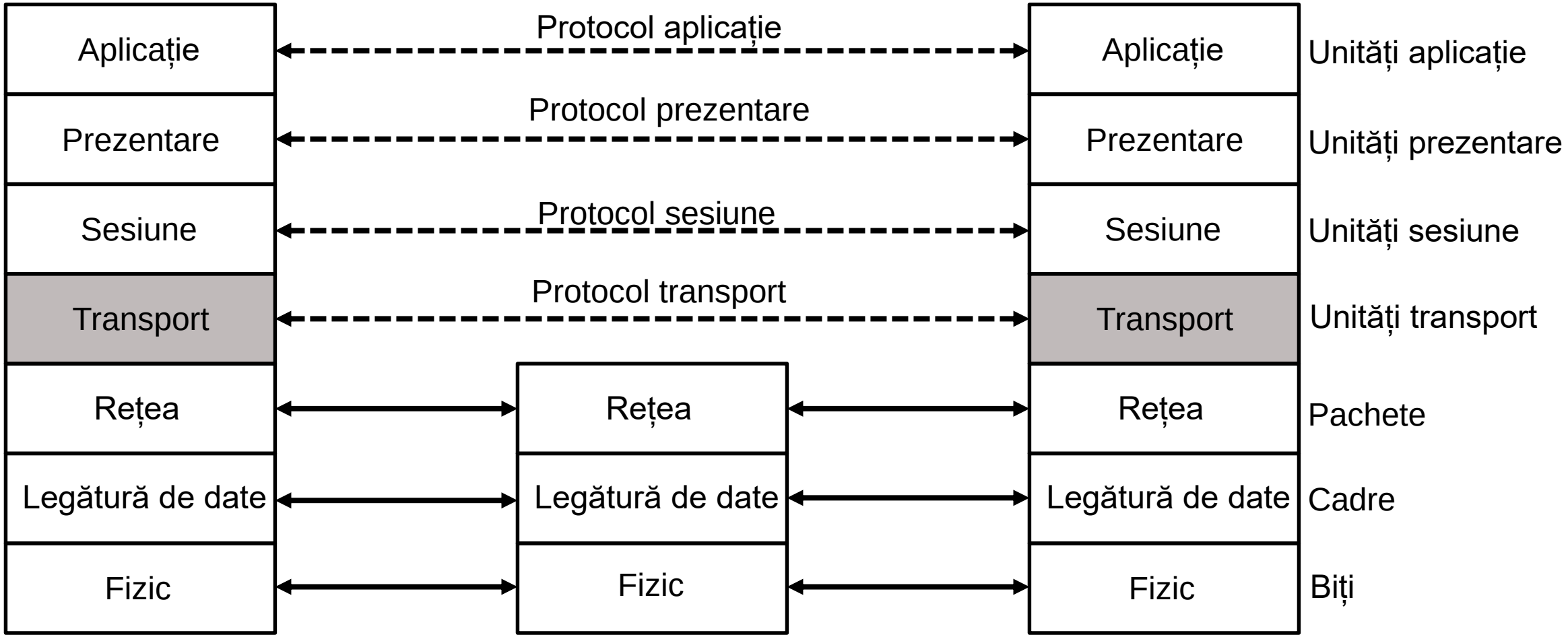
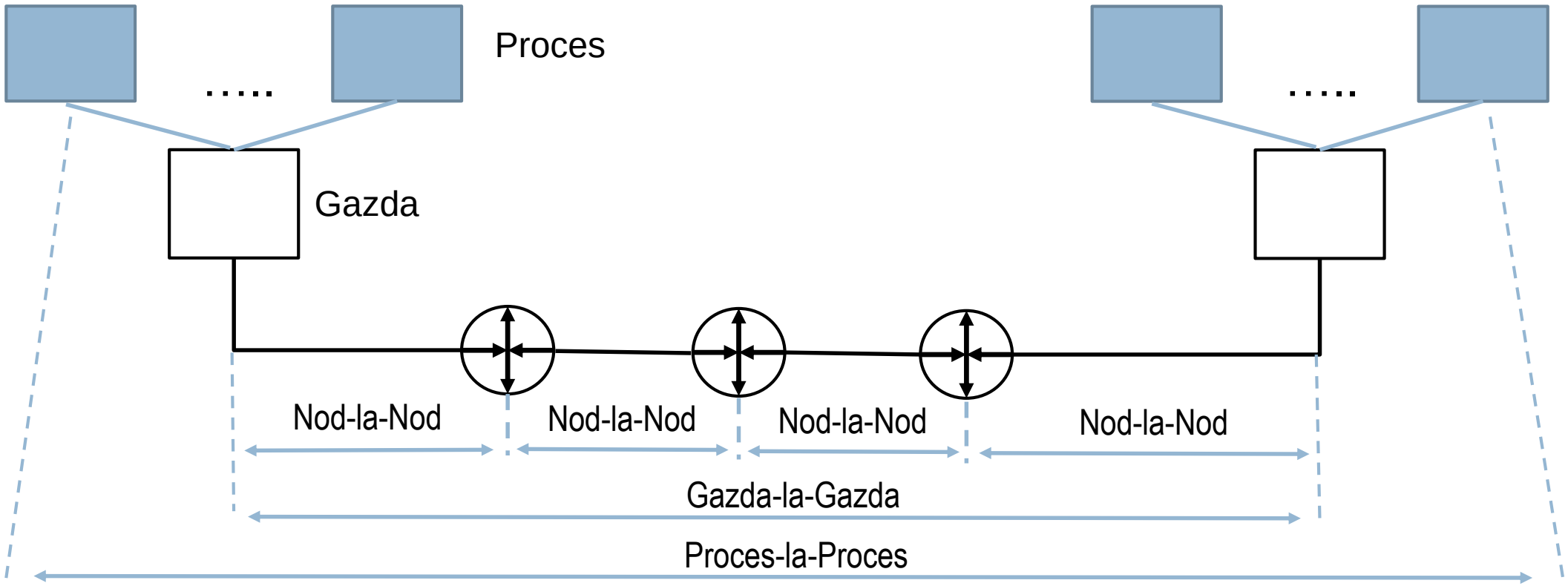
Nivelul transport

