

Rutarea în Internet

Mihai Budiu — mihaib+@cs.cmu.edu
<http://www.cs.cmu.edu/~mihaib/>

Aprilie 1998

Subiect:

Cunoștințe necesare:

Cuvinte cheie:

Cuprins

1	Internet-ul: scurt istoric	2
2	Acest articol	3
3	Protocoale	4
4	Alte rețele	5
5	Internet-ul: o rețea de rețele	6
6	IP	7
7	Problema rutării	7
8	Adrese	8
9	Rețele și noduri	9
10	Interfața cu sub-rețeaua: ARP, ifconfig	10
11	Tabelele de rutare	11
12	Rutele implicite (default)	12
	12.1 route și netstat	13
13	Ruterele și coloana vertebrală (backbone)	13
14	Protocoale de rutare	14

15 Administrarea rețelei	15
15.1 ping	15
15.2 traceroute	16
16 Alte probleme	18
16.1 Stabilitate	18
16.2 Multicast	19
16.3 Rutare mobilă	19

1 Internet-ul: scurt istoric

La mijlocul anilor '60 compartimentul de cercetare al Departamentului Apărării al Statelor Unite (DARPA: Defense Advanced Research Projects Agency), un organism beneficiind de un buget strivitor, a pornit un proiect de cercetare pentru construirea unei rețele strategice pentru transmisiunea datelor între calculatoarele Pentagonului. Rețeaua trebuia să satisfacă două cerințe:

- Să lege între ele scumpele calculatoare ale agenției și eventual ale altor centre de cercetări;
- Să reziste cu succes în fața unui atac nuclear.

Rețeaua sponsorizată de DARPA a fost numită ARPAnet (network, net = rețea). Proiectul inițial a fost făcut în 1966 iar în 1969 primul prototip care lega 4 calculatoare, amplasate la mai multe universități și laboratoare de cercetare, era operațional.

În același timp universitatea Stanford construia un alt prototip de rețea, sponsorizat tot de DARPA, care folosea undele radio pentru transmisiunea datelor (ARPAnet transmitea datele prin linii telefonice închiriate). Cercetătorii de la Stanford ar fi dorit să interopereze cu cealaltă rețea. Din cauza asta proiectanții au adoptat o serie de reguli asemănătoare pentru transmisiunea datelor, dar au construit ansamblul în așa fel încât cele două rețele componente să poată fi administrate și dezvoltate independent, fără intervenții speciale ale unei autorități centrale. De asemenea un factor crucial în design a fost decizia ca rețeaua să poată opera în același fel indiferent de natura mediului de transmisie folosit.

La ARPAnet se puteau conecta universitățile și centrele de cercetare care primeau bani de la DARPA. Multe alte universități ar fi dorit însă să participe la conexiune. Fiecare dintre ele a dezvoltat propria ei rețea peste care a suprapus tehnologiile din ARPAnet. Astfel, mai multe rețele diferite au adoptat standardele propuse în ARPAnet pentru comunicație și au putut interoperă. La începutul anilor '80 această federație de rețele, din care ARPAnet era doar o părțică, s-a numit "Internet". Numele înseamnă "inter = între", "net = rețele", indicând faptul că protocoalele permit comunicarea între mai multe rețele.

Internet a crescut cu o viteză uluitoare, legînd din ce în ce mai multe universități, centre de cercetare și alte instituții de același gen. Pentru că banii pentru funcționarea sa veneau de la guvern (inițial de la armată, apoi, de la mijlocul anilor '80 de la National Science Foundation — NSF, care este o agenție guvernamentală care sponsorizează cercetarea științifică de la buget), activitățile comerciale erau strict interzise.

Internet-ul s-a expandat în afara Statelor Unite, cuprinzând instituții de pe toate continentele. După căderea cortinei de fier el a devenit accesibil și în fostele țări comuniste (fiecare participant acoperea costurile unei anumite părți din Internet, mod de operare care este valid și astăzi).

În 1992 a fost ridicată restricția privind activitățile comerciale pe Internet, ceea ce a adus gradual în scenă o cantitate enormă de participanți. În 1993 ARPAnet a fost complet desființată. Suportul federal pentru rețea a fost încetat complet pe 1 mai 1995; NSF continuă să sponsorizeze accesul unor universități. La ora actuală comerțul pe Internet și tehnologiile Internet constituie una dintre cele mai prospere activități din Statele Unite, și, pe o scară ceva mai mică, din restul lumii. Companii ca Netscape, care s-a ridicat de la absolut nimic, în 1994, la o valoare la bursă de peste 1 miliard (sic!) de dolari, în 1995, semnifică importanța economică crescândă a rețelei.

O adevărată industrie de ISP: Internet Service Providers: firme care oferă servicii de conexiune la Internet au apărut în toată lumea. La ora actuală în Internet se estimează ca sunt conectate peste 30 de milioane de calculatoare de pe tot globul, și că peste 100 de milioane de oameni beneficiază de serviciile lui în mod regulat. Rata de creștere continuă să fie exponențială. Se estimează că 30% din populația SUA folosește Internet-ul; acesta a devenit un dispozitiv de uz casnic (aproape) la fel de obișnuit ca mașina de spălat sau televizorul.

România este legată la Internet cel puțin din 1992. La ora actuală nodul cu cea mai mare importanță este aflat la Politehnica din București; în jur de 80% din traficul din România trece prin acest punct, pentru că aici se află un nod care oferă legătură marilor centre universitare din țară (informațiile la zi pot fi obținute de la (<http://www.roedu.net>)¹). În multe orașe au apărut furnizori de servicii Internet.

Iată cum deciziile inițiale de descentralizare și inter-operare au permis o dezvoltare uluitoare care întrece imaginația celor mai vizionari dintre arhitecții Internet-ului.

Internet-ul are propriul lui corp de standardizare, numit IETF “Internet Engineering Task Force”, care este format din experți din universități, centre de cercetare și mari firme de calculatoare și telecomunicații. Ritmul de lucru la IETF este uluitor de rapid (comparat cu alte corpuri de standardizare), și este în mare măsură responsabil pentru evoluția tehnologică colosală a Internet-ului. (Trebuie spus că deși anumite decizii au rămas neschimbate de la început, o mulțime de reguli din funcționarea Internet-ului au fost introduse pe parcurs. Acesta este un fapt colosal: e ca și cum ai schimba bujiile la mașină în timp ce merge!) IETF își publică deciziile în niște documente numite “Request For Comments (RFC)” (Cerere de comentarii), și care sunt adoptate prin consensul celor implicați în funcționarea Internet-ului. RFC-urile specifică *totalitatea* regulilor pe care trebuie să le respecte calculatoarele care comunică în Internet (reguli care se numesc *protocoale*). Lista completă a RFC-urilor se poate obține...de pe Internet (<http://rs.internic.net>)². Dacă vreți să știți cum funcționează Internet-ul este suficient să citiți aceste documente. Dar nu e o treaba prea simplă... (sunt câteva sute!).

2 Acest articol

Despre rețele și Internet s-a scris enorm. O carte absolut excelentă (și foarte fierbinte) a publicat în traducere în româna editura Agora, a lui Andrew S. Tanenbaum (la un preț ridicol: ediția

¹Cele mai multe cunoștințe despre Internet le-am căpătat lucrând ca administrator la acest nod în 1995; în continuare administrația nodului este făcută în mare parte de studenți ai Politehnicii.

²[internic.net](http://rs.internic.net) este rețeaua firmei “Network Solutions”, care este responsabilă de administrarea domeniilor rădăcină din Internet. Nu vom vorbi despre domenii în acest articol.

americană costă 80 de dolari).

