Rețele de calculatoare

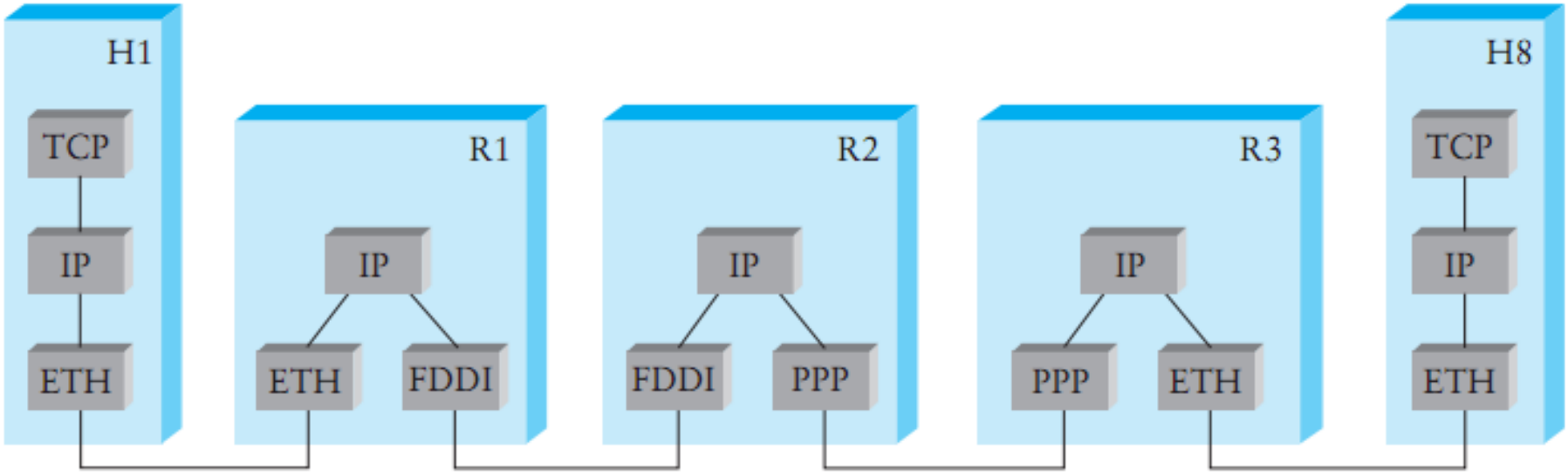
Partea a 5-a

Sebastian Fuicu

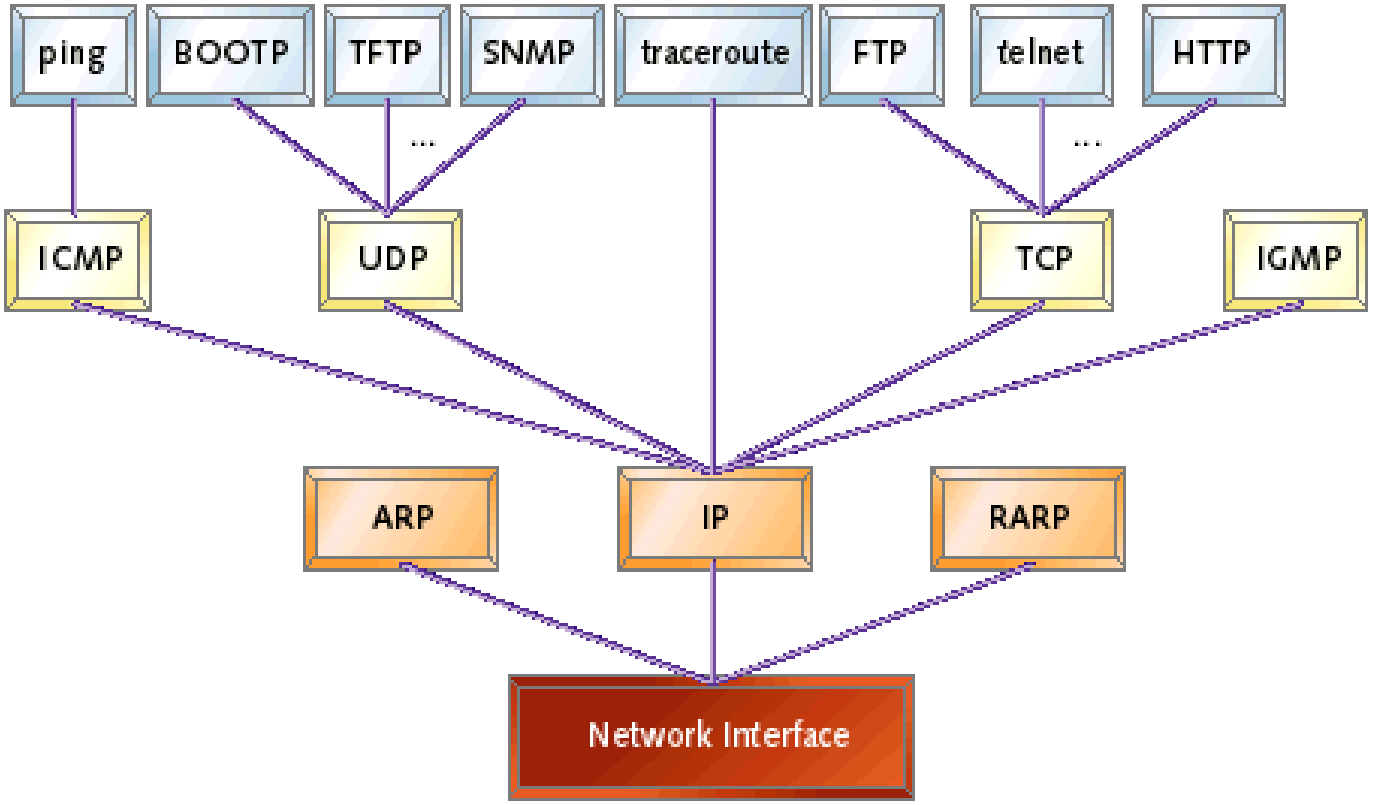
- **Interconectarea rețelelor**
- **Protocolul IP**
- **Protocolul TCP**
- **Controlul congestiei**

Interconectarea rețelelor

- Interconectarea reţelor reprezintă mecanismul prin care o colecţie arbitrară de reţele oferă împreună un serviciu de transport de tip host-to-host.
- Protocolul IP este elementul cheie folosit pentru a construi reţele interconectate.
- O instanţă a acestui protocol trebuie să ruleze pe fiecare din nodurile reţelelor interconectate.



Protocolul IP



Protocolul IP

Serviciul oferit de protocolul IP

- Modelul serviciului prezintă două componente:

  - o schemă de adresare care permite identificarea tuturor host-urilor interconectate.

  - un model de transfer al datelor, de tip datagramă, fără conexiune, model numit si „best effort”.

- Noţiunea de „best effort” se traduce prin aceea că dacă un pachet se pierde sau este afectat de eroare, reţeaua nu face nimic pentru a-l recupera, pentru ca ea a depus deja tot efortul pentru a trasporta acel pachet.

- Este posibil ca pachetele care au fost transmise să ajungă în altă ordine decât cea in care au fost transmise sau unele să fie duplicate.

