

Transmisiuni analogice și digitale

Acesta este primul curs și pornește de la zero: nu presupune că știți ce e un bit, ce e un protocol sau de ce un cablu de rețea are opt fire. Până la final veți înțelege ce se întâmplă, fizic, între momentul în care apăsați Enter și momentul în care mesajul ajunge la celălalt capăt - și veți avea vocabularul cu care se poartă toate discuțiile din restul semestrului.

OBIECTIVELE CAPITOLULUI

- Să explicați ce este un bit, un octet și de ce 100 Mbps nu înseamnă 100 MB pe secundă
- Să clasificați o rețea după acoperire: LAN, MAN, WAN
- Să spuneți, pentru orice echipament de rețea, la ce nivel lucrează și ce decizie ia
- Să explicați de ce comunicația e organizată pe niveluri și să treceți între modelele OSI și TCP/IP
- Să citiți și să desenați forma de undă pentru NRZ-L, NRZ-I, Manchester, MLT-3 și PAM-5
- Să calculați debitul binar pornind de la rata de simbol și schema de modulație

1.1 Despre ce este vorba, de fapt

Aveți două calculatoare într-o cameră și vreți ca unul să îi trimită celuilalt un fișier. Pare simplu. Nu este. Ca să funcționeze, cineva a trebuit să răspundă, pe rând, la vreo cincisprezece întrebări:

- Ce fel de fir le leagă și cât de lung poate fi?
- Cum arată, în tensiune electrică, cifra 1? Dar cifra 0?
- De unde știe receptorul unde se termină un bit și începe următorul?
- Cum știe că mesajul a început? Cum știe că s-a terminat?
- Dacă amândoi vorbesc simultan, ce se întâmplă?
- Dacă un bit se strică pe drum, cum ne dăm seama?

- Dacă sunt zece calculatoare, nu două, cum spui cui îi este destinat mesajul?
- Dacă celălalt calculator e în altă clădire? În altă țară?

Fiecare dintre aceste întrebări are un răspuns, iar răspunsurile se numesc **proto-coale**. Cursul acesta răspunde la primele cinci - cele fizice. Restul semestrului urcă, întrebare cu întrebare, până la ultima.

Dacă e prima dată când auziți asta. O rețea de calculatoare este interconectarea mai multor sisteme de calcul astfel încât ele să poată schimba informație. Definiția e scurtă, dar ascunde o diferență importantă.

În **interiorul** unui calculator, componentele comunică prin trasee de cupru pe placa de bază, lungi de câțiva centimetri, controlate de chipset. Nimeni nu se atinge de ele, nimeni altcineva nu le ascultă, iar semnalul ajunge întotdeauna intact.

Între calculatoare nu mai avem niciuna dintre aceste garanții: distanța poate fi de metri sau de mii de kilometri, mediul poate fi zgomotos, iar celălalt capăt poate să aparțină altcuiva. Tot ce urmează în acest curs există ca să compenseze aceste trei lipsuri.

1.2 Vocabularul minim: bit, octet, semnal

Nu se poate discuta despre rețele fără patru cuvinte. Le luăm pe rând, fără să presupunem nimic.

Bitul. Un **bit** este cea mai mică unitate de informație posibilă: un răspuns la o întrebare cu două variante. Da sau nu. Aprins sau stins. 1 sau 0. Nu există nimic mai mic, pentru că o întrebare cu o singură variantă nu transmite nicio informație.

Un bit singur e aproape inutil. Puterea vine din combinare: cu 2 biți exprimați 4 lucruri diferite, cu 3 biți 8, cu 8 biți 256. Regula este că **n biți exprimă 2^n valori** - și de aici vine practic toată aritmetica din acest curs.

Octetul și numerele. Un **octet** (în engleză *byte*) este un grup de 8 biți. Poate reprezenta $2^8 = 256$ de valori diferite, adică numerele de la 0 la 255. Nu e o alegere întâmplătoare: 255 e exact numărul maxim dintr-un octet al unei adrese IP, iar asta explică de ce nu există adresa 192.168.300.1.

Același octet poate fi scris în trei feluri, care înseamnă exact același lucru:

- **zecimal** - cum numărăm noi: 141
- **binar** - cum „gândește” echipamentul: 10001101
- **hexazecimal** - o prescurtare comodă a binarului: 0x8D

Conversia binar ↔ zecimal nu este magie și nu se învață pe de rost. Fiecare poziție dintr-un octet are o **greutate** - de la dreapta la stânga: 1, 2, 4, 8, 16, 32, 64, 128. Valoarea zecimală este pur și simplu suma greutateilor pozițiilor care conțin 1. Atât.



Figura 1.1 Convertor și antrenor de numerație.

Jucați-vă cu el până când conversia devine reflex. Veți avea nevoie de ea la fiecare curs de aici înainte, iar la subnetare (cursul 5) va fi singura unealtă care contează. Apăsați „exersează” și încercați să ghiciți reprezentarea binară înainte de a se afișa.

