

Curs 2: Routarea pachetelor

Routarea pachetelor in retelele de calculatoare

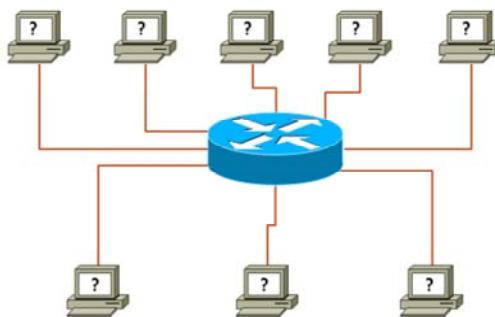
Procesul de rutare



De ce avem nevoie de rutare?

Transmiterea pachetelor între rețele

- criterii de eficiență
- criterii de politică internă sau externă



Principalul rol al unui ruter într-o topologie este rutarea pachetelor între mai multe rețele. Rutarea este procesul prin care dispozitivul de rețea analizează antetul de nivel 3 al unui pachet primit (adresa IP destinație) și apoi, pe baza anumitor reguli clar definite manual sau de protocoale specializate, decide ce să facă cu pachetul mai departe. Un ruter de nivel „enterprise” trebuie să fie capabil să proceseze în jur de 10.000 pachete pe secundă, iar acest lucru este posibil doar prin implementarea unor circuite hardware dedicate foarte rapide.

Deși din punct de vedere teoretic, rutarea pachetelor se desfășoară doar la nivelul 3, un ruter este capabil să citească informații din antete până la nivelul 7. Acest lucru este în special folosit în aplicații „mission-critical” cum ar fi VoIP sau aplicații care necesită latență extrem de mică, cu ajutorul unor sisteme de QoS (Quality of Service). QoS este un serviciu oferit de anumite rutere avansate care permite prioritizarea traficului în funcție de conținut sau destinație.

Criterii de rutare

Informațiile cu privire la rețelele cunoscute sunt stocate în tabela de rutare

- se stochează adresele rețelelor și următorul hop către fiecare destinație
- în cazul conexiunilor punct-la-punct se poate stoca direct interfața de ieșire
- același lucru se face automat pentru rețelele direct conectate

Rute statice

- configurate de administrator
- au prioritate în procesul de rutare

Rute dinamice

- învățate prin intermediul unor protocoale specializate
- algoritmii folosesc criterii de eficiență sau criterii de politică

Fiecare ruter are o bază de date salvată în RAM care conține regulile setate manual sau automat folosite pentru luarea deciziilor de rutare; această bază de date se numește tabelă de rutare.

Tabela de rutare a unui ruter stochează informații multiple cu privire la rețelele adiacente unui ruter și la calea pe care un pachet trebuie să o urmeze pentru a ajunge în rețea destinație. Astfel, pentru o destinație oarecare tabela de rutare stochează:

- masca de rețea
- metoda prin care calea respectivă a fost aflată
- adresa IP next-hop sau interfața de ieșire prin care aceasta poate fi accesată

Rutele pot fi învățate de un ruter prin două metode:

- **Static**, configurate de un administrator; acest tip de rută va fi întotdeauna preferată față de o rută dinamică
- **Dinamic**, cu ajutorul unui protocol de rutare specializat; în acest caz se folosesc algoritmi avansați pentru determinarea căii optime

Rutare statică vs. Rutare dinamică (1)

Rutare statică

- oferă control mult mai riguros administratorului asupra următorului hop ales
- este foarte ușor de învățat
- nu este deloc scalabilă

Rutarea statică se configerează manual pe rutere și oferă un management mai riguros asupra modului de stabilire a următorului hop către o anumită destinație. Un avantaj al folosirii rutării statice este efortul necesar scăzut pentru configurarea și administrarea rețelelor restrânse. Pe de altă parte, rutarea statică nu scalează optim în cazul rețelelor de dimensiuni mari, fiind necesară implementarea rutării dinamice. De asemenea, rutarea statică consumă puține resurse hardware, permitând rularea în paralel a altor aplicații performante dacă este necesar.

Rutare statică vs. Rutare dinamică (2)

Rutare dinamică

- necesită cunoștințe avansate pentru o configurare eficientă
- utilizează atât un procent din bandwidth cât și o parte din procesor
- calea aleasă de pachete nu e cunoscută în mod clar
- este o soluție scalabilă și tolerantă la defecte
- exemple: RIP, IGRP, EIGRP, OSPF, IS-IS, BGP

În cazul protocoalelor de rutare dinamice, informația despre rute este propagată automat în întreaga rețea, rutele fiind distribuite cu ajutorul unui algoritm specific. Protocoalele de rutare dinamice oferă scalabilitate și flexibilitate mărită față de metoda statică, însă consumă mai multe cicluri de procesor și utilizează o cantitate mai mare de memorie RAM. Protocoalele de rutare dinamice sunt responsabile cu păstrarea tabelei de rutare sincronizată peste toate ruterele din domeniul de rutare în cazul unei modificări în topologie. Astfel, informațiile că o rețea este invalidă sau afectată de anumite probleme hardware se propagă foarte rapid în toată rețeaua, fără intervenția administratorului. Exemple de protocole de rutare dinamice sunt RIPv1, RIPv2, IGRP, EIGRP, OSPF, IS-IS, RIPng, EIGRP IPv6 și BGP. RIPv1 și IGRP nu mai sunt utilizate, ele fiind înlocuite integral de versiunile lor îmbunătățite – RIPv2 și EIGRP. RIPng și EIGRP IPv6 sunt variantele bazate pe IPv6 ale protocoalelor de rutare asociate prin denumire.

Metrici, determinarea căii optime

Metrică

- indicator de preferință a unei rute după anumite criterii
- se calculează în funcție de hop-count, delay, bandwidth etc.
- o metrică mai mică este mai bună

Determinarea căii optime

- fiecare rută din tabel are atribuită o metrică
- ruterul alege ruta cu metrica cea mai mică

Determinarea celei mai bune căi către destinație implică evaluarea căilor disponibile în funcție de anumite criterii, dintre care se remarcă metrica. Metrica este o valoare calculată în funcție de anumite variabile asociate drumului dintre două puncte într-o rețea:

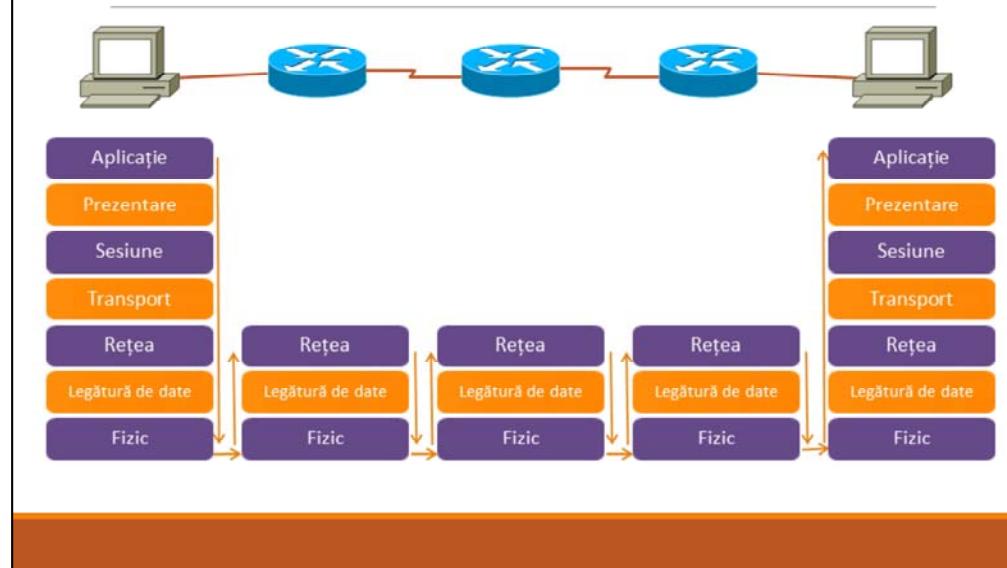
- Hop-count: numărul de rutere parcurse de pachet până la destinație
- Delay: latență specifică elementelor de legătură care alcătuiesc calea
- Bandwidth: viteza legăturilor dintr-o cale
- Load: încărcarea cu date a unei conexiuni
- Reliability: gradul de siguranță oferit de o anumită legătură

Protocolele de rutare dinamice pot utiliza una sau mai multe din aceste variabile în calculul metricii unui drum sau a unei destinații. Ulterior acestui calcul, protocolul de rutare va selecta ruta cu metrica cea mai mică și o va introduce în tabela proprie de rutare.

Manipularea pachetelor



Rutarea în cadrul stivei OSI



În cazul transmiterii de date între două stații, pachetele aflate în tranzit vor fi prelucrate de echipamentele terminale și intermediare folosind protocoalele definite în cadrul fiecărui nivel al stivei OSI. Astfel, un pachet va trece prin mai multe procese de încapsulare și decapsulare. La sursă, PDU-ul (Packet Data Unit) va fi încapsulat, la fiecare nivel adăugându-se noi informații specifice.

În cazul rutării, pachetul va fi decapsulat în fiecare ruter prin care trece până la nivelul 3, deoarece un ruter are nevoie doar de adresa IP destinație pentru a lua decizia de trimitere mai departe. Adresele IP sursă și destinație nu se vor schimba niciodată de-a lungul traseului. La nivelul Legătură de date, fiecare hop va modifica adresa MAC sursă, respectiv adresa MAC destinație. Antetul de nivel 2 se va modifica doar la trecerea într-o altă rețea, și nu la trecerea printr-un switch sau alt echipament de nivel 2. Când ajunge la destinație, pachetul este decapsulat și informația conținută este prezentată utilizatorului.

Manipularea pachetelor (1)

Adrese MAC

- adrese de nivel 2
 - folosite pentru identificarea fizică a dispozitivelor în cadrul unei rețele locale
 - se modifică la trecerea dintr-o rețea în alta

IEEE 802.3						
7	1	6	6	2	46 to 1500	4
Preamble	Start of frame delimiter	Destination Address	Source Address	Length Type	802.2 Header and Data	Frame check sequence

Pentru o comunicație eficientă în interiorul rețelei locale nu este nevoie de o adresă la nivel 3, fiind suficient antetul unui pachet la nivelul 2. Aceasta conține adresa MAC sursă cât și cea destinație, acestea fiind suficiente pentru transmiterea pachetului cu succes la destinație. Deoarece adresele fizice au numai relevanță locală, ele se vor modifica când pachetul părăsește rețeaua în care se află la un moment dat.

Cadrul 802.3 (Ethernet) este alcătuit din următoarele câmpuri:

- **Preamble:** 7 biți de 1 și 0 care alternează și au rol de sincronizare
- **Start of frame Delimiter:** un octet care semnalizează începutul cadrului
- **Destination Address:** adresa MAC a destinatarului
- **Source Address:** adresa MAC a expeditorului

Manipularea pachetelor (2)

Adrese MAC

- adrese de nivel 2
 - folosite pentru identificarea fizică a dispozitivelor în cadrul unei rețele locale
 - se modifică la trecerea dintr-o rețea în alta

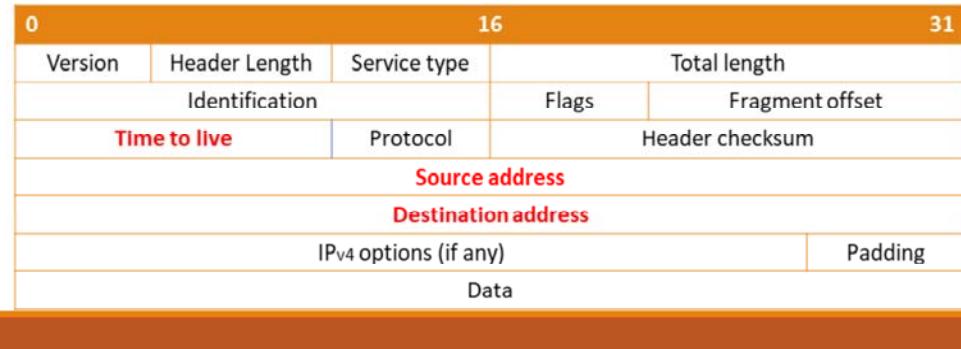
IEEE 802.3						
7	1	6	6	2	46 to 1500	4
Preamble	Start of frame delimiter	Destination Address	Source Address	Length Type	802.2 Header and Data	Frame check sequence

- **Length/Type:** doi octeți care pot reprezenta ori lungimea cadrului (dacă valoarea este mai mică decât 0x600 – echivalentul 1536 în decimal) sau tipul protocolului de nivel superior (dacă valoarea este mai mare de 0x600)
- **Data:** datele încapsulate în pachet
- **Frame Check Sequence:** 4 octeți cu rolul de a verifica integritatea datelor recepționate.

Manipularea pachetelor (3)

Adrese IPv4

- adrese de nivel 3
- folosite pentru identificarea rețelelor și a stațiilor din rețea
- se păstrează neschimbate în timpul rutării între rețele
- time-to-live poate fi folosit pentru a opri buclele de rutare



Antetul de nivel 3 este utilizat de ruter pentru a determina calea pe care trebuie să trimită pachetul în drumul spre destinație. Adresele de nivel 3 IP sursă și destinație nu se modifică niciodată în tranzit, fiind afectate alte câmpuri din antet, cum ar fi TTL (time to live) și Header Checksum. Câmpul TTL al fiecărui pachet IP se decrementează cu o unitate la fiecare trecere printr-un ruter. Decrementarea valorii TTL poate fi considerată o măsură de siguranță la nivel 3 permitând eliminarea rapidă a buclelor de rutare (în general considerând că un pachet este trimis cu o valoare TTL egală cu 16, aceasta nu este prins într-o buclă de rutare deoarece pachetul va fi aruncat după parcurgerea a 16 hopuri).

Tabela de rutare și principiile rutării



Tabela de rutare

Este folosită de ruter pentru a alege interfața de ieșire în transmiterea unui pachet

Este stocată în RAM, deci se pierde la fiecare repornire

Conține informații de tip rețea – interfață de ieșire (sau rețea – rețea intermediaрă)

- rețele direct conectate, adăugate implicit
- rețele la distanță: rute statice sau dinamice

Tabela de rutare a unui ruter reprezintă o structură de date ierarhică, unificată și organizată care stochează informații despre destinațiile cunoscute. Este stocată în RAM și nu se memorează la salvarea configurației unui ruter – ea se va reconstrui la fiecare repornire. Pe baza informațiilor conținute în tabela de rutare ruterele iau decizii cu privire la transmiterea unui pachet pe o anumită interfață de ieșire.

Tabela de rutare poate conține mai multe tipuri de rețele:

- Rețele direct conectate: sunt introduse automat în tabela de rutare, reprezentând rețelele care aparțin interfețelor active ale ruter-ului; ele nu pot fi șterse sau modificate fără o schimbare a adresării IP sau a dezactivării interfeței
- Rețele remote: configurate cu ajutorul rutelor statice sau a protoocoalelor dinamice de rutare

Tabela de rutare

Exemplu de tabelă de rutare

```
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  change change change

Gateway of last resort is 10.119.254.240 to network 10.140.0.0

o 172.150.0.0 [160/5] via 10.119.254.6, 0:01:00, Ethernet2
E 172.17.10.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
```

În momentul în care un ruter primește un pachet IP pe una din interfețe, adresa destinație din antetul pachetului va fi căutată în tabela de rutare pornind de la ruta cea mai specifică din tabelă (masca de rețea cea mai lungă) și până la ruta cea mai puțin specifică. Tabela de rutare prezintă informații despre următorul hop unde trebuie trimis un pachet pentru ca acesta să ajungă la destinația dorită. De asemenea este prezent și un timer, care în cazul protocolelor de rutare dinamice reprezintă timpul rămas până la declararea unei anumite rute invalide din cauza lipsei de activitate. Fiecare tip de rută din tabela de rutare este reprezentată de un simbol: O – OSPF, R – RIP, D – EIGRP, B – BGP, făcând mai ușoară parcurgerea acestoria și efectuarea de „troubleshooting” în caz de nevoie.

Principiile de rutare

Fiecare ruter ia decizii bazându-se doar pe propria tabelă de rutare

Nu toate ruterele au aceeași tabelă de rutare

Rutarea se face asimetric

- rutetele stocate se referă doar la drumul spre o rețea, nu și invers
- pachetele pot folosi alte căi la comunicarea în sens invers

Pentru asigurarea unei funcționări optime a procesului de rutare sunt respectate următoarele 3 principii:

- **Ruterele iau decizii de rutare independent bazându-se numai pe informațiile din propria lor tabelă de rutare; astfel, problemele de rutare sunt împiedicate de a se propaga în întreaga topologie, iar puterea de procesare pentru găsirea unei destinații este împărțită în mod egal tuturor nodurilor**
- **Tabela de rutare este unică pentru fiecare ruter deoarece aceasta conține următorul hop pentru fiecare destinație în parte; tabela de rutare a unui ruter nu va descrie niciodată întreaga cale pe care un pachet trebuie să o urmeze pentru a ajunge la destinația cerută**
- **Rutarea este asimetrică deoarece tabela de rutare nu descrie un „next hop” valabil pentru un drum dus-întors**

Clasificarea rețelelor la rutare

Conecțate

- rețelele direct conectate la interfețe ale ruterului
- rutele sunt adăugate automat după pornirea și configurarea interfeței

Cunoscute

- acele rețele către care sunt definite rute statice sau dinamice

Necunoscute

- nu există rute definite pentru aceste rețele
- se folosește ruta default, dacă e definită, sau se aruncă pachetul

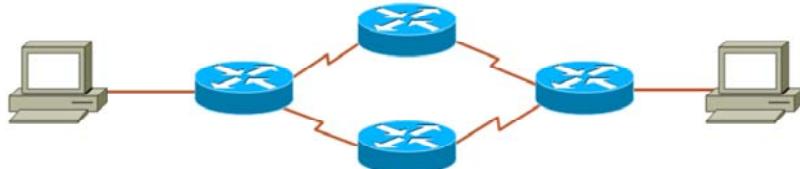
Rută implicită

- se definește static de către administrator sau este propagată dinamic
- se aplică pentru toate rutele necunoscute

Tabela de rutare a unui ruter poate fi populată de mai multe tipuri de rețele:

- **Conecțate – rețelele care aparțin interfețelor active ale ruterului, fiind introduse automat în tabela de rutare alături de interfețele de ieșire corespunzătoare**
- **Cunoscute – rețelele care au fost instalate în tabela de rutare prin rute statice sau prin protocoale de rutare dinamice**
- **Necunoscute – rețelele pentru care nu a fost găsit nici un „next hop” sau o interfață de ieșire în urma procesului de parcurgere a tabelei de rutare; în cazul definirii unei rute隐式的, ruterul va folosi această rută pentru trimiterea pachetelor destinate respectivelor rețele, altfel, vor fi aruncate**
- **Rută implicită – este ruta spre care se trimit toate pachetele pentru care nu se cunoaște o destinație specifică**

Load balancing



Pot exista mai multe rute cu aceeași metrică și către aceeași rețea

În acest caz pachetele pot fi repartizate în mod egal între rutele respective

- se obține o mai bună repartizare a traficului în rețea

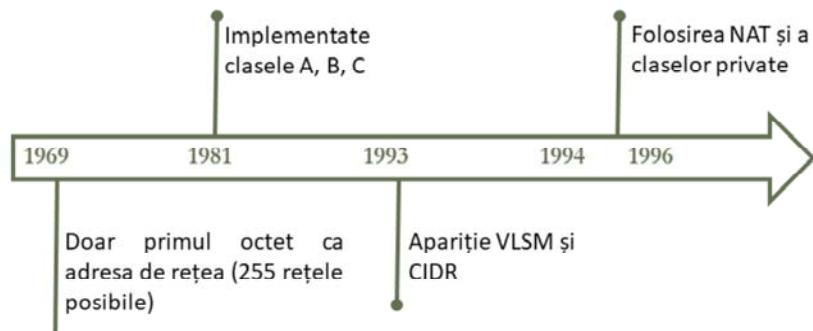
Procesul se numește „Load balancing”

Există situații în care sunt introduse în tabela de rutare mai multe rute către aceeași destinație având aceeași valoare a metrii. În acest caz, ruterul va repartiza pachetele trimise către destinație în mod egal între rutele respective. Astfel, tabela de rutare va conține pentru o anumită rețea destinație mai multe interfețe de ieșire (sau adrese IP next-hop).

Utilizarea corectă a procesului de „load balancing” poate îmbunătății eficiența și performanța rețelei. În cazul în care traficul este împărțit în mod egal între rutele către destinație, ruterul realizează procesul de „equal cost load balancing”, dar există situații în care pachetele pot fi trimise pe căi multiple chiar dacă metrica nu are aceeași valoare. Acest proces este cunoscut sub numele de „unequal cost load balancing” și poate fi realizat în cadrul protocolului de rutare EIGRP.

CIDR (Classless Inter-Domain Routing)
și VLSM (Variable Length Subnet Mask)

Evoluția spațiului de adresare



Atunci când Internetul devinea popular printre companii și instituții de cercetare, se considera că spațiul de adresare pe 32 de biți urma să fie îndeajuns de mare pentru orice evoluție ulterioară a numărului de utilizatori. Adresarea IPv4 este cea mai răspândită adresare de nivel 3 și în ziua de astăzi, permitând un număr de 2^{32} adrese IP existente (aproximativ 4 miliarde).

Dezvoltarea inițială a adresării IP presupunea existența a 3 clase majore: clasa A (mască /8), clasa B (mască /16), clasa C (mască /24).

Datorită numărului limitat de posibilități de adresare, foarte multe companii primeau mult mai multe adrese decât necesar, majoritatea rămânând neutilizate. Soluția a venit sub forma VLSM (Variable-Length Subnet Mask) și CIDR (Classless inter-domain routing), dar și prin NAT (Network Address Translation), concept care permite mai mulți utilizatori să folosească în același timp un număr limitat de adrese IP externe.

Adresarea classful

Clasa A	Network	Host	Host	Host	/8
Clasa B	Network	Network	Host	Host	/16
Clasa C	Network	Network	Network	Host	/24

Clasele au fost create pentru a suporta rețele de diferite dimensiuni

Adresele IP erau alocate inițial în funcție de clasă, existând doar 3 clase majore, cu un număr fix de adrese IP pentru fiecare.

