

Universitatea Politehnica Bucuresti

Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei

Protocoale la nivelul legaturi de date

Studenti : Madalin Nica

Ana – Maria Radu

Grupa : 442 A

Cuprins

1. Introducere (Nica Madalin).....	3
2. Protocele la nivelul legaturii de date (LD) (Radu Ana-Maria)	5
3. Protocolul High-level Data Link Control – HDLC (Radu Ana-Maria).....	8
4. Nivelul LD in Internet (Nica Madalin).....	12
4.1 Protocolul Punct la Punct (PPP)	12
4.2 Protocolul Serial Link Internet Protocol (SLIP)	14
Bibliografie	15

1. Introducere

(Madalin Nica)

Legatura de date este un ansamblu compus din elementele a doua echipamente terminale de date, care sunt controlate de un protocol si care, prin intermediul circuitului de date care le interconecteaza, permit, impreuna, transferul datelor. Schema unei legaturi de date este aratata in Figure 1 .

Nivelul legatura de date este realizat pe conexiunea fizica asigurata de un circuit (fie el ai dintr-o retea), pentru a furniza un serviciu de transfer de date fiabil nivelului retea sau, direct, nivelului aplicatie.

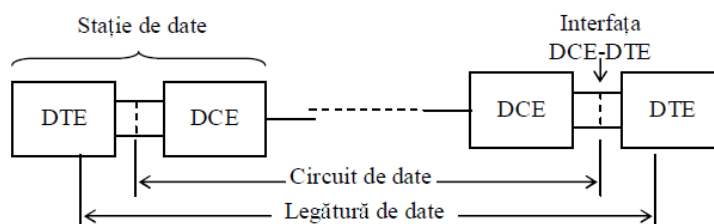


Figure 1 : Legatura de date

Circuitul de date este ansamblul format din doua canale de transmisiune asociate pentru a asigura transmiterea datelor in ambele sensuri. Statia de date este o unitate functionala care furnizeaza date pentru transmisiune, primeste datele transmise si realizeaza toate functiunile necesare pentru comunicatia cu o alta unitate functionala.

Protocolul legaturii de date este constituit dintr-un set de reguli care determina comportarea unitatilor functionale in cursul comunicatiei, urmarind ca informatia transferata sa fie receptionata si interpretata corect. Legatura de date poate fi considerata sub doua aspecte:

- fizic, cu referire la circuitul de date si transmisiunea datelor ;
- logic, legat de comanda legaturii si coordonarea transferului de date pentru a-l face sigur si eficace.

Din punct de vedere al configuratiei fizice legatura de date poate fi de mai multe tipuri: punct-la-punct, multipunct, sau in bucla, asa cum arata Figure 2.

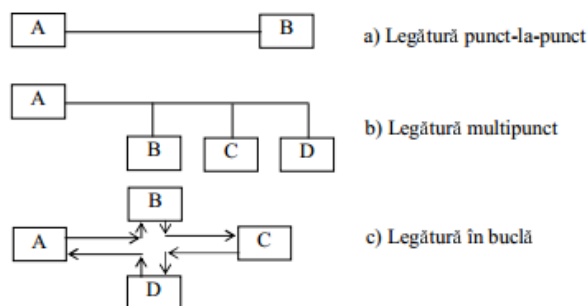


Figure 2 : Configuratii ale legaturii de date

Pentru a asigura transferul sigur si eficient al datelor protocolul legaturii de date trebuie sa realizeze urmatoarele functiuni principale:

- controlul erorii ;
- controlul fluxului ;
- formatarea datelor in cadre (blocuri) ;
- identificarea sursei si destinatiei datelor (in legaturile multipunct si in bucla).

Desigur, protocolul legaturii de date trebuie sa aiba in vedere si situatiile anormale de functionare care pot surveni in cursul transferului datelor: intreruperi ale legaturii, saturarea unei statii, erori frecvente, etc.

De asemenea, protocolul trebuie sa fie adecvat tipului aplicatiei. Aplicatiile pot fi de tipul :

- Conversational ;
- prelucrare pe loturi ;
- prelucrare in timp real ;
- transfer de fisiere, etc.

Corespunzator tipului aplicatiei, functionarea legaturii de date poate diferi de la o aplicatie la alta. Astfel :

- in **aplicatiile conversationale** un mesaj este emis intr-un sens, in celalalt sens fiind transmis mesajul de raspuns, alternativ, ca intr-o conversatie si legatura de date trebuie sa functioneze in modul bidirectional alternant ;
- in **prelucrarile pe loturi datele** sunt transmise intr-un volum mare intr-un sens, catre centrul de prelucrare. Rezultatele sunt transmise in celalalt sens, dupa terminarea prelucrarii, care poate dura un timp indelungat. Legatura functioneaza in modul bidirectional alternant, dar cu schimbari rare ale sensului de transmisiune ;
- pentru **prelucrarile in timp real** datele receptionate trebuie prelucrate rapid si rezultatele (sau deciziile) trebuie transmise imediat. Legatura trebuie sa functioneze in modul bidirectional simultan. In aplicatiile spatiale, spre exemplu, timpul devine un factor primordial si, pentru controlul erorii, se pot folosi codurile corectoare de erori.