Acest articol va trata deci foarte sumar o singură problemă legată de funcționarea Internet-ului: “rutarea” (alegerea unei rute între două calculatoare). Despre rutare numai au fost scrise mai multe cărți; referința cea mai bună este Radia Perlman “Interconnections”, Addison-Wesley, 1992. Subiectul este foarte generos, și deloc simplu; tratamentul pe care îl vom da aici este desigur doar orientativ.

În funcționarea unei rețele există două aspecte oarecum complementare: algoritmi și protocoalele care fac acea rețea să funcționeze, și aparatele, sîrmele și comenzile administratorilor care operează aparatele “reale”. Voi încerca în acest articol să le ating pe amîndouă.

Voi folosi adesea analogii cu o altă rețea reală care prezintă oarecare similitudini cu Internet-ul: poșta. La nivelul la care discutăm în acest articol Internet-ul are o singură misiune: îi dai niște date (grupate într-un *pachet*) și îi spui *unde* vrei să ajungă, și Internet-ul face tot ce poate să ducă datele acolo. La fel ca și poșta: pui la poștă un pachetel (sau un plic), scrii pe el adresa de destinație, și apoi sperii ca sistemul să-și facă datoria. Nici poșta și nici Internet-ul nu garantează că ce le-ai dat o să ajungă în partea cealaltă. Dacă vrei să fii sigur trebuie să-ți iei măsuri suplimentare de precauție³.

3 Protocoale

Un protocol este un set de reguli pe care toată lumea trebuie să le respecte. Cînd întîlnești un cunoscut îi întinzi mîna dreaptă, și el ți-o întinde pe a lui; asta e o regulă dintr-un protocol. Protocoalele fac să ne înțelegem unii cu alții.

Pentru ca două calculatoare să poată comunica unul cu celălalt amîndouă trebuie să respecte aceleași reguli. Ingineria programării ne învață că pentru a construi sisteme complexe trebuie să le spargem în piese mai simple, pe care apoi le putem construi independent. Din cauza asta, o rețea de calculatoare se specifică mai multe protocoale independente pe care calculatoarele trebuie să le respecte. Pentru fiecare sub-problemă pe care vrem s-o rezolvăm avem un “mic” protocol.

De pildă avem un protocol prin care asociem unei anumite diferențe de potențial de pe sîrmă valoarea 0 și alteia 1; protocolul acesta probabil specifică care sîrmă ce valoare cară, cît timp trebuie să dureze un semnal pentru “1”, care este limita între 0 și 1, etc.

Din cauza interferențelor electrice semnalele propagate printr-o sîrmă pot fi stricate. Un *alt* protocol specifică cum anume codificăm un mesaj în așa fel încît atunci cînd o eroare se strecoară în el avem șanse s-o detectăm.

Și tot așa, pentru fiecare problemă avem un alt protocol. Protocoalele astea se folosesc împreună; de exemplu cele două de mai sus vor funcționa în ambele capete ale unei transmisiuni. Protocolul de detecție a erorilor va funcționa “deasupra” celuiilalt, pentru că după ce transmitem datele le verificăm corectitudinea. Protocolul pentru erori folosește rezultatele oferite de protocolul pentru transmisiune: datele transmise.

De aceea în rețele avem de-a face cu o “stivă” de protocoale. De fapt Internet-ul folosește mai mult de o duzină de protocoale în mod curent, care fiecare rezolvă o altă problemă.

Protocoalele bine proiectate pot fi văzute ca niște piese de Lego, care se îmbină unele cu altele. Adesea putem avea mai multe protocoale care rezolvă aceeași problemă, și putem alege pe

³De fapt unul dintre protocoalele din Internet, TCP, se ocupă chiar de asta: retransmite copii ale pachetelor care par să se fi pierdut. În articolul de față însă nu vom ajunge “pînă” la TCP.

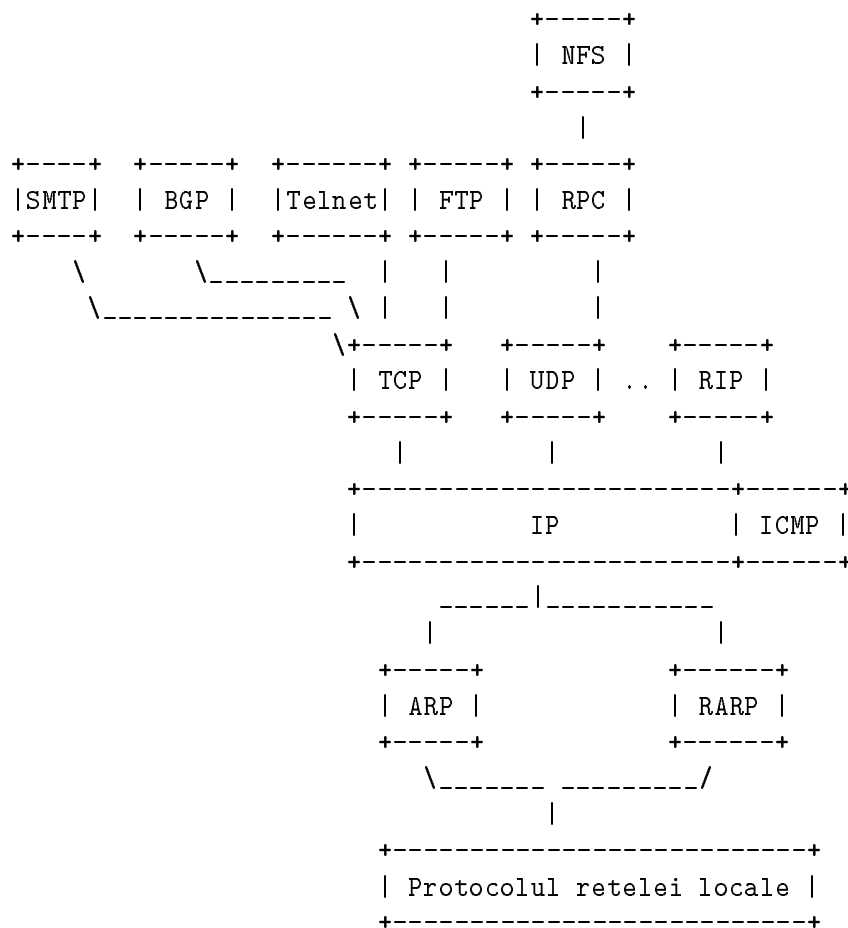
oricare dintre ele. Un protocol bine proiectat nu trebuie să depindă de celelalte împreună cu care funcționează.

Iată un exemplu din sistemul poștal: dacă vrei să-i scrii unui amic compui o scrisoare și o pui la poștă. Avem (cel puțin) două protocoale aici: conversația dintre prieteni, și sistemul poștal care mișcă scrisorile. Dacă te grăbești trimiți scrisoarea cu altă companie, de pildă DHL. Ai schimbat protocolul prin care informația se plimbă, dar conversația cu amicul este aceeași: conținutul scrisorii nu s-a schimbat deloc.

Internetul este caracterizat de folosirea anumitor protocoale. Cele mai importante protocoale folosite de calculatoarele din Internet se numesc IP (Internet Protocol) și TCP (Transport Control Protocol). TCP a fost proiectat înainte de a exista foarte multă experiență despre rețele, așa că violează regula expusă anterior: TCP funcționează numai cu IP.

Protocolul IP este cel care mută pachetele dintr-un loc într-altul, posibil peste multe conexiuni, iar TCP asigură fiabilitatea comunicației.

Iată schema unor protocoale esențiale din Internet:



4 Alte rețele

Nu trebuie să rămînem cu impresia că Internet-ul este singura rețea pentru transmisiunea de date din lume. Poate nici măcar cea mai mare! O rețea enormă dedicată transmisiunii de date este

rețeaua telefonică, care există de mai bine de 100 de ani, care folosește propriile protocoale.

