Universitatea Politehnica Bucuresti

Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei

NIVELUL RETEA IN INTERNET

Studenti: Hulia Suliman – 442A

Alexandru Nicolae Ion – 442A

Profesor Coordonatro: Stefan Stancescu

Cuprins:

Hulia Suliman

I. Introducere

In lume exista multe retele, cu echipamente si programe diverse. Perosanele conectate la o anumita retea doresc adesea sa comunice cu persoane racordate la alta. Acesta cerinta impune conectarea unor retele diferite, de multe ori incompatibile, ceea ce uneori se realizeaza utilizand masini numite **porti (gateways).** Aceasta realizeaza conectarea si asigura conversiile necesare, atat in termeni de hardware cat si de software. O colectie de retele interconectate este numita inter-reatea sau internet. Acesti termeni vor fi colositi in sens generic , spre deosebire de Internet-ul mondial (care este un internet scpecial), al carui nume va fi scris mereu cu majuscula.

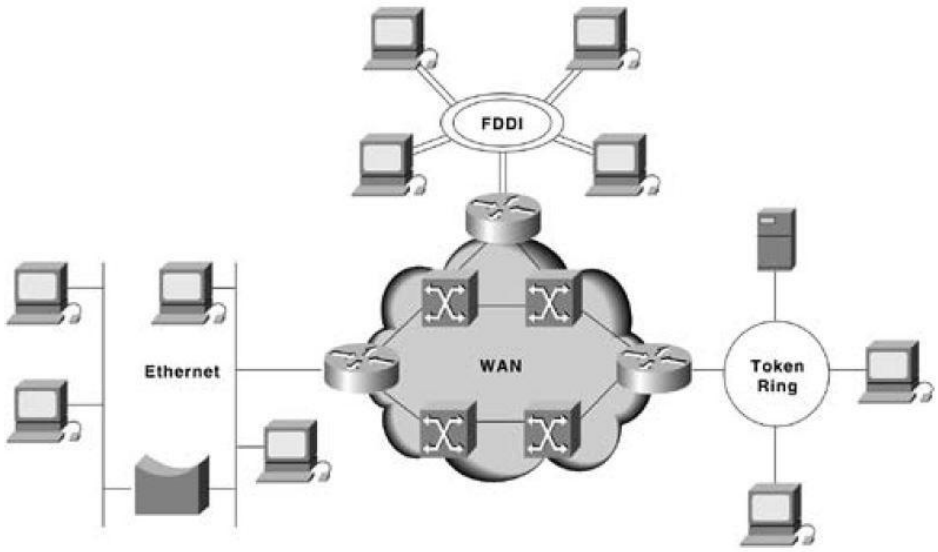
O forma comuna de inter-retea este o colectie de LAN-uri conectate printr-un WAN.

Deseori se produc confuzii intre subretele, retele si inter-retele. Termenul de subretea este mai potrivit in contextul unei retele larg raspandite geografic, unde se refera la colectia de rutere si linii de comunicatie aflate in proprietatea operatorului de retea. Ca o analogie, sistemul telefonic consta din centrele telefonice de comutare, care sunt conectate intre ele prin linii de mare viteza si sunt legate la locuinte si birouri prin linii de viteza scazuta. Aceste linii si echipamente, detinute si intretinute de catre compania telefonica , formeaza subreteaua sistemului telefonic. Telefoanele propriuzise(care corespund in aceasta analogie gazdelor) nu sunt o parte a subretelei. Combinatia dintre o subreatea si gazdele sale formeaza o retea. In cazul unui LAN, reteaua este formata din cablue si gazde. Aici nu exista cu adevara o subreatea.

O inter-retea se formeaza atunci cand se leafa intre ele retele diferite. Din punctul nostru de vedere, legarea unui LAN si a unui WAN sau legarea a doua LAN-uri formeaza o inter-retea, dar nu exista un consens asupra terminologiei din acest domeniu. O regula simpla este aceea ca daca diferite companii sunt platite sa construiasca diversa parti ale unei retele si fiecare trebuie sa isi intretina propria parte, avem o inter-retea mai degraba decat o retea.De asemenea, daca tehnologiile difera in diverse zone ale retelei (de exemplu: difuzare si punct-la - punct), probabil ca discutam nu despre una ci despre doua retele.

O inter-reatea este o colectie de retele de broadcast si rutare de pachete, conectate de catre switch-uri si rutere, care reprezinta dispozitive intermediare functionand ca o singura mare retea. Asadar, toti utilizatorii si dipozitivele pot comunica, indiferent de segmentul de retea de care apartin.

Different Network Technologies Can Be Connected to Create an Internetwork



A. Istoria Internetworking-ului

Primele retele erau de tipul “time-sharing” care foloseau “mainframes” si terminale atasate. Acest tip de medii au fost implementate de catre IMB –System Network Arhitecture (SNA) si Digital.

Local-area networks (LAN) au evoluat odata cu revolutia PC-urilor. LAN – urile ofera unor useri multipli dintr-o zona geografica relativ mica, de a schimba fisiere si mesaje, dar si de a oferi acces la dispozitive cum ar fi : servere sau imprimante.

Wide-area networks (WAN) interconecteaza LAN-uri din diverse zone geografice pentru a creea conetivitate. Cateva din tehnologiile folosite pentru interconectarea LAN-urilor sunt : T1, T3, ATM, ISDN, ADSL, Frame Relay si altele. Noi metode de a iterconecta LAN-urile apar in fiecare zi.

Astazi, LAN-uri cu viteze foarte mari, si inter-retele comutate sunt folosite la scara larga, datorita faptului ca ele opereaza la viteze foarte mari si suporta aplicatii ce necesita viteze mari, cum ar fi: multimedia sau video- conferinte.

Inter-retelele au evoluat ca o solutie la trei probleme cheie: LAN-urile izolate, resurse duplicate, si lipsa management-ului retelelor. LAN-uri izolate faceau ca comunicatia electronica intre diverse sectoare sau departamente sa fie imposibila. Duplicarea resurselor inseamna ca

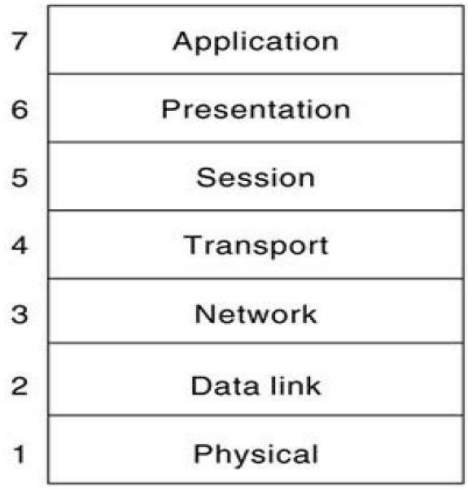
aceleasi resurse hardware sau software trebuies puse la dispozitie tuturor sectoarelor sau departamentelor, si desigur suport diferit pentru aceste.

Retelele permit calculatoarelor sa puna la dispozitie altor dispozitive (calculatoare,servere) informatie, aplicatii si chiar resurse sau dispozitive hardware. Pentru ca doua dispozitive (calculatoare) sa comunice intre ele, acestea trebuie sa urmeze un set de reguli numite protocoale. In perioada timpurie a retelelor, fiecare producator-vanzator era preocupat numai cu produsul sau. Fiecare producator isi definea propriul standard de comunicatie intre calculatoare. Acest lucru a dus la multe tipuri de standarde pentru hardware-ul si software-ul retelelor, care nu impartaseau in protocol comun. Asta inseamna, ca diferite sisteme nu puteau interactiona intre ele. Doua calculatoare care foloseau produse de retelistica produse de diversi producatori nu puteau fi conectate intre ele.

Spre sfarsitul anilor 1970, comunitatea ce opera in acest domeniu , s-a intrunit intr-un efort de a inlocui aceste sisteme inchise cu sisteme deschise. Ei doreau ca toate produsele de networking sa fie compatibile intre ele indiferent de producator. Organizatia Internationala de Standarde (International Standards Organisation ISO) a dezvoltat OSI (Open Systems Interconnection) , un model de referinta pentru retele. Acest model ofera informatii (specificatii) despre cum partile unei retele de comunicatii ar trebui sa lucreze impreuna.

Modelul de referinta OSI descrie modul in care informatia de la o aplicatie software, aflata intr-un calculator, circula prinr-o retea la o alta aplicatie software aflata pe un alt calculator. Modelul OSI este un model conceptual compus din sapte nivele , fiecare specificand in mod particular functiile retelei. Modelul a fost dezvoltate de catre ISO in 1948, si acum, este considerat principalul model arhitectural pentru comunicatia intre calculatoare. Modelul OSI divide sarcinile implicate cu transferul informatiei intre calculatoare in sapte sarcini mai mici. O sarcina sau un grup de sarcini este asignat fiecarui nivel din modelul OSI fiecare fiind implementata independent fara a afecta celelalte nivele.

The OSI Reference Model Contains Seven Independent Layers



B. Cum difera retelele si cum pot fi conectate

a) Cum difera retelele si probleme de implementare

Retelele pot sa difere in mai multe moduri. Unele dintre diferente, cum ar fi , tehnici de modulatie diferite sau formate de cadre, se gasesc in nivelele fizic(physical) si cel de date (data link). Urmatoarea lista de diferente apar in nivelul network. Sunt enumerate acele diferente care fac inter-retelele mult mai dificile de implementat si de operat fata de reletele independete.

Some of the many ways networks can differ.

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

Atunci cand pachetele trimise de catre un calculator sursa aflat intr-o retea, tranziteaza una sau mai multe retele straine inainte de a ajunge la destinatie (care, la randul ei poate fi diferita de rateaua sursa), pot aparea numeroase probleme la interfatele dintre retele. Cand pachetele dintr-o retea orientata pe conexiune (connection oriented) tranziteaza o retea connectionless, trebuiesc reordonate, lucru la care, expeditorul nu se asteapta si destinatarul nu este pregatit sa il trateze. Protocoale de conversie sunt adesea necesare, lucru ce poate fi dificil daca functionalitatea necesara nu poate fi exprimata. Conversiile de adrese vor fi, de asemenea, necesare , lucru ce presupune un fel de sistem director. Pasarea pachetelor multicast printr-o retea care nu suporta multicasting necesita generarea de pachete separate pentru fiecare destinatie.

Diferentele dintre marimea maxima a pachetelor admise pe diverse retele poate reprezenata o problema majora. Cum transmiti un pachet de 8000-biti printr-o retea , a carui marime maxima este 1500 biti? Calitatile diferite ale serviciilor sunt o problema cand, de exemplu, un pachet care are o constrangere de livrare in timp real este transmis printr-o retea care nu ofera nici o garantie de timp real.

