TCP de control** cu un modul de protocol Server-FTP, pe care schimbă comenzi FTP și răspunsuri

-când transferul de fișiere e acceptat, se stabilește **a doua conexiune TCP de transfer** a datelor și datele fișierelor sunt transferate prin ea.

Ambele niveluri ale FTP interacționează cu software-ul TCP/IP din sistemul local pentru activarea conexiunilor TCP și cu software-ul sistemului local de management a fișierelor, pentru a accesa sistemul de fișiere local și fișierele sale.

În final, există și o interfață a utilizatorului FTP, care permite unui utilizator uman sau unui program să acceseze utilizatorul FTP.

8.1.2.c. Comenzile FTP

Comenzile FTP specifică:

- **parametrii conexiunii de date** (portul de date, modul de transfer, tipul reprezentării și structura),
- **natura operării sistemului de fișiere** (memorare/store, recuperare/retrieve, comasare/append, ștergere/delete, etc) .

Comenzi și parametri de control a accesului

USER username - identifică utilizatorul pentru hostul îndepărtat;

PASS password - autentificarea utilizatorului;

ACCT account information - identifică contul utilizatorului;

REIN - un utilizator termină și șterge toate bufferele; pregătit pentru o nouă comandă USER,

Sesiune

REST marker -Treci peste fișierul specificat de markerul datelor (Restart);

SITE string -Trimite informația hostului străin, care e folosită pentru a furniza servicii specifice aceluși host

SYST - Găsirea tipului de sistem de operare la server;

STAT (pathname) - Generează răspunsul despre stare, de trimis pe conexiunea de control;

QUIT - Utilizatorul termină și închide conexiunea de control.

Parametri de transfer

PORT host-port - Specifică portul de date de folosit în conexiunile de date;

PASV - Cere serverului să asculte portul de date și să aștepte o conexiune, mai degrabă decât să inițieze o conexiune;

TYPE type-code - Specifică tipul reprezentării;

STRU structure-code - Specifică structura fișierului;

MODE mode-code - Specifică modul de transfer a datelor.

Modulul 8 Aplicații

Comenzi pentru directoare

CWD pathname	-schimba directorul de lucru
CDUP	- schimba directorul parinte
PWD	-tipareste directorul de lucru
MKD pathname	-creaza directorul
RMD pathname	-elimina directorul
SMNT	-creaza o structura de date diferita a sitemului de fisiere

Servicii FTP

RETR pathname	- recuperare, accesare, copiere; transferă fișierul de pe server la utilizator (Retrieve). * Retrieve - funcție care permite utilizatorului să copieze un fișier arhivat din memorie pe serverul de fișiere sau workstation. Copia din memoria arhivată (storage pool) nu este afectată. Este în opoziție cu arhivarea (IT Vocabulary–IBM).
STOR pathname	- memorează. Transferă fișierul de la utilizator pe server. Dacă numele de fișier specificat în numele căii există deja este în locuit (Store)
STOU	-Transferă fișierul de la utilizator pe server. Serverul creează nume de fișiere unice și le returnează utilizatorului (store unique).
APPE pathname	-atașează/comasează. Tranferă datele de la utilizator pe server. Dacă numele de fișier specificat de numele căii există deja, se atașează datele fișierului existent și dacă nu, se creează un fișier nou (append).
DELE pathname	- ștergerea fișierului (delete file).
ALLO integer [R integer]	– alocare, cerută de unele servere pentru a rezerva spațiul de memorie suficient. Primul argument este numărul de octeți. Al doilea argument, opțional, este dimensiunea paginii sau a înregistrării.
LIST pathname	-dacă numele căii specifică un director, serverul transferă lista de fișiere în director, iar dacă este un fișier serverul transferă informația curentă în fișier.
NLST pathname	- (name list) numirea listei, face ca serverul transfere o listă de fișiere.
RNFR pathname	-(rename from) redenumeste din; specifică numele de cale vechi a fișierului ce trebuie

Modulul 8

Aplicații

	redenumit. Trebuie urmată de o comandă RNTD.
HELP (string)	-determină serverul să trimită informații despre starea implementării. Argumentul opțional este un nume de comandă pentru care serverul returnează mai multe informații specifice.
NOOP	-(no-operation) serverul returnează răspusul OK.

Protocolul de transfer a datelor utilizatorului trebuie „să asculte” la un port de date specificat și serverul inițiază conexiuni de date și transferul datelor, conform cu parametrii specificați. FTP folosește protocolul Telnet pe conexiunea de control, în două moduri:

- protocolul FTP-utilizator sau protocolul FTP-server poate implementa direct regulile protocolului Telnet în propriile proceduri;
- protocolul FTP-UTILIZATOR sau protocolul FTP-SERVER poate utiliza modulul TELNET existent în sistem.

Exemplu de transfer FTP simplu fără erori

S-au folosit notațiile **UCP** pentru procesul de control al utilizatorului, (User–Control-Proces), și **SCP** pentru procesul de control al serverului (Server-Control-Proces) (fig. 8.3).

În fig. 8.3.a programul utilizatorului apelează FTP-ul utilizatorului, User-FTP. Utilizatorul dă adresa sistemului îndepărtat la care dorește accesul. Ca răspuns, protocolul User-FTP deschide o conexiune TCP, conexiunea de control, spre hostul îndepărtat. Toate comenzile și răspunsurile vor trece prin această conexiune.

În fig. 8.3.b și 8.3.c e prezentat controlul accesului. Utilizatorul (sau programul utilizatorului) furnizează numele de cont și parola sistemului local, care formează comenzile FTP și le va trimite serverului. În exemplu, numele de cont și parola sunt valide și sunt acceptate. FTP nu ia parte la procesul actual de autorizare a utilizatorului. El oferă doar modul de pasare a informației, unui mecanism oarecare din sistemul de management a fișierelor de pe server, pentru a controla accesul.

Fig. 8.3.d. User-FTP furnizează detaliile specifice transferului: numele fișierului, direcția transferului (get/put, ia sau trimite) și detalii despre tipul fișierului și modul de transfer, care sunt făcute printr-o secvență de comenzi (nereprezentate în fig. 7.3.d). Ca și la controlul accesului, s-ar putea ca utilizatorul să nu vadă toate detaliile acestui pas.

Modulul 8 Aplicații

Programul User-FTP interpretează intențiile utilizatorului și trimite o serie de comenzi FTP prin conexiunea de control.

Fig.8.3.e: deschiderea conexiunii de date. Când toate comenzile au fost recepționate și confirmate, poate începe transferul. Ambele părți sunt de acord că fișierul poate fi transferat în direcția specificată, folosind tipul datelor, tipul fișierului și modul de transfer dat de User-FTP.

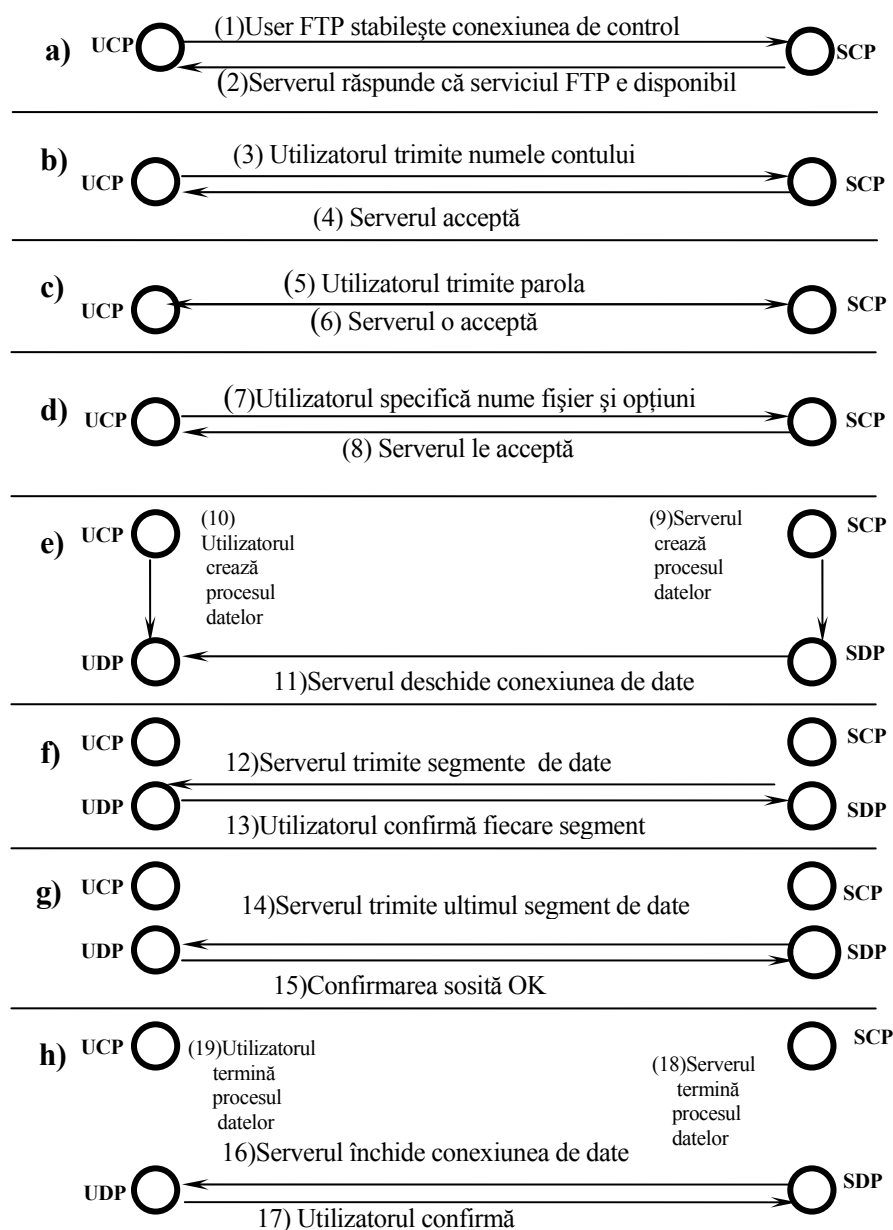


Fig.8.3. Transferul FTP

Fig. 8.3.f arată modul de transfer al datelor. Fișierul este transferat ca o secvență de segmente TCP (aici de la server la user). Serverul și userul lucrează împreună, folosind mecanismul TCP de control al fluxului și al erorilor, pentru conexiunile de date, pentru a citi

Modulul 8

Aplicații

datele din fișierul sursă, a le încapsula printr-un număr de octeți în fiecare segment TCP, și a scrie datele din segmentele recepționate în fișierul destinației la hostul utilizatorului. TCP este responsabil cu retransmisia segmentului dacă se detectează erori și pentru controlul fluxului de la sursă și de pe conexiunea TCP.