Modelul OSI


Aplicație
Prezentare
Sesiune
Transport
Rețea
Legătură de date
Fizic
Protocol aplicație
Protocol prezentare
Protocol sesiune
Protocol transport
Rețea
Legătură de date
Fizic
Aplicație
Prezentare
Sesiune
Transport
Rețea
Legătură de date
Fizic
Unități aplicație
Unități prezentare
Unități sesiune
Unități transport
Pachete
Cadre
Biți

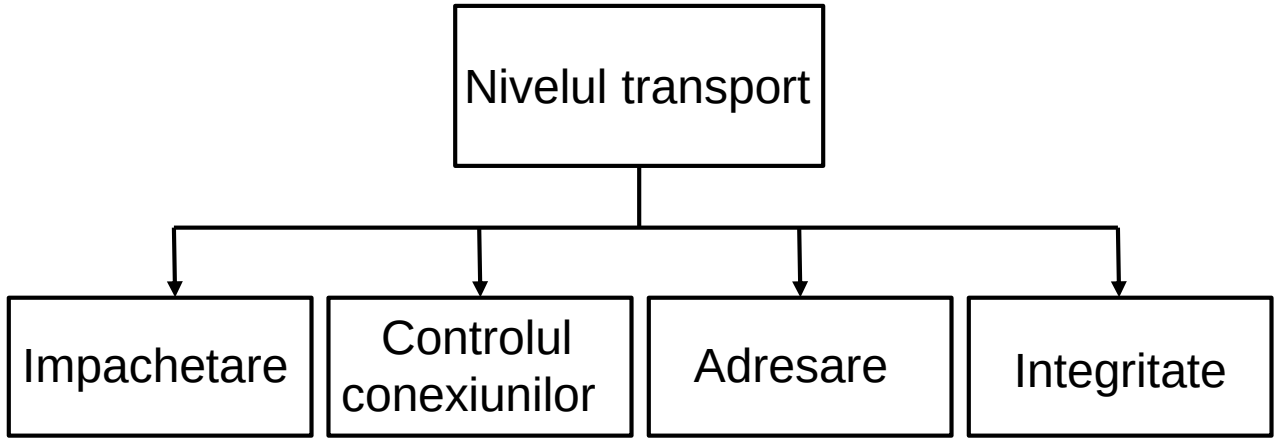
Nivelul transport

Nivelul transport este responsabil de transportul datelor intre procese, adica de transportul pachetului ca parte a unui mesaj, de la un proces la altul.



Nivelul transport

Ofera servicii nivelului sesiune si primeste servicii de la nivelul retea.



Responsabilitati:

- impărtirea datelor in **segmente / datagrame;**
- crearea de **conexiuni;**
- un nou mecanism de adresare (**porturi**);
- **controlul fluxului** (controlul congestiei);
- siguranta transmisiei (**reliability**).

Adresare nivel transport

Port = sistemul de adresare folosit de nivelul transport.

Existenta mai multor procese pe aceeasi statie necesita o **multiplexare prin porturi.**

16 biti → valori de la 0 la 65535.

Intervale de porturi (IANA):

- *Porturi rezervate* (well-known): între 0 și 1023 (ex. SSH –22, FTP –21, Telnet –23, HTTP –80);
- *Porturi inregistrate* între 1024 și 49151 (ex. Kazaa, RMI Registry, MySQL, etc.);
- *Porturi dinamice* (efemere): de la 49152 la 65535.

Adresa unui socket reprezinta o pereche formata dintr-o adresa IP si un port.

