Reţele de calculatoare / Reţele si servicii

Curs 1

Tehnologia de baza a secolului nostru este reprezentata de colectarea, prelucrarea si distribuirea informaţiilor. Reţelele de calculatoare, privite drept o colecţie de sisteme de calcul si de dispozitive interconectate prin canale de comunicaţie, permit schimbul de date intre utilizatori. Costurile scăzute, avantajele oferite si uşurinţa de utilizare au făcut ca la începutul anului 2011 sa existe nu mai puţin de 2 miliarde de utilizatori internet conform statisticilor Organizaţiei Naţiunilor Unite.

[http://news.yahoo.com/s/afp/20110126/ts_afp/untechnologytelecominternetmobile_20110126143340 Number of Internet users worldwide reaches 2 bln: UN (Agence France-Presse)]

ISTORIC RETELE DE CALCULATOARE

Apariţia reţelelor de calculatoare a fost posibila datorita progresului tehnologic si a intrepatrunderii intre domeniul comunicaţiilor si cel al calculatoarelor. Reţeaua de calculatoare leagă între ele o mulţime mai mică sau mai mare de calculatoare, astfel încât un calculator poate accesa datele, programele şi facilităţile unui alt calculator din aceeaşi reţea. De obicei este nevoie desigur şi de măsuri de restricţie/siguranţă a accesului.

Începutul reţelelor de calculatoare se afla in timpul războiului rece (anii ’60), când Departamentul Apărării al Statelor Unite ale Americii a început finanţarea agenţiei Advanced Research Projects Agency Network (ARPANET) care a avut drept unul din scopuri construirea unei reţele de comunicaţie bazata pe principii noi fata de reteaua folosita in telefonie. Acest tip nou de reţea permitea o robusteţe deosebita chiar si in cazul in care unul sau mai multe din elementele componente ii erau afectate. Totodată structura reţelei era una ierarhica, lucru care scădea complexitatea reţelei, permitea selectarea diferenţiata a echipamentelor si funcţionalităţii pentru fiecare strat si creştea eficienta in luarea deciziilor de transmitere a traficului pe trasee diferite.

Pornind de la reţeaua iniţiala, prin modem, implementata in Statele Unite ale Americii, tehnologia a fost extinsa in tot mai multe tari, iar viteza de comunicaţie a crescut continuu pentru a face fata cerinţelor societăţii informaţionale din prezent.

Pentru comunicarea intre diferite entităţi in general (si in particular intr-o reţea de calculatoare) este necesara existenta un protocol de comunicaţie. Un **protocol de comunicaţie** este o descriere formala a structurii mesajelor schimbate intre cele doua entităţi si a regulilor pentru schimbul acestor mesaje. Un protocol descrie sintaxa, semantica si sincronizarea comunicaţiei, putând fi implementat in hardware, software sau mixt. In reţele de calculatoare protocoalele pot sa definească modalitatea de accesa la mediul prin care sunt transmise datele, vitezele de transfer, etc.

Protocoalele folosite in retele de calculatoare au in general o structura similara desi pot sa indeplineasca roluri foarte diferite.

Complexitatea schimbului de date in reţe-lele de calculatoare este insa foarte de mare, atât la nivel hardware cat si la nivel software, deci este imposibila dezvoltarea unui singur protocol care sa definească toate aspectele acestei comunicaţii. Din acest motiv a fost necesara dezvoltarea unei structuri pe niveluri, in care mai multe protocoale lucrează

împreuna, fiecare îndeplinind un rol la un anumit nivel, pentru a îndeplini toate sarcinile transmisiei informaţiei.

Aceste protocoale au fost grupate in modele de comunicaţie, care prezintă detaliat modul in care diferite protocoale interacţionează pentru implementarea comunicaţiei pornind de la nivelul aplicaţiei software folosita de utilizator pana la nivelul hardware in care informaţia este transformata in semnale (electrice, optice, etc.) pentru a putea fi transmisa. Cele mai cunoscute modele de comunicatie sunt OSI (Open Systems Interconnection) si TCP/IP (Transmission Control Protocol/Internet Protocol).

In acest proces de dezvoltare a comunicaţiilor in reţele de calculatoare numeroase organizaţii pentru standardizare sunt implicate, dintre care putem aminti:

- Internet Engineering Task Force care este principala organizaţie de standardizare in domeniul internetului la nivelul serviciilor si protocoalelor;
- The International Organization for Standardization (ISO)
- The Institute of Electrical and Electronics Engineers (IEEE)
- The American National Standards Institute (ANSI)
- The International Telecommunication Union (ITU)
- The Electronics Industry Alliance/Telecommunications Industry Association (EIA/TIA)
- National telecommunications authorities such as the Federal Communication Commission (FCC) in the USA.

IETF de exemplu creează grupuri de lucru care dezbat standardele dar si alte probleme legate de acest domeniu si in urma discuţiilor publica documentele finale numite RFC (Request for Comments). Dintre acestea unele au valoare speciala si constituie standarde internet in vreme ce altele sunt doar recomandări pentru implementarea corecta a tehnologiilor internet.

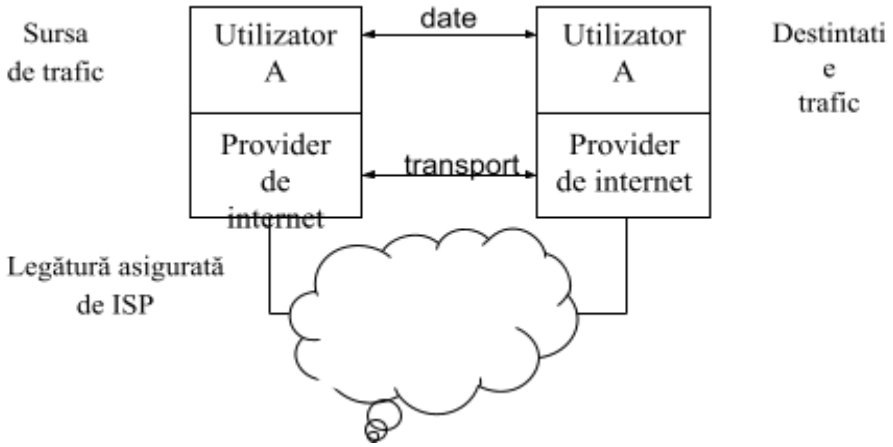
COMUNICATIA LA NIVELUL INTERNETULUI

Comunicaţia la nivelul internetului poate fi prezentata prin analogie cu transmiterea prin posta clasica a unei scrisori. Persoana care doreşte sa trimită un mesaj va scrie o scrisoare (*date*) pe care o va închide intr-un plic (*încapsulare* date) pe care trebuie sa menţioneze atât adresa destinatarului cat si adresa sursei astfel incat scrisoarea sa poată sa ajungă la destinaţie. Plicul este depus la oficiul poştal, moment in care se poate alege printre altele daca transmiterea se va face normal sau rapid si daca se solicita primirea unei confirmări ca mesajul a fost transmis. In acest moment posta (sau firma de curierat) este cea care preia scrisoarea si o sortează (*rutare*) pentru a o putea trimite in mod corect in tara/localitatea in care se afla destinaţia. Scrisoarea este in final recepţionata de destinatar care o desface (*decapsulare* date) pentru a putea citi mesajul (*date*). Au fost folosiţi in paranteza câţiva termini specifici reţelelor de calculatoare pentru a explica pe scurt semnificaţia acestora.

Pornind de la exemplul enunţat anterior, vom prezenta un exemplu de comunicaţie prin internet, aşa cum este perceputa de utilizator:

In aceasta figura utilizatorul A doreşte sa comunice cu utilizatorul B. In acest scop cei doi utilizatori trebuie sa apeleze in general la serviciile unui furnizor de internet (numit de obicei si ISP - **I**nternet **S**ervice **P**rovider) deoarece conectarea directa este in general mult prea costisitoare sau impractic de realizat. Furnizorii de servicii internet oferă suportul pentru transferul informaţiei in mod *transparent* pentru cei doi utilizatori (nu este necesata cunoasterea detaliilor tehnice legate de modul in care se r ealizează schimbul de date), in marea majoritate a cazurilor percepându-se pentru serviciile oferite o anumita taxa. Pentru conectare cei doi utilizatori nu trebuie decât sa aibă un calculator pentru a se conecta si o aplicaţiei rulând pe acestea care sa le permită schimbul de date.

Aceasta figura poate fi redesenata astfel incat sa evidenţieze nivelurile la care se efectuează schimbul de informaţii.



Din punct de vedere al dimensiunii retelele de calculatoare se pot clasifica in retele LAN –Local Area Network si retele WAN (Wide Area Network).

Retelele LAN au drept standarde dominante sunt Ethernet şi WLAN (IEEE 802.11). Separaţia (conectarea) între LAN şi MAN/WAN se realizează cu un ruter (gateway). Securitatea este controlata la nivel centralizat, de obicei de catre administratorul retelei.

In retelele WAN –Wide Area Network sunt utilizate numeroare protocoale precum: MPLS, ATM, Frame Relay, PPP. Ele sunt compuse din numeroase retele care au propria politica de securitate lucru care face imposibila stabilirea centralizarea acetora.

Modele pentru schimbul de date in retele de calculatoare

Schimbul de date intre doua echipamente intr-o retea de calculatoare poate fi diferentiat in functie de rolul pe care il au participantii al schimbul de date. Din acest punct de vedere al partitionarii sarcinilor si al timpului de lucru in retea, se pot distinge urmatoarele modele:

-client-server

-peer to peer

Modelul client-server este cel al unei aplicatii distribuita care partajeaza sarcinile intre asiguratorii de resurse (servere) si cei care solicita aceste resurse (clientii). Clientul si serverul sunt situati in general pe calculatoare diferite si comunica prin retea, insa este posibil ca acestia sa se afle pe aceeasi masina.

Serverul este un echipament care ruleaza unul sau mai multe programe care isi partajeaza resursele in retea. Resursele sunt variate pornind de la putere de calcul pana la fisiere sau dispozitive periferice. Functii precum posta electronica, accesul la pagini web sau accesul la baze de date se bazeaza pe modelul client-server.

Clientii nu isi partajeaza resursele si apeleaza datele sau functiile serverului.

In majoritatea cazurilor clientii sunt entitati active deoarece genereaza cererile catre server, iar serverul este entitate pasiva deoarece asteapta sa serveasca clientii. In unele situatii pot exista servere active care cauta clientii in retea, insa acest lucru pune o incarcare puternica, nedorita, pe echipamentul server din punct de vedere al utilizarii resurselor acesuia de calcul si de conectivitate.

Modelul client-server are la bază un protocol simplu, fără conexiune de tipul întrebare-răspuns. In consecinta, interactiunea intre server si client se poate descrie prin diagrame de secventa.

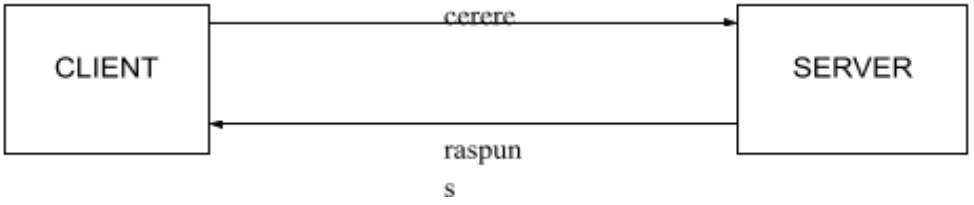


Figura 1.1 Modelul client-server

La implementarea modelului client-server se ţine seama de:
- adresarea proceselor server;
-tipul primitivelor utilizate în transferul mesajelor (sincrone/asincrone, cu/fără tampon, fiabile/nefiabile).

Clientul şi serverul se pot găsi în *acelaşi nod*, când se utilizează mecanisme de comunicaţie locală sau în *noduri diferite*, când se utilizează mecanisme de comunicaţie în reţea.

Modelul peer to peer este o arhitectura de aplicatie distribuita care imparte sarcinile in mod egal intre statiile (peer) care participa la schimbul de date. Aceste statii isi pun la dispozitie o parte din resursele lor in mod direct la dispozitia celorlalti participanti. Participantii la schimbul de date in acest model indeplinesc in acelasi timp si rolul de client dar si cel server de server prezentate anterior.

Acest model de retea poate avea ma multe variante:

- Retele peer to peer simple: nu exista nici un control la realizarea schimbului de date, toti participantii avand drepturi egale.
- Retele peer to peer hibride: exista anumite elemente in infrasctructura capabile sa acumuleze informatii despre utilizatorii prezenti pentru a facilita conectarea acestora.
- Retele peer to peer centralizate: exista servere care gestioneaza conetarea participantilor si indexarea resurselor insa acesta nu poate controla momentul sau datele care sunt accesate.

Daca vom realiza o comparatie intre modelele peer to peer si client – server, vom observa ca o retea client-server are de obicei mai multi clienti conectati la un singur server. Serverul are una sau mai multe caracteristici superioare clientilor (capacitate de stocare, viteza de procesare, dimensiune memorie) pe care le pune la dispozitia clientilor. In retelele peer to peer resursele partajate sunt cele ale unui calculator obisnuit: unitati de stocare a datelor, echipamente periferice.

Avantajul retelelor peer to peer este acela ca nu necesita instalarea de software specializat, costurile sunt reduse prin reutilizarea dispozitivelor periferice sau a aplicatiilor software de catre toti participantii la retea. Dezavantajele retelelor peer to peer privesc securitatea care nu poate fi controlata dintr-un punct central ci doar la nivelul fiecarui calculator, resursele care trebuie partajate intre utilizatorul local si cel din retea si dificultatea realizarii conexiunilor pentru arhitecturi complexe de retea.

Retelele client server au un cost initial mai mare de instalare si configurare insa imbunatatirea acestora se realizeaza usor deoarece doar serverul este cel a carui stuctura trebuie modificata. Totodata asigurarea disponibilitatii retelei si a securitatii datelor se poate face rapid si centralizat in retelele client-server deci se preteaza pentru implementarea in institutii in care aceste caracteristici sunt prioritare. Trebuie avuta in vedere insa proiectarea corespunzatoare a serverului pentru a putea face fata tuturor cererilor clientilor.

Pentru ca o firma sa isi dezvolte solutia IT este necesara o investitie prealabila in infrastructura (calculatoare, echipamente de retea, etc.), de pregatirea personalului care va administra aceasta retea si de achizitionarea de licente software. Deoarece in prezent firmele doresc sa aiba o crestere cat mai rapida si sa fie cat mai flexibile pentru a se putea adapta mai usor conditiilor si evolutiilor pietei, a crescut nevoia pentru implementarea unui model in care costurile legate de investitiile hardware (atat pentru extindere cat si pentru actualizare) sa fie cat mai reduse. Din acest motiv a aparut un nou model numit **cloud computing**. Acesta este modelul prin care resursele partajate, atat hardware cat si software sunt oferite drept un serviciu care poate fi achizitionate de catre utilizatori pentru perioada dorita, avand drept mediu de distributie internetul.

Oferirea atat a aplicatiilor cat si a infrastructurii ca un serviciu permite controlul mai eficient al accesului la resursele oferite. Accesarea de exemplu a serviciilor software se face prin intermediul browserelor de internet ca si cand aplicatiile ar rula pe calculatoarele locale.

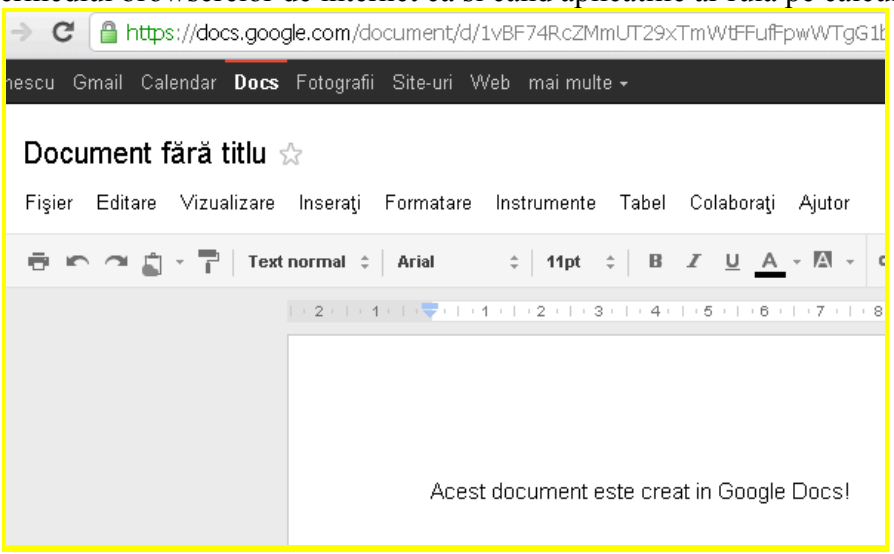


Fig. Aplicatiile cloud precum Google Docs au devenit tot mai raspandite

Resursele sunt virtualizate, lucru care asigura disponibilitatea acestora. Virtualizarea reprezinta tehnica de creare a unei versiuni software (virtuale) a unui sistem de operare, a unei resurse hardware sau a unei resurse pentru retele de calculatoare. Scopul virtualizarii este

realizarea unui sistem unitar de administrare centralizata, asigurarea scalabilitatii sistemului si utilizarea eficienta a puterii de calcul.

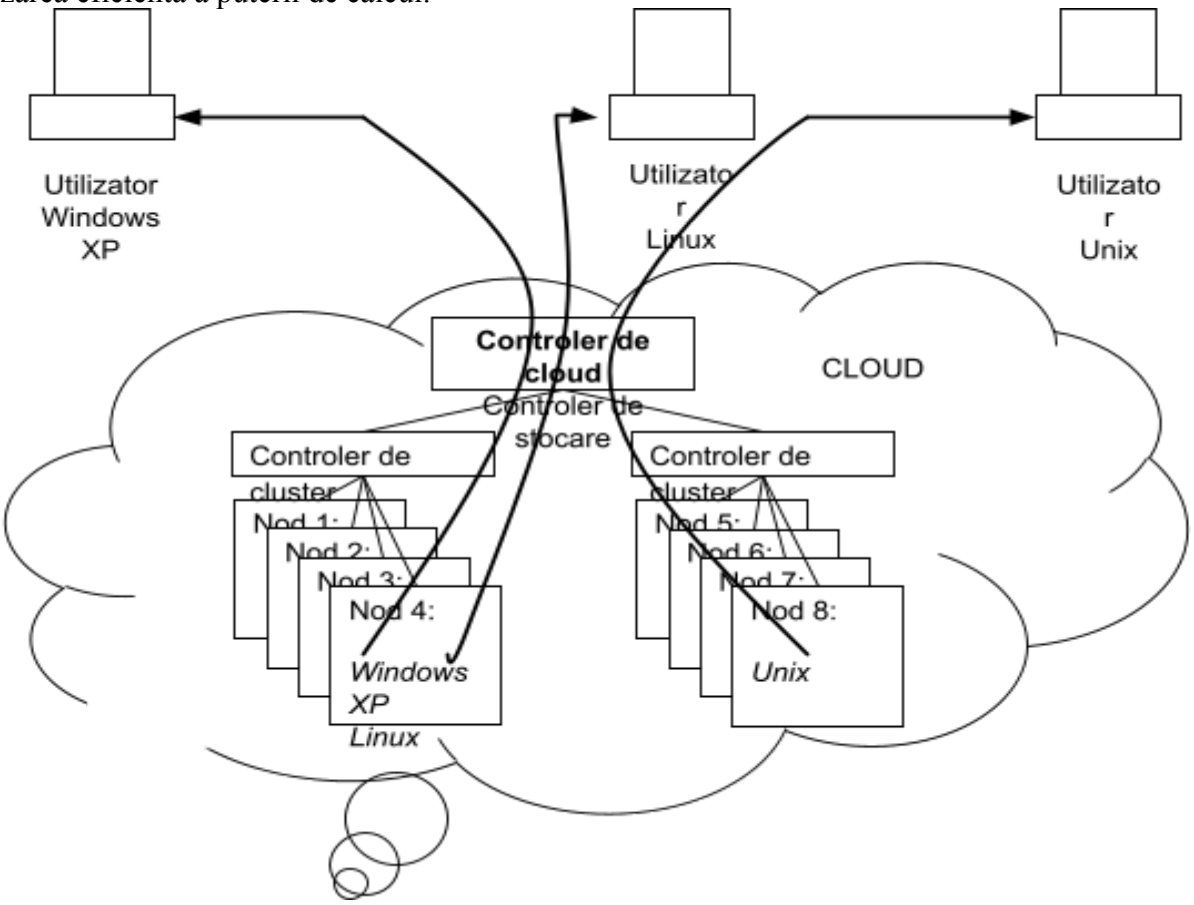


Fig. Virtualizarea sistemelor de operare intr-un cloud: utilizatorii sistemelor de operare Windows XP si Linux din figura utilizeaza acelasi echipament hardware din datacenter fara a cunoaste acest lucru

[https://help.ubuntu.com/community/UEC/CDInstall]

De exemplu virtualizarea resurselor hardware implica crearea uneia sau mai multor masini virtuale ruland diferite sisteme de operare care ruleaza insa un hardware cu sistem de operare sau structura hardware diferita. In acest caz nu este importanta localizarea fizica a echipementului server care poate utiliza una sau mai multe masini si poate migra de pe un echipament hardware pe altul in functie de cerinte. Esential este ca infrastructura este concentrata intr-unul sau mai multe locatii data center care sunt controlate prin sisteme de operare specializate. Un data center adaposteste numeroase sisteme de calcul si resursele hardware asociate organizate in scopul asigurarii disponibilitatii neintrerupte si a integritatii datelor. Operarea unui data center se realizeaza centralizat, cu numeroase programe care automatizeaza sarcinile uzuale iar singurele interventii sunt de natura exceptionala si privesc efectuarea de reparatii sau modificari la echipamente.

Numeroase tehnologii au contribuit la realizarea acestui model si prezinta caracteristici comune cu acesta. Astfel:

- tehnologia Grid – a fost proiectata pentru a servi aplicatii avand nevoie de mare putere de calcul prin crearea de statii cu mare putere de calcul folosind ca elemente componente statii mult mai ieftine. Aceasta tehnologie insa nu oferea simplitate si suport pentru automatizarea sarcinilor in interiorul grid-ului;

- virtualizarea: trecerea unor infrastructuri in forma virtuala permitea revenirea rapida la o stare de functionare in cazul in care apareau probleme, insa acest proces nu era auomatizat si necesita interventia utilizatorului uman.
- Hosting: serverele folosite pentru stocrea de date nu ofera scalabilitatea necesara in functie de cererile existente la un anumit moment, acest lucru constituindu-se intr-un dezavantaj fie prin alocarea de resurse neutilizate, fie prin lipsa de resurse.
- SaaS (Software as a Service): acestea sunt aplicatii software care ofera servicii in functie de cerere. Ele insa nu pot oferi hardware in functie de cerere, lucru posibil in cazul unui cloud.

Schimbul de informatie intre utilizatori se poate desfasura dupa unul din urmatoarele **modele de comunicatie**:

- **Comunicatia orientata pe conexiune**: este un mod de comunicare in care dispozitivele care participa la schimbul de date stabilesc o legatura (conexiune) fizica sau logica inainte de transmiterea datelor. Acest mod de comunicare asigura de obicei confirmarea receptiei datelor in cazul receptoinarii acestora corect si retransmit in mod automat datele in cazul in care se depisteaza erori. In cazul in care conexiunea permanenta stabilita este dedicata se reduce traficul necesar cu identificarea sursei si a destinatiei insa apare problema inflexibilitatii acestuia, deoarece tot canalul de comunicatie este rezervat pentru respectiva conexiune si nu poate fi utilizat de alte conexiuni (de exemplu in momentele de pauza intre schimburile de date). In cazul conexiunilor logice toate datele sunt transmise pe acelasi traseu in timpul unei sesiuni de lucru si doar primele pachete de date sunt insotite de informatii pentru identificarea sursei si a destinatiei traficului. Pachetele ulterioare sunt insotite doar de informatii care identifica canalul virtual de comunicatie stabilit.
- **Comunicatia fara conexiune**: este un mod de comunicare in care datele utilizatorilor sunt transmise in forma de pachete intre participantii la schimbul de date fara a se realiza in prealabil o semnalizare a acestui lucru. Din acest motiv nu se garanteaza receptia datelor (de exemplu echipamentul destinatie pentru traficul de date poate sa nu fie gata de receptie la momentul sosirii datelor), modul de lucru purtand denumirea de *best effort*. Pachetele de date trebuie insotite de informatii de adresare (privind echipamentul sursa si destinatie al traficului de date) deoarece circula independent unul de altul prin reteaua de calculatoare.

Cum se desfasoara insa o comunicaţie intr-o reţea de calculatoare comparativ cu alte tipuri de retele de comunicatie?

Clasificare METODE DE COMUTAŢIE

Pentru ca datele sa ajungă de la sursa la destinaţie intr-o reţea de dimensiuni mari este necesara luarea unor decizii legate de traseul pe care vor fi transmise acestea. Exista in prezent mai multe tehnologii de implementarea a modului in care se iau aceste decizii:

- comutatia de mesaje;
- comutaţia de circuite;
- comutaţia de pachete;
- comutatia de celule.

Comutatia de mesaje este tehnica folosita la inceputurile comunicatiei de date care implica stocarea mesajelor pe un anumit mediu de stocare apoi transmiterea ulterioara a acestora. Comunicarea in timp real nu este o conditie necesara. Un exemplu de astfel de sistem este cel de posta electronica (e-mail) sau de mesagerie pentru comunicatiile de voce.

Comutaţia de circuite presupune stabilirea inaintea transmiterii datelor a canalului de comunicaţie de la sursă la destinaţie, bazat pe un algoritm de optimizare a utilizării resurselor, care se păstrează pe toată durata comunicaţiei. Pentru toata durata schimbului de date intre cele doua entităţi, traseul dintre acestea este dedicat si exclusiv, iar eliberarea sa se face doar când sesiunea de comunicaţie a fost terminata. Aceasta tehnologie de comutatie a fost folosita pentru comunicatiile telefonice.

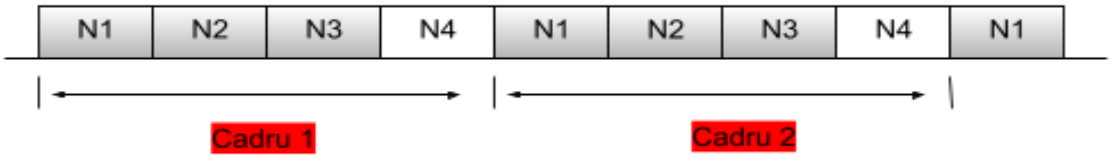
Multiplexarea conexiunilor prin acelasi mediu de transmisie s-a realizat prin tehnica TDM (Time-Division Multiplexing – Multiplexare cu diviziune in timp).

TDM este o metoda de multiplexare in care doua sau mai multe fluxuri de date provenind de la *un nod sursa de date* sunt transferate aparent simultan prin acelasi canal de comunicatie catre *un nod numit destinatie*, in practica ele accesand succesiv canalul de comunicatie. Pentru a realiza acest lucru, timpul este divizat in mai multe intervale de durata fixa, cate unul pentru fiecare canal, astfel incat sa poata fi transmise date succesiv din fiecare flux de date, iar dupa ce fiecare flux a realizat o transmisie, procesul se reia de la inceput. Un bloc de date provenind de la primul flux este transmis in primul interval de timp, un bloc de date din al doilea flux si asa mai departe, dupa cum se observa in figura urmatoare:

FIGURA

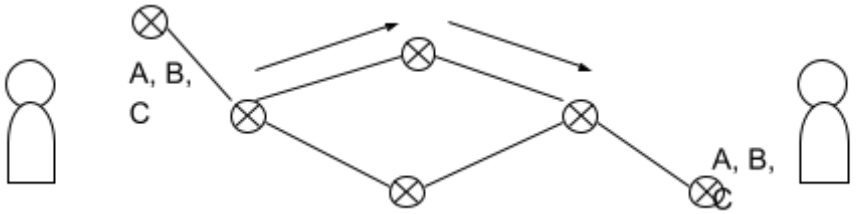
In acest mod numarul de canale este fix iar rata maxima de transfer este constanta pentru fiecare canal (Un flux de date de 64 kbps este standardul pentru telefonia digitală (DS0), derivat din teorema de eșantionare a lui Nyquist; un semnal vocal analogic de 4 kHz este eșantionat de 8.000 de ori pe secundă -de două ori frecvența maximă- și digitalizat la 8 biți pe eșantion, asigurând o reproducere eficientă.). Comutatia TDM este utilizata in comutatia de circuite fizice si cea de circuite virtuale. Circuitele virtuale se refera la acele conexiuni care folosesc la nivel logic aceasta tehnica de comutatie, desi la nivel fizic se folseste o tehnica diferita de comutatie. Acest lucru permite protocoalelor de nivel inalt sa aiba la dispozitie o conexiune permanenta intre sursa si destinatie si sa evite diferite probleme de natura fizica, insa in majoritatea cazurilor nu se mai garanteaza caracteristici precum rata de biti sau timpul necesar transmiterii informatiei de la sursa la destinate (latenta).

TDMA: time division multiple access este un caz particular al TDM in care exista mai multe surse (noduri) care trebuie sa transmita prin acelasi canal de comunicatie. Accessul la canalul de comunicatei se face pe rand, fiecare nod primind un slot de lungime fixa in care poate sa transmita. Sloturile de timp nefolosite de catre surse nu sunt reutilizate. TDMA este folosit in sistemele de telefonie celulara 2G. In exemplul urmator din patru noduri prezente in retea, doar trei au pachete de transmis (N1, N2, N3), in timp ce nodul patru nu are pachete de transmis.



Un flux de date TDM nu necesita identificatori pentru fiecare slot, deoarece numarul acestora este cunoscut si fiecare are o dimensiune fixa, prin urmare la destinatie se realiezaza aceeasi divizare si se extrag pe rand datele din sloturile corespunzatoare. Erorile sunt detectate şi corectate de fiecare canal în parte. Aceasta metoda de multiplexare este ineficienta deoarece sloturile de timp sunt alocate permanent, indiferent daca se ele sunt sau nu folosite.

Acest mod de comunicaţie este foarte sensibil la întrerupere. Odată intrerupt circuitul fizic sau virtual, conexiunea se întrerupe iar schimbul de date va fi reluat abia dupa restabilirea acestuia.



Etapele comutaţiei de circuite sunt următoarele:
- stabilirea circuitului: este etapa de formare a numărului in telefonie;
- realizarea comunicaţiei: schimbul efectiv de date;
- închidere circuit: terminarea conexiunii.

In figura se observa cum traseul stabilit (reprezentat punctat) pentru desfăşurarea schimbului de date la deschiderea conexiunii este păstrat pana la terminarea acesteia. Acest lucru face posibil schimbul de date atât in format digital cat si in format analogic.

IMAGINE INGINERIE TRAFIC CIRCUIT SWITCHING

Daca vom considera o linie de comunicaţie este de tipul T1 (cu o viteza de 1,544Mb/s) sau E1 (cu o viteza de 2,048Mb/s). Ştiind ca pentru a transmite semnalul de voce pentru un apel telefonic sunt necesari 64kb/s, cele doua linii sunt suficiente pentru aproximativ 25 de convorbiri telefonice simultane, respectiv 32 convorbiri. In prezent liniile performante pot oferi viteze de 10Gb/s, insuficiente pentru oraşele de dimensiuni mari.

Totuşi aceasta tehnologie este folosita in continuare deoarece este folosita de mult timp si ii sunt cunoscute toate posibilele probleme. Pentru comunicaţia prin internet a fost insa necesara o tehnologie nou, comutaţia de pachete.

Pentru a putea înţelege *comutaţia de pachete* este necesara înţelegerea conceptului de *pachet*. De multe ori este necesara folosirea aceleiasi linii de comunicatie de catre mai multe transmisii simultane. Din acest motiv fluxul de datele este împărţit in cantităţi mai mici, numite pachete, si sunt transmise individual. Acest lucru permite reutilizarea liniei de comunicatie de catre mai multe echipamente. Pentru ca pachete diferite sa poata fi identificate ca făcând parte din acelaşi grup de date, este necesara însoţirea acestora de catre catre informatii pentru identificarea susei si destinatiei, a ordinii de transmisie, etc.

IMAGINE INGINERIE TRAFIC PACKET SWITCHING

Prin urmare un pachet are doua componente:

A) ***datele utilizator***: formatate intr-un mod care sa le permită identificarea lor corecta atât la sursa cat si la destinaţie.

Majoritatea protocoalelor de comunicaţii transmit datele începând cu partea cea mai semnificativa (la nivel de octet sau bit) acest mod de transmitere purtând denumirea de big-endian. In reţelele de telefonie cea mai semnificativa parte a mesajului conţinea prefixul telefonic reprezentat printr-un grup determinat de cifre care corespunde unei anumite localităţi sau regiuni geografice din cadrul statului respectiv. Aceasta permitea determinarea destinaţiei numărului de telefon înainte ca acesta sa fie format integral. Protocolul internet defineşte big-endian ca reprezentarea standard in reţele de calculatoare pentru valorile numerice din antetul pachetului. Pentru ca sistemele de calcul folosind reprezentarea little-endian (in care partea semnificativa a octetului sau bitul cel mai semnificativ este ultimul), sa poata transmite sau recepţiona datele in reprezentarea big-endian ele trebuie sa transforme datele folosind functii dedicate precum:

- htonl (host-to-network-long) si htons (host-to-network-short) care transforma valorile pe 32 respectiv 16 biti din ordinea little endian in big endian;
- ntohl si ntohs care transforma valorile pe 32 respectiv 16 biti din ordinea big endian in little endian.

De exemplu sistemele de calcul Intel x86 folosesc reprezentarea little endian. Sistemele de calcul care folosesc intern reprezentarea big-endian nu trebuie sa realizeze nici o transformare pentru a transmite datele in retea.

B) ***informaţia de control***: asigura datele necesare reţelei pentru a putea trimite datele utilizator la destinaţie: informaţii pentru identificarea sursei si a destinaţiei, un număr de ordine pentru a putea reasambla datele in ordinea in care au fost transmise, informaţii de detecţie a erorilor, precum si alte informaţii de control.

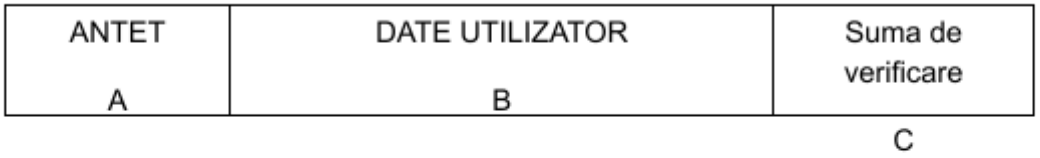
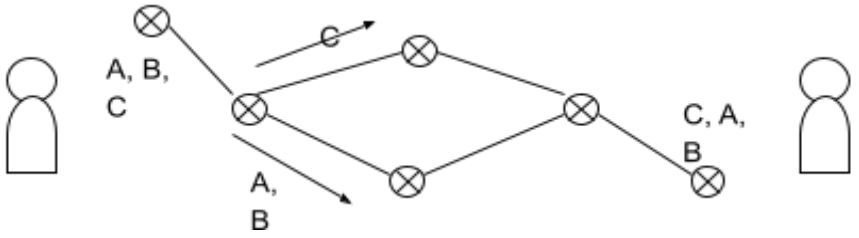


Fig. Componentele folosite de obicei pentru compunerea unui pachet de date sunt ordonate astfel: A) informaţia de control cu privire la adresare, numar de ordine, alti indicatori; B) datele utilizator; C) alta parte a informaţiei de control de obicei in scopul verificarii corectitudinii pachetului transmis.

Comutaţia de pachete este folosită în reţele de calculatoare deoarece deciziile de direcţionare (rutare) a traficului se efectuează la nivel de pachet, in mod individual. Acest lucru implică necesitatea folosirii bufferilor de recepţie deoarece pachetele individuale pot sosi în ordine diferită faţă de trimitere, daca exista trasee alternative de la sursa la destinaţie.



In cazul in care pachetele nu includ in antet informaţii pentru reasamblare ordonata (in cazul in care ordinea pachetelor la recepţie nu este aceeaşi cu ordinea de la transmisie) sau informaţii pentru retransmitere (in cazul in care unele pachete sunt pierdute) se foloseşte termenul de *datagrama*. Datagramele conţin insa informaţiile pentru detecţia erorilor.

De exemplu in figura anterioara, daca s-ar transmite datagrame, in cazul in care traseul folosit de datagrama C s-ar întrerupe iar datagrama s-ar pierde, acest lucru nu va fi semnalizat in nici un fel către sursa de trafic. Daca s-ar transmite pachete, atunci pachetul pierdut ar fi detectat prin lipsa numărului de ordine corespunzător la destinaţie, iar aceasta va semnaliza sursa pentru a retrimite pachetul.

Daca traficul de date ar sosi in reţelele de calculatoare cu o viteza constanta, atunci ar fi uşor de proiectat echipamente care sa servească un număr determinat de utilizatori. Traficul în reţea soseşte la intervale de timp inegale, iar atunci când soseşte aceasta se face în rafale. Vitezele variază frecvent deoarece diferite tipuri de date necesita tipuri de fluxuri diferite iar vârfurile de trafic nu coincid datorita imposibilitatii sincronizării traficului. Aceste caracteristici aparent dezavantajoase au trebuit sa fie utilizate intr-o tehnica de comunicaţie care le transforme in avantaje.

Cu cat mai mulţi utilizatori folosesc acelaşi canal de comunicaţie, cu atât şansele ca ei sa folosească conexiunea la momente diferite de timp cresc. Prin urmare, cu cât sunt transmise simultan mai multe fluxuri de comunicaţie cu atât ele pot fi rearanjate mai bine astfel incat momentele in care anumite conexiuni nu sunt utilizate (sau utilizate la un nivel scăzut) sa compenseze timpul in care alte conexiuni devin active.

MULTIPLEXAREA STATISTICĂ

Multiplexarea statistica cu diviziune in timp (STDM - Statistical time-division multiplexing) este o forma avansata a TDM in care atat adresa terminalului cat si datele sunt trasmise impreuna pentru o mai buna directionare a traficului.

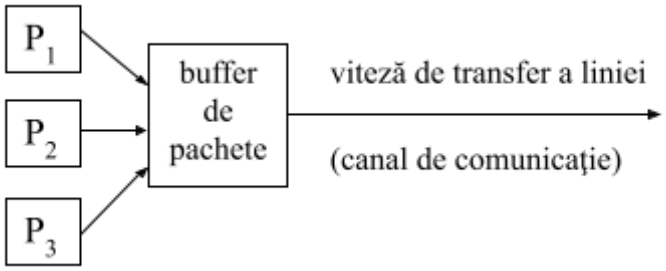
Aceasta este numita si TDM asincron (in vreme ce metoda traditionala cu sloturi de timp fixe se numeste si TDM sincron) deoarece aloca sloturile de timp in mod dinamic, adaptandu-se la cerintele instantanee de trafic ale fluxurilor de date care sunt transportate prin retea. Rata de transfer rezultata este mai mica decat suma ratelor datelor de intrare oferind castigul de multiplexare statistica.

Multiplexarea statistica implica oferirea canalului de comunicatie la cerere, fluxurile fiind servite in ordinea primul sosit, primul servit.

In figura urmatoare este prezentat modelul unui echipament de comunicatie numit router, care are rolul de a *multiplexa* (combina) mai multe fluxuri de comunicaţie care sosesc in echipament catre un canal de iesire folosnd tehnica STDM.

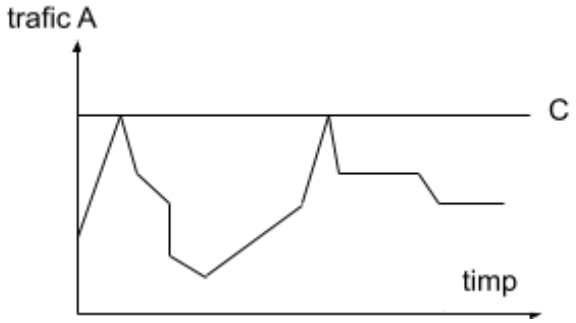
Caracteristicile acestui echipament sunt:

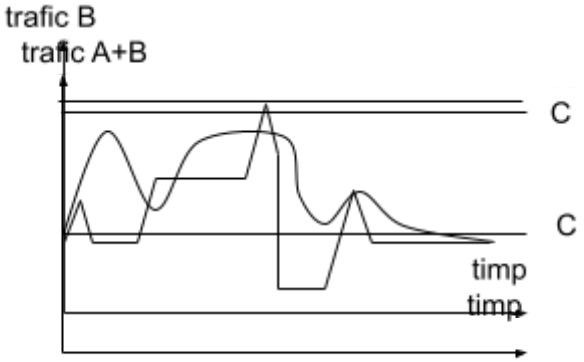
- viteză de transfer a liniei;
- canal de comunicaţie;
- fdsafesfes



Bufferul de pachete este necesar pentru a putea păstra traficul sosit simultan de la mai mulţi utilizatori pana la momentul in care dispozitivul poate sa il proceseze.

Vom reprezenta in continuare modul in care doua persoane utilizează internetul intr-un interval de o ora, fiecare având o conexiune de capacitate maxima C. Cei doi utilizatori folosesc internetul in scopuri diferite si la momente de timp diferite: primul utilizator accesează pagini de internet si citeşte materialele pe care le descărca, având numeroase momente in care nu utilizează conexiunea dar si momente in care o utilizeza intens; cel de al doilea utilizator asculta radio prin internet si urmăreşte videoclipuri, fapt care produce un trafic constat, cu momente de trafic crescut. In ultimul grafic este prezentata suma celor doua valori de trafic pentru a vedea daca sunt avantaje in combinarea acestora.

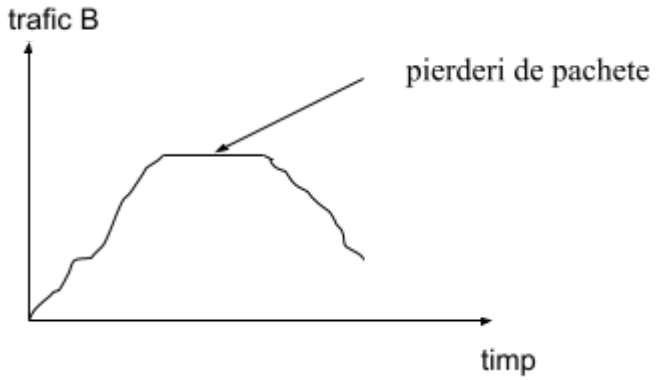




Deoarece buffer-ul poate sa stocheze pentru puţin timp pachetele, el va regulariza pentru scurt timp fluxul care intra in echipament reducând şansele ca acesta sa depăşească capacitatea sa de procesare sau sa depăşească capacitatea maxima a liniei de ieşire. Dezavantajul este ca apare o întârziere in comunicaţie, lucru care este detectat decât de aplicaţiile care necesita funcţionare in timp real (comunicaţii audio sau video). Avantajul major este ca linia de ieşire din router nu trebuie să aibă o capacitate egala cu suma capacităţii liniilor care intra in router.

Considerând R ca fiind rata de transfer, B lăţimea de banda a liniei de ieşire si N numărul de conexiuni care intra in router, atunci $B \leq R \cdot N$

Memoria unui buffer are totuşi o valoare limitată, iar daca traficul este susţinut la valoarea maxima de suficient de multe surse, atunci aceasta poate fi depăşită, rezultând pierderi de pachete.



Considerând că traficul A are valoare maximă C şi dacă traficul B are valoare maximă C, rezultă câştigul de multiplexare statistică este de $M = \frac{2C}{R}$ **M=2C/R** , în care R este viteza de transfer.

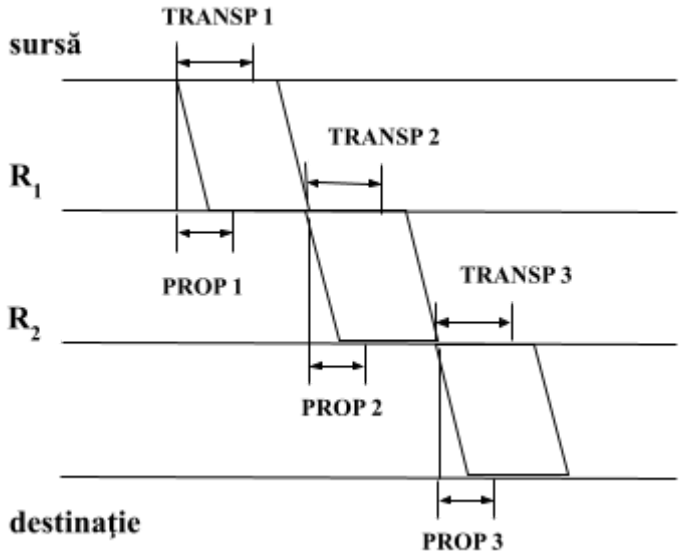
Modulaţia comutaţiei de pachete este dată de cost, eficienţă în liniile de comunicaţie, rezistenţa la cădere a liniei prin redirecţionarea pachetelor.

Consideram P ca fiind lungimea pachet exprimata în biţi, L lungimea traseului de date exprimata în metri si R rata de transfer exprimata în biţi pe secunda. Atunci putem defini:

- **întârzierea de propagare** ca fiind timpul necesar unui bit de parcurgere a liniei: **PROP=L/c**, in care c este viteza de propagare prin mediul de transmisie;
- **timpul de transmisie** este timpul necesar pentru a transmite un pachet de lungime P **TRANSM=P/R;** unde R- rata de transmisie(b/s)
- **latenţa** este timpul scurs de la momentul transmitereii primului bit pana la momentul recepţionării ultimului bit: **latenţa=PROP + TRANSP;**

Steiner -

Reprezentand grafic aceste valori



Prin urmare valoarea minima a latentei pentru transmiterea tuturor pachetelor este:

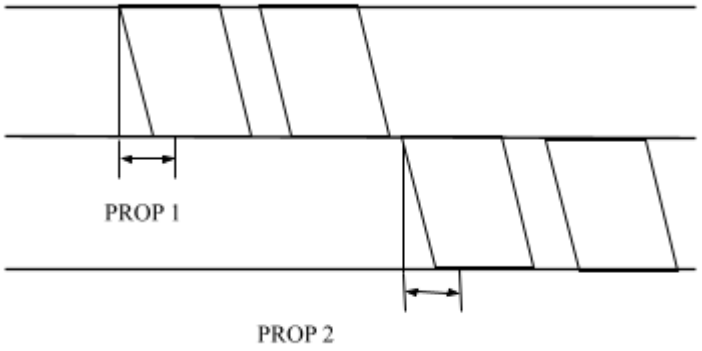
$$\text{Latenta} = \sum \text{PROP}_i + \text{TRANSM}_i$$

De multe ori insa datele care urmează a fi transmise prin reţea au o dimensiune foarte mare. Poate sa apara in aceste condiţii întrebarea: e*ste mai avantajos sa transmitem toate datele într-un singur pachet sau sa folosim pachete multiple de dimensiuni mici?

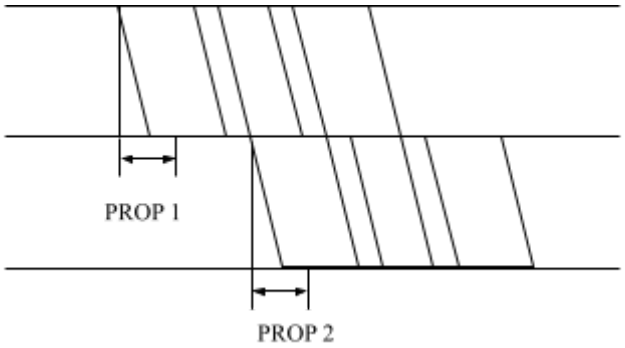
Aceasta situaţie este ilustrata comparativ in graficul următor, in care pentru situaţia in care se foloseşte un singur pachet de dimensiune mare latenta este de

$$Latenta_i = \sum (PROP_i + M/R_i)$$

deoarece am considerat un echipamentul intermediar nu poate sa retransmita pachetul catre destinatie decat in momentul in care acesta a fost receptionat integral (principiul de functionare store and forward).



Pentru situaţia in care se folosesc mai multe pachete de dimensiune mica, pentru aceeaşi cantitate totala de date, latenta este de: Latenta i =M/Rmin + ∑ PROP , o valoare mai mica decat in situatia anterioara.



In concluzie, împărţirea mesajului mare în pachete de dimensiune mai mica va duce la reducerea latenţei. In acelaşi timp, in situaţia in care linia de comunicaţie poate fi ocupata de un singur mesaj la un anumit moment de timp, pachete provenind de la alte comunicaţii pot fi intercalate intre doua pachete, evitând situaţia in care o singura transmisie blochează linia de comunicaţie.