Hexazecimalul, în trei propoziții. Baza 16 folosește cifrele 0–9 plus literele A–F pentru valorile 10–15. Motivul pentru care există este pur practic: **exact 4 biți încap într-o cifră hexazecimală**, deci un octet se scrie mereu cu două cifre hexa, iar conversia se face pe grupe de câte patru, fără niciun calcul.

$1000\ 1101 \rightarrow 8\ \text{și}\ D \rightarrow 0x8D$. Din acest motiv adresele MAC și cele IPv6 se scriu hexazecimal: sunt mult mai scurte decât în binar și mult mai ușor de convertit decât zecimalul.

Semnalul. Un bit este o idee. Ca să ajungă de la un calculator la altul, ideea trebuie transformată în ceva fizic: o tensiune pe un fir, o pulsație de lumină într-o fibră, o undă radio în aer. Acest „ceva fizic” se numește **semnal**.

Tot restul cursului se ocupă de o singură întrebare: **cum transformăm un șir de biți într-un semnal pe care celălalt capăt îl poate citi corect** - și înapoi.

Capcana unităților

Aici se produce cea mai frecventă neînțelegere dintre un administrator de rețea și un utilizator nemulțumit. Vitezele de rețea se măsoară în **biți**, iar dimensiunile fișierelor în **octeți**. Între ele e un factor de 8 - plus o a doua diferență, subtilă.

Tabelul 1.1 Capcana unităților.

Notăție	Se citește	Valoare	Unde apare
b	bit	1 bit	viteze de transmisie
B	octet (byte)	8 biți	dimensiuni de fișiere
kbps, Mbps	kilo/megabiți pe secundă	$10^3, 10^6$ biți/s	lățime de bandă
KB, MB	kilo/megaocteți	$2^{10}, 2^{20}$ octeți	capacitate de stocare

Exercițiu rezolvat 1.1. Aveți o conexiune de 100 Mbps. În cât timp, în cel mai bun caz, descărcați un fișier de 500 MB?

Rezolvare. $500 \text{ MB} = 500 \times 2^{20} \text{ octeți} = 524\,288\,000 \text{ octeți} = \mathbf{4\,194\,304\,000 \text{ biți}}$.

$100 \text{ Mbps} = 10^8 \text{ biți pe secundă}$.

$4\,194\,304\,000 / 100\,000\,000 \approx \mathbf{42 \text{ de secunde}}$ - și asta doar în cazul ideal, fără anteturi de protocol, fără retransmisii, fără nimeni altcineva pe legătură. În realitate, socotiți 15–20 % în plus.

Observați: un utilizator care se aștepta la „500 MB pe 100 Mbps înseamnă 5 secunde” a greșit de opt ori - exact factorul dintre bit și octet.

1.3 Cât de mare e o rețea: LAN, MAN, WAN

Cea mai veche clasificare a rețelelor este după distanța dintre noduri. Nu e o simplă convenție de vocabular: fiecare clasă și-a dezvoltat protocoale proprii, potrivite ordinului ei de mărime, pentru că problemele sunt cu adevărat diferite la 10 metri față de 10 000 de kilometri.

Tabelul 1.2 Cât de mare e o rețea: LAN, MAN, WAN.

Distanța între noduri	Ce acoperă	Tip de rețea	Protocoale dominante
1 mm – 1 cm	pe siliciu, între procesoare	micro-rețele	magistrale interne
~1 m	spațiul personal	PAN	Bluetooth, Zigbee
10 m	o cameră	LAN	Ethernet (IEEE 802.3),
100 m	o clădire	Local Area Network	Wi-Fi (IEEE 802.11)
1 km	un campus		
10 km	un oraș	MAN	rar întâlnite azi ca entitate distinctă
100 – 1000 km	o țară, un continent	WAN	MPLS, PPP, Frame Relay, ATM
~10 000 km	planeta	Internet	stiva TCP/IP

Disciplina se numește **Rețele Locale**, deci accentul cade pe rândurile din mijloc - dar veți observa că ele nu pot fi înțelese fără ce se întâmplă la graniță.

Unde se termină un LAN. Granița dintre rețeaua locală și restul lumii este **ruterul**, numit și *gateway* - cuvânt care în engleză înseamnă exact „poartă”. Tot ce se află de partea internă a ruterului aparține rețelei locale; tot ce trece dincolo de el intră în altă rețea.

Această frontieră organizează întregul semestru: **cursurile 2–4** se ocupă de ce se întâmplă în interiorul ei, **cursurile 5–10** de ce se întâmplă la ea și dincolo de ea, iar **cursurile 8, 11 și 12** de cine încearcă să treacă prin ea fără să aibă voie.

1.4 Echipamentele și nivelul la care lucrează

Într-un magazin, echipamentele de rețea par să difere prin număr de porturi și preț. În realitate, diferența esențială este alta: **cât de adânc se uită fiecare în ceea ce trece prin el.**

1. Un **repetor** nu se uită deloc. Primește un semnal slăbit, îl amplifică și îl trimite mai departe. Nu știe și nu îl interesează ce transportă.
2. Un **switch** se uită la adresa de nivel 2, cea scrisă în placa de rețea, și decide pe ce port să trimită cadrul.
3. Un **ruter** se uită la adresa de nivel 3, adresa IP, și decide spre ce rețea să trimită pachetul.
4. Un **firewall** se uită și mai adânc - până la conținutul propriu-zis - și decide dacă are voie să treacă.

