

# Nivelul legătură de date

---

## Contents

1.	Servicii furnizate de nivelul legătură de date .....	3
1.1.	Încadrarea .....	5
1.1.1.	Structura cadrului .....	6
1.2.	Accesul la legătură.....	6
1.3.	Livrare fiabilă .....	7
1.4.	Detecția erorilor.....	9
1.4.1.	Bitul de paritate .....	10
1.4.2.	Coduri cu redundanță ciclică.....	10
1.5.	Corecția erorilor.....	12
1.6.	Controlul fluxului .....	12
1.7.	Half-duplex/Full-duplex.....	12
2.	Unde este implementat nivelul legătură de date.....	12

Nivelul legătură de date este responsabil de transmiterea corectă a datelor printr-o legătură fizică existentă, între două dispozitive conectate direct prin acea legătură fizică.

Așa cum este indicat în Figura 1 calea de comunicație este compusă dintr-o succesiune de legături de comunicație, care încep la gazda sursă, traversează o serie de rutere, pentru ca în final să ajungă la gazda destinație.

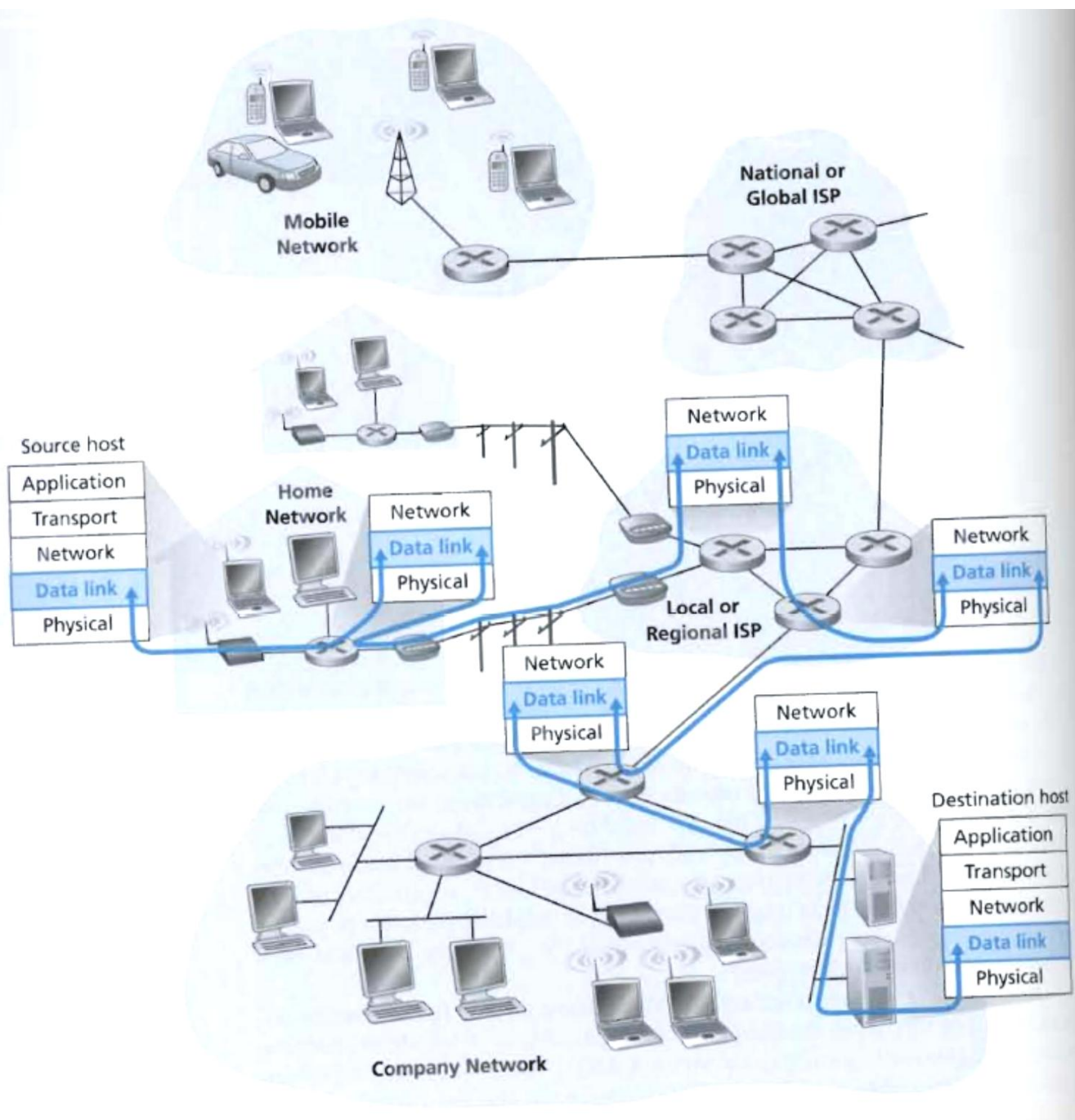


Figura 1

Pe măsură ce parcurgem de sus în jos stiva de protocoale, mai exact de la nivelul rețea la nivelul legătură de date ne punem următoarea o serie de întrebări:

- cum sunt trimise datagramele pe *legăturile individuale* ce compun calea de comunicație capăt la capăt?
- cum sunt încapsulate datagramele de nivel rețea în *cadre* de nivel legătură de date?
- poate nivelul legătură de date să asigure o livrarea fiabilă ruter-la-ruter?

