Transmisii de date digitale

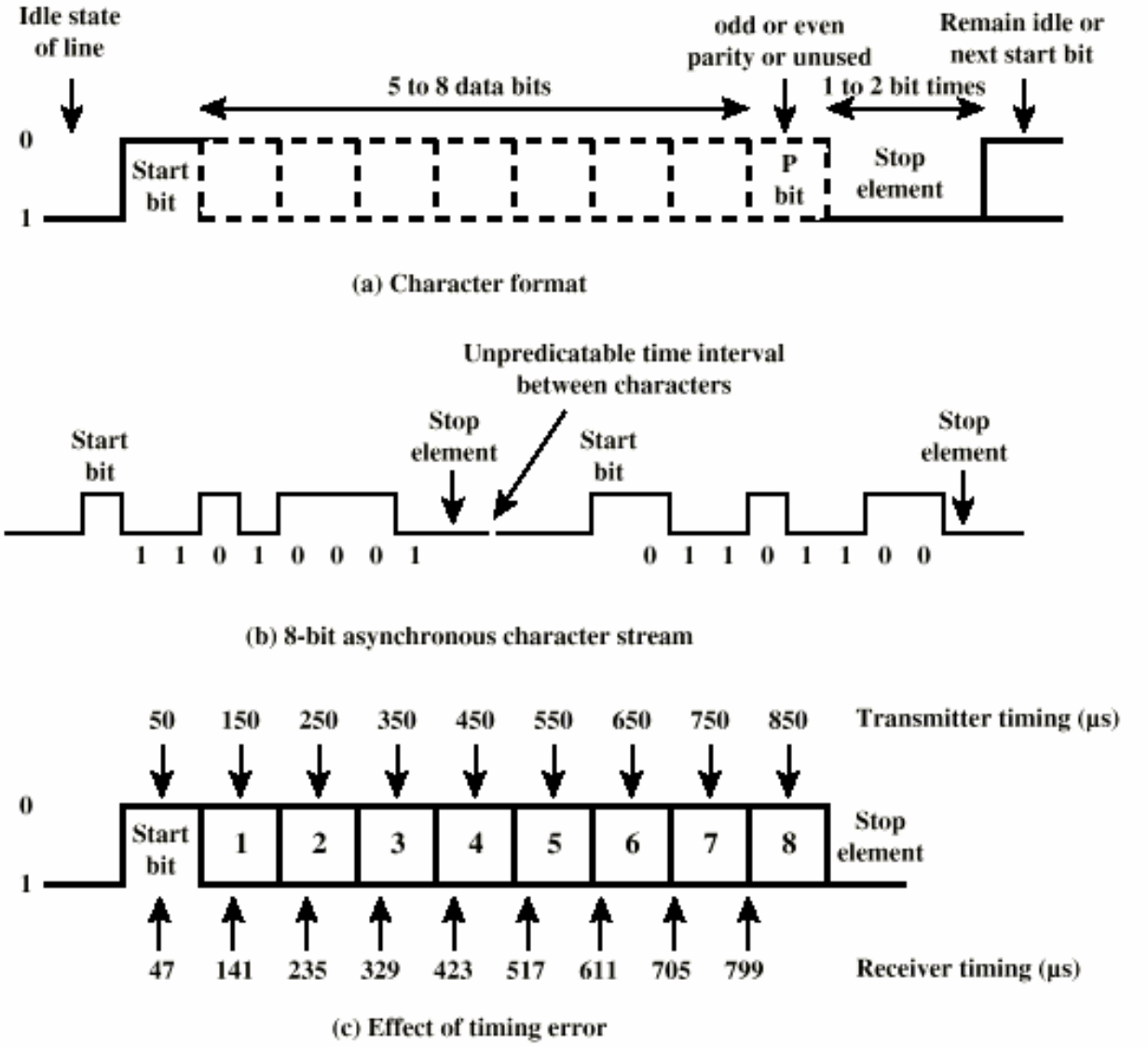
Transmisia sincronă şi asincronă

- Problemele de timing necesită un mecanism pentru sincronizare între transmiţător şi receptor
- 2 soluţii:
  - Asincron
  - Sincron

Asincron

- Datele sunt transmise un caracter după caracter
  - 5 la 8 biţi
- Sincronizarea e necesar de menţinut doar în cadrul fiecărui caracter
- Resicronizare la fiecare caracter

Asincron (diagramă)



Asincron (comportament)

- In cazul transmisiei continui, intervalul între caractere este uniform (laţimea elementului de stop)
- In starea de aşteptare (idle), receptorul caută tranziţia din 1 în 0
- Apoi eşantionează următoarele 7 intervale (lungimea caracterului)
- Apoi aşteaptă următoarea tranziţie din 1 în 0 pentru caracterul următor

- Simplu
- Ieftin
- Overhead de 2-3 biţi per caracter (~20%)
- Bun pentru date cu pauze mari (keyboard)

Sincron - Bit Level

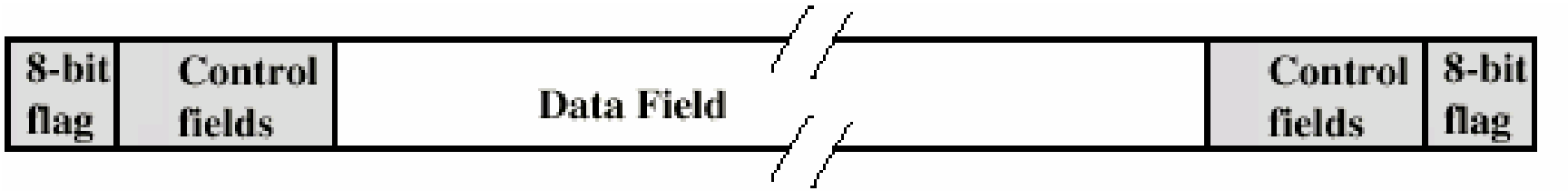
- Blocurile de date transmise fără biţi de start sau stop
- Ceasurile trebuie sa fie sincronizate
- Se pot utiliza linii de ceas dedicate
  - —Pentru distanţe scurte
  - —Afectate de imperfecţiunile mediului
- Ceas inclus în semnalul de date
  - —Codare Manchester
  - —Frecvenţă purtătoare (analog)

Sincron - Block Level

- E nevoie de indicarea blocului de start şi de stop
- Utilizează “preamble” şi “postamble”
  - Ex: serie de caractere SYN (hex 16)
  - Ex: blocuri de 11111111 terminate cu 11111110

