
Transmisii de date digitale

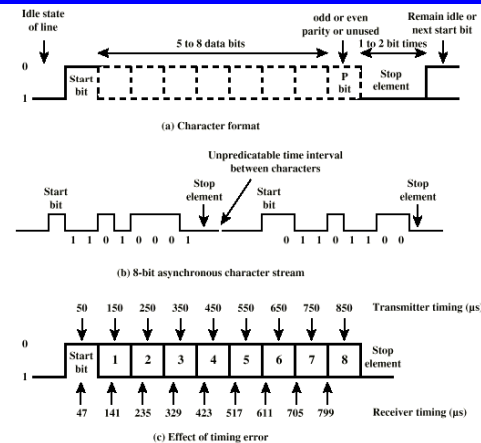
Transmisia sincronă și asincronă

- Problemele de timing necesită un mecanism pentru sincronizare între transmițător și receptor
- 2 soluții:
 - Asincron
 - Sincron

Asincron

- Datele sunt transmise un caracter după caracter
 - 5 la 8 biți
- Sincronizarea e necesar de menținut doar în cadrul fiecărui caracter
- Resincronizare la fiecare caracter

Asincron (diagramă)



Asincron (comportament)

- In cazul transmisiei continui, intervalul între caractere este uniform (lățimea elementului de stop)
- In starea de așteptare (idle), receptorul caută tranziția din 1 în 0
- Apoi eșantionează următoarele 7 intervale (lungimea caracterului)
- Apoi așteaptă următoarea tranziție din 1 în 0 pentru caracterul următor

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

Mihai IONITA
curs

Sincron - Bit Level

- Blocurile de date transmise fără biți de start sau stop
- Ceasurile trebuie să fie sincronizate
- Se pot utiliza linii de ceas dedicate
 - Pentru distanțe scurte
 - Afectate de imperfecțiunile mediului
- Ceas inclus în semnalul de date
 - Codare Manchester
 - Frecvență purtătoare (analog)

Mihai IONITA
curs

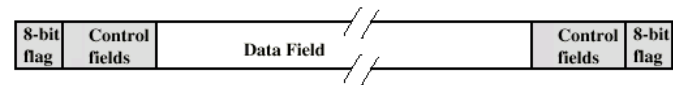
Sincron - Block Level

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

- Mai eficient (overhead scăzut) decât asincron

Mihai IONITA
curs

Sincron (diagramă)

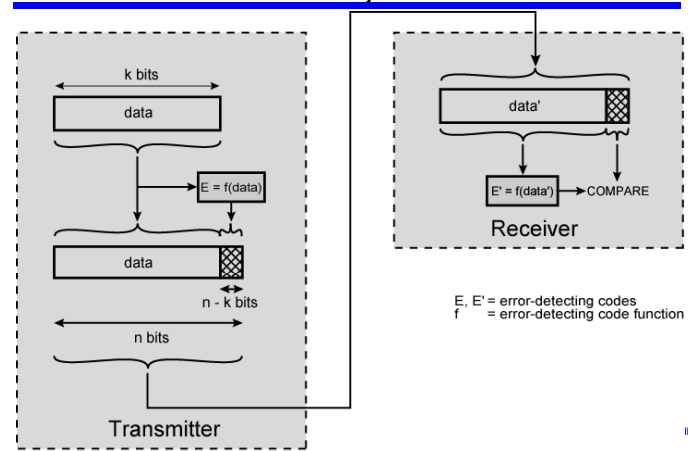


Mihai IONITA
curs

Tipuri de erori

- O eroare apare când un bit este modificat între transmisie și recepție
- Erori de 1 bit
 - 1 bit modificat
 - Biții adiacenți nu sunt afectați
 - Zgomot alb
- Erori în burst
 - Lungime B
 - Secvență continuă de B biți în care primul, ultimul și orice număr de biți intermediari sunt afectați
 - Zgomot în impuls
 - Efect mai mare la viteză mai mare

Metode de detecție a erorilor



Detecția erorilor

- Biți suplimentari sunt adăugați de transmițător pentru codul de detecție a erorilor
- Paritate
 - Valoarea bitului de paritate este astfel încât caracterul are un număr par (paritate pară) sau impar (paritate impară) de unu
 - Număr par de biți eronați nu sunt detectați