Totuşi, in reţele de calculatoare, latenta nu este compusa doar din componente de valoare constanta. Uneori intr-un interval de timp scurt sau chiar in acelaşi timp la un echipament pot sosi mai multe pachete simultan, fiecare având destinaţii diferite. Echipamentele care direcţionează traficul trebuie sa calculeze pentru fiecare pachet destinatia insa, deoarece calculul nu se poate efectua simultan, unele vor fi procesate mai rapid si altele cu o anumita intarziere. Datele care urmeaza a fi procesate vor fi stocate temporar intr-o structura de tip coada. Întârzierea datorată cozii de aşteptare Q este singura variabilă în reţeaua de calculatoare.

Prin urmare formula de calcul a latentei se modifica in modul urmator:

$$latenta = \sum_{I} (transp_I + prop_I + Q_i)$$

IN CONCLUZIE

Multiplexarea STDM este foarte similara cu TDM insa nu lasa sloturile de timp sa fie transmise nefolosite. Capacitatea liniei va fi partajata doar de fluxurile care au date de transmis

Comutaţia de circuite creaza circuit (fizic sau virtual) dedicat conexiunii care asigura faptul ca toata informaţia va ajunge la destinaţie si, mai mult, va păstra ordinea de transmisie. Acest lucru face sa nu mai fie necesara transmiterea informatiilor de identificare a ordinii in care au fost transmisie partile componente ale fluxului pentru recompunerea corecta a acestora la destinatie, prin urmare fluxul de date nu va fi incarcat cu date suplimentare.

In acest mod se poate explica calitatea uneori mai slaba a comunicaţiilor VoIP (Voice over Internet Protocol) care folosesc drept suport comutatia STDM, comparativ cu telefonica clasica care foloseste TDM.

Din punct de vedere al costurilor comutaţia de pachete este mult mai ieftina, deoarece, pe de o parte, poate folosi un singur circuit pentru mai multe comunicaţii simultane si pe de alta parte costurile comunicaţiei sunt întotdeauna locale, in vreme ce la telefonie se pot aplica costurile de apel la distanta care sunt costisitoare.

In cazul in care pierderile de pachete si întârzierile nu sunt excesive, comunicaţiile de voce folosind comutaţia de pachete sunt preferabile telefoniei clasice datorita ocupării mai eficiente a echipamentelor datorita *multiplexării statistice*.

Comutatia de celule este folosita in retelele ATM (Asynchronous Transfer Mode) si este similara cu tipul de comutatie de pachete, insa comutatia nu are loc la terminarea pachetului ci dupa receptionarea unei anumite cantitati de date numita celula, care are dimensiunea bine determinata. Avantajul este ca un acest mod pot fi transmise semnale digitale atat de date cat si de voce.

IMAGINE INGINERIE TRAFIC CELL SWITCHING

Aceasta tehnica foloseste multiplexarea cu diviziune in timp - TDM (time-division multiplexing). Aceasta asigura trafic potrivit atat pentru comunicatiile de date de care necesita transferul unui volum mare de date cat si pentru comunicatiile de voce care necesita o lantenta mica. Latenta redusa este asigurata de este dimensiunea fixa a celulelor, care provine acapararea liniei de comunicatie pentru prea mult timp de catre o singura transmisie.

Modelul folosit de retelele ATM este cel orientat pe conexiune prin urmare este necesara stabilirea unui circuit (in acest caz este vorba de un circuit virtual) intre sursa si destinatie inainte de inceperea schimbului de date.

Conceptul de circuit virtual cuprinde conceptele de Cale Virtuala (Vitrual Path) si Canal Virtual (Virtual Channel). Fiecare celula ATM are in antent o pereche compusa dintr-un identificator de cale virtuala pe 8 sau 12 biti si un identificator de canal virtual pe 16 biti. Impreuna acestea pot identifica un circuit virtual folosit in realizarea conexiunii.

The length of the VPI varies according to whether the cell is sent on the user-network interface (on the edge of the network), or if it is sent on the network-network interface (inside the network).

Pe masura ce traverseaza reteaua ATM, celulele sunt supuse procesului de comutatie (switching) care implica modificarea valorilor VP si VC (label swapping). Disi valorile VP sau VC nu sunt consistente de la un capat la altul al retelei, conceptual de circuit este consistent, spre deosebire de retelele bazare pe comutatie de pachete in care un pachet poate ajunge la destinatie pe un traseu diferit fata de celelalte pachete.

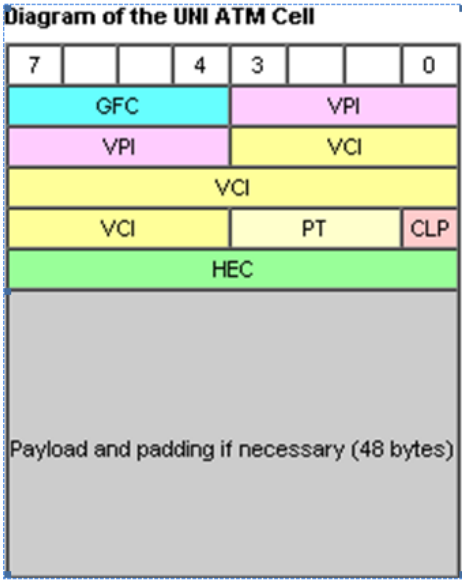
Alt avantaj al folosirii circuitelor virtual este posibilitatea de folosire a lor ca un strat de multiplexare, permitand transmiterea mai multor tipuri de servicii (precum voce, Frame Relay sau IP)

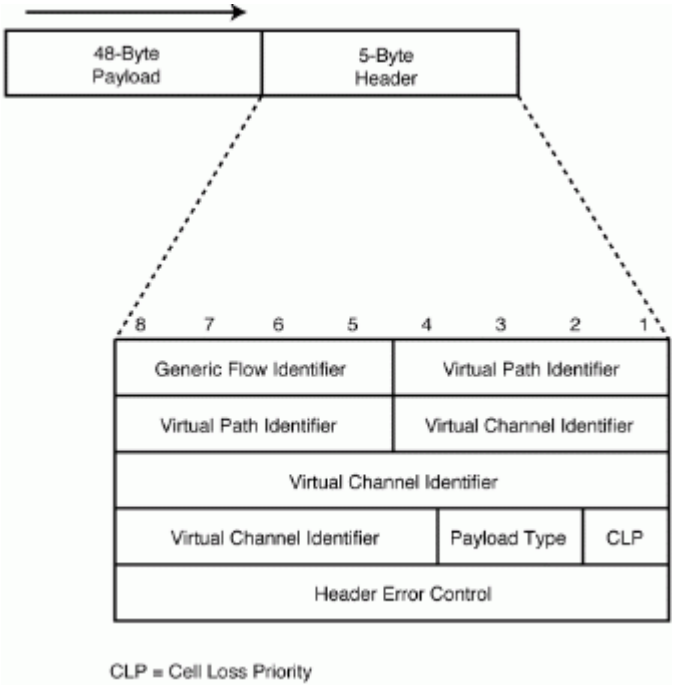
O celula ATM are o dimensiune fixa compusa din:

- 5 octeti antet
- 48 de octeti date. Aceasta dimensiune a fost aleasa ca un compromis intre dimensiunea propusa de tarile europene de 32 de octeti si cea de 64 de octeti propusa de Statele Unite ale Americii. Alegerea unei lungimi a celulei de 53 de octeti rezulta intr-o lantenta minima comparativ cu lungimea unui pachete intre-o retea clasica de calculatoare (Etherenet) care poate ajunge la o dimensiune maxima de 1526 de octeti.

La momentul proiectării ATM, Synchronous Digital Hierarchy (SDH) de 155 Mbit/s cu un trafic util de 135 Mbit/s era considerată o legătură de rețea optică rapidă, iar multe legături alternative din rețeaua digitală Plesiochronous Digital Hierarchy (PDH) erau considerabil mai lente, variind de la 1,544 la 45 Mbit/s în SUA şi de la 2 la 34 Mbit/s în Europa.

Exista doua tipuri de celule: UNI (User-Network Interface) si NNI (Network-Network Interface). Diferenta dintre acestea este ca formatul UNI prezinta un camp suplimentar (GFC) pentru controlul fluxului de date in cazul in care mai multe terminale partajeaza aceeasi conexiune la retea, in vreme ce NNI foloseste acesti biti doar pentru adresare.





GFC = Generic Flow Control (4 bits) (in mod implicit sunt 4 biti de zero) campul apare doar in formatul UNI al celulei ATM.
VPI = Virtual Path Identifier (8 biti pentru formatul UNI sau 12 biti pentru formatul NNI)

VCI = Virtual Channel identifier (16 biti) The 16-bit VCI field identifies a connection between two ATM stations communicating with one another for a specific type of application. Multiple virtual channels (VCs) can be transported within one virtual path. For example, one VC could be used to transport a disk backup operation, while a second VC is used to transport a TCP/IP-based application. T
PT = Payload Type (3 bits) PT field to designate various special kinds of cells for operations, administration, and maintenance (OAM) purposes, and to delineate packet boundaries in some AALs.
CLP = Cell Loss Priority (1-bit) The 1-bit Cell Loss Priority (CLP) field indicates the relative importance of the cell. If this field bit is set to 1, the cell can be discarded by a switch experiencing congestion. If the cell cannot be discarded, the CLP field bit is set to 0.
HEC = Header Error Control este un camp de 8 biti folosit de algoritmul de detectie a erorilor CRC, cu polinomul generator $X^8 + X^2 + X + 1$.

Traficul ATM poate asigura calitatea serviciului ("Quality of Service" - QoS) prin cele patru tipuri de trafic posibile:

CBR - Constant bit rate: este specificata o rata de varf pentru celule, care are valoare constanta.

VBR - Variable bit rate: o rata medie de transfer a celulelor este specificata, care insa poate avea varfuri de trafic pentru scurte intervale de timp.

ABR - Available bit rate: o rata minima garantata este specificata.

UBR - Unspecified bit rate: traficului ii este alocata toata capacitatea de transmisie disponibila.

ATM este in acest moment tehnologia care sta la baza comunicatiilor de voce insa in prezent aceasta metoda de comutatie a inceput sa fie inlocuita de comutatia de pachete (desi prezinta numeroase avantaje fata de aceasta) datorita extinderii retelelor NGN (next generation network) bazate exclusiv pe adresarea IP, cu viteze de 10Gbps (necesitand 1,2 us pentru a transmite un frame Etehrnet).

CARACTERISTICILE COMUNICAŢIEI IN REŢELE DE CALCULATOARE

In retele de calculatoare se foloseste predominant comutatia de pachete. Aşa cum am văzut, uneori cantitatea de date care va fi transmisa este prea mare pentru a fi transportată într-un singur pachet, din acest motiv se realizează împărţirea in mai multe pachete. Fiecare pachet este trimis individual către destinaţie. Deoarece echipamentele dintr-o reţea de calculatoare coordonează mai multe comunicaţii in acelaşi timp, fiecare având priorităţi si conţinut diferit, putem afirma despre transmiterea informaţiei in reţelele de calculatoare ca:

- Nu se garantează timpul de transmitere al informaţiei: aşa cum am văzut un router poate procesa traficul cu diferite întârzieri, datorita mărimii variabile a cozii de aşteptare.
- Nu este garantată primirea în ordine a pachetelor: condiţiile de trafic intr-o reţea se pot schimba de la un moment de timp la altul daca doua pachete de trimit pe trasee diferite,
- Nu este garantată transmiterea informaţiei: anumite routere pot sa fie închise sau sa funcţioneze necorespunzător, pierzând datele care nu au reuşit sa le transmită. Daca in primele doua situaţii pachetele ajung la destinaţie cu o anumita întârziere, in ultimul caz pachetele nu ajung la destinaţie.

Calitatatea transmisiilor de date este din ce in ce mai buna iar viteza este tot mai mare lucru care face ca dezavantajele acestui mod de comutatie sa devina nesemnificative, prin urmare tinde sa devina singurul mod de comutatie folosit in comunicatiile moderne. In unele situaţii pierderea unor pachete nu este importanta, insa in alte cazuri trebuie implementate mecanisme care sa permită detectarea pachetelor pierdute si retransmiterea acestora.

ELEMENTE DE TEORIA INFORMATIEI

Teoria informatiei serveste la a putea evalua necesarul de informatie de transmis de la emitator la receptor si de-a evalua eficienta si redundanta transmiterii informatiei.

Teoretic receptorul are nevoie de un mesaj de la emitator daca continutul acelui mesaj (si eventual momentul ın care este trimis) nu poate fi dedus de catre receptor pe baza cunostintelor sale. Putem spune ca un mesaj contine informative daca aduce receptorului ceva

ce acesta nu putea obtine singur. Prin urmare, un mesaj contine informatie numai ın masura ın care receptorul are o incertitudine pe care prin primirea mesajului o poate ınlatura. *Informatia* este deci inlaturarea unei incertitudini. Masura informatiei trebuie sa masoare incertitudinea ınlaturata prin dobandirea unei informatii.

Presupunem ca avem o multime completa de evenimente posibile, disjuncte: $A = \{x_1, x_2, \ldots, x_n\}$ cu probabilit˘at¸ile $p_i = P(x_i)$. Prin definitie, un mesaj care ınstiinteaza receptorul de producerea evenimentului x_i aduce o cantitate de informatie $i = -\log_2 p_i$. Unitatea de masura se numeste *bit*.

Teoria codarii reprezinta studiul proprietatilor codurilor si potrivirea lor pentru o anumita aplicatie. Codurile sunt folosite pentru compresia datelor, criptografie, corectia erorilor sau codarea de retea.

Doua aspect pricipale sunt tratate de teoria codarii:

- **Compresia datelor numita si codarea de sursa;** Scopul este incercarea de a comprima datele provenid de la sursa pentru a le transmite mai efficient. Aceasta tehnica este gasita in transmisiile internet in care compresia ZIP este utilizata pentru a reduce incarcarea retelei si pentru a face fisierele mai mici. Entropia unei surse este masura informatiei. Codarea de sursa incearca sa reduca redundant prezenta in sursa si sa reprezinte sursa cu mai putini biti care sa transporte mai multa informatie. Nici o tehnica de codare de sursa nu poate fi mai buna decat entropia sursei.
- **Corectia erorilor numita si codarea de canal.** Aceasta codare se realizeaza prin adaugarea de biti suplimentari la mesaj pentru a face comunicatia mai rezistenta la aparitia erorilor in canalul de comunicatie. Scopul este de a gasi coduri care transmit rapid, contin multe cuvinte de cod valide si care corecteaza (sau cel putin detecteaza) cat mai multe erori. Din acest motiv in functie de directia de optimizare dorita pot rezulta coduri optimale diferite. De exemplu CD-urile folosesc codarea Reed-Solomon pentru a corecta erorile datorate zgarierilor sau prafului.

Suppose a source sends r messages per second, and the entropy of a message is H bits per message. The information rate is R = r H bits/second.

One can intuitively reason that, for a given communication system, as the information rate increases the number of errors per second will also increase. Surprisingly, however, this is not the case.

Shannon's theorem:

A given communication system has a maximum rate of information C known as the channel capacity.

If the information rate R is less than C, then one can approach arbitrarily small error probabilities by using intelligent coding techniques.

To get lower error probabilities, the encoder has to work on longer blocks of signal data. This entails longer delays and higher computational requirements.

Thus, if R ≤C then transmission may be accomplished without error in the presence of noise.

Unfortunately, Shannon's theorem is not a constructive proof — it merely states that such a coding method exists. The proof can therefore not be used to develop a coding method that reaches the channel capacity.

The negation of this theorem is also true: if R > C, then errors cannot be avoided regardless of the coding technique used.

Teorema care sta la baza tehnicilor de codare a fost enuntata de Claude Shannon in 1948. Ea stabileste limitele pentru compresia datelor si simbolistica operationala pentru entropia Shannon. Entropia in teoria informatiei este o măsură a incertitudinii asociate unei variabile aleatoare. In acest context, entropia Shannon indică cantitatea de informație asteptată într-un mesaj, exprimată de obicei în biți sau în biți pe simbol. In acelasi timp entropia Shannon este o masura a continutului informational mediu care lipseste in cazul in care o varaibila aleatoare nu este cunoscuta.

Teorema Claude Shannon lui arata ca este imposibila compresia datelor intr-o asemenea masura incat rata de cod (raportul intre bitii de date si bitii total transmisi) sa fie mai mica decat entropia Shannon a sursei, fara a avea certitudinea ca se va pierde informatie. Totusi este posibil sa existe o rata de cod arbitrar de aproape de entropia Shannon cu o probabilitate neglijabila de pierderi de date.

Deoarece este imposibila proiectarea unei tehnici de codare perfecte, s-au dezvoltat scheme de corectie a erorilor specifice pentru erori de tip bit sau erori de tip burst (grupate). Sistemele in timp real trebuie sa tina un balans intre protectia la erori si intatzierea la codare/decodare.

Se va mai tine cont la implementare ca un cod software complex si performant este mai dispus la erori decat unul simplu, care are insa un algoritm mai putin slab. Marirea complexitatii software duce la aparitia mai multor erori de design si de implementare.

Se disting doua tipuri de codare: codare liniara si codare convolutionala.

Codurile liniare se caracterizeaza prin segmentarea mesajului in blocuri de lungime fixa, si codarea blocului o singura data pentru transmisie. Codurile convolutionale codeaza tot sirul de date intr-un singur bloc, pe care il transmit in parti mai mici.

Codurile bloc liniare se numesc astfel deoarece fiecare cuvant de cod e o combinatie liniara a unui set de cuvinte de cod generatoare: pentru a coda un mesaj de k biti, se face o simpla inmultire intre vectorul mesajului si matricea generatoare a codului.

Aceste coduri sunt usor de implementat hardware. Au o rata mare de cod (raportul intre bitii de date si lungimea totala a mesajului), de obicei peste 95%. In acelasi timp, ele nu adauga multe date redundante in mesaj. Se folosesc in situatii in care rata de eroare a bitilor este mica (BER: Bit Error Rate) si atunci cand e usor posibila retransmisia bitilor.

In aceasta categorie intre codurile SEC/DED (Single Correcting Code/Double Error Detecting) pentru memorii de calculator de viteza mare.

Un subset special de coduri liniare sunt codurile: Cyclic Redundancy Check (CRC). Acestea sunt cele mai folosite in comunicatii digitale. Deoarece se obtin prin shift-are, ele au o deosebita usurinta in implementare prin folosirea registrilor de shift-are. Datele trebuie retransmise odata ce s-a detectrat o eroare.

De obicei se folosesc impreuna cu un cod corector de erori, deoarece codurile CRC nu stiu sa faca corectia erorilor. In realizarea practica a sistemului de comunicatie in infrarosu am ales un cod CRC.

Codurile convolutionale au rata de cod mai mica (aproximativ 0,90) dar au posibilitati mari de corectie a erorilor. Totusi, ele sunt foarte greu de implemantat, in software sau hardware, de aceea sunt folosite doar in comunicatiile spatiale. O metoda bine cunoscuta pentru decodarea acestor coduri este algoritmul lui Viterbi.

Corectia erorilor este mai dificil de implementat in hardware sau software decat detectia erorilor. Corectia este utila in situatiile in care repetarea mesajului detectat a fi eronat este imposibil de realizat (cel putin in timp util) sau impune costuri foarte mari, asa cum este cazul comunicatiilor spatiale. Deoarece este usor in retele de calculatoare sa se retransmita datele care au fost detectate a fi eronate, s-a apelat la introducrerea metodelor de detectie a erorilor.

METODE PENTRU DEPISTAREA ERORILOR

Primul lucru care trebuie făcut pentru asigurarea unei comunicaţii corecte in cazul in care canalul de comunicaţie are erori, este depistarea acestora. Fără acest lucru nu se pot lua masuri de corecţie a erorilor. Exista mai multe metode pentru depistarea erorilor dintre care putem enumera:

1. repetarea fiecărui bit sau repetarea întregului pachet de un număr impar de ori (de exemplu de trei ori): in acest caz o eroare este uşor de detectat si chiar de corectat, prin regula votului. Dacă toate valorile transmite sunt identice transmisia este corecta, daca valorile sunt diferite, transmisa este eronata. In exemplul de mai jos se încearcă transmiterea a trei biţi de 1 de la sursa la destinaţie. Biţii se repeta de trei ori. Cea de a doua secvenţă transmisa are doi biţi de 1 si un bit de valoare 0, prin urmare a fost o eroare de transmisie. In general, aceasta metoda nu se recomanda a fi folosită, deoarece erorile grupate pot produce şiruri de aceeaşi valoare care vor trece uşor nedetectate.

Sursă	1	1	1	1	1	1	1	1	1
Destin	1	1	1	1	0	1	1	1	1
	OK			EROARE			OK		

2. folosirea bitului de paritate: se adăuga la sfârsitul mesajului un bit numit "bit de paritate" su[illegible] ăr impar rezultând într-o paritate para respectiv impara. Daca, de exemplu, la receptie se constata ca suma tuturor biţil[illegible]etata sa folosească paritate para, înseamnă ca a avut loc o eroare de paritate. In exemplul de mai jos este prezentat modul de calcul al parităţii pentru şapte biţi de date.

Biţi de date							Bit de paritate	
1	1	1	1	1	1	1	1	Paritate [illegible]
1	0	1	1	0	1	1	0	Paritate [illegible]
1	0	1	1	0	0	1	1	Paritate impară

Din păcate erorile care afectează un număr par de biţi trec nedetectate daca se foloseste bitul de paritate, din acest motiv uneori se grupeaza mai multe segmente intr-o matrice si se calculeaza paritatea atât pe linii cat si pe coloane. In acest mod fiecare data este verificata de doi biţi de paritate, aşa cum se observa in exemplul următor.

	Biţi de date							Biţi de paritate	
	1	1	1	1	1	1	1	1	
	1	0	1	1	0	1	1	1	*Paritate*
	1	0	1	1	0	0	1	0	*pară*
Biţi de paritate	1	1	1	1	1	1	1	1	

Nici aceasta metoda nu este sigura, anumite erori putându-se strecura nedetectate, aşa cum se observa in exemplul următor.

	Biţi de date							Biţi de paritate	
	x	1	x	1	1	1	1	1	
	1	0	1	1	0	1	1	1	*Paritate*
	x	0	x	1	0	0	1	0	*pară*
Biţi de paritate	1	1	1	1	1	1	1	1	

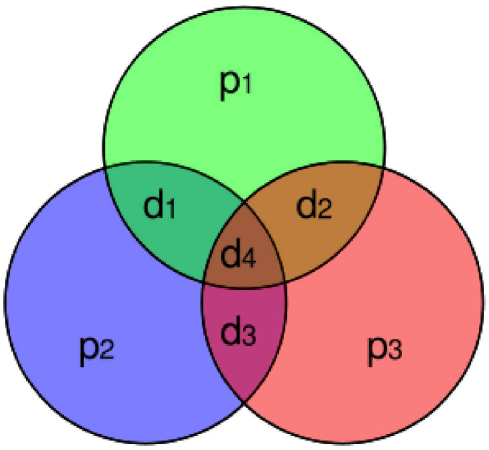
Indiferent daca cei patru biţi au valoarea de 1 sau 0, paritatea este in mod incorect verificata.

3. codul Hamming este o metodă de depistare a erorilor prin adăugarea de informaţie redundantă legată de paritate. Codul Hamming poate detecta pana la doua erori simultane de bit si pot corecta erorile de un singur bit.

Codurile Hamming sunt coduri liniare binare. Pentru fiecare întreg m≥2 exista un cod cu m biti de paritate si 2^m-m-1 biti de date.

….

Datorita simplităţii codurile Hammming sunt implementate in memoria RAM a calculatorului.



Bit #	1	2	3	4	5	6	7
Transmitted bit	p_1	p_2	d_1	p_3	d_2	d_3	d_4
p_1	Yes	No	Yes	No	Yes	No	Yes
p_2	No	Yes	Yes	No	No	Yes	Yes
p_3	No	No	No	Yes	Yes	Yes	Yes

……..

5. suma de verificare este o metoda simpla de verificare a unei secvenţe de date este însumarea cuvintelor de o anumita dimensiune, eliminând orice depăşire a dimensiunii stabilite. Rezultatului astfel obţinut ii este calculat codul complement fata de doi iar valoarea obţinuta este ataşata datelor originale. La destinaţie toate datele sunt trecute prin acelasi algoritm, inclusiv suma de verificare. In cazul in care rezultatul calculat la destinaţie este diferit de zero, o eroare s-a strecurat in şirul de date recepţionat. Aceasta metoda detectează toate erorile de un bit insa probabilitatea ca o eroare de doi biţi sa treacă nedetectata este destul de mare.

Exemplul următor arata modul in care este calculata o suma de control pe 16 biti in cadrul protocolului TCP:

Pentru a calcula suma de verificare sunt incluse următoarele valori:

- adresa internet a sursei: un numar pe 4 octeti;
- adresa internet a destinatiei: un numar pe 4 octeti;
- identificatorul de protocol: un numer pe 2 octeti;
- lungimea antetului: un numar pe 2 octeti;
- intregul antet;
- toate datele transportate de protocol;

Exemplul urmator arata modul in care este calculata o suma de control pe 16 biţi in cadrul protocolului IP:

The IP checksum is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header.

Datorita uşurinţei de calculare aceste sume sunt folosite in retelele de calculatoare pentru a realiza o verificare suplimentara a datelor impreuna cu alte metode mai bune pentru detectia erorilor.

4. folosire CRC (Cyclic Redundancy Check) – caz particular al CRC (pe 1 bit) este bitul de paritate

Această tehnică poate depista apariţia oricărei erori în şirul de date. Ea foloseşte un polinom generator cu ajutorul căruia efectuează operaţii de SAU EXCLUSIV asupra şirului de date.

```
Calcul CRC-3, folosind polinom generator 1011, pentru secventa de date 11010011101100

11010011101100 000
1011
01100011101100 000
 1011
00111011101100 000
  1011
00010111101100 000
   1011
00000001101100 000
      1011
00000000110100 000
       1011
00000000011000 000
        1011
00000000001110 000
         1011
00000000000101 000
          101 1
----------------
00000000000000 100
```

Verificare: se calculeaza CRC pentru date urmate de CRC calculat. Rezultatul trebuie sa fie numai biti de 0 daca nu au aparut erori in fluxul de date.

```
11010011101100 100
1011
01100011101100 100
 1011
00111011101100 100

......

00000000001110 100
         1011
00000000000101 100
          101 1
----------------
00000000000000 000
```

Observaţie. - Aceasta este metoda folosită în internet pentru a depista erorile CRC 32 biţi.

- Este suficientă adăugarea de 32 biţi pentru a putea verifica 1 milion de biţi.

- Este uşor de calculat.

Conditii pentru alegerea polinomului generator:

- pentru erori de un bit: de forma 1000000…000. Pentru aceasta trebuie ca

- polinomul sa aiba cel putin doi biti setati in 1
- pentru detectia erorilor de doi biti: de forma 100...000100...000, trebuie ca polinomul sa nu aiba multipli de forma 11, 101, 1001, etc. Matematicianul Tanenbaum afirma ca pentru polinomul cu biti de 1 pe pozitiile (15,14,1) exista un multiplu de forma 100…001 unde numarul de zerouri este 32767.
- Erori cu numar impar de biti: trebuie sa ne asiguram ca polinomul are un numar par de biti 1.
- Erori in grup "burst errors": trebuie ca ultimul bit sa fie 1. Tot Tanenbaum afirma ca posibilitatea de trecere nedetectata a acestor erori este de 0,5W, unde w este lungimea grupului

Algoritmi pentru corectia erorilor

In timpul unei comunicatii, in ferestrele de transmisie pot sa apara erori la nivel de bit. Acest lucru se intampla, de exemplu, din cauza interferentelor electrice si a zgomotului temic, sau in cazul transmisiilor in infrarosu, datorita obstructionarii canalului de comunicatie. Desi rare, in special pentru legaturile optice, trebuie implementate mecanisme de detectie a erorilor pentru a se putea trece la corectia lor. In caz contrar nu s-ar putea asigura o comunicatie corecta a datelor in retea.
Exista o lunga istorie de tehnici pentru tratarea erorilor la nivel de bit din retelele de calculatoare, incepand cu codurile Hamming si Reed-Solomon, care erau concepute pentru a fi folosite in stocarile de date si in primele tipuri de memorii. In retelele de calculatoare, metoda principala pentru detectarea erorilor este cea de Cyclic Redundancy Check (CRC). Este folosita de aproape toate protocoalele care lucreaza la nivelul legatura de retea (data-link layer): HDLC, DDCMP, IMP-IMP, precum si in mai cunoscutele protocoale CSMA si token Ring.
Pentru detectia erorilor se introduc in general doua metode de baza: paritatea bloc (two-dimensional parity) si sumele de control. Prima e folosita de protocolul BISYNC la transmiterea caracterelor ASCII , a doua e folosita in protocoalele de retea.
Metode de corectie a erorilor:
1. Backward Error Correction (BEC, or Reverse Error Correction): In acest mod de lucru, se trimite mesajul, si daca se detecteaza aparitia unei erori, se cere emitatorului retransmisia mesajului. Aceasta implica nu numai detectia erorilor dar si comunicatie duplex. Aceasta nu e recomandata pentru situatiile incare mai multi receptori ar trebui sa astepte retransmisia, sau atunci cand comunicatii duplex nu sunt posibile (nave spatiale). Mai exista si dezavantajul ca in cazul unei erori fixe (ex. Una din mai multe
linii este rupta) transmisia se blocheaza prin retransmiterea aceluiasi mesaj eronat de un numar nelimitat de ori, fara a exista posibilitatea de depistare a erorii. Avantajul este ca algoritmul este simplu si eficient.
2. Forward Error Correction (FEC): Aici receptorul primeste informatia si nu mai are nevoie de alte date de la emitator. Acelt luctru implica un minim de redundanta in transmisia de la emitator. Exista mai multe metode:
Trimodular Redundancy (TMR) prin care fiecare bit este trimis de 3
ori, si se permite receptorului sa "voteze" asupra valorii corecte.
Pentru valori mai mari, algoritmul isi pierde din eficienta.
- Coduri Hamming: acestea devin mai eficiente pe masura ce dimensiunea datelor de transmis devine mai mare:

$$\lim_{n\to\infty} R = \lim_{n\to\infty} \frac{2^{k-1} - k}{2^{k-1}} = 1$$

unde k – biti de paritate, iar R este rata informatiei (raportul intre numarul bitilor de date si numarul bitilor total trimisi). Dezavantajul este ca pe masura ce dimensiunea cereste, si posibilitatea ca mai mult de o eroarea sa apara in blocul de date, de aceea trebuie gasit un echilibru.

Coduri de linie

O importanta deosebita pentru transmiterea informatiei in retele de calculatoare o are formatul in care valorile digitale binare sunt reprezentate in liniile seriale folosite in retelele de calculatoare.

Codarea de linie se refera la modul in care informatia digitala este transmisa printru-un canal de date. Codarea simbolurilor binare abstracte in forme de unda reale care vor fi transmise in banda de baza se numeste codare se linie.

Banda de baza descrie semnalele a caror gama de frecvente se intinde de la aproape 0 Hz pana la o frecventa de taiere, o latime de banda maxima sau o valoare maxima a frecventei semnalului.

Sistemele de transmisie a datelor care nu folosesc procedee de modulare/demodulare, utilizând direct forma de undă numerică (semnalul dreptunghiular) asociat secventei de date, se numesc sisteme de transmisie a datelor în banda de bază, deoarece banda de frecvente a semnalului nu este translată în jurul unei frecvente superioare si contne inclusiv componenta continuă (frecventa zero).

Transmisia digitală în banda de bază înseamnă, în general, transmiterea directă pe canal a semnalului dreptunghiular cu două nivele de tensiune sau curent corespunzătoare valorilor logice de “0” si “1”. Deoarece transmisia datelor are loc în mod serial, echipamentul receptor va primi doar două nivele de tensiune care se succed în timp, din care acesta trebuie să discrimineze începutul si sfârsitul fiecărei celule de bit (sincronizarea pe bit), dar si împachetarea globală a sirului de date în caractere si blocuri de date (sincronizarea pe caracter si pe cadru). Sincronizarea receptorului cu emitătorul poate fi realizată în două moduri, în functie de relatia dintre ceasul transmitătorului si cel al receptorului.

In retelele de calculatoare (de exemplu Ethernet) cuvantul BASE indica transmisii in banda de baza (ex. 10BASE-T) adica se realizeaza codarea de linie printr-o linie de date nefiltrata. Aceasta modaliate de transmisie este diferita de cea folosita de retelele wireless sau de modemurile de cablu, numita trece banda (ex. 10PASS-TS) sau broadband (ex. 10BROAD36 – tehnologie mai veche si ITU-T G.hn – tehnologie pentru transmisii prin liniile de alimentare cu tensiune).

BROADBAND in computer networks

Many computer networks use a simple line code to transmit one type of signal using a medium's full bandwidth using its baseband (from zero through the highest frequency needed). Most versions of the popular Ethernet family are given names such as the original 1980s 10BASE5 to indicate this. Networks that use cable modems on standard cable television infrastructure are called broadband to indicate the wide range of frequencies that can include multiple data users as well as traditional television channels on the same cable. Broadband systems usually use a different radio frequency modulated by the data signal for each band.[6] The total bandwidth of the medium is larger than the bandwidth of any channel.[7]

The 10BROAD36 broadband variant of Ethernet was standardized by 1985, but was not commercially successful.[8][9] The DOCSIS standard became available to consumers in the late 1990s, to provide Internet access to cable television residential customers. Matters were further confused by the fact that the 10PASS-TS standard for Ethernet ratified in 2008 used DSL technology, and both cable and DSL modems often have Ethernet connectors on them.

Power lines have also been used for various types of data communication. Although some systems for remote control are based on narrowband signaling, modern high-speed systems use broadband signaling to achieve very high data rates. One example is the ITU-T G.hnstandard, which provides a way to create a high-speed (up to 1

Gigabit/s) local area networkusing existing home wiring (including power lines, but also phone lines and coaxial cables).

FIG COMUNICATIE RETELE (BASEBAND) VS COMUNICATIE MODEM DE CABLU

Baseband transmission in Ethernet

The word "BASE" in Ethernet physical layer standards, for example 10BASE5, 100BASE-Tand 1000BASE-SX, implies baseband digital transmission, i.e. that a line code and an unfiltered wire are used.

This is as opposed to 10PASS-TS Ethernet, where "PASS" implies passband transmission. Passband digital transmission requires a digital modulation scheme, often provided by modemequipment. In the 10PASS-TS case the VDSL standard is utilized, which is based on the Discrete multi-tone modulation (DMT) scheme. Other examples of passband network access technologies are wireless networks and cable modems.

Digital baseband transmission, also known as line coding,[3] aims at transferring a digital bit stream over base-band channel, typically an unfiltered wire, as opposed to passbandtransmission, also known as *carrier-modulated* transmission.[4] Passband transmission makes communication possible over a bandpass filtered channel, such as the telephone network local-loop or a band-limited wireless channel.

An unfiltered wire is intrinsically a low-pass transmission channel, while a line code is intrinsically a pulse wave signal that occupies a frequency spectrum of infinite bandwidth. According to the Nyquist theorem, error-free detection of the line code requires a channel bandwidth of at least the Nyquist rate, which is half the line code pulse rate.

In a digital communication system, there exists a known set of symbols to be transmitted. These can be designated as
fmı
g, i D 1; 2; : : : ; N, with a probability of occurrence fpi
g, i D 1; 2; : : : ; N, where the sequentially transmitted symbols are generally assumed to be statistically independent. The conversion or coding of these abstract symbols into real, temporal waveforms to be transmitted in baseband is the process of line coding. Since the most common type of line coding is for binary data, such a waveform can be
succinctly termed a direct format for serial bits. The concentration in this section will be line coding for binary data.

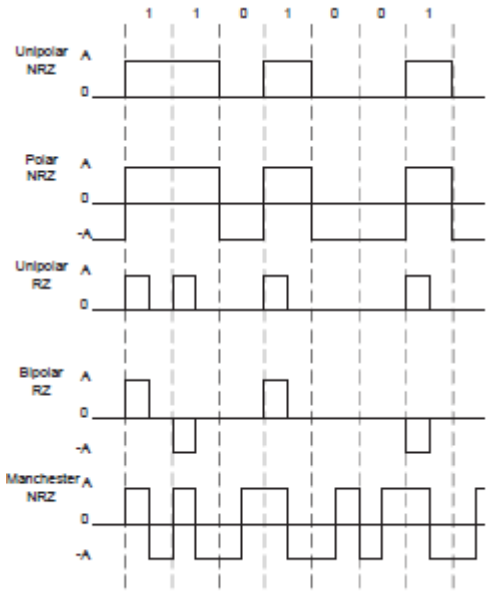
A line coding format consists of a formal definition of the line code that specifies how a string of binary digits are converted to a line code waveform. There are two major classes of binary line codes: level codes and transition codes. Level codes carry information in their voltage level, which may be high or low for a full bit period or part of the bit period. Level codes are usually instantaneous since they typically encode a binary digit into a distinct waveform, independent of any past binary data. However, some level codes do exhibit memory. Transition codes carry information in the change in level appearing in the line code waveform. Transition codes may be instantaneous, but they generally have memory, using past binary data to dictate the present waveform. There are two common forms of level line codes: one is called return to zero (RZ) and the other is called nonreturn to zero (NRZ).
In RZ coding, the level of the pulse returns to zero for a portion of the bit interval. In NRZ coding, the level of the pulse is maintained during the entire bit interval.
Line coding formats are further classified according to the polarity of the voltage levels used to represent the data. If only one polarity of voltage level is used, i.e., positive or negative (in addition to the zero level) then it is called unipolar signalling. If both positive and negative voltage levels are

being used, with or without a zero voltage level, then it is called polar signalling. The term bipolar signalling is used by some authors to designate a specific line coding scheme with positive, negative, and zero voltage levels. This will be described in detail later in this section. The formal definition of five common line codes is given in the following along with a representative waveform, the power spectral density (PSD), the probability of error, and a discussion of advantages and disadvantages. In some cases specific applications are noted.

Comunicatiile digitale utilizeaza coduri de linie pentru reprezentara semnalelor digitale

Modul de reprezentare a semnalelor binare se numeste codarea de linie.

Dintre modurile de reprezentare, cele mai populare sunt prezentate in continuare:



Exista doua categorii principale:

- Return to Zero (RZ)
- Non Return to Zero (NRZ)

Diferenta este reprezentata de faptul ca in codarea RZ semnalul cade in 0 la jumatatea bitului.

The following are some of the desirable properties of a line code:
– Self–Synchronisation: There is enough timing information built into the code so that bit synchronisers can extract the timing or clock signal. A long series of binary 1's or 0's should not cause a problem in time recovery
– Low Probability of Bit Error: Receivers can be designed that will recover the binary data with a low probability of bit error when the input data is corrupted by noise or ISI
– A Spectrum that is Suitable for the Channel: For example, if the channel is AC coupled, the PSD of the line code signal should be negligible at frequencies near 0. In addition, the signal bandwidth needs to be sufficiently small compared to the channel bandwidth, so that ISI will not be a problem
– Transmission Bandwidth: This should be as small as possible
– Error Detection Capability: It should be possible to implement this feature easily by the addition of channel encoders and decoders, or the feature should be incorporated into the line code

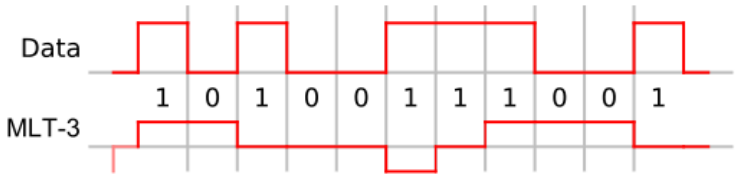
Clasificare:
-dupa comportamentul la jumatatea transmiterii unui bit:
-return to zero RZ
-non return to zero NRZ

-dupa polaritatea nivelurilor de tenstiune:

- unipolar 0V - 0 logic
+V - 1 logic

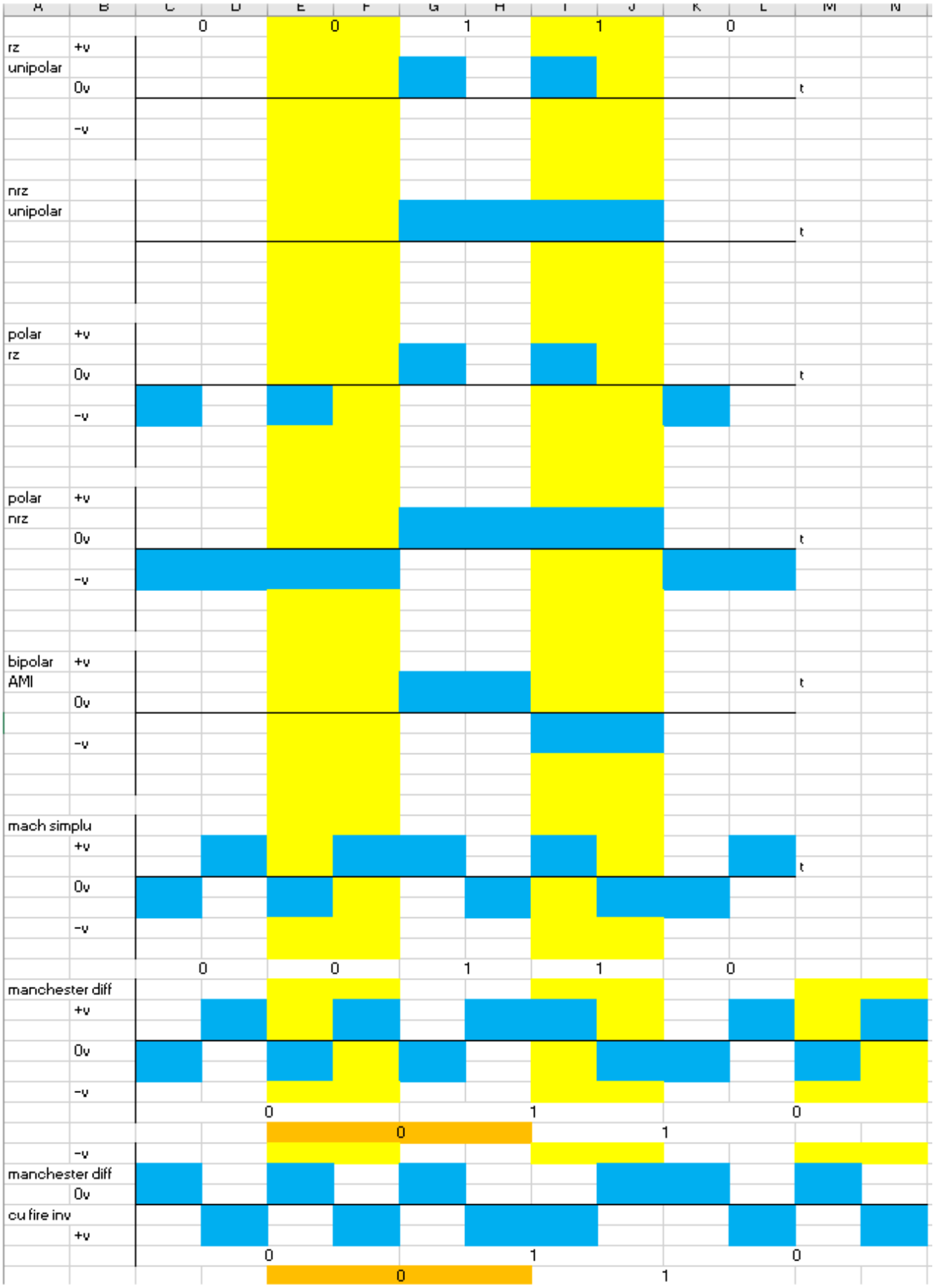
- polar:
-V -> 0 logic
+V -> 1 logic

- bipolar (pseudoternar): (alternate mark inversion (AMI)
1 logic : alternarnativ +V si -V
0 logic: 0V
- Diferentiala: codarea semanalului curent depinde atat de starea curenta cat si de starea semnalului anterior
- Multi-nivel (de ex.MLT-3) MLT-3 cicleaza intre starile -1,0,1, 0. La transmiterea unui bit de 1 trece la starea urmatoare iar la transmiterea unui bit de 0 ramane in starea curenta



Codarea
Manchester:
simplu/normal 1 logic -> 1/2 bit +V 1/2bit -V tranz negativa
0 logic -> 1/2 bit -V 1/2bit +V tranz pozitiva
..frecv dubla -> scumpe
> efic. energetic

diferential .... tranzitie in timp (split–phase encoding) .. e rezistenta la inversare fire
1 logic: anterior tranzitie pozitiva -> curent tranz negativa ..inv tranzitie intre 2 biti
0 logic: anterior tranzitie pozitiva > curent tranz pozitiva ...pastrare tranzitie intre 2 biti

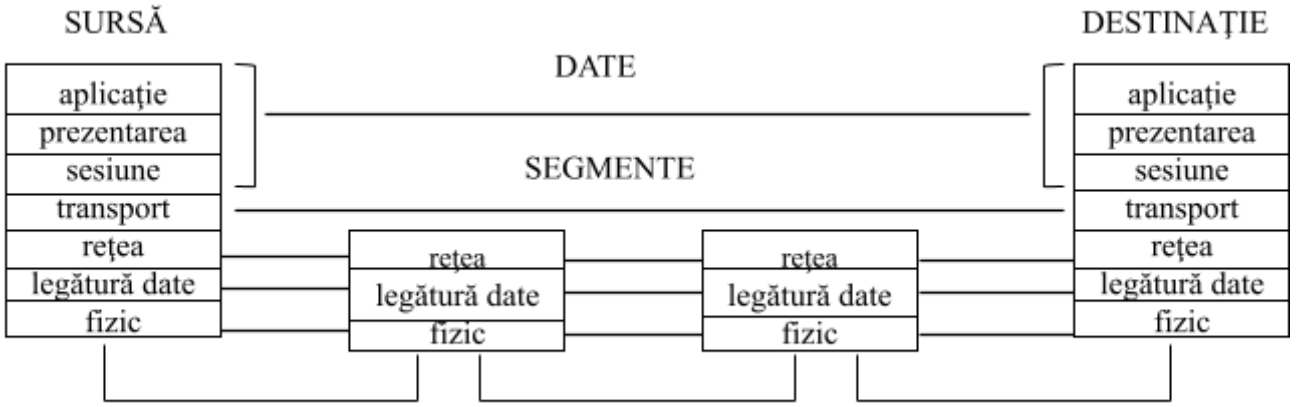

0 0 1 1 0
rz
+v
unipolar
0v
t
-v
nrz
unipolar
t
polar
+v
rz
0v
t
-v
polar
+v
nrz
0v
t
-v
bipolar
+v
AMI
t
0v
-v
mach simplu
+v
t
0v
-v
0 0 1 1 0
manchester diff
+v
0v
-v
0 1 0
0 1
-v
manchester diff
0v
cu fire inv
+v
0 1 0
0 1

Curs 2

Exemplu detaliat de comunicaţie la nivelul internetului

Odată cu răspândirea internetului (anii 1960 - 1970) s-a simţit nevoia de introducere a standardizării. În 1977 ISO (International Organisation for Standardisation) a propus modelul OSI (Open Systems Interconection). El este un model de referinţă care descrie comunicaţia şi arhitectura stratificată a interconectării. Datorită evoluţiei rapide modelul nu mai este utilizat în practică însă are un rol important educaţional.

Standardele incluse în acest model se referă la comportarea exterioară a elementelor din interior.



Aplicaţie – nivelul asigură interfaţa şi serviciile cu utilizatorul (client FTP, browser internet, etc.).
Prezentare – asigură reprezentarea unitară a datelor indiferent de tipul echiopamentului care se conecteaza la internet (telefon, PDA, PC, etc.) şi criptarea datelor.