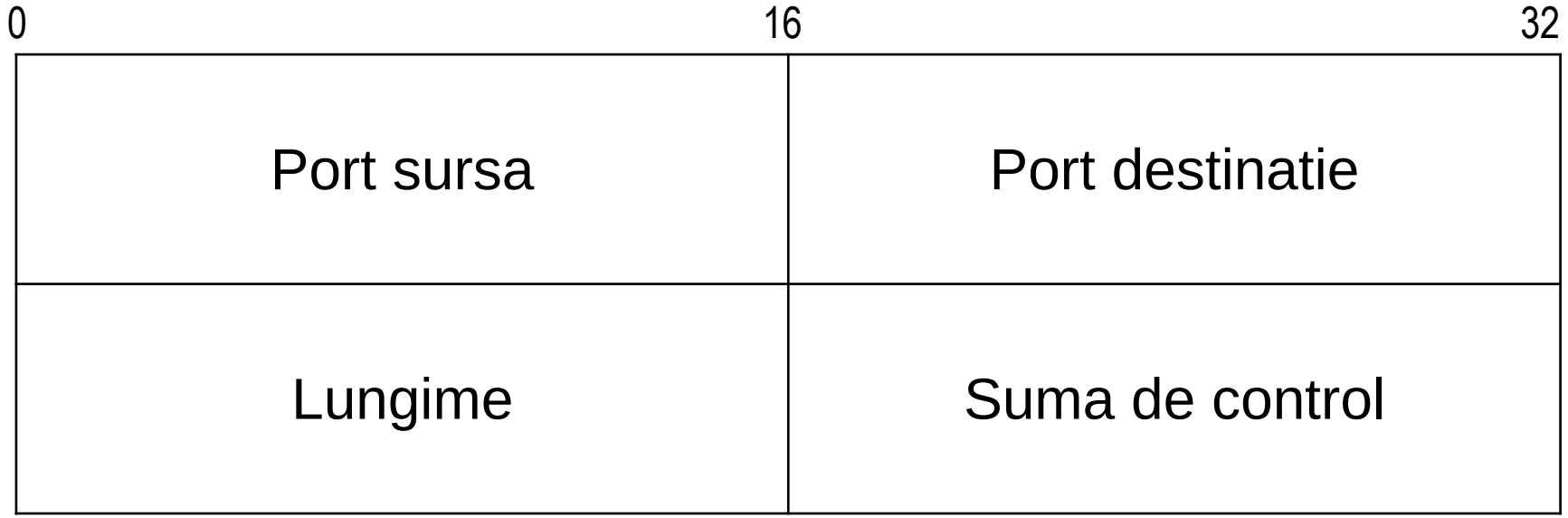
User Datagram Protocol (UDP)

- Neorientat conexiune;
- Nesigur (unreliable) (segmente pierdute);
- Fara controlul fluxului (segmente fara ordine).

Utilizare:

- atunci cand conexiunile orientate ofera un randament scazut : ex. DNS, SNMP;
- transmisii de date in timp real: ex. Skype;
- aplicatia asigura controlul integritatii: ex. TFTP.

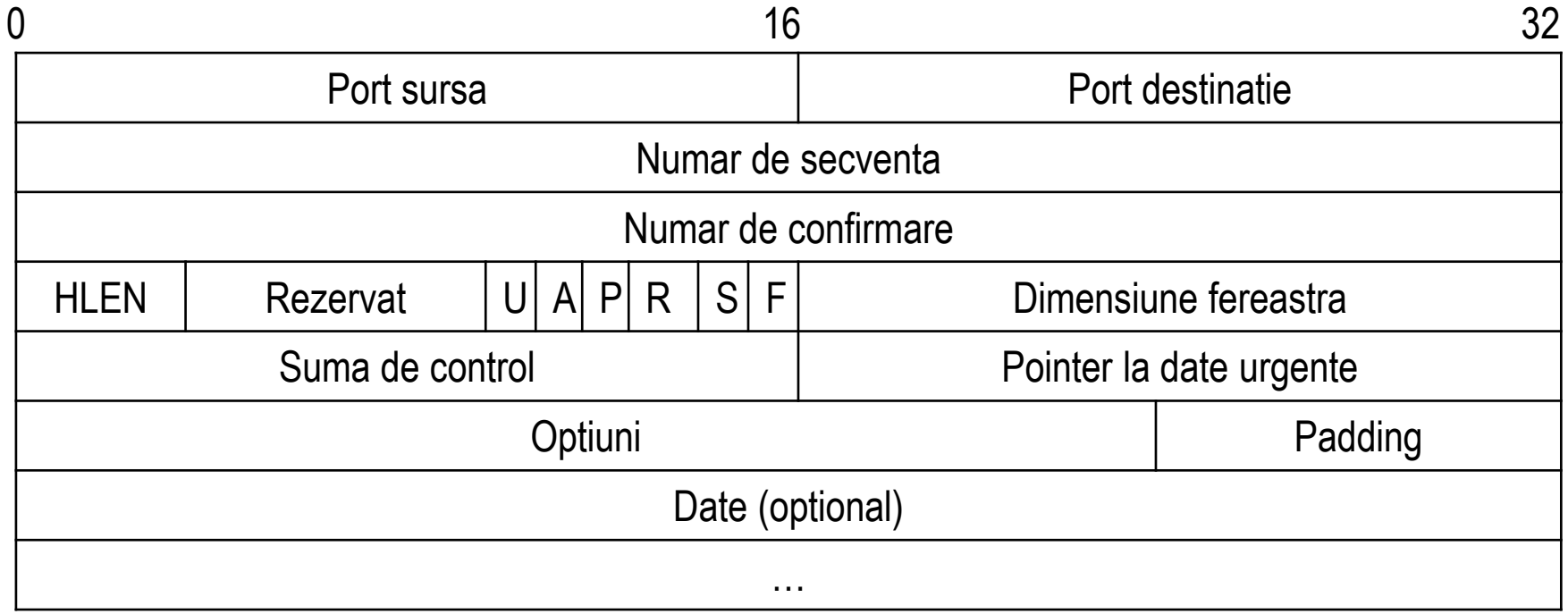
Antet UDP



Transmission Control Protocol (TCP)