- **Clasa A:** cuprinde adrese de rețea asignabile de la 1.0.0.0 la 126.0.0.0 (0.0.0.0 și 127.0.0.0 sunt rezervate), alocate celor mai mari companii cu nevoi extinse de adresare globală și „data centere”
- **Clasa B:** cuprinde adresele de rețea de la 128.0.0.0 la 191.255.255.255 cu un prefix de rețea /16, fiind alocate companiilor de dimensiuni medii
- **Clasa C:** cuprinde adrese de rețea de la 192.0.0.0 la 223.255.255.255 având un prefix de rețea /24, de obicei alocate companiilor mici

Dezavantajele adresării classful

Este transmisă întreaga adresă de rețea, în loc de subnet

Nu suportă VLSM

Clasele A și B risipeau multe adrese

Clasa C oferea prea puține adrese, în unele cazuri

Clasele A și B risipeau prea multe adrese, majoritatea companiilor nefolosind tot intervalul acordat. Pe de altă parte, clasa C oferea un spațiu de adrese prea mic pentru firmele de dimensiuni medii și mari.

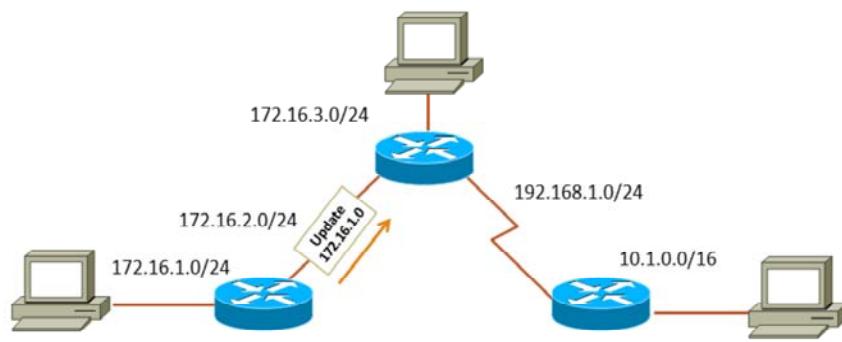
În cazul folosirii unui protocol de rutare classful, este necesară o documentare sporită a rețelei din partea administratorului, pentru a nu se genere inconsistențe la nivelul tabelei de rutare pentru ruterele din rețea.

Deoarece în cazul adresării classful nu există suport pentru VLSM, rețele de dimensiuni mari au o scalabilitate redusă și crește dificultatea de administrare și configurare a acestora.

Rutarea classful

Protocolele de rutare classful nu trimit și masca de rețea a rutei

- folosesc masca specifică fiecărei clase
- pot folosi masca de pe interfața folosită, dacă rețeaua majoră coincide



Rutarea classful este primul mod de rutare și presupune existența în topologie doar a rețelelor care aparțin claselor majore. Deoarece nu se putea aloca alt tip de adrese IP decât cele din clasele A, B sau C, nu era nevoie de o altă metodă de rutare.

Cu toate acestea, masca de rețea aferentă unei anumite rute nu este inclusă în update-urile de rutare, ea fiind determinată de valoarea primului octet a adresei IP primite. Astfel se deduce clasa majoră a cărei mască aferentă este instalată în tabela de rutare.

În cazul în care ruta primită într-un update face parte din aceeași clasă majoră cu adresa IP a interfeței pe care acesta sosește, masca de rețea instalată în tabela de rutare va fi cea de pe interfața respectivă. Problema apare când se folosesc măști de rețea mai mici decât masca clasei majore părinte.

VLSM (Variable Length Subnet Mask)

Permite utilizarea unor măști de rețea de orice dimensiune

- fiecare client primește de la ISP un spațiu de adrese potrivit nevoilor de adresare

Se folosește masca sau prefixul de rețea pentru a specifica portiunea de rețea a adresei

VLSM (Variable Length Subnet Mask) este conceptul care permite utilizarea de rețele cu mască variabilă într-o anumită topologie dată. Introducerea acestui concept alături de rutarea classless a fost necesară datorită dezvoltării exponentiale a internetului și a limitărilor existente în scalabilitatea tabelor de rutare. De asemenea există pericolul epuizării rapide a tuturor adreselor IP dacă se continuă implementarea schemei de adresare classful. Un exemplu în acest caz este scenariul des întâlnit în care o companie de dimensiuni medii avea nevoie de câteva sute de IP-uri (mai mult de 255), iar singura metodă existentă de adresare ar fi fost alocarea unei adrese din clasa B (65534 IP-uri). Astfel mii de IP-uri alocate ar fi rămas nefolosite.

Utilizând VLSM fiecare ISP are posibilitatea să repartizeze un spațiu de adrese în funcție de nevoile de adresare ale fiecărui client, fie el o companie de diferite dimensiuni sau o persoană fizică.

CIDR (Classless Inter-Domain Routing)

Standard ce impune routerelor să transmită și masca de rețea în pachetele de update

Permite suprarețea (agregarea rețelelor) adreselor la orice suprarețea, indiferent de clasa rețelei

- agregarea rețelelor în modul classful forță agregarea la dimensiunea clasei aferente

Prin agregarea rutelor se obțin tabele de rutare mai mici

Exemple de protocoale classless:

- RIPv2, EIGRP, OSPF, IS-IS, BGP

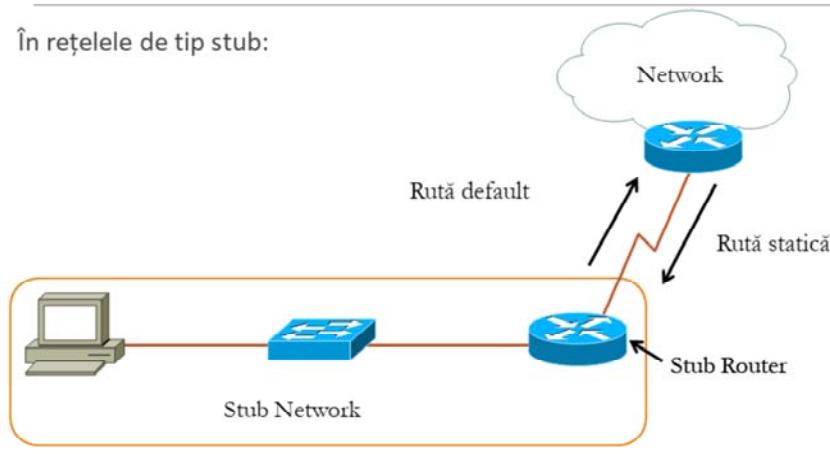
Conceptul de CIDR (Classless Inter-Domain Routing) a fost introdus pentru a eficientiza modul de utilizare al spațiului de adresare IP disponibil. CIDR permite realizarea sumarizării rutelor (procesul de anunțare a mai multor adrese prin o singură rută cu o mască de rețea mai puțin specifică). Din concepțile însușite de la protocolul RIPv1, se cunoaște faptul că rutele incluse în update-uri sunt sumarizate la o rețea dintr-o clasă majoră.

În cazul CIDR, nu mai există limitarea legată de utilizarea claselor majore, permitând sumarizarea la o rută cu o mască de rețea mai mică decât masca de rețea classful. Acest tip de sumarizare reduce numărul de rute incluse în update-uri și implicit dimensiunile tabelei de rutare. La scurt timp după apariția RIPv1 au fost dezvoltate protocoale de rutare dinamice mai performante și mai complexe capabile să suporte VLSM și CIDR: RIPv2, EIGRP, OSPF, IS-IS, BGP.

Rutarea statică

Rol rute statice

În rețelele de tip stub:



O rețea stub este o rețea care poate fi accesată doar printr-o singură rută. Astfel, în exemplu, dacă host-ul vrea să acceseze o destinație din afara rețelei sale, singurul mod de a face acest lucru este prin ruta R1-R2. De asemenea, dacă un host din afara rețelei stub vrea să acceseze un dispozitiv din interiorul rețelei, va putea face acest lucru numai prin intermediul rutei R2-R1.

În astfel de situații, folosirea unui protocol de rutare între cele două rutere ar fi redundant, deoarece există un singur mod prin care R1 poate trimite pachete în afara rețelei stub. Așadar, se va configura câte o rută statică pe fiecare ruter: o rută statică implicită din rețeaua stub spre ruterul vecin, iar apoi o rută statică de pe ruterul vecin spre rețeaua stub.

Principii de rutare

Fiecare ruter ia deciziile de rutare independent, bazându-se pe informațiile aflate în tabela sa de rutare

Dacă un ruter are anumite informații în tabela de rutare nu înseamnă că alte rutere au aceeași informație

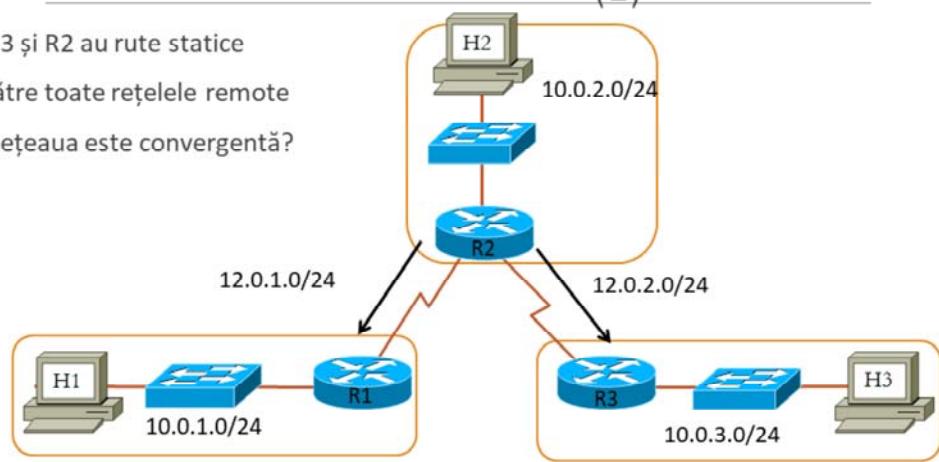
Informațiile de rutare despre o cale nu trebuie să fie aceleași pentru calea de întoarcere

Presupunând că un ruter R1 conține în tabela sa de rutare o serie de rute spre diverse destinații, orice decizie de transmitere a pachetelor va fi luată pe baza informațiilor deținute. R1 nu consultă tabelele de rutare ale vecinilor și nici nu cunoaște dacă ruterul la care va trimite pachetul are configurață o rută către destinație. În caz că adresa IP destinație a unui pachet face parte dintr-o rețea existentă în tabela de rutare, R1 va cunoaște doar către ce echipament vecin să trimită mai departe pachetul și eventual distanța până la destinație.

Dacă pachetul trimis trece printr-un ruter intermediar R2 pentru a ajunge la destinație, nu se garantează faptul că R2 va folosi aceeași cale de întoarcere prin R1. Există și posibilitatea ca R2 să nu cunoască nici o rută de întoarcere către expeditor. De aceea, este rolul administratorului de rețea să se asigure că toate destinațiile sunt accesibile.

Aplicarea principiilor (1)

R3 și R2 au rute statice
către toate rețelele remote
Rețeaua este convergentă?



Dacă H2 trimite un pachet către H1, acesta va ajunge la destinație deoarece R2 are configurate rute statice către toate rețelele remote. Pachetul va ajunge la R1 care, fiind direct conectat cu H1, va ști să îl transmită. Totuși, dacă H1 vrea să îi răspundă lui H2, pachetul va fi aruncat fiindcă R1 nu are o rută configurată către rețeaua lui H2. Se respectă aşadar principiile de rutare: dacă R2 și R3 au rute configurate către toate rețelele remote, nu înseamnă că R1 știe despre acestea. Astfel, dacă R2 are o rută către rețeaua lui R1 nu înseamnă că R1 va ști să transmită un răspuns către R2. Aceeași problemă este valabilă și în cazul comunicației între dispozitivele H2 și H3.

În concluzie, nu există conectivitate între oricare două puncte ale rețelei, rețeaua nefiind convergentă. Soluția optimă pentru rezolvarea problemei de conectivitate este configurarea unei rute statice pe ruterele R1 și R3 cu destinația 10.0.2.0/24 sau a unei rute implicate spre ruterul R2.

Aplicarea principiilor (2)

Tabelele de rutare corespunzătoare topologiei

```
R1#show ip route
***output omitted***
10.0.1.0/24 is subnetted, 1 subnets
C      10.0.1.0 is directly connected, FastEthernet0/0
12.0.0.0/24 is subnetted, 1 subnets
C      12.0.1.0 is directly connected, Serial1/0
```

```
R2#show ip route
***output omitted***
10.0.0.0/24 is subnetted, 3 subnets
S      10.0.1.0 is directly connected, Serial1/0
C      10.0.2.0 is directly connected, FastEthernet0/0
S      10.0.3.0 is directly connected, Serial1/1
12.0.0.0/24 is subnetted, 2 subnets
C      12.0.1.0 is directly connected, Serial1/0
C      12.0.2.0 is directly connected, Serial1/0
```

Vizualizarea rutelor configurate se face prin afișarea conținutului tabelei de rutare. Astfel, comanda `show ip route` oferă multiple informații despre rutele existente:

- tipul rutei, identificat printr-un caracter, de exemplu caracterul S înseamnă rută statică, iar caracterul C înseamnă rețea direct conectată; orice alt caracter alfabetic diferit de S sau C semnifică protocolul de rutare dinamic prin care ruta a fost introdusă în tabela de rutare
- adresele rețelelor din tabela de rutare împreună cu masca de rețea folosită; masca de rețea va fi afișată în dreptul rețelei classful părinte sau în dreptul fiecărei rute
- tipul interfeței de ieșire sau adresa IP a următorului hop; în unele cazuri, vor fi afișate ambele elemente (âtât interfața de ieșire cât și adresa IP a următorului hop)

Aplicarea principiilor (3)

```
R3#show ip route
***output omitted***
10.0.0.0/24 is subnetted, 3 subnets
S      10.0.1.0 is directly connected, Serial1/0
S      10.0.2.0 is directly connected, Serial1/0
C      10.0.3.0 is directly connected, FastEthernet0/0
12.0.0.0/24 is subnetted, 2 subnets
S          12.0.1.0 is directly connected, Serial1/0
C          12.0.2.0 is directly connected, Serial1/0
```

Din output-ul comenzii `show ip route`, introdusă pe ruterele R2 și R3, se observă faptul că acestea au configurat rute către toate rețelele din topologie, spre deosebire de R1 care cunoaște numai rețelele direct conectate cu acesta.

În momentul în care R1 trebuie să trimită un pachet drept răspuns la conexiunea inițializată cu ruterul R2, adresa IP destinație va face parte din rețeaua 10.0.2.0/24. Analizând tabela de rutare a ruterului R1, se observă că nu există nici o rută definită către această rețea. În absența configurației unei rute implicate, pachetul va fi ignorat, deci nu există conectivitate între rețeaua 10.0.1.0/24 și alte rețele remote.

Căutare recursivă în tabela de rutare

Procesul are loc doar la instalarea rutei în tabela de rutare

Ruta statică este specificată prin următorul hop



Se caută următorul hop în tabela de rutare

```
10.0.0.0/24 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
S         10.0.1.0 [1/0] via 10.0.0.2
```

Pentru ca un pachet să fie trimis mai departe de către un ruter, acesta trebuie, mai întâi, să găsească o cale a cărei adresă de rețea să corespundă adresei IP destinație a pachetului. Dacă un ruter primește un pachet destinat unei rețele care nu este direct conectată, acesta va căuta în tabela de rutare rețeaua destinație, iar apoi interfața pe care trebuie să transmită pachetul. Când ruterul trebuie să desfășoare mai multe căutări în tabela de rutare înainte să transmită un pachet, efectuează un proces cunoscut sub numele de căutare recursivă.

Eliminarea procesului de căutare recursivă se poate face prin definirea unei rute statice prin interfața de ieșire către destinație. Astfel, pentru descoperirea căii pe care un ruter trebuie să transmită un anume pachet se va realiza doar o singură căutare în tabelă. În cazul definirii unei rute cu adresa IP a următorului hop, ruterul va mai realiza o căutare în tabelă pentru a descoperi interfața de ieșire atașată acestuia.

Interfață de ieșire căzută

Ruta statică este ștearsă din tabela de rutare dacă interfața de ieșire pentru aceasta nu funcționează

```
RT: interface FastEthernet0/0 removed from routing table
RT: del 10.0.0.0/24 via 0.0.0.0, connected metric [0/0]
RT: delete subnet route to 10.0.0.0/24
RT: del 10.0.1.0/24 via 10.0.0.2, static metric [1/0]
RT: delete subnet route to 10.0.1.0/24
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
```

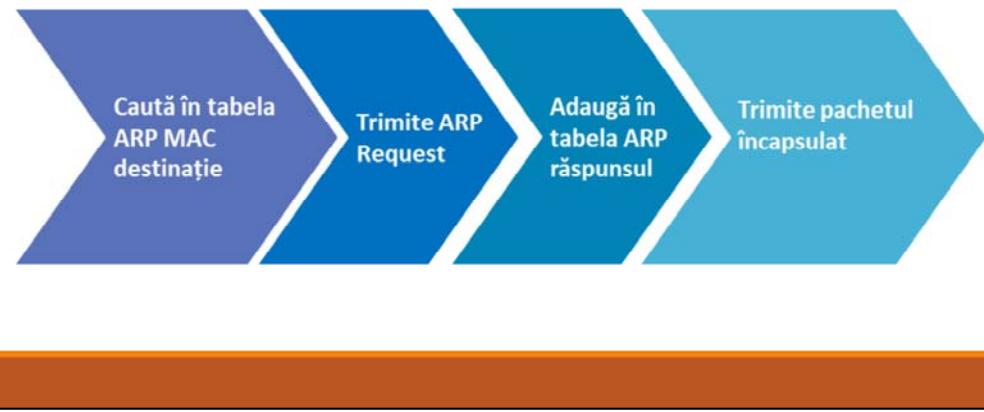
Se poate întâmpla, din diverse motive, ca o interfață să devină inutilizabilă. În acest caz, rutele statice care aveau ca interfață de ieșire pe cea căzută vor fi șterse din tabela de rutare. Pentru instalarea și menținerea unei rute în tabela de rutare trebuie să existe în prealabil o configurație IP pentru cel puțin o interfață activă.

Procesul de ștergere a unei rute statice se poate urmări cu ajutorul comenzi debug ip routing. Se observă faptul că toate rutele care aveau ca interfață de ieșire pe cea căzută sunt șterse din tabela de rutare. Aceste rute vor fi reinserate în tabela de rutare doar dacă interfața va redeveni funcționabilă.

O rută poate fi configurată în aşa fel încât să aibă asociată o interfață de ieșire, indiferent dacă rețeaua este direct conectată sau nu. Acest lucru micșorează timpul de căutare a căii destinație în tabela de rutare.

Rute statice

Un router care decide să trimită pachete la următorul hop precizat într-o rută statică trebuie să seteze adresa MAC destinație a pachetului



În situația în care între două rutere există o conexiune de tip Ethernet, cadrul unui pachet va include câmpuri pentru adresarea MAC.

Când un ruter trebuie să trimită un pachet pe o interfață Ethernet, el va căuta adresa MAC corespunzătoare IP-ului destinație sau a ruterului „next-hop” în tabela sa ARP. Dacă nu este găsită nici o corespondență, ruterul va trimite un ARP request pe interfața Ethernet. Acest request este, de fapt, un broadcast care cere adresa MAC a dispozitivului destinație sau a „next-hop”-ului. Răspunsul va fi un pachet de tip ARP reply ce conține adresa MAC căutată, informație ce va fi introdusă în tabela ARP a dispozitivului care a solicitat request-ul. Pachetul este apoi încapsulat, folosind adresa MAC obținută, și trimis mai departe.

Rețelele seriale (point-to-point) conțin numai două dispozitive legate între ele, deci nu vor avea nevoie de o adresă de nivel 2 în momentul în care se trimit un pachet pe o interfață serială.

Ruta default

Adăugând o rută default pachetele nu vor mai fi aruncate

- orice pachet face match pe ruta default

Când se folosește?

- când nici o altă rută nu decide rutarea unui pachet
- când un ruter are un singur punct de ieșire spre restul rețelei (stub router)

La primirea unui pachet, un ruter va compara adresa destinație cu adresele pe care le conține în tabela de rutare, verificând astfel dacă aceasta face parte din cadrul unei rețele cunoscute. Pachetul va fi trimis pe ruta cea mai specifică. De exemplu, dacă un pachet are destinația 192.168.0.3 și ajunge la un ruter care are în tabela sa de rutare rețelele: 192.168.0.0/24 și 192.168.0.0/16, pachetul va fi trimis spre prima rețea, deoarece 24 de biți se potrivesc cu adresa destinație, în comparație cu 16 biți în cazul celei de-a doua rețele.

O rută default este o rută care se potrivește oricărei destinații. Astfel, în momentul în care nici o rută din tabela de rutare nu se potrivește cu adresa destinație, pachetul va fi trimis pe ruta default în loc să fie aruncat. De asemenea, ruta default poate fi folosită și în cazul unei rețele de tip „stub”, deoarece orice pachet va fi trimis pe o singură cale de ieșire.

Comanda show interfaces

Verificarea configurației interfețelor

- `#show interfaces [tip_interfață număr_interfață]`

```
R1#show interfaces fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is cc00.140c.0000 (bia cc00.140c.0000)
  Internet address is 10.0.0.1/24

R1#show interface serial 1/0
Serial1/0 is administratively down, line protocol is down
```

Comanda `show interfaces` oferă informații detaliate despre starea interfețelor existente pe un ruter. Comanda poate fi introdusă fără parametri suplimentari, caz în care va afișa, sub forma unei liste, detalii despre toate interfețele instalate pe echipament, sau se poate specifica denumirea unei interfețe ca argument pentru un output mai specific. În primul rând, comanda va verifica dacă interfața este activă și dacă protocolul de nivel 2 funcționează. În output se mai afișează și adresa IP asociată interfeței, adresa MAC și modelul fizic al acesteia.

Pentru un output mai succint și mai bine organizat se poate folosi comanda `show ip interface brief`. Informațiile afișate astfel reprezintă o modalitate utilă pentru verificarea funcționalității interfețelor și corectitudinii configurației adreselor IP.