Fig. 8.3.g, 8.3.h. Când serverul a extras ultimul octet din fișierul sursă, transmite ultimul segment TCP și inițiază închiderea conexiunii. Procesul FTP user interpretează această închidere normală ca un semnal că transferul e complet și ambele părți (user și server) termină conexiunea de date. Conexiunea de control rămâne deschisă și poate fi folosită pentru controlul de date sau se poate închide în acest punct.

8.1.2.d. Opțiunile FTP

Proiectarea FTP consideră că **fișierele sunt obiecte** aflate în memoria de masă a calculatorului, care partajează anumite **proprietăți**, indiferent de tipul mașinii. Fișierele au nume simbolice care permit identificarea lor în mod unic, de către un anume sistem de fișiere sau directoare.

Un fișier are un proprietar și este protejat față de accesul neautorizat sau modificările făcute de alții. Un fișier poate fi: **citit (copiat), scris, șters, creat.**

Cu acest cadru de lucru simplu FTP reușește să transfere fișiere între diverse calculatoare prin rețele diferite. Pentru a se adapta unui calculator anumit și cu un anume sistem de operare, FTP furnizează un **mecanism de negociere a opțiunilor de transfer cu 3 dimensiuni:**

- tipul datelor,
- tipul fișierelor,
- modul de transfer

Programatorul fiecărui sistem trebuie să determine modul de asociere a unui fișier oarecare cu unul din fișierele standard.

I. Tipurile de date pot fi:

- ASCII,
- EBCDIC,
- image
- secvență logică de octeți (logical byte size).

Modulul 8

Aplicații

Tipul **ASCII** și **EBCDIC** se folosesc pentru reprezentarea fișierelor text, memorate ca un șir de caractere. Pe majoritatea mașinilor se folosesc caractere ASCII pe 8 biți, și nu mai e necesară vreo conversie de cod la sfârșit. Opțiunea EBCDIC e potrivită dacă ambele hosturi sunt IBM, folosind codul EBDIC. Fișierele de tip ASCII sau EBCDIC mai pot avea specificații suplimentare referitoare la reprezentarea lor pe imprimantă. Deoarece unele sisteme includ în textul tipărit caracterele de control pentru imprimantă, se oferă trei variante:

- **nonprint:** pentru fișiere care nu vor fi tipărite
- **telnet formatting:** caracterele de control vor fi extrase din text și folosite pentru formatarea paginii (carriage return, line feed, new line, vertical tab, form feed)
- **character control formatting:** opțiune de formatare folosită pentru limbajul FORTRAN.

Tipul **imagini (image)** este folosit pentru transferul unor fișiere arbitrare între mașini de același tip cu același sistem de operare. Un transfer de tip imagine este o replică bit-cu-bit a fișierului de pe mașina sursă, pe mașina destinație.

Tipul **secvență logică de octeți (logical byte size)** se folosește când sunt unități de date la care trebuie conservată dimensiunea. Se specifică dimensiunea în octeți (nu neapărat 8 biți). Este folosit pentru ca imaginea programului executabil, compilată pe o mașină și transmisă și memorată pe altă mașină, să fie corect interpretată și manevrată pe a doua mașină.

II. Tipuri de fișiere.

FTP definește trei tipuri de fișiere pentru a oferi o interfață convenabilă și eficientă sistemului de fișiere de la sursă și destinație. Se asigură astfel compatibilitatea cu orice sistem de operare:

a) Tipul „**file structure**” presupune că fișierul este doar un șir de octeți (definit prin opțiunea datelor) cu sfârșitul semnalat de un marker, EOF (End of File);

b) Tipul „**record structure**” se folosește când e mai convenabil ca fișierul să fie tratat ca o secvență de înregistrări (record). O înregistrare poate fi de dimensiunea agreată de hardware-ul controlerului din interfața discului sistemului. Se transmit înregistrările separate, fiecare având un marker de sfârșit (EOR – End of Record);

c) Tipul „**page structure**” e folosit pentru fișiere care sunt memorate continuu pe disc și la care trebuie păstrată structura de pagini la transfer.

Modulul 8 Aplicații

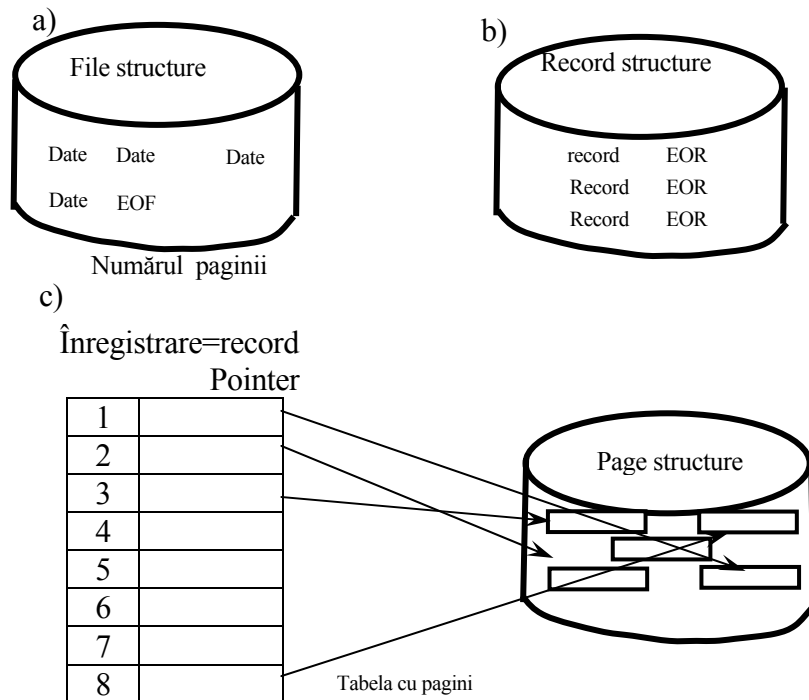


Fig.8.4. Tipuri de fișiere FTP

III. Moduri de transmisie.

Tipurile de date și de fișiere se referă la problemele sistemelor de operare de pe hosturile sursă și destinație, modul de transmisie oferă opțiuni pentru **optimizarea lucrului cu rețeaua**. Sunt definite trei moduri de transmisie:

1-Modul flux este implicit. Datele sunt transmise nemodificate pe conexiunea de date. Deoarece există procesări specializate la capetele conexiunii, acest mod cere un efort minim serverului și utilizatorului. Nu există restricții de tipuri de fișiere. Pentru fișierele cu structura „record” se folosesc 2 octeți marcând sfârșitul înregistrării și sfârșitul fișierului.

2-Modul bloc permite repornirea unui transfer eșuat (întrerupt). La apariția unei defecțiuni sau întreruperi, transferul se reia doar din acel loc și nu se retransmite întregul fișier. Dacă acest mod este acceptat, sursa înacpsulează datele în blocuri; fiecare bloc începe cu un antet din 2 câmpuri.

3-Modul comprimat oferă o cale de eficientizare a transferului permițând sursei să comprime într-un cod mai scurt secvențele de caractere de același tip. Sunt permise 4 formate diferite, dar fiecare format începe cu un antet de 8 biți:

- date necomprimate: antetul indică numărul de octeți necomprimați care urmează (maxim 127 octeți, după care este necesar un nou antet);

Modulul 8

Aplicații

- octeți replicați (copiați): când fișierul sursă conține o secvență de octeți cu aceeași valoare, ei sunt extrași și înlocuiți cu formatul replicat de octeți. Secvența de 2 octeți permite definirea unei repetiții de până la 63 octeți specificați. FTP destinație trebuie să substituie formatul octeților replicați cu numărul indicat de octeți specificat;
- șiruri de umplere (filler string): permit o compresie suplimentară pentru caractere speciale (de obicei caracterul spațiu) care apar frecvent în fișierele text. Acest format permite ca la recepție să fie inserate până la 63 asemenea caractere,
- secvența „escape” constă din octeți formați doar din zerouri urmați de octetul descriptor al codului, așa cum e definit în modul bloc.