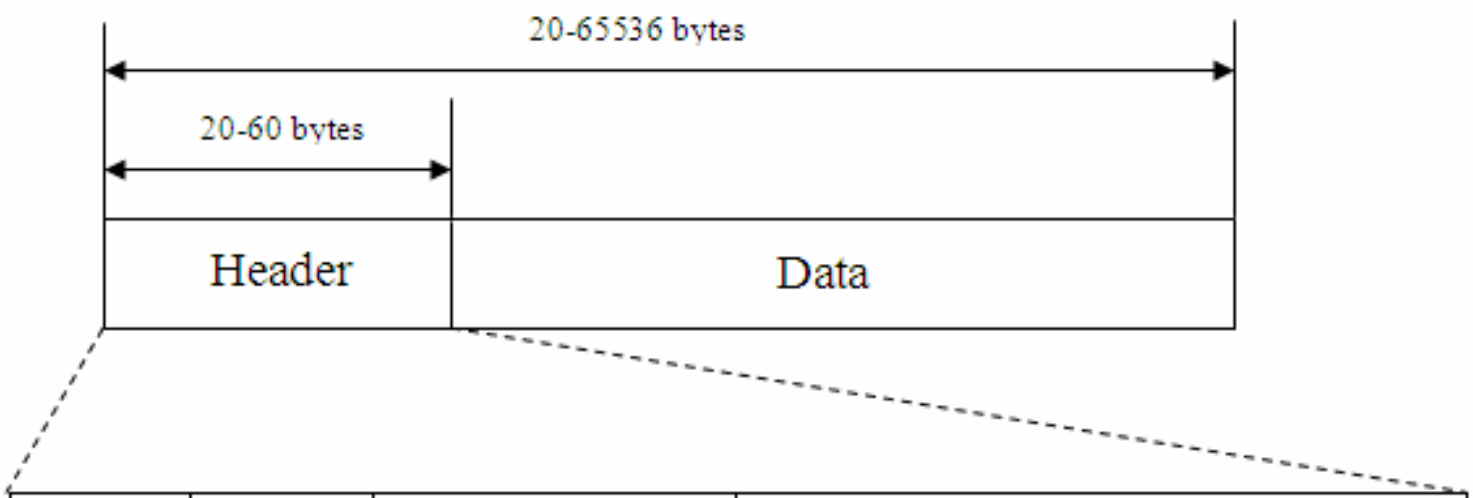
Protocolul IP

Serviciul oferit de protocolul IP

- Serviciul de tipul „best effort” fără conexiune este cel mai simplu serviciu care poate fi oferit in cazul interconectării reţelelor.
- Păstrarea simplităţii software-ului care rulează pe routere a fost unul din principalele scopuri ale protocolului IP.
- Abilitatea protocolului IP de a rula peste orice tip de reţea reprezintă o altă caracteristică fundamentală a acestuia.
- Multe din tehnologiile peste care protocolul IP rulează nici măcar nu erau inventate în momentul apariţiei acestuia.

Protocolul IP

Formatul unui pachet IP



Version 4 bits	IHL 4 bits	Service Type 8 bits	Total Length 16 bits
Identification 16 bits		Flags 3 bits	Fragment Offset 13 bits
Time to Live 8 bits	Protocol 8 bits	Header Checksum 16 bits	
Source IP address 32 bits			
Destination IP address 32 bits			
Option			

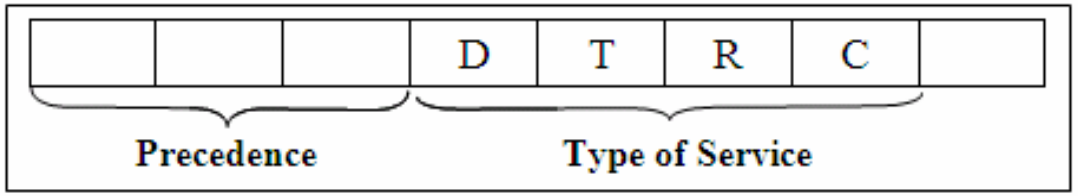
Protocolul IP

Formatul unui pachet IP

- ***Version***: Versiunea protocolului. Există două versiuni funcţionale, 4 şi 6. În acest material este prezentată doar versiunea 4.

- ***IHL*:** Câmpul ne spune lungimea header-ului, exprimată în număr de cuvinte de 4 octeţi. Dacă lungimea este 20, atunci valoare lui IHL este 5.

- ***Service Type:*** Câmpul acesta este împărţit la rândul lui în mai multe subcâmpuri.



- ***Precedence*:** a fost gândit pentru a defini prioritatea unui pachet. În versiunea 4 a protocolului IP acest subcâmp nu este folosit.

Protocolul IP

Formatul unui pachet IP

- ***Type of Service***: acest subcâmp este format din 4 biţi. Fiecare dintre ei are o anumite semnificaţie şi doar unul poate fi setat la un moment dat.

ToS	Semnificație
0000	Normal
0001	Minimizează costul
0010	Maximizează siguranța
0100	Maximizează capacitatea de transfer
1000	Minimizează întârzierea

- ***Total Length***: Acest câmp conţine lungimea totală a pachetului. Dacă se doreşte să se afle lungimea datelor, se scade din lungimea totală, valoarea câmpului Header Length înmulţită cu 4.

- ***Identification, Flags, Fragmentation Offset***: Folosite în procesul de fragmentare a pachetelor.

Protocolul IP

Formatul unui pachet IP

- ***Time to Live*:** Acest câmp este folosit pentru a stabili numărul maxim de hop-uri (routere) prin care un pachet poate trece.

- Fiecare router care procesează pachetul decrementează câmpul cu o unitate. Când valoare ajunge la zero, pachetul este eliminat din reţea şi un mesaj de eroare este generat către nodul care avea adresa trecută în câmpul *Source IP Address*.

- Valoarea de iniţializare a acestui câmp este de obicei dublul numărului maxim de router-e care se pot interpune între sursă şi destinaţie.

- Este necesar acest mecanism deoarece în absenţa lui şi în anumite circumstanţe (tabele de rutare corupte) anumite pachete ar putea călători la infinit în reţea, consumând inutil resursele reţelei.

Protocolul IP

Formatul unui pachet IP

- ***Protocol*:** Prin acest câmp se identifică protocolul de nivel superior care face uz de protocolul IP pentru a-şi transporta datele

Protocol	Valoare
TCP	6
UDP	17
ICMP	1
OSPF	89
EGP	8
Ipv6	41

- ***Checksum*:** Sumă de control aplicată pachetului.
- ***Source Address*:** Câmpul conţine adresa nodului care a trimis pachetul.
- ***Destination Address*:** Câmpul conţine adresa nodului căruia îi este destinat pachetul.

Protocolul IP

Fragmentarea şi reasamblarea pachetelor

- Fiecare protocol de pe Nivelul Legătură de Date are propriul format pentru frame-urile utilizate la transportul informaţiei. Când un pachet IP traversează diferite reţelele el trebuie să fie încapsulat în aceste frame-uri ale Nivelului Legătură de Date.
- Fiecare frame acceptă o anumită dimensiune maximă pentru câmpul de date. Astfel, dacă dimensiunea pachetului IP depăşeşte această valoare, pachetul va trebui să fie fragmentat.
- Câmpul *Flags* conţine 3 biţi. Primul este rezervat, iar următorii doi sunt notaţi cu D (*do not fragment*), respectiv M(*more fragment*). Dacă bitul D are valoarea 1, atunci pachetul nu poate fi fragmentat. Daca bitul M are valoarea 1 aceasta semnifică că pachetul nu este ultimul fragment ci mai sunt şi altele. Dacă valoarea este 0, atunci fragmentul este ultimul.
- Câmpul *Fragmentation Offset* indică poziţia relativă a unui fragment în cadrul unui pachet. Poziţia este indicată sub formă de deplasament exprimat ca multiplu de 8 octeţi.