- există posibilitatea utilizării de-a lungul căii de comunicație a unor protocoale de nivel legătură de date diferite?

La nivel legătură de date, din punct de vedere al tehnologiei folosite există două tipuri diferite de canale. Primul tip este constituit din canalele cu difuzare, utilizat cu precădere în cazul rețelelor locale (LAN), rețelor locale fără fir, rețelor prin satelit și rețelor de acces hibride fibră-coaxial (hybrid fiber-coax, HFC). În cazul canalelor cu difuzare, mai multe gazde sunt conectate la același canal de comunicații, iar pentru coordonarea transmisiilor și evitarea coliziunilor între cadre este necesar un protocol de acces la mediu. Al doilea tip este reprezentat de legăturile de comunicație punct-la-punct, cum ar fi cele dintre două rutere sau legătură de acces dintre un modem rezidențial și ruterul furnizorului de servicii. Coordonarea accesului pe o legătură punct la punct este simplă, însă totuși apar probleme ce nu pot fi neglijate legate de încadrare, transfer fiabil de date, detecția erorilor și controlul fluxului.

## 1. Servicii furnizate de nivelul legătură de date

Să demarăm prin clarificarea terminologie utilizate. La acest nivel vom referi gazdele și ruterele ca **noduri**, întrucât, așa cum vom vedea în curând, nu va fi de interes faptul că un nod este o gazdă sau un ruter. Vom referi canalele de comunicație dintre nodurile adiacente aflate de-a lungul căii de comunicație ca **legături**. Pentru ca o datagramă să poată fi transferată de la sursă la destinație, aceasta trebuie transferată pe fiecare din *legăturile individuale* din calea capăt-la-capăt. Pe o legătură dată, nodul emițător încapsulează datagrama într-un cadru de nivel legătură de date și îl transmite pe acea legătură; nodul receptor primește cadrul din care extrage datagrama.

Un protocol de nivel legătură de date este utilizat pentru transferul unei datagrame încapsulate într-un cadru peste o legătură individuală. **Protocolul de nivel legătură de date** definește formatul pachetelor schimbate între nodurile aflate la capătul legăturii, precum și acțiunile întreprinse de aceste noduri la recepția și/sau transmisia pachetelor. Reamintim că unitățile de date vehiculate de protocoalele de nivel legătură de date sunt denumite **cadre**, iar de regulă fiecare cadru încapsulează o datagramă (există totuși și cadre care nu încapsulează datagrame). Acțiunile întreprinse de un protocol de nivel legătură de date la recepția și transmisia cadrelor includ, fără a se limita la acestea: detecția erorilor, retransmisie, controlul fluxului și acces aleatoriu. Constituie exemple de protocoale de nivel legătură de date: Ethernet-ul, IEEE 802.11 (LAN fără fir, cunoscut și sub numele WiFi/WLAN), token ring, PPP. Unele dintre aceste protocoale vor fi analizate în detaliu mai târziu.

Dacă nivelul rețea are sarcina transferului segmentelor de nivel transport de la gazda sursă la gazda destinație, un protocol de nivel legătură de date are un rol mai modest, și anume transferul

datagramelor peste o singură legătură din cale. O caracteristică importantă a nivelului legătură de date este că datagramele pot fi transferate de protocoale diferite pe diversele legături ce compun calea. Spre exemplu, o datagramă poate fi transferată prin Ethernet pe prima legătură, prin PPP pe a ultima legătură și printr-un protocol WAN de nivel legătură pe una din legăturile intermediare. Este important de remarcat că serviciile furnizate de diversele protocoale de nivel legătură de date de pe calea sursă destinație pot fi diferite. Spre exemplu, unele protocoale de nivel legătură asigură transfer fiabil pe când altele nu. Astfel nivelul rețea trebuie să fie capabil să-și îndeplinească funcțiile gazdă-la-gazdă inclusiv în prezența unui set de servicii eterogene pe legăturile individuale.

Cu toate că serviciul de bază al oricărui nivel legătură de date este transferul datagramelor peste o singură legătură între două noduri adiacente, detaliile serviciilor furnizate pot varia semnificativ între protocoalele de nivel legătură de date.

La nivelul rețea al nodului sursă există un proces, care trimite datagrame către nivelul legătură de date, pentru a fi transmise la destinație. Funcția nivel legătură de date este să transmită datagramele la destinație pentru ca acolo să fie livrate nivelului rețea.