Exemplu: 12A3h

INTEL	A312	little endian
MOTOROLA	12A3	big endian

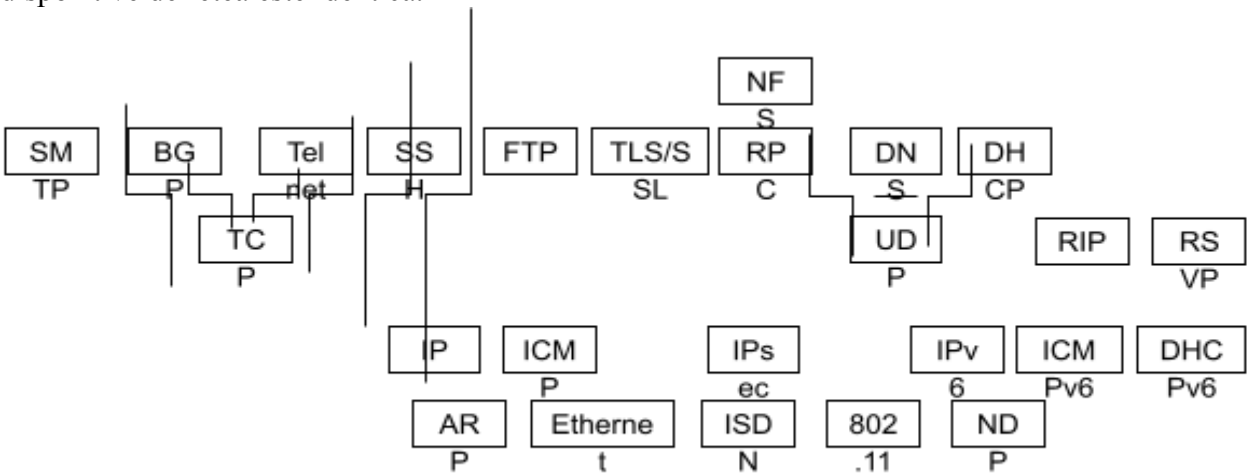
Sesiune – se ocupă cu deschiderea , menţinerea şi inchiderea conexiunii.
Transportul – asigură transmiterea datelor corectă şi în ordinea de la destinaţie.
Reţea – acest nivel asigură o cale de la sursă la destinaţie (best efort) (aici găsim adresele IP).
Legătura de date – furnizează un transport sigur, fiabil, al datelor de-a lungul unei legături fizice, realizând:

- Controlul erorilor de comunicaţie
- Controlul fluxului de date
- Controlul legăturii
- Sincronizarea la nivel de cadru

Nivelul LLC - Asigură mecanisme pentru multiplexarea mai multor protocoale prin acelasi mediu (IP si IPX) precum si controlul fluxului de date. Identificarea locală a calculatoarelor în reţeaua de date si controlul acesului la mediul de transmisie se realizeaza prin subnivelul MAC – Media Access Control.
Nivelul Fizic – Rol de transmitere a unui şir de biţi pe un canal de comunicaţie. Se precizează modulaţii, codări, sincronizări la nivel de bit. Un standard de nivel fizic defineşte 4 tipuri de caracteristici:

- Mecanice (forma şi dimensiunile conectorilor, numărul de pini)
- Electrice (modulaţia, debite binare, codări, lungimi maxime ale canalelor de comunicaţie)
- Funcţionale (funcţia fiecărui pin)
- Procedurale (succesiunea procedurilor pentru activarea unui serviciu).

Pentru ca doua calculatoare sa se inteleaga, ele trebuie sa respecte acelasi set de reguli. Protocolul de comunicaţie – este un set de reguli prin care se realizează comunicarea între 2 sisteme. Incapsulararea/Interpretarea informaţiei în nivelurile de acelaşi similare pe diferite dispozitive de retea este identica.



In practica se foloseste modelul TCP/IP (Transmission Control Protocol/Internet Protocol) care a fost creat de US DoD (US Department of Defence - Ministerul Apărării Naţionale al Statelor Unite) din necesitatea unei reţele care ar putea supravieţui în orice condiţii. DoD dorea ca, atâta timp cât funcţionau maşina sursă şi maşina destinaţie, conexiunile să rămână intacte, chiar dacă o parte din maşini sau din liniile de transmisie erau brusc scoase din funcţiune. Era nevoie de o arhitectură flexibilă, deoarece se aveau în vedere aplicaţii cu cerinţe divergente, mergând de la transferul de fişiere până la transmiterea vorbirii în timp real.

Aceste cerinţe au condus la alegerea a patru niveluri pentru modelul TCP/IP: Aplicaţie, Transport, Reţea (sau Internet) şi Acces la Reţea.

Modelul TCP/IP

Aplicatie
Transport
Internet
Acces la Reţea

.

La nivelul transport datele sunt dizivate in segmente care reprezinta unitate de protocol specifica a protocoalelor la acest. Ele sunt insoţite de un antet prin care sunt realizate cererile către calculatorul destinaţie, precum si alte informatii de verificare a integritatii si lungimii datelor transmise. Nivelul transport are 2 protocoale: TCP şi UDP. In continuare se adauga adresele sursa/destinatie IP la nivelul internet si alte date aditionale (inclusiv de verificare). Adresele IP sunt valabile la nivelul internet (exceptie adresele private). In final datele sunt incapsulate in cadre (nivel acces la retea) si sunt insotite de adresele MAC si suma de verificare CRC-32 inainte de a fi trimise in mediul de trasmisie prin tehnici specifice.

La destinatie se realizeaza procesarea datelor in sens invers pentru a scoate datele introduse la nivelul aplicatie.

Medii de transmisie

Modemurile sunt dispozitive care au fost folosite pentru transmiterea informatiei de la sfarsitul anilor ’40 pana in prezent folosind liniile telefonice.

Un modem este un dispozitiv care moduleaza un semnal analog in scopul codarii informatiei digitale sau il demoduleaza pentru a extrage informatia digitala. Denumirea de modem provine chiar de la activitatea sa (**mod**ulare si **dem**odulare).

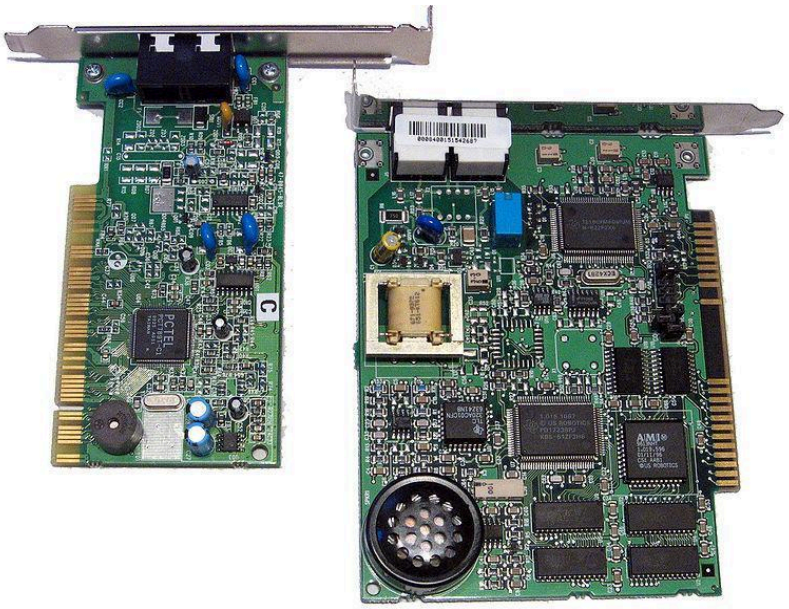
O caracteristica importanta a acestor echipamente este aceea ca un modem nu se poate conecta in mod normal direct la unul sau mai multe modemuri in scopul realizarii unei retele locale. Modemul se conecteaza la semnalul provenind de la un distribuitor de servicii de telefonie, din acest motiv exista costuri asociate schimbului de date.

O alta caracteristica este viteza de trasmisie a datelor, exprimata prin cantitatea de transmisie a datelor in unitatea de timp. O parte din vitezele specifice modemurilor sunt prezentate in tabelul urmator.

Standard	Viteza	An aparitie
V.21	0.3 kbit/s	1962
V22	1.2 kbit/s	1976
V.32	9.6 kbit/s	1989
V32BIS	14.4 kbit/s	1991
V34	28.8 kbit/s	1994
V90	56.0/33.6 kbit/s	1998
V92	56.0/48.0 kbit/s	2001
ISDN Basic Rate Interface	64/128 kbit/s	1986

In procesul de reducere a costurilor de implementare a modemurilor au aparut mai multe inventii dintre care voi aminti pe cele mai importante.

Pentru a se conecta la reteaua de telefonie, un modem necesita ridicarea manuala a receptorului si formarea numarului, proces automatizat prin aparitia unor echipamente speciale de formare numite „*dialer*” care erau controlate prin portul RS-232. Similar, la primirea unui apel telefonic (de date), era necesara ridicarea receptorului. Prin urmare existau costuri aditionale pentru automatizarea formarii si un port serial era ocupat. In 1981 Hayes Communications a propus o solutie care folosea un set bine definit de comenzi prin care calculatorul controla direct modemul fara un echipament *dialer*. Un astfel de modem opera in doua moduri: modul de date si modul de comenzi, trecerea intre modul de date si cel de comenzi fiind controlata prin transmiterea unei secvente de trei caractere „+” urmate de o pauza de o secunda, iar pentru schimbarea invarsa O. Interpretarea comenzilor se realiza de catre un microcontroler integrat in modem. Setul de comenzi propus a constituit baza comunicatiei cu modemurile moderne, la acesta adaugandu-se comenzi noi pentru a identifica noile functii aparute.



O alta inovatie a fost aparitia softmodemurilor in care interpretarea tonurilor se face prin software ruland pe calculator. In acest mod numetul componenetlor hardware se reducea semnificativ. Din pacate folosirea resurselor calculatorului a dus la scaderea performantiei acestuia iar noile produse au avut driver (care gestioneaza comunicatia intre echipament si sistemul de operare) doar pentru sistemul de operare Windows, reducand utilitatea acestor echipamente pentru alte sisteme de operare.

Compresia datelor este o alta tehnica folosita (standardele V.42, V.42bis si V.44). In functie de tipul datelor, compresia permitea transmiterea unei canti mult mai mare de date (de execmplu un text putea fi comprimat de 6 ori, reusind astfel sa trimita aproape 300kbps de text printr-un trafic de 50kbps).

Integrated Services Digital Network *(**ISDN**) is a set of communications standards for simultaneous digital transmission of voice, video, data, and other network services over the traditional circuits of the public switched telephone network. It was first defined in 1988 in theCCITT red book.[1] Prior to ISDN, the telephone system was viewed as a way to transport voice, with some special services available for data. The key feature of ISDN is that it integrates speech and data on the same lines, adding features that were not available in theclassic telephone system. There are several kinds of access interfaces to ISDN defined asBasic Rate Interface (BRI), Primary Rate Interface (PRI) and Broadband ISDN (B-ISDN).*

ISDN is a circuit-switched telephone network system, which also provides access to packet switched networks, designed to allow digital transmission of voice and data over ordinarytelephone copper wires, resulting in potentially better voice quality than an analog phone can provide. It offers circuit-switched connections (for either voice or data), and packet-switched connections (for data), in increments of 64 kilobit/s. A major market application for ISDN in some countries is Internet access, where ISDN typically provides a maximum of 128 kbit/s in both upstream and downstream directions. Channel bonding can achieve a greater data rate; typically the ISDN B-channels of 3 or 4 BRIs (6 to 8 64 kbit/s channels) are bonded.

ISDN should not be mistaken for its use with a specific protocol, such as Q.931 whereby ISDN is employed as the network, data-link and physical layers in the context of the OSI model. In a broad sense ISDN can be considered a suite of digital services existing on layers 1, 2, and 3 of the OSI model. ISDN is designed to provide access to voice and data services simultaneously.

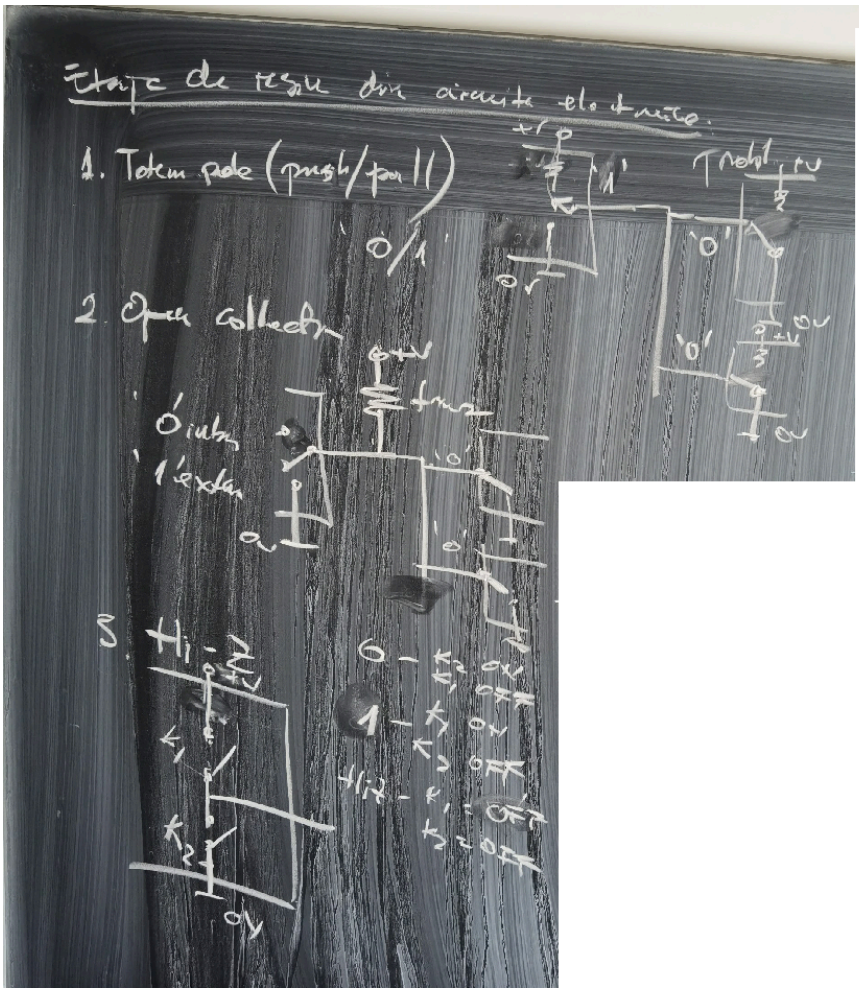
However, common use reduced ISDN to be limited to Q.931 and related protocols, which are a set of protocols for establishing and breaking circuit switched connections, and for advancedcalling features for the user. They were introduced in 1986.[2]

In a videoconference, ISDN provides simultaneous voice, video, and text transmission between individual desktop videoconferencing systems and group (room) videoconferencing systems.

Medii de transmisie:

- Cablu coaxial
- Cablu torsadat
- Fibra optica
- Wireless

Etaje de iesire in dispozitive digitale


Etapa de ieșire din circuite electronice.
1. Totem pole (push/pull)
2. Open collector
3. Hi-Z

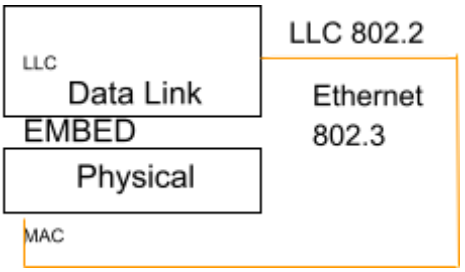
Curs 3

Ethernet si identificarea calculatoarelor la nivel local

Nivelul legătură de date asigură interfaţa între componenta fizică a reţelei şi componenta software. Tehnologiile de la acest nivel au scopul de a partaja accesul mai multor utilizatori la mediu de transmisie. Scopurile urmărite sunt:

- eficienţa mare;
- întârziere mică;
- rezistenţă la erori;
- echilibrarea traficului între utilizatori.

Ethernet este denumirea unei familii de tehnologii de reţele de calculatoare, bazate pe transmisia cadrelor (frame) şi utilizate la implementarea reţelelor locale de tip LAN. Numele provine de la "eter", care multă vreme s-a crezut că este mediul în care acţionau şi comunicau zeităţile. Ethernetul se defineşte printr-un şir de standarde pentru cablare şi semnalizare aparţinând primelor două nivele din Modelul de Referinţă OSI - nivelul fizic şi legătură de date.



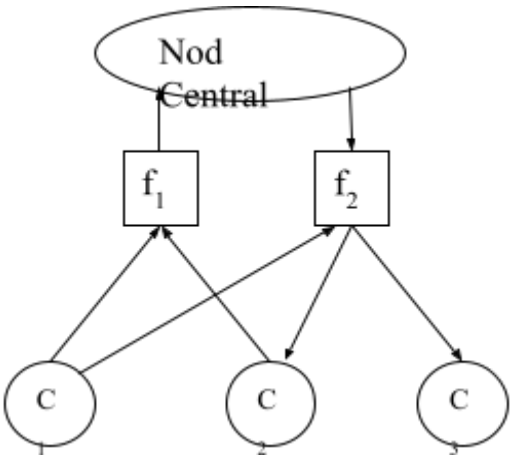
Ethernetul este standardizat de IEEE în seria de standarde 802.3. Aceste standarde permit transmisia datelor prin mai multe medii fizice: cabluri coaxiale, folosite în primele reţele Ethernet; cabluri torsadate - pentru conectarea sistemelor individuale la reţea; cabluri de Fibră optică - pentru structura internă (backbone) a reţelei.

Aşa cum se întâmplă pentru orice protocol din familia IEEE 802, nivelul legătură de date din modelul ISO este divizat în două subnivele IEEE 802 : Logical Link Control (LLC) si Media Access Control (MAC). LLC este specificat de standardul 802.2 Acest subnivel asigură interfaţa între subnivelul Ethernet MAC şi nivelele superioare din stiva de protocoale cu care lucrează ETD. El este definit pentru toate standardele IEEE 802 în general (nu este specific standardului 802.3). sub acesta se regaseste specificatia Ethernet cu:

• Subnivelul **MAC** controlează accesul nodului la resursele nivelului fizic şi este specific fiecărui protocol în parte. Toate subnivelele MAC ale protocoalelor IEEE 802.3 trebuie să întrunească un set de cerinţe de bază. Există de asemenea o serie de cerinţe definite în extensii ale protocolului, care sunt opţionale. Singurul criteriu impus pentru o legătură primară (care nu include extensii opţionale ale protocolului) între două noduri ale reţelei este că nivelul MAC al ambelor noduri trebuie să suporte aceeaşi rată a transmisiei.

• Subnivelul **Nivelul fizic** al protocoalelor 802.3 este caracterizat de către rata de transmisie dorită, de reprezentarea (codarea) semnalului prin mediul de transmisie şi de mediul de transmisie care interconectează două noduri ale reţelei. În ceea ce priveşte diferitele versiuni ale nivelului fizic ne putem referi la 10Mbps pe cablu coaxial (deja învechită), 100Mbps pe fire torsadate şi 1Gbps pe fire torsadate sau fibră optică.

Începutul ethernet s-a realizat în Hawaii şi s-a numit protocol ALOHA. Acesta implementează mecanisme elementare ethernet. În cazul apariţiei unei coliziuni (2 calculatoare transmit simultan) se aşteaptă un interval de timp aleatoriu înaintea retransmisiei. Structura reţelei ALOHA era următoarea:



Nodul central repetă către toate celelalte calculatoare informaţia primită.

Dezavantajul era în cazul unei încărcări mari a reţelei deteriorarea rapidă a performanţei prin transmiterea cadrelor care au coliziuni.

În urma îmbunătăţirilor a fost dezvoltat protocolul ***CSMA / CD*** folosit de ethernet. Protocolul CSMA / CD (*Carrier Sense Multiple Access with Collision Detection*) a fost conceput iniţial pentru a permite mai multor staţii ce folosesc acelaşi mediu fizic de comunicaţie să comunice între ele, să partajeze această resursă în absenţa unui switch. Acest protocol nu necesită prezenţa unui management centralizat, a unor token-uri de acces sau a unor intervale dedicate de transmisie pentru fiecare staţie. Nivelul MAC al fiecărei staţii va determina momentul în care staţia respectivă poate să intre în emisie. Regulile de acces ale protocolului sunt indicate în mare măsură de numele acestuia :

• Carrier Sense - sesizarea purtătoarei : Fiecare staţie "ascultă" permanent traficul pentru a şti când are dreptul de a transmite;

• Multiple Access - oricare dintre staţii poate accesa mediul (transmite) dacă nu există trafic;

• Collision Detect- dacă două sau mai multe staţii aflate în acelaşi ***domeniu de coliziune*** (în aceeaşi reţea CSMA / CD) încep transmisia în aproximativ acelaşi moment de timp, semnalele provenite de la cele două staţii vor interfera (intra în coliziune), făcând ca ambele transmisii să fie deteriorate. Dacă se întâmplă acest lucru, fiecare dintre staţii trebuie să detecteze această coliziune înainte de sfârşitul transmisiei cadrului eronat, să se oprească din transmisie şi să încerce o retransmisie după un interval de timp cvasi-aleatoriu.

În primele reţele Ethernet, semnalele transmise de către staţii erau regenerate cu ajutorul unui dispozitiv denumit hub. Acest dispozitiv primea semnalul de la una dintre staţii (pe unul dintre porturile sale) şi îl transmitea spre toate celelalte porturi (cu excepţia celui prin care recepţionase semnalul). Din acest motiv toate calculatoarele intrau în competiţie pentru accesarea mediului de comunicaţie (semnalele provenite de la oricare dintre cel puţin două staţii puteau intra în coliziune).

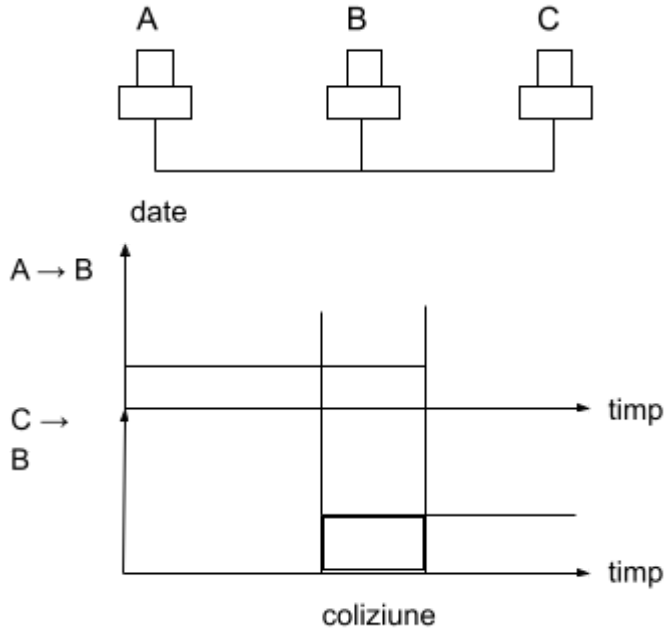
Apariţia switch-urilor (comutatoarelor de reţea) inteligente a însemnat din punctul acesta de vedere un important pas înainte. Astfel, fiecare port al unui switch oferă acces fiecărei staţii la o reţea de debit înalt (de exemplu într-o reţea fast-ethernet fiecare port poate să transmită un trafic de 100 Mbps). Mai mult decât atât, switchurile sunt capabile să filtreze traficul pe baza adresei ethernet de destinaţie a pachetului şi să transmită semnalul recepţionat

de la un anumit calculator doar pe portul ce direcţionează acest semnal spre staţia destinaţie, în loc de a-l difuza spre toate celelalte porturi.

Prin aceasta se subîmparte domeniul de coliziune posibil în mai multe subdomenii cu doar doi participanţi : portul switchului şi placa de reţea a staţiei ce este conectată la acel port. Întrucât fiecare participant este acum într-un domeniu privat de coliziune, ce îl include doar pe el însuşi şi un port al switchului, lăţimea de bandă a fiecărui utilizator al reţelei creşte semnificativ, fără a fi nevoie de o modificare a vitezei legăturii pe linia respectivă.

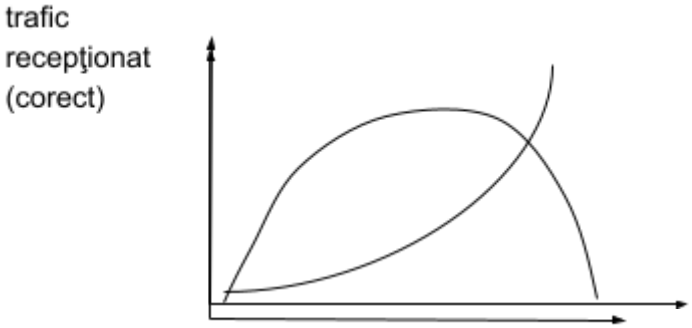
Domeniile de coliziune mai mici duc la mai putine coliziuni; deci echipamentele recomandate sunt switch-urile (sau routerele) in locul hub-urilor.

Pentru a se asigura transmiterea fără coliziuni a unui pachet de date un calculator trebuie să fie în stare să depisteze coliziunea înaintea terminării transmiterii pachetului.



Din acest motiv există o dimensiune minimă a unitatii de protocol. În mod normal aceasta respectă următoarea relaţie: $T_{propagare\ minim} > 2\ T_{propagare\ reţea}$.

Cu cât timpul petrecut pentru detectarea coliziunilor şi cauzarea acestora este mai mare cu atât reţeaua este mai ineficientă. Într-o reţea ethernet comportamentul poate fi descris prin graficul de mai jos:



În practică dimensiunea minimă a unui 2 biţi.

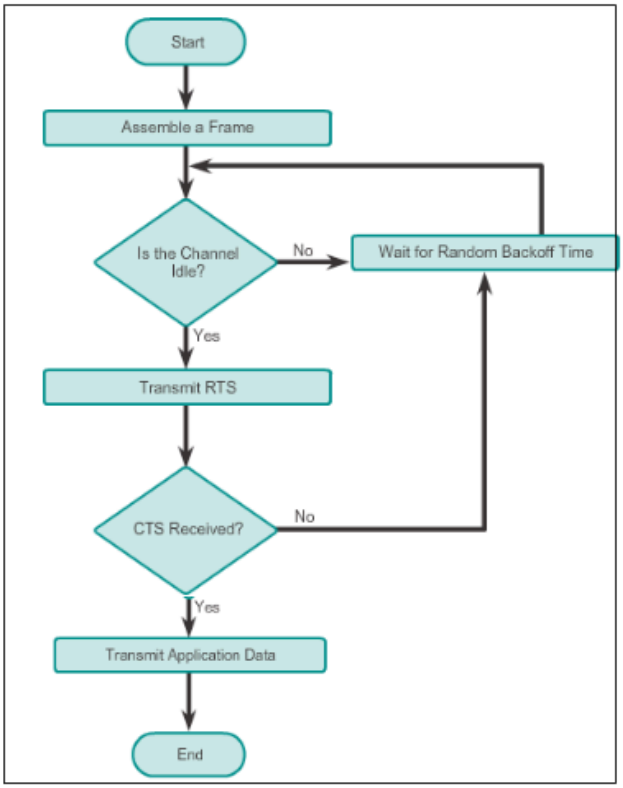
Pentru cablu coaxial distanţa maximă între 2 receptoare este de 500 m => timpul de propagare este de aproximativ 6 μ secunde => timpul minim de transport al unui pachet trebuie să fie mai mare de 12 μ secunde.

Pentru retele Wireless se foloseste CSMA/CA

Wireless Operation

CSMA/CA

CSMA/CA Flowchart



Evitarea coliziunilor este utilizată pentru a îmbunătăți performanța metodei CSMA prin încercarea de a împărți canalul în mod oarecum egal între toate nodurile de transmisie din domeniul de coliziune.

1. Carrier Sense: înainte de transmitere, un nod ascultă mai întâi mediul partajat (cum ar fi ascultarea de semnale fără fir într-o rețea fără fir) pentru a determina dacă un alt nod transmite sau nu. Rețineți că problema nodului ascuns înseamnă că poate transmite un alt nod care nu este detectat în această etapă.

2. Evitarea coliziunilor: dacă s-a auzit un alt nod, așteptăm o perioadă de timp (de obicei aleatorie) ca nodul să nu mai transmită înainte de a asculta din nou pentru un canal de comunicații liber.

• Solicitarea de trimitere/Clear to Send (RTS/CTS) poate fi utilizată opțional în acest moment pentru a media accesul la mediul partajat. Acest lucru ajută într-un fel să atenueze problema nodurilor ascunse, deoarece, de exemplu, într-o rețea fără fir, punctul de acces emite doar un Clear pentru a trimite către un nod la un moment dat. Cu toate acestea, implementările wireless 802.11 nu implementează de obicei RTS/CTS pentru toate transmisiile; ei îl pot opri complet sau cel puțin să nu îl folosească pentru pachete mici (supratenția pentru RTS, CTS și transmisie este prea mare pentru transferuri de date mici).

• Transmisie: dacă mediul a fost identificat ca fiind clar sau nodul a primit un CTS pentru a indica în mod explicit că poate trimite, acesta trimite cadrul în întregime. Spre deosebire de CSMA/CD, este foarte dificil pentru un nod wireless să asculte în același timp cu transmiterea (transmisia sa va depăși orice încercare de a asculta). Continuând exemplul fără fir, nodul așteaptă primirea unui pachet de confirmare de la punctul de acces pentru a indica că pachetul a fost primit și a fost verificat corect. Dacă o astfel de confirmare nu sosește în timp util, se presupune că pachetul s-a ciocnit cu o altă transmisie, determinând nodul să intre într-o perioadă de backoff exponențial binar înainte de a încerca retransmiterea.

Structura unui cadru (frame) Ethernet este următoarea:

7 octeţi	1 octet	6 octeţi	6 octeţi	2 octeţi	1 octet	1 octet
Preambul	Delimitator început frame	Adresă MAC destinaţie	Adresă MAC sursă	Tip protocol	Date	CRC

Iniţial placile de teţea foloseau drept oscilatoare circuite a caror frecventa varia in funtie de conditiile de mediu (temperatura). **Preambulul** este o secvenţă binară de forma: 0101... folosită pentru sincronizarea acestor echipamente. În prezent preambulul are doar rol de compatibilitate deoarece inechipamentele moderne frecventa de oscilatie este mult mai nprecisa (nu are variatii).

Tip protocol indică tipul protocolului încapsulat în frame.

CRC – ul este o sumă de verificare pe 32 de biti ce permite depistarea erorilor, modul de lucru se bazeaza pe proprietatea functiei sau exclusiv:

Se efectuează un XOR la sursă şi un XOR la destinaţie. Dacă valorile CRC calculate la sursă şi la destinaţie coincid codul este corect.

Plăcile de reţea au inscripţionate hardware o **adresă numită MAC(Media Access Control)**. Aceasta are 48 biţi organizaţi astfel:

- primii 24 identifică producătorul;
- următorii 24 identifică seria produsului.

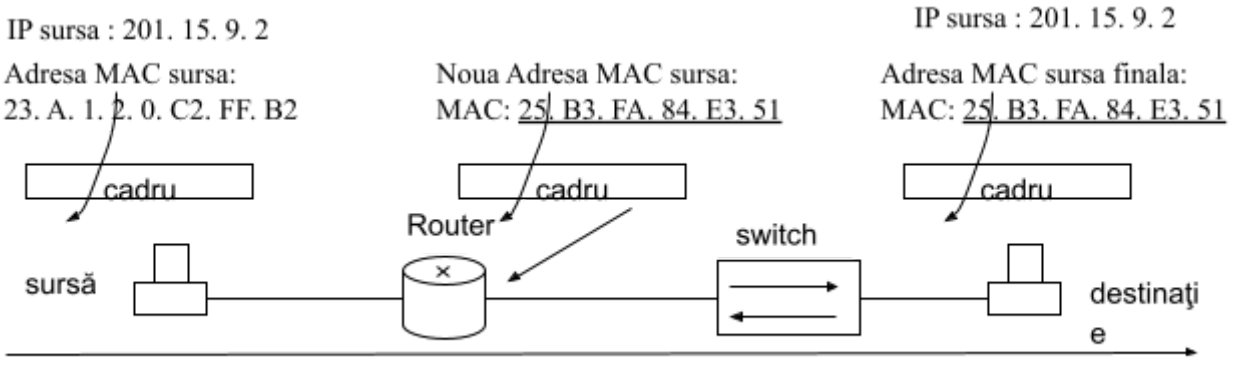
Switch–urile îşi construiesc un tabel cu adresele MAC. Pe baza acestuia datele sunt transferate între porturile switch–ului sursă şi destinaţie fără a afecta şi celelalte porturi din switch.

Adresa MAC poate fi folosită pentru a filtra traficul în internet.

Adresa MAC este utilizată numai la nivel local: sunt prezente echipamente de retea de nivelul 1 si 2 (hub, switch, repetor, bridge) si nu trece de routere.

Noua Adresa MAC sursa:

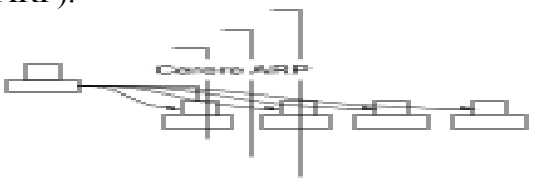
MAC: 25. B3. FA. 84. E3. 51

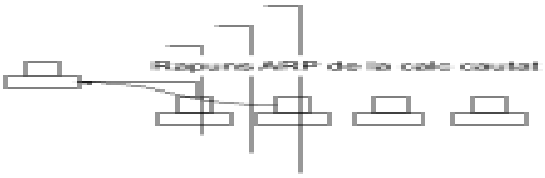


De la sursă la destinaţie adresele IP se păstrează iar adresele MAC se modifică la fiecare trecere printr-un router, dar nu la trecerea prin alte echipamente: switch, hub. Adresele folosite pentru identificarea la niveulul internetului (care raman nemodificate de la sursa la destinatie) sunt adresele IP.

O placã de retea Ethernet are adresa pe 48 de biti, adresã stabilitã de cãtre producãtorul plãcii, iar aceastã adresã nu poate fi modificatã. Ca o consecintã, dacã o placã de retea se defectează si aceasta se înlocuieste, masina în cauzã va avea o altã adresã fizicã. Mai mult, deoarece adresa Ethernet este pe 48 de biti, nu este nici o posibilitate de a o codifica pe cei 32 de biti ai adresei IP.

Solutia aleasã permite ca o nouã masinã sã fie adãugatã în retea fãrã a recompila codul, si nu necesitã mentinerea unei baze de date centralizatã. Pentru evitarea mentinerii unui tabel de mapare centralizat, proiectantii Internet au ales un protocol de nivel scãzut care leagã adresele dinamic. Acest protocol este cunoscut sub numele de "Address Resolution Protocol" (ARP).





Ideea pe care se bazează rezolutia dinamicã cu ARP este simplã si este prezentatã schematic în figura precedenta. Dacã o statia A doreste sã rezolve adresa IP a statiei B IB, trimite un pachet special prin broadcast la toate statiile din retea, prin care cere statiei cu adresa IP IB sã rãspundã cu adresa sa fizicã PB. Toate statiile receptionează pachetul, dar doar statia B îsi recunoaste propria adresã IP si rãspunde statiei A, cãreia îi cunoaste adresa fizicã chiar din pachetul receptionat. Dupã ce statia A a receptionat rãspunsul va trimite pachetele statiei B folosind adresa ei fizicã. ARP permite unei statii sã afle adresa fizicã a unei alte statii conectate la aceeasi retea fizicã, furnizând doar adresa IP a statiei destinatie.

Pentru a reduce comunicatiile inutile, statiile care folosesc ARP mentin în cache cele mai recente adresele IP rezolvate si adresele fizice corespunzãtoare. Când o statie primeste un rãspuns la o cerere ARP, ea salveazã în cache adresa IP a masinii si adresa fizicã corespunzãtoare, pentru cãutãrile ulterioare. Când statia doreste sã transmitã un pachet, ea se uitã prima datã dacã are în cache adresa fizicã pentru adresa IP doritã, dacã o are o foloseste pe aceasta, iar dacã nu o gãseste trimite un pachet ARP, si asteaptã rãspunsul cu adresa fizicã.

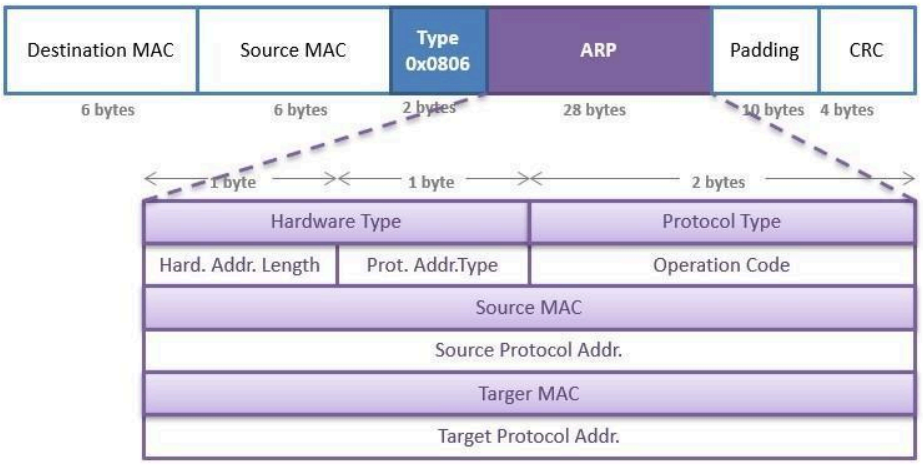
Intr-o tabelă ARP:

– putem avea mai multe adrese IP asociate cu un singură adresă MAC
– nu putem avea mai multe adrese MAC asociate cu un singură adresă IP
• Două metode de abordare:
– Default Gateway – reprezintă metoda dominantă
– Proxy ARP – folosit uneori în cazul ruterelor sau în cazul implementărilor stivei TCP/IP pe platfome embeded

Proxy ARP

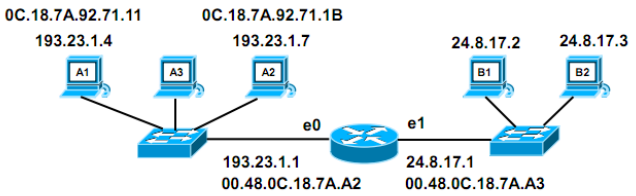
Mesajele trimise catre alte retele nu pot afla adresa MAC a calculataorelor destinatie. Din acest motiv la orice cerere ARP primita de un rotuer pentru un echipamnt care nu se afla in reteaua locala, ruterul

Pachetul ARP:



Alte informatii:
FF-FF-FF-FF-FF-FF = Broadcast
0x0806 = ARP Message
op field – ARP request = 1
ARP reply = 2
RARP request = 3
RARP reply = 4

Exemplu incarcare comnunicatie de la PC 193.23.1.4 catre PC 24.8.17.3:

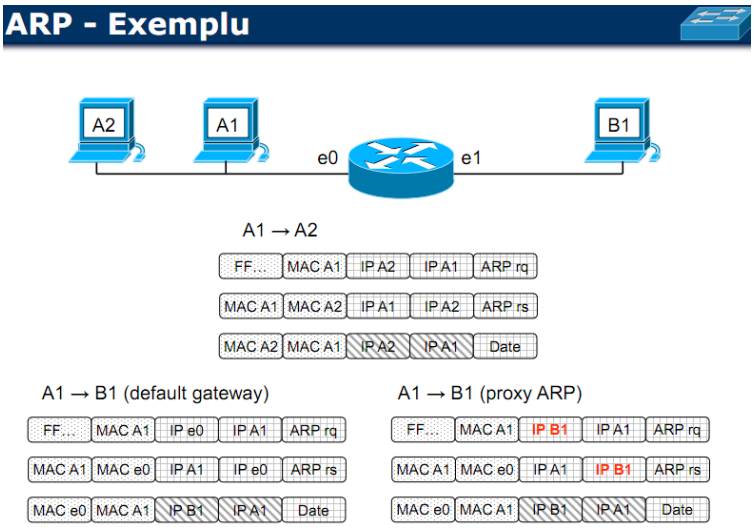


←	Antet	→		←	Date	→	
MAC dest.	MAC sursă	Tip cadru	cod operație	MAC sursă	IP sursă	MAC dest.	IP dest
FFFF: FFFF: FFFF	0C18: 7A11: 7111	0x0806	1	0C18: 7A11: 7111	193.23. 1.4	0000: 0000: 0000	193.23. 1.7

←	Antet	→		←	Date	→	
MAC dest.	MAC sursă	Tip cadru	cod operație	MAC sursă	IP sursă	MAC dest.	IP dest
0C18: 7A11: 7111	0C18: 7A92: 711B	0x0806	2	0C18: 7A92: 711B	193.23. 1.7	0C18: 7A11: 7111	193.23. 1.4

←	Antet 2	→	← Antet 3 →		←	Date	→
MAC dest.	MAC sursă	Tip cadru	IP dest	IP sursă			
0C18: 7A92: 711B	0C18: 7A11: 7111	0x0800	193.23. 1.7	193.23. 1.4			

Exemplu proxy ARP:



Format cadru in retele wireess WiFi 802.11

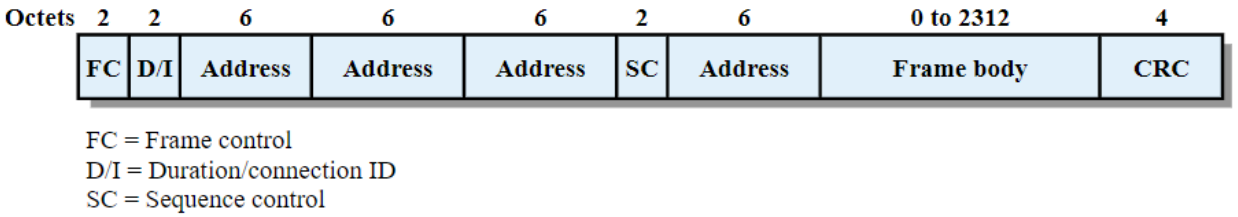


Figure 1: IEEE 802.11 MAC frame format. Image from William Stallings "Data and Computer Communications".

Următorul fragment din William Stallings „Date and Computer Communications” explică aceste domenii:

• Frame Control: indică tipul de cadru (control, management sau date) și oferă informații de control. Informațiile de control includ dacă cadrul este către sau de la un DS, informații despre fragmentare și informații despre confidențialitate.

• Duration/Connection ID: Dacă este folosit ca câmp de durată, indică timpul (în microsecunde) în care canalul va fi alocat pentru transmiterea cu succes a unui cadru MAC. În unele cadre de control, acest câmp conține un identificator de asociere sau conexiune.

• Adrese: numărul și semnificația câmpurilor de adresă pe 48 de biți depind de context. Adresa transmițătorului și adresa receptorului sunt adresele MAC ale stațiilor conectate la BSS care transmit și primesc cadre prin LAN fără fir. ID-ul setului de servicii (SSID) identifică rețeaua LAN fără fir

pe care se transmite un cadru. Pentru un IBSS, SSID este un număr aleatoriu gen-

erate în momentul formării rețelei. Pentru o rețea LAN fără fir care face parte dintr-un

configurație mai mare, SSID-ul identifică BSS-ul prin care este transmis cadrul; în mod specific, SSID-ul este adresa la nivel MAC a AP-ului pentru acest BSS

(Figura 17.4). În cele din urmă, adresa sursă și adresa de destinație sunt adresele MAC ale stațiilor, fără fir sau de altă natură, care sunt sursa finală și

destinația acestui cadru. Adresa sursă poate fi identică cu cea de transmisie

adresa terță și adresa de destinație pot fi identice cu cele ale destinatarului
abordare.

• Controlul secvenței: conține un subcâmp de număr de fragment de 4 biți, utilizat pentru fragmentare și reasamblare, și un număr de secvență de 12 biți utilizat pentru numerotare
cadre transmise între un anumit emițător și receptor.

• Frame Body: Conține un MSDU sau un fragment al unei MSDU. MSDU este o unitate de date de protocol LLC sau informații de control MAC.

• Secvență de verificare a cadrelor: o verificare a redundanței ciclice pe 32 de biți.

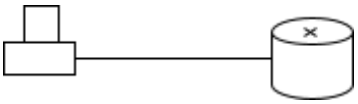
Topologii de retele de calc

Nodurile reţelei, interconectate prin mediile fizice, pot fi împărţite în două mari categorii:

- **echipament terminal de date (ETD):** dispozitive care sunt fie sursa, fie destinaţia cadrelor de date. Aceste dispozitive pot să fie PC-uri, servere de fişiere sau imprimante.
- **echipament de comunicaţii de date (ECD):** dispozitive intermediare care recepţionează şi rutează cadre în interiorul reţelei. Aceste echipamente pot fi dispozitive de sine stătătoare (hub-uri, switchuri sau routere) sau echipamente de interfaţare a reţelei cu staţia de lucru (plăci de reţea sau modemuri).

Topologia unei retele priveste modalitatea de conectare între nodurile unei reţele

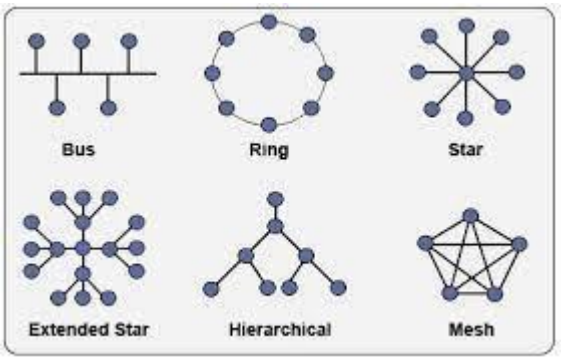
Există numeroase tipuri de topologii posibile pentru LAN-uri, dar toate acestea sunt combinaţii a trei tipuri de interconectări de bază. Cea mai simplă dintre acestea este o conexiune de tipul **punct-lapunct** care poate fi stabilită în mai multe moduri: ETD-ETD, ETD-ECD sau ECD-ECD. Un exemplu de asemenea conexiune este ilustrat în figura urmatoare:



Primele reţele Ethernet, implementate pe cablu coaxial, aveau o topologie de tip **magistrală (bus)**. Lungimea unui segment de magistrală putea ajunge până la 500m, şi pe acest segment se puteau lega până la 100 de staţii. Diferitele segmente erau interconectate prin intermediul unor repetoare sau hub-uri. Distanţa totală între staţiile situate în punctele “extreme” ale reţelei nu putea să depăşască o valoare maximă dată.

Încă de la începutul deceniului trecut, cea mai întâlnită topologie de reţea Ethernet este aceea în **stea**. Unitatea centrală într-o asemenea topologie este fie un repetor multiport (cunoscut de asemenea sub numele de hub) sau un switch (comutator de reţea). Toate conexiunile într-o topologie stea sunt de tipul punct-la-punct implementate pe cablu torsadat sau fibră optică. O evolutie este topologia de **stea extinsa.**

Alte topologii:
Inel: folosita de **Fiber Distributed Data Interface**;
Mesh (retele wireless 802.11);



Toppologii logice: sunt legate de modul de transmitere a informatiei, deci o clasificare in functie de configurarea software:

- Ex o retea ethernet este bus din punt de vedere logic insa stea dpdv fizic
- Token ring (802.5) este inel din punt de vedere logic insa stea dpdv fizic

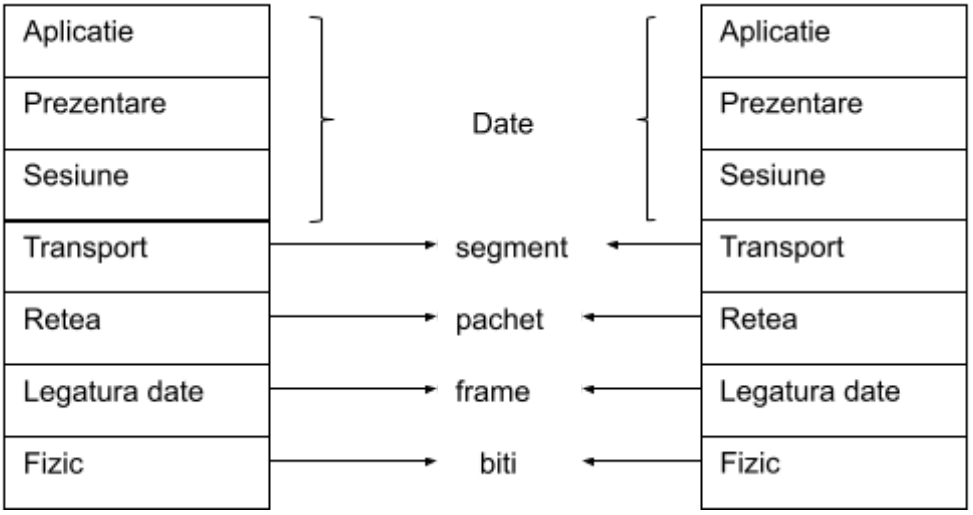
Curs 3

Nivelul de retea si protocolul internet (IP)

Nivelul de retea este responsabil de transmiterea secventelor de date de lungime variabila de la echipamentul sursa la cel destinatie la nivelul internetului, echipamente identificate prin adrese unice.

Cateva din protocoalele specifice nivelului retea sunt: IP versiunea 4, IP versiunea 6, ICMPv4 si ICMPv6, RIP, IP Sec, BGP. Unitatea de protocol la nivelul retea este pachetul.

Modelul de comunicatie este cel fara conexiune, in care pachetele cu date sosesc la destinatie fara a fi anuntate in prealabil. Acest lucru poate duce la pierderea pacehtelor in cazul in care echipamentul destinatie nu este pregatit pentru a le receptiona.



