



Tehnologii Internet

CURSUL 01 - INTERNETUL

Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Departamentul de Calculatoare
Specializarea Tehnologia informației

© 2017-2018 Adrian ALEXANDRESCU



Cuprins

1. Internetul – definiții, istoric, concepte
2. Rețele de calculatoare
3. Comunicarea client-server



1. Internetul

1.1. Introducere și definiții

1.2. Istoric

1.3. Utilizarea Internetului

1.5. Harta Internetului



1.1. Internetul

- Infrastructură globală
- Mecanism de diseminare a informației
- Mediu în care lumea poate să interacționeze și să colaboreze prin intermediul calculatorului
- Modalitate de comunicare între indivizi indiferent de locația geografică în care se află aceștia
- Magazin virtual de unde se pot cumpăra diverse produse și servicii
- Poșta electronică
- Streaming



1.1. Internetul - definiții

Rețea de calculatoare

- Ansamblu de calculatoare conectate între ele prin intermediul unui mediu de comunicare (cablu, wireless)

Internet

- Sistem global de rețele interconectate de calculatoare care utilizează standardul TCP/IP

Protocol (în OOP)

- Modalitate prin care două sau mai multe obiecte comunică între ele

Protocol de comunicare (în rețele de calculatoare)

- Reguli sau specificații pe baza cărora se realizează transferul de date inter- și intra-calculatoare



1.1. Internetul - definiții

World Wide Web - WWW, W3, Web

- Sistem de documente hipertext interconectate care sunt accesate prin intermediul Internetului

Hipertext (en., hypertext)

- Text afișat pe un dispozitiv electronic care conține referințe către alte date ce pot fi accesate de utilizator

Hiperlink (en., hyperlink)

- Referință la date care pot fi accesate de utilizator (în contextul unui hipertext)



1.2. Internetul - istoric

- 1962 - J.C.R. Licklider de la MIT și DARPA propune realizarea unui set de calculatoare conectate între ele prin intermediul cărora fiecare individ să poată accesa rapid date și programe din orice loc
- 1965 – este creată prima rețea de calculatoare prin conectarea a două calculatoare din localități diferite prin intermediul liniei telefonice (dial-up)



1.2. Internetul - istoric

- 1967 – Lawrence G. Roberts publică planul pentru ARPANET (prima rețea care folosește conceptul de "packet switching")
- Conceptul de *packet switching* presupune împărțirea mesajelor în pachete arbitrare, deciziile de rutare sunt luate la nivel de pachet



1.2. Internetul - istoric

- 1969 – primul mesaj host-to-host este trimis prin intermediul ARPANET din laboratorul lui Leonard Kleinrock de la UCLA la Stanford Research Institute
- Network Measurement Center de la UCLA a fost primul nod ARPANET
- Au fost adăugate alte două noduri, UC Santa Barbara and University of Utah
- 1972 – a fost organizată prima demonstrație publică a ARPANET la International Computer Communication Conference (ICCC)