Controlul erorilor,fluxului si congestiilor, adesea, difera de la retea la retea. Daca expeditorul si destinatarul pachetelor se asteapta ca toate pachetele sa fie expediate in secvente, fara erori, dar o retea intermediara tocmai a aruncat (discards) pachetele, lucru realizat de fiecare data cand apare o congestie, multe aplicatii vor crapa. De asemenea , daca pachetele pot “umbla” fara tinta pentru o perioada, apoi reapar dintr-o data si sunt transmise, problemele or sa apara daca acest comportament nu a fost anticipat si rezolvat. Diferite mecanisme de securitate, setarile parametrilor, contabilitatea regulilor si chiar legile natinale de intimitate (privacy) pot, de asemenea , sa cauzeze probleme.

Fiecare nivel are nevoie de un mecanism pentru a indentifica emitatorii si receptorii. Dat fiind ca o retea cuprinde in mod normal numeroase calculatoare, iar o parte dintre acestea detin mai multe procese, este necesara o modalitate prin care un proces de pe o anumita masina sa specifice cu cine doreste sa comunice. Ca o consecinta a destinatiilor multiple, pentru a specifica una dintre ele, este necesara o forma de adresare.

Un alt set de decizii de proiectare se refera la regulile pentru transferul de date. In unele sisteme datele circula intr-un singur sens, in altele pot circula in ambele sensuri. Protocolul trebuie, de asemenea , sa determine cator canale logice le corespunde conexiunea si care sunt prioritatile acestora. Multe retele dispun de cel putin doua canale logice pe conexiune, unul pentru date normal si unul pentru date urgente.

Controlul erorilor este o problema inportanta deoarece circuitele fizice de comunicatii nu sunt perfecte. Se cunosc multe coduri detecotare si corecotare de erori, dar ambele capete ale conexiunii trebuie sa se inteleaga asupra codului utilizat. In plus, receptorul trebuie sa aiba cum sa-i spuna emitatorului care mesaje au fost primite corect si care nu.

Nu toate canalele de comunicatii pastreaza ordinea mesajelor trimise. Pnetru a putea trata o eventuala perdere a secventialitatii, protocolul trebuie sa furnizeze explicit receptorului informatia necesara pentru a putea reconstrui mesajul. O solutie evidenta este numeroatarea fragmentelor, dar aceasta solutie inca nu rezolva problema fragmentelor care sosesc la receptor aparent fara legatura cu restul mesajului.

O problema ce intervine la fiecare nivel se refera la evitarea situatiei in care un emitator rapid trimite unui receptor lent date, la viteza prea mare. Au fost propuse diverse rezolvari ,undele dintre acestea presupun o anumita reactie, directa sau indirecta, prin care receptorul il informeaza pe emitator despre starea sa curenta. Altele limiteaza viteza de transmisie a emitatorului la o valoare stabilita de comun acord cu receptorul. Acest subiect se numeste controlul fluxului.

O alta problema care apare la cateva niveluri priveste incapacitatea tuturor proceselor de a accepta mesaje de lungime arbitrara. Acest fapt conduce la mecanisme pentru a dezasambla, a transmite si apoi a reasambla mesaje. O problema asemanatoare apare atunci cand procesele insista sa transmita datele in unitati atat de mici, incat transmiterea lor separata este ineficienta. In acesta situatie, solutia ese sa se asambleze impreuna mai multe mesaje mici destinate aceluiasi receptor si sa se dezasambleze la destiantie mesajul mare obtinut astfel.

Atunci cand este neconvenabil sau prea costisitor sa se aloce conexiuni separate pentru fiecare pereche de procese comincante, nicelul implicat in comunicare poate hotara sa utilizeze aceeasi conexiune pentru mai multe conversatii independente. Atata timpe cat acesta multiplexare si demultiplexare se realizeaza transparent, ea poate fu utilizata de catre orice nivel. Multiplexare este necasara, de exemplu, in nivelul fizic, unde traficul pentru toate coneiunuile trebuie sa fie transmis prin cel mult cateva circuite fizice.

Atunci cand exista mai multe cai intre sursa si destinatie, trebuie ales un anumit drum. Uneori acesta decizie trebuie impartita pe doua sau mai multe niveluiri. De exemplu, este posibil ca trimiterea unor date de la Londra la Roma sa necesite atat o decizie de nivel inalt pentru alegerea ca tara de tranzit a Frantei sau a Germaniei - in functie de legile lor de protejare a secretului datelor - cat si o decizie de nivel sacazut pentru alegerea unuia din multele trasee posibile, pe baza traficului curemt. Acest subiect poarte numele de dirijare sau rutare(routing).

b) Cum pot fi interconectate retelele

Retelele pot fi interconectate de catre diverse dispozitive.

In **nivelul fizic** (**physical layer**), retelele pot fi conectate intre ele cu ajutoarul repetoarelor sau ale hub-urilor, care doar muta bitii dintr-o retea intr-o retea identica. Aceste, sunt de regula, dispozitive analogice care nu au cunostiinte despre protocoalele digitale, ele doar regenerand semnal.

Un nivel mai sus **gasim nivelul legaturilor de date** (**data link**), care are la dispozitive de tipul punti (**bridges**) si comutatoare (**switches**), care lucreaza la nivel de date. Aceste dispozitive pot accepta cadre (frames), sa examineze adrese MAC (**Media Access Control address**) si sa trimita mai departe cadrele unei alte retele, in timp ce face translatii minore de protocol , de exemplu, de la Ethernet la FDDI sau la 802.11.

In **nivelul retea** (**network layer**), avem dispozitive de rutare (**routers**) care conecteaza doua retele. Daca doua retelele au nivelele de reatea diferite, router-ul poate fi capabil de a face o traducere intre cele doua formate de pachete, chiar daca, acum traducerea de pachete este foarte rar intalnita. Un router care poate procesa multiple protocoale este numit **multiprotocol router**.

In **nivelul transport** (**transport layer)** gasim portile de transport (**transport gateways**), care pot fi o interfata intre cele doua conexiuni de transport. De exemplu, o poarta de transport poate permite pechetelor sa circule intre o retea TCP si o retea SNA, ficare avand un protocol de transport diferit, lipind o conexiune TCP la o conexiune SNA.

In sfarsit, in **nivelul aplicatie** (**application layer**) , portile aplicatiei traduc mesajul semantic. Ca un exemplu, portile dintre un Internet e-mail (**RFC 822**) si X.400 e-mail trebuie sa analizeze (**parse**) mesajele e-mail si sa schimbe diverse campuri din header.

II. Programele de retea (retele de calculatoare)

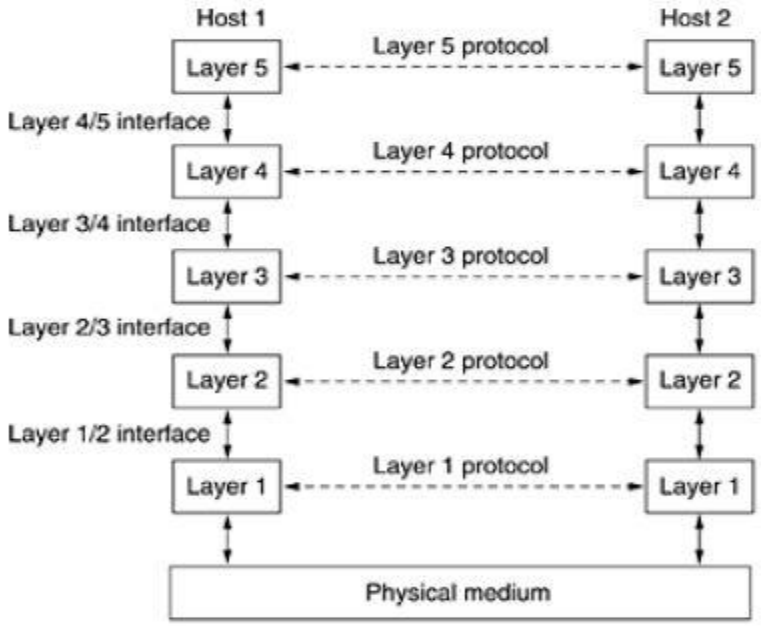
In proiectarea primelor retele de calculatoare, s-a acordat atentie in primul rand echipamentelor, iar programele au fost gandite ulterior. Acesta strategie nu mai este valabila. Programele de retea sunt acum fosrte structurate. In sectiunule urmatoare vom examina unele detalii ale tehnicii de structurare a programelor.

a) Ierarhiile de protocoale

Pentru a reduce din complexitatea proiectarii, majoritatea retelelor sunt organizate sub forma unei serii de **straturi** sau **niveluri**, fiecare din ele construit peste cel de dedesupt. Numarul de niveluri, numele fiecarui nivel, continutul si functia sa variaza de la retea la retea. Oricum, in toate reletele, scopul fiecarui nivel este sa ofere anumite servicii nivelurilor superioare, protejandu-le totoadata de detaliile privitoare la implementarea efectiva a serviciilor oferite. Intr-un anumit sens, fiecare nivel este un fel de masina virtual, oferind anumite servicii nivelului de deasupra lui.

Nivelul *n* de pe o masina converseaza cu nivelul *n* de pe alta masina. Regulile si conventiile utilizare in conversatie sunt cunoscute sub numele de **protocolul** nivelului *n* .In primcipal, un protocol reprezinta o intelegere intre partile care comunica, asupra modului de realizare a comunicarii. Incalcarea protocolului va face comunicarea mai dificila, daca nu chiar imposibila.

In figura urmatoare este ilustrata o retea cu cinci niveluri. Entitatile din niveluri corespondente de pe masini diferite se numesc egale. Entitatile egale pot fi procese, dispozitive hardware, sau chiarfiinte umane. Cu alte cuvinte, entitatiile egale sunt cele care comunica folosind protocolul.

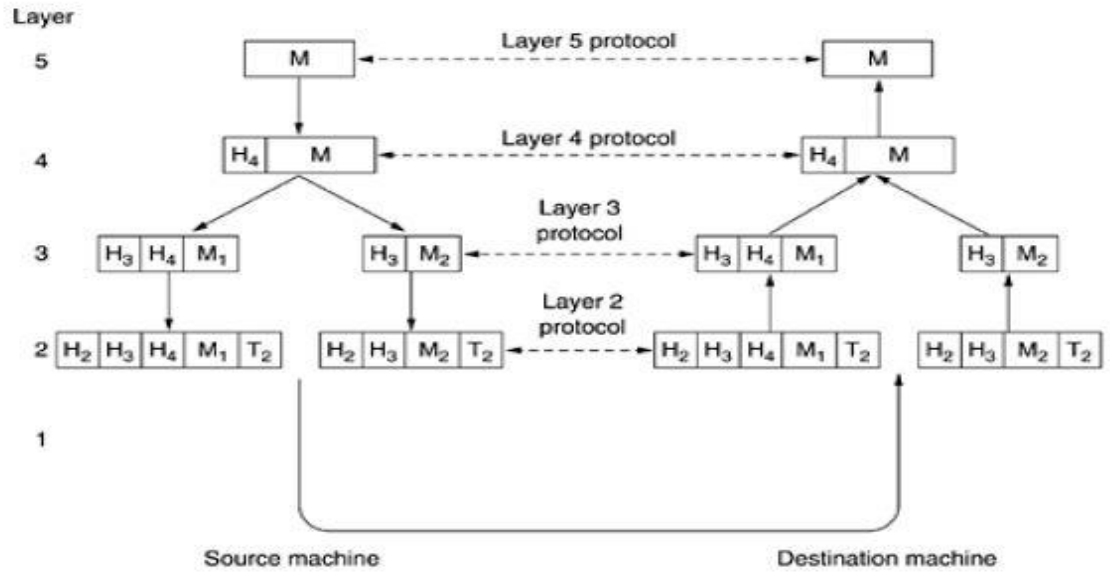


In realitate, nici un fel de date nu sunt transferate direct de pe nivelul *n* al unei masini pe nivelul *n* al altei masini. Fiecare nivel transfera datele si informatiile de control nivelului imediat inferior, pana cand se ajunge la nivelul cel mai de jos. Sub nivelul 1 se afla mediul **fizic** prin care se produce comunicarea efectiva. In figura de mai sus, comunicarea virtuala este reprezentata prin linii punctate, iar comunicarea fizica prin linii continue. Intre cele doua niveluri adiacente exista o **interfata**. Interfata defineste ce operatii si servitii primitive ofera nivelul de jos catre nivelul de sus. Cand proiectantii de retea decid cate niveluri sa inclunda intr-o retea si ce are de facut fiecare dintre ele, unul dintre considerentele cele mai importante se refera la definirea de interfete clare intre niveluri.