Comanda show interfaces (2)

Probleme Layer1

- cablu deconectat

```
R1#show interface serial 1/0
Serial1/0 is down, line protocol is down
```

Probleme Layer2

- interfața serială nu primește semnal de ceas

```
R1#show interface serial 1/0
Serial1/0 is up, line protocol is down
```

Comanda show interfaces poate fi folosită cu succes pentru depanarea problemelor de conectivitate dintr-o rețea, datorate unei interfețe inactive sau unor inconsistențe în configurarea IP-urilor.

Dacă în output este specificat că interfața este „down” atât la nivel de linie, cât și la nivel de protocol, înseamnă fie că un cablu este deconectat sau defect, fie că interfața dispozitivului de la celălalt capăt al legăturii este în modul „shutdown”.

În cazul în care output-ul indică doar protocolul de linie ca fiind „down”, acest lucru reprezintă o problema la nivelul 2, de exemplu faptul că nu a fost setată valoarea clock-rate-ului pentru sincronizarea interfețelor seriale HDLC.

Examinarea interfețelor seriale

Verificarea DCE/DTE

- `#show controllers [tip_interfață număr_interfață]`

```
R1#show controllers serial 1/0
M4T: show controller:
PAS unit 0, subunit 0, f/w version 1-45, rev ID 0x2800001, version 1
idb = 0x64090F0C, ds = 0x64091FD4, ssb=0x64092390
Clock mux=0x0, ucmd_ctrl=0x0, port_status=0x7B
Serial config=0x8, line config=0x200
maxdgram=1608, bufpool=78Kb, 120 particles
DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
line state: down
cable type : V.11 (X.21) DCE cable, received clockrate 2015232
```

Conexiunile seriale sunt realizate considerând un capăt al legăturii de tip DCE (Data Communications Equipment) și celălalt capăt de tip DTE (Data Terminal Equipment). Interfețele seriale Cisco sunt, în mod normal, DTE, dar pot fi configurate pentru a se comporta ca DCE.

Pentru a configura interfața unui ruter ca DCE se conectează capătul DCE al cablului la interfață, pe care apoi se va stabili o valoare numerică pentru clockrate.

În general, conectorii cablurilor seriale sunt marcați vizual ca fiind de tip DCE sau DTE. Un alt mod de a deosebi cele două modele este faptul că DTE are conector de tip „male”, iar cel DCE, de tip „female”.

Comanda folosită pentru a vizualiza dacă un capăt al unei conexiunii seriale este de tip DTE sau DCE este `show controllers` menționând ca parametru identificatorul interfeței respective.

Configurarea interfețelor seriale

Configurarea parametrului **clock-rate** pe interfețele seriale

- interfețele seriale necesită configurarea vitezei de comunicație (clockrate), pentru a putea funcționa
- echipamentul care dă tactul de ceas trebuie să fie DCE
- **(config-if) #clock rate <valoare>**

Configurarea parametrului clock-rate, adică a vitezei de transmisie a datelor pentru sincronizarea echipamentelor de la cele două capete ale legăturii seriale va avea efect numai în cazul interfețelor de tip DCE. Dacă se setează o valoare pentru clock-rate pe interfața DTE, sistemul de operare va ignora comanda introdusă.

Valorile numerice care pot fi atribuite clock-rate-ului (în biți pe secundă) sunt: 1.200, 2.400, 9.600, 19.200, 38.400, 56.000, 64.000, 72.000, 125.000, 148.000, 500.000, 800.000, 1.000.000, 1.300.000, 2.000.000, 4.000.000 sau 8.000.000. Nu este necesară reținerea valorilor exacte a ceasului, deoarece în cadrul configurației acesteia, poate fi introdusă orice valoare non standard între 300 și 8.000.000, iar sistemul de operare va ajusta numărul introdus la cea mai apropiată valoare suportată de către echipamentul hardware. În mod implicit, o interfață DCE nu are configurat semnalul de ceas, acest lucru fiind semnalat prin starea „down” a protocolului de linie.

Troubleshooting

Verificarea introducerii rutelor în tabela de rutare

- `#debug ip routing`

```
R1#debug ip routing
IP routing debugging is on
R1#configure terminal
R1(config)#interface fastEthernet 0/0
R1(config-if)#shutdown
RT: interface FastEthernet0/0 removed from routing table
RT: delete subnet route to 10.0.0.0/24
RT: NET-RED 10.0.0.0/24
%LINK-5-CHANGED: Interface FastEthernet0/0,changed state to
administratively down
R1(config-if)#no shutdown
RT: interface FastEthernet0/0 added to routing table
%LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,changed state
to up
```

Spre deosebire de comanda `show`, comanda `debug` este folosită pentru monitorizarea operațiilor efectuate de ruter în timp real. Comanda `debug ip routing` urmărește fiecare schimbare făcută de ruter, în procesul de rutare.

Astfel, dacă o interfață din rețea devine inactivă, `debug ip routing` va arăta faptul că orice rută care folosea interfața de ieșire respectivă a fost ștearsă. De asemenea, comanda va afișa și fiecare interfață/rută nou adăugată în tabela de rutare.

Procesele `debug` consumă o mare parte din procesor atunci când sunt activate. De aceea este recomandat să fie folosite numai la depanarea rețelelor și să se păstreze un număr cât mai mic de procese `debug` pornite, dezactivându-se cele care nu sunt necesare.

Pentru a închide toate procesele `debug` activate se folosește comanda `no debug all`, sau comanda cu același efect, `undebbug all`.

Configurarea rutelor statice

Sintaxa comenzi ip route

- (config)#**ip route adresă-rețea subnet-mask {adresă-ip | interfață-de-ieșire}**

Configurarea rutei statice default

- (config)#**ip route 0.0.0.0 0.0.0.0 {adresă-ip | interfață-de-ieșire}**

Nu se pune doar interfața de ieșire în cazul rețelelor multi-acces, trebuie obligatoriu next-hop

Pentru a adăuga o rețea remote în tabela de rutare a unui ruter în mod static, se va folosi comanda **ip route**. Aceasta va primi ca parametri adresa rețelei remote, masca ei de rețea, și adresa ip a ruter-ului „next-hop” sau interfața de ieșire.

Se poate verifica adăugarea noii rute prin activarea procesului debug sau prin folosirea comenzi show ip route după adăugarea rutei.

În cazul unei rețele „stub” este recomandată configurarea unei rute default deoarece pachetele pot ieși din rețea doar pe o singură cale. Ruta default va avea adresa de rețea 0.0.0.0 și masca /0.

În cazul rețelelor multi-access se va configura adresa IP a următorului hop deoarece doar prin configurarea interfeței de ieșire ruterul nu va avea suficiente informații pentru a determina dispozitivul „next-hop”. Așadar, fără a cunoaște ip-ul „next-hop-ului”, ruterul nu va ști ce adresă MAC destinație să încapsuleze în cadrul Ethernet de nivel 2.

Rute statice cu interfețe Ethernet

Rețeaua Ethernet este multi-acces

- nu se poate specifica doar interfața de ieșire pentru că pot exista mai multe destinații pe aceeași interfață
- trebuie să se specifice next-hop

▪ Configurarea rutei

```
R1(config)#ip route 10.0.1.0 255.255.255.0 fastEthernet 0/0 10.0.0.2
```

▪ Afisarea rutei

```
S 10.0.1.0 [1/0] via 10.0.0.2, FastEthernet0/0
```

Spre deosebire de o rețea point-to-point, cu interfețe seriale, care include doar două device-uri (cele două rutere conectate între ele), o rețea Ethernet poate conține, pe lângă rutere, și alte dispozitive (switch-uri, host-uri), ceea ce înseamnă că rețeaua Ethernet este o rețea multi-acces. În aceste condiții, trebuie ca rutele statice către rețelele remote să fie făcute prin adresa IP „next-hop”. Pentru o bună funcționare a rutelor statice configure, se recomandă specificarea ambelor căi de ieșire pentru o anumită rută (interfață de ieșire și adresa IP „next-hop”).

Comanda `ip route` va include la parametri adresa rețelei remote, masca acesteia, interfața pe care se trimite pachetul către rețea și IP-ul interfeței. Specificarea interfeței este optională. În cazul în care nu este specificată interfața, ruterul va căuta rețeaua următorului hop în tabela de rutare și va folosi interfața direct conectată la acesta.

Rutele statice în tabela de rutare

```
R1#show 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 0.0.0.0 to network 0.0.0.0

      172.16.0.0/24 is subnetted, 2 subnets
S       172.16.0.0 is directly connected, Serial1/0
S       172.16.1.0 is directly connected, Serial1/0
      10.0.0.0/24 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
S       10.0.1.0 [1/0] via 10.0.0.2
      12.0.0.0/24 is subnetted, 1 subnets
C       12.0.0.0 is directly connected, Serial1/0
S*     0.0.0.0/0 is directly connected, Serial1/0
```

Tabela de rutare se vizualizează cu ajutorul comenzi show ip route, care produce afişarea tuturor rutelor existente, fie ele direct conectate, învăţate în mod static sau prin protocole de rutare.

Output-ul generat include tipul rutei, adresa rețelei accesibile prin ruta menționată, tipul conexiunii, interfața pe care se realizează conexiunea și, dacă este cazul, metrica. Tipul unei rute existente în tabelă este specificat prin prezența unui caracter alfabetic în dreptul fiecărei rute. Astfel, o rută statică este indicată de caracterul „S” a cărui semnificație este prezentată în legenda afișată în prima parte a output-ului comenzi introduse.

În cazul configurației unei rute statice default, aceasta va fi indicată de caracterul „S” urmat de caracterul „*” la începutul liniei pe care este afișată, dar și de prezența mesajului Gateway of last resort is 0.0.0.0 to network 0.0.0.0.

Modificarea unei rute statice

Nu se poate modifica o rută statică deja creată

- se va șterge și se va crea o alta

```
R1(config)#no ip route 172.16.1.0 255.255.255.0 serial 1/0  
R1(config)#ip route 172.16.1.0 255.255.255.0 172.16.0.1
```

Odată creată o rută statică, singurul mod în care i se pot aduce modificări este ștergerea și recrearea acesteia. Pentru a șterge o rută statică se folosește aceeași comandă prin care a fost adăugată, precedată de negația no. Se va crea, apoi, o nouă rută statică specificând modificările dorite.

Verificarea configurării corecte a rutelor se poate face prin două metode: afișarea fișierului de configurare „running-config” sau vizualizarea tabelei de rutare. Comanda introdusă pentru setarea unei rute statice va fi prezentă în running-config chiar dacă ea nu apare în tabela de rutare (posibil din cauză că interfața atașată ruterului nu este activă, sau nu are configurată în prealabil o adresă IP).

Rute statice summarizate

Se sumarizează, acolo unde este posibil, pentru a avea tabele de rutare cu mai puține intrări

- rutele nesumarizate trebuie șterse

```
ip route 172.16.1.0 255.255.255.0 Serial0/0/1  
ip route 172.16.2.0 255.255.255.0 Serial0/0/1  
ip route 172.16.3.0 255.255.255.0 Serial0/0/1
```



```
ip route 172.16.0.0 255.255.252.0 Serial0/0/1
```

Conceptul de summarizare a fost introdus pentru a ajuta la micșorarea tabelelor de rutare, eficientizând astfel procesul de dirijare a pachetelor. Acest concept constă în reducerea mai multor rețele mici la o rețea mai mare, păstrând astfel o singură rețea echivalentă în tabela de rutare. Este important de reținut faptul că summarizarea se face doar pentru acele rute care au asociată aceeași interfață de ieșire sau aceeași adresă IP „next-hop”.

Procesul de summarizare se face conform următoarelor reguli:

- **Se scriu adresele de rețea în binar**
- **Se numără biții comuni de la stânga la dreapta**
- **Masca noii rețele va reprezenta numărul de biți comuni între rețelele inițiale**
- **Rețeaua rezultată reprezintă rețeaua summarizată a rutelor inițiale**

Troubleshooting pentru rute statice

Afișarea tabelei de rutare

- `show ip route`

Afișarea statusului interfețelor de pe router

- `show ip interface brief`

Verificarea conectivității Layer2 cu vecinii

- `show cdp neighbors detail`

Testarea accesului între sursă și destinație

- `ping`

Testarea traseului de la sursă la destinație

- `traceroute`

Există multe probleme care pot apărea într-o rețea, de la căderea unei interfețe până la o comandă greșit introdusă de administrator. În aceste cazuri conectivitatea rețelei poate fi ușor compromisă. Rolul administratorului de rețea este să rezolve astfel de potențiale situații apărute. În acest scop există mai multe unele care pot ajuta la depanarea rețelei:

- `show ip route` – oferă informații detaliate despre starea interfețelor și a rutelor active din tabela de rutare
- `show ip interface brief` – afișează sumar starea interfețelor
- `show cdp neighbours detail` – oferă informații detaliate despre toate dispozitivele direct conectate cu echipamentul local
- `ping` - testează conectivitatea dintre două dispozitive
- `traceroute` – identifică locația unde se poate bloca un pachet între sursă și destinație, afișând adresele parcurse de pachet

Mesaje de logging

Mesaje privind schimbări în configurație, erori, alerte, etc.

Sincronizarea afișării logurilor cu promptul

- (config-line) #**logging synchronous**

```
R1(config)#interface fastEthernet 0/0
R1(config-if)#shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to
    administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
    changed state to down
R1(config-if) #
```

De multe ori, sistemul de operare afișează diverse mesaje informative fără a fi solicitate de administrator (schimbarea descrierii unei interfețe generează un mesaj). Aceste mesaje pot crea unui administrator de rețea posibile dificultăți de vizualizare în momentul introducerii diferitelor comenzi de configurare. Deși aceste mesaje nu afectează comenziile utilizatorului în nici un fel, ele pot fi derutante, lucru pentru care se obișnuiește să se separe mesajele sistemului de operare de comanda care este scrisă. Acest lucru se realizează automat după introducerea comenzi **logging synchronous** în modul config-line accesat prin comanda **line console 0**.

Introducere în protocoale de rutare dinamice

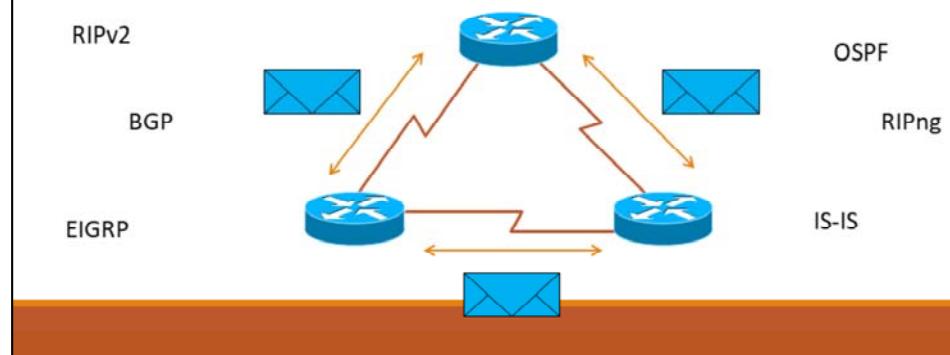


Funcțiile unui protocol de rutare dinamic

Un protocol de rutare are multiple funcții:

- partajarea dinamică a informațiilor între rutere
- adaptarea la schimbările din topologia rețelei
- determinarea căii cele mai bune spre fiecare destinație

Update-uri
de rutare
metrică



Un protocol de rutare este un set de procese, algoritmi și mesaje folosite pentru a schimba informații de rutare și a popula tabela de rutare cu căile cele mai bune către destinație.

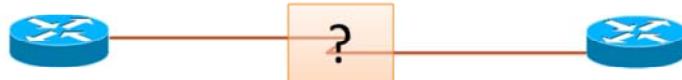
Protocolele de rutare dinamice determină calea optimă în fiecare rețea folosind algoritmi interni, cale care apoi este adăugată în tabela de rutare. Există posibilitatea ca același protocol de rutare să furnizeze mai multe rute către aceeași destinație. Pentru ierarhizarea acestora, fiecare rută are asociată o valoare numerică numită metrică, ce indică o apreciere calitativă a drumului până la destinație.

Un rol important al protocolelor de rutare dinamice este adaptarea la schimbările apărute în topologie fără intervenția administratorului. În momentul în care este detectată o modificare a rețelei, protocolul de rutare actualizează rapid informațiile proprii și în același timp își anunță vecinii de schimbările survenite.

Protocole de rutare - soluții

Un protocol de rutare eficient și scalabil trebuie să aducă soluții la următoarele probleme:

- cum se pot menține informațiile mereu actualizate în tabelele de rutare?
- cum se determină cea mai bună cale spre o destinație?
- cât de repede poate să propage protocolul o modificare apărută în rețea?
- cât de repede poate să găsească protocolul o cale alternativă spre o destinație?



Principalele probleme care trebuie rezolvate de protocolele dinamice de rutare și care determină o anumită ierarhizare în privința performanțelor acestora sunt reprezentate de:

- **modul în care se pot menține informațiile mereu actualizate în tabelele de rutare prin schimbări periodice de mesaje sau prin procese declanșate de modificări în topologie**
- **determinarea celei mai bune căi către destinație prin utilizarea unui algoritm intern în funcție de anumiți parametri**
- **viteza de propagare a unei modificări apărute în rețea și anume diminuarea timpului necesar pentru a anunța o eventuală schimbare către celelalte rutere din topologie, dar totodată și viteza de determinare a unei căi alternative spre destinație în urma procesării datelor generate de modificările apărute în topologie**

Protocole de rutare - componente

Structuri de date

- tabele sau baze de date salvate în RAM

Algoritmul intern

- folosit pentru a determina calea cea mai bună spre destinație

Mesajele protocolului de rutare

- interschimbate de vecini
- folosite pentru:
 - descoperirea vecinilor direct conectați
 - transmiterea de update-uri cu informații de rutare



Toate protocolele de rutare au același scop: să determine cea mai bună cale spre fiecare rețea destinație, și în același timp să mențină informațiile de rutare actualizate de fiecare dată când în rețea are loc o schimbare. Astfel, pentru a permite ruterelor să învețe dinamic rețelele nou conectate, dar și să găsească rute alternative, un protocol de rutare este alcătuit din mai multe seturi de componente dintre care se pot menționa următoarele:

- Structuri de date, reprezentând metode stocare eficientă a informațiilor
- Algoritm folosit de protocolul de rutare, utilizat pentru a determina calea optimă către destinație (exemplu: algoritmul lui Dijkstra)
- Mesajele protocolului de rutare, prin care sunt descoperiți vecinii configurați cu același protocol de rutare, dar prin care este menținută o consistență a informațiilor despre topologie

Recapitulare: Rutarea statică

Avantaje:

- consum minim de resurse (CPU, memorie, lățime de bandă)
- ușor de configurații de depanat (în rețele mici)
- comportament complet previzibil

Dezavantaje:

- orice modificare se execută exclusiv manual (nu scalează)
- nu detectează nicio schimbare în rețea

Utilizări generale:

- rute către diverse zone de rețea unde nu rutează protocoale de rutare
- rute default la marginea rețelei

Rutele statice sunt introduse manual de administrator, spre deosebire de rutele dinamice care sunt generate de un protocol de rutare. O rută statică apare în tabela de rutare doar dacă interfața de ieșire asociată acesteia este activă. Spre deosebire de rutarea dinamică, rutarea statică nu folosește resurse adiționale de lățime de bandă, timp de procesor sau memorie necesare funcționării protocoalelor de rutare.

Un alt avantaj al rutării statice este efortul redus necesar pentru configurarea și administrarea rețelelor de dimensiuni mici, în care implementarea unui protocol dinamic de rutare ar însemna un consum inutil de resurse.

Dimensiunile rețelelor actuale nu permit folosirea exclusivă a rutării statice, dar sunt situații în care folosirea rutelelor statice este necesară, cu scopul de a fi redistribuite apoi în protocoalele de rutare interne, sau de a se asigura conectivitatea în cazul rețelelor de tip „stub”.

Clasificarea protoalelor de rutare

Definiție: Un sistem autonom (AS) reprezintă un grup de rutere aflate sub o administrație comună.

- în multe cazuri, într-un AS rulează un singur protocol de rutare
- un AS poate aparține unei companii, ISP, și este identificat printr-un număr de 16 sau 32 de biți

Clasificarea în funcție de AS-uri:

- protoale **IGP** (Interior Gateway Protocol)
 - rutează doar în interiorul unui AS
 - RIPv2, IS-IS, OSPF, EIGRP
- protoale **EGP** (Exterior Gateway Protocol)
 - rutează informații între AS-uri
 - BGP

Datorită dimensiunii actuale a Internetului, toate protoalele rutate trebuie să suporte o schemă de adresare ierarhică. Astfel rețelele pot fi grupate în sisteme autonome. Un sistem autonom (AS) reprezintă o infrastructură de rețea aflată sub o administrație comună. O administrație comună se referă la un set comun de protoale de rutare, un set de politici de securitate și de criterii de decizie, întâlnită în cazul rețelelor interne ale companiilor sau în cazul ISP-urilor.

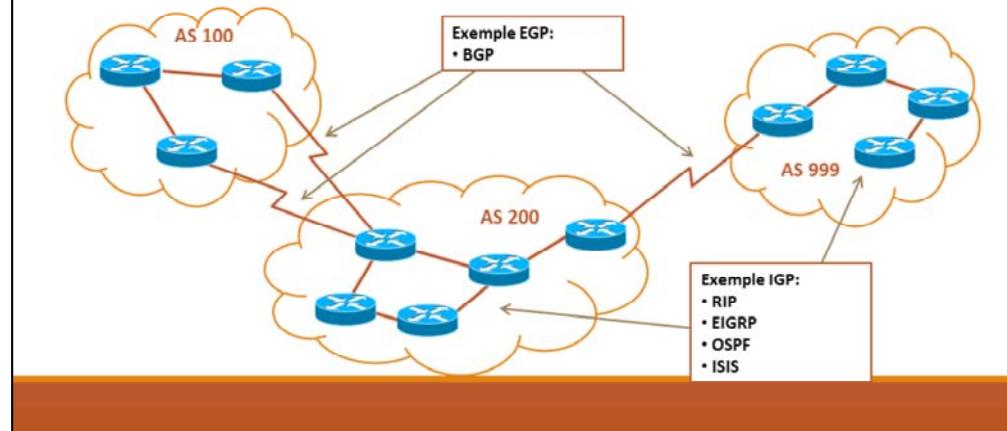
Protoale de rutare pot fi clasificate ca protoale interioare (IGP) sau exterioare (EGP) după modul de funcționare în raport cu AS-ul. Protoale de rutare IGP promovează rețelele în interiorul unui AS, în timp ce protoalele de rutare EGP schimbă informații între AS-uri.