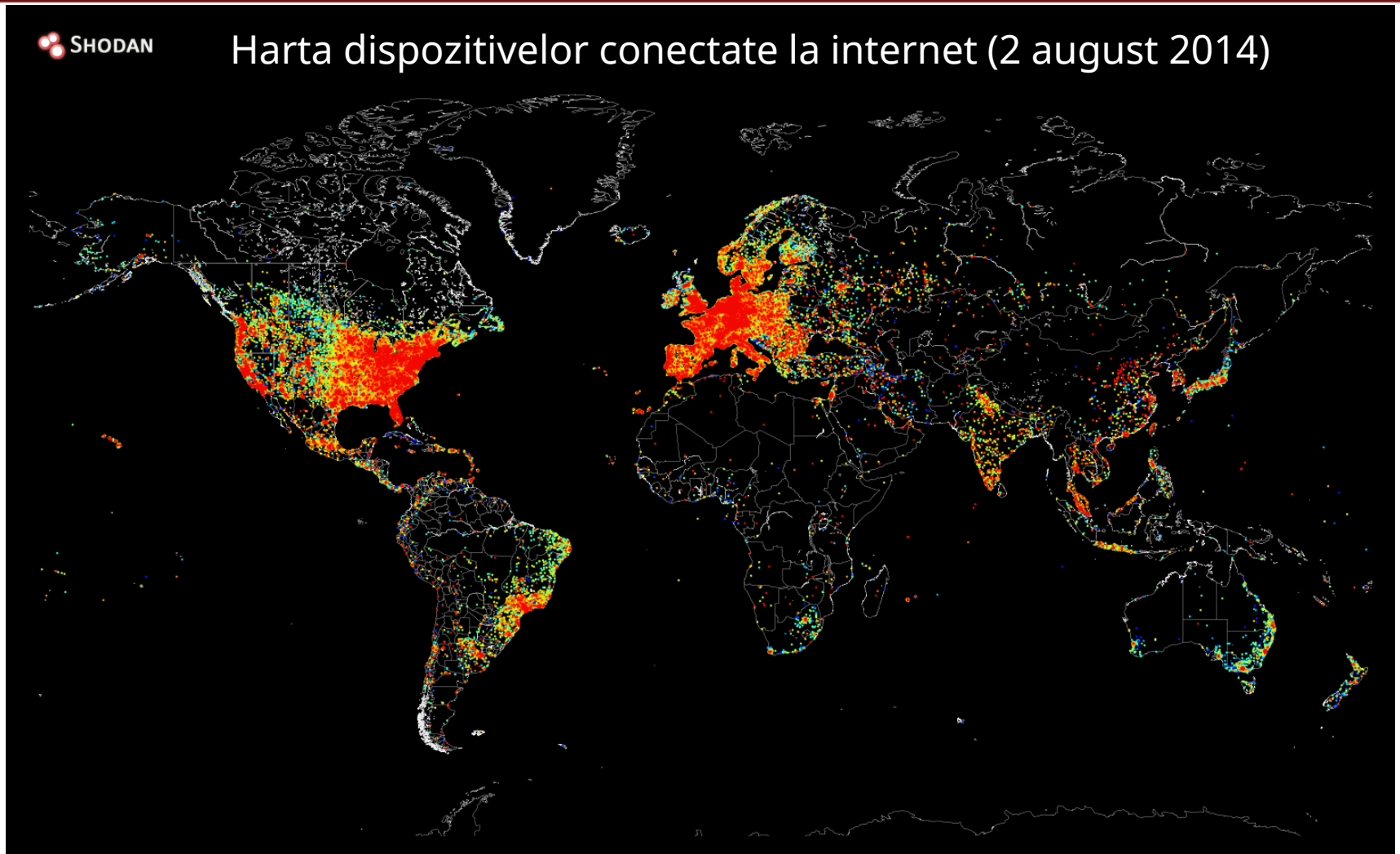


1.2. Internetul - istoric

- 1972 – apare poșta electronică
- 1983 – se realizează trecerea de la protocolul NCP (Network Control Protocol) la TCP/IP (Transmission Control Protocol / Internet Protocol)
- Bob Kahn a avut un rol important în designul ARPANET
- Ideea de bază a Internetului:
 - Mai multe rețele independente de calculatoare cu design arbitrar conectate între ele