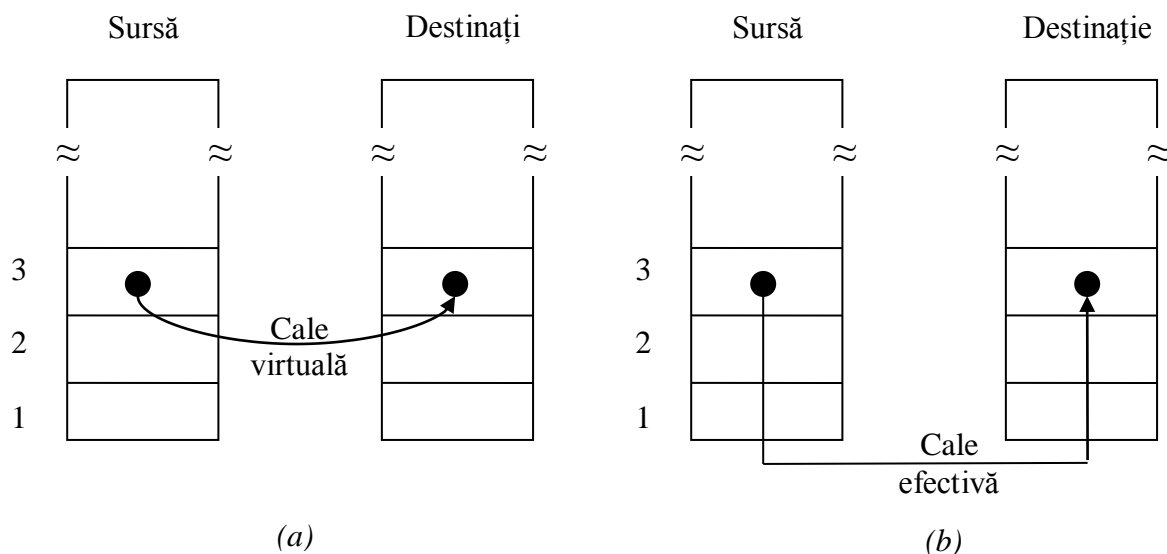


Figura 2

Transmisia efectivă urmează calea din figura (b), însă este mai ușor de înțeles și analizat în termenii a două procese ale nivelului legătură de date care comunică direct utilizând un protocol al legăturii de date. În continuare vom analiza posibile servicii ce pot fi oferite de protocoalele de nivel legătură de date.

## 1.1. Încadrarea

Înainte de transmisia efectivă, aproape toate protocoalele de nivel legătură de date încapsulează fiecare datagramă de nivel rețea într-un cadru de nivel legătură de date. Un cadru constă dintr-un câmp de date (informația utilă), unde este inserată datagrama, precum și o serie de câmpuri antet. (Un cadru poate conține și câmpuri încheiere; vom referi atât câmpurile antet cât și cele încheiere ca antete). Structura cadrelor depinde de protocolul utilizat; cu toate acestea, în scopuri didactice, este posibilă extragerea unui format general al cadrelor.

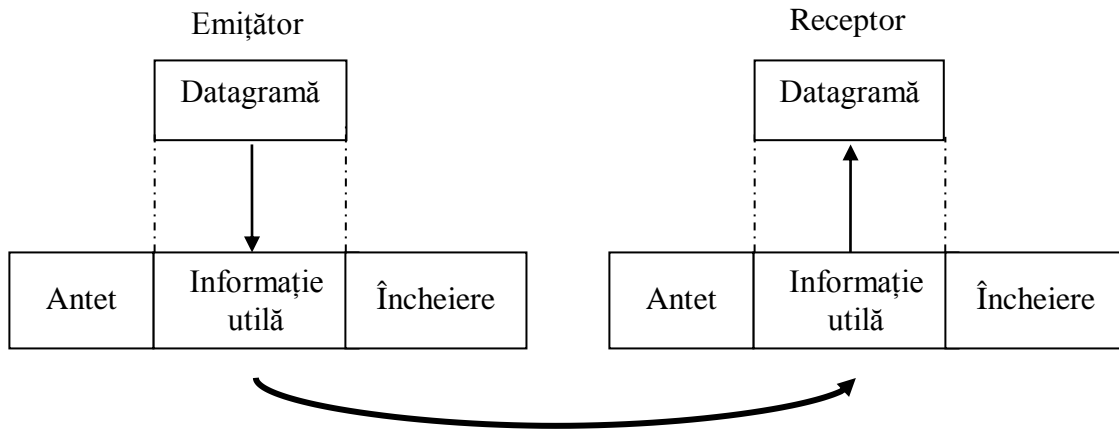


Figura 3

În vederea furnizării unui serviciu nivelului rețea, nivelul legătură de date trebuie să utilizeze serviciul furnizat de către nivelul fizic. Sarcina nivelului fizic este să primească un flux de biți și să încerce să-l trimită la destinație; nu se garantează că acest flux de biți nu conține erori. Numărul de biți recepționați poate fi mai mic, egal sau mai mare decât numărul de biți transmiși și pot avea valori diferite. Este la latitudinea nivelului legătură de date să detecteze și dacă e necesar să corecteze erorile. Abordarea uzuală este să se spargă șirul de biți în cadre discrete.

La acest nivel, fiecărei datagrame primite de la nivelul rețea îi sunt adăugate un antet și o încheiere specifice nivelului legătură de date. În antet sunt înscrise pe lângă alte date, adresele de nivel doi sursă și destinație, fără de care rețeaua nu ar putea funcționa, pentru că stațiile nu ar putea decide dacă un cadru recepționat le este adresat sau nu. De asemenea în încheiere este înscrisă informația de control care asigură detecția erorilor. Împărțirea în cadre permite obținerea de informații ce nu se puteau transmite prin șiruri de biți, cum ar fi de exemplu:

- care calculatoare comunică între ele;
- când începe și când se termină comunicarea între anumite calculatoare individuale;
- contorizarea erorilor care au apărut.

### 1.1.1. Structura cadrului

O observație importantă este faptul că formatul cadrului este dependent de tehnologie (Ethernet, WLAN), adică pentru diferite tipuri de rețele vom avea formate de cadre diferite. Vom prezenta în continuare o structură generică a unui cadru de nivel doi, ce conține anumite câmpuri compuse din unul sau mai mulți octeți.

