
Protocolul IP

Adresele MAC sunt unice pe planetă și totuși inutilizabile la scară mare, pentru că nu spun nimic despre locul unde se află echipamentul. Nivelul rețea rezolvă problema exact așa cum a rezolvat-o poșta acum două secole: printr-o adresă ierarhică, din care se poate deduce direcția fără a cunoaște destinația finală.

Din capitolul precedent.

- Adresa MAC are 48 de biți, e unică pe planetă, dar e **plată** (fără ierarhie) și **locală** (vizibilă doar în domeniul de broadcast).
- Un switch comută după adresa MAC; un **ruter** lucrează la nivelul 3 și separă domeniile de broadcast.
- Un **VLAN** este un domeniu de broadcast. Comunicația între VLAN-uri cere obligatoriu un dispozitiv de nivel 3.
- Conversia binar ↔ zecimal din cursul 1 devine, de azi, unealta de lucru zilnică.

OBIECTIVELE CAPITOLULUI

- Să explicați de ce adresarea ierarhică face rutarea posibilă la scara Internetului
- Să citiți antetul IPv4 și să spuneți la ce folosește fiecare câmp important
- Să calculați adresa de rețea și adresa de broadcast prin operații pe biți
- Să subnetați un bloc în subrețele egale și, apoi, optim cu VLSM
- Să distingeți adresele publice de cele private și să recunoașteți blocurile rezervate
- Să descrieți un schimb ARP complet și să explicați ce e proxy ARP
- Să urmăriți cele patru mesaje DHCP și să spuneți când e nevoie de un relay

5.1 De ce mai avem nevoie de un set de adrese

Adresa MAC identifică unic o interfață în toată lumea. Pare exact ce ne trebuie - și totuși nu este suficientă. Motivul e o singură proprietate: schema este **plată**.

Ce înseamnă „plată” și de ce e o problemă. Din adresa 00:FC:42:3E:34:99 nu se poate deduce absolut nimic despre locul unde se află acea placă de rețea. Poate fi în camera de alături sau în Australia. Nu există nicio regulă care să spună „adresele care încep cu X sunt în Europa”.

Consecința: un ruter care ar dirija trafic după adrese MAC ar avea nevoie de **o intrare separată pentru fiecare interfață din lume** - miliarde de rânduri, fără nicio posibilitate de grupare. Tabela nu ar încăpea nicăieri, nu ar putea fi actualizată de nimeni și nu ar putea fi căutată suficient de repede.

Analogie: codul poștal Un poștaș din Cluj primește un plic pentru „Splaiul Independenței 313, sector 6, București”. Nu are nevoie să știe unde e acea stradă. Îi ajunge prima cifră a codului poștal ca să pună plicul în sacul pentru București. Cine ia sacul mai departe folosește a doua cifră, și așa mai departe.

Fiecare nivel de decizie folosește **doar prefixul de care are nevoie**. Exact acesta este mecanismul care face rutarea IP posibilă la scara Internetului.

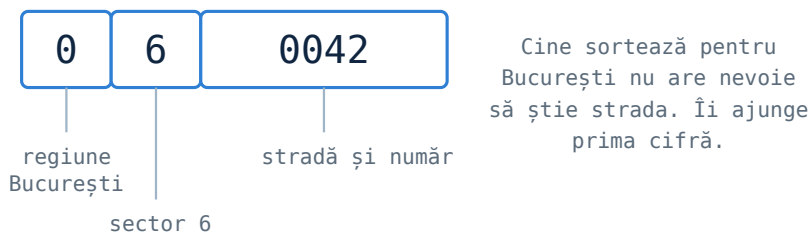


Figura 5.1 Codul poștal este o adresă ierarhică. Aceeași idee, aplicată la 32 de biți, dă adresa IP.

5.2 Cele trei funcții ale nivelului rețea

Tabelul 5.1 Cele trei funcții ale nivelului rețea.

