

Rutare: statică, rute flotante și OSPF

Aceeași topologie, două filozofii. Întâi scrieți fiecare rută de mână și vedeți exact ce știe ruterul și de unde. Apoi ștergeți tot și lăsați OSPF să facă aceeași treabă singur - după care tăiați un cablu și cronometrați cine își revine mai repede.

OBIECTIVELE CAPITOLULUI

- Configurați rutarea statică completă și demonstrați conectivitate între toate cele trei LAN-uri
- Demonstrați funcționarea unei rute flotante: arătați tabela înainte și după căderea legăturii principale
- Calculați și configurați ruta sumarizată, plus ruta spre Null0
- Ștergeți rutarea statică și configurați OSPF single-area pe toate ruterele
- Documentați tabela de vecini a fiecărui ruter, cu stările adiacenței
- Forțați OSPF să aleagă o cale suboptimă modificând costul; demonstrați cu `traceroute`
- Măsurați și raportați timpii de convergență în ambele scenarii
- Configurați `default-information originate` și verificați că R2 și R3 primesc ruta implicită, marcată `O*E2`

5.1 Ce știe un ruter și de unde

Un ruter proaspăt configurat cunoaște un singur lucru: rețelele legate direct de el. Atât. Despre tot restul lumii nu are nici cea mai vagă idee, iar pachetele care merg acolo sunt aruncate.

Există exact două moduri de a-i spune restul.

Rute statice.

- Le scrieți voi, una câte una
- Fac exact ce ați scris, previzibil
- Nu consumă bandă și nu pot fi păcălite
- La 20 de rețele și 5 rutere înseamnă 100 de comenzi, iar la fiecare modificare le rescrieți

Protocol dinamic (OSPF)

- Ruterele vorbesc între ele și își spun ce știu
- O rețea nouă se anunță o singură dată, pe un singur ruter
- Când o legătură cade, drumul se recalculează singur
- Mai complex de configurat și de depanat

DEFINIȚIA 5.1 (DISTANȚĂ ADMINISTRATIVĂ (AD)). Cât de mult are încredere ruterul în sursa unei rute. Conectat direct: 0. Rută statică: 1. OSPF: 110. Când două surse propun rute către aceeași destinație, câștigă cea cu AD *mai mică*.

DEFINIȚIA 5.2 (METRICĂ). Cât de „scump” este un drum, în interiorul aceluiași protocol. La OSPF, metrica este costul, calculat din banda interfețelor. Se compară numai între rute din aceeași sursă.

DEFINIȚIA 5.3 (LONGEST PREFIX MATCH). Când mai multe rute se potrivesc cu o destinație, ruterul o alege pe cea cu prefixul *cel mai lung* - cea mai specifică. O rută /24 bate una /16, indiferent de AD sau metrică.

Ordinea deciziei, o dată pentru totdeauna. Întâi prefixul cel mai lung. Dacă sunt mai multe cu același prefix, câștigă AD-ul cel mai mic. Dacă și AD-ul e egal, câștigă metrica cea mai mică. Aceste trei rânduri explică aproape orice comportament ciudat al unei tabel de rutare.

5.2 Echipamente necesare

Tabelul 5.1 Echipamente necesare.

Buc.	Echipament	Model	Categoria din Packet Tracer	La ce folosește aici
4	Ruter	4331	Network Devices → Routers	R1, R2, R3 și ruterul ISP
3	Switch	2960	Network Devices → Switches	câte unul pentru LAN-ul fiecărui ruter
3	Calculator	PC-PT	End Devices	o stație de test în fiecare LAN
1	Server	Server-PT	End Devices	„Internetul”, adresa 8.8.8.8
3	Cablu serial	Serial DCE	Connections	cele trei legături dintre rutere
5	Cablu drept	Copper Straight-Through	Connections	LAN-uri și legătura spre ISP

Dacă ruterul nu are destule interfețe seriale. În Packet Tracer, ruterul 4331 pornește fără module seriale. Opriți-l din butonul de alimentare din fila *Physical*, trageți un modul HWIC-2T într-un slot liber și porniți-l la loc. În bancul de lucru din pagină, interfețele S0/1/0 și S0/1/1 există deja.

5.3 Topologia

Se continuă topologia din laboratorul 4: trei rutere legate în triunghi, fiecare cu propriul LAN, plus o ieșire spre Internet prin ruterul ISP.

