

# C14

## Java si comunicatia in retea

- Retele de calculatoare:
  - concepte
  - clasificari;
  - dispozitive;
- Protocoale de retea:
  - TCP/IP
- Pachete Java si comunicatia in retea
- Comunicati intre Client si Server:
  - socket
  - stabilirea comunicatiei, prelucrai, terminare.

## Obiective

După parcurgerea acestui curs ar trebuie sa puteți:

- înțelege conceptul de rețea de calculatoare împreună cu modalitățile de clasificare a acestora;
- înțelege conceptul general de protocol de comunicație și în particular cel de TCP/IP;
- cunoaște și utiliza clasele Java pentru lucrul în rețea;
- implementa o aplicație de rețea client-server.

# Concepte despre rețele de calculatoare și protocoale de rețea

În domeniul rețelelor de calculatoare, termenul **dispozitiv utilizator** este folosit pentru a defini un echipament electric care este conectat la rețea (calculator, imprimantă, scanner etc.) în scopul schimbului de date cu un alt echipament al rețelei. Denumirea e dată din punctul de vedere al utilizatorului de rețea (de exemplu dacă discuția nu ar fi la nivel de rețea ci de calculator, dispozitive ar putea fi: disc, placă video, modem etc.). Termenul **dispozitiv de rețea** este folosit pentru a descrie echipamentele electrice necesare pentru interconectarea dispozitivelor utilizator în vederea schimbului de date. Aceste dispozitive sunt conectate prin cabluri electrice (sau, direct, prin undă radio - wireless) și asigură transmisia, recepția, gestionarea datelor ce se doresc a fi transferate între dispozitivele utilizator. Câteva dintre numele de dispozitive vehiculate mai des sunt prezentate, pe scurt, în continuare:

*Repeater (repetor)* - dispozitiv de rețea care se folosește pentru refacerea sau repetarea semnalelor (există repetitoare care pot face retransmisia datelor între două rețele care folosesc protocoale diferite).

*Hub (concentrator)* - punct de conectare comun la nivelul unei rețele, se folosesc pentru interconectarea segmentelor (segmentul este o porțiune de rețea între două bridge-uri, router-e sau switch-uri) de LAN. Hub-ul are mai multe porturi (interfețe de plăci de rețea). Atunci când un pachet (mesajele de transmis sunt divizate în pachete, fiecare pachet se transmite individual și poate parcurge o rută individuală către destinație) ajunge pe un anumit port se copiază pe toate celelalte porturi pentru ca toate segmentele de LAN să-l vadă. Există și hub-uri inteligente care citesc adresa destinație a pachetelor și le trimit numai unui singur port, acestea mai pot avea și diferite facilități de monitorizare și administrare a traficului.

*Bridge (punte)* - dispozitiv de rețea ce conectează două LAN-uri sau două segmente de LAN ce folosesc același protocol (de exemplu, Ethernet).

*Switch (comutator)* - dispozitiv de rețea ce filtrează și expediază pachete între două segmente de LAN. Switch-ul operează la nivelul legăturii de date (nivelul 2 OSI - Open System Interconnection) eventual la nivelul numit rețea (nivelul 3 OSI).

*Router* - un dispozitiv ce transferă pachete de date între rețele distincte. Tipic, router-ul este conectat la două LAN-uri sau WAN-uri respectiv între un LAN și furnizorul de servicii Internet. Router-ele realizează dirijarea eficientă a pachetelor de date pe liniile cele mai convenabile.

*Gateway* - un calculator din rețea care se folosește pentru accesarea unei alte rețele, deoarece este unicul punct de interacțiune cu alte rețele, deseori, va fi și proxy server și firewall; dacă rețeaua este subnetată (are subrețele) este router-ul care transferă traficul în afara subrețelei din care face parte stația transmițătoare.

*Proxy server* - un server (un calculator care gestionează anumite resurse) care este “așezat” între un client de aplicație, de exemplu un navigator de Web, și un server real, de exemplu un server de Web. Proxy server-ul va intercepta cererile către server-ul real și va încerca să le îndeplinească singur, dacă nu poate transferă cererile server-ului real. Avantajele folosirii unui proxy stau în creșterea performanțelor (răspunsul este dat imediat, nu mai ajunge la server-ul real care este bombardat de mii de cereri și cam ocupat cu ele) și filtrarea cererilor.

*Firewall* - un sistem electric sau o aplicație ce gestionează accesul (filtrează) la sau de la o rețea privată.

### **Modalități de clasificare a rețelelor**

Există multe criterii de clasificare a rețelelor, iată câteva posibilități:

- extinderea geografică: LAN, MAN, WAN etc.;
- topologie: stea, magistrală, inel etc.;
- mediul de transmisie: cupru, fibră optică, radio, microunde, satelit;
- proprietar: public, privat;
- tehnologia folosită: comunitată (switched) / conexiune permanentă (permanent links), cu legătură fizică (physical link) sau virtuală (virtual link), orientată pe conexiune (connection-oriented) sau fără conexiune (connectionless), cu difuzare (broadcast) sau punct la punct (point-to-point) etc.;
- viteză: bandă largă (broadband), bandă îngustă (narrowband).