Protocolul legaturii de date trebuie sa tina seama si de caracteristicile traficului de date in cele doua sensuri de transmisiune. De lungimea cadrelor si volumul mesajelor care trebuie transmise vor depinde dimensiunile memoriilor de emisie si de receptie, procedurile de reluare in cazurile de functionare anormala, timpii de asteptare si de raspuns. Protocolul legaturii de date trebuie sa tina seama, de asemenea, si de caracteristicile circuitului de date: timpul de propagare, debitul datelor si de procentul de erori.

2. Protocoale la nivelul legaturii de date (LD)

(Ana – Maria Radu)

Principalul serviciu oferit nivelului LD este transferul datelor de la un nod la altul între sursa și destinație prin verificarea și corectarea în nodurile intermediare a erorilor de transmisie. El trebuie să pună la dispoziția nivelului rețea linii sigure între două noduri sau cel puțin să le facă să se comporte ca fiind sigure, chiar dacă pe ele apar erori sau alte probleme de transmisie. În felul acesta, nivelul rețea găsește canale sigure între sursa și destinație, indiferent prin câte linii și noduri intermediare trece, indiferent dacă sunt linii fir, radio, satelit, fibra optică etc. Acest serviciu poate fi executat în mai multe moduri:

- **neconfirmat și fara conexiune:** mașina sursă trimite cadre unul după altul, pe rute stabilite pe moment, fără ca destinația să confirme recepția lor corectă. Confirmarea presupune un răspuns în sens invers. Dacă un cadru s-a pierdut, nu se încearcă retransmiterea lui, această sarcină revenind nivelurilor superioare (transport). Lucrul acesta este admisibil dacă rata erorilor de transmisie este mică. Este adecvat transmisiile de date în timp real, cum ar fi voce sau imagini video în timp real, aplicații care nu suportă întârzieri variabile;
- **confirmat fara conexiune:** fiecare cadru este confirmat individual, astfel încât transmitatorul știe în fiecare moment dacă fiecare cadru a fost recepționat corect. Este adecvat transmisiilor pe canale nesigure, cum ar fi cele fără fir, cu zgomot mare sau gaze mobile;
- **confirmat și orientat pe conexiune:** este cel mai sigur și pe lângă confirmarea recepției corecte, asigură și ordinea firească a cadrelor. El presupune existența celor trei faze: stabilirea conexiunii, transferul datelor și desfacerea conexiunii. În prima fază mașinile din nodurile adiacente stabilesc conexiunea, rezerva resurse (banda, memorii tampon etc.), initializează contoare și alte variabile de control pentru a ține evidența cadrelor transmise și recepționate corect, a celor retransmise etc. În faza a doua se transmit cadrele de date (individual sau blocuri, menținând evidența confirmării lor). În faza a treia se eliberează conexiunea, se eliberează resursele angajate și se resetează variabilele setate anterior.

Nivelul LD este în mod obișnuit împărțit în două subnivele:

- subnivelul de **control al legaturii logice (LLC – Logical Link Control)**: se ocupă de formarea cadrelor, controlul erorilor, servicii de confirmare dacă este cazul, interfața cu nivelul superior etc. indiferent cum este partajat mediul de transmisie. El creează o interfață uniformă între nivelele superioare și subnivelul MAC;
- subnivelul de **control al accesului la mediu (MAC- Medium Access Control)**: are două roluri majore: stabilirea și respectarea regulilor de acces la mediu comun de transmisie a mai multor utilizatori și adaptarea la mediul fizic, astfel încât, să ascundă diferențele legate de diferite medii de transmitere, forme de semnal, coduri de linie etc.

Formatul cadrelor diferă de la un protocol la altul, dar o formă generală este cea din Figure 3.

Start	Adresa	Control	Date utilizator	Verificare	Stop
-------	--------	---------	-----------------	------------	------

Figure 3 : Formatul general al unui cadru LD

Campurile *Start* si *Stop* au structura fixa si reprezinta delimitatori de cadru. Campul *Adresa* contine adresele de nivel fizic (sau MAC) ale sursei si ale destinatiei. Campul *Control* are rolul de a permite controlul transmisiei în functie de timpul de receptie, inclusiv prelucrare si retransmisie in caz de erori. Blocul de verificare este destinat monitorizarii erorilor de transmisie. Cea mai simpla metoda de control a erorilor este bitul de paritate. O alta metoda mai elaborata este suma de control. Ea se efectueaza la emisie, se înscrie în campul de control si se verifica la receptie. Daca valorile sunt diferite, rezulta ca in timpul transmisiei au aparut erori si se iau decizii în consecinta. Verificarea erorilor se poate face pentru tot blocul de date (tot cadrul) sau numai pentru antet.