8.1.3. Poșta electronică, e-mail

Poșta electronică (electronic mail sau e-mail) este cea mai utilizată aplicație din rețelele de comunicații. Pentru început, când se transmiteau mesaje simple, a fost suficient protocolul **SMTP** (Simple Mail Transfer Protocol), dar mai recent a fost dezvoltat protocolul **MIME** (Multi-Purpose Internet Mail Extension) pentru a permite transmiterea unor tipuri de date variate: voce, imagini, videoclipuri.

8.1.3.a Protocolul SMTP

Protocolul SMTP (Simple Mail Transfer Protocol) este protocolul standard de transfer a poștei electronice între hosturile având implementate stiva de protocoale TCP/IP (**RFC 821**). Deși formatul mesajelor transferate de SMTP păstrează de obicei formatul specificat de **RFC 822**, care va fi descrisă în continuare, nu există reguli cu privire la conținutul mesajului.

SMTP folosește doar informația din antetul mesajului, anvelopa. Singurele reguli sunt că SMTP folosește codarea ASCII, pe 7 biți a caracterelor și adaugă informație (log information) mesajului livrat, indicând calea urmata de acesta. SMTP nu are grija de formatul sau conținutul mesajului.

Operațiile de bază pentru e-mail. În figură e prezentat fluxul de e-mail într-un sistem tipic, deși o mare parte nu face parte din SMTP.

Modulul 8

Aplicații

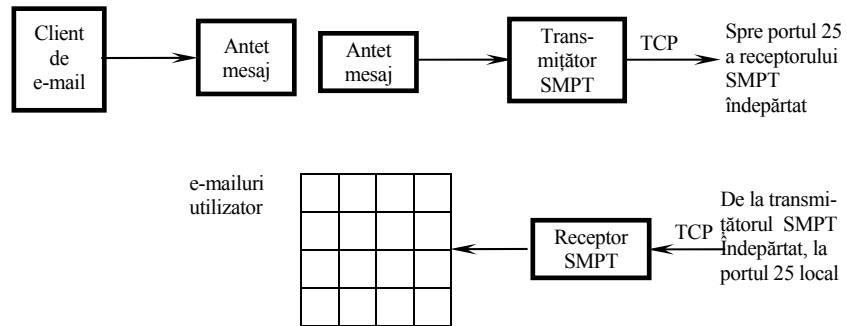


Fig.7.5. Fluxul de e-mail

La intrarea utilizator este creat mail-ul de programul agent utilizator. Fiecare **mesaj** are un **antet** ce conține adrese și alte informații și **corpul mesajului** cu informațiile de trimis. Mesajele sunt memorate într-o coadă de așteptare și sunt oferite ca intrare programului de transmisie SMTP care e un program server prezent permanent pe host. Orice mesaj din coadă are 2 părți (indiferent de sistemul de operare al hostului):

1. –**textul mesajului**, format din

-**antetul RFC 822**, care **constituie anvelopa** și include indicații despre destinatar/destinatari,

-**corpul mesajului** sau conținutul, compus de utilizator.

2. o **listă a destinațiilor email-ului**, dedusă de agentul utilizator din antetul 822 al mesajului. Uneori destinația/destinațiile sunt specificate literal în antet, altele agentul utilizator trebuie să extindă numele din lista de e-mailuri, să îndepărteze duplicatele, să înlocuiască numele mnemonice cu nume din lista de e-mailuri. Dacă se cere o copie „oarbă” BCC (blind carbon copy) agentul utilizator trebuie să pregătească mesajul în acest sens. Ideea de bază este că diferitele stiluri și formate, preferate de utilizatorul uman în interfața utilizator, sunt înlocuite cu forma standard adecvată pentru transmiterea SMTP.

Transmițătorul SMTP ia mesajul din coada de mail de ieșire și-l transmite hostului destinație printr-o tranzacție SMTP, pe una sau mai multe conexiuni, la portul 25 a hostului destinație. Un host poate avea transmițătoare SMTP multiple active simultan, dacă are un volum mare de mailuri de transmis și de asemenea poate crea la comandă receptoare SMTP multiple, astfel încât mailul de la un host să nu fie întârziat de mailuri de la alte hosturi.

La livrarea unui mesaj, transmițătorul SMTP șterge respectiva destinație din lista destinațiilor, iar când l-a livrat tuturor destinatarilor, șterge mesajul coadă. Transmițătorul SMTP poate optimiza lucrul cu coada. Astfel dacă un mesaj e trimis la mai mulți utilizatori de pe același host, textul mesajului e transmis o singură dată. Dacă mai multe mesaje

Modulul 8

Aplicații

sunt gata de transmis aceluiși host, transmițătorul SMTP deschide o singură conexiune TCP, transferă toate mesajele și închide conexiunea, în loc să deschidă câte o conexiune pentru fiecare mesaj.

Exemplu

Transmițătorul SMTP trebuie să rezolve o serie de **erori**: **hostul destinație nu poate fi găsit**, poate fi **inactiv** sau **conexiunea TCP** poate să **defecteze** în timpul transferării mailului. Transmițătorul poate reîncărca mailul în coadă pentru livrarea ulterioară, dar după un timp abandonează, nu-l ține la infinit acolo. Altă situație frecventă este **adresa destinației greșită**, fie scrisă greșit de utilizator fie din cauza unei adrese noi a destinatarului pe un host diferit. Dacă poate, transmițătorul SMTP redirecționează mesajul și dacă nu, returnează sursei un mesaj de eroare.

RCF821: protocolul SMTP este folosit pentru a transfera mesajul de la transmițătorul SMTP la receptorul SMTP printr-o conexiune TCP. SMTP încearcă să opereze singur, dar nu garantează recuperarea mesajelor pierdute. SMTP nu returnează o confirmare cap-la-cap sursei mesajului, de livrare cu succes. Nu se garantează nici returnarea mesajelor de eroare. Cu toate acestea sistemele de mail folosind SMTP sunt considerate sigure în general.

Receptorul SMTP acceptă fiecare mesaj care sosește și fie îl plasează în cutia poștală a utilizatorului, fie îl copiază într-o coadă locală de ieșire dacă se cere expedierea. Receptorul SMTP trebuie să fie capabil să verifice destinațiile locale ale mailului și să trateze erorile, inclusiv erorile de transmisie și memoria insuficientă.

Transmițătorul SMTP este responsabil cu extragerea mesajului când receptorul SMTP indică sfârșitul transferului, deși acest lucru înseamnă doar că mesajul a ajuns la receptorul SMTP și nu că mesajul a fost livrat destinatarului final. Referitor la erori, receptorul SMTP se limitează să renunțe la conexiunile TCP care eșuează sau sunt inactive prea mult timp, astfel că transmițătorul are majoritatea responsabilității. Erorile care apar în timpul încheierii indicației de eroare pot duce la duplicarea mesajelor, dar nu la pierderea lor.

De obicei, mesajele circulă pe o singură conexiune TCP de la sursă la destinație. Uneori, mailul poate trece prin mai multe mașini intermediare, folosind posibilitatea de „forwarding SMTP” (expediere), caz în care mesajul traversează mai multe conexiuni TCP. O posibilitate de a face acest lucru este ca transmițătorul să specifice ruta spre destinație ca o secvență de servere. Mai des este cazul când forwarding-ul este necesar din cauza mutării utilizatorului.

Operare SMTP constă dintr-o serie de comenzi și răspunsuri schimbate între transmițător și receptorul SMTP. Inițiativa aparține transmițătorului SMTP care stabilește conexiunea TCP. Odată ce

Modulul 8 Aplicații

conexiunea a fost stabilită, transmițătorul SMTP transmite comenzi, prin conexiune, receptorului SMTP. Fiecare comandă generează o replică de la receptorul MTP.

Fiecare comandă constă dintr-o linie de text, începând cu codul comenzii din 4 litere și urmat eventual de argumente. Replicile sunt de obicei dintr-o singură linie, dar există și replici multilinii.

In comenzi se folosesc următoarele notații:

<CRLF> Carriage Return, Line Feed, <SP> Space,
[...] parantezele drepte indică elemente opționale.

Comenzi SMTP

HELLO <SP> <domain><CRLF>	identificare
MAIL <SP>FROM:<reverse-path><CRLF>	identifică sursa emailului
RCPT <SP>TO:<forward-path><CRLF> emailului	identifică recipientul emailului
DATA <CRLF>	transferă textul mesajului
RSET <CRLF>	renunță la tranzacția curentă de email (abort)
NOOP <CRLF>	no operation
QUIT <CRLF>	închide conexiunea TCP
SEND <SP>FROM:<reverse-path><CRLF>	trimite emailul terminalului
SOML <SP>FROM:<reverse-path><CRLF>	trimite emailul terminalului, dacă e posibil, altfel spre mailbox
SAML <SP>FROM:<reverse-path><CRLF>	trimite la terminal și la mailbox
VRFY <SP> <string><CRLF>	confirmă numele utilizatorului
EXPN <SP> <string><CRLF>	returnează apartenența la lista de email
HELP [<SP> <string>]<CRLF>	trimite documentație specifică sistemului
TURN <CRLF>	inversează rolul transmițătorului și receptorului

Replici SMTP

Confirmarea pozitivă a terminării operației: acțiunea cerută s-a terminat cu succes și poate fi inițiată o nouă cerere.