- Mai eficient (overhead scăzut) decât asincron

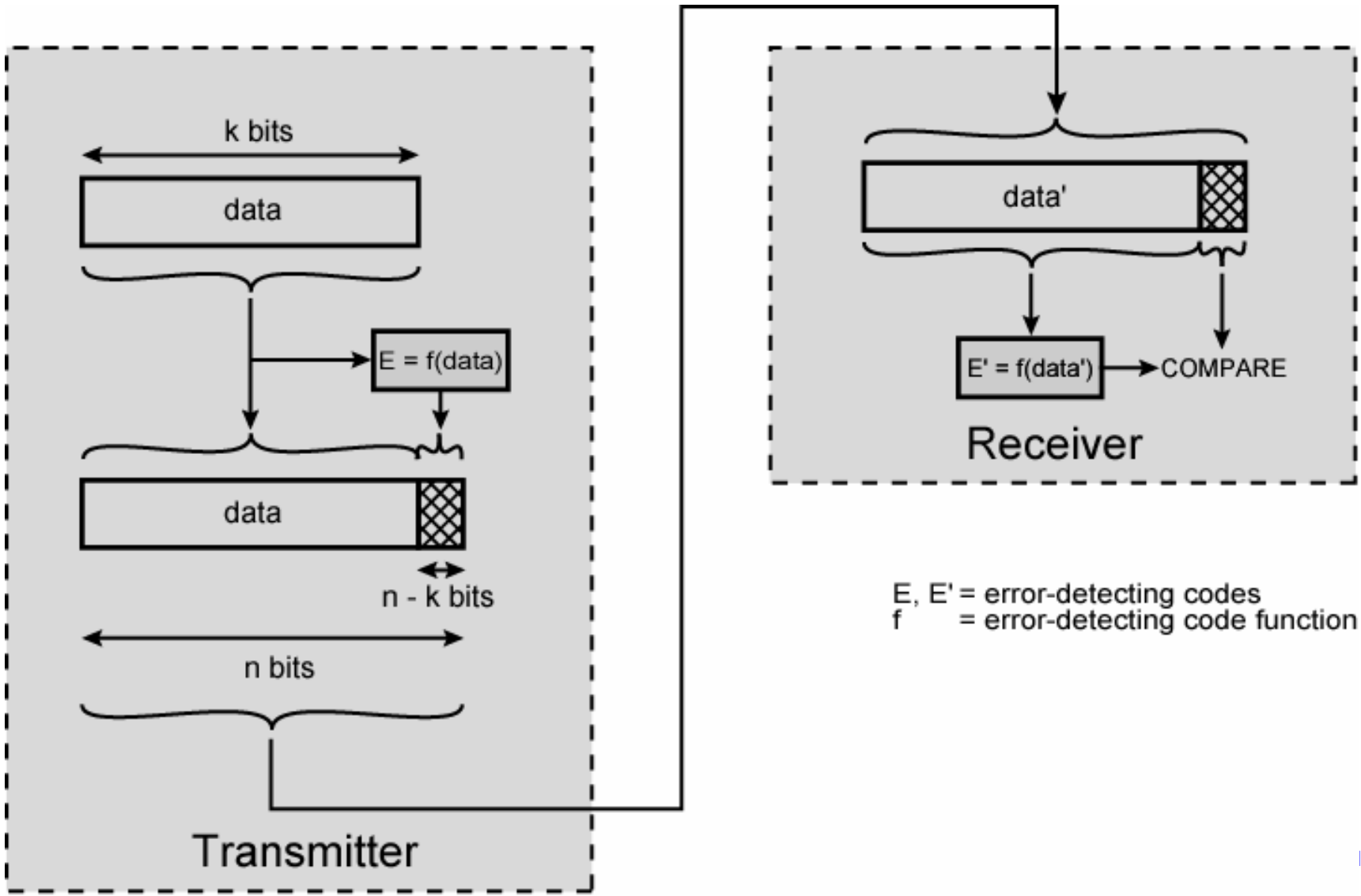
Sincron (diagramă)



Tipuri de erori

- O eroare apare când un bit este modificat între transmisie şi recepţie
- Erori de 1 bit
  - 1 bit modificat
  - Biţii adiacenţi nu sunt afectaţi
  - Zgomot alb
- Erori în burst
  - Lungime *B*
  - Secvenţă continuă de *B* biţi în care primul, ultimul şi orice număr de biţi intermediari sunt afectaţi
  - Zgomot în impuls
  - Efect mai mare la viteză mai mare

Metode de detecție a erorilor



Detecția erorilor

- Biţi suplimentari sunt adăugaţi de transmiţător pentru codul de detecţie a erorilor
- Paritate
  - —Valorea bitului de paritate este astfel încât caracterul are un număr par (paritate pară) sau impar (paritate impară) de unu
  - —Număr par de biţi eronaţi nu sunt detectaţi

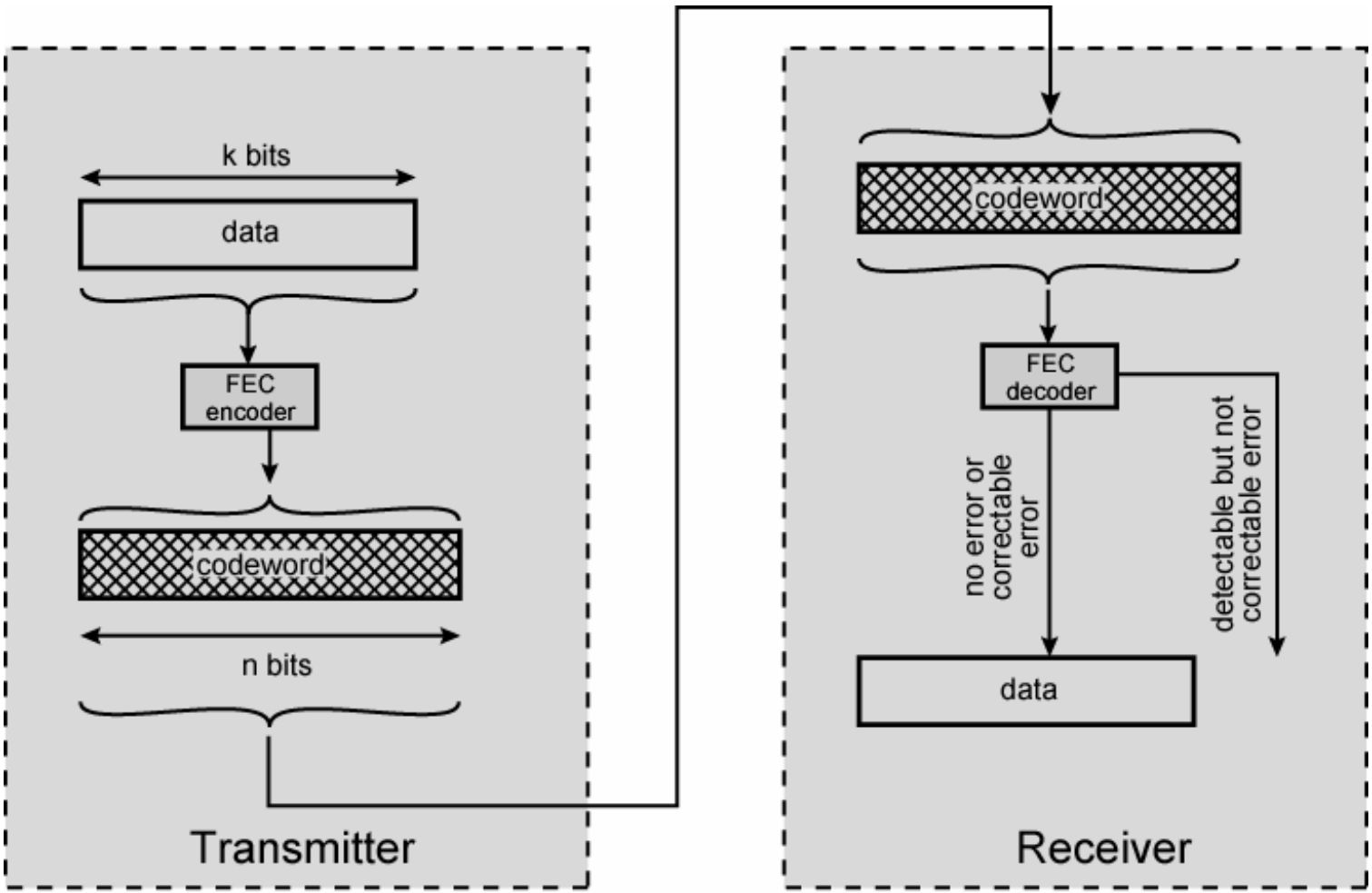
Cyclic Redundancy Check

- Pentru un număr de k biţi transmiţătorul generează o secvenţă de n biţi
- Transmite *k+n* biţi care se divide exact cu un anumit număr
- Receptorul divide secvenţa recepţionată cu acest număr:
  - Dacă nu extistă rest, se presupune că nu sunt erori

Corectarea erorilor

- De obicei, corecţia erorilor detectate se face prin retransmisie (vezi Nivelul 2 – data link)
- Nu este potrivit pentru aplicaţii wireless
  - —Rata erorilor e mare
    - Multe retransmisii
  - —Intârzierile de propagare pot fi mari (sateliţi) comparat cu timpul de transmisie a unui pachet
    - Ar rezulta rezulta retransmisia pachetului eronat şi a numeroase pachete ulterioare
- Trebuie corectate erorile pe baza biţilor recepţionaţi

Diagrama metodei de corectare a erorilor



Corectarea erorilor

- Fiecare bloc de k biţi este mapat într-un bloc de n biţi ($n > k$)
  - Cuvânt de cod
  - Forward error correction (FEC) encoder
- Cuvântul de cod (codeword) este transmis
- Şirul de biţi recepţionaţi este similar cu cel transmis, dar poate conţine erori
- Cuvântul de cod recepţionat este procesat de FEC decoder
  - Dacă nu sunt erori, blocul de date original este recepţionat
  - Unele configuraţii de erori pot fi detectate şi corectate
  - Unele configuraţii de erori pot fi detectate dar nu corectate
  - Unele configuraţii de erori (rare) nu pot fi detectate
    - Rezultă în date eronate la ieşirea FEC

