Tabela de rutare (detaliere)

Tabela de rutare (1)

- Organizare ierarhică
- Rutele sunt stocate pe nivele
- Conține:
  - adresele rețelelor direct conectate
  - rute statice
  - rute învățate prin protocoale de rutare dinamice

Tabela de rutare reprezintă o organizare ierarhică a rutelor prezente pe echipamentul de rețea la un moment dat. Aceasta este salvată în RAM, motiv pentru care se reface automat la fiecare repornire a ruter-ului. Pentru eficiență, tabela de rutare este împărțită în rute de nivel 1 și 2.

Tabela de rutare conține mai multe tipuri de rute:

•Rețelele direct conectate: sunt adăugate automat în tabelă la configurarea interfeței aferente

•Rute statice: sunt configurate manual de către administrator și vor fi preferate întotdeauna unei rute obținute printr-un protocol dinamic

•Rute învățate prin protocoale dinamice: sunt introduse automat în tabela de rutare ulterior configurării protocolului respectiv și a primirii informațiilor despre rute de la echipamentele vecine; cu ajutorul a diferiți algoritmi, update-urile de rutare sunt propagate în tot domeniul de rutare.

Tabela de rutare (2)

- Exemplu

```
Router#show ip route

Codes: I - IGRP derived, R - RIP derived, O - OSPF derived,
C - connected, S - static, E - EGP derived, B - BGP derived,
* - candidate default route, IA - OSPF inter area route,
i - IS-IS derived, ia - IS-IS, U - per-user static route,
o - on-demand routing, M - mobile, P - periodic downloaded static route,
D - EIGRP, EX - EIGRP external, E1 - OSPF external type 1 route,
E2 - OSPF external type 2 route, N1 - OSPF NSSA external type 1 route,
N2 - OSPF NSSA external type 2 route

Gateway of last resort is 10.119.254.240 to network 10.140.0.0

O E2   10.130.0.0 [160/5] via 10.119.254.6, 0:00:59, Ethernet2
E      10.10.0.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
    172.110.0.0 is variably subnetted, 2 subnets, 2 masks
C      172.110.232.32/28 is directly connected, Ethernet0
S      172.110.0.0/16 is directly connected, Ethernet0
```


Pentru a afișa tabela de rutare a unui ruter, se folosește comanda **`show ip route`**. Inițial este prezentată o legendă a acronimelor utilizate în afișarea rutelor, apoi un tabel cu detalii sumare privind rutele învățate. Tabelul cuprinde:

•Modul cum a fost învățată ruta respectivă: static (S), dinamic (se menționează protocolul utilizat: O – OSPF, D – EIGRP), direct conectată

•Adresa IP a rețelei destinație (10.130.0.0)

•Distanța administrativă împreună cu metrica specificate între paranteze drepte

•Adresa IP next-hop

•Timpul până când o rută devine invalidă

•Interfața de ieșire pentru pachetul trimis spre destinația respectivă

Tabela de rutare (3)

- Deși tabela suportă atât adresare classful cât și classless, structura ei este bazată pe adresarea classful
- Ierarhizarea tabelei ajută la determinarea rapidă a căii pe care să fie trimis pachetul
- Rutele sunt organizate pe două niveluri

Pentru păstrarea compatibilității vizuale, tabela de rutare implementează o structură de tip classful, dar în același timp sunt integrate și rutele care folosesc adresarea classless. Structura classful permite o ierarhizare eficientă a rutelor, ceea ce duce la determinarea rapidă a căii spre destinație. Se remarcă faptul că, deși parcurgerea tabelei de rutare se face secvențial, începând cu prima rută existentă, în construcția tabelei inserarea unei noi rute se face înaintea primei rute cu un prefix mai general decât aceasta. Astfel, informațiile referitoare la rețelele mai specifice se vor găsi înaintea informațiilor despre rețelele de dimensiuni mai mari.

Rutele pot fi clasificate în funcție de nivelul acestora astfel:

•Rute de nivel 1 – rutele cu o mască de rețea de o dimensiune mai mică sau egală decât cea classful a adresei respective

•Rute de nivel 2 – rute care reprezintă subrețele ale rețelelor classful

Level 1 Routes

- Au masca de rețea mai mică sau egală cu masca classful a rețelei
- Pot funcționa ca:
  - „Default Route”

```
S*   0.0.0.0 [1/0] via 192.168.1.2
```

  - „Supernet Route” (masca de rețea strict mai mică decât masca classful)

```
C   192.168.2.0/22 is directly connected, Ethernet0
```

  - „Network Route” (masca de rețea egală cu masca classful)

```
C   128.137.0.0/16 is directly connected, Ethernet0
```


O rută de nivel 1 are masca de rețea mai mică sau egală decât masca de rețea a rețelei classful din care face parte, și la rândul ei poate fi încadrată în următoarele tipuri de rute:

•Rută default, care reprezintă o rută statică cu adresa 0.0.0.0/0, spre care vor fi trimise toate pachetele pentru care nu se cunoaște o destinație specifică

•Supernet Route, rută a cărei adresă de rețea are o mască mai mică decât rețeaua classful

•Network Route, rută care are masca de rețea egală cu masca rețelei classful; aceasta poate fi și o rută părinte, în momentul în care în tabelă există rețele sau subrețele care aparțin aceluiași bloc de adrese IP classful, dar cu o mască de rețea mai specifică

Rutele de nivel 1 pot fi de tipul direct conectate, definite static sau învățate printr-un protocol de rutare dinamic.

Ultimate Routes

- Rutele care includ:
  - o adresă next-hop
  - și/sau o interfață de ieșire
- Pot fi atât rute Level 1 cât și rute Level 2
- Exemplu

```
S*   192.168.2.0 [1/0] via 192.168.1.2
                is directly connected, FastEthernet1/0
```

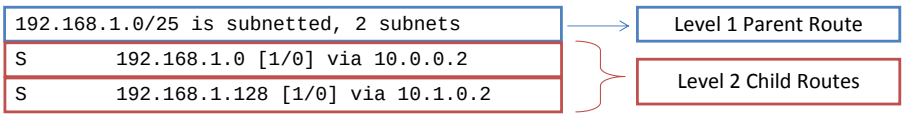

„Ultimate routes” sunt acele rutele care includ o adresă IP next-hop și/sau o interfață de ieșire. Practic reprezintă rutele pe care un pachet va face „match” înainte de a fi trimis către destinație.

Întrucât rutele Level 1 și rutele Level 2 pot fi definite printr-o adresă IP „next hop” sau o interfață de ieșire, rutele ultimate pot aparține oricăruia dintre cele două niveluri. Astfel, o rută de tip ultimate poate să aibă masca de rețea mai mică, egală sau mai mare decât cea a rețelei classful din care face parte, dar niciodată nu va putea să fie o rută părinte.

În exemplul de mai sus se poate observa o rută ultimate de nivel 1 definită atât printr-un IP „next-hop” cât și printr-o interfață de ieșire.

Parent & Child Routes

- Parent route
  - rută Level 1
  - nu conține o adresă destinație sau o interfață de ieșire
  - este adăugată automat când este introdusă în tabelă o subrețea a unei rețele classful (rută Level 2)
- Child route
  - rută Level 2
  - reprezintă o subrețea a unei rețele classful
  - conține o adresă destinație și/sau o interfață de ieșire



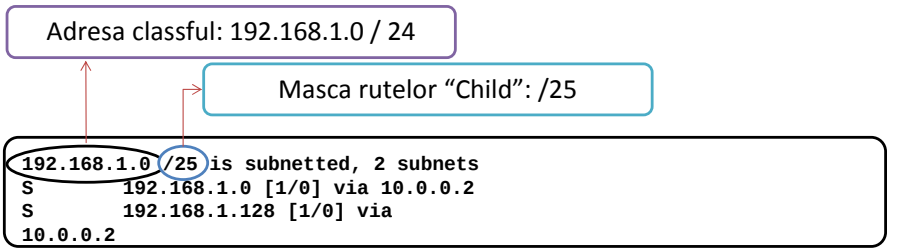
În cadrul rutelor de nivel 1 și 2 se disting rutele de tip „Parent route” și rutele de tip „Child route”. Astfel, rutele părinte se încadrează în categoria rutelor Level 1 și nu sunt definite printr-o adresă IP „next-hop” sau o interfață de ieșire. O rută părinte este creată automat de fiecare dată când un nou subnet este introdus în tabela de rutare. Ruta cu masca de rețea mai mare decât ruta părinte se încadrează în tipul „Child route”.

Rutele Child route fac parte din categoria rutelor nivel 2 și reprezintă un subnet al unei clase majore. La fel ca și în cazul rutelor de nivel 1, acestea pot fi introduse în tabela de rutare ca rute direct conectate, rute statice sau printr-un protocol de rutare dinamic.

Datorită faptului că tabela de rutare folosește o adresare classful, chiar și în cazul în care un subnet instalat în tabelă are ca sursă un protocol de rutare classless, va fi introdusă automat o rută părinte de nivel 1 având ca adresă IP rețeaua majoră classful a rutei „child”.

Parent & Child Routes: Classful

- Adresa rețelei din ruta „Parent” este adresa clasei majore
- Masca de rețea a rutei „Parent” este masca pentru rutele sale „Child”



Modul de ierarhizare al rutelor în tabela de rutare diferă în funcție de tipul de adresare, classful sau classless. În cazul adresării classful, ruta „Parent” va indica vizual adresa clasei majore și masca de rețea a subrețelelor din care fac parte rutele „Child”. Masca de rețea menționată imediat în dreapta adresei classful este afișată doar dacă rutele „Child” au atașată aceeași mască de rețea. De asemenea, în output-ul tabelei de rutare se indică pe aceeași linie cu ruta părinte și numărul de rute „Child” existente pentru o anumită rută părinte. În cazul de față, „2 subnets”.

Rutele „Child” sunt rute de nivel 2 și sunt considerate rute „Ultimate” întrucât conțin o adresă IP next-hop și/sau o interfață de ieșire.

Parent & Child Routes: Classless

- Adresa și masca rețelei din ruta „Parent” corespund clasei majore
- Este precizat numărul de subrețele și numărul de măști folosit
- Fiecare subrețea specifică adresa subrețelei și masca

```
172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       172.16.1.4/30 is directly connected, Serial0/0/0
C       172.16.1.8/30 is directly connected, Serial0/0/1
C       172.16.1.16/24 is directly connected, Serial0/1/0
```


În cazul în care rutele „Child” ale unei rute părinte folosesc o schemă de adresare VLSM, adică au mască de rețea de lungime variabilă, acest lucru este indicat în tabela de rutare prin textul „is variably subneted”, menționat ca o scurtă descriere a rutei părinte.

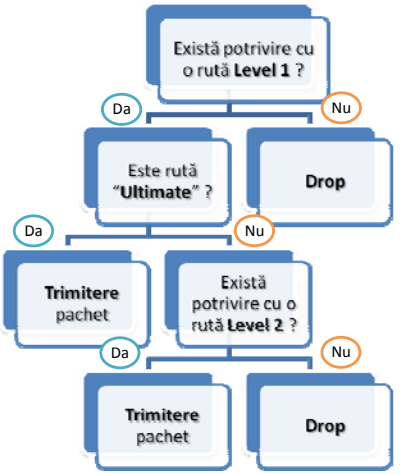
Există câteva diferențe majore de ierarhizare în cazul rutelor părinte și „child” ce folosesc o schemă de adresare classless spre deosebire de cea classful:

•Ruta părinte are afișată în dreptul acesteia masca de rețea classful proprie și nu cea a rutelor „Child”

•Sunt menționate numărul de subrețele (rute Child) și numărul de măști de rețea diferite utilizate de subrețelele respective

•Fiecare rută Child este afișată împreună cu masca de rețea proprie

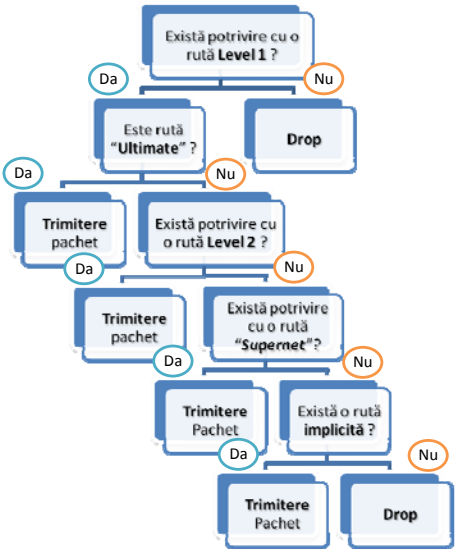
Căutarea în tabela de rutare (Classful)



Modul în care ruterul determină cea mai bună rută destinație pentru a trimite pachetele este diferită în funcție de tipul de căutare: classful sau classless. Astfel, în cazul rutării cu comportament classful, ruterul va compara inițial adresa IP destinație cu fiecare rută de nivel 1 pentru a găsi o potrivire. În caz că ruta găsită este de tip ultimate, adică are specificată o interfață de ieșire și/sau o adresă IP „next-hop", aceasta va fi folosită pentru trimiterea pachetului către destinație. Dacă ruta găsită nu este o rută ultimate, acesta va fi o rută părinte și deci se va căuta o potrivire cu rutele „Child" de nivel 2. Dacă este descoperită o potrivire cu o astfel de rută, pachetul va fi trimis mai departe, iar în caz contrar pachetul va fi aruncat.

Odată ce s-a realizat o potrivire cu o rută părinte, comportamentul classful nu va mai permite existența unei alte potriviri cu o altă rută părinte sau o rută default.

Căutarea în tabela de rutare (Classless)



Odată cu implementarea conceptului de VLSM și a utilizării adresării classless, limitările comportamentului de rutare classful a dus la utilizarea pe o scară din ce în ce mai largă a procesului classless de localizare a informațiilor de rutare. Începând cu versiunea de IOS 11.3, datorită creșterii în popularitate a schemei de adresare classless, a fost necesară folosirea procesului de căutare classless „by default”.

Căutarea classless în tabela de rutare este identică cu cea classful până la momentul localizării unei potriviri cu rutele „child” a unei rute părinte. În momentul în care nu este găsită nicio potrivire cu rutele child, pachetul nu va fi aruncat (deosebire față de comportamentul classful), ci se va continua căutarea unei potriviri printre celelalte rute „Supernet” de nivel 1. Pachetul va fi aruncat doar în cazul în care nu este găsită nicio potrivire cu o rută „Supernet” sau nu este configurată în tabelă o rută implicită.

Classful vs. Classless

- Activare mod classless
  - **(config)#ip classless**
  - este implicit pe versiunile de IOS >= 11.3
- Activare mod classful
  - **(config)#no ip classless**

Comportamentul utilizat pentru localizarea informațiilor din tabela de rutare poate fi modificat manual de către administrator. Anterior versiunii de IOS 11.3 comportamentul „default” în cazul ruterelor Cisco era classful. Acest lucru poate fi verificat prin afișarea fișierului de configurare, observându-se prezența comenzii **`no ip classless`**.

Ideea inițială de la care a pornit implementarea unui comportament de rutare classful era existența schemei de adresare IP împărțite pe clase. Astfel, înainte ca Internetul să ajungă la dimensiunile actuale, organizațiile de diferite dimensiuni primeau adrese IP doar din cadrul celor 3 clase de rețele majore: A, B sau C. Din această cauză, realizarea unei căutări în afara spațiului de adrese alocat era inutilă.

Activarea procesului de rutare classless se face prin introducerea în modul global de configurare a comenzii **`ip classless`**. Ulterior versiunii de IOS 11.3, modul implicit de căutare este setat ca fiind classless.

Rezumat

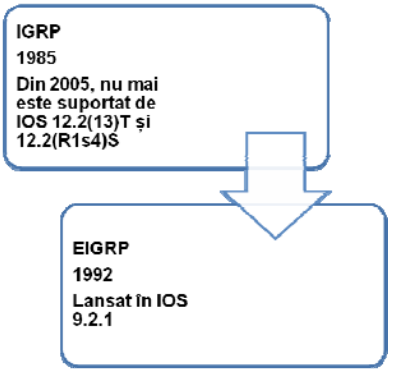
- Structura tabelei de rutare
- Comportament Classful
- Comportament Classless
- Căutarea unei adrese destinație în tabela de rutare

1. Care sunt tipurile de rute care pot exista într-o tabelă de rutare la un moment dat?
2. Care este motivul pentru care trebuie reconstruită tabela de rutare la fiecare pornire a unui ruter ?
3. Ce reprezintă rutele de nivel 2 ?
4. Cărui nivel poate aparține o rută „Ultimate” ?
5. În cazul rutarii cu comportament classful, este posibilă trimiterea unui pachet pe o rută default ?

EIGRP

IGRP

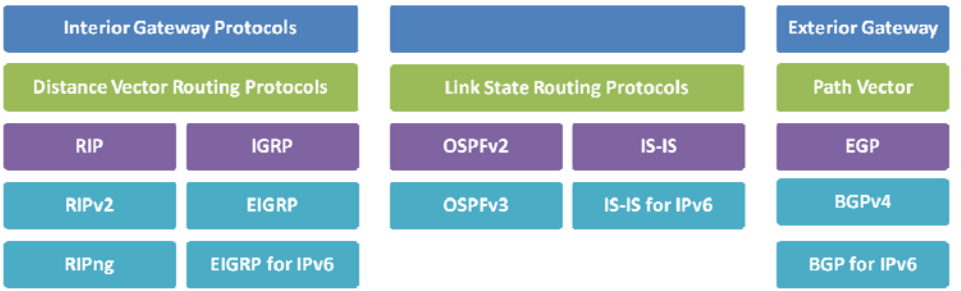
- Totul a pornit de la IGRP
  - dezvoltat în 1985 pentru a întrece limitarea numărului de hop-uri impuse de RIPv1
  - protocol de rutare distance-vector
  - metrica IGRP se bazează pe:
    - lățimea de bandă (folosită implicit)
    - delay (folosită implicit)
    - reliability
    - load



Protocolul IGRP a fost dezvoltat de Cisco în scopul găsirii unui protocol simplu (distance-vector), dar care să fie o îmbunătățire față de RIP. Deși puțin mai complex decât RIP-ul și incompatibil cu acesta, IGRP-ul împarte unele caracteristici cu RIPv1: ambele sunt protocoale distance vector classful (nu țin cont de masca de rețea), își trimit actualizările prin intermediul unor mesaje periodice de tip broadcast și ambele protocoale își determină rutele optime spre destinații cu ajutorul algoritmului Bellman-Ford. Ceea ce are în plus față de RIPv1 este utilizarea mai multor variabile pentru calcularea metricii (RIPv1 utiliza doar hop-count) și creșterea dimensiunii maxime a unei rețele, limitată la 15 noduri.