211- starea/servirea sistemului (system help), ajutor

214- mesaj de ajutor (help): informație despre modul de folosire a receptorului sau sensul unor anumite comenzi nestandard. Replica folosește doar utilizatorului uman

220 <domain> serviciu pregătit

221 <domain> serviciu închizând canalul de transmisiune

250 acțiunea cerută s-a încheiat cu succes

251 utilizatorul nu e local; se va expedia mai departe pe <forward-path>

Replica pozitivă intermediară: comanda a fost acceptată, dar acțiunea cerută este suspendată temporar, în așteptarea unor informații suplimentare, pe care transmițătorul mai trebuie să le furnizeze. Se folosește pentru grupuri de secvențe de comenzi.

354 începe emailul, se termină cu <CRLF>.<CRLF>

Confirmare negativă tranzitorie a încheierii acțiunii: comanda nu a fost acceptată și acțiunea cerută nu apare.

421 <domain> serviciu nedisponibil, canal cu pierderi de date: poate fi răspunsul la orice comandă, când serviciul știe că trebuie să se dezactiveze

450 acțiunea cerută de mail nu va fi îndeplinită; mailbox nedisponibil/ocupat

451 acțiunea cerută este respinsă; există erori în procesarea locală

452 acțiunea cerută nu este preluată; spațiu de memorie insuficient

Confirmarea negativă permanentă a încheierii acțiunii: comanda nu a fost acceptată și acțiunea cerută nu va avea loc

500 eroare de sintaxa, comanda nerecunoscută: sunt incluse și erori ca linie de comanda prea lungă

501 eroare de sintaxa, în parametri sau argument

502 comanda neimplementată

Modulul 8 Aplicații

503 secvență greșită de comenzi

504 parametri de comanda neimplementați

550 acțiunea cerută nu e preluată; mailbox nedisponibil (nu e găsit, nu e acces, etc.)

551 utilizatorul nu e local; încercați <forward-path>

552 respingerea acțiunii cerută de mail; memoria e depășită

553 acțiunea cerută nu e preluată; nume de mailbox nepermis (sintaxa incorectă)

554 tranzacție eșuată

Operațiile de bază SMTP se desfășoară în **trei faze**:

- stabilirea conexiunii,
- schimbul de perechi comenzi-răspunsuri (una sau mai multe)
- terminarea conexiunii.

Exemplu de operații SMTP

1- Stabilirea conexiunii: transmițătorul SMTP va încerca să stabilească o conexiune TCP cu hostul îndepărtat, când există unul sau mai multe mailuri destinate acestuia. Secvența este:

- 1- transmițătorul deschide o conexiune TCP cu receptorul
- 2- după stabilirea conexiunii, receptorul se identifică cu „220, Service Ready”
- 3- transmițătorul se identifică cu o comanda HELLO
- 4- receptorul acceptă cu „250, OK” identificarea transmițătorului

Dacă la destinație nu este disponibil serviciul de email, hostul îndepărtat returnează „421, Service Not Available” în pasul 2 și procesul se termină.

2-Transferul e-mail-ului: după stabilirea conexiunii, transmițătorul poate transmite una sau mai multe mesaje receptorului SMTP. Există trei faze logice la transferul mesajului:

- 1- comanda MAIL identifică sursa adică inițiatorul mesajului

Modulul 8

Aplicații

2- una sau mai multe comenzi RCPT identifică mesajul

3- comanda DATA transferă textul mesajului

Comanda MAIL arată calea de retur care poate fi folosită pentru raportarea erorilor. Dacă receptorul este pregătit să accepte mesajele de la sursă, returnează replica „250 OK”. În caz contrar, răspunsul indică eșecul executării comenzii (codurile 451,452,453) sau o eroare în comandă (codurile 421,500, 501).

Comanda RCPT indică recipientul datelor din mail. Pentru fiecare comandă RCPT se returnează un răspuns separat:

- 1- răspunsul 250: receptorul acceptă destinația, adică cutiile poștale (mail-box) vizate se află la receptor.
- 2- răspunsul 251: destinația va expedia mai departe mail-ul
- 3- răspunsul 551: destinația ar trebui transferată dar receptorul nu redirectează mail-ul; expeditorul trebuie să trimită din nou adresa de expediere (forwarding)
- 4- răspunsul 550: nu există pe host cutie poștală pentru recipient
- 5- răspunsurile 450, 451, 452, 552, 553: destinația este respinsă din alte motive decât cele precedente, sau 421,500,501,503 sunt erori în comandă.

Faza RCPT separată are avantajul că transmițătorul nu va trimite mesajul până nu e sigur că receptorul este pregătit pentru recepția mesajului, pentru minim un recipient, evitându-se astfel supraîncărcarea de a transmite întregul mesaj, doar ca să constate că destinația este necunoscută.

Odată ce receptorul SMTP a fost de acord să primească mesaje pentru cel puțin un recipient, transmițătorul SMTP inițiază transferul mesajului cu comanda DATA. Dacă receptorul SMTP se mai pregătește încă de recepție, returnează 354, altfel returnează o replică indicând eșecul efectuării comenzii (451, 554), sau eroare de comandă (421, 500, 501, 503).

Dacă primește 354, transmițătorul SMTP începe transmisia mesajului pe conexiunea TCP, ca o secvență de linii ASCII. Sfârșitul mesajului este o linie ce conține doar un punct. Receptorul SMTP răspunde cu 250 OK dacă mesajul este acceptat, sau cu un cod de eroare adecvat (451, 452, 552,554).

Modulul 8 Aplicații

Un **exemplu, din RFC 821**, ilustrează procesul:

```
S: MAIL FROM : <Smith @Alpha.ARPA>
R: 250 OK
S: RCPT TO : <Jones @Beta. ARPA>
R: 250 OK
S: RCPT TO : <Green @Beta.ARPA>
R: 550 No such user here
S: RCPT TO : < Brown @Beta. ARPA>
R: 250 OK
S: DATA
R: 354 Start mail input; end with < CRLF>.<CRLF>
S: Blach, blah, blah, ...
S: ....etc, etc, etc
S: <CRLF>.<CRLF>
R: 250 OK
```

Transmițătorul SMTP transmite de la utilizatorul : <Smith @Alpha.ARPA> un mail spre 3 destinatari de pe mașina Beta. ARPA, Jones, Green, și Brown. Receptorul anunță că are cutii poștale doar pentru Jones și Brown, dar că nu știe nimic despre Green. Fiind confirmat cel puțin un recipient, transmițătorul expediază mesajul text.

3- Închiderea conexiunii se face în doi pași de transmițătorul SMTP:

1. Se transmite o comandă QUIT și se așteaptă răspunsul;
2. Închiderea conexiunii TCP: R inițiază propria închidere de conexiune TCP după transmiterea lui QUIT.

RFC 822 definește formatul mesajelor text trimise prin e-mail. Mesajele conțin:

- o **anvelopă**, cu informațiile necesare transiterii și livrării;
- **conținut. Doar la acesta se referă RFC 822.**

Liniile de antet conțin: **cuvânt cheie, coloană, argumentul cuvântului cheie**. Liniile lungi sunt transformate în mai multe linii.

Cuvintele cheie uzuale sunt: **From, To, Subject, Date** (iar **Message ID conține un identificador unic al mesajului**)

Exemplu de conținut de email conform **RFC 822**

```
From : „John Willey” <jw @host.com>
Subject :The Syntax in RFC 822
To : Smith @ other-host.com
CC : Jones @ Yet-Another Host.com
BCC : Green @ Thirt-Host.com
```


Modulul 8 Aplicații

Hello. This section begins the actual message body, which is delimited from the message heading by a blank line.

Deși protocolul SMTP este bine definit de RFC 821, pot apărea **probleme:**

-lungimea mesajelor: la implementările vechi e de 64kB;

-expirarea timpului (time-out): dacă diferă la client și server, s-ar putea ca unul să renunțe, întrerupând neașteptat conexiunea, în timp ce celălalt este încă ocupat;

- schimburile infinite de mesaje, dacă mașina 1 păstrează lista de poștă A, mașina 2 păstrează lista de poștă B, fiecare listă conține o intrare pentru cealaltă, atunci orice mesaj trimis oricărora dintre liste generează un trafic de email nesfârșit.

Unele din aceste probleme au fost rezolvate de **protocolul SMTP extins ESMTP**. Clienții care doresc să-l utilizeze trimit inițial mesajul **EHLO** (nu HELO). Dacă este rejectat, serverul e standard, cu SMTP și clientul procedează normal. Dacă este acceptat, sunt permise comenzi și parametri noi.

Porțile de e-mail

Folosind SMTP, poșta electronică funcționează cel mai bine dacă ambele părți sunt în Internet și pot stabili conexiuni TCP. Există însă cazuri când mașinile nu sunt conectate la Internet (în companii, pentru securitate), transmițătorul înțelege doar RFC 822 (SMTP) și receptorul alte protocole de e-mail (X.400 sau standard de proprietar). Aceste probleme se rezolvă prin porți de e-mail la nivelul aplicației.

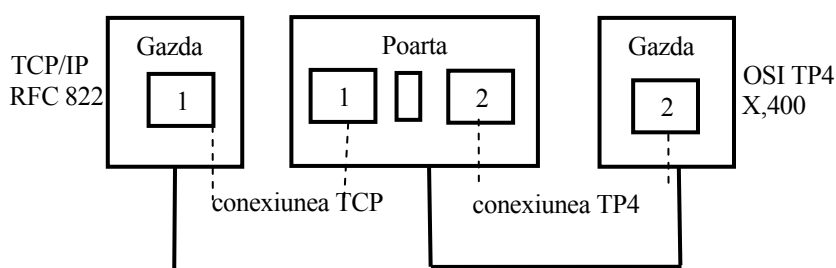


Fig.8.6. Porți de e-mail la nivelul aplicației.