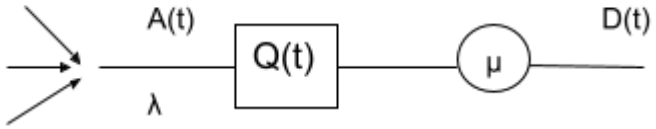
Nivelul retea din modelul OSI are un echivalent numit nivelul internet in modelul TCP/IP, insa exista diferente intre cele doua niveluri, specificatia din modelul TCP/IP fiind de fapt un subset al specificatiei din nivelul OSI. Nivelul internet este special dedicat suitei de protocale care folosiesc protocolul internet (IP).

Echipamentele folosite in retele de calculatoare la acest nivel sunt ruterele.

Modelul unui dispozitiv router

La transmitere (sursa) informaţia este împachetată la sursa şi la receptie (destinaţie) este despachetată. Routerele iau decizii legate de adresele valabile la nivelul întregului internet.

Routerul este echipamentul de reţea care direcţionează pachetele de date la nivelul internetului. Modelul sa poate fi reprezentat astfel:

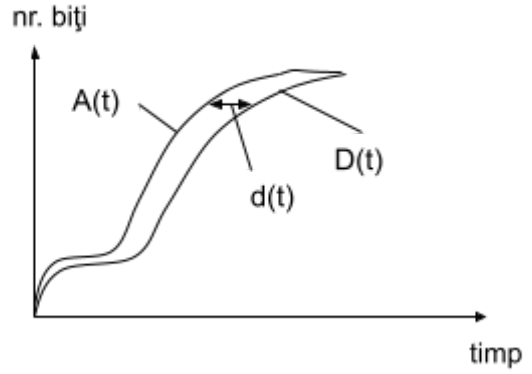


A(t) – numărul de sosiri de pachete pe intervalul de timp [0, t]
λ – valoarea medie a sosirilor de pachete
D(t) – numărul de plecări de pachete pe intervalul [0, t]

$\frac{1}{\mu}$ $\frac{1}{\mu}$ – valoarea medie de timp necesară tratării unui pachet

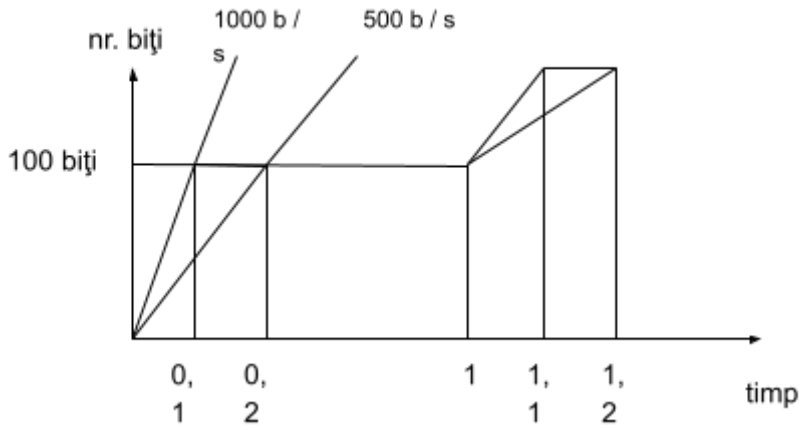
Q(t) = A(t) – D(t)

Q(t) – reprezintă ocuparea cu biţi a cozii de pachete



d(t) – întârzierea cozii de procesare
d(t) este timpul pe care îl pierde un bit din coada de pachete de la sosirea sa până la tratare.

Exerciţiu 1.
În fiecare secundă sosește un grup de 100 biţi cu viteza de 1000 b / s. Cunoaştem viteza de plecare, 500 b/s. Care este numărul de biţi care ocupă în medie coada de aşteptare ?



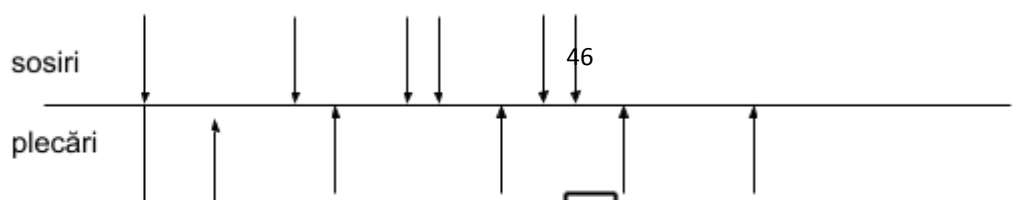
0 – 0,2 buffer ocupat
0,2 – 1 buffer gol

Încărcarea medie cât timp bufferul are biţi:

$$\overline{Q(t)} = \frac{1}{2}\left(500\frac{b}{s} \times 0{,}1\ s\right) = 25\ biţi$$

$$\overline{Q_{t_{tot}}} = 0{,}2 \cdot 25 = 5\ biţi$$

Evoluţia în timp a cozii de pachete:



Exerciţiu 2.

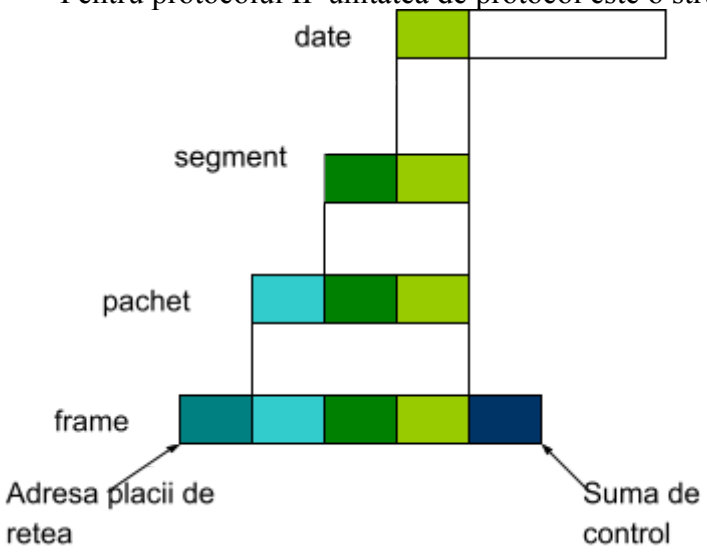
Sa se calculeze pentru exemplul de mai sus cantitatea de date care trebuie sa intre in router in fiecare secunda cu viteza specificata mai sus pentru a umple un buffer de 2kb in 30 de secunde.

1000 b / s -500 b / s =500 b / s, maxim pot sa intre in router, care pot umple buffer-ul de 2kb in aprox 4 secunde.

Pentru a fi umplut in 30 de secunde, tb sa ramana 2kb/30=67 biti in buffer in fiecare secunda. Deci viteza de intrare este de 567kb/s.

Rafalele de date (grupuri de pachete rapide) cresc întârzierea. În reţelele de calculatoare însă sosirile sunt aleatoare => dacă valoarea medie a sosirilor este egală cu valoarea ieşirilor echipamentele se vor comporta în parametrii. Deci predictibilitatea pachetelor minimizează întârzierea.

Pentru protocolul IP unitatea de protocol este o structură numită pachet.



Antetul este adaugat la informatia primita de la nivelul superior si contine numeroase informatii intre care cele mai importante sunt prezentate in conitnuare :

Structura unităţii de protocol IP v4

Versiunea	HLEN	TOS	Lungimea totală	
ID			Indici	Deplasament fragmentare
TTL		Protocol	Sumă verificare	
IP Sursă				
IP Destinaţie				
Opţiuni				PAD

Antent IPv4:

• versiune – precizeaza versiunea protocolului

• lungime antet – precizeaza lungimea antetului. Poate avea între 20 şi 60 de octeţi (multipli de 4 octeţi) si în vasta majoritate a cazurilor antul IP va fi doar de 20 octeti

• TOS (Type of Service) – folosit pentru implementarea QoS

• TTL (Time To Live) – folosit pentru diminuarea efectului buclelor. Fiecare ruter va decrementa valoarea acestui câmp iar un pachet cu TTL 1 nu va părăsi reţeua locală. Este singurul câmp din antetul de nivel 3 ce este modificat la trecerea printr-un ruter.

• suma de control a antetului va fi recalculată de fiecare ruter, doar datorită operaţiei de decrementare a TTL. Valorile acestui câmp sunt comune în IPv4 şi IPv6

• Indică tipul următorului antet: 1 – ICMP pentru IPv4; 2 – IGMP pentru IPv4; 4 – IP in IP; 6 – TCP; 17 – UDP ; 41 – IPv6; 58 – ICMP pentru IPv6; 59 – nu mai exista alt antet

• lungime pachet [16] defineşte lungimea pachetului. Limitează la maxim 64KB dimensiunea unei datagrame.-

• identificator secvenţă [16] identifică datagrama IP.

• flags [3]

– bitul 49 – rezervat (are valoare 0)

– bitul 50 – DF (“do not fragment”)

– bitul 51 – MF (“more fragments”)

• offset [13]

– defineşte poziţia fragmentului în cadrul datagramei IP

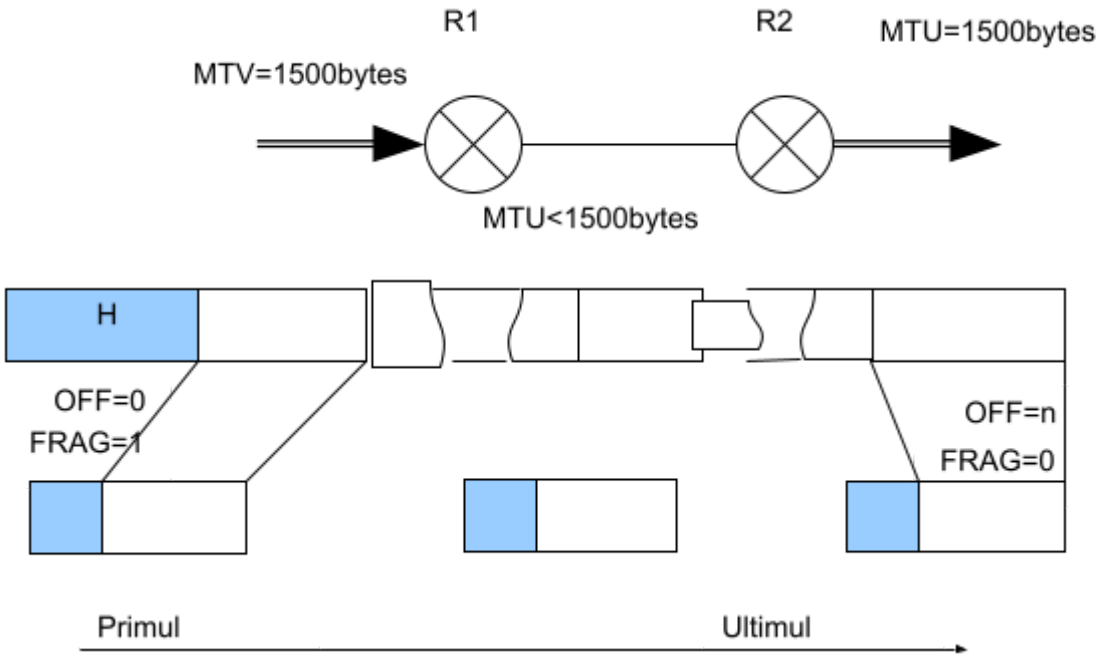
– RFC 3514

Protocolul IP are următoarele caracteristici:

- este fără conexiune (datele pot veni în orice ordine);
- nu este sigur (poate să piardă pachete);
- este “best effort” (face cel mai mare efort pentru a duce pachetele la destinaţie);
- pachetele sunt transmise individual.

Deplasamentul datorat fragmentării

Un router poate primi pachete mai mari decât dimensiunea maximă permisă de reţea (MTU – maximum transmision unit). In cazul in care route-rul primeste un pachet mai mare decat unitatea de transmisie maxima permisa (MTU) el va fragmenta pachetul .



MTU (Maximum Transfer Unit) reprezinta dimensiunea maximă a datelor înainte de encapsularea de nivel 2. Ethernet defineşte dimensiunea maximă a datelor înainte de encapsulare 1500 octeti. Fragmentele sunt reasamblate la destinaţie nu de routerele intermediare. Pentru evitarea fragmentării se recomandă folosirea celui mai mic MTU regăsit de-al lungul căii de comunicaţie.

MSS (Maximum Segment Size) reprezinta dimensiunea maximă a datelor înainte de encapsularea de nivel 4.

Tehnologia dominantă de transfer este TCP/IP/Ethernet. TCP nu limitează dimensiunea datelor dintr-un segment. IP defineşte dimensiunea maximă a datelor după encapsulare 64 KB

Fragmentarea este o operaţiune costisitoare. În implementările actuale se încearcă evitarea dublei fragmentării prin stabilirea dimensiunii maxime a datelor de la nivelul 3

MTU = MSS + 40

Adresarea în internet

Pentru a identifica un calculator la nivelul internetului se folosesc adresele IP.

Se folosesc 2 versiuni:

- IPv4 are adresele pe 32 de biţi care sunt grupati în 4 grupuri a câte 8 biţi, iar scrierea numerelor se face in baza 10
- IPv6 are adrese pe 128 de biţi , iar scrierea numerelor se face in baza 16

IPv4 a fost prima apărută şi deoarece echipamentele nu sunt compatibile cu IPv6 răspândirea acestuia din urmă se realizează foarte greu.

Echipamentele moderne IPv6 proiectate să fie compatibile cu IPv4 împachetează datele în format IPv4 când acestea traversează echipamente vechi.

Motivarea măririi numărului de biţi de adresă pentru modificarea calculatoarelor a venit la sfârşitul anului 1990 când explozia internetului a avut loc numerele de adrese IPv4 utilizate a crescut foarte mult existând temerea terminării adreselor de acest tip.

IPv6 asigură 2^{128} adrese, mai mult decât sunt necesare în viitorul apropiat.

Datorită lipsei de compatibilitate a echipamentelor au fost găsite soluţii pentru IPv4 de extindere al numarului de adrese.

Structura adreselor IPv4

Valoarea primului octet:

– Clasă A – valoarea primului octet este între 0 şi 127
– Clasă B – valoarea primului octet este între 128 şi 191
– Clasă C – valoarea primului octet este între 192 şi 223

• Clasele de adrese sunt caracterizate de o mască de reţea:

– Clasă A – 255.0.0.0
– Clasă B – 255.255.0.0
– Clasă C – 255.255.255.0

Masca de reţea trebuie să respecte condiţia de continuitate. Diferenţa ţine de raţiuni istorice deoarece din punct de vedere practic nu există diferenţe între o reţea şi o subreţea (atâta vreme cât este precizată masca de reţea. O adresă de reţea este o adresă ce are toţi biţii din câmpul de staţie 0. O astfel de adresă nu poate fi asociată de obicei unei staţii sau unei interfeţe de ruter.

A	0000 0111	$0-127$	R	U	U	U
B	1000 0111	$128-191$	R	R	U	U
C	1100 0111	$192-223$	R	R	R	U

Clasa 127.0.0.0 contine adresa de loop back (127.0.0.1).Aceasta identifica propriul calculator . Transmiterea comenzii (ping 127.0.0.1) verifica instalarea corecta a stivei TCP IP pe propriul calculator . Adresa nu se aloca calculatoarelor . Router_le folosesc mascarea pentru a extrage adresa de retea dintr_o adresa IP .

Masca are două formate de reprezentare:

– Zecimal: 255.255.0.0

– Prefixat: /16

Adresa de subreţea identifică reţeaua în care se află o staţie

• Fie staţia cu adresa IP 172.168.0.1 şi masca de subreţea

255.255.0.0 (/16)

– Adresa de subreţea se obţine prin folosirea operaţiei de ŞI logic (ŞI pe

biţi între adresa IP şi masca de subreţea)

10101100 10101000 00000000 00000001 – 172.168.0.1

11111111 11111111 00000000 00000000 – 255.255.0.0

10101100 10101000 00000000 00000000 – 172.168.0.0

– Adresa de subreţea este 172.168.0.0/16

– Se spune că staţia are adresa 172.168.0.1/16 sau că are adresa 172.168.0.1 cu masca de subreţea 255.255.0.0

Adresa de difuzare a unei reţele se mai numeste adresa de broadcast. Orice (sub)reţea are o adresă de broadcast folosită pentru a transmite un pachet către toate staţiile din reţea. Adresa de broadcast este adresa din reţea pentru care toţi biţii de staţie sunt 1.

• Exemplu:

– Adresa de staţie: 172.168.0.1

– Masca de subreţea: 255.255.0.0

– Primii 16 biţi sunt biţii de subreţea, ultimii 16 biţi sunt biţii de staţie

– Adresa de broadcast va fi, aşadar:

172.168.11111111.11111111

adică 172.168.255.255

• Un pachet trimis către adresa de broadcast a unei reţele se numeşte broadcast direcţionat

• Un pachet trimis către adresa 255.255.255.255 se numeşte broadcast global

Exemplu 1

$$IP \rightarrow 130.50.75.28 \wedge$$
$$\underline{\quad 255.255.00.0}$$

$$A.R.130.50.0.0.$$
$$A.B.130.50.255.255$$

Masca are bitii de 1 in partea de retea si bitii de 0 in partea de utilizator .

Pornind de la aceasta adresa putem construi adresa de broadcast . Adresa de broadcast are toti bitii de utilizator 1 la care se adauga adresa de retea .

Exemplu 2

$$IP \rightarrow 250.80.32.16 \wedge$$
$$\underline{\quad 255.255.255.0}$$

$$A.R.200.80.32.0.$$
$$A.B.200.80.32.255$$

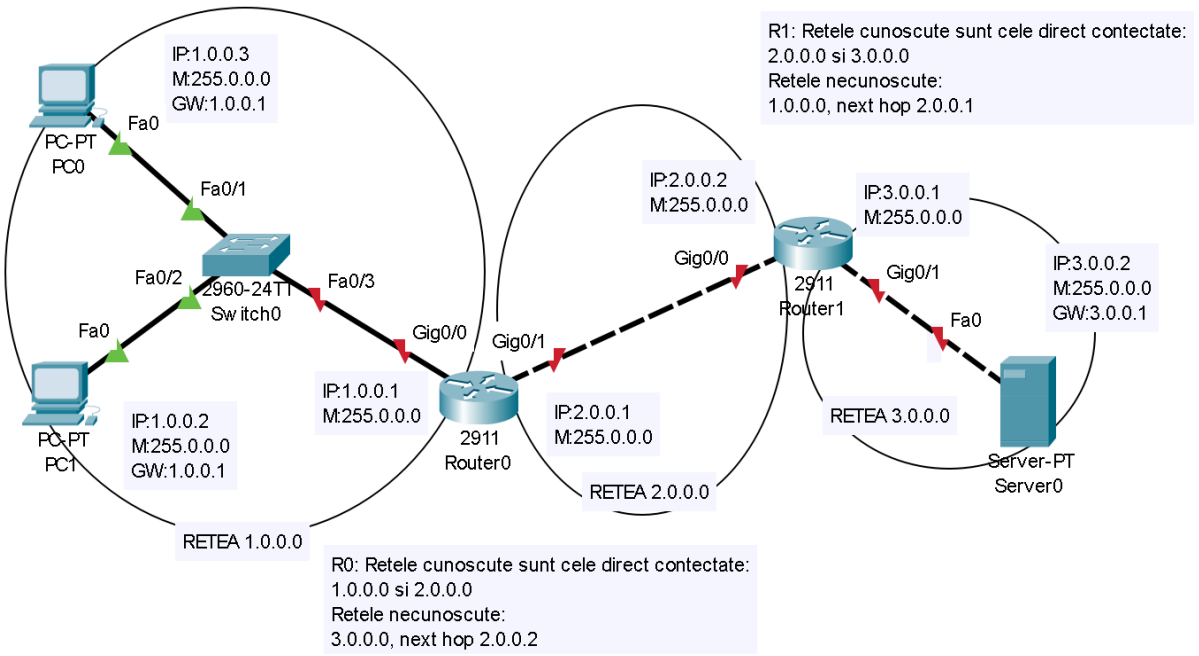
Adresele IP versiunea 4 sunt pe 32 biti . La sfarsitul anilor ’90 explozia site-urilor pe internet a dus la restrangerea adreselor IP disponibile .S-au propus mai multe solutii pentru a depasi aceasta problema .

-Folosirea versiunii IP V6. Adresele sunt pe 128 biti rezulta 2^{128} adrese de unde rezulta in continuare aproximativ 2^{95} adrese pentru fiecare locuitor al planetei . Desi solutia exista de mult timp exista probleme la implementare . Printre acestea enumeram :

- ✔ Dificultatea retinerii acestor adrese. Exemplu
  2001 :0DB8 :85AB :08D3:1319:8A2E:0370:7347
- ✔ Nu prezinta compabilitate cu IP V4
- ✔ Lipsa implementarii in sistemele de operare (xp)

O solutie a fost inpachetarea adreselor IP V6 in adrese IP V4.

Exemplu retea IPv4 cu 2 rutere care delimiteaza retelele si un switch care extinde reteaua in care se afla, permitand contectarea mai multor calculatoare



Folosirea adreselor private

O gama de adrese IP a fost rezervata pentru adrese locale .Aceste adrese nu pot fi folosite la nivelul internetului decat numai cele publice . Scopul este sa permita fiecarei organizatii sa aiba la dispozitie un numar mare de adrese locale la dispozitie, independent de numarul de adrese publice care ii sunt alocate.

Clasa A 10.*.*.*
Clasa B 172.16.*.*-172.31.*.*
Clasa C 192.168.*.*

Adresele private sunt valabile doar la nivel local, din acest motiv doua calculatoare din doua retele locale diferite pot avea acelasi IP fara sa apara conflict de adrese. Comunicatia intre cele doua retele locale este intermediata de rutere care au adrese publice pe interfata conectata la internet. Ruterele inlocuiesc intr-un pachet IP originar din retaua locala adresa privata cu adresa interfetei publice. Procedeul de transformare a adreselor private in adrese publice se numeste NAT (Network Address Translation). La receptionarea raspunsului, ruterul recunoaste sursa traficului si trimite in reteaua locala pachetul de date.

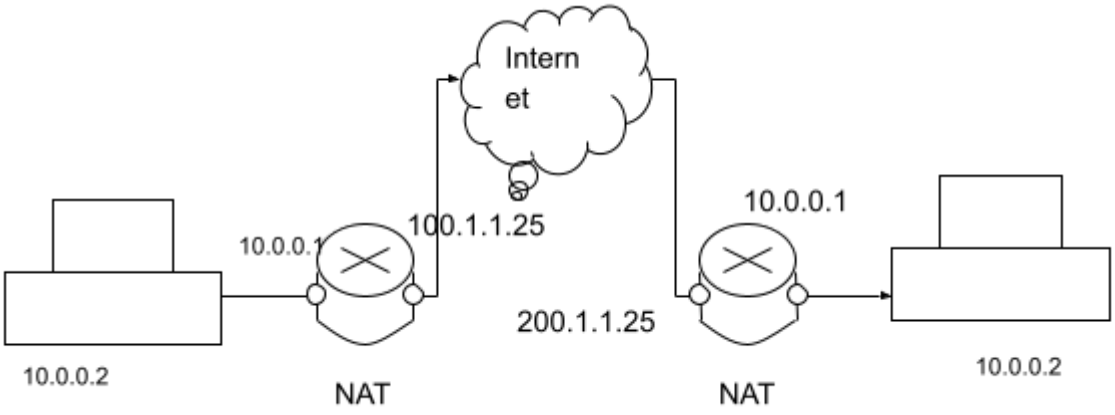


Fig 1 Calculatoare cu acelasi IP privat in retele diferite pot comunica deoarece prin internet datele folosesc IP-urile publice ale ruterelor

Pentru o utilizare eficienta a adreselor IP la nivelul local in ultimi ani sa realizat utilizarea subretelelor de calculatoare .

Aplicatie

O firma are alocata o adresa de retea de clasa C . Se doreste separarea retelei de contabilitate de a retelei generale .Prezntati solutii ?

Solutii

- Daca nu se utilizeaza subretelele ar mai fi nevoie de o clasa C . Daca firma are 12 calculatoare in reteaua de contabilitate si 12 pentru restul utilizatorilor se pierd 2*256-48=464 cazuri
- Daca se folosesc subretele . Subretelele implica inprumutarea de biti de la utilizator in partea de retea . Se imprumuta minim 2 biti.Cu 2 biti se creaza 4 subretele .

$$200.80.75.0.$$
$$\Downarrow$$
$$00 \text{---------} \begin{matrix} 200.80.75.0\ AR \\ 200.80.75.63\ AB \end{matrix}$$
$$01 \text{---------} \begin{matrix} 200.80.75.64\ AR \\ 200.80.75.127\ AB \end{matrix}$$
$$10 \text{---------} \begin{matrix} 200.80.75.128\ AR \\ 200.80.75.191\ AB \end{matrix}$$
$$11 \text{---------} \begin{matrix} 200.80.75.192\ AR \\ 200.80.75.255\ AB \end{matrix}$$

Nu se recomanda folosirea primei subretele deoarece coincide cu adresa de retea si nici a ultimei subretele deoarece coincide cu adresa de broadcast a retelei .Intr_o astfel de retea raman doar 2^6-12=52 adrese neutilizate in retea .

Formula de calcul pentru numarul de subretele create este : $2^x - 2$ unde x = numarul de biti imprumutati.

Exemplu ; 3 biti imprumutati rezulta 2^3-2=6 subretele utilizate .

Scopul introducerii subretelelor este pentru a micsora numarul de adrese IP pierdute si a creste securitatea retelei .

Aplicatie

O firma doreste sa imparta calculatoarele in sapte subretele utilizabile , stiind ca firma are alocata o adresa de clasa C .Precizati numarul de calculatoare posibile instalate in fiecare subretea ?

Solutie

2^4 – 2 = 14 calculatoare maxim 16

Curs 4

Scrierea simplificată a adreselor IP

Pentru a identifica un calculator nu este necesară utilizarea întregii adrese IP la nivelul internetului.

Adresele sunt împărţite în 2 secţiuni: adresa de teţea şi adresa de utilizator.

Pentru cele 3 clase repartizarea se face astfel:

Partea de utilizator este folosită doar în reţeaua locală.

Pentru a identifica reţeaua echipamentele router folosesc mascarea pentru a extrage partea de reţea.

```
   12.  1.  10.  200
  255.  0.   0.    0
---------------------------
AR  12.  0.   0.  0

   200.    5.    50.  4
   255.  255.  255.  0
---------------------------------
AR  200.  255.  255.  0
```

Adresele de reţea folosesc notarea simplificată în care numărul de biţi din mască este trecut după număr:

```
 12.   1.  10. 200  / 8 biţi
200.   5.  50.   4  /24 biţi
130.  20.   3.   3  /16 biţi
```

Subreţelele

Deoarece spaţiul de adrese trebuie utilizat cât mai eficient reţelele se împart în subreţele prin împrumutarea de biţi din zona utilizată pentru reţea.

Numărul minim recomandat de biţi împrumutaţi este 2 deoarece se poate confunda prima reţea cu prima subreţea.

```
1 bit  192.  168.  0.    0  /25
       192.  168.  0.  128  /25

2 biţi  192.  168.  0.    0  /26
        192.  168.  0.   64  /26
        192.  168.  0.  128  /26
        192.  168.  0.  192  /26
```

Prima subreţea nu este utilizată deoarece adresa de subreţea coincide cu adresa de reţea.

Transmisiile broadcast au toţi biţii din partea de utilizator 1.

Tentru subreţea adresa de broadcast este:

AB: 0011 1111 = $63_{(10)}$

AB: 0111 1111 = $127_{(10)}$

AB: 1011 1111 = $191_{(10)}$

AB: 1111 1111 = $255_{(10)}$

Deoarece adresa de broadcast a subreţelei coincide cu adresa de broadcast a reţelei nici ultima subreţea nu e utilizată.

192. 168. 20. 0 /27

192. 168. 20. 0 □ 31
192. 168. 20. 32 □ 63
192. 168. 20. 64 □ 95
192. 168. 20. 96 □ 127
192. 168. 20. 128 □ 159
192. 168. 20. 160 □ 191
192. 168. 20. 192 □ 223
192. 168. 20. 224 □ 255

Observatţie. Prima subreţea nu este utilizată pentru că adresa de subreţea coincide cu adresa de reţea.

Adresa de broadcast permite transmiterea datelor la toate calculatoarele din subreţea. Adresa de broadcast are partea de utilizator numai biţi de 1 (este ultima adresă din subreţea).

Adresa de broadcast a reţelei este 191. 168. 20. 255 , ea coincide cu adresa de broadcast din ultima subreţea => mesajele transmise în reţea vor ajunge în toate subreţelele prin urmare nici ultima subreţea nu se recomandă a fi utilizată.

192. 168. 20. 130 /27

Identificarea adresei de subreţea:
AR: 1000 0000 □192. 168. 20. 128
AB: 1001 1111 □ 192. 168. 20. 159

Exemplu.

Pentru reţeaua 192. 168. 20. 190 /28 precizaţi adresa de subreţea şi adresa de broadcast a subreţelei din care face parte. Poate fi adresa obţinută utilizată în reţea?

$190_{(10)}$ =10111110
AR: 10110000 176
AB: 10111111 191

AR: 192. 168. 20. 176
AB: 192. 168. 20. 191

Deoarece nu este nici adresă de reţea nici adresă de broadcast este adresă utilizabilă.

Exerciţiu. Îmtr-o reţea se recomandă împărţirea eficientă în subreţele. Pentru o adresă de clasă C alegeţi dimensiunea măştii astfel încât să acopere 6 săli de laborator cu 12 calculatoare în fiecare sală (se doreşte maximizarea numarului de subreţele).

12 calculatoare => $2^x - 2 \geq 12$ => 4 biţi pe partea de utilizator
8 – 4 = 4 biţi rămaşi pentru partea de reţea
$2^4 - 2 > 6$ => corespunde
În concluzie masca care poate fi folosita este /28.

VLSM

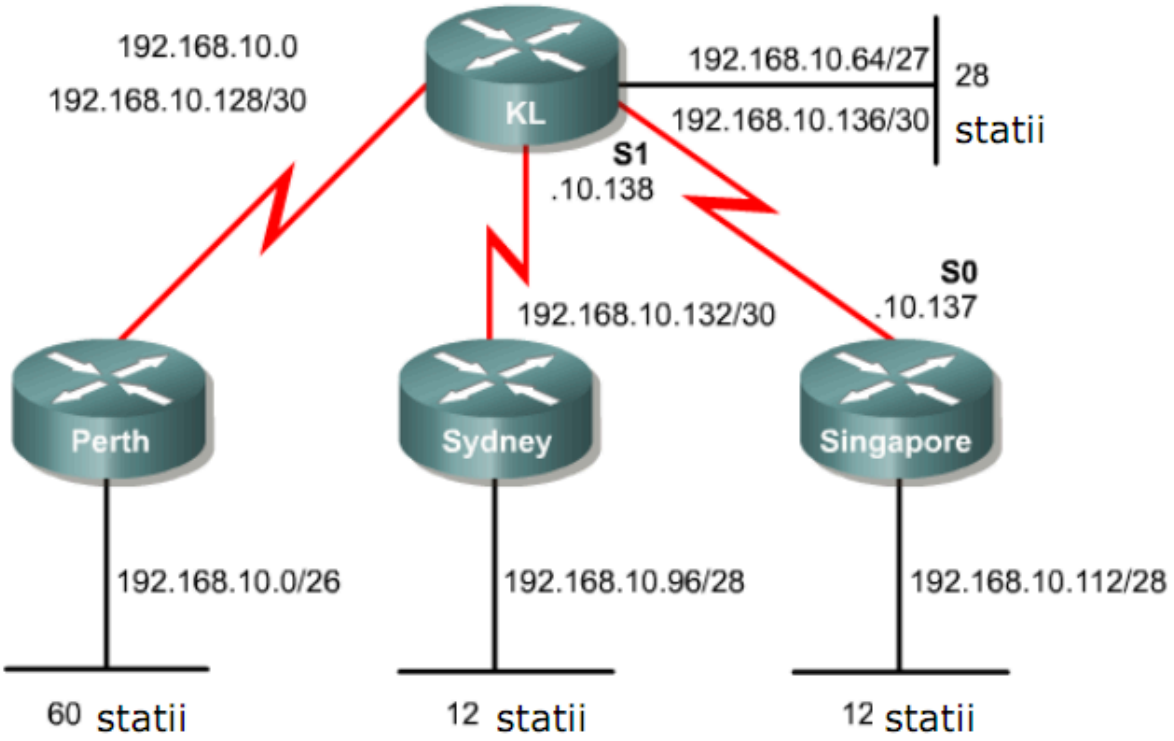
O adresă IP este un şir de 32 de biţi

• Principalul mecanism de scalare a schemei de adresare IP este separarea adresei în două câmpuri: partea de reţea şi partea de staţie

• Pentru a permite alocarea cu pierderi minime, spaţiul de adrese a fost împarţit în 5 clase de adrese: A, B, C, D, E

• Modul in care a evoluat Internet-ul a scos la iveala limitările acestei împărţiri rigide

Pentru a preveni epuizarea spaţiului de adrese IP au fost dezvoltate mai multe soluţii:
– CIDR – reducerea dimensiunii tabelei de rutare
– VLSM – flexibilizarea alocării adreselor IP
– NAT – posibilitatea folosiri de adrese private
– IPv6 – extinderea spaţiului de adrese (adrese pe 128 de biţi)

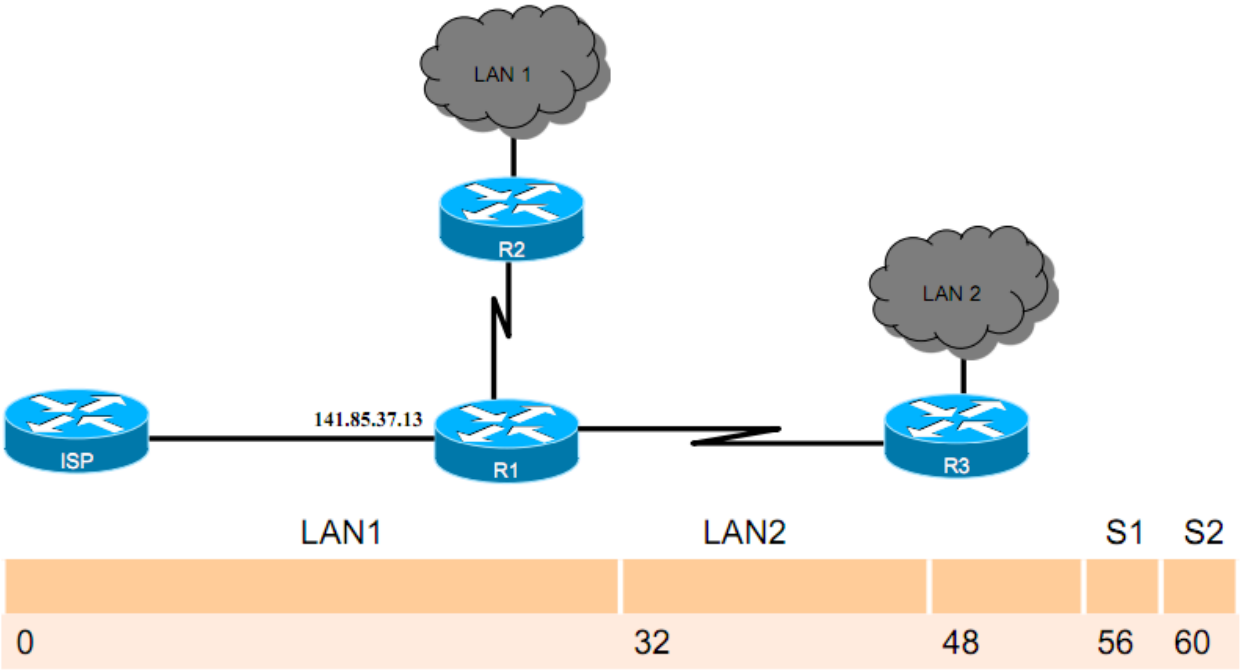


Exercitiu: Fie spaţiul de adrese 18.78.32.0/22. Care este adresa celei de a 29-a staţii din cea de a 29-a subreţea?

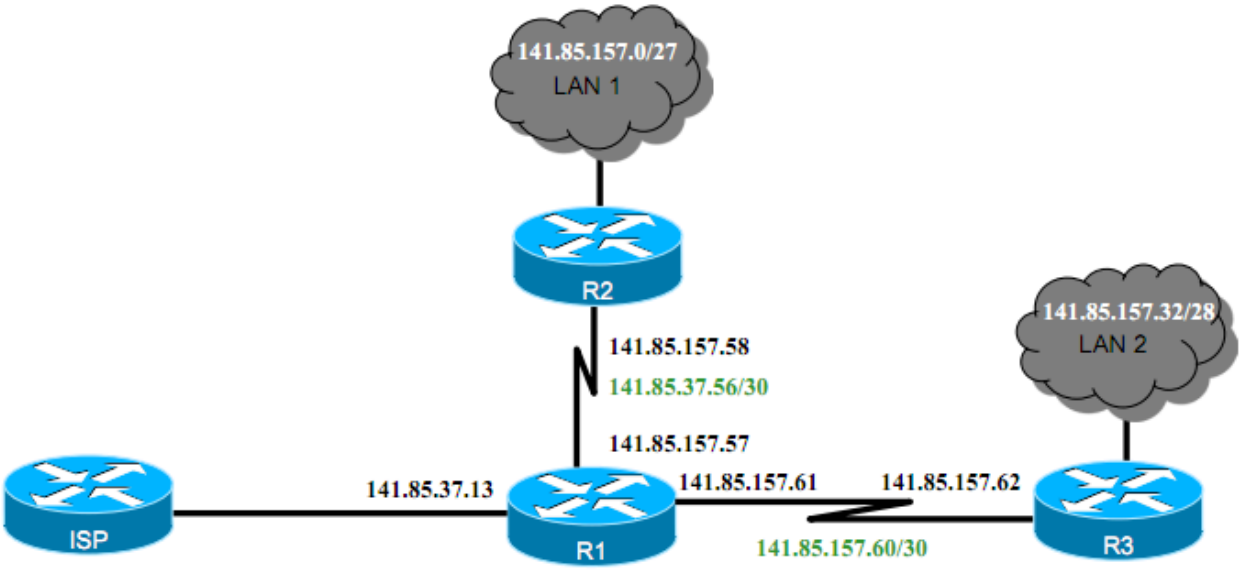
• 29 de staţii → 5 biţi de staţie
• 29 de subreţele → 5 biţi de subreţea
• masca rezultată va fi 22 + 5 = 27

18.78.001000 00.000 00000/22
18.78.001000 11.100 11101/27 → 18.78.35.157

Exercitiu:
Distribuiti adrese din spatiul 141.85.157.0/26 pentru urmatoarea figura



LAN1 141.85.157.0/27 32-2-1=29 adrese de staţie
LAN2 141.85.157.32/28 16-2-1=13 adrese de staţie
liber 141.85.157.48/29
S1 141.85.157.56/30
S2 141.85.157.60/30



Curs 5

DNS (Domain Name System) – PORT 53

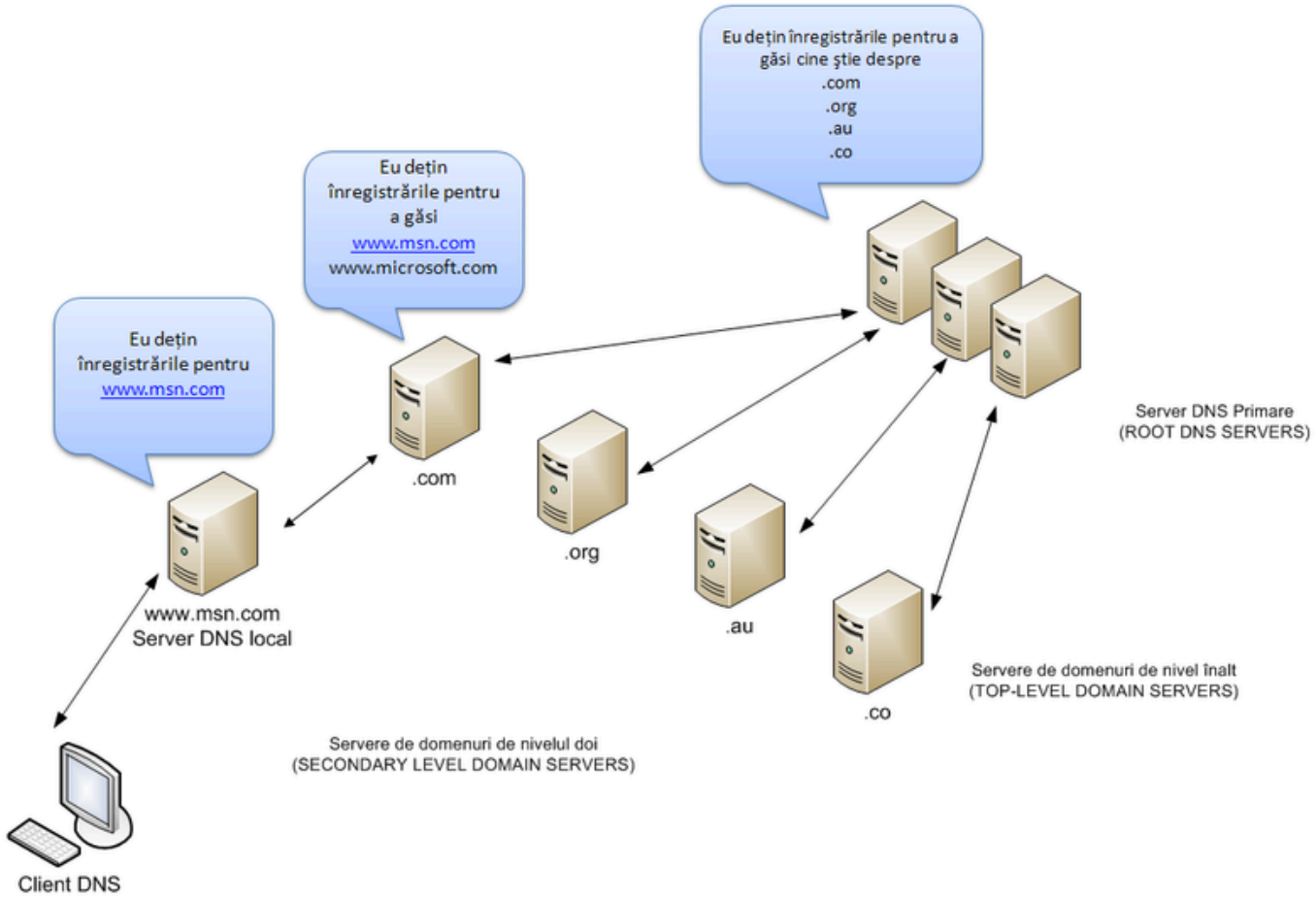
Adresele IP sunt greu de reţinut (atât IPv4 cât şi mai ales IPv6). Din această cauză s-au desemnat nume unice pentru fiecare organizaţie. Numele sunt gestionate de organizaţia

ICANN (Internet Corporation For Assigned Names And Numbers). Numele au o structură ierarhică şi aparţin unor domenii. De exemplu .com ; .edu ; .org ; .gov ; .net ; .ro ; .de etc.
Domain Name System (DNS) este un sistem distribuit de păstrare şi interogare a unor date arbitrare într-o structură ierarhică. Cea mai cunoscută aplicaţie a DNS este gestionarea domeniilor în Internet.
Caracteristicile sistemului de nume (DNS) sunt:

- foloseşte o structură ierarhizată;
- deleagă autoritatea pentru nume;
- baza de date cu numele şi adresele IP este distribuită.

Fiecare implementare TCP/IP conţine o rutină software (name resolver) specializată în interogarea serverului de nume (DNS) în vederea obţinerii translatării nume/adresă IP sau invers.
Există 2 tipuri de rezoluţie de nume:

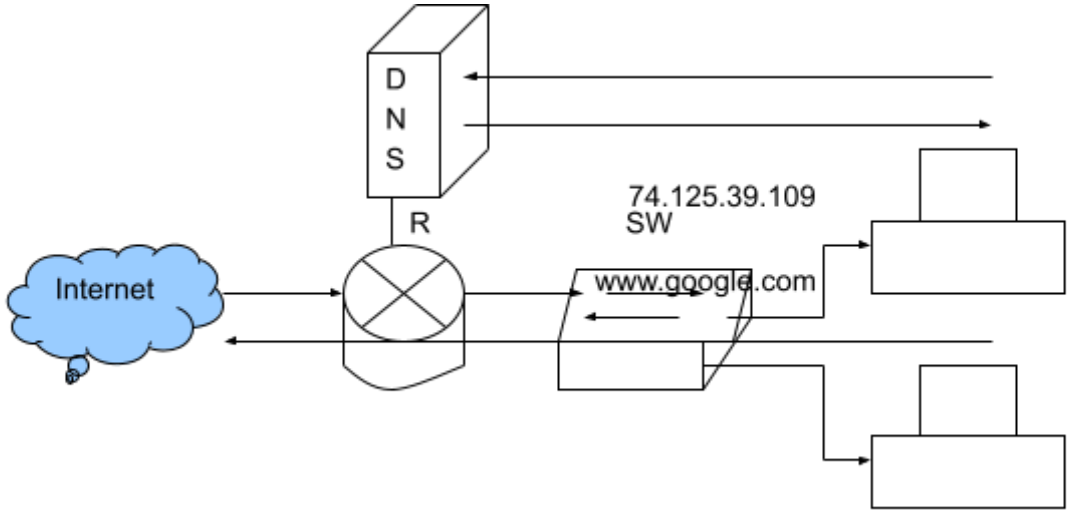
- rezoluţie recursivă (name resolverul cere serverului de nume să facă translatarea);
- rezoluţie iterativă (name resolverul cere serverului de nume să îi furnizeze adresa IP a unui server care poate face translat/area).



Tipic, procesul de rezoluţie a numelor se desfăşoară astfel:

1. Name resolverul primeşte de la o aplicaţie client TCP/IP un nume; acesta formulează o interogare primului server de nume din lista serverelor;
2. Serverul de nume (DNS) determină daca este mandatat (autorizat) pentru domeniul respectiv (dacă există configurată o zonă DNS care conţine numele respectiv);
3. Dacă este autorizat, transmite răspunsul clientului;

4. Dacă nu, transmite o interogare altui server de nume pentru un răspuns autorizat; obţine răspunsul autorizat şi transmite clientului un răspuns neautorizat; totodată stochează răspunsul local pentru a răspunde la alte cereri pentru acelaşi nume.
5. Resolverul de nume transmite răspunsul aplicaţiei utilizator şi îl păstrează într-un cache pentru o anumită perioadă;
6. Dacă name resolverul nu primeşte un răspuns într-un anumit timp, transmite cererea următorului server de nume din listă. Când lista este epuizată, va genera o eroare.



Protocolul Internet versiunea 6

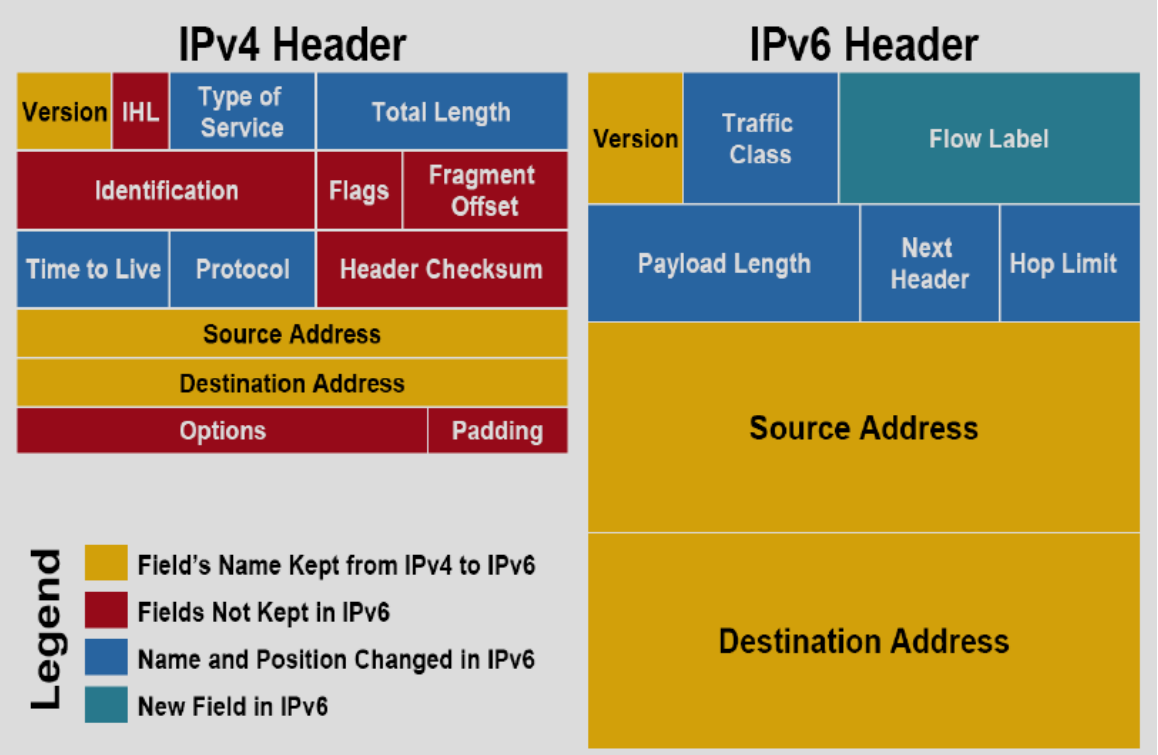
Protocolul IPv6 este o versiune a protocolului IP menita sa il inlocuiasca deoarece inca din anii ′80 devenise evident ca evolutia internetului facea insuficient numarul adreselor IP versiunea 4 care foloseau 32 de biti pentru adresare.

