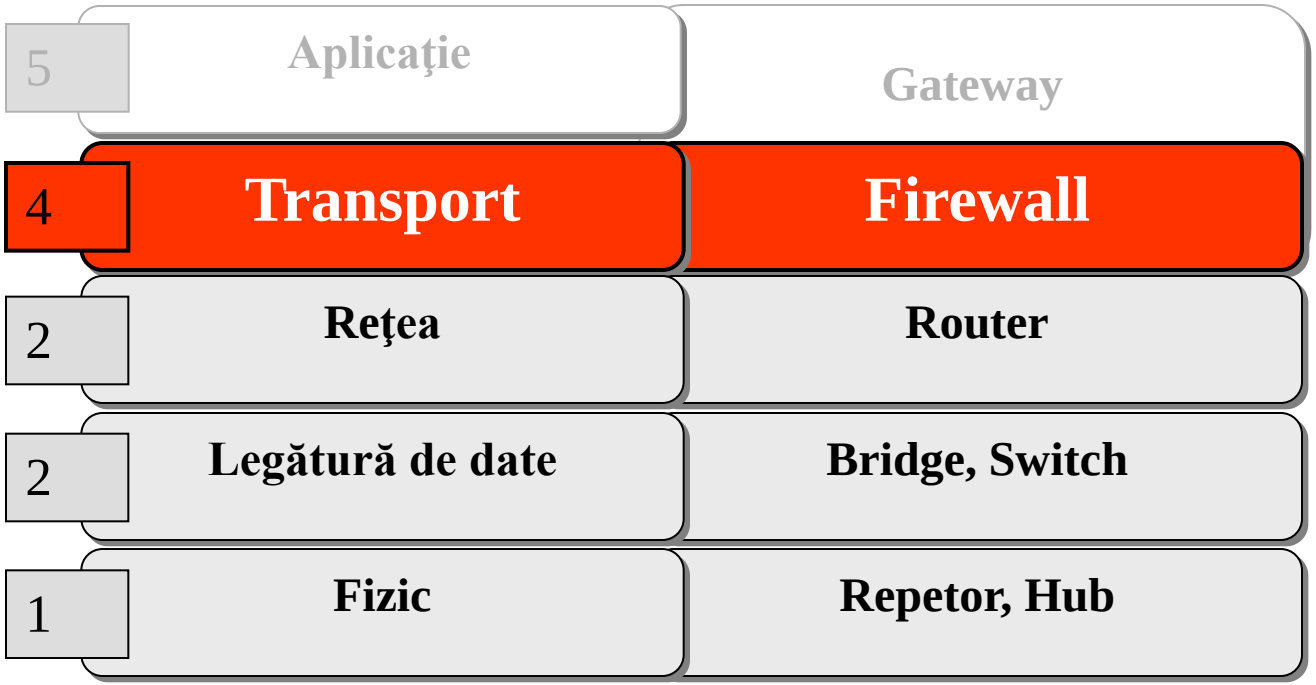
6. Nivelul Transport



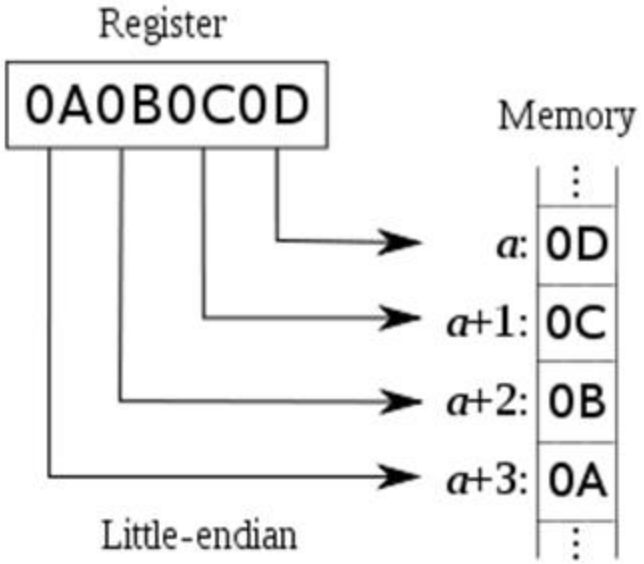
Cuprins

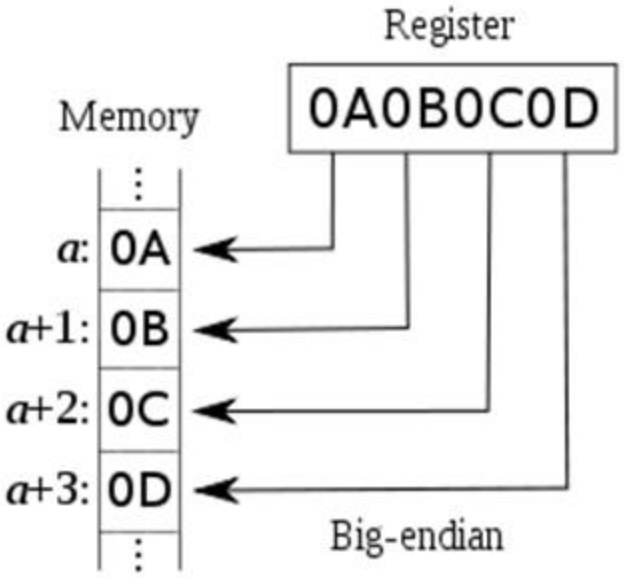
- Ordinea octeților (little și big endian)
- Modele de servicii
- Porturi
- Protocolul TCP
- Protocolul UDP
- Controlul fluxului de date
- NAT
- Firewalls

Ordinea octeţilor

Little Endian
- Intel 80x86
- DEC VAX
- DEC PDP-11

Big Endian/network byte order
- IBM 370
- Motorola 68000
- Sun



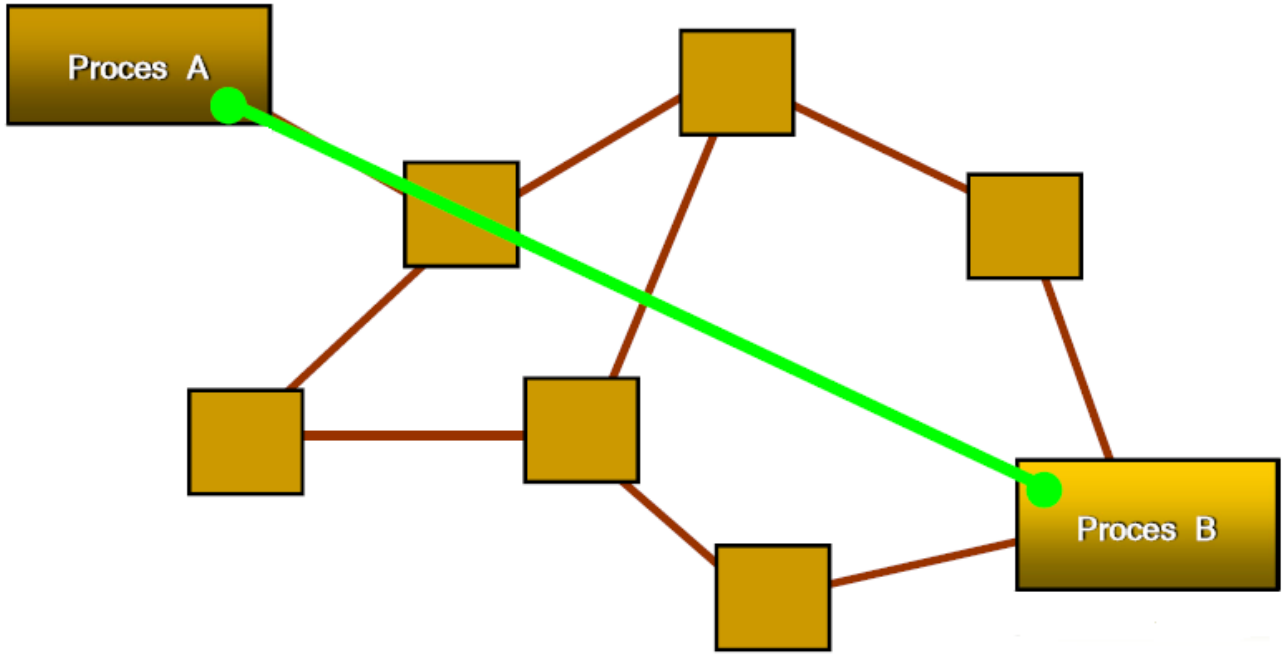


- htonl (host-to-network-long) și htons (host-to-network-short)
- ntohl and ntohs (network-to-host order)

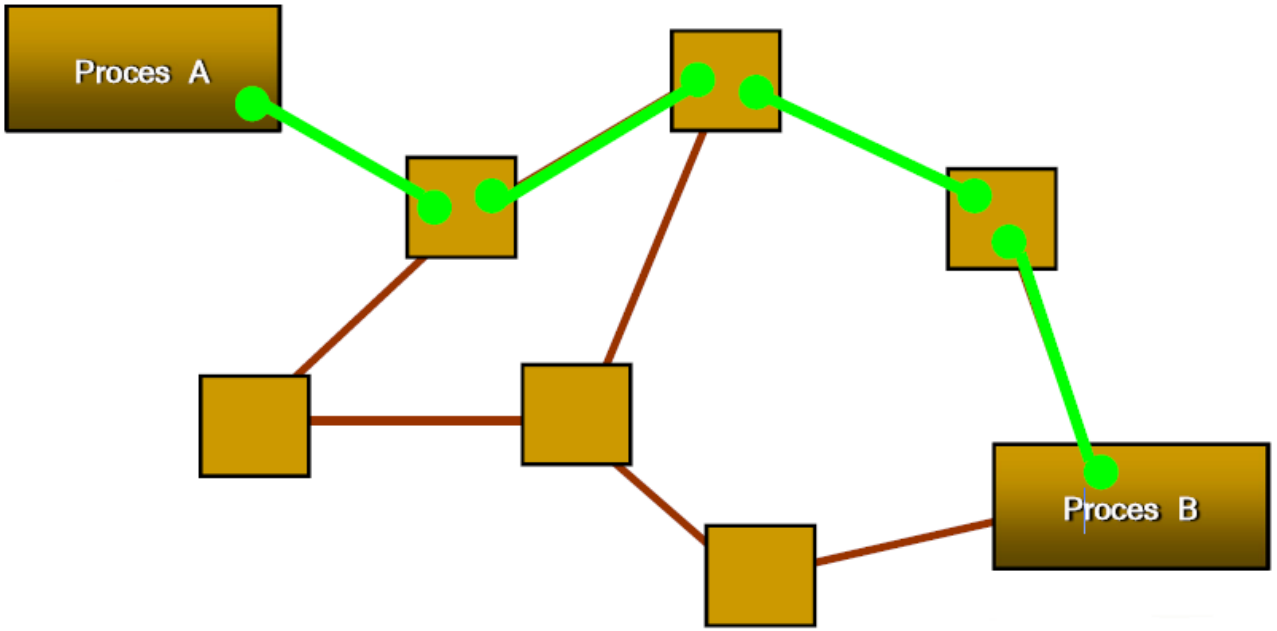
Modele de servicii

- Controlul fluxului/erorilor se pot realiza:
  – între punctele terminale ale comunicaţiei (*end-to-end*)
  – între fiecare 2 noduri ale drumului dintre cele 2 puncte terminale (*hop-by-hop*)
- Comunicaţiile pot fi mai eficiente via *buffer*-e