Poarta trebuie să extragă mesajele sosite dintr-o coadă și să le transfere în cealaltă coadă. Dar problema nu este simplă, deoarece:

- adresele Internet și cele X.400 diferă complet, trebuie puse în corespondență. - anvelopa/antetul au câmpuri diferite.

Modulul 8

Aplicații

- dacă un sistem, are clase de prioritate și celălalt nu, se vor pierde informații valoroase într-un sens și în sensul celălalt trebui generate din nimic;

- rezolvarea altor probleme ca: un mesaj se referă la un fișier audio ce trebuie obținut prin FTP, și celălalt sistem nu acceptă acest concept; un sistem X.400 care în caz de eșec mesajul să fie trimis prin fax, dar folosirea faxului nu face parte din RFC822.

Soluțiile sunt mai simple doar pentru mesaje simple (text ASCII), dar în rest sunt foarte complicate adică sunt necesare convertoare de protocol.

Livrarea finală

1- POP3 (Post Office Protocol) RFC1225. Mulți utilizatori lucrează pe PC-uri neconectate la Internet, dar compania unde se află, are servere de e-mail, cu care acești utilizatori comunică pentru a trimite/primi emailuri, folosind un protocol simplu POP3, care are comenzi de conectare/deconectarea utilizatorilor, aducerea/ștergerea mesajelor.

2- IMAP (Interactive Mail System Protocol) RFC 1064. A fost proiectat pentru utilizatorii care folosesc mai multe calculatoare (la serviciu, acasă, portabil). Serverele trebuie să păstreze un depozit central de mesaje și accesul la el să fie posibil de pe orice mașină. Spre deosebire de POP3, IMAP nu copiază emailul pe mașina utilizatorului, fiindcă acesta poate avea mai multe. IMAP are multe facilități, ca de exemplu nu adresează un mesaj prin numărul de sosire ci prin atribute (dă-mi primul mesaj de la Sam). Cutia poștală e un sistem de baze de date relaționale și nu o secvență liniară de mesaje.

3-DMSP (Distributed Mail System Protocol) RFC 1056, este o parte a sistemului PCMAIL. Nu se presupune că tot emailul se află la nivelul unui singur server (ca la POP și IMAP). Permite utilizatorilor să descarce e-mailul pe calculatorul propriu și să se deconecteze apoi. Utilizatorul poate citi/scrie mesajele cât e deconectat și când se reconectează transmite e-mailurile și sistemul e resincronizat.

Facilitățile pentru procesarea suplimentară a mesajelor permite:

- construirea filtrelor; filtrele conțin un set de reguli ce specifică o condiție și o acțiune, ce se verifică la pornirea utilizatorului sau la primirea mesajelor.
- redirecționarea (temporară) a e-mailurilor: un alt calculator sau un serviciu de comunicații (care va contacta utilizatorul prin radio/satelit afișând un mesaj);

Modulul 8

Aplicații

- demonul de vacanță, care trimite un mesaj automat, în absența utilizatorului;
- robotul de e-mail: verifică fiecare mesaj sosit, să vadă dacă este de la un corespondent nou și dacă da, returnează un răspuns standard: nu mai pot citi personal toate e-mailurile, dar am produs un document care răspunde la majoritatea întrebărilor frecvente (FAQ – Frequently Asked Questions). De obicei grupurile de știri au documente FAQ, rareori persoanele.

8.1.3.b. Protocolul MIME

La început ARPANET, poșta electronică era formată din mesaje text, în engleză, în cod ASCII. RFC 822 specifica antetele, dar conținutul era lăsat în seama utilizatorului. Astăzi, abordarea nu mai este actuală. Este necesară transmisia și recepția de mesaje: în limbi cu accent (franceza, germana), cu alfabet neelatin (slav, ebraic), fără alfabet (chineză, japoneză), mesaje fără text (audio, video).

MIME (Multiple Internet Mail Extensios RFC 1341,1521) continuă să folosească formatul RFC 822, dar **adaugă structură corpului mesajului** și definește **reguli de codare pentru mesaje non-ASCII**. Mesajele MIME pot fi trimise cu programele de transmitere și recepție:

Exemple

Antet RFC 822, legat de transport de mesaje.

To:	Adresa/ele de e-mail a/le destinatarului/ilor primari
Cc:	Adresa/ele de e-mail a/le destinatarului/ilor secundar/i
Bcc:	Adresa/ele de e-mail pentru „blind carbon copy”
From:	Persoana/ele care au creat mesajul
Sender:	Adresa/ele de e-mail a transmițătorului curent
Received:	Linie adăugată de fiecare agent de transfer de pe traseu
Return-Path:	Pentru identificarea căii de retur la transmițător

Câmpuri suplimentare de antet RFC 822 folosite de utilizatorii sau receptorii umani.

Date:	Data,ora,minut,secundă la care a fost trimis mesajul
Reply-To:	Adresa de e-mail la care ar trebui trimise

Modulul 8

Aplicații

	răspunsurile
Message-Id:	Numărul unic/identificator folosit ca referință pentru acest mesaj
In-Reply-To:	Identificator mesajului al cărui răspuns este mesajul curent
References:	Alți identificatori de mesaje relevanți
Keywords:	Cuvinte cheie alese de utilizator
Subject:	Rezumatul mesajului, pe o singură linie

Antete RFC 822 adăugate de MIME

MIME-Version:	Identifică versiunea
Content-Description:	Rezumatul mesajului
Content-ID:	Identificator unic
Content-Transfer-Encoding:	Cum e împachetat corpul mesajului pt.transmisie
Content-Type:	Natura mesajului

Mime-Version: mesaj MIME și versiunea (1.0 deocamdată). Mesajele care nu conțin acest câmp se presupun în limba engleză și sunt procesate ca atare.

Content-description: caractere ASCII ce specifică ce este în mesaj. E necesar receptorului să știe dacă merită să decodifice/citească mesajul (de exemplu, dacă acele caractere spun că „mesajul conține fotografia poneiului lui Carmen” și receptorul nu e interesat de ponei, probabil că va arunca mesajul, decât să-l decodifice într-o fotografie color de înaltă rezoluție).

Content-Id identifică conținutul, are același format cu Message-Id

Content-Transfer-Encoding arată cum este împachetat pentru transmisie corpul mesajului, într-o rețea care poate ridica obiecții la alte caractere decât literele: cifre/semne de punctuație. Sunt furnizate 5 scheme și o evadare spre noi scheme:

1. Codarea ASCII pe 7 biți: caracterele pot fi transferate direct de protocolul de email, dacă linia nu depășește 1000 caractere;
2. Codarea pe 8 biți, încalcă protocolul original de email; trebuie respectată lungimea maximă a liniei. Se folosește în locurile din Internet care au implementate extensii ale protocolului original;

Modulul 8

Aplicații

3. Codarea binară; nu e respectată nici lungimea liniei (de exemplu, programele executabile); nu sunt garanții de livrare corectă;
4. Codarea în baza 64 (ramură ASCII): 24 de biți sunt împărțiți în grupe de 6 biți, iar celor 6 biți li se asociază câte un caracter ASCII. Astfel, pot fi transmise sigur texte binare arbitrare.
5. Codare afișabilă marcată (quoted–printable–encoding): pentru texte în majoritate ASCII, unde caracterele cu cod mai mare ca 127 apar ca egal plus caracterul codat prin 2 cifre hexazecimale.

Datele binare ar trebui codate în forma 4 sau 5, dar există și varianta codării definite de utilizator.

Content–Type, cel mai interesant, specifică natura corpului mesajului (7 tipuri).

Tip	Subtip	
Text	Plain	Text neformatat
	Richtext	Text cu comenzi simple de formatare
Image	Gif	Imagini fixe în format GIF
	JPEG	Imagini fixe în format JPEG
Audio	Basic	Sunet
Video	MPEG	Film în format MPEG
Application	Octet-stream	Secvență neinterpretată de octeți
	Post-script	Un document afișabil în Post Script
Message	RFC 822	Un mesaj MIME RFC 822
	Parțial	Mesajul a fost fragmentat pentru transmisie
	External-body	Mesajul în sine trebuie adus din rețea
Multipart	Mixed	Părți independente în ordine specificată
	Alternative	Același mesaj în formate diferite
	Parallel	Părțile trebuie vizualizate simultan
	Digest	Fiecare parte e un mesaj RFC 822 complet

8.2. Aplicații moderne

Dintre aceste aplicații moderne amintim câteva: accesarea Web cu HTTP, serviciul de ghidare prin internet, DNS, transmiterea vocii

Modulul 8

Aplicații

prin internet și suport multimedia, SIP, etc. Creșterea rapidă a folosirii Web-ului e datorată standardizării tuturor elementelor ce suportă aplicații Web. Elementul cheie este **HTTP** (Hypertext Transfer Protocol), pentru schimbul de informații între browserele Web (programe de navigare) și serverele Web. În rețelele HTTP pot fi folosite trei tipuri de **dispozitive intermediare**: **proxy-uri**, **gateway-uri** și **tunele**.