In luna februarie a anului 2011 au fost asignate ultimele adrese IPv4 de catre autoritatea Internet Assigned Numbers Authority (IANA). Solutiile prezentate pentru utilizarea eficienta a adreselor IPv4 nu au putut rezolva problema numerului limitat al acestora. Prin urmare a fost adoptat in anul 1994 aceasta noua versiune a protocolului internet care propunea intre altele si un spatiu de adresare pe 128 de biti.

Desi ar parea ca acest protocol este o simpla extensie a vechiului protocol, de fapt sunt incorporate in acesta o serie de elemente care sa simplifice comunicarea intre echipamente la nivelul internetului si sa asigure functii noi, care nu au fost prevazute in protocolul IPv4, lucru care face protocolul IPv6 incompatibil cu vechiul protocol.

Deoarece pentru o perioada buna de timp cele doua protocoale distincte vor coexista, ele sunt prezentate separat, in detaliu.

Astfel pentru IPv4 prezinta numeroare campuri in antentul sau optionale care sunt neutilizate (…… eexmplu ……..) sau redundante (campul pentru suma de verificare). Din acest motiv antetul IPv6 este simplificat iar campurile optionale au fost deplasate in zone a caror prezenta nu este obligatorie.



Conceptul de fragmentare nu mai exista in IPv6 deoarece echipamentele trebuie sa determine care este valoarea MTU permisa fie vor trimite pachete de dimensiune mai mica decat MTU de 1280 OCTETI.

IPv4 continea in antet un camp cu rol de detectie a erorilor, insa verificari similare se realizau atat la nivelul legatura de date (CRC-32) cat si la nivlul transport (suma pe 16 biti). Antetul IPv6 nu mai contine acest camp de verificare lasand pe seama nivelului transport aceasta operatie. In acest mod se usureaza sarcina ruterelor care vor avea de efectuat un numar redus de calcule.

Campul TTL a carui denumire nu corespundea cu functionalitatea sa a fost redenumit corespunzator: Hop Limit (Numar de salturi).

Spre deosebire de IPv4, IPv6 suporta echipamentele mobile …

IPv6 are suport pentru viitoare functii precum QoS, etc. ….

Pentru inceput s-a avut in vedere faptul ca adresele IPv4 au fost ineficient alocate, o mare parte a adreselor ramanand neutilizate si in prezent.

Principul multicast-ului a fost integrat in specificatia de baza a IPv6, in acest mod un echipament putand transmite in retea dintr-o singura comanda un pachet la mai multe calculatoare.

ICMPv6 este folosit, spre deosebire de protocolul cu nume similar disponibil pentru IPv4, pentru configurarea automata a echipamentelor IPv6. In IPv6, ICMP packets are also used in the neighbor discovery protocol and path MTU discovery.

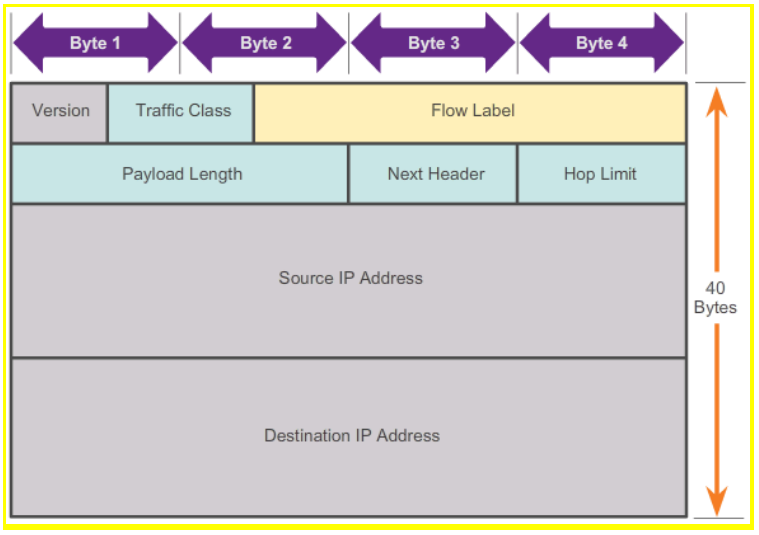
Securitatea este o prioritate a noului protocol, fiind integrata in acesta in diferite campuri ale antetului. Folosirea protocolului IPsec este obligatorie pentru IPv6 in vreme ce pentru IPv4 aceasta era optionala.

Pachetul IPv6

Integrarea in sistemele de operare

Metode de tunelare a traficului IPv

Îmbunătățirile oferite de IPv6 includ:
• Spațiu de adrese crescut - adresele IPv6 se bazează pe adresarea ierarhică de 128 de biți, spre deosebire de IPv4 cu 32 de biți. Acest lucru crește dramatic numărul de adrese IP disponibile.
• Gestionare îmbunătățită a pachetelor - Antetul IPv6 a fost simplificat cu mai puține câmpuri. Acest lucru îmbunătățește gestionarea pachetelor de către routerele intermediare și oferă, de asemenea, suport pentru extensii și opțiuni pentru o scalabilitate/longevitate sporită.
• Elimină necesitatea NAT - Cu un număr atât de mare de adrese IPv6 publice, Network Address Translation (NAT) nu este necesară. Site-urile clienților, de la cele mai mari întreprinderi până la gospodării unice, pot obține o adresă publică de rețea IPv6. Acest lucru evită unele dintre problemele aplicațiilor induse de NAT cu care se confruntă aplicațiile care necesită conectivitate end-to-end.
• Securitate integrată - IPv6 acceptă nativ capabilități de autentificare și confidențialitate. Cu IPv4, au trebuit implementate caracteristici suplimentare pentru a face acest lucru.
Antetul simplificat IPv6 oferă mai multe avantaje față de IPv4:
• Eficiență de rutare mai bună pentru performanță și scalabilitate a ratei de redirecționare
• Nicio cerință pentru procesarea sumelor de control
• Mecanisme de antet de extensie simplificate și mai eficiente (spre deosebire de câmpul Opțiuni IPv4)
• Un câmp Flow Label pentru procesarea per flux, fără a fi nevoie să deschideți pachetul interior de transport pentru a identifica diferitele fluxuri de trafic



Câmpurile din antetul pachetului IPv6 includ:
• Versiune - Acest câmp conține o valoare binară de 4 biți care identifică versiunea pachetului IP. Pentru pachetele IPv6, acest câmp este întotdeauna setat la 0110.
• Clasa de trafic - Acest câmp pe 8 biți este echivalent cu câmpul Servicii diferențiate IPv4 (DS). Conține, de asemenea, o valoare DSCP (Differentiated Services Code Point) pe 6 biți, utilizată pentru a clasifica pachetele și o notificare explicită de congestie (ECN) pe 2 biți, utilizată pentru controlul congestionării traficului.
• Flow Label - Acest câmp de 20 de biți oferă un serviciu special pentru aplicații în timp real. Poate fi folosit pentru a informa ruterele și comutatoarele să mențină aceeași cale pentru fluxul de pachete, astfel încât pachetele să nu fie reordonate.
• Lungimea încărcăturii - Acest câmp de 16 biți este echivalent cu câmpul Lungimea totală din antetul IPv4. Acesta definește dimensiunea întregului pachet (fragment), inclusiv antetul și extensiile opționale.

• Antet următor - Acest câmp de 8 biți este echivalent cu câmpul Protocol IPv4. Indică tipul de încărcare utilă de date pe care îl transportă pachetul, permițând stratului de rețea să transmită datele la protocolul corespunzător de nivel superior. Acest câmp este utilizat și dacă există antete de extensie opționale adăugate la pachetul IPv6.
• Hop Limit: - Acest câmp de 8 biți înlocuiește câmpul IPv4 TTL. Această valoare este redusă cu una de către fiecare router care transmite pachetul. Când contorul ajunge la 0, pachetul este aruncat și un mesaj ICMPv6 este redirecționat către gazda care trimite, indicând faptul că pachetul nu a ajuns la destinație.
• Adresă sursă - Acest câmp pe 128 de biți identifică adresa IPv6 a gazdei care primește.
• Destination Address - Acest câmp pe 128 de biți identifică adresa IPv6 a gazdei care primește.

Cum se directioneaza pachetele in retea (rutare)

Un alt rol al stratului de rețea este de a direcționa pachetele între gazde, proces numit rutare. O gazdă poate trimite un pachet către:
• Sine - Aceasta este o adresă IP specială de 127.0.0.1, care este denumită interfață loopback. Această adresă de loopback este atribuită automat unei gazde atunci când rulează TCP/IP. Capacitatea unei gazde de a-și trimite un pachet utilizând funcționalitatea de rețea este utilă în scopuri de testare. Orice IP din cadrul rețelei 127.0.0.0/8 se referă la gazda locală.
• Gazdă locală - Aceasta este o gazdă în aceeași rețea cu gazda de trimitere. Gazdele partajează aceeași adresă de rețea.
• Gazdă la distanță - Aceasta este o gazdă într-o rețea la distanță. Gazdele nu partajează aceeași adresă de rețea.
Tabelul de rutare local (route print sau netstat -r) al gazdei conține de obicei:
• Conexiune directă - Aceasta este o rută către interfața loopback (127.0.0.1).
• Rută de rețea locală - Rețeaua la care este conectată gazda este populată automat în tabelul de rutare a gazdei.
• Rută implicită locală - Ruta implicită reprezintă ruta pe care trebuie să o parcurgă pachetele pentru a ajunge la toate adresele de rețea la distanță. Ruta implicită este creată atunci când pe gazdă este prezentă o adresă de gateway implicită. Adresa implicită a gateway-ului este adresa IP a interfeței de rețea a routerului care este conectat la rețeaua locală. Adresa de gateway implicită poate fi configurată pe gazdă manual sau învățată dinamic.

Tabela de rutare Ipv6 afișează patru coloane care identifică:
• Dacă - Listează numerele de interfață din secțiunea Lista de interfețe a comenzii netstat –r. Numerele interfeței corespund interfeței capabile de rețea de pe gazdă, inclusiv adaptoarele Ethernet, Wi-Fi și Bluetooth.
• Metric - Listează costul fiecărei rute către o destinație. Numerele mai mici indică rutele preferate.
• Network Destination - Listează rețelele accesibile.
• Gateway - Listează adresa utilizată de gazda locală pentru a redirecționa pachetele către o destinație de rețea la distanță. On-link indică faptul că gazda este conectată în prezent la ea.

```
IPv6 Route Table
===========================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
 18    281 fe80::/64                On-link
  9    306 fe80::/64                On-link
  9    306 fe80::8f2a:b733:42d8:d934/128
                                    On-link
 18    281 fe80::f453:6f89:bf2a:5faa/128
                                    On-link
  1    331 ff00::/8                 On-link
 18    281 ff00::/8                 On-link
  9    306 ff00::/8                 On-link
===========================================================
Persistent Routes:
  None
```

In tabel (obtinut prin comada **netstat –r)** putem intalni urmatoarele valori:

- **::/0 -** echivalentul IPv6 echivalentul rutei locale implicite (0.0.0.0 in IPv4).
- **::1/128 -** echivalentul adresei IPv4 loopback (127.0.0.0/8) pentru servicii locale. Functioneaza chiar daca echipamentul nu este conectat la retea
- **2001::/32** – prefix folosit in comunicatii globale unicast.
- **fe80::/64 -** This is the **local link** network route address and represents all computers on the local link IPv6 network.
- **fe80::8f2a:b733:42d8:d934/128** – adresa link local IPv6 a calculatorului. Interfețele din IPv6 au de obicei două adrese IPv6: o adresă locală (link local) și o adresă unicast globală. De asemenea, nu există adrese de broadcast în IPv6!
- **ff00::/8** – Rezervate pentru **multicast** (clasa D a IPv4 **224.x.x.x**).

Note: Interfaces in IPv6 commonly have two IPv6 addresses: a link local address and a global unicast address. Also, notice that there are no broadcast addresses in IPv6. IPv6 addresses will be discussed further in the next chapter.

IPv4 issues

There is not a single date to move to IPv6. The IETF has created various protocols and tools to help network administrators migrate their networks to IPv6. The migration techniques can be divided into three categories:

- **Dual Stack** – As shown in Figure 1, dual stack allows IPv4 and IPv6 to coexist on the same network. Dual stack devices run both IPv4 and IPv6 protocol stacks simultaneously.
- **Tunneling** – As shown in Figure 2, tunneling is a method of transporting an ***IPv6 packet over an IPv4 network***. The IPv6 packet is encapsulated inside an IPv4 packet, similar to other types of data.
- **Translation** – As shown in Figure 3, Network Address Translation 64 (NAT64) allows IPv6-enabled devices to communicate with IPv4-enabled devices using a translation technique similar to NAT for IPv4. An IPv6 packet is translated to an IPv4 packet, and vice versa.

IPv6 Addressing

IPv6 addresses are 128 bits in length and written as a string of hexadecimal values. Every 4 bits is represented by a single hexadecimal digit; for a total of 32 hexadecimal values.

How to reduce the notation of IPv6 addresses

The first rule to help reduce the notation of IPv6 addresses is any leading 0s (zeros) in any 16-bit section or hextet can be omitted. For example:

- 01AB can be represented as 1AB
- 09F0 can be represented as 9F0
- 0A00 can be represented as A00
- 00AB can be represented as AB

This rule only applies to leading 0s, NOT to trailing 0s, otherwise the address would be ambiguous. For example, the hextet “ABC” could be either “0ABC” or “ABC0”.

The second rule to help reduce the notation of IPv6 addresses is that a double colon (::) can replace any single, contiguous string of one or more 16-bit segments (hextets) consisting of all 0s.

The double colon (::) can only be used once within an address, otherwise there would be more than one possible resulting address. When used with the omitting leading 0s technique, the notation of IPv6 address can often be greatly reduced. This is commonly known as the compressed format.
2001:DB8:0:1111::200

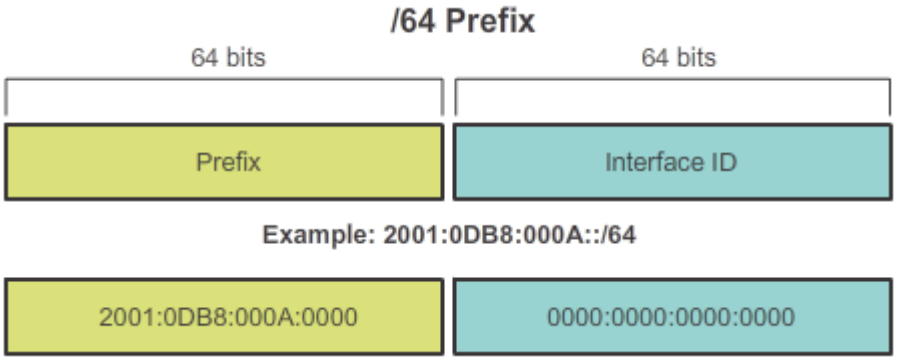
1
There are three types of IPv6 addresses:

- **Unicast** - An IPv6 unicast address uniquely identifies an interface on an IPv6-enabled device. As shown in the figure, a source IPv6 address must be a unicast address.
- **Multicast** - An IPv6 multicast address is used to send a single IPv6 packet to multiple destinations.

- **Anycast** - An IPv6 anycast address is any IPv6 unicast address that can be assigned to multiple devices. A packet sent to an anycast address is routed to the nearest device having that address. Anycast addresses are beyond the scope of this course.

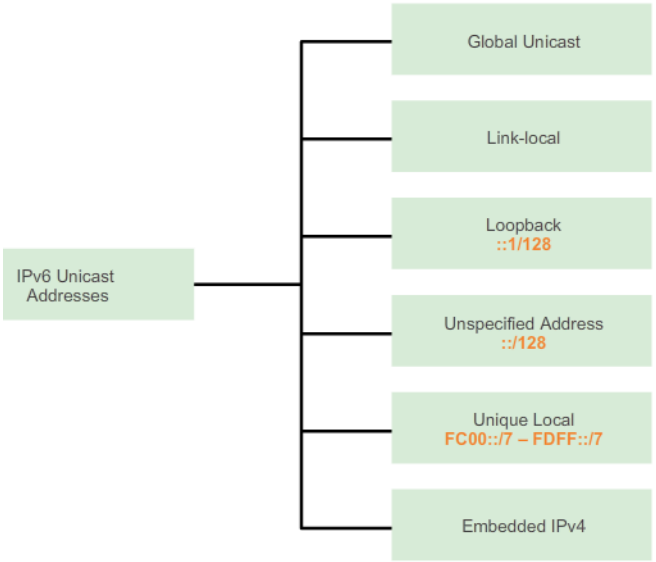
Obs. IANA is to record the allocation of the IPv6 global unicast address prefix 2001:DB8::/32 as a documentation-only prefix in the IPv6 address registry. No end party is to be assigned this address. Can be safely used in laboratory!

Unlike IPv4, IPv6 does not have a broadcast address. However, there is an IPv6 all-nodes multicast address that essentially gives the same result. **Multicast addresses in IPv6 have the prefix ff00::/8.**



IPv6 uses the prefix length to represent the prefix portion of the address. IPv6 does not use the dotted-decimal subnet mask notation. ***The prefix length is used to indicate the network portion of an IPv6 address using the IPv6 address/prefix length.***

The prefix length can range from 0 to 128. A typical IPv6 prefix length for LANs and most other types of networks is /64. This means the prefix or network portion of the address is 64 bits in length, leaving another 64 bits for the interface ID (host portion) of the address.



An IPv6 unicast address uniquely identifies an interface on an IPv6-enabled device. A packet sent to a unicast address is received by the interface that is assigned that address. Similar to IPv4, a source IPv6 address must be a unicast address. The destination IPv6 address can be either a unicast or a multicast address.

There are six types of IPv6 unicast addresses.

Global unicast

A global unicast address is similar to a public IPv4 address. These are globally unique, Internet routable addresses. Global unicast addresses can be configured statically or assigned dynamically.

There are some important differences in how a device receives its IPv6 address dynamically compared to DHCP for IPv4. Use 2001:DB8::/32 for execises!

Link-local

Link-local addresses are used to communicate with other devices on the same local link. With IPv6, the term link refers to a subnet. Link-local addresses are confined to a single link. Their uniqueness must only be confirmed on that link because they are not routable beyond the link. In other words, routers will not forward packets with a link-local source or destination address. IPv6 link-local addresses are in the FE80::/10 range

Loopback

The loopback address is used by a host to send a packet to itself and cannot be assigned to a physical interface. Similar to an IPv4 loopback address, you can ping an IPv6 loopback address to test the configuration of TCP/IP on the local host. The IPv6 loopback address is all-0s except for the last bit, represented as ::1/128 or just ::1 in the compressed format.

Unspecified address

An unspecified address is an all-0s address represented in the compressed format as ::/128 or just :: in the compressed format. It cannot be assigned to an interface and is only be used as a source address in an IPv6 packet. An unspecified address is used as a source address when the device does not yet have a permanent IPv6 address or when the source of the packet is irrelevant to the destination.

Unique local

IPv6 unique local addresses have some similarity to RFC 1918 private addresses for IPv4, but there are significant differences as well. Unique local addresses are used for local addressing within a site or between a limited number of sites. These addresses should not be routable in the global IPv6. Unique local addresses are in the range of FC00::/7 to FDFF::/7.

With IPv4, private addresses are combined with NAT/PAT to provide a many-to-one translation of private-to-public addresses. This is done because of the limited availability of IPv4 address space. Many sites also use the private nature of RFC 1918 addresses to help secure or hide their network from potential security risks. However, this was never the intended use of these technologies and the IETF has always recommended that sites take the proper security precautions on their Internet facing router. Although, IPv6 does provide for site specific addressing, it is not intended to be used to help hide internal IPv6-enabled devices from the IPv6 Internet. IETF recommends that limiting access to devices should be accomplished using proper, best-practice security measures.

Note: The original IPv6 specification defined site-local addresses for a similar purpose, using the prefix range FEC0::/10. There were several ambiguities in the specification and site-local addresses were deprecated by the IETF in favor of unique local addresses.

IPv4 embedded

The last type of unicast address type is the IPv4 embedded address. These addresses are used to help transition from IPv4 to IPv6. IPv4 embedded addresses are beyond the scope of this course.

IPv6 Link-Local Communications

IPv6 Packet

Source IPv6 Address	Destination IPv6 Address
FE80::AAAA	FE80::DDDD

An IPv6 link-local address enables a device to communicate with other IPv6-enabled devices on the same link and only on that link (subnet). Packets with a source or destination link-local address cannot be routed beyond the link from where the packet originated.

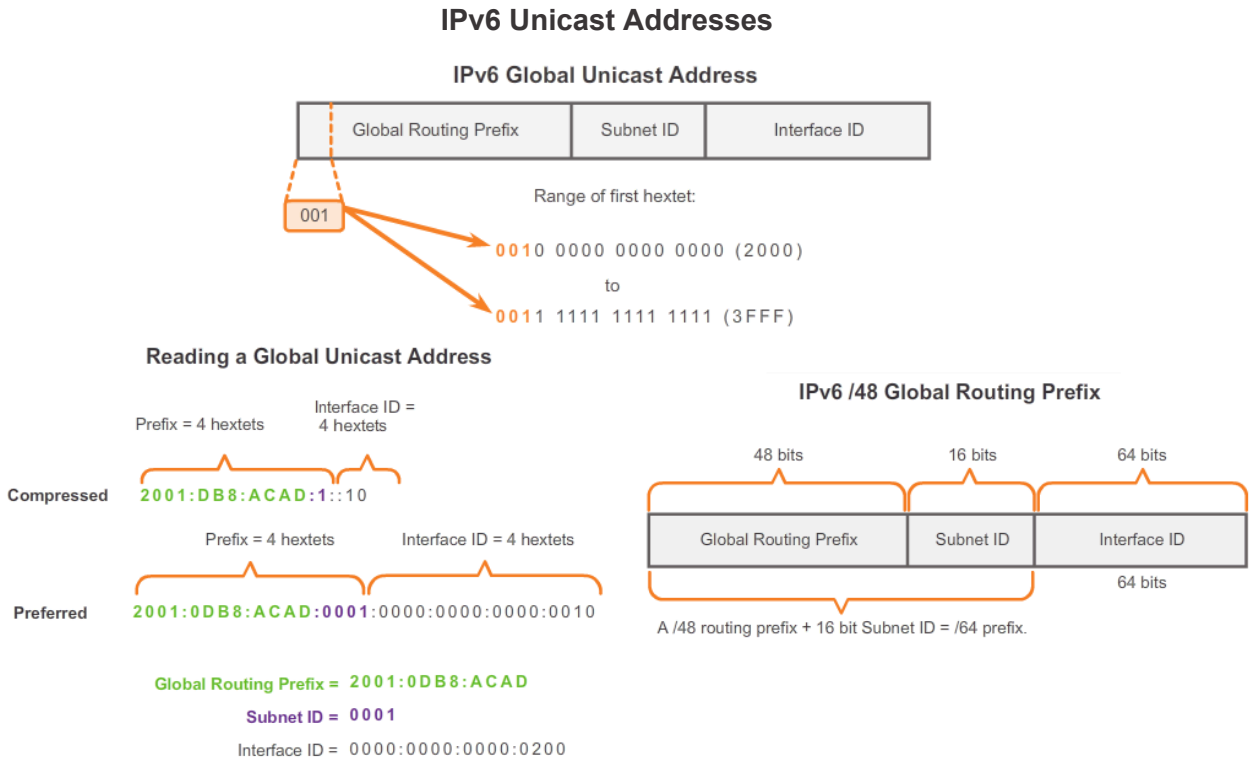
Unlike IPv4 link-local addresses, IPv6 link-local addresses have a significant role in various aspects of the network. The global unicast address is not a requirement; however, every IPv6-enabled network interface is required to have a link-local address.

If a link-local address is not configured manually on an interface, the device will automatically create its own without communicating with a DHCP server. IPv6-enabled hosts create an IPv6 link-local address even if the device has not been assigned a global unicast IPv6 address. This allows IPv6-enabled devices to communicate with other IPv6-enabled devices on the same subnet. This includes communication with the default gateway (router).

IPv6 link-local addresses are in the FE80::/10 range. The /10 indicates that the first 10 bits are 1111 1110 10xx xxxx. The first hextet has a range of 1111 1110 10**00 0000** (FE80) to 1111 1110 10**11 1111** (FEBF).

IPv6 link-local addresses are also used by IPv6 routing protocols to exchange messages and as the next-hop address in the IPv6 routing table. Link-local addresses are discussed in more detail in a later course.

Note: Typically, it is the link-local address of the router and not the global unicast address that is used as the default gateway for other devices on the link.



IPv6 global unicast addresses are globally unique and routable on the IPv6 Internet. These addresses are equivalent to public IPv4 addresses. The Internet Committee for Assigned Names and Numbers (ICANN), the operator for Internet Assigned Numbers Authority (IANA), allocates IPv6 address blocks to the five RIRs. Currently, only global unicast addresses with the first three bits of 001 or 2000::/3 are being assigned. This is only 1/8th of the total available IPv6 address space, excluding only a very small portion for other types of unicast and multicast addresses.
Note: The 2001:0DB8::/32 address has been reserved for documentation purposes, including use in examples.

A global unicast address has three parts:

- Global routing prefix
- Subnet ID
- Interface ID

Global Routing Prefix

The global routing prefix is the prefix, or network, portion of the address that is assigned by the provider, -such as an ISP, to a customer or site. Currently, RIRs assign a /48 global routing prefix to customers. This includes everyone from enterprise business networks to individual households. This is more than enough address space for most customers.

Figure 2 shows the structure of a global unicast address using a /48 global routing prefix. /48 prefixes are the most common global routing prefixes assigned and will be used in most of the examples throughout this course.

For example, the IPv6 address 2001:0DB8:ACAD::/48 has a prefix that indicates that the first 48 bits (3 hextets) (2001:0DB8:ACAD) is the prefix or network portion of the address. The double colon (::) prior to the /48 prefix length means the rest of the address contains all 0s.

Subnet ID

The Subnet ID is used by an organization to identify subnets within its site.

Interface ID

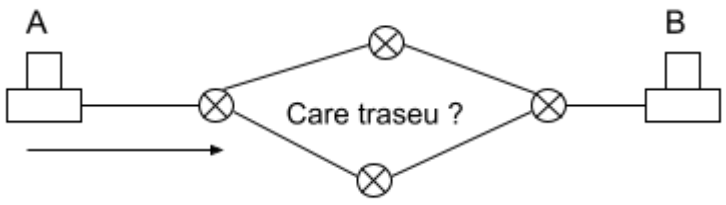
The IPv6 Interface ID is equivalent to the host portion of an IPv4 address. The term Interface ID is used because a single host may have multiple interfaces, each having one or more IPv6 addresses.

Note: Unlike IPv4, in IPv6, the all-0s address can be assigned to a device because there are no broadcast addresses in IPv6. However, the all-0s address is reserved as a Subnet-Router anycast address, and should be assigned only to routers.

An easy way to read most IPv6 addresses is to count the number of hextets. As shown in Figure 3, in a /64 global unicast address the first four hextets are for the network portion of the address, with the fourth hextet indicating the Subnet ID. The remaining four hextets are for the Interface ID.

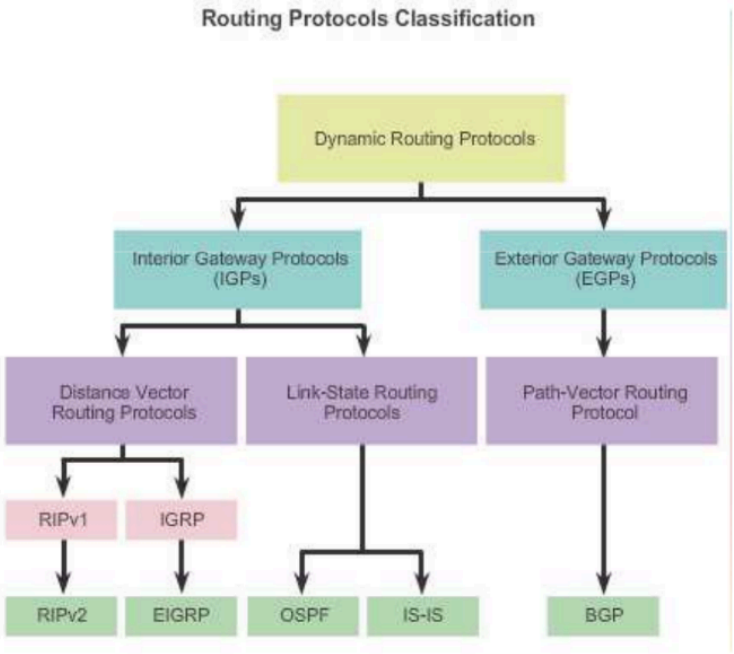
Rutarea în reţele de calculatoare

Pentru a redirectiona traficul in retea routerele au nevoie de o vedere unica a traseelor in retea . Aparitia sau disparitia traseelor trebuie sa fie semnalizata iar acest lucru este realizat de protocoalele de routare .Informatia de routare poate fi transmisa catre toate celelalte routere (FLOATING) putand fi trimisa pe cai selective folosind grafuri (arbori) fara a transmite datele de mai multe ori (redundante).



Types of Routing Protocols

Classifying Routing Protocols



Distance Vector or Link-State Routing Protocols

Putem propune 3 metode:

1. Tehnica Flooding (inundare cu pachete)

2. Tehnica vector distanţă - algoritmul Bellman – Ford

3. Algoritmul bazat pe transmiterea stării legăturilor (link state) – algoritmul lui Dijkstra

Termenul folosit pentru a măsura calea către destinaţie este **metrică**:

A metric is a measurable value that is assigned by the routing protocol to different routes based on the usefulness of that route:

Used to determine the overall "cost" of a path from source to destination.

Routing protocols determine the best path based on the route with the lowest cost.

- **Întârzierea**. La transmiterea unui pachet se preferă legăturile rapide şi apropiate
- **Lăţimea de bandă**
- **Utilizarea liniei** (load balancing)
- **Stabilitatea liniei** (dacă este pornită sau oprită)

The RIP and OSPF are two interior gateway protocols (IGP) that intensively used in computer networks to specify the best routes for data transmission. RIP (Routing Information Protocol) is one of the oldest routing protocols in service, whereas OSPF (Open Shortest Patch First) serves as the most widely adopted IGP for large enterprise networks. Network admins may find themselves in a dilemma when choosing between RIP vs OSPF. So, we will present a detailed description of these two routing protocols and address key RIP vs OSPF differences.

RIP vs OSPF: What Is RIP Protocol?

Routing Information Protocol (RIP), is an example of distance vector routing for local networks. RIP works to deliver the whole routing table to all active interfaces in every 30 seconds. In RIP protocol, hop count is the only metrics to decide the best path to a remote network. Let's take an example to see how RIP protocol

works: Assuming, we have two paths available from the source (R1) to the destination (R7). It is clear that Path 2 will be selected by RIP protocol since it has less hop counts

Path 1: R1-R2-R4-R6-R7

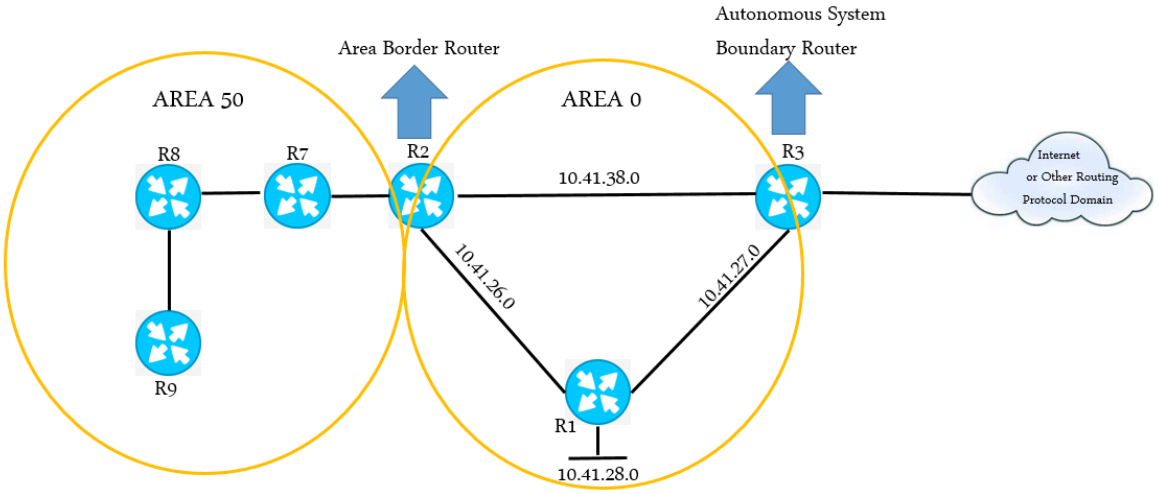
Path 2: R1-R3-R5-R7

Pros and Cons of RIP Protocol

RIP is a great fit for small networks - It's easy to understand and configure while also being supported by almost all routers. The hop counts of RIP is limited to 15 hops, so any router beyond that distance is considered as infinity, and hence unreachable. When implementing in a large network, RIP can create a traffic bottleneck by multicasting all the routing tables every 30 seconds, and it has very slow network convergence. Since any routing update in RIP will take up great bandwidth, the resources for critical IT processes are hence limited. Moreover, RIP doesn't support multiple paths on the same route, which may generate more routing loops. While using fixed hop count metric to select the best routes, RIP fails to work when routes are compared based on real-time data. This causes a packet loss and overloads network operations due to repeated processes.

RIP vs OSPF: What Is OSPF in Networking?

OSPF (Open Shortest Path First), a link state routing protocol, is massively adopted in large enterprise networks. OSPF routing protocol collects link state information from routers in the network and determines the routing table information to forward packets. This occurs by creating a topology map for the network. Unlike RIP, OSPF only exchanges routing information when there's a change in network topology. OSPF best fits for complex networks that comprise multiple subnet working to ease network administration and optimize traffic. It effectively calculates the shortest path with minimum network traffic when the change occurs. Foloseste algoritm Djikstra.

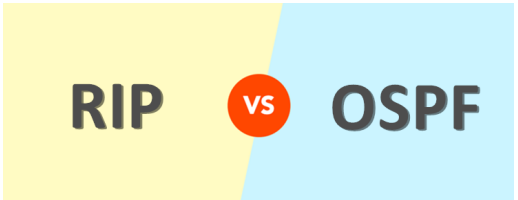


Pros and Cons of OSPF Protocol

Using OSPF protocol demands advanced knowledge about complex networks. So OSPF routing protocol allows routers to calculate routes based on incoming requests. Unlike RIP protocol that has only 15 hops at most, OSPF has no limitations in hop count. So OSPF converges faster than RIP, and has better load balancing. The drawbacks of OSPF, however, is that it doesn't scale when there are more routers added to the network. And this lack of scalability in OSPF makes it unsuitable for routing across the Internet.

RIP vs OSPF: What Is the Difference?

The RIP and OSPF are the IGP that routing information within an autonomous system, and RIP vs OSPF differs in many aspects.



Routing Protocol Type: The RIP is a distance vector protocol whereas the OSPF is a link state protocol. A distance vector protocol uses the distance or hop counts to determine the transmission path. The link state protocol analyzes different sources like the speed, cost and path congestion while identifying the shortest path.

Network table construction: The RIP requests the routing table from the devices around the router that uses RIP. Then the router consolidated that information and constructs its own routing table. This table is sent to those neighboring devices at a regular interval and the consolidated routing table of the router is updated. In OSPF, the router consolidates routing table by getting only required information from the neighboring devices. It never gets the entire routing table of the devices and the routing table construction is really simpler.

Hop Count Restriction: The RIP allows only up to 15 hops, whereas in OSPF protocol, there is no such restriction.

Algorithm used: The RIP routing protocol uses the distance vector algorithm whereas the OSPF uses the shortest path algorithm Dijkstra to determine the transmission routes.

Network classification: In RIP, the networks are classified as areas and tables. In OSPF, the networks are classified as areas, sub areas, autonomous systems and backbone areas.

Complexity level: The RIP is relatively simpler whereas the OSPF is much more complex.

RIP vs OSPF Application: The RIP suits better for smaller networks as it has hop count restrictions. The OSPF serves great for larger networks.

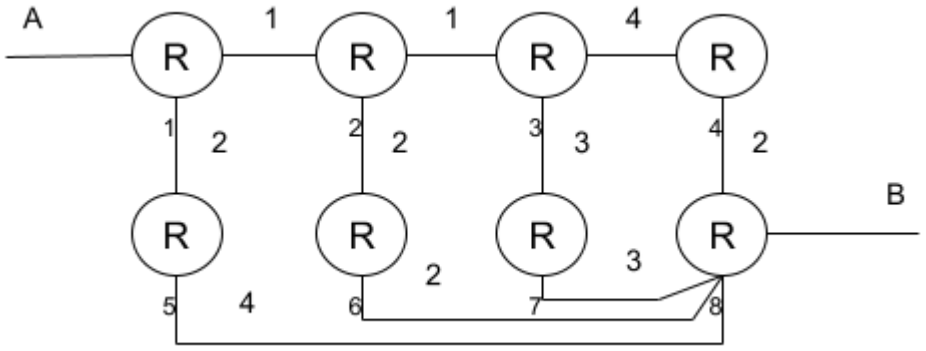
Other RIP vs OSPF differences are presented in the chart below.

Attribute	RIP	OSPF
Convergence	Slow	Fast
Network Size	For small to medium networks	For large and small networks
Need of Device Resources	Much less memory and CPU intensive than OSPF	Memory and CPU intensive
Need of Network Resources	Bandwidth consuming; whole routing table is sent	Less than RIP; only small updates are sent
Metric	Based on hop count	Based on bandwidth
Design	Flat network	Hierarchical network possible

Conclusion

After comparing RIP vs OSPF differences, it’s clear that RIP is ideal for small networks that are simple and non-hierarchical, whereas OSPF fits best for large and hierarchical enterprise networks. In a complex network, you may have multiple routing protocols operating simultaneously.

Exercitiu:
Determinaţi traseul optim de rutare a informaţiei intre calculatoarele A şi B.



Costurile posibile includ: distanţă, viteza, preţul, întârziere, încărcarea liniei.

Un tip de protocol de routare numit vector distanta ia decizii bazate pe legaturile adiacente invecinate si propune solutia . R1R2R3R7R8.

Un algoritm numit protocol de routare cu transmiterea legaturilor analizeaza toate legaturile din retea si propune solutia optima : R1R2R6R8, insa este mai mult de calculat.

1. Tehnica flooding de rutare implică transmiterea pachetelor de către router către toate porturile mai puţin cel de intrare.

Avantaje: - este simplă şi orice destinaţie poate fi atinsă

Dezavantaje: - pachetele primite de mai multe ori

- pachete în buclă înfinită

Tehnicile moderne de transmitere în reţea se bazează pe arbori spanning tree (arbori de conexiune care conţin toate vârfurile grafului însă o singură cale către acestea). În arbori nu există bucle.

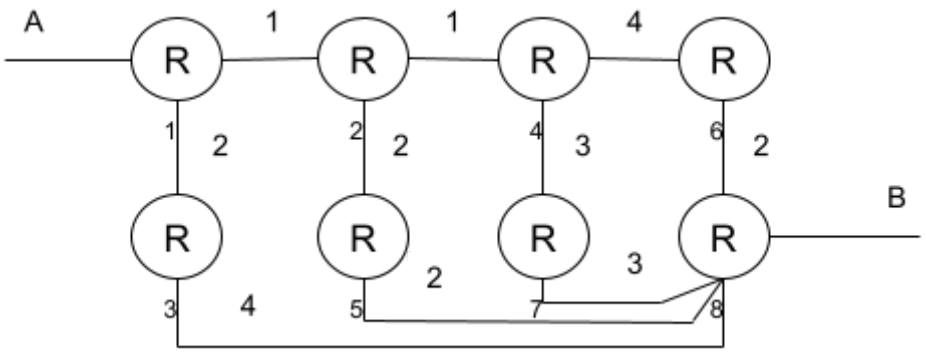
2. Algoritmul vector distanţă (RIP).

Distance vector protocols use routers as sign posts along the path to the final destination.

Algoritmul presupune iniţial toate distanţele infinite la fiecare T secunde routerul trimite către vecinii săi informaţia despre celelalte rute disponibile (acesta este vectorul distanţă).

Dacă routerul primeşte o informaţie cu un cost mai mic decât cel cunoscut informaţia se actu-alizează. Dacă nu, rămâne nemodificată.

Presupunem reţeaua de mai jos:



Paşii pentru construirea tabelei de rutare pentru router R_8

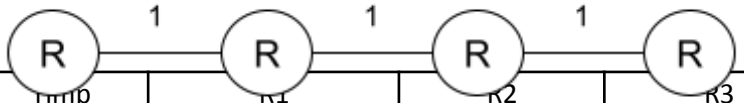
Pasul 1	Pasul 2	Pasul 3	Pasul 4
R_1 ∞	R_1 ∞	R_1 6, R_3	R_1 5, R_2
R_2 ∞	R_2 ∞	R_2 4, R_5	R_2 4, R_5
R_3 ∞	R_3 4, R_3	R_3 4, R_4	R_3 4, R_4
R_4 ∞	R_4 ∞	R_4 6, R_7	R_4 5, R_2
R_5 ∞	R_5 2, R_5	R_5 2, R_5	R_5 2, R_5
R_6 ∞	R_6 2, R_6	R_6 2, R_6	R_6 2, R_6
R_7 ∞	R_7 3, R_7	R_7 3, R_7	R_7 3, R_7

Convergenţa unei reţele

O reţea este convergenta dacă toate routerele cunosc corect toate traseele din reţea.

Algoritmul vector – distanţă converge greu. Din această cauză pot să apară probleme dacă unul dintre trasee îşi modifică costul sau dacă este deconectat.

Exemplu.



Timp 1	R1 2	R2 3	R3 4	
0	3,R2	2,R3	1,R4	
1	5,R2	4,R3	3,R2	R1,R4 este intrerupt
2	7,R2	6,R3	5,R2	
3	…	…	…	…Catre infinit

Această problemă poartă denumirea de numărare la infinit.

Solutii

a. Limitarea numărului de salturi (de obicei 15 salturi).

b. Split horizon. Deoarece un router primeşte de la un vecin un cost mai mic nu îi mai răspunde dacă ştie un cost mai mare.

c. Split horizon cu anunţuri inverse. Se vor transmite actualizări cu distanţe infinite.

3. Actualizări bazate pe starea legăturilor (OSPF)

A link-state routing protocol is like having a complete map of the network topology. The sign posts along the way from source to destination are not necessary, because all link-state routers are using an identical map of the network. A link-state router uses the link-state information to create a topology map and to select the best path to all destination networks in the topology.

Algoritmul implică transmiterea de mesaje de actualizare doar la modificarea unei legături. Fiecare router cunoaşte starea tuturor legăturilor din reţea. Traficul este masiv iniţial (tehnica flooding) dar ulterior nu mai există trafic decât la intreruperea unei coneziuni. Vectorii distanţă necesită mult mai puţină tehnică de calcul. Necesar de memorie mai mare pentru algoritmul cu starea legăturilor.

Nu există probleme cu buclele infinite.

Securitate mai bună pentru vectorii distanţă deoarece ascunde topologia reţelei.Sistemele bazate pe starare legaturilor nu faca acest lucru.

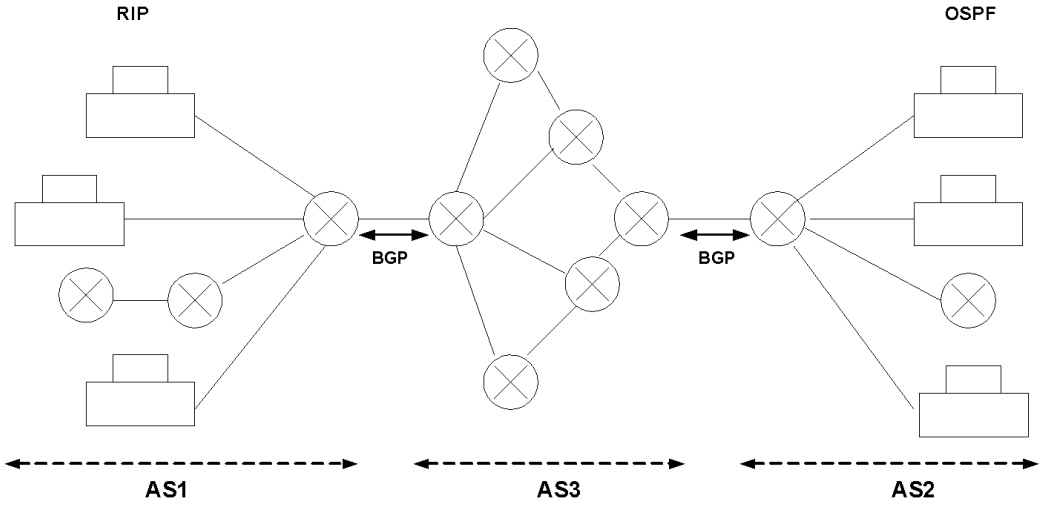
Curs 6

Protocoalele de rutare în internet

În internet se folosesc aşa numitele ***sisteme autonome*** care sunt gestionate de un administrator unic. Numerele de identificare a sistemelor autonome sunt gestionate de IANA (**Internet Assigned Numbers Authority**) si trebuie inregistrate in mod oficial. În intervalul sistemelor autonome se folosesc protocoalele RIP (Routing Information Protocol) – vector distanţă; sau OSPF (Open Shortist Path First) – stare legături. În special OSPF – ul implică cunoaşterea topologiei reţelei = > probleme de securitate = > necesitatea unui administrator comun. Între sistemele autonome se folosesc protocoale care permit transferul de date indiferent de structura internă a sistemului. Cei mai cunoscuţi sunt: BGP (Border Gateway Protocol).

Există 3 tipuri de sisteme autonome (AS):

- ***stub*** (sisteme autonome închise) (AS_1 , $AS2_2$) - au decat o cale către exterior
- ***transit*** sisteme autonome pentru providerii de internet (au cel puţin 2 cai de ieşire).
- ***multihomed*** sisteme autonome cu mai multe cai catre exetrior, dar care nu permit tranzitarea lor de catre traficul generat in alte sisteme autonome.



Protocoale de routare exterioare (BGP)

Foloseste TCP pentru a mentine conexiuni intre rutere situate in AS diferite.

Acestea trebuie sa resolve urmatoarele probleme:

- Probleme de topologie (trebuie sa compatibilizeze structuri diferitelor sisteme autonome);
- Autonomia sistemelor autonome (gasirea cailor de la sursa la destinatie este lasata la latitudinea sistemelor autonome); nu se poate selecta o cale optima de la sursa la destinatie;
- Increderea (anumite sisteme autonome nu isi fac publice informatii despre traficul intern; de exemplu 2 provideri aflati in concurenta);
- Politica diferita de routare in interiorul sistemelor autonome (numar minim de salturi contar cost minim)

Protocoale de rutare interioara (RIP, OSPF)

În protocoalele de rutare în interiorul unui sistem autonom sunt:

- RIP (Routing Information Protocol)
  - foloseşte algoritmul vector – distanţă (algoritmul distribuit Bellman - Ford);
  - actualizările se trimit la 30 secunde;
  - nu solicită autentificare;
  - folosit adesea în începutul apariţiei internetului datorită necesarului mai mare al puterii de calcul
- OSPF (Open Shortest Path First)
  - algoritm de rutare cu actualizarea stării legăturilor;
  - foloseşte inundarea reţelei (flood) cu mesaje de actualizare doar când este necesar;
  - foloseşte algoritmul Dijkstra;
  - necesită autentificare;
  - poate diviza sistemele autonome;
  - este folosit în present datorită evoluţiei sistemelor de calcul.

Între sistemele autonome se folosesc protocoale de rutare exterioare care trebuie să resolve următoarele probleme:

- securitate: priveşte promovarea traseelor corecte către destinaţie;
- topologie: fiecare reţea are propria topologie;
- politici diferite de rutare De exemplu rutarea în cel mai mic număr de salturi, rutare pe linii cu cost minim;
- autonomia sistemelor autonome: nu este posibilă calcularea costului minim al unui traseu datorită definirii costului în mod diferit de către sistemele autonome.

