



Rutarea; Protocolul RIPv1 și RIPv2

18 octombrie 2012

Proiectarea Rețelelor - 2012

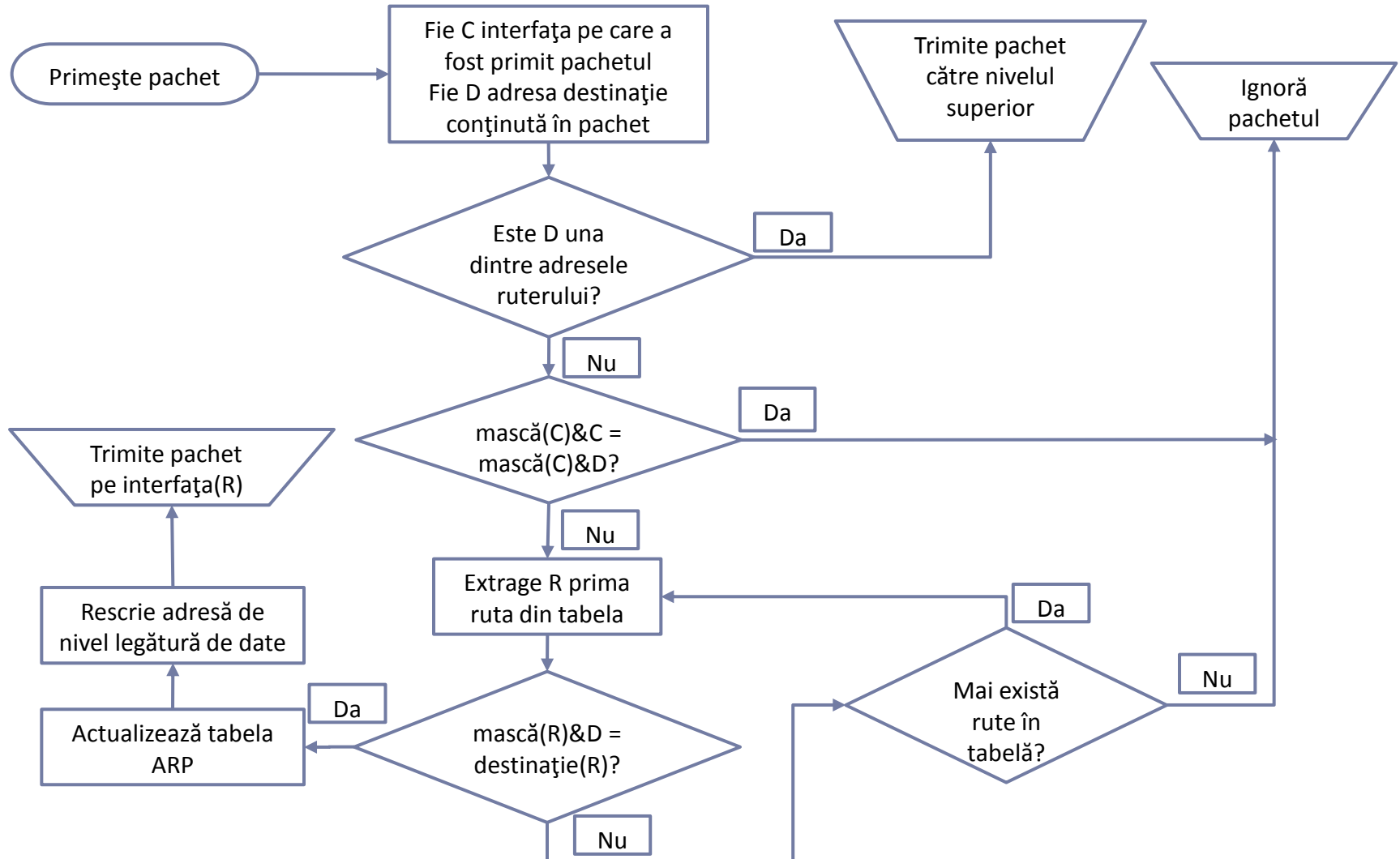
Overview

- ▶ Rutarea
 - ▶ Rutare statică
 - ▶ Rutare dinamică
- ▶ Protocoale de rutare Distance Vector
 - ▶ Generalități
 - ▶ Probleme: bucle de rutare
 - ▶ Soluții pentru acestea
- ▶ Protocolul RIPv1 și RIPv2
 - ▶ Generalități
 - ▶ Configurare
 - ▶ Verificare

