
Rutare

Un ruter nu cunoaște Internetul. Nu are o hartă a lumii și nu întreabă pe nimeni unde e destinația. Are doar o listă de direcții și o regulă de alegere, pe care le aplică din nou și din nou, pentru fiecare pachet în parte. Cursul acesta arată cum arată acea listă, cum ajung rutele în ea și de ce ordinea liniilor contează atât de mult.

Din capitolul precedent.

- Adresa IP e **ierarhică**: partea de rețea plus partea de host, separate de **maskă**.
- **ȘI între adresă și maskă** dă adresa de rețea. Această operație e inima cursului de azi.
- Adresele IP rămân **constante** de la un capăt la altul; cele MAC se rescriu la fiecare salt.
- **TTL** este decrementat de fiecare ruter și oprește buclele.

OBIECTIVELE CAPITOLULUI

- Să explicați ce e o rută și ce rol are metrica
- Să spuneți, pentru orice rută, din ce sursă provine și ce distanță administrativă are
- Să citiți o tabelă de rutare și să prevedeați exact ce se întâmplă cu un pachet dat
- Să configurați rute statice, cu next-hop sau cu interfață de ieșire, și să știți când merge fiecare
- Să scrieți o rută implicită și o rută flotantă de rezervă
- Să calculați o rută sumarizată și să folosiți interfața Null0

6.1 Ce este, de fapt, un ruter

Din punct de vedere arhitectural, un ruter este un **calculator specializat**: are procesor, memorie și dispozitive de intrare/ieșire. Diferențele față de un calculator obișnuit sunt însă semnificative, iar toate servesc aceluiași scop - să ia decizii de dirijare cât mai repede.

Tabelul 6.1 Ce este, de fapt, un ruter.

Componentă	Rol într-un ruter	Se pierde la repornire?
ROM	codul de pornire (bootstrap) și o imagine de urgență	nu - e permanentă
Flash	sistemul de operare, adică imaginea IOS	nu
RAM	configurația activă , tabela de rutare, tabela ARP, cozile de pachete	da - tot
NVRAM	configurația de pornire	nu
ASIC-uri	circuite dedicate care execută comutarea pachetelor în hardware	-

Distincția care salvează o configurație. `running-config` trăiește în **RAM** și este ce rulează chiar acum. `startup-config` trăiește în **NVRAM** și este ce se va încărca la următoarea pornire.

O configurație perfectă, nesalvată cu `copy running-config startup-config`, dispare la prima pană de curent - sau la prima repornire făcută de altcineva. Este eroarea numărul unu în laboratoare și, din păcate, și în rețele reale.

De ce un ruter e mai lent decât un switch. Un switch citește primii 12 octeți ai cadrului - cele două adrese MAC - și caută într-o tabelă cu potrivire **exactă**. Operația se face în hardware, într-un singur ciclu.

Un ruter trebuie să decapsuleze nivelul 2, să citească antetul IP, să **decrementeze TTL**, să **recalculeze suma de control**, să caute în tabela de rutare cu potrivire de **prefix** - nu exactă - și apoi să reîncapsuleze cadrul cu alte adrese MAC, ceea ce poate necesita o rezoluție ARP.

Ruterele moderne fac și asta în hardware, dar rămâne mai multă muncă. De aici și diferența de preț per port.

Pe lângă funcția de bază, ruterele moderne mai fac: filtrarea traficului cu ACL-uri (cursul 7), translatarea adreselor cu NAT și PAT (cursul 9), stabilirea de tuneluri, atribuirea de adrese ca server DHCP și proxy ARP.

6.2 Ce este o rută și ce e metrica

DEFINIȚIA 6.1 (RUTA). Triplețul (**destinație, direcție, distanță**).

„Pentru a ajunge la Brașov pot trece prin Ploiești; distanța totală va fi de 120 km.” - (Brașov, Ploiești, 120 km).

Triplețul e suficient nu doar pentru a merge, ci și pentru a **alege** între mai multe posibilități.

În rețele, „distanța” în kilometri nu spune nimic util. Mărimea asociată unei rute se numește **metrică** și poate fi oricare dintre următoarele, sau o combinație:

Tabelul 6.2 Ce este o rută și ce e metrica.