Nu voi discuta toate tipurile posibile, voi aminti însă cele mai uzuale denumiri.

### **Rețele LAN (Local Area Networks)**

Rețeaua locală permite interconectarea unor calculatoare aflate la distanță mică (un birou sau mai multe birouri) în scopul partajării fișierelor din calculatoare, a imprimatelor și a comunicației locale. Multe dintre LAN-urile actuale folosesc uzual tehnologia Ethernet. Alte tehnologii folosite pentru rețele LAN sunt Token Ring și FDDI.

### **Rețele WAN (Wide Area Networks)**

Rețeaua WAN este, tipic, răspândită pe o arie geografică mai mare decât LAN-ul (distanța între procesoare este de ordinul kilometrilor sau mai mult). WAN-ul se formează prin conectarea mai multor LAN-uri și asigură accesul la calculatoarele locale, server-e, imprimate de la o distanță mare respectiv comunicația între calculatoare. Câteva dintre tehnologiile specifice WAN-urilor sunt: modem-urile, ISDN (Integrated Service Digital Network), DSL (Digital Subscriber Line), SONET (Synchronous Optical Network).

### **Rețele MAN (Metropolitan Area Networks)**

MAN este o rețea de date ce are o răspândire geografică între LAN și WAN, cel mai des la nivelul

unui oraș. Majoritatea MAN-urilor beneficiază de conexiuni private dar foarte rapide pe fibră optică sau alte medii de conexiune digitală rapidă. Se formează prin legarea mai multor LAN-uri, eventual și prin unde radio pentru a oferi aceste servicii în spații publice (aeroport, magazine, etc.)

### **Rețele VPN (Virtual Private Networks)**

VPN este o rețea care folosește un mediu de transmisie public pentru a forma o rețea privată de calculatoare. Oricine are acces la această rețea va putea lucra cu toate facilitățile ei, în condiții de securitate bună, chiar dacă se află la distanță fizică mare de rețea și se leagă la ea folosind o conexiune clasică la Internet (care este nesecurizată). Diferite metode de criptare și mecanisme de securizare sunt folosite pentru a permite doar accesul utilizatorilor autorizați la rețea și pentru evitarea interceptării datelor.

### **Condițiile de conectare a unui calculator la o rețea**

Pentru ca un calculator să poată fi legat la o rețea, prin care să putem avea acces la Internet și la WWW, acesta trebuie să:

- aibă cel puțin o placă de rețea;
- folosească un sistem de operare ce știe lucra cu protocoale TCP/IP (Transmission Control Protocol/Internet Protocol);
- aibă instalate aplicațiile ce interpretează și afișează datele transferate prin rețea într-o formă inteligibilă pentru oameni.

### **Placa de rețea**

Pentru conectarea unui calculator la o rețea de calculatoare acesta trebuie să conțină, la nivel fizic, un dispozitiv de rețea numit placa de rețea (NIC - Network Interface Card). Acesta asigură comunicația electrică într-o rețea. Placa de rețea folosește o linie de întrerupere (IRQ - Interrupt ReQuest), un spațiu de adrese I/O (Input/Output - Intrare/Ieșire) și o zonă de RAM (Random Access Memory - memorie volatilă) pentru a interacționa cu SO (Sistemul de Operare). O linie de întrerupere este o linie electrică specială în calculator prin care acesta poate fi oprit din funcționare normală de către un dispozitiv care transmite un semnal electric pe linia respectivă. Semnalul întrerupe temporar funcționarea curentă a calculatorului pentru ca acesta să poată decide ce urmează să facă. Dacă procesul care prezintă date este rapid, atunci calculatorul va prelua datele, altfel acestea riscă să se piardă, după care își va relua activitatea din care a fost oprit. Deoarece mai multe semnale de întrerupere pot sosi pe rând sau simultan pe o singură linie de întrerupere, fiecărui dispozitiv ce poate întrerupe calculatorul trebuie să i se dea un număr unic pentru a putea fi identificat de calculator. Această valoare este specificată atunci când respectivul dispozitiv este introdus fizic în calculator și se poate face manual sau automat (asta dacă dispozitivul este PnP - Plug and Play). Placa de rețea este un astfel de dispozitiv ce trebuie să primească un număr de IRQ. Aceasta trebuie instalată pe orice echipament ce va accesa rețeaua deoarece asigură cuplarea la nivel electric cu aceasta. Există mai multe tipuri de plăci de rețea:

- ◆ notebook-urile pot avea placa de rețea încorporată sau atașată extern pe prin PCMCIA;
- ◆ PC-urile pot avea și ele placă de bază cu placă de rețea încorporată sau pot fi atașate ca plăci distincte.