EIGRP

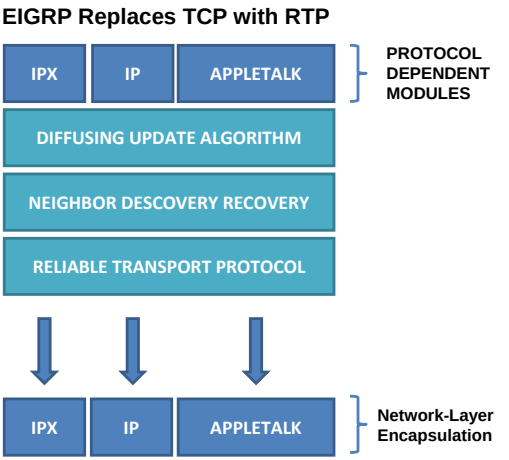
- Classless
- Distance vector – considerat în trecut un protocol hibrid, aflat la granița dintre distance vector și link state
- Distanța administrativă 90
- Suportă autentificare



EIGRP este un protocol de rutare proprietar Cisco care combină avantajele protocoalelor link-state și distance vector. Cu toate acestea, EIGRP își are rădăcinile într-un protocol distance vector, și de aici rezultă comportamentul său previzibil în diferite scenarii de rutare. Ca și predecesorul său, EIGRP este ușor de configurat și se poate adapta la o mare varietate de topologii de rețele. Ceea ce îl face pe EIGRP un protocol avansat distance vector sunt adăugarea a numeroase caracteristici ale protocoalelor link-state, cum ar fi descoperirea dinamică a vecinilor. EIGRP este un IGRP îmbunătățit, datorită convergenței rapide și a garanției că orice rută nu conține bucle. De asemenea, EIGRP este un protocol classless, având suport pentru CIDR și VLSM. Distanța administrativă de 90 îl face protocolul preferat după rutele statice. Pentru calcularea drumului minim, EIGRP utilizează algoritmul DUAL (Difussing Update Algorithm), un automat finit determinist care decide care este calea optimă până la destinație.

RTP

- Folosit pentru transmiterea și recepționarea mesajelor
- Livrare reliable și unreliable a pachetelor EIGRP:
  - reliable: necesită acknowledgement de la destinație
  - unreliable: nu necesită acknowledgement de la destinație
- Pachetele pot fi trimise:
  - unicast
  - multicast la 224.0.0.10



RTP (Reliable Transport Protocol) este protocolul de nivel 4 folosit de EIGRP pentru trimiterea și recepționarea pachetelor cu informații de rutare. Datorită complexității EIGRP, acesta necesită în unele cazuri trimiterea de mesaje reliable, pe când în alte cazuri nu este nevoie de confirmarea lor. RTP poate îndeplini această funcție; astfel, pachetele Hello (cele pentru descoperirea vecinilor și păstrarea legăturilor între rutere vecine) nu sunt trimise în mod reliable. Acestea sunt trimise de ruter pe o adresă de multicast (224.0.0.10) și au un câmp special în header-ul RTP care informează ruter-ul destinatar că acest pachet nu trebuie confirmat. Alte tipuri de pachete mai importante, cum ar fi cele de update, vor fi trimise cu cerere de confirmare. O facilitate a protocolului RTP este că permite transmiterea de mesaje ce necesită sau nu necesită confirmare chiar înainte de primirea tuturor confirmărilor anterioare.

Header-ul EIGRP

- Header data link frame – conține adresele MAC sursă și destinație
- Header IP packet – conține adresele IP sursă și destinație
- Header EIGRP packet – conține numărul sistemului autonom
- Type/Length/Field – porțiune de date a mesajului EIGRP

Encapsulated EIGRP Message			
Data Link Frame Header	IP Packet Header	EIGRP Packet Header	Type/ Length/ Values Types

Ca majoritatea protocoalelor de rutare, EIGRP se bazează pe pachete IP pentru transmiterea informațiilor. Procesul de rutare EIGRP este o funcție a nivelului transport. Pachetele de nivel 3 în care sunt încapsulate datele EIGRP au numărul de protocol 88 în header-ul IP.

Protocol Dependent Modules

- Oferă capabilitatea de a ruta mai multe protocoale de nivel 3:
  - IP
  - IPX
  - AppleTalk

Neighbor Table-Apple Talk	
Neighbor Table-IPX	
Neighbor Table-IP	
Next-Hop Router	Interface

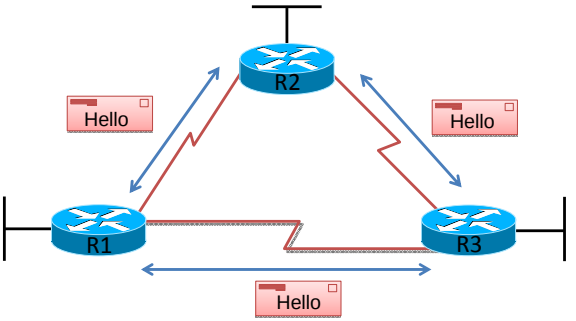
Topology Table-Apple Talk	
Topology Table-IPX	
Topology Table-IP	
Destination1	Successor
Destination2	Feasible Successor

Routing Table-Apple Talk	
Routing Table-IPX	
Routing Table-IP	
Destination1	Successor
Destination2	Feasible Successor

Modularitatea EIGRP-ului este dată de încapsularea PDM-urilor. Fiecare PDM (Protocol Dependent Modules) este specific fiecărui protocol (IPv4, IPv6, IPX, Apple Talk) și oferă o interfață comună pentru restul componentelor EIGRP-ului. Astfel, indiferent peste ce protocol rulează EIGRP-ul sau ce fel de rute transportă, protocolul va aplica același agoritm. Modularitatea este dată prin transmiterea de pachete ce au un header EIGRP generic payload de tip TLV (Type/Length/Value). În versiunea curentă de IOS, EIGRP conține inclusiv un PDM ce îi oferă suport pentru IPv6.

Pachete Hello (1)

- Folosite pentru:
  - descoperirea vecinilor
  - formarea de adiacențe cu vecinii
  - menținerea de adiacențe între vecini



Pachetele de tip Hello sunt utilizate de EIGRP pentru stabilirea adiacenței cu vecinii care rulează același proces de rutare și pentru mecanismele de keep-alive. Pachetele Hello sunt transmise întotdeauna pe adresa multicast 224.0.0.10 și nu necesită confirmare. Atunci când un ruter care rulează EIGRP primește un pachet Hello de la alt ruter din același AS (Autonomous System) spunem că între cele două rutere s-a realizat o adiacență. Cu ajutorul pachetelor Hello, fiecare ruter își construiește propriul său tabel de vecini, în care sunt memorate diferite informații despre aceștia, cum ar fi adresa IP sau timpul scurs de când s-a realizat adiacența.

Pachete Hello (2)

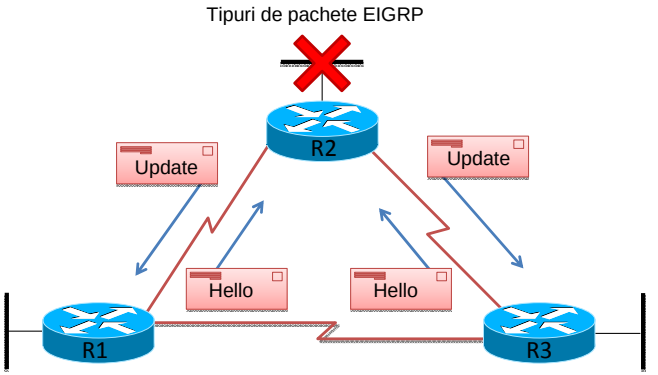
- Intervalul pachetelor de tip hello:
  - pentru majoritatea rețelelor (viteza > 1.544 Mbps)
    - la fiecare 5 secunde
  - pentru rețelele multipoint non broadcast multi-access (X.25, Frame Relay, ATM - viteza < 1.544 Mbps), dar și broadcast
    - la fiecare 60 de secunde, iar în cazul non broadcast trimiterea în mod unicast trebuie specificată folosind comanda neighbor
- Holdtime
  - timpul maxim înainte ca un vecin să fie considerat down
  - implicit = 3*intervalul hello

Pachetele „Hello” sunt trimise o dată la 5 secunde în mod multicast în majoritatea rețelelor (cu viteze mai mari de T1) și în mod unicast în rețele NBMA , o dată la fiecare 60 de secunde. Holdtime-ul este timpul care trebuie să treacă fără a primi un pachet Hello înainte de a declara vecinul picat. Holdtime-ul implicit are o valoare de 3 ori mai mare decât Hello-Interval.

În cazul în care un ruter declară legătura cu un vecin invalidă, va trimite imediat update-uri de rutare în mod multicast către toți vecinii, informându-i ca rețeaua respectivă nu mai este accesibilă. Este important de reținut ca două rutere care rulează EIGRP pot stabili adiacență chiar dacă hello-timer-ele nu sunt aceleași, astfel permițând configurarea independentă a temporizatoarelor per ruter.

Pachete de actualizare (1)

- Folosite pentru propagarea informațiilor de rutare
- Trimise la apariția unei schimbări de topologie



Pachetele de update sunt utilizate de EIGRP pentru a comunica schimbări despre topologie tuturor ruterelor din domeniu. Ele sunt trimise întotdeauna cu cerere de confirmare și sunt trimise ca multicast atunci când se descoperă o ruta nouă sau când o rută devine inaccesibilă. Pachetele de actualizare sunt utilizate și în cazul în care un ruter nou se conectează la o topologie EIGRP deja convergentă, caz în care ruterul nou venit va primi pachete de update unicast în timpul fazei de descoperire a rețelei.

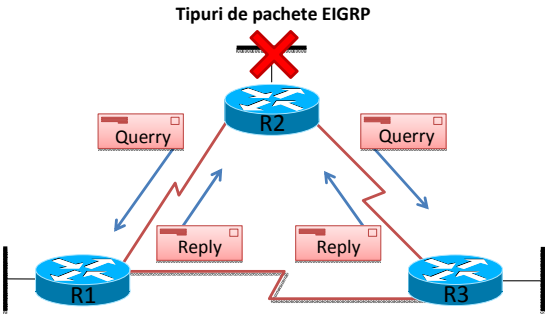
Pachete de actualizare (2)

- Actualizări parțiale
  - conțin informații numai despre ruta modificată
  - NU este trimisă întreaga tabelă de rutare
- Actualizări bounded
  - sunt trimise numai acelor routere care sunt afectate de schimbare
- Folosirea actualizărilor bounded parțiale minimizează folosirea bandei de lățime

EIGRP va trimite întotdeauna actualizări parțiale conținând numai informații despre schimbările din topologie. Întreaga tabelă de rutare se va trimite doar în cazurile de stabilire a adiacenței cu un ruter nou. În acest caz, ruter-ul care primește un pachet Hello va răspunde cu încă un pachet Hello urmat imediat de întreaga sa tabelă de rutare (în afară de rutele primite pe aceeași interfață – se aplică strategia Split Horizon). De asemenea, aceste pachete ajung doar la ruterele care au nevoie de această informație – bounded updates, minimizând congestionarea legăturilor.

Pachete EIGRP

- Query
  - folosite de DUAL pentru căutarea de rețele
  - unicast sau multicast
- Reply
  - pachete răspuns
  - numai unicast
- Acknowledgement
  - folosite pentru a confirma pachetele de actualizare, query și reply

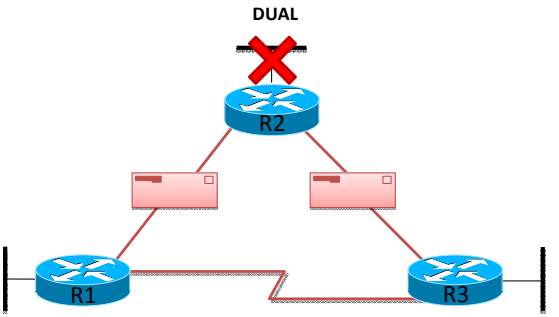


Pachetele Query și Reply oferă lui EIGRP o abordare pro-activă la pierderea unei rute către destinație. Odată picată o rută, EIGRP va trimite mesaje de Query către toți vecinii cu care are adiacență. Când un vecin primește pachetul de Query analizează tabela sa de rutare pentru ruta pierdută de vecinul său. Dacă ruterul nu are rută, este forțat să întrebe mai departe vecinii săi. Astfel, căutarea unei noi rute se propagă în rețea de la ruter la ruter prin fiecare Query. Ruterul care a originat primul Query trebuie să primească într-un interval de 3 minute răspunsuri de la toți vecinii săi, fie acestea răspunsuri negative sau pozitive. Dacă acest lucru nu se întâmplă, ruta intră în starea SIA (Stuck in Active) și administratorul este obligat să intervină pentru a restarta procesul EIGRP.

Pachetele Acknowledgement sunt folosite pentru confirmarea atât a pachetelor de update conținând informații de rutare, cât și a pachetelor de tip query și reply descrise anterior.

DUAL

- Diffusing Update Algorithm
- Folosit pentru prevenirea buclelor de rutare
- Folosește o listă de rute de backup (rutele nu au bucle) – oferă timp rapid de convergență



DUAL (Diffusing Update Algorithm) este algoritmul utilizat de EIGRP pentru găsirea căii cele mai scurte între două noduri și pentru evitarea buclelor de rutare. EIGRP urmărește toate rutele trimise de către toți vecinii și cu ajutorul unei metrici avansate selectează calea optimă către destinație, oferind un timp foarte rapid de convergență. Algoritmul DUAL reține în tabela de topologie a protocolului EIGRP o listă cu toate rutele către orice destinație. Astfel, în cazul în care o anumită legătură este declarată invalidă, DUAL va introduce în tabela de rutare o altă rută calculată anterior (Feasible Successor), însă doar daca această îndeplinește anumite condiții legate de valoarea metricii. Aceasta schimbare este aproape instantanee, EIGRP nefiind nevoit să recalculeze DUAL în cazul în care o astfel de rută este disponibilă în tabela de topologie.

Distanța administrativă

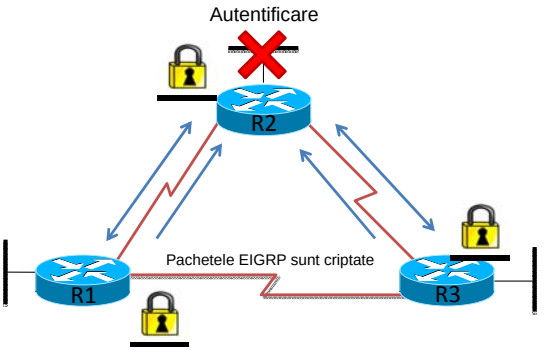
- Încrederea într-un protocol de rutare
- Distanța administrativă are valoare locală si este vendor specific
- Distanțele administrative EIGRP implicite pentru:
  - rutele sumarizate = 5
  - rutele interne = 90
  - rutele importate = 170

Distanța administrativă standard a EIGRP pentru rute interne este 90 – mai mică decât distanțele administrative pentru OSPF, RIP sau ISIS.

Rutele sumarizate EIGRP au distanța administrativă foarte mică (5) deoarece sumarizările sunt configurate manual de administrator. Rutele importate din alte protocoale de rutare cu ajutorul comenzii **`redistribute`** prezintă o încredere scăzută și sunt introduse în procesorul de EIGRP cu o distanță administrativă de 170.

Autentificare

- EIGRP poate:
  - adăuga un hash la sfârșitul update-ului
  - autentifica informațiile de rutare



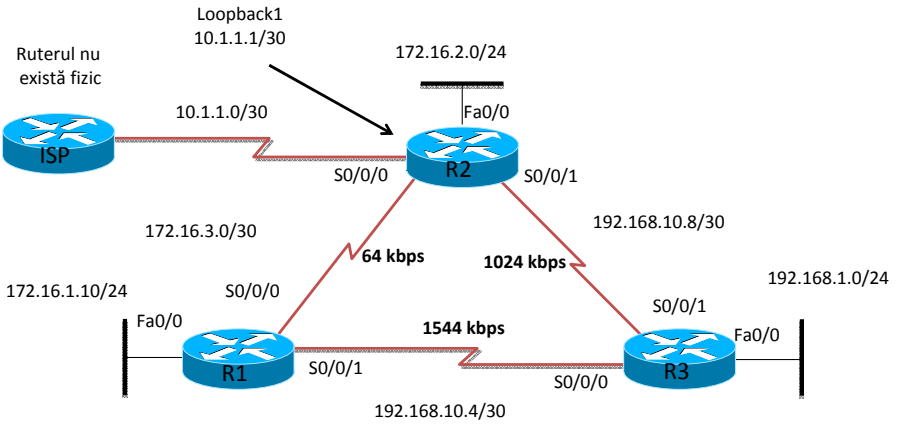
Ca și alte protocoale de rutare, EIGRP suportă autentificarea informațiilor de rutare trimise între rutere. Autentificarea este de două tipuri:

•Simplă (plain-text password authentication); cheia de autentificare este trimisă în update-urile de rutare în plain text, ceea ce o face foarte ușor de interceptat

•MD5 – se generează un hash MD5 al pachetului EIGRP trimis, care este verificat cu cheia privată la destinație; pentru autentificarea cu MD5, un identificator de cheie și o cheie de autentificare trebuie specificate pentru ambele rutere care comunică; fiecare identificator de cheie are atașată o cheie distinctă; mai multe chei pot fi grupate într-un lanț de chei

Sumarizare

- EIGRP va sumariza automat rutele la măștile classful

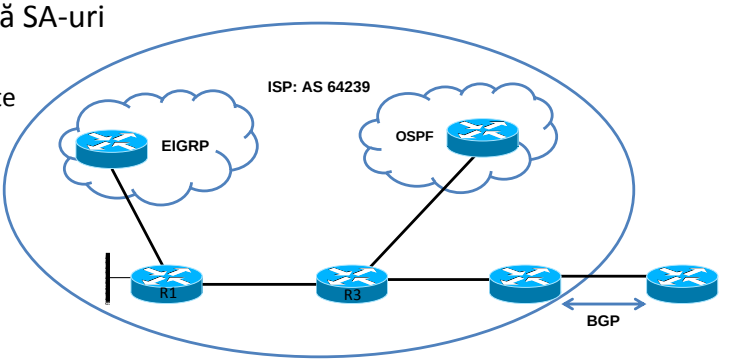


Conform cu comportamentul default al RIPv2, EIGRP va sumariza automat toate rutele la măștile classful. Acest comportament poate avea un efect benefic asupra mărimii tabelei de rutare, a resurselor ruterului și a dimensiunii update-urilor de rutare. Aceasta este o funcție EIGRP activată implicit – cu toate acestea se recomandă dezactivarea acesteia de către administrator pentru facilitarea sumarizării manuale în punctele critice din rețea.