DNS (Domain Name System) este un serviciu de căutare ierarhizat ce asociază numele host-ului de pe Internet cu adresa sa numerică. DNS folosește o bază de date distribuită, ierarhizată, pentru asocierea nume-adresă și pentru a oferi informații despre hosturi.

SIP (Session Initiation Protocol) este un protocol de control de nivel aplicație, pentru stabilirea, modificarea și terminarea sesiunilor de timp-real între participanți, printr-o rețea de date IP.

SDP (Session Description Protocol) este folosit de SIP pentru a descrie conținutul media ce este folosit pe durata sesiunii.

API (Application Programming Interface) pentru socket-uri oferă o cale standardizată, convenabilă pentru scrierea programelor de aplicații, care folosesc comunicații TCP sau IP.

8.2.1. Accesul la Web – protocolul HTTP

HTTP este protocolul care stă la baza WWW (World Wide Web) și poate fi folosit în orice aplicație client/server ce presupune hipertext (hypertext). De fapt, transmiterea informației se face cu eficiența necesară pentru realizarea salturilor de hipertext. Pentru utilizatorul Web-ului, documentele răspândite în toată lumea constituie niște **pagini**, care pot conține legături spre alte pagini. Selectând legătura (clic) se ajunge la noua pagină, etc. Despre paginile care indică alte pagini, se spune că utilizează **hipertext**. Datele transferate de HTTP pot fi texte, hipertexte, audio, imagini sau alte informații accesibile pe internet. În cele ce urmează sunt prezentați câțiva **termeni cheie** referitori la **http**.

Cache este o memorie intermediară rapidă, de program, în care se memorează mesajele de răspuns și subsistemul care controlează memorarea, regăsirea și ștergerea mesajelor sale. Memorarea în cache se face pentru reducerea timpului de răspuns și a consumului de bandă a rețelei, pentru viitoarele cereri. Orice client sau server poate include cache-ul, deși la tunelare serverul nu o poate folosi.

Client este un program aplicație care stabilește o conexiune cu scopul de a trimite cereri.

Conexiune este un circuit virtual de nivel transport stabilit între două programe aplicație în vederea comunicării.

Modulul 8

Aplicații

Entitate este o reprezentare particulară sau redare a unei resurse de date, sau răspunsul de la o resursă de serviciu, care poate fi inclusă într-un mesaj de cerere sau de răspuns. O entitate constă din antet și corp.

Gateway (poarta) este un server care lucrează ca intermediar pentru un alt server. Spre deosebire de proxy, poarta primește cererile ca și cum era serverul original pentru resursa solicitată; clientul care emite cererea poate să nu fie conștient că el comunică cu un gateway. Poarta e deseori folosită ca un portal pe partea serverului prin firewall-urile rețelei și ca un translator de protocol pentru accesarea resurselor de pe sisteme non-HTTP.

Mesajul este unitatea de bază a comunicației HTTP, constând dintr-o secvență de octeți transmisă prin conexiune.

Serverul origine este serverul pe care resursa există sau va fi creată.

Proxy este program intermediar care acționează atât ca server cât și ca client, cu scopul de a face cereri din partea altor servere, eventual traduse. Proxy-ul trebuie să interpreteze, și dacă e necesar să rescrie, mesajele de cerere înainte de expedierea lor (forwarding). Deseori, proxy-urile sunt folosite ca portaluri pe partea utilizatorului prin firewall-urile rețelei și ca aplicații „helper” pentru manevrarea cererilor prin protocoale neimplementate de agentul utilizator.

Resursă este un obiect de date rețea sau serviciu, care poate fi identificat prin URI (Uniform Resource Identifier).

Server este un program care acceptă conexiuni de cereri de serviciu, trimițând înapoi răspunsuri.

Tunelul este un program intermediar, care acționează ca un releu „orb” între două conexiuni. Odată activat, tunelul nu este considerat ca parte a comunicației HTTP, deși tunelul poate fi inițiat de o cerere HTTP. Tunelul încetează să existe când închid ambele părți cărora le-a facilitat comunicarea. Tunelurile se folosesc când este necesar un portal și sistemele intermediare nu pot, sau nu vor, să interpreteze comunicația facilitată (relayed).

Agentul utilizator este un client care inițiază o conexiune și deseori sunt browser-e, editoare, spider-e sau alte programe din sistemele finale. Spiderul este un căutător de informații prin Internet, care indexează bazele de date și facilitează astfel lucrul motoarelor de căutare.

HTTP este un protocol client-server orientat pe tranzacții. Tipic, este folosit între un browser Web și un server Web. Folosește TCP

Modulul 8 Aplicații

pentru siguranță. Astfel, va fi creată la fiecare tranzacție, o nouă conexiune TCP între client și server, care se închide la sfârșitul tranzacției. Aplicația tipică presupune utilizator-browser Web, de găsim a unor secvențe de pagini și documente, distribuite pe servere foarte dispersate, acțiuni care se desfășoară foarte rapid (ideal).

HTTP este flexibil în ceea ce privește formatele. Cererea unui client la server poate include o listă cu priorități de formate iar serverul răspunde cu formatul potrivit. De exemplu, browser-ul lynx nu poate manevra imagini, deci serverul Web nu trebuie să transmită nici o imagine de pe paginile Web. Se evită astfel transmiterea informațiilor inutile și se permite extinderea setului de formate cu specificații standard, brevetate.

Exemple de operații HTTP

a) agentul utilizator stabilește o conexiune directă cu serverul_origine (un browser Web și un server ce conține paginile dorite).

Clientul deschide o conexiune TCP end-to-end cu serverul, după care generează o cerere HTTP, care constă dintr-o comandă (metodă), o adresă [URL Uniform Resource Locator], și un mesaj de tip MIME ce conține parametrii solicitați, informații despre client și eventual alte informații. Când serverul primește cererea, încearcă să execute operația de stare, codul de succes/eroare și un mesaj de tip MIME cu informații despre server, despre răspunsul propriu zis și eventual un corp. Apoi conexiunea se închide.

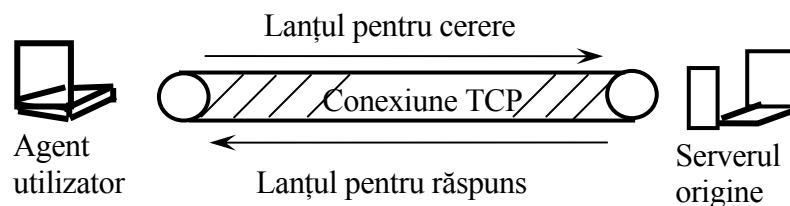


Fig.8.7. Operație HTTP: agentul utilizator stabilește o conexiune directă cu serverul_origine

b) Nu există o conexiune TCP end-to-end între agentul utilizator și serverul origine. Există câteva sisteme intermediare.

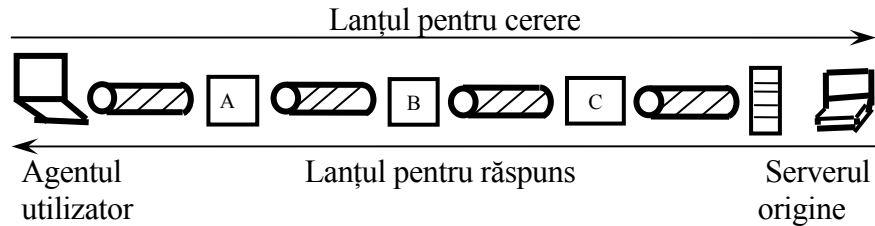


Fig.8.8. Operație HTTP: Nu există o conexiune TCP end-to-end între agentul utilizator și serverul origine. Există câteva sisteme intermediare.

Între sistemele adiacente există conexiuni TCP. Fiecare sistem intermediar acționează ca un releu transferând cererile/răspunsurile.

Există specificate, de către HTTP, 3 tipuri de sisteme intermediare: proxy, gateway și tunel.

1-Proxy-ul prezintă serverului cererile clienților și răspunsurilor din partea serverului la clienți.

Există două scenarii de apel a proxy-ului:

1. Dacă există un intermediar de securizare, care separă clientul și serverul, proxy-ul e plasat între client și firewall. Clientul face parte din rețeaua securizată de firewall, dar serverul nu, deci serverul trebuie să se autentifice la firewall pentru a stabili conexiunea cu proxy. Proxy-ul acceptă răspunsurile de la server, după ce au trecut prin firewall.

2. Dacă clientul și serverul au versiuni HTTP diferite, proxy-ul trebuie să facă adaptarea.

Proxy-ul este deci un agent de înaintare, primind cereri de obiect URL, modificând cererile și înaintând cererile spre serverul identificat de URL.

2-Gateway-ul este folosit de serverele care nu pot comunica cu clientul direct și le apare acestora ca un server origine. Există și aici două scenarii de utilizare a gateway-ului.

1. Dacă există un intermediar de securizare (firewall), cu gateway-ul înspre serverul origine, atunci serverul face parte din rețeaua securizată dar clientul nu, deci trebuie să se autentifice la gateway, care apoi va pasa cererea serverului.

2. Serverul nonHTTP (de exemplu, FTP sau Gopher), pretinde ca browserele Web să înglobeze funcțiile de adaptare cu HTTP sau gateway-uri cu aceste funcții. Clientul face o cerere HTTP spre gateway-ul server, care contactează serverul FTP/Gopher pentru a

Modulul 8

Aplicații

obține rezultatul dorit, pe care apoi îl va converti HTTP și-l va transmite clientului.

