

Curs 2: Recapitulare protocoale de rutare

Retele WAN

Silviu Vasile
vsl@fmi.unibuc.ro

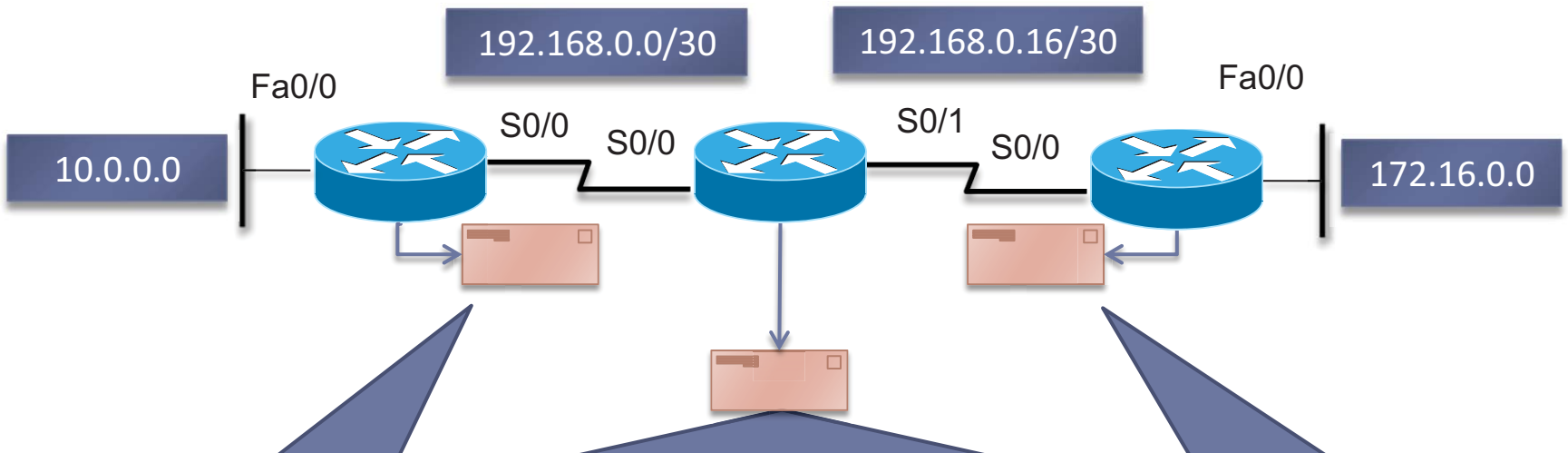
Protocoloale DV - Generalități

▶ DV

- ▶ Transmit informații la vecini
 - ▶ Transmit întreaga tabelă de rutare
 - ▶ Update-uri periodice
 - ▶ Folosesc mai puține resurse
 - ▶ Convergență greoaie
 - ▶ Puțin scalabile
-

Protocele DV – funcționare

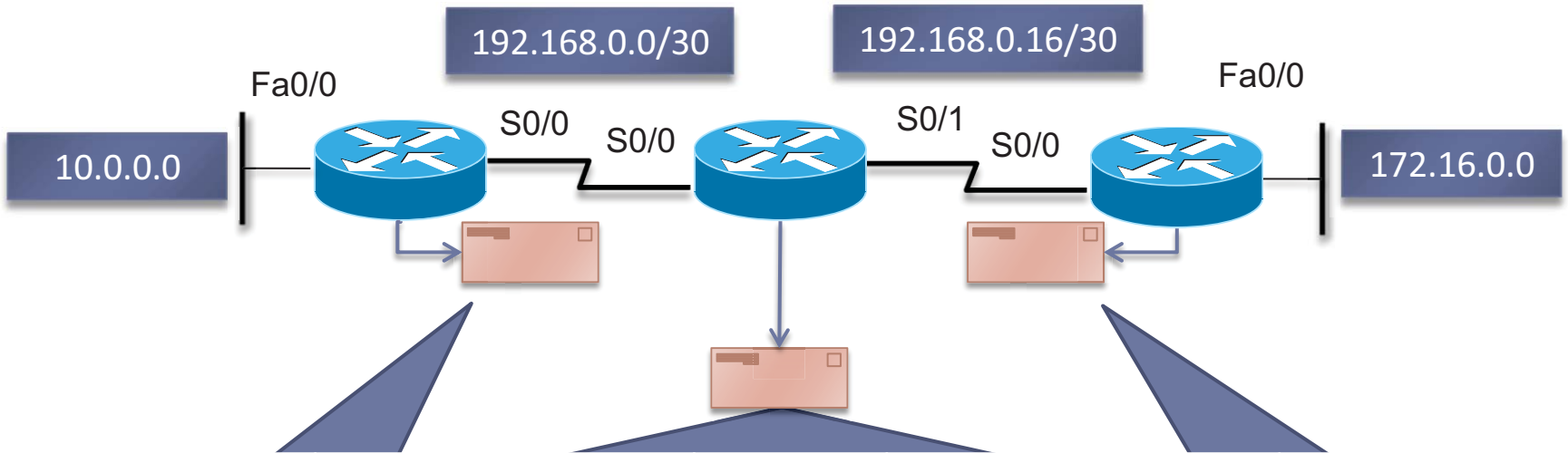
► Tabela de rutare după primul update



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	S0/1	1			

Protocele DV – funcționare

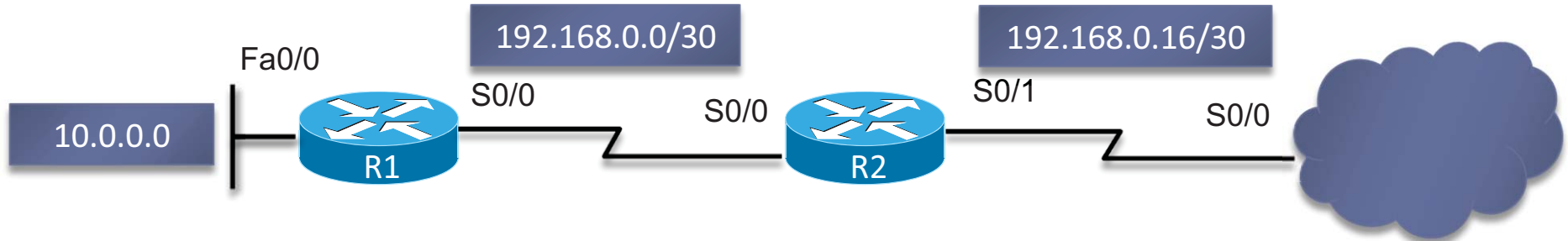
- ▶ Tabela de rutare după al doilea update
- ▶ Rețeaua a convers



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

Bucle de rutare

- ▶ Din cauza comportamentului dat de funcționarea timerelor la intervale fixe, se pot produce inconsistențe de rutare

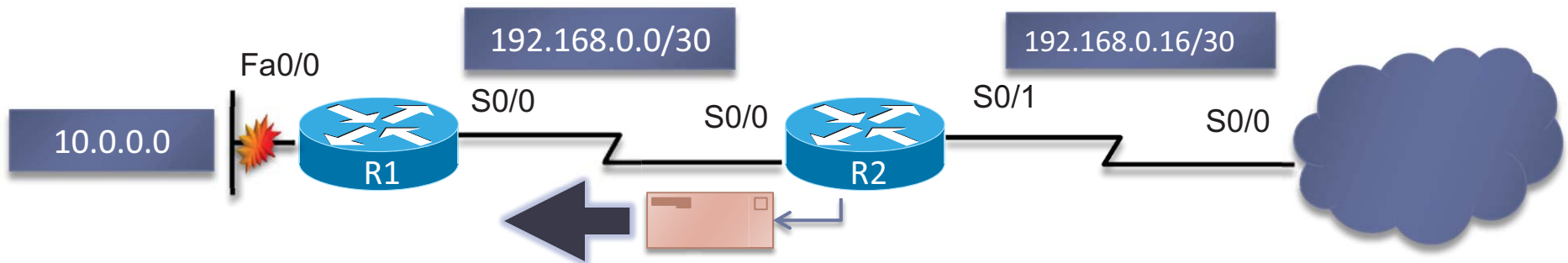


Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

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

Bucle de rutare

- ▶ Când rețeaua direct conectată a lui R1 pică, acesta o scoate din tabela de rutare
- ▶ Înainte ca R1 să poată trimite update despre rețeaua picată, R2 anunță aceeași rețea cu cale de 1 hop prin el