De asemenea, în cazul sumarizării automate pot apărea anumite probleme de conectivitate sau pierdere de pachete în cazul în care se folosește o schemă de adresare de tip VLSM.

Sistemul autonom

- Un sistem autonom este o colecție de rețele aflate sub administrație comună și care prezintă o politică de rutare comună (RFC 1930)
- Atribuite de IANA
- Entități ce necesită SA-uri
  - ISP-uri
  - Instituții conectate la alte instituții ce folosesc numere de SA



Un sistem autonom este un conglomerat de rutere aflate sub aceeași administrație. Capacitatea EIGRP de a suporta multiple sisteme autonome (AS) îl face un protocol extrem de scalabil și folositor în cazul rețelelor de dimensiuni mari și foarte mari. Avantajul utilizării sistemelor autonome este capacitatea de a separa domeniile de rutare – astfel, tabelele ruterelor devin mai mici, se eliberează resurse hardware pentru task-uri mai importante și se decongestionează rețelele de legătură între rutere pentru că nu trebuie să trimită update-uri despre absolut fiecare rețea existentă în topologie.

Numerele de AS globale sunt asignate de către IANA (“ Internet Assigned Numbers Authority” – entitate ce se ocupa global cu alocarea adreselor IP si cu alte atribuții la nivelul Internet Protocol) și pot fi rutate în internet cu ajutorul BGP, acesta fiind singurul protocol capabil să realizeze rutarea între sisteme autonome. Numerele AS globale sunt reprezentate pe 16 biți, în ziua de astăzi existând necesitatea extinderii acestora pe o dimensiune de 32 biți.

Activare EIGRP

- Comanda globală ce activează EIGRP

```
Router(config)#router eigrp autonomous-system
```

- SA-ul funcționează ca un proces ID – o instanță a protocolului de rutare ce rulează pe un router
- Toate router-ele din domeniul EIGRP trebuie să folosească același număr de SA

```
R1(config)#router eigrp 1
R1(config-router)#
```

```
R2(config)#router eigrp 1
R2(config-router)#
```

```
R3(config)#router eigrp 1
R3(config-router)#
```


EIGRP se activează pe un ruter în mod asemănător cu protocolul RIP, singura diferență fiind necesitatea menționării numărului de AS. Este necesar ca fiecare ruter dintr-un anumit EIGRP să aibă același număr AS în cazul care dorim ca acesta să conveargă. Fiecare instanță de EIGRP de pe un ruter funcționează cu un număr de AS separat. Deși din punct de vedere teoretic putem configura mai multe instanțe ale procesului EIGRP pe un ruter, de obicei se configurează unul singur.

Comanda *network*

- Funcțiile comenzii sunt
  - activează interfețele pentru a trimite și primi actualizări EIGRP
  - include rețeaua sau subrețeaua în actualizările EIGRP

```
Router(config-router)#network network-address
```

```
R1(config)#router eigrp 1
R1(config-router)#network 172.16.0.0
R1(config-router)#network 192.168.10.0
```

```
R2(config)#router eigrp 1
R2(config-router)#network 172.16.0.0
%DUAL-5-NBRCHANGE: IP_EIGRP 1: Neighbor 172.16.3.1 (Serial 0/0/0)
is up: new adjancey
```


Comanda **network** specifică rețelele care fac parte din sistemul autonom, indicând ce interfețe vor participa la procesul de EIGRP. Protocolul EIGRP suportă modul de definire al rețelei din RIP – adică specificarea doar a adresei de rețea, caz în care comanda **network** este una classful. Acest lucru poate fi dăunător unui mediu în care se dorește utilizarea VLSM pentru adresare, deoarece comanda dată în acest mod poate cuprinde mai multe rețele decât a fost intenționat.

Soluția classless este utilizarea unui wildcard mask pentru identificarea exacta a rețelelor pe care dorim să le introducem în procesul de EIGRP. Wildcard mask-ul reprezintă „masca de rețea” în care toți biții de 1 vor avea valoarea 0 și toți biții de 0 valoarea 1.

Comanda *network* cu Wildcard Mask

- Opțiune folosită pentru publicarea unor subrețele specifice

```
Router(config-router)#network network-address [wildcard mask]
```

```
R1(config)#router eigrp 1
R1(config-router)#network 172.16.0.0
R1(config-router)#network 192.168.10.0
```

```
R2(config)#router eigrp 1
R2(config-router)#network 172.16.0.0
%DUAL-5-NBRCHANGE: IP_EIGRP 1: Neighbor 172.16.3.1 (Serial 0/0/0)
is up: new adjancey
R2(config-router)#network 192.168.10.8 0.0.0.3
```

```
R3(config)#router eigrp 1
R3(config-router)#network 192.168.10.0
%DUAL-5-NBRCHANGE: IP_EIGRP 1: Neighbor 192.168.10.5 (Serial 0/0/0)
is up: new adjancey
R3(config-router)#
%DUAL-5-NBRCHANGE: IP_EIGRP 1: Neighbor 192.168.10.9 (Serial 0/0/1)
is up: new adjancey
R3(config-router)#network 192.168.1.0
```


În cazul în care rețele care vor fi anunțate în protocolul EIGRP au asociată o mască de rețea classless, pentru identificarea exactă a rețelelor pe care dorim să le introducem în procesul de rutare EIGRP soluția este utilizarea unui „wildcard mask”. „Wildcard mask”-ul reprezintă „masca de rețea” în care toți biții de 1 vor avea valoarea 0 și toți biții de 0, valoarea 1.

Sumarizare EIGRP

- Comanda **`auto-summary`** permite sumarizarea automată la rețeaua classful
- Comanda **`no auto-summary`** dezactivează sumarizarea automată
  - vecinii EIGRP fac schimb de actualizări ce nu vor mai fi sumarizate automat
  - vor apărea schimbări în
    - tabela de rutare
    - tabela de topologie

Protocolul de rutare EIGRP face sumarizare automată în mod implicit la masca de rețea classful a clasei de IP din care face parte rețeaua. EIGRP va adăuga automat o rută sumarizată către null0 în cazul în care sunt îndeplinite una dintre condițiile urmatoare:

- Există cel puțin un subnet învățat prin EIGRP
- Sumarizarea automată este activată

Pentru ca protocolul EIGRP să poată suporta rutarea între rețele cu mască de rețea variabilă se recomandă dezactivarea sumarizării automate cu ajutorul comenzii **`no auto-summary`**. În momentul introducerii acestei comenzi, algoritmul DUAL va reface toate adiacențele și toate ruterele din domeniu de rutare EIGRP vor trimite imediat update-uri cu rutele nesumarizate. Astfel, rutele vor fi instalate cu masca lor reală în tabela de rutare și în tabela de topologie.

Sumarizare manuală

- Poate include supernet-uri
  - EIGRP fiind un protocol de rutare classless poate include masca de rețea în actualizare

```
R3(config)#interface serial 0/0/0
R3(config-if)#ip summary-address eigrp 1 192.168.0.0 255.255.252.0
R3(config-if)#interface serial 0/0/1
R3(config-if)#ip summary-address eigrp 1 192.168.0.0 255.255.252.0
```


EIGRP poate fi configurat să sumarizeze mai multe rute, trimițând update-uri care conțin masca de rețea a supernetului astfel creat.

Sumarizarea în EIGRP se face la nivel de interfață, iar ruta sumarizată este trimisă ca orice altă rută EIGRP internă cu distanța administrativă de 90. Avantajul rutelor sumarizate este micșorarea mărimii tabelei de rutare astfel că procesul de căutare a unei rute se execută mai eficient. De asemenea, se utilizează mai puțin bandwidth pentru trimiterea update-urilor de rutare datorită dimensiunii acestora mai reduse.

„Fine-tuning” EIGRP

- Implicit, EIGRP va folosi până 50% din banda unei legături pentru a trimite mesajele EIGRP
- Pentru a schimba acest comportament, este folosită comanda la nivel de interfață

```
Router(config-if)#ip bandwidth-percent eigrp <as number> <procent>
```

- Pentru configurare intervalelor hello și hold-time

```
Router(config-if)#ip hello-interval eigrp <as number> <seconds>
Router(config-if)#ip hold-time eigrp <as number> <seconds>
```

- EIGRP va funcționa chiar dacă nu sunt setate aceleași timere la ambele capete ale unei legături

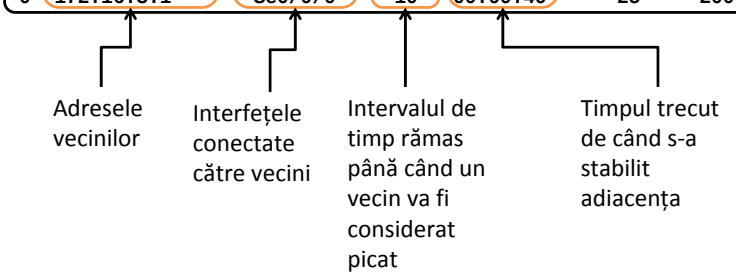
Pentru a calcula calea către o destinație, EIGRP trimite mesaje de Query la ruterele vecine, care se vor propagă apoi în rețea. Acest comportament alături de „flood”-ul de update-uri poate încărca destul de mult lățimea de bandă. De aceea, protocolul EIGRP introduce o limitare a „bandwidth”-ului folosit pe o legătură pentru mesajele sale specifice. Standard se folosește maxim 50% din „bandwidth”-ul disponibil al unei interfețe. Pentru un mai bun management al rețelei, administratorul poate controla valoarea procentului de bandă folosit de EIGRP cu comanda **`ip bandwidth-percent eigrp`** ***`as number procent`*** introduse pe interfață.

Timer-ele hello-interval și hold-time pot fi configurate pe interfață, dar trebuie avut în vedere ca valoarea hold-time să fie mai mare decât cea a hello-interval pentru a nu se pierde adiacența. Spre deosebire de alte protocoale de rutare cum este OSPF-ul, EIGRP va funcționa chiar dacă timerele nu sunt configurate identic la capetele unei legături.

Verificare EIGRP (1)

- Router-ele EIGRP trebuie să stabilească adiacențe cu vecinii lor înainte ca actualizările să fie trimise sau primite
- Comanda folosită pentru verificarea adiacențelor este:

```
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address         Interface   Hold  Uptime     SRTT    RTO   Q     Seq   Type
                                (sec)            (ms)        Cnt    Num
1192.168.10.10    Se0/1       10    00:01:41   20      200   0     7
0  172.16.3.1       Se0/0/0      10    00:09:49     25      200    0      28
```



Verificarea stabilirii adiacenței în protocolul EIGRP se poate efectua din meniul global de configurare cu ajutorul comenzii **`show ip eigrp neighbors`**. Pentru fiecare ruter se observă adresele IP ale ruterelor adiacente, interfețele de legătură cu aceștia și alte informații specifice (hold-down timer – intervalul de timp rămas până când un vecin este considerat inactiv; Uptime – timpul scurs din momentul în care este stabilită adiacența).

Verificare EIGRP (2)

```
R3#show ip protocols
Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Automatic address summarization:
    192.168.10.0/24 for FastEthernet0/0, Serial0/0/0
          Sumarizing with metric 2169856
    172.16.0.0/16 for Serilal0/0/1
          Sumarizing with metric 28160
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.10.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)          90       00:03:29
    192.168.10.6           90       00:02:09
    Gateway         Distance      Last Update
    172.16.3.2             90       00:02:12
  Distance: internal 90 external 170
```


Pentru verificarea activării protocolului EIGRP se va utiliza comanda **`show ip protocols`** care afișează detalii despre fiecare protocol rulat de un anumit ruter. Distanța administrativă a protocolului folosit este de asemenea afisată, în cazul EIGRP, având valoarea 90.

Verificare EIGRP (3)

```
Router#show ip route
```

- Rutele învățate prin EIGRP sunt reprezentate prin litera “D”
- Implicit, EIGRP sumarizează rutele la rețeaua classful

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

   192.168.10.0/24 is variably subnetted, 3 subnets, 2 maks
D    192.168.10.0/24 is a summary, 00:03:50, Null0
C    192.168.10.4/30 is directly connected, Serial0/0/1
D    192.168.10.8/30 [90/2681856] via 192.168.10.6, 00:02:43, Serial0/0/1
<code output omitted>
```


Din output-ul comenzii **`show ip route`** se poate observa introducerea de către EIGRP a unei rute sumarizate către Null0 atunci când este activată sumarizarea automată. În cazul în care pachetul nu face „match” pe una din rutele parinte de nivel 1, pachetul va face „match” pe ruta catre Null0 si astfel pachetul va fi aruncat.

Concepte DUAL (1)

- Succesor – router-ul vecin folosit pentru a trimite pachete către o anumită rețea pe rută de cost minim
- Feasible distance (FD) – cea mai mică metrică posibilă pentru a ajunge la o rețea
  - metrica anunțată de succesor

```
R1#sh ip route
<code output omitted>
   192.168.10.0/24 is variably subnetted, 3 subnets, 2 maks
D     192.168.10.0/24 is a summary, 00:00:50, Null0
D     192.168.10.8/30 [90/2681856] via 192.168.10.6, 00:02:43, Serial0/0/1
C     192.168.10.4/30 is directly connected, Serial0/0/1
   172.16.0.0/16 is variable subnetted, 4 subnets, 3 masks
D     172.16.0.0/16 is a summary, 00:00:15, Null0
D     172.16.1.0/24 [90/40514560] via 172.16.3.1, 00:00:15, Serial0/0/0
C     172.16.2.0/24 is directly connected, FastEthernet0/0
C     172.16.3.0/30 is directly connected, Serial0/0/0
   10.0.0.0/30 is subnetted, 1 subnets
C     10.1.1.0 is direclty connected, Loopback1
D  192.168.1.0/24 [90/3014400] via 192.168.10.10, 00>00>15, Serial0/0/1
                    ↑                  ↑
          feasible distance        succesor
```


Protocolul EIGRP, prin intermediul algoritmului DUAL folosește o metoda complexă dar foarte eficientă și rapid-convergentă pentru determinarea celei mai scurte căi până la fiecare destinație din rețea.

După stabilirea adiacenței între rutere, fiecare ruter începe analizarea propriei sale tabele de topologie în căutarea celei mai bune rute către o anumită destinație. În acest proces, fiecare ruter va determina câte un succesor pentru fiecare rețea destinație – acesta poate fi considerat next hop, și este ruterul folosit pentru a trimite pachete către o anumită rețea pe ruta de cost minim. Feasible Distance reprezintă cea mai mică metrică de la ruterul sursă până la ruterul destinație, iar Reported Distance (RD) reprezintă cea mai mică distanță de la ruterul vecin până la destinația dorită.

Când se găsește acest succesor, el este automat instalat în tabela de rutare, având metrica egală cu Feasible Distance. Pot exista până la 4 succesori cu FD egala – EIGRP va realiza equal-cost load balancing.

Concepte DUAL (2)

- Feasible succesor (FS) – ruter vecin ce îndeplinește condiția de succesor viabil
  - loop free backup rute către destinația succesorului
  - acest router este ales automat ca succesor dacă actualul succesor pică
- Reported distance (RD) – metrica raportată de un ruter vecinului său despre costul său către o rețea
- Condiția de succesor viabil
  - RD-ul unui vecin este mai mic decât FD-ul ruter-ului local către aceeași rețea destinație

Ruterul care rulează EIGRP, având numeroase date despre topologie conținute în tabela de topologie, calculează rute de back-up care să ia locul celor instalate în tabela de rutare în cazul în care ruta principală devine neviabilă. Pe lângă succesor, care este calculat ținând cont de cel mai mic Feasible Distance, se încearcă calcularea unui Feasible Succesor care ar putea lua locul rutei instalate momentan în tabela de rutare. Condiția ca o rută din tabela de topologie să devină Feasible Succesor este ca aceasta să aibă Reported Distance (distanța de la ruterul vecin până la destinație) mai mică decât Feasible Distance a succesorului curent instalat în tabela de rutare. Acest proces asigură, în cazul în care există, alegerea unei rute fără bucle de rutare.

Concepte DUAL (3)

- Comanda *show ip eigrp topology* afișează:
  - succesorii
  - succesorii viabili
- O rută poate fi în două stadii:
  - P Passive – ruta este bună şi funcţionează normal
  - A Active – ruta este în procedeul de recalculare dictat de DUAL