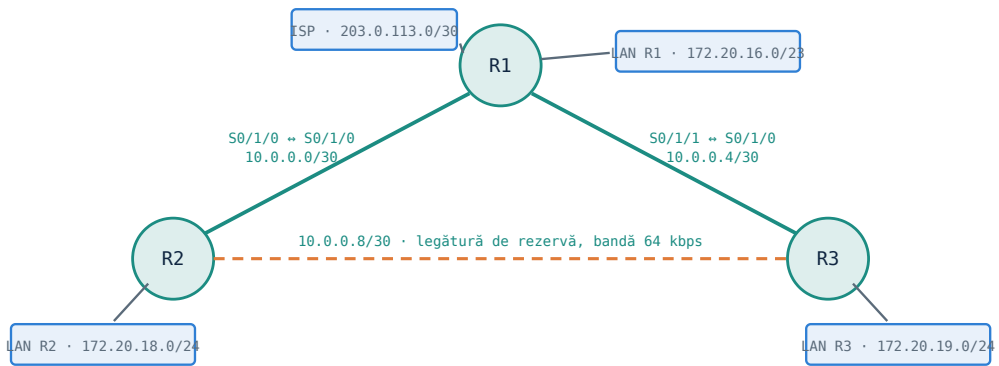


Figura 5.1 Trei rutere, două legături directe și una de rezervă, lentă. Diferența de bandă este esențială: ea face ca alegerea căii să conteze.

Tabelul 5.2 Topologia.

Ruter	LAN	Interfețe WAN	Router ID
R1	172.20.16.0/23 · G0/0/0 .1	S0/1/0: 10.0.0.1/30 · S0/1/1: 10.0.0.5/30 · G0/0/1: 203.0.113.1/30	1.1.1.1
R2	172.20.18.0/24 · G0/0/0 .1	S0/1/0: 10.0.0.2/30 · S0/1/1: 10.0.0.9/30	2.2.2.2
R3	172.20.19.0/24 · G0/0/0 .1	S0/1/0: 10.0.0.6/30 · S0/1/1: 10.0.0.10/30	3.3.3.3
ISP	8.8.8.0/24 · G0/0/1 .1	G0/0/0: 203.0.113.2/30	-

5.4 Bancul de lucru

Topologia completă, cablată, cu stațiile deja adresate. Ruterele sunt goale - le configurați voi, întâi static, apoi cu OSPF.

Obiectivele descriu starea finală. Cele două legate de OSPF rămân stinse cât timp lucrați cu rute statice - este normal. Restul se bifează încă din partea I.

5.5 Partea I: rutare statică

1. Adresați interfețele

pe R1

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# no ip domain-lookup

R1(config)# interface serial 0/1/0
R1(config-if)# ip address 10.0.0.1 255.255.255.252
R1(config-if)# clock rate 128000
R1(config-if)# no shutdown
R1(config-if)# exit
```

```
R1(config)# interface serial 0/1/1
R1(config-if)# ip address 10.0.0.5 255.255.255.252
R1(config-if)# clock rate 128000
R1(config-if)# no shutdown
R1(config-if)# exit
```

```
R1(config)# interface gigabitEthernet 0/0/0
R1(config-if)# ip address 172.20.16.1 255.255.254.0
R1(config-if)# no shutdown
R1(config-if)# exit
```

```
R1(config)# interface gigabitEthernet 0/0/1
R1(config-if)# ip address 203.0.113.1 255.255.255.252
R1(config-if)# no shutdown
R1(config-if)# end
```

Repetăți pentru R2, R3 și ISP, cu adresele din tabel. Pe legătura de rezervă R2–R3, declarați și banda, pentru ca OSPF să știe mai târziu că este lentă:
pe R2 și pe R3, interfața S0/1/1

```
R2(config)# interface serial 0/1/1
R2(config-if)# ip address 10.0.0.9 255.255.255.252
R2(config-if)# bandwidth 64
R2(config-if)# no shutdown
```

bandwidth nu schimbă viteza reală Comanda declară doar ce bandă are legătura, pentru calculele protocoalelor de rutare. Este o etichetă informativă - dar de ea depinde ce drum alege OSPF, deci trebuie să fie corectă.

2. Verificați punctul de plecare

pe R1

```
R1# show ip route
```

Trebuie să vedeți doar rute C (conectate) și L (locale). Ping-ul între LAN-uri nu funcționează - ruterele nu știu nimic unele despre altele. Verificați și acest lucru, de pe PC1: `ping 172.20.19.10` trebuie să eșueze.

3. Scrieți rutele, cu next-hop

pe R1