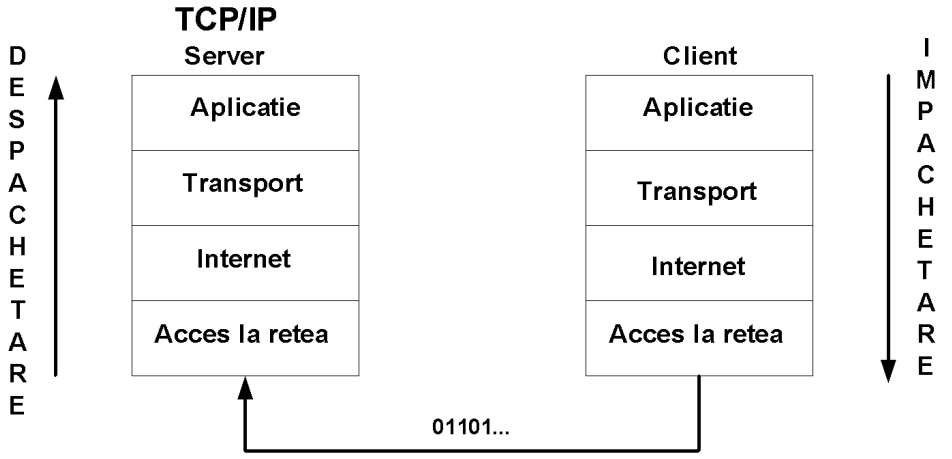
Protocolul de rutare între sisteme autonome este BGP (Border Gateway Protocol). BGP foloseşte sistemul de rutare vector de trasee: sunt transmise informaţii despre lista sistemelor autonome care trebuiesc parcurse pentru a ajunge la destinaţie.

Buclele de rutare sunt depistate şi ignorate. Politicile locale de rutare se aplică pentru alegerea căii optime.

Comparatie RIP-OSPF

- Dimensiune actualizari/convergenta ;
  - Pachete mai mari pentru OSPF
  - Trafic constant de dimensiuni mai mici pentru RIP
  - Trafic de dimensiuni mari la aparitia modificarilor la OSPF
- Tabele de routari/securitate;
  - OSPF cunoaste intreaga topologie
  - RIP cunoaste doar starea vecinilor
- Robustete ;
  - RIP poate transmite mesaje incarcate
  - OSPF poate corecta mesajele primite pe baza celorlalte mesaje

Nivelul transport şi protocoalele nivelului transport (TCP şi UDP)



Nivelul transport asigura transmiterea informatiei de la sursa la destinatie furnizand mijloacele pentru depistarea aparitiei erorilor si reasamblarea ordonata a informatiei la destinatie.

Cele mai cunoscute protocoale sunt: Transmission Control Protocol (TCP), UDP (User Datagram Protocol). Unitatea de protocol la nivelul transport este segmentul pentru protocolul TCP si datagrama pentru protocolul UDP.

La acest nivel gasim identificatoare pe 16 biti ale aplicatiie ruland pe un sistem de calcul conectata la retea, numite porturi:

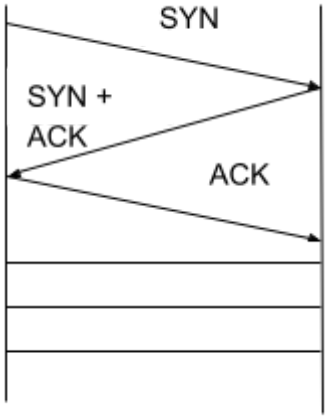
Ports 0 - 1023 well-known ports.
1024 - 49151 Registered ports.
49152 - 65535 can be used dynamically by applications.

Caracteristicile TCP

TCP (Transmision Control Protocol) – protocol orientat pe conexiune deoarece foloseste un schimb de date (inainte de transmiterea datelor utilizatorilor) in 3 etape pentru stabilirea conexiunii si o secventa 2 x 2 pentru intreruperea conexiunii. Algoritmul TCP este similar desfasurarii unei convorbiri telefonice intre doua persoana deoarece informatia care nu este inteleasa este retrimisa imediat.

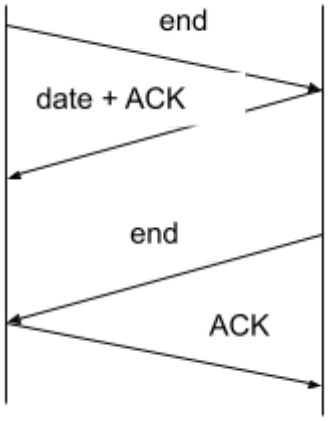
TCP – ul este orientat pe conexiune şi foloseşte algoritmul în 3 paşi de stabilire a conexiunii. În acest algoritm clientul are rol activ şi serverul are rol pasiv.

Etapele se desfăşoară astfel:



SYN – sincronizare , ACK - acknowledge
end – terminare transfer de date

Algoritmul pentru închiderea conexiunii este urmatorul:



Algoritmul de închidere a conexiunii are 4 paşi. TCP transmite octeţii în stream – uri de octeţi numite segmente.

TCP este fiabil:

- primirile de date sunt confirmate cu un ACK;
- se folosesc sume de control pentru depistarea datelor eronate;
- se folosesc numere de secvenţă pentru reasamblarea în ordine a datelor primite;
- datele eronate se retransmit după o pauză;
- TCP realizează controlul traficului de date pentru a împiedica umplerea bufferului de recepţie.

Transmiterea datelor în segmente

Un segment TCP include mai mulţi octeţi, limitarea dimensiunii acestuia fiind dată de:

- atingerea dimensiunii maxime permise
- expirarea timpului de transmisie a sistemului
- forţarea transmisiei de catre aplicaţie.

Structura segmentului TCP

Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

0	Port sursa										Port destinatie
32	Numar de secventa										
64	Numar ACK										
96	Offset date	Reservat	C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Dimensiune fereastra
128	Suma de verificare										Indicator urgenta
160 ...	Optiuni (daca Data Offset > 5) ...										
	Date										

Portul în reţele de calculatoare este un număr care identifică aplicaţia utilizată. Asocierea PORT + IP se numeste socket.

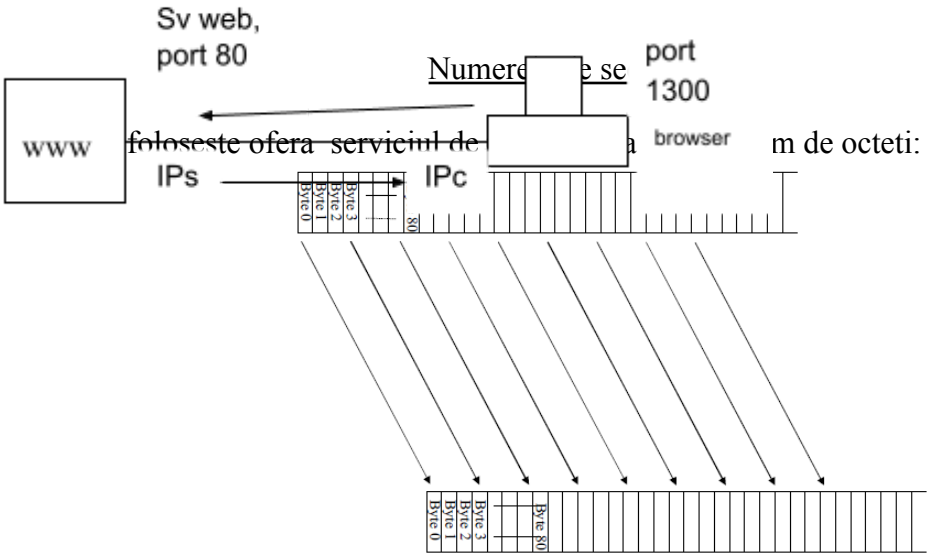
TCP flolseşte fereastra de comunicaţie pentru a transmite mai multe date fără confirmare.

Porturi Sunt 2^{16} porturi.

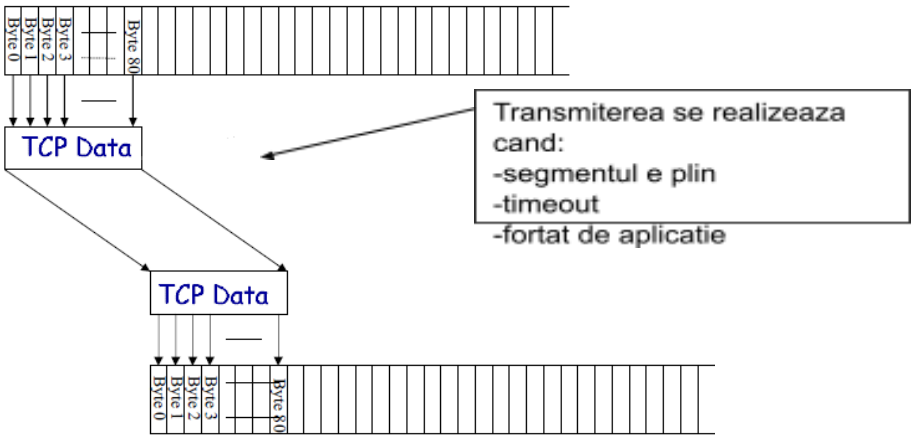
0 – 1023 - rezervate pentru aplicaţii bine cunoscute gen FTP, HTTP , etc.

1023 – 49151 – porturi rezervate aplicaţiilor comerciale

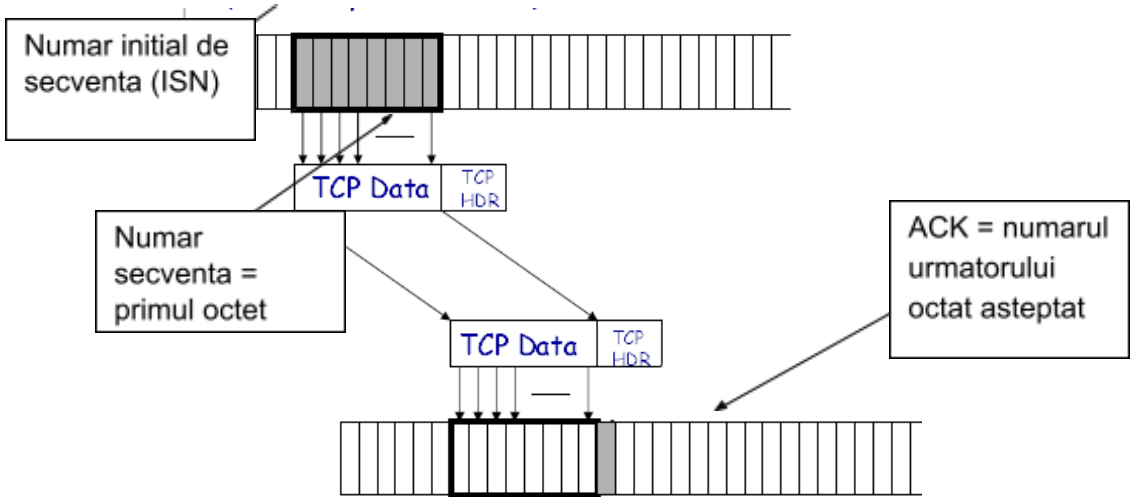
49152-65535 – porturi dinamice care sunt alocate la client in mod dinamic la conectarea la un server.



Care este emulat prin folosirea numerelor de secventa:



Numerele de secvenţă sunt folosite pentru a identifica următorul octet ce urmează a fi trimis.



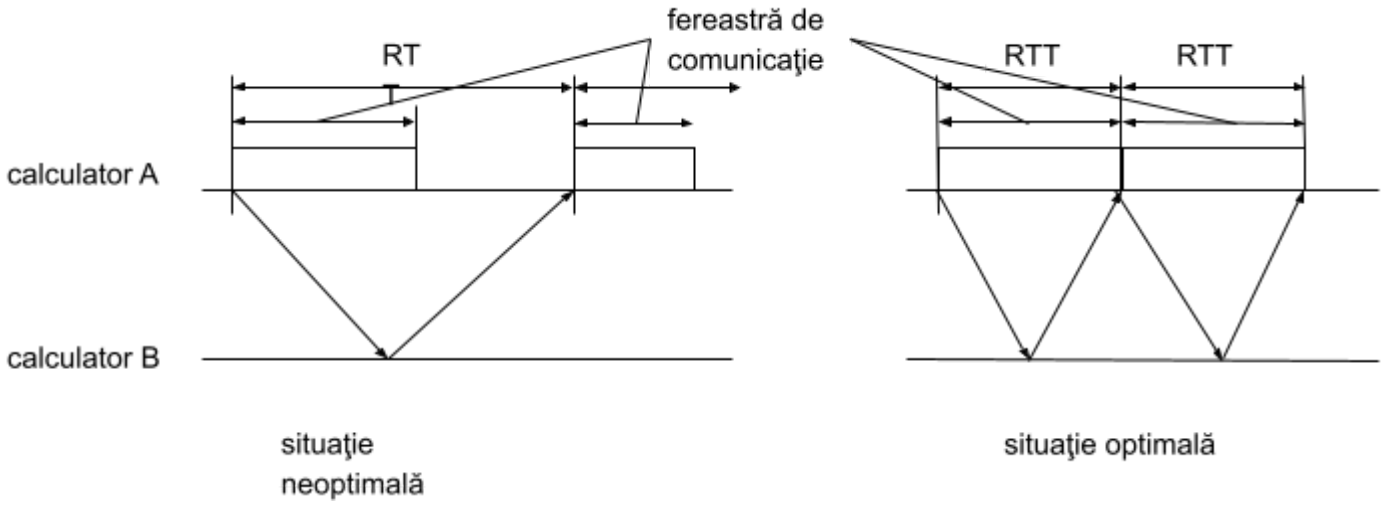
Daca apare o eroare, TCP trebuie sa retransmita informatia de la ultimul ACK corect receptionat.

Curs 7

Fereastra de comunicare TCP

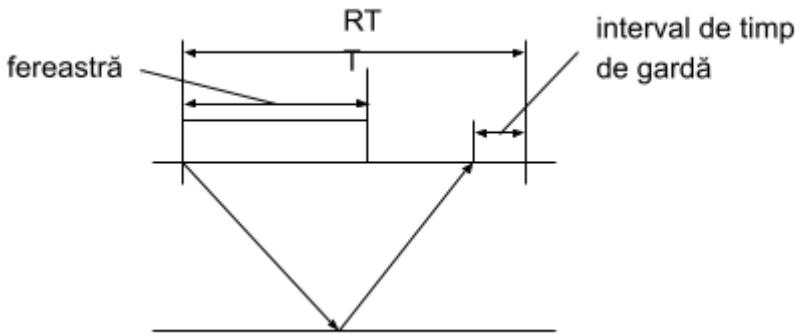
In cazul in care s-ar trimite confirmari dupa fiecare segment TCP, transmisia ar fi ineficienta pentru ca acestea sunt necesare in cadrul protocolului pentru a verifica receptia corecta a segementelor la dfestinatie. Fereastra de transmisie reprezintă cantitatea de date transmisă înainte de a se astepta primirea unei confirmări a recepţiei => 4-8 KOcteti.

Un termen asociat cu dimensiunea ferestrei este RTT (Round Trip Time) adică durata de timp de la transmitere mesaj până la prinire confirmare. Dimensiunea ferestrei are valoare optimă dacă se apropie de RTT.



Este important ca timpul până la primirea confirmării să fie cât mai mic (şă nu se piardă timp aşteptând confirmarea). Parametrul care monitorizează intervalul de la transmisie până la recepţia confirmării se numeşte RTT (Round Trip Time).

Deoarece încărcarea reţelei variază între diferite momente, routerele trebuie să aibă o marjă de eroare în valoarea RTT numita interval de timp de gardă.

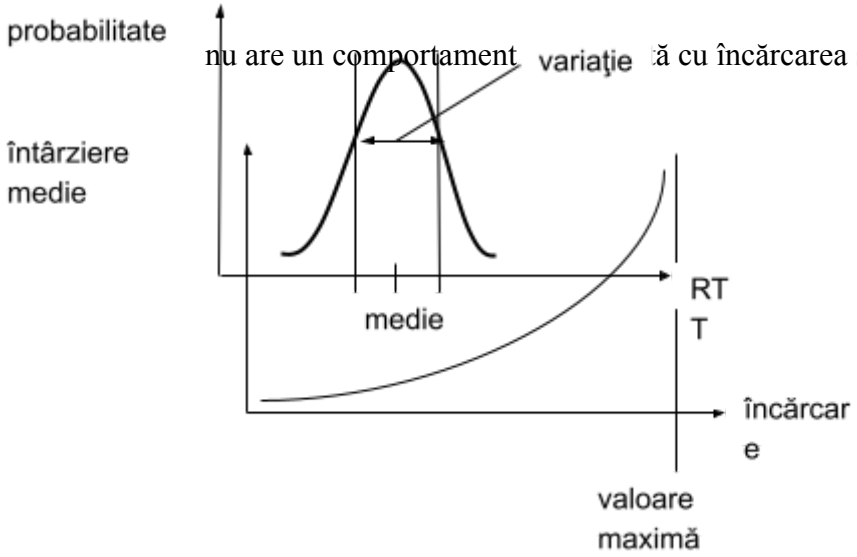


Dacă valoarea timpului de gardă este prea mare se va aştepta prea mult înainte de a transmite un pachet.

Dacă valoarea timpului de gardă este prea mică vor fi retransmise segmente în mod inutil.

Valoarea totală a timpului trebuie să fie egală cu valoarea estimată a timpului de transmisie până la recepţie.

Valorile retransmisie într-o reţea de calculatoare are o variatie cu distribuţia de forma clopotului lui Gauss.



nu are un comportament ă cu încărcarea sa.

Protocolul UDP (User Datagram Protocol)

UDP (User Datagram Protocol)– nu are o secventa de initializare, prin urmare pachetele pot urma oricand, fara a fi anuntate in prealabil. Are o structură mai simplă, deci pachetul de date va fi mai mic.

Pentru serviciile care nu sunt critice (DNS) se recomandă utilizarea UDP, ca protocol de suport, deoarece încarcă mai puţin reţeaua.

Datele transmise de UDP sunt însoţite de o sumă de verificare care permite împreună cu indicatorul de lungime să se ia decizia dacă datele au ajuns corect la destinaţie.

Nu conţine mecanisme pentru detectarea pachetelor lipsă sau celor care nu sunt în ordine. Nu există mecanism pentru controlul traficului de date.

Aplicaţiile UDP sunt: DHCP, TFTP, DNS, aplicaţii audio, video

Alte aplicaţii care nu au nevoie de transmisii sigure sunt aplicaţiile de monitorizare a reţelei.

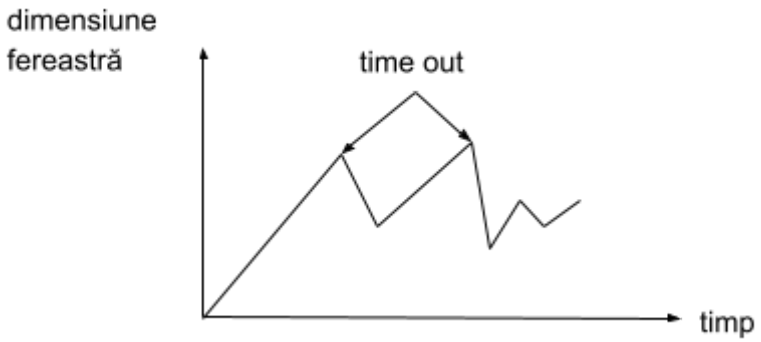
Controlul congestiei traficului în reţele de calculatoare

Congestia este inevitabilă. Există mai multe cauze ale congestiei:
- mai multi utilizatori accesează reţeaua într-o oră de vârf;
- este folosită la maxim toată lăţimea de bandă;
- trafic masiv primit de router într-un interval scurt de timp.

Pentru evitarea congestiei protocolul utilizat este TCP.

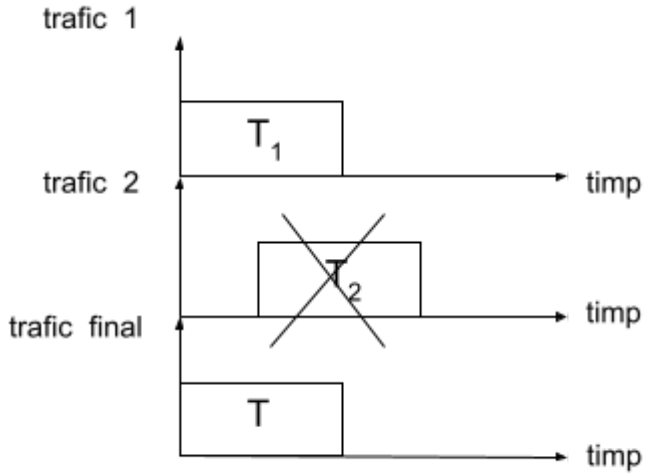
TCP foloseşte algoritmul creştere aditivă, descreştere multiplicativă.

TCP foloseşte mecanismul de repornire numit slow start.

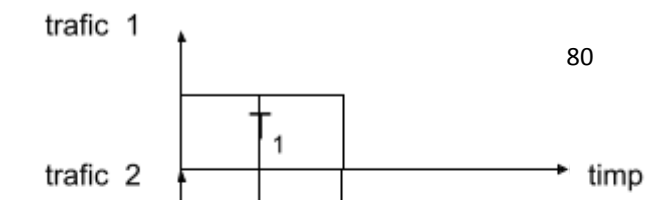


Mecanisme pentru tratarea congestiei în reţea pentru routere

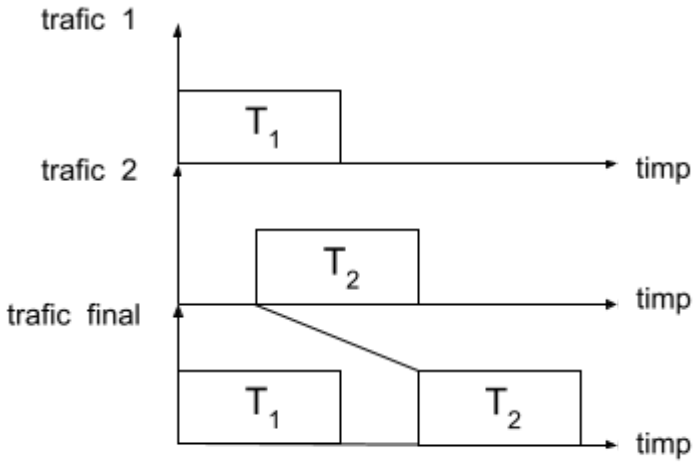
Se elimină un trafic de date.



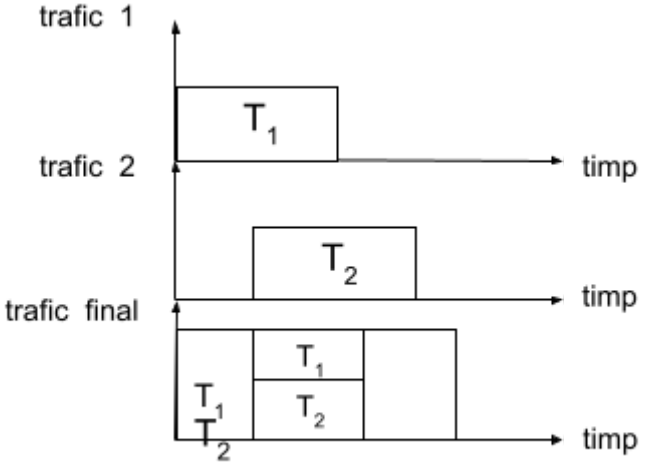
Dacă bufferul permite traficul 2 el este întârziat până la terminarea traficului 1.



Reprogramarea traficului pentru un moment de timp ulterior.



Reducerea traficului pentru fiecare din transmisii şi gestionarea lor în paralel.



O măsură a performanţei reţelei este raportului între încărcare şi întârziere:

$$performanța = \frac{\text{încărcare}}{\text{întârziere}}$$

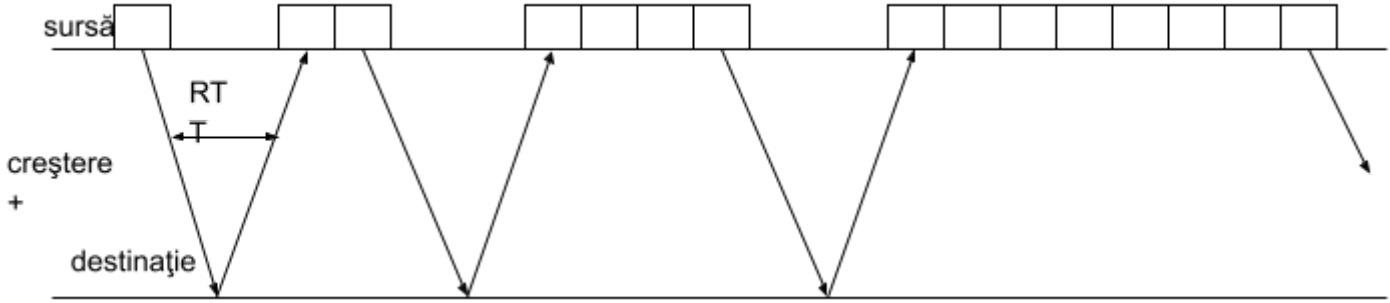
Curs 8

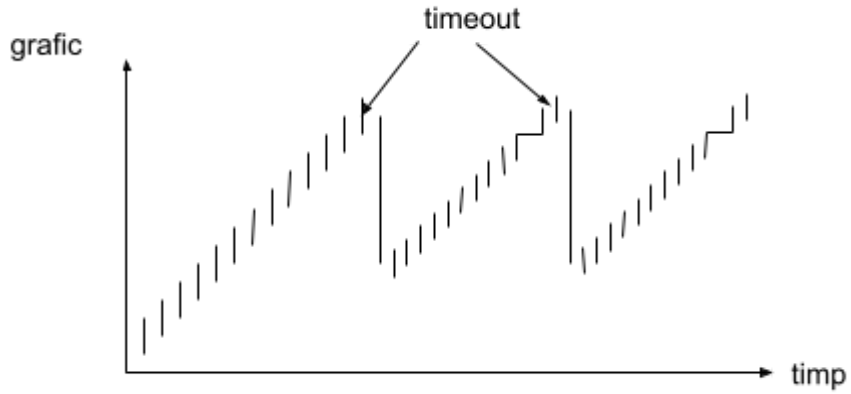
Controlul congestiei în reţelele de calculatoare (cont.)

Protocolul TCP transmite pachete şi reacţionează la pierderea pachetelor. Sursa încearcă astfel să determine capacitatea disponibilă la destinaţie. Prin urmare comunicaţia se efectuează la minimul dintre fereastra propusă de sursă şi cea propusă de destinaţie: fereastră = min{fereastră sursă; fereastră destinaţie}.

Fereastra de comunicaţie este modificată printr-un algoritm de crestere aditivă: $fereastra_{I+1} = fereastra_I + 1$

descreştere multiplicativă : fereastră$_{I+1}$ = fereastră / 2 = fereastră OS





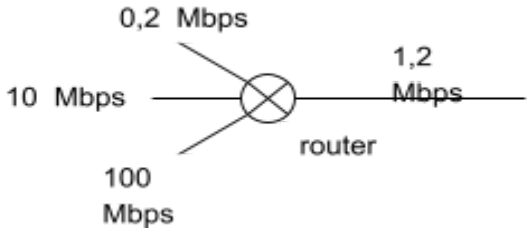
Viteza de transmisie TCP

Este determinată de raportul între dimensiunile ferestrei şi timpul de la transmiterea pachetului până la primirea confirmării. Pentru sistemul de operare Windows dimensiunea maximă a ferestrei este de 12 pachete. Pentru Linux dimensiunea maximă este de 40 pachete.

Pentru a nu se depăşi dimensiunile bufferului de recepţie acesta are o valoare mare: aproximativ 200 pachete => routerul nu trebuie să piardă pachete.

Evitarea congestiei

Mecanismele descrise anterior reacţionează după apariţia congestiei. În general routerele au implementate mecanisme pentru evitarea congestiei. Acestea sunt încă în continuă dezvoltare.



Mecanisme de evitare a congestiei

- folosirea bitului de congestie (avertizarea surselor despre apariţia congestiei);
- anunţarea sursei şi destinaţiei prin picarea intenţionată a pachetelor (aleatoriu);
- routerele modifică RTT – ul (întârzierea) şi semnalizează astfel sursa şi destinaţia.

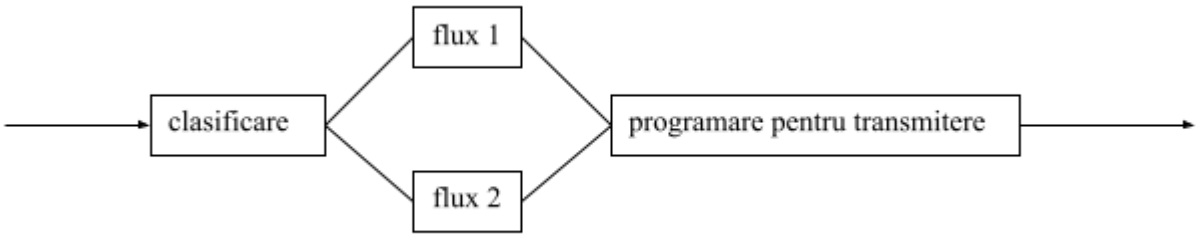
Calitatea serviciului

QoS (Quality of Service) este un mecanism prin care se încearcă măritea şanselor pentru realizarea unei transmisii de către o anumită sursă de trafic. Este totodată şi o metodă de control în reţele de calculatoare.

TCP este un protocol care favorizează transmisiile să îşi efectueze transferul mini=m necesar trebuiesc stabilite restricţii.

Mecanisme QoS

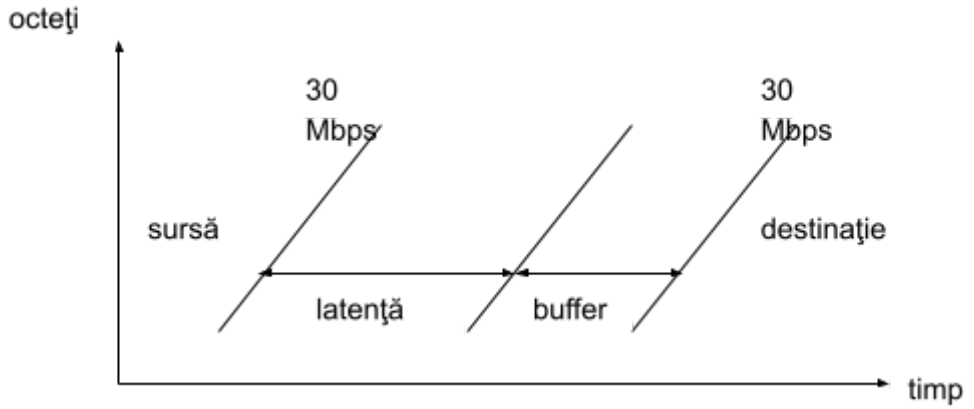
Alocarea traficului se face per flux de date. Fiecare flux este programat succesiv şi tratat în funcţie de priorităţi mecanismul numindu-se Bit by Bit Fair Queuing.



Al doilea mecanism este Bit by Bit Fair Queuing – Ponderat. Prioritatea de tratare se face în funcţie de ponderea dată fiecărei linii de comunicaţie. Pentru liniile cu trafic mai mare sunt procesaţi mai mulţi biţi.

Exemplu. Să se calculeze necesarul de trafic de date pentru a transmite video de 10 imagini pe secundă de dimensiunea 640 x 480 cu imagini 32 de biti.

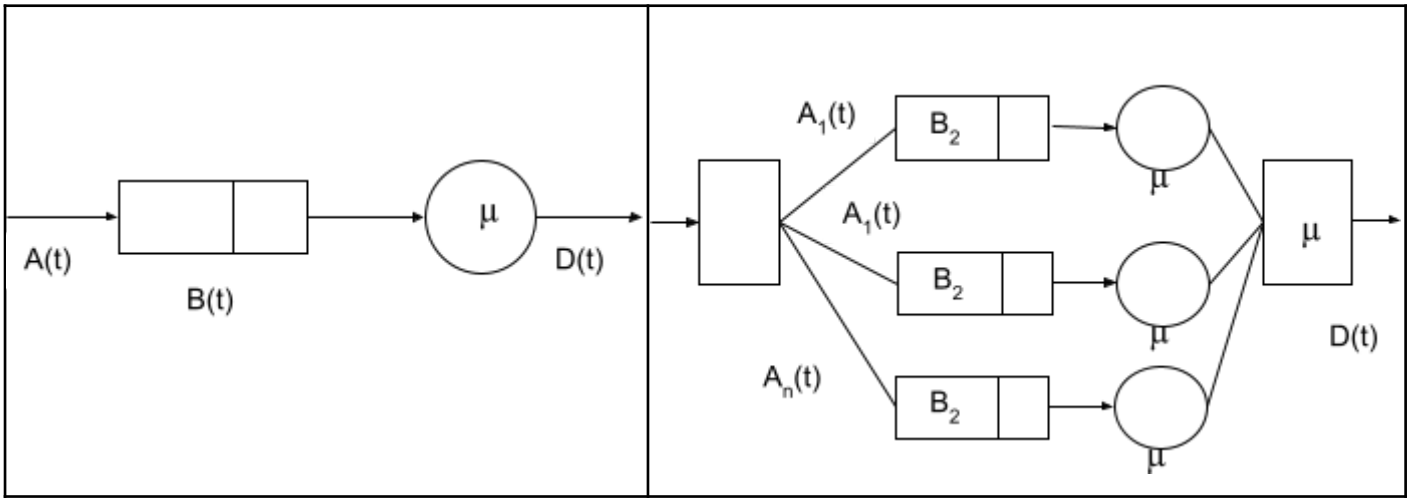
$640 \cdot 480 \cdot 10 \cdot 32 = 98$ Mb



Întârzierea de la sursă la destinaţie poate să varieze. Variaţiile trebuiesc acoperite de buffer.

Un router care suportă QoS este mai scump decât un router normal. Structura lui este următoarea:

Router fără QoS	Router cu QoS



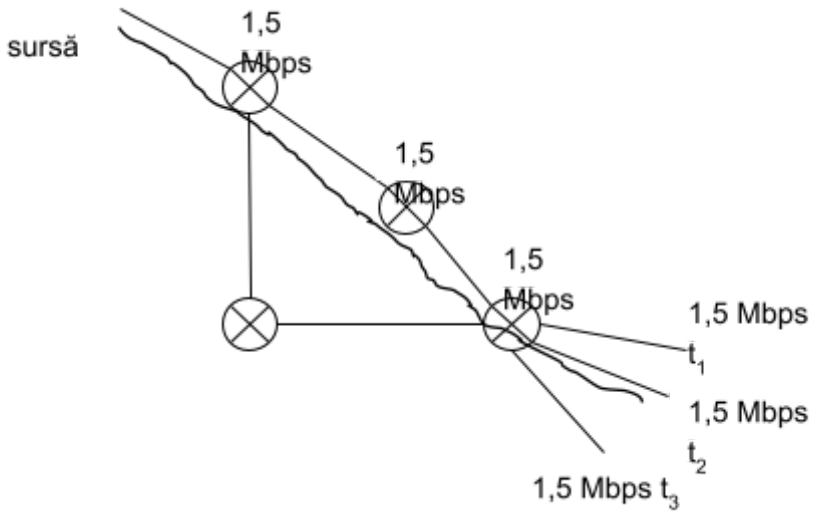
Mod de operare trafic cu prioritate

1. Sursa interoghează reţeaua pentru determinarea întârzierii garantate
2. Se negociază cu fiecare router de pe traseu o valoare a întârzierii garantate
3. Negociază cu fiecare router un anumit delay. Prin urmare sursa poate estima latenţa reţelei.
4. Routerele rezervă traficul
5. Se transmit pachete care sunt separate de router în diferite fluxuri de date.

Un protocol care este folosit în asigurarea calităţii serviciului este RSVP (Resource Reservation Protocol).

Protocolul RSVP este cel care negociază traficul între sursă şi destinaţie de-al lungul întregii căi de comunicaţie.

Exemplu. Două staţii solicită întârzieri de 100 ms, respectiv 50 ms. Routerul alege 50 ms. Routerul primeşte cereri de la 100 solicitanţi de 1,5 Mb trafic video ale aceleiaşi nideoconferinţe, routerul alocă doar 1,5 Mbps pentru toţi.



Curs 9

Nivelul aplicatie

Hypertext Transfer Protocol (HTTP/HTTPS) - TCP Port 80/443
Simple Mail Transfer Protocol (SMTP) - TCP Port 25
SMTP+ TLS:587
Post Office Protocol (POP) - TCP Port 110; POP+TLS **995**
Telnet - TCP Port 23 // SSH -22
Domain Name System (DNS) - TCP/UDP Port 53
Dynamic Host Configuration Protocol - UDP Ports 67 and 68
TFTP - UDP 69
File Transfer Protocol (FTP) - TCP Ports 20 and 21 (si pentru TLS)
SNMP UDP 161 trap 162
IMAP 143 IMAP+TLS **993**
Syslog UDP 514. Alternativ, TCP 601

Urmatoarele protocoale importante de nivel aplicatie sunt accesibile pe Web:

- E-mail (Simple Mail Transport Protocol or SMTP) - Posta electronica; Distribuie electronic mesaje si fisiere catre una sau mai multe adrese
- Telnet (Telnet Protocol) - Ofera posibilitatea de conectare la un calculator gazda si de a executa diverse comenzi pe acel calculator. Portul standard folosit este TCP 21.
- FTP (File Transfer Protocol) -Protocol de transfer de fisiere; Transfera fisiere text sau binar intre un server si un client FTP. Versiunea securizata se numeste SFTP (Secure FTP).
- HTTP (HyperText Transfer Protocol) - Protocol de transfer HyperText; Serviciu Internet ce permite receptarea de informatii organizate în standard HTML. Aceste informatii sunt prelucrate si apoi afisate de browser-ul dvs. Versiunea securizata se numeste HTTPS (Secure HTTP).
- SSH (Secure Shell) este un set de protocoale de internet standard şi asociate care permite stabilirea unui canal de comunicare sigur între un computer local şi unul controlat de la distanţa (remote). Este folosită criptarea prin cheie publica pentru a autentifica computerul controlat şi (opţional) să permită computerului controlat să autentifice utilizatorul. SSH asigură confidenţialitatea şi integritatea datelor schimbate între cele două computere prin folosirea criptării şi a codurilor de autentificare a mesajelor. SSH este de obicei folosit pentru conectarea de la distanţă şi executarea de comenzi pe acel computer , dar suportă de asemeni tunnelare; poate face transfer de fişiere folosind protocoalele asociate SFTP sau SCP . Un server SSH , implicit , ascultă pe portul standard TCP 22.
- SNMP are 3 versiuni, introducand securizare precum si utilizator si parola in ultima versiune. Acest protocol poate transmite la cerete date despre traficul realizat prin acel echipament.
- **Syslog** este un protocol ce permite schimbul de informatii intre echipamentele de retea si un server. Sunt transmise evenimenele care au loc pe echipamente pentru a putea identifica actiunile effectuate si eventualele probleme. Syslog foloseste portul UDP 514. Alternativ, TCP 601 este uneori folosit.

Serviciile Telnet si SSH

Permite utilizatorilor sa se conecteze la un calculator din retea. Utilizatorii pot incepe o sesiune la distanta specificand un calculator distant (prin nume sau IP) la care vor sa se conecteze. Din acest moment pana la sfarsitul sesiunii orice actiune a utilizatorului este transmis calculatorului distant. Statia locala a utilizatorului se va comporta ca un terminal simplu al calculatorului distant.

Acest serviciu foloseste protocolul cu acelasi nume ale modelului TCP/IP. Oricare din aplicatiile client de telnet functioneaza pe nivelul Aplicatie al modelului TCP/IP.
SSH (Secure Shell) este un serviciu specific sistemelor de operare (SO) Linux. Avantajul acestui protocol fata de protocolul Telnet este securizarea comunicatiei intre cele doua calculatoare.

Mesageria electronica. Sistemul e-mail

Permite unui utilizator transmiterea de mesaje electronice (e-mail) altor utilizatori din retea. In vederea transmiterii unui astfel de mesaj, cel care initiaza comunicatia trebuie sa gaseasca o cale de comunicatie catre sistemul apelat. Acest lucru intra in sarcina unui server e-mail sau MTA (Mail

Transfer Agent), sistem ce va transfera e-mail-ul de la utilizator (prin intermediul unui client de e-mail) unui alt server e-mail, ce are conexiune directa cu sistemul destinatie.

Serverele de e-mail comunica intre ele folosind protocolul SMTP (Simple Mail Transfer Protocol) pentru trimiterea si receptia mesajelor. Protocolul SMTP transporta mesajele e-mail in format ASCII folosind protocolul TCP. Fiecare utilizator are o cutie postala (mailbox) pe serverul de e-mail, unde ajung mesajele provenite de la ceilalti utilizatori.

Exista o serie de modalitati pentru clientii de e-mail pentru a colecta mesajele unui utilizator: folosind programe ce acceseaza direct serverul de e-mail sau folosind protocoale de retea. Cele mai des utilizate protocoale sunt : POP3 (Post Office Protocol) si IMAP (Internet Message Access Protocol), care folosesc protocolul TCP pentru transportul datelor. Desi clientii de e-mail folosesc aceste protocoale speciale pentru colectarea mesajelor, in cele mai multe cazuri folosesc protocolul SMTP pentru transmiterea lor.

Pentru a verifica daca serverul de e-mail este disponibil si corect configurat se testeaza portul SMTP (25) sau portul POP3 (110) cu comanda telnet. Se poate folosi urmatoarea comanda, introdusa in linia de comanda a sistemului Linux sau Windows, pentru testarea serverului de e-mail de la IP-ul respectiv:

[student@bodhi /student]$ **telnet 192.168.1.1 25**

POP supports download-and-delete requirements for access to remote mailboxes (termed maildrop in the POP RFC's).[2] Although most POP clients have an option to leave mail on server after download, e-mail clients using POP generally connect, retrieve all messages, store them on the client system, and delete them from the server. Other protocols, notably the Internet Message Access Protocol (IMAP) provide more features of message management to typical mailbox operations. A POP3 server listens on well-known port number 110 for service requests. Encryptedcommunication for POP3 is either requested after protocol initiation, using the STLS command, if supported, or by POP3S, which connects to the server using Transport Layer Security (TLS) or Secure Sockets Layer (SSL) on well-known TCP port number 995.

Available messages to the client are fixed when a POP session opens the maildrop, and are identified by message-number local to that session or, optionally, by a unique identifier assigned to the message by the POP server. This unique identifier is permanent and unique to the maildrop and allows a client to access the same message in different POP sessions. Mail is retrieved and marked for deletion by message-number. When the client exits the session, the mail marked for deletion is removed from the maildrop.

Protocolul SMTP nu ofera o securitate foarte buna si nu presupune nici o autentificare. Prin urmare administratorii de retea nu permit calculatoarelor care nu fac parte din reteaua locala sa foloseasca serverul propriu de e-mail pentru a transmite mesaje e-mail.

Un server SMTP trebuie sa cunoască cel puţin următoarele comenzi:

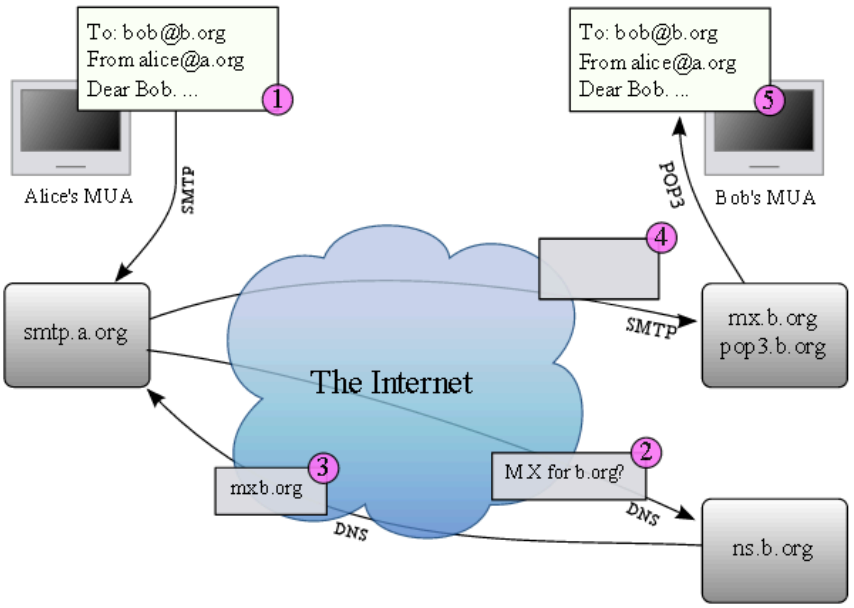
- HELO – identificare calculator expeditor;
- EHLO – identificare calculator expeditor cu cerere de mod extins;
- MAIL FROM – specificare expeditor;
- RCPT TO – specificarea destinatarului;
- DATA – conţinutul mesajului;
- RSET – Reset;
- QUIT – termina sesiunea;
- HELP – ajutor pentru comenzi;
- VRFY – verifica o adresa;
- EXPN – expandeaza o adresa;
- VERB – informaţii detaliate.

POP3 trebuie sa cunoasca urmatoarele comenzi:
STAT –indica nr de mesaje si dimensiunea acestora
LIST – aficare mesaje detaliat
RETR 1 – descarcare mesaj
DELE 1 – stergere mesaj

USER / PASS

Diagrama urmatoare arată o secvenţă tipică de evenimente, care are loc atunci când Alice compune un mesaj folosind agentul ei de mail (MUA). Ea intră în adresa de e-mail a corespondent ei, şi apasa butonul "Trimite".

1. MUA formateaza mesajul în format de e-mail şi utilizează Simple Mail Transfer Protocol (SMTP) pentru a trimite mesajul către agentul de mail locală de transfer (MTA), în acest caz smtp.a.org, condusa de providerul de internet - Internet Service Provider (ISP).
2. MTA se uită la adresa de destinaţie prevăzuta în protocolul SMTP (nu din antetul mesajului), în acest caz, bob@b.org. O internet adresa de e-mail este un şir de formatul utilizator@domeniu.com. Partea înainte de semnul @ este partea locala, de multe ori numele de utilizator al destinatarului, iar partea de după semnul @ este un nume de domeniu sau un nume de domeniu complet calificat. MTA rezolvă un nume de domeniu pentru a determina numele de domeniu complet definit al serverului de schimb de mail în Domain Name System (DNS).
3. Serverului DNS pentru domeniu b.org, ns.b.org, răspunde cu toate înregistrările MX listarea servere de mail de schimb pentru acest domeniu, în acest mx.b.org caz, un server de a alerga de ISP-ul lui Bob.
4. smtp.a.org trimite mesajul pentru a mx.b.org folosind SMTP, care se livrează la căsuţa poştală de bob de utilizator.
5. Bob apasa "Get Mail", buton pe agentul sau MUA, care preia mesaj folosind Post Office Protocol (POP3).