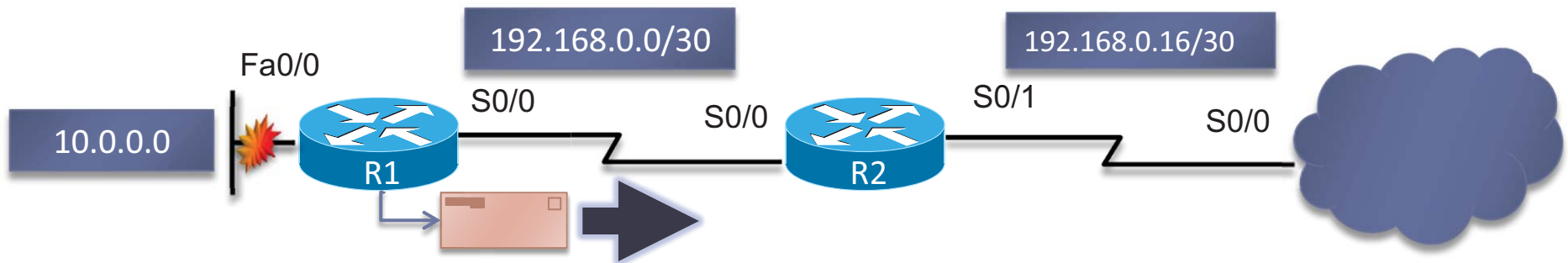


Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

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

Bucle de rutare

- ▶ R1 instalează ruta către 10.0.0.0 crezând că R2 cunoaște altă cale către această rețea
- ▶ Se creează o buclă de rutare între R1 și R2



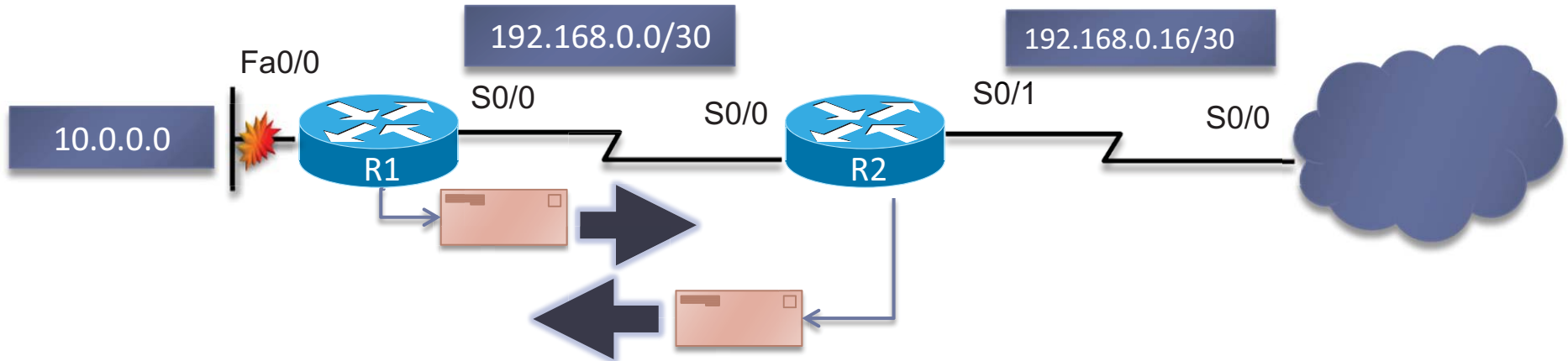
Rețea	Interfață	Număr de hopuri
10.0.0.0	S0/0	2
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

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	1

Prevenirea buclelor de rutare

- ▶ Cauze posibile ale buclelor:
 - ▶ Actualizări incorecte sau inconsecvente datorate convergenței lente după o schimbare în topologie
 - ▶ Informații de rutare incorecte sau incomplete
 - ▶ Protocoalele DV implementează mai multe metode de evitare a buclelor:
 - ▶ Max hop count
 - ▶ Split horizon
 - ▶ Route Poisoning
 - ▶ Hold-down timers
-

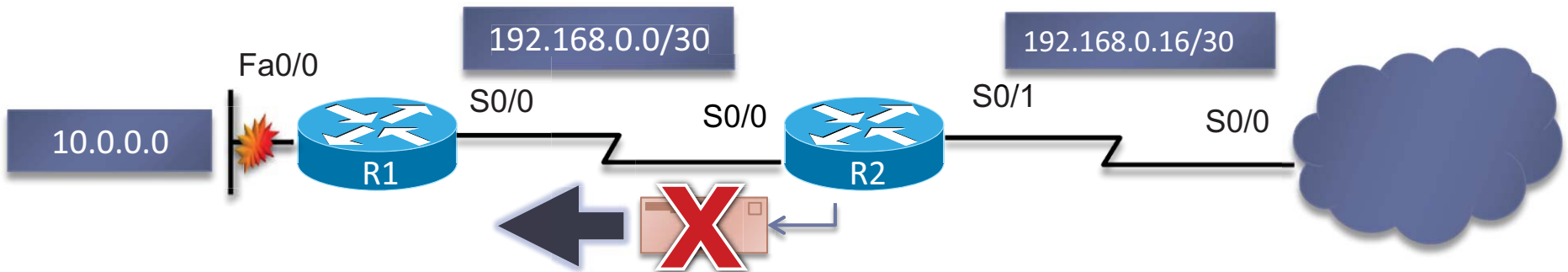
Max hop count



- ▶ În cazul apariției buclelor de rutare, ruterele pot ajunge să schimbe informații de rutare către destinația ce a cauzat bucla, rutele având metrica din ce în ce mai mare
- ▶ Fenomenul este cunoscut ca “count to infinity”
- ▶ Astfel, resursele ruterului vor fi consumate pentru a procesa informații de rutare despre o destinație ce cauzează o buclă de rutare.
- ▶ Pentru a limita acest comportament, se limitează metrica pe care o poate avea o rută
- ▶ În cazul RIP, Max Hop Count = 15.

Split horizon

- ▶ Este un mecanism de prevenire a buclelor
- ▶ Presupune ca un update despre o rută să nu fie trimis pe interfața de ieșire a rutei respective
- ▶ În cazul de mai jos, R2 nu va trimite niciodată un update despre 10.0.0.0 lui R1, acesta fiind ruterul prin care are această rută în tabela de rutare

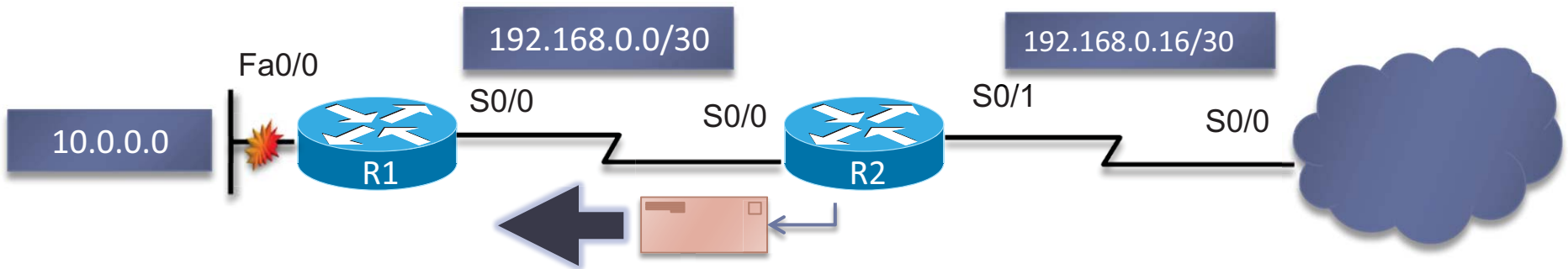


Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

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

Split horizon with poison reverse

- ▶ În general split horizon este implementat cu poison reverse
- ▶ Regula “poison reverse” spune că ruta 10.0.0.0 ar trebui trimisă de la R2 la R1, dar cu metrică infinită conform protocolului de rutare (pentru RIP = 16)
- ▶ Se merge pe ideea că “a primi vești proaste este mai bine decât a nu primi vești deloc”