Protocoale de rutare și rutate

- ▶ Protocoale de rutare: permit ruterelor să facă schimb de informații pe baza cărora fiecare își actualizează tabela de rutare
 - ▶ Exemple: RIP, IGRP, EIGRP, OSPF
- ▶ Protocoale rutate: permit identificarea nodurilor din rețea pe baza unei scheme de adresare menită să ofere unicitate, dar și ierarhizarea spațiului de adrese
 - ▶ Exemple: IP, IPX, AppleTalk

Funcționarea unui ruter



Procesul de rutare

Acest proces este alcătuit din două mecanisme:

- ▶ **Determinarea căii optime:** este folosită tabela de rutare;
- ▶ **Comutarea pachetelor (forwarding):** primirea unui pachet pe o interfață și trimitere lui pe alta.

Ruterele creează tabele de rutare

O rută conține:

- ▶ spațiul de adrese destinație (adresă de rețea și masca asociată)
- ▶ adresa următorului hop sau/și interfața de ieșire

Trei feluri de rute:

- ▶ Rețele direct conectate;
- ▶ Rute statice;
- ▶ Rute dinamice.

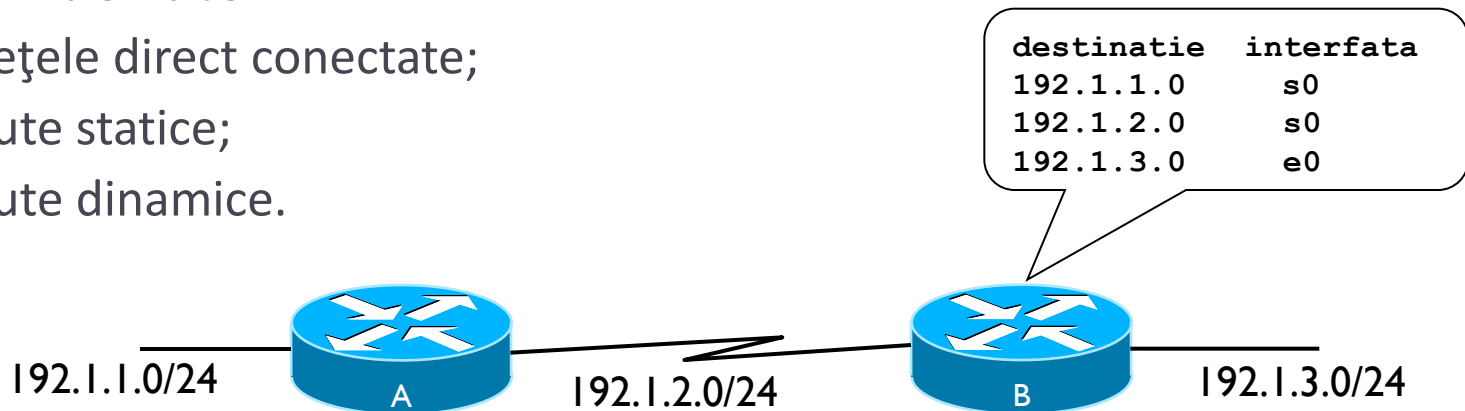
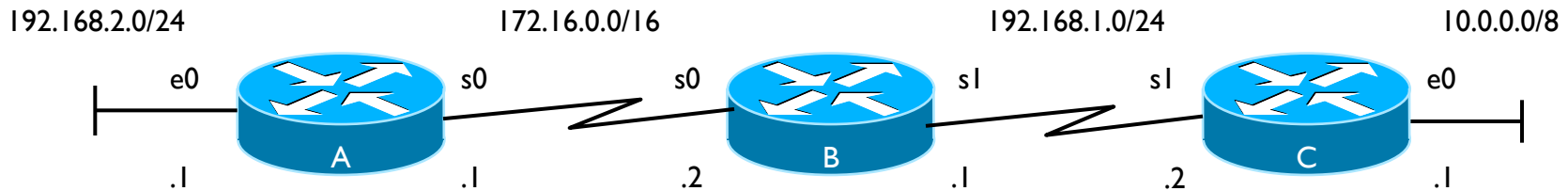