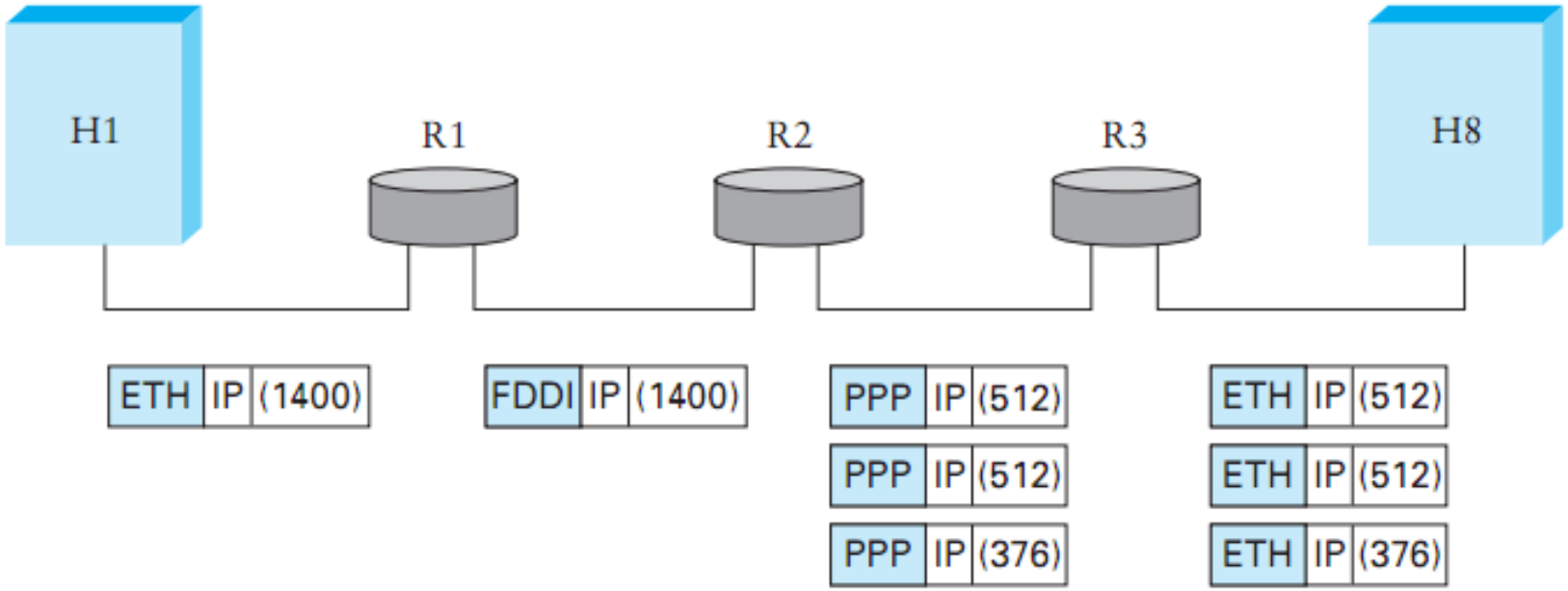
Protocolul IP

Fragmentarea şi reasamblarea pachetelor

- Câmpul *Identification* are şi el un rol important în procesul de fragmentare. Fiecare pachet primeşte un număr de identificare care va fi stocat în acest câmp. Acest număr este generat prin incrementare cu unu pentru fiecare nou pachet trimis. Valoarea de la care se porneşte este una pozitivă aleasă în mod aleator. Astfel, pachetele vor fi identificate în mod unic folosind această etichetă şi adresa sursei. Atunci când este necesar ca un pachet să fie fragmentat, fragmentele care au rezultat vor avea acelaşi număr de identificare cu cel al pachetului din care provin. În acest fel va putea fi refăcut pachetul original.

Protocolul IP

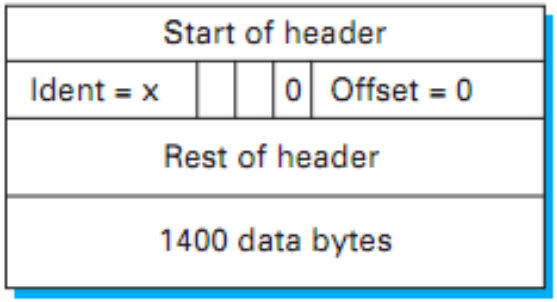
Fragmentarea şi reasamblarea pachetelor



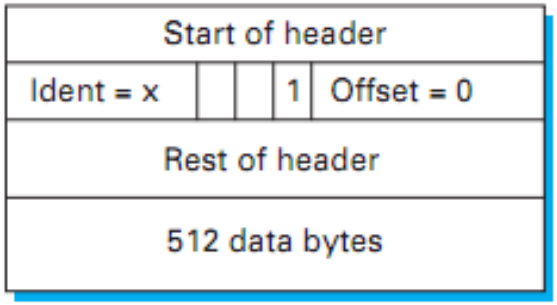
Protocolul IP

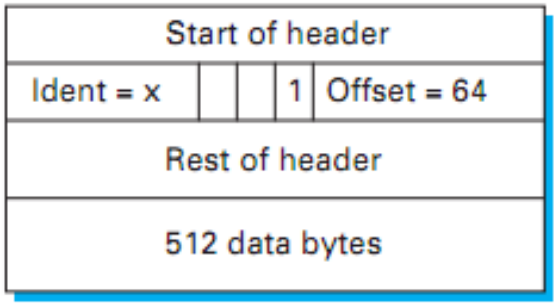
Fragmentarea şi reasamblarea pachetelor

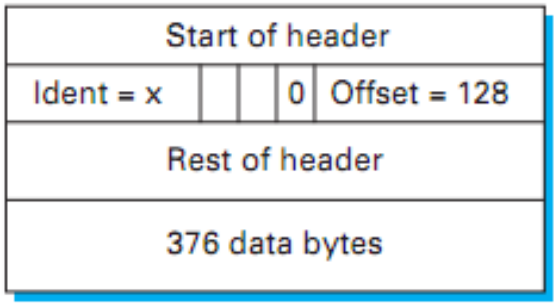
(a)



(b)





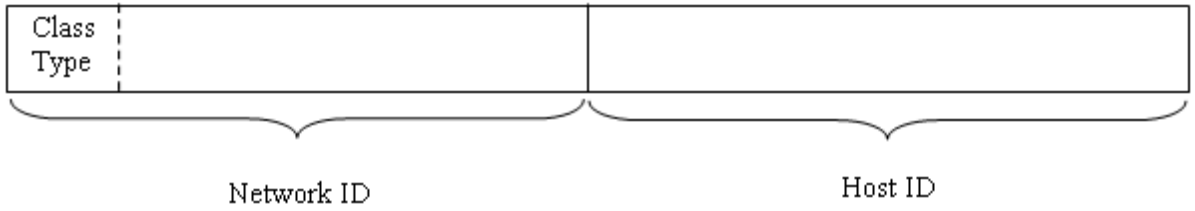


Header fields used in IP fragmentation. (a) Unfragmented packet; (b) fragmented packets.

