

Modul 4_2



Protocollo ARQ

ERROR CONTROL



(Automatic Repeat reQuest)

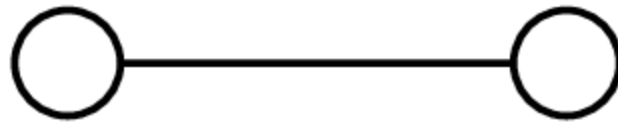


Continut

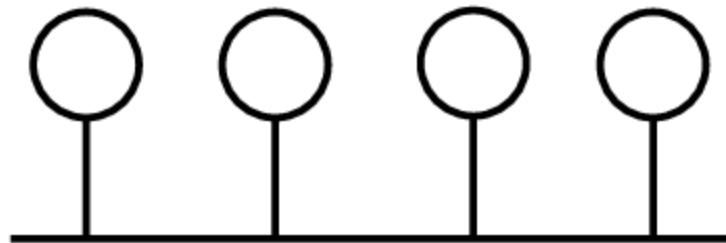
- Contextul utilizării protocoalelor pentru corectarea erorilor
- Protocolul Stop and Wait
- Protocolul Go Back N
- Protocolul Selective Repeat

Canale de comunicare

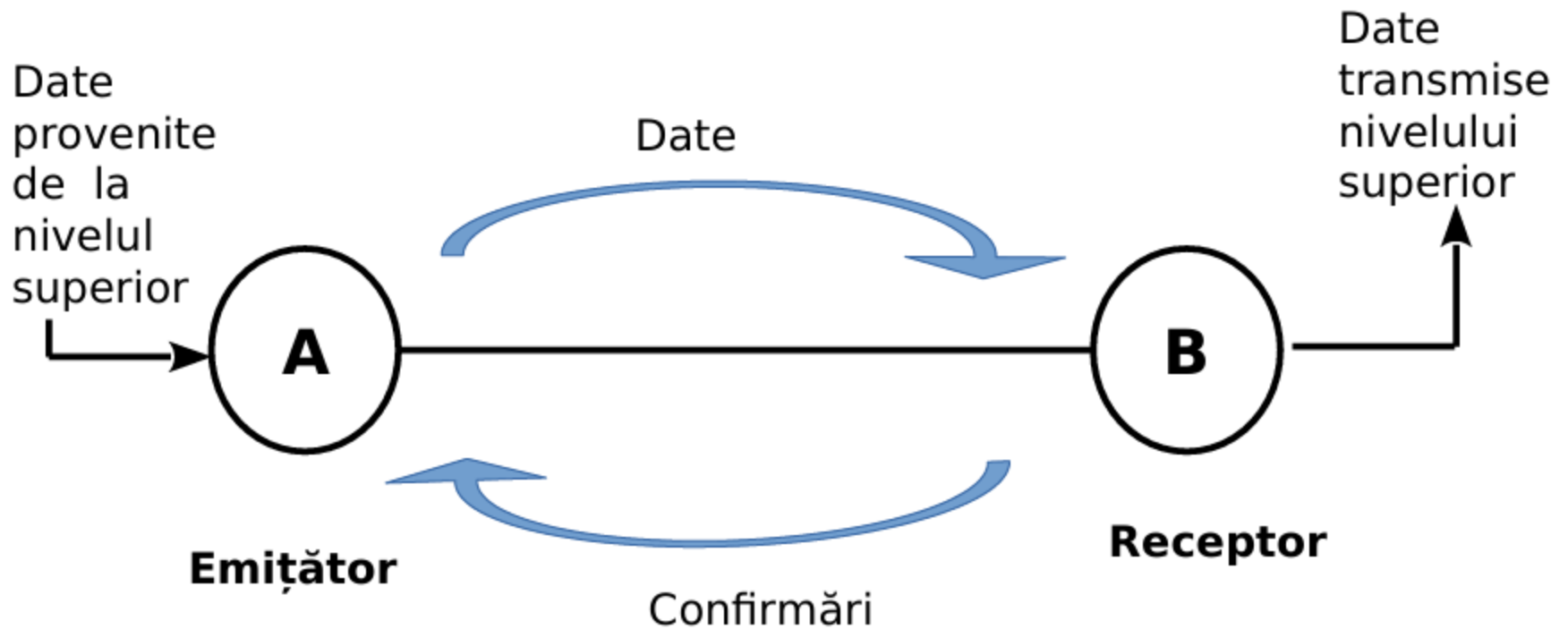
Point-to-point



Multipoint (mediu partajat)



Model de comunicare



Fenomene perturbatorii pentru transmiterea corectă a datelor

- **Suport de comunicare neideal**
 - Debit limitat
 - Prezența erorilor
 - zgomot intern
 - perturbări externe
- **Trafic eterogen și neregulat**
- **Încărcarea neuniformă a rețelei** (echipamente și canale de comunicare)
 - rețineri mari la transmiterea datelor
 - pierderi
- **Pene și disfuncționalități ale echipamentelor și canalelor**

Condiții pentru transmiterea fiabilă de date

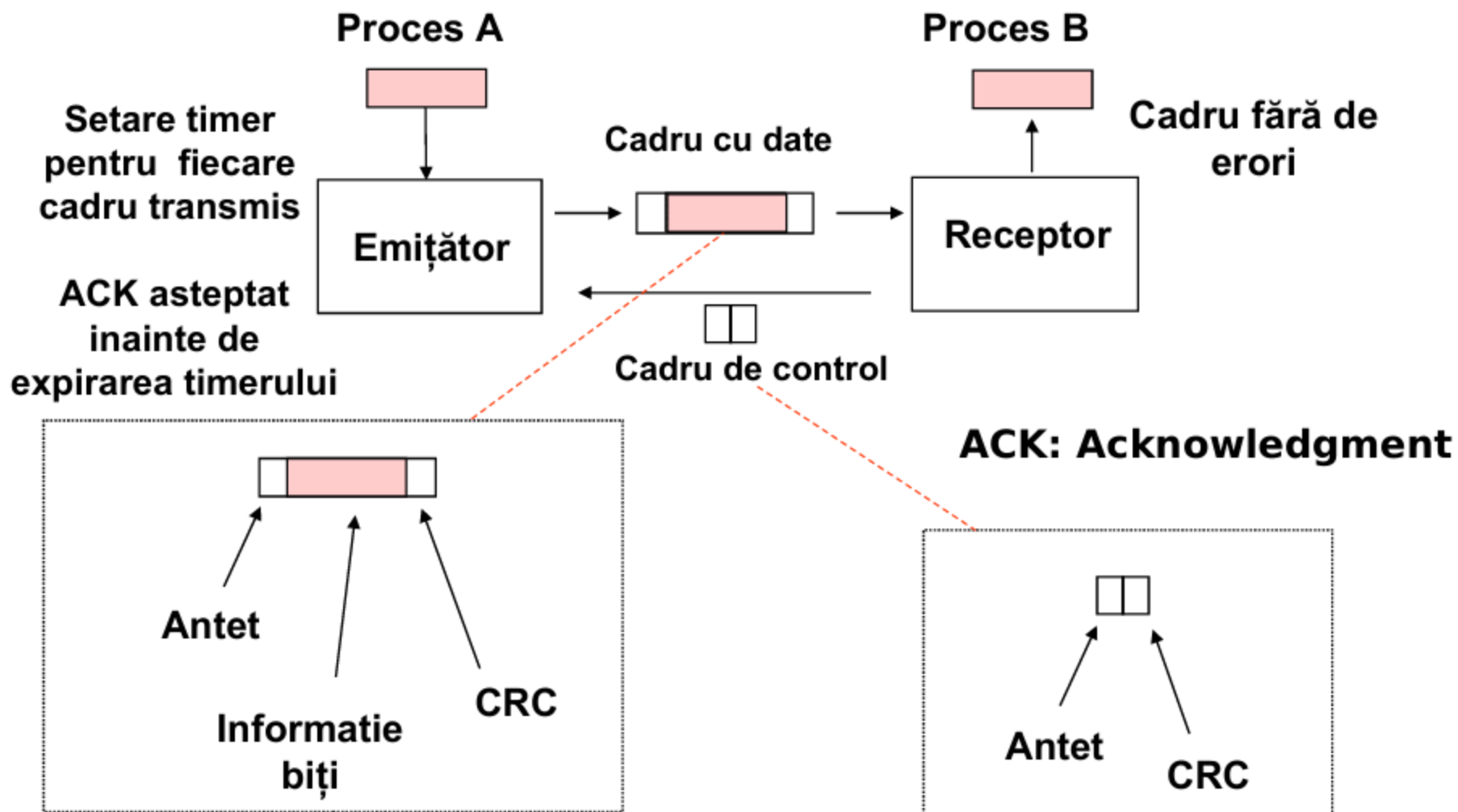
- **Gestionarea erorilor** (detectarea și la corectarea)
 - Include următoarele acțiuni:
 - Detectarea erorilor
 - Confirmare pozitivă (Positive Acknowledgement (ACK)): în cazul absenței erorilor în datele recepționate
 - Confirmare negativă (Negative Acknowledgement (NAK)): dacă au fost detectate erori în datele recepționate
 - Retransmiteri la expirarea timeout-ului: datele sunt retransmise dacă nu sosește confirmare după o anumită durată de timp.
 - Aceste acțiuni sunt cunoscute ca **protocoale ARQ (Automatic Repeat reQuest)**
- **Controlul fluxului**
 - Proceduri care restricționează volumul de date transmis de emițător către receptor pentru a evita inundarea lui – datele sunt transmise doar după confirmarea bunei recepții a celor precedente

Protocoloale Automatic Repeat reQuest (ARQ)

- ARQ oferă un serviciu de transport fiabil la traversarea mediilor de comunicare nefiabile
- ARQ asigura transmiterea fiabila a datelor :
 - Fără de erori
 - Fără dubluri
 - Cu respectarea ordinii de transmitere
 - Fără de pierderi