Tabela de rutare



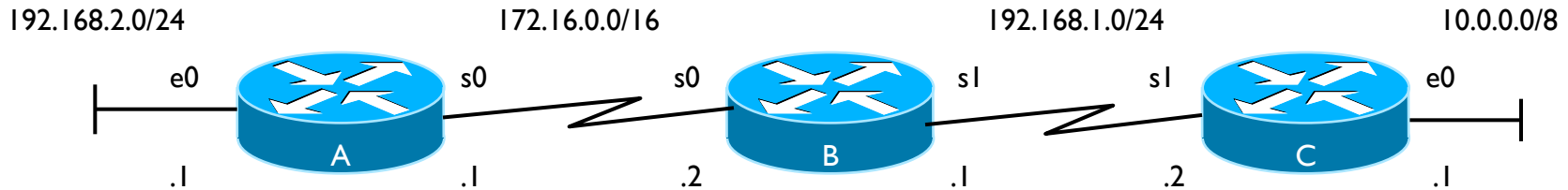
- ▶ Există două modalități de evaluare a unei rute:
 - ▶ Distanță administrativă
 - ▶ Metrică

```
A# show ip route
```

```
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
       default
       U - per-user static route, o - ODR
```

```
R      192.168.1.0/24 [120/1] via 172.16.0.2, 00:00:24, serial 0/0
```

Rețele direct conectate



Implicit, ruterele cunosc doar rețelele lor direct conectate: nu au nici un protocol de rutare configurat.

```

RTA# show ip route
Codes: C - connected,.. <Other codes and gateway information omitted>
C    172.16.0.0/16 is directly connected, Serial0
C    192.168.2.0/24 is directly connected, Ethernet0

RTB# show ip route
Codes: C - connected,.. <Other codes and gateway information omitted>
C    172.16.0.0/16 is directly connected, Serial0
C    192.168.1.0/24 is directly connected, Serial1

RTC# show ip route
Codes: C - connected,.. <Other codes and gateway information omitted>
C    10.0.0.0/8 is directly connected, Ethernet0
C    192.168.1.0/24 is directly connected, Serial1
  
```

Rute statice

```
Router(config)# ip route prefix mask {addr | interf} [dist]
```



```
ip route 192.168.2.0 255.255.255.0 192.168.1.1 E0
```

- ▶ Rutele statice care specifică următorul hop au distanță administrativă 1
- ▶ Rutele statice ce specifică interfața de ieșire au distanța administrativă 0
- ▶ Nu este necesară configurarea de rute statice pentru rețele direct conectate
- ▶ În cazul rutelor statice via legături punct la punct este indicată specificarea numai a interfeței de ieșire, deoarece adresa următorului hop nu este folosită în acest caz
- ▶ În cazul rutelor statice via rețele de multiacces (Ethernet) este indicată specificarea următorului hop, doar interfața de ieșire nefiind suficient

Ruta default

- ▶ Poate fi adăugată ca o rută statică:

```
R(config)#ip route 0.0.0.0 0.0.0.0 {address|interface}
```

- ▶ sau poate proveni dintr-un pachet de actualizare:

```
A# show ip route
```

```
< ... >
```

```
R * 0.0.0.0/0 [120/1] via 192.168.2.2, 00:00:69, serial0/0
```

- ▶ Toate pachetele pentru care nu există o rută explicită vor fi trimise pe aceasta rută
- ▶ Mai poartă denumirea de “quad zero route”

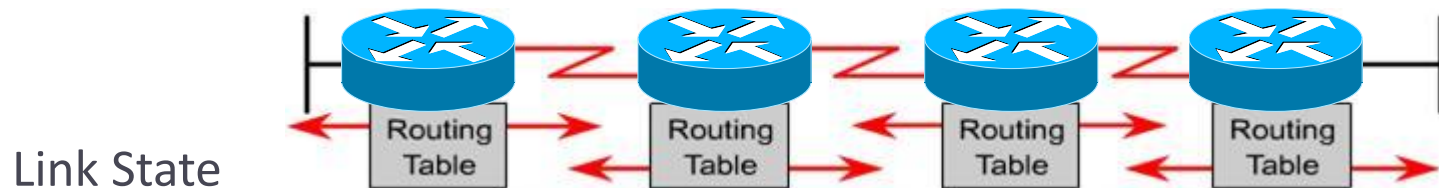
Distanța administrativă

Tipul de rută	Distanța administrativă
Direct conectată	0
Rută statică	1
Sumarizare EIGRP	5
BGP extern	20
EIGRP	90
OSPF	110
IS-IS	115
RIP	120
EIGRP extern	170
BGP intern	200
necunoscută	255