```
R2#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

<output omitted>
P 192.168.10.0/24, 1 successors, FD is 3014400
          via 192.168.10.10 (3014400/28160), Serial0/0/1
          via 172.16.3.1 (41026560/2172416), Serial0/0/0
P 192.168.10.8/30, 1 succesors, FD is 3011840
          via Connected, Serial0/1
```


Tabela de topologie se poate afișa folosind comanda **`show ip eigrp neighbors`**. În output este menționat sistemul autonom în cadrul căruia rulează procesul de EIGRP, dar și identificatorul ruterului pe care a fost introdusă comanda. Identificatorul este ales în felul următor: cea mai mare adresă IP de pe interfețele de loopback sau, în caz că nu există configurate interfețe de loopback, se va alege cea mai mare adresă dintre interfețele fizice active.

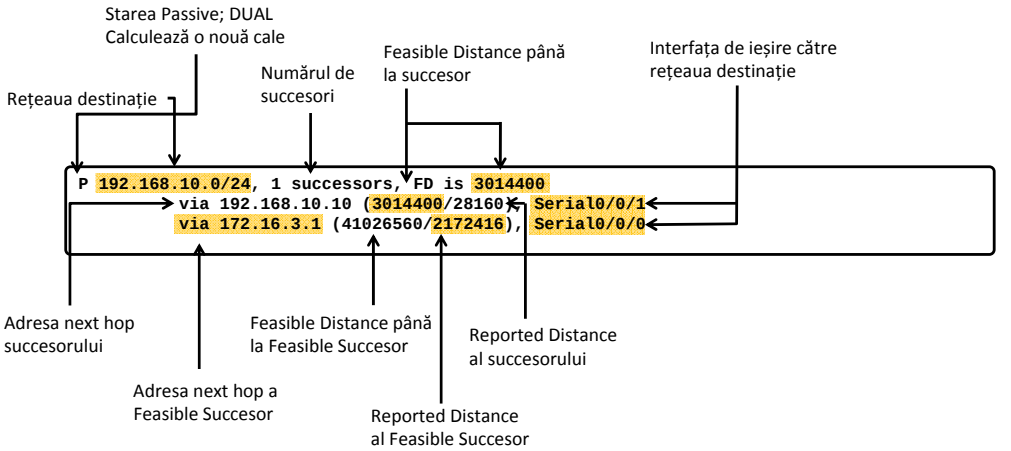
În prima coloană a informațiilor afișate în tabela de topologie se specifică starea rutei respective, care poate fi:

•P (Passive) – rețeaua este accesibilă și astfel ruta se poate instala în tabela de rutare

•A (Active) – rețeaua nu este accesibilă și nu există un succesor valabil de „back-up”, caz în care se reia algoritmul DUAL pentru recalcularea unei rute optime

Concepte DUAL (4)

- Tabela de topologie EIGRP:



În cadrul tabelei de topologie EIGRP, pentru fiecare destinație menționată sunt indicate următoarele informații:

- Numărul de succesori
- Valoarea Feasible Distance către succesor
- Adresa ruterului next hop
- Valoarea Feasible Distance către next hop
- Reported Distance al ruterului next hop către destinație
- Interfața de ieșire pentru a accesa rețeaua destinație

Când un ruter învată mai multe rute către o destinație folosind multiple protocoale de rutare, pune în tabela de rutare ruta cu distanța administrativă cea mai mică. În cazul în care avem mai multe rute ale aceluiași protocol cu distanța administrativă egala, ruterul alege calea având costul(sau metrica) cea mai mică.

Fiecare protocol calculeaza costul diferit si astfel avem nevoie de un

Concepte DUAL (5)

- Există posibilitatea să nu existe nici un "feasible successor"
  - motiv: condiţia de viabilitate nu a fost îndeplinită

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(192.168.10.5)
<output omitted>
P 192.168.10.0/24, 1 successors, FD is 2169856
          via Surrary (2169856/0), Null0
P 192.168.10.4/30, 1 successors, FD is 2169856
          via Connected, Serial0/0/1
P 192.168.1.0/24, 1 sucessors, FD is 2172416
          via 192.168.10.6 (2172416/28160), Serial0/0/1
P 192.168.10.8/30, 1 successors, FD is 3523840
          via 192.168.10.6 (3523840/3011840), Serial0/0/1
<output omitted>
```

```
R1#show ip eigrp topology
<output omitted>
P 192.168.1.0/24, 1 sucessors, FD is 2172416, aerno 5
          via 192.168.10.6 (2172416/28160), Serial0/0/1
          via 172.16.3.2 (41026560/3014400), Serial0/0/0
<output omitted>
```


În ciuda complexității algoritmului DUAL, există posibilitatea să nu se găsească un "feasible successor" pentru destinația căutată, din cauza că nu este fizic posibil sau nu îndeplinește condiția de fezabilitate. În acest caz, ruterul care rulează algoritmul DUAL va trimite un „Query multicast” celorlalte rutere din procesul de EIGRP cerând informații suplimentare despre calea spre acea rețea destinație, reluând algoritmul DUAL. Răspunsul primit poate fi pozitiv sau negativ, însă dacă nu se primește un răspuns în termen de 3 minute, există pericolul ca acest ruter să rămână în starea SIA – „Stuck in Active”, adică nu a primit răspuns la o cerere de la un vecin pe o ruta și elimina acel vecin – caz în care este necesară repornirea manuală a procesului EIGRP.

Rezumat

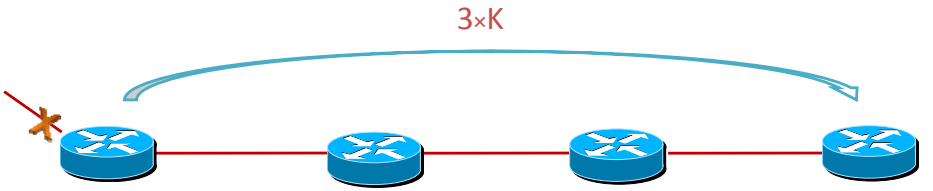
- Header-ul EIGRP
- Tipuri de pachete EIGRP
- DUAL
- Configurarea EIGRP

1. Care este rolul folosirii pachetelor de tip Hello?
2. Care este algoritmul care stă la baza funcționării EIGRP și care sunt parametrii folosiți pentru calcularea metricii?
3. Care sunt cele două tipuri de autentificare a informațiilor de rutare?
4. Ce principiu este folosit pentru calcularea unei rute de back-up pentru rutele existente în tabela de rutare populată folosind EIGRP?
5. Este necesar ca timerele configurate pe o legătură EIGRP să fie identice la ambele capete pentru stabilirea adiacenței ?

Protocoale link-state. OSPF single area

Limitările DV

- Scalabilitate
  - peste câte hopuri poate RIP să transmită un update de rutare?
- Convergență
- Hop-by-hop routing
- EIGRP ?



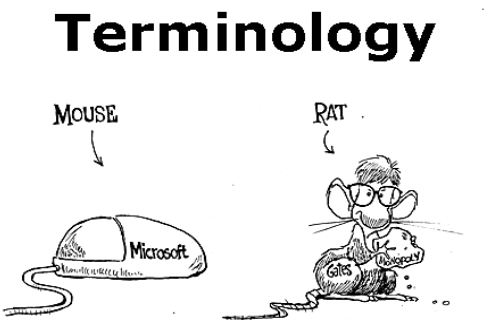
K = interval de trimire a update-urilor

Protocoalele de rutare Distance Vector (DV) sunt cele mai simple protocoale de rutare – nu au cunoștințe despre rețeaua din jurul lor, singura informație pe care o dețin fiind următorul hop spre care să trimită un anumit pachet. Practic, spre deosebire de protocoalele Link-State, cele Distance Vector nu au tabele speciale în care să rețină informații despre întreaga topologie aflată în domeniul de rutare – astfel, ele au un timp de convergență mai mare, în special din cauza faptului că nu se rețin rute care să servească drept back-up în cazul în care ruta principală deja instalată devine indisponibilă. De asemenea, pachetele cu update-uri de rutare se trimit la un anumit interval (30 secunde pentru RIP), astfel încât ar dura aproape două minute ca o rută să se propage peste 4 rutere.

Protocoalele principale Distance-Vector sunt: RIPv1, RIPv2 și IGRP. Deși în anumite publicații EIGRP este considerat un protocol Distance Vector, el este numit oficial de către Cisco un protocol hibrid.

Protocoale link-state (1)

- Link = interfață a unui ruter
- Link-state = informația despre starea link-urilor
- Cea mai bună rută se alege prin costuri acumulate de la sursă la destinație

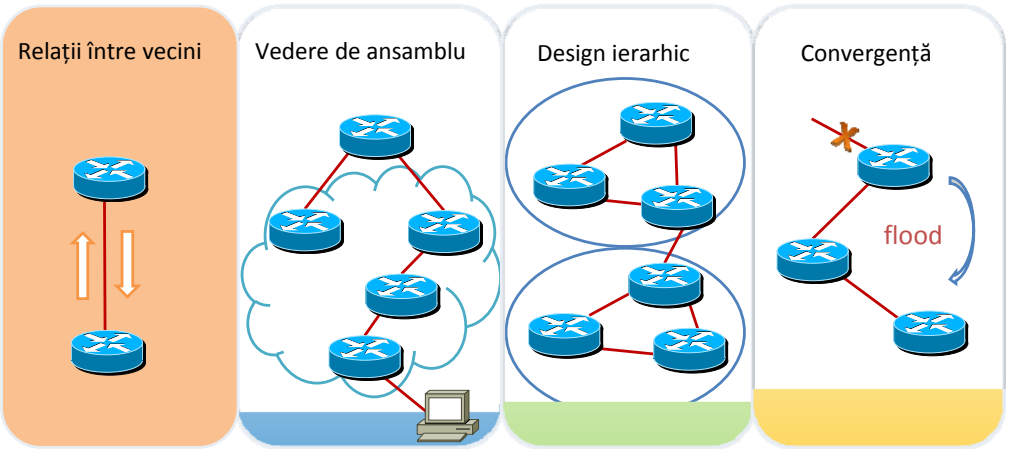


Protocoalele de rutare link-state folosesc diverși factori pentru a calcula metrica spre o anumită destinație. De exemplu, OSPF folosește pentru calcularea metricii o formulă care ia în considerare doar viteza teoretică a unui anumit link, ignorând ceilalți factori care caracterizează o anumită conexiune. O altă caracteristică a protocoalelor link-state este păstrarea legăturii – asemenea protocolului EIGRP, protocoalele link-state au implementate mecanisme pentru a se asigura că vecinii direct conectați încă sunt porniți și rulează corect instanța respectivă a protocolului. Astfel, se asigură un răspuns rapid și eficient în cazul diferitelor probleme apărute în rețea.

Pentru a se asigura funcționarea corectă a protocoalelor de rutare link-state, ca și în cazul celor distance-vector, un ruter trebuie să aibă configurată cel puțin o interfață cu o adresă IP împreună cu masca de rețea atașată. De asemenea, starea logică a interfeței trebuie să fie up.

Protocoale link-state (2)

- Concepte introduse



Principalele concepte și idei introduse în protocoalele link-state:

•Relații între vecini - două rutere care rulează un protocol link-state și se află în același domeniu de rutare pot stabili o relație de adiacență; stabilirea adiacenței este necesară pentru ca ruterele vecine să poată schimba informații despre rețelele cunoscute

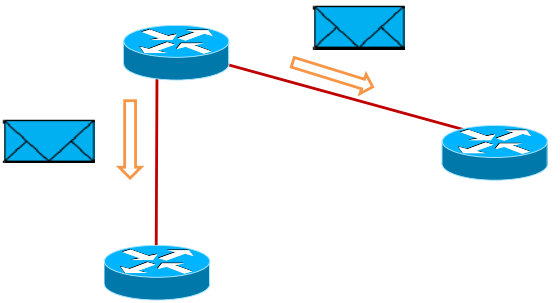
•Vedere de ansamblu - cu ajutorul unui algoritm specific, fiecare ruter realizează o vedere de ansamblu asupra întregii topologii

•Design ierarhic – protocoale de tip link-state cum sunt OSPF și IS-IS utilizează conceptul de divizare a topologiei în domenii separate, numite arii; astfel, este eficientizat procesul de rutare prin utilizarea de rute sumarizate, dar și prin posibilitatea izolării rapide a unei probleme apărute într-o arie

•Convergență – transmiterea imediată către vecini a pachetelor link-state (LSP) asigură o viteză de convergență mărită

Relație între vecini

- Adiacență
- Protocol de Hello
  - are funcție de keep-alive
- Tabelă de vecini

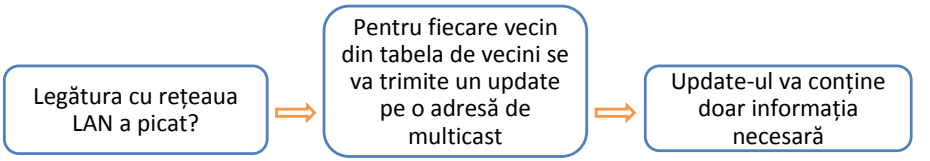


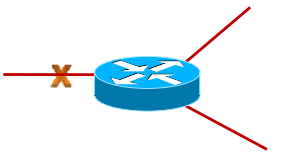
Ruterele care rulează un protocol de rutare Link-State trebuie să realizeze adiacențe cu vecinii înainte de a schimba informații de rutare. Acest lucru se realizează cu ajutorul pachetelor „Hello” – ele sunt cele cu ajutorul cărora diferitele procese de rutare realizează conexiunea între acestea, și apoi o mențin. Pachetele „Hello” se trimit la un interval regulat din ambele părți ale conexiunii; în cazul în care un ruter nu primește nici un pachet „Hello” o anumită perioadă de timp, atunci declară acea legatură invalidă și pornește procesul de scoatere a acesteia din tabela de rutare.

Fiecare ruter care rulează un protocol Link-State își realizează propria sa tabelă de vecini, în care reține informații relevante despre toate ruterele care sunt direct conectate și rulează un proces de rutare compatibil. Informațiile includ adresa IP a vecinului, starea legăturii și timpul rămas până când legatura va fi declarată invalidă.

Convergență

- Triggered updates -> convergență foarte bună





- Flapping interface?

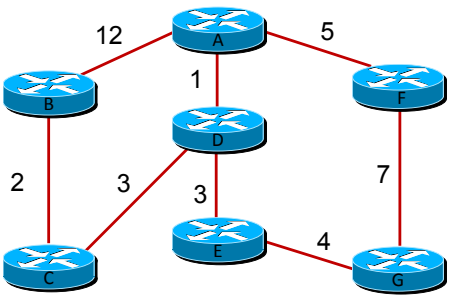
Rețelele care utilizează un protocol de rutare link-state converg mult mai rapid decât cele care utilizează un protocol distance-vector. Avantajul constă în faptul că update-urile despre rețelele noi sau rețelele care au devenit inaccesibile sunt trimise imediat ce evenimentul are loc. Adresa pe care se trimit aceste update-uri este o adresă de multicast, astfel încât doar ruterele care rulează protocolul respectiv de rutare primesc informațiile. O altă caracteristică a update-urilor de rutare trimise de protocoalele link-state este faptul că includ doar informația necesară (modificările din topologie) și nu întreaga tabelă de rutare a echipamentului respectiv.

Spre deosebire de unele protocoale de rutare distance-vector, cele link-state nu trimit update-uri de rutare periodice. După stabilirea adiacenței între vecini și schimbarea informațiilor de rutare, actualizările pentru protocoalele link-state se fac doar în momentul apariției unei schimbări în topologie.

SPF

- Fiecare ruter realizează un arbore în vârful căruia se pune pe el însuși
- Djikstra

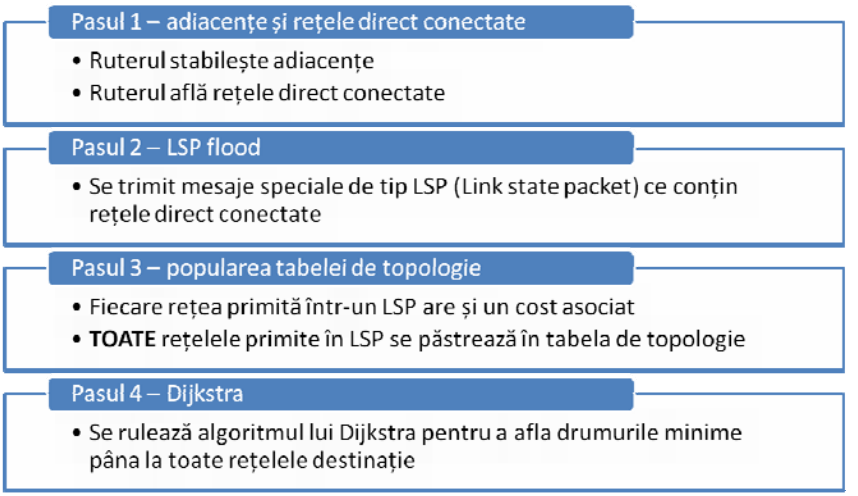
B	Dist = **6** (prin **C**)
C	Dist = **4** (prin **D**)
D	Dist = **1** (prin **A**)
E	Dist = **4** (prin **D**)
F	Dist = **5** (prin **A**)
G	Dist = **8** (prin **E**)



Algoritmul lui Dijsktra este denumit și algoritmul SPF (Shortest Path First). Algoritmul adună costurile de-a lungul unei căi, determinând drumul de cost minim de la o sursă la fiecare destinație. Arborii astfel generați de fiecare ruter din topologie oferă posibilitatea construirii unei hărți cu drumuri optime între oricare două puncte ale rețelei.

În figura de sus, fiecare cale are o valoare aleasă aleator pentru cost. Costul cel mai mic al unei căi de la A la C este 4 (A -> D -> C). Se observă că 4 nu este costul unic al unei căi de la orice ruter din topologie până la C. Fiecare ruter își calculează propria cale până la orice ruter din topologie. Altfel spus, fiecare ruter calculează costul din propriul punct de vedere, aplicând algoritmul SPF (Dijkstra).

Construirea arborelui SPF



Pentru costruirea arborelui SPF o topologie trece prin mai multe etape:

•Adiacența și rețelele direct conectate: pentru a putea schimba update-uri, ruterele trebuie să realizeze înainte o relație de adiacență; ruterul află rețelele direct conectate și construiește tabela de vecini

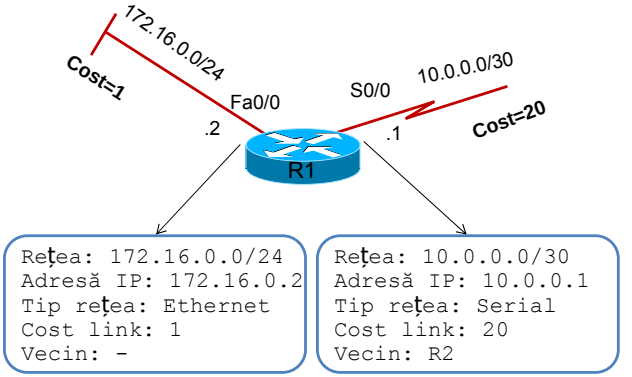
•LSP Flood: LSP (Link-State Packet) sunt pachetele utilizate de un protocol Link-State pentru a schimba informații de rutare; primul pas în construirea arborelui de SPF este trimiterea unor astfel de pachete care conțin informații doar despre rețelele direct conectate

•Popularea tabelei de topologie: cu ajutorul pachetelor LSP primite, fiecare ruter își populează propria tabelă de topologie, utilizând costurile asociate fiecărei legături primite

•Dijkstra: după ce toate ruterele și-au format propria tabelă de topologie a rețelei, se aplică algoritmul Dijkstra pentru calcularea drumului minim între oricare două puncte din topologie

LSP

- Link-state Packet
  - ID vecin
  - tipul de link
  - bandwidth link
  - starea link-ului



LSP (link-state packets) sunt pachetele utilizate de protocoalele de rutare link-state pentru a comunica între ele. Acest pachet este construit de fiecare ruter și conține informații despre starea fiecărui vecin direct conectat. Aceste informații sunt:

•ID Vecin: un cod de identificare unic pentru fiecare nod din topologie

•Tipul de link: se specifică tipul link-ului către vecin. De exemplu, Ethernet, Serial ș.a.