Cu cât se uită mai adânc, cu atât ia decizii mai bune - și cu atât consumă mai mult timp. Acesta este compromisul care revine în tot restul cursului.



Figura 1.2 Potrivii fiecare echipament cu rolul lui.

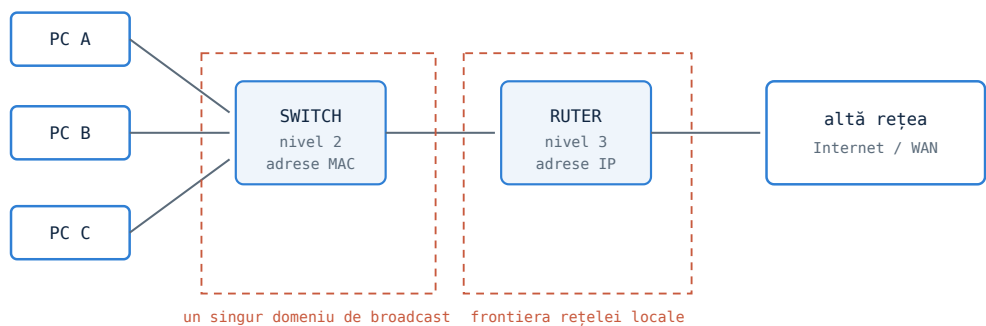


Figura 1.3 Switch-ul extinde rețeaua locală; ruterul o încheie. Toate stațiile din stânga ruterului se aud între ele fără intermediar; ca să vorbească cu ceva din dreapta, trebuie să ceară voie ruterului.

Tabelul 1.3 Echipamentele și nivelul la care lucrează.

Echipament	Nivel OSI	Decizie luată pe baza	Efect asupra domeniilor
Repetor, hub	1 - fizic	nimic; regenerează semnalul	extinde și coliziunea, și broadcast-ul
Switch (bridge)	2 - legătură de date	adresa MAC destinație	separă coliziunea, extinde broadcast-ul
Ruter	3 - rețea	adresa IP destinație	separă și coliziunea, și broadcast-ul
Firewall / IPS	3-7	anteturi și conținut	filtrează, nu doar dirijează

Ce înseamnă „domeniu”. Nu vă alarmați dacă ultimele două coloane nu vă spun încă nimic - sunt explicate pe larg în cursul 3. Deocamdată, două propoziții:

Un **domeniu de coliziune** este grupul de dispozitive care se pot deranja reciproc dacă vorbesc în același timp. Un **domeniu de broadcast** este grupul care aude un anunț general dat de oricare dintre ele. Ambele sunt cu atât mai bune cu cât sunt mai mici.



Figura 1.4 Exercițiu: câte domenii vedeți în topologia asta?.

Interfața de rețea

Termenul **interfață** desemnează un punct de comunicație cu o rețea: o placă de rețea a unui calculator, un port al unui switch, un port al unui ruter. Un calculator cu o singură placă are o singură interfață; unul cu două plăci are două. Un ruter are mai

multe și tocmai de aceea poate lega rețele diferite - practic, aceasta este întreaga lui rațiune de a exista.

În sistemele de operare, interfața este și o abstracție software. Pe Linux, plăcile Ethernet apar ca `eth0`, `eth1`, iar o interfață virtuală numită **loopback** (`lo`, adresa `127.0.0.1`) permite unui calculator să își adreseze propria stivă de protocoale ca și cum ar fi în rețea. Este primul lucru pe care îl testați când „nu merge nimic”: dacă nici loopback-ul nu răspunde, problema nu e în rețea, e în calculator.

1.5 Protocolul: de ce trebuie să vorbim aceeași limbă

DEFINIȚIA 1.1 (PROTOCOL). Un set de reguli care guvernează modul în care două entități schimbă informație. Ambele capete ale unei transmisii trebuie să folosească **același** protocol; altfel semnalul ajunge, dar mesajul nu.

Analogie Doi oameni se întâlnesc. Unul spune „早上好”. Celălalt aude perfect fiecare sunet - semnalul a ajuns intact - dar nu înțelege nimic, pentru că nu împart aceeași convenție despre ce înseamnă sunetele. Aceasta e diferența dintre **a auzi** și **a înțelege**, iar în rețele e diferența dintre nivelul fizic și tot ce vine deasupra lui.

Un protocol de rețea stabilește, tipic, patru lucruri:

1. **Sintaxa** - cum arată mesajul: ce câmpuri are, în ce ordine, pe câți biți fiecare.
2. **Semantica** - ce înseamnă fiecare câmp și ce trebuie făcut cu el.
3. **Sincronizarea** - cine vorbește când, cât așteaptă, ce face dacă nu primește răspuns.
4. **Tratarea erorilor** - ce se întâmplă când ceva se strică.