Metrică	Ce măsoară	Cine o folosește
Hop count	numărul de rutere până la destinație	RIP
Cost	invers proporțional cu lățimea de bandă	OSPF
Metrică compusă	bandă, întârziere, încărcare, fiabilitate	EIGRP
Atribute de politică	preferințe administrative, nu performanță	BGP

Exemplul 6.1. Ruterul S are două căi spre aceeași destinație: una prin A, cu 2 hopuri, dar prin două legături seriale de 64 kbps; alta prin B, cu 3 hopuri, dar prin fibră de 10 Gbps. Ce alege RIP?

Rezolvare. RIP alege calea prin A, cu 2 hopuri - pentru că este singurul lucru pe care îl măsoară.

Rezultatul: traficul trece prin legături de 64 kbps, în loc de fibră de 10 Gbps. Diferența este de peste 150 000 de ori.

Exact această slăbiciune a dus la înlocuirea RIP cu OSPF, care măsoară costul invers proporțional cu banda. Detalii în cursul 10.

6.3 De unde vin rutele

Când un ruter neconfigurat pornește, nu cunoaște **nicio** rută. Rutele apar din trei direcții:

Tabelul 6.3 De unde vin rutele.

Simbol	Sursă	Cum apare	Efort
C	Connected	automat, pentru fiecare interfață activă cu adresă IP configurată	niciunul
L	Local	automat, adresa /32 a interfeței însăși	niciunul
S	Static	scrisă manual de administrator	o linie per rețea, per ruter
R, D, O, B	dinamic	învățată automat de la vecini prin RIP, EIGRP, OSPF, BGP	configurare inițială, apoi automat

Diferența dintre C și L. Când configurați `192.168.1.1/24` pe o interfață, ruterul creează **două** intrări:

- C `192.168.1.0/24` - „rețeaua asta e direct atașată la mine, pe interfața X”
- L `192.168.1.1/32` - „adresa asta sunt chiar eu”

A doua există ca ruterul să știe imediat, fără niciun calcul, când un pachet îi este destinat lui însuși - de exemplu un ping către interfața lui, sau o conexiune SSH de administrare.

**Figura 6.1** Terminal: explorați tabela de rutare a unui ruter.

6.4 Distanța administrativă

Metrica permite compararea a două rute care provin din **aceeași** sursă. Dar cum comparăm o rută OSPF cu metrica 20 cu o rută RIP cu metrica 3?

De ce numerele nu sunt comparabile. Metrica RIP este un **număr de rutere**: 3 înseamnă „trei salturi”.

Metrica OSPF este un **cost** calculat din lățimea de bandă: 20 poate însemna „două legături de 100 Mbps” sau „o legătură serială lentă”.

Sunt unități complet diferite, ca și cum ați compara 3 kilograme cu 20 de secunde. Numerele nu au nicio relație între ele.

Este nevoie de un al doilea criteriu, aplicat **înaintea** metricii: **distanța administrativă** (AD), care exprimă cât de multă încredere are ruterul în sursa rutei. Cu cât mai mică, cu atât mai multă încredere.

Tabelul 6.4 Distanța administrativă.

Simbol	Sursă	AD	Logica din spate
C	Connected	0	ruterul vede el însuși rețeaua - nimic nu e mai sigur
S	Static	1	administratorul a decis explicit; se presupune că știe ce face
D	EIGRP	90	protocol proprietar, metrică bogată
O	OSPF	110	link-state, vedere completă a topologiei
i	IS-IS	115	tot link-state
R	RIP	120	distance-vector, metrică simplistă
–	necunoscută / neîncrezătoare	255	ruta nu va fi folosită niciodată

Regula completă de selecție

1. **Prefixul cel mai lung câștigă întotdeauna.** O rută /26 bate o rută /8, indiferent de sursă, indiferent de metrică. Acesta e criteriul suprem și nu se negociază.
2. La **același prefix**, câștigă **distanța administrativă** mai mică.
3. La **aceeași sursă**, câștigă **metrica** mai mică.
4. Dacă și metrica e egală, ruterul păstrează **ambele** căi și distribuie traficul între ele - mecanism numit *equal-cost load balancing*.

Exercițiu rezolvat 6.2. Un ruter cunoaște următoarele rute. Care ajung în tabela de rutare?

#	Rută
1	C (10.0.0.8/30, Fa0/0)
2	R (141.85.37.0/24, metrică 3)
3	C (10.0.0.4/30, Fa0/1)
4	R (141.85.37.0/24, metrică 2)
5	D (200.0.0.0/16, metrică 31452)
6	R (200.0.0.0/16, metrică 3)

Rezolvare. Rutele 1 și 3 intră: sunt singurele către destinațiile lor, și oricum au AD 0.