Protocolul Stop-and-Wait

Trimite un cadru, așteaptă un ACK

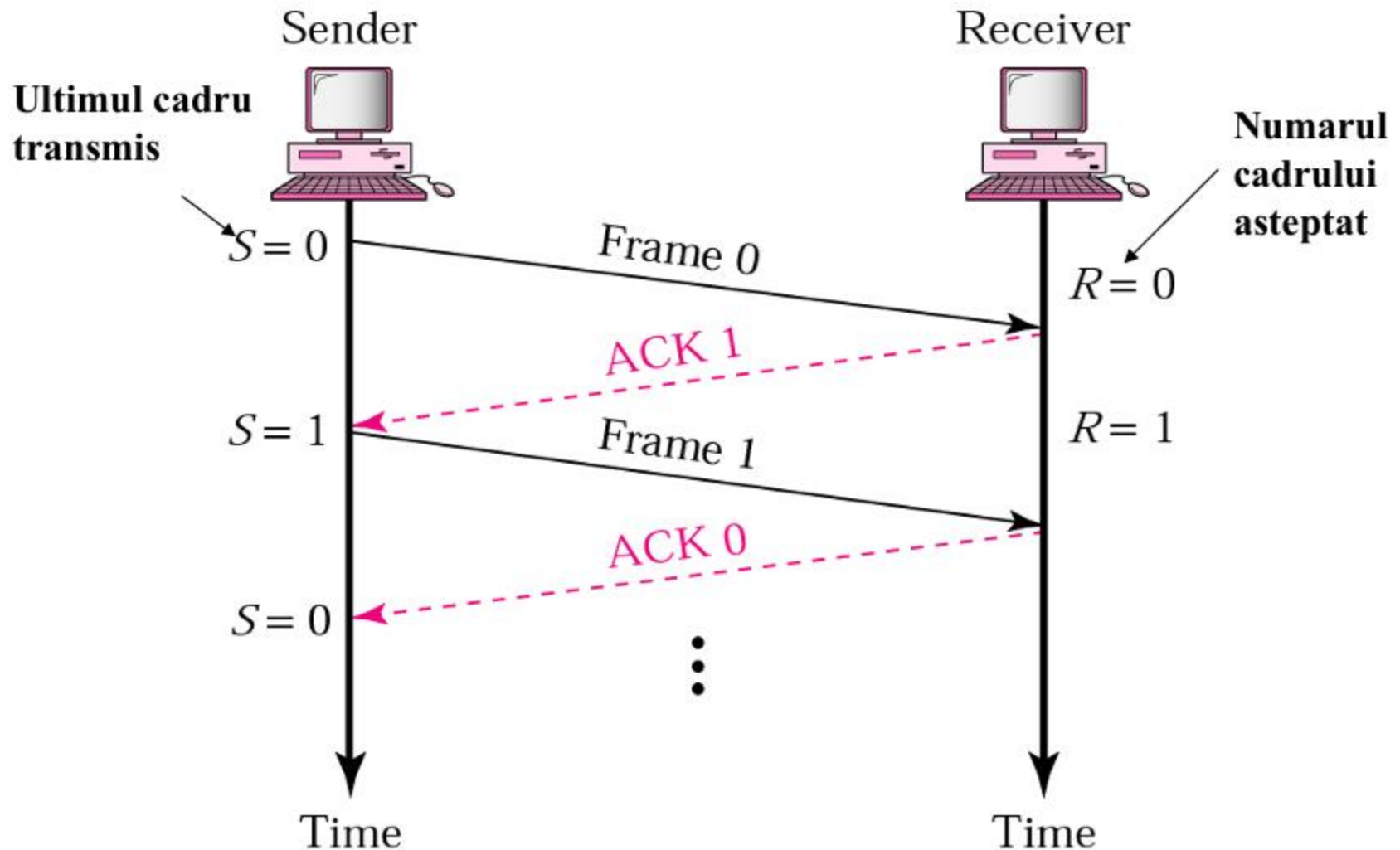


Protocolul Stop-and-Wait

Funcționare în prezenta erorilor și a pierderilor

- Cadrele și ACK-urile sunt numerotate alternativ (0,1) pentru a evita dublurile și pierderile (e suficient 1 bit!)
- Când receptorul este în așteptarea cadrului cu numărul 1, la recepția reușită a acestui cadru el va emite ACK0 anunțând emițătorul numărul cadrului așteptat (nr. 0).
- Numerele de secvența sunt incrementate modulo 2.
 $1+1 = 0$, $0+1=1$
- Receptorul dispune de un contor (R) în care păstrează numărul cadrului așteptat. R este incrementat cu 1 modulo 2 în cazul când cadrul așteptat este recepționat cu succes.
- Emițătorul dispune de un contor (S) în care se păstrează numărul ultimului cadru transmis. S se incrementează cu 1 modulo 2 când primește confirmarea pentru ultimul cadru transmis, în continuare transmite cadrul următor.