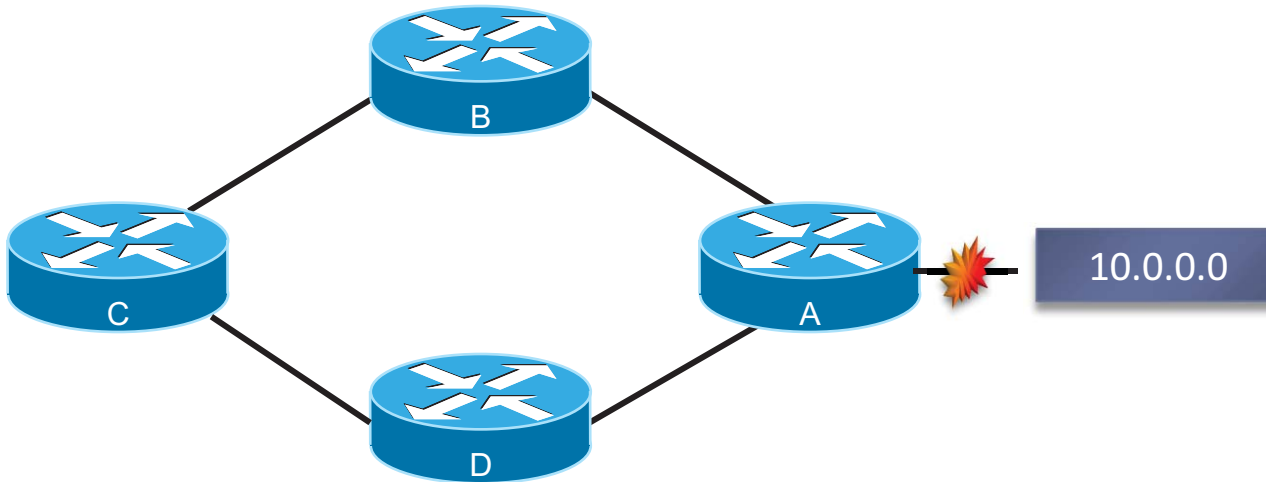


Rețea	Interfață	Număr de hopuri
10.0.0.0	S0/0	16
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

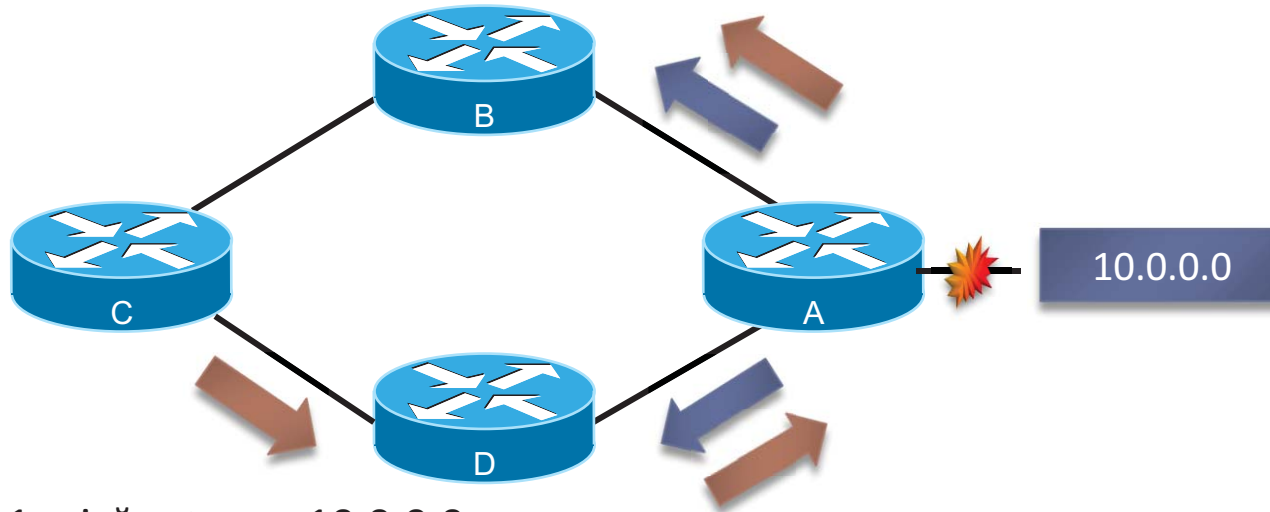
Rețea	Interfață	Număr de hopuri
192.168.0.0	S0/0	0
192.168.0.16	S0/1	0
10.0.0.0	S0/0	1

Este split-horizon îndeajuns?

- ▶ Pentru a împiedica buclele între vecini, da.
- ▶ Pentru bucle mai complexe avem nevoie de mecanisme adiționale



Este split-horizon îndeajuns?

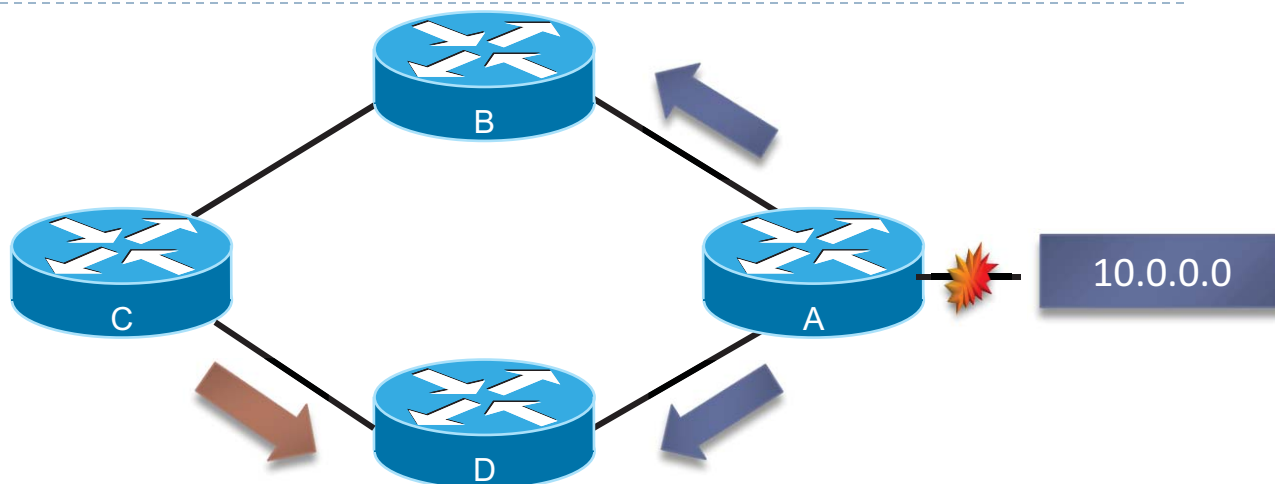


- ▶ Pasul 1: pică rețeaua 10.0.0.0
- ▶ Pasul 2: A trimite update-uri către B și D informându-i de acest aspect
- ▶ Pasul 3: C trimite un update către D spunând că el cunoaște rețeaua 10.0.0.0 cu o metrică de 3.
- ▶ Q: În trecut D i-a trimis ruta 10.0.0.0 lui C. De ce pasul 3 nu încalcă regula split-horizon?
- ▶ Pasul 4: D instalează ruta prin C cu metrică de 3 și trimite update către A
- ▶ Pasul 5: se creează o buclă în momentul în care A anunță ruta către B cu metrică de 4

Soluția: hold-down timers

- ▶ Când o rețea este marcată ca inaccesibilă (în urma unui update primit de la vecinul de la care a învățat-o), un ruter porneste un holddown timer.
 - ▶ După expirarea acestuia ruta este eliminată din tabela de rutare.
 - ▶ Dacă înainte de expirare primește informații că ruta este din nou accesibilă, atunci:
 - ▶ dacă informația a venit de la același vecin, rețeaua marcată ca accesibilă
 - ▶ dacă informația a venit de la alt vecin, cu o metrică mai proastă, informația este ignorată
 - ▶ dacă informația a venit de la alt vecin, cu o metrică mai bună, informația este considerată corectă, timerul este oprit și se modifică tabela de rutare în mod corespunzător
-