Veți vedea aceste patru componente la fiecare protocol studiat în semestru. Când un protocol nou vi se pare complicat, întrebați-vă care dintre cele patru vă scapă - de obicei e a treia.

1.6 De ce comunicăm pe niveluri

Să presupunem că scriem singuri un program de e-mail, fără să existe nimic înainte. Ar trebui să ne ocupăm de tensiunea de pe fir, de retransmiterea pachetelor pierdute, de ordinea lor, de formatul mesajului și de multe altele - toate în același cod. Ar fi imposibil de întreținut. Și, mai rău, ar trebui rescris de la zero de fiecare dată când apare un tip nou de cablu.

Soluția, veche de peste patruzeci de ani, este **stratificarea**: împărțim problema în niveluri, iar fiecare nivel oferă un serviciu nivelului de deasupra și folosește serviciile

nivelului de dedesubt, fără să știe cum sunt ele implementate.

Analogie Trimiteți un colet. Îl împachetați și scrieți adresa - acesta e nivelul vostru. Îl duceți la poștă, care îl pune într-un sac cu alte colete pentru același oraș - alt nivel. Sacul urcă într-un camion - alt nivel. Șoferul nu știe și nu îl interesează ce e în colet; voi nu știți și nu vă interesează pe ce autostradă merge camionul. Fiecare nivel are propriul lui „ambalaj” și propria lui adresă, iar la destinație totul se despachetează în ordine inversă.

Ce câștigăm.

- Un nivel poate fi înlocuit fără să afecteze restul: trecerea de la cupru la fibră nu schimbă nimic la aplicație
- Problemele se localizează: „nu merge” devine „nu merge la nivelul 3”
- Producători diferiți pot implementa niveluri diferite
- Se poate învăța pe rând, ceea ce face exact acest curs

Ce pierdem

- Fiecare nivel adaugă antet - deci trafic care nu e informație utilă
- Informația utilă unui nivel nu îi este accesibilă altuia; NAT, din cursul 9, este considerat o încălcare tocmai din acest motiv
- Trecerea prin niveluri costă timp de procesare

Modelul OSI: șapte niveluri

Apăsați pe fiecare nivel din stânga. Nu încercați să memorați tabelul - reveniți la el ori de câte ori aveți nevoie. Ce trebuie să rămână este ideea: **fiecare nivel are propriul tip de adresă și propria unitate de date.**



Figura 1.5 Explorați stiva OSI, nivel cu nivel.

Stiva TCP/IP: patru niveluri, dar reale

Modelul OSI este un model de **referință**: îl folosim ca vocabular comun. Stiva TCP/IP este ce rulează efectiv în Internet. Ele nu se contrazic - TCP/IP comasează nivelurile pe care OSI le separă, pentru că în practică nimeni nu implementează separat sesiunea și prezentarea.

Tabelul 1.4 Stiva TCP/IP: patru niveluri, dar reale.

OSI	TCP/IP	Ce rezolvă	Exemple
7. Aplicație 6. Prezentare 5. Sesiune	Aplicație	servicii pentru utilizator, reprezentarea datelor, controlul dialogului	HTTP, DNS, SSH, SMTP
4. Transport	Transport	controlul fluxului, siguranța transmisiei, multiplexarea pe porturi	TCP, UDP
3. Rețea	Internet	adresare globală și alegerea căii spre destinație	IPv4, IPv6, ICMP
2. Legătură de date 1. Fizic	Acces la mediu	accesul la mediul partajat, încadrarea, transmisia binară	Ethernet, 802.11, PPP

Încapsularea: cum coboară un mesaj prin stivă

Aceasta este cea mai importantă figură din tot cursul. Urmăriți-o pas cu pas.

**Figura 1.6** Încapsularea, pas cu pas.

De reținut, chiar dacă uitați restul. Nivelul 2 lucrează cu **adrese MAC** și **cadre**. Nivelul 3 cu **adrese IP** și **pachete**. Nivelul 4 cu **porturi** și **segmente**.

Când depanați o rețea, prima întrebare utilă este întotdeauna: **la ce nivel se rupe?** Restul semestrului vă dă uneltele cu care răspundeți la ea.

1.7 Nivelul fizic și cele patru sarcini ale lui

Nivelul fizic are o misiune enunțabilă într-o propoziție: **să transforme un bit într-un semnal și înapoi**. În practică, misiunea aceasta se desface în patru sarcini distincte.

1. **Transformarea bit → semnal.** Alegerea nivelurilor de tensiune, a lungimii de undă sau a frecvenței care reprezintă valorile logice.
2. **Sincronizarea biților.** Receptorul trebuie să știe unde se termină un bit și unde începe următorul. Pare un detaliu; este, de fapt, motivul pentru care există jumătate din codările pe care le vom vedea imediat.
3. **Controlul vitezei.** Negocierea ratei de transmisie pe care ambele capete o pot susține - motivul pentru care o placă gigabit și un switch de 100 Mbps ajung, totuși, la o înțelegere.