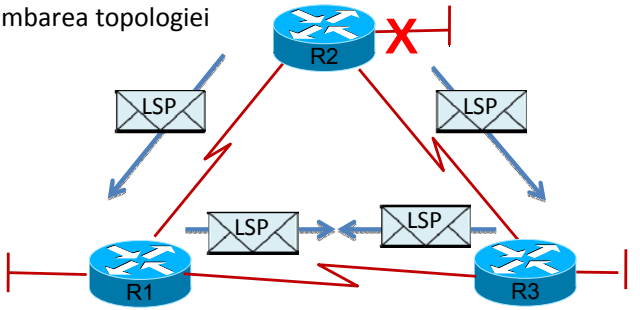
•IP-ul vecinului: adresa IP a ruterului direct conectat

•Bandwidth-ul legăturii: valoarea care reprezintă viteza teoretică a unei anumite legături; aceasta nu este neapărat viteza reală a legăturii, putând fi modificată oricând la nivel de interfață cu ajutorul comenzii **`bandwidth`** urmată de un parametru numeric

•Starea link-ului: se specifică dacă link-ul este „up” sau „down”

Flooding LSPs

- LSP se trimit numai:
  - la inițializarea router-ului sau a procesului de rutare
  - la schimbarea topologiei



- Imediat ce un ruter a primit un LSP, acesta face flood cu LSP-ul pe toate celelalte interfețe

Există două tipuri de scenarii în care un ruter trimite pachete LSP pe interfețele configurate cu un protocol de rutare link-state:

•Atunci când se pornește procesul de rutare, ruterul va trimite informații despre rețelele direct conectate pe toate interfețele sale

•Când intervine o schimbare în topologie, ruterul va trimite un LSP care conține doar informații despre schimbarea în cauză

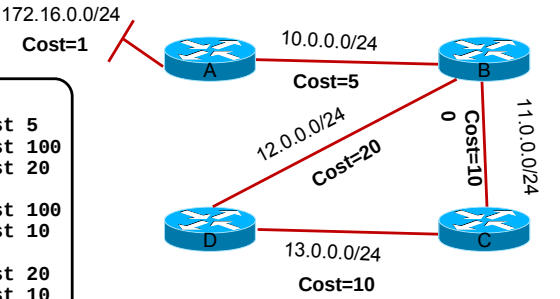
Atunci când un ruter primește un LSP de la unul dintre vecinii săi, îl va trimite imediat pe toate interfețele sale, în afară de interfața pe care l-a primit. Astfel, schimbarea semnalată în acel LSP va fi propagată la toate ruterele din domeniu.

Când un ruter primește un LSP, acesta stochează imediat informațiile în baza sa de date și anume în tabela proprie de topologie.

Construirea bazei de date

- Link-state database
  - conține toate LSP primite de la ruterele ce rulează același protocol de rutare
  - pe baza lor se calculează arborele SPF
- Database pentru A:

```
LSP de la B:
• Conectat cu A la rețeaua 10.0.0.0/24, cost 5
• Conectat cu C la rețeaua 11.0.0.0/24, cost 100
• Conectat cu D la rețeaua 12.0.0.0/24, cost 20
LSP de la C:
• Conectat cu B la rețeaua 11.0.0.0/24, cost 100
• Conectat cu D la rețeaua 13.0.0.0/24, cost 10
LSP de la D:
• Conectat cu B la rețeaua 12.0.0.0/24, cost 20
• Conectat cu C la rețeaua 13.0.0.0/24, cost 10
Link-states A:
• Conectat cu B la re'eaua 10.0.0.0/24, cost 5
•  Are rețeaua 172.16.0.0/24, cost 1
```



În momentul în care un ruter primește un LSP de la un ruter vecin, îl va memora imediat în tabela sa de topologie (Link-state Database). Rutele sunt ierarhizate conform costului acestora, fiind preferate rutele cu costul cel mai redus.

Apoi, va aplica algoritmul SPF pe datele din tabela de topologie, pentru a calcula care este calea cea mai scurtă către fiecare destinație din rețea. Arborele astfel rezultat reprezintă o hartă a topologiei cu ajutorul căreia un ruter poate determina calea optimă între oricare două puncte din domeniul de rutare. În cazul defectării unei legături sau a unui ruter din topologie, protocolul link-state va găsi rapid o cale de „backup" datorită faptului că are cunoștință de întreaga hartă a topologiei.

Avantaje și dezavantaje (1)

Avantaje	Dezavantaje
- vedere unitară asupra rețelei - convergență bună - scalabilitate: protocoalele link-state utilizează un model ierarhic – ușurință în agregarea rutelor - triggered updates	- necesită un grad de competență mai mare al administratorului de rețea - consum de memorie - consum mare de procesor - consum de lățime de bandă

- Fred Brooks: *There is no silver bullet*

Protocoalele de rutare link-state au numeroase avantaje față de cele distance-vector, însă și unele dezavantaje datorate complexității de procesare a algortimului SPF. Printre avantajele principale se numară:

•Fiecare ruter din interiorul unui domeniu de rutare în care rulează un protocol link-state are o vedere unitară asupra rețelei; acest lucru se datorează faptului că fiecare ruter aplică algoritmul SPF pe propria tabelă de topologie, obținând calea cea mai eficientă către fiecare destinație

•Datorită faptului că LSP-urile se trimit numai în cazul unei modificări de topologie (căderea unei legături, adăugarea unei noi rute în procesul de rutare etc.) rețeaua converge mult mai repede; de asemenea, datorită procesului de LSP flooding, orice schimbare de topologie este propagată foarte rapid către toate ruterele participante în domeniul protocolului link-state

Avantaje și dezavantaje (2)

Avantaje	Dezavantaje
- vedere unitară asupra rețelei - convergență bună - scalabilitate: protocoalele link-state utilizează un model ierarhic – ușurință în agregarea rutelor - triggered updates	- necesită un grad de competență mai mare al administratorului de rețea - consum de memorie - consum mare de procesor - consum de lățime de bandă

- Fred Brooks: *There is no silver bullet*

- Cum protocoalele de rutare link-state au o filosofie de funcționare ierarhică, management-ul rutelor devine mult mai ușor și mai eficient; astfel, se pot izola numeroase probleme (bucle de rutare sau erori în topologie) doar într-o anumită arie

Dezavantaje:

- Datorită complexității algoritmului SPF și a mecanismelor de „keep-alive” (menținere a legăturii), protocoalele link-state consumă mult mai multe resurse ale ruterului (memorie, CPU) și ale rețelei (bandwitdh, congestionare) în comparație cu rutarea statică sau un protocol distance-vector
- Configurarea corectă și eficientă a unui protocol de rutare link-state în interiorul unei rețele necesită un grad mai mare de competență și mai mult efort din partea administratorului de rețea față de metodele de rutare discutate în prezentările anterioare

Rezumat

- Protocoale Link-State
- SPF

1. Care sunt deosebirile principale dintre un protocol de rutare link-state în comparație cu un protocol de rutare distance-vector?
2. Care sunt diferențele transmiterii pachetelor de Update OSPF într-o rețea multiaccess în comparație cu o rețea punct-la-punct?
3. Ce algoritm stă la baza calculării drumului de cost minim de la un ruter spre toate celălalte rutere din domeniu ?
4. Care sunt criteriile de alegere a „Router ID”-ului ?
5. Care sunt pașii folosiții pentru construirea arborelui SPF?

Protocoale link-state. OSPF single area

OSPF

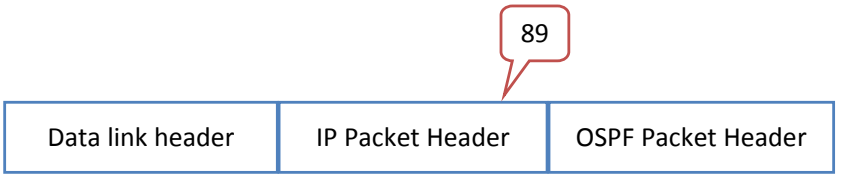
- Open Shortest Path First
- Cel mai popular protocol link state
- Open Source
- Suport IPV6

OSPF este protocol de rutare de tip link-state și reprezintă cea mai răspândită implementare a acestei categorii. Este un **protocol deschis**, standardizat în 1998, bazat, în versiunea curentă pentru IPv4, pe RFC 2328, iar puterea și scalabilitatea sa îl recomandă pentru administrarea rutării în cadrul unor rețele mai complexe. În anul 2008, protocolul a fost reimplementat ca OSPF versiunea 3, introducând schimbări semnificative pentru adaptarea IPv6 (RFC 5340). OSPF utilizează **algoritmul Dijkstra** pentru a găsi cea mai scurtă cale de la un ruter până la celelalte destinații din rețea. Algoritmul rulează în paralel pe fiecare ruter și, astfel, creează un arbore cu cele mai scurte căi. Pentru a putea determina acest arbore, trebuie să fie definit un mod de evaluare a costului unei legături, în cazul OSPF acesta fiind doar lățimea de bandă.

OSPF este un protocol de rutare avansat, utilizând mecanisme de keep-alive (pachete Hello) și metode de reducere a bandwidth-ului utilizat pentru prin reducerea numărului de pachete trimise în rețea (Triggered Updates).

Protocolul OSPF

- Nu are încapsulare de nivel 4
- Sistem reliable de trimitere a mesajelor
- Protocol 89 în campul IP
- Distanță administrativă: 110
- Folosește multicast: 224.0.0.5 – all OSPF routers



Pachetele OSPF nu au încapsulare de nivel 4, acestea ajungând până la nivelul 3 al stivei OSI (Network Layer). Pachete OSPF sunt trimise reliable, primirea lor fiind confirmată printr-un pachet special numit LASack. Un packet al OSPF este format din:

•Header Data Link (Nivel 2): conține informații despre adresa MAC destinație spre care trebuie să se trimită pachetul respectiv; aceasta este de tip multicast: 01-00-5E-00-00-05 sau 01-00-5E-00-00-06

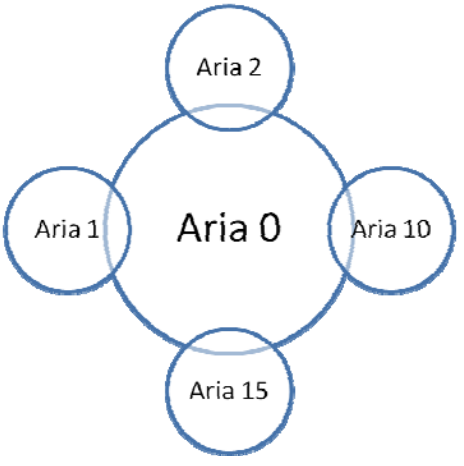
•Header IP: se folosește protocolul 89 în câmpul IP; adresa destinație poate fi reprezentată de 2 adrese IP multicast: 224.0.0.5 și 224.0.0.6

•Header pachet OSPF: include tipul de pachet OSPF - Hello, Database Description, Link-state Request, Link-state Update, Link-State Acknowledgment; include și ID-uri unice pentru ruter și arie

Protocolul OSPF are o distanță administrativă de 110, însemnând că EIGRP (proprietar CISCO, AD de 90) va fi preferat în tabela de rutare.

Scalabilitatea OSPF

- Conceptul de segmentare pe arii



Avantaj: algoritmul SPF se rulează la nivel de arie

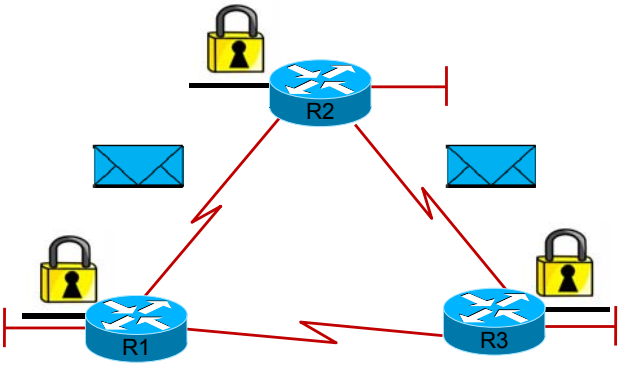
Aria 0 trebuie conectată la toate celelalte arii

Protocolul OSPF folosește un model de rutare bazat pe arii – astfel, putem separa diferite domenii de rutare pentru a evita congestionarea rețelei și supraîncărcarea bazei de date OSPF a fiecărui ruter.

Există în terminologia OSPF conceptul de Backbone Area (Area 0), aceasta fiind aria care va avea întotdeauna conectivitate cu toate celelalte arii. De obicei, în aria 0 se află echipamente foarte performante care primesc toate rutele din celelalte arii și sunt capabile să realizeze conexiunea între acestea. Totuși, procesul de rutare din ariile diferite de aria 0 rămâne izolat; algoritmul SPF de determinare a arborelui optim de acoperire se rulează separat pe fiecare arie și astfel nu se comunică, de exemplu, rute între aria 1 și aria 2 fără intervenția administratorului și a unor configurații suplimentare la nivelul echipamentelor din aria 0.

Autentificare

- OSPF poate
  - cripta informațiile de rutare
  - autentifica informațiile de rutare

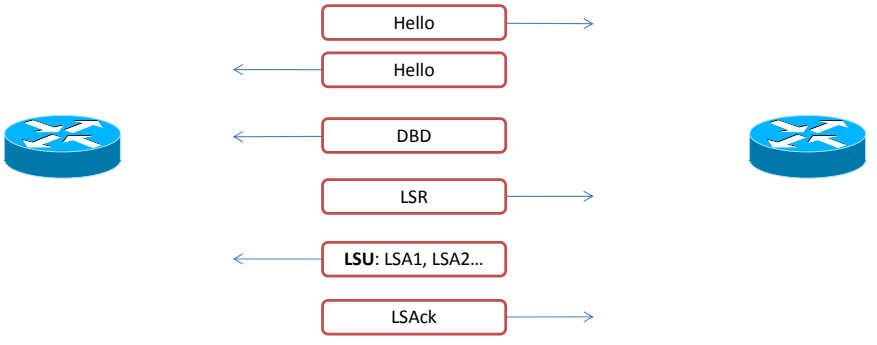


OSPF este un protocol de rutare avansat și astfel suportă autentificarea informațiilor de rutare trimise între rutere. Aceasta poate fi de două tipuri: autentificare simplă cu parolă (numită și plain-text authentication) și autentificare MD5 (care presupune criptarea parolei de autentificare pentru ca aceasta să nu poată fi interpretată de un posibil atacator).

În cazul autentificării MD5, ruterul va genera un „message-digest” (numit și „hash”) pentru fiecare pachet OSPF în parte, această informație fiind adăugată LSA-ului. Este important de reținut că ambele capete ale unei legături care are configurată autentificare trebuie să aibă aceeași parolă pentru ca cele două rutere să poată comunica între ele.