- ▶ Distanța administrativă este o metodă de a cuantifica încrederea avută în diferitele metode de învățare a rutelor.
- ▶ Dacă ruterul află două rute către aceeași destinație, va pune în tabela de rutare pe cea cu DA mai bună.

Rutare dinamică

- ▶ Rutarea dinamică se bazează pe folosirea unui protocol de rutare
- ▶ Există două clase de protocoale:
 - ▶ Distance Vector
 - ▶ Algoritmul de rutare transmite tuturor rutelor vecine o copie a tabelului de rutare
 - ▶ Ruterele vecine își actualizează tabelul de rutare în funcție de informațiile primite, apoi generează un nou pachet de actualizare



- ▶ Pachetele de actualizare ajung în toată rețeaua (nu doar la vecini)
- ▶ tabela de rutare - tabela celor mai bune căi
- ▶ tabela de topologie - colecția tuturor rutelor
- ▶ tabela de adiacență - lista tuturor vecinilor

Comparatie DV-LS

▶ DV

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

▶ LS

- ▶ Transmit informații în întreaga rețea (porțiuni din tabela de rutare)
- ▶ Imagine de ansamblu a rețelei
- ▶ Update-uri determinate de schimbări în topologie
- ▶ Cerințe mai mari de hardware și lățime de bandă
- ▶ Mai puțin predispuse la bucle de rutare
- ▶ Convergență rapidă



