

Lucrul în rețea

Utilizarea Sistemelor de Operare

Paul Irofti

Universitatea din București
Facultatea de Matematică și Informatică
Department de Informatică
Email: paul.irofti@fmi.unibuc.ro

Intranet sau rețea

- ▶ alcătuit din noduri (calculatoare) numite *host-uri*
- ▶ legături fizice: cabluri de rețea, canale wireless
- ▶ legături logice: calea prin intermediul legăturilor fizice dintre un nod sursă și unul destinație
- ▶ probleme: parcurgere de graf, calea cea mai scurtă, comis-voiajor
- ▶ o rețea poate fi conectată la una sau mai multe rețele

Internet: set de rețele publice interconectate

Modul de comunicare în rețea se desfășoară urmând unul sau mai multe protocoale

- ▶ descrise în documente *Request for Comments* (RFC)
- ▶ Internet Engineering Task Force (IETF)
- ▶ Internet Society (ISOC)
- ▶ comitete formate din ingineri și informaticieni
- ▶ <https://www.rfc-editor.org/retrieve/>
- ▶ exemple: IP, TCP, DNS, SMTP, IMAP, POP, FTP, SSH
- ▶ majoritatea construite peste TCP și IP
- ▶ **atenție:** protocolul este implementat, nu reprezintă o implementare

Internet Protocol (IP)

Fiecare host este identificat printr-una sau mai multe adrese IP

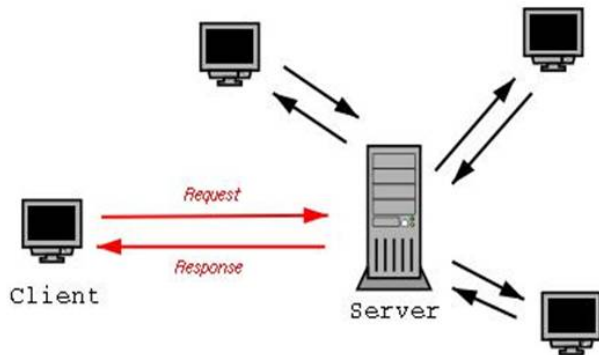
- ▶ versiunea inițială (IPv4) folosește 32-biți (RFC791)
- ▶ versiunea curentă este pe 128-biți (IPv6)
- ▶ IPv4 în continuare cea mai folosită
- ▶ 32 de biți reprezintă 4 bytes care sunt grupați separat prin .
- ▶ interval: 0.0.0.0 – 255.255.255.255
- ▶ anumite subintervale sunt folosite pentru adresare privată
- ▶ restul adreselor publice sunt accesibile de orice nod din rețea
- ▶ **exemplu:** 192.168.1.1, 10.0.0.1, 216.58.214.238, 8.8.8.8

Domain Name System (DNS)

Numele asociat unei adrese IP

- ▶ mai ușor de ținut minte pentru utilizatorii umani
- ▶ protocol descris în RFC1035
- ▶ dacă o adresa IP se schimbă, numele rămâne
- ▶ inițial toată lumea ținea o agendă locală în `/etc/hosts`
- ▶ astăzi procesul e centralizat
- ▶ toată lumea folosește aceeași agendă
- ▶ mai multe *host-uri*, numite *name server*, care conțin o copie a agendei
- ▶ serverele principale de la care copiază informația restul se numesc *root name servers*
- ▶ **exemplu:** `fmi.unibuc.ro` → `193.226.51.6`
- ▶ comenzi în Unix: `nslookup(1)`, `whois(1)`, `dig(1)`

Client-Server



<http://abcnetworking.wikispaces.com>

Comunicarea client-server se face prin *sockets*

- ▶ adresă: adresa IP a unui *host*
- ▶ port: intrare a *host*-ului
- ▶ socket: definește o posibilă conexiune IP:port
- ▶ o conexiune se face cu doi socketi și un protocol comun
- ▶ un socket pentru server și unul pentru client
- ▶ **exemplu:** <TCP,192.168.1.6:4444,193.226.51.6:80>

World Wide Web (`www`)

- ▶ creat la CERN și publicat în 1991
- ▶ resurse identificate prin Uniform Resource Locators (URL)
- ▶ pagini web prezentate prin hypertext
- ▶ HyperText Transfer Protocol (HTTP) RFC2616
- ▶ servesc pagini web pe portul 80
- ▶ acces encriptat prin HTTPS pe port 443 (curând implicit)
- ▶ **exemple:**
 - ▶ `http://fmi.unibuc.ro:80,`
 - ▶ `http://fmi.unibuc.ro`
 - ▶ `http://fmi.unibuc.ro:443`
 - ▶ `https://fmi.unibuc.ro:443`
 - ▶ `https://fmi.unibuc.ro`
- ▶ browser Unix: `lynx(1)`, `w3m(1)`
- ▶ crawler Unix: `curl(1)`, `wget(1)`

Acces la poștă

- ▶ Post Office Protocol (POP) RFC1939
- ▶ POP face o copie locală a mesajelor și le șterge de pe server
- ▶ Internet Message Access Protocol (IMAP) RFC3501
- ▶ IMAP operează asupra mesajelor direct pe server
- ▶ IMAP poate ține o copie (parțială) local
- ▶ porturi: POP 110, POP3S 995, IMAP 143, IMAPS 993
- ▶ mesageria ținută în format mbox sau maildir
- ▶ mbox: ține toate mesajele într-un singur fișier
- ▶ maildir: ține fiecare mesaj într-un fișier separat