Fiecare placă de rețea are un număr de identificare unic numit adresă MAC (Media Access Control).

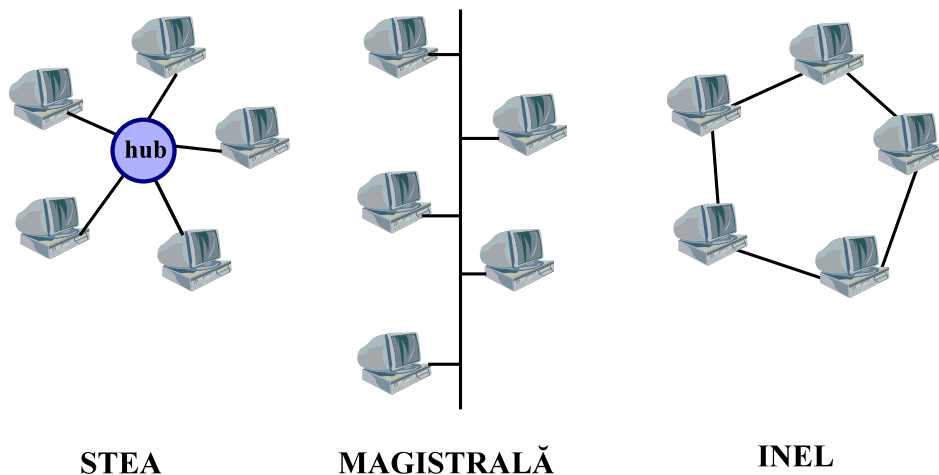
Acest număr este folosit (de exemplu, în foarte răspândita tehnologie de rețea Ethernet - o arhitectură de rețea locală dezvoltată de firma Xerox) pentru a derula comunicația de date a stației (numită host) prin rețea.

Rețeaua Ethernet apare în 1976, trecând ulterior prin mai multe revizii și standardizări. Aceasta folosea pentru conectarea fizică a plăcilor de rețea un singur cablu coaxial (cablul de rețea). Datele transmise în această rețea erau sparte în pachete. Denumirea de pachet este, azi, un termen generic, reprezentând unitatea de date ce se transmite într-o rețea. Pachetele de date erau transmise pe baza unui algoritm CSMA/CD (Carrier Sense Multiple Access / Collision Detection). În situația în care mai multe pachete de date încercau să fie transmise simultan de calculatoare diferite ale rețelei pe cablul de rețea apărea o coleziune (coleziune era detectată și pachetul era retransmis după un timp aleator). Implementarea unei astfel de rețele este simplă, dar ca urmare a competiției calculatoarelor în vederea accesului la cablul de rețea (mediul de transmisie), caracteristicile de transmisie nu sunt deterministe. Viteza tipică a transferului de date pe această rețea era de 10 Mbps. Ca urmare a simplității, tehnologia Ethernet a evoluat repede mărindu-și viteza transferului de date la 100 Mbps (Fast Ethernet) respectiv 1 Gbps (Gigabit Ethernet).

### Topologii de rețea

Topologia se referă la modul de aranjare a legăturilor fizice între calculatoarele rețelei. Câteva dintre ele se văd în **Figura 1** (star - stea, bus - magistrală, ring - inel). Topologia stea conectează toate cablurile (pe fiecare cablu este un singur calculator) la un singur punct central (hub-ul). Topologia magistrală folosește un singur cablu pe care se leagă toate calculatoarele, cea inel leagă două câte două calculatoare și primul cu ultimul rezultând o rețea de cablu în forma inelară. Termenul de topologie logică este folosit pentru de defini modul de comunicație în rețea și poate fi cu difuzare (broadcast) sau cu transfer de jetoane (token passing). În cazul difuzării, un calculator transmite mesaje către toate celelalte calculatoare ale rețelei, prin cablul de rețea.

Figura 1 -  
Topologii de  
rețea



Nu există o ordine specifică în care calculatoarele folosesc mediul de transmisie, din acest motiv pot să apară și conflicte (după cum am spus deja, Ethernet funcționează după acest model). În cazul

lucrului cu jeton, acesta este transmis secvențial fiecărui calculator din rețea. Calculatoul care are jetonul (token-ul) poate transmite date în rețea, dacă acesta nu are date de transmis va pasa jetonul mai departe (nu-l ține). Acest proces se repetă cu parcurgerea tuturor calculatoarelor din rețea. TokenRing este o rețea care lucrează după această procedură.