Modul de funcționare a corecției de erori

- Adaugă date redundante la mesajul transmis
- Poate reface originalul la apariţia unui anumit nivel de erori
- Exemplu: pentru corectarea erorilor pe bloc
  - In general, se adaugă ($n - k$) biţi la sfârşitul blocului
    - Rezultă un bloc de n biţi (cuvânt de cod)
    - Toţi k biţi originali sunt incluşi în cuvântul de cod
  - Unele FEC mapează k biţi în n biţi astfel încât originalul nu mai apare

Legătura de date

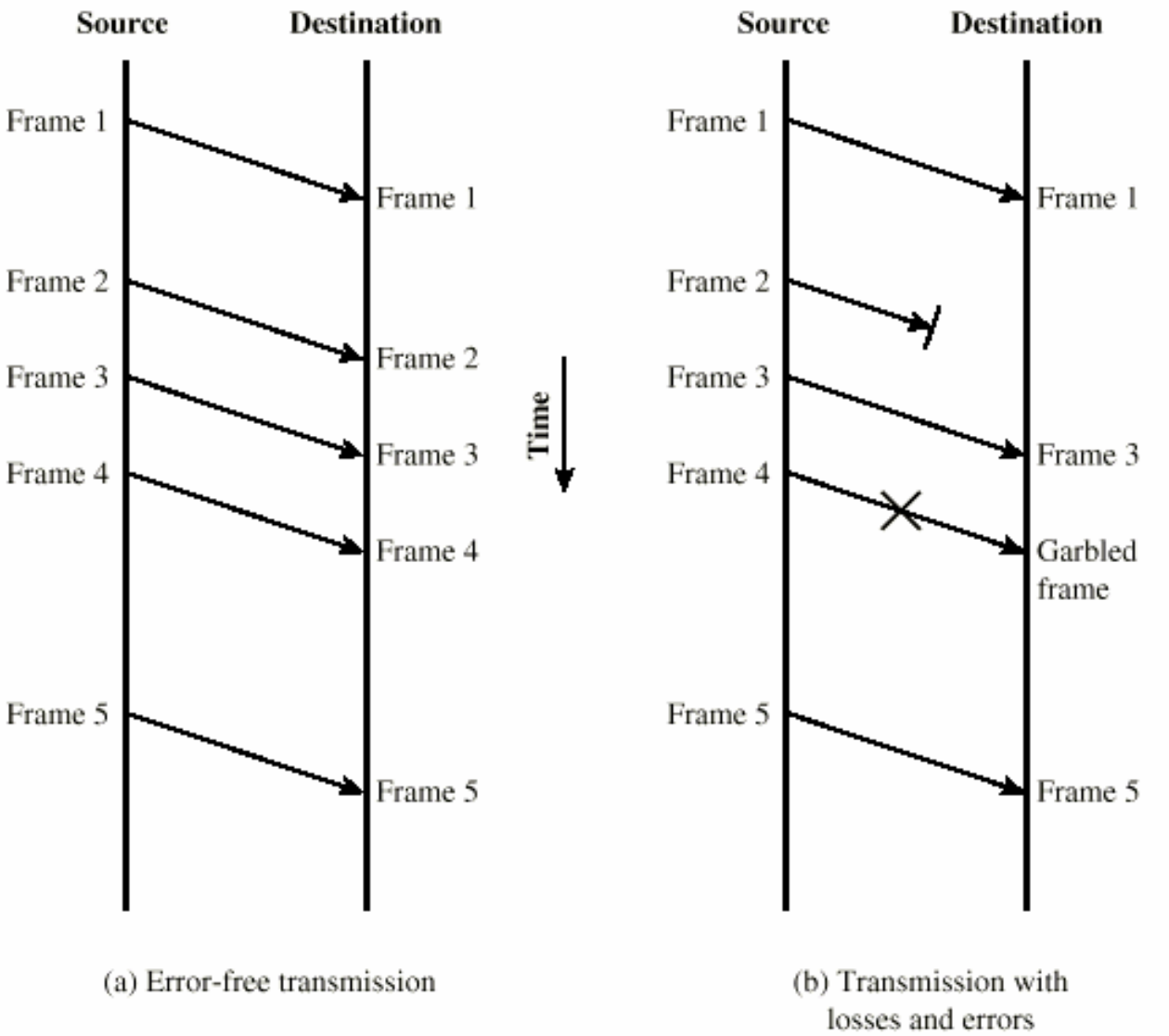
Cerintele nivelului : Data Link

- Sincronizarea frame-urilor;
- Flow control;
- Controlul erorilor;
- Adresarea;
- Control si date pe aceeasi legatura;
- Managementul legaturii;
- HDLC

Controlul fluxului de date

- Asigură ca entitatea de transmisie nu depăşeşte capacitatea enităţii de recepţie
  - —Previne umplerea buffer-ului de recepţie
- Timpul de transmisie
  - —Timpul necesar pentru a emite toţii biţii prin mediu
- Timpul de propagare
  - —Timpul necesar unui bit pentru a traversa legătura

Model de transmisie pe cadre



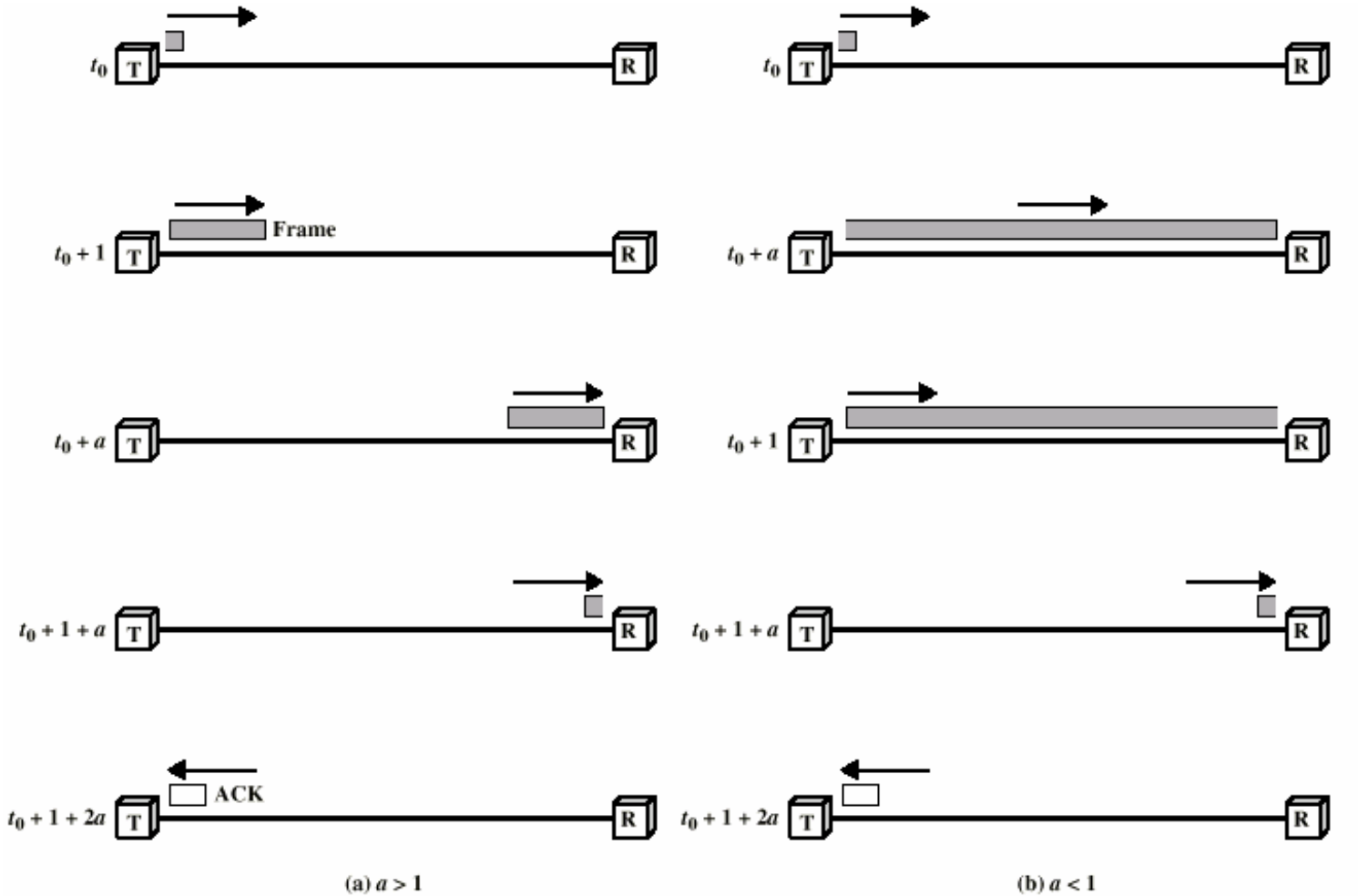
Stop and wait

- Sursa trimite un cadru
- Destinaţie recepţionează cadrul şi răspunde cu confirmare
- Sursa aşteaptă confirmarea înainte de a trimite cadrul următor
- Destinaţia poate opri fluxul de date dacă nu mai trimite confirmarea
- Merge bine pentru puţine cadre mari