```
R1(config)# ip route 172.20.18.0 255.255.255.0 10.0.0.2
R1(config)# ip route 172.20.19.0 255.255.255.0 10.0.0.6
R1(config)# ip route 10.0.0.8 255.255.255.252 10.0.0.2
```

pe R2

```
R2(config)# ip route 172.20.16.0 255.255.254.0 10.0.0.1
R2(config)# ip route 172.20.19.0 255.255.255.0 10.0.0.10
R2(config)# ip route 10.0.0.4 255.255.255.252 10.0.0.1
```

pe R3

```
R3(config)# ip route 172.20.16.0 255.255.254.0 10.0.0.5
R3(config)# ip route 172.20.18.0 255.255.255.0 10.0.0.9
R3(config)# ip route 10.0.0.0 255.255.255.252 10.0.0.5
```

Atenție la mască LAN-ul lui R1 este un /23, deci masca este 255.255.254.0, nu 255.255.255.0. Este exact genul de greșeală care produce un simptom bizar: jumătate din stații răspund, cealaltă jumătate nu.

4. Testați și citiți tabela

pe R1

```
R1# show ip route
R1# show ip route static
R1# ping 172.20.19.10
R1# traceroute 172.20.19.10
```

O intrare arată așa, și fiecare bucată înseamnă ceva:
anatomia unei rute

```
S      172.20.19.0/24 [1/0] via 10.0.0.6, Serial0/1/1
|
|
|
|
|
|
|
|
|
|
└─ sursa rutei: S = statică, C = conectată, 0 = OSPF
|
└─ destinația și prefixul
|
└─ distanța administrativă (1 = statică)
|
└─ metrica (0 pentru rute statice)
|
└─ next-hop: cui i se dă pachetul
|
└─ interfața de ieșire
```

5. Experiment: interfața de ieșire în loc de next-hop

pe R1

```
R1(config)# no ip route 172.20.18.0 255.255.255.0 10.0.0.2
R1(config)# ip route 172.20.18.0 255.255.255.0 serial 0/1/0
R1(config)# end
R1# show ip route
R1# ping 172.20.18.10
```

Pe o legătură serială punct-la-punct funcționează perfect, pentru că nu există decât un singur destinatar posibil. Refaceți acum aceeași încercare pe o legătură Ethernet dintre două rutere și observați diferența.

Întrebare

De ce varianta cu interfață eșuează pe Ethernet, dar merge pe serial?

► Vezi răspunsul

Reveniți apoi la varianta cu next-hop.

5.6 Ruta implicită și ruta flotantă

1. Ruta implicită spre Internet

pe R1

```
R1(config)# ip route 0.0.0.0 0.0.0.0 203.0.113.2
```

0.0.0.0 0.0.0.0 se potrivește cu absolut orice destinație - și tocmai de aceea are prefixul cel mai scurt posibil, deci pierde în fața oricărei alte rute. Este „dacă nu știu unde, trimite acolo”. Testați de pe PC1: `ping 8.8.8.8`.

Pe ISP, configurați ruta de întoarcere spre firmă - altfel răspunsurile nu se mai întorc:

pe ISP

```
ISP(config)# ip route 172.20.16.0 255.255.252.0 203.0.113.1
```

2. Ruta flotantă, de rezervă

Traficul spre LAN-ul lui R3 merge normal direct. Dacă legătura cade, vrem să treacă prin R2. Se obține dând rutei de rezervă o **distanță administrativă mai mare**:

pe R1

```
R1(config)# ip route 172.20.19.0 255.255.255.0 10.0.0.2 200
```

Verificați cu `show ip route`: ruta cu AD 200 **nu apare** în tabelă atât timp cât cea cu AD 1 este validă. Ambele există în configurație, dar numai una intră în tabelă.

Acum închideți legătura directă și priviți din nou:

pe R1