Aceasta presupune ca, la randul sau, fiecare nivel sa execute o colectie specifica de functii clar definite. Pe langa minimizarea volumului de informatii care trebuie transferate intre niveluri, interfetele clare permit totodata o mai simpla inlocuire a implementarii unui nivel cu o implementare complet diferita. Asa ceva este posibil, pentru ca tot ceea ce I se cere noii implementari este sa furnizeze nivelului superior exact setul de servicii pe care il oferea vechea impelementare. De altfel, este un fapt obisnuit ca doua gazde sa foloseasca implementari diferite.

O multime de niveluri si protocoale este numita **arhitectura de retea**. Specificatia unei arhitecturi trebuie sa contina destule informatii pentru a permite unui proiectant sa scrie programele sau sa contruiasca echipamentele necesare fiecarui nivel, astfel incat nivelurile sa indeplineaswca corect protocoalele corespunzatoare. Nici detaliile de implementare si nici

specificatiile interfeteleor nu fac parte din arhitectura, deoarece acestea sunta scunse in interiorul masinilor si nu sunt vizibile din afara. Nu este necesar nici macar ca interfetele de pe masiniile dintr-o retea sa fie aceleasi – cu conditia, insa, ca fiecare masina sa poata utiliza corect toate protocoalele. O lista de protocolae utilizate de catre un numit sistem, cate un protocol pentru fiecare nivel, se numeste **stiva de protocoale**.



Sa consideram un exemplu tehnic: cum se realizeaza comunicarea la ultimul nivel din reteaua cu cinci niveluri din ficura de mai sus. O aplicatie care se executa in nivelul 5 produce un mesaj M si il furnizeaza nivelului 4 pentru a-l transmite. Nivelul 4 insereaza un antet in fata mesajului, pentru a identifica respectivul mesaj si paseaza rezultatul nivelului 3. Antetul include informatii de control, de exemplu numere de ordine care ajuta nivelul 4 de pe masina destinatie sa livreze mesajele in ordinea conrecta in cazul in care nivelurile inferiaore nu pastreaza acesta ordine. Pe unele niveluri, antelele contin de asemenea campuri de control pentru marime,timp si alte informatii.

In numeroase retele nu exista nici o limita cu privire la marimea mesajelor transmise in protocolul nivelului 4, dar exista aproape intotdeauna o limita impusa de protocolul nivelului 3. In consecinta, nivelul 3 trebuie sa sparga mesajele primite in unitati mai mici, pachete, asezand fiecarui pachet un antet specific nivelului 3. In aces exemplu, M este descompus in doua parti, M_1 si M_2.

Nivelul 3 decide ce linie de transmisie sa utilizeze si transimte pachetele nivelului 2. Nivelul 2 adauga nu numai cate un antet pentru fiecare bucata, ci si o incheiere, dupa care furnizeaza unitatea rezulatanta nivelului 1 pentru a o transmite fizic. In masina receptoare mesajul este transmis in sus, din nivel in nivel, pe parcurs fiind eliminate succesiv toate antetele. Nici un antet corespunzator nivelelor de sub *n* nu este transmis in sus nivelului *n.*

Ceea ce este important de inteles este relatia dintre comunicatia virtual sic ea efectiva si diferenta intre protocoale si interfete. De exemplu, procesele egale de la nivelul 4 isi imagineaza conceptual comunicarea ca relizandu-se pe “orizontala”, utilizand protocolul nivelului 4. Desi fiecare dintre ele are, probabil, o procedura de genul *TrimiteInCealaltaPartei* si o alta *PrimesteDinCealaltaParte*, aceste procedure nu comunica de fapt cu cealata parte, ci cu nivelurile inferioare prin interfata ¾.

Abstractizarea proceselor pereche este crucial pentru proiectarea intregii retele. Cu ajutoarul ei, acesta sarcina practice imposibila poate fi descompusa in problem de proiectare mai mici, rezolvabile, si anume proiectarea nivelurilor individuale.

b) Servicii orientate pe conexiuni si servicii fara conexiuni

Nivelurile pot oferi nivelurilor de deasupra lor doua tipuri de servicii: orientate pe conexiuni si fara conexiuni.

Serviciul **orientat pe conexiuni** este modelat pe baza sistemului telefonic. Cand vrei sa vorbesti cu cineva, mai intai ridici receptorul, apoi formezi numarul, vorbesti si inchizi. Similar, pentru a utilize un serviciu orientat pe conexiuni, beneficiarul trebuie mai intai sa stablileasca o conexiune, sa foloseasca acesta conexiune si apoi sa o elibereze. In esenta conexiunea functioneaza ca o teava: emitatorul introduce obiectele (**bitii**) la un capat, iar receptorul le scoate afara, in aceeasi ordine, la celalalt capat. In majoritatea cazurilor ordinea este mentinuta, astfel incat bitii sa ajunga in aceeasi ordine in care au fost trimisi.

In anumite cazuri cand se stabileste o conexiune,transmitatorul, recetorul si subreteaua negociaza parametrii care vor fi dolositi, cum sunt dimensiunea maxima a mesajului, calitatea impusa a serviciilor, si alte problem de acesti tip. De obicei, una dintre parti face o propunere si cealalta parte poate sa o accepte, sa o rejecteze sau sa faca o contrapropunere.

Sistemul fara conexiuni este modelat pe baza sistemului postal. Toate mesajele contin adresele complete de destinatie si fiecare mesaj circula in system independent de celelalte. In mod normal, atunci cand doua mesaje sunt trimise la aceeasi destinatie, primul expediat este primul care ajunge. Totusi, este posibil ca cel care a fost expediat primul sa intarziue sis a

ajunga mai repede ca al doilea. In cazul unui serviciu orientat pe conexiuni, asa ceva este imposibil.

Fiecare serviciu poate fi caracterizat printr-o **calitate a serviciului .** Unele servicii sunt sigure in sensul ca nu pierd date niciodata. De obicei, un serviciu sigur se implementeaza obligand receptorul sa confirme primirea fiecarui mesaj, astfel incat expeditorul sa fie sigur ca mesajul a ajuns la destinatie. Procesul de confirmare introduce un timp suplimentar si intarzieri. Aceste dezavantaje sunt adese acceptate, insa uneori este trebuie evitate.

Transferul de fisiere este una din situatiile tipice in care este adecvat un serviciu sigur orientat pe conexiuni. Proprietarul fisierului doreste sa fie sigur ca toti bitii ajung corect si in aceeasi oridine in care au fost trimisi. Foarte putin utilizatori ai transferului de fisiere ar prefer un serviciu care uneori amesteca sperde cativa biti, chiar daca acest serviciu ar fi mult mai rapid.

Serviciul sigur orientat pe conexiuni admite doua variante: secventele de mesaje si fluxurile de octeti. Prima varianta mentine delimitarea intre mesaje. Cand sunt trimise doua mesaje de 1024 de octeti, ele vor sosi sub forma a doua mesaje distinct de 1024 de octeti, niciodata ca un singur mesaj de 2048 de octeti. In a doua varianta, conexiunea este un simplu flux de octeti si nu exista delimitari intre mesaje. Cand receptorul primeste 2048 de octeti, nu exista nici o modalitate de a spunde daca ei au fost trimisi sub forma unui mesaj de 2048 octeti, a doua mesaje de 1024 de octeti sau a 2048 mesaje de cate 1 octet. Daca paginile unei carti sunt expediate unei masini fotografice de tiparit printr-o retea, sub forma de mesaje, atunci delimitarea mesajelor poate fi importanta. Pe de alta parte, in cazul unui utilizator care se conecteaza la un server aflat la distanta, este nevoie de numai un flux de octeti de la calculatorul utilizatorului la server. Delimitare mesajelor nu mai este relevanta.

Asa cum am mentionat mai sus, intarzierile introduse de confiramri sunt inacceptabile pentru unele aplicatii. O astfel de aplicatie se referea la treficul de voce digitalizata. Pentru abonatii telefonici este preferabil sa existe putin zgomot pe linie sau sa auda ocazional cate un cuvant distorsionat decat sa se produca o intarziere din cauza asteptarii confirmarii. Similar, atunci cand se transmite o video-conferinta, cativa pixeli diferiti nu reprezinta o problema, in schimb intreruperile pentru a corecta erorile ar fi extrem de suparatoare.

Nu orice aplicatie necesita conexiune. De exemplu, in masura in care posta electronica devine ceva tot mai uzual, se poate sa nu apara foarte curand publicitiatea prin posta electronica? Expeditorul de publicitate prin posta electronica probabil ca nu vrea sa complice stabilind si apoi eliberand o conexiune doar printru un singur mesaj. Nici furnizarea la destinatie cu o rata de corectitudine de 100% nu este esentiala, mai ales daca lucrul acesta costa mai

mult. Tot ceea ce se cere este un mijloc de a trimite un singur mesaj cu o probabilitate mare de a ajunge la destinatie, dar fara o garantie in acest sens. Serviciul nesigur fara conexiuni este deseori numit **serviciu datagrama** ,prin analogie cu serviciul de telegrame- care, la randul sau ,nu prevede trimiterea unei confirmari catre expeditor.

In alte situatii, avantajul de a nu fi necesara stabilirea unei conexiuni pentru a trimite un mesaj scurt este de dorit, dar siguranta este de asemenea estentia. Aceste aplicatii pot utiliza **serviciul datagrama confirmat**. Este ca si cum ai trmite o scrisoare recomandata si ai solicita o confirmare de primire. In clipa in care soseste confirmarea, expeditorul este absolut sigur ca scrisoarea a fost livrata la destinatia cirecta si nu a fost pierduta pe drum.