Funcție	Ce înseamnă concret
Adresare globală	fiecare dispozitiv este identificat unic printr-o adresă IP, organizată ierarhic, astfel încât adresele apropiate să poată fi grupate
Comunicație end-to-end	protocolul este de tip best-effort : nu stabilește conexiuni, nu garantează livrarea, nu garantează ordinea. Garanțiile sunt treaba nivelului 4
Rutare (dirijare)	echipamente intermediare aleg calea fiecărui pachet pe baza adresei destinație

Ce înseamnă „best-effort”. IP promite un singur lucru: va **încerca** să livreze pachetul. Nu promite că va reuși, nu promite ordinea și nu vă anunță dacă a eșuat. Un pachet poate fi aruncat de orice ruter aglomerat de pe traseu, fără nicio ceremonie.

Pare o slăbiciune. Este, de fapt, decizia de proiectare care a făcut Internetul posibil: rămânând simplu și fără stare, un ruter poate procesa milioane de pachete pe secundă fără să țină minte nimic despre ele. Complexitatea - retransmisii, ordonare, confirmări - a fost mutată la capete, în TCP.

Tabelul 5.2 Cele trei funcții ale nivelului rețea.

Protocol	Rol
IPv4	protocolul de bază; subiectul acestui curs
IPv6	succesorul, cu adrese de 128 de biți; cursul 9
ICMP	mesaje de control și eroare: ping, traceroute, „destinație inaccesibilă”
IGMP	gestionarea grupurilor multicast

5.3 Antetul IPv4

IPv4 este definit în RFC 791, publicat în 1981 - și, cu excepții minore, funcționează neschimbat de atunci. Unitatea de date rezultată din încapsularea IP se numește **pachet**.



Figura 5.2 Antetul IPv4, câmp cu câmp.

Cum funcționează traceroute - un truc elegant. Trimiteți un pachet cu **TTL = 1**. Primul ruter îl decrementează la 0, îl aruncă și vă trimite un mesaj ICMP *time exceeded* - în care își pune propria adresă. Ați aflat cine e primul salt.

Trimiteți apoi cu TTL = 2, apoi 3, apoi 4... Fiecare pachet moare cu un salt mai departe și fiecare ruter se prezintă singur. Așa se construiește lista completă a traseului, folosind un câmp care fusese gândit ca protecție împotriva buclelor.

5.4 Adresa, masca și cele două operații

O adresă IPv4 are 32 de biți, scriși de obicei în format zecimal cu punct - patru numere între 0 și 255, câte unul pentru fiecare octet.

141 . 85 . 241 . 139

10001101.01010101.11110001.10001011

Adresa se împarte în două părți: **partea de rețea** și **partea de host**. Dispozitivele care au aceeași parte de rețea se află în aceeași rețea și pot comunica **direct**, fără ruter.

Grați la dintre cele două părți nu se poate deduce din adresă - trebuie spusă separat, prin **masca de rețea**: un șir continuu de 1 urmat de un șir continuu de 0.

Operațiile ȘI și SAU pe biți, în treizeci de secunde. Nu au nimic misterios. Se aplică **bit cu bit**, pe poziții corespunzătoare:

De reținut, practic: **ȘI cu 1 păstrează bitul, ȘI cu 0 îl șterge. SAU cu 1 aprinde bitul, SAU cu 0 îl lasă în pace.** Atât vă trebuie pentru tot restul cursului.

DEFINIȚIA 5.1 (CELE DOUĂ OPERAȚII FUNDAMENTALE). **Adresa de rețea** = adresa IP ȘI masca. Masca are 1 pe partea de rețea, deci acei biți se păstrează, și 0 pe partea de host, deci acei biți se șterg. Rezultatul: toți biții de host puși pe 0.

Adresa de broadcast = adresa IP SAU inversa măștii. Inversa are 0 pe partea de rețea (nu schimbă nimic) și 1 pe partea de host (aprinde tot). Rezultatul: toți biții de host puși pe 1.

Niciuna dintre ele nu poate fi atribuită unei stații.

Notația CIDR

Notația zecimală a măștii - 255.255.255.0 - e greoaie. S-a introdus prescurtarea **CIDR**: se scrie pur și simplu **numărul de biți de 1** din mască, precedat de bară.