Ruta 4 intră, ruta 2 nu: aceeași destinație, aceeași sursă (RIP), deci decide metrica - 2 < 3.

Ruta 5 intră, ruta 6 nu - și aici e capcana. Metrica 3 pare mult mai bună decât 31452, dar cele două numere provin din protocoale diferite și **nu sunt comparabile**. Se aplică distanța administrativă: EIGRP are 90, RIP are 120, deci ruta EIGRP câștigă.

Dacă vi s-a părut că 6 ar trebui să câștige, ați aplicat criteriul greșit - exact ce se întâmplă la examen.



Figura 6.2 Sursa rutei și încrederea în ea.

Reținerea acestui tabel se face cel mai ușor prin logica din spatele lui: cu cât ruterul are mai multă informație de primă mână despre o rută, cu atât numărul e mai mic. O interfață proprie e adevărul absolut (0). O rută scrisă de un om e o afirmație verificabilă (1). Un protocol link-state, care are harta, e mai credibil (110) decât unul care repetă zvonuri (120).

6.5 Tabela de rutare și procesul de decizie

Tabela de rutare este harta ruterului către rețeaua din jur. Este organizată de la rutele cele mai **specifice** (mască lungă) spre cele mai **generale**, iar procesul de dirijare a unui pachet are exact doi pași.

1. **Primirea pachetului.** Se decapsulează antetul de nivel 2, se citește adresa IP destinație din antetul de nivel 3, se decrementează TTL și se recalculează suma de control.

2. **Găsirea rutei.** Se parcurge tabela; pentru fiecare rută se face ȘI între masca ei și adresa destinație, iar rezultatul se compară cu adresa de rețea a rutei. **Prima potrivire câștigă.** Dacă nu se găsește nimic, pachetul e aruncat și se trimite un ICMP *destination unreachable*.

Cea mai lungă potrivire de prefix. Ordonarea tabeli de la specific la general nu e o convenție estetică - este chiar algoritmul. Parcurgând în această ordine și oprindu-se la prima potrivire, ruterul găsește automat **cea mai specifică** rută care se potrivește.

De aceea o rută /26 bate una /8: apare mai devreme în tabelă. Iar ruta implicită, /0, e ultima - se aplică doar dacă nimic altceva nu s-a potrivit.



Figura 6.3 Consultarea tabeli de rutare - testați destinații.

Experimentele de făcut aici.

1. 13.1.0.5 - se potrivește atât cu /14, cât și cu /8. Câștigă prefixul mai lung. Urmăriți coloana din dreapta ca să vedeți de ce.
2. 13.128.0.1 - nu mai încapă în /14, deci cade pe /8. Aceeași rețea „13”, altă rută.
3. 192.168.3.7 - se potrivește și cu /24 (spre Null0), și cu /20. Câștigă /24, deci pachetul este **aruncat** deliberat.
4. 8.8.8.8 - nu se potrivește cu nimic specific și cade pe ruta implicită.
5. 172.30.14.9 - nu se potrivește cu niciuna dintre cele două rute conectate. Verificați de ce, calculând ȘI-ul.

Cele două legi ale rutării. 1. Rutarea se face **individual, pentru fiecare pachet în parte**. Două pachete ale aceleiași conexiuni pot lua căi diferite, dacă între timp s-a schimbat ceva.

2. Fiecare ruter ia decizia **exclusiv pe baza propriei tabeli**. Nu există coordonare, nu există confirmare că pachetul a ajuns și - foarte important - nu există nicio garanție că drumul de întoarcere este același.

Exemplul 6.3. Configurați o rută pe ruterul A spre rețeaua din spatele lui C. Dați ping. Nu merge. De ce?

Rezolvare. Pentru că pachetul **a ajuns** la destinație. Dar destinația a răspuns, iar răspunsul nu are pe unde să se întoarcă: ruterul C nu știe unde e rețeaua lui A.

Rutarea trebuie configurată **în ambele direcții**, întotdeauna. Este cea mai frecventă greșeală de început și produce simptomul cel mai derutant: „pachetul pleacă, dar nu primesc răspuns”.

Cum verificați: dați ping **din C spre A**. Dacă nici acela nu merge, știți sigur unde e problema.