Mai exista un serviciu, si anume **servicul cerere-raspuns.** In acest serviciu emitatorul transmite o singura datagrama care contine o cerere; replica primita de la receptor contine raspunsul. In acesta categorie intra, de exemplu, un mesaj catre biblioteca locala in care se intreaba unde este vorbita limba Uighur. Serviciul cerere-raspuns este utilizat in mod frecvent pentru a implementa comunicarea in modelul client-server: clientul lanseaza o cerere si serverul raspunde la ea. In figura urmatoare sunt rezumate tipurile de servicii discutate mai sus.

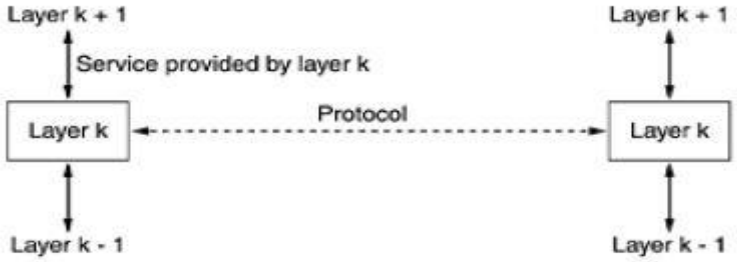
Six different types of service.

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

c) Relatia dintre servicii si protocoale

Desi sunt adesea confundate, serviciile si protocoalele reprezinta concepte distincte. Diferenta intre ele este atat de importanta, incat o subliniem din nou in aceasta sectiune. Un *serviciu* este un set de primitive (operatii) pe care un nivel le furnizeaza nivelului de deasupra sa. Serviciul defineste ce operatii este pregatit nivelul sa realizeze pentru utilizatorii sai, dar nu spune nimic despre cum sunt implementate aceste operatii. Un serviciu este definit in contextul unei interfete intre doua niveluri, nivelul inferior fiind furnizorul servicului si nivelui superios fiind utilizatorul serviciului

Prin contrast, un *portocol* este un set de regului care guverneaza formatul si semnificatia cadrelor, pachetelor sau mesajelor schimbate intre ele de entitatile pachete dintr-un nivel. Entitatile folosesc protocoale pentru a implementa definitiile serviciului lor. Ele sunt libere sa iti schimbe protocoalele dupa cum doresc, cu conditia sa nu modifice serviciul pe care il vad utilizatorii. In acest fel, serviciul si protocolul sunt complet decuplate.



Cu alte cuvinte, serviciile sunt legate de interfetele dintre niveluri, dupa cum este ilustrat si in fugura de mai sus. Prin contrast, protocoalele sunt legate de pachetele trimiste intre entitatile pereche de pe diferite masini. Este important sa nu existe confuzii intre cele doua concepte.

Merita sa facem o analogie cu limbajele de programare. Un serviciu este ca un tip de date abstracte sau ca un obiect intr-un limbaj orientat pe obiecte. Acesta defineste operatiile care pot fi aplicate pe un obiecte, dar nu specifica modul de implementare al operatiilor. Un protocol se refera la *implementarea* serviciului si nu este vizibil pentru utilizatorul serviciului.

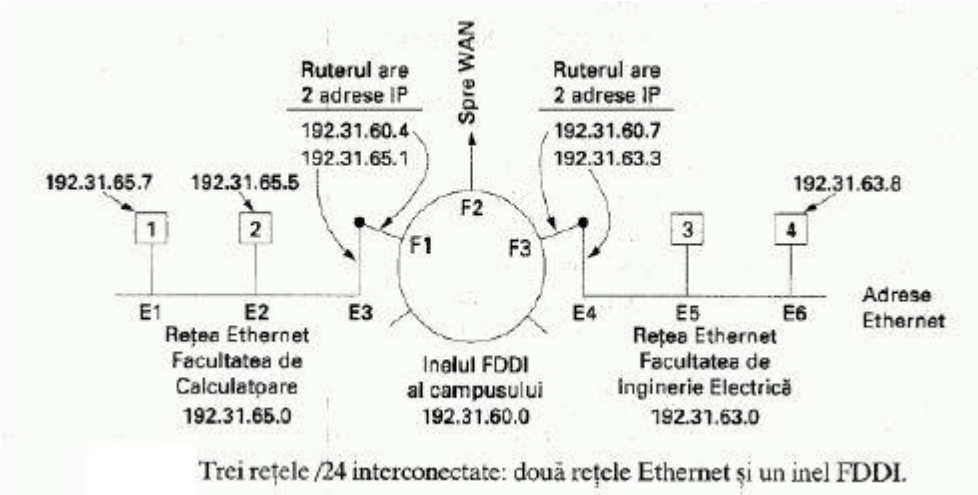
Multe protocoale mai vechi nu faceau diferenta intre serviciu si protocol. Ca urmare, un nivel tipic putea avea o primitiva de serviciu SEND PACKET in care utilizatorul furniza o referinta catre un pachet complet asamblat. Acest aranjament inseamna ca toate modificarile protocolului erau imediat vizibile pentru utilizatori. Majoritatea proiectantilor de retele privest acum un astfel de mecanism ca pe o eroare grava.

III. Protocoale de control in Internet

Internet-ul are cateva protocoale de control la nivelul de retea, incluzand CMP,ARP,RARP,BOOTP si DHCP.

A) Protocolul de rezolutie a adresei: ARP

Desi fiecare masina din Internet are una sau mai multe adrese IP, aceste nu pot fi folosite de fapt pentru trmiterea pachetelor deoarece harware-ul nivelului legaturii de date nu intelege adresele Internet. Actualmente, cele mai multe gazde sunt atasate la un LAN printr-o placa de interfata care intelege numai adresele LAN. De exemplu, fiecare placa Ethernet fabricata pana acum vine cu o adresa Ethernet de 48 de biti. Fabricantii placilor Ethernet cer un spatiu de adrese de la o autoritate centrala pentru a se asigura nu exista doua placi cu aceeasi adresa. Placine trimit si primesc cadre pe baza adresei Ethernet de 48 de biti. Ele nu stiu absolut nimic despre adresele IP pe 32 de biti.



Trei rețele /24 interconectate: două rețele Ethernet și un inel FDDI.

Se pune atunci intrebarea: Cum sunt transformate adresele IP in adrese la nivelul legaturii de date, ca de exemplu Ethernet? Pentru a explica care este functionarea, vom folosi figura de mai sus, in care este ilustrata o universitate mica care are cateva retele de clasa C (denumita acum /24). Avem doua retele Ethernet, una in facultatea de Calculatoare, cu adresa IP 192.31.65.0 si una in facultatea de Inginerie Electrica, cu adresa IP 192.31.63.0. Acestea sunt conectate printr-un inel FDDI la nivelul campusului, care are adresa IP 192.31.60.0. Fiecare masina dintr-o retea Ethernet are o adresa Ethernet unica, etichetata de la E1 la E6, iar fiecare masina de pe inelul FDDI are o adresa FDDI, etichetata de la F1 la F3.

Sa incepem prin a vedea cum trimite un utilizator de pe gazda 1 un pachet unui utilizator de pe gazda 2. Presupunem ca expeditorul stie numele destinatarului, ceva de genu arh@pub.edu. Primul pas este aflarea adresei IP a gazdei 2, cunoscuta ca arh@pub.edu. Acesta cautare este facuta de sistemul numelor de domenii (DSN). Vom considera ca DSN-ul intoarce adresa IP a gazdei 2 (192.31.65.5).

Programele de la nivelurile superioare ale gazdei 1 construiesc un pachet cu 192.31.65.5 in campul *adresa desitantarului*, pachet care este transmos programelor IP pentru a-l transmite. Programele IP se uita la adresa si vad ca destinatarul se afla in propria retea, dar au nevoie de un mijloc prin care sa determine adresa Ethernet a destinatarului. O solutie este sa avem undeva in sistem un fisier de configrare care transfoma adresele IP in adrese Ethernet. Aceasta solutie este posibila, desigur, dar pentru organizatii cu mii de masini, mentrinerea fisierelor actualizate este o actiune consumatoare de timp si care poate genera erori.

O solutie mai buna este ca gazda 1 sa trimita un pacjet de difuzare in reteau Ethernet intreband: “Cine este proprietarul adresei IP 192.31.65.5?”. Pachetul de difuzare va ajunge la toate maisinile din reteaua Ethernet 192.31.65.0 si fiecare isi va verifica adresa IP. Numai gazda 2 va raspunde cu adresa sa Ethernet (E2). In acest mod gazda 1 afla ca adresa IP 192.31.65.5 este pe gazda cu adresa Ethernet E2. Protocolul folosit pentru a pune astfel de intrebari si a primi raspunsul se numeste ARP (Address Resolution Protocol – Protocolul de rezolutie a adresei). Aproape toate masinile din Internet il folosesc.

Avantajul folosirii ARP fata de fisierele de configurare il reprezinta simplitatea. Administratorul de sistem nu trebuie sa faca prea multe, decat sa atribuie fiecarei masini o adresa IP si sa hotarasca mastile subretelelor. Arp_ul face restul.

In acest punc, programele IP de pe gazda 1 construiesc un cadru Ethernet adresat catre E2, pun pachetul IP (adresat catre 192.31.65.5) in campul informatie utila si il lanseaza pe reteaua Ethernet. Placa Ethernet a gazdei 2 detecteaza acest cadru, recunoaste ca este un cadru pentru ea, il ia repede si genereaza o intrerupere. Driverul Ethernet extrage pachetul IP din informatia utila si il trimite programelor IP, care vad ca este corectat adresat si il prelucreaza.

Pentru a face ARP-ul mai eficient sunt posibile mai multe optimizari. Pentru inceput, la fiecare executie a ARP, masina pastreaza rezultatul pentru cazul in care are nevoie sa contacteze din nou aceeasi masina in scurt timp. Data viitoare va gasi local corespondentul adresei, evitandu-se astfel necesitatea unei a doua difuzari. In multe cazuri, gazda 2 treuie sa trimita inapoi un raspuns, ceea ce forteaza sa execute ARP, pentru a determina adresa Ethernet a expeditorului. Acesta difuzare ARP poate fi evitata obligand gazda 1 sa includa in pachetul

ARP corespondenta dintre adresa sa IP si adresa Ethernet. Cand pachetul ARP ajunge la gazda 2, perechea (192.31.65.7,E1) este memorata local de ARP pentru o folosire ulterioara. De fapt, toate masinile din reteau Ethernet pot memora acesta relatie in memoria ARP locala.

Alta optimizare este ca fiecare masina sa difuzeze corespondenta sa de adrese la pornirea masinii. Acesta difuzare este realizata in general printr-un pachet ARP de cautare a propriei adrese IP. Nu ar trebui sa exista un raspuns, dar un efect lateral al difuzarii este introducerea unei inregistriri in memoria ascunsa ARP a tuturor. Adesi totusi soseste un raspuns (neasteptat), inseamna ca doua masini au ceeasi adresa IP. Noua masina ar trebui sa-l informeze pe administratorul de sistem si sa nu porneasca.

Pentru a permite schimbarea relatiei, de exemplu, cand o placa Ethernet se strica si este inlocuita cu una noua (si astfel apare o noua adresa Ethernet), inregistrarile din memoria ascunsa ARP ar trebui sa expire dupa cateva minute.