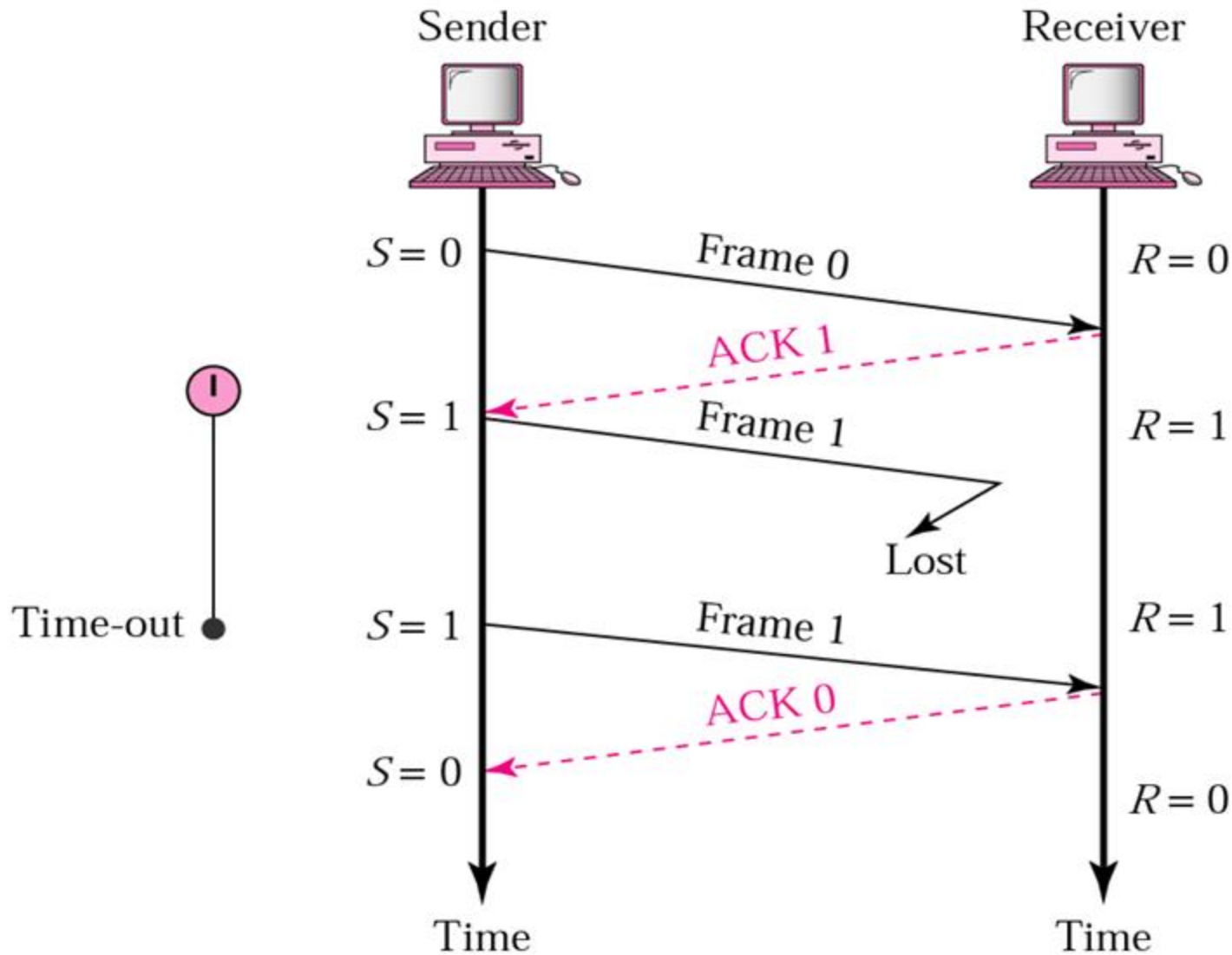
Funcționare normală



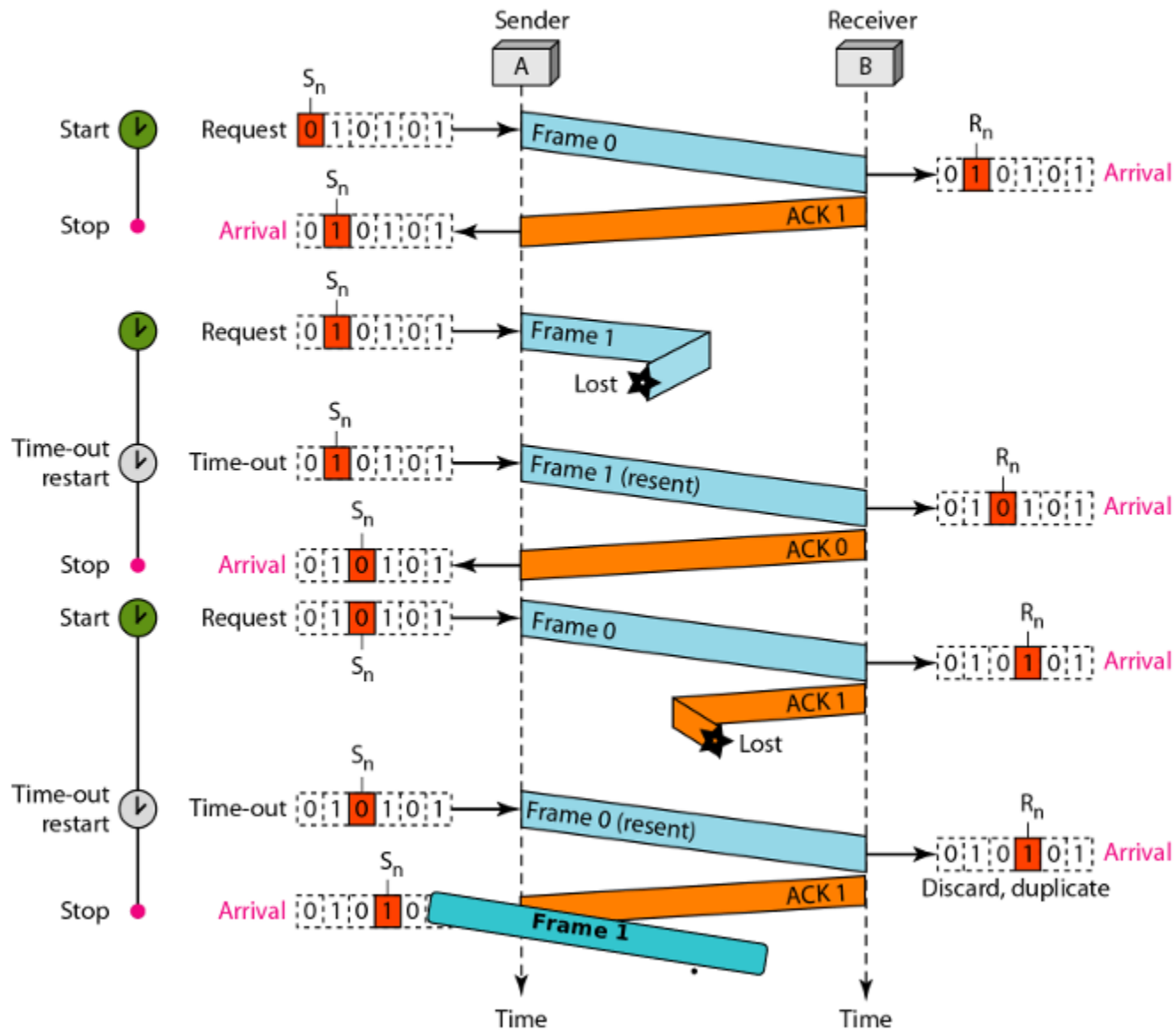
Protocolul Stop-and-Wait

- Dacă o eroare este detectată, receptorul ignorează cadrul respectiv
- Dacă numărul cadrului recepționat nu corespunde cu cel așteptat, receptorul ignorează acest cadru și retransmite ultimul ACK
- Dacă emițătorul primește un ACK cu număr diferit de $S+1$, el îl ignorează
- Emițătorul păstrează o copie a ultimului cadru transmis până în momentul când primește un ACK pentru cadrul respectiv
- Emițătorul setează un timer la transmiterea fiecărui cadru. Dacă el nu recepționează un ACK pentru acest cadru înainte de expirarea temporizatorului, cadrul respectiv este retransmis.

Protocolul Stop-and-Wait cu pierderi



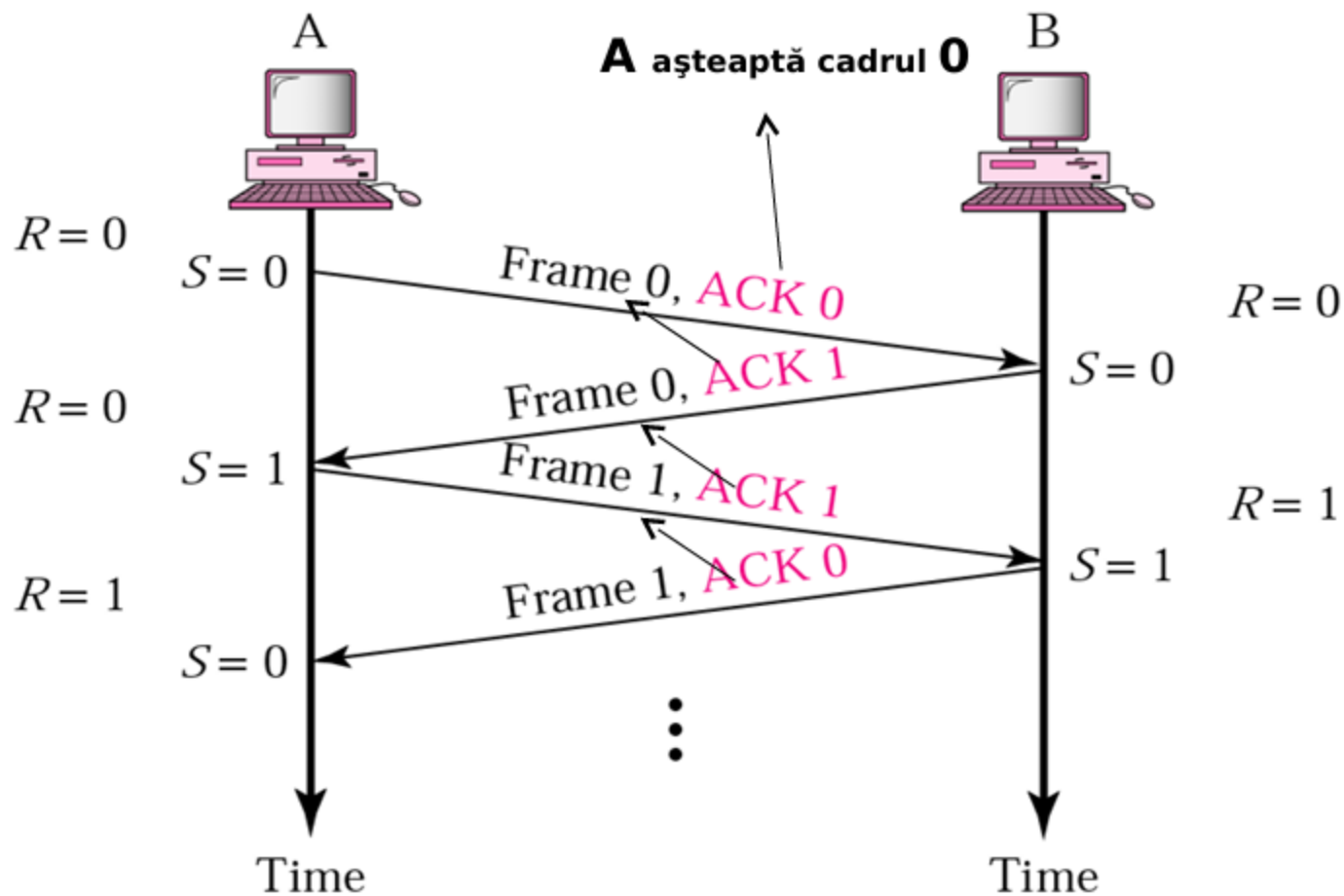
Protocolul Stop-and-Wait: exemplu



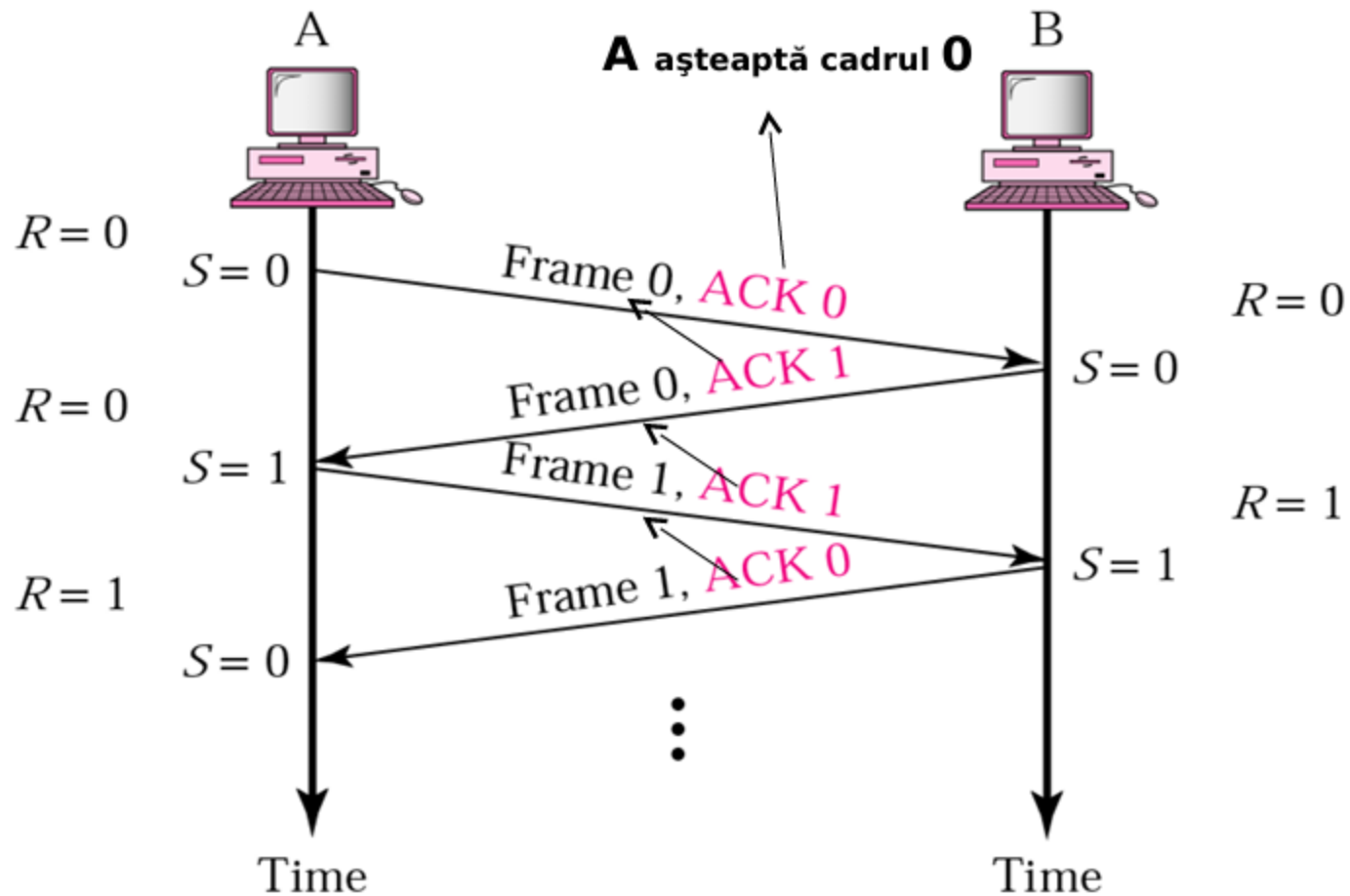
Căi de ameliorare S&W

- Protocolul precedent este unidirecțional (simplex), însă el poate funcționa și în regim bidirecțional (full-duplex): ambele noduri ar putea transmite/recepționa date
- Pentru a nu încărca inutil canalul un nod transmite împreună datele + ACK-ul pentru cadrul așteptat din cealaltă parte, ambele fiind incluse în același cadru (Piggybacking)
- Dacă nodul primește un cadru și nu are date de transmis ACK-ul respectiv va fi transmis într-un cadru fără de date.