Controlul receptiei cadrelor se face in mod uzual prin proceduri de:

- **confirmare pozitiva (positive acknowledgment, abreviat ack)**, sau
- **confirmare negativa (negative acknowledgment, nak)** trimise pe o cale de reactie de la receptor la transmitator.

Fiecare confirmare pozitiva arata un cadru receptionat corect, fiecare confirmare negativa arata o eroare si presupune retransmiterea cadrului. Daca dupa un timp prestabilit nu soseste nici ack nici nak, cadrul este retransmis automat.

Exista trei moduri de realizare a functiilor ack/nak:

- **Protocolul stop and wait** : Se transmite un cadru si se asteapta ack sau nak. Daca nu apare nici o confirmare, cadrul se retransmite;
- **Transmisia go back to N** : Cadrele se transmit continuu daca exista, fara a astepta confirmarea. Cand apare primul nak sau dupa un timp prestabilit se retransmit cadrul in cauza si toate de dupa el;
- **Protocolul repetarii selective** : Se retransmite numai cadrul nevalidat.

In ceea ce priveste controlul fluxului exista doua abordari des utilizate. Prima se bazeaza pe reactie, atunci cand transmitatorul emite in ritmul sau prestabilit pana cand receptorul se satureaza si semnalizeaza pe calea inversa sa opreasca sau sa reduca rata emisiei. A doua metoda bazata pe controlul ratei de reansmisie presupune o adaptare permanenta a ratei de transmisie la starea receptorului (a posibilitatii lui de a receptiona fluxul de date).

Monitorizarea si/sau detectia si corectia erorilor

Exista mai multe posibilitati. Cea mai simpla este bitul de paritate. Acesta este un bit plasat intr-un camp special in antet si care este 0 sau 1, ales astfel incat suma tuturor bitilor din cadru (sau numai din unele campuri din cadru) sa aiba o valoare prestabilita, para sau impara. Aceasta suma se verifica la receptie si daca

nu se confirma valoarea asteptata, rezulta ca exista erori de transmisie. Metoda este foarte simpla si usor de aplicat. Marele neajuns este ca nu poate sesiza decat existenta unui numar impar de biti eronati.

O metoda mai sofisticata de detectie si /sau corectie a erorilor consta în folosirea codurilor detectoare / corectoare de erori. O notiune de baza in teoria acestor coduri este **distanța Hamming**. Aceasta reprezinta numarul de pozitii prin care difera între ele oricare doua cuvinte de cod. Pentru a detecta d erori, distanta Hamming trebuie sa fie $d+1$ iar pentru a corecta d erori, distanta trebuie sa fie $2d+1$. Prima afirmatie este adevarata deoarece prin schimbarea a d biti nu exista posibilitatea de a da peste un alt cuvânt de cod valabil. Codurile detectoare/corectoare de erori sunt coduri redundante. Ele au k biti informationali si r biti redundanti. Cele mai folosite coduri detectoare de erori sunt codurile polinomiale, iar metoda de detectie se numeste **verificarea ciclica a redundantei (CRC – Cyclic Redundancy Check)**. Aceste coduri au la baza un polinom generator $G(x)$ asupra caruia cad de acord si transmitatorul si receptorul. Mesajul de transmis $M(x)$ este impartit la $G(x)$ iar restul obtinut se adauga la $M(x)$ ca informatie redundanta. La receptie face din nou impartirea mesajului receptionat la $G(x)$. Daca restul impartirii este 0 rezulta ca transmisia s-a facut fara erori. Daca restul impartirii este diferit de zero inseamna ca au fost erori de transmisie si se trece la corectia lor. Metodele de corectie sunt mai complicate si pot fi studiate din literatura de specialitate.

Exemple de protocoale LD

La nivel legatura de date sunt standardizate mai multe protocoale elaborate de organisme de standardizare:

- **ISI** : HDLC, ISO 3309, ISO 4305;
- **CCITT** : LAPB, LAPD;
- **IBM** : SDLC, BSC;
- **DEC** : DDCMP;
- **ANSI** : ADCCP (Advanced Data Communication Control Protocol);
- **Internet** : PPP.

In continuare, le vom prezenta pe cele mai importante dintre acestea.