Pînă în anii '90 nu era deloc clar că suita de protocoale folosite în Internet (TCP/IP) va domina lumea. Marile firme de calculatoare dezvoltaseră protocoalele proprii, care nu erau publice (structura lor era secret comercial, deci nu puteau fi implementate de oricine; de fapt asta a și dus la pieirea lor). Cele mai faimoase suite de protocoale aparțineau firmei Digital (protocoalele DECnet) și IBM (SNA: System Network Architecture). Și la ora actuală rețelele automatelor bancare din întreaga lume folosesc în mod preponderent SNA. Multe din protocoalele din Internet se inspiră puternic din aceste protocoale de firmă.

Pe deasupra mai exista o suită de protocoale (numite OSI: Open System Interconnection) standardizate de organizația mondială a standardelor; suita de protocoale X: X25, X400, etc. Multe rețele europene folosesc în continuare aceste protocoale, mai ales rețelele bancare.

O altă suită de protocoale a fost dezvoltată de marile companii de telecomunicații, dar nu a fost încă implementată pe scară largă: protocoalele numite generic ATM: Asynchronous Transfer Mode. ATM promitea anumite calități care structural scapă Internet-ului, dar nu este prea clar dacă va avea sau nu succes.

Deși Internet-ul are o mulțime de hibe, este extrem de bine testat: funcționează încontinuu de aproape 30 de ani. Lumea a învățat o grămadă de lecții despre proiectarea rețelilor cu Internet-ul, care au fost înglobate în structura lui. Dificultatea constă în faptul că nimeni nu poate de fapt înțelege funcționarea unui sistem atît de complex, cu atît mai puțin prezice. Chiar dacă alte protocoale sunt proiectate, nu este deloc limpede că vor funcționa la fel de bine în practică. De altfel la începutul anilor '90 se preconiza ca Internet-ul să fie desființat și înlocuit cu o rețea care folosește protocoale standardizate internațional. Acest lucru nu s-a întîmplat, și a devenit clar că nu se va mai întîmpla niciodată.

5 Internet-ul: o rețea de rețele

Observația cheie pe care trebuie să o facem ca să înțelegem cum lucrează Internet-ul este următoarea: Internet-ul *nu* leagă calculatoare, ci *rețele*. Distincția este crucială, după cum vom vedea.

O rețea este un număr de calculatoare care pot comunica între ele, fiecare cu fiecare, într-un oarecare mod. Cel mai răspîndit fel de rețea este Ethernet, care leagă mai multe calculatoare între ele folosind un cablu coaxial. Mai avem calculatoarele care vorbesc unele cu altele prin modemuri, folosind felurite protocoale din familia V (de exemplu modemurile de 33,6 kilobiți pe secundă folosesc V.34bis). Și așa mai departe.

Ceea ce face Internet-ul este să ia o colecție de astfel de rețele, poate foarte diferite arhitectural (primul exemplu istoric de mai sus avea o rețea prin telefon și una prin radio), și să pună *peste* mediile de comunicare ale acestor rețele o serie de reguli uniforme. Astfel toate calculatoarele din Internet vor avea nume de același fel și vor vorbi același set de protocoale.

În analogia noastră cu sistemul poștal lucrurile ar sta cam așa: să ne imaginăm că fiecare țară din lume are alte reguli și metode pentru a trimite obiecte între locuitori; în unele locuri obiectele se transportă cu camioane, în alte locuri cu avioane (de exemplu în Hawaii, probabil). Fiecare țară are propria ei rețea de distribuție. Dacă toate țările din lume vor să schimbe între ele scrisori (ceea ce se și întîmplă), atunci ele cad de acord asupra unui standard comun, independent de modul în care-și transportă obiectele. De pildă faptul că se folosesc timbre, și asupra modului în care se scrie o adresă pe plic. Dacă toată lumea face aceste lucruri la fel atunci rețelele poștale pot fi făcute să interopereze.

6 IP

Cel mai important protocol din Internet este protocolul numit IP: Internet Protocol. Scopul principal acestui protocol este să împingă pachetele de la un calculator sursă la o destinație din aproape în aproape, folosind legăturile existente între vecini. Dacă problema asta vi se pare simplă atunci încercați să o rezolvați dumneavoastră: eu vă dau un pachet și numele unui amic de-al meu; dumneavoastră aveți misiunea să faceți acest pachet să-i parvină, fără a avea cea mai mică idee unde stă amicul meu sau cum se ajunge la el. (IP are și un scop secundar, acela de a sparge pachetele în bucățele atunci când trebuie să traverseze o rețea ale cărei caracteristici nu permit tranzitul pachetelor mari. Dar asta e mai puțin interesant.) IP este descris în detaliu în standardul RFC 791.

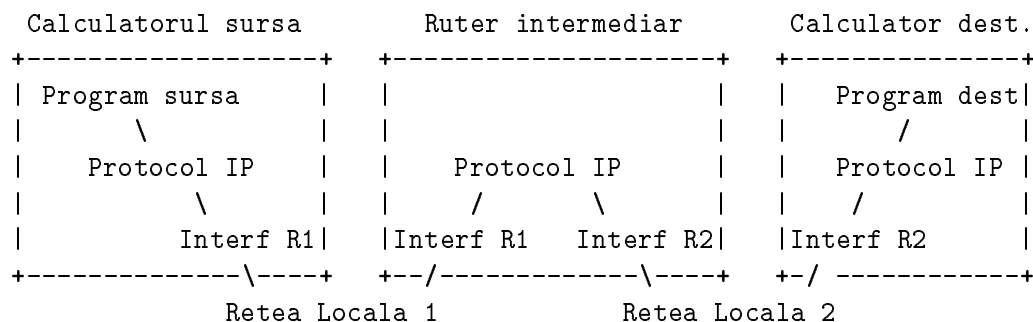
IP este un protocol care mută pachetele care i se dau. În interior rețeaua mai folosește un protocol special pentru control, care mută pachete generate de rețea însăși. Acest protocol se numește ICMP: Internet Control Message Protocol. Ca să ne facem o idee despre ICMP: dacă un calculator primește un pachet care se plimbă de mult prea multă vreme în rețea⁴ atunci calculatorul distruge pachetul și simultan trimite spre sursă un mesaj ICMP care anunță sursa de pierderea informației. Sursa va trebui să ia acțiuni corective.

Să vedem ce se dă și ce se cere lui IP.

7 Problema rutării

Fiecare calculator aparține unei rețele, cu ale cărei componente poate comunica oricând direct, fără alte intervenții. Vom vedea imediat cum află un calculator care sunt vecinii lui imediați; aceasta este o problemă de configurare care cade în sarcina administratorului de sistem.

Pe de altă parte anumite calculatoare au conexiuni la mai multe rețele simultan; acestea se mai numesc “gateways”: “porți” între rețele. Un pachet care este primit de un astfel de calculator pe *interfața* dinspre o rețea poate fi transmis pe cealaltă rețea. Acest proces se numește “forwarding” (“înaintare”). Drumul ales de un pachet de la sursă la destinație poartă denumirea de “rută” (route). Operația de alegerea a unei cărări pentru un pachet se numește “rutare” (routing); britanicii ar pronunța asta aproximativ “ruting”, pe când americanii ar zice “rauting”. Iată un desen care explică fenomenul:



Există mai multe feluri de “gateways”, dar noi în acest articol o să ne preocupăm cu preponderență de porțile care se numesc “rutere” (“router” în engleză). *Un ruter este deci în general un calculator*

⁴Fiecare pachet are un contor care scade spre zero la fiecare pas făcut. Evenimentul descris se petrece atunci când contorul ajunge la 0.

conectat simultan la mai multe rețele.