- orientat pe conexiune;
- circuit virtual in care are loc comunicatia;
- protocol sigur (reliable) (datele ajung garantat la destinatie);
- datele ajung in ordine la destinatie;
- numere de secventa si numere de confirmare;
- controlul fluxului: corelare sender si receiver prin fereastra glisanta;
- controlul congestiei;
- controlul erorii: suma de control.

Antet TCP


0
16
32
Port sursa
Port destinatie
Numar de secventa
Numar de confirmare
HLEN
Rezervat
U
A
P
R
S
F
Dimensiune fereastra
Suma de control
Pointer la date urgente
Optiuni
Padding
Date (optional)
...

TCP - numere de secventa

Numar de secventa:

- indexul primului octet din segmentul TCP;
- in cadrul fiecarui segment;

Exemplu:

- primul octet are numarul de secventa 1000;
- cel de-al 100-lea octet are numarul de secventa 1099.

Numar de confirmare:

- indexul urmatorului octet pe care receptorul se asteapta sa-l primeasca de la transmitator;
- confirmarea primirii datelor de pana la acest numar;
- nu este prezent in toate segmentele;
- activat de prezenta campului (fanionului) ACK.

TCP - campul de control

Grup de 8 biti din antetul TCP.
Identifica diverse stari ale protocolului.
Mai multi biti pot fi activi simultan.

URG

- semnalizeaza transmisia unor date urgente;
- activeaza campul “pointer la date urgente”.

PSH

- fanionul determina golirea imediata a bufferelor: livrare imediata (pentru eficienta TCP foloseste buffere de intrare si iesire).

Exemplu:

- transmiterea secventei “login:” in retea.

TCP - Campul de control

RST

- utilizat pentru resetarea conexiunii;
- invalideaza numerele de secventa.

ACK

- utilizat pentru confirmarea pachetelor transmise;
- activeaza campul “numar de confirmare”.

SYN

- utilizat in protocolul de initiere a conexiunii (handshake).

FIN

- utilizat in protocolul de incheiere a conexiunii.

TCP - Alte campuri

HLEN (Header Length)

- lungimea antetului TCP în cuvinte de 32 de octeți.

Dimensiune fereastra

- spațiul pentru stocare date neconfirmate (receptor);
- valoarea maxima este 65535;
- optiune de scalare a ferestrei.

Suma de control

- antet + date.