Tabelul 5.3 Notația CIDR.

Prefix	Mască zecimală	Biți de host	Adrese totale	Adrese utilizabile
/24	255.255.255.0	8	256	254
/25	255.255.255.128	7	128	126
/26	255.255.255.192	6	64	62
/27	255.255.255.224	5	32	30
/28	255.255.255.240	4	16	14
/29	255.255.255.248	3	8	6
/30	255.255.255.252	2	4	2

Tabelul acesta merită învățat - nu pe de rost, ci prin logică. Fiecare rând înjumătățește numărul de adrese față de cel de deasupra, iar „utilizabile” e mereu „totale minus 2” (rețea și broadcast).



Figura 5.3 Calculator de subrețea.

Experimentele de făcut aici.

1. Schimbați prefixul de la /26 la /24 și priviți cum biții trec dintr-o culoare în alta. Mutarea graniței cu **un singur bit** dublează numărul de stații și înjumătățește numărul de subrețele.
2. Puneți adresa 192.168.10.64 cu prefix /26. Observați că adresa introdusă este chiar adresa de rețea - deci nu poate fi dată unei stații.
3. Încercați /31 și /32. Sunt cazuri speciale: /31 se folosește pe legături punct-la-punct între rutere, /32 pentru o singură adresă (o interfață de loopback).

Exercițiu rezolvat - verificarea unei configurații 5.1. Trei stații sunt conectate la același switch: A cu 192.168.17.64/26, B cu 192.168.17.35/26, C cu 192.168.17.31/27. Există probleme?

Rezolvare. Da, **trei** probleme distincte:

- A are configurată o **adresă de rețea**. 64 în binar este 01000000, iar cu /26 ultimii 6 biți sunt de host și sunt toți zero. O adresă de rețea nu poate fi atribuită unei interfețe.
- C are configurată o **adresă de broadcast**. Cu /27, rețeaua lui este 192.168.17.0/27, iar broadcast-ul acesteia este exact .31.
- C mai are și o **mască diferită** de a celorlalte. Chiar și cu adrese corecte, A și B s-ar considera în rețeaua 192.168.17.0/26, iar C în 192.168.17.0/27 - comunicația ar funcționa asimetric și derutant.

Verificați fiecare caz în calculatorul de mai sus. Introduceți adresa și prefixul și comparați cu ce afișează.

5.5 Clase, adrese private, adrese rezervate

Istoric, adresele erau împărțite în cinci clase identificate după primii biți, fiecare cu masca ei implicită și fixă.

Tabelul 5.4 Clase, adrese private, adrese rezervate.

Clasă	Primii biți	Interval	Mască implicită	Statut azi
A	0...	0.0.0.0 – 127.255.255.255	/8	istoric
B	10...	128.0.0.0 – 191.255.255.255	/16	istoric
C	110...	192.0.0.0 – 223.255.255.255	/24	istoric
D	1110...	224.0.0.0 – 239.255.255.255	-	multicast, în uz
E	1111...	240.0.0.0 – 255.255.255.255	-	rezervat, experimental

De ce s-a renunțat la clase. Sistemul era rigid până la absurd. O organizație care avea nevoie de 300 de adrese primea fie o clasă C cu 254 - insuficient - fie o clasă B cu 65 534, din care irosea 65 000.

Înlocuitorul se numește **CIDR** (Classless Inter-Domain Routing) și face un lucru simplu: masca poate fi orice, nu doar /8, /16 sau /24. Pentru 300 de adrese primiți un /23, cu 510 adrese utilizabile.

Clasele au dispărut din echipamente, dar au rămas în vocabular. Când cineva spune „o rețea de clasă C”, înțelegeți „un /24” - și nu vă mirați.

Adrese private

RFC 1918 a alocat trei spații pentru uz privat. Ele **nu pot fi rutate în Internet**, deci pot fi refolosite de oricine - și sunt, în orice locuință și în orice firmă din lume. Ieșirea în Internet se face prin translație (NAT), subiectul cursului 9.

Tabelul 5.5 Adrese private.