3. Protocolul High-level Data Link Control - HDLC

(Ana – Maria Radu)

Formatul general al cadrului pentru multimea de protocoale HDLC este:

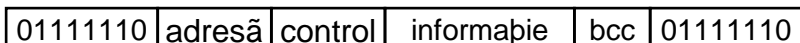


Figure 4 : Formatul general pentru protocolul HDLC

In Figure 4, campul *adresa* identifica pe liniile multipunct una din statiile secundare; pentru uniformitate se foloseste si pe liniile punct-la-punct. Campul *control* contine numere de secventa, confirmari, tipul cadrului si are configuratii diferite dupa tipul cadrului. Campul *bcc* reprezinta un bloc de control al cadrului, si este o varianta a codului ciclic detector de erori CRC ITU-T (CCITT).

In functie de structura campului *control* exista 3 tipuri de cadre:

- **Cadre de informatie** : sunt singurele cadre numerotate; ele pot fi confirmate de cadrele S si I care contin informatie de confirmare (nr) ;

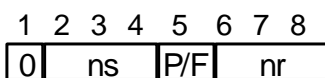


Figure 5 : Cadre de informatie

Unde : *ns* = numarul cadrului (de) transmis, *nr* = numarul cadrului asteptat la receptie, iar *P/F* are urmatoarele semnificatii : in cadrele de comanda *P/F* = P (Poll), iar *P*=1 are semnificatia de invitatie la emisie, respectiv in cadrele de raspuns *P/F* = F (Final) si *F*=1 are semnificatia de sfarsit de transmisie.

- **Cadre supervizoare** : pot fi de urmatoarele tipuri :
 - *Receiver Ready (RR)* - confirma receptia tuturor cadrelor anterioare lui *nr* si anunta ca este gata sa receptioneze cadrul cu numarul *nr* ;
 - *Receiver Not Ready (RNR)* - confirma receptia tuturor cadrelor anterioare lui *nr* si cere oprirea transmisiei, urmând sa anunte transmitatorul cand va fi gata de receptie cu un cadru RR, REJ sau SREJ ;
 - *Reject (REJ)* - confirma receptia tuturor cadrelor anterioare lui *nr* si cere retransmisia cadrelor începând cu *nr*. Toate cadrele ce sosesc inaintea cadrului *nr* vor fi ignorate ;
 - *Selective Reject (SREJ)* - confirma receptia tuturor cadrelor anterioare lui *nr* si cere retransmisia cadrului *nr*. Toate cadrele diferite de *nr* primite vor fi memorate in limita ferestrei de receptie.

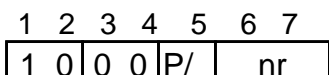


Figure 6 : Cadru RR

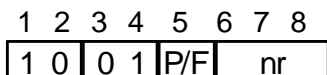


Figure 7 : Cadru RNR

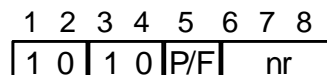
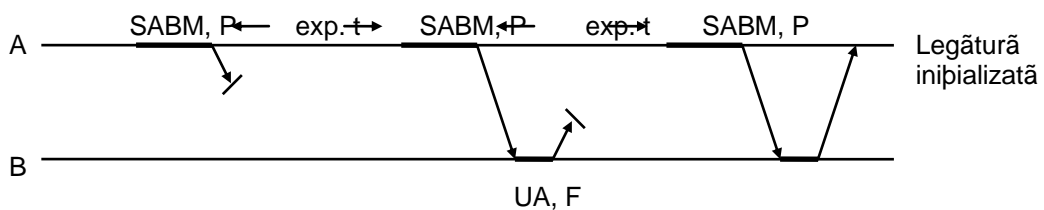


Figure 8 : Cadru REJ

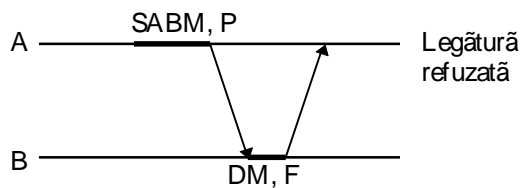
- **Cadre nenumerotate (de gestiune a legaturii) :** pot fii :
 - *Set Asynchronous Balanced Mode (SABM)* - comanda de initializare a unei legaturi echilibrate de date in modul asincron ;
 - *Disconnect (DISC)* - comanda de desfiintare a legaturii de date;
 - *Unnumbered Acknowledge (UA)* – raspuns de confirmare a cadrului de gestiune a legaturii de date receptionat ;
 - *Disconnected Mode (DM)* - raspuns de respingere a unei comenzi de initializare a unei legaturi (SABM) sau de acceptare a unei comenzi de desfiintare a unei legaturi (DISC);
 - *Unnumbered Information (UI)* - comanda si raspunsul de acceptare a comenzii de trecere a legaturii de date respective la serviciul fara confirmare si fara conexiune.