End-to-end



Hop-by-hop



Nivelul transport

• transmite date de la sistemul sursă la sistemul destinaţie – comunicare *end-to-end*
• asigură transportul datelor de la aplicaţie la aplicaţie
• asigură fluxuri de octeți în mod fiabil (*reliable*), orientat pe conexiune
• oferă servicii mult mai **fiabile** decât nivelul reţea (IP)
  – *e.g.*, pachetele pierdute/incorecte la nivelul reţea pot fi detectate/corectate la nivelul transport
  – Comunicaţii orientate flux de date (*stream*-uri) sau datagrame
  – Conectare prin circuite virtuale
  – Transfer de date via zone tampon (*buffers*)
• unitatea de date pentru transport este TPDU (*Transport Protocol Data Unit*) **adresăIP:port**

Nivelul TRANSPORT este deservit, în principal, de două protocoale UDP (User Datagram Protocol) şi TCP (Transmission Control Protocol)

Protocolul IP se ocupă cu distribuirea datelor între calculatoarele reţelei, pe când TCP şi UDP distribuie datele între aplicaţii în funcţie de porturile asignate.

PORTURI

• corespunzătoare adreselor IP de la nivelul reţea
• se asociază unei aplicaţii (serviciu) şi nu unei gazde
• un proces poate oferi mai multe servicii (poate utiliza mai multe porturi)
• un serviciu poate corespunde la mai multe procese

În Internet, fiecărui protocol îi este asociat un număr de port.

Protocol Internet	Port	Protocol Internet	Port
Echo	7	BOOTPS	67
Discard	9	BOOTPC	68
File Transfer Protocol (FTP)	21	Trivial File Transfer Protocol (TFTP)	69
Telnet Protocol	23	Finger Protocol	79
Simple Mail Transfer Protocol (SMTP)	25	HyperText Transfer Protocol (HTTP)	80
Time of Day	37	Network Time Protocol (NTP)	123
DNS (Domain Name System)	53		

Asignarea porturilor (0-65535):
- numere mai mici de 255 – aplicaţii publice
- numere între 255 si 1023 – aplicaţii comerciale
- numere mai mari de 1023 – porturi disponibile

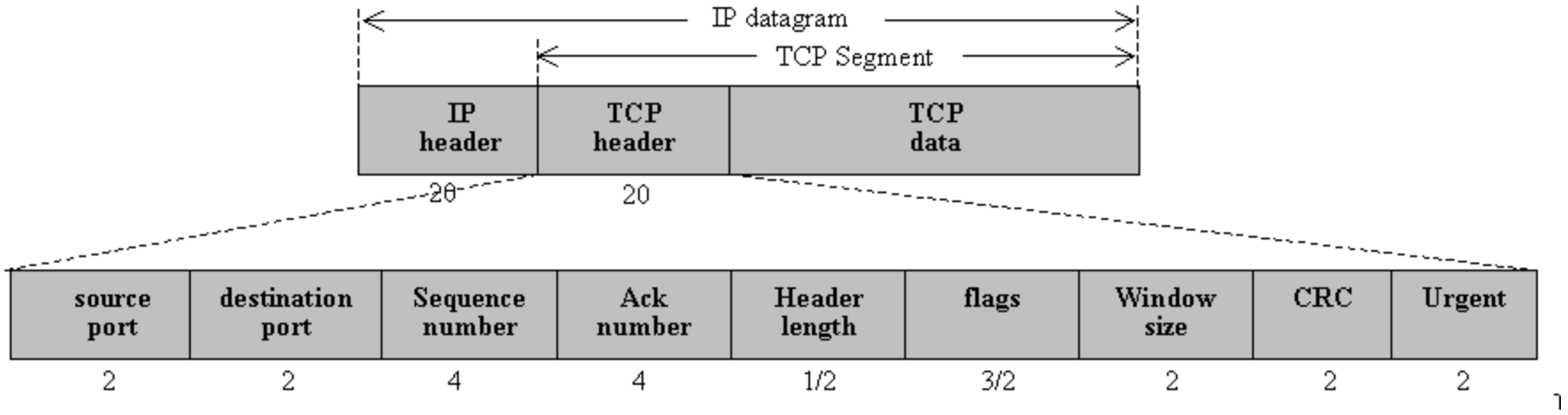
Protocoalele TCP şi UDP utilizează porturile în mod diferit:
- UDP fiind un protocol fără conexiune, lasă pur şi simplu datele pe port.
- TCP se concentrează pe conexiune – nu pe port.

Prin urmare, aplicaţiile care folosesc TCP pot să deschidă pe acelaşi port mai multe conexiuni fără să apară probleme de transmisie.

TCP

- servicii orientate pe conexiune, fiabile
- vizează oferirea calităţii maxime a serviciilor
- integreaza mecanisme de stabilire si de eliberare a conexiunii
- controleaza fluxul de date (*stream oriented*)
- utilizat de majoritatea protocoalelor de aplicaţii: TELNET, SMTP, HTTP,…

Antetul TCP



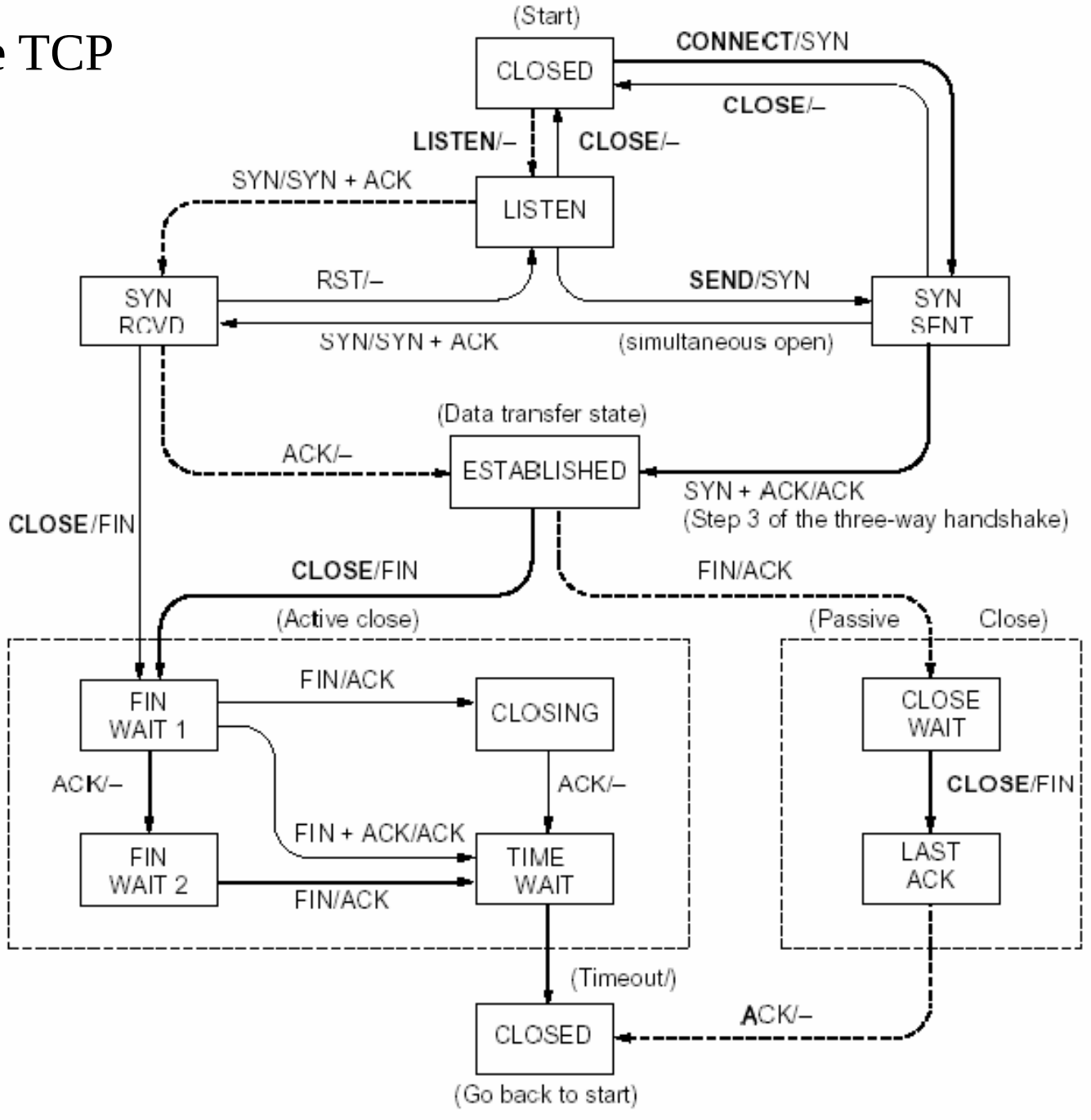
flags:

- URG – indicator de urgenţă
- ACK – numărul de confirmare este valid
- PSH – forţează partenerul de a răspunde imediat
- RST – este 1 când se refuză un segment sau o conexiune
- SYN – este 1 când se cere sau se acceptă o conexiune
- FIN – folosit pentru închiderea conexiunii

Diagrama de stare TCP

—— client

------ server



– Stabilirea conexiunii:

CLOSED – din această stare se poate cere o deschidere activă (se trece în **SYN_SENT**) sau pasivă (**SYN_RCVD**)

LISTEN – se poate trimite o cerere de conexiune activă (se trece în **SYN_SENT**) ori pasivă (**SYN_RCVD**)

– Conexiune stabilită:

ESTABLISHED – poate începe transmisia de date (din această stare se poate trece în **CLOSE_WAIT** sau **FIN_WAIT_1**)

– Deconectare iniţiată de procesul partener

CLOSE_WAIT, **LAST_ACK**, **CLOSE**

– Stări ce intervin în procesul de deconectare

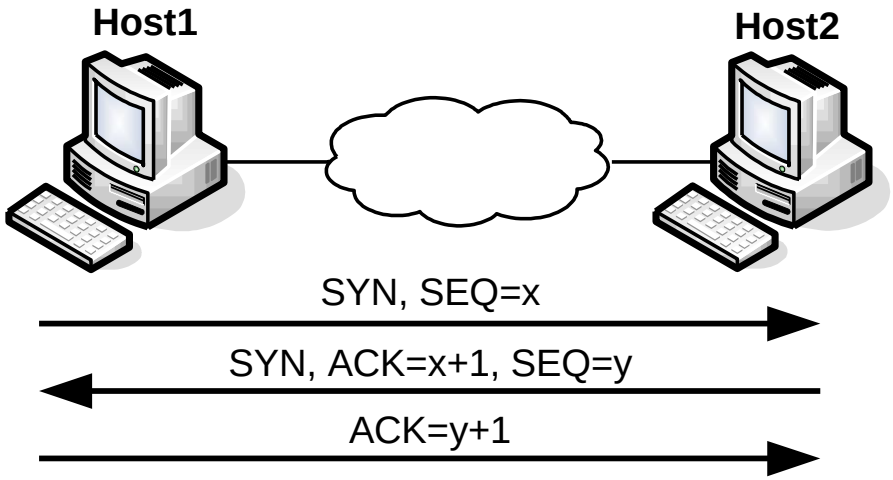
FIN_WAIT_1, **FIN_WAIT_2**, **CLOSING**, **TIME_WAIT**

Vizualizarea conexiunilor: **netstat -a**

Controlul Fluxului

- Se utilizează bitul ACK la nivelul comunicării duplex
  - **tree-way handshake**, care sincronizează sistemele sursă şi destinaţie pentru transferul de date. Deoarece datagramele se pot pierde, fiecare datagramă este confirmată la recepţie.
- **Fereastra glisantă** (***sliding window***)
- La transmitere, datele se acumuleaza într-un *buffer* cu 3 pointeri
  - Mărimea ferestrei depinde de numărul secvenţei de confirmare
- La receptie, *buffer*-ul conţine 3 pointeri
  - Datele primite şi confirmate
  - Datele care pot fi primite
  - Datele care nu pot fi încă primite

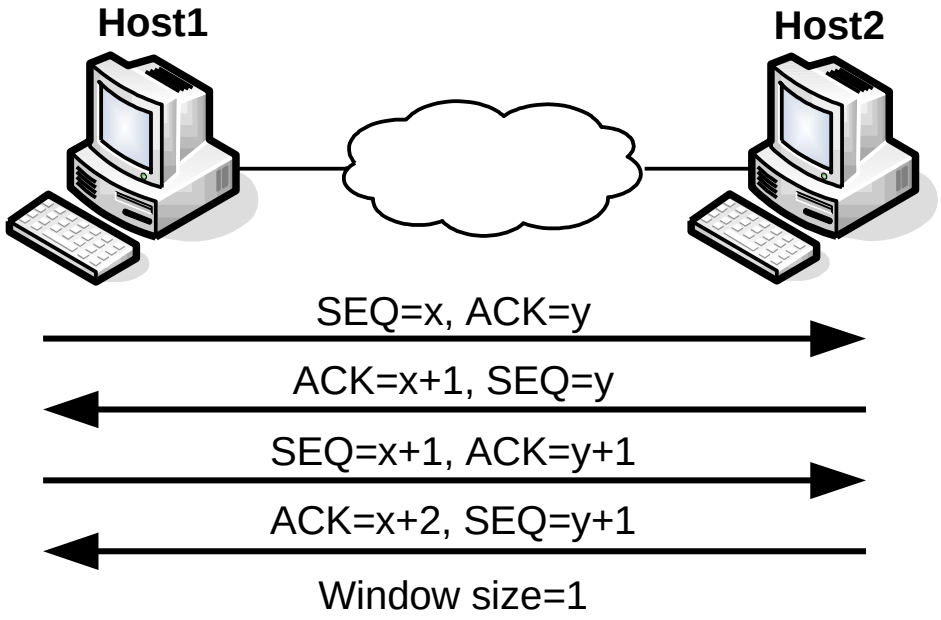
TCP tree-way handshake/Open Connection



Hostul 1 iniţiază o conexiune trimiţând un pachet cu SYN=1 (cerere de conexiune) şi numărul de secvenţă SEQ=x.

Hostul 2 recepţionează pachetul cu SEQ=x, răspunde cu un mesaj de confirmare ACK=x+1, incluzând numărul de secvenţă proprie SEQ=y. ACK=x+1 înseamnă că au fost recepţionate toţi octeţii incluzând x şi se aşteaptă pachetul x+1. Pentru închiderea conexiunii se foloseşte aceeaşi metodă, dar se setează flag-ul FIN în loc de SYN.

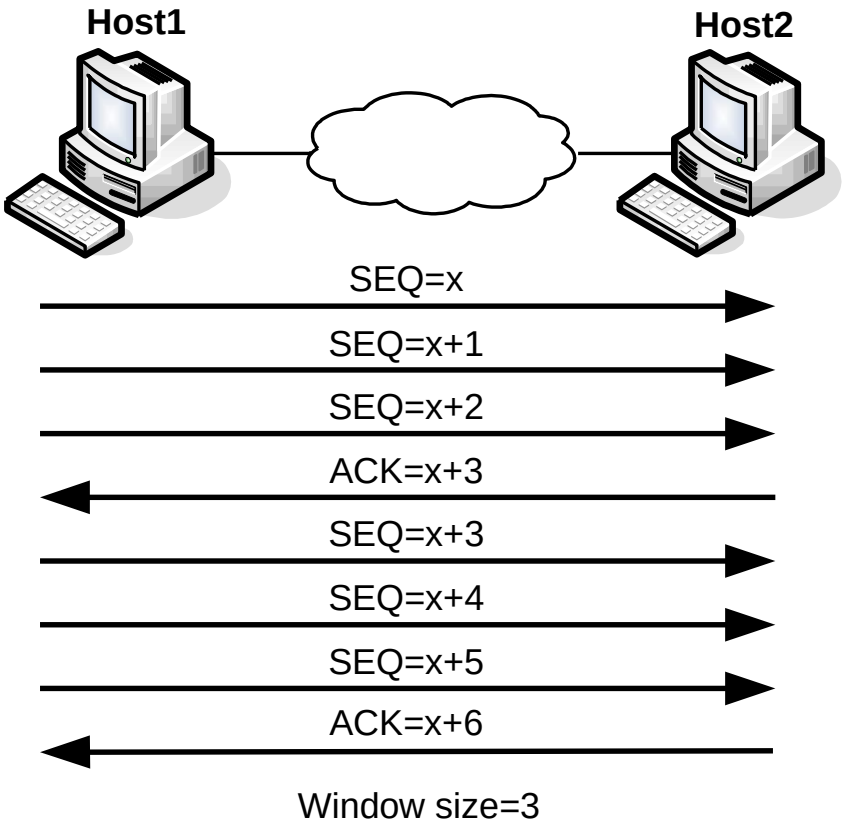
PAR (Positive Acknowledgment Retransmission)



Multe protocoale pentru siguranţa transmisiei datelor folosesc PAR, care constă în transmiterea de către receptor al unui mesaj ACK în cazul recepţionării unui pachet.

Transmiţătorul odată cu transmiterea pachetului porneşte un timer. Dacă transmiţătorul nu primeşte acest pachet ACK până la expirarea timpului stabilit, se retransmite segmentul şi se reporneşte timer-ul. Window size determină numărul de octeţi care se pot recepţiona.