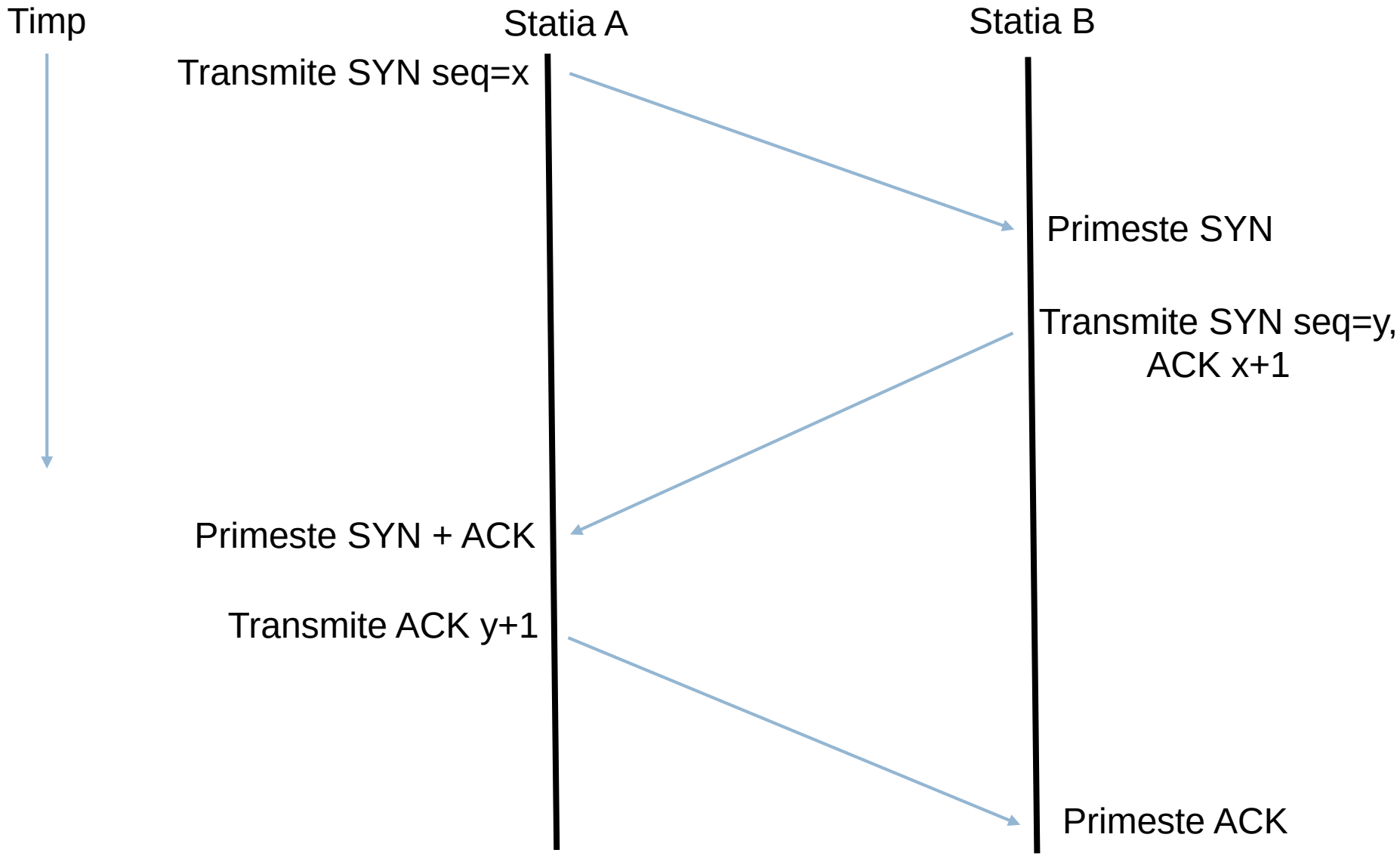
Optiuni

- diverse optiuni / extensii definite in RFC.

  Exemplu:

  - specificarea MSS (Maximum Segment Size).

TCP - Initierea conexiunii



TCP - Initierea conexiunii

Clientul este entitatea activa – initiaza conexiunea.

Campul SYN este activat.
ISN (Initial Sequence Number) = numărul de secvență dintr-un segment cu SYN activat.

Protocolul de initiere de conexiune - **3-way handshake.**

- primul pachet (SYN)
  - stabilirea ISN pentru comunicatia de la client la server.
- al doilea pachet (SYN+ACK)
  - confirmarea primului pachet;
  - stabilirea ISN pentru comunicatia de la server la client.
- al treilea pachet (ACK)
  - confirmarea celui de-al doilea pachet.

Cele doua ISN sunt generate aleator.

TCP - Full duplex

De ce sunt necesare două numere de secventa ?

- comunicația este full duplex.