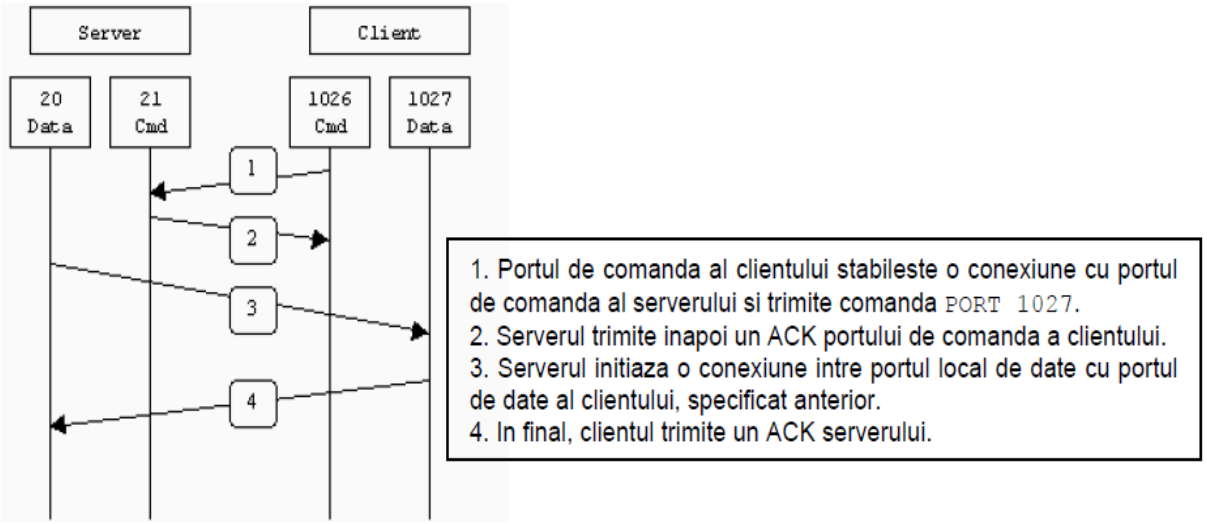


Serviciul FTP

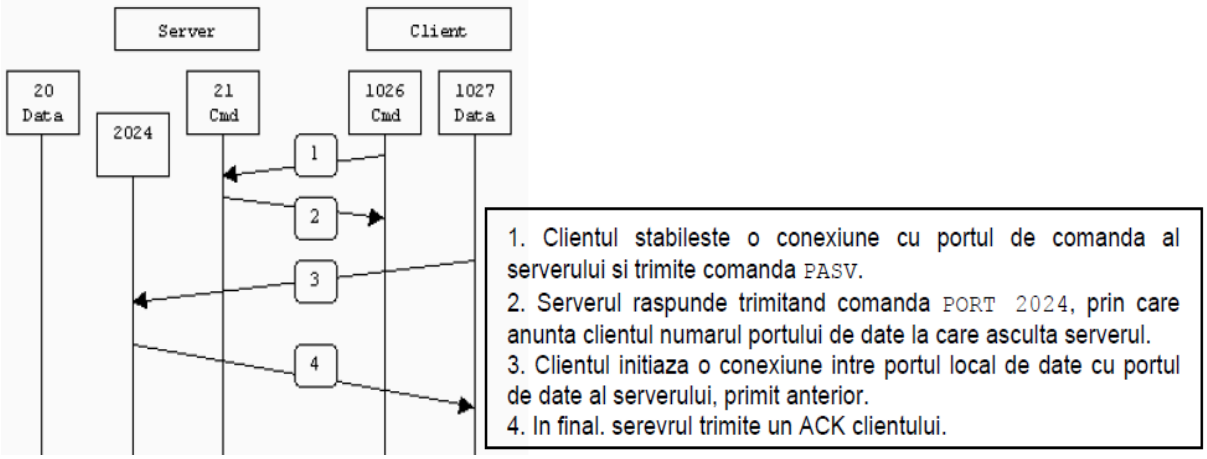
Permite transferul fisierelor intre calculatoarele dintr-o retea, prin copiere sau mutare. Este un serviciu stabil, orientat pe conexiune. Protocolul FTP utilizeaza TCP pentru trasnsferul datelor.
Atunci cand fisierele sunt copiate de pe un server ce suporta FTP, acest serviciu stabileste mai intai o conexiune de control intre client si server. A doua conexiune ce se stabileste ulterior, reprezinta legatura intre cele doua calculatoare prin care se transfera datele. Portul de comanda folosit este portul 21 respectiv portul pentru date este 20. Transferul de date se poate efectua in mod ASCII sau in mod binar. Aceste moduri determina codarea fisierelor de date.
La finalul tranferului de fisiere, conexiunea de date se termina automat. Conexiunea de control se termina cand utilizatorul inchide complet sesiunea.

In plus, exista doua tipuri de conexiuni la un server de FTP : activ si pasiv.

In modul FTP activ, clientul stabileste conexiunea de la un port neprivilegiat (*N*>1023) la portul de comanda 21 al serverului FTP. Dupa aceea, clientul va incepe sa asculte la portul *N+1* si trimite comanda FTP, PORT N+1 serverului FTP. La randul sau, serverul se va conecta de la portul de date local 20 la portul de date specificat anterior de client.



In modul FTP pasiv, clientul va deschide aleator doua porturi neprivilegiate (*N*>1023 si *N+1*). Primul din aceste porturi va contacta serverul pe portul 21 si trimite comanda FTP, PASV. Rezultatul acestei comenzi transmise de client, este deschiderea de catre serverul FTP unui port neprivilegiat aleator (*P*>1023) si trimiterea comenzii FTP, PORT *P*. In final, clientul va stabili o conexiune intre portul *N+1* si portul *P* al serverului FTP pentru tranferul datelor.



Protocolul HTTP

HTTP (Hypertext Transfer Protocol) este metoda cea mai des utilizată pentru accesarea informaţiilor în Internet care sunt păstrate pe servere World Wide Web (WWW). HTTP oferă o tehnică de comunicare prin care paginile web se pot transmite de la un computer aflat la distanţă spre propriul computer. Dacă se apelează un link sau o adresă de web cum ar fi http://www.example.com, atunci se cere calculatorului host să afişeze o pagină web (index.html sau altele). În prima fază numele (adresa) www.example.com este convertit de protocolul DNS într-o adresă IP. Urmează transferul prin protocolul TCP pe portul standard 80 al serverului HTTP, ca răspuns la cererea HTTP-GET. Informaţii suplimentare ca

de ex. indicaţii pentru browser, limba dorită ş.a. se pot adăuga în header-ul (antetul) pachetului HTTP. În urma cererii HTTP-GET urmează din partea serverului răspunsul cu datele cerute, ca de ex.: pagini în (X)HTML, cu fişiere ataşate ca imagini, fişiere de stil (CSS), scripturi (Javascript), dar pot fi şi pagini generate dinamic (SSI, JSP, PHP şi ASP.NET). Dacă dintr-un anumit motiv informaţiile nu pot fi transmise, atunci serverul trimite înapoi un mesaj de eroare. Modul exact de desfăşurare a acestei acţiuni (cerere şi răspuns) este stabilit în specificaţiile HTTP.

http/2 ….. transfer de date in format binar si transferul resurselro exclusiv asincron

Versiuni

În prezent se utilizează două versiuni ale protocolului, HTTP/1.0 şi HTTP/1.1. La versiunea HTTP/1.0 se stabileşte o nouă conexiune TCP înaintea cererii, iar după transmiterea răspunsului conexiunea trebuie închisă. Astfel dacă un document HTML cuprinde 10 imagini, vor fi necesare 11 conexiuni TCP, pentru ca pagina să fie afişată complet (în browser). La versiunea 1.1 se pot emite mai multe cereri şi răspunsuri pe aceeaşi conexiune TCP. Astfel pentru documentul HTML cu 10 imagini este necesară doar o singură conexiune TCP. Deoarece - datorită algoritmului Slow-Start - viteza conexiunii TCP este la început mică, dar acum el e necesar doar o singură dată, se scurtează semnificativ durata totală de încărcare a paginii. La aceasta se adaugă şi faptul că versiunea 1.1 poate relua şi continua transferuri întrerupte.

La HTTP se pierd informaţiile cererilor vechi (deci este un protocol fără reţinerea stării). Prin utilizarea de cookes-uri în header, se pot realiza însă aplicaţii care pot utiliza informaţii de stare (opţiunile utilizatorului din sesiunea actuală, coş de cumpărături ş.a.). Chiar şi o recunoaştere a utilizatorului este astfel posibilă. În mod normal se pot citi informaţiile transmise care parcurg reţeaua pe computere şi rutere. Prin HTTPS transferul se poate şi cripta.

Metode

Metodele disponibile sunt :

GET: este cea mai folosită metodă, fiind utilizată atunci când serverului i se cere o resursă.

PUT: metoda este folosită pentru a depune documente pe server, fiind inversul metodei GET.

HEAD: se comportă exact ca metoda GET, dar serverul returnează doar antetul resursei, ceea ce permite clientului să inspecteze antetul resursei, fără a fi nevoit să obţină şi corpul resursei.

POST: a fost proiectată pentru a trimite date de intrare către server.

DELETE: este opusul metodei PUT.

TRACE: este o metodă folosită de obicei pentru diagnosticare, putând da mai multe informaţii despre traseul urmat de legătura HTTP, fiecare server proxy adăugându-şi semnătura în antetul Via.

OPTIONS: este folosită pentru identificarea capacităţilor serverului Web, înainte de a face o cerere.

CONNECT: este o metodă folosită în general de serverele intermediare.

[modifică]Exemplu

Cererea clientului:

GET / HTTP/1.1

Host: www.example.com

Răspunsul serverului:

HTTP/1.1 200 OK

Date: Mon, 23 May 2009 22:30:34 GMT
Server: Apache/1.4.27 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html

Transferul argumentelor

Deseori utilizatorul doreşte să transmită informaţii speciale la website. Aici HTTP pune la dispozitie două posibilităţi:

- Transferul datelor în combinaţie cu o cerere pentru o resursă (HTTP-metoda "GET")
- Transferul datelor în combinaţie cu o cerere specială (HTTP-metoda "POST")

Datele transferate vin deseori %-codate. La metoda GET se utilizează partea de cerere Uniform Resource Identifiers (URI) cu simbolul ?. Această metodă se utilizează pentru a transfera o listă de parametri, pe care partea opusă trebuie să o ia în considerare la prelucrarea cererii.

Deseori această listă cuprinde perechi de valori separate prin semnul &, care sunt alcătuite din numele parametrului, semnul = şi valoarea parametrului. Rareori se mai utilizează şi semnul ; pentru separarea înregistrărilor listei [1].

Exemplu: la pagina de start dela Wikipedia.ro utilizatorul introduce în câmpul de căutare termenul "pisici", alege categoria "articole" şi apasă butonul de căutare. Atunci browserul trimite următoarea cerere la server:

GET /wiki/special:Search?search=test&go=articol HTTP/1.1 Host: ro.wikipedia.org ...

Serverului Wikipedia află că utilizatorul doreşte să vadă articole despre cuvantul cheie „test". Serverul prelucrează cererea, dar nu trimite un fişier ci redirectează browserul cu un Location-Header spre pagina dorită:

HTTP/1.0 302 Moved Temporarily Date: Fri, 13 Jan 2008 15:12:44 GMT Location: http://ro.wikipedia.org/wiki/test ...

Browserul execută indicaţia şi, pe baza noilor informaţii, emite o nouă cerere:

GET /wiki/test HTTP/1.1 Host: ro.wikipedia.org ...

Serverul răspunde şi oferă pagina cu articole despre cuvantul cheie „test":

HTTP/1.0 200 OK Date: Fri, 13 Jan 2008 15:12:48 GMT Last-Modified: Tue, 10 Jan 2008 11:18:20 GMT Content-Language: ro Content-Encoding: gzip Content-Type: text/html; charset=utf-8

*.‹..... ´ZKs.¹.>Û¿.ž-[¶KÃ!õ²ÌÇlô²¬¬ìuVò*ÉÖ– 3.r`Î+.F"xÊ!ÿ ×.ý.ö´7ý"ü't.ó"9ÔÊ›Ä®.A.ÐÝ*

Partea de date este mai lungă şi de necitit din cauza compresiei gzip.

În cazul unei cereri POST variabilele nu se află în URI, ci în partea body:

POST /wiki/special:Search HTTP/1.1 Host: ro.wikipedia.org Content-Type: application/x-www-form-urlencoded Content-Length: 24

search=test&go=articol

Serverul răspunde astfel : *HTTP/1.0 302 Moved Temporarily Date: Fri, 13 Jan 2008 15:32:43 GMT* Location: *http://ro.wikipedia.org/wiki/test*

In anumite cazuri sunt **returnate coduri de stare HTTP**. Cele mai importante sunt:

1xx – coduri de stare informaţionale: această clasă a status-ului indică un răspuns provizoriu al serverului şi conţine numai linia de status (de răspuns) sau alte aplicaţii opţionale. Nu sunt aplicaţii necesare pentru acestă clasa de răspuns/status. Aceste status-uri pot fi ignorate.

2xx - coduri de stare care indica un răspuns reuşit: clasa de răspuns/status ce indică utilizatorului că cererea a fost primită, înţeleasă şi acceptată cu succes.

200 - ok: Această cerere a fost executată cu succes. Informaţia a revenit cu un răspuns pozitiv, indiferent de modul în care s-a făcut cererea.

3xx – clasa de coduri de stare care pentru redirectări: această clasă de răspuns/status indică faptul că acţiunile următoare vor trebui făcute de browser pentru a putea fi îndeplinita cererea. Cererea ar putea fi direcţionată de browser fără a interacţiona cu utilizatorul dacă şi numai dacă metoda folosită în cea de a doua cerere este „Primit/recepţionat" sau „Direcţionat/condus".

301 - modificat/mutat permanent:

Cererii i-a fost atribuite o sursă nouă şi permanentă URI iar cererile următoare ar trebui să folosească una din sursele returnate URI. Dacă acest mesaj/cod este primit ca răspuns al unei cereri tip „Primit/recepţionat" sau „Direcţionat/condus", browser-ul nu trebuie să redirectţioneze automat cererea, doar dacă nu poate fi confirmată de către utilizator.

4xx - coduri de stare erori ale utilizatorilor: această clasă de mesaje/statusuri este folosită în cazurile în care utilizatorul ar putea greşi formularea cererii. Excepţia fiind răspunsurule pentru cererile tip „Direcţionat/condus", atunci serverul ar trebui să conţină o intrare cu o explicaţie a situaţiei erorii şi dacă e o eroare temporară sau pemanentă. Aceste răspunsuri sunt aplicabile pentru orice fel de cerere. Browser-ele ar trebui să arate orice intrare cerută de utilizator.

400 - cerere greşită: Cererea nu a putut fi înţeleasă/percepută de către server din cauza unei sintaxe greşite/incomplete. Utilizatorul ar trebui să nu repete cererea fără ca aceasta să suporte modificări.

403 - interzis: Serverul a înţeles cererea, dar refuză să o îndeplinească. Autorizarea nu ajută în nici un caz, iar cererea nu ar mai trebui repetata.

404 - negăsit: Serverul nu a găsit nimic care să corespundă cererii URI. Nu se dau indicaţii despre condiţia temporară sau permanentă.

5xx - erori de server: răspunsurile/status-urile ce încep cu unitatea/cifra 5 indică cazurile în care serverul e conştient de greşelile produse sau e incapabil să execute cererea. Excepţie facând răspunsul unei cereri „Direcţionat/condus", serverul ar trebui, în acest caz sa includă o afişare cu o explicaţie a situaţiei de eroare, fie că e temporara sau permanentă.

500 - eroare internă de server: Server-ul a întâmpinat o condiţie neaşteptată şi o previne spre a putea îndeplini cererea.

505 - versiunea HTTP nu e suportată/susţinută: Serverul nu suportă sau refuză să suporte versiunea de protocol a HTTP ce a fost folosită în formularea cererii. Server-ul sugerează că e incapabil să completeze/termine cererea folosind aceeaşi versiune cu cea a clientului.

Curs 11

Securitatea Reţelelor de Calculatoare

Obiective: *1. Stabilirea aspectelor securităţii în reţele de calculatoare*

2. Înţelegerea tipurilor de atacuri ce pot fi folosite de hackeri pentru a submina securitatea unei reţele

3. Înţelegerea tipurilor de vulnerabilităţi ce pot fi prezente într-o reţea

4. Definirea modelului de securitate în retele de calculatoare si a politicilor de securitate

Cuprins: *1. Introducere*

2. Vulnerabilităţi în reţele de calculatoare

3. Atacuri în reţele de calculatoare

4. Prevenirea atacurilor

1. Introducere

Securitatea reprezintă abilitatea de a evita neplăcerile produse de orice risc, ameninţare sau pericol. În practică, acest lucru este imposibil de realizat.

Incident de securitate este un eveniment apărut in cadrul reţelei, provenind din interiorul ori exteriorul reţelei, cu implicaţii asupra securităţii unui calculator sau a reţelei.

Se poate introduce un limbaj comun pentru descrierea incidentelor:

Principalul obiect de studiu: ***eveniment*** *(incident legat de securitate)*

– consta dintr-o ***acţiune*** *executata asupra unei* ***ţinte***

– acţiunea poate fi executata cu o ***unealta***

– exploatând un anumit tip de ***vulnerabilitate***

– cu un anumit ***rezultat9*** *(in mod normal neautorizat)*

Suprafata de atac:...

Abordarea problemei securităţii datelor intr-o reţea presupune in primul rând *identificarea cerinţelor de funcţionare* pentru acea reţea, apoi *identificarea tuturor ameninţărilor posibile* (împotriva cărora este necesara protecţia). Aceasta analiza consta in principal in trei sub-etape:

- *analiza vulnerabilităţilor* - identificarea elementelor potenţial slabe ale reţelei
- *evaluarea ameninţărilor* - determinarea problemelor care pot apărea datorata elementelor slabe ale reţelei si modurile in care aceste probleme interfera cu cerinţele de funcţionare
- *analiza riscurilor* - posibilele consecinţe pe care problemele le pot crea

Următoarea etapa consta in *definirea politicii de securitate*, ceea ce înseamnă sa se decidă:

- care ameninţări trebuie eliminate si care se pot tolera
- care resurse trebuie protejate si la ce nivel
- cu ce mijloace poate fi implementata securitatea
- care este preţul (financiar, uman, social etc.) masurilor de securitate care poate fi acceptat

O retea LAN are un singur administrator si o singura politica de securitate, in vreme ce retelele WAN au mai multi adminiustratori si po;itici de securitate multiple.

Odată stabilite obiectivele politicii de securitate, următoare etapa consta in *selecţia serviciilor de securitate* - funcţiile individuale care sporesc securitatea reţelei. Fiecare serviciu poate fi implementat prin metode (mecanisme de securitate) variate pentru care sunt necesare aşa-numitele funcţii de gestiune a securităţii. Gestiunea securităţii intr-o reţea consta in controlul si distribuţia informaţiilor către toate sistemele deschise ce compun acea reţea in scopul utilizării serviciilor si mecanismelor de securitate si al raportării evenimentelor de securitate ce pot apărea către administratorii de reţea.

Aspecte ale securităţii in reţele de calculatoare:

– ***Identitatea***: va cuprinde elementele de ***autentificare*** şi ***autorizare*** la nivelul reţelei.

– Autenticitatea: presupune ca două entităţi aflate într-un schimb de mesaje se pot identifica una pe cealaltă. În prima fază, la iniţierea conexiunii, acest serviciu asigură că cele două entităţi sunt autentice. În al doilea rând, autenticitatea presupune că transferul de date dintre cele două entităţi nu este interferat astfel încât o a treia entitate poate să se legitimeze ca fiind una din ele.

– Autorizarea: este abilitatea de a limita şi controla accesul în reţea (la sisteme sau aplicaţii). Pentru a realiza acest serviciu, fiecare entitate care încearcă să aibă acces trebuie mai întâi identificată şi apoi verificate drepturile de acces în sistem.

– ***Integritatea***: este o componentă a securităţii care cuprinde infrastructura de securitate (accesul fizic şi logic) precum şi securizarea perimetrului. Ea se referă la asigurarea consistentei informaţiilor, încrederea in date sau resurse (în cazul transmiterii unui mesaj prin reţea, integritatea se referă la protecţia împotriva unor tentative de falsificare a mesajului);

– ***Confidenţialitatea***: asigura faptul că transmisiile de date de-a lungul reţelei au caracter privat. Există numeroase posibilităţi pentru confidenţialitatea informaţiilor de la protecţie fizică până la algoritmi matematici.

– ***Disponibilitatea***: va asigura faptul că toate resursele reţelei sunt disponibile personalului sau proceselor autorizate.

– ***Nerepudierea***: măsură prin care se asigură faptul că, după emiterea/recepţionarea unei informaţii într-un sistem de comunicaţii securizat, expeditorul/destinatarul nu poate nega, în mod fals, că a expediat/primit informaţii *Se previne ca nici o entitate să nu refuze să recunoască un serviciu executat. Când un mesaj este trimis, destinatarul poate demonstra că mesajul primit este cel trimis de emiţător. Similar, când un mesaj este primit, emiţătorul poate demonstra că mesajul primit este cel primit de destinatar.*

– ***Auditul***: este necesar pentru monitorizarea şi verificarea securităţii la nivelul firmei

2. Vulnerabilităţi in reţele de calculatoare

Vulnerabilitatea este o slăbiciune a unui sistem hard/soft ce permite utilizatorilor neautorizaţi sa aibă acces asupra sa. Nici un sistem nu este integral sigur. De multe ori vulnerabilităţile apar/sunt facilitate si datorită proastei administrări.

Tipuri de vulnerabilităţi: – permiterea refuzului serviciilor (DOS – Denial Of Service); – permiterea utilizatorilor locali cu privilegii limitate sa-si mărească aceste privilegii fara autorizaţie;

– permiterea utilizatorilor externi sa acceseze reţeaua/sistemul local in mod neautorizat;

Cauzele posibile ale existentei vulnerabilităţilor (*90% sunt datorate utilizatorilor)*: – Erori existente in programe introduse deseori neintenţionat – Ignorarea/nedocumentarea erorilor existente – Configurarea necorespunzătoare a programelor, serverelor si reţelelor – Lipsa suportului din partea producătorilor (e.g. rezolvarea greoaie a erorilor) – Comoditatea sau necunoaşterea problemelor de securitate de către administrator ori de conducerea organizaţiei.

3. Atacuri in reţele de calculatoare

Atac este un eveniment potenţial distrugător provocat intenţionat de persoane răuvoitoare.

Atribute ce trebuie considerate pentru a estima reuşita unui atac (cunoaşterea profilului atacatorului): – Resursele disponibile (financiare, tehnice,… + pregătirea in domeniu) – Timpul alocat (atacatorii răbdători vor avea mai mult succes) – Riscul asumat – depinde de obiective (atacul ar putea fi revendicat sau nu de cracker) – Accesul la Internet si calitatea acestuia: tip (dial-up, conexiune satelit,…), mod de alocare a adreselor IP etc. – Obiectivele urmărite (recunoaştere mondiala, denigrarea ţintei, furt de informaţii sau bani etc.)

Niveluri de atac – ***Oportunist****:* Atacul are un scop "recreational" si nu are obiective/ţinte clar definite. Se utilizează programe disponibile liber pentru a scana sau testa vulnerabilitati uzuale. Nu necesita acces in interiorul sistemului. Cunoştinţe vagi despre sistemul/organizaţia ţinta • Masuri de precauţie: – ziduri de protecţie (firewall-uri) – actualizarea versiunilor de programe

– ***Intermediar*** Obiectiv este conturat, la nivelul organizaţiei. Se efectuează aceleaşi acţiuni ca la atacul “recreational”, dar se încearcă ascunderea lor. Atacatorul are mai multa răbdare decât in cazul unui atac oportunist . Cunoştinţe tehnice mai profunde. Probabilitate mai mare de succes, posibil efecte mai puternice

– ***Sofisticat:***• Obiectiv foarte bine conturat. Ţinta este de cele mai multe ori o organizaţie. Atacurile pot trece peste masurile de prevedere •Atacatorul va avea multa răbdare. Se investeşte timp pentru adunarea de informaţii despre sistemul/organizaţia ţinta • Necesita foarte bune abilitaţi tehnice si are o probabilitate mare de succes.

Tipuri de atac ***–Accesul utilizator*** •Atac prin acces via utilizator obişnuit sau cu privilegii superioare • Etape: – Colectarea de informaţii – utilizatori, vulnerabilităţi,… – Exploatarea – Deteriorarea » Modificare de informaţii »Acces la date importante »Asigurarea accesului ulterior la sistem » Modificarea jurnalelor de sistem

–Accesul de la distanta la servicii de retea • Nu necesita acces utilizator la sistem • Creaza refuzuri de servicii prin cereri incorecte, eventual cu “caderea” serviciilor prost proiectate • Etape: – Colectarea de informatii – identificarea de servicii – Exploatarea – trimiterea de pachete la portul gasit – Deteriorarea » Distrugerea unui serviciu de retea » Defectarea/incetinirea (temporara) a unui serviciu sau a sistemului

–Accesul de la distanta la diverse aplicatii •Trimitere de date invalide aplicatiilor, nu serviciilor de retea (traficul nu e afectat) • Nu necesita obtinerea de acces utilizator • Etape: – Colectarea de informatii – identificarea aplicatiei (e.g. server sau client Web, aplicatie gen MS Office) – Exploatarea – trimiterea conţinutului, direct sau indirect (i.e. via e-mail), spre aplicaţie – Deteriorarea »Stergerea/copierea fisierelor utilizatorilor » Modificarea fisierelor de configuratie

Ţinta atacurilor in reţele de calculatoare

– Organizaţii publice sau guvernamentale • Recunoaştere in rândul cracker-ilor • Captarea atenţiei mass-media • Revendicări etice, politice,… – Furnizori de servicii Internet • Sabotarea activităţii – Companii private • Discreditare • Furt de informaţii • Răzbunare din partea foştilor angajaţi – Persoane fizice • Cu scop recreaţional

First Mac Botnet Activated, Engages in DDoS Attacks

posted by Thom Holwerda on Sat 18th Apr 2009 09:27 UTC

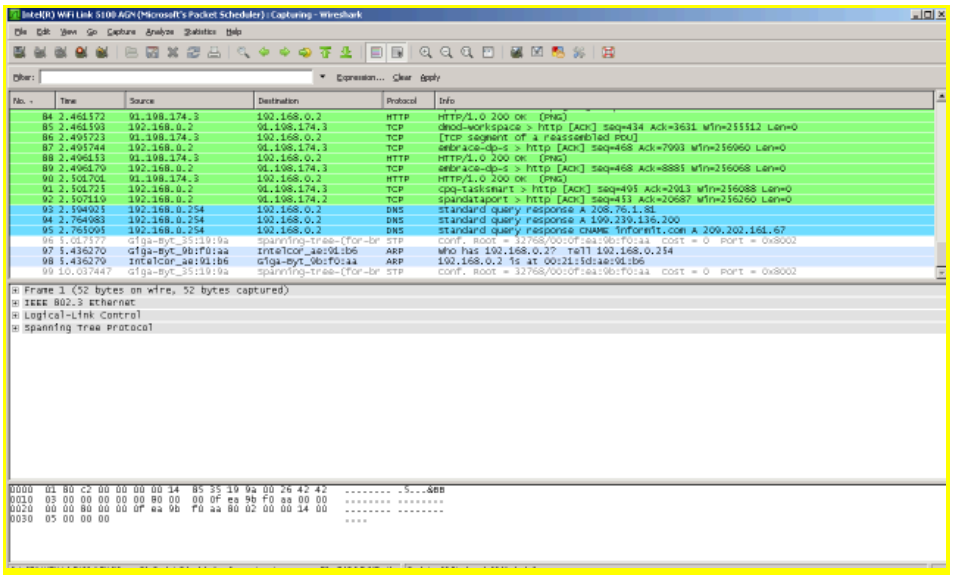
Remember the Mac trojan that we reported about earlier this year? A trojan was found piggybacking on the back of copies of iWork and Photoshop CS4 found on warez sites and networks, and it would install itself after the user had entered his or her administrator password during the software's installation. This trojan didn't seem like much of a threat back then, but as it turns out, it's now in use in the first Macintosh botnet.

Security researchers from Symantec have found evidence that said trojans, OSX.Iservice and OSX.Iservice.B, are being used in creating a botnet used for DDoS attacks. There's at least one documented case of these trojans being used for DDoS attacks, and the researchers have found out that the botnet has encryption, a peer-to-peer engine, and remote startup capabilities.

Moduri de atac *– Bomba e-mail (e-mail bombing)* •Trimiterea repetata a unui mesaj (de dimensiuni mari) spre o adresa e-mail a unui utilizator • Încetineşte traficul, umple discul • Unele atacuri pot folosi adrese e-mail multiple existente pe serverul ţinta • Se poate combina cu falsificarea adresei (e-mail spoofing) *– Spam (e-mail spamming)* •Trimiterea de mesaje nesolicitate (reclame) •Adresa expeditorului e falsa • Efectul atacului e accentuat daca mesajul este trimis pe o lista de discuţii *–Abonarea la liste de discuţii* •“Atac” ce determina enervarea victimei, facilitat de diverse programe disponibile in Internet • Cauzează trafic inutil de reţea *– Falsificarea adresei expeditorului (e-mail spoofing)* • Folosita pentru ascunderea

identit. expeditorului sau pentru determinarea utiliz. sa răspundă la atac ori sa divulge informaţii (e.g. parole) • Slăbiciunea e datorata modului in care operează protocolului SMTP • Utilizatorii trebuie educaţi sa nu răspundă expeditorilor necunoscuţi si sa nu divulge informatii confidentiale – *Refuzul serviciilor (Denial Of Service)* • Degradează calitatea funcţionarii unor servicii sau conduce la dezafectarea lor • Bombardament cu pachete (packet flood) – se trimite un număr mare de pachete spre o anumita gazda de la o singura sursa ori provenind de la surse multiple (***Distributed DOS***) – Segmente TCP (cu setarea SYN, ACK sau RST) – Pachete ICMP (ping flood) – Pachete UDP • Se poate falsifica adresa IP a expeditorului (IP spoofing) • Se pot modifica porturile sursa/destinaţie (pentru a trece de firewall-uri) • Exemple – SYN flood – cereri multiple de realizare de conexiuni – Ping of death – atac cu pachete ICMP mari –Teardrop – exploatarea implementărilor TCP/IP care nu gestionează corect pachetele IP – Smurf – atac ICMP asupra adresei de broadcast – *Depasirea capacitatii bufferelor (buffer overflow)* • Unele programe pot aloca spatiu insuficient pentru unele date, depasirile survenite pot produce executarea de comenzi ca root • Uzual, atacul provine din interior, dar poate fi si din exterior (via un cal troian) – *Interceptarea reţelei (IP sniffing)* • Monitorizarea datelor care circula printr-o interfaţa de reţea

– se pot detecta parole transmise necriptate •Atacul provine din interior • Pentru reţele de viteza mare (100 M/s) unele pachete nu pot fi captate de sniffer • Soft-ul de interceptare trebuie supravegheat • Exemple: tcpdump, Wireshark



– *Cai troieni (trojan horses)* • Programe rău intenţionate, “deghizate” sub forma unor executabile utile •Apelează programe neautorizate sau sunt modificate, incluzând cod nelegitim •Actiuni: colectarea de informaţii, distrugerea de informaţii, lansarea de atacuri spre alte sisteme • Exemple: sendmail sau “vaduva neagra” (blocheaza sau corupe browsere Web) – *Usi ascunse (back doors / traps)* • Caz particular de cai troieni • Creaza o “poarta” (e.g. utilizator, port,…) care permite accesul ulterior la calculator si/sau câştigarea de privilegii – *Viermi (worms)* • Programe care se multiplica, transferându-se pe alte calculatoare si efectuând distrugeri. Exemplu: Internet Worm (1988) – *Ghicirea parolelor (password guessing)* • Folosirea unui program ce determina parolele prost alese (prea simple) – Prea scurte, utilizează cuvinte de dicţionar, numerice • Protectie prin /etc/shadow, reguli stricte de schimbare a parolelor, educarea utilizatorilor, folosirea de programe de tip spărgător de parole (password cracker) – *Virusi* • Programe ce efectuează operatii nedorite (distructive), cu capacitati de “multiplicare”. Ele realizează infectarea altor programe (uzual, executabile) • Mai puţin răspândiţi in Unix/Linux, de obicei

având efect doar daca se executa sub auspicii de root • Pot genera si e-mail bombing • Remedii: utilizarea de antiviruşi si porţi de e-mail

4. Prevenirea atacurilor

Modelul de securitate pentru un sistem (un calculator sau o retea de calculatoare) poate fi vazut ca având mai multe straturi ce reprezintă nivelurile de securitate ce inconjoara subiectul ce trebuie protejat. Fiecare nivel izoleaza subiectul si il face mai dificil de accesat in alt mod decat cel in care a fost prevazut.

1. *Securitatea fizica* reprezinta nivelul exterior al modelului de securitate si consta, in general, in incuierea echipamentelor informatice intr-un birou sau intr-o alta incinta precum si asigurarea pazei si a controlului accesului. Aceasta securitate fizica merita o consideratie speciala. Una dintre problemele mari o constituie salvarile sub forma de copii de rezerva (backup) ale datelor si programelor, precum si siguranta pastrarii suportilor de salvare. Retelele locale sunt, in acest caz, de mare ajutor, copiile de rezerva putandu-se face prin retea pe o singura masina ce poate fi mai usor securizata. O alta problema importanta in securitatea unui sistem informatic o constituie pur si simplu sustragerile de echipamente. In plus, celelalte masuri de securitate (parole etc.) devin nesemnificative in cazul accesului fizic neautorizat la echipamente.
2. *Securitatea logica* consta din acele metode logice (software) care asigura controlul accesului la resursele si serviciile sistemului. Ea are, la randul ei, mai multe niveluri impartite in doua grupe mari : *niveluri de securitate a accesului* si *niveluri de securitate a serviciilor*.
    - *Securitatea accesului* cuprinde:
        - accesul la sistem, care este raspunzator de a determina daca si cand este reteaua accesibila utilizatorilor si in ce conditii. El poate fi raspunzator de asemenea si de gestionarea evidentei accesului. Accesul la sistem poate efectua si deconectarea fortata in anumite cazuri (ex. expirarea contului, ora de varf, ...)
        - accesul la cont care verifica daca utilizatorul ce incearca sa se conecteze are un nume si o parola valida.
        - drepturile de acces (la fisiere, resurse, servicii etc.) care determina de ce privilegii dispune un utilizator (sau un grup de utilizatori) dat.
    - *Securitatea serviciilor* (care se afla "sub" securitatea accesului) controleaza accesul la serviciile unui sistem (masina, retea). Din acest nivel fac parte:
        - *controlul serviciilor* care este responsabil cu functiile de avertizare si de raportare a starii serviciilor, precum si de activarea si dezactivarea diverselor servicii oferite de catre sistemul respectiv
        - *drepturile la servicii* care determina exact cum foloseste un anumit cont un serviciu dat (acces la fisiere, resurse, prioritate,...)

Elaborarea de politici de securitate implica: – Planificarea cerinţelor de securitate • Confidenţialitate, integritate, disponibilitate, control – Evidenţierea riscurilor – Analiza raportului cost-beneficii • Costurile prevenirii, refacerii după dezastru etc. – Stabilirea politicilor de securitate • Politica generala (naţionala, organizaţionala,…) • Politici separate pentru diverse domenii protejate • Standarde & reglementari (recomandări) • Masurile luate pot fi tehnice si non-tehnice

Elaborarea de politici de securitate – exemplu:

– Gestionarea accesului (nume de cont, alegerea si modul de schimbare a parolelor, blocarea terminalului, politica de acces din exterior,…)
– Clasificarea utilizatorilor (grupuri, permisiuni, utilizatori speciali, utilizatori administratori
– Accesul la resurse (drepturi de acces la fişiere, directoare, criptarea fişierelor importante,…)
– Monitorizarea activităţii (fişiere de jurnalizare)
– Administrarea copiilor de siguranţa (tipuri de salvări, medii de stocare, durata păstrării,…)

Supravieţuirea reprezintă capacitatea unui sistem (calculator/reţea) de a-si îndeplini misiunea, in timp util, in prezenta *atacurilor*, *defectelor* sau *accidentelor*

• Atac =eveniment potenţial distrugător provocat intenţionat de persoane răuvoitoare • Defect =eveniment potenţial distrugător cauzat de deficiente ale sistemului sau ale unui factor de care depinde sistemul (e.g. defecte hard, bug-uri soft, erori ale utilizatorilor) • Accident =evenimente aleatoare (neprevăzute); exemple: dezastre naturale, căderi de tensiune

• Sistemul trebuie sa-si duca pana la capăt misiunea chiar daca unele componente sau parţi din sistem sunt afectate ori scoase din uz • Sistemul trebuie sa susţină măcar îndeplinirea funcţiilor vitale – Identificarea serviciilor esenţiale • Proprietati ale sistemului: – Rezistenta la atacuri – Recunoaşterea atacurilor si efectelor lor – Adaptarea la atacuri • Instrumente sub Linux (Unix): – Utilitare de reţea: ping, traceroute, netstat, ifconfig, route, host, finger, telnet – Scanere de porturi: NMAP – Interceptoare de retea: tcpdump, wireshark – Testarea securităţii locale: /etc/shadow, Crack, Titan – Verificări asupra sist. de fisiere: tripwire, showmount – Salvari de siguranta: tar, dump, amanda – Verificarea daemonilor: chkconfig – Protecţia TCP/IP: iptables (firewall), activarea mecanismului SYN cookies in nucleu

2.2. Firewall

2.2.1. Tipuri de firewall

În ultimii ani, Internet-ul s-a umplut de viruşi, troieni, aplicaţii maliţioase, exploit-uri, site-uri capcană etc. E din ce în ce mai dificil pentru utilizator să se protejeze şi să scape de aceste probleme, precum şi de implicaţiile lor.

Firewall-ul poate împiedica persoanele străine să intre pe computerul nostru prin Internet. Un firewall poate lua două forme, software sau hardware, şi oferă o izolare protectoare care ajută la ţinerea deoparte a "invadatorilor" din Internet.

Înainte de a construi un firewall trebuie hotărâtă politica sa (permiterea sau blocarea de mesaje care nu respecta setul sau de reguli predefinite), pentru a şti exact care va fi funcţia sa şi în ce fel se va implementa această funcţie.

Politica firewall-ului se poate alege urmând câţiva paşi simpli:

- se aleg întâi serviciile care trebuie oferite de firewall

- se desemnează grupurile de utilizatori care vor fi protejaţi

- se defineşte amănunţit gradul de protecţie de care are nevoie fiecare grup de utilizatori şi cum vor fi implementate protecţiile necesare

- se face cunoscut utilizatorilor că oricare alte forme de acces nu sunt permise

Politicile definite la un moment dat tind să se complice cu timpul, dar la început este bine ca ele să fie simple şi la obiect.

Principala funcţie a unui firewall este de a verifica traficul intre calculatoarele din reţele cu diferite nivele de securitate. Exemplul tipic este Internetul, care este o zona fara siguranţa si reţeaua locala, care este o zona de siguranţa sporita. O zona de siguranţa intermediara, care se situează intre Internet si reţeaua locala este DMZ (DeMilitarized Zone) – zona in care anumite servicii sunt accesibile direct la nivelul internetului in vreme ce altele sunt accesibile doar la nivel local.

Funcţia firewall-ului in retea este atât pentru a preveni accesului intruşilor intr-o reţea privata dar si pentru a intarzia răspândirea aplicaţiilor si serviciilor către celelalte reţele.

Un firewall poate să:

- monitorizeze căile de pătrundere în reţeaua privată, permiţând în felul acesta o monitorizare mai bună a traficului şi deci o detectare mai uşoară a încercărilor de infiltrare;
- blocheze la un moment dat traficul spre şi dinspre Internet;
- selecteze accesul în spaţiul privat pe baza informaţiilor conţinute în pachetele de date;
- permită sau interzică accesul la reţeaua publică, de pe anumite staţii de lucru specificate;
- şi, la fel de important, poate izola spaţiul privat de cel public, realizând interfaţa între cele două.

Pe de altă parte, o aplicaţie firewall nu poate:

- interzice importul/exportul de informaţii dăunătoare vehiculate ca urmare a acţiunii răutăcioase a unor utilizatori aparţinând spaţiului privat (ex: căsuţa poştală şi ataşamentele);
- interzice scurgerea de informaţii pe alte căi care ocolesc firewall-ul (acces prin dial-up ce nu trece prin router);
- apăra reţeaua privată de utilizatorii ce folosesc sisteme fizice mobile de introducere a datelor în reţea (USB Stick, dischetă, CD, etc.)
- preveni manifestarea erorilor de proiectare ale aplicaţiilor ce realizează diverse servicii, precum şi punctele slabe ce decurg din exploatarea acestor greşeli.

De aceea, pentru o protecţie maximă împotriva pericolelor din Internet, pe lângă un firewall mai este nevoie şi de alte componente de pază. Fara configurare adecvata, un firewall poate deveni fara valoare. Practicile standard ale securităţii dictează un set de reguli pentru permisia/interzicerea traficului prin firewall (in mod implicit decizia este de blocare a traficului), in care conexiunile permise sunt doar cele care au fost declarate in mod explicit. Din păcate, o astfel de configurare necesita înţelegerea detaliata a aplicaţiilor de reţea si a endpoint-urilor necesare lucrului zilnic. In multe rețele, aceasta înţelegere detaliata lipsește si de aceea este implementat un altfel de set de reguli (default-allow), in care traficul este permis doar daca nu exista o regula clara pentru a-l bloca. Aceste configuraţii duc la nepotrivirea conexiunii de reţea si a sistemului.

Tehnologia firewall s-a dezvoltat spre sfârşitul anilor 1980, când Internetul era o tehnologie destul de noua privind folosirea lui globala si a conectivităţii. Predecesorii firewall-ului pentru securitatea in reţea au fost routerele folosite pentru a separa reţelele intre ele. Părerile despre Internet ca o comunitate de useri compatibili care doreau colaborarea si schimbul de fișiere au fost eliminate de o serie de breșe in securitate, aceste brese apărând spre sfârșitul anilor 1980. În funcţie de modul de implementare firewall-urile se pot împărţi grob în două mari categorii:

- dedicate, în care dispozitivul care rulează software-ul de filtrare este dedicat acestei operaţiuni şi este practic "inserat" în reţea (de obicei chiar după router). Are avantajul unei securităţi sporite.

- combinate cu alte facilităţi de networking. De exemplu, routerul poate funcţiona în acelaşi timp şi pe post de firewall, iar în cazul reţelelor mici acelaşi calculator poate juca în acelaţi timp mai multe roluri: de firewall, router, file server, print server ş.a.

Fig. Echipament hardware Firewall oferit de firma Cisco PIX-501

Firewall-urile pot fi clasificate după:

- layerul (stratul) din stiva de reţea la care operează
- modul de implementare

Astfel, în funcţie de layerul din stiva TCP/IP la care operează, firewall-urile pot fi:

● Layer 2 (MAC) şi 3 (datagram): packet filtering.

● Layer 4 (transport): tot packet filtering, dar se poate diferenţia între protocoalele de transport şi există opţiunea de "stateful firewall", în care sistemul ştie în orice moment care sunt principalele caracteristici ale următorului pachet aşteptat, evitând astfel o întreagă clasă de atacuri

● Layer 5 (application): application level firewall (există mai multe denumiri). În general se comportă ca un server proxy pentru diferite protocoale, analizând şi luând decizii pe baza cunoştinţelor despre aplicaţii şi a conţinutului conexiunilor. De exemplu, un server SMTP cu antivirus poate fi considerat drept application firewall pentru e-mail.

Prima generaţie de firewall – filtrare de pachete

Digital Equipment Corporation (DEC) a fost cea care a lucrat in domeniul filtrelor de pachete in 1988. Sistemul a fost prima generaţie a ceea ce va deveni o caracteristica foarte evoluata a securităţii pe internet. La AT&T Bell Labs, Bill Cheswick and Steve Bellovin au continuat dezvoltarea acestor filtre împachetate si au produs un model pentru compania lor pornind de la modelul original.

Filtre controlează pachetele, care reprezintă unitatea de bază de transfer de date între computere de pe Internet pentru nivelul reţea. Dacă un pachet se potriveşte setului de reguli de filtrare de pachete, filtrul de pachete il va elimina (eliminare in linişte), sau il va respinge (renunţa la el si trimite „eroare de răspuns” la sursă).

Acest tip de pachete de filtrare nu plăteşte nici o atenţie dacă un pachet este parte a unui flux de trafic existent (acesta nu stochează informaţii pe conexiune « de stare »). În schimb, filtrează pentru fiecare doar prin informaţiile conţinute în pachetul în sine (cel mai frecvent, folosind o combinaţie a pachetului sursă şi destinaţie, adresa, protocolul său, şi, pentru trafic TCP şi UDP, care cuprinde cele mai de internet de comunicare, numărul de port).

Pentru ca traficul TCP şi UDP prin convenţie se utilizează porturile bine cunoscute pentru anumite tipuri de trafic, un filtru de pachete fără analiza stării le poate distinge între, şi, astfel, si controla aceste tipuri de trafic (cum ar fi navigarea web, la distanţă de imprimare, transmitere de poştă electronică, transfer de fişiere), cu excepţia cazului în care maşinile de pe fiecare parte a filtrului de pachete folosesc aceeaşi porturile non-standard.

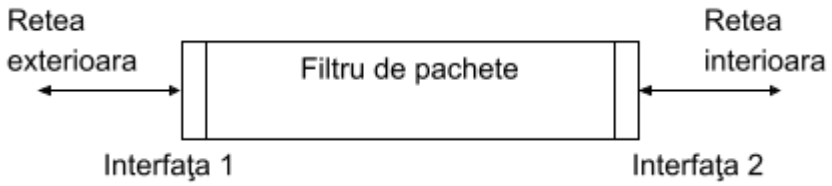


Fig. Structura unui filtru de pachete

A doua generaţie – filtrarea stării

Înainte de aparitia firewall-urilor cu analiza starii, un sistem firewall trata fiecare interval de reţea (sau pachet), în mod izolat. Un astfel de firewall nu are nici o posibilitate de a cunoaşte dacă o anumită parte a pachetului este o conexiune existentă, daca încearcă să stabilească o nouă conexiune, sau este doar un pachet nedorit. Firewall-urile moderne sunt cunosc starea conexiunilor, oferind administratorilor de reţea un control mai fin al traficului de reţea.

Intre 1980-1990 s-a dezvoltat cea de-a doua generaţie de sisteme firewall, numite firewall-uri cu filtrarea stării. Aceasta tehnologie menţine o evidenţă a tuturor conexiunilor care trec printr-un sistem firewall având posibilitatea de a determina dacă un pachet este fie un început a unei noi conexiuni, o parte dintr-o conexiune deja existenta, sau este un pachet invalid . Deşi nu există încă un set de reguli statice într-un astfel de firewall, starea de o conexiune în sine poate fi unul dintre criteriile care pune în mişcare a unor reguli specifice.

Acest tip de sistem firewall poate ajuta la prevenirea atacurilor care exploatează conexiunile existente, sau atacurile numite Denial-of-Service (care se bazează pe transmiterea a numeroase pachete sau deschiderea de conexiuni, fara a continua traficul prin legăturile deschise) .

În calculatoare, un firewall **cu filtrarea stării conexiunii** (orice firewall care efectuează inspecţia de pachete (SPI –Stateful Packet Inspection)) este un sistem firewall ce monitorizează starea conexiunilor de reţea (cum ar fi fluxurile de TCP, UDP de comunicare) care il traversează. Doar pachetele care se potrivesc cu o conexiune cunoscuta vor fi permise de firewall; celelalte vor fi respinse.

Primele încercări in producţia de sisteme firewall au operat la nivelul Aplicaţie al modelului OSI, dar acest lucru e cerut viteza procesorului sporita. Fitre de pachete funcţionează la nivelul de reţea (layer-3) mai eficient, deoarece ele verifica doar antetul dintr-un pachet. Cu toate acestea, fitrele de pachete nu au nici un concept de stare, astfel cum

sunt definite de stiinta calculatoarelor, folosind termenul Automat finit şi sunt supuse unor atacuri de falsificare şi exploatare.