Bloc privat	Dimensiune	Unde apare de obicei
10.0.0.0/8	16,7 milioane de adrese	rețele mari de întreprindere
172.16.0.0/12	1 milion de adrese	rețele medii
192.168.0.0/16	65 536 de adrese	rețele mici, echipamente de acasă

Adrese cu semnificație specială

Tabelul 5.6 Adrese cu semnificație specială.

Bloc	Ce înseamnă
127.0.0.0/8	Loopback. 127.0.0.1 se referă la calculatorul însuși. Un ping aici testează stiva locală, nu rețeaua.
169.254.0.0/16	APIPA. Adresa pe care și-o dă singură o stație când serverul DHCP nu răspunde.
0.0.0.0	„această rețea” sau „orice rețea”, după context. Ca destinație într-o rută, înseamnă ruta implicită.
255.255.255.255	Broadcast limitat - nu trece niciodată de un ruter.
224.0.0.0/4	Multicast. 224.0.0.5 = toate ruterele OSPF, 224.0.0.6 = DR și BDR.

Un simptom care spune totul. Dacă o stație are o adresă din 169.254.x.x, nu are rost să căutați problema în altă parte: **serverul DHCP nu a răspuns**. Fie nu există, fie cablul e în alt VLAN, fie relay-ul nu e configurat, fie rezervorul de adrese s-a epuizat.

5.6 Subnetarea

DEFINIȚIA 5.2 (SUBNETARE). Împărțirea unui bloc de adrese în blocuri mai mici, care să acopere un set de cerințe. Se face „împrumutând” biți din partea de host și adăugându-i la partea de rețea.

Într-o rețea modernă, un „subnet” nu e cu nimic diferit de o rețea; termenul indică doar că face parte din spațiul unei rețele mai mari.

De ce am vrea să împărțim. Ați primit blocul 192.168.10.0/24 - 254 de adrese utilizabile. Aveți trei departamente. Ați putea pune toate calculatoarele într-o singură rețea plată, dar atunci reveniți la toate problemele din cursul 4: un singur domeniu de broadcast, nicio separare, nicio politică diferențiată.

Subnetarea vă dă trei rețele separate, fiecare cu gateway-ul ei, între care traficul trece controlat, printr-un ruter.

Varianta simplă: subrețele egale

Se împrumută atâția biți câți sunt necesari pentru numărul dorit de subrețele. Pentru trei subrețele avem nevoie de 2 biți, pentru că $2^2 = 4 \geq 3$.

Exemplu: 192.168.10.0/24 pentru trei rețele cu 60, 30 și 15 stații.
 rețeaua inițială 11000000.10101000.00001010.00000000 /24
 subrețeaua 1 11000000.10101000.00001010.00000000 /26 → 192.168.10.0
 subrețeaua 2 11000000.10101000.00001010.01000000 /26 → 192.168.10.64
 subrețeaua 3 11000000.10101000.00001010.10000000 /26 → 192.168.10.128
 (subrețeaua 4) 11000000.10101000.00001010.11000000 /26 → 192.168.10.192

Fiecare /26 oferă 62 de adrese utilizabile, deci toate trei cerințele sunt acoperite. Risipa e însă considerabilă:

Tabelul 5.7 Varianta simplă: subrețele egale.

Cerință	Alocat	Utilizabile	Risipite
60 stații	/26	62	2
30 stații	/26	62	32
15 stații	/26	62	47
Total risipit			81 de adrese

Exemplul 5.2. De ce pentru 15 stații nu ajunge un /28, care oferă 14 adrese utilizabile?

Rezolvare. Pentru că $14 < 15$. Trebuie mers la /27, care oferă 30.

Și mai important: la cele 15 stații trebuie adăugat și **gateway-ul**, care consumă și el o adresă utilizabilă. Deci cerința reală e 16, iar /28 e cu atât mai insuficient.

Aceasta e greșeala numărul unu la examen: se uită gateway-ul.

5.7 VLSM: subnetarea optimă

VLSM (Variable Length Subnet Mask) renunță la constrângerea ca toate subrețelele să aibă aceeași mască. Fiecare subrețea primește exact cât îi trebuie.