Transmitere full-duplex: piggybacking



Transmitere full-duplex: piggybacking

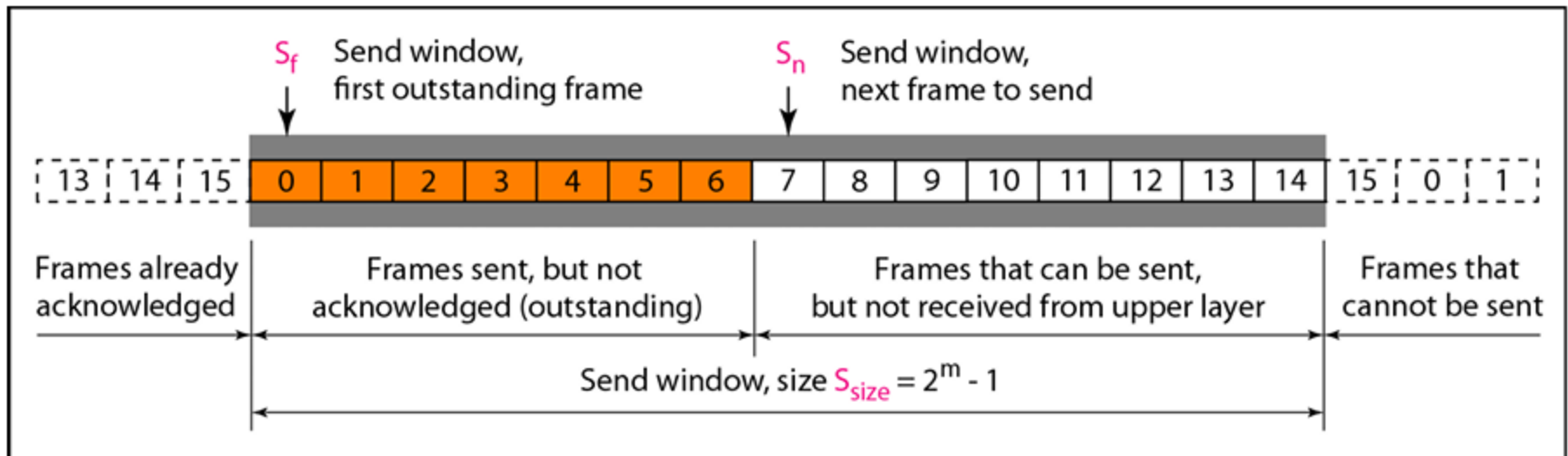


Protocolul Go-Back-N

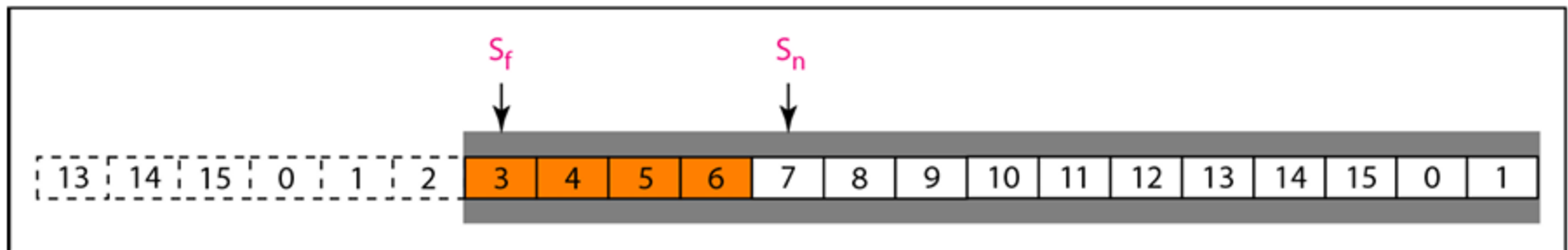
Setare fereastra emițător

- Cadrele sunt numerotate in antet secvențial utilizând m biți
 - Pentru o secvența de m biți numerele vor fi între 0 și $(2^m - 1)$, și se vor repeta
 - $m=3 \rightarrow (0,1,2,3,4,5,6,7,0, 1,2,3,4,5,6,7, \dots)$
- Emițătorul este autorizat să trimită mai multe cadre fiind în așteptarea ACK-urilor, numărul total al cadrelor neconfirmate nu poate depăși $W = 2^m - 1$ ceea ce se numește fereastra
- Fereastra conține toate numerele cadrelor care sunt în tranzit și ale celor care mai pot fi transmise
- Cadrele din stânga ferestrei sunt deja confirmate iar cele din dreapta ei nu pot fi trimise până în momentul când fereastra va "glisa" spre ele.
- Cadrele cu numerele din interiorul ferestrei așteaptă confirmări (adică ACK-uri) sau vor fi trimise imediat ce vor recepționate de la nivelul superior
- Începutul ferestrei emițătorului va glisa la numărul (i) imediat ce ACK i este recepționat

Fereastra emițătorului pentru protocolul Go-Back-N

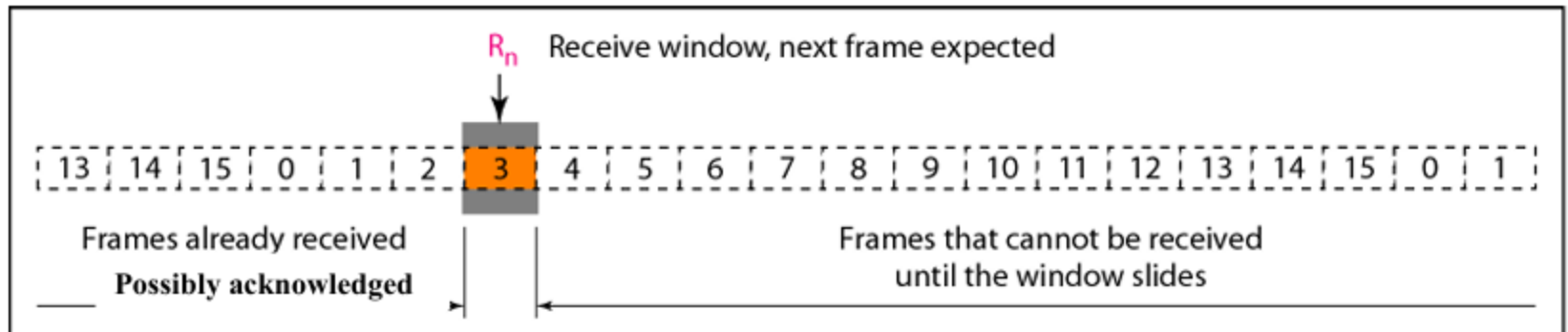


a. Send window before sliding

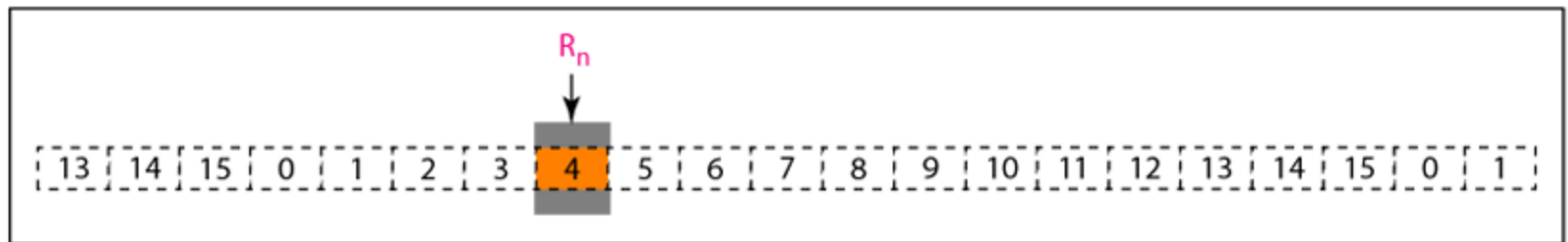


b. Send window after sliding

Fereastra receptorului pentru protocolul Go-Back-N



a. Receive window



b. Window after sliding

Imediat ce cadrul e recepționat fără de erori și in ordine, fereastra se deplasează cu o poziție spre dreapta