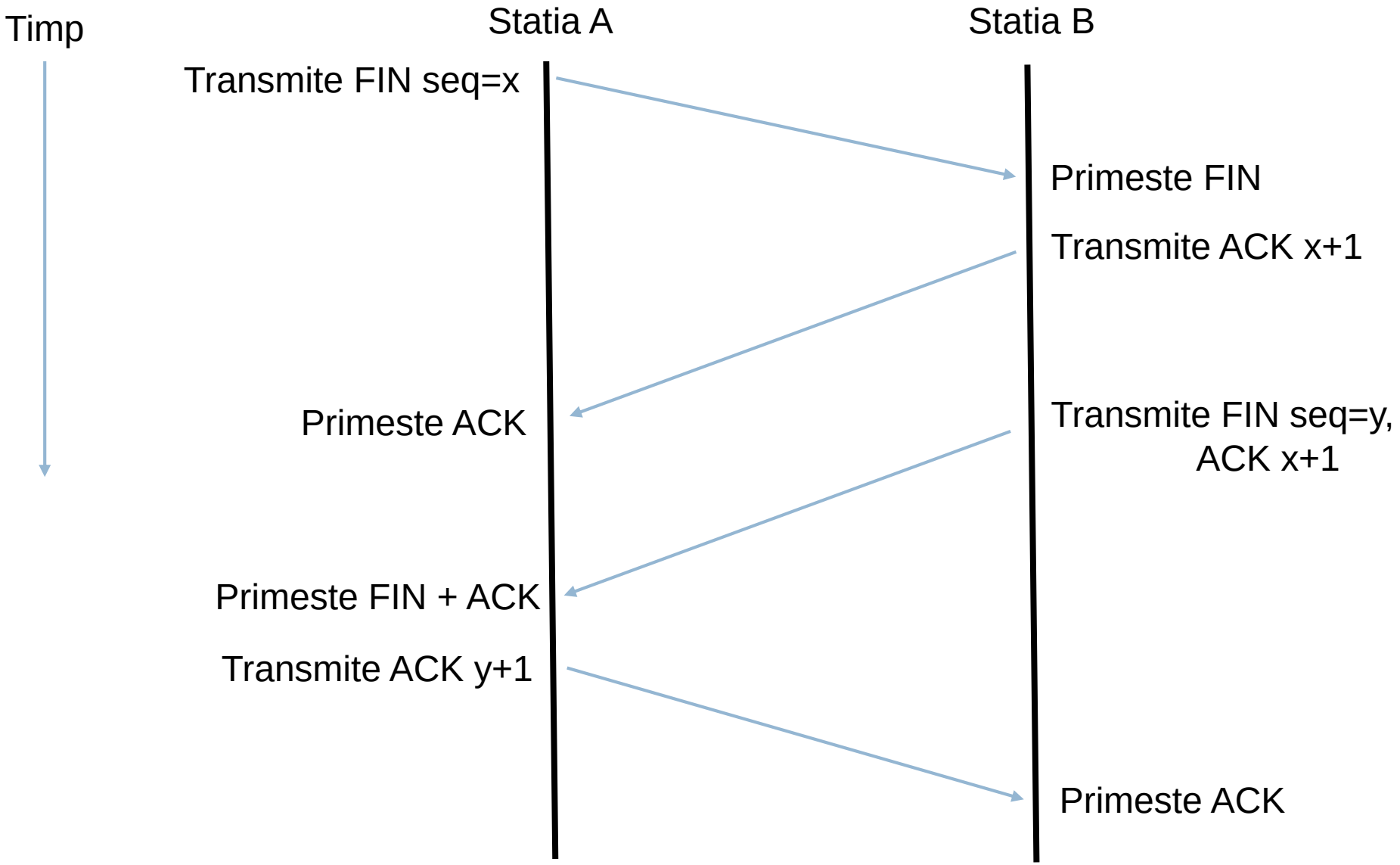
O conexiune TCP

- doua canale virtuale de comunicatie:
  - client → server;
  - server → client.

Un socket = două buffere (citire / scriere).

Este posibila comunicatia half-duplex prin inchiderea unui capat al conexiunii.

TCP - Incheierea conexiunii



TCP - Incheierea conexiunii

Initiata de oricare capat al transmisiei.

Campul FIN activat.

Protocol de tipul **4-way handshake:**

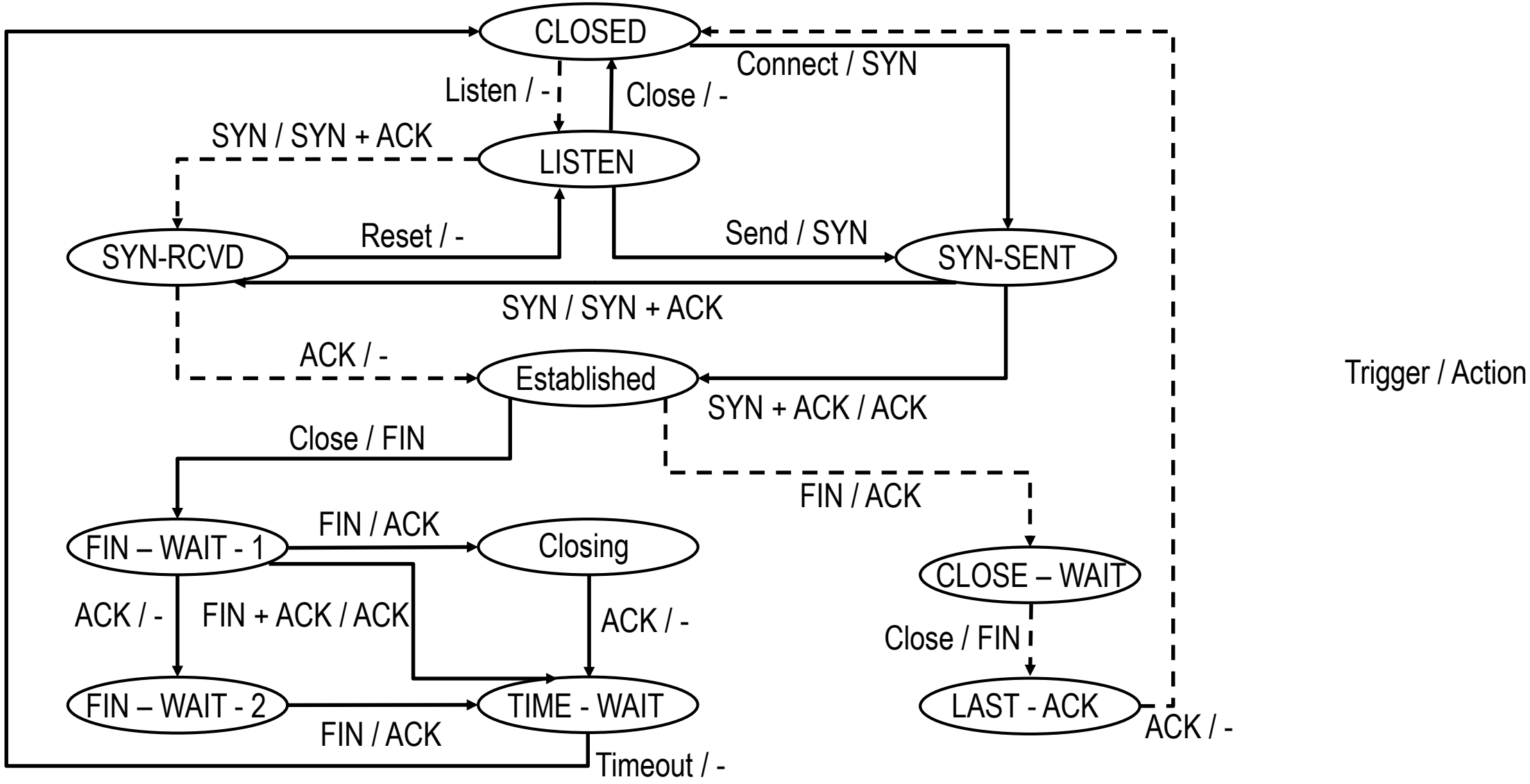
- primul segment are campul FIN activ;
- al doilea segment este o confirmare a primului;

Dupa al doilea pachet transmis conexiunea este pe jumatate inchisa (HALF CLOSED): comunicatia este intr-un singur sens.