Sa privim din nou in figura de mai sus, numai ca de acesta data gazda 1 vrea sa trimita un pachet catre gazda 4 (192.31.63.8). Folosirea ARP va esua pentru ca gazda 4 nu va vedea difuzarea (ruterele nu trimit mai departe difuzarile de nivel Ethernet). Exista doua solutii. Prima: routerul facultatii de Calculatoare poate fi configurat sa raspunda la cererile ARP pentru reteaua 192.31.63.0 (si posibil si pentru alte retele locale). In acest caz, gazda 1 va memora local perechea (192.31.63.8,E3) si va trimite tot traficul pentru gazda 4 catre ruterul local. Acesta solutie se numeste ARP cu intermediar (proxy ARP). A doua solutie este ca gazda 1 sa-si dea seama imediat ca destinatia se afla pe o retea aflata la distanta si sa trmita tot traficul catre o adresa Ethernet implicita care trateaza tot traficul la distanta, in acest caz E3. Acesta solutie nu necesita ca routerul facultatii de Calculatoare sa stie ce retele la distanta deserveste.

In ambele cazuri, ceea ce se intampla este ca gazda 1 impacheteaza pachetul IP in campul informatie utila dintr-un cadtu Ethernet adresat catre E3. Cand routerul facultatii de Caluculatoare primeste cadrul Ethernet, extrage pachetul IP din campul informatie utila si cauta adresa IP din tabelel sale de dirijare. Descopera ca pachetele pentru reteaua 193.31.63.0 trebuie sa mearga catre ruterul 192.31.60.7. Daca nu cunoaste inca adresa FDDI a lui 193.31.60.7, difuzeaza un pachet ARP pe inel si afla ca adresa din inel este F3. Apoi insereaza pachetul in campul informatie utila al unui cadru FDDI adresat catre F3 si il transmite pe inel.

Driverul FDDI al ruterului facultatii de Inginerie Electrica scoate pachetul din campul informatie utila si il trimite programelor IP care vad ca trebuie sa trimita pachetul catre 192.31.63.8. Daca acesta adresa IP nu este in memoria ascunsa ARP, difuzeaza o cerere ARP pe reteaua Ethernet a facultatii de Inginerie Electrica si afla ca adresa destinatie este E6, astfel incat construieste un cadru Ethernet adresat catre E6, pune pachetul in campul informatie utila

si il trimite in reteaia Ethernet. Cand cadrul Ethernet ajunge la gazda 4, pachetul este extras din cadru si trimis programelor IP pentru procesare.

Transferul intre gazda 1 si o retea la distanta peste un WAN functioneaza in esenta asemanator, cu exceptia ca de data asta tabelele ruterului facultatii de Calculatoare ii vor indica folosirea ruterului WAN, a carui adresa FDDI este F2.

Principala structura a pachetului ARP este evidentiata in figura urmatoare, care ilustraza cazul in care retele IPv4 ruleaza pe Ethernet. In acest scenariu pachetele au campuri de 48-biti.

Internet Protocol (IPv4) over Ethernet ARP packet		
bit offset	**0 – 7**	**8 – 15**
0	Hardware type (HTYPE)	
16	Protocol type (PTYPE)	
32	Hardware address length (HLEN)	Protocol address length (PLEN)
48	Operation (OPER)	
64	Sender hardware address (SHA) (first 16 bits)	
80	(next 16 bits)	
96	(last 16 bits)	
112	Sender protocol address (SPA) (first 16 bits)	
128	(last 16 bits)	
144	Target hardware address (THA) (first 16 bits)	
160	(next 16 bits)	
176	(last 16 bits)	
192	Target protocol address (TPA) (first 16 bits)	
208	(last 16 bits)	

B) RARP,BOOT si DHCP

ARP-ul rezolva problema aflarii adresei Ethernet corespunzatoare unei adrese IP date. Cateodata trebuie rezolvata problema inversa: dandau-se o adresa Ethernet, care este adresa IP corespunzatoare? In particular, aceasta problema apare cand se porneste o statie de lucru fara disc. O astfel de masina va primi, in mod normal, imaginea binara a sistemului sau de operare de la un server de fisiere la distanta. Dar cum isi afla adresa IP?

Prima solutie proiectata a fost **RARP** (**Reverse Address Resolution Protocol –** Protocol de rezolutie inversa a adresei) (definit in RFC 903). Acest protocol permite unei statii de lucru de-abia pornita sa difuzeze adresa sa Ethernet si sa spuna: “Adresa mea Ethernet de 48 de biti este 14.04.05.18.01.25. Stie cineva adresa mea IP?” Serverul RARP vede acesta cerere, cauta adresa Ethernet in fisierele sale de configurare si trimite inapoi adresa IP corespunzatoare.

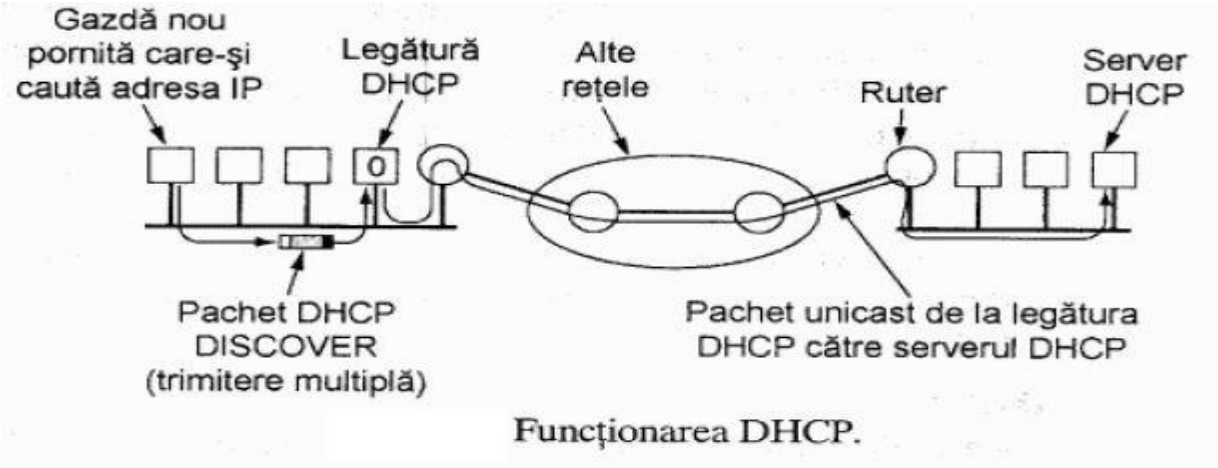
Folosirea RARP este mai buna decat introducerea unei adrese IP in imaginea de memorie,pentru ca permite ca aceeasi imagine sa fie folosita pe toate masinile. Daca adresa IP ar fi fixata inauntrul imaginii, atunci fiecare statie de lucru ar necesita imaginea sa proprie.

Un dezavantaj al RARP este ca, pentru a ajunge la serverul RARP, foloseste o adresa destinatie numai din 1-uri (difuzare limitata). Cu toate acestea, asemenea difuzarii nu sunt propagate de rutere, asa incat este necesar un server RARP in fiecare retea. Pentru a rezolva acesta problema, a fost inventat un protocol alternativ de pornire, numit BOOTP. Spre deosebire de RARP, acesta foloseste mesaje UDP, care sunt propagate prin rutere. De asemenea furnizeaza unei staii de lucru fara disc informatii suplimentare, care includ adresa IP a serverului de fisiere care detine imaginea de memorie, adresa IP a ruterului implicit si masca de subretea care se foloseste.

O problema serioasa cu BOOTP este necesara configurarea manuala a corespondentelor intre adresele IP si adresele Ethernet. Atunci cand o noua gazda este adaugata la LAN, nu poate fololsi BOOTP decat atunci cand un administrator ii aloca o adresa IP si introduce manual (adresa Ethernet, adresa IP) in tabelele de configurare BOOTP. Pentru a elimina acest pas predispus la erori, BOOTP a fost extins si i-a fost dat un nou nume: DHCP (Dynamic Host Configuration Protocol – Protocol dinamic de configurare a gazdei). DHTCP permite atat atribuirea manuala de adrese IP, cat si atribuirea automata. In majoritatea sistemelor, a inlocuit in marea parte RARP si BOOTP.

Ca si RARP si BOOTP, DHCP este bazat pe ideea unui server special care atrebuie adrese IP gazdelor care cer una. Acest server nu trebuie sa se afle in acelasi LAN cu gazda care face cererea. Deoarece serverul DHCP s-ar putea sa nu fie accesibil prin difuzare, este nevoie ca in

fiecare LAN sa existe un agent de legatura DHCP (DHCP relay agent) asa cum se vede in figura de mai jos.



Funcţionarea DHCP.

Pentru a-si afla adresa IP, o masina tocmai pornita difuzeaza un pachet DHCP DISCOVER. Agentul de legatura DHCP din LAN intercepteaza toate difuzarile DHCP. Atunci cand gaseste un pachet DHCP DISCOVER, ii trimite ca pachet unicast serverului DHCP, posibil intr-o retea departata. Singura informatie de care are nevoie agentul este adresa IP a serverului DHCP.

O problema care apare cu atriburea automata a adreselor IP dintr-o rezerva comuna este cat de mult ar trebui alocata o adresa IP. Daca o gazda paraseste reteaua si nu returneaza adresa sa IP serverului DHCP, acea adresa va fi pierduta permanent. Dupa o perioada de timp vor fi pierdute multe adrese. Pentru a preveni acesta, atribuirea adresei IP va fi pentru o perioada fixa de timp, o tehnica numita inchiriere. Chiar inainte ca perioada de inchiriere sa expire, gazda trebuie sa ii ceara DHCP-ului o reinnoire. Daca nu reuseste sa faca cererea sau daca cererea este respinsa , gazda nu va mai putea folosi adresa IP care ii fusese data mai devreme.

Bibliografie:

1. Cisco.Press.Internetworking.Technologies.Handbook.Fourth.Edition

2. Tanenbaum, Andrew S. - Computer Networks - 4th Edition

3. http://en.wikipedia.org/wiki/Address_Resolution_Protocol

4. http://arpon.sourceforge.net/

5. http://www.whatismyip.com/ip-faq/what-is-dhcp/

6. Ralph Droms; Ted Lemon (2003). *The DHCP Handbook*

Alexandru Ion

C) DHCP

DHCP (prescurtat de la **Dynamic Host Configuration Protocol**) este un protocol de rețea de calculatoare folosite de gazde (clienți DHCP) care atribuie adrese IP și alte informații de configurare de rețea importante în mod dinamic.

Istoric

RFC 1531 a definit inițial DHCP ca protocol standard în octombrie 1993, urmând Bootstrap Protocol (BOOTP). Următorul update, RFC 2131, lansat în 1997, este definiția DHCP curenta pentru rețele Internet Protocol version 4 (IPv4). Extensiile DHCP pentru IPv6 (DHCPv6) au fost publicate în RFC 3315.