Protocolul IP

Adresarea

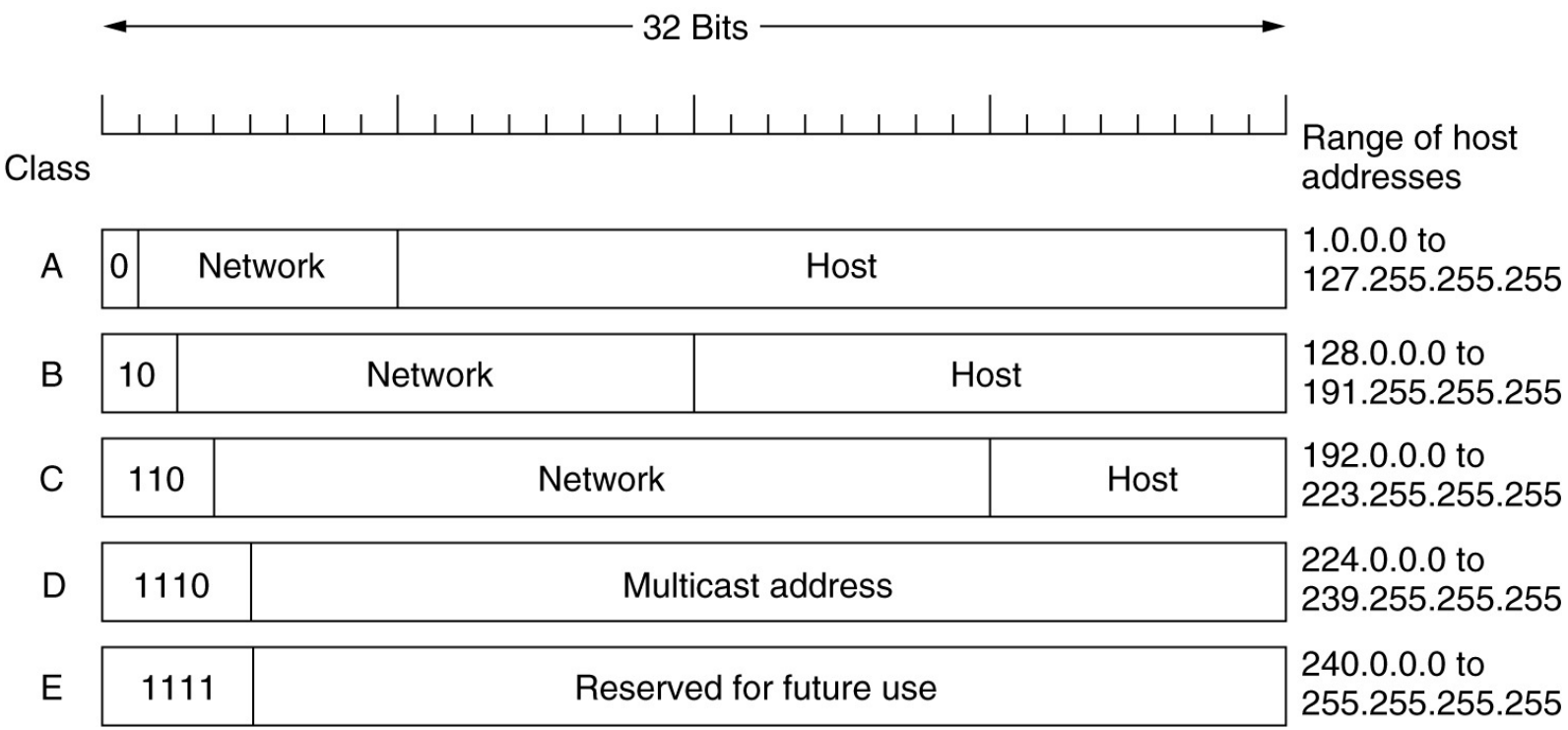
- Protocolul IP implementează o schemă de adresare globală.
- Întregul spaţiu de adrese IP a fost împărţit în 5 clase notate de la A la E.
- O adresă IP conţine doua părţi: una care ne dă adresa reţelei şi cealaltă ne spune adresa host-ului în cadrul reţelei.
- Valuarea primilor biţi dintr-o adresă IP ne spune clasa din care face parte acea adresă. Fiecare clasă alocă un număr diferit de biţi pentru partea de Network ID şi pentru Host ID.



Class	High Order Bits	Netid	Hostid
A	0	7 bits	24 bits
B	10	14 bits	16 bits
C	110	21 bits	8 bits
D	1110	28 bits multicast groupe number	
E	1111	reserved	

Protocolul IP

Adresarea



Protocolul TCP

- Protocolul TCP are sarcina de a transforma Nivelul Reţea, reprezentat în cazul nostru prin protocolul IP, dintr-un nivel nesigur într-unul sigur.
- Tot protocolul TCP este responsabil şi cu implementarea anumitor mecanisme de control al fluxului şi al congestiei.
- Protocolul TCP este de tipul capăt la capăt, adică este necesar să existe o instanţă a acestui protocol doar pe maşina sursă şi pe maşina destinaţie, nu şi în nodurile intermediare care vor fi tranzitate de către pachete.

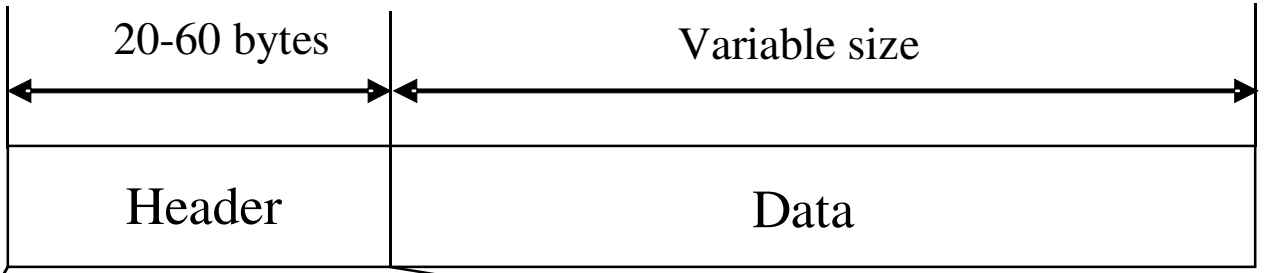
Protocolul TCP

Serviciile oferite de către TCP sunt:

- Stream Data Service
  - Aceasta presupune că pachetele care ajung la destinaţie să fie recepţionate exact în ordinea în care au fost trimise.
  - Dacă trebuie transmis un mesaj, care din cauza lungimii va fi fragmentat în mai multe pachete, la receptor se verifică sosirea tuturor pachetelor şi se aşează în ordinea în care au fost trimise.
  - Pentru a putea furniza acest serviciu, TCP face uz de buffer-e atât pe partea de transmisie, cât şi pe partea de recepţie.
- Full-duplex service
  - Presupune că transferul de informaţii între două noduri poate fi făcut simultan în ambele direcţii.
  - Când un pachet pleacă de la unul din noduri către celălalt, transportă şi un mesaj de confirmare pentru un pachet recepţionat anterior (piggybacking).
- Reliable service
  - TCP foloseşte un mecanism de confirmări pentru a se asigura că pachetele nu au fost afectate de erori şi ca au sosit în ordinea corectă.