- urmatoarele doua segmente inchid conexiunea in celalalt sens.

Este posibil si un protocol de tipul 3-way handshake atunci cand cele doua entitati inchid conexiunea in acelasi timp. In acest caz al doilea si al treilea segment sunt “unite”.

TCP- Diagrama de stari



TCP – Fereastra glisanta

Automatic Repeat reQuest (**ARQ**) – metodă de control al integritatii/erorilor studiată la nivelul legătură de date.

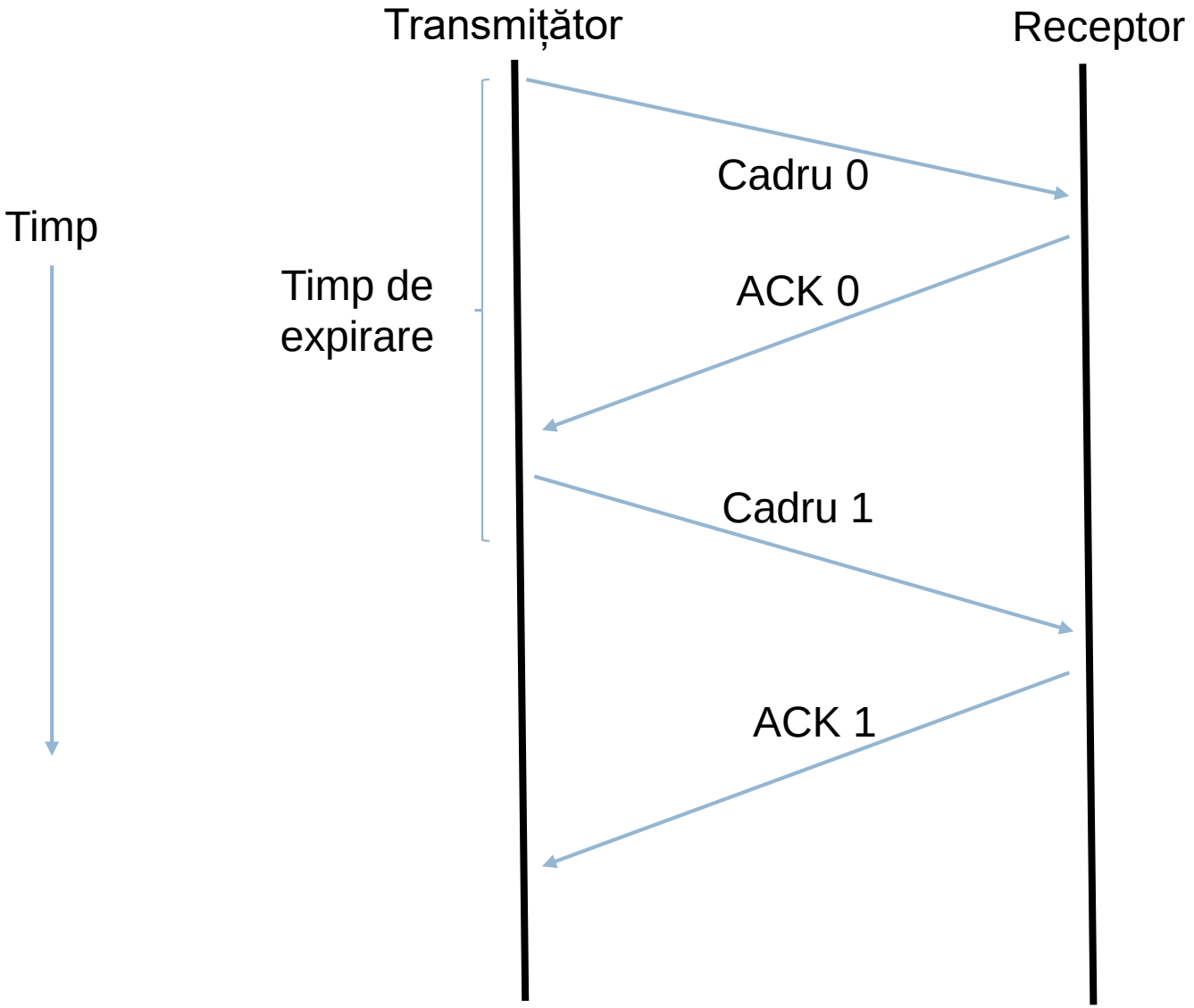
- folosește mesaje de confirmare (ACK) și timpi de expirare pentru a asigura fiabilitatea transmisiei.