Mesaje OSPF

- În antetul OSPF există un câmp **type** ce codifică fiecare mesaj



Mesajele utilizate de protocolul OSPF sunt de mai multe tipuri, în funcție de scopul fiecăruia:

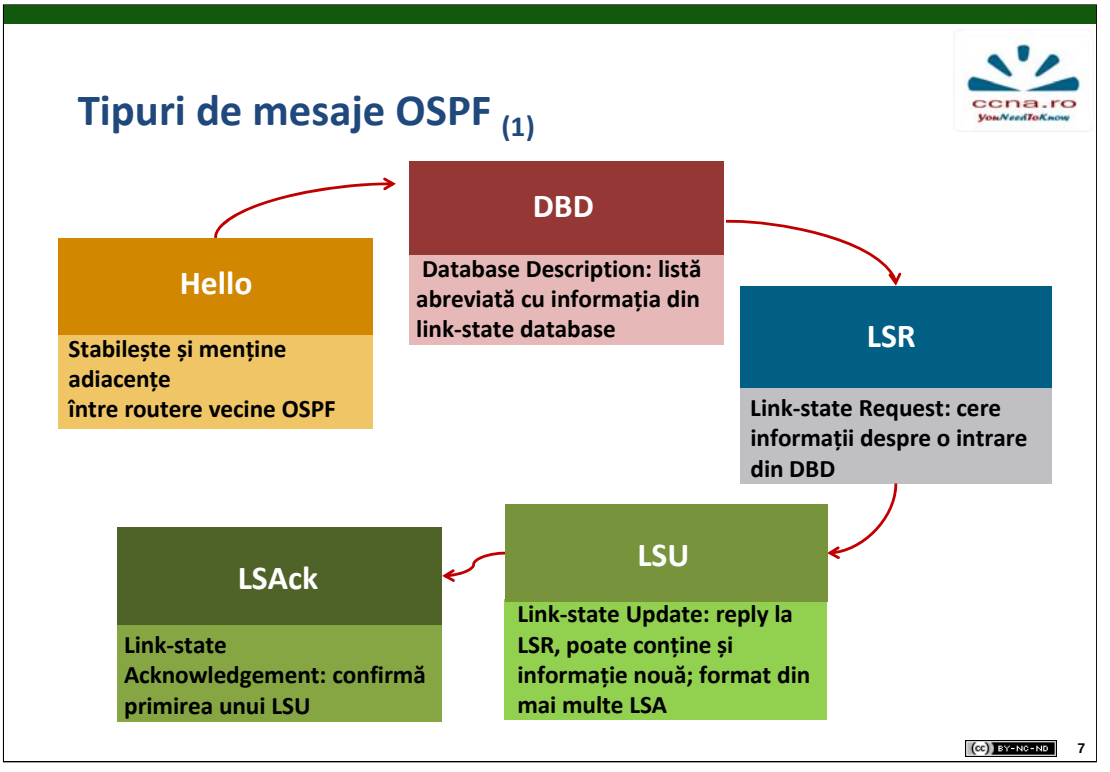
•Pachete „Hello” - folosite pentru a descoperi și a păstra legătura cu vecinii care rulează același proces de rutare

•Database Description - reprezintă o prezentare succintă a tabelei topologice a unui ruter, incluzând lista tuturor rutelor cunoscute și ultimul număr de secvența pentru fiecare

•Link-State Request – reprezintă tipul de LSU (Link-State Update) solicitat și ID-ul ruter-ului care dispune de informație

•Link-State Update – reprezintă pachetul trimis ca răspuns unor cereri de tip LSR

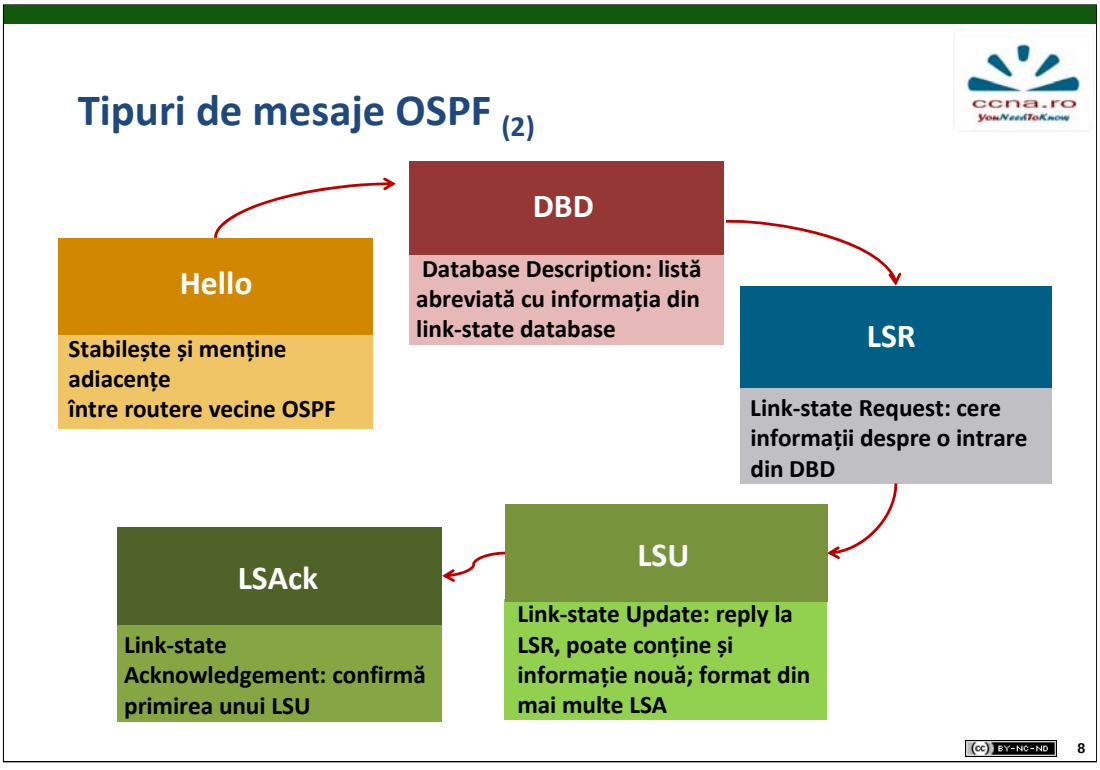
•LSA acknowledgement – pachetul de confirmare trimis de un ruter la primirea altor tipuri de pachete



Fiecare tip de pachet al protocolului OSPF are rolul de a asigura funcționarea optimă a acestuia după cum urmează:

•Pachetele „Hello” sunt utilizate pentru descoperirea și menținerea relației bidirecționale între ruterele vecine. Ruterele care rulează OSPF trebuie să recunoască și să identifice vecinii înainte de a schimba informații despre rute; fiecare interfață care participă la OSPF va trimite pachete de „Hello” la un interval de timp regulat (standard 10 secunde pe mediile multi-access și point-to-point) către adresa multicast 224.0.0.5 (All OSPF Routers)

•Pachete Database Description – pachete speciale care conțin o listă abreviată a Link-State Database a ruterului și care se trimit doar în fazele incipiente ale stabilirii adiacenței



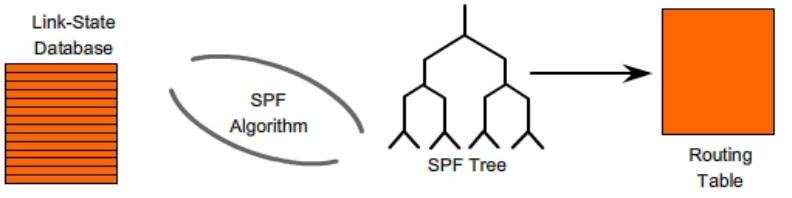
- Link-State Request – este un pachet utilizat de un ruter pentru a afla mai multe informații despre un anumit link învățat în urma unui pachet DBD (Database Description). Cum pachetele DBD conțin informații sumare despre vecinii fiecărui ruter, se trimite acest pachet LSR pentru solicitarea unor informații suplimentare despre o anumită rută.
- Link-State Update – este trimis ca un răspuns la LSR și conține mai multe LSA incluzând informații complete despre ruta solicitată. Aceste LSA-uri pot fi de mai multe tipuri, în funcție de tipul de informație transmisă (rute din aceeași rețea, rute din alte arii etc.)
- Link-State Acknowledgment – sunt trimise cu rol de confirmare pentru celelalte tipuri de pachete primite; datorită acestui tip de pachet schimbul de pachete între vecinii care rulează protocolul OSPF poate fi considerat „reliable”

_____

_____

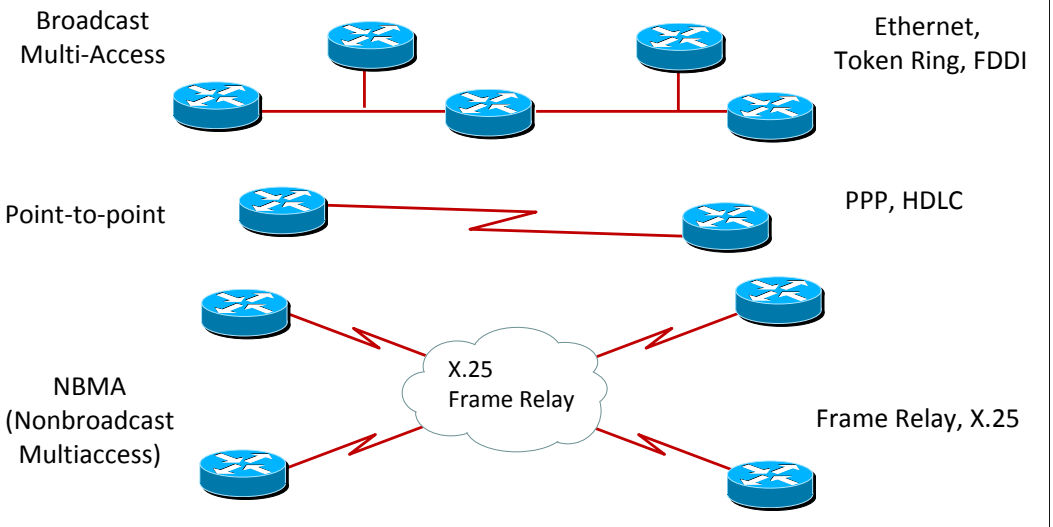
Algoritmul OSPF

- Fiecare ruter OSPF are propria Link-state Database conținând LSA-urile primite de la vecini
- OSPF folosește algoritmul SPF Dijkstra pe Link-state Database pentru a crea arborele SPF având ca rădăcină ruterul
- Cele mai bune căi astfel aflate sunt puse în tabela de rutare



Protocolul OSPF utilizează algoritmul Dijkstra pentru a determină calea optimă de la orice ruter dintr-o rețea către orice destinație din domeniu. Fiecare ruter utilizează informațiile din baza sa de date proprie (Link-State Database - alcătuită în urma primirii de pachete DBD și Link-State Request) pentru a determina independent un drum de cost minim de la acesta până la fiecare ruter din domeniul de rutare. Astfel, se alcătuiește un arbore de cost minim (SPF tree) și se asigură că „hop”-urile generate pentru fiecare rețea destinație reprezintă căile optime. Rutele noi calculate sunt apoi instalate în tabela de rutare.

Tipuri de rețele OSPF (1)



Rețelele în care rulează OSPF sunt de mai multe tipuri, în funcție de topologia de nivel logic:

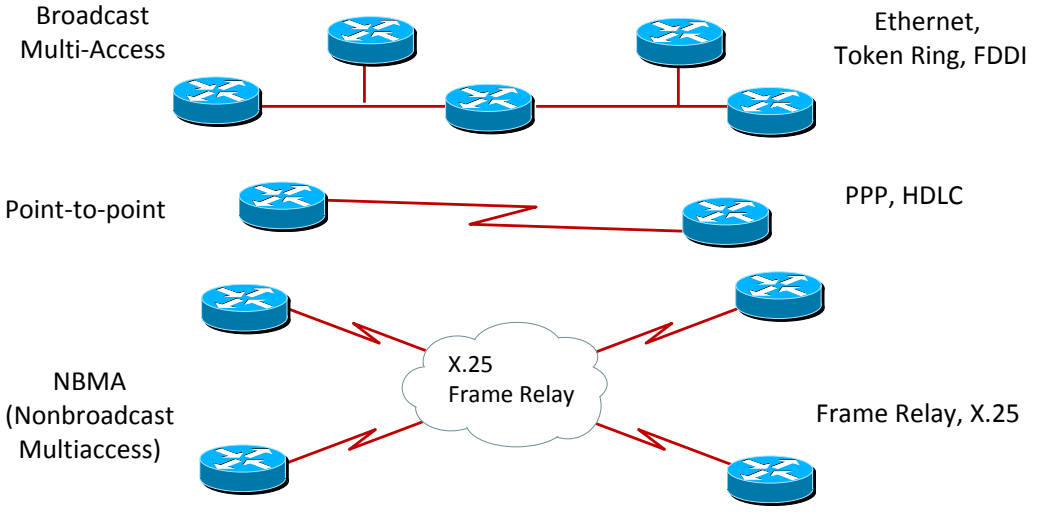
•Broadcast Multi-Access: tip de rețea folosit în medii partajate în care toate ruterele ar trebui să realizeze adicențe între ele; pentru a evita congestia în rețea, se aleg două rutere care au un rol central în dirijarea LSP; acestea se numesc DR (Designated router) și BDR (Backup designated router), backup pentru DR; astfel ruterele vor încerca să formeze o adiacență completă doar cu DR și BDR; aceste două rutere desemnate asigură că informațiile de tip link state din întreaga topologie sunt sincronizate corect și actualizate

•Point-to-point: tipul de rețea folosit pentru a realiza adicență doar între două rutere; astfel se trimit inițial pachete de „Hello” spre adresa de multicast 224.0.0.5 după care LSP-urile sunt trimise direct către ruterul vecin; deoarece sunt doar două rutere implicate nu se va mai efectua procesul de alegere a DR și BDR

_____

_____

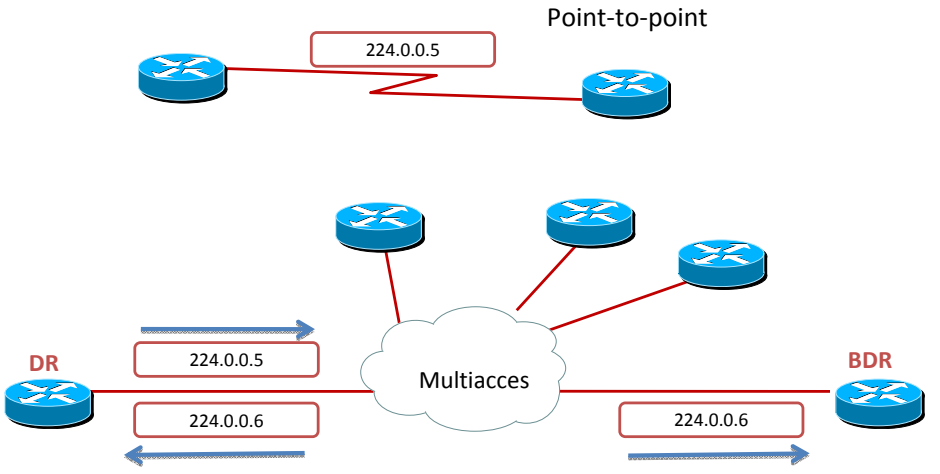
Tipuri de rețele OSPF (2)



- NBMA (Nonbroadcast Multiaccess): tip de rețea care nu are suport pentru multicast, ceea ce reprezintă o problemă majoră în cazul stabilirii adiacenței OSPF; dacă topologia NBMA nu este un „full mesh” (configurarea adiacenței între oricare doi vecini) pachetul de broadcast sau multicast trimis de către ruter nu va ajunge la destinație; pentru a rezolva această problemă este necesară implementarea unor mecanisme specializate care să asigure o metodă eficientă de rutare

Mesajele de tip „Hello” sunt trimise o dată la 10 secunde pe medii multiacces sau point-to-point, și o dată la 30 secunde pe medii NBMA (non-broadcast multiaccess).

Comunicarea OSPF



Toate informațiile despre o rută descoperită sunt trimise de către OSPF într-o structură de date numită LSA (Link-State Advertisment). Din motive de eficiență și conservare a lățimii de bandă, OSPF trimite mai multe LSA-uri în același tip de pachet numit LSP (Link-State Packet).

Modul de distribuție ale acestor update-uri depinde de mediul de transmisie utilizat. Dacă pe rețelele point-to-point toate update-urile se transmit folosind adresa de multicast 224.0.0.5, în cele multiaccess ar trebui să se trimită pachetul către n(n-1)/2 rutere ale topologiei. În acest caz s-ar consuma atât timp de procesare cât și lățime de bandă, rezultând în supraîncărcarea rețelei cu informațiile transmise de OSPF. Pentru a rezolva această problemă, OSPF alege o soluție centralizată de distribuție a update-urilor folosind un ruter cu rol de arbitru – Designated Router. Astfel, doar un singur ruter din rețea va distribui update-urile către ceilalți vecini, prevenind congestionarea acesteia.

_____

_____

DR. BDR

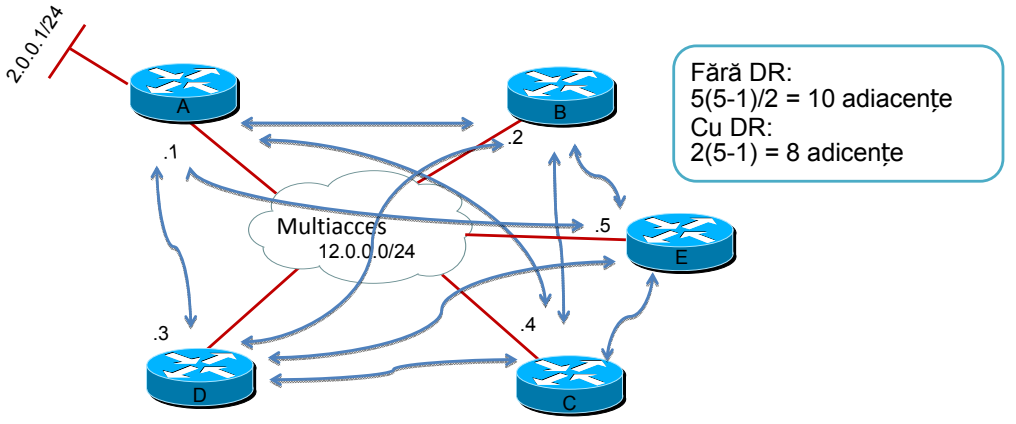
- Folosite în rețele **multiacces**
- OSPF alege un DR (Designated Router) și un BDR (Backup Designated Router) pentru a reduce traficul OSPF
  - DR se ocupă cu updatarea celorlalte routere (DROthers) când au loc schimbări de topologie
  - BDR monitorizează DR, și în cazul în care DR nu mai funcționează îi ia locul

DR (Designated Router) și BDR (back-up designated router) sunt numele asociate unor rutere cu rol special în cadrul unei rețele broadcast multiacces în care rulează OSPF. Acestea se aleg automat în timpul stabilirii adiacenței în topologie și au rolul de a minimiza numărul de adiacențe din rețea și a limita congestionarea acesteia. DR-ul se ocupă cu propagarea pachetelor LSA către DROthers (celelalte rutere din domeniu) și cu păstrarea rețelei sincronizate în cazul unor schimbări de topologie. DROthers vor stabili adiacența doar cu DR-ul și BDR-ul. În cazul în care legătura cu DR-ul pică sau acesta este oprit, BDR va prelua rolul DR-ului; acest schimb se desfășoară foarte rapid, deoarece BDR-ul are deja stabilite adiacențe cu toți ceilalți DROthers.

DR-ul și BDR-ul au o adresă de multicast dedicată – 224.0.0.6. Pachetele trimise către această adresă de multicast vor fi primite doar de DR sau BDR. Adresa de multicast 224.0.0.5 (All OSPF Routers), este folosită de DR pentru a comunica cu DROthers.