```
R1(config)# interface serial 0/1/1
```

```
R1(config-if)# shutdown
```

```
R1(config-if)# end
```

```
R1# show ip route
```

```
R1# traceroute 172.20.19.10
```

Ruta [200/0] via 10.0.0.2 a apărut, iar `traceroute` arată un salt în plus. Redeschideți interfața și verificați că ruta de rezervă se retrage singură.

5.7 Sumarizare și Null0

Cele trei LAN-uri sunt 172.20.16.0/23, 172.20.18.0/24 și 172.20.19.0/24. ISP-ul nu are niciun motiv să știe despre ele separat - îi ajunge o singură rută.

Exemplul 5.1.

Rezolvare. În binar, al treilea octet: 16 = 00010000, 18 = 00010010, 19 = 00010011. Blocurile acoperite sunt .16, .17 (din /23), .18 și .19 - adică patru valori consecutive începând de la 16. Primii 6 biți ai octetului sunt comuni, deci prefixul este $16 + 6 = /22$.

Ruta sumarizată: 172.20.16.0/22, adică 172.20.16.0 255.255.252.0.

Pe R1, adăugați o rută spre Null0 pentru porțiunea nefolosită a blocului sumarizat, ca traficul spre adrese inexistente să fie aruncat imediat:

```
R1(config)# ip route 172.20.16.0 255.255.252.0 null0
```

Listingul 5.1 pe R1 copiază

Testați `ping 172.20.20.5` de pe ISP: pachetul ajunge la R1 și moare acolo, în loc să circule prin rețea căutând o destinație inexistentă. Destinațiile reale continuă să funcționeze, pentru că rutele lor au prefixul mai lung și câștigă.

**Figura 5.2** Tabela de rutare a lui R1 - testați destinații.

Încercați `172.20.19.10` (LAN-ul lui R3), `172.20.20.5` (în blocul sumarizat, dar inexistent) și `8.8.8.8` (Internet). Fiecare cade pe altă rută, din motive diferite - urmăriți coloana de test ca să vedeți exact de ce.

5.8 Partea a II-a: OSPF**1. Ștergeți tot ce ați scris**

pe fiecare ruter

```
R1(config)# no ip route 172.20.18.0 255.255.255.0 10.0.0.2
R1(config)# no ip route 172.20.19.0 255.255.255.0 10.0.0.6
R1(config)# no ip route 172.20.19.0 255.255.255.0 10.0.0.2 200
R1(config)# no ip route 10.0.0.8 255.255.255.252 10.0.0.2
```

Păstrați ruta implicită spre ISP și cea spre Null0. Verificați cu `show ip route` că au rămas doar rutele conectate - și că ping-ul între LAN-uri a încetat din nou să funcționeze. Este punctul de plecare curat pentru partea a doua.

2. Activați OSPF

pe R1

```
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 172.20.16.0 0.0.1.255 area 0
R1(config-router)# network 10.0.0.0 0.0.0.3 area 0
R1(config-router)# network 10.0.0.4 0.0.0.3 area 0
R1(config-router)# passive-interface gigabitEthernet 0/0/0
R1(config-router)# default-information originate
R1(config-router)# end
```

pe R2

```
R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 172.20.18.0 0.0.0.255 area 0
R2(config-router)# network 10.0.0.0 0.0.0.3 area 0
R2(config-router)# network 10.0.0.8 0.0.0.3 area 0
```

```
R2(config-router)# passive-interface gigabitEthernet 0/0/0
```

```
R2(config-router)# end
```

Pe R3, la fel, cu `router-id 3.3.3.3` și rețelele `172.20.19.0 0.0.0.255`, `10.0.0.4 0.0.0.3` și `10.0.0.8 0.0.0.3`.

Măștile wildcard nu sunt măști de rețea Comanda `network` din OSPF folosește *wildcard*: inversul măștii. Pentru un `/24` se scrie `0.0.0.255`, pentru un `/30` `0.0.0.3`, pentru un `/23` `0.0.1.255`. Regula rapidă: scădeți fiecare octet al măștii din 255.

Cele două comenzi opționale care contează `passive-interface` pe interfețele dinspre stații: rețeaua rămâne anunțată, dar nu se mai trimite Hello-uri inutile spre calculatoare - economie de bandă și o ușă de atac închisă.

`default-information originate` pe R1: anunță celorlalte rutere că el are ieșire spre Internet. Fără ea, R2 și R3 nu vor ști niciodată de ruta implicită.

3. Verificați adiacențele

pe R1

```
R1# show ip ospf neighbor
```

ce trebuie să apară

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/BDR	00:00:35	10.0.0.2	Serial0/1/0
3.3.3.3	1	FULL/BDR	00:00:35	10.0.0.6	Serial0/1/1

Ambele adiacențe trebuie să fie în starea `FULL`. Dacă lipsesc:

Symptom	Cauză cea mai probabilă
Niciun vecin în listă	rețeaua nu e inclusă în <code>network</code> , wildcard greșit, sau interfața e <code>passive</code>
Blocat în <code>INIT</code>	Hello-urile pleacă dar nu se întorc: filtrare, VLAN greșit, cablu
Blocat în <code>EXSTART</code> sau <code>EXCHANGE</code>	MTU diferit între cele două interfețe
Adiacența oscilează	intervale <code>hello/dead</code> diferite, sau legătură instabilă

alte verificări utile