Go-Back-N în detalii

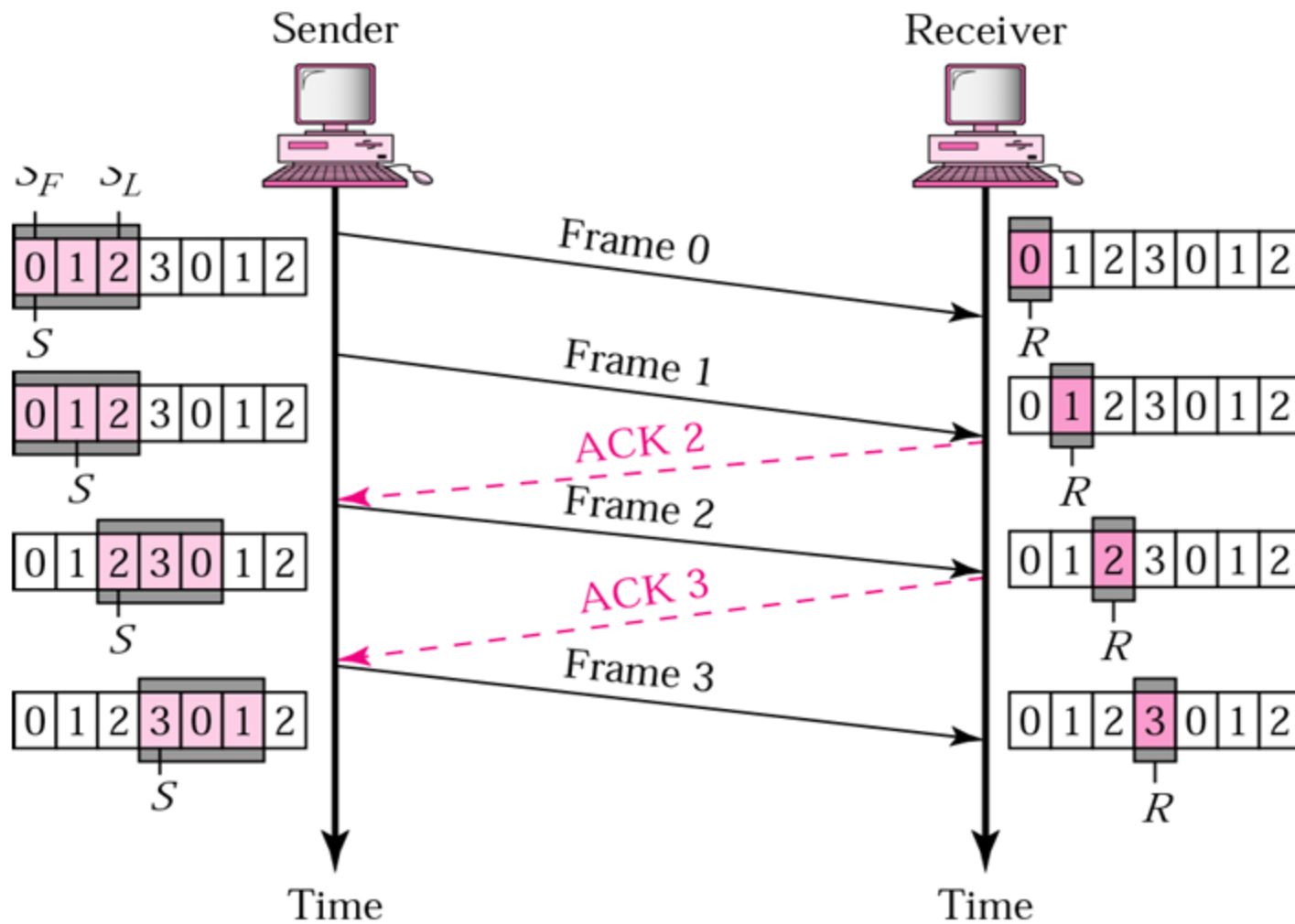
Setarea ferestrei receptorului

- Dimensiunea ferestrei receptorului este întotdeauna egală cu 1 și conține numărul cadrului așteptat
- Prin urmare cadrele trebuie să sosească în ordine pentru a fi acceptate.
- Dacă cadrul așteptat este recepționat fără de erori, fereastra receptorului va glisa cu 1 către numărul de secvență următor

Funcționare

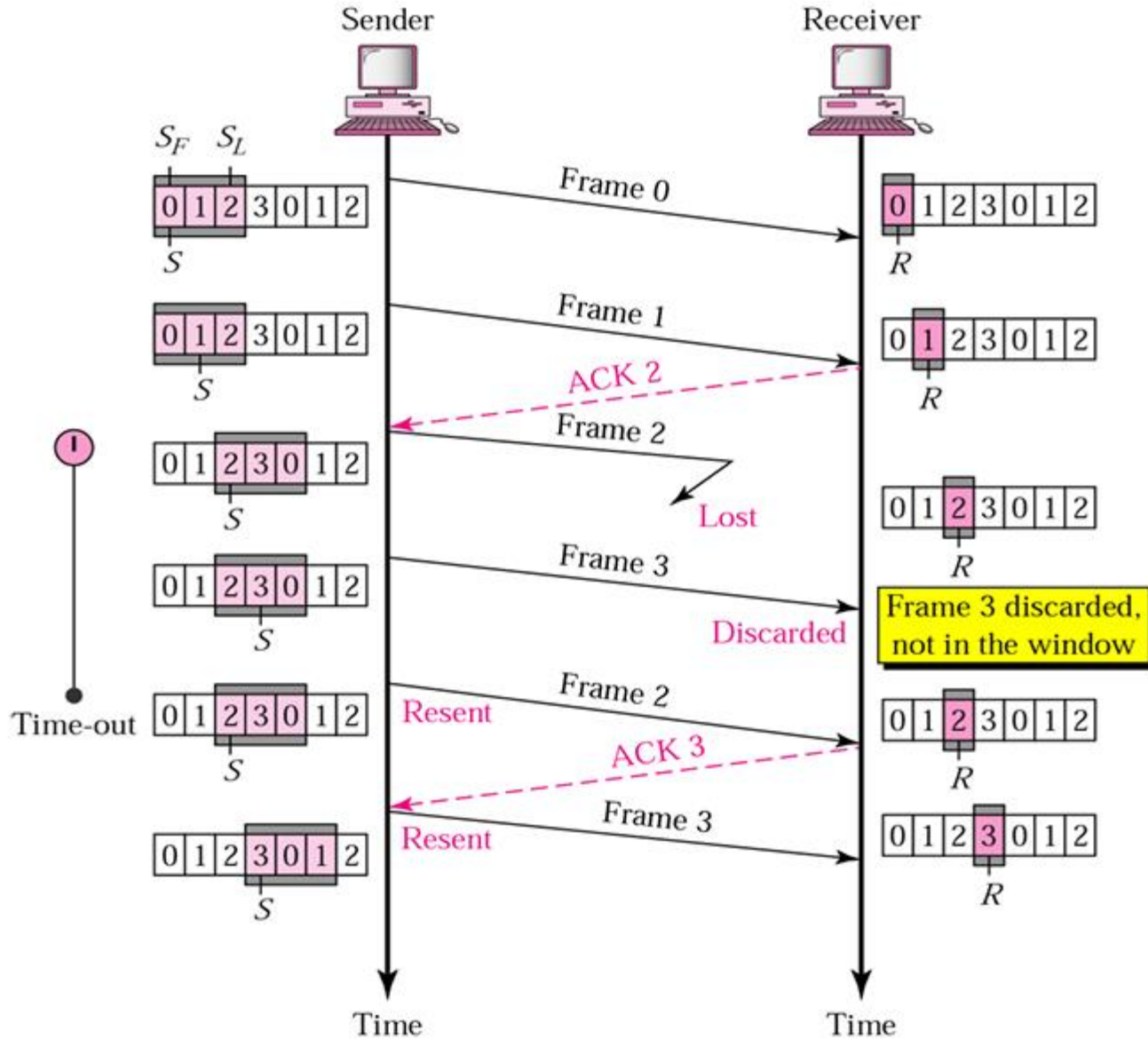
- Receptorul trimite un ACK pozitiv dacă cadrul primit este fără de erori și are numărul de ordine așteptat
- Receptorul nu este obligat să confirme individual fiecare cadru primit corect
- Receptorul poate trimite ACK-uri cumulative pentru un grup de cadre (de ex. ACK5 confirmă cadrele (0,1,2,3,4) și setează cadrul așteptat cu nr. 5)
- Dacă cadrul este cu erori sau nu are un număr de ordine bun, receptorul îl va ignora silențios, va ignora de asemenea și toate cadrele cu numere mai mari decât cel așteptat.
 - În acest caz nu se transmite ACK
- Dacă timer-ul emițătorului expiră înainte de a primi un ACK la cadrul respectiv, *TOATE* cadrele începând cu cel cu numărul așteptat vor fi retransmise (de aici denumirea Go-Back-N).

Functionare Go-Back-N



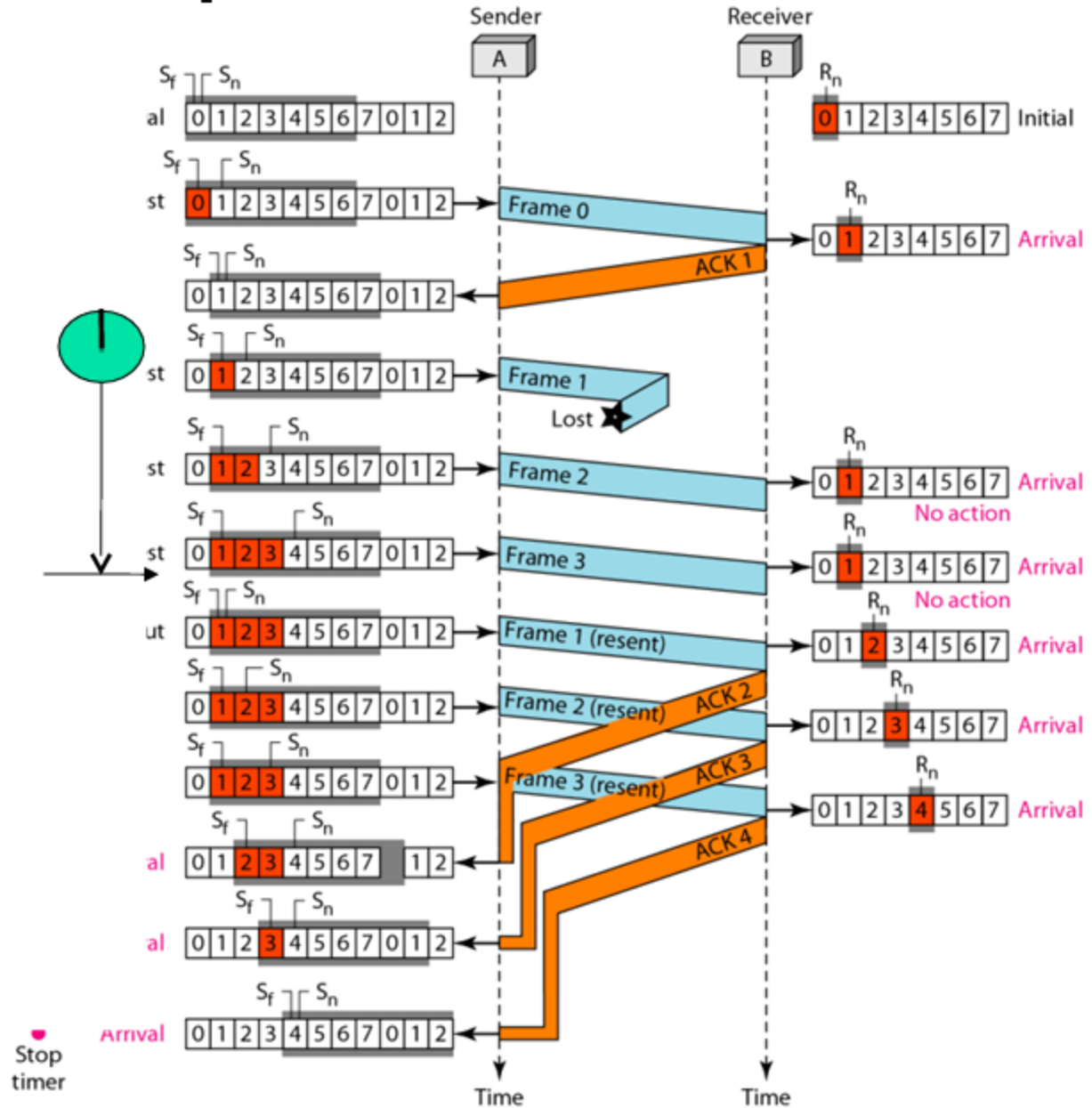
Fereastra emitatorului se deplaseaza la numarul cadrului i in momentul cand el receptioneaza (ACK i)