Protocolul TCP

Formatul unui pachet TCP



Source port number 16 bits								Destination port number 16 bits
Sequence number 32 bits								
Acknowledgement number 32 bits								
Header length 4 bits	Reserved 6 bits	URG	ACK	PSH	RST	SYN	FYN	Window size 16 bits
Checksum 16 bits								Urgent pointer 16 bits
Option (if any)								

Protocolul TCP

Formatul unui pachet TCP

- ***Source port number*** **:** Numărul portului folosit de către aplicaţia care rulează pe maşina care trimite pachetele.
- ***Destionation port number*** **:** Numărul portului folosit de aplicaţia care rulează pe maşina care primeşte pachetele.
- ***Sequence number:*** TCP numerotează fiecare octet trimis şi foloseşte acest câmp pentru a indica numărul de secvenţă al primului octet de date din pachet. Când se iniţiază o comunicaţie între două maşini, numărul de secvenţă pentru primul octet trimis este ales aleator.
- ***Acknowledgement number:*** Acest câmp este folosit pentru a confirma octeţii recepţionaţi. El se calculează însumând la valoarea din câmpul *Sequence number* al pachetului primit, dimensiunea câmpului de date recepţionat, plus 1. Astfel valoarea acestui câmp reprezintă de fapt următoarea valoare pentru câmpul *Sequence number* care va fi folosită de către maşina care recepţionează pachetul de confirmare.

Protocolul TCP

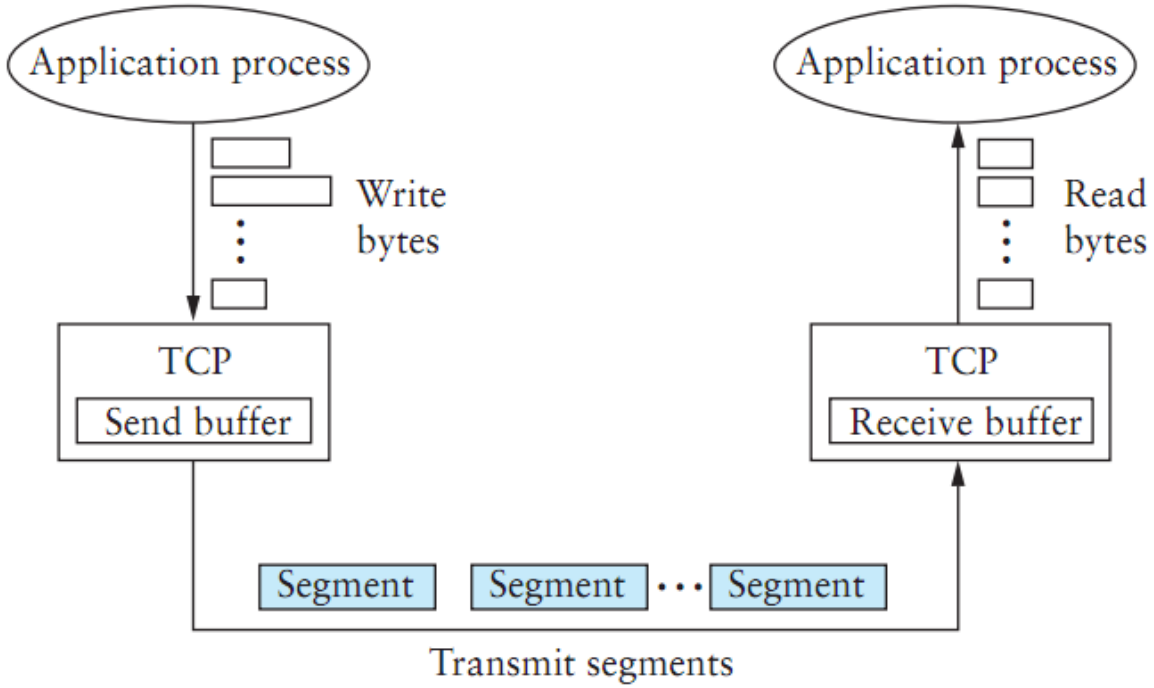
Formatul unui pachet TCP

- ***Header length:*** Conţine dimensiunea header-ului exprimată in cuvinte de 32 de biţi.
- ***Control field:*** Conţine 6 biţi a căror semnificaţie va fi explicată atunci când se va discuta modul cum se iniţiază şi se desfăşoară o sesiune de comunicaţie.
- ***Window size:*** reprezintă spaţiul disponibil din buffer-ul de recepţie.

Protocolul TCP

Conexiunea de tip end-to-end

- TCP-ul îşi bazează funcţionarea pe un algoritm cu fereastră glisantă.
- Protocoalele cu fereastră glisantă au fost gândite pentru legături punct la punct.
- În cazul TCP-ului se vor stabili conexiuni logice între procese care pot rula pe oricare doua calculatoare din Internet.



Protocolul TCP

Conexiunea de tip end-to-end

- Mecanismul ferestrei glisante garantează un transfer sigur al datelor:
  - mecanism de retransmisie al datelor
  - recepţia pachetelor în ordinea în care au fost transmise
  - multiplexarea traficului
  - permite controlul fluxului între transmiţător şi receptor

- Un alt mecanism esenţial implementat de către TCP este controlul congestiei.

Protocolul TCP

Iniţierea conexiunii:

- Nodul care iniţiază conexiunea, de obicei este de tip *client*, iar cel care răspunde cererii venite de la *client*, este *server-ul*.
- Pentru a iniţia o conexiune este nevoie de trei pachete (three-way handshake):

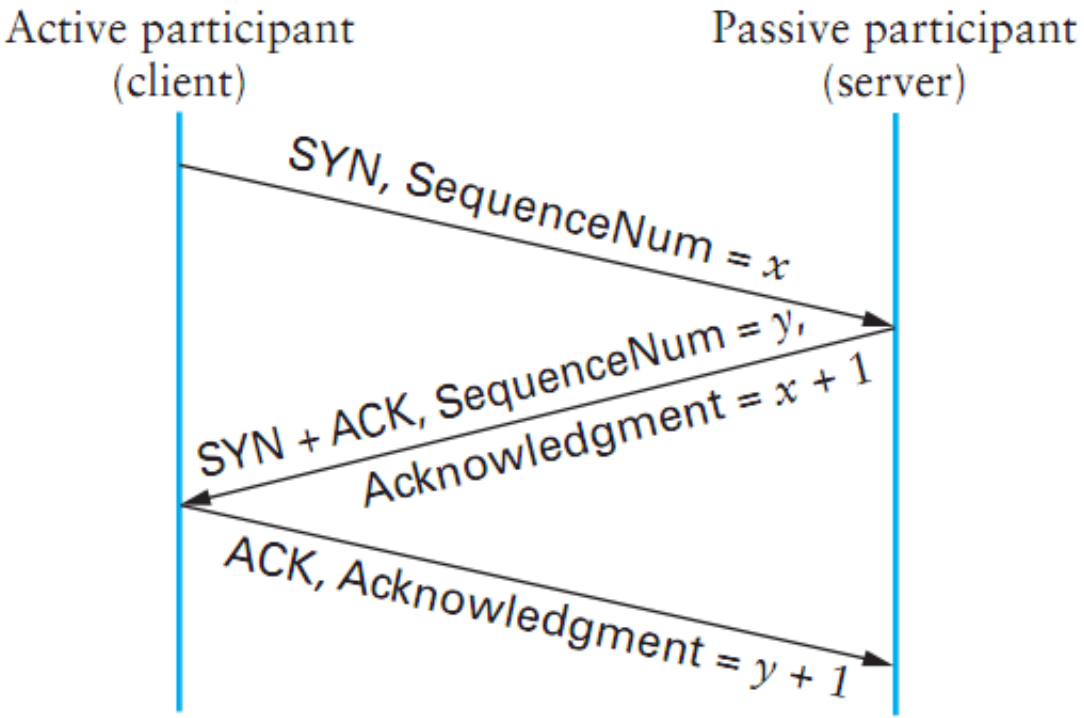
  - Primul pachet vine din partea clientului şi are setat flag-ul SYN. Acest pachet conţine numărul portului pe care clientul îl va folosi pe durata conexiunii, precum şi numărul de secvenţă iniţial.