3-Tunelul, spre deosebire de proxy și gateway, nu face nici o operație la cereri sau răspunsuri HTTP. El este un releu între două conexiuni TCP și mesajele HTTP trec neschimbate dacă există o singură conexiune HTTP între utilizator și serverul origine. Apare când este necesar un sistem intermediar, dar care să nu înțeleagă conținutul mesajelor, de exemplu un firewall prin care un client/server, extern rețelei protejate, stabilește o conexiune autenticată, pe care o menține apoi pentru tranzacții HTTP.

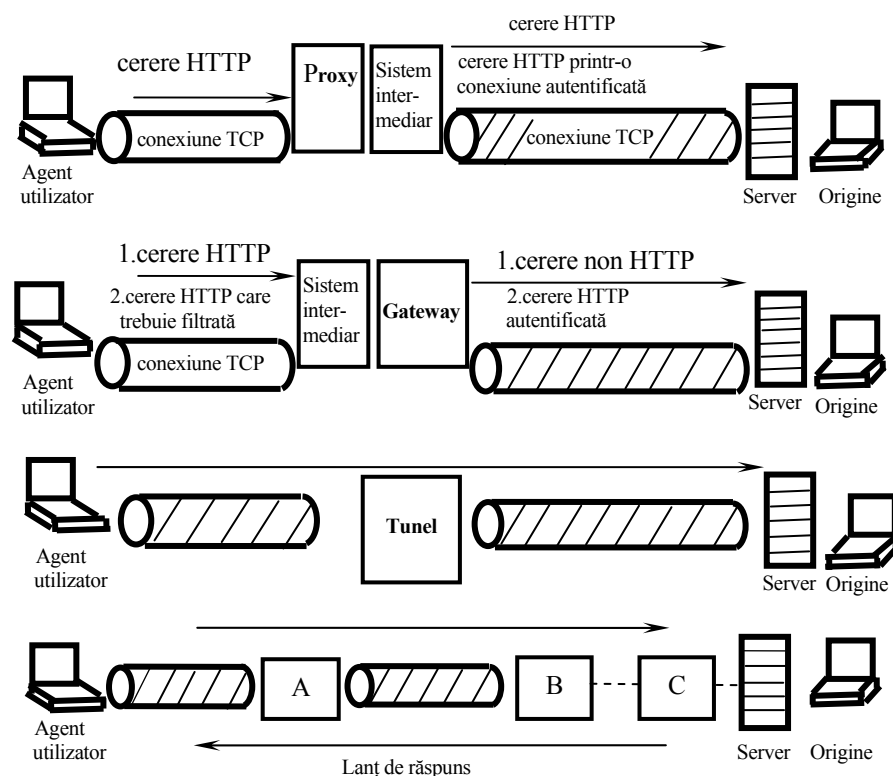


Fig.8. 9. Sisteme HTTP intermediare

c) **exemplu de cache** (facilitate de memorare a cererilor/răspunsurilor anterioare, pentru manevrarea noilor cereri). Dacă noua cerere este identică cu una veche, cache-ul poate oferi răspunsul direct, în loc să acceseze din nou resursa indicată de URL. Cache-ul poate opera la client, la server sau în sistemele intermediare. În figura, sistemul intermediar B memorează în cache tranzacția cerere/răspuns, astfel că cererea nouă e manevrată de B și nu mai trebuie să traverseze rețeaua până la serverul origine. Nu toate tranzacțiile pot fi memorate în cache și clientul sau serverul pot dicta memorarea unor tranzacții pe o durată limitată.

8.2.2. Sistemul numerelor de domenii DNS

DNS (Domain Name System) este un serviciu de asociere între numele de host și adresa IP numerică. Conține 4 elemente:

- **spațiul numelor de domenii:** DNS folosește pentru spațiul numerelor de domenii o structură de arbore, pentru identificarea resurselor din Internet.

- **baza de date DNS:** fiecare nod și frunză al arborelui specifică o serie de informații (adresă IP, tipul resursei), care sunt conținute în înregistrare resursei, RR (Resource Record). Toate înregistrările RR sunt organizate într-o bază de date distribuită.

- **serverele de nume:** sunt programe server ce mențin informațiile despre o porțiune a arborelui numerelor de domenii și RR-urile asociate.

- **rezolverele:** sunt programe care extrag informația de la serverele de nume la cererea clienților. O cerere tipică este pentru adresa IP corespunzătoare unui nume de domeniu.

Numele de domeniu: adresa IP de 32 biți, ce identifică unic dispozitivul atașat la Internet, este interpretată ca: număr de rețea + număr de host.

Dar, în practică, folosirea adreselor de IP prezintă două probleme:

1- ruterele crează o cale prin Internet pe baza numărului rețelei. Dacă fiecare ruter ar menține o tabelă completă cu toate rețelele și căile preferate spre acestea, manevrarea acestei tabele ar fi dificilă și foarte lentă. Astfel că, pentru simplificarea dirijării se grupează rețelele.

2- adresa de 32 de biți, scrisă ca 4 grupe de numere zecimale corespunzătoare celor 4 octeți de adresă, este utilă procesării pe calculator, dar foarte greu de manevrat de utilizatorii umani, care rețin mai degrabă nume decât numere.

Acestor probleme le răspunde conceptul de **domeniu**, care se referă la un grup de hosturi, controlate de o singură entitate administrativă (o companie sau o agenție guvernamentală). Domeniile sunt organizate ierarhic, astfel că fiecare domeniu constă din câteva domenii subordonate. Fiecare domeniu e denumit de calculator în arbore, până la rădăcină:

- eng.sun.com, componentele fiind separate cu punct,
- /com/sun/eng, în UNIX separarea se face cu slash.

Modulul 8 Aplicații

O porțiune a spațiului numelor de domenii din Internet e prezentată în figura următoare:

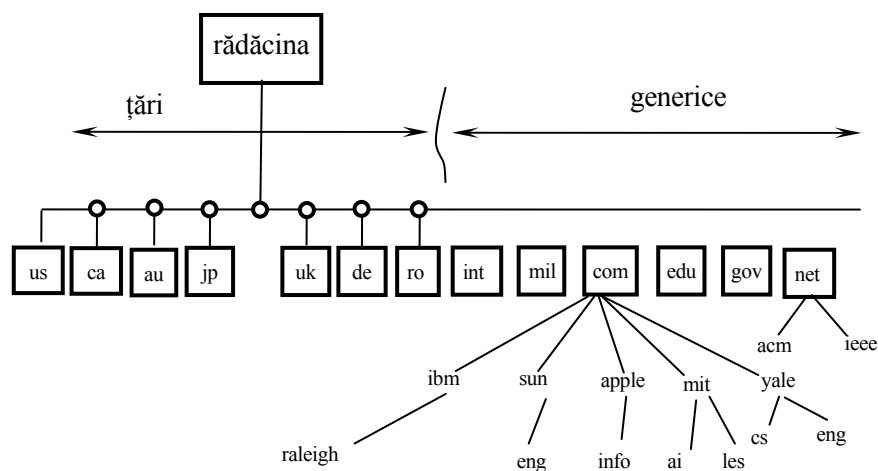


Fig. 8.10. O porțiune a spațiului numelor de domenii din Internet

com-comercial, **edu**-educație, **gov**-guvernul federal al SUA, **int**-organizații internaționale, **mil**-forțe armate SUA, **org**-organizații nonprofit, **network**-centru pentru servicii de rețea, **biz**-afaceri private, **info**-utilizare nerestricționată, **pro**-profesii, medicale, legale, economice, **aero**-comunitate aviatică, **coop**-organizații cooperatiste ca uniunile de credit, **name**-individuale, pentru adrese de email sau domenii personalizate, **museum**-pentru muzee și organizații/persoane lucrând în domeniul muzeelor.

Pentru tot Internetul o organizație alocă numele domeniilor. Fiecare domeniu controlează alocarea domeniilor/subdomeniilor inferioare lui. Pentru înregistrare unui nou domeniu se cere permisiunea domeniului în care va fi inclus. Coborând în jos prin arbore se ajunge la hosturi, care au alocate adrese Internet. Alocarea adreselor se face mai jos în ierarhie.

Baza de date DNS – înregistrarea resurselor

DNS e construit pe o bază de date ierarhică ce conține înregistrările resurselor **RR (resource records)** ce includ numele, adresa IP și alte informații despre hosturi. Formatul RR este:

Modulul 8
Aplicații

Numele domeniului	
Tip	Clasă
Timp de viață	
Lungimea câmpului Rdata	
Rdata	

Fig. 8.11. Înregistrările resurselor RR (resource records)

Pentru un host, înregistrarea normală este chiar adresa sa IP.

Nume domeniu, reprezintă cheia de căutare primară, fiind format dintr-o serie de caractere alfanumerice și cratime (pe care operatorul uman le poate citi) separate de puncte.

Tipul, identifică tipul de resurse din RR:

A– adresă, adresă de host. Se asociază numele cu adresa IP. Ruterele cu adrese multiple au RR separate pentru fiecare adresă.

CNAME – canonical name, specifică numele alias pentru host și-l asociază cu numele canonic (adevărat).

HINFO – host information, specifică procesorul și sistemul de operare al hostului.

MINFO – mailbox or mail-list information, asociază aceste informații la numele hostului.

MX – mail exchange, identifică hostul din organizație, care transferă e-mailurile.

NS – authoritative name server for this domain, numele serverului supus autorității acestui domeniu.