Go-Back-N, cadru pierdut

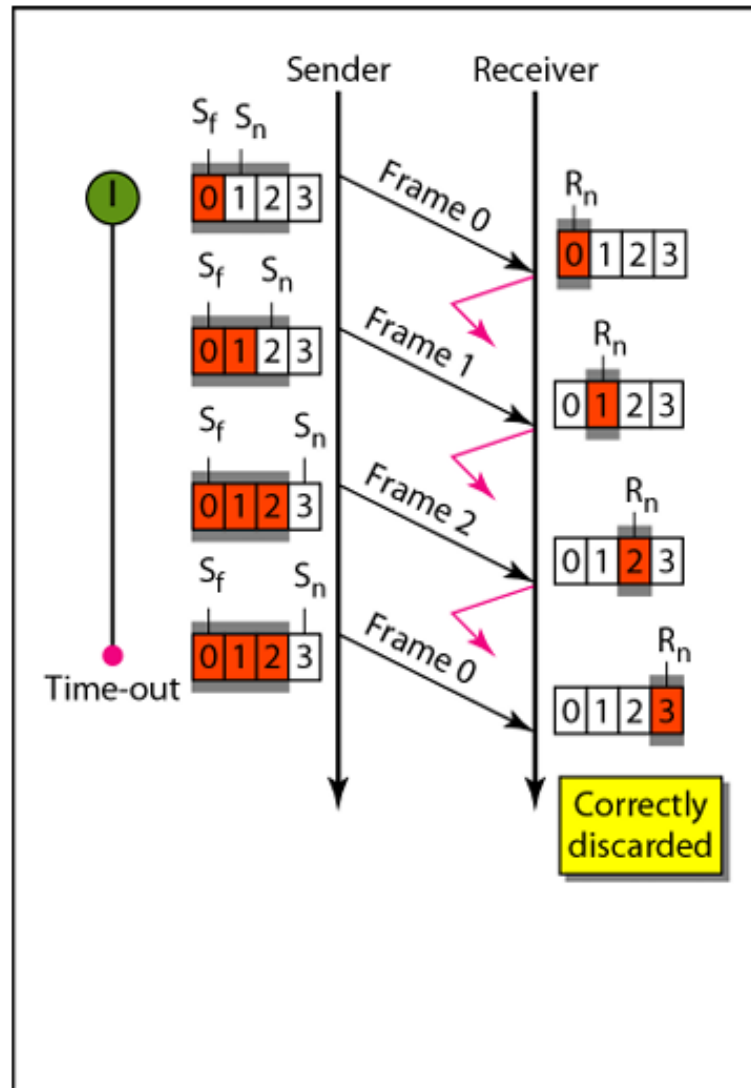


Exemplu

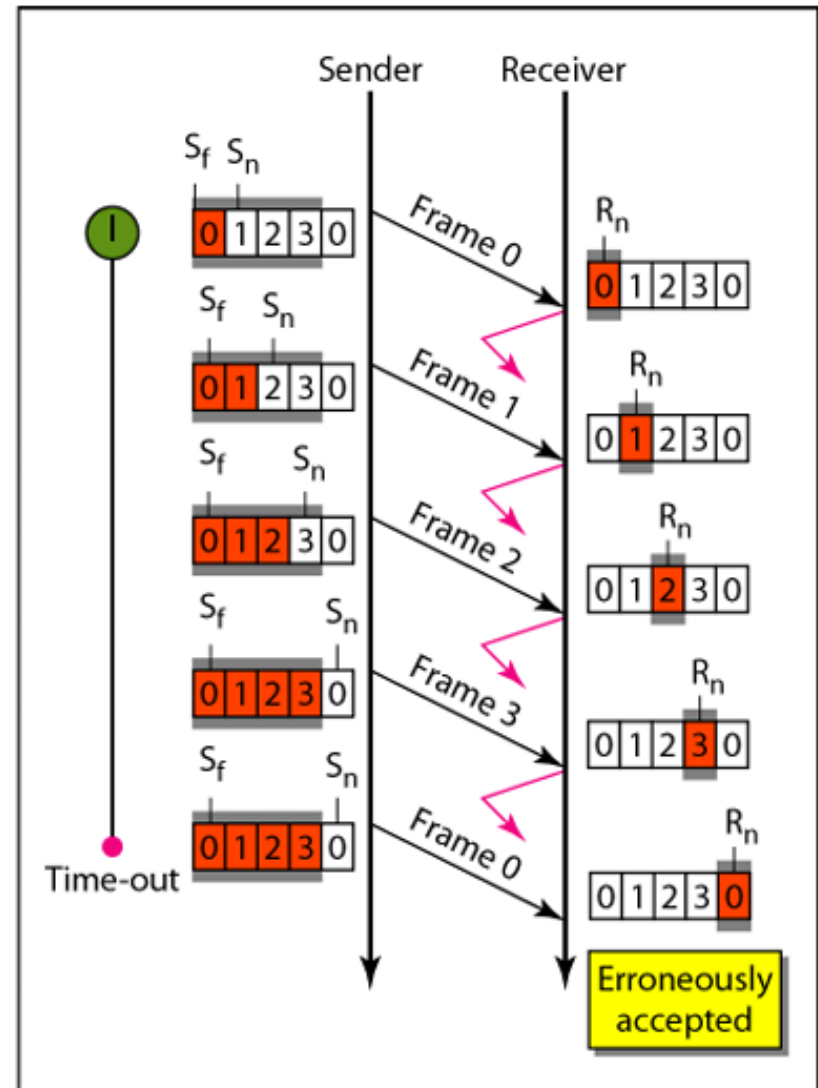
Timer-ul pentru cadrul 1 expirat



Dimensionarea ferestrei Go-Back-N



a. Window size $< 2^m$



b. Window size $= 2^m$

Functioneaza corect doar pentru fereastra $W < 2^m$

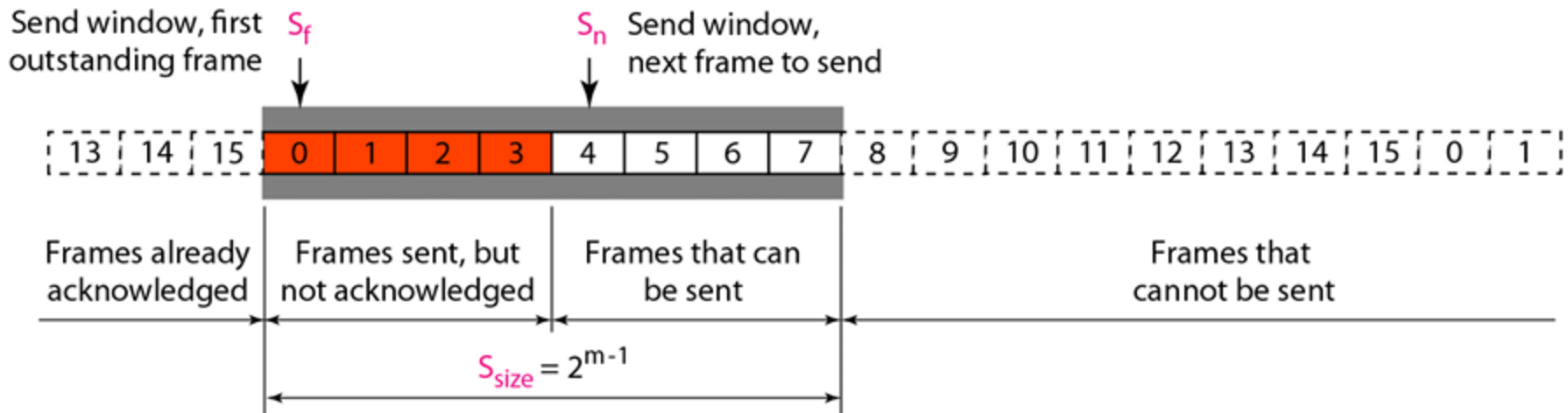
Performanța

- Mai buna decât cea a protocolului Stop and Wait
- Nu acceptă cadre recepționate care nu sunt în ordine chiar dacă ele sunt fără de erori
- Funcționează ineficient in cazul când rata erorilor de transmisie a canalului e mare