Initializarea si desfiintarea unei legaturi

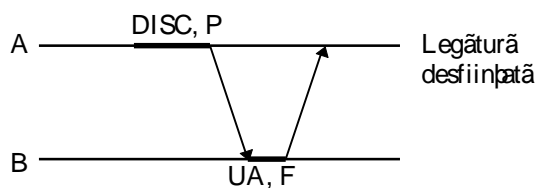
1. Stabilirea cu erori a legaturii



2. Respingerea cererii de stabilire a unei legaturi



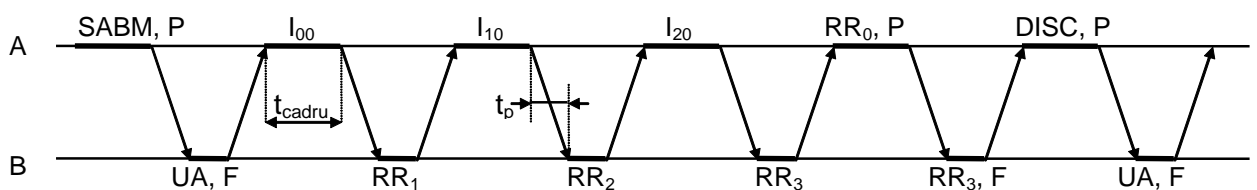
3. Desfiintarea unei legaturi



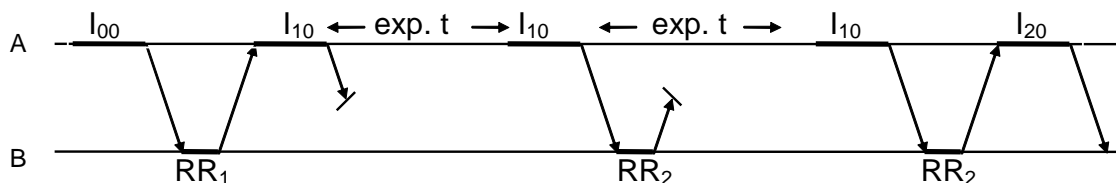
Protocolul de oprire si asteptare (stop and wait)

Acest protocol are eficienta redusa pentru t_p/t_{cadru} mare, unde $t_p = t_{propagare} + t_{prelucrare}$

1. A transmite 3 cadre

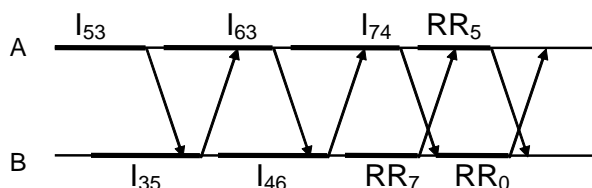


2. A transmite 3 cadre, al doilea fiind eronat



Protocolul cu transmisie continua, cu REJ (uzual)

1. A transmite 3 cadre, B transmite 2 cadre, nu exista erori ($nt = 7, nr = 1$)



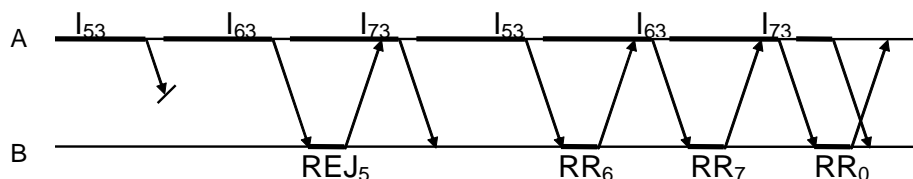
Unde :

- Fereastra la transmisie = domeniul numerelor de secvență care pot fi folosite la un moment dat pentru cadre transmise (nt) ;

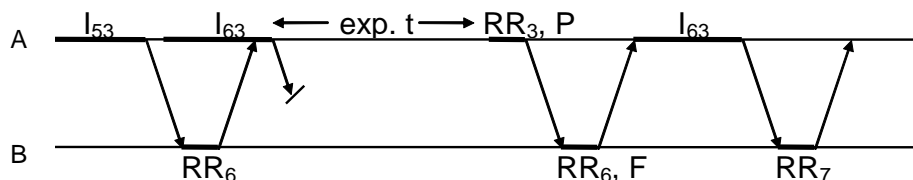
- Fereastra la receptie = domeniul numerelor de secvență ale cadrelor ce pot fi acceptate la receptie (nr).

Fiecare statie are 3 variabile de control: va = numarul primului cadru transmis si neconfirmat, vs = numarul urmatorului cadru de transmis si vr = numarul cadrului asteptat la receptor.