### Despre protocoale de rețea

Rețelele de comunicație între calculatoare pot fi organizate ca o grupare de protocoale aproape independente, fiecare dintre acestea operând la un anumit nivel. Tipic, nivelul cel mai de jos asigură comunicația fizică între calculatoare, în timp ce nivelul cel mai înalt constă în aplicațiile utilizatorului. Fiecare nivel curent, se folosește de nivelul inferior, pentru a pune la dispoziția nivelului superior, un grup de servicii. La fiecare nivel, programe rulate pe calculatoarele rețelei folosesc protocoale corespunzătoare nivelului pentru a comunica între ele. Avantajul protocoalelor pe nivele constă în specificațiile clare ale metodelor de transmitere a informațiilor de la un nivel la altul ca parte a protocolului. Astfel, orice modificare la nivelul protocolului nu afectează celelalte nivele, această independență simplificând proiectarea și întreținerea programelor de comunicație.

Pentru cele ce urmează voi considera că suita de procoale TCP/IP este organizată pe 5 nivele de protocoale. Prezentarea este mult simplificată fiind meținute numai acele concepte care au fost deja discutate, respectiv cele ce vor fi utilizate mai departe la Web și Internet. Anumitor nivele le corespund anumite concepte și dispozitive de rețea, după cum urmează:

Nivel TCP/IP	Descriere	Dispozitive de rețea și concepte specifice nivelului
5 - Aplicație	Asigură transparența rețelei, alocarea de resurse etc. la nivelul utilizatorului rețelei.	
4 - Transport (TCP)	Asigură realizarea unei conexiuni un punct A și unul B, pentru ca datele să ajungă fără erori și în ordinea corectă.	
3 - Internet (IP)	Asigură determinarea rutei pachetelor de date de la transmițător la receptor.	router, subrețea, adresă IP
2 - Placă de rețea	Asigură fragmentarea datelor în pachete ce conțin informații de indentificare și urmărire în vederea transmierii la nivelul fizic.	switch, adresă MAC, Ethernet
1 - Fizic	Corespunde conexiunilor electrice și mecanice.	hub

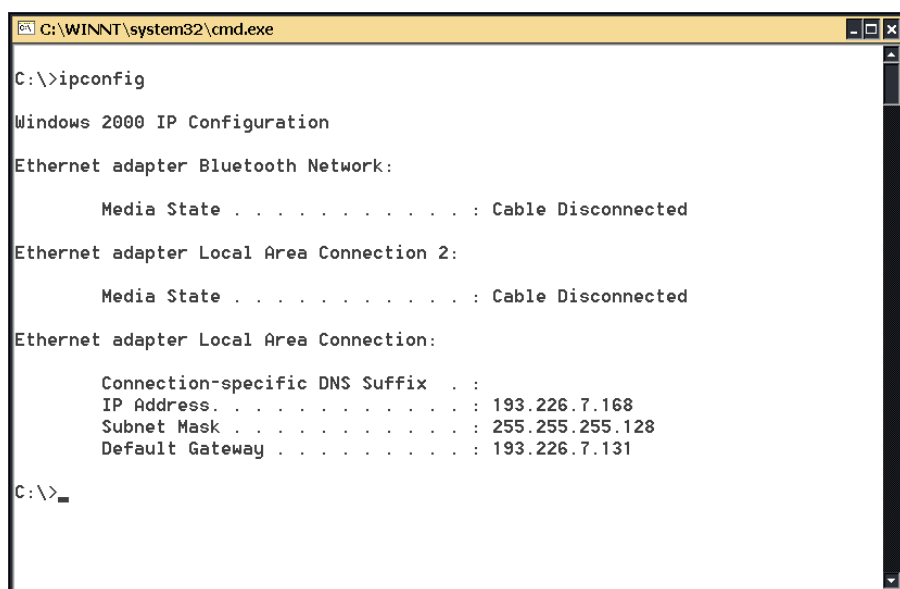
### Introducere în TCP/IP

Aceast protocol a devenit interesant atunci când proprietarii de LAN-uri au înțeles că poate fi utilizat atât pentru transportul datelor într-o LAN cât și pentru interconectarea mai multor LAN-uri. La interconectarea LAN-urilor datele pot fi transferate între calculatoarele unor LAN-uri distincte, dar

numai cu ajutorul unor dispozitive de rețea de tipul router sau gateway. TCP este un protocol de comunicație sigur (garantează că toate datele ajung la destinație fără erori, în ordinea în care s-au trimis), asigură controlul fluxurilor (atunci când receptorul nu mai poate accepta date acesta va opri transmisia cu un semnal, iar când recepția devine din nou posibilă o va reporni cu un semnal), este multiplexat (mai multe semnale electrice sunt combinate pentru a partaja mediul de transmisie respectiv cablul de rețea) și orientat pe conexiune (transportul de date între calculatoare se face într-un flux continuu, în trei faze bine definite: stabilirea conexiunii, transferul de date și eliberarea conexiunii). Protocolul IP este fără conexiune, ce lucrează cu priorități (tipurilor de trafic li se atribuie priorități, în funcție de prioritate traficul poate sau nu tolera întârzieri, de exemplu videoconferințele trebuie să “meargă” în timp real, în timp ce e-mail-ul poate fi întârziat) și comutare de pachete (pachetele de date sunt expediate individual între calculatoare, fără existența unei rute prestabilite). IP asigură transportul pachetelor pe diferite rute, fragmentarea (proces prin care un pachet este divizat în porțiuni mai mici numite fragmente, cu scopul adaptării acestora la cerințele rețelei fizice prin care pachetele trebuie să treacă; procesul invers se numește reasamblare) și reasamblarea pachetelor.