PTR – domain name pointer, indică spre altă parte a spațiului numerelor de domenii.

SOA – start of zone authority, care parte a ierarhiei e implementată .

SRV – indică serverul (ele) care furnizează serviciul specificat.

TXT – arbitrary text, text oarecare, permite adăugarea de comentarii text bazei de date.

Modulul 8

Aplicații

WKS – well known services, listează serviciile aplicație disponibile pe host.

Clasa identifică familia de protocoale = IN pentru Internet sau alte coduri pentru informația noninternet.

Timp de viață, dă indicații despre cât de stabilă e informația (ex: e 86.400 sec pentru o informație stabilită sau 60 sec pentru o informație instabilă). De obicei, când e găsită RR din numeservser, gășitorul va memora în cache RR, pentru a nu interoga serverul de mai multe ori. Câmpul indică timpul de memorare în cahe a RR. Valoarea zero înseamnă că RR nu va fi memorată ci folosită doar pentru tranzacția actuală.

Lungimea câmpului Rdata, ca număr de octeți. Rdata, e un șir de octeți de lungime variabilă , care descrie resursa. Formatul informației e funcție de tipul RR. De exemplu, pentru tipul A , adresa, Rdata e o adresă IP de 32 de biți, iar pentru CNAME e un nume de domeniu.

Exemplu de operare DNS

Operarea DNS conține următoarele faze:

1. programul utilizatorului cere o adresă pentru numele domeniului;
2. modulul resolver din hostul sau ISP-ul local generează o interogare (query) spre serverul de nume local, din același domeniu ca și resolverul;
3. serverul de nume local verifică dacă numele se află în baza sa locală de date sau în cache și dacă e așa, returnează adresaIP la cel ce interoghează. Astfe, interoghează, ierarhic alte servere.(recursiv $A \rightarrow B \rightarrow C \rightarrow B \rightarrow A$ sau iterativ: lui A i se returnează următorul nume de server C,și A va interoga pe C).
4. când sosește răspunsul la serverul de nume local, el memorează în cache asocierea nume / adresă și o ține acolo cât specifică timpul de viață al RR regășite.
5. programului utilizator i se returnează adresa IP sau un mesaj de eroare.

Baza de date distribuită trebuie actualizată des, din cauza schimbărilor frecvente și extinderii internetului.

Modulul 8

Aplicații

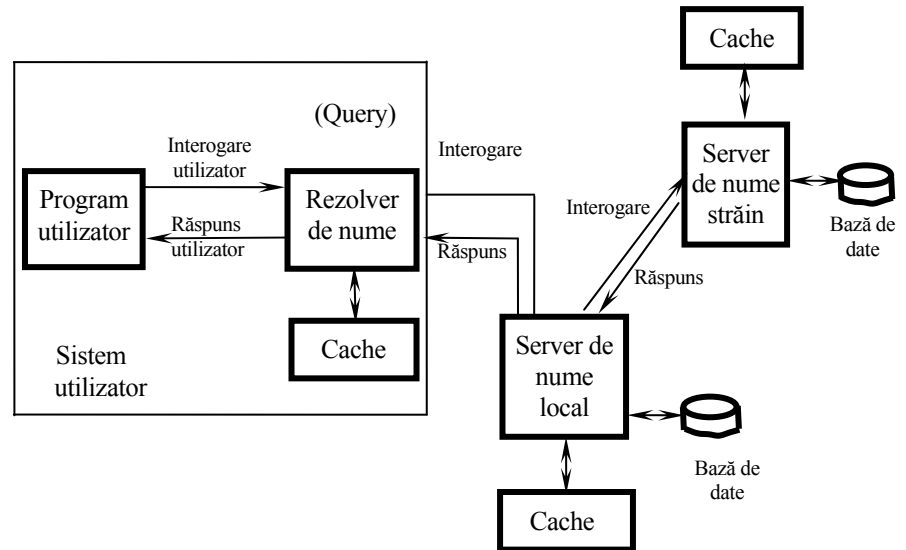
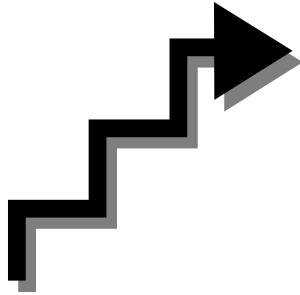


Fig.8.12. Exemplu de operare DNS

Ierarhia serverelor conține în nivelul superior 13 **servere de nume rădăcină** (root name servers), responsabile cu nivelul superior. Sunt necesare mai multe servere pentru ca serverul rădăcină să nu stranguleze traficul (bottleneck).

Baza de date DNS e distribuită ierarhic pe mai multe servere DNS de nume, răspândite în tot Internetul. Serverele de nume sunt operate de organizația care ține domeniul și e responsabilă cu subarborile respectiv al domeniului. Fiecare server de nume e configurat cu un subset al spațiului numerelor de domenii, numit **zonă**. Zona conține unul, mai multe, sau toate subdomeniile din domeniu, împreună cu RR-urile respective. Acest set de date se numește **al autorității** (authoritative), serverul de nume fiind responsabil cu menținerea unui set de RR-uri corecte despre acea porțiune a ierarhiei numelor de domenii. Structura ierarhică se poate extinde în profunzime, în jos, oricât de mult.



Rezumat

Aplicațiile tradiționale ale internetului sunt comunicarea cu ajutorul protocolului TELNET, transferul de fișiere cu ajutorul protocolului FTP și poșta electronică, sau e-mailul. TELNET face posibilă comunicarea unor terminale simple cu un host, local sau la distanță, folosind un terminal de rețea virtual, NVT și un protocol de terminal virtual VTP. Transmiterea comenzilor urgente se face cu TELNET SYNCH. Opțiunile TELNET pot fi obținute prin negociere.

Transferul de fișiere se face sigur și eficient cu protocolul FTP, care protejează utilizatorul de variațiile sistemelor de memorare a fișierelor de pe hosturi diferite. Pentru transfer se stabilesc două conexiuni TCP, una de control și una de transfer. Apoi se negociază opțiunile de transfer: tipul datelor, tipul fișierelor, modul de transfer, flux, bloc sau comprimat, permite eficientizarea lucrului cu rețeaua.

Poșta electronică sau e-mail este cea mai utilizată aplicație din rețelele de comunicații. La început, când se transmiteau mesaje simple, a fost suficient protocolul, dar apoi a fost dezvoltat protocolul MIME pentru a permite transmiterea de voce, imagini, videoclipuri. Dacă transmițătorul înțelege doar SMTP și receptorul alte protocoale de e-mail, se folosesc portile de e-mail. Livrarea finală se face cu protocolul POP3, pentru utilizatorii care lucrează pe PC-uri neconectate la Internet și se conectează ocazional la servere de e-mail, cu IMAP pentru utilizatorii care au mai multe calculatoare și emailurile lor sunt stocate centralizat pe un server, respectiv DMS, care permite nu numai citirea, ca la IMAP, ci și descărcarea emailurilor.

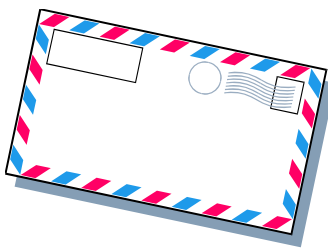
Aplicațiile moderne cele mai cunoscute sunt accesul la web, cu HTTP și sistemul numelor de domenii DNS.

HTTP este un protocol client-server orientat pe tranzacții, care stă la baza WWW (World Wide Web) și poate fi folosit în orice aplicație client/server ce presupune hipertext. Transmiterea informației se face cu eficiența necesară pentru realizarea salturilor de hipertext. Pentru utilizatorii Web-ului, documentele răspândite în toată lumea constituie niște pagini, care pot conține legături spre alte pagini adică utilizează hipertext. Datele transferate de HTTP pot fi texte, hipertexte, audio, imagini sau alte informații accesibile pe internet.

DNS este un serviciu de asociere între numele de host și adresa IP numerică.

ÎNTREBĂRI ȘI PROBLEME

Care sunt aplicații tradiționale ale internetului ?
La ce servește protocolul TELNET?
Ce este terminalul virtual NVT ?
Ce este protocolul de terminal virtual-VTP?
Cum lucrează protocolul de transfer Telnet ?
Ce este mecanismul TELNET SYNCH ?
Care sunt opțiunile TELNET ?
Ce este FTP ?
Cum se face transferul de fișiere cu protocolul FTP?
Care sunt opțiunile de transfer FTP?
Care sunt modurile de transfer FTP?
Care este forma unui e-mail?
Ce face protocolul SMTP?
Ce face protocolul MIME?
Care sunt aplicațiile moderne ale internetului ?
Ce este HTTP?
Ce este DNS și cum operează?
Ce înseamnă cache?
Ce înseamnă client?
Ce înseamnă tunel?
Ce înseamnă proxy?



TEMĂ

1. Dați un exemplu de încercare de stabilire a conexiunii cu protocolul SMTP când la destinație nu este disponibil serverul de e-mail.
2. Dați un exemplu de încercare de stabilire a conexiunii cu protocolul SMTP când receptorul se pregătește încă pentru recepție.
3. Dați un exemplu de e-mail, conform RFC 821 și 822, dacă aveți mai mulți destinatari și doriți să trimiteți copii la alte două persoane, dar o copie să fie invizibilă.
4. Dați un exemplu de domeniu DNS.