# Explicati cum folosesc protocoalele sumele de control si confirmarile si numerele de secventa.

Protocoalele folosesc **sumele de control** si **confirmarile** pentru a realiza una dintre functiile nivelului legatura de date si anume controlul erorilor. Pentru aceasta emitatorul calculeaza suma de control a cadrului (folosind de exemplu CRC), o ataseaza cadrului in antetul acestuia, trimite cadrul si porneste un contror (pentru a nu astepta la infinit confirmarea acestuia). Atunci cand receptorul primeste cadrul, calculeaza si el suma de control a cadrului si o compara cu suma de control atasata cadrului. In functie de protocol, receptorul poate corecta erorile, sau poate cere din nou cadrul receptionat gresit (in functie de protocol poate trimite o confirmare negativa sau nu trimite nici o confirmare). Daca dupa scurgerea timpului, aplicatia receptoare nu primeste nici o confirmare pentru cadrul trimis, aceasta il retrimite.

Atunci cand cadrele sunt transmise de mai multe ori, exista pericolul ca receptorul sa accepte acelasi cadru de mai multe ori si sa-l transmite de mai multe ori in retea. Pentru a evita aceasta situatie folosim **numere de secventa** pentru cadrele de iesire.

**Confirmarile** reprezinta o solutie pentru a preveni inundarea unui receptor lent de catre un emitator rapid. Presupunem ca avem un canal fara erori, canal pe care se transmit pachete ce contin anumite date. Transmitatorul trimite un pachete, receptorul il primeste si trimite un cadru de confirmare. Transmitatorul primeste cadrul de confirmare si doar atunci poate trimite urmatorul pachet.

# Coduri corectoare vs Coduri detectoare de erori.

Corectarea erorilor este mai complexă decât detectarea lor.

Pe canale cu siguranta mare, cum ar fi fibra optica, este mai eficient sa utilizam un cod detector de erori sis a retransmitem blocul in care s-au detectat erori.

In cazul canalelor de comunicatie fara fir, este indicat sa adaugam destula informative redundanta fiecarui bloc, in care sa ne bazam pe retransmisie, care poate sa fie la randul sau afecatata de erori.

Codurile corectoare de erori sunt adesea utilizate pe canale fără fir, care sunt cunoscute ca fiind predispuse la erori, în comparaţie cu firele de cupru sau fibra optică. Fără coduri detectoare de erori, comunicaţia ar fi greu de realizat. In cazul firelor de cupru sau a fibrei optice rata erorilor este mult mai mică, aşa că detectarea erorilor şi retransmisia este de obicei mai eficientă aici ca metodă de tratare a erorilor care apar ocazional.

Detectia erorilor utilizează sindromul, o combinatie liniară a simbolurilor cuvântului de cod, pe cand corectarea erorilor necesită aflarea zerourilor unor polinoame cu coeficienti functii rationale de componentele sindromului.

Numarul de pozitii binare prin care 2 cuvinte de cod difera se numeste distanta Hamming. Semnificaţia sa este că dacă două cuvinte de cod sunt despărţite de o distanţă Hamming d, sunt necesare d erori de un singur bit pentru a-l converti pe unul în celălalt. Pentru a detecta d erori, este nevoie de un cod cu distanta d+1. Similar pentru a corecta d erori este nevoie de un cod cu distanta 2d+1.

Corectia erorilor foloseste si metoda codului polinomial.

Codurile polinomiale sunt bazate pe tratarea şirurilor de biţi ca reprezentări de polinoame cu coeficienţi 0 şi 1. Un cadru de k biţi este văzut ca o listă de coeficienţi pentru un polinom cu k termeni, de la xk-1 la x0. Se spune că un astfel de polinom este de gradul k-1. Bitul cel mai semnificativ (cel mai din stânga) este coeficientul lui xk-1; următorul bit este coeficientul lui xk-2 ş.a.m.d

# CRC “CB” cu polinomul x4+x+1

C(67): 01000011

B(66): 01000010

Polinomul: 10011

CB: 01000011 01000010

→ 01000011 01000010 0000 → restul: 0100 → 01000011 01000010 0100

# Enumerati functiile nivelului legatura de date

Functiile nivelului legatura de date sunt:

* 1. incadrarea
  2. transmiterea trasnparenta
  3. controlul erorilor
  4. controlul fluxurilor
  5. gestiunea legaturii.