Regula de aur a VLSM. Alocarea se face **de la cea mai mare cerință spre cea mai mică**. Ordinea nu e o convenție estetică - e obligatorie.

Motivul: un bloc de dimensiune 2^n trebuie să înceapă la o adresă **aliniată** la 2^n . Dacă alocați mai întâi câteva blocuri mici, alinierea se pierde, iar următorul bloc mare nu mai poate începe decât mult mai departe - spațiul dintre ele devine inutilizabil.

Aceleași cerințe ca mai sus, rezolvate cu VLSM. Calculul biților de host necesari:

1. **60 de stații** + gateway = 61 $\rightarrow 2^6 - 2 = 62 \geq 61 \rightarrow$ **6 biți de host** $\rightarrow /26$
2. **30 de stații** + gateway = 31 $\rightarrow 2^5 - 2 = 30 < 31 \rightarrow$ nu ajunge; $2^6 - 2 = 62 \rightarrow /26$.
Dacă cerința e strict 30 *inclusiv* gateway-ul, /27 este suficient.
3. **15 stații** + gateway = 16 $\rightarrow 2^5 - 2 = 30 \geq 16 \rightarrow$ **5 biți** $\rightarrow /27$

Tabelul 5.8 VLSM: subnetarea optimă.

Cerință	Subrețea alocată	Adrese utilizabile	Risipă
60 stații	192.168.10.0/26	62	2
30 stații	192.168.10.64/27	30	0
15 stații	192.168.10.96/27	30	15
Total risipit			17 în loc de 81

În plus, rămâne liber tot blocul de la 192.168.10.128 în sus - 128 de adrese pentru extinderi viitoare, față de niciuna în varianta cu subrețele egale.

**Figura 5.4** Împărțitor VLSM - verificați-vă calculele.

Exercițiu rezolvat, complet 5.3. Subnetăți optim 172.18.240.0/23 pentru: o rețea cu 200 de stații, una cu 90, două cu 20, una cu 6 și trei cu 4 stații.

Rezolvare. **Pasul 1 - ordonăm descrescător:** 200, 90, 20, 20, 6, 4, 4, 4.

Pasul 2 - calculăm biții de host pentru fiecare:

Cerință	Necesar cu gateway	Biți de host	Prefix	Subrețea	Broadcast
200	201	8 → 254 utilizabile	/24	172.18.240.0	172.18.240.255
90	91	7 → 126	/25	172.18.241.0	172.18.241.127
20	21	5 → 30	/27	172.18.241.128	172.18.241.159
20	21	5 → 30	/27	172.18.241.160	172.18.241.191
6	7	3 → 6	/29	172.18.241.192	172.18.241.199
4	5	3 → 6	/29	172.18.241.200	172.18.241.207
4	5	3 → 6	/29	172.18.241.208	172.18.241.215
4	5	3 → 6	/29	172.18.241.216	172.18.241.223

Pasul 3 - verificarea: fiecare subrețea începe exact acolo unde s-a terminat cea dinainte (broadcast + 1). Dacă undeva există o gaură sau o suprapunere, ați greșit calculul.

Rămâne liber blocul de la 172.18.241.224 în sus - 32 de adrese.

Legăturile dintre rutere. Pentru o legătură punct-la-punct între două rutere folosiți /30 - două adrese utilizabile, exact cât trebuie. Pe echipamentele care îl suportă, /31 este și mai eficient: două adrese, fără rețea și broadcast.

Un /24 alocat unei legături între două rutere risipește 252 de adrese pentru absolut nimic. Este risipa clasică din proiectele de începător.

5.8 ARP: de la adresa IP la adresa MAC

Iată o problemă practică pe care nu am rezolvat-o încă. O stație vrea să trimită un pachet și cunoaște adresa IP a destinației. Dar cadrul Ethernet în care trebuie încapsulat pachetul cere o **adresă MAC** - și pe aceea nu o are.

Protocolul care face traducerea se numește **ARP** (Address Resolution Protocol).