  - Al doilea pachet implicat în stabilirea conexiunii vine ca răspuns din partea serverului, având setat flag-ul SYN, precum şi flag-ul ACK, fiind un pachet de confirmare pentru primul pachet. El mai conţine, de asemenea şi numărul de secvenţă iniţial folosit de server.

  - Al treilea pachet vine din partea clientului si conţine confirmarea pentru pachetul SYN trimis de server.

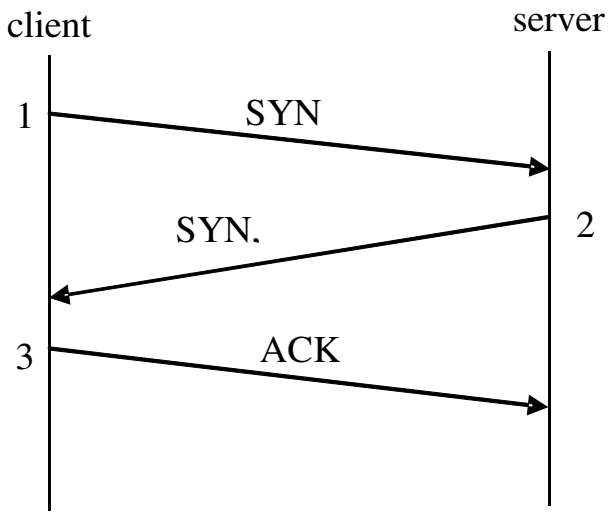
Protocolul TCP

Iniţierea conexiunii:



Protocolul TCP

Iniţierea conexiunii:

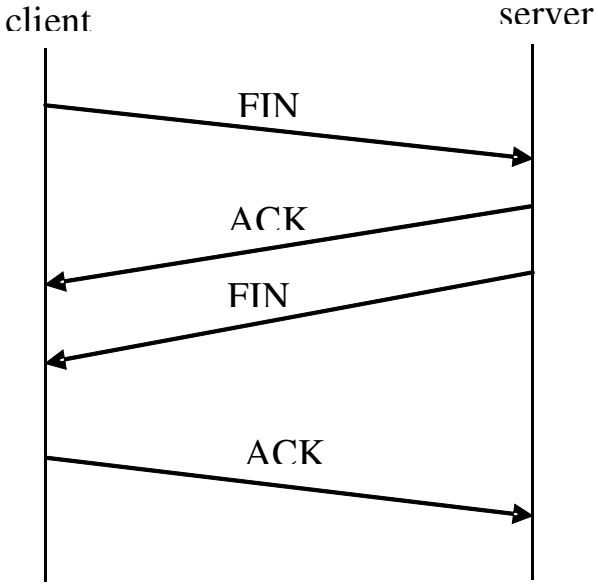


No.	Time	Source	Destination	Protocol	Info
3	0.000309	192.168.0.13	212.112.238.74	TCP	1193 > ftp [SYN] Seq=0 Ack=0 Win=64240 Len=0 MSS=1460
4	0.131235	212.112.238.74	192.168.0.13	TCP	ftp > 1193 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
5	0.131369	192.168.0.13	212.112.238.74	TCP	1193 > ftp [ACK] Seq=1 Ack=1 Win=64240 Len=0

Protocolul TCP

Încheierea unei conexiuni

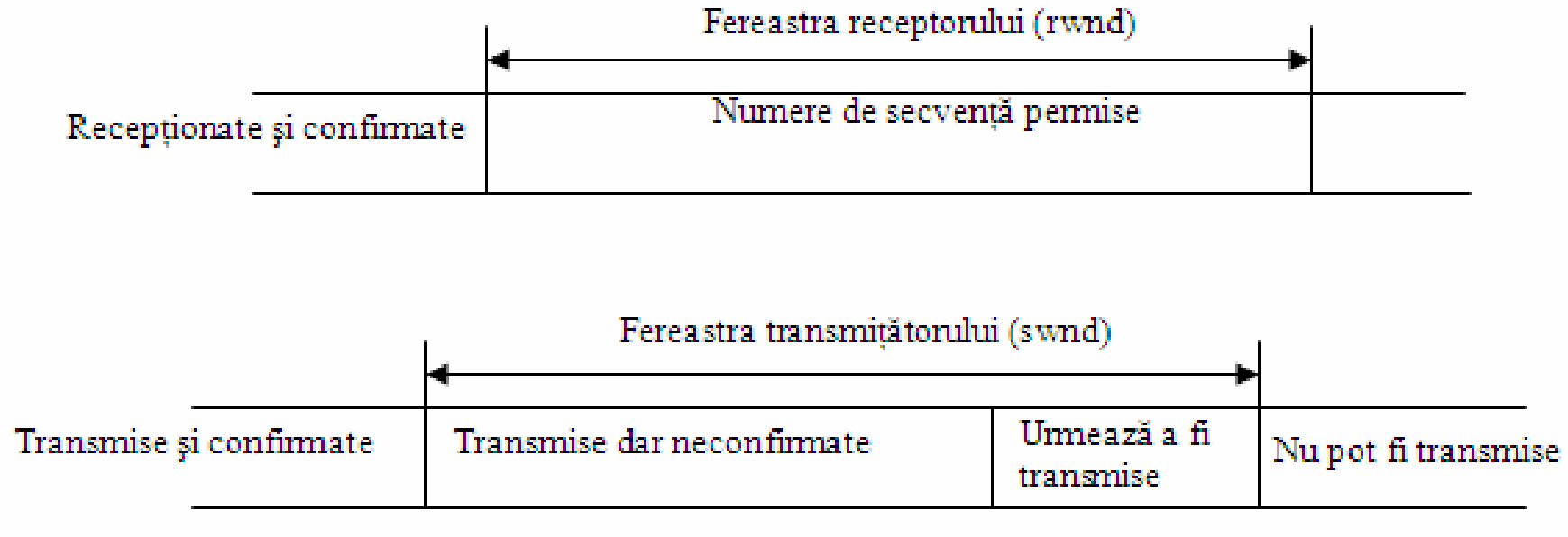
- Încheierea unei conexiuni poate fi făcută în mod bilateral, adică ambele noduri vor trimite câte un pachet de tip FIN, confirmate prin câte un pachet ACK, sau unilateral, când doar unul dintre noduri trimite un pachet FIN care va fi confirmat print-un pachet ACK.