### Configurarea conexiuni TCP/IP în Windows XP și 2000

Sistemul de operare Windows trebuie să fie configurat să lucreze cu TCP/IP pe fiecare calculator al rețelei pentru ca să poată exista o comunicație între acestea. Pentru vizualizarea setărilor TCP/IP, în Windows 95, 98 și Me se folosește aplicația `winipcfg`, în NT/2000 și XP se folosește `ipconfig`.



```
G:\WINNT\system32\cmd.exe
C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Bluetooth Network:

    Media State . . . . . : Cable Disconnected

Ethernet adapter Local Area Connection 2:

    Media State . . . . . : Cable Disconnected

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . . :
    IP Address . . . . . : 193.226.7.168
    Subnet Mask . . . . . : 255.255.255.128
    Default Gateway . . . . . : 193.226.7.131

C:\>=
```

Calculatorul trebuie să fie conectat la rețea pentru ca datele (vezi **Figura 2**) cu privire la conexiune să poată fi afișate.

Pentru ca un grup de calculatoare să fie într-o rețea locală trebuie ca la:

- **IP Address:** primele 3 grupe de numere să fie aceleași, iar numerele din ultima grupă se fie, toate, distincte;
- **Subnet Mask:** să fie aceeași;
- **Default Gateway:** să fie aceeași.

Funcționarea conexiunii la rețea se face cu aplicația `ping` (Packet InterNet Groper). Aceasta verifică dacă o adresă de IP există și dacă acceptă cereri. `ping` lucrează trimițând un pachet special numit

ICMP (Internet Control Message Protocol Echo Request) către o anumită destinație. Fiecare pachet trimis este o cerere de răspuns, `ping` se folosește pentru a verifica dacă funcțiile de transmisie și recepție ale NIC, configurarea TCP/IP și a rețelei sunt funcționale.

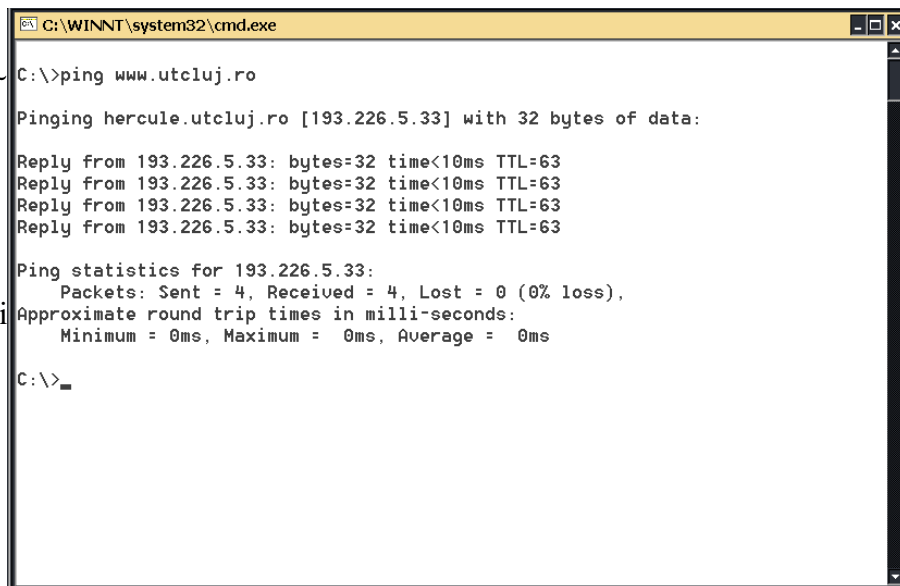
`ping 127.0.0.1 - 127.0.0.1` este o adresă rezervată pentru testare a conexiunii între client și server (loop back address test). Aceasta permite realizarea unei conexiuni între client și server prin TCP/IP pe același calculator (nu este nevoie de alte elemente de rețea). Fiecare calculator are adresa 127.0.0.1 asignată interfeței de loop back care mai poartă și numele de “local host”, majoritatea NIC-urilor au implementată această buclă de testare internă. Orice transmisie pe acest port nu iese din NIC pe cablul de rețea ci este redirectată spre interfața de testare și ajunge din nou în coada de intrare ip. Se emite o cerere de ecou și un răspuns la acesta prin mesaj ICMP prin care se verifică condițiile de funcționare ale NIC-ului și a instalării respectiv ale configurării corecte ale TCP/IP.