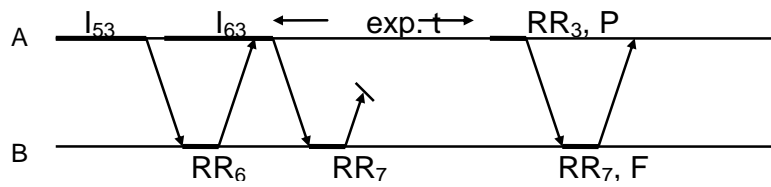
2. A transmite 3 cadre, primul eronat



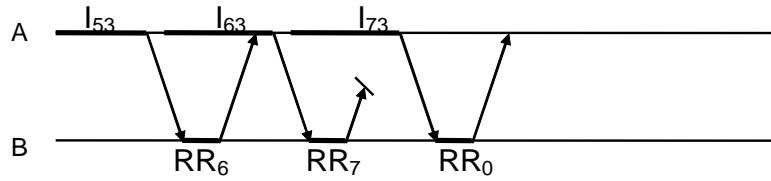
3. A transmite 2 cadre, ultimul eronat (expirare de timp)



4. A transmite 2 cadre, confirmarea ultimului cadru este eronată (expirare de timp)

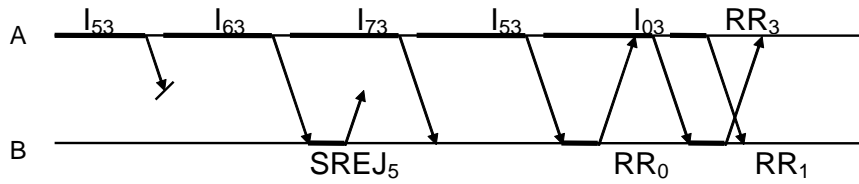


5. A transmite 3 cadre, confirmarea celui de-al doilea se pierde



3.4 Protocolul cu retransmisie selectiva, cu SREJ

A transmite 4 cadre, primul eronat; $nt = 4$, $nr = 4$



4. Nivelul LD in Internet

(Madalin Nica)

Internetul este o colectie de calculatoare interconectate prin rutere si linii de comunicatii. În cadrul unei cladiri sunt instalate LAN-uri de diverse tipuri care deservesc nevoile de aplicatii ale unui grup restâns si oarecum omogen de utilizatori. MAN-urile si WAN-urile interconecteaza retelele locale folosind o infrastructura de comunicatii ce cuprinde linii închiriate în configuratie punct la punct. O varianta des folosita de utilizatorii casnici este aceea a legaturilor comutate bazate pe modemuri. Calculatorul personal de la domiciliu apeleaza ruterul unui furnizor de servicii Internet (ISP). În ambele situatii, si pe linie închiriată si pe linie comutată, este nevoie de un protocol la nivel legatura de date punct la punct pentru formarea cadrelor, controlul erorilor si alte functii ale legaturii de date. Cel mai folosit protocol este PPP.

4.1 Protocolul Punct la Punct (PPP)

Pentru a imbunatati situatia, IETF a definit un grup pentru proiectarea protoalelor legaturii de date pentru liniile punct-la-punct, care sa rezolve toate aceste probleme si care sa poata deveni un Standard Internet. Aceasta activitate a culminat cu PPP care este definit în RFC 1661 si dezvoltat în alte câteva RFC-uri (de exemplu RFC-urile 1662 si 1663). PPP face detectia erorilor, suporta mai multe protoale, permite ca adresele IP sa fie negociate in momentul conectarii, permite autentificarea si are multe alte imbunatatiri fata de SLIP. In timp ce multi furnizori de servicii Internet ofera inca suport atat pentru SLIP cât si pentru PPP, viitorul apartine cu siguranta PPP-ului, atat pentru liniile comutate cat si pentru liniile inchiriate ruter-ruter.

PPP furnizeaza trei lucruri:

- **Metoda de impartire in cadre** care delimiteaza, fara ambiguitate, sfarsitul unuia si inceputul urmatorului. Formatul cadrului permite si detectia de erori ;
- Un **protocol de legatura** pentru a obtine liniile, a le testa, a negocia optiunile si pentru a elibera liniile atunci cand nu mai este nevoie de ele. Acest protocol se numeste *LCP(Link Control Protocol - protocolul de control al legaturii)* ;
- Un **mod de a negocia optiunile nivelului retea** intr-un mod independent de protocolul folosit pentru nivelul retea. Metoda aleasa este de a avea un *NCP (Network Control Protocol - protocolul de control al retelei)* pentru fiecare nivel de retea suportat.

Formatul cadrului PPP

A fost ales foarte asemanator cu formatul cadrului HDLC deoarece nu exista nici un motiv pentru a se reinventa roata. Diferenta majora între PPP si HDLC este ca primul este mai degraba orientat pe caractere decât pe biti. In particular, PPP, ca si SLIP, foloseste umplerea cu caractere pe liniile comutate prin modem, astfel încât toate cadrele au un numar întreg de octeti. Nu este posibil sa se trimita un cadru constand din 30.25 octeti, asa cum era la HDLC. Cadrele PPP pot fi transmise nu

numai pe liniile telefonice comutate, ele pot fi transmise si pe linii SONET sau linii HDLC, cu adevarat orientate pe biti (de exemplu pentru conexiuni ruter-ruter). Formatul protocolului PPP este aratat in Figure 9.