Pentru că avem peste 30 de milioane de calculatoare este clar că nu se poate configura informația despre unde se află fiecare în toți membrii rețelei (de altfel această informație este foarte volatilă: în fiecare secundă probabil mii de calculatoare se conectează și se deconectează). Știm că calculatoarele sunt foarte bune la prelucrat date, așa că sarcina calculării rutelor le care lor în sarcină.

Inițial fiecare calculator are oarecare cunoștințe despre vecinătatea lui mai mult sau mai puțin imediată; aceste cunoștințe sunt introduse de administratorul de sistem.

Problema rutării se enunță astfel:

Atunci când primești (tu, un ruter) un pachet pentru o anumită destinație trebuie să decizi pe care dintre interfețele tale să-l trimiți.

Una dintre interfețe trebuie să fie întotdeauna mai aproape de destinație; dacă fiecare ruter care primește un pachet îl trimite apoi pe o interfață mai apropiată de destinația finală, atunci (prin inducție matematică) eventual pachetul va ajunge la destinație. Simplu, nu?

Problema rutării este (aproape) banală pentru calculatoarele care au o singură interfață. Este clar că oricare ar fi destinația, dacă ea nu este chiar calculatorul local⁵, atunci există o singură interfață pe care pachetele pot fi scoase.

Problema rutării este mai complicată pentru calculatoarele care au de-a face două alegeri. Dar înainte de a vedea cum este ea rezolvată, să vedem concret cum se specifică o adresă și o destinație.

8 Adrese

Protocoalele din Internet au fost proiectate cu o grijă deosebită pentru administrare descentralizată. Astfel proiectanții au construit supape de siguranță pentru evoluția protocoalelor. Dacă protocolul evoluează este imposibil să se facă *simultan* în întreaga lume “upgrade”. Atunci rețeaua poate funcționa cu părți din ea rulând protocoale mai avansate (care, desigur, trebuie să rămână compatibile (backwards compatible) cu versiunile anterioare).

Versiunea curentă a protocolului IP este 4. Practic toate calculatoarele din Internet rulează versiunea 4, numită și IPv4. Câteva calculatoare rulează versiunea experimentală următoare, încă nestandardizată, numită 6 (nimeni nu știe unde a dispărut versiunea 5). Articolul de față discută numai IPv4.

Ca să poți să indici pe cineva anume trebuie să ai un nume pentru el. Pentru un calculator, mai pregnant decât pentru un om, un obiect care nu are un nume nu există! Numele calculatoarelor din Internet sunt întregi pe 32 de biți (4 octeți), numite *adrese IP*. De fapt trebuie să fim preciși: fiecare *interfață* (la o rețea) a unui calculator are o altă adresă; un calculator cu 10 interfețe (un ruter) va avea simultan 10 adrese IP. Adresele IP se scriu convențional indicând în baza 10 valorile celor 4 octeți separate de semnul (.) punct. Iată o adresă IP: “1.2.3.4”. Tern, nu? (Un sistem sofisticat există care permite calculatoarelor să aibă nume mai “umane”, cum ar fi “apolo.cs.pub.ro”, dar asta este cu totul altă poveste, și este treaba unor protocoale aflate mult deasupra IP.)

Fiecare adresă IP trebuie să apară *o singură dată* în Internet. Pentru a asigura acest lucru adresele IP sunt alocate de o autoritate centrală în grupe unor sub-autorități. Fiecare sub-autoritate apoi alocă adresele la rîndul ei. De exemplu Politehnica din București a primit 65536

⁵Se întâmplă ca programe de pe același calculator să comunice între ele folosind protocoalele de rețea; în acest caz sursa și destinația pachetelor vor fi identice.

de adrese, care toate încep cu octeții 141.85 (deci adresele între 141.85.0.0 și 141.85.255.255). Politehnica apoi a împărțit adresele fiecărui departament sau laborator, etc. În felul acesta responsabilitatea poate fi relativ clar trasată.

Dacă un calculator într-o anumită parte a Internet-ului își alocă o adresă care nu i se cuvine în cel mai bun caz nu va funcționa, iar în cel mai rău caz va deranja funcționarea altor calculatoare; cel mai probabil este că administratorii rețelei la care este conectat calculatorul vor trebui să ia măsuri pentru a-l deconecta.

Există un program numit **whois** (“cine e”) care interoghează anumite baze de date care conțin informații despre administrația Internet-ului. Din păcate există mai multe astfel de baze de date, acoperind părți diferite ale rețelei. În principiu puteți afla informații despre un anumit calculator, sau administrator, sau altceva tastând **whois informatie-de-afla**.

Există deci teoretic 2^{32} adrese IP diferite, ceea ce înseamnă peste 4 miliarde. Aparent adresele ar trebui să ajungă, însă în zilele inițiale ale Internet-ului s-a cam făcut risipă; de pildă Politehnica din București nu a folosit toate cele 65000 de adrese, așa că multe sunt acum inutile, pentru că nu pot fi luate înapoi. Acest fenomen s-a petrecut în mai multe locuri.

Nu este obligatoriu ca adresele să fie administrate atât de grijuliu pe atât de multe planuri; există standarde de rețele în care adresele sunt astfel făcute încât intervenția administratorului este absolut inutilă; astfel de rețele se numesc “plug-and-play”, pentru motive evidente. Ethernet este o astfel de rețea: oricând iei două calculatoare care vorbesc Ethernet și le pui laolaltă și se vor înțelege *fără nici un fel de configurare*! Rețelele din IPv4 necesită administrare pentru că sunt prea puține; IPv6, care are adrese pe 16 octeți (128 biți!) va putea fi și el plug-and-play.

Atunci când un calculator vrea să trimită date spre un altul va pune datele într-un pachet (sau mai multe), și la începutului fiecărui pachet va pune un *antet* (header) care conține printre altele adresa IP a sursei și a destinației. Cea din urmă este folosită de rutere pentru a înainta pachetul.

9 Rețele și noduri

Pentru a simplifica problema rutării adresele în Internet nu sunt “plate”, ci *ierarhice*. Asta înseamnă că o adresă dă și o oarecare indicație despre *traseul* care trebuie urmat pînă la acea adresă.

Ca să înțelegem mai bine diferența, numele unui individ este dintr-un spațiu de nume plat: nu-ți spune absolut nimic despre locul în care este acel individ (trebuie să folosești ceva gen o carte de telefon).

Pe de altă parte o adresă ca cea pusă pe o scrisoare este ierarhică, pentru că îți indică cu o precizie din ce în ce mai mare (dacă citești de sus în jos) *unde* este destinația. O adresă de genul:

4108 Wean Hall
Carnegie Mellon University
4000 Forbes Avenue
Pittsburgh, PA 15217
USA

îți spune că locul este în USA, din USA apoi trebuie să alegi statul Pennsylvania (PA), orașul Pittsburgh, și așa mai departe. Din ce în ce mai precis. Aproape sigur mai există undeva o stradă numită “Forbes”, dar în orașul Pittsburgh există una singură (asta este treaba primăriei: să asigure această lipsă de ambiguitate). Astfel adresa este foarte precisă.

Adresele din Internet sunt tot ierarhice, avînd însă numai două componente. Prima componentă este numărul *rețelei*, iar a doua componentă este numărul *calculatorului din rețea*. Reamintiți-vă că Internet-ul este o colecție de rețele. Toate calculatoarele care fac parte din aceeași rețea vor avea deci aceeași valoare pentru numărul rețelei.

O adresă Internet trebuie deci să fie însoțită de o *mască*, al cărei folos este să ne indice care din cei 32 de biți indică rețeaua și care calculatorul din rețea. (Un calculator într-o rețea se numește pe engleză și “gazdă” — host.) Masca este întotdeauna un șir de biți “1” consecutivi plecînd de la cel mai important.