Fragmentarea

- Blocurile mari de date pot fi sparte în cadre mici
  - —Mărimea buffer-ului este limitată
  - —Erorile sunt detectate mai repede (când tot cadrul este recepţionat)
  - —În caz de erori, este necesară retransmiterea unui bloc mic
  - —Evită ocuparea mediului de către o staţie pentru un timp îndelungat
- Protocolul Stop and wait devine inadecvat

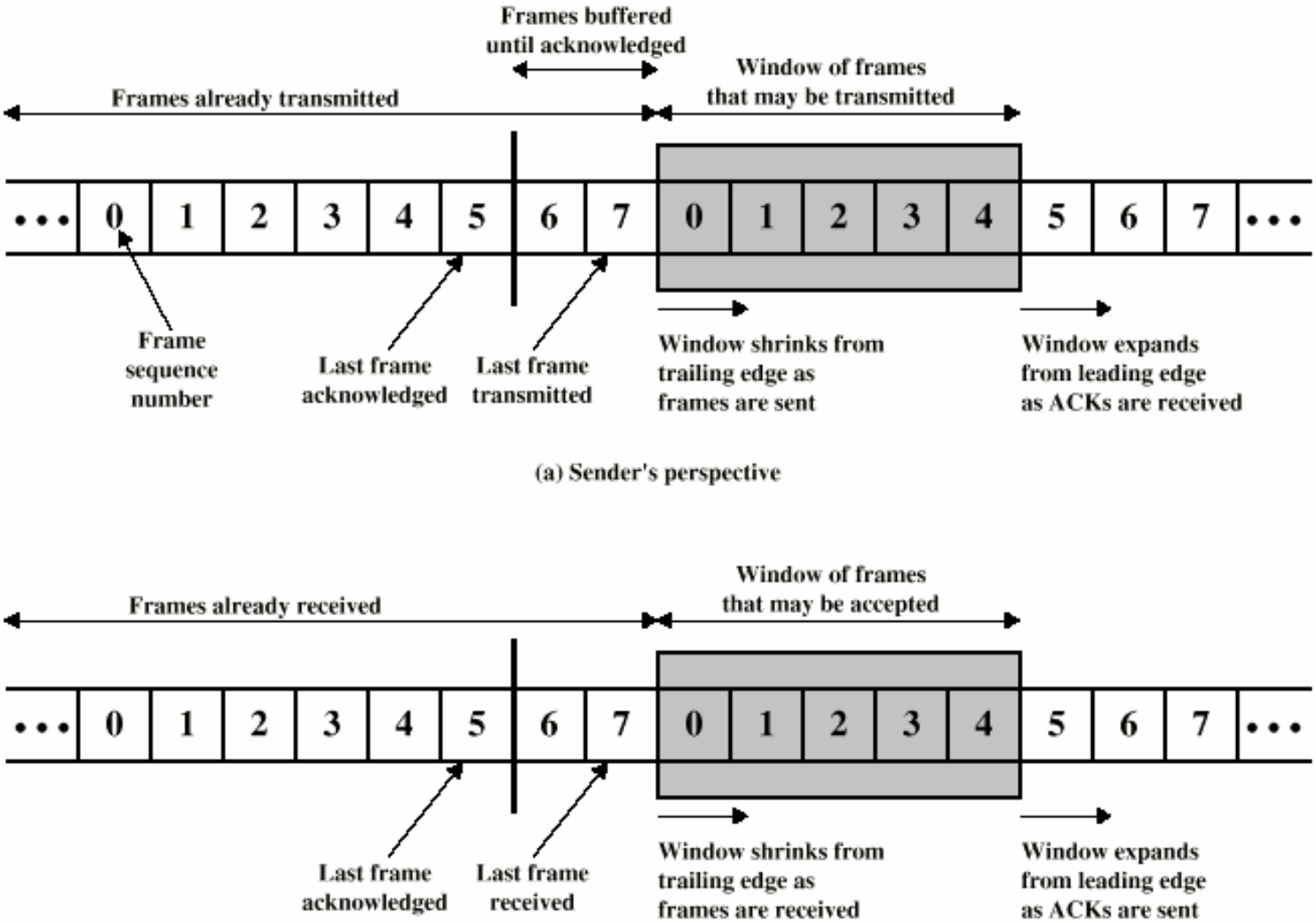
Utilizarea protocolului stop and wait



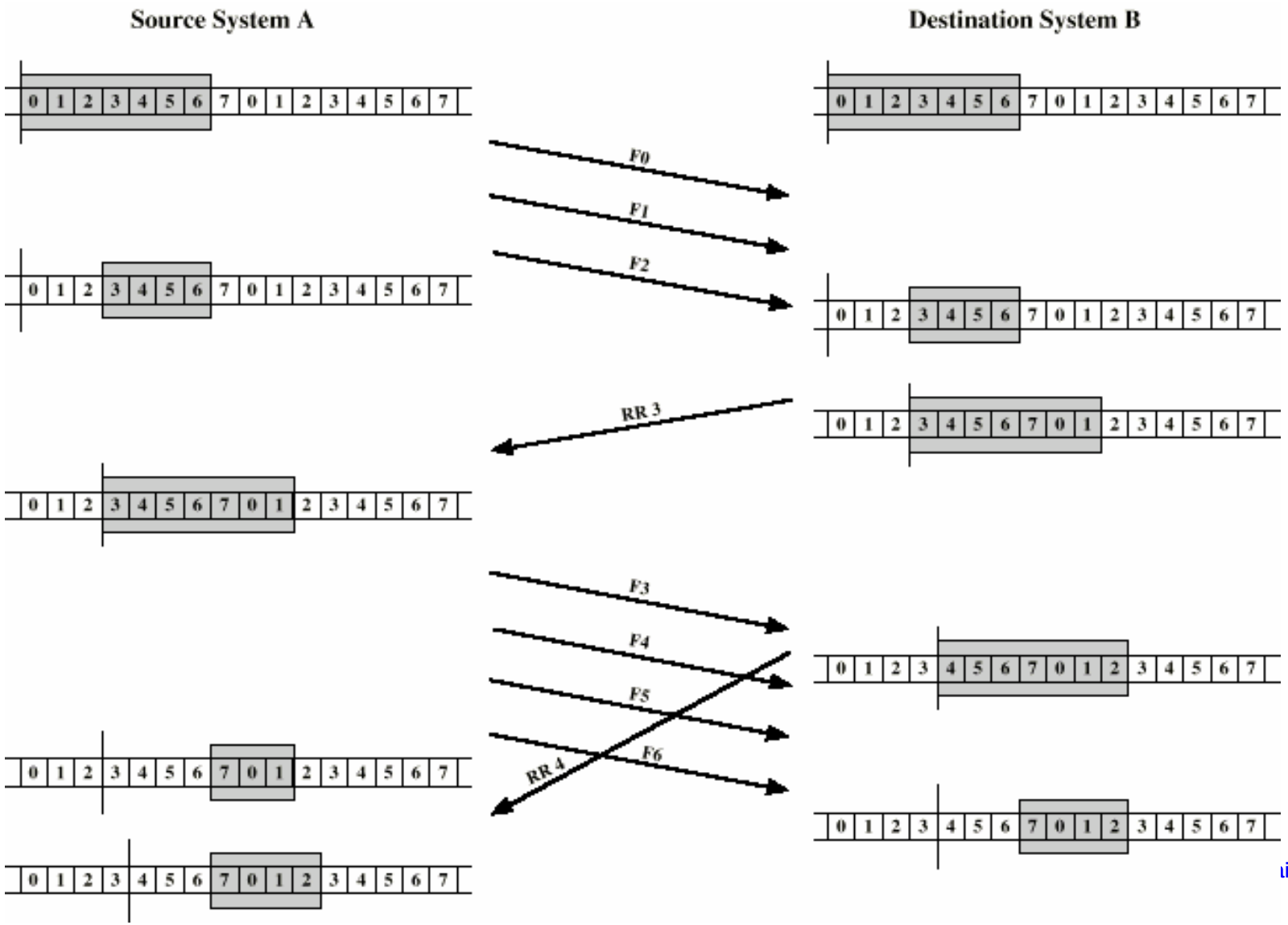
Controlul de flux cu fereastră alunecătoare