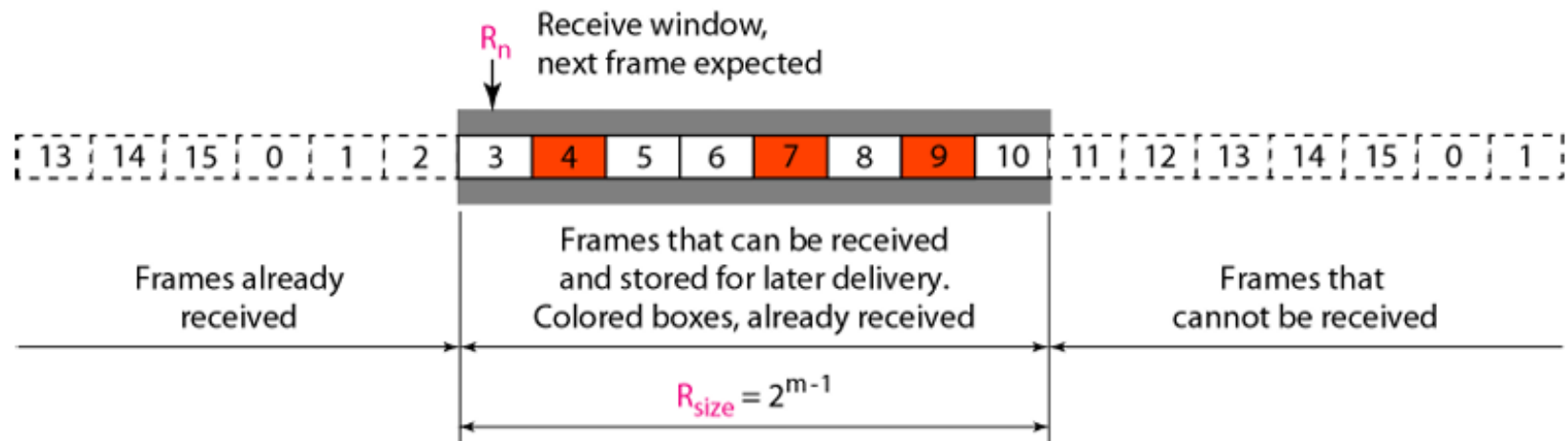
Repetare Selectivă

- **Setarea dimensiunii ferestrelor (atât pentru emițător cat și pentru receptor)**
 - Pentru numere de ordine de m biți dimensiunea ferestrei nu trebuie sa depășească 2^{m-1}
- **Funcționarea**
 - Emițătorul retransmite cadrele eronate sau pierdute în cazul când:
 - ele au fost achitate negativ
 - timer-ul lor expira
 - Receptorul acceptă cadre in dezordine și le păstrează in memoria tampon pana la sosirea cadrelor pierdute sau eronate din fereastra de recepție
 - Receptorul nu achita cadrele in dezordine ci doar le păstrează in memoria-tampon.
 - Daca receptorul primește un cadru fără de erori dar in dezordine el va trimite un ACK negativ (Negative Acknowledge (NAK)) cu numărul cadrului pentru care se cere retransmiterea.
 - NAK ameliorează performanta deoarece se transmit doar acele cadre de care este nevoie (cu timerele expirate)

Fereastra emițătorului pentru Selective Repeat



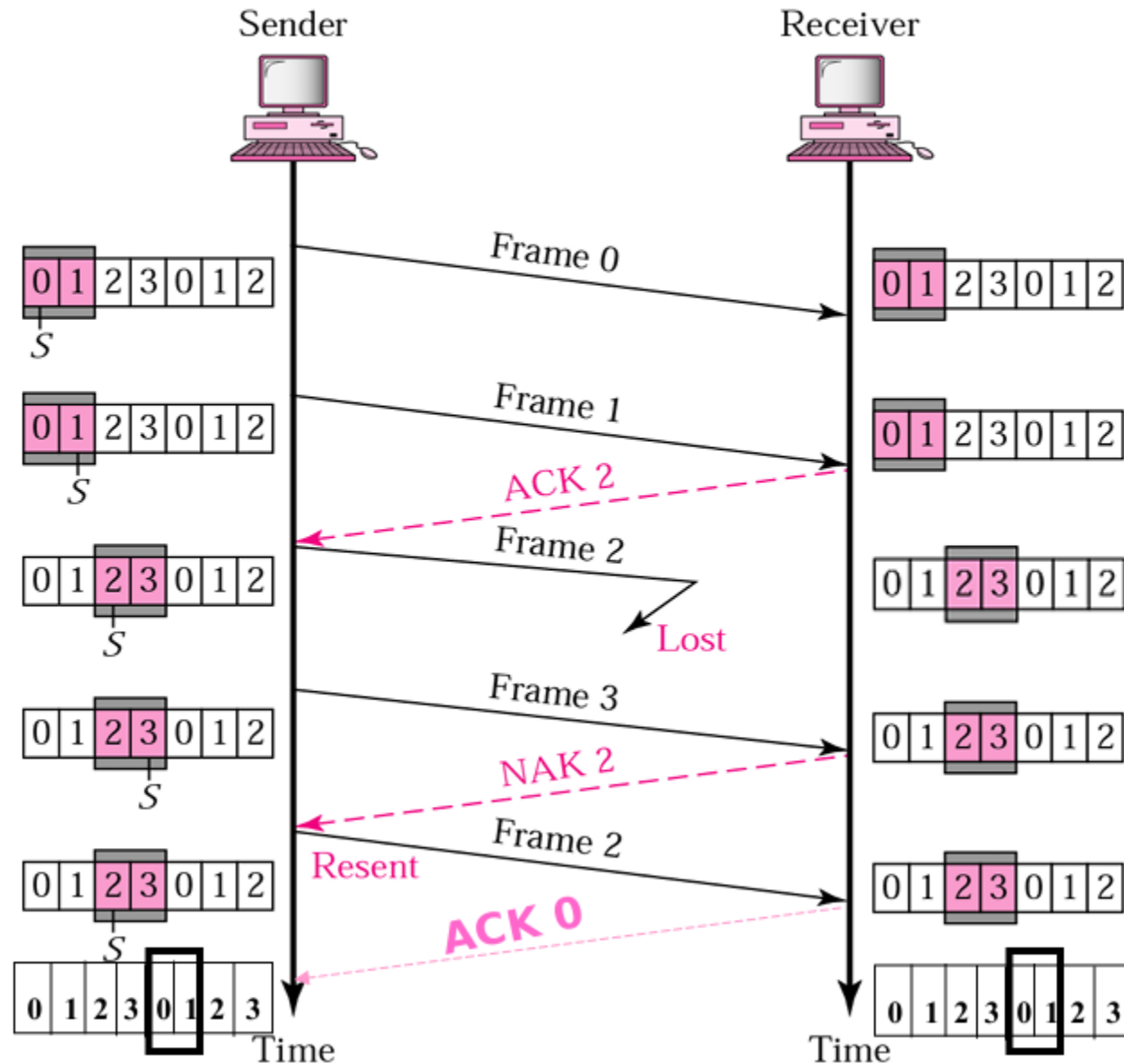
Fereastra receptorului pentru Selective Repeat



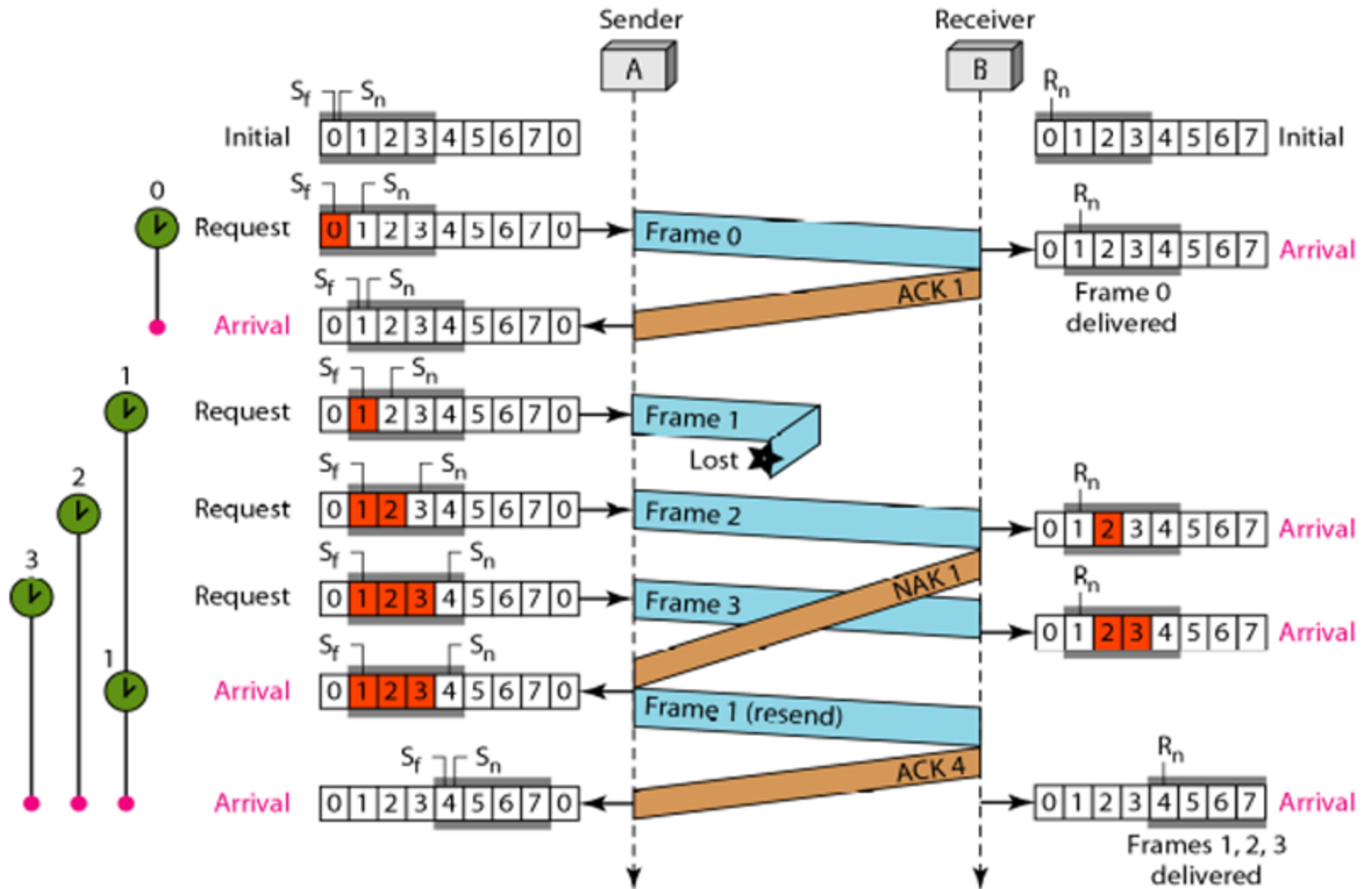
Fereastra receptorului se deplaseaza spre pozitia (i) doar cand receptorul primeste ACK i