Figura 6.4 Decizia de rutare, pas cu pas.

6.6 Ruta implicită

Ruta $0.0.0.0/0$, numită și *quad-zero*, are masca formată numai din zerouri. Operația ȘI cu ea dă întotdeauna $0.0.0.0$, care coincide întotdeauna cu adresa de rețea a rutei - deci se potrivește cu **orice** destinație.

Fiind cea mai generală, ocupă ultima poziție în tabelă și se aplică doar dacă nimic altceva nu s-a potrivit. Este ruta pe care orice rețea mică o are spre furnizorul de Internet.

Analogie Un birou de sortare a corespondenței are rafturi pentru destinațiile pe care le cunoaște: „Brașov”, „Cluj”, „București”. Iar la capăt are un raft mare pe care scrie „restul lumii”. Tot ce nu se potrivește pe niciun raft specific ajunge acolo și pleacă spre centrul de sortare superior.

Ruta implicită este acel raft. Iar ruterul furnizorului de Internet este centrul de sortare superior.

Cine are și cine nu are rută implicită. O rețea de întreprindere are, de regulă, o singură rută implicită - spre furnizor.

Ruterele din **nucleul** Internetului nu au niciuna. Ele trebuie să cunoască explicit toate cele peste 900 000 de prefixe globale, pentru că nu există „mai sus” unde să trimită ce nu cunosc. Aceasta e diferența dintre a fi la marginea Internetului și a fi în mijlocul lui.

6.7 Rute statice

O rută statică este scrisă manual de administrator. Direcția poate fi exprimată în două feluri, iar alegerea între ele nu e indiferentă.

```
! prin adresa urmatorului salt - merge intotdeauna
R1(config)# ip route 192.168.10.0 255.255.255.0 10.0.0.3

! prin interfata de iesire - doar pe legaturi punct-la-punct
R1(config)# ip route 192.168.10.0 255.255.255.0 serial 0/0/0

! ambele, varianta cea mai explicita si mai rapida
R1(config)# ip route 192.168.10.0 255.255.255.0 serial 0/0/0 10.0.0.3

! ruta implicita
R1(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.3
```

Listingul 6.1 cele trei forme ale unei rute statice copiază



Figura 6.5 De ce next-hop-ul contează pe Ethernet.

Există și o portiță - dar nu vă bazați pe ea. Forma cu interfață *poate* funcționa pe Ethernet dacă vecinul are activat **proxy ARP**: el va răspunde la cererea ARP în locul destinației, cu propria adresă MAC.

A construi o rețea pe această bază înseamnă însă a construi o rețea care se strică atunci când cineva, absolut corect, dezactivează proxy ARP din motive de securitate. Scrieți next-hop-ul.

Exercițiu rezolvat 6.4. Trei rutere A, B, C legate în triunghi. A are LAN-ul 172.16.0.0/16, B are 144.13.248.0/21, C are 140.20.0.0/20. Legătura A-C este 10.0.0.4/30 (A are .5, C are .6). Configurați rute statice astfel încât LAN A să comunice cu LAN C, pe calea optimă.

configurație

```
A(config)# ip route 140.20.0.0 255.255.240.0 10.0.0.6
C(config)# ip route 172.16.0.0 255.255.0.0 10.0.0.5
```

Rezolvare. Ambele rute sunt necesare - legea a doua a rutării. Pachetul ajunge la destinație cu prima, dar răspunsul nu are pe unde să se întoarcă fără a doua.

Atenție și la măști: /20 înseamnă 255.255.240.0, nu 255.255.255.0. O mască greșită produce un simptom bizar: unele stații răspund, altele nu.

Ruta flotantă

Uneori vrem o cale de rezervă, care să se activeze **doar** dacă cea principală dispare. Se obține dându-i rutei de rezervă o distanță administrativă **mai mare**.

```
! principala - AD implicit 1
R1(config)# ip route 172.20.19.0 255.255.255.0 10.0.0.6

! rezerva - AD 200, deci nu intra in tabela cat timp prima e valida
R1(config)# ip route 172.20.19.0 255.255.255.0 10.0.0.2 200
```

Listingul 6.2 rută principală și rută flotantă copiază

Verificați cu `show ip route`: ruta cu AD 200 **nu apare** în tabelă atât timp cât cea cu AD 1 e validă. Închideți interfața principală și priviți din nou - a apărut. Redeschideți-o și verificați că se retrage.