- Mai bine "sliding window"
- Permite ca mai multe cadre să fie în tranzit
- Receptorul are un buffer de lungime W
- Transmiţătorul poate trimite până la W cadre fără ACK
- Fiecare cadru primeşte un număr
- ACK include numărului cadrului următor aşteptat
- Numărul secvenţei este limitat de dimensiunea câmpului (k)
  - Cadrele sunt numărate modulo 2^k

Diagrama sliding window



Exemplu de funcționare a protocolului



Îmbunătățiri pentru sliding windows

- Receptorul poate confirma cadre fără a permite transmisii ulterioare (Receive Not Ready)
- Trebuie să trimită o confirmare normală pentru reluarea transmisiei
- Dacă transmisia e de tip duplex, se utilizează piggybacking
  - Dacă nu sunt date de transmis, se trimite un cadrul de confirmare
  - Dacă nu este confirmare de transmis, se trimite ultimul număr confirmat din nou

Detecția erorilor

- Biţi adiţionali adăugaţi pentru codul de detecţie a erorilor
- Bit de paritate
- Cyclic Redundancy Check

Controlul erorilor

- Tipuri de erori:
  - Cadre pierdute
  - Cadre stricate
- Tehnici de control al erorilor
  - Confirmare pozitivă
  - Retransmisie după timeout
  - Confirmare negativă şi retransmisie

Automatic Repeat Request (ARQ)

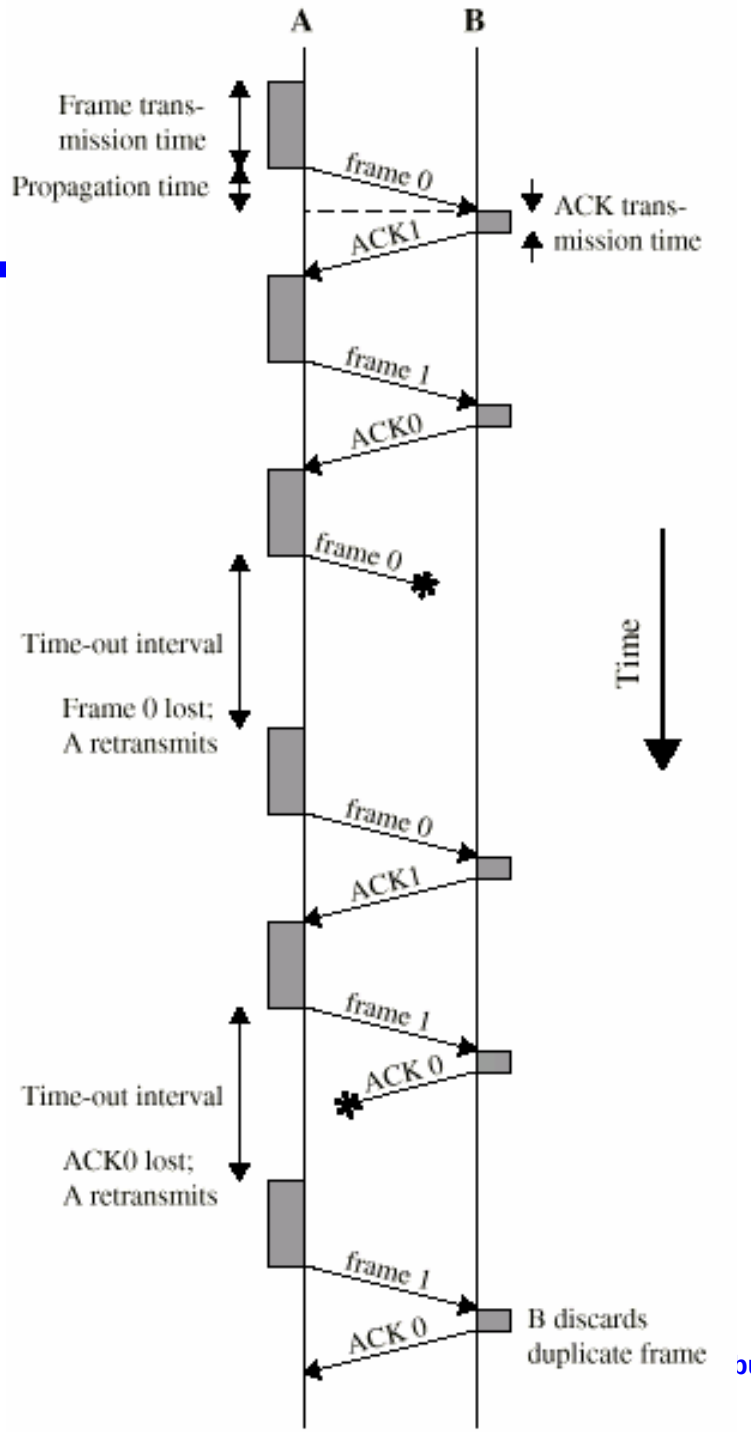
- Stop and wait ARQ
- Go back N ARQ
- Selective reject (retransmisie selectivă) ARQ

Stop and wait

- Sursa transmite un singur cadru
- Aşteaptă ACK
- Dacă cadrul recepţionat este stricat, se aruncă
  - Transmiţătorul are timeout
  - Dacă nu se confirmă până la timeout, se retransmite
- Dacă ACK e afectat, transmiţătorul nu îl recunoaşte
  - Transmiţătorul va retransmite
  - Se recepţionează 2 copii ale cadrului
  - Se foloseşte ACK0 şi ACK1

Diagrama stop and wait

- 2 frame-uri corecte;
- frame pierdut;
- time out, retransmite frame;
- pierdere ack;
- time out, retransmite farme;
- primeste ack.



Go Back N (1)

- Bazat pe protocolul sliding window
- Daca nu sunt erori, ACK ca de obicei cu urmatorul cadru asteptat
- Daca sunt erori, răspunde cu respingerea cadrului
  - —Aruncă cadrul eronat şi toate cadrele următoare până ce cadrul eronat e recepţionat corect
  - —Transmiţătorul trebuie să retransmită (sa pastreze cardele care nu au primit ack) acel cadru şi toate cadrele următoare (go back N)

Go Back N – Cadre eronate

- Receptorul detectează eroare în cadrul j
- Receptorul trimite rejection-*i*
- Transmiţătorul primeşte rejection-*i*
- Transmiţătorul retransmite cadrul j şi toate cadrele următoare

Go Back N – Cadre pierdute (1)

- Cadrul *i* se pierde
- Transmiţătorul trimite *i+1*
- Receptorul primeşte cadrul *i+1* în afara ordinii aşteptate
- Receptorul trimite reject *i*
- Transmiţătorul merge înapoi la cadrul *i* şi retransmite

Go Back N – Cadre pierdute (2)

- Cadrul i se pierde şi nu se trimite alt cadru
- Receptorul nu primeşte nimic şi nu răspunde nici cu confirmare, nici cu rejecţie
- Transmiţătorul atinge time-out şi trimite un cadru de confirmare
- Receptorul interpretează asta ca o comandă pe care o confirmă cu cadrul următor pe care îl aşteaptă (cadrul i)
- Atunci transmiţătorul retransmite cadrul i