4. **Multiplexarea.** Punerea mai multor fluxuri logice pe același mediu fizic. Subiect tratat pe larg în cursul următor.

De ce sincronizarea e o problemă reală. Imaginați-vă că vă dictez o serie de cifre, dar fără nicio pauză între ele și fără să vă spun cât durează una. Dacă vă dictez zece de „zero” la rând, cu voce constantă, câte ați notat? Nu aveți cum să știți - poate erau opt, poate douăsprezece.

Exact aceasta e problema receptorului: dacă tensiunea rămâne constantă mult timp, el nu are cum să numere biții. Soluția e fie un ceas separat (scump: încă un fir), fie o codare care **garantează schimbări dese ale semnalului**, din care ceasul se poate reconstitui. Aproape toate codările moderne aleg a doua variantă.

Patru combinații de date și semnal

Cuvintele „digital” și „analogic” se aplică **separat** datelor și semnalului. Rezultă patru situații posibile - toate cu utilizare reală.

Tabelul 1.5 Patru combinații de date și semnal.

Datele sunt	Semnalul este	Operația se numește	Exemplu din viața reală
digitale	digital	codare de linie	Ethernet pe cablu torsadat
digitale	analogic	modulație	Wi-Fi, DSL, modem pe linie telefonică
analogice	digital	eșantionare și cuantizare	vocea digitalizată, VoIP
analogice	analogic	modulație analogică	radio AM și FM, telefonie clasică

De unde vine cuvântul „modem”. Un modem este un **MO**dulator/**DE**modulator: transformă datele digitale ale calculatorului în semnal analogic potrivit mediului și face operația inversă la recepție. Cuvântul a supraviețuit tehnologiei care l-a produs - cutia de la furnizorul de Internet tot modem se numește, deși înăuntru nu mai seamănă deloc cu strămoșul ei de pe linia telefonică.

1.8 Codări de linie: cum arată un bit pe fir

O **codare de linie** este regula prin care un șir de biți devine o formă de undă. Este cea mai concretă parte a acestui curs: aici se vede, la propriu, cum arată informația pe cablu.

O codare bună rezolvă simultan trei probleme:

1. **Distincția.** Receptorul trebuie să poată deosebi fără ambiguitate un 1 de un 0, chiar și cu puțin zgomot pe fir.

2. **Sincronizarea.** Receptorul trebuie să își poată recupera ceasul din chiar semnalul primit.
3. **Absența componentei continue.** Media semnalului trebuie să fie zero, pentru că transformatoarele de izolare din plăcile de rețea nu transmit componenta continuă. Un semnal care stă mult timp „sus” pur și simplu nu trece prin ele.

Codările se clasifică după numărul de niveluri de tensiune folosite: **unipolare** (un nivel plus absența semnalului), **polare** (două niveluri, pozitiv și negativ) și **bipolare** sau multinivel (trei sau mai multe).



Figura 1.7 Generator de forme de undă - încercați fiecare codare.

Ce trebuie să observați, pe rând

Nu treceți mai departe fără să faceți efectiv experimentele de mai jos. Sunt cinci minute și înlocuiesc o oră de citit.

Tabelul 1.6 Ce trebuie să observați, pe rând.

Codare	Regula	Experiment de făcut	Ce veți observa
NRZ-L Non-Return to Zero Level	nivelul <i>este</i> valoarea bitului	scrieți 00000000	o linie perfect dreaptă - receptorul nu mai are ce număra
NRZ-I NRZ Inverted	1 = schimbă nivelul, 0 = păstrează-l	scrieți 11111111, apoi 00000000	unii produc tranziții la fiecare bit, zerourile tot niciuna
Manchester	tranziție obligatorie la mijlocul fiecărui bit	orice șir	semnalul se schimbă de două ori mai des decât rata de bit
Manchester diferențial	tranziția de la mijloc rămâne; informația e în prezența unei tranziții la început	comparați cu Manchester simplu	rezistă la inversarea firelor
MLT-3	trei niveluri parcurse ciclic, avansează doar la 1	scrieți 11111111	ciclul $0 \rightarrow +1 \rightarrow 0 \rightarrow -1$; frecvența scade de patru ori
PAM-5	doi biți per simbol, pe niveluri de amplitudine	orice șir par	jumătate din numărul de tranziții pentru aceiași biți

Exemplul 1.2. Cu NRZ, un șir alternat 101010... la 100 Mbps produce o schimbare de nivel la fiecare bit, adică o frecvență fundamentală de 50 MHz. Cablul Cat 5 este certificat până la 100 MHz, dar la 50 MHz emite deja destulă radiație cât să deranjeze perechile vecine și echipamentele din jur.

MLT-3 avansează în ciclul de patru stări doar la biții de 1, deci îi trebuie **patru** biți ca să revină la punctul de plecare. Frecvența fundamentală scade la 25 MHz. Aceasta este, esențial, ideea care a permis ca 100 Mbps să circule pe cablul telefonic ieftin deja tras prin clădiri - și de aici a plecat tot restul.