La ora actuală, BGP-ul este singurul protocol de rutare EGP utilizat în internet, fiind un protocol de tip „path vector” (specifică AS-urile prin care trebuie să treacă un pachet până la destinație).

IGP & EGP

Distinctia intre IGP-uri si EGP-uri se face pe baza scalabilitatii

- un ruter ce ruleaza BGP trebuie sa suporte intreaga tabela de rutare a Internetului (~ 300.000 de rute)



O cerință esențială pentru un protocol de tip EGP este puterea sporită de procesare a unor tabele semnificativ mai mari decât cele întâlnite în interiorul unui AS. O tabelă de rutare în Internet, care este schimbată între două rutere de graniță din sisteme autonome diferite, poate cuprinde aproximativ 180.000 de rute.

O altă caracteristică a protocolelor de tip EGP este cea de flexibilitate, BGP-ul folosind un algoritm complex de comparare a două sau mai multe rute.

Pe de altă parte, cerințele de convergență pentru un EGP sunt destul de reduse, datorită faptului că legăturile de nucleu sunt foarte stabile. Astfel, timpul de convergență pentru BGP este de ordinul orelor mai degrabă decât al minutelor.

Distance-vector și Link-state

Distance vector:

- rutele sunt descrise prin *distanță* și *direcție*
- ruterele nu au o vedere completă a topologiei
- folosesc update-uri periodice
- update-urile nu țin cont de informațiile deja trimise anterior
- se trimit tabelele de rutare întregi

Link state:

- ruterele dețin o vedere topologică completă a rețelei
- ruterele vecine mențin adiacențe
- update-urile sunt trimise doar când e necesar
- update-urile pot descrie doar modificările apărute în rețea

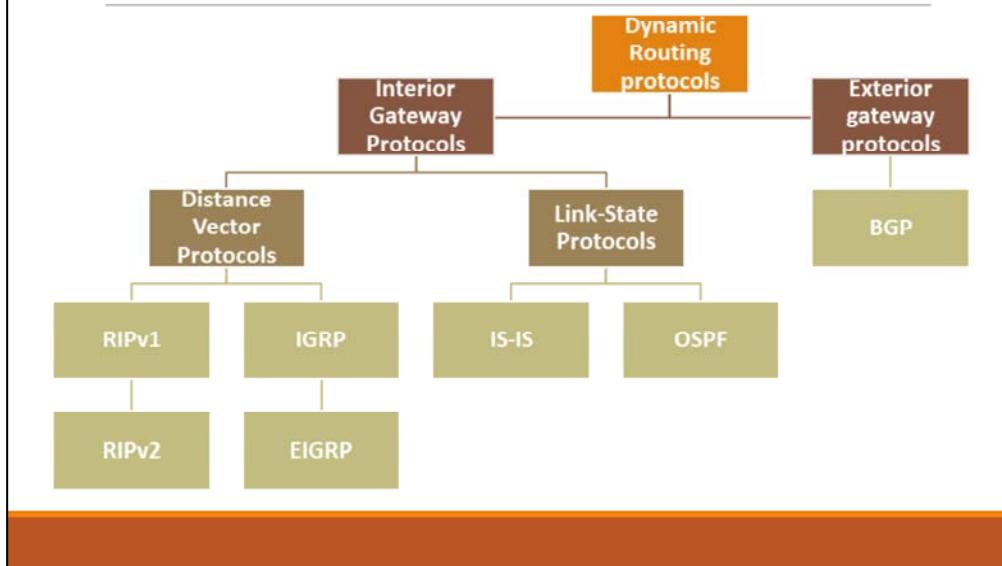
BGP e un caz special, considerat un protocol „path vector”

Protoalele de rutare IGP pot fi clasificate în funcție de modul de învățare a rutelor în protoale de rutare Distance-vector și Link-state.

Primele protoale de rutare IGP de tip distance vector folosesc algoritmi de tip Bellman-Ford pentru a-și construi tabela de rutare. Acestea promovează rutele ca vectori de distanță și direcție, trimițând periodic rutelor vecine întreaga tabelă de rutare. Distanța poate fi numărul de hopuri (RIP), iar direcția unei rute, adresa IP a echipamentului „next-hop”.

Protoalele de rutare link-state rezolvă câteva din limitările protoalelor distance-vector. Ruterele trimit informații care sunt propagate către toate ruterele din topologie pe măsură ce se modifică stările link-urilor. Fiecare ruter calculează căile optime către fiecare destinație, creând un arbore de cost minim cu el însuși ca rădăcină și având astfel o imagine de ansamblu asupra rețelei.

Clasificare (rezumat)



Caracteristicile relevante ale unui protocol de rutare determină dacă acesta este:

- distance-vector, link-state sau hibrid – în funcție de modul de învățare al rutelor
- interior sau exterior – în funcție de utilizare în rețele private sau Internet
- classless (permite CIDR) sau classful – permite agregarea de rute (supernetare) în schimbul de informații dintre rutere
- capabil să proceseze măști de rețea de lungime fixă sau variabilă (VLSM) – VLSM permite conservarea de adrese într-o rețea
- uniform sau ierarhic – determină scalabilitatea de adresare în rețele extinse IPv4 sau IPv6, protocole de rutare noi fiind utilizate în rețelele IPv6 (RIPng)

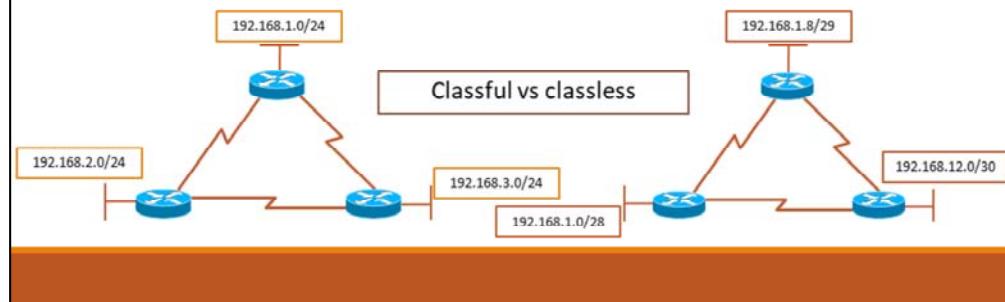
Classful vs classless

Protocole de rutare **classful**

- NU TRIMIT masca de rețea în update-urile de rutare; exemple: RIPv1, IGRP
- cu ce mască va introduce ruterul o astfel de rută în tabelă?

Protocole de rutare **classless**

- TRIMIT masca de rețea în update-urile de rutare; exemple: RIPv2, EIGRP, OSPF, ISIS, BGP
- ce avantaje prezintă un astfel de comportament?



La începutul anilor '90, o rută nu conținea mască de rețea atașată, întregul proces de rutare fiind **classful**. Astfel, o primă clasificare a protocolelor de rutare se poate face în funcție de tipul procesului de rutare, **classful** sau **classless**. Odată cu trecerea timpului, dezvoltarea Internetului a dus la utilizarea tabelelor de rutare **classless**.

În procesul de rutare **classful**, se evaluează primul octet din adresa IP destinație extrasă din antetul unui pachet ajuns la ruter, determinându-se astfel clasa de adrese și implicit masca de rețea.

În mod implicit, există și situația în care o rută face parte din același supernet **classful** al rețelei de pe interfața de pe care a fost primită, caz în care ruterul îi va atribui masca configurată pe propria sa interfață.

Protocolele **classless** au multiple avantaje, cum ar fi: în tabela de rutare pot apărea rute discontinue, masca de rețea este inclusă în update-urile de rutare, suportă anunțarea rețelelor de tip VLSM.

Convergența unui protocol de rutare

Un protocol de rutare este în stare de **convergență** atunci când toate tabelele de rutare au o stare consistentă în raport cu topologia

Timpul de convergență este un factor important în alegerea unui protocol de rutare

- reflectă viteza cu care protocolul răspunde la modificările din rețea
- timpii de convergență mari pot crea (temporar) erori logice în rețea:
 - bucle de rutare
 - black-hole routing
 - rutare suboptimală

Convergență lentă: RIP, IGRP

Convergență rapidă: OSPF, EIGRP, ISIS



O rețea este convergentă atunci când tabelele de rutare conțin informații consistente despre rutele spre toate destinațiile existente. În general, o rețea este inutilizabilă sau dificil de controlat în timp ce converge, de aceea protoalele de rutare urmăresc să atingă un timp de convergență cât mai mic. În cazul unei convergențe lente, datorită inconsistenței tabelelor de rutare, în rețea pot apărea erori logice: bucle de rutare, „black-hole routing” sau rutare suboptimală. Convergența depinde de fiecare ruter în parte, deoarece ruterele ce participă într-un protocol de rutare trebuie să calculeze independent căile cele mai bune spre toate destinațiile cunoscute.

Astfel, protoalele de rutare pot fi clasificate în funcție de viteza de convergență:

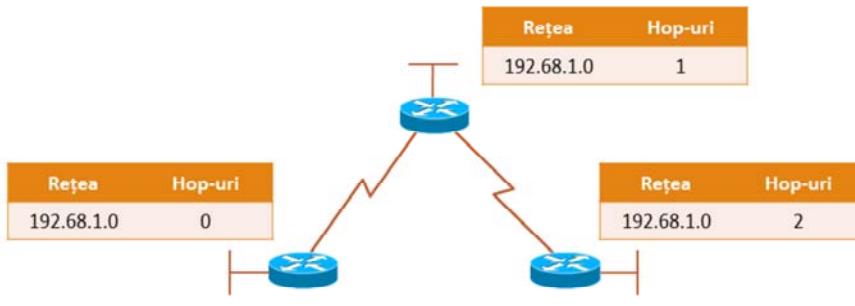
- RIP și IGRP au un timp de convergență redus
- EIGRP și OSPF au un timp de convergență rapid

Metrica unei rute

Metrica reprezintă o valoare folosită de protocolele de rutare pentru a decide care rute sunt mai „bune” decât altele

Exemplu simplu de metrică bazată pe hop-count:

- metoda este implementată în RIP



Un protocol de rutare poate să furnizeze două sau mai multe rute către aceeași destinație și astfel este necesară specificarea unui mecanism de comparare a rutelor între ele. În acest scop este folosită metrica.

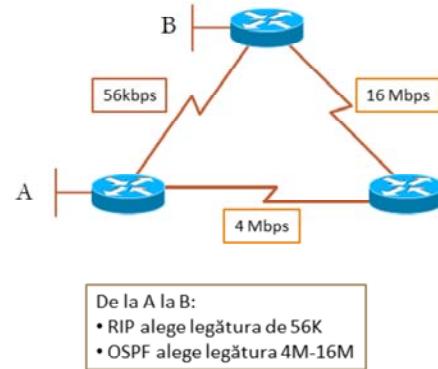
Metrica unei rute reprezintă un număr, rezultat din aprecierea calității unui drum spre o anumită destinație conform unor criterii, specific fiecărui protocol de rutare. Astfel, nu are sens compararea metricilor unor rute obținute prin protocole de rutare diferite.

În funcție de algoritmul intern de determinare a căii optime caracteristic fiecărui protocol, metrica este aleasă depinzând de parametrii mai mult sau mai puțin complecși. RIP spre exemplu folosește o metrică simplă ce determină numărul de rutere pe care ruta respectivă le-a traversat înainte de a ajunge în punctul curent. Alte protocole mai avansate pot folosi metrici complexe care să includă: lățimea de bandă („bandwidth”) sau încărcarea unei legături („load”).

Tipuri de metrii

Protoalele de rutare folosesc diferite valori pentru a măsura rutele:

- hop count
- bandwidth
- cost
- delay
- load
- reliability
- diverse combinații ale acestora



Așadar, există diverse tipuri de metrii utilizate de protoalele de rutare:

- Hop count: numărul de rutere traversate de un pachet până la destinație
- Bandwidth: este preferată calea către destinație cu cea mai mare lățime de bandă
- Load: capacitatea traficului utilizat pe o legătură
- Delay: timpul necesar transmiterii unui semnal peste o legătură
- Reliability: probabilitatea unei legături de a deveni inactivă
- Cost: o valoare care indică preferința pentru o anumită rută

Metrica folosită de RIP este cea a numărului de hop-uri, în timp ce EIGRP folosește o metrică compusă având următoarele componente: „bandwidth”, „delay”, „reliability” și „load”.

Metrica în tabela de rutare

Metrica este relevantă doar în raport cu protocolul de rutare care a generat-o

Ruterele compară doar metricile acelaiași protocol de rutare

```
Router# show ip route
Gateway of last resort is not set

    172.16.0.0 255.255.0.0 is variably subnetted, 4 subnets, 2 masks
O      172.16.0.21 255.255.255.255
          [110/51] via 172.18.1.2, 00:04:56, Serial0/0/0.100
R      172.16.0.12 255.255.255.255
          [110/2] via 172.18.1.6, 00:04:56, Serial0/1/0
C      172.16.0.11 255.255.255.255 is directly connected, Loopback0
O      172.16.1.4 255.255.255.252
          [110/113] via 172.18.1.6, 00:04:56, Serial0/1/0
```

Protoalele de rutare determină cea mai bună cale către destinație folosind ruta cu cea mai mică valoare a metricii.

Metrica asociată cu fiecare rută poate fi vizualizată în tabela de rutare utilizând comanda `show ip route`, fiind a doua valoare dintre parantezele drepte din dreptul fiecărei înregistrări. În output se pot observa rute învățate prin protoale de rutare diferite. În cazul **protocolului de rutare RIP**, rețeaua destinație 172.18.1.12 se află la două hopuri distanță. Metrica are rol de apreciere a calității unui drum către o anumită destinație doar în cadrul unui anumit protocol de rutare, fiind inutilă compararea metricilor a două rute învățate prin protoale de rutare diferite. Ierarhizarea diferitelor protoale de rutare se realizează cu ajutorul primei valori menționate între parantezele drepte din dreptul fiecărei rute – distanță administrativă. La fel ca și în cazul metricii, va fi preferat protocolul de rutare care are asociată o valoare mai mică a distanței administrative.

„show ip route [address]”

Comanda **show ip route** poate afișa informații specifice pentru fiecare rută

Pentru a afișa doar rutele unui anumit protocol se folosește:

- **show ip route [protocol]**

```
Router#show ip route 10.0.0.1

Routing entry for 10.0.0.1/32
 Known via "isis", distance 115, metric 20, type level-1
 Redistributing via isis
 Last update from 223.191.255.251 on Fddi1/0, 00:00:13 ago
 Routing Descriptor Blocks:
 * 10.22.22.2, from 223.191.255.247, via Serial2/3
   Route metric is 20, traffic share count is 1
   223.191.255.251, from 223.191.255.247, via Fddi1/0
     Route metric is 20, traffic share count is 1
```

Comanda show ip route poate fi apelată împreună cu un parametru suplimentar care specifică o adresă IP a unei rețele destinație din cadrul tabeliei de rutare astfel: show ip route [address]. Din outputul afișat, se pot observa informații detaliate despre ruta primită ca parametru:

- Adresa IP a rețelei, împreună cu masca de rețea atașată
- Modul de învățare a rutei, în cazul de față, prin protocolul dinamic de rutare IS-IS, împreună cu informații specifice protocolului menționat
- Distanța administrativă și metrica rutei
- Interfața de ieșire prin care se vor trimite pachetele a căror IP destinație aparține rețelei definite în cadrul rutei

Load balancing

Equal-cost load balancing

- abilitatea unui ruter de a folosi multiple căi spre aceeași destinație, dacă acestea au metrici egale
- majoritatea protocolelor de rutare pot introduce căi multiple în tabela de rutare

Unequal-cost load balancing

- rutele spre destinație pot avea metrici diferite, dar într-un interval predefinit

```
Router#show ip route
Gateway of last resort is not set

R    192.168.6.0 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
                                [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
```

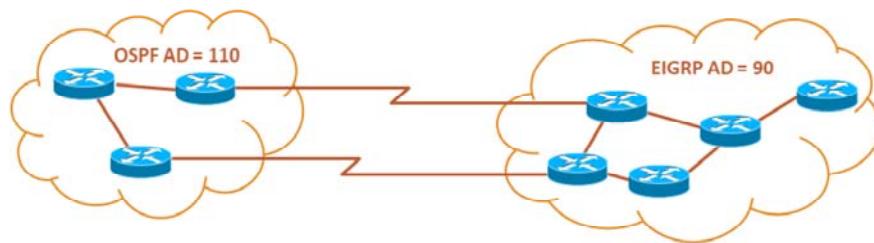
După procesul de construire a tabelei de rutare există posibilitatea existenței mai multor rute către aceeași destinație având atașate aceeași metrică. În acest caz, ruterul va distribui traficul în mod egal către toate interfețele care au asociate rute cu metrici egale. Procesul desfășurat este cunoscut sub denumirea de „equal-cost load balancing” și poate fi vizualizat în tabela de rutare atunci când două sau mai multe rute au asociate aceeași rețele destinație.

O caracteristică importantă a anumitor protocoale de rutare cum este protocolul EIGRP este faptul că poate echilibra traficul pe mai multe rute de cost diferit, ținând cont de o valoare a variației metricilor folosite în procesul de „load balancing”. Mai precis, vor fi introduse în tabela de rutare toate rutele cu metrică mai mică decât valoarea obținută prin înmulțirea valorii variației cu metrica rutei de cost minim.

Distanța administrativă (AD) a unei rute

Distanța administrativă este o valoare ce reprezintă gradul de „preferință” pentru originea unei anumite rute

- rutele statice, direct conectate și cele dinamice au valori AD diferite
- valoarea mai mică este preferată
- metricile sunt comparate doar pentru rute cu AD-uri egale



Două sau mai multe protocoale de rutare diferite pot să furnizeze câte o cale către aceeași destinație, **cu aceeași adresă și mască de rețea** atașată. Criteriul principal de diferențiere a rutelor generate este reprezentat de distanța administrativă (AD). Ruta cu valoarea distanței administrative mai mică este preferată și introdusă în tabela de rutare. Așadar, distanța administrativă este o valoare ce reprezintă gradul de „preferință” pentru originea unei anumite rute. În practică, aceasta determină o ierarhie bine definită a tuturor modurilor posibile prin care o rută poate fi dobândită.

În cadrul aceluiași protocol de rutare, diferențierea rutelor se va face pe baza valorii metricii, ruta cu o valoare mai mică a acesteia fiind introdusă în tabela de rutare.

AD în tabela de rutare

Fiecare rută dinamică afișează [AD / metrică]

Valorile aparțin intervalului [0..255]

Doar rutele direct conectate au AD = 0 (și nu poate fi schimbat)

Un AD = 255 indică o rută ce nu va fi inclusă niciodată în tabela de rutare

```
R1#show ip route
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.7.7.2/32 [90/21024000] via 10.1.1.1, 00:14:05, Serial0
D        10.7.7.0/24 [90/21024000] via 10.1.1.1, 00:14:05, Serial0
C        10.1.1.0/24 is directly connected, Serial0
C        10.1.1.1/32 is directly connected, Serial0
C        192.168.0.0/24 is directly connected, Ethernet0
```

În cadrul tabelei de rutare, distanța administrativă este reprezentată de prima valoare numerică afișată între paranteze drepte. Acest număr va apartine intervalului 0-255, valoarea cea mai mică fiind preferată pentru alegerea rutei către destinație. Astfel, ruta cu distanță administrativă 0, asociată rutelor direct conectate, va fi tot timpul aleasă înaintea altor rute existente. Pe de altă parte, o rută cu o valoare AD („administrative distance”) egală cu 255 nu va fi instalată niciodată în tabela de rutare. După rutele direct conectate vor fi preferate rutele statice, deoarece acestea au atașată o distanță administrativă implicită egală cu 1, iar apoi protocoalele dinamice de rutare.

Valori AD standard

Sursa rutei	Distanța administrativă
Direct conectată	0
Statică	1
Rută summarizată EIGRP	5
Rută BGP externă	20
Rută EIGRP internă	90
IGRP	100
OSPF	110
ISIS	115
RIP	120
Rută EIGRP externă	170
Rută BGP internă	200

Valorile standard pot fi modificate... doar la CCNP ☺

Distanțele administrative pentru cele mai utilizate protocoale de rutare sunt precizate în tabelul alăturat. Se remarcă valorile pentru următoarele tipuri de rute:

- Direct conectate: 0
- Statice: 1
- RIP: 120
- EIGRP: 90
- OSPF: 110

În funcție de valorile AD-urilor se pot efectua următoarele concluzii: AD RIP > AD EIGRP deoarece EIGRP este mai performant decât RIP, AD IGRP > AD EIGRP deoarece EIGRP a fost dezvoltat ca o îmbunătățire a IGRP. Rutele externe vor avea o distanță administrativă mai mare datorită faptului că sunt provenite din alte domenii de rutare.

Rute direct conectate și statice

Atenție la rutele statice date prin interfața de ieșire!

- ele apar în tabela de rutare ca fiind „directly connected”
- dar au AD = 1

```
R1#show ip route

S    192.168.0.0/24 is directly connected, Serial0/0

R1#show ip route 192.168.0.0
Routing entry for 192.168.0.0/24
Known via "static", distance 1, metric 0 (connected)
  Routing Descriptor Blocks:
    * directly connected, via Serial0/0
      Route metric is 0, traffic share count is 1
```

Rutele statice sunt introduse manual de către administratorul de rețea cu scopul de a configura o cale optimă către destinație. Valoarea implicită a distanței administrative a rutelor statice este 1, deci, după rutele direct conectate care au distanță administrativă egală cu 0 sunt preferate rutele statice.