Cyclic Redundancy Check

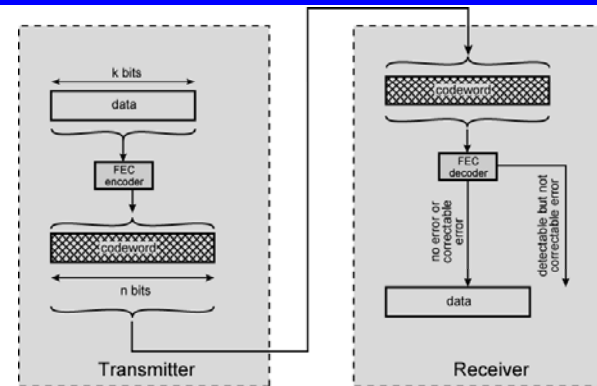
- Pentru un număr de k biți transmițătorul generează o secvență de n biți
- Transmite $k+n$ biți care se divide exact cu un anumit număr
- Receptorul divide secvența recepționată cu acest număr:
 - Dacă nu există rest, se presupune că nu sunt erori

Corectarea erorilor

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

Mihai IONITA
curs

Diagrama metodei de corectare a erorilor



Mihai IONITA
curs

Corectarea erorilor

- Fiecare bloc de k biți este mapat într-un bloc de n biți ($n > k$)
 - Cuvânt de cod
 - Forward error correction (FEC) encoder
- Cuvântul de cod (codeword) este transmis
- Șirul de biți recepționați este similar cu cel transmis, dar poate conține erori
- Cuvântul de cod recepționat este procesat de FEC decoder
 - Dacă nu sunt erori, blocul de date original este recepționat
 - Unele configurații de erori pot fi detectate și corectate
 - Unele configurații de erori pot fi detectate dar nu corectate
 - Unele configurații de erori (rare) nu pot fi detectate
 - Rezultă în date eronate la ieșirea FEC

Mihai IONITA
curs

Modul de funcționare a corecției de erori

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

Mihai IONITA
curs

Legătura de date

Cerintele nivelului : Data Link

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

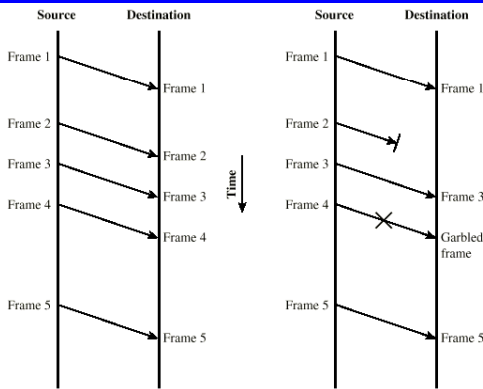
Mihai IONITA
curs

Controlul fluxului de date

- Asigură ca entitatea de transmisie nu depășește capacitatea entității de recepție
 - Previne umplerea buffer-ului de recepție
- Timpul de transmisie
 - Timpul necesar pentru a emite toții biții prin mediu
- Timpul de propagare
 - Timpul necesar unui bit pentru a traversa legătura

Mihai IONITA
curs

Model de transmisie pe cadre



(a) Error-free transmission

(b) Transmission with losses and errors

Mihai IONITA
curs

Stop and wait

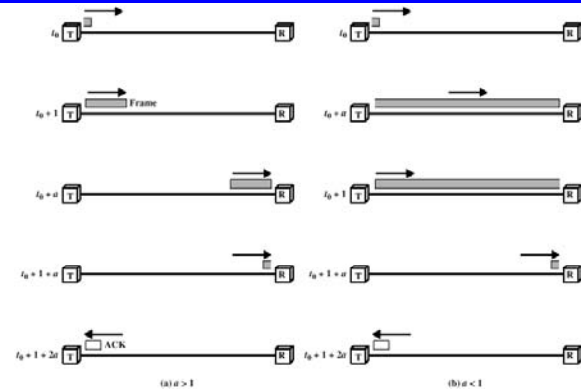
- Sursa trimite un cadru
- Destinație recepționează cadrul și răspunde cu confirmare
- Sursa așteaptă confirmarea înainte de a trimite cadrul următor
- Destinația poate opri fluxul de date dacă nu mai trimite confirmarea
- Merge bine pentru puține cadre mari

Mihai IONITA
curs

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 recepționat)
 - În caz de erori, este necesară retransmiterea unui bloc mic
 - Evită ocuparea mediului de către o stație pentru un timp îndelungat
- Protocolul Stop and wait devine inadecvat