Hold-down timer - funcționare



- ▶ Pasul 1: pică rețeaua 10.0.0.0
 - ▶ Pasul 2: A trimite update-uri către B și D informându-i de acest aspect
 - ▶ Pasul 3: B și D pornesc un hold-down timer care durează de obicei de 4 ori intervalul dintre update-uri de rutare
 - ▶ Pasul 4: C trimite un update către D spunând că el cunoaște rețeaua 10.0.0.0 cu o metrică de 3.
 - ▶ Pasul 5: D nu acceptă update-ul deoarece a fost primit cât hold-down timerul era încă activ și are o metrică mai proastă decât cea anterioară ($3 > 1$)
-

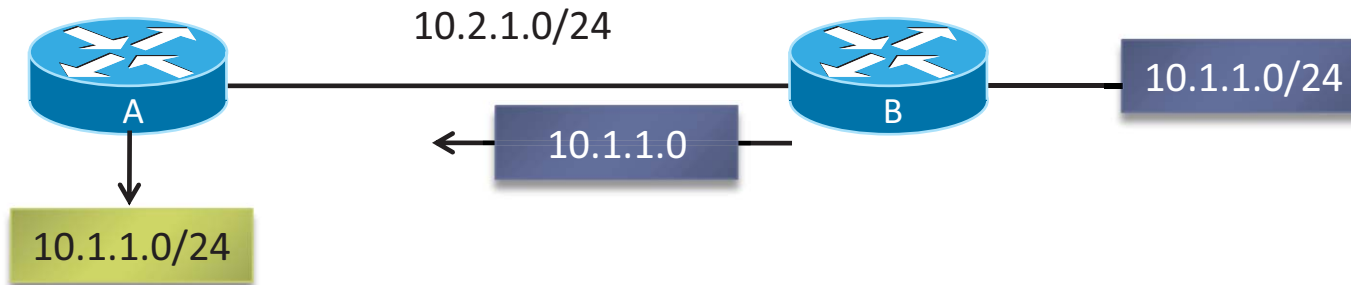
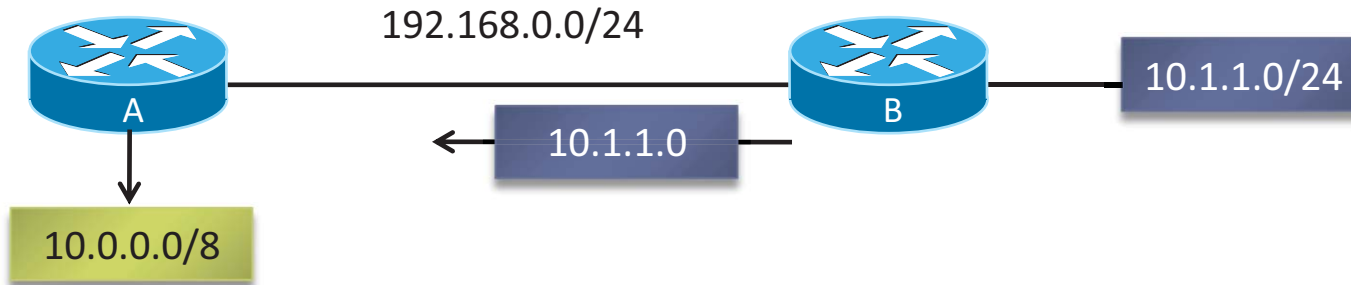
RIP

▶ Generalități:

- ▶ Protocol distance vector
 - ▶ Update periodic: 30 secunde
 - ▶ Metrica: hop count (max=15)
 - ▶ Două versiuni:
 - ▶ RIP v1: - classful
 - ▶ RIP v2: - classless
 - ▶ RIP implementează mecanismele de Split Horizon with Poison reverse și Holddown timer
 - ▶ folosește protocolul UDP pe portul 520 pentru trimiterea actualizărilor
 - ▶ folosește triggered updates pentru a grăbi propagarea informației în rețea
-

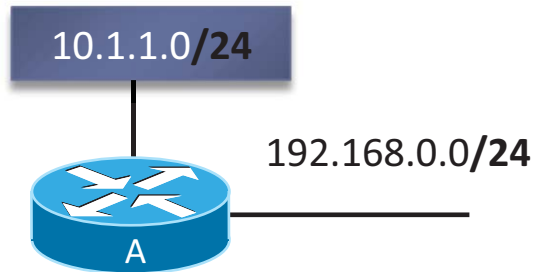
RIPv1

- ▶ Protocol de rutare classful
 - ▶ Nu trimite masca de rețea în update-uri
- ▶ Comportament RIPv1 la primirea update-urilor



RIPv1 – trimiterea updateurilor

- ▶ Comportamentul RIPv1 la trimiterea update-urilor
 - ▶ Dacă masca de pe interfața de ieșire coincide cu masca rețelei ce trebuie inclusă în update, update-ul este trimis



- ▶ Dacă masca rețelei ce trebuie inclusă în update este /32 atunci update-ul este trimis
 - ▶ Dacă masca rețelei ce trebuie inclusă în update este diferită de masca de pe interfață, atunci update-ul nu este trimis
 - ▶ Concluzie de design RIPv1: se pot folosi măști non-classful, dar trebuie folosită aceeași mască în toată rețeaua (excepție făcând /32 pentru loopback-uri)
-

Configurare RIPv1

- ▶ Exemplu de configurare:



```
Flash(config)# router rip
Flash(config-router)# network 10.10.11.0
Flash(config-router)# network 10.10.12.0
SAU
Flash(config)# router rip
Flash(config-router)# network 10.0.0.0
```

- ▶ Comanda *network* are 3 funcții:
 - ▶ ce rețele vor fi incluse în updateuri
 - ▶ pe ce interfețe să trimită actualizări
 - ▶ pe ce interfețe să asculte pentru actualizări
 - ▶ Obs: Comanda *network* se folosește numai pentru rețelele direct conectate
 - ▶ Obs: RIP are comportament *classful* la activare
-

Configurare RIP

- ▶ Se pot face configurări mai avansate, pentru mărirea performanței:

- ▶ Dezactivarea Split-horizon

```
Router(config-router)# no ip split-horizon
```

- ▶ Setarea timpilor (update, invalid, holddown, flush)

```
Router(config-router)# timers basic 30 180 180 240
```

- ▶ Setarea intervalului de update

```
Router(config-router)# update-timer 40
```

- ▶ Dezactivarea update-urilor pe o interfață

```
Router(config-router)# passive-interface f0/0
```

- ▶ Folosirea RIP v2

```
Router(config-router)# version 2
```

- ▶ Interpretarea pachetelor primite (versiune)

```
Router(config-router)# ip rip receive version 1 2
```

Verificare RIP

```
Flash# sh ip protocols
```

```
Routing Protocol is "rip"
```

```
Outgoing update filter list for all interfaces is not set
```

```
Incoming update filter list for all interfaces is not set
```

```
Sending updates every 30 seconds, next due in 0 seconds
```

```
Invalid after 180 seconds, hold down 180, flushed after 240
```

```
Redistributing: rip
```

```
Default version control: send version 1, receive any version
```

Interface	Send	Recv	Triggered	RIP	Key-chain
Ethernet0/0	1	1	2		
Ethernet0/1	1	1	2		
Loopback0	1	1	2		

```
Automatic network summarization is in effect
```

```
Maximum path: 4
```

```
Routing for Networks:
```

```
10.0.0.0
```

```
172.16.0.0
```

```
Routing Information Sources:
```

Gateway	Distance	Last Update
---------	----------	-------------

```
Distance: (default is 120)
```