6.8 Traseul complet al unui pachet

Aici punem cap la cap tot ce am învățat: ARP din cursul 5, comutarea din cursul 3 și rutarea de azi. Este exercițiul care apare, într-o formă sau alta, la fiecare examen.



Figura 6.6 Un pachet, de la un capăt la altul.

6.9 Sumarizarea rutelor și interfața nulă

Un ruter care are patru rute spre patru subrețele consecutive le poate înlocui pe toate cu una singură, dacă acestea au un prefix comun. Procesul se numește **sumarizare** (sau agregare).

Cum se calculează

Se scriu adresele în binar și se observă câți biți de la stânga sunt **comuni tuturor**. Acel număr este prefixul rutei summarize.

```
192.168.1.0/26   192.168.1.00000000
192.168.1.64/26  192.168.1.01000000
192.168.1.128/26 192.168.1.10000000
192.168.1.192/26 192.168.1.11000000
```

comun: primii 24 de biți → 192.168.1.0/24

Exercițiu propus 6.5. Care este ruta sumarizată pentru 10.4.0.0/16, 10.5.0.0/16, 10.6.0.0/16 și 10.7.0.0/16?

Rezolvare. Se compară al doilea octet, în binar:

4 = 00000**100**

5 = 00000**101**

6 = 00000**110**

7 = 00000**111**

Primii 6 biți sunt comuni: 000001. Primul octet contribuie cu 8 biți comuni, deci total $8 + 6 = 14$.

Ruta sumarizată: 10.4.0.0/14.

Verificare: /14 acoperă $2^{(32-14)} = 262\,144$ adrese, adică exact patru blocuri /16. Corect.



Figura 6.7 Cât economisește o sumarizare.

Alegeți /26 sumarizat la /24: patru rute devin una. Alegeți /30 sumarizat la /22 - cazul legăturilor punct-la-punct dintr-o rețea mare - și veți vedea de ce sumarizarea nu este un moft: **256** de linii înlocuite cu una singură, în tabela fiecărui ruter din rețea.

De ce contează, la scară. Avantajul imediat: tabele de rutare mai mici, căutări mai rapide, mai puține actualizări atunci când o subrețea apare sau dispare.

La scara Internetului, sumarizarea este **singurul motiv** pentru care tabela globală de rutare are sute de mii de intrări în loc de sute de milioane. Fără ea, Internetul pur și simplu nu ar funcționa la dimensiunea actuală.

Efectul secundar și interfața nulă

Sumarizarea are un preț. Dacă din cele patru subrețele una - să zicem 192.168.1.128/26 - nu mai există, ruta sumarizată 192.168.1.0/24 continuă să atragă traficul pentru ea și îl trimite mai departe, unde va fi aruncat abia la capătul lanțului. În cazul unei bucle, traficul poate chiar oscila între rutere până la expirarea TTL-ului.

Soluția este o rută statică spre **Null0**, interfața nulă: o destinație care aruncă imediat tot ce primește. Fiind mai specifică decât ruta sumarizată, ea câștigă și oprește

traficul chiar la sursă.

```
R1(config)# ip route 192.168.1.128 255.255.255.192 null0
```

Listingul 6.3 ruta „groapă de gunoi” copiază

Echivalentul din lumea Linux este `/dev/null`; ideea e aceeași - un loc în care lucrurile dispar, la cerere și fără eroare.

6.10 Greșeli frecvente

Greșeli frecvente.

„Am configurat ruta, dar ping-ul nu merge”

În nouă cazuri din zece, lipsește ruta de întoarcere . Pachetul ajunge; răspunsul nu are pe unde să vină înapoi.

Rutarea se configurează întotdeauna în ambele direcții.

„Am comparat metricile a două protocoale diferite”

Nu sunt comparabile. Se aplică întâi distanța administrativă.

Prefix mai lung → AD mai mic → metrică mai mică. În această ordine.

„Am scris ruta cu interfața de ieșire, pe Ethernet”

Ruterul nu știe ce adresă MAC să pună în cadru, pentru că pe segment sunt mai mulți candidați.

Pe Ethernet, scrieți întotdeauna next-hop-ul.

„Am salvat configurația... cred”

running-config e în RAM. Fără copy running-config startup-config , se pierde la repornire.

Salvați după fiecare bloc de modificări, nu la final.