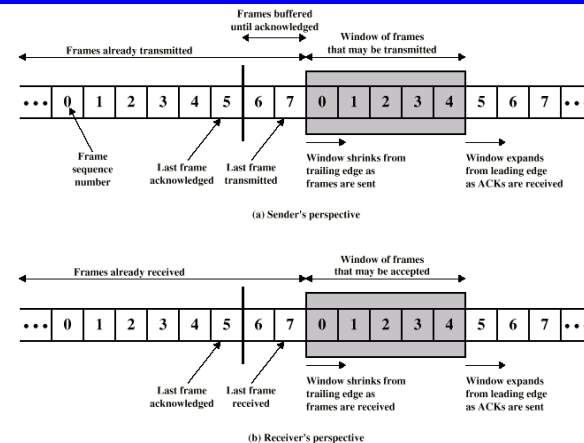
Utilizarea protocolului stop and wait



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
- Transmițătorul poate trimite până la W cadre fără ACK
- Fiecare cadru primește un număr
- ACK include numărului cadrului următor așteptat
- Numărul secvenței este limitat de dimensiunea câmpului (k)
 - Cadrele sunt numărate modulo 2^k

Diagrama sliding window



Go Back N – Cadre eronate

- Receptorul detectează eroare în cadrul j
- Receptorul trimite rejection- i
- Transmițătorul primește rejection- i
- Transmițătorul retransmite cadrul j și toate cadrele următoare

Go Back N – Cadre pierdute (1)

- Cadrul i se pierde
- Transmițătorul trimite $i+1$
- Receptorul primește cadrul $i+1$ în afara ordinii așteptate
- Receptorul trimite reject i
- Transmițătorul merge înapoi la cadrul i și retransmite

Go Back N – Cadre pierdute (2)

- Cadrul i se pierde și nu se trimite alt cadru
- Receptorul nu primește nimic și nu răspunde nici cu confirmare, nici cu rejecție
- Transmițătorul atinge time-out și trimite un cadru de confirmare
- Receptorul interpretează asta ca o comandă pe care o confirmă cu cadrul următor pe care îl așteaptă (cadrul i)
- Atunci transmițătorul retransmite cadrul i

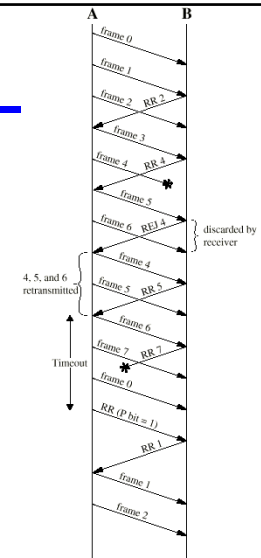
Go Back N – Confirmare eronată

- Receptorul primește cadrul i și trimite confirmare care se pierde
- Confirmările sunt cumulative și următoarea confirmare ($i+n$) poate sosi înainte ca transmițătorul să ajungă la time-out pentru cadrul i
- Dacă transmițătorul 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)

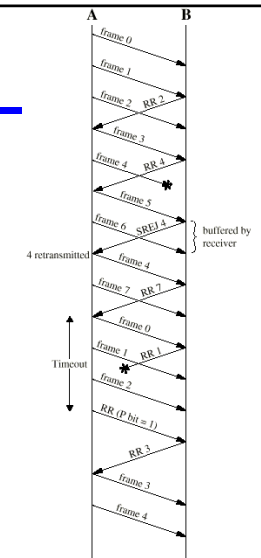
Go Back N Diagramă



Rejecție selectivă

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

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 stații HDLC

- Stație primară
 - Controlează modul de operare a legăturii
 - Transmite cadre numite comenzi
 - Menține legături logice separate cu fiecare stație secundară
- Stație secundară
 - Funcționează sub controlul stației primare
 - Transmite cadre numite răspunsuri
- Stație combinată
 - Transmite comenzi și răspunsuri

Configurații ale legăturii HDLC

- Neechilibrat
 - O stație primară și mai multe stații secundare
 - Poate fi full duplex sau half duplex
- Echilibrat
 - Două stații combinate
 - Poate fi full duplex sau half duplex

Moduri de transfer HDLC (1)

- Normal Response Mode (NRM)
 - Configurație neechilibrată
 - Stația primară inițiază transferul spre stația secundară
 - Stația secundară poate transmite date doar ca răspuns a unei comenzi de la stația principală

Moduri de transfer HDLC (2)

- Asynchronous Balanced Mode (ABM)
 - Configurație echilibrată
 - Orice stație poate iniția transmisia fără a cere permisiunea
 - Cel mai folosit
 - Nu e necesar polling

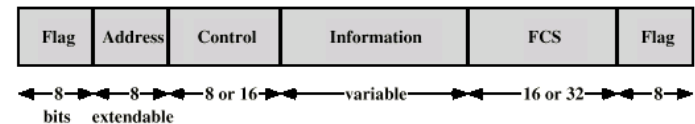
Moduri de transfer HDLC (3)

- Asynchronous Response Mode (ARM)
 - Configurație neechilibrată
 - Stația secundară poate iniția transmisia fără permisiunea stației primare
 - Stația 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 și control