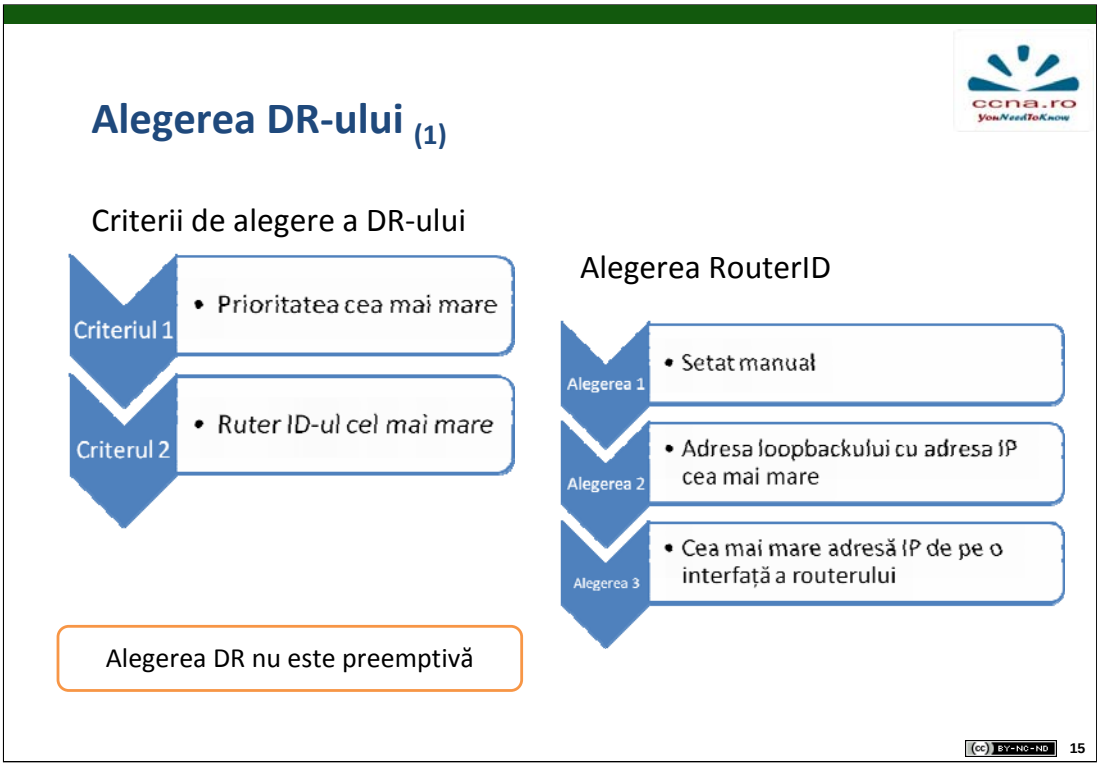
De ce este nevoie de DR?

- Fără DR ar fi n(n-1)/2 adiacențe
- Cu DR sunt (n-1) adiacențe + (n-1) cu BDR = 2(n-1)



Principalul motiv pentru care este nevoie de un DR este reducerea numărului de adiacențe, rezultând o diminuare a traficului și implicit o mărire simțitoare a performanței rețelei. Fără existența unui ruter desemnat pentru trimiterea actualizărilor, singura metodă folosită pentru propagarea mesajelor către toate ruterele este realizarea a n(n-1)/2 adiacențe, unde n reprezintă numărul de rutere din rețeaua multiaccess. Prin alegerea unui DR și a unui BDR, numărul de adiacențe necesare se reduce semnificativ, fiind realizate un număr total de 2(n-1) adiacențe (n-1 rutere comunică cu DR-ul și cu BDR-ul).

Rezultatul implementării unei astfel de soluții este că, la un moment dat, există un singur ruter care realizează trimiterea de pachete LSA către DROthers (ruterele din rețea care comunică cu DR-ul și BDR-ul).

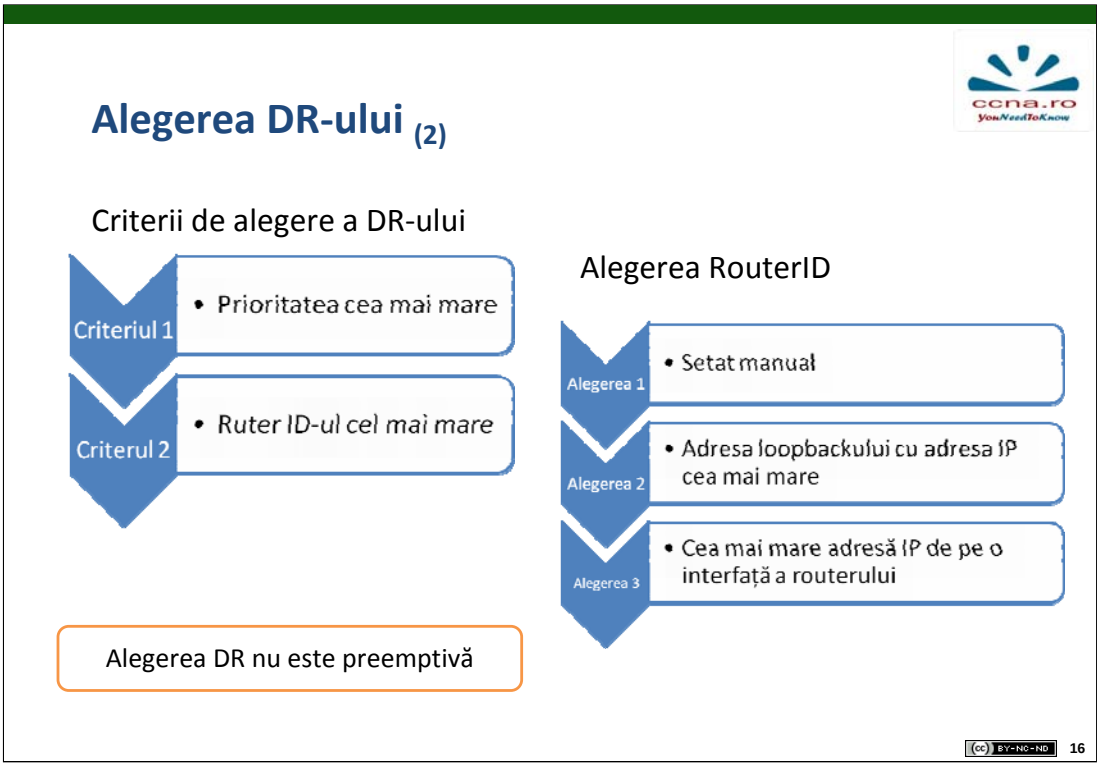


Într-o rețea multiaccess, există 3 tipuri de rutere din punct de vedere al rolului jucat în distribuția de update-uri: DR, BDR și DROTHER. Orice ruter DROTHER va transmite orice update către DR și BDR folosind adresa de multicast asociată acestora, și anume 224.0.0.6. DR-ul este apoi singurul care transmite celorlalte rutere DROTHER update-uri folosind adresa de multicast 224.0.0.5. BDR-ul nu transmite nici un mesaj, doar ascultă mesajele din rețea și formează adiacențe cu ceilalți DROTHERS pentru a prelua rolul DR-ului în cazul în care acesta devine indisponibil.

DR-ul se alege într-o rețea de tip multiaccess în funcție de prioritatea cea mai mare sau, dacă acestea sunt egale, în funcție de Router ID-ul cel mai mare. Prioritatea unui ruter fi setată manual și este o valoare între 0 și 255, 255 reprezentând cea mai mare prioritate, și 0 dacă nu se dorește ca acel ruter să participe în procesul de alegere DR/BDR.

_____

_____



Router ID este o valoare, teoretic, unică pentru fiecare ruter dintr-o topologie, și se alege automat la inițializarea procesului de OSPF utilizând cea mai mare adresă de loopback configurată pe ruter și inclusă în procesul de OSPF. Dacă nu există nici o adresă de loopback configurată, procesul de OSPF va lua în considerare ca Router ID cea mai mare adresă IP a unei interfețe. Cu toate acestea, pentru un control optim al procesului de alegere DR/BDR, Router ID se poate configura și manual.

Alegerea DR nu este preemptivă, însemnând că la apariția unui nou ruter cu o prioritate sau un Router ID mai mare, nu se va iniția automat procesul de alegere DR/BDR.

Ruter IDs duplicate

- Când două sau mai multe rutere au același ruter-id, rutarea poate să nu mai funcționeze intuitiv
  - dacă două routere vecine au același ID este posibil să nu formeze adiacență
- Când se întâmplă ca două rutere să aibă același ID, IOS afișează:
  - %OSPF-4-DUP_RTRID1: Detected ruter with duplicate router ID

Este posibil ca într-un mediu de producție două rutere să aibă același „router ID”. În acest caz, consola va afișa un mesaj de eroare și procesul de rutare nu va funcționa corect. În funcțiile de detaliile furnizate în mesajul apărut, se poate determina dacă identificatorul are un duplicat în interiorul ariei sau într-o arie diferită.

În cazul apariției unor astfel de mesaje pe un ruter din topologie se recomandă intervenția imediată a administratorului de rețea pentru soluționarea problemei prin asigurarea că ruterele din cadrul domeniului de rutare OSPF au asociate un identificator unic.

Scenarii de alegere DR

- DR nu mai funcționează
  - BDR îi ia locul și se alege din DROthers cel cu router-id-ul cel mai mare ca BDR
- Apare un nou router în OSPF
  - nu se întâmplă nimic, procesul este nepreemptiv
- BDR nu mai funcționează
  - se alege din DROthers cel cu router-id-ul cel mai mare ca BDR
- Și DR și BDR nu mai funcționează
  - se aleg din DROthers cele cu router-id-ul cel mai mare ca DR,BDR

În cadrul unei topologii în care rulează OSPF, este posibil să apară unele probleme care să interfereze cu funcționarea normală a procesului de rutare:

•În cazul în care DR-ul nu mai funcționează, BDR-ul îi va lua locul imediat, și va începe procesul de alegere a unui nou BDR dintre DROthers

•În cazul în care apare un nou ruter în OSPF, având un ruter ID mai mare decât actualul DR sau BDR, nu va produce nici o schimbare în ierarhia DR/BDR

•În cazul în care BDR nu mai funcționează, se va alege din DROthers un nou BDR

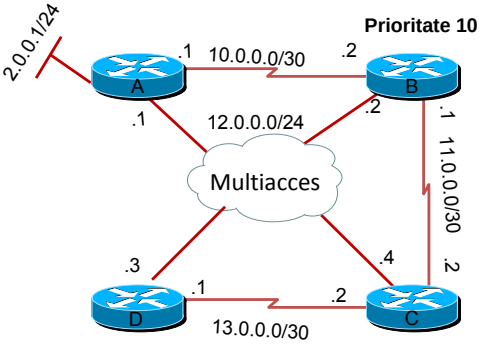
•În cazul în care nici DR și nici BDR nu mai funcționează, se vor alege noi DR-uri și BDR-uri din DROthers, respectând regula cu prioritatea/„ruter ID”-ul cea/cel mai mare

_____

_____

Exemplu alegere DR

- Interfața aleasă ca router-id
  - nu trebuie neapărat să ruleze OSPF pe ea
  - trebuie să fie “up”




```
Router ID A: 2.0.0.1
Router ID B: 12.0.0.2
Router ID C: 13.0.0.2
Router ID D: 13.0.0.1

DR: B (prioritate 10 > prioritate default 1)
BDR: C (router-id cel mai mare)
```


În exemplul de mai sus, deși există 5 rețele diferite în topologie, alegerea DR/BDR se face doar pe rețeaua multiacces 12.0.0.0/24. Dacă toate ruterele din figură sunt pornite în același timp, ruterul B va deveni DR din cauza priorității sale modificate la valoarea 10. Ca BDR va fi ales ruterul C pentru că are cel mai mare identificator dintre cele 3 rutere rămase. Dacă ruter-ul B va avea vreodată o defecțiune hardware sau software și nu va mai putea rula procesul de OSPF, ruter-ul C va deveni direct DR în rețea, indiferent de valoarea priorității sau „ruter-id”-ului său din acel moment. BDR-ul va fi ales în continuare folosind cele 2 criterii rămase.

Stări OSPF (1)

- Stările OSPF de adiacență

- Pentru a putea schimba rute vecinii OSPF trebuie să se afle în starea FULL

Atunci când sunt pornite, ruterele care rulează OSPF trec printr-un proces de inițializare facilitat de protocolul „Hello”. În timpul acestui proces, ruterele trec prin mai multe stări:

•INIT: Ruterul trimite pachete „Hello” pe toare interfețele în procesul de OSPF în încercarea de a stabili adiacențe cu vecinii către adresa multicast 224.0.0.5; toate ruterele direct conectate primesc acest pachet „Hello” și adaugă ruterul sursă în tabela lor de vecini; aceste rutere sunt acum în starea INIT

•Two-Way: Ruterele care sunt acum în starea INIT trimit înapoi pachete unicast către adresa de unde au fost primite, acestea conținând informațiile lor specifice (exemplu: toate ruterele vecine); atunci cand le primește, ruterul care a inițiat schimbul a format acum adiacența de tip „two-way” cu vecinii săi

Stări OSPF (2)

- Stările OSPF de adiacență

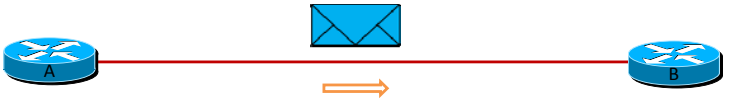
- Pentru a putea schimba rute vecinii OSPF trebuie să se afle în starea FULL

- Exstart: se începe procesul de alegere DR/BDR; în acest proces, se iau în considerare prioritățile/„router-id”-urile fiecărui ruter și se selectează cele două rutere semnificative (DR și BDR)
- Exchange: ruterele schimbă între ele pachete DBD, reprezentând header-ele Link-state Database-ului fiecăruia; astfel, fiecare ruter își populează propriul LSDB cu informații elementare despre topologie
- Loading: pe baza informațiilor primite în starea anterioară, fiecare ruter trimite pachete de tip LSR către DR, cerând informații suplimentare despre un anumit link. Răspunsul vine ca un LSU, care din motive de eficiență conține mai multe LSA-uri. Fiecare ruter își completează informațiile din LSDB
- Full: după ce toate ruterele și-au completat LSDB-ul, putem spune că ruterul respectiv este pregătit să trimită pachete pe baza informațiilor din tabela de rutare

Descoperirea vecinilor

- Protocolul de Hello asigură trecerea din starea **init** în starea **two-way**

Network Mask		
Hello Interval	Options	Router Priority
Dead Interval		
Designated Router		
Backup Designated Router		
Neighbour Router ID		
Neighbour Router ID		
<alte câmpuri de tip Neighbour Router ID, dacă sunt necesare>		



Pentru a-și îndeplini funcțiile, protocolul Hello implementează un format specific de mesaje Hello pe care le trimite între 2 rutere OSPF care doresc să realizeze adiacențe. Toți parametrii care trebuie să se potrivească între cele 2 instanțe de OSPF sunt incluși în antetul mesajului „Hello”. Antetul are lungime variabilă și conține la sfârșitul său o listă de „Neighbour Router ID”. Când un ruter R1 primește de la vecinul său R2 un „Hello”, îl adaugă în această listă și îi trimite un pachet „Hello” înapoi. Când R2 primește „Hello-ul” răspuns și vede că ID-ul său este în lista menționată, cele 2 rutere intră în starea de adiacență numită „Two-Way”. OSPF are definite timere implicite de trimitere a unui „Hello” și de expirare a unei adiacențe (Dead Timer). Acestea pot fi modificate de administrator în funcție de cerințele rețelei, însă, fiind transmise în mesajul Hello, ele trebuie să se potrivească între vecini pentru a fi posibilă realizarea unei adiacențe.

Adiacențe OSPF

- În rețelele multiacces
  - FULL: se formează doar cu DR și BDR
  - 2-WAY: între DROthers
- În rețelele point-to-point
  - FULL între cele două rutere și nu se alege DR

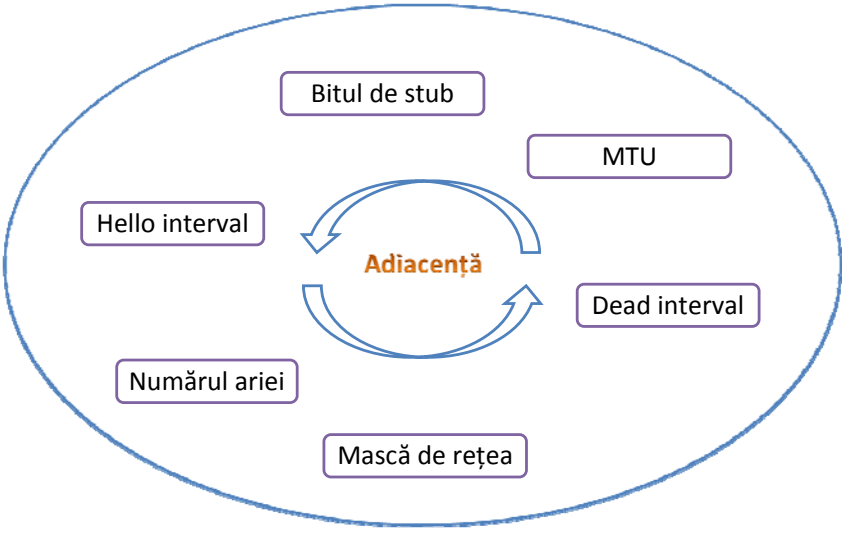
```
RouterA#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address     Interface
12.0.0.2          0   FULL/  -        00:00:37    10.0.0.2    Serial1/0
12.0.0.2         10   FULL/DR         00:00:37    12.0.0.2    FastEthernet0/0
13.0.0.1          1   2WAY/DROTHER    00:00:36    12.0.0.3    FastEthernet0/0
13.0.0.2          1   FULL/BDR        00:00:37    12.0.0.4    FastEthernet0/0
```


Rețelele multiaccess sunt alcătuite din mai mult de două dispozitive care comunică printr-un mediu partajat. Rețele local de tip Ethernet, în cadrul cărora deseori este întâlnit și un dispozitiv de nivel 2 (switch) reprezintă rețele multiaccess. Un pachet transmis pe o adresă de broadcast va fi primit de toate dispozitivele din cadrul rețelei, întrucât acestea au alocat câte un IP din aceeași rețea. Din motive de eficientizare a distribuției „update”-urilor, adiacența completă se realizează doar între perechile DROthers – DR, DROthers – BDR. Între ruterele DROthers adiacența se va realiza doar până în starea 2Way.

În rețelele point-to-point, doarece există doar două dispozitive conectate între ele, alegerea unui DR respectiv a unui BDR este inutilă. Adiacența realizată între cele două rutere vecine este completă, pentru a permite desfășurarea procesului de schimbare a informațiilor de rutare.