Privire de ansamblu

Dynamic Host Configuration Protocol automatizează alocarea parametrilor de rețea la dispozitive de către unul sau mai multe fault-tolerant servere DHCP. Chiar și în rețele mici, DHCP este folositor, deoarece simplifica adăugarea unor noi echipamente în rețea.

Când un client configurat (un computer sau orice alt dispozitiv de rețea) se conectează la rețea, clientul DHCP trimite o broadcast interogare în ce privește informația necesară la serverul DHCP. Serverul DHCP gestionează o rezervă de adrese IP și informații despre configurarea parametrilor clientului, ca gateway-ul implicit, numele domeniului,serverul DNS, alte servere ca serverul de timp, ș.a.m.d. La primirea unei cereri valide, serverul atribuie calculatorului o adresă de IP, un contract de leasing (perioada de validitate a alocării respective), precum și alți parametri de configurare de IP, cum ar fi masca de subrețea și gateway-ul implicit . Interogarea este de obicei inițiată imediat după boot, și trebuie să fie completată, înainte ca clientul să poată iniția comunicareaIP cu alte gazde.

Funcție de implementare, serverul DHCP poate avea trei metode de alocare a adreselor IP:

- alocare dinamică: Un administrator de rețea atribuie o serie de adrese IP la DHCP, și fiecare computer din LAN este configurat să ceară o adresa de IP de la serverul DHCP în timpul inițializării de rețea. Procesul de cerere și aprobare folosește un concept ca un contract de leasing pe o perioada determinată, permițând serverului DHCP să revendice (și să realoce) adrese IP disponibile (refolosirea dinamică de adrese de IP).
- alocare automată: Serverul DHCP alocă în permanență o adresă de IP disponibilă, din gama definită de administrator, către un client. Acest proces este asemănător alocării dinamice, dar serverul DHCP păstrează un tabel cu alocările de IP anterioare, astfel încât să poată atribui preferențial pentru un client aceeași adresă de IP pe care acesta a avut-o anterior.

- alocare statică: Serverul DHCP alocă adresa de IP în baza unui tabel cu perechi adresa MAC/adresa IP, acestea fiind completate manual (probabil de către administratorul rețelei). Numai clienții care au adresa MAC lisată în acest tabel vor primi o adresă de IP. Această caracteristică (care nu e suportată de orice router) este denumităStatic DHCP Assignment (by DD-WRT), fixed-address (by the dhcpd documentation), *DHCP reservation* or *Static DHCP* (by Cisco/Linksys), și *IP reservation* sau *MAC/IP binding* (de către diverși alți producători de routere).

Închirierea adreselor DHCP

Închirierea este fundamentală pentru procesul DHCP. Fiecare adresă oferită de un server DHCP are o perioadă de închiriere asociată, perioadă în care clientul are permisiunea să folosească adresa. Perioada de închiriere este denumită ”lease time” și poate avea orice valoare, de la câteva minute până la câteva luni, ani sau chiar pentru totdeauna. Închirierea pe perioadă nelimitată de timp transformă adresarea dinamică în adresare statică.

Dacă s-a scurs peste 50% din timpul de închiriere al adresei, clientul trimite serverului care i-a închiriat adresa, o cerere de prelungire a perioadei de utilizare a adresei (”renew”). Dacă acest client nu a reușit prelungirea perioadei de închiriere de la serverul de la care a primit inițial închirierea (”lease-ul”) la scurgerea a 87,5% (7/8) din timp, clientul trimite un ”broadcast packet”, încercând să închirieze o adresă IP de la orice server existent în rețea. Procesul de închiriere poate fi anulat atât de client cât și de server înaintea perioadei stabilite inițial. De asemenea, serverul DHCP are posibilitatea de a trimite mesaje clienților obligându-i să înnoiască contractul de închiriere înainte de terminarea lui.

Confirmarea cererilor DHCP

Când serverul DHCP primește mesajul DHCPREQUEST de la client, procesele de configurare intră în faza finală. Faza de confirmare presupune trimiterea unui pachet DHCPACK clientului. Acest pachet include durata contractului de leasing, precum și orice alte informații de configurare pe care clientul le-ar putea fi solicitat. În acest moment, procesul de configurare IP este finalizat.

Protocolul așteaptă ca clientul DHCP să-și configureze interfața de rețea cu parametri negociați.

DHCPDISCOVER

UDP Src=0.0.0.0 sPort=68 Dest=255.255.255.255 dPort=67			
OP	**HTYPE**	**HLEN**	**HOPS**
0x01	0x01	0x06	0x00
XID			
0x3903F326			
SECS		**FLAGS**	
0x0000		0x0000	
CIADDR (Client IP Address)			
0x00000000			
YIADDR (Your IP Address)			
0x00000000			
SIADDR (Server IP Address)			
0x00000000			
GIADDR (Gateway IP Address)			
0x00000000			
CHADDR (Client Hardware Address)			
0x00053C04			
0x8D590000			
0x00000000			
0x00000000			
192 octets of 0s, or overflow space for additional options. BOOTP legacy			
Magic Cookie			
0x63825363			
DHCP Options			
DHCP option 53: DHCP Discover			
DHCP option 50: 192.168.1.100 requested			
DHCP option 55: Parameter Request List: Request Subnet Mask (1), Router (3), Domain Name (15), Domain Name Server (6)			

DHCPOFFER

UDP Src=192.168.1.1 sPort=67 Dest=255.255.255.255 dPort=68			
OP	**HTYPE**	**HLEN**	**HOPS**
0x02	0x01	0x06	0x00
XID			
0x3903F326			
SECS		**FLAGS**	
0x0000		0x0000	
CIADDR (Client IP Address)			
0x00000000			
YIADDR (Your IP Address)			
0xC0A80164			
SIADDR (Server IP Address)			
0xC0A80101			
GIADDR (Gateway IP Address)			
0x00000000			
CHADDR (Client Hardware Address)			
0x00053C04			
0x8D590000			
0x00000000			
0x00000000			
192 octets of 0s. BOOTP legacy			
Magic Cookie			
0x63825363			
DHCP Options			
DHCP option 53: DHCP Offer			
DHCP option 1: 255.255.255.0 subnet mask			
DHCP option 3: 192.168.1.1 router			
DHCP option 51: 86400s (1 day) IP lease time			
DHCP option 54: 192.168.1.1 DHCP server			
DHCP option 6: DNS servers 9.7.10.15, 9.7.10.16, 9.7.10.18			

Protocolul de mesaje de control pentru Internet (ICMP)

Protocolul de mesaje de control pentru Internet (ICMP) este unul dintre protocoalele fundamentale din suita TCP/IP care foloseste la semnalizarea si diagnosticarea problemelor din retea.

Este utilizat, in primul rand, de sistemele de operare ale calculatoarelor din retea pentru a trimite mesaje de eroare, indicand de exemplu, ca serviciul solicitat nu este disponibil sau ca gazda sau ruterul nu putut fi atinse. Acest protocol poate fi utilizat si pentru a schimba mesaje cu intrebari (“query messages”).

Ca diferenta intre ICMP si protocoalele de trensport, cum ar fi, TCP si UDP, este faptul ca ICMP nu este de regula folosit pentru a schimba date intre sisteme, si nici nu este ocupat de catre aplicatiile unui user.

ICMP pentru **Internet Protocol version 4** (**IPv4**) este cunoscut si ca ICMPv4. IPv6 are de asemenea, un protocol asemanator ICMPv6.

Mesajele ICMP sunt, de regula folosite cu scopul de a diagnostica sau de a controla, ori sunt generate ca raspuns la erorile aparute in operatiile cu IP. Erorile ICMP sunt directionate catre sursa IP care a trimis pachetul.

ICMP este un protocol care functioneaza la nivelul 3 al modelului OSI (**Nivelul Retea**)**,** nefiind necesara utilizarea unui protocol de transport (TCP sau UDP) sau a unui port de comunicatie. El este o parte componenta a protocolului IP. Acest protocol permite incapsularea in interiorul cadrului IP a unor informatii, care o data ajunse la destinatia specificata, determina generarea unui raspuns catre sursa ICMP, din care se poate deduce timpul de raspuns pe un canal de comunicatie.

Parametrii ICMP pot fi astfel configurati incat sa determine generarea unui raspuns din partea fiecarui echipament de comunicatie tranzitat de pachetele ICMP , de exemplu comanda “tracert” sau “ping route”, obtinandu-se si o imagine a traseului fizic corespunzator canalului de comunicatie. In cazul in care nodul de destinatie sau un nod tranitat nu raspunde la un pachet ICMP, este asociat un mesaj de eroare, care poate oferi informatii utile in stabilirea cauzelor pentru care nu poate fi atinsa o destiantie: cale de cumunicatie nefunctionala, rute IP necorespunzatoare etc.

Functiile ICMP:

- Ruterii transmit altor ruteri sau sistemelor mesaje de eroare sau de control : numai pentru raportarea erorilor, nu pentru crestearea fiabilitatii IP.
- Pentru datagrame fragmentate, mesajele ICMP sunt transmise numai pnetru eventuale erori produse in cazul primului fragment.
- Mesajele ICMP nu sunt transmise ca raspuns la o problema legata de o datagrama care nu are sursa ce desemneaza un sistem unic (unicast) - adresa sursa nu poate fi zero, o adresa de transmisie in bucla (loopback), o adresa de difuzare (broadcast) sau o adresa de grup (multicast);

Exemplu de utilizare:

Probabil cele mai utilizate programe care se bazeaza pe ICMP sunt ping si traceroute.Ping trimite mesaje ICMP de tip **echo request** (“cerere de ecou”) catre calculatorul tinta si asteapta de la acesta mesaje ICMP de tip **echo reply** (“respuns de tip echo”). Daca acestea nu sunt primite, se poate presupune ca este ceva neinregula cu conexiunea dintre cele doua calculatoare.

Toate pachetele IP au in antet un camp special numit TTL (“**Time to live**”)**.** Acest camp este decrementat de fiecare data cand trece printr-un ruter. Pentru a evita bulele de rutare, in momentul in care campul TTL ajunge la zero pachetul nu mai este trimis mai departe. In acesta situatie, ruter-ul care a decrementat campul TTL la zero trimite catre calculatorul-origine al pachetului, adresa acestuia se afla tot in prologul IP, un mesaj ICMP de tip **time exceeded.**

Programul **traceroute** profita de acest mecanism si trimite catre calculatorul tinta, pachete UDP cu valori ale campului TTL din ce in ce mai mari, cu scopul de a obtine mesaje **time exceeded** de la toate routerele aflate pe traseu.

1) **Aplicatia Ping**

Aceasta aplicatie este una din cele mai utile unelte de depanare a retelelor. Numele ii vine de la sonarul submarinelor – se trimite o salva scurta si se asculta ecoul ‘ping’ rezultat. Practic intr-o retea IP salva consta intr-un singur pachet iar raspunsul asteptat este de asemenea un singur pachet. Deoarece astfel se testeaza insasi functionalitatea de baza a retelei IP – de a transmite un pachet, putem afla multe lucruri dintr-un simplu ping. Implementarea functiei ping se face utilizand pachetele ICMP de ecou si ecou raspuns.