Iată un exemplu de o adresă (am generat-o aleator) însoțită de o mască, de data asta scrisă în baza 2:

```
00101101.01011011.11100011.00010110 --- adresa
11111111.11111111.11100000.00000000 --- masca
\-----/\-----/
  numarul rețelei      nr. host-ului

00101101.01011011.11100000.00000000 --- adresa rețelei = adresa & masca
00000000.00000000.00000011.00010110 --- host = adresa & ~masca
```

În baza 10 adresa de mai sus ar avea valoarea: 55.91.227.20. Adresa rețelei este 55.91.224.0, iar adresa calculatorului din cadrul rețelei este 0.0.3.20.

Numerotarea asta poate părea confuză, și în oarecare măsură și este.

Informația purtată de o adresă și o mască este foarte importantă: un calculator știe că dacă ajunge la un calculator dintr-o rețea (oricare), atunci ajunge la orice calculator din acea rețea. Cu alte cuvinte, toate calculatoarele care încep cu același număr de rețea (cu biții 55.91.224.0) sunt direct conectate unul la altul.

De asemenea, acest lucru simplifică enorm tabelele de rutare ale rutelor: un ruter nu trebuie să știe drumul de parcurs pentru fiecare calculator care începe cu 55.91.224.0: ajungă să știe unul singur, pentru că aceste calculatoare sunt toate la un loc.

În trecut adresele din Internet aveau masca implicită și se împărțeau în 3 clase mari, numite A, B și C. Această organizare este acum depășită, așa că nu o menționăm decît în trecere, pentru a nu spori confuzia.

10 Interfața cu sub-rețeaua: ARP, ifconfig

Administratorul de sistem configurează un calculator ca la pornire să-și inițializeze adresele (și măștile) pe toate interfețele. Adresa se poate inițializa și inspecta pe o stație Unix cu comanda `ifconfig` (InterFace CONFIGuration). Iată un exemplu de informație pentru o interfață așa cum este el tipărit de `ifconfig`:

```
ifconfig le0
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
      inet 132.236.102.10 netmask ffff0000 broadcast 132.236.0.0
```

Numele interfeței este `le0`, care vine de la Local Ethernet 0 (nu există nici un fel de standardizare pentru aceste nume; pe anumite sisteme Unix puteți afla toate interfețele cu comanda `ifconfig -a`). Mai știu că adresa IP este 132.236.102.10, și că masca (tipărită în baza 16) este

255.255.0.0. Cu alte cuvinte adresa rețelei este formată din primii doi octeți (132.236.0.0), iar adresa calculatorului din această rețea este 0.0.102.10.

Prin convenție într-o rețea nu poate exista nici un calculator cu o adresa numai din cifre 0 sau numai din cifre 1. Aceste adrese sunt rezervate pentru alte scopuri, pe care le vom ignora aici.

Calculatorul meu chiar dacă știe numai acest lucru știe un lucru foarte important: că toate celelalte calculatoare care au adresele de forma 132.236.xxx.xxx sunt accesibile direct prin interfață 1e0. Cu alte cuvinte, dacă dau ca destinație un astfel de calculator, nu este nevoie de nici un fel de “forwarding”: pachetul poate fi pus direct pe interfața 1e0 și el va ajunge la destinație imediat!

Cum se poate așa ceva? O mare parte din rețelele locale au o proprietate foarte interesantă: comunicația în ele se face prin “difuzare”: cînd cineva pune un pachet în rețea absolut toată lumea din rețea vede acel pachet. De altfel este ușor de înțeles: Ethernet-ul seamănă tehnologic cu televiziunea prin cablu, în care un transmițător pune un semnal pe o sîrmă și toți îl aud. Toți se uită la pachet, dar numai cel căruia îi este destinat îl copiază în memoria proprie. (Destinatarul se auto-identifică comparînd adresa destinației din antetul pachetului cu propria lui destinație.)

Era prea frumos ca să fie adevărat: lucrurile sunt în realitate ceva mai complicate (deși conceptual asta e ideea). Întotdeauna rețeaua locală are la rîndul ei propriile mecanisme de identificare și numire a calculatoarelor. Deși este adevărat că fiecare calculator de pe un cablu Ethernet⁶ le poate auzi pe toate, comunicația se face de obicei tot între perechi de calculatoare astfel: fiecare calculator are o adresă Ethernet, care este unică pe glob, pentru că fiecare fabricant de plăci Ethernet a promis asta unei autorități centrale. Cînd un calculator pune un pachet pe un cablu, indică și adresa Ethernet a destinației.

O interfață plasată pe o rețea Ethernet are deci (pe lîngă adresa IP) și o adresă Ethernet. Cînd calculatorul vrea să comunice cu altul din același Ethernet, trebuie să indice de fapt adresa Ethernet a destinației, pentru că rețeaua Ethernet habar-n-are de adrese IP. Pentru asta fiecare calculator menține o tabelă de corespondențe: pentru fiecare adresă IP din rețeaua locală știe ce adresă Ethernet corespunde. Tabela asta se construiește cu ajutorul unui alt protocol, numit ARP (Address Resolution Protocol).

- Pentru cazul unor rețele ca Ethernet-ul se folosește proprietatea de “broadcast” a rețelei: cînd vrei să-i trimiți ceva lui 1.2.3.4 “strigi” pe Ethernet cu ajutorul unui pachet special: “Băi, care ai adresa IP 1.2.3.4?”. Cel în seamă vede pachetul tău Ethernet, care conține adresa ta Ethernet, așa că poate să-ți răspundă cu un alt pachet, din care afli corespondența între adresele sale IP și Ethernet.
- Pentru rețele în care nu se poate face broadcast administratorii configurează un server special care știe toate traducerile;

În Unix puteți inspecta tabela ARP cu comanda `arp`.

11 Tabelele de rutare

Ce ne facem însă atunci cînd vrem să comunicăm cu un calculator care nu este conectat în aceeași rețea locală cu noi? Pachetul nostru va trebui să treacă prin mai multe gateway-uri pînă la destinație. Problema care se pune pentru calculatorul meu este: “care este primul gateway”?

Informația de acest gen este menținută de nivelul IP într-o *tabelă de rutare*. Această tabelă descrie pentru fiecare grup de adrese interfața care trebuie folosită pentru a ajunge la ele și primul calculator căruia datele trebuie să-i fie trimise.

Cum arată tabela de rutare și de unde vin informațiile din ea?

Răspunsul la această întrebare se poate obține în trei feluri diferite, depinzînd de importanța calculatorului care pune întrebarea. Putem distinge trei feluri de calculatoare conectate în Internet, după cum ilustrează figura 1.

⁶Puteți substitui în loc de Ethernet orice alt tip de rețea locală; lucrurile vor sta la fel.