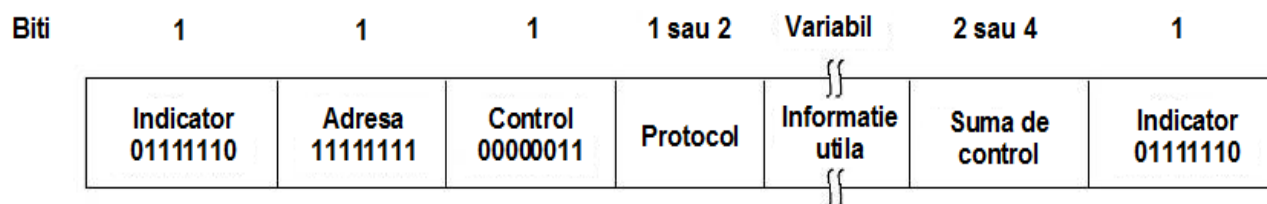


Figure 9 : Protocolul PPP

Motivele pentru care PPP lucreaza cu inserare de caractere si lucreaza cu bytes sunt :

- PPP a fost proiectat pentru a fi implementat software, nu hardware ca in cazul HDLC-ului. In cazul implemenatarilor software , lucrul cu bytes este mult mai facil decat lucrul cu biti ;
- PPP a fost creat pentru a lucra cu modem-uri , iar modem-urile accepta si transmit date in bytes, nu in biti.

Fazele unei linii

Diagrama simplificata din Figure 10 arata fazele prin care trece o linie atunci cand este stabilita, folosita si eliberata. Secventa se aplica atat pentru conexiunile prin modem cat si pentru conexiunile ruter-ruter.

Cand o linie este "Moarta", nu este prezenta nici o purtatoare la nivel fizic si nu exista nici o conexiune fizica. Dupa ce este stabilita conexiunea fizica, linia trece in "Stabilire", in acest punct incepe negocierea optionala LCP care, daca reuseste, conduce la "Autentificare". Acum cele doua parti pot sa-si verifice una alteia identitatea, daca doresc. Cand se intra in faza "Retea", este invocat protocolul NCP corespunzator pentru a configura nivelul retea. Daca configurarea se face cu succes, este atinsa faza "Deschidere" si poate avea loc transportul datelor. Gand transportul datelor este terminat, linia este trecuta in faza "Terminare" si, de aici, inapoi in "Moarta" unde purtatoarea este intrerupta.