Rutele statice pot fi configurate utilizând adresa IP „next-hop” sau interfața de ieșire, în ambele cazuri având distanță administrativă implicită, și anume 1. Totuși, în cazul rutelor statice configurate folosind interfața de ieșire, valoarea distanței administrative nu este afișată la introducerea comenzi show ip route. Rutele direct conectate vor apărea în tabela de rutare imediat după configurarea adreselor IP pe interfețe și activarea acestora. Distanța administrativă fiind 0, va fi întotdeauna ruta preferată.

În cazul în care o rută statică configurată nu apare în tabela de rutare se va verifica dacă interfața de ieșire este configurată corect și activată.

Protocole de rutare

Distance Vector

Protocolele Distance Vector (1)

Ce implică ideea de Distance Vector ?

- rutele sunt reținute și propagate sub forma unui vector
- fiecare intrare reține rețeaua destinație, direcția către ea și distanța până la ea

Un ruter care folosește un protocol Distance Vector

- nu cunoaște toată calea pe care va fi transmis un pachet
- cunoaște următorul "hop"
- cunoaște distanța până la rețea

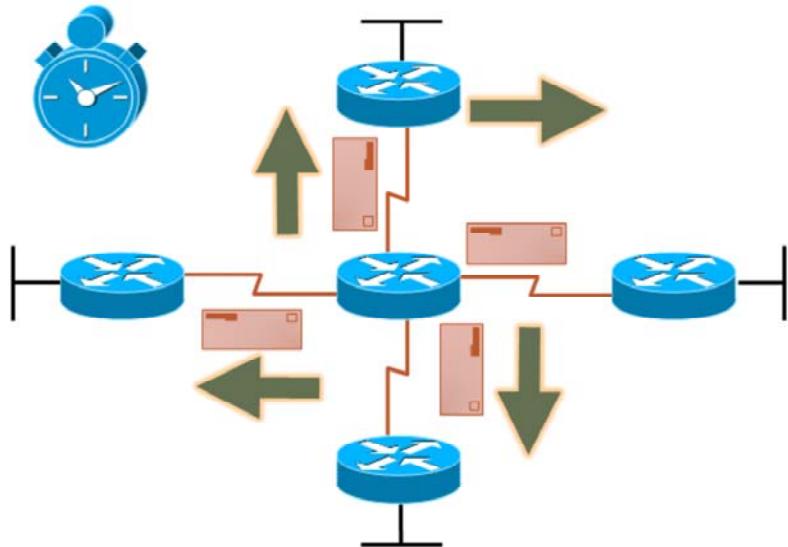
Un protocol distance vector își reține rutele în forma unui „vector de distanță și de direcție”. Distanța, în acest context, se definește cu ajutorul unei metriki.

Această metrică poate fi reprezentată, de exemplu, de numărul de hop-uri (protocolul RIP) sau se poate referi la bandwidth, delay, reliability, load, MTU (protocolul EIGRP). Direcția reprezintă următorul hop, sau interfața pe care ruterul trimite pachete către o anumită rețea.

Dacă un ruter folosește un protocol distance vector, acesta va cunoaște direcția pe care o va lua un pachet și distanța până la destinație. Totuși, ruterul nu va ști întreaga cale către rețeaua destinație.

Protocolele Distance Vector (2)

Efectuează update-uri periodice prin broadcast sau multicast



Protocolele distance vector prezintă un set de caracteristici comune:

Sunt trimise **update-uri periodice** către vecini, la intervale fixe, chiar dacă topologia rețelei nu s-a schimbat pe parcursul unui **anumit** interval de timp.

Vecinii unui ruter sunt acele rute care sunt direct conectate la acesta și care folosesc același protocol de rutare.

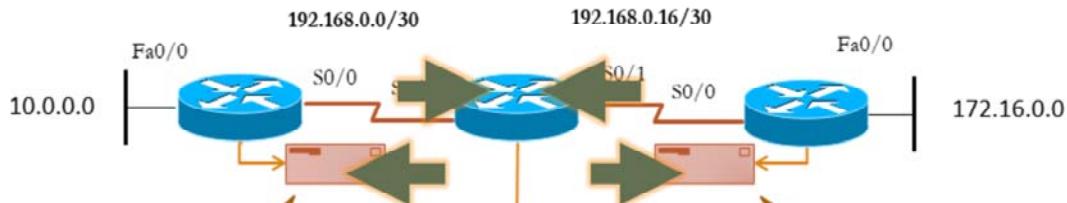
Update-urile sunt trimise prin **broadcast**. Ruterele vecine vor procesa aceste update-uri. Toate celelalte echipamente din rețea vor decapsula aceste pachete până la nivelul 3, după care le vor arunca.

Update-uri cu tabela de rutare vor fi trimise de protocolele distance vector, cu câteva excepții (EIGRP), către vecinii săi. Vecinii care primesc aceste update-uri vor procesa întregul pachet pentru a găsi informații pertinente și vor arunca restul datelor.

Protocolele Distance Vector (3)

Orice ruter își cunoaște doar vecinii direcți

Se trimite întreaga tabelă de rutare vecinilor



Rețea	Interfață	Număr de hopuri	Rețea	Interfață	Număr de hopuri	Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0	192.168.0.0	S0/0	0	172.16.0.0	Fa0/0	0
192.168.0.0	S0/0	0	192.168.0.16	S0/1	0	192.168.0.16	S0/0	0

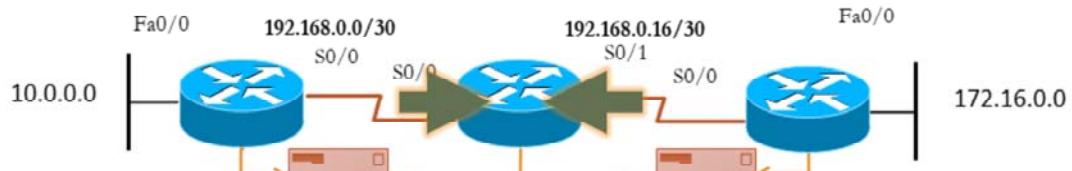
În acest exemplu, cele 3 rutere își trimit tabela de rutare numai către vecinii lor. Se observă că primul ruter trimit un pachet către al doilea ruter, dar nu și către al treilea ruter, în timp ce ruterul 2 trimit pachete ruterului 1 și 3.

Un protocol distance vector va folosi un algoritm specific pentru instalarea rutelor în tabela de rutare, pentru trimitera de update-uri vecinilor și pentru luarea deciziilor de rutare. Acesta va avea definite următoarele mecanisme:

- Mecanism pentru primirea și trimiterea informației de rutare
- Mecanism pentru calcularea celor mai eficiente rute și instalarea lor în tabela de rutare
- Mecanism pentru detectarea și adaptarea la schimbări în topologia rețelei

Protocolele Distance Vector (4)

Tabela de rutare după primul update



Retea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

Retea	Interfață	Număr de hopuri
192.168.0.0	S0/0	0
192.168.0.16	S0/1	0
10.0.0.0	S0/0	1
172.16.0.0	S0/1	1

Retea	Interfață	Număr de hopuri
172.16.0.0	Fa0/0	0
192.168.0.16	S0/0	0
192.168.0.0	S0/0	1

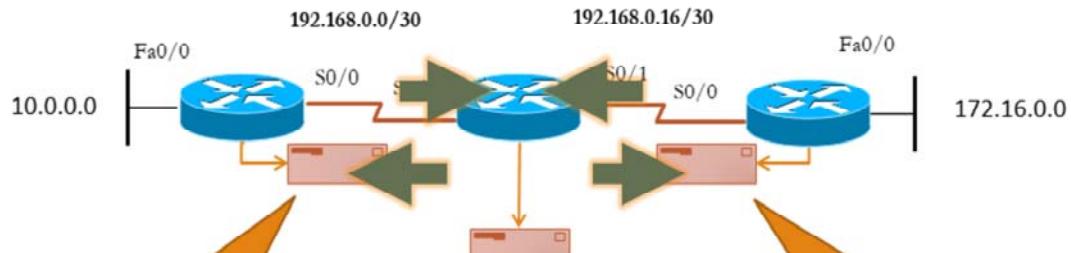
În cazul de față, cele trei rutere primesc informații despre cel puțin o rețea anterior necunoscută. Fiecare ruter va analiza independent update-ul primit, va calcula cea mai scurtă cale către noile adrese de rețea, iar apoi acestea vor fi adăugate sau actualizate în tabela de rutare.

Atunci când un ruter primește un update care conține întreaga tabelă de rutare a unui vecin, va păstra numai partea din pachet care îi aduce informații noi, iar pe cealaltă o va arunca. Astfel, ruterul 2 trimite către ruterul 1 ambele adrese afilate în tabela sa de rutare (192.168.0.0, 192.168.0.16), dar ruterul 1 instalează numai ruta pe care nu o cunoștea (192.168.0.16). La primirea unei rute deja existente în tabela de rutare, dar cu o metrică mai bună, vechea rută va fi actualizată conform noilor informații primite.

Protocolele Distance Vector (5)

Tabela de rutare după al doilea update

Putem observa că rețeaua a convergit



Rețea	Interfață	Număr de hopuri	Rețea	Interfață	Număr de hopuri	Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0	192.168.0.0	S0/0	0	172.16.0.0	Fa0/0	0
192.168.0.0	S0/0	0	192.168.0.16	S0/1	0	192.168.0.16	S0/0	0
192.168.0.16	S0/0	1	10.0.0.0	S0/0	1	192.168.0.0	S0/0	1
172.16.0.0	S0/0	2	172.16.0.0	S0/1	1	10.0.0.0	S0/0	2

După al treilea update, fiecare ruter din exemplu va avea informații complete despre topologie. Convergența este realizată atunci când fiecare ruter are o imagine completă și corectă asupra întregii topologii. Practic informațiile tuturor ruterelor despre rețele oferă posibilitatea existenței unei conexiuni cu orice destinație din cadrul domeniului de rutare.

Să presupunem că la un moment dat, din diferite motive, rețeaua 172.16.0.0 devine inaccesibilă. În această situație ruterul 3 va trimite un update provocat (triggered update) către vecinul său, ruterul 2, care va scoate rețeaua din tabela sa de rutare. La rândul său, al doilea ruter va trimite un update către ruterul 1, care va șterge și el adresa respectivă din tabela sa proprie de rutare.

Protocolele Distance Vector (6)

Avantaje

- configurare și întreținere simplă
- consumul de resurse al ruterului este redus

Dezavantaje

- timp de convergență ridicat
- scalabilitate limitată
- pot apărea bucle de rutare

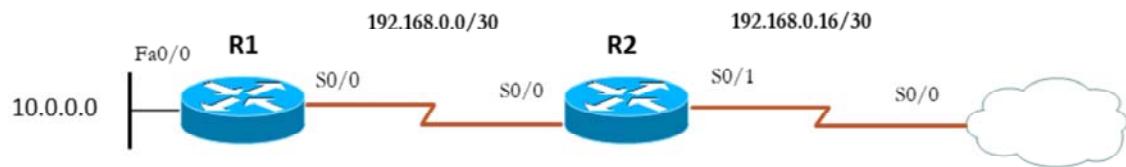
Datorită simplității configurației unui protocol de rutare distance vector, nivelul de cunoștințe al unui administrator necesar pentru implementarea și întreținerea ulterioară a protocolului de rutare distance vector într-o rețea este redus.

Complexitatea redusă a algoritmilor utilizați de protocolele de rutare distance vector, dar și informațiile nu foarte detaliate despre rutele schimbate între vecini nu necesită o cantitate mare de memorie și nici o putere de procesare sporită. În cazul folosirii unor echipamente cu resurse reduse, protocolele distance vector reprezintă alegerea ideală pentru dirijarea pachetelor în mod dinamic în rețea.

Timpul de convergență al protocolelor de rutare distance vector este ridicat datorită update-urilor de rutare trimise la un anumit interval de timp definit, ceea ce implică o scalabilitate redusă, un număr mare de echipamente crescând posibilitatea apariției bulelor de rutare.

Bucle de rutare (1)

Exemplu: Problema “count-to-infinity”



Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

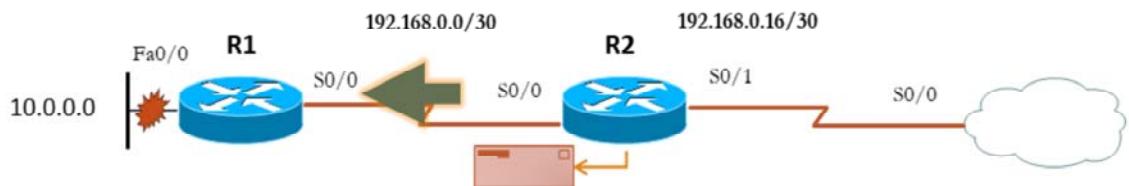
Rețea	Interfață	Număr de hopuri
192.168.0.0	S0/0	0
192.168.0.16	S0/1	0
10.0.0.0	S0/0	1

Problema „count to infinity” apare în momentul în care între rutere circulă update-uri eronate despre o rețea indisponibilă.

Să presupunem, în topologia de mai sus în care s-a realizat convergența, că rețeaua 10.0.0.0 devine indisponibilă. În același timp, R2 îi trimite un update lui R1 cu informația că are o rută către 10.0.0.0 cu metrica 1, cunoșcând faptul că are o rută în tabela de rutare către 10.0.0.0 prin ruterul vecin R1. În această situație, R1 introduce în tabela sa proprie de rutare o rută către 10.0.0.0 cu metrica 2. După un timp, R1 trimite update la R2 conținând toate rutele sale. R2 vede că ruta către 10.0.0.0 nu mai are metrica 1, ci 2, așa că va mări metrica la 3. Când R2 va trimite update la R1, acesta va vedea că metrica rutei către 10.0.0.0 a crescut, așa că va incrementa metrica în tabela sa de rutare. Acest proces se poate repeta la infinit, generând problema „count to infinity”.

Bucle de rutare (2)

Rețeaua 10.0.0.0 devine inaccesibilă, simultan R2 trimite un update lui R1



Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	s0/0	0
192.168.0.16	s0/0	1

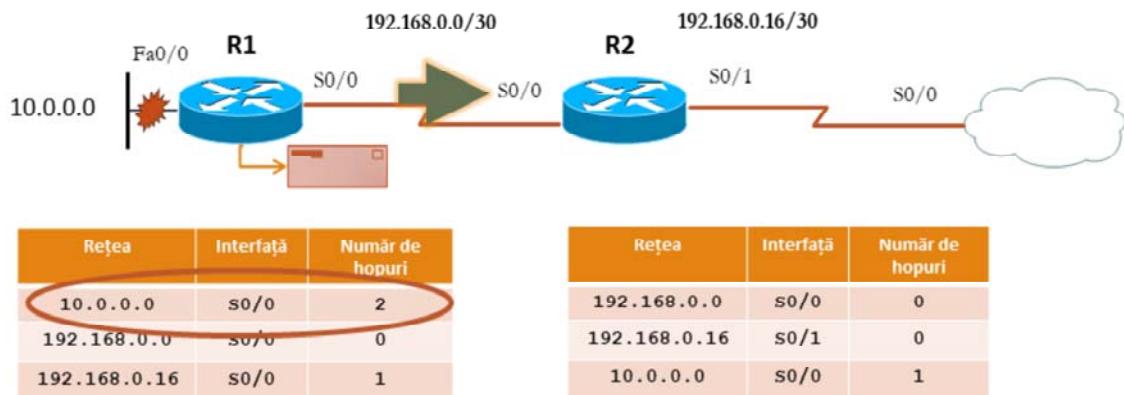
Rețea	Interfață	Număr de hopuri
192.168.0.0	s0/0	0
192.168.0.16	s0/1	0
10.0.0.0	s0/0	1

O buclă de rutare este o situație în care un pachet este transmis în mod continuu între o serie de rutere, fără a ajunge la destinația dorită. Acest lucru se întâmplă atunci când anumite rutere din rețea au informații incorecte despre o cale care apare ca fiind validă, către o destinație care, în mod real, nu există.

Un generator tipic pentru o astfel de buclă este exemplul de mai sus. După cum se poate observa, rețeaua 10.0.0.0 devine inaccesibilă, dar înainte ca R1 să trimită un mesaj cu această informație, primește un update de la R2 care are o rută către 10.0.0.0. Aceasta este o altă situație când apare problema „count to infinity”.

Bucle de rutare (3)

R1 reintroduce rețeaua în tabela de rutare și trimite și el update



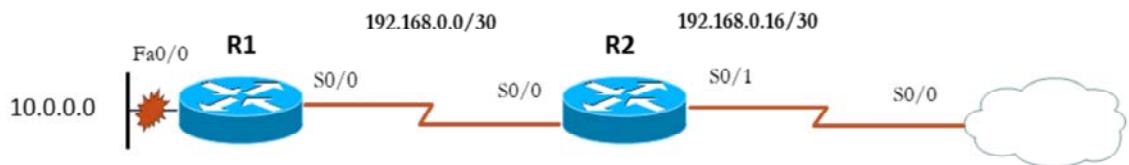
Deoarece ruterul nu cunoaște topologia rețelei, va presupune că informația primită de la R2 este corectă și va adăuga în tabela de rutare adresa rețelei 10.0.0.0, cu hop-ul 2. În acest moment, un pachet trimis de pe R1 cu destinația din rețeaua 10.0.0.0 va ajunge la R2. Mai departe, R2 cunoaște ca next-hop pentru rețeaua 10.0.0.0 ruterul R1. Astfel s-a format o buclă de rutare, pachetul circulând între ruterele R1 și R2 până va fi aruncat când valoarea TTL va ajunge 0.

Buclele de rutare pot apărea în diferite circumstanțe:

- Rute statice configurate incorrect
- Redistribuire de rute configurate incorrect (redistribuția este un proces prin care pot fi transmise informații introduse manual sau învățate prin intermediul altui protocol de rutare)
- Tabele de rutare inconsistentă, datorită unui timp de convergență crescut

Bucle de rutare (4)

Bucla se repetă la infinit



Rețea	Interfață	Număr de hopuri
10.0.0.0	s0/0	2
192.168.0.0	s0/0	0
192.168.0.16	s0/0	1

Rețea	Interfață	Număr de hopuri
192.168.0.0	s0/0	0
192.168.0.16	s0/1	0
10.0.0.0	s0/0	3

O buclă de rutare se poate dovedi extrem de dăunătoare pentru o rețea, ducând la performanțe scăzute sau chiar la blocarea rețelei. Printre efectele unei bucle de rutare se numără:

- Utilizarea excesivă a bandwidth-ului care poate determina congestiunea rețelei datorită fluxului de pachete care ciclează într-o buclă de rutare
- Procesorul ruter-ului va fi suprasolicitat datorită procesării numărului mare de operații într-un interval scăzut de timp
- Poate fi afectată convergența rețelei, ducând la rutare suboptimală sau la pierderi de pachete
- Update-urile periodice se pot pierde sau pot fi întârziate, generând astfel mai multe bucle de rutare

Bucle de rutare (5)

Apar datorită cunoașterii reduse a rețelei și a convergenței lente

Cum le prevenim ?

- setarea unei valori maxime a metricii unei rute
- hold-down timer (ruterele sunt instruite să ignore update-uri despre o anumită rută un interval de timp)

Protocolele distance vector sunt foarte simple în ceea ce priveste operațiile efectuate. Totuși, această simplicitate prezintă dezavantaje precum buclele de rutare. O buclă de rutare poate fi foarte dăunătoare unei rețele, ocupând excesiv bandwidh și resurse. Există o serie de mecanisme pentru prevenirea buclelor de rutare:

- Setarea unei valori maxime a metricii unei rute, prevenind astfel problema „count to infinity” (ex.: RIP poate avea metrică maxim 15)
- Definirea unui hold-down timer; în momentul când un ruter este informat că o rețea este inaccesibilă, acesta pornește un contor de timp pe durata căruia ruterul nu va accepta nici un update despre rețeaua respectivă

Bucle de rutare (6)

Cum le prevenim ?

- split-horizon (o rută nu este trimisă înapoi pe calea pe unde a fost învățată)
- poison reverse (se face un update cu o metrică “infinită” pentru rutele care devin inaccesibile)
- câmpul TTL din antetul IP (asigură că pachetele nu vor circula la infinit)

- „**Split horizon**” împiedică un ruter să trimită update despre o anumită rută pe aceeași interfață pe care a primit inițial informații despre existența ei
- „**Route poisoning**” sau „**poison reverse**” se referă la cazul în care se trimit explicit informația că o rețea este inaccesibilă (este trimis un update în care rețeaua respectivă are metrica maximă, astfel că ruterele care o primesc să o considere inaccesibilă)
- **TTL-ul din antetul IP** asigură faptul că un pachet va putea traversa un număr finit de hop-uri, după care va fi aruncat; la fiecare traversare a unui hop valoarea TTL-ului este decrementată, astfel că atunci când ajunge la valoarea 0, pachetele vor fi ignorate
- **Update-urile provocate (triggered updates)** sunt folosite de unele protocoale în cazul în care o rețea devine inaccesibilă, pentru a informa rapid celelalte rutere explicit despre acest eveniment

RIP

Protocol open standard

Metrica folosită: numărul de hop-uri

- metrica maximă pentru o rețea este de 15 – nu poate fi folosit în rețele de diametru mai mare de 15

Trimite update-uri la fiecare 30 de secunde (implicit)

Trimite și update-uri provocate de modificări ale topologiei (triggered updates)

Equal cost load balancing

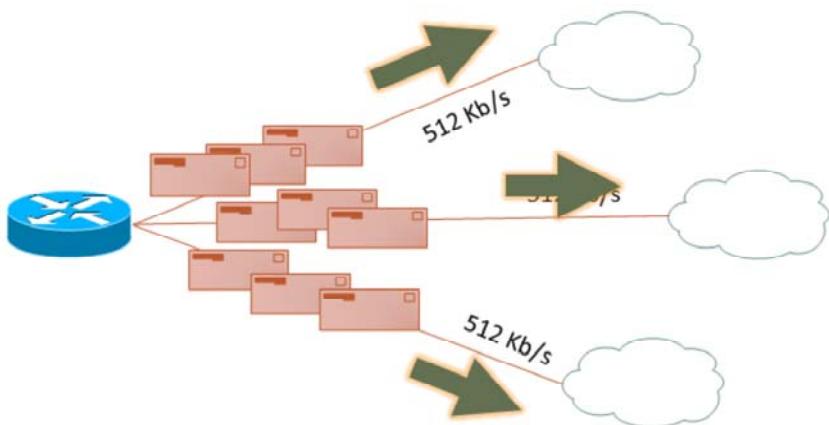
Protocolul RIP (Routing Information Protocol) este primul protocol dinamic folosit. Acesta a evoluat, în timp, de la un protocol classful (RIPv1) la un protocol classless (RIPv2). Deoarece este „open standard”, RIP poate fi implementat de către orice producător de rutere.