Protoziale Distance Vector

Protocoale DV - Generalități

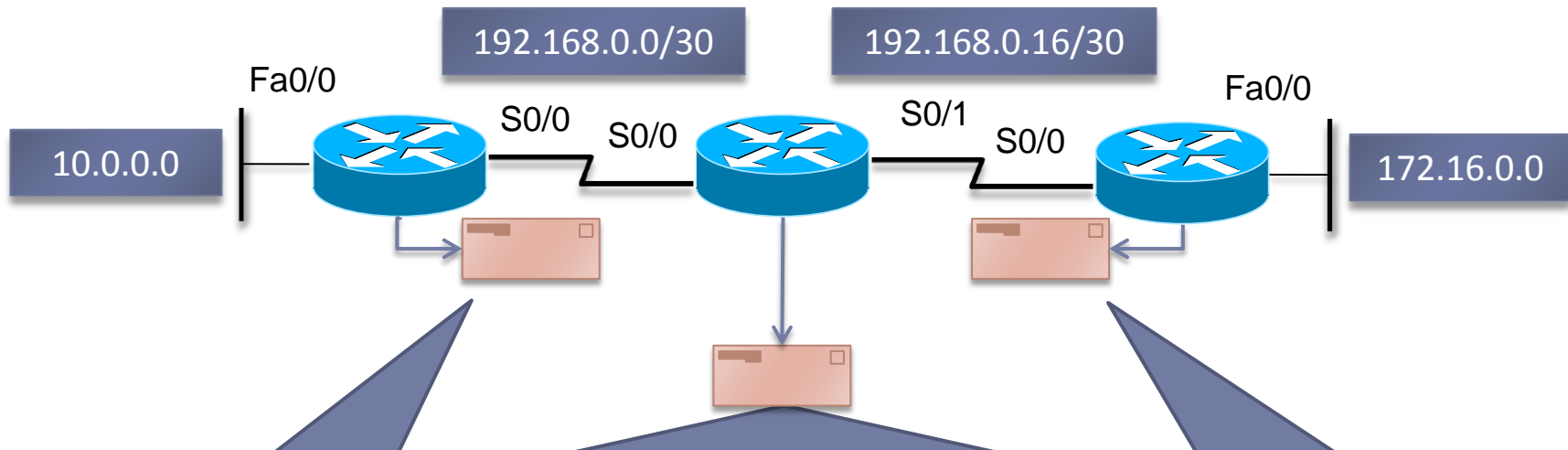
▶ DV

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



Protocele DV – funcționare

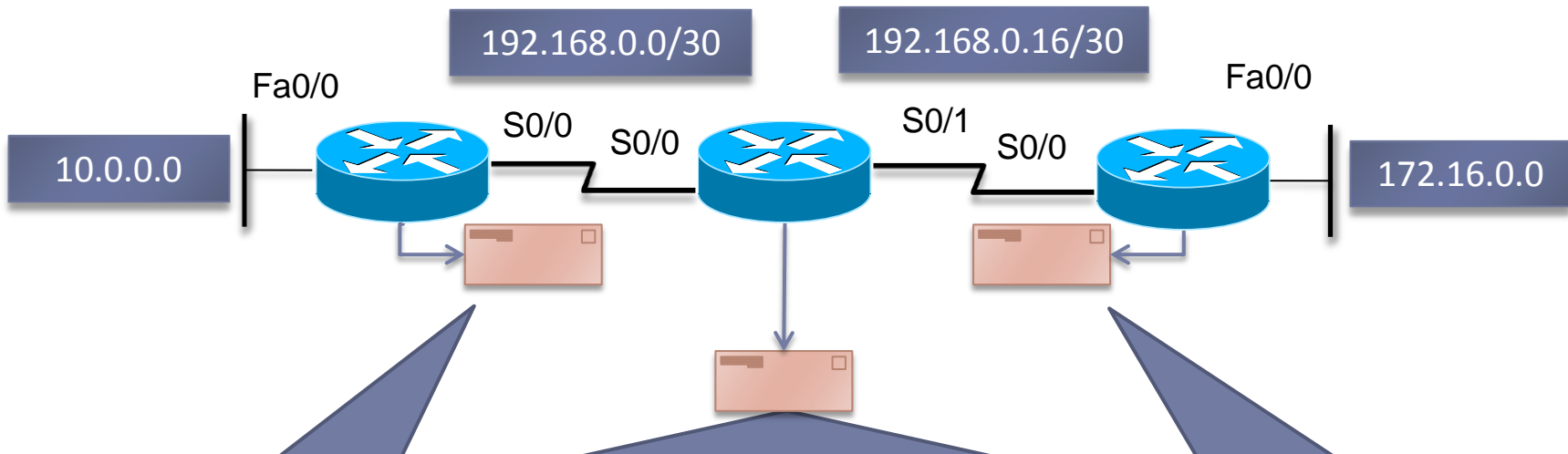
- ▶ Orice ruter își cunoaște doar vecinii direct conectați
- ▶ Se trimite întreaga tabelă de rutare vecinilor