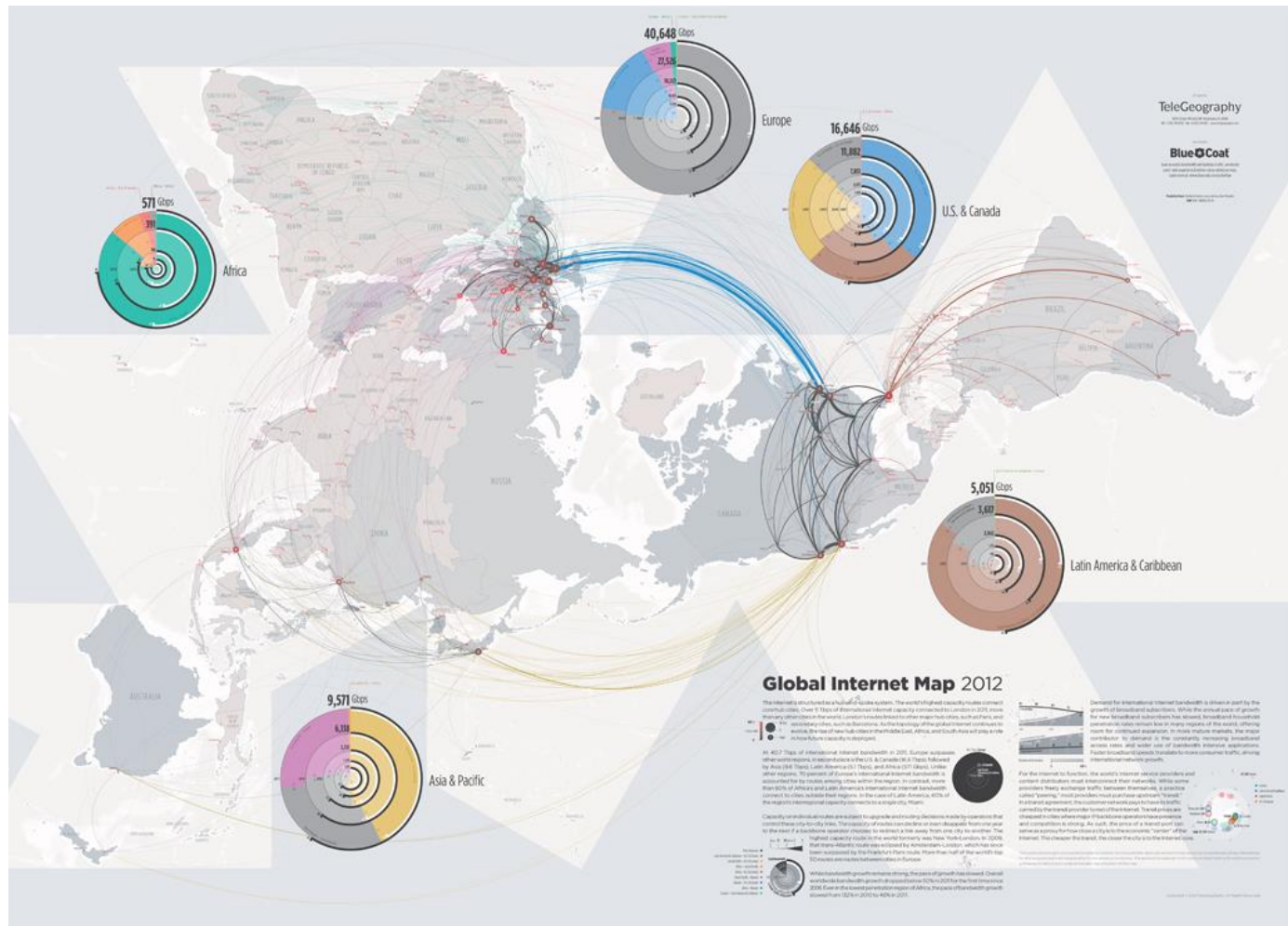
1.3. Utilizarea Internetului



Sursa: <http://www.iflscience.com/technology/map-shows-all-devices-world-connected-internet> 11