Functiile indeplinite de aplicatie:

- Plaseaza numere de secventa in fiecare pachet transmis si analizeaza secventa de pachete receptionata. Astfel putem afla daca avem pachete abandonate, duplicate sau reordonate.
- Calculeaza suma de control a fiecarui pachet, astfel verificand eventuale distrugeri ale pachetelor
- Plaseaza o marca de timp in fiecare pachet, care este intoarsa de la destinatie si astfel utilizata pentru a calcula timpul de parcurgere dus-intors (Round Trip Time RTT)
- Raporteaza nu in ultimul rand si celelalte mesaje ICMP aparute, care altfel ar fi ramas invizibile utilizatorului (de exemplu raportul unui router de 'destinatie de neatins')

2) **Aplicatia Traceroute**

Traceroute incearca sa traseze drumul parcurs de pachete pana la destinatie. Altfel spus, determina ruta acestor pachete. Da, intr-adevar, exista in specificatiile IP un mecanism pentru memorarea rutei parcurse de pachet, dar aceste specificatii sunt vag definite, foarte rar implementate si adesea dezactivate din motive de securitate, astfel incat nu ne putem baza in nici un fel pe ele. Traceroute isi face treaba in alt fel, putem sa-i spunem 'ilegal'.

Traceroute trimite pachete cu valori ale TTL foarte mici. TTL (Time To Live) este un camp al headerului IP prevazut sa previna posibilitatea intrarii pachetelor in bucle infinite de rutare. Fiecare router scade cu 1 valoare campului TTL, iar in momentul cand aceasta ajunge zero, pacehtul este considerat expirat si abandonat. In mod normal routerul care face aceasta genereaza un mesaj ICMP de tipul Time Exceeded catre emitatorul pachetului.

Folosind deci pachete cu valori ale TTL mici, traceroute forteaza routerele de pe traseu sa ne trimita aceste mesaje ICMP ce vor fi folosite pentru a identifica routerele. Un TTL de 1 va forta primul router sa ne trimita un mesaj, 2 – un mesaj de la al doilea router samd.

Pachetele trebuie in principiu transmise destinate unui port nefolosit pe destinatar, altfel aplicatia ascultand pe acel port va primi pachete de neinteles pentru ea.

Structura segmentului ICMP

Un pachet ICMP este un pachet IP in care campul *protocol* are valoarea 1 pentru ICMPv4, respectiv 58 pentru ICMPv6, iar zona de date utile este structurata conform standardului ICMP.

Pachetele ICMP sunt de obicei generate de modulul de retea al unui nod, ca urmare a unei erori aparute in livrarea unui pachet IP. Pachetele ICMP mai pot fi generate si de programe utilizator, prin intermediul socket-urilor de tip SOCK_RAW. Astfel de aplicatii servesc la testarea functionarii retelei.

O data generat, un pachet ICMP este transmis prin retea ca orice alt pachet IP. Ajuns la destinatie, modulul de retea (IP) al nodului destinatie examineaza compul *protocol* si, constatand ca este vorba de un pachet ICMP, il livreaza modulului ICMP al nodului destinatie. Modulul ICMP trebuie sa fie prezent si functional in orice nod IP; in implementarile obisnuite este parte a nucleului sistemului de operare al calculatorului ce constituie nodul.

Datele utile sunt formate conform standardului ICMP si incep cu doi intregi pe cate 8 biti reprezentand tipul si subtipul mesajului ICMP (tabelul de mai jos). Formatul retului pachetului depinde de tipul mesajului ICMP: in majoritatea cazurilor, este prezenta o copie a primilor cateva zeci de octeti din pachetul IP care a dus la generarea pachetului ICMP.

Antetul

Antetul sau header-ul ICMP incepe imediat dupa antetul IPv4. Toate Pachetele ICMP dispun de un antet de 8 octeti si de o sectiune de date de lungime variabila. Structura antetului ICMP este redata in figura de mai jos.

Biti	0-7	8-15	16-23	24-31
0	Tip	Cod	Suma de control	
32	Restul antetului			

TIP – tipul pachetului ICMP

Cod - subtipul pachetului ICMP in functie de tipul slectat anterior

Suma de control – suma de control calculata in functie de campurile antet ICMP + sir de date

Restul antetului – camp de 4 biti ce variaza ca si continut pe baza tipului/codului antetului ICMP.

Tip	Subtip	Ce semnalizează
3 — Destination unreachable	0 — network unreachable 1 — host unreachable 3 — protocol unreachable	pachet nelivrabil, conform § 10.2.5.1
	4 — fragmentation needed	pachet prea mare şi flagul *nu fragmenta* setat; vezi § 10.2.6.1
	5 — source route failed	pachetul a avut opţiunea *dirijare de către sursă* şi ruta specificată este invalidă.
11 — time exceeded	0 — TTL exceeded	pachetul se află de prea mult timp în reţea (probabil ciclează), § 10.2.5.3
	1 — fragment reassembly time exceeded	probabil fragment pierdut, § 10.2.6.1
12 — parameter problem		pachet neconform cu standardul
4 — source quench		cerere încetinire sursă, § 10.2.5.4
5 — redirect	0 — network 1 — host 2 — TOS & network 3 — TOS & host	redirecţionare, § 10.2.5.5
8 — echo request		cerere ecou, § 10.2.5.2
9 — echo reply		răspuns ecou, § 10.2.5.2

Tipuri şi subtipuri mai importante de pachete ICMPv4

Tip – Acest camp poate lua una din urmatoarele valori (8 biti), in functie de tipul mesajului:

- 0 - Raspuns ecou (Echo reply),
- 3 - Destinatie inaccesibila (Destination unreachable),
- 4 - Oprirea sursei (Source quench),
- 5 - Redirectare,
- 8 - Cerere ecou,
- 9 - Anuntarea unui ruter,
- 10 - Solicitarea unui ruter,
- 11 - Depasire timp,
- 12 - Problema legata de un parametru,
- 13 - Cerere eticheta de timp,

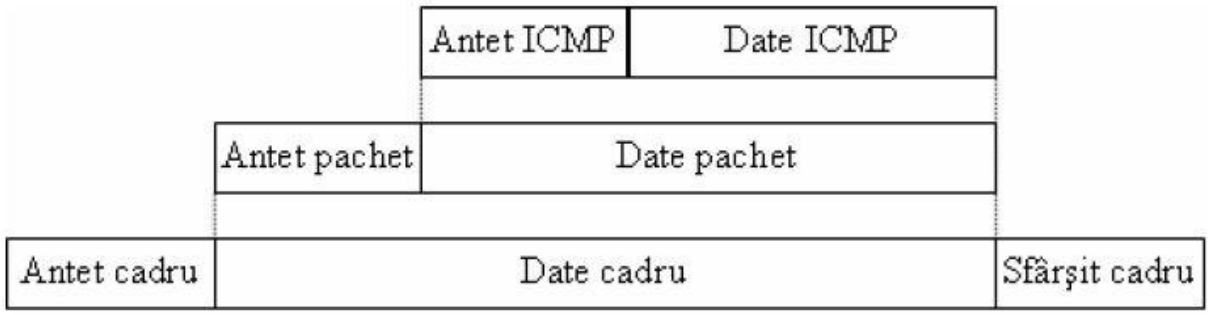
- 14 - Raspuns eticheta de timp,
- 17 - Cerere masca de adrese,
- 18 - Raspuns masca de adrese,
- 30 - Descoperire ruta (Traceroute),
- 37 - Cerere nume domeniu,
- 38 - Raspuns nume domeniu

Cod – Contine codul erorii pentru datagrama raportata de acest mesaj ICMP. Interpretarea acestui camp depinde de tipul mesajului. Acest camp furnizeaza inforamtii suplimentare despre tipul mesajului.

Suma de control – Contine suma de verificare, 16 biti, folosind acelasi algoritma ca si IP dar veriificand numai mesajul ICMP, incepand cu campul dedicat tipului mesajului. Daca valoare sumei nu coincinde cu valoarea calculata la receprie pe baza continutului receprionat, atunci datagrama este eliminata.

Restul antetului – contine informatia corespunzatoare mesajului ICMP curent. De cele mai multe ori, aces camp contine o portiune din pachetul IP original, cel pentru care a fost generat mesajul ICMP curent.

Fiecare mesaj ICMP este inclus in campul de date al unui pachet.



Încapsularea mesajului ICMP.

Tipuri de mesaje ICMP

- **Mesajul Destinatie Inaccesibila (Destination Unreachable)** este folosit atunci cand subreteaua sau un ruter nu pot localiza destinatia, sau un pachet cu bitul DF nu poate fi livrat deoarece o retea cu “pachete mici” ii sta in cale.

- **Mesajul Timp Depasit (Time Exceeded)** este trimis cand un pachet este eliminat datorita ajungerii contorului sau la zero. Acest mesaj este un simptom al buclarii pachetelor, al unei enorme congestii sau al stabilirii unor valori prea mici pentru ceas.Daca acest mesaj este receptionat de la un ruter intermediar înseamna ca valoarea din câmpul TTL a unui pachet IP a ajuns la zero. Daca mesajul este receptionat de la un sistem de destinatie înseamna ca timpul TTL dintr-un fragment IP a expirat în timpul reasamblarii, datorita întarzierii unui fragment.

- **Mesajul Problema De Parametru (Parameter Problem)** indica detectarea unei valori nepermise intr-un camp din antet. Acesta problema indica o eroare in programele IP ale gazdei emitatoare sau eventual in programele ubnui ruter tranzitat.

- **Mesajul Oprire Sursa (Source Quench)** a fost folosit pe vremuri pentru a limita traficul gazdelor care trimiteau prea multe pachete. Cand p gazda primea acest mesaj, era de asteptat sa incetineasca ritmul de transmisie. Este folosit arareori, deoarece cand apare o congestie, aceste pachete au tendinta de a turna mai mult gaz pe foc.

- **Mesajul Redirectare (Redirect)** este folosit atunci cand un ruter observa ca un pachet pare a fi dirijat gresit. Este folosit de ruter pentru a spune gazdei emitatoare despre eroarea probabila. Daca acest mesaj e receptionat de la un ruter intermediar înseamna ca sistemul sursa ar trebui sa trimita urmatoarele datagrame catre ruterul a carui adresa IP este specificata în mesajul ICMP.Acest ruter preferat va fi întotdeauna aflat în aceeasi subretea cu sistemul emitent al datagramei si ruterul care a returnat datagrama.

- **Mesajele Cerere Ecou (Echo Request)** si **Raspuns Ecou (Echo Reply)** sunt folosite pentru a vedea daca o anumita destinatie este accesibila si activa. Este de asteptat ca la receptia mesajului Ecou, destinatia sa raspunda printr-un mesaj **Raspuns Ecou**.Cererea contine un câmp de date optionale. Raspunsul va contine o copie a acestor date. În felul acesta se poate verifica daca o anumita destinatie este accesibila si raspunde.