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

Protocele DV – funcționare

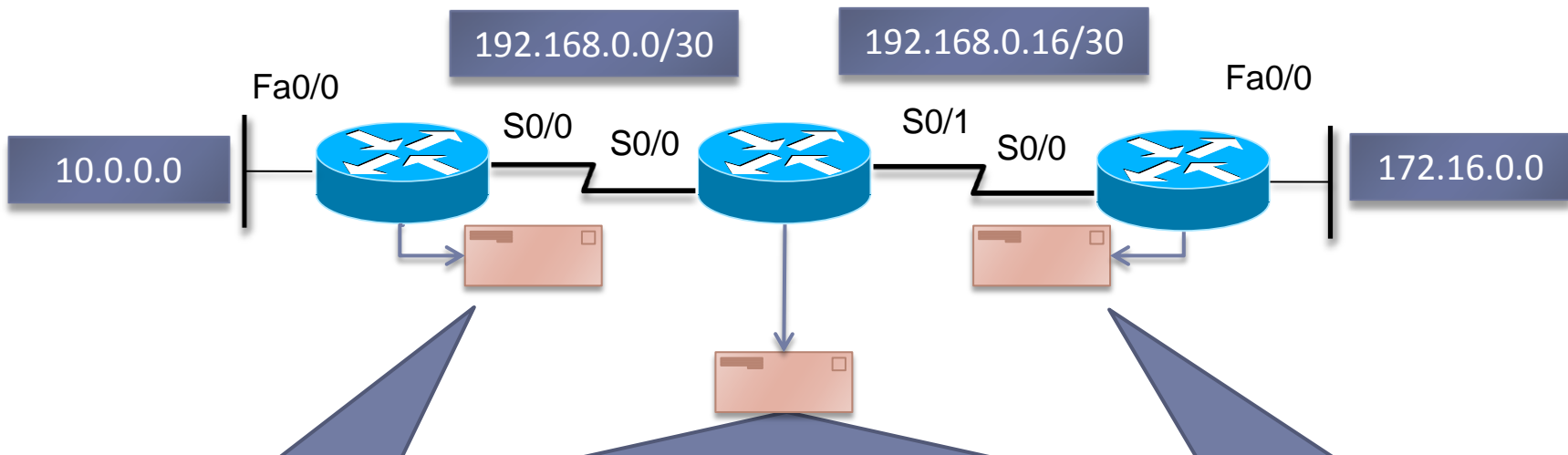
► Tabela de rutare după primul update



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

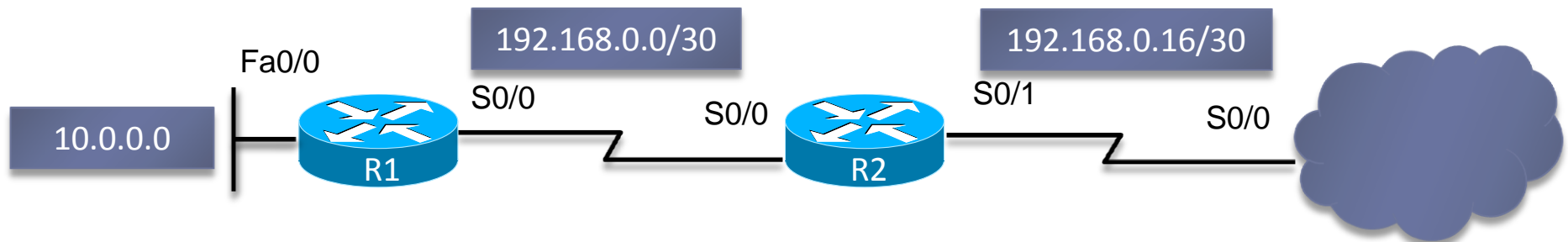
- ▶ Tabela de rutare după al doilea update
- ▶ Rețeaua a convers