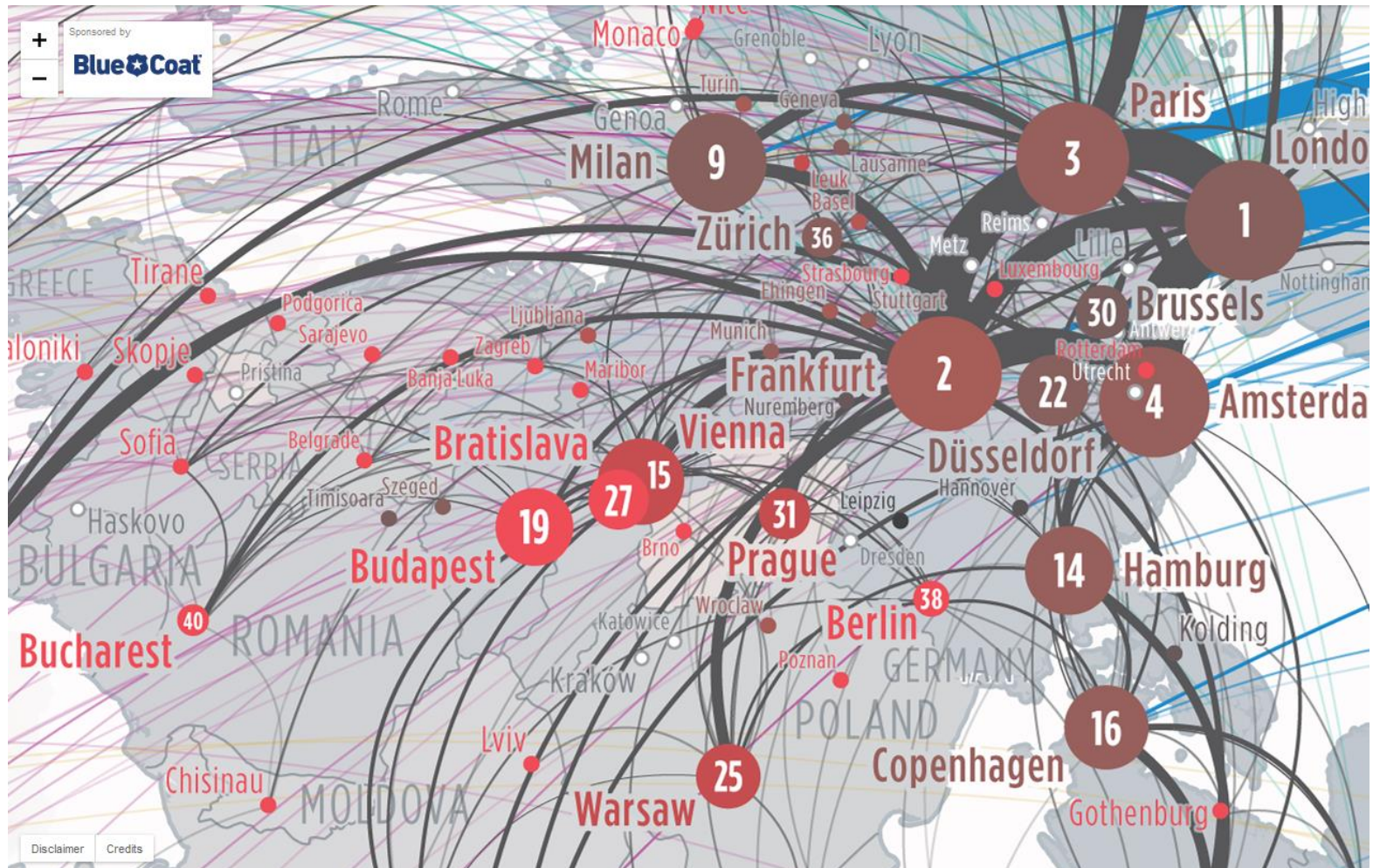
1.4. Harta Internetului



Sursa: <http://global-internet-map-2012.telegeography.com/>



1.4. Harta Internetului



Sursa: <http://global-internet-map-2012.telegeography.com/>



2. Rețelele de calculatoare

2.1. Definiții

2.2. Tipuri de rețele de calculatoare

2.3. Topologii rețea

2.4. Nodurile/componentele unei rețele de calculatoare

2.5. Modelul OSI



2.1. Rețele de calculatoare – definiții

Rețea de calculatoare

- Ansamblu de calculatoare conectate între ele prin intermediul unui mediu de comunicare (cablu, wireless)
- Prin intermediul unui calculator din rețea se pot accesa resursele (hardware, software, fișiere) puse la dispoziție de un alt calculator din aceeași rețea.