Codarea în bloc: 4B/5B

Problema șirurilor lungi de zerouri se poate rezolva elegant nu la nivelul semnalului, ci **înaintea** lui: rescriem datele astfel încât să nu mai conțină secvențe problematice.

Codarea **4B/5B** înlocuiește fiecare grup de 4 biți de date cu un grup de 5 biți ales dintr-un tabel construit special: niciun cuvânt de cod nu are mai mult de trei zerouri consecutive. Receptorul face traducerea inversă și recuperează datele originale.

Tabelul 1.7 Codarea în bloc: 4B/5B.

date (4b)	cod (5b)	date (4b)	cod (5b)	simbol de control	cod (5b)
0000	11110	1000	10010	Q - linie liniștită	00000
0001	01001	1001	10011	I - inactiv	11111
0010	10100	1010	10110	J - început flux	11000
0011	10101	1011	10111	K - început flux	10001
0100	01010	1100	11010	T - sfârșit flux	01101
0101	01011	1101	11011	R - resetare	00111
0110	01110	1110	11100	S - set	11001
0111	01111	1111	11101	H - eroare	00100

Costul redundanței, și partea bună a lui. 4B/5B introduce 25 % supraîncărcare: ca să transmiteți 100 Mbps de date utile, linia trebuie să ducă **125 Mbaud**. Pare risipă.

Dar din cele 32 de cuvinte posibile de 5 biți, doar 16 sunt folosite pentru date. Celelalte 16 nu se irosesc - devin **simboluri de control** care marchează începutul cadru-lui, sfârșitul lui sau starea de repaus a liniei. Cu alte cuvinte, redundanța cumpărată pentru sincronizare rezolvă gratuit și problema delimitării mesajelor.

Cum se combină în tehnologiile reale

Niciun standard nu folosește o singură tehnică. Se combină în lanț, fiecare rezolvând altă problemă:

Tabelul 1.8 Cum se combină în tehnologiile reale.

Standard	Mediu	Lanțul de codare	De ce așa
100BASE-TX	cupru, Cat 5	date → 4B/5B → MLT-3	4B/5B dă sincronizarea, MLT-3 coboară frecvența
100BASE-FX	fibră multimod	date → 4B/5B → NRZ-I	pe fibră frecvența nu e o problemă, deci NRZ-I ajunge
1000BASE-SX / LX	fibră	date → 8B/10B → NRZ	8B/10B echilibrează și componenta continuă
1000BASE-T	cupru, Cat 5e	date → PAM-5, pe 4 perechi simultan	250 Mbps pe fiecare pereche, în ambele sensuri

1.9 Modulația: date digitale peste semnal analogic

Dacă mediul nu suportă tranziții bruște de tensiune - o linie telefonică, aerul, un canal radio - biții nu pot fi transmiși direct. Trebuie transportați modificând un semnal continuu, numit **purtătoare**.

Ce e o purtătoare. Imaginați-vă un fluierat constant, la aceeași înălțime și aceeași intensitate. În sine nu transmite nimic - dar dacă îl faceți mai tare și mai încet după o convenție dinainte stabilită, ați transmis informație **peste** fluierat. Fluieratul este purtătoarea; modificarea lui este modulația.

O undă sinusoidală are exact trei parametri pe care îi putem modifica - și, prin urmare, există exact trei modulații fundamentale.

Tabelul 1.9 Modulația: date digitale peste semnal analogic.

Parametrul modificat	Denumire	Cum arată	Unde apare
amplitudinea (înălțimea undei)	ASK - Amplitude Shift Keying	1 = undă puternică, 0 = undă slabă sau absentă	fibră optică, RFID
frecvența (cât de des oscilează)	FSK - Frequency Shift Keying	1 și 0 sunt două frecvențe distincte	modemuri vechi, Bluetooth clasic
faza (decalajul undei)	PSK - Phase Shift Keying	1 și 0 diferă prin poziția în ciclu	Wi-Fi, satelit, DVB

Rata de bit nu este rata de simbol

DEFINIȚIA 1.2 (BIT RATE ȘI BAUD RATE). Rata de bit (bps) este numărul de biți transmiși într-o secundă.

Rata de simbol (baud) este numărul de schimbări de stare ale semnalului într-o secundă.

Întotdeauna $\text{baud} \leq \text{bps}$, iar raportul dintre ele este exact numărul de biți codificați într-un simbol.

Distincția asta este cheia întregii evoluții a comunicațiilor. Un canal are o limită fizică pentru rata de simbol, impusă de lățimea lui de bandă - nu o puteți depăși oricât ați plăti. Singura cale de a transmite mai mulți biți pe secundă este să **codificați mai mulți biți în fiecare simbol**.

Aici intervin schemele combinate. **QAM** (Quadrature Amplitude Modulation) variază simultan amplitudinea și faza, obținând mai multe puncte distincte în plan - fiecare punct fiind un simbol.