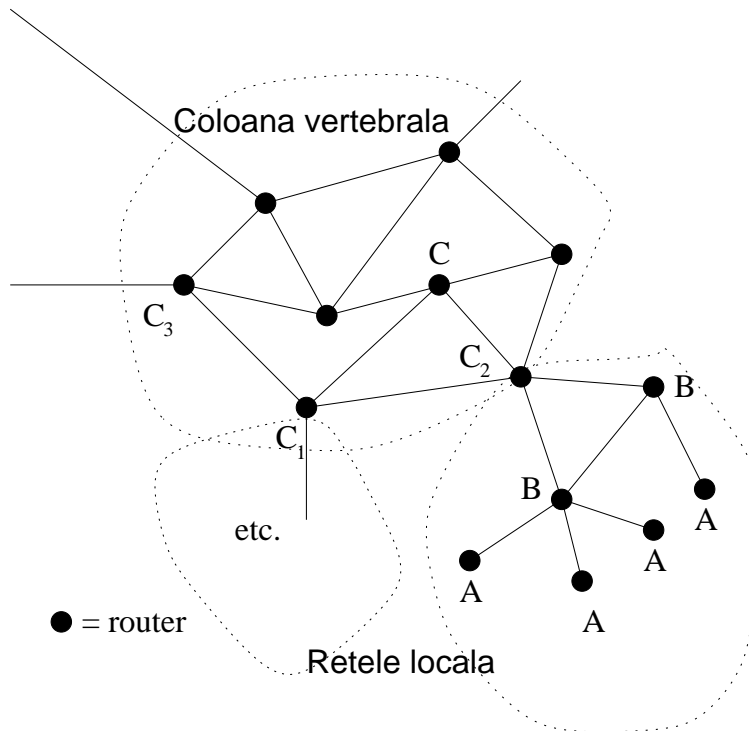


Figura 1: Fragment din Internet

Frunze: orice calculator care are o singură interfață este o “frunză” (termen împrumutat din teoria grafurilor) (A în figură) (Atenție: literele A, B, C nu sunt nicidecum denumiri standard; le-am inventat eu pentru a ilustra acest articol; de exemplu nu trebuie nicidecum confundate cu rețelele de tip A, B, C, care sunt altceva).

Rutere “mici”: un astfel de calculator separă o rețea relativ mică de restul Internet-ului; toate pachetele care trec dintr-o parte într-alta trebuie să treacă prin el (B în figură).

Rutere din backbone: “miezul” Internet-ului arată ca o mare pânză de păianjen; aceasta este “coloana vertebrală” a Internet-ului, formată din trunchiuri de mare viteză care se încrucișează în rutere. Pentru un astfel de ruter nu se poate spune că “taie” Internet-ul în două părți (C în figură).

Modul în care funcționează cele trei feluri de calculatoare din punct de vedere al rutării este complet diferit.

12 Rutele implicite (default)

Frunzele trebuie să știe care este ruter-ul “mic” (terminologia nu este standard, dar nu am o idee mai bună) care leagă propria lor rețea de Internet. În figura anterioară, calculatorul A trebuie să știe că rețeaua sa locală are ca legătură pe B. Acest lucru este de obicei configurat de administratorul rețelei. De ce?

Pentru că atunci când A are de trimis un pachet la o adresă algoritmul lui va fi următorul:

- Dacă pachetul este pentru rețeaua locală, atunci pachetul este transmis pe interfață proprie indicînd destinația dorită;
- Altfel pachetul este înaintat calculatorului B cel mai apropiat.

De ce? Pentru că nu există alt drum spre lumea dinafară decît prin B. B este ceea ce se numește “ruter implicit” pentru A (default router); majoritatea pachetelor se vor duce la el.

12.1 route și netstat

Dacă frunza rulează sistemul de operare Unix, atunci comanda cu care se adaugă și șterg rute în tabela de rutare se numește **route**. De obicei calculatorul execută un script de inițializare la bootare prin care instalează ruterul implicit. Puteți vedea toate rutele instalate ale unui calculator cu comanda **netstat -r**, de la NETwork STATistics, Routes. Ca în desenul nostru, uneori frunzele pot accesa simultan mai multe porți diferite spre Internet; de obicei una dintre ele este declarată inițial poartă implicită.

Iată de exemplu rezultatul acestei comenzi pe un calculator frunză:

```
# netstat -r
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
128.2.0.0        *               255.255.0.0    U        1500 0          0 eth0
127.0.0.0        *               255.0.0.0      U        3584 0          0 lo
default          GW.CS.CMU.EDU  0.0.0.0        UG       1500 0          0 eth0
```

De aici află că:

- Interfața **eth0** este conectată direct (adică nu prin intermediul unui gateway: *) la rețeaua cu numărul 128.2.0.0 (masca 255.255.0.0). U înseamnă “up”, adică “funcționează”;
- Interfața **lo** este conectată direct la rețeaua 127.0.0.0 (aceasta este o interfață fictivă prezentă în toate calculatoarele; adresele care încep cu 127 sunt rezervate pentru utilizări speciale. De exemplu adresa 127.0.0.1 reprezintă întotdeauna calculatorul local — este echivalentul pronumelui “eu”);
- În fine, interfața **eth0** ne leagă la o rețea pe care se află ruterul implicit, care este **gw.cs.cmu.edu**. Masca 0.0.0.0 ne arată că acesta este un ruter implicit, deci orice pachet pentru o adresă care nu se potrivește cu celelalte trebuie trimisă lui.

13 Ruterele și coloana vertebrală (backbone)

Pentru celelalte tipuri de calculatoare situația este diferită. Un router “mic” poate fi încă configurat manual: el trebuie să știe care sunt toate frunzele pentru care el este “responsabil”. Un astfel de calculator va schimba permanent informații despre topologia rețelei cu vecinii săi. Protocolul prin care ruterele schimbă informații se numește *protocol de rutare*. La ora actuală sunt standardizate aproape o duzină de astfel de protocoale pentru Internet. În figura noastră B-urile vor schimba informații: fiecare îi va spune celuilalt cu ce calculatoare “frunză” este direct conectat, iar ambele îi vor raporta lui C acest lucru.

Caracteristica centrală a unui astfel de ruter este că trebuie să memoreze relativ puține informații în tabela de rutare (zeci sau sute de înregistrări), putându-se baza în continuare pe rute implicite, care trec prin cel mai apropiat ruter de tip C.

Astfel de rutere pot fi tot stații de lucru sau calculatoare personale rulând Unix, sau calculatoare specializate (cum vom vedea mai jos că sunt folosite întotdeauna pentru ruterele de tip C). Foarte adesea astfel de rutere rulează ceea ce se numesc “ziduri de foc” (firewalls). Acestea sunt programe speciale care inspectează fiecare pachet care trece prin ruter și interzic transferul unor pachete suspicioase. Aceasta previne o sumedenie de atacuri informatice prin Internet și restrânge și controlează fluxul de informații din/spre rețeaua locală.

14 Protocoale de rutare

Ruterele de tip B sau C nu pot fi configurate manual; cel mai important motiv este că o rețea mai mare suferă în mod frecvent schimbări, care sunt greu de centralizat și administrat. Din cauza asta astfel de rutere calculează singure topologia rețelei.

Ruterele de tip C sunt întotdeauna calculatoare special construite pentru acest scop; au sisteme de operare speciale, nu au discuri (folosesc memorii nevolatile flash pentru a ține sistemul de operare), pot suporta foarte multe plăci de rețea (zeci simultan) și nu fac tot timpul altceva decât să permită administrarea rețelei și calculul de tabele de rutare. Cam 80% din ruterele din Internet sunt fabricate de firma Cisco din Statele Unite. Un ruter de mare performanță poate transfera date de ordinul a 10Gbps (giga biți pe secundă)!

Sarcina unui ruter de tip C este de cu totul altă natură decât cea a rutereleor de tip A sau B. Spre deosebire de acelea, un ruter de tip C trebuie să știe *întreaga topologie a Internet-ului*, și nu numai rețelele care se învecinează direct cu el.

Ca să înțelegem de ce se întâmplă acest lucru să considerăm ruterele C_2 și C_3 din figura de mai sus. Ce se întâmplă dacă C_2 primește un pachet pentru un calculator din rețeaua conectată la C_3 ? Dacă fiecare C și-ar cunoaște numai vecinii, atunci C_2 n-ar avea de unde să știe că vecinul lui, C_1 este la rîndul lui vecinul lui C_3 . Din cauza asta fiecare ruter de tip C va spune tuturor vecinilor săi *toate* informațiile pe care le are.