Port (hardware - fizic)

- "Priză" specializată pe un dispozitiv la care se poate conecta un cablu
- E.g., port USB, Firewire, Serial, VGA, HDMI



2.2. Tipuri de rețele de calculatoare

După răspândirea geografică

- LAN (Local Area Network)
 - conectarea unor calculatoare și dispozitive dintr-o zonă restrânsă geografic (casă, școală, firmă)
 - rate mari de transfer a datelor
- WAN (Wide Area Network)
 - conectarea dispozitivelor dintr-un oraș, țară sau chiar zone mai mari
 - un LAN poate fi conectat la un WAN printr-un router
- PAN, HAN, SAN, CAN, MAN



2.2. Tipuri de rețele de calculatoare

- Internet backbone
 - WAN-uri și routere care fac legătura dintre toate rețelele conectate la Internet
- VPN (Virtual Private Network)
 - Rețea privată peste o rețea publică (e.g., peste internet)
 - Aplicații client VPN: Hamachi, Tunngle



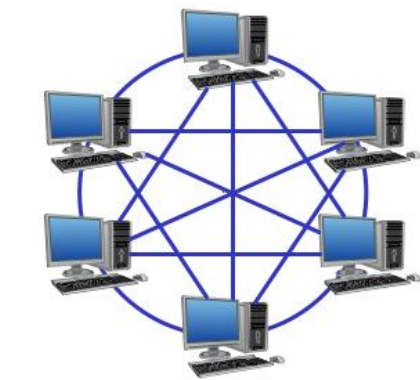
2.3. Topologii rețea

Topologie rețea

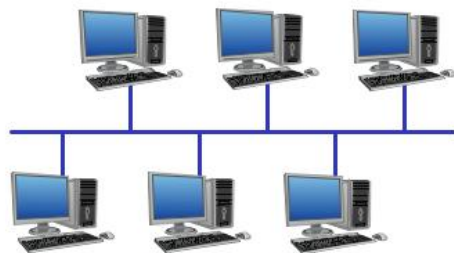
- Modul cum sunt aranjate elementele unei rețele de calculatoare
- Tipuri de topologii: linie, inel, stea, magistrală (bus), mesh, graf complet