Început de cadru	Adrese	Lungime/ Tip	Date	FCS
------------------	--------	-----------------	------	-----

Figura 4 Structura generică a unui cadru de nivel legătură de date

Aceste câmpuri sunt:

- *Început de cadru* este folosit pentru sincronizare, anunțând începerea unui cadru. Conține o secvență de semnalizare specifică fiecărei tehnologii în parte.
- *Adrese*. Adresarea este esențială pentru a ști cui îi este destinat acel cadru și de la cine provine. Fiecare protocol are propriul lui tip de adresare; de exemplu în cazul Ethernet-ului sau Wi-Fi adresarea se face prin intermediul adreselor IEEE pe 48 biți.
- *Lungime/Tip*. Anumite tipuri de cadre conțin și lungimea cadrului, iar altele conțin un câmp special, numit „Protocol field” sau „Type field” prin intermediul căruia se specifică protocolul de nivel superior căruia i se adresează datagrama încapsulată.
- *FCS* (Frame Check Sequence, rom. secvența de verificare a cadrului). În timpul unei transmisii datele sunt susceptibile la eroare, motiv pentru care în structura unui cadru de nivel doi este prevăzut un câmp care încearcă să determine pe cât posibil dacă a apărut sau nu o eroare. Acest câmp funcționează astfel: în momentul în care stația sursă încapsulează cadrul calculează FCS (există mai multe variante în funcție de tehnologie) și atașează rezultatul la cadru. Stația care recepționează cadrul calculează și ea la rândul ei FCS și îl compară cu cel citit de la sfârșitul cadrului primit. Dacă nu coincid înseamnă că a apărut o eroare.

### 1.2. Accesul la legătură

Un protocol de acces la mediu (medium acces protocol, MAC) specifică regulile prin care un cadru este trimis pe legătură. În cazul legăturilor punct la punct care dispun de un singur emițător și un singur receptor la celălalt capăt, protocolul MAC este simplu (sau inexistent) – emițătorul poate trimite un nou cadru imediat ce legătura devine inactivă. Cazul mai interesant este atunci când mai multe noduri

partajează o singură legătură cu difuzare – așa numita problemă a accesului multiplu. Aici protocolul MAC servește la coordonarea transmisiilor nodurilor; vom studia protocoale MAC în unitatea de curs următoare.

### 1.3. Livrare fiabilă

Atunci când un protocol de nivel legătură de date asigură un serviciu de livrare fiabilă, acesta garantează că va transfera datagrama de nivel rețea fără erori peste legătură. Anumite protocoale de nivel transport (cum ar TCP) pun de asemenea la dispoziție un serviciu fiabil de transfer. În mod similar serviciului de transfer fiabil de la nivelul transport, este posibilă implementarea unui serviciu de transfer fiabil la nivel legătură de date folosind confirmări și retransmisii. Serviciile de transfer fiabil sunt utilizate adesea în cazul legăturilor predispuse la rate mari de eroare cum ar fi legăturile radio, cu scopul de a corecta erorile locale (pe legătura unde a apărut eroarea), fără a forța o retransmisie capăt-la-capăt de către nivelul transport sau nivelul aplicație. În schimb, în cazul unor legături cu rate de eroare mici cum ar fi fibre optice, cabluri coaxiale sau torsadate, mecanismele de livrare fiabilă constituie o supraîncărcare inutilă. Din acest motiv, multe din protocoalele de nivel legătură de date folosite pe cabluri nu asigură un serviciu de livrare fiabil.

În ceea ce privește fiabilitatea transmisiilor la nivelul legătură de date există trei posibilități de bază, oferite în mod curent:

- Serviciul neconfirmat fără conexiune
- Serviciul confirmat fără conexiune
- Serviciul confirmat orientat conexiune.

*Serviciul neconfirmat fără conexiune* constă din aceea că stația sursă trimite cadre independente către destinație, fără ca nodul destinație să trebuiască să confirme primirea lor. În acest caz, nu sunt necesare stabilirea sau desființarea unei legături logice. Dacă un cadru este pierdut datorită zgomotului de pe linie, la nivel legătură de date nu se face nici o încercare de recuperare. Această clasă de servicii este adecvată atunci când rata de eroare este foarte scăzută, însă recuperarea este lăsată în sarcina nivelurilor superioare. De asemenea este adecvată pentru traficul în timp real, cum ar fi cel de voce, unde a primi date cu întârziere este mai rău decât a primi date eronate. Majoritatea LAN-urilor cablate utilizează la nivel legătură de date serviciu neconfirmat fără conexiune.

*Serviciul confirmat fără conexiune* constă din aceea că, încă nu se utilizează conexiuni, dar fiecare cadru transmis este confirmat individual. În acest mod, emițătorul știe dacă un cadru a ajuns sau nu cu bine. Dacă nu a ajuns într-un interval de timp specificat, poate fi trimis din nou. Acest serviciu este adecvat pentru canale nesigure, cum ar fi rețelele fără fir.

Trebuie menționat faptul că asigurarea confirmării la nivel legătură de date este doar o optimizare, niciodată o cerință. Nivelul rețea poate întotdeauna să trimită o datagramă și să aștepte confirmarea; dacă aceasta nu apare în timp util, atunci emițătorul poate retransmite datagrama. Problema cu această strategie este că de obicei, cadrele au o lungime maximă impusă de hardware, iar datagrama nivelului rețea nu are (cel puțin teoretic) o astfel de limitare. Dacă o datagramă este divizată în 10 cadre și 20% din totalul cadrelor sunt pierdute, transmiterea corectă a întregii datagrame ar lua foarte mult timp. În cazul în care cadrele individuale sunt confirmate și retransmise, datagramele întregi vor fi transmise mult mai rapid. Pe canale sigure, precum fibra optică, costul suplimentar implicat de un astfel de protocol al legăturii de date poate fi nejustificat, în schimb pe canale fără fir, costul este pe deplin justificat datorită nesiguranței acestora.