```
R1# show ip protocols
```

```
R1# show ip ospf interface brief
```

```
R1# show ip route ospf
```

4. Citiți ce a învățat ruterul

`show ip route ospf` pe R1 trebuie să arate ceva de forma:

ce trebuie să apară

```
0    172.20.18.0/24 [110/65] via 10.0.0.2, Serial0/1/0
0    172.20.19.0/24 [110/65] via 10.0.0.6, Serial0/1/1
```

110 este distanța administrativă a OSPF, 65 este costul cumulativ: 64 pentru legătura serială plus 1 pentru interfața LAN de la capăt. Verificați pe R2 că a apărut și ruta implicită, marcată `O*E2` - semnul că a fost injectată din afara OSPF-ului.

Testați acum toate ping-urile: totul funcționează fără nicio rută scrisă de mână. Aceasta este diferența, în trei comenzi pe fiecare ruter în loc de trei rute pe fiecare ruter - și diferența crește cu fie-

care rețea nouă.

5.9 Costul, care decide drumul

OSPF alege drumul cu costul total cel mai mic. Costul unei interfețe se calculează din bandă:

Tabelul 5.3 Costul, care decide drumul.

Interfață	Bandă	Cost = 100 Mbps / bandă
Serial, implicit	1544 kbps	64
Serial cu bandwidth 64	64 kbps	1562
FastEthernet	100 Mbps	1
GigabitEthernet	1 Gbps	1 - costul minim este 1

Ultimul rând este o problemă reală: cu referința implicită, o legătură de 100 Mbps și una de 10 Gbps sunt indistingibile. Se corectează ridicând referința:

```
R1(config)# router ospf 1
R1(config-router)# auto-cost reference-bandwidth 10000
```

Listingul 5.2 pe toate ruterele, aceeași valoare copiază

Aceeași valoare pretutindeni. Dacă un ruter are altă referință decât vecinii lui, calculează costuri incompatibile și rezultă rutare asimetrică sau bucle. IOS afișează un avertisment; nu îl ignorați.

Forțați o cale suboptimă