No.	Time	Source	Destination	Protocol	Info
158	24.365722	10.23.3.21	10.23.3.11	TCP	telnet > 1100 [FIN, ACK] Seq=6442 Ack=129 Win=5840 Len=0
159	24.366030	10.23.3.11	10.23.3.21	TCP	1100 > telnet [ACK] Seq=129 Ack=6443 Win=64171
160	24.366613	10.23.3.11	10.23.3.21	TCP	1100 > telnet [FIN, ACK] Seq=129 Ack=6443 Win=64171
161	24.366778	10.23.3.21	10.23.3.11	TCP	telnet > 1100 [ACK] Seq=6443 Ack=130 Win=5840 Len=0

Protocolul TCP

Funcţionarea ferestrei glisante:



Protocolul TCP

Funcţionarea ferestrei glisante:

- Atunci când un pachet este recepţionat se verifică dacă numărul lui de secvenţă coincide cu numărul de secvenţă de la începutul ferestrei receptorului, adică este următorul număr de secvenţă aşteptat.
- Daca numărul de secvenţă nu coincide dar se află în interiorul ferestrei, atunci este introdus în buff-er dar nu este confirmat, iar receptorul trimite un pachet de confirmare care conţine în câmpul ACK aceiaşi valoare cu cea din pachetul de confirmare corespunzător ultimului pachet de date valid.
- Dacă numărul de secvenţă nu este cel aşteptat şi nici nu se află în interiorul ferestrei de recepţie, atunci pachetul este ignorat.
- În momentul când soseşte pachetul aşteptat, atunci acesta este confirmat, iar limita din stânga a ferestrei se deplasează spre dreapta.
- Limita din dreapta a ferestrei se va deplasa spre dreapta doar în momentul în care pachetele care au fost confirmate sunt scoase din buffer pentru a fi procesate.

Protocolul TCP

Retransmisia datelor:

- Retransmisia datelor se realizează fie când pachetele ajung la destinaţie dar sunt afectate de erori, fie când s-au pierdut pe drum.
- Pentru ca procedeul de retransmisie să funcţioneze este nevoie de folosirea unor timere şi a unui mecanism de confirmări pozitive (*positive acknowledgements*).
- Confirmarea pozitivă însemnă că sunt confirmate doar pachetele care au ajuns neafectate de erori.
- Pentru a optimiza mecanismul de retransmisie se foloseşte o metodă cumulativă de confirmări, care permite confirmarea printr-un singur mesaj a unui grup de pachete consecutive.
- Pentru transmisia confirmărilor sunt folosite pachete de date, procedeul purtând denumirea de *piggybacking*. Există şi posibilitatea de a transmite distinct doar pachete de confirmare.

Controlul congestiei

- Atunci când resursele reţelei nu mai reuşesc să facă faţă traficului şi reţeaua devine suprasolicitată, parametrii în care se desfăşoară traficul se degradează tot mai mult, în felul acesta instalându-se congestia. Congestia se manifestă prin întârzieri tot mai mari, printr-un număr mare de pachete pierdute şi în final se poate ajunge la colaps total, adică blocarea reţelei.

- În funcţie de locul unde apare congestia există două categorii de congestie:

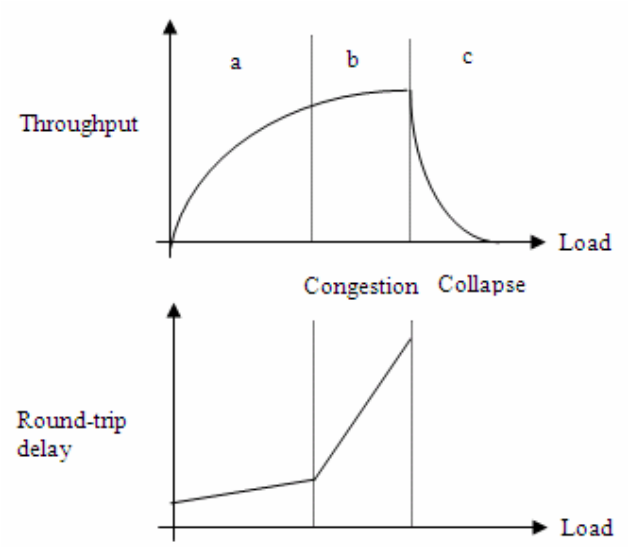
  - Congestie care apare din cauza suprasolicitării server-elor de aplicaţii (Server Side Congestion): apar prea multe cereri din partea clienţilor la un moment dat astfel încât noile cereri pentru conexiuni vor fi respinse.

  - Cealaltă categorie de congestie apare pe partea de client, atunci când mai mulţi clienţi împart în comun aceleaşi conexiune fizice. În acest caz congestia apare în nodurile intermediare care nu mai pot să gestioneze numărul mare de conexiuni care apar la un moment dat.

- Într-o reţea bazată pe comutarea de pachete resursele sunt distribuite la nivelul fiecărui nod din reţea. Aceste resurse pot fi definite prin trei elemente: capacitatea de procesare a informaţiilor, dimensiunea buffer-elor şi capacitatea de transport a liniilor.

Controlul congestiei

- Luând în considerare capacitatea de transport şi întârzierile care apar, instalarea congestiei arată grafic ca în figura de mai jos.
- Se observă că atunci când reţeaua lucrează în parametrii optimi, ea răspunde corect atunci când apare o încărcare mai mare (zona a). La început apare o creştere exponenţială, apoi o zonă în care reţeaua nu mai reacţionează la o creştere suplimentară a încărcării (zona b), pentru ca apoi dintr-un anumit punct capacitatea de transfer să scadă brusc, iar dacă nu sunt luate măsuri, se ajunge la colaps (zona c).
- În ceea ce priveşte întârzierea introdusă de reţea se observă că se păstrează o valoare aproximativ constantă în zona a, apoi pe măsură ce reţeaua începe să fie congestionată (zona b) întârzierea introdusă de reţea devine tot mai mare.



Controlul congestiei

- Pentru controlul congestiei există două abordări şi anume:

  - un control în buclă deschisă

  - un control în buclă închisă (cu *feedback*)

- În primul caz se stabilesc de la început parametrii în care va funcţiona reţeaua, luându-se măsuri în faza de proiectare pentru prevenirea apariţiei problemelor, deoarece controlul se va face fără informaţii despre situaţia concretă de la un moment dat din reţea. Controlul în buclă deschisă se face de obicei la periferia reţelei, prin supravegherea traficului (*traffic policing*) şi prin formarea traficului (*traffic shaping*) pentru nodul care a primit acces la resursele reţelei. Formarea traficului presupune uniformizarea ratei medii de transmisie a datelor.