Verificare RIP – tabela de rutare

```
Batman#sh ip route
```

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

```
Gateway of last resort is not set
```

```
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks  
R 10.10.11.0/24 [120/1] via 10.10.14.1, 00:00:24, Ethernet0/0  
R 10.10.12.0/24 [120/1] via 10.10.13.1, 00:00:22, Ethernet0/1  
C 10.10.13.0/24 is directly connected, Ethernet0/1  
C 10.10.14.0/24 is directly connected, Ethernet0/0  
C 10.10.16.0/26 is directly connected, Loopback0  
R 11.0.0.0/8 [120/1] via 10.10.13.1, 00:00:22, Ethernet0/1
```

Load Balancing folosind RIP

- ▶ Presupune transmiterea pachetelor către o destinație pe mai multe căi în același timp
 - ▶ Căile trebuie să fie de cost egal
 - ▶ RIP suportă maxim 6 căi (4 implicit)
 - ▶ Două posibilități:
 - ▶ Fast Switching (sau CEF): transmiterea în mod per-destination
 - ▶ Process Switching: transmiterea în mod per-packet
-

Redistribuirea rutelor statice

- ▶ Rutele statice sunt importante, deoarece cu ajutorul lor se poate configura Gateway of Last Resort (default route)
 - ▶ O rută statică poate fi configurată ca alternativă la o rută dinamică dacă se setează distanța administrativă mai mare decât la cea dinamică
 - ▶ O rută statică va fi scoasă din tabela de rutare dacă interfața este cazută sau următorul hop nu mai e valabil
 - ▶ Pentru a redistribui rutele statice în RIP, folosiți comanda:
`Router(config-router)#redistribute static`
-

RIPv2

- ▶ Versiunea a două a protocolului a dus îmbunătățiri notabile
 - ▶ Comportament clasless (masca de rețea e trimisă în update-uri)
 - ▶ Autentificare folosind clear-text sau MD5
 - ▶ De ce există opțiunea de a face autentificare clear-text?
 - ▶ Sumarizarea manuală
 - ▶ În loc de trimiterea rutelor specifice, administratorul poate decide să trimită o rută mai generală pe care o specifică manual
 - ▶ În mod implicit RIPv2 face sumarizare automată la classful boundary
 - ▶ Update-urile sunt trimise folosind multicast (224.0.0.9)
 - ▶ Care e diferența între a trimite update-uri folosind 255.255.255.255 vs multicast?
-

RIPv1 vs RIPv2

RIPv1	RIPv2
Max. 15 hops	Max. 15 hops
Classful Nu transmite informații despre subnet INUTILIZABIL cu VLSM	Classless Transmite informații despre subnet Poate fi folosit cu VLSM
Fără autentificare	Autentificare clear text sau MD5
Update-uri trimise ca broadcast (255.255.255.255)	Update-uri trimise ca multicast (224.0.0.9)
Sursa update-ului este next-hop pentru ruta respectivă	Poate redirecționa ruterele ce primesc o rută spre alte rutere din același subnet
-	Folosește tag-uri pentru rute externe

Activarea RIPv2

- ▶ Implicit, routerele 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

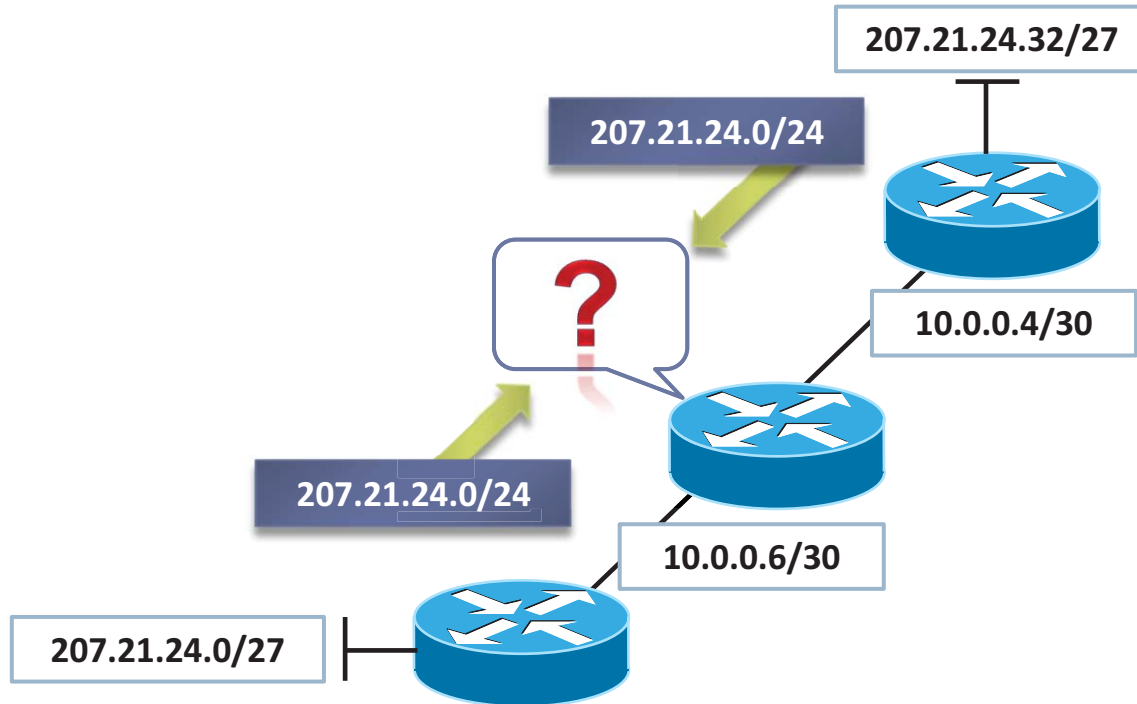
```
R1(config-router)#version ?
<1-2>  version
R1(config-router)#version 2
```

Auto-sumarizarea

- ▶ Implicit, RIPv2 trimite masca de rețea dar face aceeași sumarizare classful, ca și RIPv1:

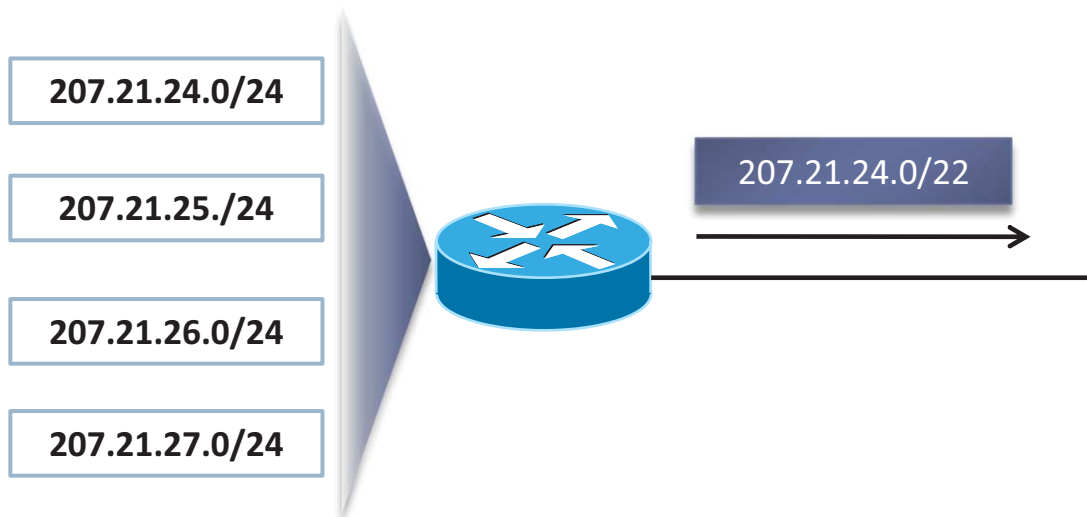
```
R1(config-router)#do show ip protocols
Routing Protocol is "rip"
[...]
  Default version control: send version 2, receive version 2
  Interface                Send  Recv  Triggered RIP  Key-chain
  Serial0/0/0               2    2
  Serial0/0/1               2    2
Automatic network summarization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

Probleme de auto-sumarizare



- ▶ Implicit, RIP sumarizează rețele anunțate la limita clasei
 - ▶ Dezactivarea sumarizării:
`Router(config-router)# no auto-summary`
-

Configurarea sumarizării manuale



- ▶ Dacă se dorește sumarizarea la o mască non-classful, administratorul poate realiza acest lucru în RIPv2
- ▶ Configurarea se face la nivel de interfață