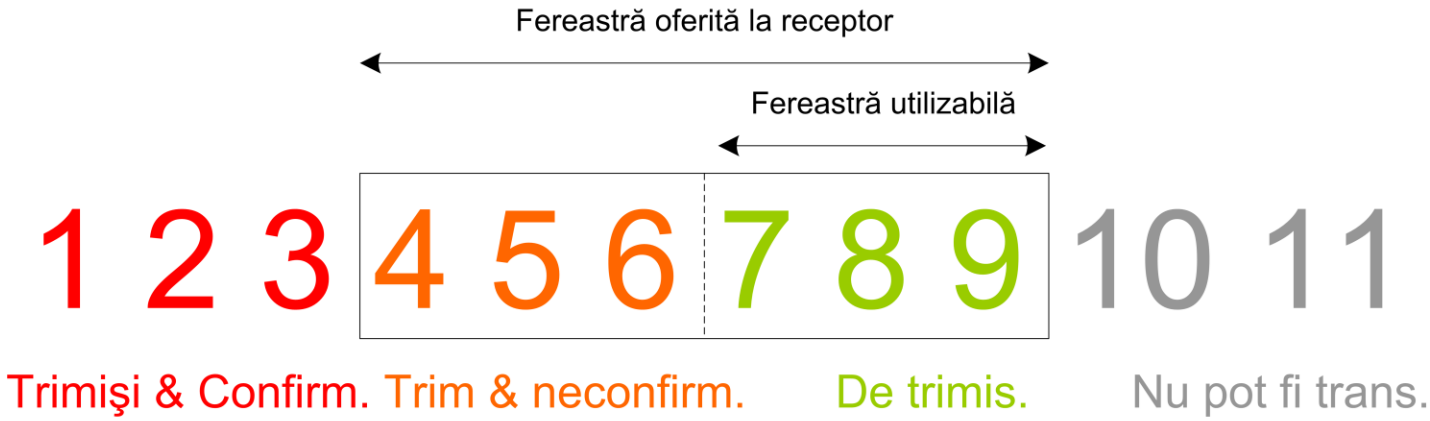
TCP Sliding Window



Metoda Sliding Window se bazează pe recepţionarea mesajului de confirmare după trimiterea unui anumit număr de octeţi. De exemplu, dacă window size este 3, după transmiterea a trei octeţi, transmiţătorul aşteaptă un mesaj de confirmare, după care se transmit următorii octeţi. Dacă nu se primeşte confirmarea se retransmit datagramele.

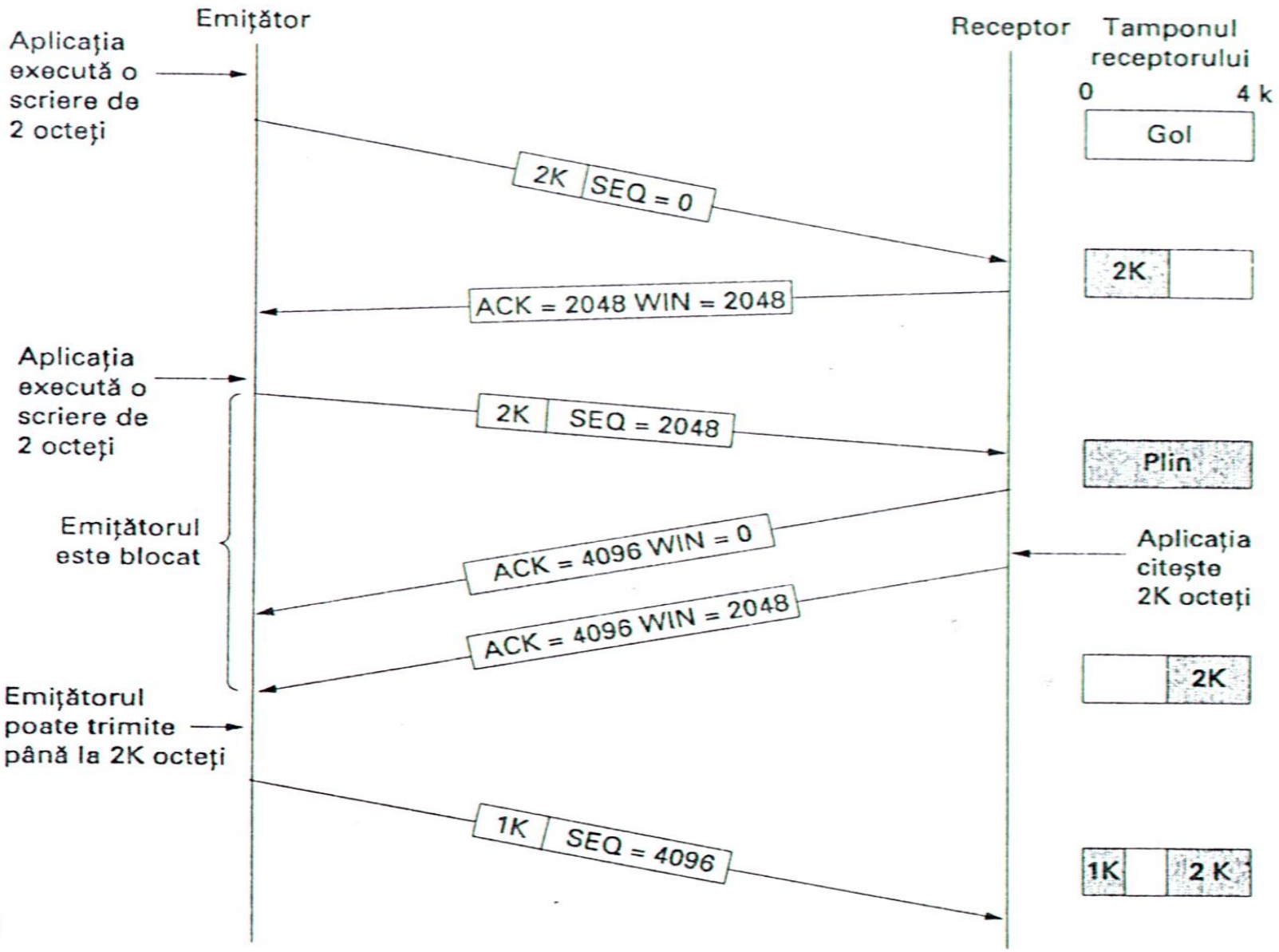
Fereastra glisantă

- 1 2 3 transmişi şi confirmaţi
- 4 5 6 transmişi şi neconfirmaţi
- 7 8 9 pot fi transmişi cât de curând posibil
- 10 11 nu se pot transmite fără mutarea ferestrei



Fereastra glisantă la transmiterea datelor

Managementul ferestrei în TCP



Detecţia erorilor & retransmiterea datelor

– Fiecare segment trimis conţine un numar de secvenţă (*Sequence Number*) indicând poziţia octeţilor transmişi în cadrul fluxului de date

– Gazda destinatar verifică numărul de secvenţă pentru fiecare segment (se testează dacă anumite segmente se pierd, sunt duplicate sau nu sunt în ordine) şi trimite înapoi pentru fiecare segment un număr de confirmare (*Acknowledgment Number*), specificând numărul de secvenţă al următorului octet care se aşteaptă a fi recepţionat

– Segmentele pierdute sunt detectate folosindu-se un *timer* de retransmisie a datelor

– Pentru detectarea erorilor se utilizează şi *checksum*-uri

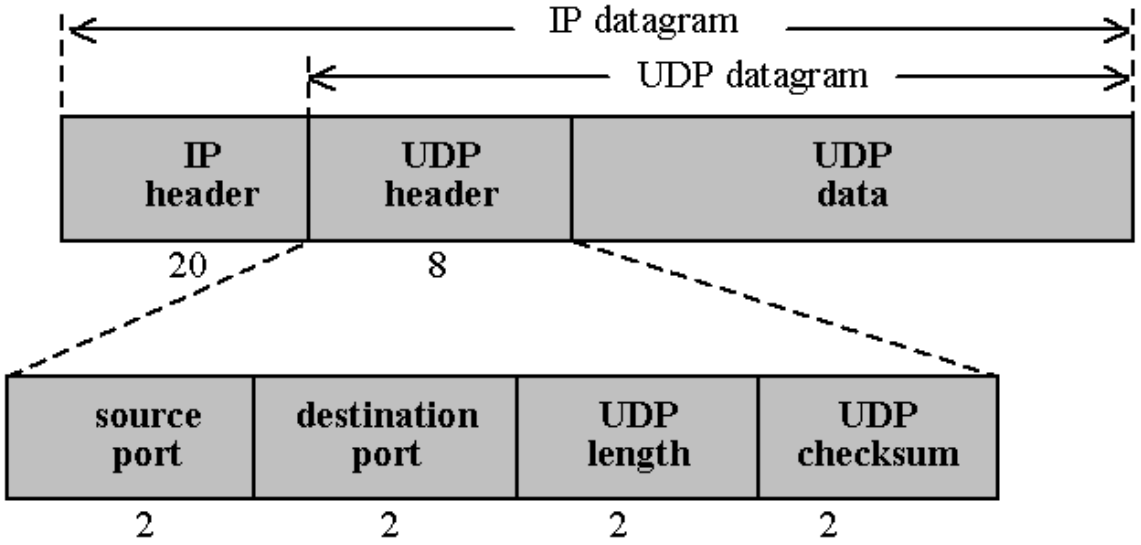
UDP

- este definit în RFC768
- servicii neorientate pe conexiune, nesigure
- nu oferă nici o calitate suplimentară a serviciilor
- nu recurge la negocieri sau la confirmări ale primirii datelor
- utilizat la apelul procedurilor la distanţă via RPC (*Remote Procedure Call*)
- ca şi TCP, pentru a oferi servicii de comunicare între procese foloseşte porturi
- porturile TCP sunt independente de porturile UDP

Antetul UDP

- UDP a fost proiectat în 1980 şi definit în RFC 768.

- Încapsularea UDP:



port – identifică procesele transmiţător şi receptor.

UDP length – este lungimea header şi date UDP în bytes

UDP checksum - nu este obligatoriu

Protocoale care folosesc UDP:

- TFTP (Trivial File Transport Protocol)
- SNMP (Simple Network Management Protocol)
- DHCP (Dynamic Host Configuration Protocol)
- DNS (Domain Name System)
- RIP (Routing Information Protocol)
- Aplicaţii streaming media IPTV, VoIP
- Jocuri online

UDP flood attack

- este un atac de tip denial of service (DoS)

- constă în trimiterea unui număr mare de pachete UDP la porturi generate random. Ca rezultat, sistemul atacat va:

    - verifică dacă vreo aplicaţie ascultă portul respectiv

    - constată că nici o aplicaţie nu ascultă portul respectiv

    - trimite înapoi un pachet ICMP Destination Unreachable

- sistemul atacat va fi forţat să trimită multe pachete ICMP nefiind disponibil pentru alte cereri.

- atacatorul poate ascunde adresa sursă, astfel încât răspunsul ICMP să nu ajungă la atacator (atacator anonym).