Metrica RIP se bazează numai pe numărul de hop-uri. Metrica maximă este 15. Aceast lucru înseamnă că nu poate fi folosit în rețele cu diametru (hop-count) mai mare de 15 hopuri. Totuși, este ușor de configurat și de întreținut, ceea ce îl face o alegere bună pentru rețele de dimensiuni mici.

RIP trimite update-uri periodice la intervale de 30 de secunde, dar poate trimite și update-uri provocate (triggered updates), dacă are loc o schimbare a topologiei. O altă funcționalitate a protocolului RIP este faptul că poate face „equal cost load balancing”.

Equal cost load-balancing

Având mai multe legături cu aceeași metrică se împarte traficul în mod egal pe ele



În situația în care un ruter cunoaște mai multe căi către o singură destinație, iar metrica lor are aceeași valoare (în funcție de protocol: același număr de hop-uri, aceeași lățime de bandă, etc.) spunem că avem o situație de „equal cost metric”.

În tabela de rutare, „equal cost load balancing” se traduce prin existența unei singure intrări cu adresa rețelei, care are asociate mai multe interfețe de ieșire sau adrese IP next-hop. Un dezavantaj în cazul RIP-ului care ia în considerare doar hop-count-ul este că, dacă există, spre exemplu, două rute cu același număr de hop-uri către aceeași destinație, faptul că una este de 2Mb iar cealaltă de 512Kb nu va reprezenta un criteriu de diferențiere a celor două rute. Ele vor fi tratate egal, RIP realizând procesul de „equal cost load balancing”.

RIPv1 (1)

Update timer – durata până la trimitera următorului update de rutare (implicit 30 secunde)

Invalid timer – durata până când o rută este marcată nevalidă (setarea metricii 16) dacă lipsește din update-urile vecinilor (implicit 180 secunde)

Hold-down timer – folosit pentru prevenirea buclelor de rutare (implicit 180 secunde)

Flush timer – durata până când rutele nevalide sunt scoase din tabela de rutare (implicit 240 secunde)

Protocolul RIPv1 prezintă o serie de caracteristici specifice protocoalelor distance vector:

- **Update-uri periodice, trimise la un interval de timp fix, setat la valoarea de 30 de secunde**
- **Invalid timer – o rută este setată ca fiind invalidă dacă timp de 180 secunde lipsește din update-urile vecinilor, fiind totuși păstrată în tabela de rutare până la expirarea flush timer-ului**
- **Hold-down timer – sunt opriate orice schimbări în tabela de rutare pentru un interval de timp (180 secunde), pentru a preveni instalarea incorectă a unei rute inaccesibile, semnalate printr-un update de rutare primit de la un vecin care poate să nu fi aflat despre modificarea survenită**
- **Flush timer – previne ștergerea unei rute pentru un timp fixat chiar dacă aceasta a devenit inaccesibilă (240 secunde)**

RIPv1 (2)

Aflarea timpului trecut de la ultimul update

```
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

R 172.150.0.0 [120/5] via 10.119.254.6, 0:01:00, Ethernet2
R 172.17.10.0 [120/8] via 10.119.254.244, 0:02:22, Ethernet2
R 172.70.132.0 [120/5] via 10.119.254.6, 0:00:59, Ethernet2
R 10.130.0.0 [120/3] via 10.119.254.6, 0:00:59, Ethernet2
```

Rutele care conțin caracterul „R” în fața adresei de rețea au fost adăugate la tabela de rutare prin protocolul RIP.

La apelarea comenzi show ip route se va afișa, în dreptul fiecărei adrese, timpul care a trecut de la ultimul update, în secunde. În caz că rutele au fost învățate prin RIP în acest fel se poate observa dacă există posibilitatea ca ruterul să efectueze load balancing. În acest caz se va afișa o rețea instalată cu mai multe interfețe de ieșire sau adrese IP next-hop.

Tot în tabela de rutare sunt afișate în dreptul fiecărei rute două valori numerice între paranteze drepte, separate prin caracterul „/”. Aceste valori reprezintă distanța administrativă împreună cu metrica, specifice protocolului de rutare implementat. În acest caz, se observă că RIP are AD egală cu 120, iar metrica, în funcție de fiecare rută, numărul de echipamente de nivel 3 prin care trece un pachet până la destinație.

RIPv1 (3)

Afișarea timerelor protocolului

```
Router#show ip protocols

Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 2 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240

<output omis>

  Routing for Networks:
    172.19.0.0
    2.0.0.0
    10.3.0.0
  Routing Information Sources:
    Gateway Distance Last Update
    Distance: (default is 120)
```

Comanda `show ip protocols` furnizează informații despre protocoalele de rutare active. Timpul trecut de la ultimul update apare sub eticheta „Last Update”. Pe lângă acesta, comanda va afișa când va fi trimis următorul update.

Se pot vizualiza, de asemenea, valorile pentru timer-ele invalid, hold-down și flush. Timer-ul invalid reprezintă timpul după care o rută este marcată invalidă dacă nu se mai primesc informații despre aceasta, timer-ul flush pornește când o rută devine invalidă și reprezintă timpul după care va fi ștersă din tabela de rutare dacă nu își revine, iar timer-ul hold-down este folosit pentru a evita buclele de rutare când o rută devine invalidă.

În output-ul comenzii se pot vedea și rețelele classful despre care ruterul respectiv va trimite update-uri.

RIPv2

Folosește Split Horizon și Poison Reverse

Funcționează classless

Are metode de autentificare

Suportă VLSM

Suportă sumarizarea manuală a rutelor

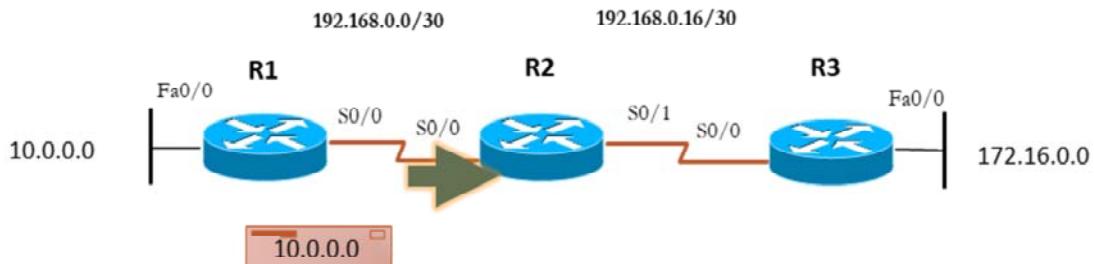
RIPv2 este un protocol de rutare classless, ceea ce înseamnă că include masca atașată unei adrese de rețea în update-urile trimise către celelalte rutere. VLSM reprezintă prescurtarea de la „Variable Length Subnet Mask”, adică există posibilitatea subnetării rețelelor classful, folosind diferite măști de rețea.

RIPv2 folosește mecanisme de autentificare pentru a securiza update-urile. Acest lucru are o importanță majoră când accesul în rețea nu poate fi strict controlat, existând potențiale încercări de interceptare și modificare a update-urilor.

RIPv2 preia de la RIPv1 tehniciile split horizon și poison reverse utilizate pentru prevenirea apariției buclelor de rutare. În loc de adrese broadcast, protocolul RIPv2 folosește adrese multicast pentru transmiterea update-urilor. De asemenea, spre deosebire de RIPv1, acest protocol suportă sumarizarea manuală a rutelor.

Split Horizon (1)

R1 propagă ruta 10.0.0.0 către R2

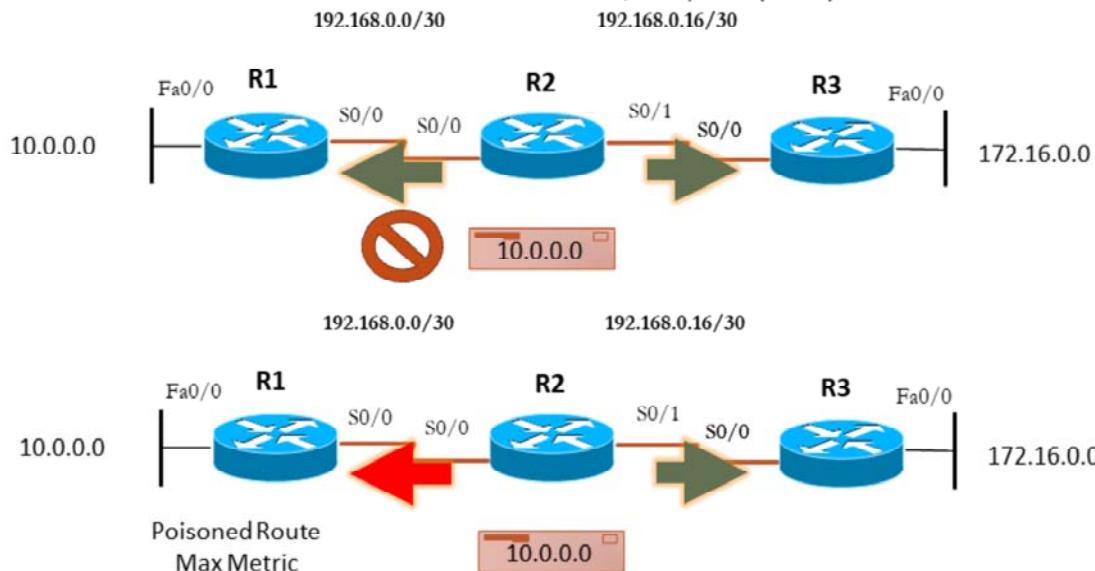


Split horizon este folosit pentru a preveni buclele de rutare cauzate de timpul de convergență crescut. Această regulă spune că un ruter nu poate transmite un update cu o anumită rețea pe aceeași interfață pe care a primit inițial informații despre adresa respectivă. Când un ruter află pentru prima oară despre o rută de la unul dintre vecini, se consideră că vecinul respectiv este mai aproape de destinație. În consecință, primul ruter nu îi va trimite update-uri vecinului conținând aceeași rută pentru a evita suprascrierea tabelei de rutare a acestuia cu informații neactualizate.

În exemplul de mai sus, R1 trimite lui R2 un update cu adresa 10.0.0.0. R2 va adăuga această adresă în tabela sa de rutare.

Split Horizon (2)

R2 transmite ruta 10.0.0.0 doar către R3, nu și înapoi spre R2



Respectând regula split horizon, ruta 10.0.0.0, pe care R2 a învațat-o de la R1, nu va fi inclusă în update-ul trimis care R1, dar va fi transmisă către R3.

O metodă alternativă de a preveni buclele de rutare este folosirea tehnicii de „route poisoning”. Aceasta implică faptul că se va transmite un update explicit despre rută care va fi marcată ca înaccesibilă, în loc ca aceasta să nu fie inclusă în viitoarele update-uri și să fie marcată ca invalidă la expirarea timer-ului invalid. Această metodă micșorează timpul de convergență. Update-urile vor conține o valoare a metricii egală cu 16, ceea ce indică în RIP o metrică infinită. Informația despre rețeaua înaccesibilă este astfel propagată în întreaga rețea de către rutele vecine, nemaifiind nevoie să se aștepte până la expirarea anumitor timere, procesul de convergență fiind accelerat semnificativ.

Interior Gateway Routing Protocol

Protocol proprietar Cisco

Protocol Distance Vector

A fost înlocuit de protocolul EIGRP

Folosea o formulă în funcție de bandwidth, reliability și delay pentru a calcula metrica

Update-urile de rutare erau trimise implicit la 90 de secunde

Protocolul IGRP (Interior Gateway Routing Protocol) este un protocol distance vector, care, spre deosebire de RIP, este proprietar Cisco, ceea ce înseamnă că va funcționa numai pe echipamente Cisco. Acest protocol nu mai este folosit în prezent, el fiind înlocuit de EIGRP.

Acest protocol a fost dezvoltat pentru a depăși neajunsurile protocolului RIP. Deoarece metrica RIP se poate dovedi ineficientă, IGRP suportă metriki multiple pe rute, cum ar fi: bandwidth, load, MTU, reliability și delay. Pentru a compara două rute, aceste metriki vor fi combinate într-o singură metrică cu ajutorul unei formule.

IGRP face update-uri periodice la un interval de 90 de secunde. Acestea sunt transmise prin broadcast.

EIGRP (1)

Protocol proprietar Cisco

Protocol Distance Vector avansat

Poate face load balancing pe rute de cost inegal ("unequal cost load-balancing")

Folosește algoritmul DUAL pentru a calcula calea cea mai scurtă de la sursă la destinație

Update-urile de rutare nu se trimit periodic, ci doar în cazul unor modificări în rețea ("triggered updates")

Enhanced IGRP (EIGRP) a fost dezvoltat pe baza protocolului IGRP. Acesta este un protocol distance vector classless, cu unele caracteristici similare protocoalelor link-state. Caracteristicile EIGRP includ:

- Nu are update-uri periodice, ci folosește update-uri provocate atunci când există schimbări în topologia rețelei
- Folosește o tabelă de topologie pentru a reține toate rutele primite de la vecini, nu numai pe cele mai eficiente
- Suportă VLSM și summarizare manuală; acestea permit crearea de topologii mari, structurate ierarhic
- Poate face load balancing pe rute de cost inegal ținând cont și de alte caracteristici în afară de hop-count
- Metrica sa este bazată pe bandwidth-ul și pe delay-ul unei rute

EIGRP (2)

Protocol proprietar Cisco

Protocol Distance Vector avansat

Poate face load balancing pe rute de cost inegal ("unequal cost load-balancing")

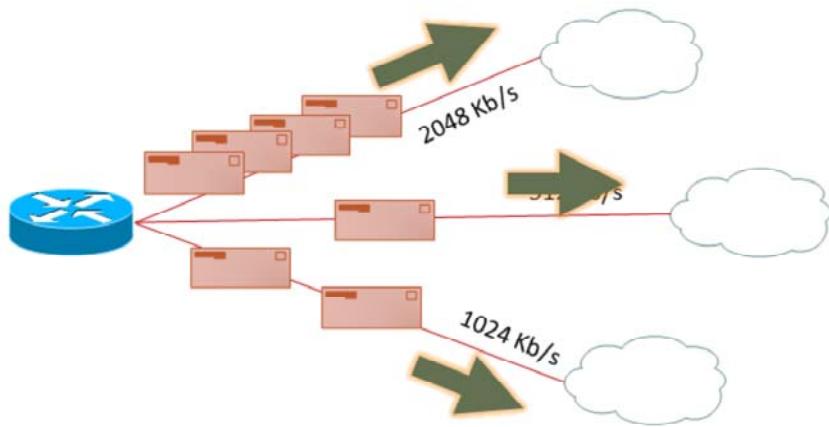
Folosește algoritmul DUAL pentru a calcula calea cea mai scurtă de la sursă la destinație

Update-urile de rutare nu se trimit periodic, ci doar în cazul unor modificări în rețea ("triggered updates")

- **Folosește algoritmul DUAL (Diffusion Update Algorithm) pentru a calcula cea mai scurtă rută până la o anumită destinație.** Acesta permite inserarea rutelor de back-up în tabela de topologie EIGRP, care sunt folosite dacă ruta primară devine inaccesibilă. Astfel, trecerea la ruta de back-up în cazul unei probleme este aproape imediată și nu presupune nici o acțiune din partea altor rutere. În cazul inexistenței rutei de back-up, algoritmul DUAL este reinicțiat pentru descoperirea unor posibile rute alternative.
- **EIGRP suportă diferite protocoale de nivel 3, cum ar fi IP, IPX sau AppleTalk.** Astfel, indiferent peste ce protocol rulează EIGRP-ul sau ce fel de rute transportă, el va aplica același algoritm.

Unequal cost load-balancing

EIGRP poate distribui traficul pe legături în funcție de metrica lor



Protocolul EIGRP va distribui traficul pe mai multe legături, în funcție de metrica care la rândul ei depinde de delay și bandwidth. Faptul că ține cont de bandwidth și că știe să facă unequal load-balancing este un avantaj major în comparație cu RIP.

EIGRP folosește bounded updates, adică numai ruterele care au nevoie de o anumită informație primesc pachetele de update, minimizând congestiunea legăturilor. O asemănare cu protocolele de rutare link-state este faptul că EIGRP transmite informații numai atunci când există o schimbare în topologia rețelei inclusiv informații numai despre modificările care au avut loc.

RIPv1

Caracteristici RIPv1

Protocol Distance Vector

Numărul de hop-uri(Hop Count) reprezintă metrica

- o metrică mai mare strict ca 15 face ca ruta să fie unreachable

Mesajele de notificare

- sunt transmise între vecini sub formă de broadcast
- sunt trimise la fiecare 30 de secunde
- sunt transmise peste UDP, port 520

Nu se mai folosește în real life!

RIPv1 este primul protocol de rutare dinamică. Este un protocol distance vector, având următoarele caracteristici:

- Utilizează numărul de hopuri ca metrică pentru alegerea rutei optime. Practic, rutele sunt evaluate doar în funcție de numărul de rutere (hop-uri) prin care trece un pachet, până la destinație
- Metrica maximă suportată de RIPv1 este 15, valoarea 16 fiind folosită pentru a desemna o metrică „infinită”, sau o rută inaccesibilă
- RIPv1 trimit mesaje de notificare, sub formă de broadcast, la un interval definit la 30 de secunde. Mesajul RIP este încapsulat într-un segment UDP, transmis folosind portul 520
- Datorită limitărilor întâlnite în cadrul RIPv1, în anul 1994 a fost dezvoltată o nouă versiune mai performantă, RIPv2.

Formatul mesajelor RIPv1

Command	1=Request 2=Reply
Version	1=RIPv1 2=RIPv2
Address Family Identifier	2=IP 0=Request pentru toată tabela de rutare
IP Address	Adresa destinație (de rețea, subnet sau host)
Metric	Hop-count, între 1 și 16

Câmpurile mesajelor de update folosite de protocolul RIPv1 se împart în două secțiuni:

- „RIP header” care include câmpurile command, în care se specifică tipul mesajului („Request” sau „Reply”), version care poate să aibă valoarea 1 pentru RIPv1, respectiv valoarea 2 pentru RIPv2 și câmpul must be zero, câmp ce oferă spațiu pentru o dezvoltare ulterioară a protocolului
- „Route Entry” este descris de câmpurile: “Address Family Identifier”, care poate conține două valori, 2 pentru IP și 0 dacă ruterul solicită întreaga tabelă de rutare; IP Address, în care este specificată adresa destinație; și câmpul metric, în care este menționată metrica specifică protocolului RIPv1, și anume numărul de hop-uri

Un update RIP poate conține maxim 25 de intrări, dimensiunea maximă a cadrului fiind 504 bytes, fără a include antetele IP sau UDP.

Procesul de cerere/răspuns

La pornire, ruterul configurat cu RIPv1 trimite o cerere



Vecinii trimit tabelelor lor de rutare ca răspuns



Se evaluatează răspunsurile

- dacă ruta nu există în tabela de rutare, este adăugată
- dacă ruta există, dar are metrică mai mare decât cea primită, este modificată

Fiecare interfață configurată cu protocolul RIPv1 trimite, la activarea protocolului, o cerere prin care solicită tuturor vecinilor să trimită tabela lor de rutare; doar cei care rulează un proces RIPv1 vor răspunde cererii. În momentul în care un ruter primește un răspuns, evaluatează fiecare rută, astfel:

- Dacă ruta nu se află în tabela de rutare, ea va fi adăugată
- Dacă ruta există, ea va fi înlocuită cu ruta nou învățată, doar în cazul în care aceasta din urmă are o metrică mai mică

Următorul pas este ca ruterul să trimită un update declanșat de evenimente (triggered update), conținând tabela proprie de rutare, pe toate interfețele sale pentru a-și informa vecinii despre noile schimbări. În urma acestui proces, se urmărește atingerea stării de convergență, prin trimiterea de update-uri, atât periodice, cât și declanșate de schimbări în topologie.

Rutare classless/classful

RIPv1 nu trimite masca de rețea în update

- se folosește de masca pusă pe interfețe
- se folosește de clase

Clasa A

255	0	0	0
Network	Host	Host	Host

Clasa B

255	255	0	0
Network	Network	Host	Host

Clasa C

255	255	255	0
Network	Network	Network	Host

Spațiul IPv4 este împărțit în trei clase principale: A, B și C. Fiecare clasă are atribuită o mască de rețea implicită: clasa A are masca /8, clasa B, /16, iar clasa C, /24.

RIPv1 este un protocol de rutare classful. Acest lucru înseamnă că retelele direct conectate vor fi anunțate în pachetele de update fără masca lor aferentă. Așadar, un ruter va instala ruta primită cu masca de rețea configurată pe interfața locală, doar dacă rețeaua anunțată face parte din aceeași clasă majoră cu IP-ul configurat pe interfața respectivă. Altfel, se va salva cu masca de rețea specifică clasei din care face parte IP-ul rutei. Datorită acestei limitări, adresarea rețelelor configurate folosind RIPv1 nu poate fi discontinuă și nici nu poate suporta măști de rețea de lungime variabilă (VLSM).

Distanța administrativă

RIP are AD=120

- mai puțin preferat față de IS-IS, IGRP, OSPF, EIGRP

```
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

R 172.150.0.0 [120/5] via 10.119.254.6, 0:01:00, Ethernet2
R 172.17.10.0 [120/8] via 10.119.254.244, 0:02:22, Ethernet2
R 172.70.132.0 [120/5] via 10.119.254.6, 0:00:59, Ethernet2
R 10.130.0.0 [120/3] via 10.119.254.6, 0:00:59, Ethernet2
```

Distanța administrativă reprezintă o valoare cuprinsă între 0 și 255 care desemnează un grad de „încredere” sau preferință pentru anumite rute, în detrimentul altora. O rută cu o distanță administrativă mai mică va fi întotdeauna preferată de ruter și instalată în tabela de rutare.