*Serviciul confirmat orientat-conexiune* este cel mai complex serviciu pe care nivelul legătură de date îl pune la dispoziția nivelului rețea. În cazul acestui serviciu, nodurile sursă și destinație stabilesc o conexiune înainte de a transfera date. Fiecare cadru transmis pe conexiune este numerotat iar nivelul legătură de date garantează că fiecare cadru este recepționat exact o dată și toate cadrele sunt recepționate în ordinea corectă. În schimb, în cazul serviciului fără conexiune, este posibil ca datorită unei confirmări pierdute, un cadru să fie transmis de mai multe ori și prin urmare recepționat de mai multe ori. Se spune că serviciul orientat conexiune furnizează proceselor de la nivelul rețea un flux de biți sigur. Atunci când este utilizat serviciul orientat conexiune, transferurile au trei faze distincte. În prima fază este stabilită conexiunea, ambele părți inițializând variabile și contoare, utilizate pentru a ține evidența cadrelor care au fost recepționate și a celor care nu au fost. În a doua fază sunt transmise unul sau mai multe cadre. În cea de a treia și ultima fază, conexiunea este desființată, eliberând variabilele și alte resurse utilizate.

Când un cadru ajunge la ruter, hardware-ul verifică absența erorilor (vom vedea mai târziu cum) și trimite cadrul programelor nivelului legătură de date. Programele nivelului legătură de date verifică dacă acesta este cadrul așteptat și dacă este așa trimit mai departe nivelului rețea câmpul informație utilă. La nivelul rețea se alege linia de ieșire corespunzătoare și se trimite datagrama înapoi, către programele nivelului legătură de date corespunzătoare liniei respective. Programele nivelului rețea doresc de obicei ca operația să fie executată corect, ceea ce presupune conexiuni sigure pe fiecare linie. Este sarcina protocolului legătură de date să facă liniile de comunicație nesigure să pară perfecte, sau cel puțin, suficient de bune.



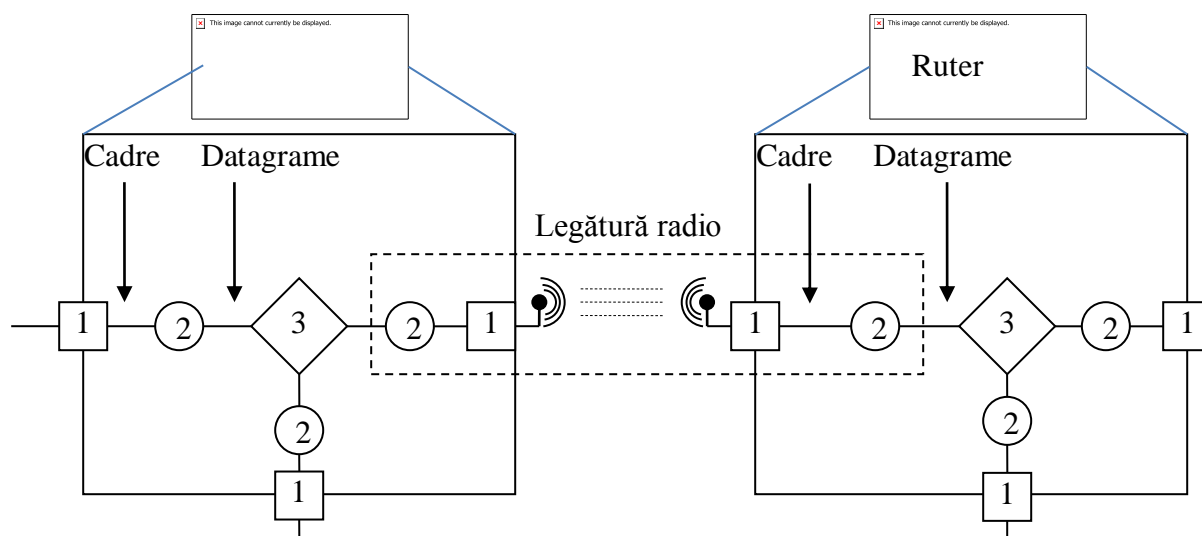


Figura 5

## 1.4. Detecția erorilor

Hardware-ul nivelului fizic de la nodul receptor poate decide incorect dacă un bit recepționat este zero sau unu. Astfel de erori apar datorită atenuării semnalului sau zgomotelor electromagnetice. Întrucât retransmiterea unei datagrame afectate de erori este inutilă, multe din protocoalele de nivel legătură de date pun la dispoziție mecanisme de detecție a unor astfel de erori de bit. Acest lucru este posibil prin includerea în cadru de către nodul emițător a unor biți de detecție a erorilor, fiind necesar ca nodul receptor să efectueze verificări. Spre deosebire de mecanismele limitate de detecție a erorilor prezente la nivel transport (suma de control Internet), nivelul legătură de date utilizează algoritmi complecși de regulă implementați în hardware.