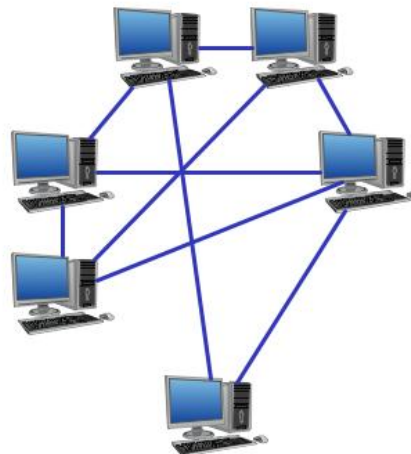
2.3. Topologii rețea



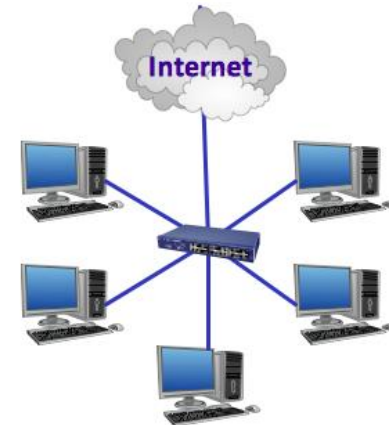
Fully Connected Network Topology



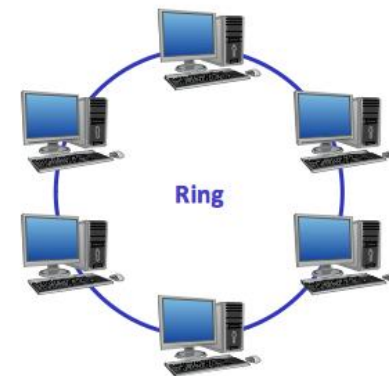
Common Bus Network Topology



Mesh Network Topology



Star Network Topology



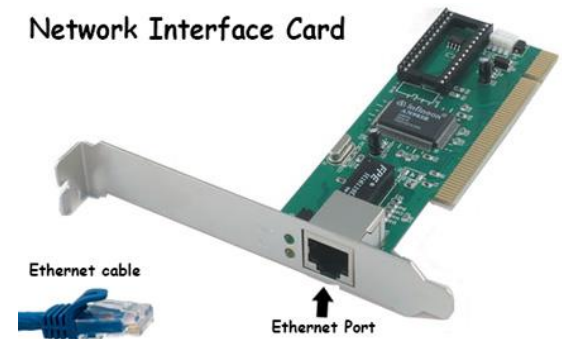
Ring Network Topology

Sursa: <http://www.conceptdraw.com/examples/network-topologies>



2.4. Nodurile unei rețele de calculatoare

- NIC (Network Interface Controller)
 - Permite conectarea unui calculator la rețea
 - Placa de rețea
- Modem (**m**odulator-**d**emodulator)
 - Transformă datele digitale de la un calculator în semnale electrice în vederea transmiterii prin intermediul liniilor telefonice





2.4. Nodurile unei rețele de calculatoare

- Hub
 - Permite conectarea mai multor dispozitive de rețea => dispozitivele se comportă ca un singur segment de rețea
 - Semnalul de la intrarea unui port apare la ieșirea celorlalte porturi





2.4. Nodurile unei rețele de calculatoare

- Switch (Switching hub)
 - Similar cu Hub-ul, dar cu funcții mai avansate
 - Un mesaj este transmis doar unui sau anumitor dispozitive (*multicast*), în comparație cu transmiterea mesajului tuturor dispozitivelor conectate (*broadcast*)





2.4. Nodurile unei rețele de calculatoare

- Router
 - Dispozitiv care transmite pachete între rețele
 - Este conectat la cel puțin două rețele (LAN-LAN, LAN-WAN, WAN-WAN, LAN-ISP network)
 - Permite crearea unei rețele interne



ISP – Internet Service Provider



2.4. Nodurile unei rețele de calculatoare

- Wireless Access Point (AP)
 - Dispozitiv care permite dispozitivelor wireless să se conecteze la o rețea cu fir folosind Wi-Fi
 - AP-ul poate să fie un dispozitiv de sine stătător sau integrat într-un router (wireless router)
 - Hotspot – AP conectat la internet



- Wi-Fi (WiFi) – tehnologie care permite transferul de date sau conectarea la internet a dispozitivelor prin intermediul undelor radio



3. Comunicarea client-server

3.1. Paradigma client-server

3.2. Adresa IP

3.3. Port-ul

3.4. Serverul

3.5. Clientul

3.6. Implementarea unui server în Java

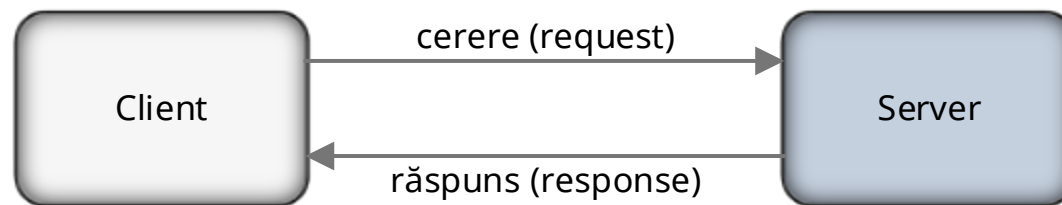
3.7. Implementarea unui client în Java

3.8. Observații



3.1. Paradigma client-server

- **Server** = instanță a unei aplicații care primește cereri și oferă răspunsuri
- **Client** = instanță care accesează serviciile puse la dispoziție de un server
- Serverul așteaptă ("ascultă") conexiuni de la potențiali clienți
- Clientul se conectează la server și trimite mesaje (cereri), iar serverul răspunde