ACK i se transmite daca toate cadrele pana la (i-1) au fost receptionate corect

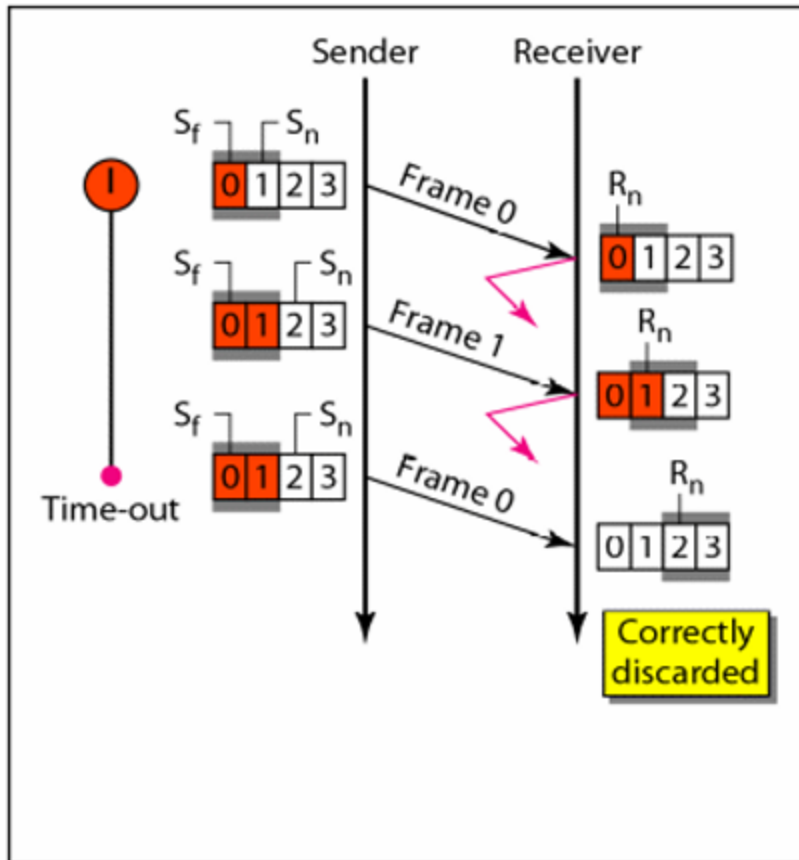
Selective Repeat cu pierderi



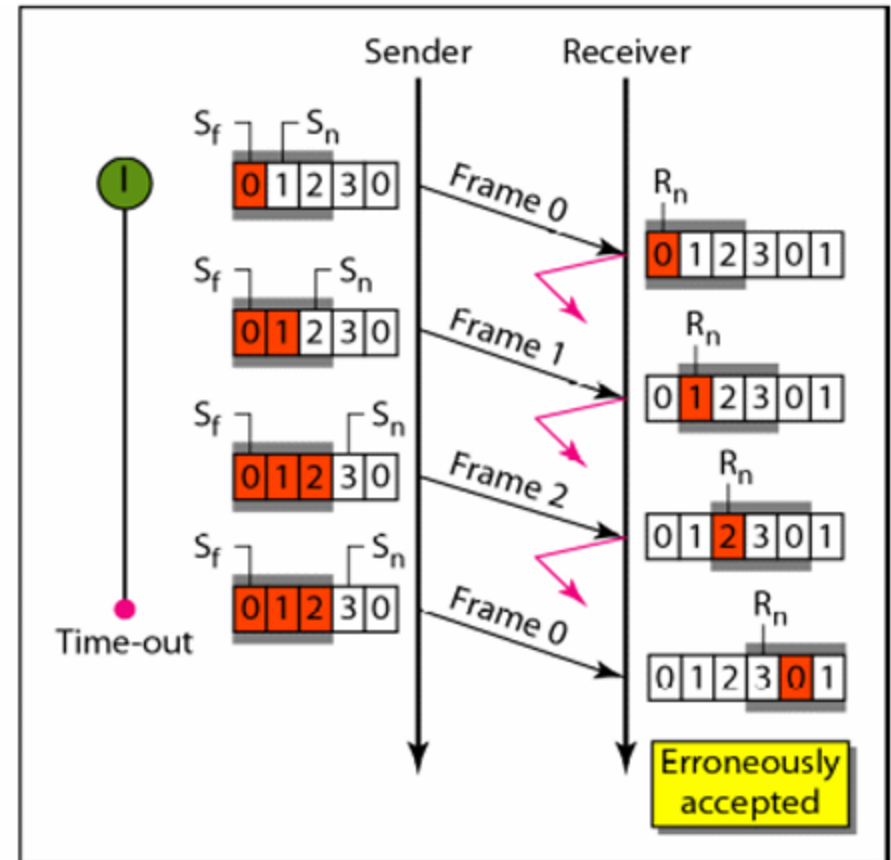
Exemplu functionare Selective Repeat



Dimensionare fereastră Selective Repeat



a. Window size = $2^m - 1$



b. Window size $> 2^m - 1$

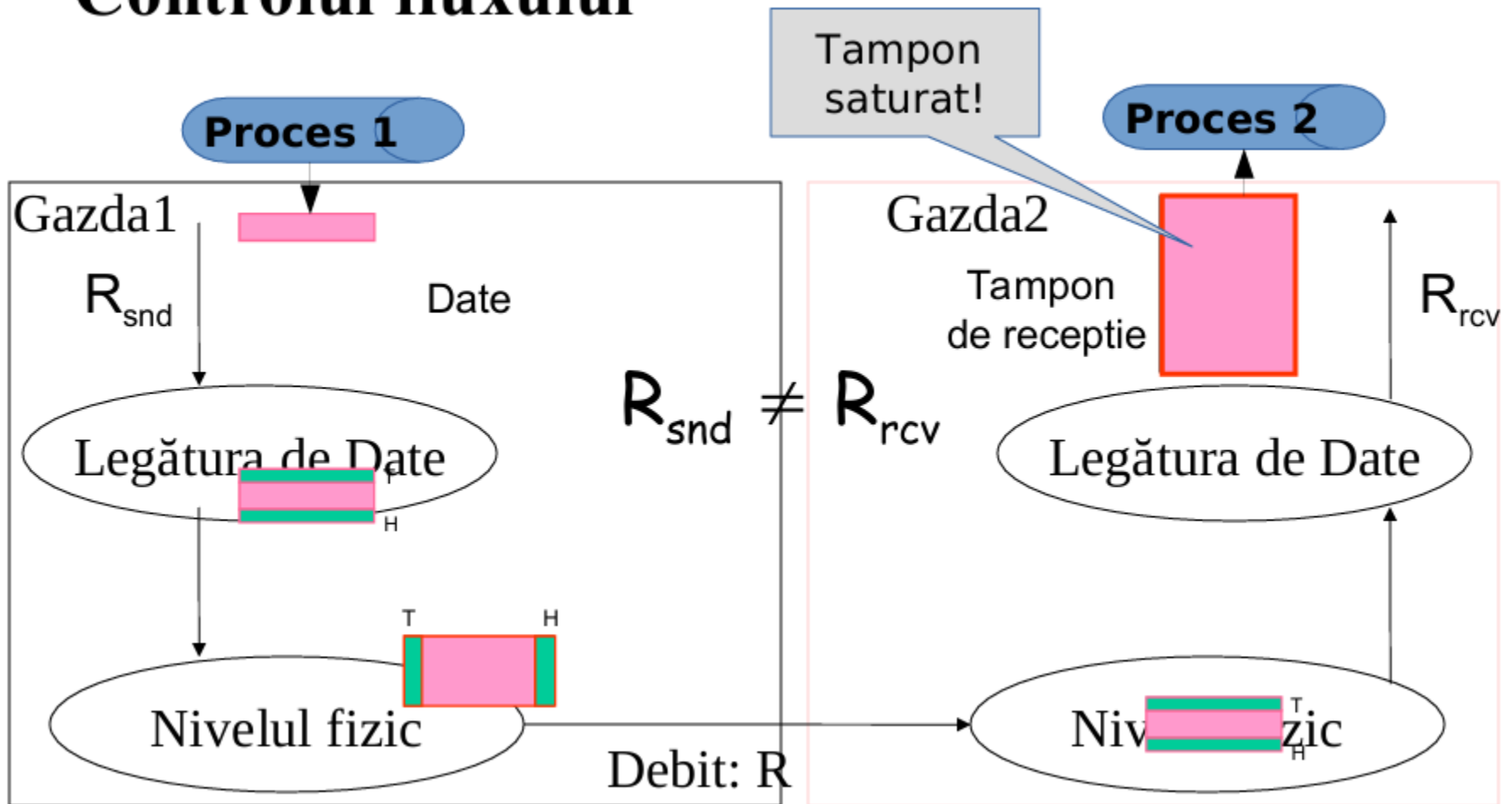
Pipelining

- Pipelining: începerea unui task înainte ca task-ul precedent să se termine
- Protocolul Stop-And-Wait nu utilizează pipelining-ul
- Pipelining-ul e implementat in protocoalele Go-Back-N si Selective Repeat
- Pipelining-ul mărește eficiența pentru ca păstrează linia ocupată in loc de a aștepta

Performanța SR

- Mai eficient decât celelalte două protocoale, deoarece reduce numărul de retransmisii pentru legături cu zgomote
- Logica de funcționare a receptorului și a emițătorului este mai complexă
 - Receptorul trebuie să fie capabil să reintroducă cadrul retransmis (pierdut, întârziat, deteriorat) în ordinea corespunzătoare în fluxul de ieșire
 - Emițătorul ar trebui să poată trimite cadre cu numărul de ordine solicitat de către receptor prin NAK
 - Necesită mai multă memorie decât Go-Back-N în partea de recepție, volumul ei depinde de dimensiunea ferestrei.

Controlul fluxului



Controlul fluxului

- Controlul fluxului permite evitarea inundării receptorului de către emițător
- Receptorul controlează emiterea cadrelor prin ACK-uri
- Protocoalele ARQ, pe lângă gestionarea erorilor, asigură și controlul fluxului.
- Există și alte metode de control al fluxului
 - ON-OFF
 - Anunțare de fereastră