`ping IP_adresa_calc_local` - verifică dacă configurarea TCP/IP a rețelei este corectă, adică dacă IP-ul este adăugat corect rețelei locale și dacă dispozitivul de rețea la care este legat calculatorul este activ (pornit).

`ping IP_adresa_default_gateway` - verifică dacă conexiunea cu router-ul (default gateway-ul) funcționează corect și dacă se poate face legătura cu alte calculatoare din rețeaua locală.

`ping IP_adresa_calc_alta_retea` - verifică dacă se poate comunica prin router cu calculatorul aflat la distanță într-o altă rețea.

Dacă totuși funcționează bine `ping` întoarce adresa verificată. Dacă răspunsul este “Request timed out” atunci nu s-a primit răspuns în timpul alocat de la calculatorul `ping`-uit. Durata de timp alocată așteptării răspunsului se modifică folosind opțiunea `w`, de exemplu pentru așteptare de 5 secunde scriem `ping -w 5000 adr_IP`.



```
C:\WINNT\system32\cmd.exe
C:\>ping www.utcluj.ro
Pinging hercule.utcluj.ro [193.226.5.33] with 32 bytes of data:

Reply from 193.226.5.33: bytes=32 time<10ms TTL=63
Reply from 193.226.5.33: bytes=32 time<10ms TTL=63
Reply from 193.226.5.33: bytes=32 time<10ms TTL=63
Reply from 193.226.5.33: bytes=32 time<10ms TTL=63

Ping statistics for 193.226.5.33:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>_
```

**Figure 3** - ping www.utcluj.ro

Pe Windows XP și 2000 mai puteți utiliza și `pathping` care face mult mai multe (determină IP-ul unui calculator dintr-o altă rețea, întârzierile și pierderile de pachete de date).

În rețelele bazate pe protocoalele TCP/IP (aici este inclus și Internet-ul) calculatoarele se mai numesc



și gazde (host), fiecare gazdă trebuie să aibă o adresă de IP-unică. TCP permite stabilirea unei conexiuni între două host-uri cu scopul interschimbării unor fluxuri de date. TCP garantează transferul de date și recepția pachetelor în aceeași ordine în care s-au transmis. Protocolul IP se ocupă numai de pachete. Aici este specificat formatul pachetelor de date (numite și datagrame) și schema de adresare. IP se aseamănă cu sistemul poștal. Permite adresarea și punerea unui pachet la poștă, dar fără să facă o legătură directă între expeditor și destinatar. TCP este cel care face această legătură virtuală între sursă și destinație. Împreună, TCP/IP, permit stabilirea unei conexiuni între două gazde în scopul transferului de mesaje pentru o durată limitată de timp. Versiunea curentă a protocolului IP este 4, de aceea se mai folosește și scrierea IPv4. Ca urmare cerințelor de creștere a Internetului, atât în număr de gazde cât și de trafic, a fost creat noul standard IPv6.

Adresa de IP identifică unic un calculator sau alt dispozitiv (imprimantă) al rețelei TCP/IP. Rețelele ce folosesc protocolul TCP/IP dirijează mesajele pe baza adresei IP a destinatarului. Forma unei adrese IP este un număr de 32 de biți, scrise ca 4 numere întregi separate prin puncte. Fiecare număr poate fi între 0 și 255, de exemplu 168.212.226.204 poate fi o adresă de IP. Într-o rețea izolată de Internet atribuirea acestor adrese se poate face aleator, condiția fiind aceea ca numerele folosite sunt unice. Dacă această rețea privată va fi însă legată la Internet este obligatorie folosirea unor adrese de IP înregistrate (pentru evitarea dublurilor). Cele patru numere se folosesc pentru a identifica o anumită rețea și o gazda particulară. 168.212.226.204 se scrie în binar 10101000.11010100.11100010.11001100. Dacă subnet mask-ul este 255.255.255.000, adică 11111111.11111111.11111111.00000000 în binar, primii 24 de biți din IP se folosesc pentru identificarea rețelei, iar ultimii 8 identifică calculatorul gazdă (host-ul) din rețea. Dimensiunea rețelei este funcție de numărul de biți folosiți la definirea gazdelor. Dacă subnet mask-ul are 8 biți pot exista cel mult 256 de adrese de gazde pentru o anumită rețea. Dacă subnet mask-ul are 16 biți (255.255.0.0) atunci sunt 65536 de adrese de gazde disponibile în rețea.

În concluzie, adresa IP este formată din două regiuni, una care identifică rețeaua, alta care identifică gazda (nodul din rețea). Clasa adresei determină care parte a adresei aparține de rețea și care aparține de gazdă. Toate gazdele unei rețele trebuie să aibă același prefix de rețea și trebuie să aibă o valoare de gazdă unică.