Ca rezultat al proceselor fizice care le generează, erorile din unele medii de transmisie tind să vină mai curând în rafală decât izolate. Sosirea erorilor în rafală are atât avantaje cât și dezavantaje în raport cu erorile izolate de un singur bit. Avantajul este acela că datele de la un calculator sunt trimise întotdeauna în blocuri de biți. Spre exemplu să presupunem că dimensiunea unui bloc este de 1000 de biți iar rata medie de eroare este de  $10^{-3}$  ( $BER=10^{-3}$ ). Dacă erorile sunt fi independente, rata de eroare la nivel de pachet poate fi calculate astfel:  $PER=1-(1-BER)^L$  unde  $L$  este lungimea pachetului exprimată în biți; la înlocuirea în formulă se obține  $PER \approx 1$  ceea ce înseamnă că aproape fiecare pachet va fi eronat. Dacă în schimb erorile vin în rafale de câte 100 de biți, în medie vor fi afectate doar una sau două pachete din 100. Dezavantajul erorilor în rafală este că sunt mai greu de detectat și de corectat decât erorile izolate.

### 1.4.1. Bitul de paritate

Bitul de paritate este un bit care se adaugă pentru a ne asigura că numărul total de biți transmiși având valoarea 1 este întotdeauna par (sau impar).

7 biți de date	Sir transmis (paritate pară)	Sir transmis (paritate impară)
0000000	0000000 0	0000000 1
1010001	1010001 1	1010001 0
1111111	1111111 1	1111111 0

Dacă un număr impar de biți sunt modificați (incluzând bitul de paritate) atunci bitul de paritate va fi incorect indicând o eroare. Aceasta înseamnă că datele respective vor fi pierdute, fiind necesară o retransmisie. Dacă unui bloc i se adaugă un singur bit de paritate și blocul este deformat de o eroare în rafală (lungă), probabilitatea ca eroarea să fie detectată este de numai 50%, valoarea insuficientă.

Pentru creșterea probabilității de detectare a erorilor se pot utiliza mai mulți biți de paritate ca în figura de mai jos. Dacă blocul este puternic deformat de o rafală lungă sau rafale scurte multiple, probabilitatea ca oricare din cele  $n$  coloane să aibă accidental paritate corectă este de 0,5 deci probabilitatea ca un bloc eronat să fie acceptat ca fiind corect  $n$ -ar trebuie să depășească  $(0,5)^n$ .

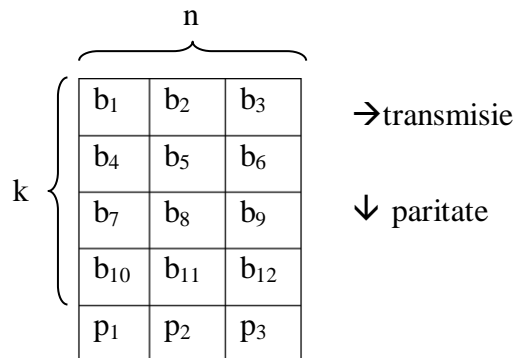


Figura 1.1 Extinderea schemei de paritate

### 1.4.2. Coduri cu redundanță ciclică

Codurile cu redundanță ciclică (eng. Cyclic Redundancy Code, CRC), cunoscute și sub denumirea de coduri polinomiale, sunt bazate pe tratarea șirurilor de biți ca reprezentări de polinoame cu coeficienți 0 și 1. Un cadru de  $k$  biți este văzut ca o listă de coeficienți pentru un polinom cu  $k$  termeni, de la  $X^{k-1}$  la  $X^0$ . Aritmetica polinomială este de tip modulo 2, nu există transport la adunare și nici împrumut la scădere, adunările și scăderile fiind identice cu sau-exclusiv (XOR).

Pentru utilizarea metodei, emițătorul și receptorul se pun de acord în avans asupra unui polinom generator  $G(x)$ , de grad  $r$ , la care atât bitul cel mai semnificativ cât și bitul cel mai puțin semnificativ să fie 1. Pentru a calcula suma de control a unui cadru de  $m$  biți, corespunzător polinomului  $M(x)$ , cadrul

trebuie să fie mai lung decât polinomul generator. Ideea este de a se adăuga la sfârșitul cadrului o sumă de control astfel încât polinomul rezultat să fie divizibil cu  $G(x)$ .

m biți	r biți	
M: biții de date	R: suma de control	Sir de biți (date+CRC)

Algoritmul de calcul este următorul:

1. Se adaugă r biți de 0 la capul mai puțin semnificativ al cadrului, astfel încât șirul rezultat va avea m+r biți, ce corespund polinomului  $X^r \cdot M(X)$

2. Se împarte șirul de biți ce corespund lui  $G(x)$  într-un șir de biți corespunzând lui  $X^r \cdot M(X)$ , utilizând împărțirea modulo 2.

Practic se aliniază biții care corespund intrării pe o linie, li se poziționează biții corespunzători polinomului generator sub marginea din stânga a liniei. Dacă bitul de intrare aflat deasupra bitului ce mai din stânga al divizorului este 0 se deplasează divizorul al dreapta cu 1 bit; altfel se face XOR între divizor și intrare, după care divizorul este deplasat la dreapta. Se repetă procedura până când divizorul atinge extrema dreaptă.