Figura 5.5 Un schimb ARP complet.

Ce se întâmplă când destinația e în altă rețea

Aici e partea care produce cea mai multă confuzie la început.

1. Sursa verifică dacă destinația e în aceeași rețea, aplicând propria mască peste ambele adrese.
2. Dacă **da** - emite ARP pentru adresa IP a destinației, ca mai sus.
3. Dacă **nu** - emite ARP pentru adresa IP a **propriului default gateway**. Pentru că acolo trebuie să ajungă cadrul, fizic.

Regula de aur a celor două adrese. Adresele **IP** sursă și destinație rămân **neschimbate** de la un capăt la altul al comunicației.

Adresele **MAC** sursă și destinație se **rescriu pe fiecare segment** Ethernet parcurs.

Consecință practică: dacă vedeți într-o captură un pachet cu adresa IP sursă din altă rețea și adresa MAC sursă a ruterului local, totul este perfect normal. Dacă vedeți adresa **IP** sursă rescrisă, atunci acolo se face NAT - subiectul cursului 9.

DEFINIȚIA 5.3 (PROXY ARP). Un ruter poate răspunde cu **propria sa adresă MAC** la o cerere ARP pentru o adresă IP din afara rețelei emițătorului, dacă are o rută spre acea destinație. Stația crede că vorbește direct cu destinația, dar cadrele ajung, de fapt, la ruter.

Mecanismul salvează configurații greșite - de exemplu o stație fără gateway configurat - dar maschează probleme reale și de aceea este dezactivat implicit pe multe echipamente moderne.

Câte tabele ARP are un ruter? Câte **una pentru fiecare interfață multiacces activă**. Un switch, în schimb, are **zero**: el nu se uită la adrese IP, deci nu are ce traduce. Excepția e adresa IP de management a switch-ului, care aparține interfeței virtuale, nu funcției de comutare.

5.9 DHCP: adrese fără configurare manuală

Configurarea manuală a adreselor este realistă pentru rutere și servere. Pentru sutele de laptopuri și telefoane care intră și ies dintr-o rețea în fiecare zi, este imposibilă.

DHCP (Dynamic Host Configuration Protocol) automatizează procesul prin patru mesaje, ușor de memorat sub acronimul **DORA**.



Figura 5.6 DORA: cele patru mesaje DHCP.

DHCP relay

Mesajul Discover este un broadcast către 255.255.255.255, iar ruterele **nu propagă broadcast-uri** - asta le e treaba. Într-o rețea cu mai multe VLAN-uri și un singur server DHCP central, cererea nu ar ajunge niciodată la el.

Soluția este configurarea unui **DHCP relay** pe ruterul din rețeaua locală: acesta preia broadcast-ul, îl transformă într-un unicast către adresa serverului și îi atașează informația despre rețeaua din care provine - ca serverul să știe din ce rezervor să aloce.

```
R1(config)# interface gigabitEthernet 0/0.20
R1(config-subif)# ip helper-address 10.0.100.5
```

Listingul 5.1 DHCP relay pe interfața dinspre clienți copiază



Figura 5.7 Terminal: comenzile de diagnostic pe o stație.

5.10 Greșeli frecvente

Greșeli frecvente.

„Am uitat gateway-ul din calculul de adrese”

Gateway-ul consumă și el o adresă utilizabilă. Pentru 30 de stații plus gateway vă trebuie un /26, nu un /27.

Cerința reală = numărul de stații + 1.

„Am alocat subrețelele VLSM în ordinea din enunț”

Ordinea trebuie să fie descrescătoare după dimensiune . Altfel alinierea se pierde și blocurile mari nu mai încap.

Ordonăți întotdeauna înainte de a alocă.

„Măștile pot fi diferite pe stațiile aceluiași segment”

Nu produc o eroare vizibilă la configurare, dar produc comportament asimetric: o stație crede că destinația e locală, cealaltă crede că e la distanță.

Aceeași mască pe toate stațiile unui segment, fără excepție.