„Am sumarizat și acum unele destinații nu mai răspund”

Ruta sumarizată acoperă și subrețele care nu există sau nu sunt ale voastre.

Adăugați rute spre Null0 pentru porțiunile neutilizate ale blocului.

6.11 Rezumat și glosar

1. O **rută** este tripletul (destinație, direcție, distanță). Distanța se numește **metrică** și depinde de protocol.
2. Selecția se face în ordine: **prefix mai lung → AD mai mic → metrică mai mică**.

3. Tabela e ordonată de la specific la general, iar **prima potrivire câștigă**. Ruta implicită e ultima.
4. Pe Ethernet, o rută statică are nevoie de **next-hop**, pentru că altfel lipsește adresa MAC destinație.
5. Adresele IP rămân constante pe tot traseul; cele MAC se rescriu la fiecare salt; TTL scade cu 1 la fiecare ruter.
6. **Sumarizarea** micșorează tabelele; **Null0** oprește traficul spre porțiunile neutilizate.

Exerciții

1. Ce înseamnă [110/65] într-o intrare din tabela de rutare?
 - a. 110 hopuri și 65 de secunde
 - b. Distanța administrativă 110 și costul cumulativ 65
 - c. Costul 110 și metrica 65
 - d. Adresa 110.0.0.65
2. O rută statică are AD 200 și există și o rută OSPF spre aceeași destinație. Care intră în tabelă?
 - a. Cea statică, pentru că rutele statice au prioritate
 - b. Cea OSPF, pentru că 110 este mai mic decât 200
 - c. Amândouă, pentru echilibrarea traficului
 - d. Depinde de metrica
3. Un pachet pentru 13.1.0.5 găsește în tabelă atât 13.0.0.0/8 cât și 13.0.0.0/14. Ce se întâmplă?
 - a. Se folosește prima găsită în tabelă
 - b. Câștigă /14, pentru că prefixul este mai lung
 - c. Câștigă /8, pentru că acoperă mai mult
 - d. Pachetul este duplicat pe ambele căi
4. O rută statică prin interfața de ieșire nu funcționează pe Ethernet. De ce?
 - a. Pentru că Ethernet nu suportă rute statice
 - b. Pentru că routerul nu știe ce adresă MAC destinație să pună în cadru, pe un mediu multiacces
 - c. Pentru că interfața trebuie să fie în stare shutdown
 - d. Pentru că Ethernet nu are TTL

5. Care este ruta sumarizata pentru 10.4.0.0/16, 10.5.0.0/16, 10.6.0.0/16 si 10.7.0.0/16?
 - a. 10.0.0.0/8
 - b. 10.4.0.0/14
 - c. 10.4.0.0/16
 - d. 10.4.0.0/22

6. La ce foloseste o ruta statica spre Null0?
 - a. Pentru a crea o bucla controlata
 - b. Pentru a arunca imediat traficul spre o subretea inexistentă, acoperita de o ruta sumarizata
 - c. Pentru a redirectiona traficul spre firewall
 - d. Pentru a testa interfata de loopback

7. Unde se afla configuratia care se pierde la repornirea ruterului?
 - a. In Flash
 - b. In NVRAM
 - c. In RAM - este running-config
 - d. In ROM

8. Un ping catre o retea indepartata nu primeste raspuns, desi ruta e configurata pe ruterul local. Ce verificati intai?
 - a. Cablul
 - b. Daca exista ruta de intoarcere pe ruterul indepartat
 - c. Viteza interfetei
 - d. Adresa MAC a destinatiei

Note bibliografice

Rutele statice sunt exacte și previzibile, dar nu reacționează la nicio schimbare și nu scalează. Cursul 10 introduce protocoalele care întrețin tabela singure. Până atunci, cursurile 7 și 9 arată ce **altceva** mai poate face un ruter cu pachetele care trec prin el: să le filtreze și să le rescrie adresele.

La **laboratorul 5** configurați exact topologia de aici, întâi cu rute statice și apoi cu OSPF, apoi tăiați un cablu și cronometrați cine își revine mai repede.

- RFC 1812 - cerințele pentru rutere IPv4
- RFC 4632 - CIDR și agregarea rutelor

- Cisco - *IP Routing: Protocol-Independent Configuration Guide*
- Radia Perlman, *Interconnections: Bridges, Routers, Switches and Internetworking Protocols*

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