Există mai multe organizații (de exemplu, pentru Europa avem Réseaux IP Européens Network Coordination Centre.) care înregistrează și administrează adresele de IP din următoarele clase:

Clasă	Cei mai semnificativi biți ai adresei	Domenii de adrese posibile	Adresa de subnet mask	Rețele pe clasă	Gazde pe clasă	Utilizare
A	0xxx	0.0.0.0 - 127.255.255.255	255.0.0.0	128	16.777.214	Rețele foarte mari, întotdeauna subnetate
B	10xx	128.0.0.0 - 191.255.255.255	255.255.0.0	16,384	65,535	Rețele mari, tipic, subnetate
C	110x	192.0.0.0 - 223.255.255.255	255.255.255.0	2,097,152	254	Rețele mici
D	1110	224.0.0.0 - 239.255.255.255	255.255.255.255	268,435,456	0[2]	Grupuri de adrese de multicast (nu există gazde)

E	1111	240.0.0.0 - 255.255.255.255	nedefinită	nedefinite	nedefinite	Rezervate în scopuri experimentale
---	------	-----------------------------	------------	------------	------------	------------------------------------

Este posibil ca administratorul unei rețele să divizeze o rețea în mai multe rețele virtuale numite subrețele (procesul este numit “subnetare”). Pentru aceasta biții din subnet mask care țin de gazdă se “împrumută” adresei de rețea prin trecerea biților doriți din 0 în 1. De exemplu, fie adresa de rețea 192.168.10.0 cu subnet mask-ul 255.255.255.0. Pentru a face din această rețea două rețele subnet mask-ul se face 255.255.255.128 (adică din 11111111.11111111.11111111.00000000 acesta devine 11111111.11111111.11111111.10000000). Aceasta înseamnă ca în loc de 24 de biți se folosesc 25 de biți pentru rețea și numai 7 pentru host-uri. Rezultă două rețele distincte cu 128 de adrese de host posibile fiecare. Prima rețea va avea domeniul de adrese în 192.168.10.0 - 192.168.10.127, iar cea de-a doua 192.168.10.128 - 192.168.10.255.

Toate rețelele trebuie să rezerve orice adresă de gazdă ce are toți biții 0 respectiv 1 pentru rețea. Astfel fiecare subrețea va avea o adresă specifică de rețea și una de difuzare (broadcast). Într-o rețea cu 256 de gazde, 0 și 255 vor fi adrese rezervate, deci, vor fi numai 254 de adrese de gazde posibile. Pentru exemplul de mai sus, când numai 7 biți se folosesc pentru adresele de gazde, numărul maxim ar fi de 126 din 128 (pentru prima subrețea 192.168.10.0 este adresa de rețea iar 192.168.10.127 adresa de difuzare).

Toate gazdele care dorim să **comunică direct** în rețea trebuie să fie în aceeași rețea, adică să aibă același subnet mask. Dacă se folosesc subnet mask-uri diferite vor crede că sunt în rețele diferite și nu vor putea comunica direct, ci doar indirect, cu ajutorul unui router.

Fără a intra în prea multe detalii, protocolul TCP/IP este cel ce stă la baza Internetului. Acesta asigură transportul datelor între aplicațiile, rulate pe calculatoarele unor rețele distincte. Ceea ce numim generic Internet, este cea mai mare (în sensul răspândirii geografice) grupare de rețele de calculatoare. Are o structură ierarhică cu trei nivele: rețele de tranzit cum sunt ARPAnet, NSFNet, MILNET - asigură traficul de volum foarte mare între rețele pe care le leagă, rețele de mijloc - rețele intermediare, care conectează rețelele mici la cea de tranzit, rețele mici - la nivelul unei astfel de rețele traficul este de tipul local.

# Conceptul de socket in contextul aplicatiilor client-server

Pachetul `java.net` asigură suportul Java pentru lucrul cu rețele. Aceste clase au la bază conceptul de **socket** dezvoltat în cadrul Berkley Software Distribution (BSD) la Universitatea Berkley din California. Un socket (priză) de rețea este un mecanism de creare a unei conexiuni virtuale între procese. Socket-ul este o interfață între procese și facilitățile de comunicare în rețea. Socket-urile se creează pe baza claselor specifice ce întorc un descriptor către socket-ul creat prin care acesta poate fi accesat în codul Java. Socket-ul are asociat o adresă formată dintr-un număr de port și din adresa de rețea mașinii (IP-ul) pe care se rulează codul.

Aplicațiile client-server menționate în contextul rețelelor de calculatoare au la bază socket-uri. Server-ul este o mașină care partajează anumite resurse, iar clienții sunt mașini care doresc să acceseze respectivele resurse. Noțiunea de socket permite ca o singură mașină să poată deservi, simultan, mai mulți clienți și în același timp să poată servi mai multe tipuri de informații. Aceasta se realizează prin introducerea conceptului de port, care este un număr întreg asociat socket-ului mașinii. Se zice că server-ul ascultă un anumit port până când un client se leagă la acesta. Server-ul poate accepta mai multe conexiuni, simultan, de la mai mulți clienți același port, fiecare sesiune fiind însă unică. Pentru gestionarea conexiunilor cu clienți multipli server-ul trebuie să lucreze cu fire de execuție multiple sau să implementeze o metodă de multiplexare a operațiilor de intrare/ieșire simultane.