RIP are distanța administrativă 120. Compartat cu alte protocoale interne de rutare, RIP este cel mai puțin preferat protocol, în special datorită limitărilor în privința performanței și a scalabilității. Ulterior au fost dezvoltate și alte protocoale mai complexe cum sunt IS-IS, OSPF, IGRP și EIGRP. Datorită tehnologiilor implementate, acestea au o distanță administrativă mai mică decât RIP.

Interfețe pasive

Interfețe configurate în RIP, pe care nu se transmit update-uri

Avantaj față de scoaterea definitivă a adresei interfeței din protocol

- adresa interfeței este propagată în continuare de ruter

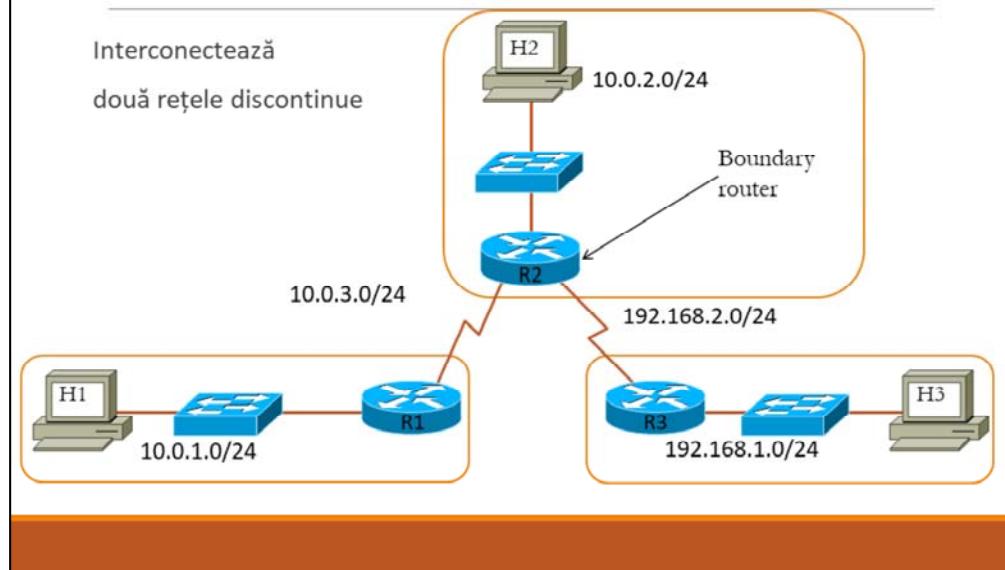
Folosite pentru a nu consuma bandwidth și timp de procesare cu update-uri nedorite

Update-urile nedorite pot crea găuri în securitate

Există anumite situații în care o rețea locală trebuie anunțată în protocolul de rutare, dar nu este necesară trimiterea update-urilor pe interfața de legătură deoarece nu există rutere care să ruleze RIP în acea rețea. În general, este de dorit evitarea traficului inutil în aceste segmente atât pentru că poate reprezenta un risc de securitate, putând fi interceptat și modificat. Un alt dezavantaj îl reprezintă faptul că broadcast-urile de nivel 3 pot fi procesate până la nivelul transport.

O soluție intuitivă de a opri transmiterea de update-uri într-o rețea este eliminarea rețelei din protocol, folosind comanda `no network [rețea]`. În acest caz, ruterul nu va mai trimite pachete de rutare către rețeaua respectivă, dar nici nu o va mai putea include în update-urile trimise către celelalte rutere. Soluția preferată este folosirea comenzi `passive interface [nume interfață]`, care previne trimiterea update-urilor de rutare pe interfață specificată, dar va permite ca acea rețea să fie anunțată în continuare către vecini.

Boundary router



Fiind un protocol de rutare classful, RIPv1 va summariza automat rețelele conform adresării classful. În figură, se observă că R2 este conectat la mai multe rețele classful, din această cauză ruterul R2 este considerat un ruter de tip „boundary router”.

Deoarece „boundary router”-ul R2 va summariza subrețelele RIP la o clasă majoră, update-urile pentru rețelele 10.0.1.0/24, 10.0.2.0/24 și 10.0.3.0/24 vor anunța rețeaua classful 10.0.0.0/8 pe interfața serială către R3. Așadar, ruterul R3 va instala o singură rută în tabela de rutare, summarizând toate cele 3 subrețele.

Reguli de procesare a update-urilor (1)

Rețeaua destinație din update aparține aceleiași clase majore cu interfața pe care a venit update-ul

- masca interfeței pe care s-a primit update-ul va coincide cu masca rutei în tabela de rutare

Rețeaua destinație din update nu aparține aceleiași clase majore cu interfața pe care a venit update-ul

- ruta adăugată în tabela de rutare este sumarizată automat la clasa majoră din care face parte

Deoarece este protocol de rutare classful, RIPv1 nu va include masca de rețea a rutelor anunțate în update-uri. Întrucât rețelele anunțate trebuie să fie totuși instalate în tabela de rutare cu o mască de rețea atașată, update-urile RIPv1 sunt procesate în funcție de următoarele două reguli majore:

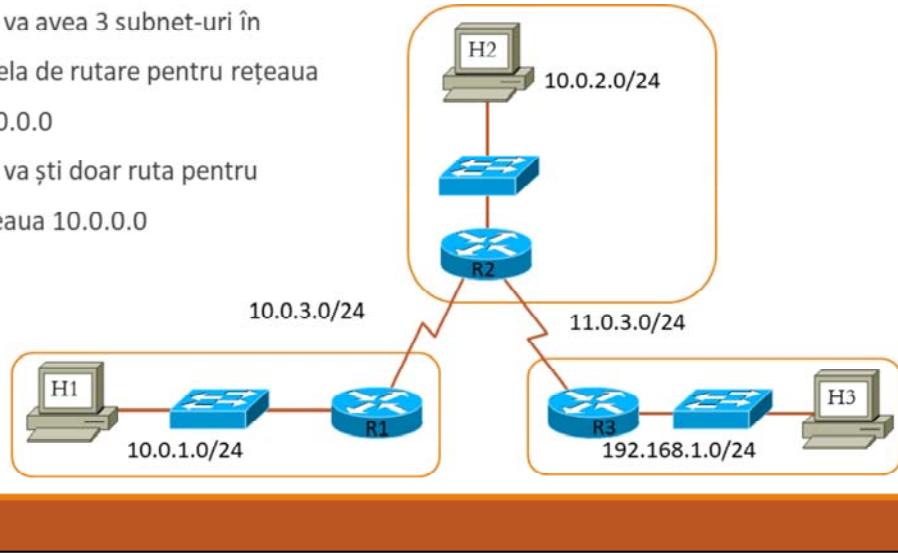
- Dacă un update de rutare conține o rută ce aparține aceleiași clase majore cu rețeaua interfeței pe care este primit acesta, masca de rețea a interfeței este aplicată rutei nou învățate
- Dacă un update de rutare nu aparține aceleiași clase majore cu interfața pe care este primit, ruta va fi adăugată în tabela de rutare cu masca de rețea classful din care face parte

Conform acestor reguli, un dezavantaj al RIPv1 este faptul că toate subrețelele unei clase majore vor utiliza masca de rețea a clasei din care fac parte.

Reguli de procesare a update-urilor (2)

R1 va avea 3 subnet-uri în
tabela de rutare pentru rețeaua
10.0.0.0

R3 va ști doar ruta pentru
rețeaua 10.0.0.0



În topologia din figură se poate observa aplicarea regulilor de procesare a update-urilor astfel: R1 va primi de la R2 un update cu rețeaua 10.0.2.0, care face parte din aceeași clasă majoră ca și rețeaua de pe interfață către R2 (10.0.0.0/8), astfel va introduce în tabela proprie de rutare rețeaua 10.0.2.0, cu masca de pe interfață locală, și anume /24.

În cazul ruterului R3, acesta va primi rețelele 10.0.1.0, 10.0.2.0, 10.0.3.0, care aparțin aceleiași clase majore cu interfața pe care a primit update-ul. Astfel, în tabela sa de rutare este instalată o singură rută classful, care sumarizează cele trei rețele, 10.0.0.0/8.

Avantajele summarizării automate

La limita dintre două clase diferite sunt trimise update-uri summarizate, care consumă puțină lățime de bandă

Tabela de rutare este mică, aşadar căutările sunt rapide

Pentru exemplul anterior, tabela ruterului R3 arată astfel :

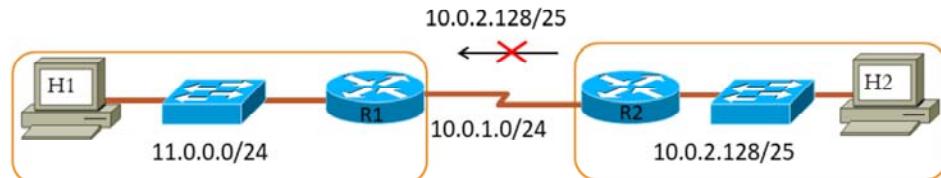
```
R3#show ip route
R 10.0.0.0/8 [120/1] via 11.0.3.2, 0:01:00, Serial0/0
  11.0.3.0/24 is subnetted, 1 subnets
    C 11.0.3.3 is directly connected, Serial0/0
    C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

Sumarizarea automată prezintă mai multe avantaje: update-urile de rutare trimise și primite sunt de dimensiuni mai mici, utilizând astfel o lățime mai mică de bandă; de asemenea timpul de procesare al update-urilor se diminuează, iar consumul de resurse de memorie și CPU se reduc.

După cum se observă în output-ul comenzi show ip route, pentru rețeaua 10.0.0.0/8, indiferent de numărul de subrețele se va trimite doar o rută summarizată rezultând un proces de rutare mai rapid.

Dezavantajele summarizării automate (1)

Lipsa suportului pentru VLSM și pierderea conectivității



```
R1#show ip route
      10.0.0.0/24 is subnetted, 1 subnets
c        10.0.1.0 is directly connected, FastEthernet0/0
      11.0.0.0/24 is subnetted, 1 subnets
c        11.0.0.0 is directly connected, Loopback0
```

În topologia din figură se remarcă un dezavantaj principal al protocoalelor de rutare classful, cum este RIPv1, și anume lipsa suportului pentru VLSM (Variable Length Subnet Mask).

Protocolele de rutare classful nu includ masca de rețea în mesajele de update RIPv1. Astfel, ruterul R2 nu trimite mai departe rețeaua 10.0.2.128/25. Prin urmare, ruterul R1 nu va avea conectivitate către rețeaua 10.0.2.128/25. Pentru rezolvarea acestei probleme, se recomandă utilizarea unei adresări classful sau folosirea unui protocol de rutare classless cum sunt: RIPv2, OSPF, EIGRP.

Dezavantajele summarizării automate (2)

Lipsa suportului pentru rețelele discontinue
R2 va avea rută pentru 10.0.0.0/8 și prin R1, și prin R3 – se pierd pachete



Un alt dezavantaj major al summarizării automate este reprezentat de lipsa suportului pentru rețelele discontinue. În topologia din figură, RIPv1 nu va putea identifica toate rețelele existente datorită discontinuității acestora. R1 și R3 au rolul de „boundary router” pentru subrețelele atașate din clasa majoră 10.0.0.0/8, acestea fiind separate de o altă rețea majoră, 11.0.0.0/8. Din cauza faptului că RIPv1 nu include în update-uri masca de rețea a rutei anunțate, nu va trimite corect informația de rutare către rețelele 10.0.1.0/24, respectiv 10.0.2.0/24. În acest caz, apar următoarele probleme în topologie:

- R1 nu are nicio rută către LAN-ul atașat lui R3
- R3 nu are nicio rută către LAN-ul atașat lui R1
- R2 va face load balancing între subrețelele din clasa majoră 10.0.0.0/8; acest lucru înseamnă că R1 și R3 vor primi, fiecare, jumătate din pachetele trimise de R2

Configurare RIP

Activare RIP

- (config) #**router rip**

Modul de configurare al protocolului

- R1 (config-router) #

Dezactivare RIP

- (config) #**no router rip**

Pentru a activa un protocol de rutare dinamic, în modul global de configurare se utilizează comanda `router`, urmată de numele protocolului care se doresă să fie configurat. Se observă că intrarea în modul de configurare al unui protocol de rutare este reflectată și în cadrul prompt-ului afișat: `config-router`.

Comanda nu pornește efectiv procesul RIP, ci doar creează contextul pentru comenzi de configurare ale acestuia. Dezactivarea unui protocol de rutare se face prin atașarea cuvântului `no` înainte de comanda `router` și numele protocolului dorit. Dezactivarea are ca efect și eliminarea configurațiilor protocolului de rutare.

Comanda network

Activează RIP pe toate interfețele specificate

- trimite update-uri pe interfața respectivă
- ascultă update-uri de pe interfața respectivă

Anunță rețelele configurate în update-urile trimise vecinilor

Comanda este prin definiție classful

Sintaxa

- (config-router) #**network [adresă-de-rețea-classful-direct-conectată]**

Înainte ca protocolul să poată efectiv funcționa, ruterul trebuie să știe ce interfețe vor fi folosite pentru a comunica cu ruterele vecine și ce rețele vor fi anunțate în protocolul de rutare. Comanda **network** specifică aceste informații prin introducerea adresei classful a uneia sau mai multor rețele direct conectate. Fiind o comandă cu un comportament classful, în caz că administratorul specifică un subnet al unei rețele classful ca parametru al comenzi **network**, sistemul de operare va converti respectiva rețea la clasa majoră din care face parte.

Comanda **network** activează RIP pe toate interfețele care se încadrează în adresa de rețea specificată ca parametru, și adaugă rețelele din care acestea fac parte în update-urile RIP.

Specificarea rețelelor în RIP

Deși se pot specifica adrese de host cu comanda network, IOS-ul corectează inputul

```
R(config)#router rip
R(config-router)#network 172.16.0.1
R(config-router)#network 12.0.0.2
R#show running-config
!
router rip
  network 12.0.0.0
  network 172.16.0.0
!
```

Introducerea unei adrese de subrețea va avea ca efect trunchierea adresei la lungimea prefixului clasei respective. Cu alte cuvinte, comanda network 172.16.0.1 introdusă pentru o interfață din rețeaua 172.16.0.0/16 va lua în considerare rețeaua 172.16.0.0, cu masca sa classful (/24). Prin definiție comanda network este o comandă classful. De exemplu, în caz că ruterul este conectat la mai multe subrețele din clasa majoră 10.0.0.0/8, este îndeajuns introducerea comenzii network 10.0.0.0, nefiind necesară rularea comenzii pentru fiecare subrețea.

Verificarea configurării

Verificarea că protocolul este funcțional pe o interfață

```
# show ip interface brief
```

Asigurarea că rutele propagate prin RIP au ajuns în tabela de rutare

```
# show ip route
```

Verificarea parametrilor protocolului

```
# show ip protocols
```

Diagnosticarea protocolului

```
# debug ip rip
```

Înainte de configurarea oricărui protocol de rutare, trebuie să existe o configurație IP a cel puțin unei interfețe active. Informații sumare despre fiecare interfață a unui ruter pot fi obținute prin folosirea comenzi show ip interface brief.

Rutele propagate prin RIP pot fi vizualizate în tabela de rutare prin comanda `show ip route`. Pentru vizualizarea informațiilor generale despre toate protocolele de rutare care rulează la un moment dat, există comanda `show ip protocols`. Comanda afișează interfețele active din fiecare protocol de rutare, comenziile network introduse și vecinii cu care se realizează schimbul de informații.

Pentru o examinare mai amănunțită a unei probleme de conectivitate apărute într-o rețea configurată cu RIP, se folosește comanda `debug ip rip` care afișează un output alcătuit din toate procesele generate de protocolul RIP.

Comanda show ip protocols (1)

1. Protocolele de rutare active

```
R2#show ip protocols  
Routing Protocol is "rip"  
***output omitted***
```

2. Timerele folosite și valorile lor

```
R2#show ip protocols  
***output omitted***  
Sending updates every 30 seconds, next due in 24 seconds  
Invalid after 180 seconds, hold down 180, flushed after 240
```

Comanda show ip protocols este utilă pentru vizualizarea detaliilor legate de protocolele de rutare implementate.

Astfel, prima informație afișată în output reprezintă denumirea protocolului configurat și activat la momentul respectiv. Pentru pornirea protocolului RIP este necesară cel puțin o interfață activă a cărei adresă IP să aparțină clasei rețelei specificate de comanda network.

Update-urile generate de protocolul de rutare configurat, în cazul de față, RIPv1, vor fi trimise vecinilor conform timerelor specificate – o dată la 30 de secunde. De asemenea, sunt precizate și valorile celorlalte timere existente:

- Invalid timer – 180 secunde
- Hold down timer – 180 secunde
- Flush timer – 240 secunde

Comanda show ip protocols (2)

3. Filtrare de update-uri și redistribuire numai cu RIP

```
R2#show ip protocols
***output omitted***
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
***output omitted***
```

4. Interfețele pe care RIP este activat și versiunea de RIP acceptată

```
R2#show ip protocols
***output omitted***
Default version control: send version 1, receive any version
Interface          Send Recv Triggered RIP Key-chain
Ethernet0/0           1      1 2
Serial1/0              1      1 2
Loopback0              1      1 2
Loopback1              1      1 2
***output omitted***
```

Protocolul de rutare RIP oferă posibilitatea filtrării anumitor update-uri în funcție de o serie de criterii bine stabilite, dar și posibilitatea redistribuirii anumitor rute în cadrul domeniului de rutare. Un scenariu des întâlnit este acela în care creăm pe un „boundary-router” o rută default către exterior și o redistribuim în tot protocolul de rutare pentru ca astfel pachetele cu destinație necunoscută să fie dirijate către acesta.

În continuare, în output-ul comenzi show ip protocols sunt incluse informații legate de versiunea protocolului, dar și interfețele care participă la procesul de trimitere și primire a update-urilor.

Comanda show ip protocols (3)

5. Sumarizare automată, la clasa adresei și equal cost load-balancing

```
R2#show ip protocols
***output omitted***

Automatic network summarization is in effect
  Maximum path: 4
***output omitted***
```

6. Rețelele classful configurate cu RIP

```
R2#show ip protocols
***output omitted***
Routing for Networks:
  10.0.0.0
  12.0.0.0
  172.16.0.0
***output omitted***
```

În output este precizat faptul că sumarizarea automată este activată în cadrul protocolului de rutare prin mesajul: „Automatic network summarization is in effect”. Astfel, dacă RIPv1 identifică mai multe subnet-uri aparținând aceleiași rețele majore și care utilizează aceeași cale de ieșire, va reduce rutele individuale la o singură rută classful.

În caz că protocolul RIPv1 va instala în tabela de rutare mai multe rute către aceeași destinație având aceeași metrică, numărul maxim de căi pe care RIPv1 poate realiza „equal cost load balancing” este egal cu 4.

Rețelele classful configurate cu ajutorul comenzi `network` sunt afișate în continuare în output. Aceste rețele vor fi incluse în update-urile RIPv1 și vor fi trimise mai departe vecinilor din cadrul domeniului de rutare.

Comanda show ip protocols (4)

7. Adresele vecinilor cu care comunică prin update-uri
 - include AD vecin, când s-a primit ultimul update de la vecin
 - ultima linie afișează AD-ul ruterului

```
R2#show ip protocols
***output omitted***

Routing Information Sources:
  Gateway          Distance      Last Update
    10.0.0.1           120          00:00:11
    12.0.0.1           120          00:00:24
  Distance: (default is 120)
***output omitted***
```

Vecinii RIP sunt afișați sub forma unui tabel în care sunt incluse următoarele detalii:

- **Gateway-ul** – adresa IP a vecinului care trimite update-uri
- **Distance** – distanță administrativă folosită pentru update-urile trimise de vecin
- **Last Update** – conține secundele scurte de la ultimul update primit de la vecin

Interfețe pasive

Configurare pentru a nu trimite update-uri pe interfață

```
R1(config)#router rip  
R1(config-router)#passive-interface fastEthernet 0/0
```

Interfața Fa0/0 face parte din rețeaua 10.0.0.0 pentru care RIP încă rutează

```
R1#show ip protocols  
Interface          Send   Recv   Triggered RIP  Key-chain  
    Serial1/0           1      1 2  
Automatic network summarization is in effect  
Maximum path: 4  
Routing for Networks:  
  10.0.0.0  
  12.0.0.0  
Passive Interface(s):  
  FastEthernet0/0
```

Unele interfețe pot fi legate către LAN-uri în care nu se află nici un dispozitiv care să ruleze RIP, deci trimiterea update-urilor nu mai este necesară. O rețea locală poate fi afectată prin faptul că update-urile trimise broadcast vor consuma o parte din bandwidth, ca apoi să fie procesate de dispozitive până la nivel 4 înainte de a fi aruncate.

De asemenea, trimitera update-urilor RIPv1 broadcast într-o rețea locală poate reprezenta un risc de securitate întrucât acestea pot fi interceptate de programe malicioase (sniffing software), modificate și apoi trimise mai departe determinând răspândirea unor date corupte.

Modalitatea corectă de a opri trimitera de update-uri RIP într-un LAN, este folosirea comenzi **passive-interface**, rețeaua locală fiind în continuare anunțată pe celelalte interfețe din domeniul de rutare.

Vizualizarea interfețelor pasive configurate se face prin comanda **show ip protocols**, fiind afișate în secțiunea „**Passive Interface(s)**”.

Propagarea unei rute default în RIP₍₁₎

Ruterul pe care este pusă ruta default trebuie configurat cu comanda

- (config-router) #**default-information originate**

```
R1(config)#ip route 0.0.0.0 0.0.0.0 Serial 0/0
R1(config)#router rip
R1(config-router)#default-information originate

R1#debug ip rip
*Mar 1 00:34:43.343: RIP: sending v1 update to 255.255.255.255
via FastEthernet0/0 (10.0.1.1)