3. Se scade restul obținut din șirul de biți corespunzători lui  $X^r \cdot M(X)$ .

Considerăm șirul de biți 1101011011 și polinomul generator  $G(X) = X^4 + X + 1$  (10011).

$$\begin{array}{r}
 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\
 1 \ 0 \ 0 \ 1 \ 1 \\
 \hline
 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\
 1 \ 0 \ 0 \ 1 \ 1 \\
 \hline
 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\
 1 \ 0 \ 0 \ 1 \ 1 \\
 \hline
 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \\
 1 \ 0 \ 0 \ 1 \ 1 \\
 \hline
 0 \ 0 \ 1 \ 1 \ 1 \ 0 \text{ Suma de control}
 \end{array}$$

La nivel internațional au fost definite polinoame generatoare standardizate pe 8, 12, 16 și 32 de biți. Fiecare standard CRC poate detecta erori în rafale de până la r biți inclusiv. Mai mult, în anumite condiții pot fi detectate erori mai lungi de r biți cu probabilitatea  $1-0,5^r$ . De asemenea fiecare standard CRC poate detecta orice șir impar de biți eronați

## **1.5. Corecția erorilor**

Corecția erorilor este similară detecției erorilor, cu observația că receptorul nu doar sesizează faptul că a apărut o eroare în cadru ci și determină exact unde anume în cadru a apărut eroarea (pe care astfel o poate corecta). Unele protocoale de nivel legătură de date pun la dispoziție mecanisme de corectare a erorilor doar din antetul cadrului (nu din întreg cadrul).

## **1.6. Controlul fluxului**

Nodurile de la capetele legături au capacități limitate de memorare a cadrelor. Acest fapt poate deveni o problemă în momentul în care nodul receptor primește cadre la o rată mai mare decât capacitatea sa de procesare. Fără controlul fluxului, memoria tampon a receptorului va fi depășită iar cadrele vor fi pierdute. În mod similar nivelului transport, un protocol de nivel legătură de date poate asigura controlul fluxului în scopul prevenirii inundării nodului receptor de către nodul emițător.

## **1.7. Half-duplex/Full-duplex**

În cazul unei transmisii full-duplex, nodurile de la capetele legăturii pot transmite simultan pachete. În cazul unei transmisii half-duplex la un moment dat, un nod poate fie să transmită, fie să recepționeze (nu sunt posibile ambele).

Așa cum am menționat mai sus, multe din serviciile asigurate de nivelul legătură de date au corespondențe cu serviciile asigurate de nivelul transport. Spre exemplu, atât nivelul legătură de date cât și nivelul transport pot asigura livrarea fiabilă. Cu toate că mecanismele utilizate pentru livrare fiabilă sunt similare la ambele niveluri, cele două servicii de livrare fiabilă nu sunt identice. Un protocol de transport asigură livrarea fiabilă capăt-la-capăt a segmentelor între două procese; un protocol de livrare fiabilă la nivel legătură de date garantează livrarea fiabilă între două noduri conectat printr-o singură legătură. În mod similar, atât protocoalele de nivel legătură de date cât și cele de nivel transport pot asigura controlul fluxului și detecția erorilor; din nou, controlul fluxului la nivel transport este asigurat capăt-la-capăt, pe când la nivel legătură de date este asigurat între două noduri adiacente.

## **2. Unde este implementat nivelul legătură de date**

În continuare vom încerca să răspundem la întrebarea *unde* este implementat nivelul legătură de date de pe o stație gazdă. Este acesta implementat în hardware sau în software? Este acesta

implementat pe un card sau chip distinct, și cum este interfațat cu hardware-ul gazdei sau sistemul de operare?

Figura 6 prezintă o arhitectură tipică pentru o gazdă. În cea mai mare parte nivelul legătură de date este implementat în **adaptorul de rețea** cunoscut și sub denumirea de card pentru interfața la rețea (eng. Network interface card, NIC). Nucleul adaptorului de rețea este controlerul nivelului legătură de date, care este de regulă un chip specializat, independent care implementează multe din serviciile nivelului legătură de date (încadrare, acces la legătură, controlul fluxului, detecția erorilor etc.). Astfel, mare parte din funcționalitatea nivelului legătură de date este implementată în hardware. Până în anii 2000, majoritatea adaptoarelor de rețea erau carduri distincte (ISA, PCI, PCMCIA), însă după aceea din ce în ce mai multe adaptoare de rețea au fost integrate pe placa de bază a gazdei (așa numita configurație „on-board LAN”).

La nodul emițător, controlerul preia datagrama creată și stocată în memoria principală de către nivelurile superioare ale stivei, încapsulează datagrama într-un cadru de nivel doi (prin completarea câmpurilor specifice, calcularea secvenței de verificare a cadrului), după care, în baza protocolului de acces la legătură, transmite cadrul pe legătură de comunicație. La receptor, controlerul recepționează întreg cadrul, efectuează detecția erorilor prin recalcularea secvenței de control după care extrage datagrama de nivel rețea.