Structura cadrului



(a) Frame format

Flag

- Delimitează cadrul
- 01111110
- Poate închide un cadru și deschide altul
- Receptorul urmărește secvența de flag pentru sincronizare
- Pentru a evita confuzia cu datele de tip 01111110, se folosește "bit stuffing"
 - 0 inserat după fiecare secvență de cinci 1
 - Dacă receptorul detectează cinci 1, verifică bitul următor
 - Dacă e 0, se șterge
 - Dacă e 1 și al șaptelea bit e 0, se acceptă un flag
 - Dacă bitul al șaselea și al șaptelea e 1, se semnalizează "abort"

Mihal IONITA
curs

Bit Stuffing

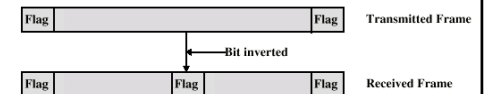
Original Pattern:

111111111111011111101111110

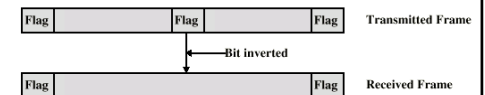
After bit-stuffing

11111011111011011111010111111010

(a) Example



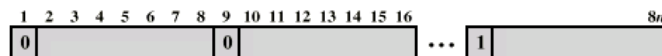
(b) An inverted bit splits a frame in two



(c) An inverted bit merges two frames

Adresa

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



(b) Extended Address Field

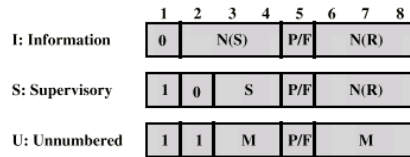
Mihal IONITA
curs

Control

- Diferiți în funcție de tipul cadrului
 - Information – datele de transmis către user (nivelul superior)
 - Controlul de flux și de erori folosind piggyback
 - Supervisory – ARQ când nu se folosește piggyback
 - Unnumbered – control suplimentar al legăturii
- Primul sau primii 2 biți determină tipul cadrului

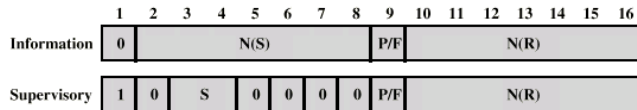
Mihal IONITA
curs

Control - diagramă



N(S) = Send sequence number
N(R) = Receive sequence number
S = Supervisory function bits
M = Unnumbered function bits
P/F = Poll/final bit

(c) 8-bit control field format



(d) 16-bit control field format

IONITA
curs

Poll/Final Bit

- Folosit în funcție 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

Mihai IONITA
curs

Informația

- Numai în cadre de tip information și unele cadre de tip unnumbered
- Trebuie să conțină un număr întreg de octeți
- Lungime variabilă

Mihai IONITA
curs

Frame Check Sequence

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

Mihai IONITA
curs

Modul de operare HDLC

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

Inițializarea

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

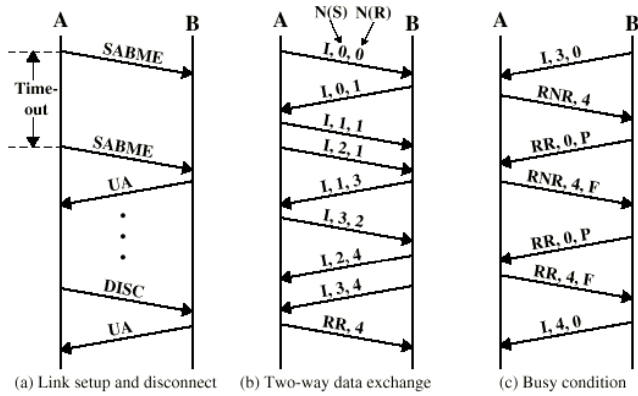
Transferul de date

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

Deconectarea

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

Exemplu (1)



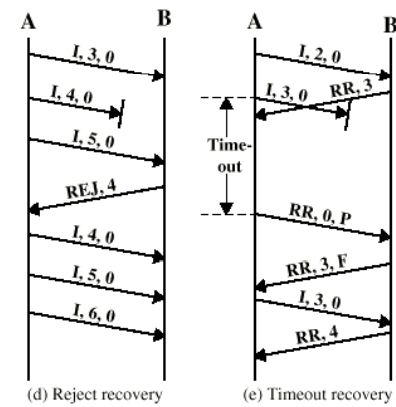
(a) Link setup and disconnect

(b) Two-way data exchange

(c) Busy condition

Mihai IONITA
curs

Exemplu (2)



(d) Reject recovery

(e) Timeout recovery

Mihai IONITA
curs