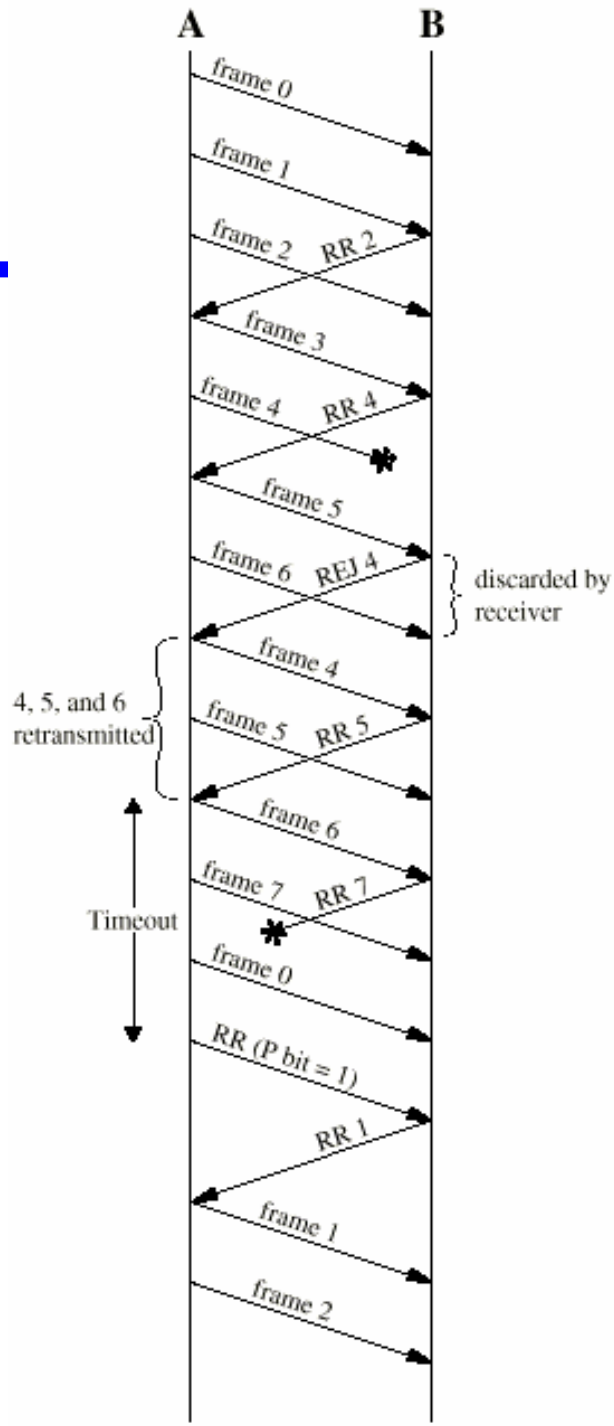
Go Back N – Confirmare eronată

- Receptorul primeşte cadrul i şi trimite confirmare care se pierde
- Confirmările sunt cumulative şi următoarea confirmare (i+n) poate sosi înainte ca transmiţătorul să ajungă la time-out pentru cadrul i
- Dacă transmiţătorul ajunge la time-out, trimite un cadru de confirmare ca înainte
- Aceasta se poate repeta de un număr de ori, înainte ca o procedură de reset să fie activată

Go Back N – Rejecție eronată

- Vezi cazul: cadre pierdute (2)

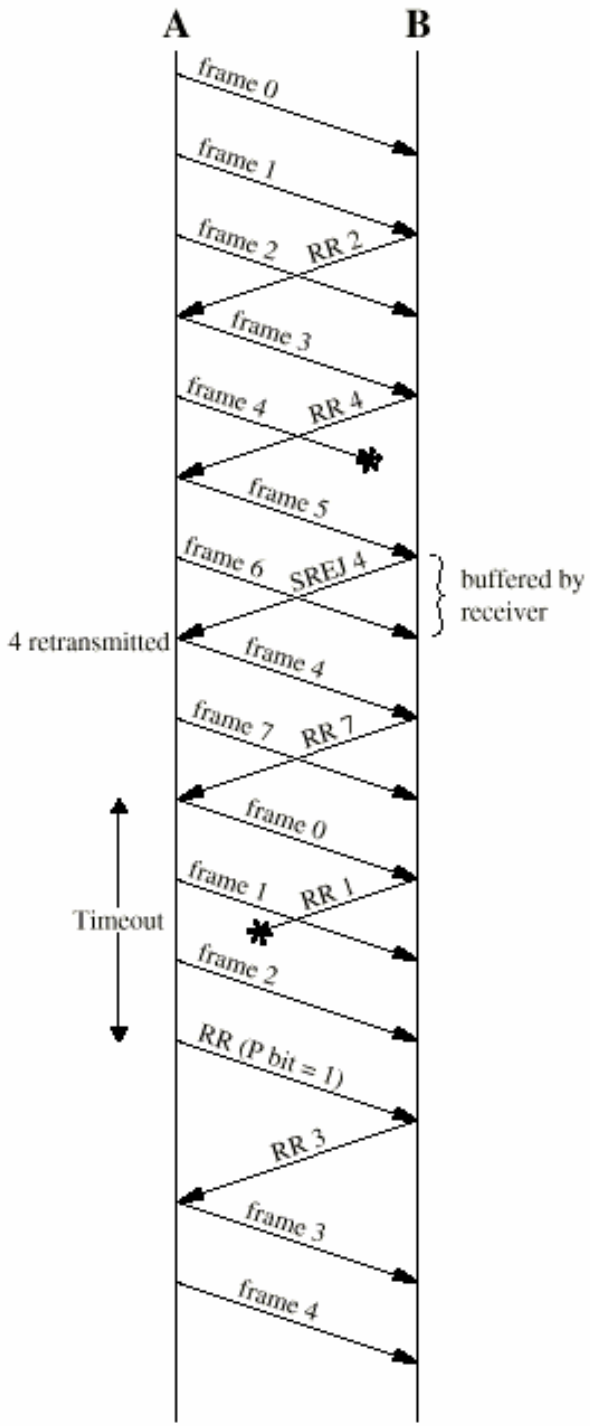
Go Back N Diagramă



Rejecție selectivă

- Denumită şi retransmisie selectivă
- Numai cadrele rejectate se retransmit
- Cadrele următoare sunt acceptate de receptor şi memorate
- Minimizează retransmisia
- Receptorul trebuie să aibă un buffer destul de mare
- Mai multă logică la transmiţător

Rejecție selectivă - diagramă



High Level Data Link Control

- HDLC
- ISO 33009, ISO 4335
- 3 - Tipuri de statii
- 2 – Tipuri de configuratii de legatura
- 3 - Moduri de transmisiune

Tipuri de staţii HDLC

- Staţie primară
  - —Controlează modul de operare a legăturii
  - —Transmite cadre numite comenzi
  - —Menţine legături logice separate cu fiecare staţie secundară
- Staţie secundară
  - —Funcţionează sub controlul staţiei primare
  - —Transmite cadre numite răspunsuri
- Staţie combinată
  - —Transmite comenzi şi răspunsuri

Configuraţii ale legăturii HDLC

- Neechilibrat
  - —O staţie primară şi mai multe staţii secundare
  - —Poate fi full duplex sau half duplex
- Echilibrat
  - —Două staţii combinate
  - —Poate fi full duplex sau half duplex

Moduri de transfer HDLC (1)

- Normal Response Mode (NRM)
  - Configuraţie neechilibrată
  - Staţia primară iniţiază transferul spre staţia secundară
  - Staţia secundară poate transmite date doar ca răspuns a unei comenzi de la staţia principală

Moduri de transfer HDLC (2)

- Asynchronous Balanced Mode (ABM)
  - —Configuraţie echilibrată
  - —Orice staţie poate iniţia transmisia fără a cere permisiunea
  - —Cel mai folosit
  - —Nu e necesar polling

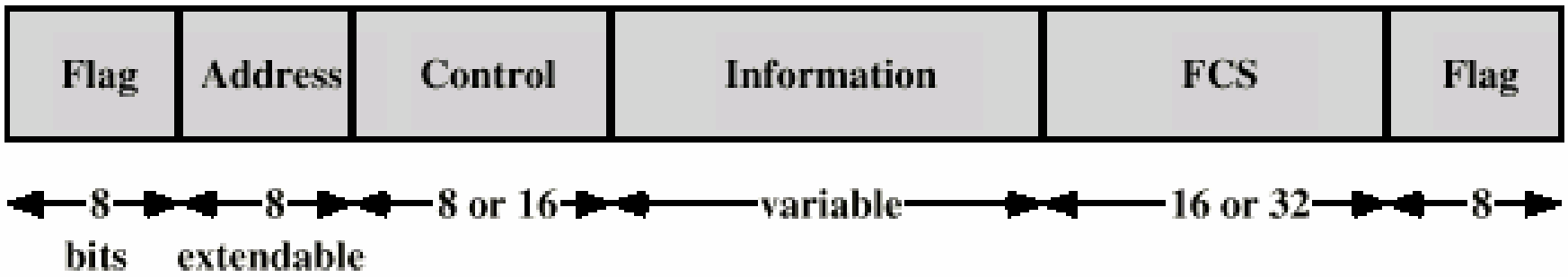
Moduri de transfer HDLC (3)