TCP vs. UDP

- ambele se bazează pe IP, utilizează porturi
- unitatea de transmisie se numeşte:
  - segment TCP
  - pachet UDP

	IP	UDP	TCP
Orientare pe conexiune	nu	nu	da
Limitarea lungimii mesajului	da	da	nu
Checksum la date	nu	da	da
Răspuns la validare	nu	nu	da
Timeout şi retransmitere	nu	nu	da
Detectarea pachetelor duplicate	nu	nu	da
Secvenţiere	nu	nu	da
Controlul fluxului de date	nu	nu	da

NAT (Network Address Translation)

NAT oferă posibilitatea schimbării unei adrese IP cu o altă adresă din antetul unui pachet IP. În practică NAT se foloseşte pentru a permite staţiilor ce utilizează adrese IP private să acceseze Internetul.

Astfel pentru un grup (o reţea locală) de calculatoare se folosesc un număr de adrese IP de obicei inferior numărului de calculatoare care doresc să acceseze Internetul.

Adrese private (vezi Cap. Nivelul rețea. Adrese IP)
10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16

NAT (cont)

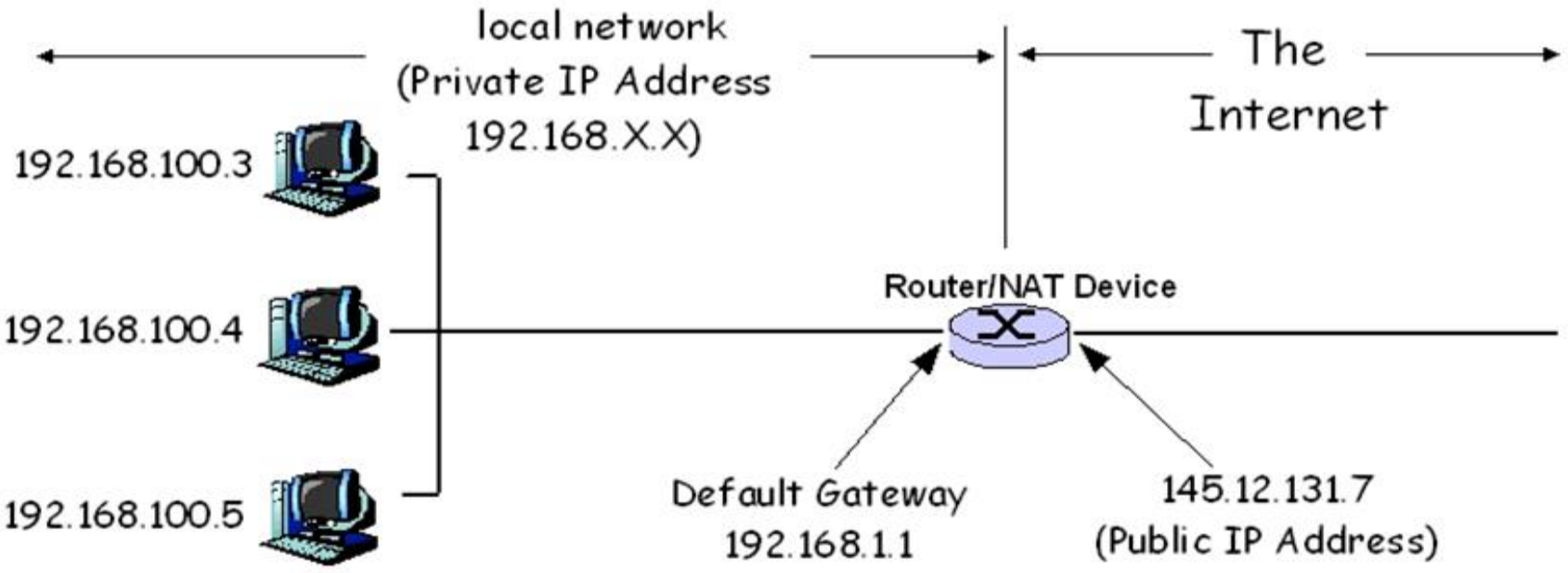
Fie *i* numarul de adrese care sunt alocate calculatoarelor din LAN şi *e* numărul de adrese IP externe (rutabile).

- Translatarea statică – maparea între adresele locale şi cele externe este de 1:1. Ruterul NAT face maparea pe baza unei tabele interne. Cazul e >= i.

- Translatarea dinamică – translatarea se face între o adresă internă şi prima adresă externă liberă. Cazul i > e.

- Translatarea adreselor cu supraîncărcare (masquerading) – cazul e = 1. Se folosesc porturi diferite pentru fiecare conexiune iniţiată. Pachetele se transmit pe porturi diferite, deci vor fi recepţionate pe porturi diferite, putând fi astfel identificat calculatorul de către serverul NAT căruia i se adresează acel pachet.

Ideea de bază din spatele NAT este de a atribui o singură adresă IP unui dispozitiv NAT. Vom numi aceasta adresa IP publică. În cadrul rețelei locale din spatele dispozitivului NAT, fiecărui dispozitiv de calcul i se atribuie o adresă IP privată, așa cum este ilustrat mai jos:



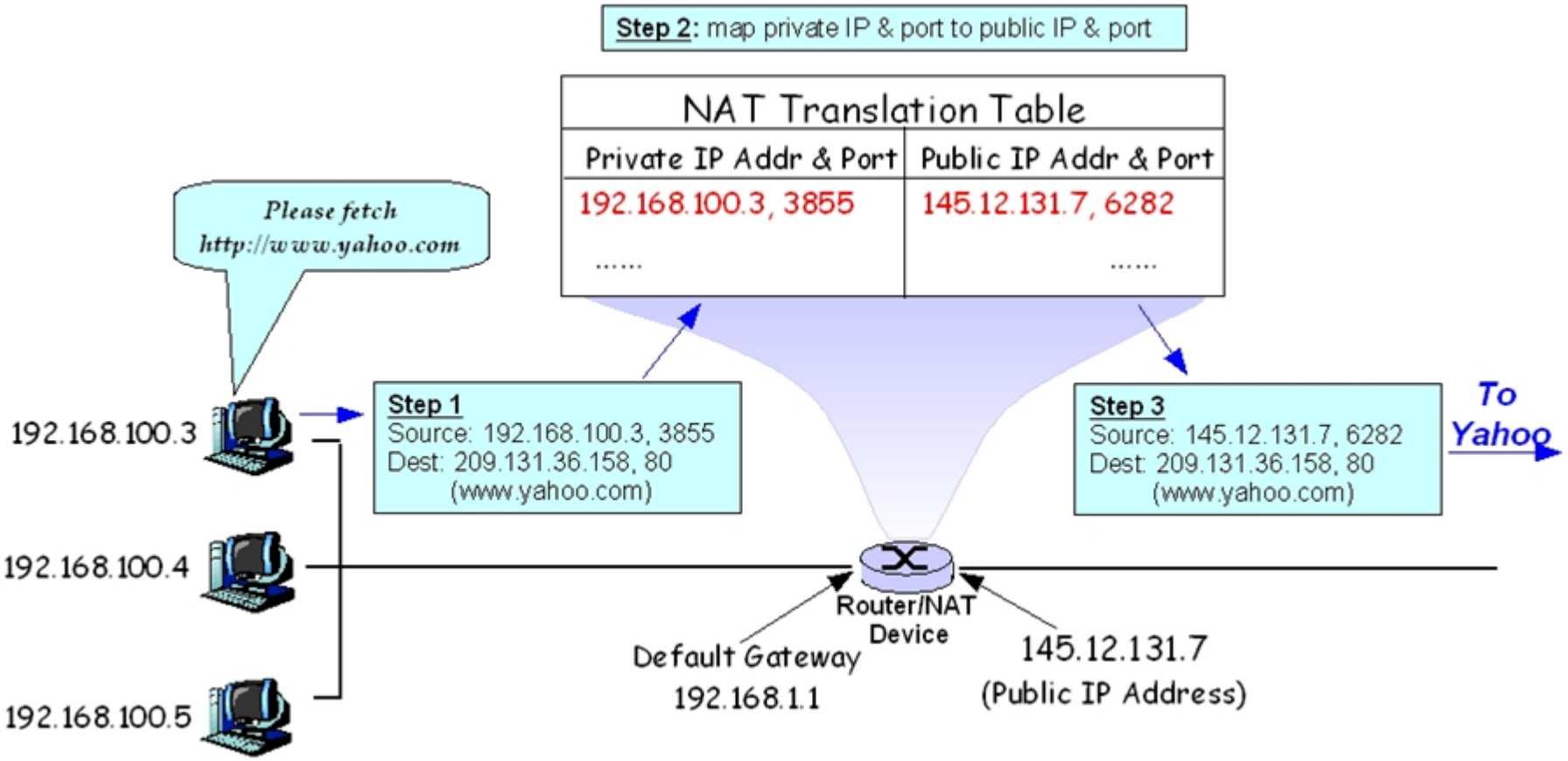
La pasul 1, gazda la adresa IP privată 192.168.100.3 solicită pagina de pornire a www.yahoo.com printr-o solicitare HTTP prin portul 3855.

Când pachetul HTTP ajunge la dispozitivul NAT (pasul 2), acesta caută în tabelul de translatare o intrare publică existentă (adresă IP, port) pentru această combinație privată (adresă IP, port). Dacă nu există nicio intrare existentă, atunci dispozitivul NAT va crea o nouă intrare publică (adresă IP, port). Dacă există o intrare existentă, atunci procesul de traducere va folosi intrarea existentă.