Comunicația în rețea se face pe baza protocoalelor. IP (Internet Protocol) este un protocol rutare de nivel jos ce sparge datele în pachete și le trimite prin rețea fără a avea însă garanția livării lor la destinație. TCP (Transmission Control Protocol) este un protocol de nivel înalt care gestionează robust pachetele de date prin sortarea și retransmiterea lor dacă este cazul pentru a ajunge corect la destinație.

Orice mașină de pe Internet are o adresă care o identifică unic. Adresa de IP are 32 de biți și se scrie ca o secvență de 4 numere întregi între 0 și 255 separate prin punct (de ex.: 168.212.226.204). Adresa IP este formată din două regiuni, una care identifică rețeaua, alta care identifică mașina din rețea. Clasa adresei determină care parte a adresei aparține de rețea (se specifică în subnetmask) și care aparține de mașinii. Toate mașiniile unei rețele trebuie să aibă același prefix de rețea și trebuie să aibă o valoare de mașină unică.

# Obiectul Socket client si server

Socket-urile TCP/IP implementează o conexiune sigură din punctul de vedere al transmiterii datelor, bidirecțională, punct-la-punct, bazată pe fluxuri între mașini din rețea sau de pe Internet. Socket-ul se folosește pentru a conecta sistemul de intrare/ieșire din Java cu alte programe care sunt rezidente local sau pe altă mașină din rețea sau de pe Internet. Există două feluri de socket-uri în Java, socket-ul server care așteaptă conectarea unui client înainte de face orice altceva și socket-ul client care se conectează la server și inițiază transferul de date pe Internet. Crearea unui obiect **Socket** stabilește implicit o conexiune între client și server. Nu există metode sau constructori care să expună detaliile stabilirii unei conexiuni. Constructorul pentru crearea unui socket client este `Socket(IP, port)`, iar pentru crearea unui server socket este `ServerSocket(port)`. IP-ul din socket-ul este cel al server-ului, iar portul trebuie să fie același.

Codul tipic pentru crearea unui server este:

```
ServerSocket srv= new ServerSocket(port, lungimecoada);
```

Gestionarea conexiunii cu un client se face prin clasa `Socket`. Așteptarea conectării unor clienți la server se face prin apelarea metodei `accept()` într-un ciclu infinit codul tipic fiind:

```
Socket con = srv.accept();
```

care întoarce un obiect `Socket` dacă conexiunea s-a realizat cu un client.

Server-ul trimite date clientului prin obiectul `OutputStream`, iar server-ul primește date de la client prin obiectul `InputStream`. Server-ul apelează metoda `getOutputStream()` pe `Socket` pentru a stoca o referință la fluxul de ieșire și apelează metoda `getInputStream()` pentru a stoca o referință la fluxul de intrare al Socket-ului. Transmiterea de date se face efectiv prin apelul metodelor `out.write()` și `in.read()`.

```
ObjectInputStream in = new ObjectInputStream(con.getInputStream());  
ObjectOutputStream out = new ObjectOutputStream(con.getOutputStream());
```

După transmisie urmează prelucrarea datelor și în final terminarea comunicației prin apelul metodei `close()`.

```
in.close();  
out.close();  
con.close();
```

Dacă se dorește implementarea unui server ce poate accepta conexiuni multiple trebuie să avem câte un Thread pentru fiecare conexiune de client. Server-ul porneste câte un Thread nou pentru fiecare conexiune nouă, codul tipic fiind:

```
while (true) {
    Socket s = srv.accept();

    Runnable r = new ServerMThread(s);
    Thread t = new Thread(r);

    t.start();
}
```

Această variantă de server rulează fără oprire. Clasa ServerMThread trebuie sa implementeze o interfață Runnable. Codul trebuie scris în metoda run () care primește, interpretează și trimite răspunsuri înapoi la client.

La client realizarea conexiunii cu server-ul se face prin :

```
Socket cli = new Socket(IPserver, port);
```

Comunicația cu server-ul se derulează folosind codul:

```
ObjectInputStream cin = new ObjectInputStream(cli.getInputStream());
ObjectOutputStream cout = new ObjectOutputStream(cli.getOutputStream());
```

prin obiectele cin si cout.

În cazul în care se dorește utilizarea de fulxuri text se poate lucra cu clasele PrintWriter pentru scriere și Scanner pentru citire.

## **C14- Întrebări**

1. Care este rolul unei rețele de calculatoare?
2. Ce este TCP/IP?
3. Unde se folosesc ServerSocket și Socket în cod?
4. Ce metodă se folosește pentru așteptarea conexiunilor de clienți?
5. Cum se implementează un server care poate accepta, simultan, conexiuni multiple?