3.1. Paradigma client-server

- Mesajul transmis este o secvență de octeți, dar poate fi interpretat la destinație ca fiind un număr, un șir de caractere sau un obiect mai complex.
- Pentru a crea un server este nevoie de un socket
- **Socket** = instanță la unul din cele două capete ale unei căi de comunicații formată din IP și port
- **IP** = adresa de internet a calculatorului
- **Port** = identificatorul unui capăt al unei căi de comunicații



3.2. Adresa IP

- **Adresa IP** = etichetă numerică asignată unui dispozitiv conectat la internet
- Există două tipuri de adrese IP: IPv4 și IPv6
- **Adresa IPv4** este reprezentată ca un număr pe 32 de biți – 4 octeți – 4 grupuri a 8 biți (e.g., 192.168.1.101)
- **Adrese IPv4 private**

Start	Sfârșit	Nr. de adrese
10.0.0.0	10.255.255.255	16.777.216
172.16.0.0	172.31.255.255	1.048.576
192.168.0.0	192.168.255.255	65.536



3.2. Adresa IP

- **Adresa IPv6** este reprezentată ca un număr pe 128 de biți – 16 octeți – 8 grupuri a 16 biți (e.g., 2001:db8::fe00:42:8313)
- Adresa IPv6 poate fi scurtată astfel:
 - Zerouri de la începutul fiecărui grup pot fi omise
 - Secțiunile consecutive de zerouri pot fi înlocuite cu :: (indiferent de câte secțiuni sunt)
- De exemplu:
 - 2001:0db8:0000:0000:0000:fe00:0042:8313
 - 2001:db8::fe00:42:8313
- Spațiul privat de adrese IPv6 este de forma: fdxx:xxxx:xxxx:....



3.3. Port-ul

Port-ul

- Este folosit pentru a identifica în mod unic o anumită aplicație sau un serviciu care se execută pe un calculator în vederea comunicării printr-o rețea de calculatoare
- Este o valoare numerică pe 16 biți (2 octeți) => valorile posibile sunt în intervalul 1 - 65535



3.3. Port-ul

Conform IANA (Internet Assigned Numbers Authority) porturile se împart în trei categorii:

- Porturi cunoscute (well-known): 0 - 1023
- Porturi înregistrate (registered): 1024 - 49151
- Porturi dinamice sau private: 49152 - 65535



3.3. Port-ul

Exemple de porturi cunoscute

- 20 & 21: File Transfer Protocol (FTP)
- 22: Secure Shell (SSH)
- 23: Telnet
- 25: Simple Mail Transfer Protocol (SMTP)
- 53: Domain Name System (DNS)
- 80: Hypertext Transfer Protocol (HTTP)
- 110: Post Office Protocol (POP3)
- 143: Internet Message Access Protocol (IMAP)
- 443: HTTP Secure (HTTPS)
- 465: SMTP Secure (SMTPS)



3.3. Port-ul

Exemple de porturi înregistrate

- *1194*: OpenVPN
- *1214*: Kazaa
- *1220*: QuickTime Streaming Server
- *1293*: IPSec (Internet Protocol Security)
- *1801*: Microsoft Message Queuing
- *2049*: Network File System
- *3306*: MySQL database system
- *3690*: Subversion (SVN) version control system



3.4. Serverul

Pașii necesari creării unei aplicații server la care să se poată conecta un client:

1. Crearea unui socket pe un anumit port. La socketul respectiv se conectează clienții
2. Apelarea unei metode care să aștepte conectarea unui potențial client (i.e., `accept()`)
3. În momentul în care s-a conectat un client se creează un nou socket care reprezintă capătul dinspre server al căii de comunicație dintre server și clientul conectat
4. Noul socket are un flux de intrare prin care se pot primi mesaje de la client și un flux de ieșire prin care se pot trimite mesaje către client
5. Închiderea conexiunii cu clientul și închiderea serverului



3.5. Clientul

Pașii necesari creării unei aplicații client care să se poată conecta un server:

1. Crearea unui socket cu specificarea adresei IP și a portului serverului
2. Apelarea unei metode care să încerce realizarea conexiunii cu serverul (i.e., `connect()`)
3. Dacă s-a reușit conectarea, socketul creat anterior reprezintă capătul dinspre client al căii de comunicație dintre server și client
4. La fel ca în cazul serverului, pentru a trimite date clientul scrie în fluxul de ieșire (*send*), iar pentru a primi date clientul citește din fluxul de intrare (*receive*)
5. Închiderea conexiunii cu serverul.