- Asynchronous Response Mode (ARM)
  - —Configuraţie neechilibrată
  - —Staţia secundară poate iniţia transmisia fără permisiunea staţiei primare
  - —Staţia primară e responsabilă pentru linie
  - —Rar utilizată

Structura cadrului

- Transmisie sincronă
- Toate transmisiile se fac în cadre
- Un singur format de cadru pentru schimbul de date şi control

Structura cadrului



(a) Frame format

Flag

- Delimitează cadrul
- 01111110
- Poate închide un cadru şi deschide altul
- Receptorul urmăreşte secvenţa de flag pentru sincronizare
- Pentru a evita confuzia cu datele de tip 01111110, se foloseşte “bit stuffing”
  - 0 inserat după fiecare secvenţă de cinci 1
  - Dacă receptorul detectează cinci 1, verifică bitul următor
  - Dacă e 0, se şterge
  - Dacă e 1 şi al şaptelea bit e 0, se acceptă un flag
  - Dacă bitul al şaselea şi al şaptelea e 1, se semnalizează “abort”

Bit Stuffing

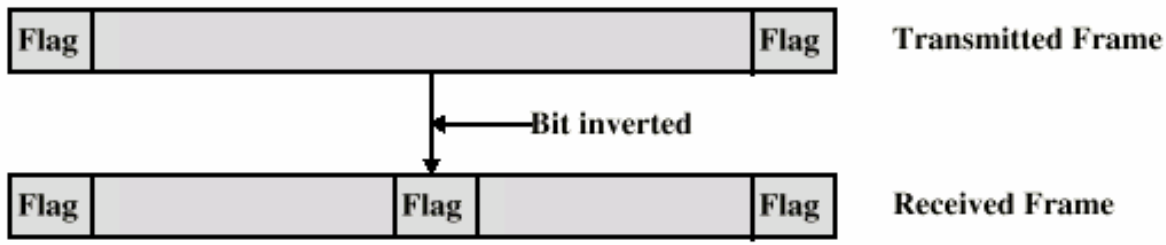
Original Pattern:

```
111111111111011111101111110
```

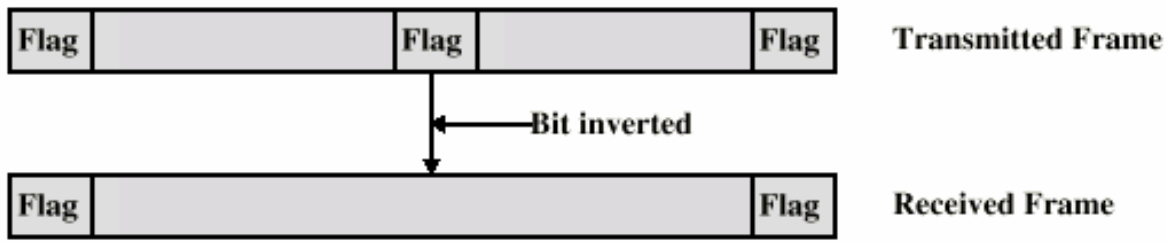
After bit-stuffing

```
1111101111110110111110101111101010
```

(a) Example



(b) An inverted bit splits a frame in two



(c) An inverted bit merges two frames

Adresa

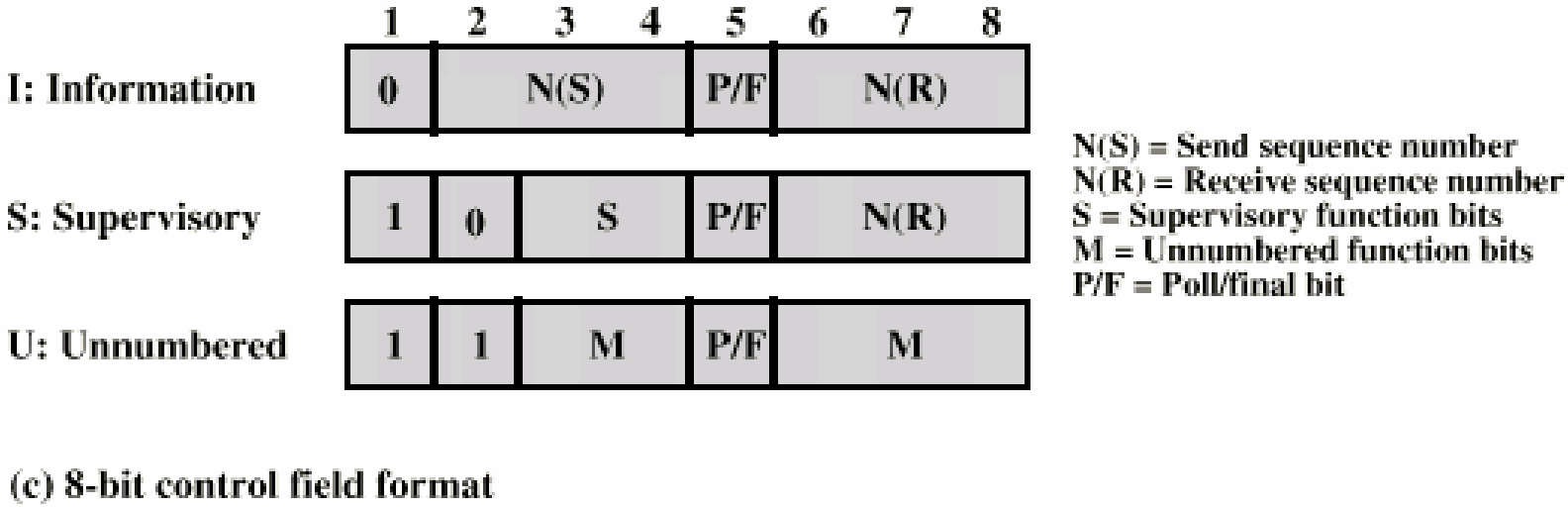
- Identifică staţia secundară care trimite sau recepţionează cadrul
- De obicei are 8 biţi
- Se poate extinde la multipli de 7 biţi
  - —LSB din fiecare octet indică dacă este ultimul octet (1) sau nu (0)
- 11111111 este broadcast

(b) Extended Address Field

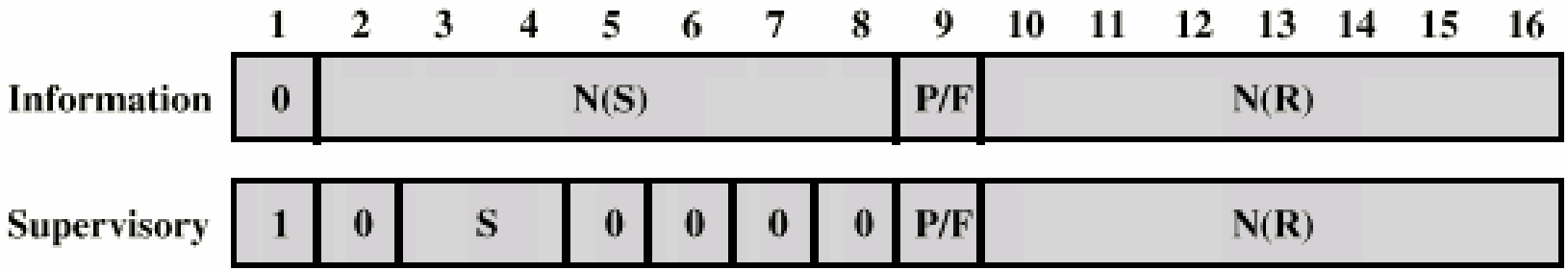
Control

- Diferiţi în funcţie de tipul cadrului
  - —Information – datele de transmis către user (nivelul superior)
    - Controlul de flux şi de erori folosind piggyback
  - —Supervisory – ARQ când nu se foloseşte piggyback
  - —Unnumbered – control suplimentar al legăturii
- Primul sau primii 2 biţi determină tipul cadrului

Control - diagramă



(c) 8-bit control field format



(d) 16-bit control field format

Poll/Final Bit

- Folosit în funcţie de context
- Cadre de comandă:
  - P bit
  - 1 cere răspuns de la celălalt capăt (poll)
- Cadre de răspuns:
  - F bit
  - 1 indică răspuns la o comandă

Informația

- Numai în cadre de tip information şi unele cadre de tip unnumbered
- Trebuie să conţină un număr întreg de octeţi
- Lungime variabilă