# Incadrare

Incadrarea este o functie a nivelului de legatura date. Acesta se ocupa de ea adaugand caractere de control pentru delimitarea cadrului.

Metode folosite pentru realizarea incadrarii:

* 1. Numararea caracterelor: utilizează un câmp din antet pentru a specifica numărul de caractere din cadru. Atunci când nivelul legătură de date de la destinaţie primeşte contorul de caractere, ştie câte caractere urmează şi unde este sfârşitul cadrului. Problema cu acest algoritm este că valoarea contorului poate fi alterată de erori de transmisie. Chiar dacă suma de control este incorectă şi destinaţia ştie că a primit cadru eronat, nu există nici o posibilitate de a determina unde începe următorul cadru. Nu ajută nici trimiterea unui cadru înapoi la sursă, cerând o retransmisie, deoarece destinaţia nu ştie peste câte caractere să sară pentru a începe retransmisia. Din acest motiv, metoda contorizării caracterelor este rar utilizată.
  2. Caractere de control: înlătură problema resincronizării după o eroare, prin aceea că fiecare cadru începe şi se termină cu o secvenţa specială de octeţi. Octetul care indica inceputul si sfarsitul cadrului sunt identici. O problemă serioasă cu această metodă apare atunci când se transmit date binare, cum ar fi un obiect sau numere în virgulă mobile. O cale de rezolvare a acestei probleme este ca nivelul legătură de date al emiţătorului să insereze un octet special (ESC) înaintea fiecărei apariţii „accidentale” a indicatorului în date. Nivelul legătură de date al receptorului va elimina acest octet special înainte de a pasa datele nivelului reţea.

# Descrieti principiul protocoalelor cu fereastra glisanta

Esenta protocoalelor cu fereastra glisanta este aceea ca, la orice omement de timp emitatorul mentine o multime de numere de secventa care corespund cadrelor pe care are permisiunea sa le trimita (fereastra de transmisie). Similar, receptorul mentine o fereastra de receptie corespunzatoare multimii de cadre care pot fi acceptate.

Cand soseste un cadru cu date, in locul emiterii imediate a unui cadru de confirmare, receptorul sta si asteapta urmatorul pachet. Confirmarea e transportata pe gratis de catre urmatorul cadru. Tehnica intarzierii confirmarii,astfel incat sa fie agatata de urmatorul cadru de date,se numeste piggybacking.

Avantaj, la acest tip de protocol, este faptul ca lungimea de banda este folosita mai eficient.

Dezavantajul este ca nu stim cat timp trebuie sa astepte nivelul legatura, pachetul pe care sa ataseze confirmarea. (Solutia dezavantajului este asteptarea pentru un numar fixat de milisecunde. Daca un pachet soseste mai repede, confirmarea este adaugata in el. Daca pana la sfarsitul perioadei de timp nu a aparut un nou pachet, se trimite un cadru de confirmare).

# Diferenta intre repetare selectiva si neselectiva

In cazul ferestrei neselective, daca soseste alt cadru decat cel asteptat, protocolul elimina cadrele nesosite in ordine. In cazul ferestrei selective, chiar daca un cadru nu soseste in ordine, protocolul memoreaza acest cadru pana cand devine util.

# Protocol start-stop

* dupa transmiterea unui cadru, emitatorul declanseaza contorul de timp.
* pentru ca receptorul sa distinga intre un cadru pe care il vede pentru prima data si o retransmisie, emitatorul trebuie sa adauge numere de secventa in antetul fiecarui cadru pe care il transmite