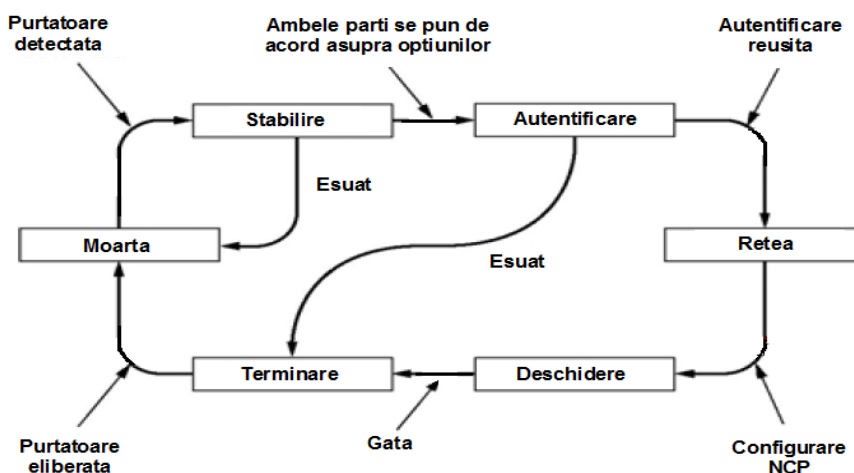


Figure 10 : Fazele unei linii de legatura

4.2 Protocolul Serial Link Internet Protocol (SLIP)

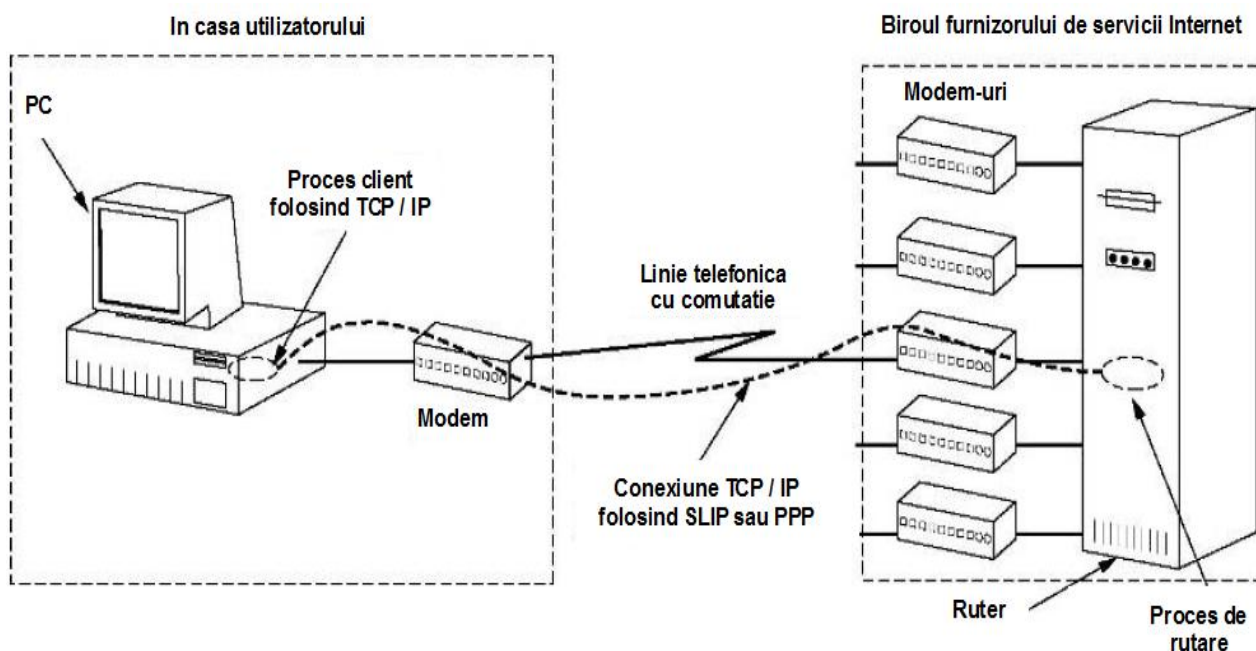


Figure 11 : Principiul protocolului SLIP

Principii generale ale protocolului SLIP

- Nu face nici un fel de detectie sau corectie a erorilor, astfel incat trebuie ca niveluri superioare sa detecteze si sa recupereze cadrele pierdute, distruse sau amestecate;
- Este un protocol de incadrare a pachetelor;
- Suporta doar IP, odata cu cresterea Internet-ului, au aparut si retele care nu folosesc IP ca limbaj nativ, ceea ce face ca aceasta restrictie sa devina din ce in ce mai serioasa;
- Nici o adresa nu poate fi asociata dinamic;
- Nu furnizeaza nici o forma de autentificare, astfel incat nici una din entitati nu stie cu cine urmeaza sa comunice;
- Este folosit pentru conexiuni seriale punct-la-punct peste care ruleaza TCP/IP intre gazde si rutere;
- Nu este un standard Internet.

Reguli Protocol:

- Defineste doua caractere speciale: END si ESC;
- O gazda SLIP trimite date in pachet;
- END in pachet inlocuit cu ESC si octal 334;
- ESC in packet inlocuit cu ESC si octal 335;
- Dupa ultimul octet din pachet se transmite END.

Bibliografie

- [1] «Transportul datelor pe o legatura de date,» [En ligne]. Available: <http://etti.poly.ro/>. [Accès le Ianuarie 2013].
- [2] «The Data Link Layer (Layer2),» [En ligne]. Available: http://defoenet.com/ccna/osi_l2.html. [Accès le Ianuarie 2013].
- [3] «Nivelul legatura de date,» [En ligne]. Available: http://shannon.etc.upt.ro/laboratoare/pc/luc6/2_Nivelul_2.htm. [Accès le Ianuarie 2013].
- [4] «PPP and SLIP Protocols,» [En ligne]. Available: <http://en.kioskea.net/contents/internet/ppp.php3>. [Accès le Ianuarie 2013].
- [5] «HDLC - A Technical Overview,» [En ligne]. Available: <http://vkalra.tripod.com/hdlc.html>.
- [6] «Protocole la nivel LD,» [En ligne]. Available: <http://clodel.com>. [Accès le Ianuarie 2013].
- [7] «Cum functioneaza modelul OSI si modelul de interconectare a sistemelor deschise,» [En ligne]. Available: <http://www.scientia.ro/tehnologie/34-cum-functioneaza-calculatorul/419-cum-functioneaza-osi-modelul-de-interconectare-a-sistemelor-deschise.html>. [Accès le Ianuarie 2013].
- [8] «Nivelul Legaturilor de Date,» [En ligne]. Available: <http://www.scrigroup.com/calculatoare/NIVELUL-LEGATURILOR-DE-DATE62142.php>. [Accès le Ianuarie 2013].