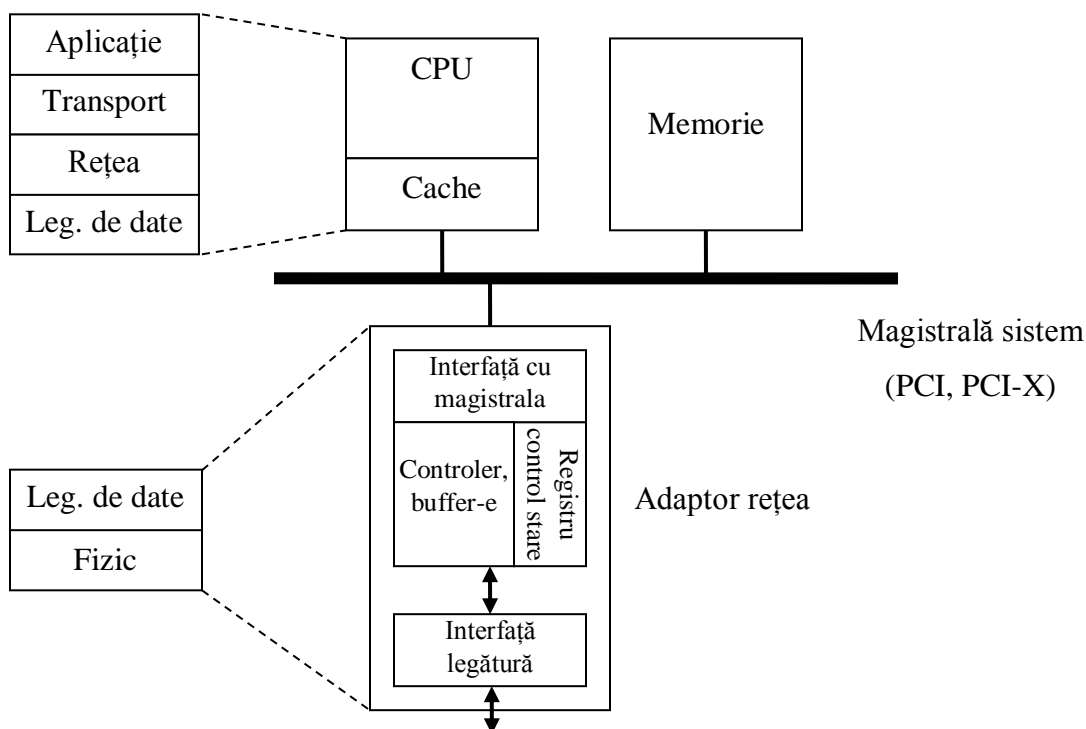


Figura 6

Figura 6 prezintă un adaptor de rețea atașat la magistrala sistemului gazdă (PCI, PCI-X etc), acesta fiind din perspectiva sistemului un dispozitiv I/O oarecare. De asemenea se poate observa că deși în mare nivelul legătură de date este implementat în hardware, o parte din funcționalitatea acestuia este implementată în software (driver de dispozitiv) ce rulează pe procesorul gazdei. Componentele software ale nivelului legătură de date implementează de regulă funcționalitățile de nivel înalt cum, a r fi recepționarea datagramei de la nivelul rețea, asamblarea informațiilor de adresare și activarea hardware-ului controlerului. În ceea ce privește nodul receptor software-ul de la nivel legătură de date se ocupă de tratarea condițiilor de eroare și transferul datagramei către nivelul rețea.

Un adaptor de rețea, ca orice alt dispozitiv este în ultimă instanță programat prin software-ul ce rulează pe procesorul gazdei. Din perspectiva procesorului, adaptorul exportă un *registru de control al stării* (eng. control status register, CSR). CRS-ul este în mod tipic localizat la o anumită adresă de memorie, fiind posibilă citirea și scrierea acestuia la fel ca orice locație de memorie. Procesorul scrie CSR pentru a solicita transmiterea/recepția unui cadru și citește CSR pentru a afla starea curentă a adaptorului.

Procesorul gazdei ar putea spre exemplu sta într-o buclă infinită, citind permanent registrul de control al stării, până la apariția unei eveniment de interes după care întreprinde acțiunile corespunzătoare. Aceasta tehnică, denumită *așteptare ocupată* (eng. busy waiting/pooling), poate fi utilizată în anumite situații (spre exemplu un ruter care nu face altceva decât să aștepte următorul cadru), însă nu este adecvată pentru o gazdă care utilizează procesorul pentru rularea altor aplicații. În loc să interogheze în permanență adaptorul de rețea, majoritatea gazdelor acordă atenție adaptorului de rețea doar când acesta din urmă întrerupe gazda. Dispozitivul generează o întrerupere atunci când apare un eveniment ce necesită intervenția procesorului, spre exemplu transmiterea sau recepționarea cu succes a unui cadru sau apariția unui eveniment de eroare legat de încercarea de a transmite sau recepționa un cadru. La apariția unei întreruperi controlul este transferat unei rutine de tratare a întreruperii care de regulă face parte din driverul de dispozitiv.

Una din problemele importante în proiectarea adaptoarelor de rețea este modul în care octeții sunt transferați în adaptor și memoria sistemului. Există două mecanisme: **acces direct la memorie** (eng. direct memory acces, DMA) sau **I/O programat** (eng. programmed I/O, PIO). În cazul DMA, adaptorul citește și scrie direct în memoria sistemului fără intervenția CPU; procesorul doar adaptorului adresele de memorie din care să citească/la care să scrie. În cazul PIO, procesorul este direct responsabil de transferul datelor între adaptor și memoria principală: pentru a trimite un cadru, procesorul intră într-o buclă în care mai întâi citește un cuvânt din memorie după care îl scrie la adaptor; pentru a primi un cadru procesorul citește cuvinte de la adaptor și le scrie în memorie. În cazul

DMA adaptorul nu necesită memorie locală (eventual doar pentru un cadru) pe când la PIO adaptorul are nevoie să memoreze eventual chiar mai multe cadre întrucât procesorul s-ar putea să fie ocupat cu alte sarcini atunci când sosesc cadrele.

Pe lângă funcțiile nivelului legătură de date anumite adaptoare de rețea (de ex. Intel 8254x) sunt capabile să efectueze în hardware și funcții ce aparțin nivelului rețea (suma de control a antetului IP) sau transport (suma de control TCP/UDP). Deși o astfel de abordare încalcă flagrant independența nivelurilor, avantajul este dat de faptul că în hardware sumele de control sunt calculate mult mai rapid decât în software.