Frame Check Sequence

- FCS
- Detecţia erorilor
- 16 bit CRC
- Optional 32 bit CRC

Modul de operare HDLC

- Schimb de cadre de tip information, supervisory si unnumbered între două staţii
- Trei faze
  - Iniţializare
  - Transfer de date
  - Deconectare

Inițializarea

- Cerut de orice parte
- 3 scopuri:
  - Cerere de iniţializare spre cealaltă parte
  - Specifică modul de transfer (NRM, ABM, ARM)
  - Specifică dacă numărul de secvenţă e pe 3 sau 7 biţi
- Dacă cealaltă parte acceptă, trimite un cadru tip UA (unnumbered acknowledge)
- Dacă cererea e respinsă, trimite un cadru de tip DM (disconnect mode)

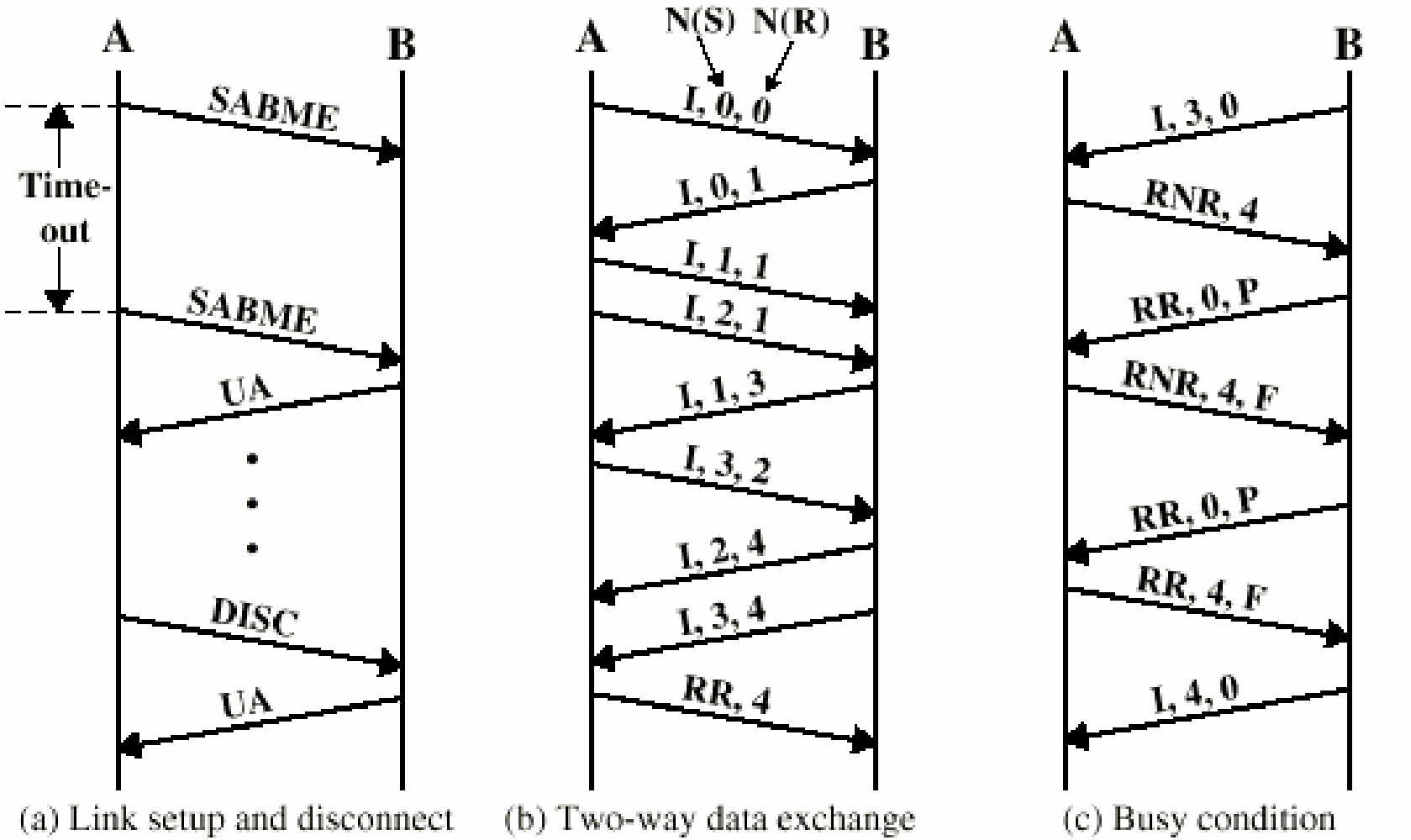
Transferul de date

- Transfer de cadre tip I, începând cu numărul de secvenţă 0
- Cadre de tip S, pentru controlul de flux şi a erorilor:
  - RR (receive ready)
  - RNR (receive not ready)
  - REJ (reject)
  - SREJ (selective reject)

Deconectarea

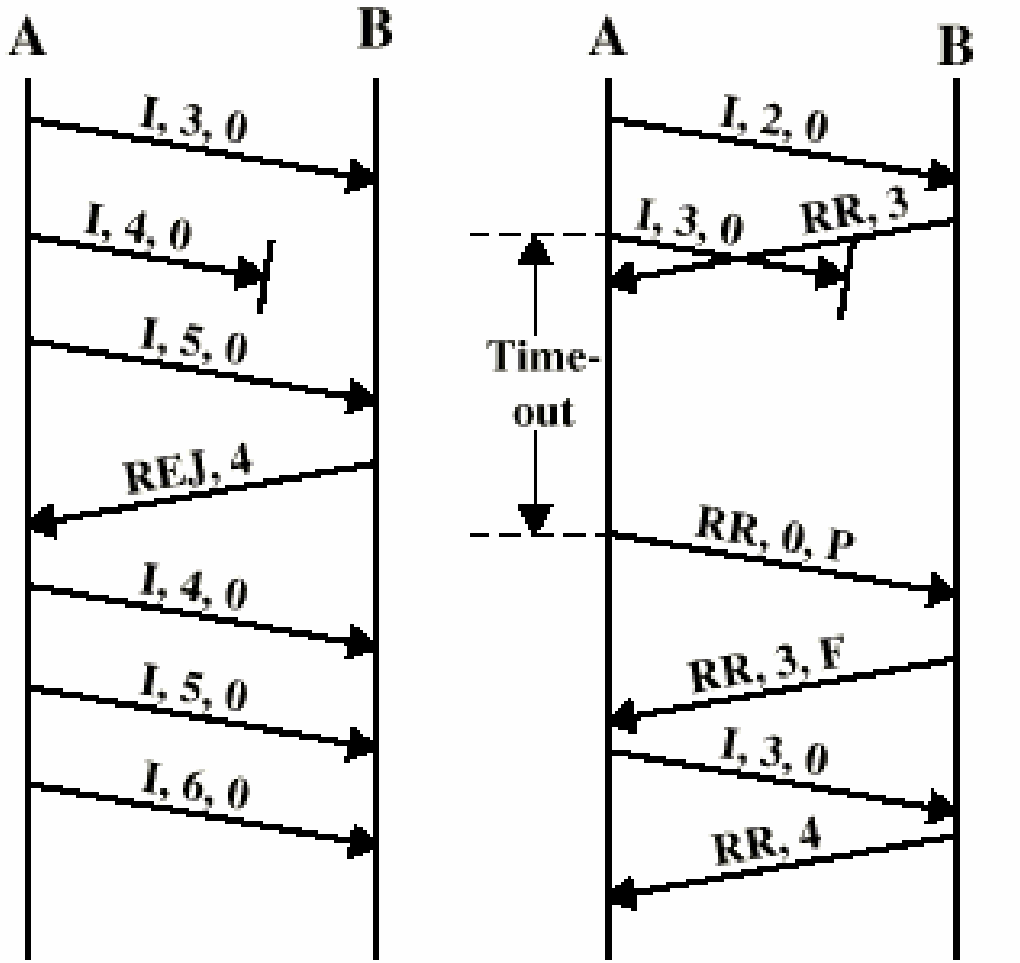
- Iniţiat de orice parte
- Din proprie iniţiativă, în caz de erori sau la cererea nivelului superior
- Cadru de tip disconect (DISC)
- Acceptarea deconectării se răspunde cu UA

Exemplu (1)



(a) Link setup and disconnect (b) Two-way data exchange (c) Busy condition

Exemplu (2)



(d) Reject recovery

(e) Timeout recovery