*Mar 1 00:34:37.151: RIP: build update entries
*Mar 1 00:34:37.151:   subnet 0.0.0.0 metric 1
*Mar 1 00:34:37.155:   network 10.0.0.0 metric 1
```

Ruterele pot fi configurate cu o rută default pentru a fi folosită ca destinație pentru pachetele care nu pot fi trimise pe o rută mai specifică. Astfel, pachetul va fi trimis către next hop-ul sau interfața de ieșire indicată de respectiva rută. Există situații când o rețea care rulează RIP să aibă configurată o singură conexiune cu exteriorul, de exemplu spre Internet. Pentru ca toate ruterele să poată trimită pachete spre această conexiune, ele ar trebui configurate individual cu câte o rută statică default. RIP, ca și alte protocoale de rutare, oferă posibilitatea propagării unei astfel de rute în întreg domeniul de rutare.

O rută default este propagată într-un domeniu de rutare RIP cu ajutorul comenzi **default-information originate** dacă în tabela de rutare a ruterului pe care este rulată comanda este definită static o astfel de rută. La adăugarea parametrului **always** o rută default va fi propagată independent de existența unei rute statice default.

Propagarea unei rute default în RIP₍₂₎

Celelalte rutere vor avea următoarea intrare în tabela de rutare în urma procesării update-urilor

```
R2#show ip route

Gateway of last resort is 10.0.1.1 to network 0.0.0.0

    10.0.0.0/24 is subnetted, 2 subnets
C       10.0.2.0 is directly connected, Loopback0
C       10.0.1.0 is directly connected, Ethernet0/0
R       11.0.0.0/8 [120/1] via 10.0.1.1, 00:00:12, Ethernet0/0
R*      0.0.0.0/0 [120/1] via 10.0.1.1, 00:00:12, Ethernet0/0
```

Așadar, o rută default va fi instalată pe toate ruterele din domeniul de rutare RIP, putând fi vizualizată în tabela de rutare a acestora. Aceasta este identificată prin caracterul R deoarece este învățată prin protocolul de rutare RIP, alături de caracterul *, indicând o rută default.

Ruterul care originează ruta default se va anunța pe sine ca fiind next-hop al acesteia, fiind astfel folosit de către toți vecinii ca destinație pentru noua rută instalată în tabela de rutare.

RIPv2

Similarități între RIPv1 și RIPv2

Aceleași tehnici de prevenire a buclelor

- timere
- split horizon
- poison reverse

Triggered updates

Același mod de calcul al metricii

- număr maxim de hop-uri: 15

RIPv2 este o versiune revizuită a lui RIPv1. Așadar, cele două protocoale au o serie de atribute comune:

- Sunt folosite aceleași timere (**invalid, hold-down, flush**)
- Se folosesc update-uri periodice pentru a menține informația din tabelele de rutare corectă
- În cazul unei schimbări de topologie, ambele protocoale utilizează update-uri provocate pentru a propaga informația
- Amândouă protocoalele folosesc o metrică bazată pe numărul de hopuri, care, în ambele cazuri, poate fi maxim 15
- Se folosesc **split horizon**, sau **split horizon** în combinație cu **poison reverse**, pentru a preveni apariția buclelor de rutare

Diferențe între RIPv1 și RIPv2

RIPv1

- classful și distance vector
- nu suportă rețelele discontinue
- nu suportă VLSM
- nu trimită masca de rețea în update
- update-urile sunt broadcast

RIPv2

- classless și distance vector
- masca de rețea este inclusă în update-uri
- update-urile sunt multicast
- permite autentificarea

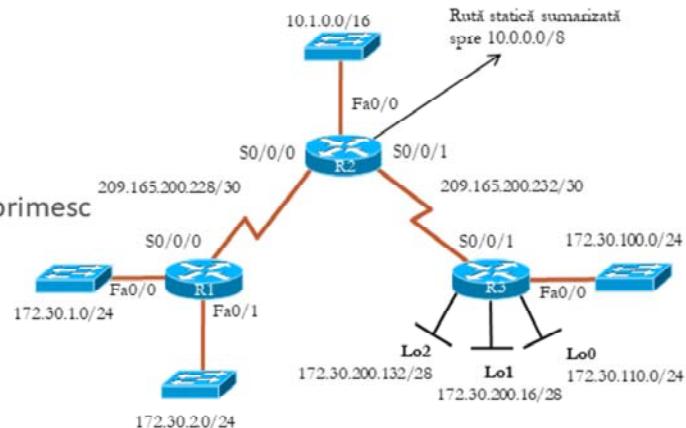
Fiind o extensie a versiunii anterioare, RIPv2 aduce o serie de îmbunătățiri protocolului original, făcând-l mai eficient și mai flexibil. Printre aceste îmbunătățiri, cele mai importante sunt:

- **RIPv2 este un protocol classless, ceea ce înseamnă că masca de rețea este inclusă în update-urile trimise de ruter.**
- **Folosește adrese multicast pentru a trimite update-uri, ceea ce are ca efect economisirea lățimii de banda în cadrul rețelelor multiacces**
- **Permite autentificarea, pentru o mai bună securizare a rețelei și un control sporit al procesului de rutare**
- **Suportă summarizarea manuală a rutelor**

Scenariu – Ruta statică

Ruta statică summarizată

- ieșire prin interfața NULL0
- traficul este ignorat
- ruta este propagată
- interfețele NULL nu trimit sau primesc trafic
- recomandabil în laborator



Exemplu configurare:

```
R2(config)#ip route 10.0.0.0 255.0.0.0 Null0
```

Orice pachet care are ca interfață de ieșire o interfață null este aruncat. Acestea se folosesc în mai multe scopuri:

1. Pentru a bloca traficul către o anumită rețea
2. Pentru a summariza mai multe rețele

În cazul în care facem o summarizare folosind o rută statică, setăm interfața de ieșire null. Traficul către rețelele summarizate nu va fi blocat deoarece ruterul nu va lăsa în considerare ruta generică (cea către interfață null) pentru că are rețelele care au fost summarizate, acestea fiind mai puțin generice. Avantajul îl reprezintă faptul că protocolul de rutare trimite în update-urile sale doar rețea summarizată.

O rută către interfață null se adaugă cu comanda `ip route <retea> <masca> <interfata-de-iesire>` la fel ca orice altă rută statică.

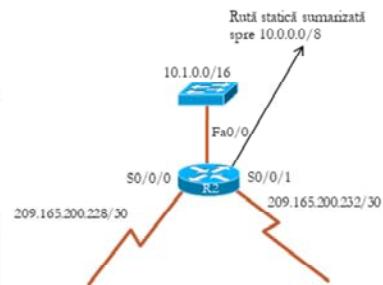
Scenariu – Redistribuire de rute

Prin redistribuire se pot introduce:

- rute statice într-un protocol de rutare
- rute direct conectate într-un protocol de rutare
- rute dintr-un protocol de rutare în altul

Exemplu R2:

```
R2(config)# router rip  
R2(config-router)# network 10.0.0.0  
R2(config-router)# network 209.165.200.0  
R2(config-router)# redistribute static
```



Rețelele conectate sunt introduse prin comanda network.

- ruta statică este introdusă prin redistribuire.
- RIP rulează pe interfața NULL0?

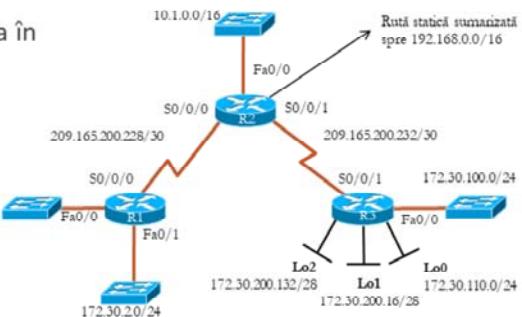
Un ruter care rulează protocolul RIP va trimite în update-uri informații din tabela sa de rutare. Totuși, în mod normal, protocolul va include în update numai rutele care au fost învățate prin RIP (incluzând și rețelele direct conectate pentru care s-a dat comanda `network`). Dacă un ruter are, de exemplu, configurația o rută statică, ea nu va fi trimisă prin update-urile RIP. Pentru aceste cazuri se folosește redistribuirea.

Redistribuirea presupune adăugarea rețelelor de la un tip de sursă (protocol dinamic, rută statică, conexiune directă) în update-urile unui protocol de rutare. Pentru a redistribui rute statice se folosește comanda `redistribute static` în modul de configurare al ruterului. O rețea direct conectată va fi introdusă prin comanda `network` sau `redistribute connected`.

Scenariu – Update-uri classful

RIPv1 nu trimite masca de rețea în update-uri.

Comanda `debug ip rip`:



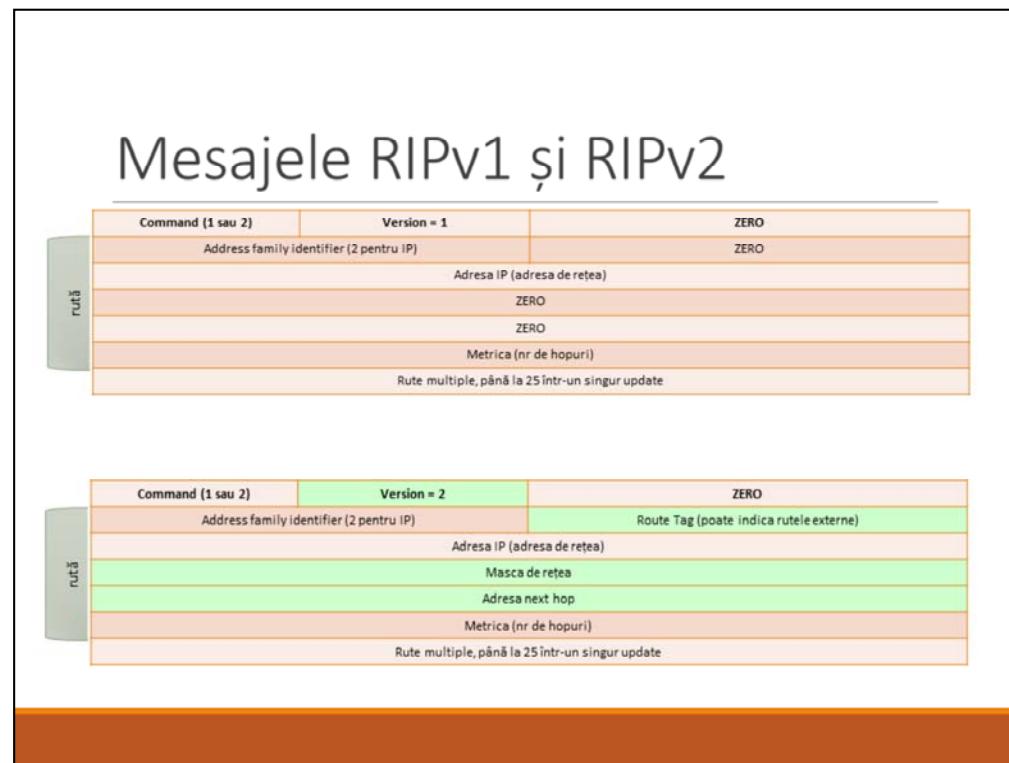
```
R2#debug ip rip
RIP protocol debugging is on

RIP: sending v1 update to 255.255.255.255 via Serial0/0/0 (209.165.200.229)
RIP: build update entries
    network 10.0.0.0 metric 1
    subnet 209.165.200.232 metric 1
```

Un mare dezavantaj al protocolelor classful este faptul că nu se trimit masca de rețea în update-uri. În momentul în care rutерul trimite un update, se face summarizare la rețelele classful din care fac parte adresele prezente în update-uri.

Fie situația în care R1 și R3 trimit update-uri către R2. Ambele rutere vor trebui, mai întâi, să sumarizeze adresele tuturor subrețelelor lor 172.30.0.0/16 la 172.30.0.0 pentru a trimite update la R2. Aceasta va vedea ambele rute ca fiind căi de cost egal către 172.30.0.0/16 și le va instala pe amândouă, astfel obținând două căi către aceeași rețea. Deoarece R2 are o singură rețea, cu două căi de acces, orice pachet trimis la una dintre subrețelele legate la R1 sau R3 are șanse să nu ajungă, doarece pachetul poate fi trimis pe calea greșită.

Mesajele RIPv1 și RIPv2



După cum se poate observa, mesajul protocolului RIPv2 este similar cu cel al protocolului RIPv1, având două sau trei câmpuri în plus.

Prima extensie importantă a mesajului RIPv2 este masca de rețea. Aceasta este plasată într-un câmp de 32 de biți.

A doua extensie semnificativă a mesajului RIPv2 este adresa de next-hop.

Aceasta va fi folosită pentru a trimite pachetul pe cea mai bună rută pentru a ajunge la destinație. Dacă acest câmp este setat la 0.0.0.0, adresa de la care se trimite update-ul este cel mai bun next-hop.

Câmpul „Route Tag” este folosit pentru a marca ruturile care au fost importate (redistribuite) din alte protocole de rutare. Când un ruter primește informații despre o rețea ca fiind importată, acesta va conserva valoarea acestui câmp.

Activarea RIPv2

Implicit, ruterele pornesc RIP în versiunea 1.

```
R2(config-router)#do show ip protocols
Routing Protocol is "rip"
[...]
Default version control: send version 1, receive any version
  Interface      Send   Recv  Triggered RIP  Key-chain
  Serial0/0/0        1      1 2
  Serial0/0/1        1      1 2
```

- RIPv1 este forward-compatible.
- Primește orice versiune dar trimite doar v1.
- Activarea RIPv2 (sau revenirea la v1):

```
R1(config-router)#version ?
<1-2>  version
R1(config-router)#version 2
```

În mod normal, când un ruter Cisco este configurat cu un protocol RIP, el va rula RIPv1. Acesta va trimite mesaje RIPv1, dar va putea interpreta mesaje atât de tip RIPv1, cât și RIPv2. Ruterul configurat cu RIPv1 va ignora, pur și simplu, câmpurile specifice RIPv2. Acest lucru înseamnă că RIPv1 este forward-compatible.

Pentru a verifica ce protocol este folosit de ruter vom folosi comanda `show ip protocols`. Pentru a schimba versiunea protocolului RIP, se va folosi comanda `version <versiune_RIP>`.

Auto-sumarizarea

Implicit, RIPv2 trimită masca de rețea, dar face aceeași summarizare classful ca RIPv1:

```
R1(config-router)#do show ip protocols
Routing Protocol is "rip"
[...]
Default version control: send version 2, receive version 2
  Interface      Send   Recv  Triggered RIP  Key-chain
  Serial0/0/0        2       2
  Serial0/0/1        2       2
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
  209.165.200.0
  10.0.0.0
[...]
Distance: (default is 120)
```

- Dezactivarea summarizării automate:

```
R1(config-router)#no auto-summary
```

Deși RIPv2 trimită în update-uri și masca de rețea, acesta face același tip de summarizare a adreselor pe care îl face RIPv1. Așadar, în topologia anterioară, ruterul R2 va avea în tabela de rutare tot o adresă classful cu două căi asociate.

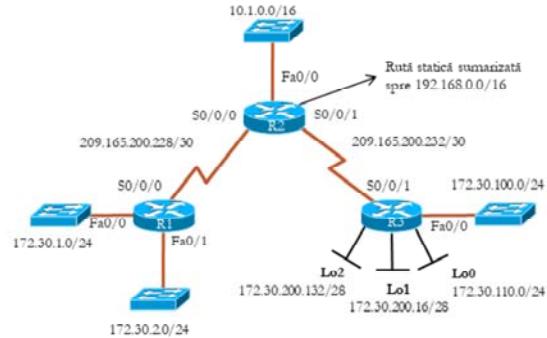
Pentru ca RIPv2 să nu mai facă summarizare, se folosește comanda `no auto-summary`. Acest lucru va face ca protocolul RIPv2 să includă în update-urile sale toate adresele subrețelelor împreună cu măștile lor. În această situație, comanda `show ip protocols` va afișa „Automatic network summarization is not in effect”.

Comanda `no auto-summary` nu va avea efect pe un ruter care implementează RIPv1. Deși pe Cisco IOS se va putea da comanda `no auto-summary`, sistemul va ignora comanda în cazul protocolului RIPv1.

RIPv2 și supernet-uri

RIPv2 include în update-uri masca supernet-urilor.

Verificare cu `debug ip rip`:



```
R2# debug ip rip
RIP protocol debugging is on
RIP: sending v2 update to 224.0.0.9 via Serial0/0/0 (209.165.200.229)
RIP: build update entries
  10.1.0.0/16 via 0.0.0.0, metric 1, tag 0
  172.30.100.0/24 via 0.0.0.0, metric 2, tag 0
  172.30.110.0/24 via 0.0.0.0, metric 2, tag 0
  172.30.200.16/28 via 0.0.0.0, metric 2, tag 0
  172.30.200.32/28 via 0.0.0.0, metric 2, tag 0
  192.168.0.0/16 via 0.0.0.0, metric 1, tag 0
  209.165.200.232/30 via 0.0.0.0, metric 1, tag 0
```

Unul dintre obiectivele CIDR (Classless Inter-Domain Routing) este de a oferi un mecanism de agregare a informației de rutare. Un supernet este un bloc de adrese classful continue, care sunt adresate ca o singură rețea. Supernet-urile au măști de rețea mai generale decât masca classful. Pentru ca un supernet să fie inclus într-o tabelă de rutare, protocolul de rutare trebuie să aibă capacitatea de a transmite masca aceluia supernet, deci va trebui să fie un protocol classless, precum RIPv2.

Se va folosi comanda `debug ip rip` pentru a vedea dacă un supernet este inclus în tabela de rutare. Nu este nevoie ca sumarizarea automată să fie dezactivată într-un protocol classless pentru ca supernet-urile să fie incluse în tabela de rutare.

Supernet-urile se definesc manual. Dacă s-a creat un supernet, sumarizarea automată nu mai are efect asupra rețelelor din supernet chiar dacă este activă. Practic sumarizarea este deja efectuată.

Autentificarea în RIPv2

Avantaj:

- minimizarea riscului de a accepta informații de rutare nevalide
 - inclusiv împotriva atacurilor de rutare

Protocolele de rutare ce suportă autentificare:

- RIPv2
- EIGRP
- OSPF
- IS-IS
- BGP

O mare problemă a protocolelor de rutare este faptul că își trimit update-urile folosind pachete IP, ceea ridică probleme de securitate. De exemplu, un ruter poate să accepte update-uri invalide inițiate de un atacator care intenționează să captureze pachetele trimise, păcălind ruterul să trimită mesaje către o destinație greșită. O altă sursă de update-uri invalide poate fi un ruter incorect configurat sau un echipament conectat la rețea, care rulează un protocol de rețea, fără știință utilizatorului.

Oricare ar fi motivul, este o bună practică să se folosească autentificarea între ruterele care transmit informații. RIPv2, EIGRP, OSPF, IS-IS și BGP pot fi configurate să utilizeze autentificare. Acest lucru asigură faptul că ruterele vor accepta numai pachetele trimise de surse care cunosc datele de autentificare.

Configurarea autentificării pe un ruter nu va cripta tabela de rutare în momentul trimiterii.

Configurarea autentificării în RIPv2

Autentificarea se realizează la nivel de interfață.

Primul pas – crearea unui key chain

- numele key chain-ului este MYRIP, cu o cheie (parolă) „cisco”
- indexul cheii nu e relevant

```
R2(config)#key chain MYRIP  
R2(config-keychain)#key 1  
R2(config-keychain-key)#key-string cisco
```

Al doilea pas: activarea autentificării pe interfață

```
R2(config)#interface serial 0/0/1  
R2(config-if)#ip rip authentication mode ?  
    md5      Keyed message digest  
    text     Clear text authentication  
  
R2(config-if)#ip rip authentication mode md5  
R2(config-if)#ip rip authentication key-chain MYRIP
```

Autentificarea, în RIPv2, se face la nivel de interfață. Se va configura un „key-chain” cu cel puțin o parolă.

Un key-chain reprezintă un set de parole care vor fi utilizate pentru autentificare. Pentru ca autentificarea să funcționeze fără erori este necesar ca ambele echipamente să folosească același set de parole pe interfețele asociate aceleiași legături.

Parolele dintr-un key-chain sunt rotite periodic după o regulă cunoscută de ambele echipamente și pot fi transmise în clear-text sau folosind un hash md5.

Erori frecvente

Conflict de versiuni

- ruterele comunică, nu se trimit avertizări
- duce la apariția de rețele classful

Comenzi network:

- comanda **network** e classful
- poate cuprinde mai multe interfețe decât se dorește

Sumarizarea automată

- duce la absența rutelor corecte din tabela de rutare
- rețeaua nu poate fi convergentă dacă adresarea e discontinuă

Autentificare eronată

- doar la un capăt, conflict de parolă, conflict de mod

Atunci când se face depanarea RIPv2, se au în vedere următorii factori:

Versiunea: deși RIPv1 și RIPv2 sunt parțial compatibile (RIPv1 primește update-uri de la RIPv2 dar nu și invers), RIPv1 nu suportă subrețele discontinue, VLSM sau CIDR, ceea ce poate duce la apariția destinațiilor invalide sau la nepropagarea anumitor rețele. În general, este recomandat ca toate ruterele dintr-o rețea să ruleze același protocol de rutare, cu excepția cazului în care circumstanțe speciale cer să fie folosite protocoale diferite.

Comanda network: Comanda **network** pornește trimitera și recepționarea de actualizări pe toate interfețele care aparțin rețelei classful menționate ca parametru. Dacă există două interfețe care au asociate două subrețele aparținând aceleiași rețele classful, folosind comanda **network** activăm trimitera de actualizări pe ambele interfețe. Dezactivarea trimiterii de actualizări pe o singură interfață se face cu ajutorul comenzi **passive-interface**.

Erori frecvente

Conflict de versiuni

- ruterele comunică, nu se trimit avertizări
- duce la apariția de rețele classful

Comenzi network:

- comanda **network** e classful
- poate cuprinde mai multe interfețe decât se dorește

Sumarizarea automată

- duce la absența rutelor corecte din tabela de rutare
- rețeaua nu poate fi convergentă dacă adresarea e discontinuă

Autentificare eronată

- doar la un capăt, conflict de parolă, conflict de mod

Autentificare: o medodă de autentificare configurată greșit va genera conflicte care pot cauza erori în tabela de rutare și în final pierderea pachetelor.

Sumarizarea automată: dacă există nevoie să se trimită pachete la o anumită subrețea, folosirea sumarizării automate poate cauza probleme. Sumarizarea automată face ca RIPv2 să se comporte ca RIPv1 în ceea ce privește rețelele classless.