Modul de operare al unui firewall cu inspecţia stării conexiunilor este ilustrata in figura următoare:

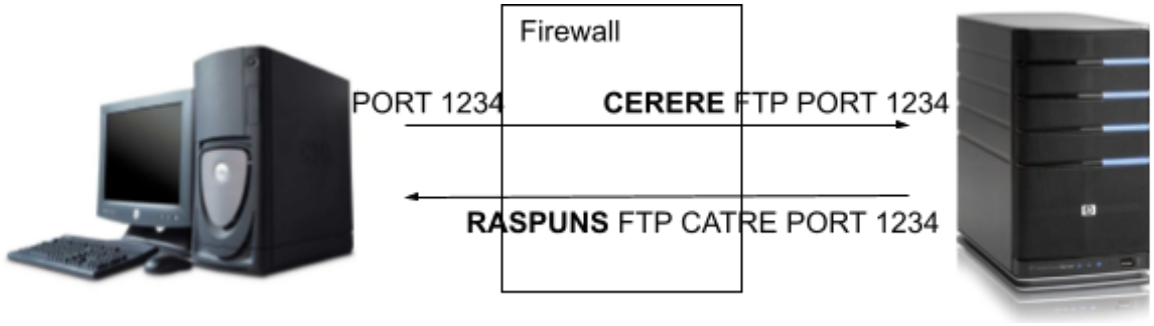


Fig. Firewall-ul recunoaşte ca iniţializarea cererii s-a efectuat din interior si accepta răspunsul din exterior

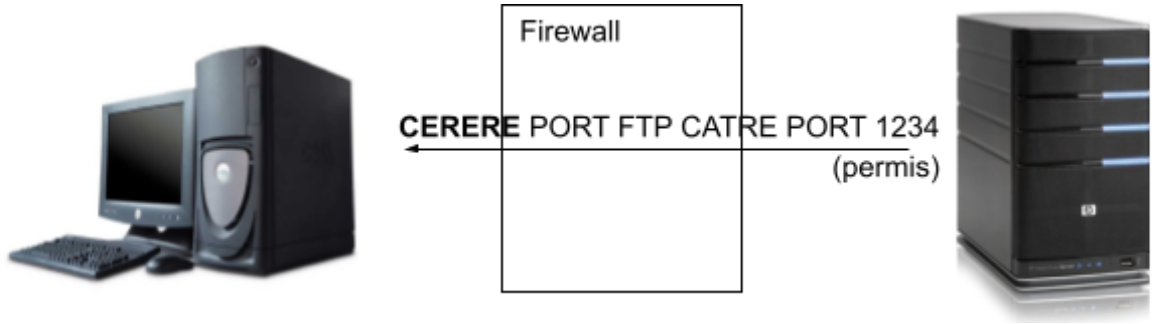


Fig. Firewall-ul primeşte din exterior o iniţializare de conexiune către un port permis (1234), pe care o poate accepta

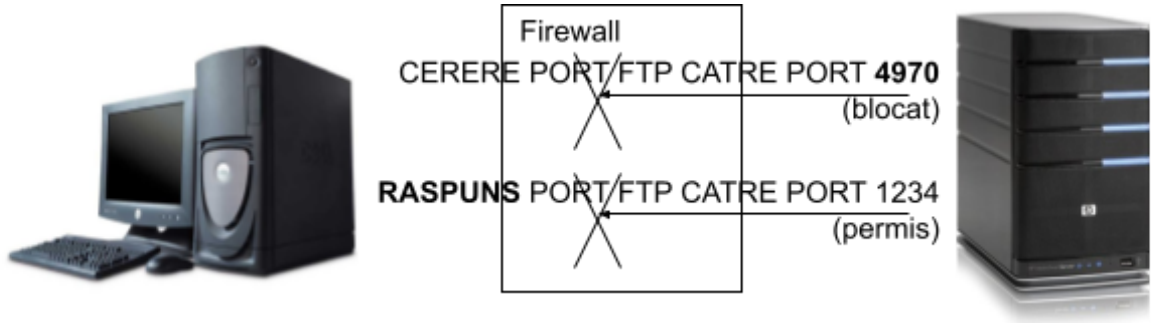


Fig. Firewall-ul cu inspecţia stării conexiunii primeşte din exterior o iniţializare de conexiune către un port necunoscut, de la care nu a pornit nici o conexiune, deci o va respinge

Exemplul clasic este File Transfer Protocol, pentru că prin design deschide noi conexiuni pe porturi arbitrare. FTP, printre alte protocoale, trebuie să fie capabil sa deschidă conexiuni pe porturi arbitrare pentru a funcţiona corect. Deoarece un firewall nu are nici un

fel de a ştii că pachetul este destinat unei retele protejate, la portul 4970 al gazdei , este o parte dintr-o sesiune de FTP reala, va refuze pachetul. Un firewall cu inspecţia stării conexiunilor firewall rezolva această problemă prin menţinerea unui tabel de conexiuni deschise şi asocierea inteligenta a noilor cereri de conexiune cu conexiuni legitime existente.

Un sistem firewall cu inspecţia stării conexiunilor este în măsură să salveze în memorie atributele semnificative din fiecare conexiune, de la început până la sfârşit. Aceste atribute, care sunt cunoscute sub numele colectiv de starea de conexiune, pot include adresele IP, porturile implicate în conexiune şi secvenţa de numere de pachete care traversează conexiunea. Cele mai intense controale sunt efectuate în momentul de configurare a conexiunii. Toate pachetele după configurare sunt procesate rapid, pentru că este simplu şi rapid de a stabili dacă acesta aparţine unei sesiuni existente,deja verificata. După ce a luat sfârşit , la intrarea sesiunii în tabelul de stat este strearsa.

Firewall-ul stateful depinde de celebrul three-way handshake al protocolului TCP. Când un client initiaza o noua conexiune, se transmite un pachet SYN de biti setati în antetul pachetului. Toate pachetele cu bitii SYN setati , sunt considerate de firewall-ul conexiuni noi. Dacă serviciul care clientul l-a solicitat este disponibil pe server, serviciul va raspunde la pachetul SYN, cu un pachet în care ambele pachete SYN si ACK sunt stabilite. Clientul va răspunde cu un pachet în care doar de biţii ACK sunt setati şi conexiunea va intra in starea ESTABLISHED. Un astfel de firewall va permite trecerea tuturor pachetelor dar va accepta doar pachetele care sunt parte din conexiunea stabilita, asigurându-se că hackerilor nu pot porni conexiunile nesolicitate cu serverul protejat.

-+Pentru a împiedica umplerea stivei, o sesiune va expira daca nu s-a mai efectuat un transfer de data pentru a anumita perioada de timp. Aceste conexiuni vor fi înlăturare din stiva de conexiuni. Din aceasta cauza multe aplicatii trimit mesaje de “keepalive” in mod periodic pentru a impiedica firewall-ul sa întrerupă sesiunea datorita netransmiterii mesajelor. De menţionat ca cel mai răspândit mod de atac “Denial of service” pe internet in zilele noastre este “SYN flood” in care un utilizator nepermis transmite pachete mari de informaţii la un server pentru a-I bloca stiva de conexiuni împiedicându-l sa accepte conexiuni noi.

Multe firewall-uri “stateful” sunt capabile sa detecteze aceaste date venite pe conexiuni fara protocoale, ca de exemplu UDP. Aceste conexiuni primesc imediat starea ESTABLISHED dupa ce primul pachet a fost tirmis de firewall. Sesiunile cu conexiuni fara protocoale pot fi terminate doar prin time-out.

Prin mentinere rutei conexiunii, forewall-urile “stateful” asigura o eficienta sporita in inspectia pachetelor. Acest lucru de datoreaza faptului ca pentru o conexiune existenta, firewall-ul trebuie sa verifice doar stiva, in loc de a verifica setul de reguli ale sale împotriva

pachetului. Verificarea pachetului prin setul de reguli va genera un cost suplimentar mai ales daca setul de reguli al firewall-ului s-a modificat (stiva de conexiuni va fi stearsa). De asemenea conceptul de “deep packet inspection” nu este legat de cel de firewall “stateful”, pachetele sunt inspectate intr-un mod asemănător cu cel al firewall-urilor “application layer”.

Filtre la nivel aplicaţie

Cu toate acestea, filtrele de pachete nu sunt considerate destul de sigure. Pentru a bloca efectiv traficul de retea peer-to-peer, este nevoie de un firewall care efectueaza filtrare pentru protocoale de nivel aplicaţie (de exemplu FTP – File Transfer Protocol, DNS sau HTTP). Aceasta filtrare este o extensie a verificării pachetelor efectuat de firewall-urile “stateful”. Firewall-urile “stateful” pot determina ce tip de protocol este trimis pe fiecare port, dar filtrele de aplicaţie verifica pentru ce este folosit un protocol. De exemplu, un astfel de filtru poate detecta diferenţa dintre traficul HTTP folosit pentru a deschide o pagina Web si cel folosit pe schimbul de fisiere, in timp de un firewall “stateful” va considera toate traficele HTTP egale.

Firewall-urile la nivelul aplicaţie diferă de celelalte firewall-uri prin mai multe caracteristici. Ele suporta aplicaţii multiple pe un singur firewall. Serverul intermediar este situat intre client si server si transmite date catre cele 2 endpoint-uri. Datele care genereaza suspiciuni sunt eliminate, iar clientul si serverul nu comunica niciodată intre ele direct. Din cauza ca firewall-urile la nivelul aplicaţie sunt mai sigure, pot lucra cu uşurinţa cu protocoale complexe, ca de exmplu H.323 (folosit pentru videoconferinta) si VoIP (trafic de voce intre IP). De asemenea pot fi transparente pentru client si server, in acest caz nu este necesare nici o configurare din partea celor 2 endpoint-uri, sau netransparente, lăsând serverul sau clientul sa acceseze direct serverul intermediar. Transparenta sau netransparenta este mai degraba o problema de implementare si de adresare ascunsa decât de securitate.

Ultimul sistem de operare oferit de Microsoft, Microsoft Vista, foloseşte scalarea ferestrei TCP pentru conexiuni non-HTTP. Acest mecanism este folosit si de kernelul de Linux superior versiunii 2.6.8. Aceasta caracteristica este incompatibila cu unele firewall-uri care folosesc SPI (Inspectie de Pachete “Stateful”) precum Checkpoint NG R55, Cisco PIX IOS versiunea sub 6.3.1, NetApp Cache Appliances, SonicWall, D-Link DI-724U, Netgear WGR614 si Linksys WRT54GS. Versiunea Windows Vista are deja multe probleme, incluzând cele legate de neconectarea la http prin firewall-uri SPI.

Curs 12
Criptografia

Multe dintre serviciile Internet transmit informaţii importante - cum ar fi nume de utilizatori şi parole - fără a le proteja, în acest sens, *stiva de protocoale TCP/IP* a fost concepută să fie transparentă, oricine având acces la pachetele de date. O persoană rău intenţionată poate monitoriza traficul din reţea şi depista cu uşurinţă aceste informaţii.
Criptarea informaţiilor este soluţia cea mai des utilizată pentru prevenirea acestor atacuri.
Criptografia permite comunicaţia sigură între părţi. Aceasta înseamnă atât siguranţa că un mesaj poate fi citit doar de către persoana căreia îi este destinat, cât şi siguranţa că mesajul nu a fost modificat pe parcurs.
Procesul de transformare a informaţiilor din forma iniţială într-o formă imposibil de citit, fără a avea câteva cunoştinţe suplimentare (o cheie), se numeşte ***criptare***. *Decriptarea este operaţiunea inversă, adică transformarea informaţiei criptate în forma sa iniţială.*
Criptarea, respectiv decriptarea necesită în general utilizarea unei informaţii secrete, numită ***cheie***. Informaţiile originale sunt criptate utilizând o cheie de criptare, sunt transmise, iar la destinaţie sunt decriptate folosind tot o cheie, identică sau diferită de prima cheie.
Criptografia (de la cuvântul grec kryptos logos, adică cuvânt ascuns, secret) este ştiinţa care se ocupă de protecţia informaţiilor prin codificarea acestora. ***Criptanaliza*** studiază metodele de determinare a informaţiilor originale sau a cheilor de criptare.
Criptografia a apărut ca ştiinţă acum mii de ani, fiind utilizată de-a lungul timpului îndeosebi pentru ascunderea secretelor militare. Există două tipuri de ***sisteme de criptare***:
- *criptarea cu cheie secretă* (sau *criptografia simetrică*) foloseşte aceeaşi cheie atât la criptarea, cât şi la decriptarea informaţiilor. Dintre algoritmii de criptare de acest tip putem enumera **DES** şi **AES**.
- *criptarea cu chei publice* (sau *criptografia asimetrică*) foloseşte chei distincte pentru criptare, respectiv decriptare, cheile fiind dependente una de alta. Prima cheie, numită *cheia privată*, este ţinută secretă şi este cunoscută doar de proprietarul ei. A doua cheie, numită *cheia publică*, este cunoscută şi de expeditor, şi de destinatar. Informaţiile pot fi criptate de către orice persoană care posedă cheia publică, dar vor putea fi decriptate doar de către persoana care cunoaşte cheia privată. Cel mai cunoscut sistem de criptare cu chei publice este **RSA**.

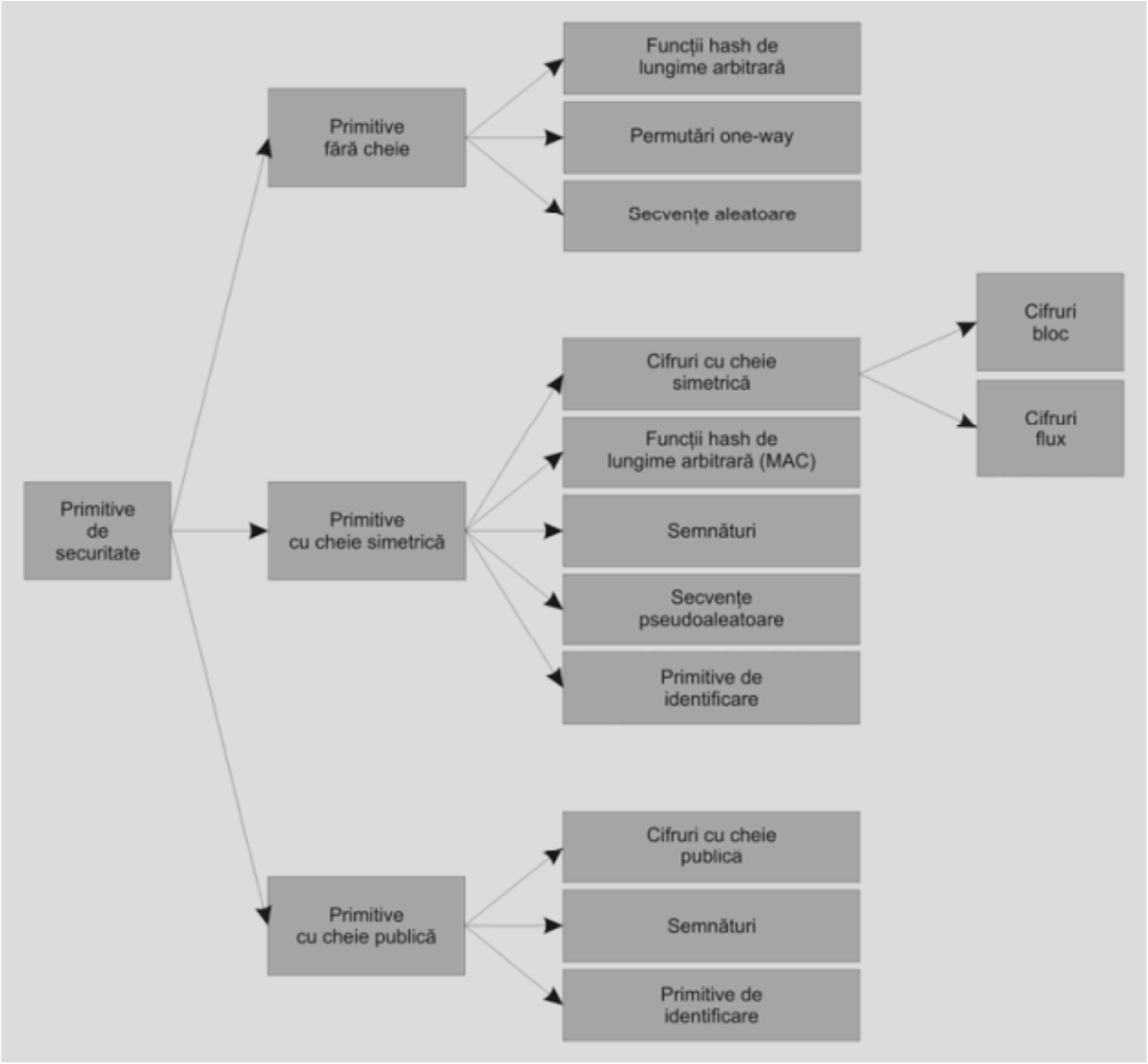


FIGURA I.5. Taxonomia primitivelor criptografice

Problemele afacerilor electronice din punct de vedere al securităţii pot fi împărţite, în mare, în patru domenii interconectate: ***confidenţialitate, autentificare, nerepudiere şi controlul integrităţii.***

Confidenţialitatea se referă la păstrarea informaţiei departe de utilizatorii neautorizaţi. Aceasta este ceea ce vine de obicei în mintea oamenilor atunci când se gândesc la *securitatea unei tranzacţii*.

Autentificarea reprezintă determinarea identităţii persoanei cu care se vorbeşte înainte de a dezvălui informaţii importante.

Nerepudierea implică semnături: cum se poate dovedi că un client a făcut întradevăr o comandă pentru o sută de mii de produse de 65 de eurocenţi fiecare, dacă, mai târziu, el pretinde că preţul era de 35 de eurocenţi?

Controlul integrităţii survine din întrebarea: Cum se poate garanta că un mesaj ce a fost primit a fost cel trimis cu adevărat şi nu unul pe care un atacator l-a modificat în tranzit?

La *nivelul fizic*, ascultarea firelor poate fi împiedicată prin închiderea liniilor de transmisie în tuburi sigilate conţinând gaz de argon la presiuni înalte. Orice încercare de

a sfredeli tubul va duce la pierderi de gaz, reducând presiunea şi trăgând alarma. Anumite sisteme militare folosesc această tehnică.
La *nivelul legătură de date*, pachetele transmise pe o linie punct-la-punct pot fi codificate când părăsesc una dintre maşini şi decodificate când intră în cealaltă. Toate detaliile pot fi manipulate la nivelul legătură de date, fără ca nivelurile mai înalte să aibă cunoştinţă de ceea ce se petrece. Această soluţie eşuează, totuşi, atunci când pachetele trebuie să traverseze mai multe routere, deoarece pachetele trebuie decriptate în fiecare router, făcându-le astfel vulnerabile la atacurile din interiorul routerelor. De asemenea, ea nu permite ca anumite sesiuni să fie protejate (de exemplu, acelea ce implică cumpărăturile on-line prin cărţi de credit), iar altele nu. Cu toate acestea, *criptarea legăturii (link encryption)*, cum este numită această metodă, poate fi adăugată cu uşurinţă la orice reţea si este adeseori utilă.
La *nivelul reţea*, pot fi instalate firewall-uri pentru a păstra pachetele în interior sau pentru a păstra pachetele în afara acestuia.
La *nivelul transport*, conexiuni întregi pot fi criptate, de la un capăt la celălalt, adică de la un proces la celălalt. Cu toate că aceste soluţii conduc la realizarea confidenţialităţii şi mulţi oameni muncesc din greu pentru a le îmbunătăţi, nici una dintre ele nu soluţionează *problema autentificării sau nerepudierii* într-un mod suficient de general. Pentru a rezolva aceste probleme, soluţiile trebuie să se găsească la *nivelul aplicaţie*.

În cazul cifrurilor bloc, unde mesajul în clar este împărţit în blocuri de dimensiune fixă (de exemplu: 64 sau 128 de biţi) şi supus, apoi, procedurii (algoritmului) de cifrare, este folosită îmbinarea tehnicilor de substituţie cu tehnici de transpoziţie, în mod repetitiv, cu scopul de a furniza cifrului proprietăţile de confuzie şi de difuzie, introduse de Claude Shannon în [9].

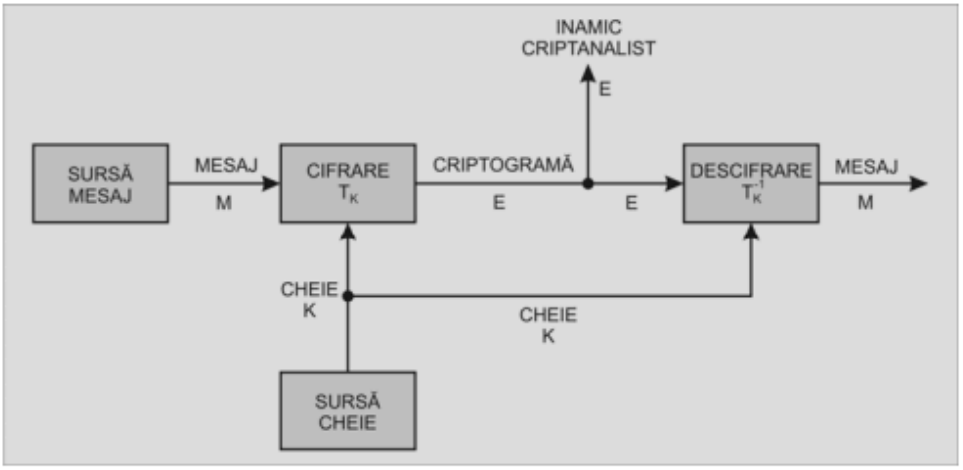


FIGURA I.1. Modelul lui Shannon pentru un criptosistem
(Sursa: Claude E. Shannon - *Communication Theory of Secrecy Systems* [9])

Modelul lui Shannon cuprinde:
- *sursa de mesaj* care emite mesajul M;
- *sursa de cheie* care emite cheia K;
- *funcţia de cifrare* care aplică transformarea T_K asupra mesajului M, generând criptograma E;
- *funcţia de descifrare* care aplică transformarea inversă T_K^{-1} asupra criptogramei E, generând mesajul M;
- *inamicul criptanalist*, entitate care atacă, folosind metode de criptanaliză, atât criptograma E, cât şi cheia K.

Confuzia reprezintă cerinţa ca relaţia dintre cheie şi mesajul cifrat să fie cât mai complexă cu putinţă. Difuzia este cerinţa ca influenţa unui singur bit al mesajului în clar să fie cât mai „împrăştiată”, asupra biţilor mesajului cifrat. În cifrurile bloc, fenomenul de difuzie propagă modificarea unui bit prin tot blocul. Acest tip de structură, în care sunt utilizate tehnici simple, precum substituţia, transpoziţia sau operaţii aritmetice modulare (operaţii cu modulo), se numeşte cifru produs, iar o categorie a cifrurilor produs o reprezintă cifrurile (reţelele) Feistel (după numele autorului acestora, criptologul american de la IBM, Horst Feistel).

In cryptography, a **substitution cipher** is a method of encrypting by which units of plaintext are replaced with ciphertext, according to a fixed system; the "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver deciphers the text by performing the inverse substitution.

Substitution ciphers can be compared with transposition ciphers. In a transposition cipher, the units of the plaintext are rearranged in a different and usually quite complex order, but the units themselves are left unchanged. By contrast, in a substitution cipher, the units of the plaintext are retained in the same sequence in the ciphertext, but the units themselves are altered.

There are a number of different types of substitution cipher. If the cipher operates on single letters, it is termed a **simple substitution cipher**; a cipher that operates on larger groups of letters is termed **polygraphic**. A **monoalphabetic cipher** uses fixed substitution over the entire message, whereas a **polyalphabetic cipher** uses a number of substitutions at different positions in the message, where a unit from the plaintext is mapped to one of several possibilities in the ciphertext and vice versa.

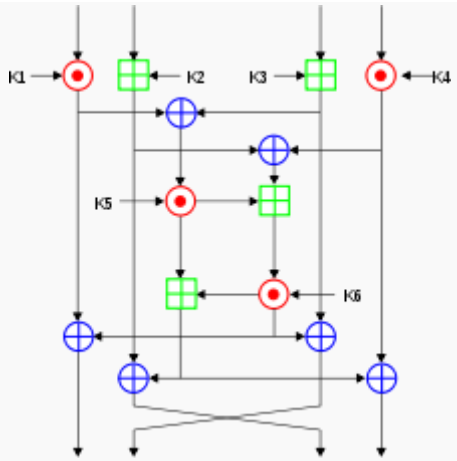
Algoritm criptografic	Confidențialitate	Autentificare	Integritate	Management al cheilor
Algorimi de cifrare simetrică	Da	Nu	Nu	Da
Algoritmi de cifrare asimetrică	Da	Nu	Nu	Da
Algoritmi de semnătură digitală	Nu	Da	Da	Nu
Algoritmi de stabilire a cheii	Da	Opțional		Da
Funcții hash cale-unică	Nu	Nu	Da	Nu
Coduri de autentificare a mesajelor	Nu	Da	Da	Nu

TABELUL I.1. Proprietățile algoritmilor de cifrare
(Sursa: Bruce Schneier - *Applied Cryptography (2nd Edition)* [13])

Criptografia moderna

Symmetric-key cryptography

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976.[10]



One round (out of 8.5) of the patentedIDEA cipher, used in some versions of PGPfor high-speed encryption of, for instance,e-mail

The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications. A block cipher is, in a sense, a modern embodiment of Alberti's polyalphabetic cipher: block ciphers take as input a block of plaintext and a key, and output a block of ciphertext of the same size. Since messages are almost always longer than a single block, some method of knitting together successive blocks is required. Several have been developed, some with better security in one aspect or another than others. They are themodes of operation and must be carefully considered when using a block cipher in a cryptosystem.

The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designatedcryptography standards by the US government (though DES's designation was finally withdrawn after the AES was adopted).[12] Despite its deprecation as an official standard, DES (especially its still-approved and much more secure triple-DES variant) remains quite popular; it is used across a wide range of applications, from ATM encryption[13] to e-mail

privacy[14] and secure remote access.[15] Many other block ciphers have been designed and released, with considerable variation in quality. Many have been thoroughly broken; see Category:Block ciphers.[11][16]

Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the one-time pad. In a stream cipher, the output stream is created based on a hidden internal state which changes as the cipher operates. That internal state is initially set up using the secret key material. RC4 is a widely used stream cipher; see Category:Stream ciphers.[11] Block ciphers can be used as stream ciphers; see Block cipher modes of operation.

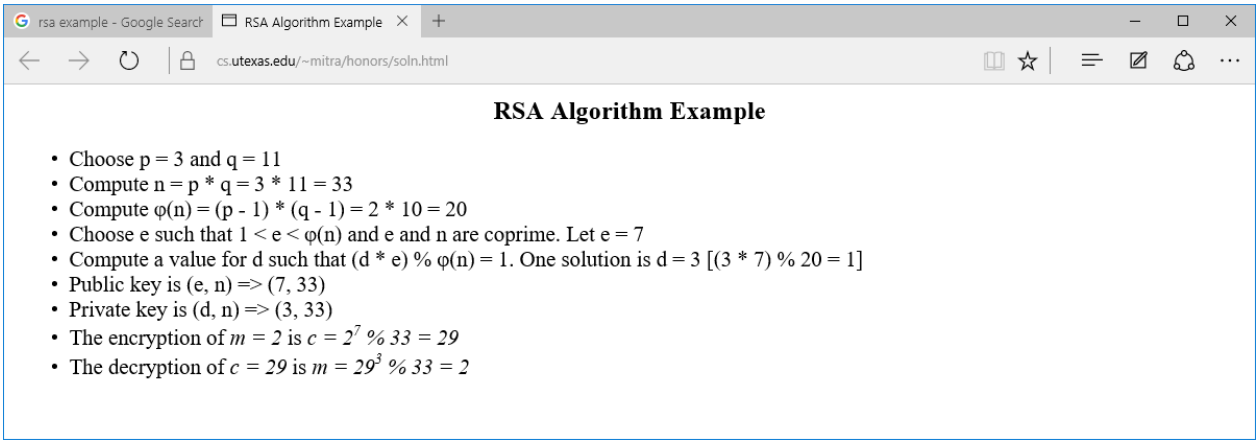
Cryptographic hash functions are a third type of cryptographic algorithm. They take a message of any length as input, and output a short, fixed length hash which can be used in (for example) a digital signature. For good hash functions, an attacker cannot find two messages that produce the same hash. MD4 is a long-used hash function which is now broken; MD5, a strengthened variant of MD4, is also widely used but broken in practice. The U.S. National Security Agency developed the Secure Hash Algorithmseries of MD5-like hash functions: SHA-0 was a flawed algorithm that the agency withdrew; SHA-1 is widely deployed and more secure than MD5, but cryptanalysts have identified attacks against it; the SHA-2 family improves on SHA-1, but it isn't yet widely deployed, and the U.S. standards authority thought it "prudent" from a security perspective to develop a new standard to "significantly improve the robustness of NIST's overall hash algorithm toolkit."[17] Thus, a hash function design competition is underway and meant to select a new U.S. national standard, to be called SHA-3, by 2012.

Message authentication codes (MACs) are much like cryptographic hash functions, except that a secret key can be used to authenticate the hash value[11] upon receipt.

Public-key cryptography

Main article: Public-key cryptography

Symmetric-key cryptosystems use the same key for encryption and decryption of a message, though a message or group of messages may have a different key than others. A significant disadvantage of symmetric ciphers is the key management necessary to use them securely. Each distinct pair of communicating parties must, ideally, share a different key, and perhaps each ciphertext exchanged as well. The number of keys required increases as the square of the number of network members, which very quickly requires complex key management schemes to keep them all straight and secret. The difficulty of securely establishing a secret key between two communicating parties, when a secure channel does not already exist between them, also presents a chicken-and-egg problem which is a considerable practical obstacle for cryptography users in the real world.



Whitfield Diffie and Martin Hellman, authors of the first published paper on public-key cryptography

In a groundbreaking 1976 paper, Whitfield Diffie and Martin Hellman proposed the notion of *public-key* (also, more generally, called *asymmetric key*) cryptography in which two different but mathematically related keys are used—a *public* key and a *private* key.[18] A public key system is so constructed that calculation of one key (the 'private key') is computationally infeasible from the other (the 'public key'), even though they are necessarily related. Instead, both keys are generated secretly, as an interrelated pair.[19] The historian David Kahn described public-key cryptography as "the most revolutionary new concept in the field since polyalphabetic substitution emerged in the Renaissance".[20]

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. The *public key* is typically used for encryption, while the *private* or *secret key* is used for decryption. Diffie and Hellman showed that public-key cryptography was possible by presenting the Diffie–Hellman key exchange protocol.[10]

In 1978, Ronald Rivest, Adi Shamir, and Len Adleman invented RSA, another public-key system.[21]

In 1997, it finally became publicly known that asymmetric key cryptography had been invented by James H. Ellis at GCHQ, a British intelligence organization, and that, in the early 1970s, both the Diffie–Hellman and RSA algorithms had been previously developed (by Malcolm J. Williamson and Clifford Cocks, respectively).[22]

The Diffie–Hellman and RSA algorithms, in addition to being the first publicly known examples of high quality public-key algorithms, have been among the most widely used. Others include the Cramer–Shoup cryptosystem, ElGamal encryption, and various elliptic curve techniques. See Category:Asymmetric-key cryptosystems.

Padlock icon from the FirefoxWeb browser, meant to indicate a page has been sent in SSL or TLS-encrypted protected form. However, such an icon is not a guarantee of security; any subverted browser might mislead a user by displaying such an icon when a transmission is not actually being protected by SSL or TLS.

In addition to encryption, public-key cryptography can be used to implement digital signature schemes. A digital signature is reminiscent of an ordinarysignature; they both have the characteristic that they are easy for a user to produce, but difficult for anyone else to forge. Digital signatures can also be permanently tied to the content of the message being signed; they cannot then be 'moved' from one document to another, for any attempt will be detectable. In digital signature schemes, there are two algorithms: one for *signing*, in which a secret key is used to process the message (or a hash of the message, or both), and one for *verification,* in which the matching public key is used with the message to check the validity of the signature. RSAand DSA are two of the most popular digital signature schemes. Digital signatures are central to the operation of public key infrastructures and many network security schemes (e.g., SSL/TLS, many VPNs, etc).[16]

Public-key algorithms are most often based on the computational complexity of "hard" problems, often from number theory. For example, the hardness of RSA is related to the integer factorization problem, while Diffie–Hellman and DSA are related to the discrete logarithm problem. More recently, *elliptic curve cryptography* has developed in which security is based on number theoretic problems involving elliptic curves. Because of the difficulty of the underlying problems, most public-key algorithms involve operations such as modular multiplication and exponentiation, which are much more computationally expensive than the techniques used in most block ciphers, especially with typical key sizes. As a result, public-key cryptosystems are commonly hybrid cryptosystems, in which a fast high-quality symmetric-key encryption algorithm is used for the message itself, while the relevant symmetric key is sent with the message, but encrypted using a public-key algorithm. Similarly, hybrid signature schemes are often used, in which a cryptographic hash function is computed, and only the resulting hash is digitally signed.[11]

Exercitii:

Steganografie

Se cere ascunderea secvenței de biti: A7 in secvența de pixeli (RGBA)

2C 2D B3 FE => 2E 2E B1 FF

Se vor folosi ultimii 2 biti din fiecare octet.

Rezolvare

Se extrag bitii valorii A7: A7=10 10 01 11

Apoi se introduc in ordine in ultimii doi biti ai fiecarui octet.

A7=**10** 10 01 11

2C= 0010 1100 0010 11**10**=2E

A7=10**10** 0111

2D= 0010 1101 0010 11**10**=2E

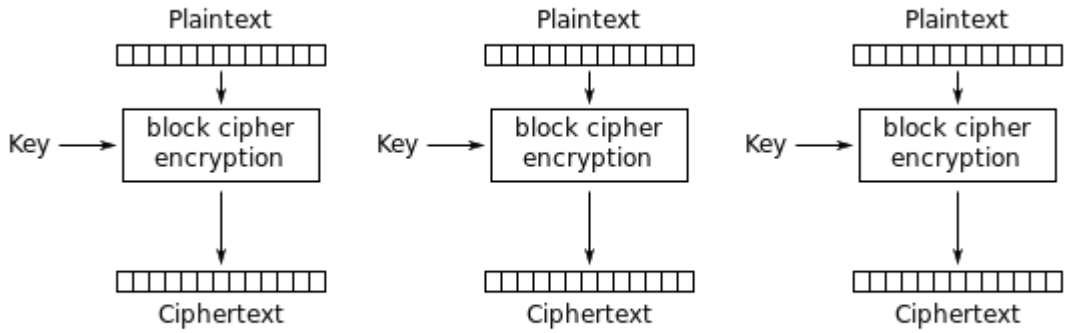
A7=1010 **01**11

B3= 1011 0011 1011 00**01**=B1

A7=1010 01**11**

FE= 1111 1110 1111 11**11**=FF

Criptare ECB ELECTRONIC CODEBOOK



Electronic Codebook (ECB) mode encryption

Se va cripta folosind XOR si algoritmul ECB secventa: 29 A1 B5

Folosind cheia A7. Realizati si operatia decriptare.

Rezolvare

29– 00 10 10 01 A1- 10 10 00 01 B5- 10 11 01 01

A7- 10 10 01 11 A7- 10 10 01 11 A7- 10 10 01 11

---------- XOR

10 00 11 10 00 00 01 10 00 01 00 10

8E 06 12 (criptare)

Criptare CBC CHYPHER BLOCK CHAINING

Se va cripta folosind XOR si alg CBC secventa: 29 A1 B5

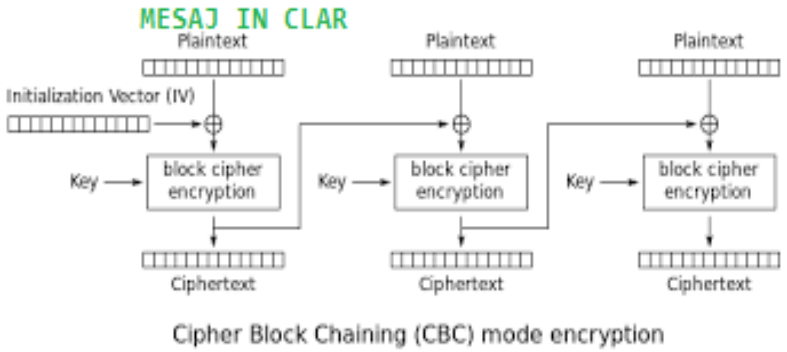
Folosind cheia A7 si vectorul de initalizare B2.

Rezolvare

Se transforma valorile in binary pentru a putea face operatii B2= 10 11 00 10

29 -00 10 10 01 A1 -10 10 00 01 B5 - 10 11 01 01

A7- 10 10 01 11 A7- 10 10 01 11 A7- 10 10 01 11



Cipher Block Chaining (CBC) mode encryption

XOR

29 -00 10 10 01	A1 -10 10 00 01	B5 - 10 11 01 01
B2 - 10 11 00 10	3C- 00 11 11 00	3A - 00 11 10 10
----------	----------	----------
10 01 10 11	10 01 11 01	10 00 11 11
A7- 10 10 01 11	A7- 10 10 01 11	A7- 10 10 01 11
----------	----------	----------
00 11 11 00	00 11 10 10	00 10 10 00

Bibliografie:

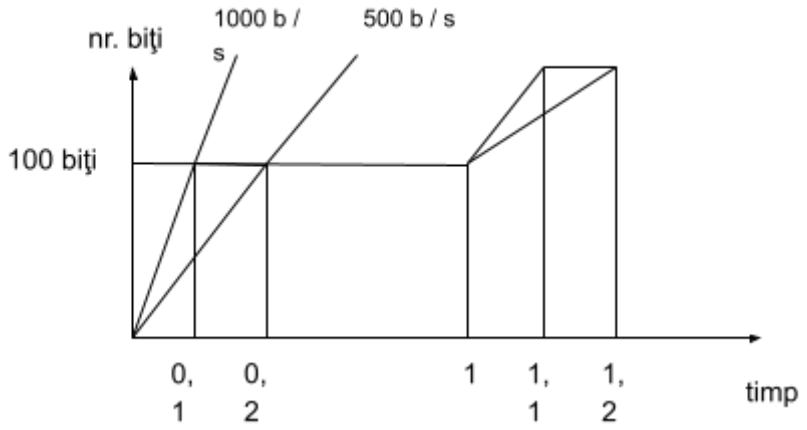
1) V.M. Ionescu, I. Sima, E. Sofron, „Aplicatii software pentru protocoale de comunicatie”, Ed. MatrixROM, 2008, ISBN 987-973-755-302-7
2) Tutănescu I., Buzuloiu A., Călugăreanu M., Vlad I. - sub coordonarea Prof. univ. dr. ing. Sofron E. - “Protecţia reţelelor de calculatoare conectate la Internet”, Editura Universităţii din Piteşti, 2000, ISBN 973-8212-06-5.
3) Andrew S. Tanenbaum - "Reţele de calculatoare", Ediţia a patra, Ed. Teora, 2004;
4) Ion Bănică - "Reţele de comunicaţii între calculatoare", Ed. Teora, Bucureşti, 1998;
5) James Chellis, Charles Perkins, Mathew Strebe - "Elemente fundamentale ale reţelelor de calculatoare", Editura All Educational, Bucureşti, 2000;
6) Zoltan Kasa, Horia Pop - "Comunicare în Internet", 1999;
7) Lary Schumer - "Utilizare UNIX", 1998;
8) Russel Sage - "UNIX pentru profesionişti", 1998;
9) Iosif Ignat - "UNIX - Gestionarea proceselor", 1996;
10) Steve Oualline - "Descoperiţi sistemul LINUX", 1998;
11) Debra Niedermiller - "Administrarea sistemelor Novell NetWare", 1997;
12) Terry Ogletree - „Reţele de calculatoare: depanare si modernizare”, Ed. Teora, 2001;
13) Claudiu Bulaceanu, “Retele locale de calculatoare: Arhitecturi prezente si viitoare”, Ed. Tehnica, Bucuresti, 1995
14) V.V. Patriciu - "Criptografia şi securitatea reţelelor de calculatoare", Editura Tehnică, Bucureşti, 1994;
15) V.V. Patriciu, Bica Ion - "Securitatea informatică în UNIX şi Internet", Ed. Techica Bucuresti, 1998;
16) Lars Klander - "Anti Hacker - Ghidul securităţii reţelelor de calculatoare", 1998;
17) Tutănescu I., Călugăreanu M., Bondor K., sub coordonarea prof. univ. dr. ing. Maghiar T. - “Sisteme de prelucrare şi protecţie a informaţiei”, Editura Universităţii din Oradea, 2001, ISBN 973-8219-58-2.
19) Rosca, Ion Gh., Tapus, Nicolae, “Internet si intranet: Concepte si aplicatii” Ed. Economica, Bucuresti, 2000
20) Sabin Buraga şi Gabriel Ciobanu, “Atelier de programare în reţele de calculatoare“, Polirom, Iaşi, , Polirom, Iaşi, 2001
21) L.Scripcaru, I.Bogdan, S.V.Nicolaescu, “Securitatea Reţelelor De Comunicaţii”, Casa de Editură VENUS, Iaşi 2008

Aplicatii

1. Calculati CRC-4 si verificati corectitudinea pentru: secventa de 8 biti: 10111110 si polinomul generator CRC-4: 10101.
2. Stiind ca intra intr-un router pachete cu viteza de 700b/s si ies cu viteza de 350b/s, iar dimensiunea bufferului este de 2kbiti calculati:

a. numarul de pachete care tb sa intre in router intr-o secunda pentru a umple bufferul dupa 10 secunde. Ilustrati grafic acest proces
b. Ocuperea medie cu biti de date a routerului daca in fiecare secunda intra 150biti.

3. Adresarea IP:
   a. precizati o adresa de clasa A. B si C. precizati pentru fiecare masca, adresa de retea si adresa de braodcast. Arătaţi cum se calculează acestea!
   b. se da adresa IP de clasa C: …. Se cere sa se realizeze un număr de subretele care sa permita 30 de calculatoare intr-o subreatea. Precizaţi masca rezultata si adresele de retea, precum si cele de broadcast pentru fiecare din sub-reţelele rezultate. Explicaţi modul in care ati determinat aceste valori!



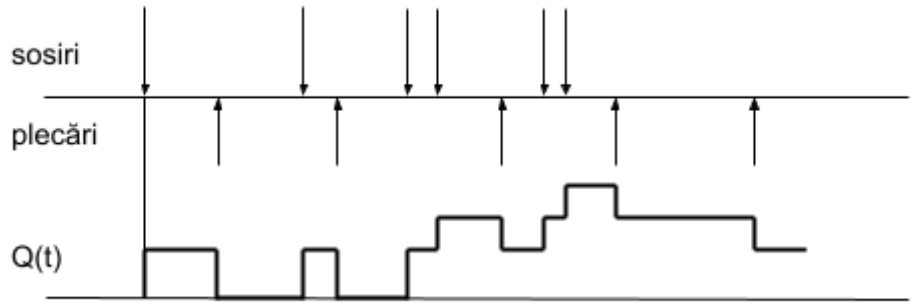
0 – 0,2 buffer ocupat
0,2 – 1 buffer gol

Încărcarea medie cât timp bufferul are biţi:

$$\overline{Q(t)} = \frac{1}{2}\left(500\frac{b}{s} \times 0{,}1\ s\right) = 25\ biți$$

$$\overline{Q_{t_{tot}}} = 0{,}2 \cdot 25 = 5\ biți$$

Evoluţia în timp a cozii de pachete:



Subiecte examen Reţele feb 2010

1.Detalierea comunicaţiei de la sursa la destinaţie la nivelul internetului. (modele de comunicaţie, protocoale, etc.) Precizaţi totodată echipamentele de reţea si mediile de transmisie ce pot fi folosite.

2. Caracteristicile comunicaţiei folosind pachete de date.

3. O firma doreşte sa asigure acces internet la cele 5 departamente ale sale, in acelaşi timp maximizând numărul de adrese disponibile pentru utilizatori. Deoarece firma are multe calculatore in fiecare departament, a hotărât sa folosească pentru acestea o adresa de clasa B impartita in subretele.

A. Ştiind ca unul dintre departamente foloseşte adresa: 172.18.65.255, precizaţi masca, adresele de subreţea si adresele de broadcast folosite pentru cele 5 departamente. Calculati din ce subretea face parte adresa precizata.

B. Prezentaţi o soluţie pentru conectarea la internet a acestei reţele private si exemplificaţi grafic, printr-o schema, modul de conectare al unuia dintre calculatoarele din reţeaua anterioara la internet. Precizaţi pentru fiecare dispozitiv prezent: adresele IP, gateway, masca.

C. Propuneţi mai multe soluţii pentru prevenirea atacurilor asupra reţelei

4. Stiind ca intr-un router intra pachete cu viteza de 1kb/s si ies cu viteza de 400b/s, iar dimensiunea bufferului este de 1500biti calculaţi:

a. numarul de biti care tb sa intre in router intr-o secunda pentru a umple bufferul dupa 10 secunde.

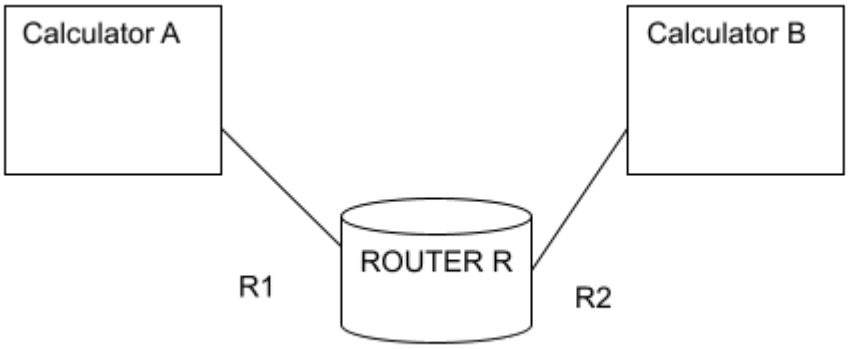
b. Ilustrati grafic evoluţia în timp a cozii de pachete

c. Ocuperea medie cu biti de date a routerului daca in fiecare secunda intra 200biti.

1.Detalierea comunicaţiei de la sursa la destinaţie la nivelul internetului. (modele de comunicaţie, protocoale, etc.) Precizaţi totodată echipamentele de reţea si mediile de transmisie ce pot fi folosite.

2. O firma doreşte sa asigure acces internet la cele 5 departamente ale sale, in acelaşi timp maximizând numărul de adrese disponibile pentru utilizatori. Deoarece firma are multe calculatore in fiecare departament, a hotărât sa folosească pentru acestea o adresa de clasa B impartita in subretele. Ştiind ca unul dintre departamente foloseşte adresa: 172.18.65.255, precizaţi masca, adresele de subreţea si adresele de broadcast folosite pentru cele 5 departamente. Calculati din ce subretea face parte adresa precizata.

3. Daca un router R are conectate 2 calculatoare ca in figura, cu adresele IP:
-A=192.168.90.81 ;B=192.168.90.105; R1=192.168.90.110;R2=192.168.90.92; MASCA /29
-A=192.168.90.178 ;B=192.168.90.229; R1=192.168.90.190;R2=192.168.90.235; MASCA /29
-A=192.168.90.186 ;B=192.168.90.145; R1=192.168.90.190;R2=192.168.90.147; MASCA /30



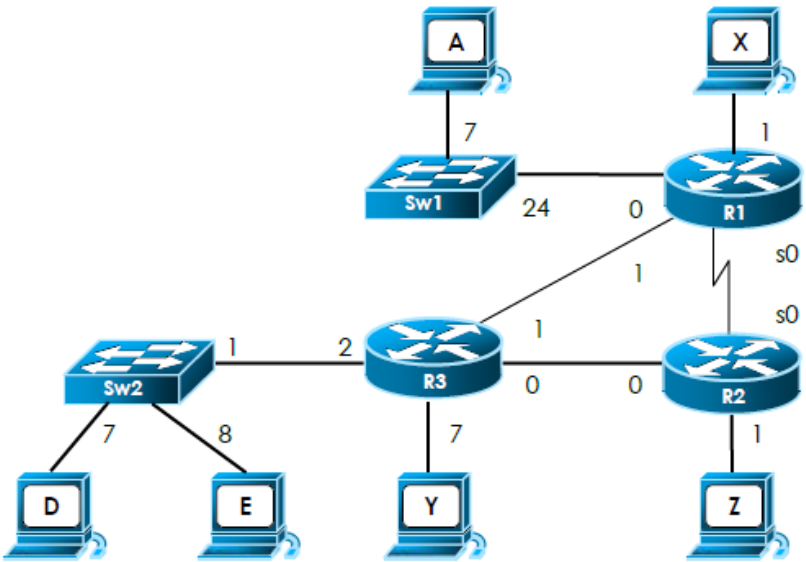
Calculaţi care reţea va funcţiona corect si care nu. Explicati de ce.

4. medii de transmisie. avantaje si dezavantaje
5. comenzi folosite in retele de calculatoare (minim 6). Detalii
6. un administrator de retea proiecteaza extinderea retelei la 5 noi locaţii distincte. Se doreste folosirea unei clase de adrese C. cate calculatoare poate avea maxim fiecare locatie?

7. comparatie TCP si UDP
8. prezentati serviciile de retea http, FTP, DNS, DHCP

În reţeaua dată pentru cele două reţele cu switchuri s-au folosit adrese private, astfel R1 şi R3 vor asigura translatare de adresă cu supraîncărcare (PAT). Vom considera că tabelele ARP din reţea au fost configurate static pentru toate destinaţiile.

1. Descrieţi antetele pachetelor apărute în reţea în cazul în care A trimite un singur pachet către E.
2. Ambele switchuri sunt repornite. Staţia E trimite un pachet către Z, iar X un pachet către D. Ce intrări vor exista în tabela de comutare a switchului sw2 în final?
3. Câte domenii de difuzare (broadcast) sunt în topologia dată?



4. Cum se realizează securitatea memoriei la nivelul sistemului de operare?
5. Descrieţi comparative stiva de protocoale OSI şi stiva de protocoaleTCP/IP
6. Tipuri de codificare în transmisia digitală
7. Care este rolul câmpului durată din antetul 802.11?
8. Pentru configurarea unei zone de nume pe un server bind este necesară editarea a 3 fişiere.Care este rolul fiecăruia dintre cele 3 fişiere?
9. La ce se referă sintagma „social engineering”?
10. În urma unui handshake între un client (C) şi un server (S) se stabilesc numerele de secvenţă 1000, respective 2000 în cele două sensuri de comunicaţie.Clientul interoghează serverul folosind 3 pachete successive de 100 de octeţi (payload TCP),iar serverul îi răspunde(pentru fiecare pachet) folosind pachete de 1000 de octeţi. După transmiterea celor 3 pachete şi a răspunsurilor la acestea, clientul iniţiază încheierea conexiunii.Descrieţi antetul TCP(flag-uri şi numerele de secvenţă) al segmentelor schimbate(inclusiv încheierea conexiunii).