# Pseudocod pentru protocol pentru canal cu erori

void sender3(void) {

seq\_nr xt\_frame\_to\_send; /\* numărul de secvenţă al următorului cadru trimis \*/

frame s; /\* variabilă temporară \*/

packet buffer; /\* tampon pentru pachetul transmis \*/

event\_type event;

next\_frame\_to\_send = 0; /\* iniţializează numerele de secvenţă de ieşire \*/

from\_network\_layer(&buffer); /\* preia primul pachet \*/

while (true) {

s.info = buffer; /\* construieşte un cadru pentru transmitere \*/

s.seq = next\_frame\_to\_send; /\* inserează în cadru un număr de secvenţă \*/

to\_physical\_layer(&s); /\* îl trimite pe traseu \*/

start\_timer(s.seq); /\* dacă răspunsul întârzie prea mult, timpul va expira \*/

wait\_for\_event(&event); /\* frame\_arrival, cksum\_err, timeout \*/

if (event == frame\_arrival) { /\* preia confirmarea \*/

from\_physical\_layer(&s);

if(s.ack==next\_frame\_to\_send) {

stop\_timer(s.ack); /\* opreşte ceasul \*/

from\_network\_layer(&buffer);

/\* preia următorul pachet de transmis \*/

inc(next\_frame\_to\_send);

/\* inversează next\_frame\_to\_send \*/

}

}

}

}

void receiver3(void) {

seq\_nr frame\_expected;

frame r, s;

event\_type event;

frame\_expected = 0;

while (true) {

wait\_for\_event(&event); /\* variante: frame\_arrival, cksum\_err \*/

if (event == frame\_arrival) { /\* a sosit un cadru corect \*/

from\_physical\_layer((r); /\* preia cadrul nou sosit \*/

if (r.seq == frame\_expected) { /\* este cel pe care îl aşteptam \*/

to\_network\_layer((r.info); /\* transferă datele nivelului reţea \*/

inc(frame\_expected); /\* incrementează număr de secvenţă \*/

}

s.ack=1-frame\_expected; /\* spune care frame este confirmat \*/

to\_physical\_layer(&s); /\* nici unul dintre câmpuri nu este utilizat \*/

}

}

}

# Explicati cum functioneaza transmiterea transparenta

In zona de date pot aparea caractere de control (datorita incadrarii). STX se pune la inceput, marcand inceputul mesajului, si ETX la sfarsit. Pot aparea 2 probleme:

* 1. La trimiterea datelor (binare), datele pot contine secvente identice cu secventele speciale.

Rezolvare: In loc de STX si ETX vom pune. DLE STX si DLE ETX, si le definim la inceputul si sfarsitul.

* 1. Unele date contin flag fals de terminare.

Rezolvare: La trimitere, punem un 0 dupa 5 biti consecutivi de 1. La primire, eliminam 0 care e dupa 5 biti consecutivi de 1.

# Avantaje si dezavantaje ale protocolului start-stop

Avantajul protocolului start-stop este faptul ca e usor de implementat, nu necesita un hardware performant sau conditii speciale.

Dezavantajul este utilizarea largimii de banda care este cu mult sub optim.

# Go Back N

# Permite emitatorului sa trimita pana la W cadre, in loc de 1 (maxim 26)

# Cu o alegere potrivita a lui W, emitatorul va putea trimite pana la W cadre pentru un timp egal cu timpul de transit, fara a umple fereastra

# Pentru a putea opera la eficienta maxima, emitatorul trebuie sa fie capabil sa umple o fereastra fara sa se opreasca (BANDA DE ASAMBLARE)

# In cazul evenimetului de tip “sosire cadru” = emitatorul preia de la nivelul fizic cadrul sosit, cadrele sunt acceptate doar in ordine, preda pachetul nivelului de retea si creste limita inferioara a ferestrei emitatorului.

# In cazul evenimentului de tip “time-out” = emitatorul retransmite toate cadrele neconfirmate.

# Cum recunoaste receptorul ca s-a produs o eroare de transmisie? Ce face receptorul in cazul unei erori de transmisie?

# Dacă un cadru ajunge corect, receptorul poate transmite către nivelul reţea, în ordine, cadrele pe care le-a memorat. Repetarea selectivă este deseori combinată cu utilizarea confirmărilor negative (NAK), care sunt trimise atunci când se detectează o eroare, de exemplu când se primeşte un cadru cu suma de control incorectă sau cu număr de secvenţă necorespunzător. Confirmările negative simulează retransmisia înainte de expirarea contorului de timp corespunzător, îmbunătăţind astfel performanţa.