Expediere mesaje

- ▶ Simple Message Transfer Protocol (SMTP) RFC821
- ▶ SMTP trimite un mesaj către un destinatar
- ▶ protocol vechi, nu cere exepeditor → spam
- ▶ protocol vechi, nu cere autentificare → spam
- ▶ porturi: SMTP 25, SMTPS
- ▶ soluții anti-spam
 - ▶ reverse DNS: legătură IP → nume
 - ▶ autentificare: utilizator și parolă, criptare
 - ▶ Sender Policy Framework (SPF): anti-email spoofing
 - ▶ baze de date cu IP-uri de spameri

File Transfer Protocol (ftp)

Transfer de fișiere

- ▶ protocol vechi din 1971, activ și astăzi RFC959
- ▶ autentificare și criptare FTPS
- ▶ port separat de comandă (21) și date (20)
- ▶ **exemplu:**
ftp://[user[:password]@]host[:port]/url-path
- ▶ comandă Unix: ftp(1)
- ▶ **exemple:**
 - ▶ \$ ftp alex@fmi.unibuc.ro
 - ▶ \$ ftp -a fmi.unibuc.ro
 - ▶ \$ ftp fmi.unibuc.ro -P 2121

Tranfer de fișiere distribuit fără server

- ▶ comunicație numită peer-to-peer
- ▶ toți clienții transmit și primesc date
- ▶ cei ce doar transmit se numesc *seeds*
- ▶ cei ce primesc se numesc *leechers*
- ▶ mod eficient de transfer pentru date mari
- ▶ elimină presiunea de pe un singur server
- ▶ cel ce inițiază crează un fișier mic `.torrent` în care există informațiile legate de date și locul de unde se pot accesa

Secure Shell (ssh)

Protocol sigur de conectare la alt calculator

- ▶ specificat în RFC4253, port 22
- ▶ implementarea ubicuă OpenSSH creată de grupul OpenBSD
- ▶ <http://www.openssh.com>
- ▶ comandă Unix: `ssh(1)`
- ▶ **exemplu:** `$ ssh [user[:password]@]host[:port]`
- ▶ autentificare parolă, cheie asimetrică etc.
- ▶ o dată autentificat se deschide un terminal pe acel *host* în care puteți începe să dați comenzi
- ▶ se poate folosi interfața grafică cu argumentul `-X`
- ▶ folosit peste tot în programare și administrare

Copiere prin protocolul SSH

- ▶ Secure Copy Protocol (SCP) nu există RFC
- ▶ funcționează similar cu comanda `cp(1)`
- ▶ are în plus prefixat *host-ul*
- ▶ comenzi Unix: `scp(1)`
- ▶ **exemple:**
 - ▶ `$ scp hello.c fmi.unibuc.ro:`
 - ▶ `$ scp hello.c alex@fmi.unibuc.ro:code/`
 - ▶ `$ scp -r project/ alex@fmi.unibuc.ro:`

Funcționalitate FTP prin protocolul SSH

- ▶ SSH File Transfer Protocol (SFTP) nu există RFC
- ▶ funcționează similar cu comanda ftp(1)
- ▶ profită de autentificare și criptografia de desubt (SSH)
- ▶ de obicei tot pe port 22 este servit
- ▶ comenzile protocolului SSH diferențiază între tipuri de sesiuni
- ▶ comenzi Unix: sftp(1)
- ▶ **exemplu:** \$ sftp alex@fmi.unibuc.ro

Virtual Network Computing (vnc)

- ▶ bazat pe protocolul Remote Framebuffer (RFB) RFC6143
- ▶ oferă acces la interfața grafică de la distanță
- ▶ funcționează cu Windows, macOS și X (sistemul grafic Unix)
- ▶ VNC este cea mai cunoscută aplicație ce implementează și folosește RFB
- ▶ clientul se conectează la server pe port 5900
- ▶ comenzi Unix: `vncviewer(1)`, `vncserver(1)`
- ▶ **exemplu:**
 - ▶ `$ vncserver :0` – pornește server pe primul display
 - ▶ `$ vncviewer fmi.unibuc.ro`
 - ▶ `$ vncviewer fmi.unibuc.ro:2` – conectează la display-ul 3
 - ▶ `$ vncserver -kill :0` – oprește server

Remote Desktop (rdp)

- ▶ Remote Desktop Protocol, proprietar Windows
- ▶ funcționează similar cu VNC
- ▶ oferă în plus acces la sistemul audio și imprimantă
- ▶ clientul se conectează la server pe port 3389
- ▶ pe Windows clasic e permis un singur utilizator conectat
- ▶ pe Windows server se pot conecta mai mulți deodată
- ▶ comenzi Unix: `rdesktop(1)`, `xfreerdp(1)`
- ▶ **exemplu:**
 - ▶ `$ rdesktop 192.168.1.6`
 - ▶ `$ rdesktop fmi.unibuc.ro`
 - ▶ `$ rdesktop -r sound:remote -g 90% 141.85.225.153`