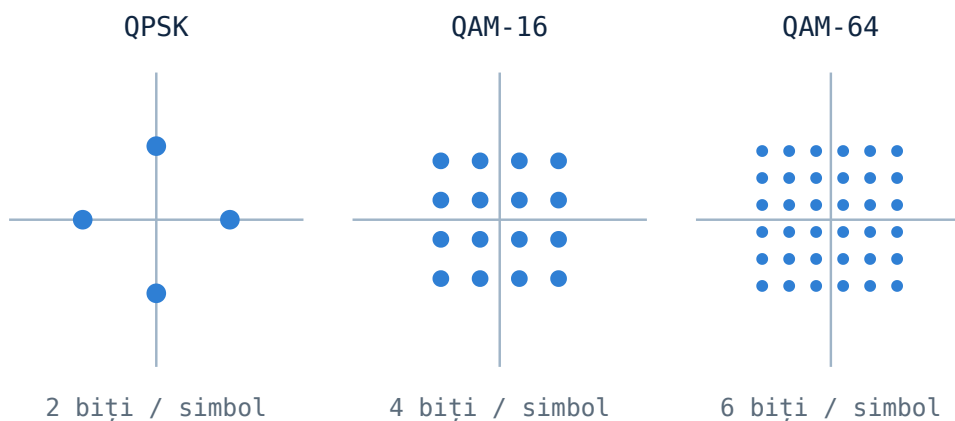


Figura 1.8 Diagrame de constelații. Fiecare punct este o combinație distinctă de amplitudine și fază, deci un simbol. Cu cât punctele sunt mai multe și mai apropiate, cu atât transmitem mai mulți biți per simbol - și cu atât mai puțin zgomot este nevoie ca receptorul să confunde două puncte vecine.

Compromisul care se repetă peste tot. Mai multe puncte de constelație = mai mulți biți per simbol = debit mai mare. Dar punctele devin mai apropiate, deci e nevoie de un semnal mai curat pentru a le distinge.

De aceea, când vă îndepărtați de routerul Wi-Fi, viteza scade: echipamentul **coboară automat** pe o modulație cu mai puține puncte, mai robustă. Nu s-a stricat nimic - se negociază, în timp real, între debit și fiabilitate.



Figura 1.9 Calculator: de la rata de simbol la debit.

Exercițiu rezolvat (clasic de examen) 1.3. Se consideră o linie cu o capacitate de 2400 baud. Câți biți de date pot fi trimiși pe secundă dacă se folosește QAM-16 pentru modulare?

Rezolvare. QAM-16 are 16 puncte de constelație. Numărul de biți codificați într-un simbol este $\log_2 16 = 4$.

$$4 \text{ biți/simbol} \times 2400 \text{ simboluri/s} = \mathbf{9600 \text{ bps}}$$

Aceeași linie, cu QAM-64, ar duce $6 \times 2400 = 14\,400$ bps - fără nicio modificare fizică, doar schimbând schema de modulație. Verificați ambele în calculatorul de mai sus.

Exercițiu propus 1.4. Un canal Wi-Fi transmite 1000 de simboluri pe microsecundă folosind QAM-256, pe un singur flux spațial. Ce debit brut rezultă? Dar dacă se folosesc patru fluxuri MIMO simultan?

Rezolvare. $\log_2 256 = 8$ biți/simbol. $1000 \text{ simboluri}/\mu\text{s} = 10^9 \text{ simboluri/s}$. Rezultă **8 Gbps** brut pe flux; cu patru fluxuri, 32 Gbps.

Debitul **util** este semnificativ mai mic - codarea de corecție a erorilor, anteturile și intervalele de gardă consumă o parte importantă. Este observația care revine la fiecare tehnologie din acest curs: cifra de pe cutie nu este cifra pe care o vedeți.

1.10 Greșeli frecvente

Greșeli frecvente.

„100 Mbps înseamnă 100 MB pe secundă”

Factor de 8 între bit și octet, plus diferența dintre puterile lui 10 și ale lui 2. O legătură de 100 Mbps transferă, în cel mai bun caz, aproximativ 11,9 MB pe secundă.

Verificați întotdeauna dacă litera este mică (bit) sau mare (octet).

„Un switch e un hub mai bun”

Nu sunt din aceeași familie. Hub-ul repetă orbește semnalul pe toate porturile; switch-ul citește adresa și trimite doar unde trebuie. Diferența e de nivel, nu de calitate.

Rețineți nivelul: hub = 1, switch = 2, ruter = 3.

„OSI și TCP/IP sunt două lucruri concurente”

Nu sunt. OSI e un model de referință, folosit ca vocabular; TCP/IP e ce rulează efectiv. Se suprapun, nu se contrazic.

Când cineva spune „nivelul 3”, vorbește în OSI, chiar dacă rețeaua rulează TCP/IP.

„Modulația și codarea de linie sunt același lucru”

Codarea de linie produce un semnal digital (trepte de tensiune). Modulația modifică un semnal analogic continuu.

Întrebați-vă ce fel de semnal iese: trepte → codare; undă → modulație.