```
Flash(config-if)#ip summary-address rip 207.21.24.0 255.255.252.0
```

Verificarea RIPv2

```
# show ip protocols  
# show ip route  
# show ip interface brief  
# show running-config
```


IGRP

- ▶ 1985: protocol dezvoltat de Cisco pentru a combate limitarea impusă de RIP (metrica: hop count)
- ▶ Protocol de tip distance vector, classful
- ▶ Metrica este una mult mai complexă:

$$[k1 * BW_{IGRP(min)} + (k2 * BW_{IGRP(min)}) / (256 - LOAD) + k3 * DLY_{IGRP(sum)}] \times [k5 / (RELIABILITY + k4)]$$

- ▶ Permite balansarea folosind căi de cost inegal
 - ▶ Introduce conceptul de domenii (Autonomous System)
 - ▶ Actualizări periodice
 - ▶ Un timp de trei ori mai mare față de RIP
 - ▶ Broadcast
-

to EIGRP

- ▶ Principala motivație a fost trecerea la un comportament de tip classless

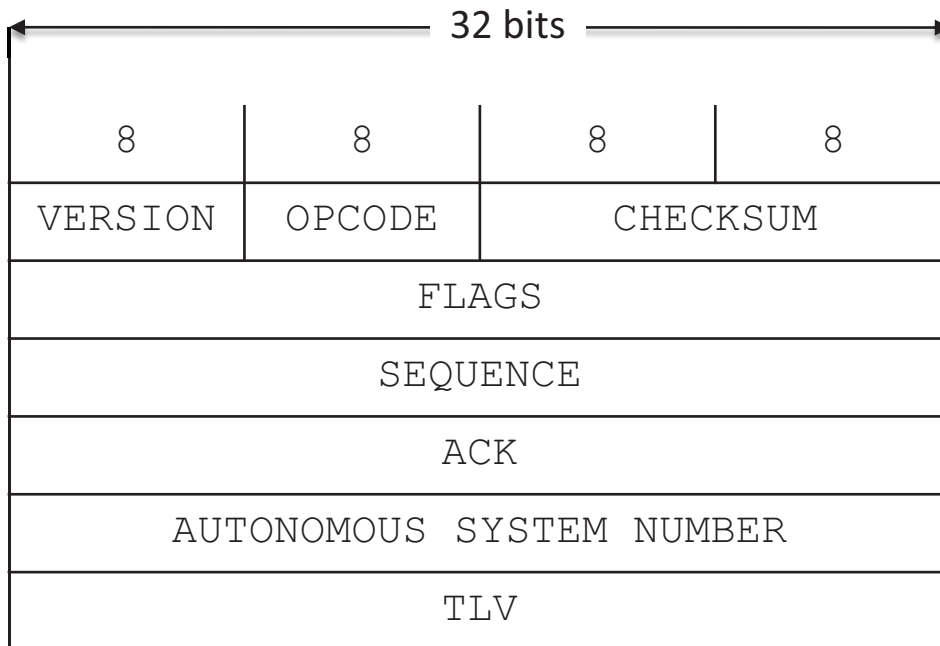
 - ▶ Schimbarea algoritmului folosit pentru selectarea rutelor
 - ▶ Trecerea de la Bellman-Ford la DUAL
 - ▶ EIGRP este considerat un protocol distance vector (sau hibrid)
-

to EIGRP

- ▶ Trecerea la actualizări neperiodice, parțiale și direcționate
 - ▶ Pot fi trimise atât multicast (224.0.0.10) cât și unicast
 - ▶ Distanța administrativă: 90/170 (internă/externă)
 - ▶ Independent față de protocoalele rutate (PDM)
 - ▶ IPv4, IPX, IPv6
-

EIGRP Packet Header

- ▶ EIGRP folosește protocolul RTP (Reliable Transport Protocol)
 - ▶ Protocol de nivel 4 proprietar Cisco
 - ▶ Mesajele pot fi transmise reliable/unreliable



TLVs :

Type/Length/Value

Câmpul Type poate avea valoarea:

- ▶ 0x0001 -> EIGRP Parameters
 - ▶ 0x0102 -> Internal Routes
 - ▶ 0x0103-> External Routes
-

Tabele EIGRP

- ▶ Folosind mesajele definite anterior EIGRP construiește următoarele trei tabele:
 - ▶ Tabela de vecini
 - ▶ Construită și menținută prin mesaje de tip Hello
 - ▶ Tabela de topologie
 - ▶ Construită și actualizată prin mesaje de tip Update/Query/Reply
 - ▶ Tabela de rutare
 - ▶ Construită din tabela de topologie folosind algoritmul DUAL
-

Tabela de vecini

- ▶ Pentru a putea stabili adiacențe trebuie activat procesul:

```
router(config)#router eigrp AS
```

- ▶ AS definește domeniul și funcționează ca un “process ID”

- ▶ Activarea pe o anumită interfață:

```
router(config-router)#network ADRESA_RETEA [WILDCARD]
```

- ▶ Se include și rețeaua (și masca de rețea) în actualizările EIGRP

- ▶ Verificarea adiacențelor:

```
router#show ip eigrp neighbors
```

Tabela de vecini

- ▶ Hello interval:
 - ▶ 60 sec - NBMA(X.25, Frame Relay, ATM), viteza < 1544 Mbps
 - ▶ 5 sec - T1, Ethernet, viteza >1544 Mbps
 - ▶ Se poate modifica per interfață
 - ▶ `ip hello-interval eigrp`
- ▶ Hold time
 - ▶ Timp maxim înainte ca un vecin sa fie considerat inaccesibil
 - ▶ Implicit 3 x Hello interval
 - ▶ Se poate modifica per interfață
 - ▶ `ip hold-time eigrp`

```
# Frame 3 (64 bytes on wire, 64 bytes captured)
# Cisco HDLC
# Internet Protocol, Src: 192.168.30.4 (192.168.30.4), Dst: 224.0.0.10 (224.0.0.10)
# Cisco EIGRP
  Version = 2
  Opcode = 5 (Hello)
  Checksum = 0xee6a
  Flags = 0x00000000
  Sequence = 0
  Acknowledge = 0
  Autonomous System : 100
# EIGRP Parameters
  Type = 0x0001 (EIGRP Parameters)
  Size = 12 bytes
  K1 = 1
  K2 = 0
  K3 = 1
  K4 = 0
  K5 = 0
  Reserved
  Hold Time = 15
# Software version: IOS=12.2, EIGRP=1.2
  Type = 0x0004 (Software Version)
  Size = 8 bytes
  IOS release version = 12.2
  EIGRP release version = 1.2
```

Tabela de vecini

- ▶ Condițiile necesare pentru stabilirea adiacenței:
 - ▶ Același număr de AS
 - ▶ Aceleași valori pentru parametri K
 - ▶ Trebuie ca adresa IP sursă a pachetului primit să fie în aceeași rețea cu interfața pe care a fost primit acest pachet
 - ▶ Succesul procesului de autentificare (opțional)
 - ▶ O adiacență este considerată nefuncțională atunci când pe un interval de timp (egal cu valoarea Hold time) nu s-au primit pachete Hello
 - ▶ Se ia în considerare valoarea primită de la vecin, nu cea trimisă
-

Tabela de topologie

- ▶ Pentru a modifica valorile K folosim comanda

```
router(config-router)#metric weights tos k1 k2 k3 k4 k5
```

- ▶ Pentru a modifica metrica EIGRP se modifică parametrii per interfață

- ▶ DELAY