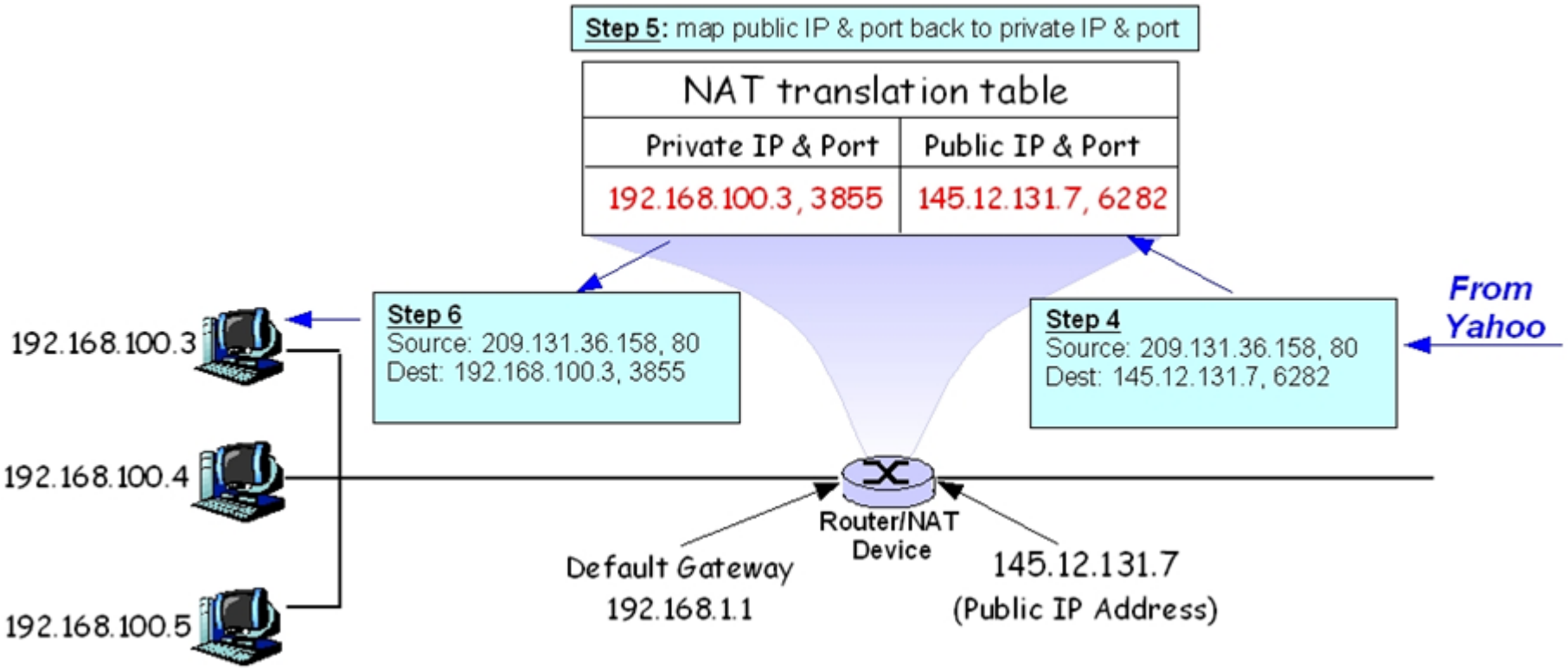
Fiecare intrare din tabelul de translatare trebuie să rămână întotdeauna unică! După ce căutarea în tabel este completă, pachetul IP este apoi modificat astfel încât noua adresă IP și portul să le înlocuiască pe cele vechi. În cele din urmă, la pasul 3, pachetul modificat este direcționat către www.yahoo.com.

Acest întreg proces de traducere a adreselor de rețea este complet transparent pentru gazdele finale. Cu alte cuvinte, nici gazda de la 192.168.100.3, nici serverul web Yahoo nu realizează că pachetul a fost schimbat.

Adresele IP private sunt valide numai în rețeaua locală. Nu este recunoscut pe internetul public. Pentru pachetele care provin dintr-o adresă IP și un port privat, acestea trebuie convertite într-o adresă și un port IP public unic înainte de a putea fi trimise pe Internet. Maparea de la adresa IP privată și un port la o adresă IP și un port public se face de obicei printr-un tabel de translatare în interiorul dispozitivului NAT. Un exemplu este prezentat mai jos:

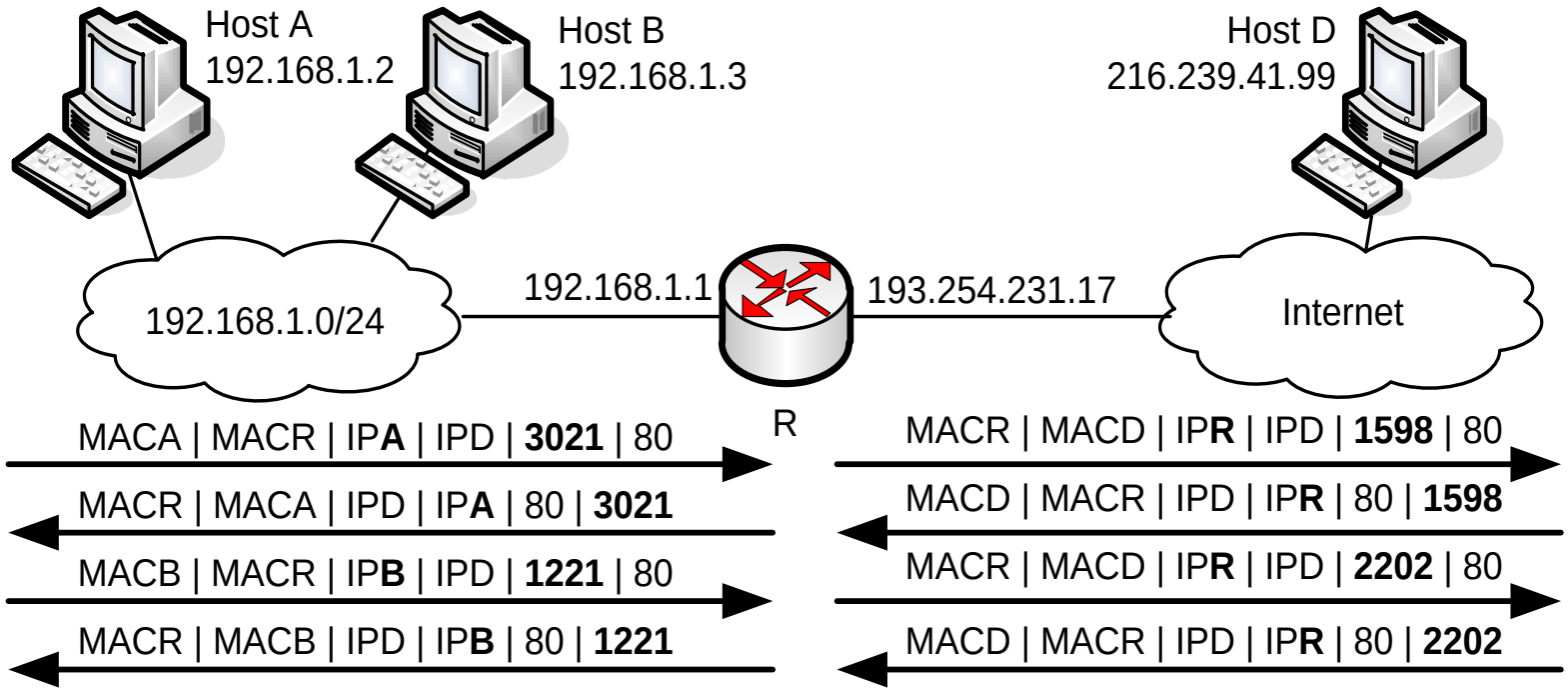


Procesul invers este similar cu procesul de translatare originală. Acesta va căuta în tabel perechea privată corespunzătoare (adresă IP, port) atunci când i se oferă perechea publică (adresă IP, port). Singura diferență este că o intrare lipsă va duce la aruncarea pachetului. Odată ce căutarea și modificarea sunt finalizate (pasul 5), pachetul (acum conține informațiile private originale (adresă IP, port)) este trimis gazdei solicitante la 192.168.100.3 portul 3855.



Funcţionare NAT

Host A şi B din reţeaua locală (privată) vor să acceseze Host D pe portul 80 de pe Internet.



NAT(cont)

Avantajele NAT:

- oferă o schemă de adresare rapidă şi comodă.

- deoarece adresele staţiilor nu sunt accesibile din afara reţelei, NAT este una dintre cele mai eficiente politici de securitate.

Dezavantajele NAT:

- ruterul prin care reţeaua privată accesează Internetul va trebui să fie capabil să facă conversia adreselor private în adrese publice, deci să ruleze un serviciu de NAT.

- NAT impune o latenţă suplimentară pentru fiecare pachet ce tranzitează ruterul.

- în interiorul unei reţele private nu pot fi plasate calculatoare ce oferă servicii publice, deoarece este imposibil de iniţiat conexiuni din exterior către acestea.

Port forwarding

Translatarea permanentă a unui port pe gateway-ul rețelei către o adresă IP și un port din rețeaua privată se numește **Port Forwarding** sau **Port Mapping**. Practic deschidem un port în router pentru a permite accesul către un server (de exemplu http sau ftp) aflat în spatele unui firewall.
La unele rutere aceste setări se găsesc la secțiunea Virtual Server.