Inițial, la boot-are, un ruter C va ști numai vecinii săi (introduși tot de administrator). După prima rundă de comunicații va afla însă și vecinii vecinilor săi, după care vecinii vecinilor vecinilor săi, și așa mai departe. Cum Internet-ul la ora actuală are un diametru în jur de 30 (adică distanța dintre oricare două calculatoare trece prin mai puțin de 30 de rutere), după 30 de runde de comunicație între vecini orice ruter cunoaște întreaga topologie a rețelei! Asta înseamnă zeci de megaocteți de informație. Dar fără această informație calculatoare îndepărtate n-ar avea nici o idee cum să ajungă unul la altul.

În realitate tabelele de rutare ar fi foarte mari, așa încît pentru întregul Internet se folosește în mod normal o adresare ierarhică pe 3 nivele. Nivelul cel mai de jos este calculatorul, apoi rețeaua (pe astea deja le-am văzut), iar deasupra vine *sistemul autonom* (Autonomous System, AS). În general un sistem autonom este o unitate administrativă. De exemplu întreaga rețea educațională a României este un sistem autonom. Informația este apoi compactată astfel în ruterele de tip C:

- Un ruter cunoaște topologia completă a întregului sistem autonom din care face parte;
- Pentru celelalte sisteme autonome un ruter știe numai două feluri de informații:
 - Ce rețele sunt în fiecare;
 - Ce șir de sisteme autonome trebuie traversat pentru a ajunge la un sistem autonom anume.

Ca să înțelegem de ce asta comprimă foarte tare informația, să ne gândim din nou la sistemul poștal, în care o țară ar fi un sistem autonom și un oraș ar fi o rețea. Să presupunem că toate scrisorile circulă terestru. Atunci informația completă de rutare între București și Londra ar trebui să fie lista tuturor orașelor care trebuie traversate pe drum. Pe de altă parte lista sistemelor autonome ar fi numai lista *țărilor* care trebuie traversate. Ideea este că, dacă destinația este înafara propriului sistem autonom, dai pachetul primei țări pe drum (il duci în capitală = ruter de margine al sistemului autonom), care se descurcă.

Ruterele vorbesc în Internet două feluri diferite de protocoale de rutare, schimbînd feluri diferite de informație. Anumite protocoale sunt vorbite *între* ruterele din același sistem autonom, și astfel de protocoale se numesc *protocoale interne*. Cele mai folosite sunt RIP (Routing Internet Protocol) și OSPF (Open Shortest Path First).

Ruterele vorbesc cu vecinii lor din alte sisteme autonome protocoale *externe*. Cele mai faimoase sunt EGP (External Gateway Protocol), care este acum treptat înlocuit cu BGP (Border Gateway Protocol).

15 Administrarea rețelei

Administrarea și configurarea unei rețele de calculatoare este o slujbă foarte interesantă, mai ales a unei rețele de mare întindere. Problema este că adesea operatorul se află la foarte mare distanță de echipamentele supravegheate, așa că administrația rețelei se face...folosind rețeaua. Administratorul are conturi pe ruterele din domeniile administrate și le configurează de la distanță (presupunînd că funcționează...).

Există un protocol special numit SNMP (Simple Network Management Protocol), dar care nu este deloc simplu, prin care ruterele comunică evenimente deosebite și prin care administratorul de rețea configurează parametrii de funcționare ai ruterelelor.

Dar cele mai folosite scule ale administratorului sunt două mici programe: **ping** și **traceroute**.

15.1 ping

ping vine de la Packet INternet Gropper (pachet care bijbăie prin Internet). Programul **ping** trimite la o destinație un pachet ICMP cu rugămintea destinației de a întoarce pachetul de îndată ce-l primește. Este metoda cea mai bună de a verifica dacă rutele între sursă și destinație sunt bine configurate (în ambele sensuri!), și de a măsura încărcarea rețelei (din timpul de întârziere între transmisie și recepție). Iată un exemplu de folosire a lui **ping**:

```
# ping pub.pub.ro
PING pub.pub.ro (141.85.128.1): 56 data bytes
64 bytes from 141.85.128.1: icmp_seq=0 ttl=238 time=722.7 ms
64 bytes from 141.85.128.1: icmp_seq=1 ttl=238 time=730.6 ms
64 bytes from 141.85.128.1: icmp_seq=2 ttl=238 time=825.5 ms
64 bytes from 141.85.128.1: icmp_seq=3 ttl=238 time=991.5 ms
```

Asta arată cum fiecare pachet se întoarce de la adresa 141.85.128.1 după un timp cuprins între 722 și 991 milisecunde.

15.2 traceroute

traceroute afișează întregul traseu al unui set de pachete spre o anumită destinație (fiecare pas făcut). Cu opțiunea **-n** adresele sunt afișate sub formă numerică (IP) numai. Iată un exemplu:

```
# traceroute pub.pub.ro
traceroute to pub.pub.ro (141.85.128.1), 30 hops max, 40 byte packets
 1 GW.CS.CMU.EDU (128.2.254.36)  0.757 ms  0.738 ms  1.029 ms
 2 RTRBONE.NET.CMU.EDU (128.2.1.2)  1.378 ms  1.271 ms  1.292 ms
 3 nss5.psc.net (192.88.114.254)  3.46 ms  4.278 ms  3.543 ms
 4 atm0-0-3.pit-bb1.cerf.net (134.24.137.21)  3.513 ms *  37.252 ms
 5 atm8-0.bos-bb1.cerf.net (134.24.46.93)  15.473 ms  16.077 ms  15.966 ms
 6 atm10-0.nyc-bb2.cerf.net (134.24.46.5)  20.593 ms  20.937 ms  20.022 ms
 7 atm0-0-0.sr1.nyc1.alter.net (134.24.32.50)  21.252 ms  23.044 ms  21.642 ms
 8 105.ATM2-0-0.XR2.NYC1.ALTER.NET (146.188.177.158)  21.338 ms  20.904 ms  20.39 ms
 9 194.ATM9-0-0.GW1.NYC5.ALTER.NET (146.188.177.229)  120.15 ms  23.331 ms  22.024 ms
10 421.ATM9-0-0.BR1.NYC5.Alter.Net (137.39.30.118)  21.395 ms  22.958 ms  20.692 ms
11 225.ATM1-0-0.CR1.AMS2.Alter.Net (146.188.5.10)  104.605 ms  108.978 ms  105.051 ms
12 213.Hssi1-0.CR1.STK2.Alter.Net (146.188.2.194)  127.12 ms  128.184 ms  128.107 ms
13 Fddi0-0.GW1.STK2.Alter.Net (146.188.31.99)  143.882 ms  141.473 ms  139.756 ms
14 Taide-gw.customer.ALTER.NET (146.188.48.10)  150.409 ms  144.739 ms  142.156 ms
15 SE-STO-MFS-0.taide.net (193.219.194.145)  145.877 ms  164.948 ms  144.165 ms
16 NO-NIT-TN-0.taide.net (193.219.193.93)  164.505 ms  162.902 ms  159.266 ms
17 RO-BUC-PUB-1.taide.net (193.219.193.170)  746.259 ms  720.339 ms  791.329 ms
18 waiting.pub.ro (141.85.130.16)  706.019 ms  696.28 ms  753.865 ms
19 pub.pub.ro (141.85.128.1)  730.402 ms  718.742 ms  725.471 ms
```

Se văd cei 19 pași pe care trebuie să-i facă un pachet plecând de la calculatorul de pe biroul meu pînă la un calculator (`pub.pub.ro`) din Politehnica din București. Numele calculatoarelor sunt afișate atît în formă textuală cît și în formă IP. Pentru fiecare destinație sunt trimise 3 pachete, și timpii de propagare dus-întors sunt afișați.