„Stația trimite ARP pentru adresa IP a destinației finale”

Doar dacă destinația e în aceeași rețea. Altfel trimite ARP pentru gateway , pentru că acolo trebuie să ajungă fizic cadrul.

Întâi se aplică masca, apoi se decide pe cine se întreabă.

„169.254.x.x e o adresă validă, deci rețeaua merge”

Este semnul clar că DHCP-ul nu a răspuns. Două stații cu astfel de adrese se pot chiar vedea între ele, ceea ce sporește confuzia.

Vedeți 169.254 → căutați serverul DHCP sau relay-ul, nimic altceva.

5.11 Rezumat și glosar

1. Adresa IP e **ierarhică**, iar asta face rutarea posibilă la scară globală - spre deosebire de adresa MAC, care e plată.
2. Masca separă partea de rețea de partea de host. **ȘI cu masca** dă adresa de rețea; **SAU cu inversa** dă broadcast-ul.
3. **VLSM** alocă fiecărei subrețele exact cât îi trebuie, în ordine descrescătoare.
4. **ARP** traduce IP → MAC în interiorul unui segment. Adresele IP rămân constante, cele MAC se rescriu la fiecare salt.
5. **DHCP** distribuie automat adrese, în patru mesaje. Peste granițele de broadcast e nevoie de **relay**.

Exerciții

1. Care este adresa de broadcast a rețelei 192.168.17.0/26?
 - a. 192.168.17.255
 - b. 192.168.17.63
 - c. 192.168.17.127
 - d. 192.168.17.64
2. De ce se alocă subrețelele VLSM începând cu cea mai mare cerință?
 - a. Ca să fie mai ușor de citit tabelul
 - b. Pentru că un bloc de dimensiune 2^n trebuie să înceapă la o adresă aliniată la 2^n
 - c. Pentru că protocoalele de rutare cer această ordine
 - d. Nu contează ordinea
3. O stație vrea să trimită un pachet către o adresă IP din altă rețea. Pentru cine emite cererea ARP?
 - a. Pentru adresa IP a destinației finale
 - b. Pentru adresa IP a propriului default gateway
 - c. Pentru adresa de broadcast a rețelei destinație
 - d. Nu emite nicio cerere ARP
4. Ce se întâmplă cu adresele MAC ale unui pachet care traversează trei rutere?
 - a. Rămân neschimbate, ca și cele IP
 - b. Se rescriu pe fiecare segment Ethernet
 - c. Se șterg complet după primul router
 - d. Sunt înlocuite cu adrese IP
5. O stație are adresa 169.254.13.7. Ce s-a întâmplat?
 - a. A primit adresa de la DHCP
 - b. Nu a primit răspuns la Discover și și-a atribuit singură o adresă APIPA
 - c. Gateway-ul este configurat greșit
 - d. Mască este prea mică

6. De ce mesajul DHCP Request se trimite în broadcast, deși clientul știe deja adresa serverului?
- Pentru ca protocolul cere așa
 - Ca să afle și serverele respinse și să își elibereze rezervările
 - Pentru ca este mai rapid
 - Pentru ca serverul nu are încă adresa clientului
7. Câte adrese utilizabile oferă un /29?
- 8
 - 6
 - 4
 - 14
8. La ce folosește câmpul TTL din antetul IPv4?
- Măsoară latența pachetului
 - Este decrementat de fiecare router și oprește buclele de rutare
 - Indică prioritatea pachetului
 - Conține suma de control

Note bibliografice

Avem adrese ierarhice și mecanismele care le leagă de lumea fizică. Cursul următor arată cum se folosește ierarhia: cum decide un router, pentru fiecare pachet în parte, pe unde să îl trimită mai departe.

La **laboratorul 4** proiectați planul complet de adresare al unei firme, cu cifre reale, și îl automatizați prin DHCP.

- RFC 791 - Internet Protocol
- RFC 826 - Address Resolution Protocol
- RFC 1918 - spații de adrese pentru rețele private
- RFC 2131 - Dynamic Host Configuration Protocol
- RFC 4632 - CIDR: arhitectura de alocare și agregare a adreselor
- RFC 3927 - adresarea link-local (APIPA)