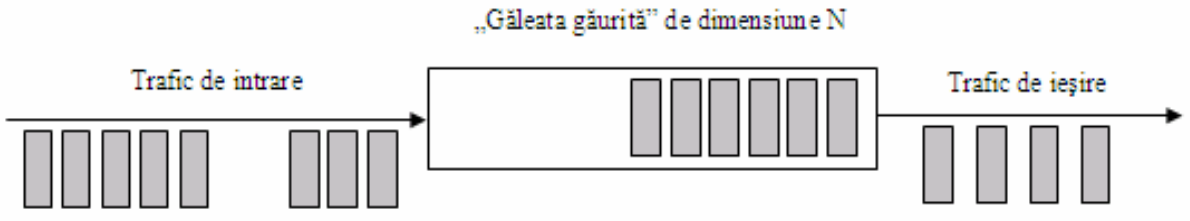
- Al doilea mod de control al congestiei este cel în buclă închisă. Aici măsurile care sunt întreprinse se bazează pe informaţii culese în permanenţă din interiorul reţelei. Informaţiile primite pot fi implicite sau explicite. Un tip de informaţie implicită ar putea fi numărul de pachete pierdute, sau întârzierile din reţea. Informaţiile explicite sunt cele generate în mod special pentru a avertiza despre apariţia congestiei. Aceste informaţii pot fi pachete suplimentare care să conţină date despre congestie sau ar putea fi folosite anumite câmpuri în cadrul pachetelor şi care vor fii setate cu anumite valori atunci când apare congestia.

Controlul congestiei

- Pentru controlul congestiei în buclă deschisă, vom da două exemple de algoritmi pentru formarea traficului:

Algoritmul găleţii găurite

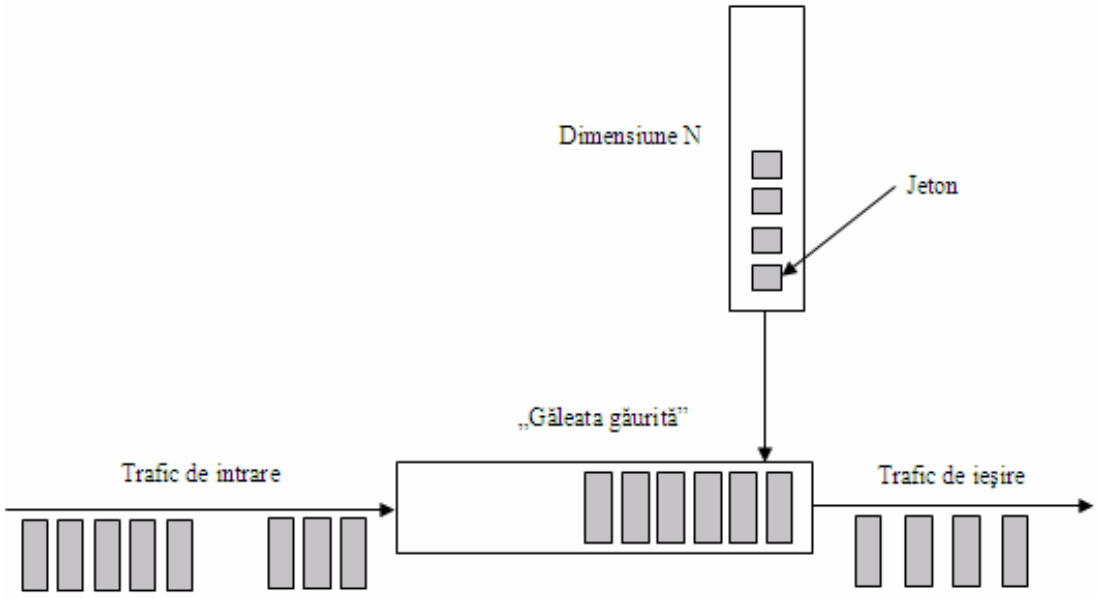
- Algoritmul poartă această denumire deoarece se face analogie cu o găleată care are un orificiu pe fund. Indiferent de debitul cu care apa intră în găleată, ea se va scurge prin orificiu cu un debit constant. La un moment dat, dacă găleata continuă să fie alimentată ea se poate umple şi apa se pierde. Acelaşi principiu poate fi aplicat şi în cazul unei transmisii de date. Pachetele care sunt trimise dintr-un anumit nod în reţea sunt trecute printr-o „găleată găurită”, adică un buffer de tip FIFO care acceptă pachete la orice rată de transfer, dar le transmite mai departe cu o rată fixă. Dacă buffer-ul se umple atunci pachetele care sosesc ulterior se pierd. În felul acesta se realizează în mod implicit şi controlul traficului, adică urmărirea dacă un utilizator depăşeşte parametrii de trafic care i-au fost atribuiţi. Marele avantaj al acestei metode este acela că nu permite traficului în rafală care ar putea veni din partea nodului transmiţător să pătrundă sub această formă în reţea, aceasta fiind principala cauză a apariţiei congestiei, chiar şi atunci când resursele reţelei ar părea ca sunt suficiente.



Controlul congestiei

Algoritmul găleţii cu jeton

- Acest algoritm se aseamănă în mare măsură cu cel anterior, dar spre deosebire de acesta permite flexibilitate în ceea ce priveşte rata traficului de ieşire. În cazul găleţii găurite rata acestui trafic era fixă. Algoritmul funcţionează în felul următor: găleata acumulează jetoane generate cu o rată de un jeton la ΔT secunde. Pentru ca un pachet să poată fi trimis el trebuie să găsească un jeton în găleată, pe care să-l distrugă. Acest mod de abordare a problemei permite ca în momentul în care la intrare avem date în rafală, iar în găleată avem jetoane disponibile, atunci datele vor fi transmise tot în rafală, dar lungimea rafalei are maxim valoarea egală cu numărul de jetoane din găleată.



Controlul congestiei

Controlul în buclă închisă

- Pentru a se evita funcţionarea reţelei în zona de congestie este nevoie să se folosească procedee de monitorizare şi control a congestiei. Când se pune problema controlului congestiei două mecanisme trebuie luate în discuţie: evitarea congestiei (*congestion avoidance*) şi ieşirea din congestie (*congestion recovery*). Aceste mecanisme pot fi implementate pe de o parte la nivelul routerelor, care reprezintă nodurile intermediare, atunci când are loc un transfer de date între sursă şi destinaţie, cât şi la nivelul sursei şi a destinaţiei, adică al celor două noduri între care se desfăşoară transferul. Se spune că în acest caz realizăm un control capăt la capăt.

- Într-o reţea ideală, pentru ca cele două tipuri de control să fie eficiente este nevoie ca pe de o parte reţeaua să ofere feedback, pentru ca resursele să fie folosite în mod eficient, iar pe de altă parte este nevoie ca fluxurile de date să fie protejate unele faţă de altele, în cazul în care unii utilizatori ar avea tendinţa să acapareze mai multe resurse decât cele care le-au fost alocate. Această protejare a fluxurilor de date se poate face prin mecanisme de tip QoS.

- Principalul avantaj al unei reţele bazate pe comutarea de pachete este faptul că toate resursele reţelei vor fi folosite împreună de către toţi utilizatorii, iar reţeaua va încerca să aloce maximul de resurse disponibile fiecărui utilizator în parte. Ceea ce creează probleme este natura impredictibilă şi în rafală a traficului, aceasta putând să conducă la apariţia congestiei . Pentru scurte momente de timp reţeaua poate deveni supraîncărcată şi pentru a se evita intrarea în congestie este necesar ca într-un anume fel, utilizatorii să fie înştiinţaţi de acest lucru şi să treacă la o diminuare a încărcării reţelei, evitându-se astfel apariţia congestiei. Acest mecanism va funcţiona, făcând presupunerea că toţi utilizatorii reţelei vor coopera şi vor lua în considerare semnalele care avertizează asupra apariţiei congestiei. Pentru a se asigura acest lucru au fost implementate mecanisme de evitare a congestiei chiar în protocoalele de comunicaţie în Internet.