Putem afla multe informații interesante din acest tabel. Cea mai mare diferență de timp este între pașii 15 și 16: pînă la 15 pot ajunge în 150ms, dar pînă la 16 în 750ms. Asta pentru că legătura între 16 și 17 se face printr-un satelit geosincron, care zboară la o altitudine de cam 36,000km. Semnalului electromagnetic îi trebuie cam jumătate de secundă pentru a traversa distanța asta în sus și în jos.

Ce e amuzant este că ambele capete ale legăturii prin satelit se află în Europa. Prefixul “NO” din nume mă îndeamnă să ghicesc că legătura este între Norvegia și România. Zic “amuzant” pentru că a doua diferență ca mărime este între 10 și 11. Legătura asta este transcontinentală, între New York și Amsterdam, dar întîrzierea de propagare este de numai 80ms! Asta pentru că folosește un cablu submarin, probabil de fibră optică.

Putem afla și care sunt companiile ale căror rețele sunt tranzitate. În ordine pachetul trece prin:

- `NET.CMU.EDU`: rețeaua universității mele, Carnegie Mellon;
- `psc.net`: Pittsburgh Supercomputing Center;
- `cerf.net`;
- `ALTER.NET`: o rețea a firmei UUNET Technologies;
- `taide.net`: o rețea a țărilor baltice;

- pub.ro: rețeaua politehnicii din București.

Am aflat informații despre aceste rețele folosind `whois`, descrisă mai sus, interogând baza de date de la `internic.net`

```
# whois -h rs.internic.net taide.net
Upnet Baltic Taide Network AB (TAIDE-DOM)
  Radiskis Farm
  Vidiskiai, 4126
  LT

Domain Name: TAIDE.NET

Administrative Contact:
  Bjerkeland, Vidar (VB50) vidar.bjerkeland@ARCTICMAIL.COM
  +47 9002 74 66 (FAX) +47 9027 74 66
Technical Contact, Zone Contact:
  Vosylius, Saulius (SV104) saulius@TAIDE.LT
  +370-2-223541
Billing Contact:
  Bjerkeland, Vidar (VB50) vidar.bjerkeland@ARCTICMAIL.COM
  +47 9002 74 66 (FAX) +47 9027 74 66

Record last updated on 04-Mar-98.
Record created on 24-Aug-95.
Database last updated on 12-Apr-98 03:37:05 EDT.

Domain servers in listed order:

PARABOL.TAIDE.NET          193.219.197.130
MIDUS.TAIDE.NET           193.219.195.130
```

The InterNIC Registration Services Host contains ONLY non-military Domains and POC's. Other associated whois servers:

```
American Registry for Internet Numbers whois server - whois.arin.net
European IP Address Allocations whois server - whois.ripe.net
Asia Pacific IP Address Allocations whois server - whois.apnic.net
US Military whois server - whois.nic.mil
```

Putem afla și alte lucruri interesante din numele ruterelor tranzitate, cum ar fi tipul rețelei locale: unde apare "ATM" este vorba de o rețea cu protocoale ATM, unde apare "Hssi" e o rețea de tip "High Speed Serial Interface", unde apare "Fddi" e o rețea de tip "Fiber Distributed Data Interface", o rețea cu fibra optică. Ș.a.m.d.

16 Alte probleme

Voi încheia aici această discuție despre Internet și protocoalele de rutare. În clipa asta am o listă mai mare de lucruri pe care aș fi dorit să le spun decât de lucruri pe care am reușit să le povestesc.

O să menționez în treacăt câteva probleme fascinante ale rutării, care influențează enorm arhitectura și protocoalele rețelei, dar despre care va trebui poate să revin altădată.

16.1 Stabilitate

Discuțiile de mai sus presupun adesea că rețeaua funcționează impecabil. Ce ne facem când avem de-a face cu rutere defecte, sau mai rău, cu atacuri informatice.

De exemplu, în primele zile ale ARPAnet-ului, un ruter a avut o cădere de memorie, care a șters parțial tabelele de rutare, dar nu l-a scos complet din funcțiune. Ruterul a ajuns la concluzia că se află la distanță 0 de oricare alt ruter din Internet. Când a schimbat informații cu vecinii săi, acești s-au bucurat că au găsit o ruta mai rapidă pentru pachetele lor și au început să-i dea toate pachetele acestui ruter. Apoi vestea cea bună s-a dus mai departe. În câteva ore toate pachetele din Internet veneau direct la ruterul defect. Asta a produs o congestie enormă, care a oprit literalmente rețeaua.

Lección este: nu poți construi un sistem cu 30 de milioane de bucăți dacă nici una nu se poate strica. De atunci protocoalele de rutare au devenit din ce în ce mai robuste, fiind capabile să tolereze malfuncții și configurări greșite în părți ale rețelei. În plus ruterele sunt configurate să se autentifice reciproc (folosind parole) și să nu primească informații de rutare decât de la calculatoare de încredere. “Încredere” înseamnă practic sub administrația unei organizații care poate fi trasă la răspundere pentru erori, și eventual pedepsită cu deconectarea totală.

Un al doilea tip de stabilitate este următorul: dacă un ruter reboot-ează sau o nouă legătură este introdusă sau scoasă din rețea, după un timp finit ajung toate ruterele de acord asupra topologiei? Deși poate părea evident că răspunsul este “da”, lucrurile nu stau chiar așa; multe protocoale din Internet de fapt au mari probleme să încorporeze astfel de tranziții.

Un al treilea tip de problemă este cauzat de faptul că informațiile despre modificări se propagă în timp, așa că anumite părți ale rețelei funcționează cu idei despre o topologie care nu mai există! Se pot astfel dezvolta calamități cum ar fi bucle de rutare, în care un pachet se plimbă în cerc între două (sau mai multe rutere), pentru că fiecare din ele crede că celălalt este mai aproape de destinație. Astfel de pachete sunt distruse într-o “gaură neagră” după ce le expiră timpul de viață. Inutil de spus că astfel de probleme sunt adevărate coșmare pentru administratorii de rețea (care le pot crea atunci când introduc rute greșite), și sunt extrem de greu de depistat și depanat.

În general, rețeaua poate dezvolta tot felul de anomalii, ca în cazul în care protocoale de rutare au ales o rută între două orașe americane aflate la 100km unul de altul prin Japonia, pentru că un segment de rețea pe acolo era mai “ieftin”. La ora actuală o mare parte din cercetarea Internet-ului este de natură empirică: dai drumul la niște pachete și după aia te miri ce se întâmplă cu ele și încerci să explici. Să observăm că nu există nici o garanție că două pachete ale aceleiași transmisiuni vor merge pe același drum! Sistemul este mult prea mare.

16.2 Multicast

O problemă absolut fascinantă care a fost rezolvată cu succes este transmisiunea spre multiple destinații simultan. Anumite rutere, care vorbesc un protocol special, sunt grupate într-o “rețea

virtuală” numită MBONE (Multicast Bone), pe care ceea ce un calculator trimite primesc toate cele interesate. Am văzut astfel în direct pe monitorul meu aterizarea navei spațiale, transmisă în direct de la NASA. Un fel de televiziune prin cablu, dacă vreți, dar cu potențiali milioane de receptori oriunde în lume și a cărei topologie se schimbă permanent.

16.3 Rutare mobilă

O altă problemă colosală a fost rezolvată: cum poți comunica cu un calculator care umblă tot timpul? Presupunerea noastră despre faptul că toate calculatoarele cu adrese în aceeași rețea sunt la un loc nu mai sunt valabile. Scheme deosebit de ingenioase permit acum unui individ să călătorească oriunde în lume și să folosească permanent Internet-ul. Schema este analoagă telefoniei celulare: calculatorul mobil își schimbă tot timpul ruterul implicit, dar detaliile sunt mult mai complicate.