„Cu cât modulația e mai densă, cu atât e mai bine”

Doar dacă semnalul e suficient de curat. QAM-1024 pe o legătură zgomotoasă produce mai multe erori și retransmisii decât QPSK - deci, în final, un debit mai mic.

Debitul real depinde de raportul semnal-zgomot, nu doar de schema aleasă.

1.11 Rezumat și glosar

Cursul acesta a construit vocabularul și a coborât până la fir. Trei idei rezistă:

1. Comunicația e organizată pe **niveluri**, fiecare cu adresa și unitatea lui de date. Când ceva nu merge, prima întrebare este la ce nivel.
2. Nivelul fizic transformă biți în semnal, iar problema lui grea nu e distincția între 1 și 0, ci **sincronizarea**.
3. Debitul se obține fie transmițând mai multe simboluri pe secundă - limitat fizic - fie punând mai mulți biți în fiecare simbol. A doua cale a produs toată evoluția din ultimii treizeci de ani.



Figura 1.10 Puneți în ordine pașii încapsulării.

Exerciții

1. Care este unitatea de date de la nivelul 3?
 - a. Cadru
 - b. Pachet
 - c. Segment
 - d. Bit
2. De ce este util ca stiva sa fie impartita pe niveluri?
 - a. Pentru ca merge mai repede
 - b. Pentru ca un nivel poate fi inlocuit fara sa afecteze restul
 - c. Pentru ca reduce dimensiunea pachetelor
 - d. Pentru ca elimina erorile
3. De ce codarea Manchester nu isi pierde niciodata sincronizarea?
 - a. Pentru ca foloseste trei niveluri de tensiune
 - b. Pentru ca fiecare bit contine obligatoriu o tranzitie la mijloc
 - c. Pentru ca transmite ceasul pe un fir separat
 - d. Pentru ca inlocuieste grupurile de 4 biti cu grupuri de 5
4. Ce rezolva codarea 4B/5B in lantul 100BASE-TX?
 - a. Cripeaza datele pe fir
 - b. Elimina secventele lungi de zerouri, care ar face receptorul sa piarda ceasul
 - c. Reduce numarul de fire de la patru la doua
 - d. Corecteaza erorile de transmisie
5. Cate valori distincte pot fi reprezentate pe 6 biti?
 - a. 12
 - b. 36
 - c. 64
 - d. 128

6. Care echipament separa domeniile de coliziune, dar extinde domeniile de broadcast?
 - a. Hub-ul
 - b. Switch-ul
 - c. Ruterul
 - d. Repetorul
7. O linie de 4800 baud folosește QPSK. Ce debit binar obține?
 - a. 4800 bps
 - b. 9600 bps
 - c. 19200 bps
 - d. 2400 bps
8. În stiva TCP/IP, carui nivel OSI îi corespunde nivelul Internet?
 - a. Nivelului 2, legătura de date
 - b. Nivelului 3, rețea
 - c. Nivelului 4, transport
 - d. Nivelurilor 5-7
9. Ce adresă folosește nivelul transport ca să livreze datele programului potrivit?
 - a. Adresa MAC
 - b. Adresa IP
 - c. Numărul de port
 - d. Numele de domeniu
10. Ce înseamnă că un semnal nu are componentă continuă?
 - a. Ca nu se transmite niciodată
 - b. Ca media lui în timp este zero, deci poate trece prin transformatoarele de izolare
 - c. Ca frecvența este constantă
 - d. Ca folosește doar două niveluri
11. Un octet are valoarea binară 11000000. Cât este în zecimal?
 - a. 96
 - b. 128
 - c. 192
 - d. 224

12. De ce se folosește hexazecimalul pentru adresele MAC și IPv6?
- Pentru ca ocupa mai puțină memorie
 - Pentru ca o cifră hexa reprezintă exact 4 biți, deci conversia este imediată
 - Pentru ca este mai sigur
 - Pentru ca zecimalul nu suportă valori mari

Note bibliografice

Cursul următor rămâne la nivelul fizic, dar schimbă întrebarea: nu *cum* arată semnalul, ci *prin ce* îl trimitem. Cupru, fibră sau aer - ce poate duce fiecare, cât de departe, ce se strică pe drum și cum punem mai multe conversații pe un singur fir.

La **laboratorul 1** construiți prima topologie în Packet Tracer și urmăriți un pachet real de la un capăt la altul, cu toate anteturile pe care le-ați văzut aici.

- IEEE 802.3 - standardul Ethernet, capitolele despre nivelul fizic (PCS, PMA)
- Andrew S. Tanenbaum, *Computer Networks*, capitolul 2 - nivelul fizic
- James Kurose, Keith Ross, *Computer Networking: A Top-Down Approach*, capitolul 1
- RFC 1122 - cerințele stivei TCP/IP pentru stații
- Pentru cine vrea partea matematică: teorema lui Nyquist și formula lui Shannon pentru capacitatea unui canal cu zgomot

Extras din volumul „Rețele Locale - Note de curs și lucrări de laborator”. Varianta online a acestui capitol conține figurile în formă interactivă.