Dacă transmițătorul nu primiște un mesaj de confirmare din partea receptorului intr-un interval de timp predefinit, acesta va retransmite mesajul.

Tipuri de ARQ:

Stop-and-wait
Go-back-N
Selective Repeat

Stop-and-wait ARQ



TCP - Fereastra glisanta

Sliding Window (SW) – generalizare Stop-and-Wait
Permite transmisia a W pachete / RTT.

Exemplu limitare Stop-and-wait ARQ clasic:

Lățime bandă (B) canal = 1 Mbps
Latență (L) = 50 ms => RTT = 100 ms
Dimensiune Pachet = 10000 biți = 1250 bytes
Rată de transfer = (1s / 100ms) Pachete * Dimensiune Pachet = 100 Kbps << B

Eficiență maximă pentru W = 2BL / Dimensiune Pachet.

Exemple de implementare:
Go-Back-N – simplu, ineficient;
Selective Repeat – complex, eficiență sporită.

Sliding Window – Transmițător

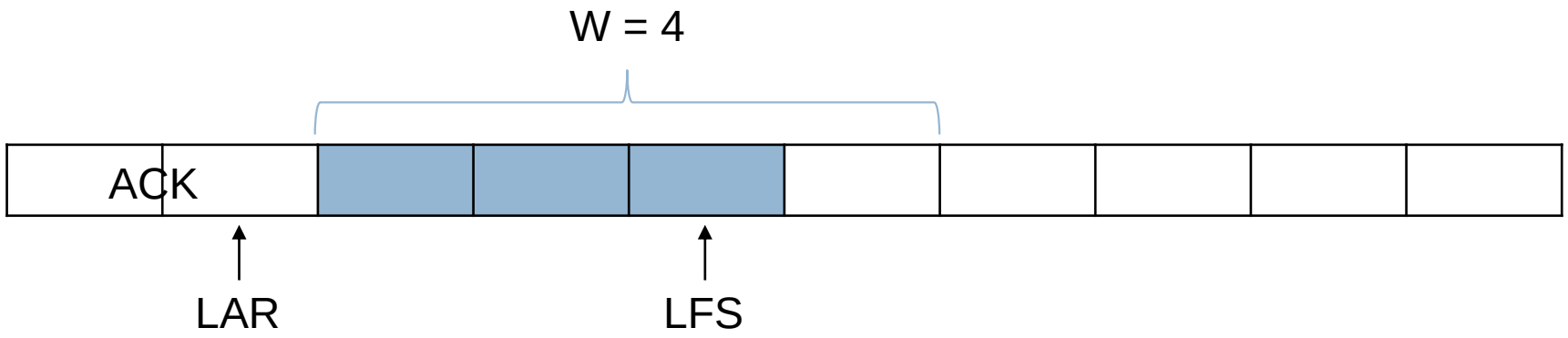
Transmițătorul menține în memoria tampon până la W segmente până când acestea sunt confirmate.

Variabile de stare:

LFS – Last Frame Sent

LAR – Last ACK Received

Funcționare: trimite segmente atâta timp cât LFS-LAR ≤ W



Sliding Window – Receptor

Go-Back-N

Receptorul menține o memorie tampon pentru următorul segment

Variabilă de stare:

LAS – Last ACK Sent

Funcționare:

La recepție:

- Dacă numărul de secvență este LAS + 1 acceptă segmentul și transmite-l aplicației
- Dacă numărul de secvență diferă de LAS + 1 distruge pachetul (out of order)

Sliding Window – Receptor

Selective Repeat

Receptorul:

- transmite datele către aplicație în ordine;
- menține o memorie tampon de dimensiune W pentru segmentele ajunse în ordine incorectă

Se transmit mesaje de confirmare pentru ultimul mesaj ajuns în ordine

Variabilă de stare:

LAS – Last ACK Sent

Funcționare:

La recepție:

- Menține în memoria tampon segmentele cu numărul de secvență între [LAS+1, LAS+W]
- Trimite mesajul având numărul de secveță LAS+1 la aplicație și actualizează LAS
- Trimite ACK pentru LAS

Sliding Window – Retransmisia

Go-Back-N – folosește un singur timp de expirare

- La expirare retransmite pachetele salvate în memoria tampon începând cu LAR+1

Selective Repeat – folosește câte un timer pentru fiecare segment neconfirmat

- La expirare retransmite doar segmentul asociat

TCP – Fereastra glisanta

Timpul de expirare:

$$SRTT_{N+1} = 0.9 * SRTT_{N} + 0.1\ RTT_{N+1}$$

$$Svar_{N+1} = 0.9 * Svar_{N} + 0.1 * (RTT_{N+1} - SRTT_{N+1})$$

$$Timeout_{N} = SRTT_{N} + 4 * Svar_{N}$$

SRTT = Smoothed RTT

Svar = Smoothed Variance

Controlul fluxului

Dimensiunea ferestrei transmitatorului este controlată de cea a receptorului.

Receptorul definește o fereastră de control al fluxului WIN pe care o trimite transmițătorului.

Transmițătorul folosește fereastra cu dimensiunea cea mai mică dintre SW și WIN.

TCP - Controlul fluxului

Memorie tampon receptor 4K