```
R1# traceroute 172.20.19.10
R1(config)# interface serial 0/1/1
R1(config-if)# ip ospf cost 2000
R1(config-if)# end
R1# show ip route ospf
R1# traceroute 172.20.19.10
```

Listingul 5.3 pe R1 copiază

Cu costul 2000 pe legătura directă, drumul prin R2 ($64 + 1562 + 1 = 1627$) devine mai ieftin decât cel direct ($2000 + 1 = 2001$). `traceroute` arată acum un salt în plus. Reveniți cu `no ip ospf cost` și verificați că drumul se întoarce.

Ce ați demonstrat. Nu ați schimbat niciun cablu și nicio adresă. Ați schimbat un număr, iar întreaga rețea a recalculat drumurile singură, în ambele sensuri. Acesta este, în esență, tot avantajul rutării dinamice.

5.10 Experimentul care justifică tot laboratorul

Aveți acum aceeași rețea funcționând în două moduri. Comparați-le pe singurul criteriu care contează într-o defecțiune: **cât durează până revine traficul.**

1. De pe PC2, porniți un ping continuu spre PC3 (în Packet Tracer: `ping -t 172.20.19.10`).
2. Închideți legătura pe care circulă traficul: pe R1, `interface s0/1/1` urmat de `shutdown`.
3. Numărați câte pachete se pierd până când ping-ul revine.
4. Repetați experimentul cu rutare statică, folosind ruta flotantă configurată în partea I.

Tabelul 5.4 Experimentul care justifică tot laboratorul.

	Rutare statică cu rută flotantă	OSPF
Detectarea căderii	doar dacă interfața trece în down	prin Hello: cel mult 40 s, tipic mult mai puțin
Timp până la comutare	imediat, dacă interfața a căzut fizic	secunde: detectare + flooding + Dijkstra
Dacă legătura e „sus” dar nu transportă	nu detectează niciodată	detectează prin lipsa Hello-urilor
Efort de configurare la adăugarea unei rețele	rută nouă pe fiecare ruter	o comandă <code>network</code> , pe un singur ruter

Concluzia experimentului. Ruta flotantă comută rapid, dar *doar* dacă interfața locală cade fizic. Dacă legătura rămâne electric activă și se rupe undeva în mijloc - un scenariu foarte frecvent pe legăturile închiriate - ruta statică nu observă nimic și traficul dispare în neant. OSPF observă, pentru că nu se bazează pe starea cablului, ci pe faptul că vecinul răspunde.

5.11 Provocare

Exemplul 5.2. Reveniți la rutare statică și construiți intenționat o buclă de rutare: traficul de la LAN-ul lui R2 spre o anumită destinație trebuie să circule $R2 \rightarrow R1 \rightarrow R2 \rightarrow R1 \dots$ la nesfârșit.

1. Ce rute trebuie scrise, exact, pe fiecare ruter?
2. Vor circula pachetele la infinit? Argumentați.
3. Rulați `traceroute` spre acea destinație și descrieți ce vedeți.
4. Ar putea OSPF să producă aceeași buclă? De ce da sau de ce nu?
5. Ce mecanism din nivelul 3 lipsește la nivelul 2 și de aceea buclele de acolo sunt fatale?

Rezolvare. 1. Este suficient ca fiecare dintre cele două rutere să indice către celălalt pentru aceeași destinație: pe R2 `ip route 192.168.99.0 255.255.255.0 10.0.0.1` și pe R1 `ip route 192.168.99.0 255.255.255.0 10.0.0.2`.

2. Nu. Fiecare trecere printr-un ruter decrementează TTL-ul; la zero pachetul este aruncat și se trimite un ICMP *time exceeded* înapoi la sursă.

3. `traceroute` va afișa o alternanță perfect vizibilă R1, R2, R1, R2... până la epuizarea TTL-ului. Este cea mai clară demonstrație vizuală a unei bucle de rutare.

4. Nu, atât timp cât toate ruterele sunt în aceeași arie: fiecare are harta completă a topologiei și calculează drumuri minime pe un graf - algoritmul lui Dijkstra nu produce cicluri prin construcție. Buclele în OSPF apar doar la granițele dintre arii sau la redistribuirea între protocoale diferite, adică exact acolo unde vederea de ansamblu se pierde.

5. Câmpul TTL. Antetul Ethernet nu are niciun echivalent, motiv pentru care o buclă de nivel 2 multiplică cadrele la infinit - și pentru care STP a trebuit inventat.

5.12 Ce se predă

Tabelul 5.5 Ce se predă.

Livrabil	Format	Punctaj
Fișier Packet Tracer cu rutarea statică completă, inclusiv rută flotantă și Null0	.pkt	25 %
Fișier Packet Tracer cu OSPF configurat și funcțional	.pkt	25 %
Tabelele de vecini și de rutare, comentate	document	20 %
Măsurătorile de convergență, cu concluzia proprie	document	15 %
Provocarea: bucla construită, cu răspunsurile la cele cinci întrebări	document	15 %

Exerciții

- Ce înseamnă [110/65] într-o intrare din tabela de rutare?
 - 110 hopuri și 65 de secunde
 - Distanța administrativă 110 și costul cumulativ 65
 - Costul 110 și metrica 65
 - Adresa 110.0.0.65
- O rută statică are AD 200 și există și o rută OSPF spre aceeași destinație. Care intră în tabelă?
 - Cea statică, pentru că rutele statice au prioritate
 - Cea OSPF, pentru că 110 este mai mic decât 200
 - Amandouă
 - Depinde de metrica
- Comanda network din OSPF folosește:
 - Mască de rețea
 - Mască wildcard
 - Nicio mască
 - Adresa de broadcast
- Adiacenta OSPF rămâne blocată în starea EXCHANGE. Ce verificăm întâi?
 - Adresa IP a vecinului
 - MTU-ul celor două interfețe
 - Parola de enable
 - Cablul
- De ce o rută statică nu detectează o legătură care este electric activă dar nu transportă trafic?
 - Pentru că nu are metrica
 - Pentru că se bazează exclusiv pe starea interfeței locale, nu pe răspunsul vecinului
 - Pentru că AD este prea mare
 - Pentru că nu folosește ARP

6. Doua rute se potrivesc cu destinația: una /16 cu AD 1 și una /24 cu AD 110. Care castiga?
- a. Cea /16, pentru că are AD mai mică
 - b. Cea /24, pentru că prefixul mai lung se compară primul
 - c. Depinde de metrica
 - d. Niciuna

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