3.6. Implementarea unui server în Java

```
ServerSocket ss = new ServerSocket(5678);
Socket s = ss.accept();
PrintWriter socketWriter = new
    PrintWriter(s.getOutputStream(), true);
BufferedReader socketReader = new BufferedReader(
    new InputStreamReader(s.getInputStream()));
BufferedReader consoleReader = new BufferedReader(
    new InputStreamReader(System.in));
socketWriter.println("hello");
String line;
while (!(line = socketReader.readLine()).equals("bye")) {
    System.out.println(line);
    line = consoleReader.readLine();
    socketWriter.println(line);
}
socketWriter.println("bye");
s.close();
ss.close();
```



3.7. Implementarea unui client în Java

```
Socket s = new Socket("localhost", 5678);
```

```
PrintWriter socketWriter = new  
    PrintWriter(s.getOutputStream(), true);
```

```
BufferedReader socketReader = new BufferedReader(  
    new InputStreamReader(s.getInputStream()));
```

```
BufferedReader consoleReader = new BufferedReader(  
    new InputStreamReader(System.in));
```

```
String line;
```

```
while (!(line = socketReader.readLine()).equals("bye")) {  
    System.out.println(line);  
    line = consoleReader.readLine();  
    socketWriter.println(line);
```

```
}
```

```
socketWriter.println("bye");
```

```
s.close();
```



3.8. Observații

- Metoda `accept()` de la server este blocantă, ceea ce înseamnă că execuția programului este blocată până când se conectează un client
- La client, când se creează obiectul de tip `Socket`, automat se apelează metoda `connect()`. Metoda `connect()` este blocată până când se realizează conexiunea sau până când a trecut un anumit interval de timp (*timeout*)



3.8. Observații

- Pentru a se conecta la server, clientul are nevoie de adresa serverului și de portul folosit de server.
- Adresa serverului (*host*) este un șir de caractere care reprezintă adresa IP sau poate fi un nume de domeniu (*domain name*)
- Dacă aplicația client este lansată pe același calculator ca și serverul atunci la adresa serverului se poate folosi IP-ul **127.0.0.1** sau aliasul acestuia, **localhost**



3.8. Observații

- Pentru fiecare *send* de la server trebuie să fie un apel de *receive* la client și, invers, pentru fiecare *send* de la client trebuie să fie un apel de *receive* la server
- De ce?

Server	Client
send	send
receive	receive
send	send
receive	receive

Server	Client
send	receive
receive	send
send	receive
receive	send



3.8. Observații

- Pentru ca atât serverul cât și clientul să poată să trimită și să primească mesaje în același timp trebuie folosite fire de execuție (*thread-uri*)
- Un fir de execuție se ocupă cu trimiterea mesajelor și unul cu primirea lor.
- *Cum comunică cele două fire de execuție între ele?*
- *Cum se poate face ca un server să permită conectarea mai multor clienți?*
- *Cum se poate ca serverul să comunice în același timp cu clienții conectați?*



Bibliografie

- <http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet>
- Douglas Comer. 1988. *Internetworking with Tcp/Ip: Principles, Protocols, and Architecture*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Frenzel, Louis E., *Principles of Electronic Communication Systems*, 3rd edition, McGraw Hill, 2008.
- Simoneau, Paul, *The TCP/IP and OSI Models*, Global Knowledge Training LLC, 2011.
- <http://docs.oracle.com/javase/7/docs/api/java/net/Socket.html>
- <http://docs.oracle.com/javase/tutorial/networking/sockets/>