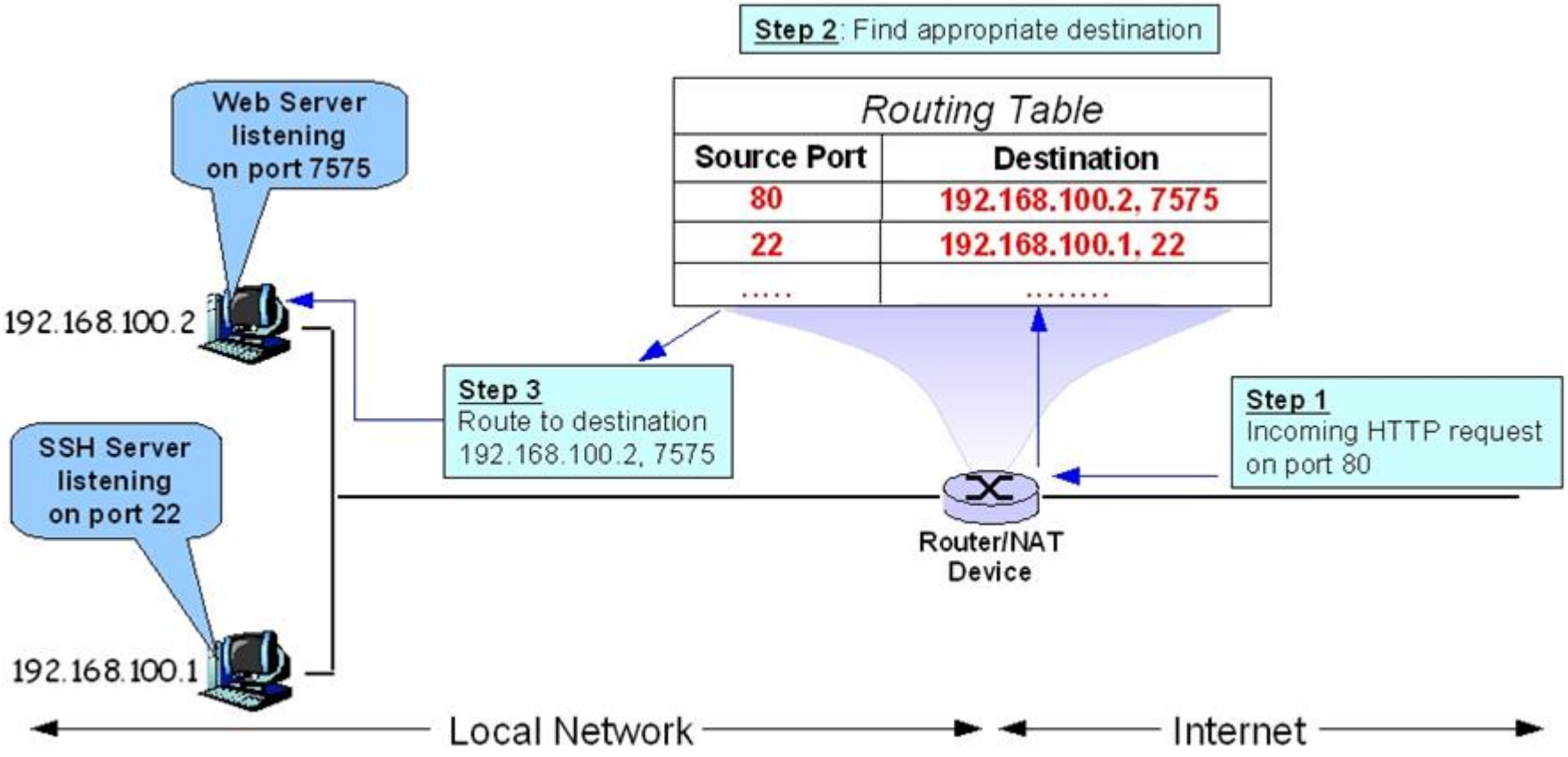
Port triggering

Anumite aplicații (jocuri, video conferințe) folosesc conexiuni multiple și nu vor funcționa pe un ruter uzual.
După configurare, secvența de operare este:
1. host-ul local menține o conexiune pe portul specificat la "triggered port" cu un host extern
2. ruterul înregistrează această conexiune și deschide portul/porturile asociate pentru host-ul respectiv.

3. host-ul extern va putea conecta la host-ul local pe unul dintre porturile specificate la "forwarded port"

Port Address Translation (PAT)

PAT permite sesiunilor de intrare, care sunt inițiate de la o gazdă externă, să se mapeze la o anumită gazdă și port intern.



- toate cererile de intrare către portul 80 al routerului sunt redirecționate către gazda internă 192.168.100.2 portul 7575. De asemenea, toate conexiunile de intrare către portul 22 sau routerul sunt redirecționate către gazda 192.168.100.1 portul 22. Acest tip de configurare este obișnuit pentru utilizatori care doresc să ruleze un server în spatele unui dispozitiv NAT. Singurul dezavantaj al PAT este că este limitat la o singură intrare per port de router.

Rezumat:

Network Address Translation (NAT) este o soluție utilizată pe scară largă pentru lipsa de adrese IP. NAT introduce conceptul de adresă IP „privată” care este valabilă numai într-o rețea locală (LAN) și trebuie tradusă la adresa IP „publică” care este utilizată pe Internet. Cu NAT, putem avea mai multe adrese IP private care partajează o singură adresă IP publică, întârziind astfel nevoia de a implementa soluții pe termen lung pentru lipsa de adrese IP.

Firewalls

Sistemele conectate la Internet sunt supuse la următoarele tipuri de pericole:
- pericolul **scurgerii de informaţii confidenţiale** sub formă electronică
- pericolul **infiltrării de informaţii nedorite** (viruşi, hackeri)
- pericolul unui atac de tip **DoS** (Denial of Service), în care calculatorul este supus unui bombardament cu cereri de informaţie şi pentru a răspunde, ajunge la limită în privinţa resurselor.
- atac de tip **flood**
- **găurile de securitate** din sistemul de operare sunt portiţe de intrare în sistem.

Un firewall este o componentă hardware sau software (sau o combinaţie a lor) care funcţionează într-un mediu de reţea pentru a împiedica anumite tipuri de comunicare; această reducere a posibilităţilor de comunicare este stabilită de o politică de securitate, în conformitate cu scopurile propuse pentru reţeaua respectivă.

Scopul principal al unui firewall este împiedicarea propagării ameninţărilor din Internet înspre reţeaua proprie.

Tipuri de firewall

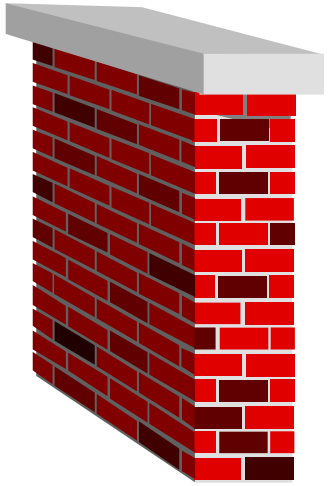
- Layer 2 (MAC) şi 3 (datagram) – filtru de pachete

- Layer 4 (transport) – filtru de pachete, poate face diferenţa între protocoale de transport (statefull firewall)

- Layer 5 (aplicaţie) – se comportă ca un server proxy pentru diferite protocoale, analizând şi luând decizii pe baza cunoştiinţelor despre aplicaţii şi pe baza conexiunilor.

Packet filtering firewalls

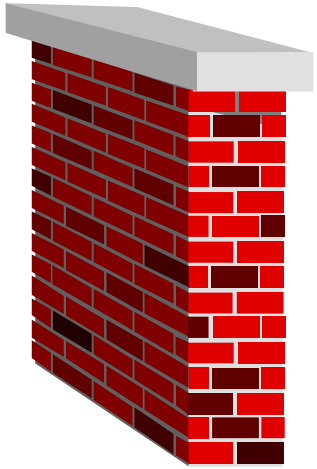
- pachetele se filtrează pe baza:
  - adreselor IP sursă şi destinaţie
  - câmpul de protocol
  - numărul portului sursă şi destinaţie
  - setările flag-ului SYN
  - informația din pachetul de date
- filtrarea pachetelor se face pe baza unor reguli de tipul (permite sau interzice)



- nu se ţine cont de starea conexiunii
- sunt susceptibile de atacuri de la nivelul aplicaţie

Statefull packet filtering firewalls

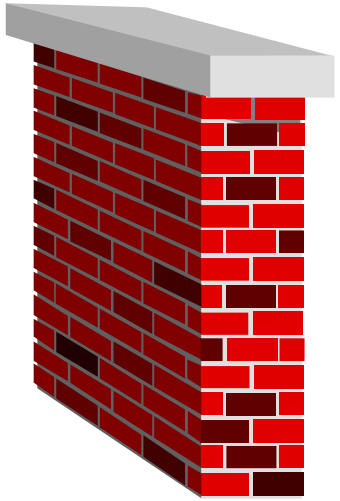
- păstreaza starea conexiunii (starea sesiunii curente)
- odată deschisă conexiunea nu se aplică alte reguli pentru închiderea lui
- se pot implementa reguli complexe



- susceptibile de atacuri de la nivelul aplicaţie
- funcţionează fără autentificarea utilizatorilor

Circuit level gateway

- operează la nivelul aplicaţie din stiva de protocoale TCP/IP
- funcţioneaza ca un proxy între aplicaţia server şi client
- gateway-ul ascunde adresele IP din reţeau privată
- odată realizată conexiunea, nu se mai aplică alte reguli de securitate