Condiții pentru realizarea adiacenței

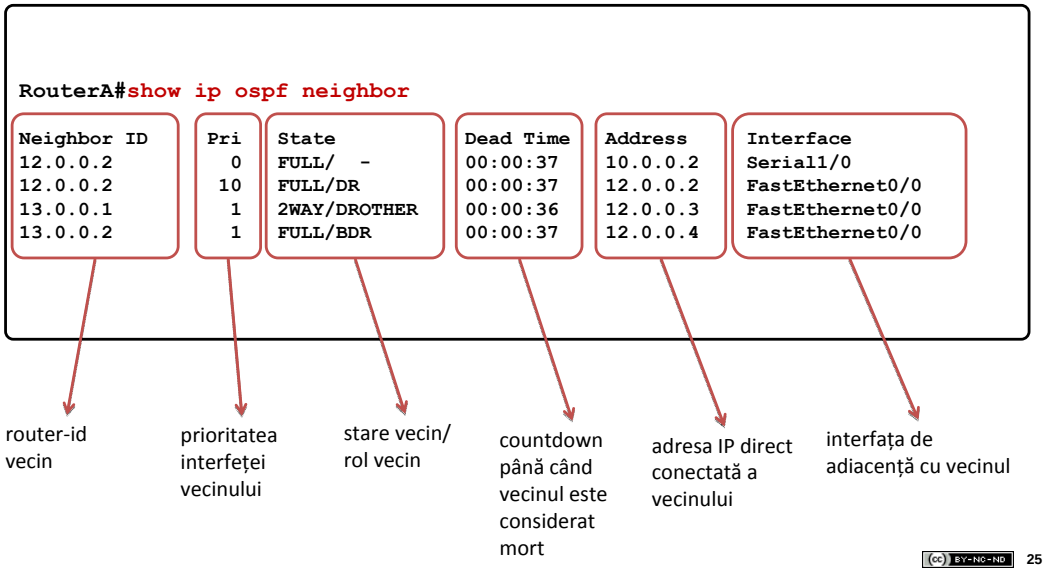


Pentru a obține adiacența OSPF între 2 rutere trebuie ca anumiți parametrii configurați să coincidă. Acești parametrii sunt introduși în mesajul de „HELLO” și trimiși vecinului. Pentru ca toate ruterele să ajungă în starea FULL de adiacență, trebuie ca mai mulți astfel de parametrii să coincidă; implicit, majoritatea parametrilor se vor potrivi. Astfel, două rutere care rulează OSPF vor schimba informații despre rute doar dacă sunt în aceeași arie (sau unul din ele este în aria 0). De asemenea, trebuie ca timpul dintre 2 mesaje de „HELLO” trimise să fie identic, la fel cum și valorile intervalului „Dead interval” trebuie să coincidă. Implicit, valoarea „Dead interval” este de 4 ori mai mare decât valoarea „Hello interval”. Adiacența nu se poate realiza dacă valoarea MTU și măștile de rețea nu sunt configurate la fel în ambele capete ale interfețelor ruterelor adiacente.

_____

_____

Verificarea adiacențelor



După ce două rutere au realizat adiacența (au ajuns în starea FULL) ele continuă să schimbe pachete HELLO la intervale de 10 secunde pentru rețelele multiaccess, respectiv 30 de secunde pentru rețele NBMA. În cazul în care nu se primește un pachet „Hello” în intervalul reprezentat de „Dead Time”, ruta este declarată invalidă și va fi scoasă din tabela de rutare imediat. De asemenea, se va trimite un LSU către toate interfețele în procesul de OSPF prin care se anunță că acea rețea nu mai este accesibilă, mesajul fiind trimis de către DR către ceilalți DROthers. În cazul în care se pierde legătura cu o anumită rețea, ruterul în cauză va reporni procesul de SPF utilizând datele existente în tabela de topologie și va găsi o nouă cale către acea rețea cu ajutorul algoritmului lui Dijkstra.

Activarea OSPF

- Wildcard
  - matematic, este opusul unei măști de rețea
  - funcțională, poate fi și discontinuă
- Numărul de proces
  - are semnificație locală
  - folosirea numărului de proces pentru a separa comunicarea OSPF nu este recomandată de CISCO

```
Router(config)#router ospf <process-id>
Router(config-router)#network <address> <wildcard-mask> area <area-id>
```


Activarea protocolului de routare OSPF pe un ruter se face din modul global de configurare, cu ajutorul comenzii r**outer ospf** urmată de un număr de proces unic. Acest număr de proces are numai relevanță locală; nu afectează domeniul de rutare, astfel încât două rutere vecine pot avea același număr de proces OSPF fără a întâmpina probleme. Acest lucru este în contrast cu EIGRP, în cazul căruia trebuia specificat numărul sistemului autonom pentru care se va face configurarea.

Comanda **network** se folosește pentru identificarea interfețelor /rețelor care vor fi incluse în procesul OSPF. În acest caz, este necesară și obligatorie adăugarea „wildcard-mask”-ului după adresa IP a rețelei pentru identificarea și delimitarea corectă a acesteia. După aceste două comenzi, urmează câmpul de „area-id”, pentru identificarea domeniului de rutare în care se dorește adăugarea rețelei în cadrul procesului de OSPF.

Metrica OSPF

$$\frac{10^8}{bandwidth}$$

Mediu	Cost
56 kbps – serial	**1785**
T1 (1.544 Mbps – serial)	**64**
E1 (2.048 Mbps – serial)	**48**
4 Mbps Token Ring	**25**
Ethernet	**10**
16 Mbps Token Ring	**6**
100 Mbps Fast Ethernet, FDDI	**1**

OSPF este un protocol de rutare link-state care utilizează algoritmul Dijkstra pentru calcularea arborelui de cost minim. Acest cost este influențat, în cazul OSPF, doar de bandwith-ul legăturii respective. Utilizând formula de mai sus, se obțin costuri diferite pentru diferite medii de transmisie, aceste costuri fiind utilizate în determinarea căii optime minime către orice destinație. Astfel, în mod teoretic, un ruter care rulează OSPF va prefera întotdeauna o cale care presupune 63 de legături Fast-Ethernet decât o cale cu o singură legătură T1 (situația nu este practică deoarece TTL-ul unui pachet IP este mult mai mic decât 63, iar OSPF suportă în jur de 50 de rutere într-un domeniu).

Modificarea metricii OSPF

- Modificarea costului atribuit unei interfețe:
  - `(config-if)#ip ospf cost <cost_nou>`
- Modificarea bandwidth-ului unei interfețe:
  - `(config-if)#bandwidth <bandwidth_nou_Kb>`
- Modificarea bandwidth-ului de referință:
  - `(config-if)#auto-cost reference-bandwith <bandwith_nou_MB>`

Pentru un control optim al procesului OSPF, un administrator de rețea poate modifica manual atât costul unei anumite legături, cât și lățimea de bandă acesteia. Este important de reținut ca lățimea de bandă la nivelul unei interfețe nu modifică decât viteza care va fi luată în considerare de diversele protocoale de rutare; viteza fizică a legăturii nu va fi afectată. De asemenea, configurări avansate OSPF permit setarea numărătorului fracției folosit în calculul costului la altă valoare, mai mare de 10^8, permițând astfel un calcul exact și pentru viteze mai mari de 100 Mbps.

Timere OSPF

- Timere
  - rețele broadcast multiacces și point-to-point:
    - Hello: 10 secunde
    - Dead: 40 secunde
  - rețele NBMA
    - Hello: 30 secunde
    - Dead: 120 secunde
- Un LSA are max-age 60 minute
  - o dată la 30 minute se face flooding cu un LSU pentru fiecare LSA deținut
  - `(config-if)#ip ospf hello-interval <time>`
  - `(config-if)#ip ospf dead-interval <time>`

Protocolul de rutare OSPF are anumite timer-e setate standard pentru cele trei tipuri majore de rețele: broadcast, multiaccess, point-to-point, NBMA. Un LSA are un max-age de 60 de minute, însemnând că informația conținută în aceasta va fi declarată invalidă după acest timp. Din acest motiv, fiecare ruter care rulează OSPF va trimite cate un LSU conținând fiecare LSA deținut o data la 30 de minute, din motive de sincronizare a rețelei.

Hello interval și Dead Interval se pot modifica, cu condiția ca ele să fie identice pentru două rutere adiacente. Acest lucru este diferit de EIGRP, unde Hello Timer și Hold-Down timer nu sunt relevante pentru păstrarea adiacenței.

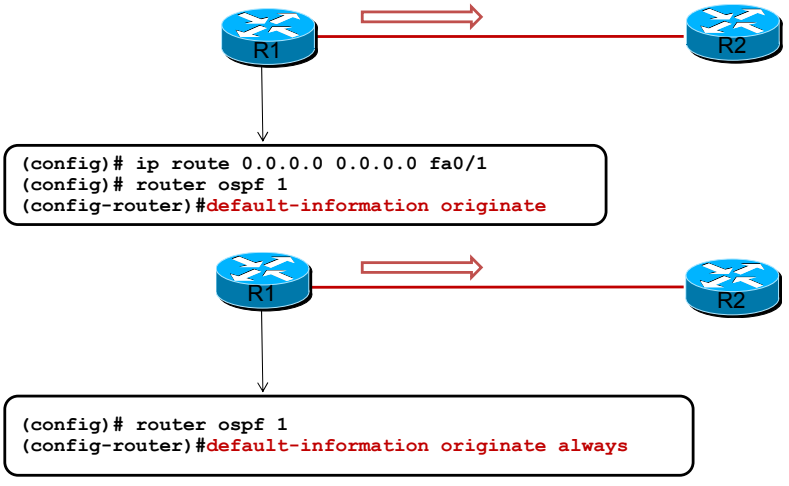
Influențarea alegerii DR

- Se poate modifica prioritatea pe interfață
  - 0 înseamnă ca ruterul nu va putea fi niciodată DR sau BDR
  - 1 este by default
  - `(config-if)#ip ospf priority <prioritate>`
- Setarea manuală a RouterID-ului
  - `(config-router)#router-id <router-id>`
  - `#clear ip ospf processes`
- **Atenție**: restartarea procesului OSPF are efect doar dacă s-a configurat manual un router-id

Un administrator de rețea poate modifica anumiți parametri care influențează alegerea DR/BDR. Dintre acestea, cea mai semnificativă este modificarea priorității per-interfață a unui ruter, fiind preferat în procesul de alegere DR valoarea cea mai mare a priorității. Prioritatea se setează per-interfață deoarece procesul de alegere DR/BDR se petrece doar într-o rețea de tip multiaccess, fiecare segment de rețea alegându-și propriul DR/BDR (cu excepția rețelelor point-to-point). În cazul în care două rutere au aceeași prioritate, alegerea DR-ului va fi determinată de cea mai mare valoare a ruter id-ului. Acesta poate fi setat manual la o valoarea preferabilă pentru situația existentă.

Dacă ulterior alegerii automate a unui ruter id pe baza adresei de loopback sau a adresei IP se configurează un nou loopback mai mare, ruter id-ul nu se va modifica în urma restartării procesului de rutare. În acest caz, dacă se dorește influențarea procesului de alegere DR/BDR trebuie setată manual prioritatea sau setat manual alt ruter ID.

Redistribuirea rutei default



Un ruter care rulează OSPF poate să se comporte ca un gateway pentru restul rețelei (toate cererile pentru rute inexistente în tabela de rutare să fie trimise catre el). Acest lucru se realizează cu ajutorul comenzii **`Default-information originate`**, care se poate utiliza doar dacă a fost configurată o rută default anterior pe ruter. Dacă nu a fost configurată, se va folosi parametrul suplimentar **`always`**.

Fine-tuning OSPF

- Modificarea lățimii de bandă de referință
  - by default 100Mb =>
  - bw>=100Mbps are un cost = 1
  - se recomandă schimbarea lățimii de bandă de referință pe toate routerele din aceeași arie

$$\frac{10^8}{bandwidth}$$

```
(config-router)#auto-cost reference-bandwidth <bw-referință-Mbps>
```


În cazul în care într-o topologie există legături cu o lățime de bandă asociată mai mare de 100MBps, cum sunt cele care suportă viteze de ordinul GBps, costul calculat va avea întotdeauna valoarea 1. Pentru evitarea unor astfel de situații apărute în topologie care pot genera o rutare suboptimală, se recomandă modificarea lățimii de bandă de referință folosită în calculul formulei costului asociat unei legături.

Comanda utilizată în acest scop este: **auto-cost reference-bandwidth *bandwidth-interfață-MBps***

Autentificare

- Nulă

```
(config-if)#ip ospf authentication null
```

- Plain text

```
(config)#router ospf 1
(config-router)#area 0 authentication
(config)#int fa0/0
(config-if)#ip ospf authentication-key cisco
```

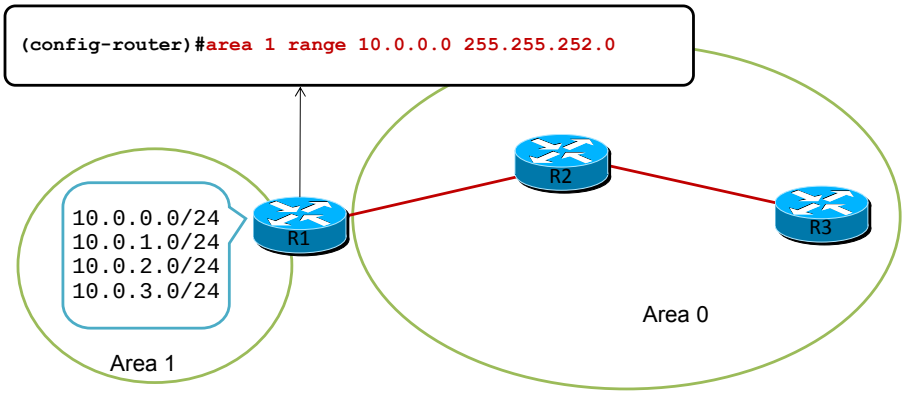
- MD5

```
(config)#router ospf 1
(config-router)#area 0 authentication message-digest
(config)#int fa0/0
(config-if)#ip ospf message-digest-key 1 md5 cisco
```


Autentificarea OSPF este de două tipuri: plain-text sau MD5. Autentificarea și tipul acesteia trebuie să fie activate atât în modul de configurare al protocolului cât și din modul de configurare al interfeței. În cazul în care un capăt al unei conexiuni dintre două rutere care rulează un proces compatibil de OSPF are configurată autentificare, însa celălalt capăt nu are configurată autentificare sau are configurată o altă parolă, cele două rutere nu vor forma adiacență OSPF.

Sumarizarea unei rute

- Se realizează pe ABR (Area Border Router)
- R1 va transmite în Area 0 o rută sumarizată a celor 4 rute



Sumarizarea unei rute se realizează de obicei pe un ABR (Area Border Router). Un ABR este acel ruter care se află la intersecția dintre două sau mai multe arii și facilitează atât sumarizarea cât și transmiterea informației despre rețele între cele două arii conectate. Sumarizarea are ca beneficii creșterea scalabilității și reducerea consumului de memorie și procesor. Astfel, se reduce numărul de pachete LSA propagate și dimensiunea tabelelor de topologie/rutare.

Verificarea configurării OSPF

```
RouterA#show ip protocols
Routing Protocol is "ospf 11"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 2.0.0.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    0.0.0.0 255.255.255.255 area 0
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    13.0.0.1             110      00:08:02
    12.0.0.2             110      00:08:02
    13.0.0.2             110      00:08:02
  Distance: (default is 110)
```


Comanda **`show ip protocols`** oferă informații despre toate protocoalele de rutare dinamice care rulează pe ruter. Din outputul afișat pentru procesul OSPF 11 se poate vizualiza id-ul ruterului curent, rețelele introduse în procesul de rutare, dar și detalii despre vecinii acestuia. De asemenea este specificată și valoarea implicită a distanței administrative pentru OSPF, si anume 110.

Verificarea configurării interfeței

```
RouterB#show ip ospf interface fa 0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 12.0.0.2/24, Area 0
  Process ID 11, Router ID 12.0.0.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 10
  Designated Router (ID) 12.0.0.2, Interface address 12.0.0.2
  Backup Designated router (ID) 13.0.0.2, Interface address 12.0.0.4
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:04
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 3, Adjacent neighbor count is 3
    Adjacent with neighbor 2.0.0.1
    Adjacent with neighbor 13.0.0.1
    Adjacent with neighbor 13.0.0.2  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```


Parametrii OSPF configurați pe fiecare interfață, precum și starea DR/BDR pe segmentul respectiv, pot fi vizualizate prin comanda **`show ip ospf interfaces`** urmată de numele interfeței ce se dorește a fi inspectată. De asemenea, pot fi vizualizate informații legate de tipul rețelei, costul calculat de OSPF, dar și prioritatea interfeței, în exemplul din output având valoarea 10, diferită de valoarea implicită.

Vizualizarea configurațiilor

```
Router#show ip ospf
```

- Afișează RouterID-ul, timere și statistici

```
Router#show ip ospf interface
```

- Afișează RouterID-ul, AreaID și informații de adiacență

```
Router#show ip route ospf
```

- Afișează rutele OSPF

```
Router#show ip protocols
```

- Afișează informații despre toate protocoalele de rutare active

```
Router#show ip ospf neighbors
```

- Afișează tabela de vecini

Verificarea configurațiilor efectuate în implementarea protocolului OSPF într-o rețea se poate face cu următoarele comenzi:

• **`show ip ospf`** – se menționează identificatorul procesului OSPF, identificatorul ruterului sau aria în care rulează protocolul; pentru afișarea detaliilor OSPF pentru o anumită interfață, se mai adaugă comenzii parametrii suplimentari: **`sh ip ospf interface`** ***`nume_interfață`***

• **`show ip protocols`** – se afișează informații detaliate despre toate protocoalele active pe un ruter la un moment dat

• **`show ip ospf neighbors`** – se folosește în cazul în care se dorește afișarea vecinilor cu care s-au stabilit adiacențe, dar și pentru identificarea ruterelor DR și BDR dintr-o rețea multiaccess

Rezumat

- OSPF
- Configurare OSPF

1. Care sunt deosebirile principale dintre un protocol de rutare link-state în comparație cu un protocol de rutare distance-vector?
2. Care sunt diferențele transmiterii pachetelor de Update OSPF într-o rețea multiaccess în comparație cu o rețea punct-la-punct?
3. Ce algoritm stă la baza calculării drumului de cost minim de la un ruter spre toate celălalte rutere din domeniu ?
4. Care sunt criteriile de alegere a „Router ID”-ului ?
5. Care sunt pașii folosiții pentru construirea arborelui SPF?