```
Delay <tens of microseconds>
```

- ▶ BANDWIDTH

```
Bandwidth <kilobits>
```

Tabela de topologie

- ▶ Pentru menținerea unei tabele fără bucle se folosește algoritmul DUAL (Diffusing Update Algorithm)
 - ▶ Acesta salvează toate căile fără buclă existente către o destinație
 - ▶ Se definesc următoarele concepte pentru EIGRP
 - ▶ “reported distance (RD)” – distanța primită de la un vecin despre o rețea
 - ▶ Cunoscută și sub numele de “advertised distance (AD)”
 - ▶ “distance (D)” – distanța până la o rețea
 - ▶ $RD + \text{costul între router și vecin}$
 - ▶ “feasible distance (FD)” – cea mai mică distanță până la o rețea
 - ▶ $\min(D)$
-

Tabela de topologie

▶ Toți vecinii pentru care este satisfăcută condiția $D = FD$ vor construi tabela de rutare și poartă denumirea de “succesor(S)”

▶ Implicit doar patru, maxim șase

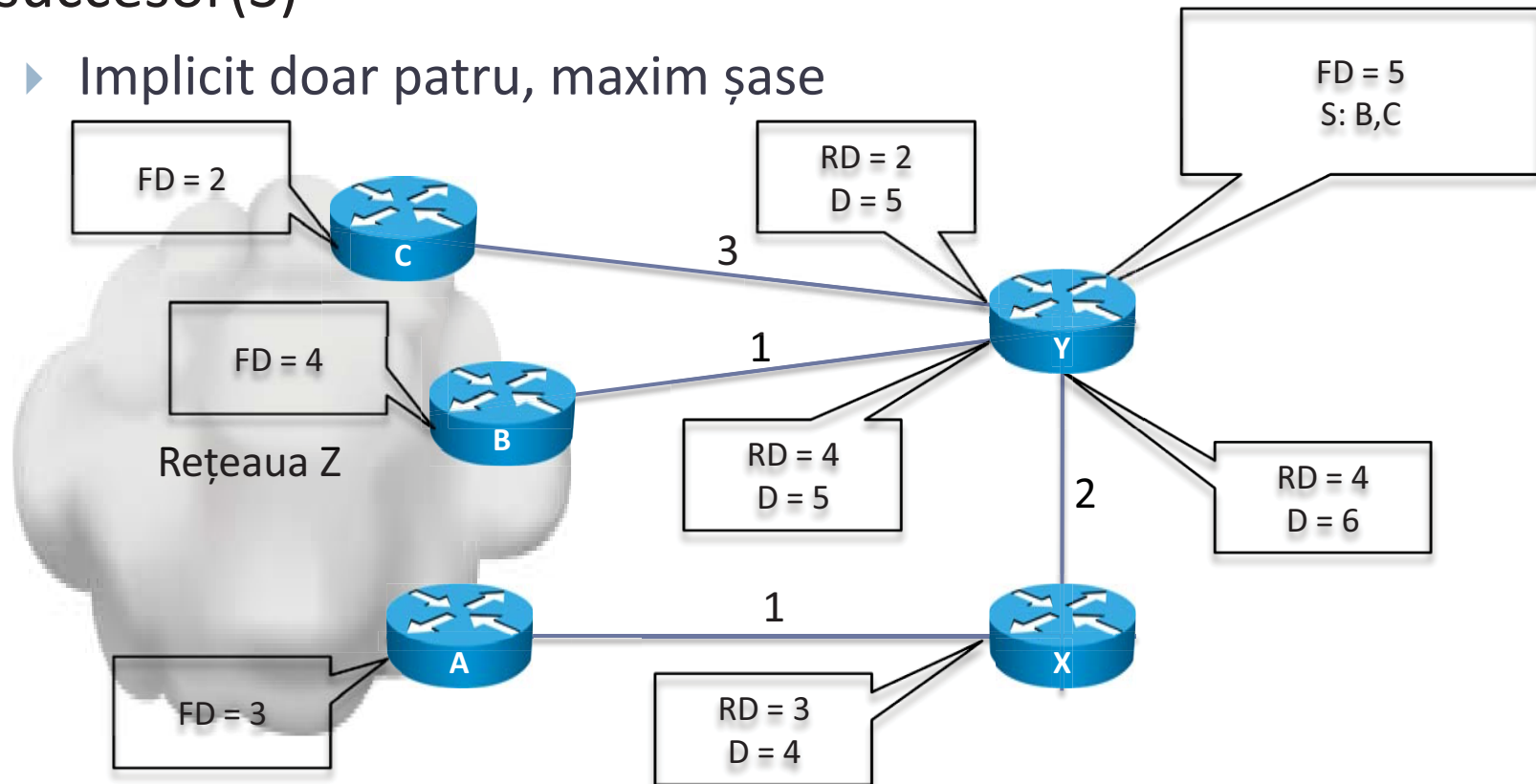
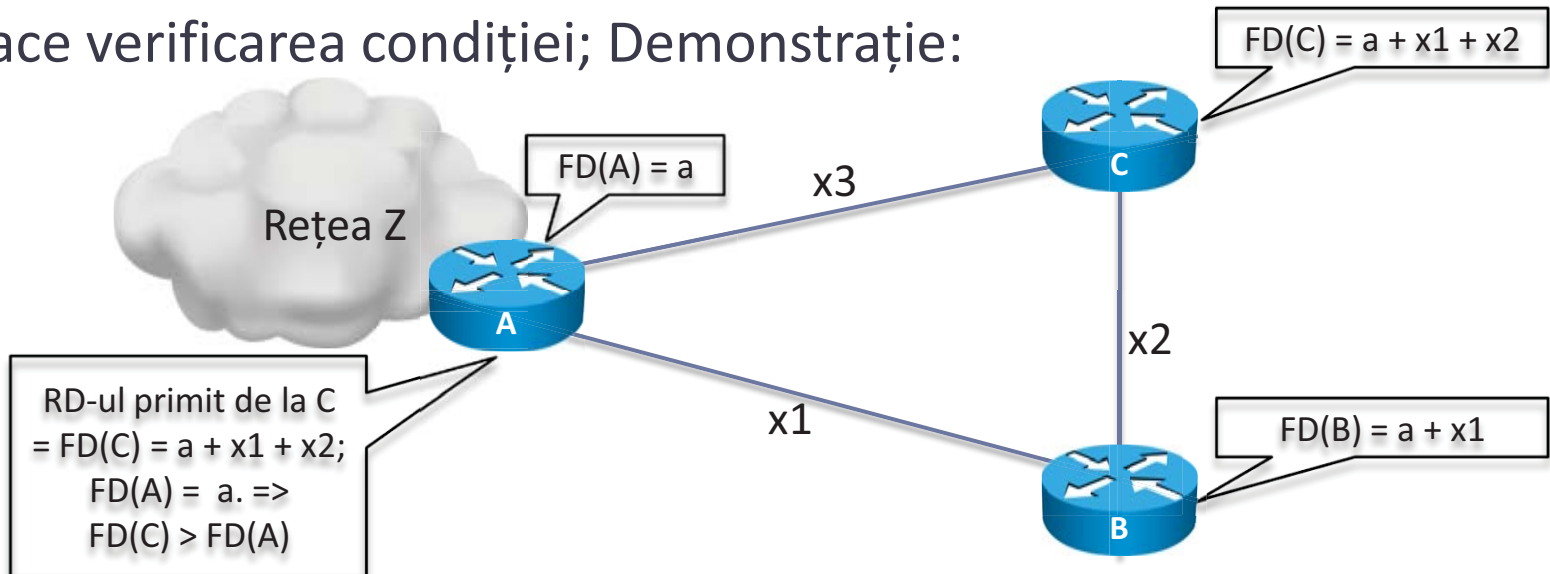


Tabela de topologie

► Condiția pentru ca un ruter să fie acceptat ca alternativă către o anumită destinație este: $RD < FD$

- Poartă numele de “feasible condition (FC)”
- Această condiție nu permite bucle prin echipamentul care face verificarea condiției; Demonstrație:



► Un router ce nu este S și trece condiția FC poartă numele de “feasible successor (FS)”

Tabela de topologie

- ▶ Există posibilitatea să nu existe succesori viabili
 - ▶ Fie datorită inexistenței redundanței fie datorită configurării incorecte a rețelei; Demonstrație:

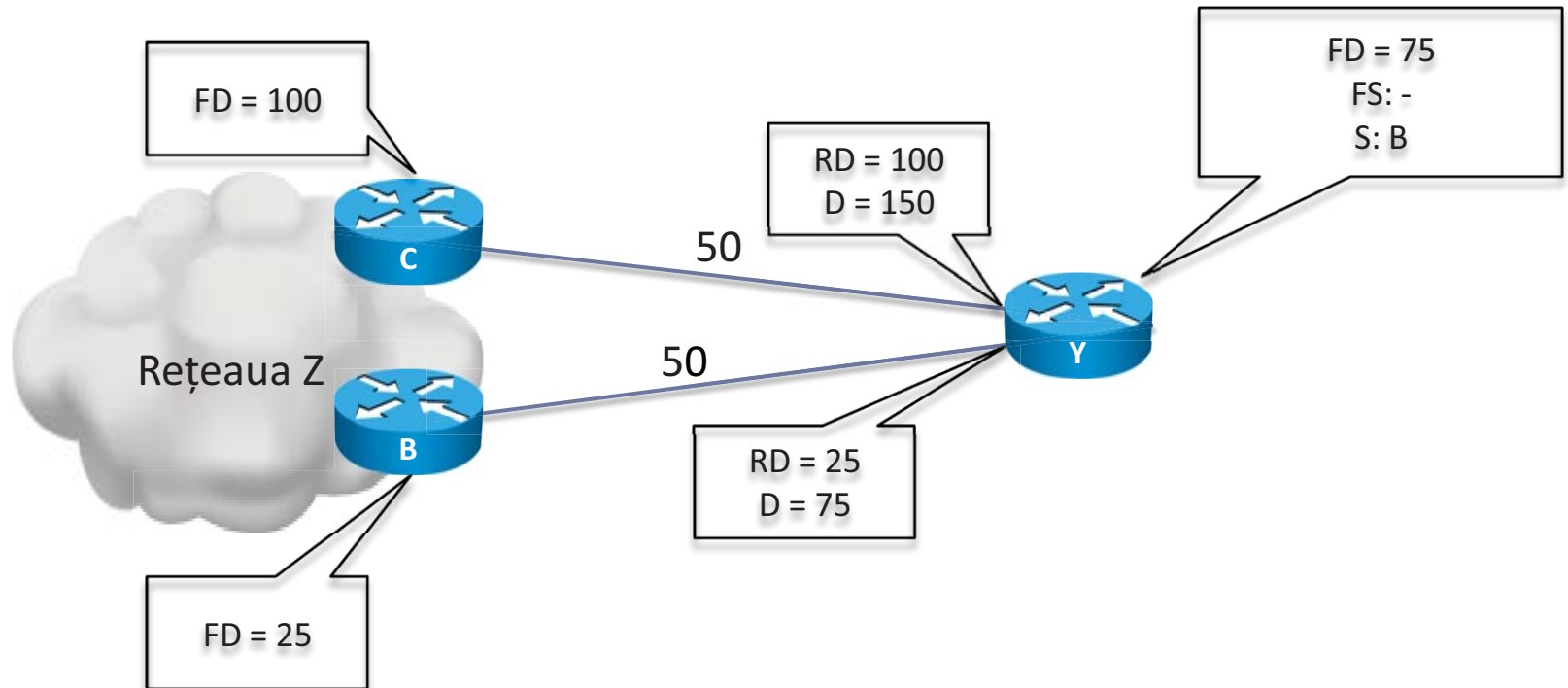


Tabela de topologie

- ▶ Pentru vizualizarea tabelii de topologie se folosește:

```
router#show ip eigrp topology  
router#show ip eigrp topology all-link
```

- ▶ Rutele din tabela de topologie pot fi în două stadii:
 - ▶ P Passive – ruta este bună și funcționează normal
 - ▶ A Active – ruta este în procedeu de recalculare DUAL
-

Tabela de topologie

► Procesul de recalculare DUAL

Dacă o conexiune pică: DUAL caută în tabela de topologie o rută alternativă



Daca nu se găsește rută alternativă, ruta este marcată ca activă (Active)



Sunt trimise pachete Query către toți vecinii - se cer informații despre topologie



Toate ruterele vecine trebuie să trimită un pachet Reply ca răspuns la pachetul Query



Se recalculează topologia (S și FS)

Tabela de topologie

- ▶ Atunci când se începe procesul de recalculare a unei rute se pornește un cronometru, implicit trei minute
- ▶ Dacă un vecin nu răspunde la QUERY până la expirarea timpului, acesta va fi scos din tabela de vecini
 - ▶ SIA - Stuck in active
- ▶ O rută ce se află în starea ACTIVE nu este folosită

```
NewYork#sh ip eigrp topology
  IP-EIGRP Topology Table for process 10

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply, r - Reply status
[...]
A 172.16.50.0/24, 0 successors, FD is 2195456, Q
  1 replies, active 00:00:06, query-origin: Local origin
Remaining replies: via 172.16.251.2, r, Serial1
```

Tabela de topologie

- ▶ Dezactivarea auto-sumarizării:

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

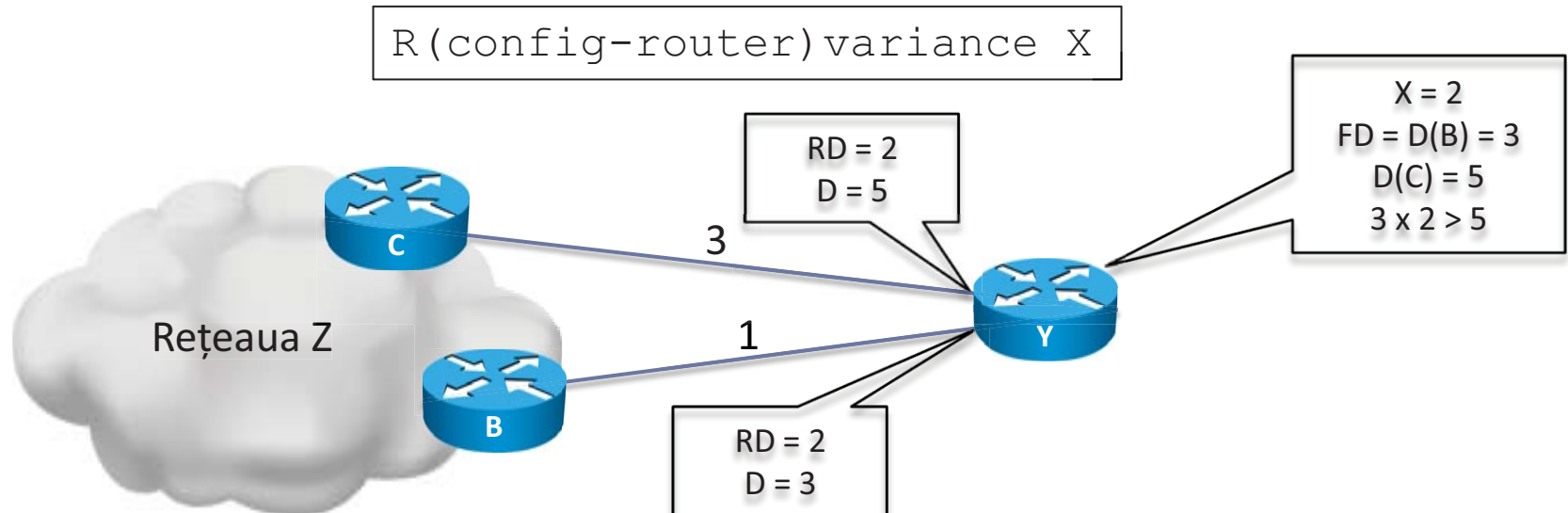
- ▶ Configurarea sumarizării manuale:

```
R(config-if)#ip summary-address eigrp <AS> <summ-address> <netmask> [<AD>]
```

- ▶ Distanța administrativă pentru rutele EIGRP sumarizate este 170.
 - ▶ În tabela de rutare locală se instalează o rută către Null0 cu distanța administrativă 5.
-

Tabela de rutare

- ▶ Sunt permise atât rute de cost egal cât și rute de cost inegal
 - ▶ Implicit doar cele cu cost egal
- ▶ Pentru a permite balansarea traficului pe rute de cost inegal se definește un factor de multiplicare raportat la FD



Multumesc!