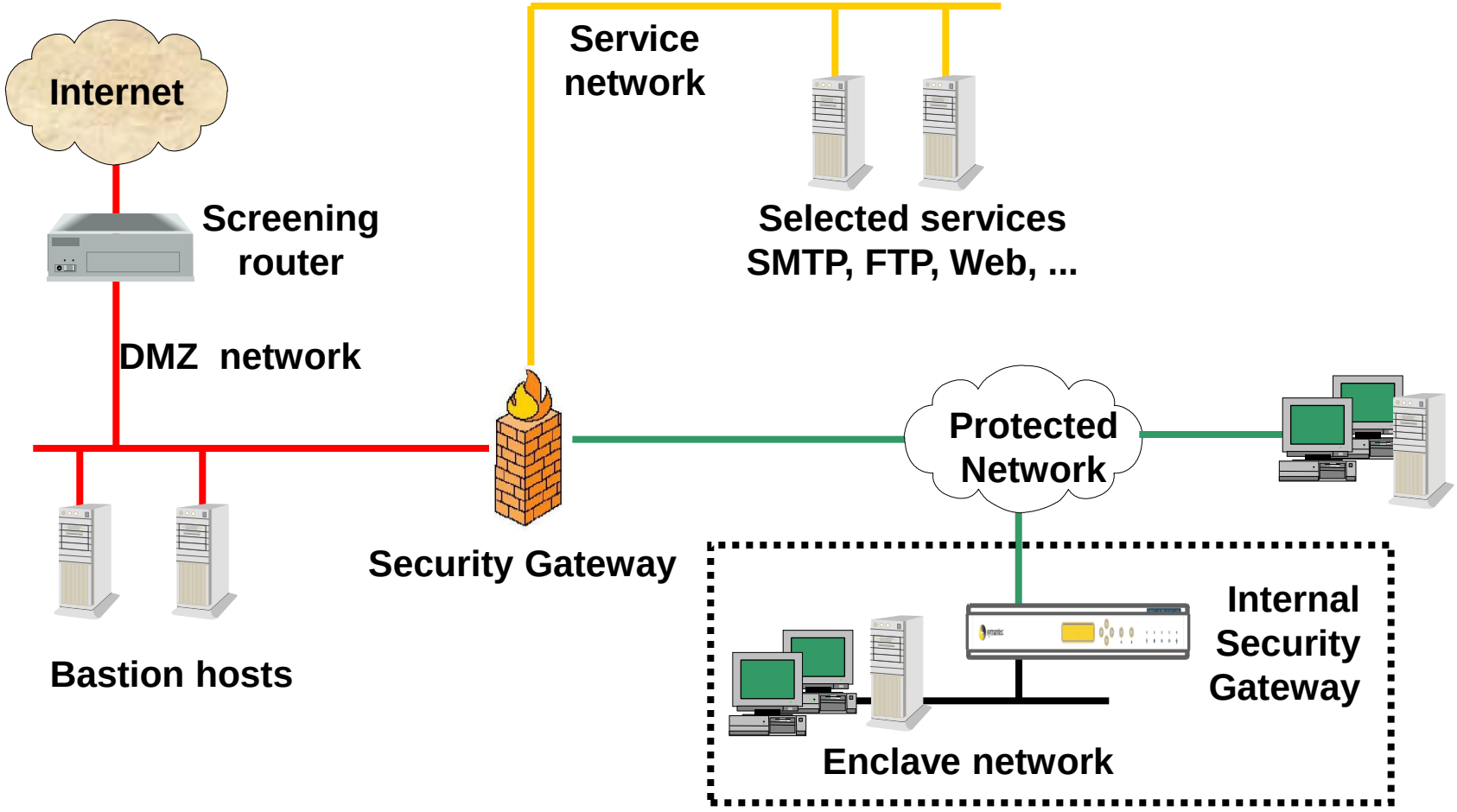


Network Topology

Topologia reţelei după conectarea printr-un firewall va include diferite tipuri de reţele, ca:

- Internet
  - reţeaua conectată la interfaţa nesigură a firewall-ului
- De-Militarized Zone (DMZ)
  - reţeaua de la interfaţa neprotejată a firewall-ului
- Extranet
  - în cazul VPNs, se pot include sisteme mobile
- Service network
  - o reţea protejată de firewall dar separată de reţeaua internă protejataă pentru a oferii servicii publice (web, ftp, email)
- Protected network
  - reţeaua internă protejată de firewall
- Enclave network
  - reţea privată protejată

Network Topology (cont)



Implementare Firewall în Linux

Folosind iptables:

- regula implicită: aruncă toate pachetele cărora nu se potriveşte nici o regula

iptables –P INPUT –j DROP

- deschiderea porturilor 80, 21

iptables –A INPUT –p TCP –dport 80 –j ACCEPT

iptables –A INPUT –p TCP –dport 21 –j ACCEPT

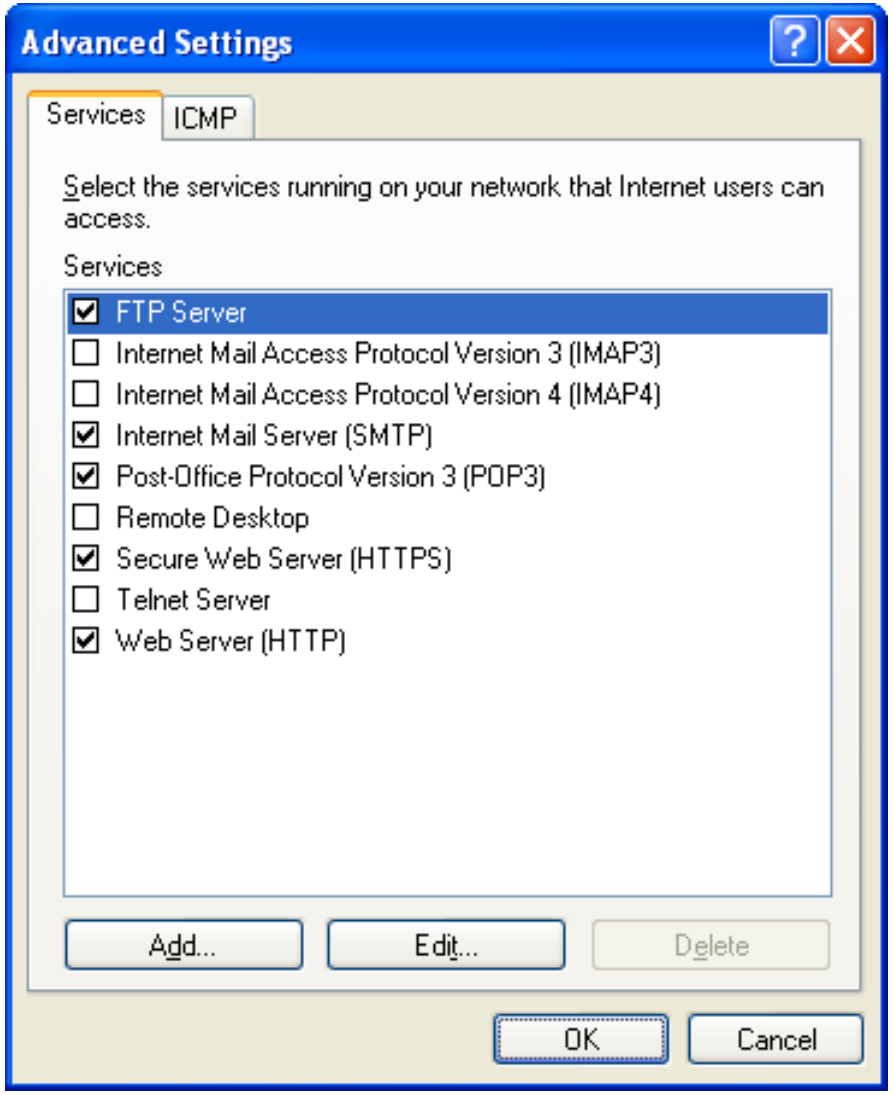
- filtrarea protocolului ICMP

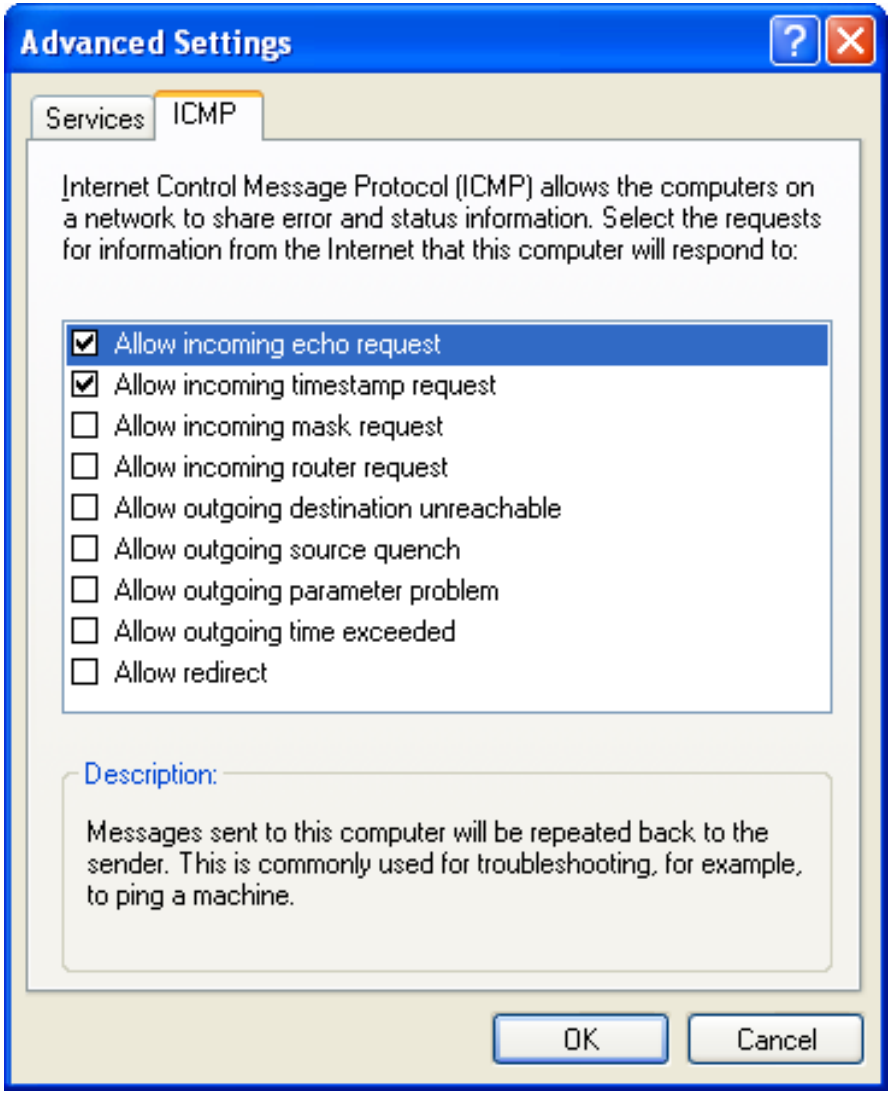
iptables –A INPUT –s 0/0 –d $IPEXT –p ICMP
–icmp-type echo-request –j REJECT

- activarea NAT

iptables –I FORWARD –s $LOCALNET/24 –d 0/0 –j MASQ

Firewall în Windows





Bibliografie

- http://en.wikipedia.org/wiki/Transmission_Control_Protocol
- http://en.wikipedia.org/wiki/User_Datagram_Protocol
- Terry William Ogletree – FIREWALLS - Protecţia Reţelelor Conectate la Internet, Ed. Teora
- http://www.r-c.ro/download/Manual_Utilizare_TL-WR841N_romana.pdf