- **Mesajele Cerere eticheta de** timp si **Raspuns eticheta de timp** sunt utilizate pentru depanare si masurare a performantelor. Acestea nu sunt utilizate pentru sincronizarea de ceas. Transmitatorul initializeza identificatorul si numarul de secventa (care se utilizeaza în cazul în care sunt transmite mai multe etichete de timp), stabileste eticheta initiala de timp si transmite pachetul catre destinatie. Statie destinatie actualizeaza etichetele de timp asociate receptiei si transmisiei, modifica tipul etichetei de timp din cerere în raspuns si o returneaza statiei sursa. Pachetul contine doua etichete de timp daca exista o diferenta semnificativa de timp între timpul de receptie si timpul de emisie.

- **Mesajele Cerere de masca de adrese** si **Raspuns cu masca de adrese**. Cererea de masca de adrese este utilizata de catre un sistem pentru a determina masca subretelei folosita în cadrul unei retele asociate. Cele mai multe sisteme sunt configurate cu masca (sau mastile) de subretea asociata. Totusi, unele sisteme, cum ar fi statiile de lucru fara disc, trebuie sa obtina aceasta informatie de la server. Un sistem foloseste protocolul RARP (Reverse Address Resolution Protocol) pentru a obtine adresa sa IP. Pentru a obtine masca de subretea, sistemul transmite prin difuzare cererea de masca de adresa.

- Mai exista si alte mesaje ICMP pentru semnalizarea unor situatii de congestie (atunci când un ruter este prea încarcat pentru a prelucra un nou pachet, care din acest motiv va fi pierdut), semnalizarea unei rutari ciclice (o ruta infinita, propagare în bucla), etc.

ICMP Router Discovery Messages

Prezentarea generala a protocolului

Inainte ca un host sa poata sa trimita datagrame IP dincolo de reteaua la care este atasat, el trebuie sa afle adresa ultimului ruter operational din acea retea. De obicei, acest lucru este realizat prin citirea unei liste care contine una sau mai multe adrese ale unor rutere, dintr-un fisier de configuratie la creearea retelei. In legaturile muticast, unele hosturi descopera adresele ruterelor, ascultand traficul protocolului de rutare. Ambele metode au neajunsuri importante: configuratia fisierelor trebuie mentinuta manual, lucru ce reprezinta o povara destul de grea pentru administrator, si nu permit detectarea schimbarilor dinamice in disponibilitatea ruterelor.

Vom discuta despre o alterantiva la solutiile de mai sus pentru descoperirea ruterelor, folosind o pereche de mesaje ICMP pentru legaturi multicast. Elimina nevoia pentru a folosi configurarea manuala a adreselor ruterelor si este independent de orice protocol de rutare.

Mesajele de descoperire a ruterelor mai sunt numite si “Anunturi de rutere” ("Router Advertisements") sau “Solicitari de ruter ” ("Router Solicitations"). Periodic, fiecare router face astfel de anunturi pentru fiecare dintre interfetele sale, anuntand adresa IP a acelei interfete. Gazdele (hosts) descopera adresele ruteri-lor vecini doar prin ascultarea acelor anunturi. Atunci cand un calculator sau un host se leaga la o retea sau cand acesta porneste , va trimite mesaje in intreaga reatea de tipul “Solicitare de ruter” , cerand astfel, “anunturi de rutere” de la celelalte rutere. Face acest lucru, decat sa astepte pana la urmatoarea difuzare perioadica in retea a ruterelor. Daca aceste “anunturi” nu sunt trimise gazda poate sa retransmita solicitarea, dar acest lucru se face de un numar mic de ori.

Unele ruterele care nu erau pornite sau care nu au fost identificate datorita pierderilor de pachete vor fi identificate la trimiterea lor periodica de anunturi.

Descoperirea mesajelor de catre un router nu reprezinta un protocol de rutare: el doar permite unei gazde identificarea sau descoperirea ruterilor vecini, dar nu analizeaza care ruter este cel mai bine sa fie “contactat” intr-o situatie anume. Daca o gazda alege pentru o anumita destinatie ca prim hop, un ruter ce nu este optim pentru acea ruta, gazda va primi de la ruter un mesaj ICMP Redirect, identificand o ruta mai buna.

Un mesaj de tip “anunt ” al unui ruter include un “nivel de preferinta” pentru fiecare adresa de ruter anuntata. Cand o gazda trebuie sa aleaga un ruter default, este de asteptat sa aleaga dintre acele adrese de rutere care au cel mai mare nivel de pregerinta. Un administrator

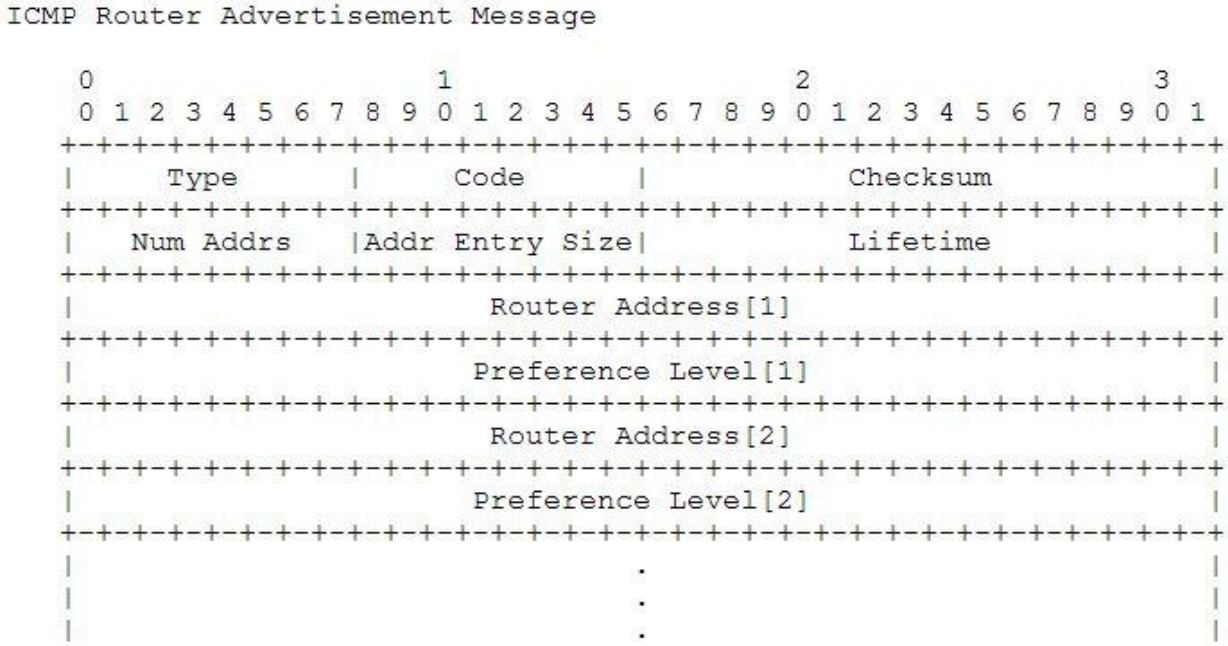
de retea poate configura nivelele de preferinta ale adreselor ruterelor pentru a incuraja sau descuraja folosirea unor rutere anumte ca rutere defaut.

Un “Anunt de ruter” contine si un camp numit “timp de viata” (lifetime), specificand durata maxima de timp in care acel anunt este considerat ca fiind valud, adica sa contina o adresa de ruter valida, in absenta altor mesaje. Acest lucru ne asigura ca gazdele ‘uita’ de ruterele care cad, devin inacccesibile sau care se opresc ca sa functioneza ca rutere.

Frecventa cu care mesajele de anunturi sunt difuzare in retea este de 7-10 minute, si durata de viata este de 30 de minute. Asta inseamna ca, prin folosirea valorilor implicite ale anunturilor nu sunt suficiente ca mecanism de detectie a gaurilor negre – detectia unei erori pe primul hop al unei cai- ideal aceste gauri negre ar trebui sa fie detectate destul de repede pentru a schimba la alt ruter inainte ca conexiunile de transport sau sesiunile de nivel mai mare sa expire. Este presupus ca gazda are mecanism de detectie a gaurilor negre. Gazdele nu pot depinde de anunturile ruterelor pentru indeplinirea acestui scop, datorita faptului ca acele rutere pot fi oprite sau incacesibile.

Formatul Mesajelor

ICMP Router Advertisement Message



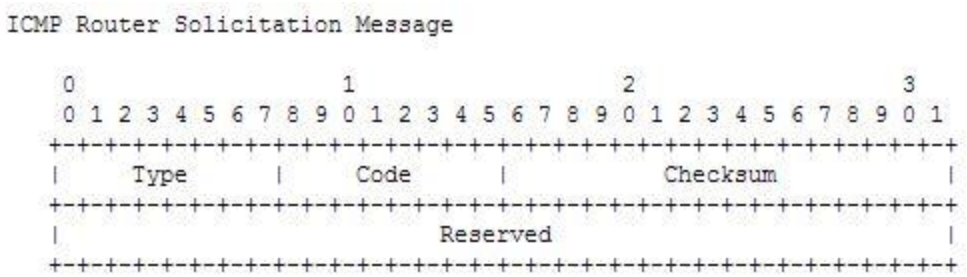
Campurile IP:

- **Source Address :** Reprezinta adresa IP a interfetei care a trimis mesajul
- **Destination Address:** Adresa hostului vecin
- **Time-to-Live:** 1 daca adresa destinatie este o adresa IP multicast, si cel putin 1 altfel

Campurile ICMP:

- **Type: 9**
- **Code : 0**
- **Checksum**
- **Num Addrs:** Adresa ruterului anuntat in acest mesaj
- **Addr Entry Size:** Contine 32-biti de cuvinte de informatie pentru fiecare adresa de ruter
- **Lifetime :** Numarul maxim de secunde pentru care adresa ruterului poate fi considerata valida
- **Router Address[i] :** adresa IP a ruterului care transmite mesajul
- **Preferente Level[i] :** nivelul de preferinta a fiecarei adrese de ruter.

ICMP Router Solicitation Message



Bibliografie

1. Cisco.Press.Internetworking.Technologies.Handbook.Fourth.Edition

2. Tanenbaum, Andrew S. - Computer Networks - 4th Edition

3. *A Reverse Address Resolution Protocol*, R. Finlayson, T. Mann, J. Mogul, M. Theimer (June 1984)

4. http://www.whatismyip.com/ip-faq/what-is-dhcp/

5. Ralph Droms; Ted Lemon (2003). *The DHCP Handbook*

6. http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

7. Colton, Andrew. *OSPF for Cisco Routers*

8. Moy, John T.. *OSPF: Anatomy of an Internet Routing Protocol*

9. Berkowitz, Howard (1999). "OSPF Goodies for ISPs

10. http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol#Control_messages

11. Retele de calculatoare –Principii Radu-Lucian Lup_sa

12. http://www.ietf.org/rfc/rfc792.txt

13. http://www.ietf.org/rfc/rfc1256.txt