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

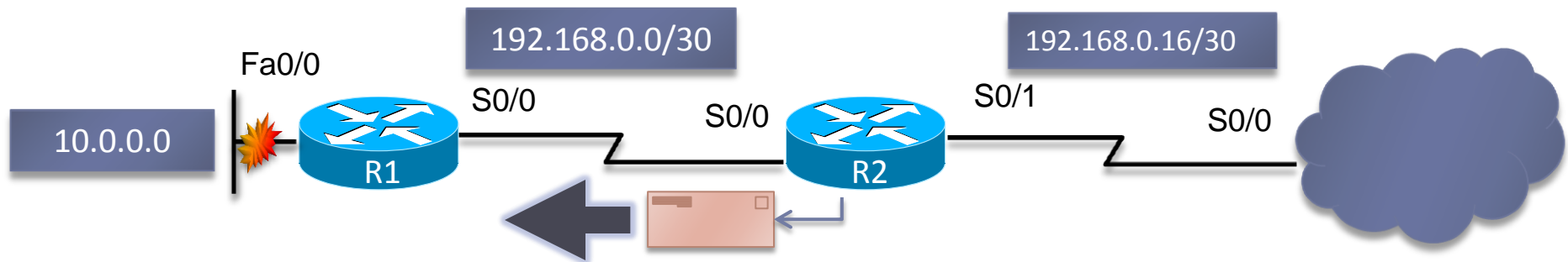


Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

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

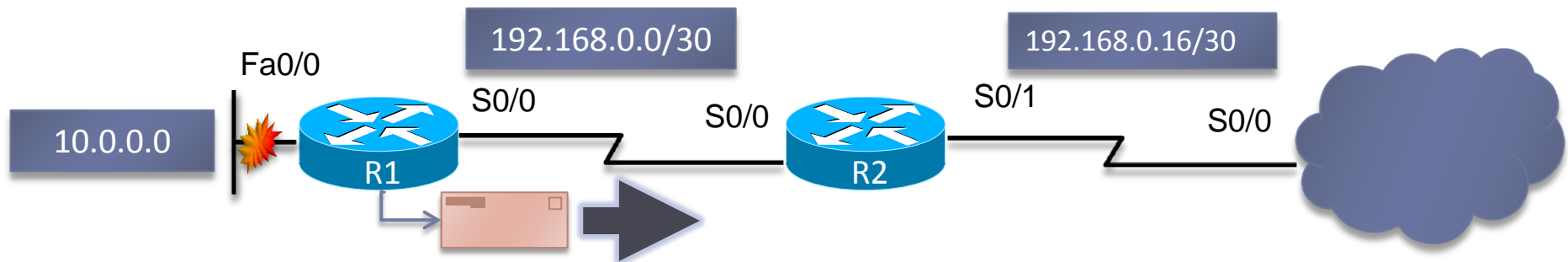


Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

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



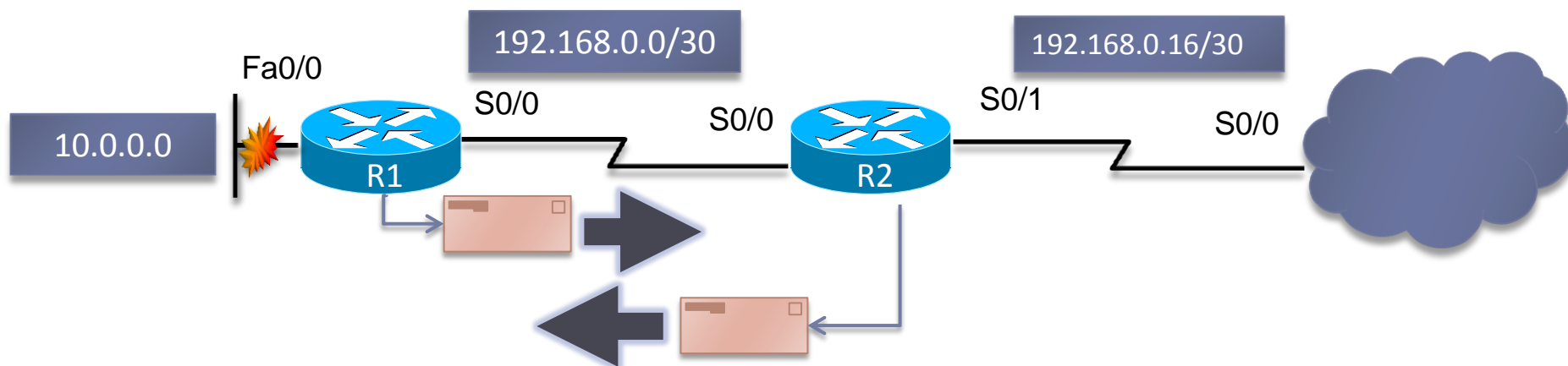
Rețea	Interfață	Număr de hopuri
10.0.0.0	S0/0	2
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

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

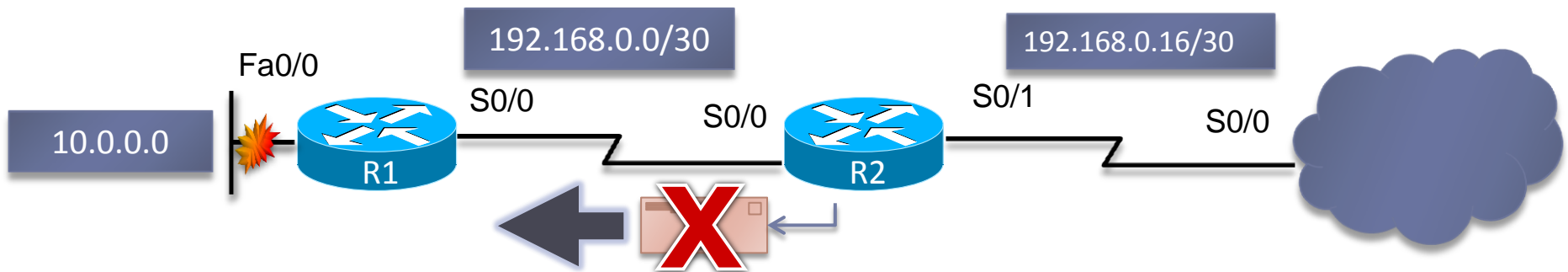
Max hop count



- ▶ 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

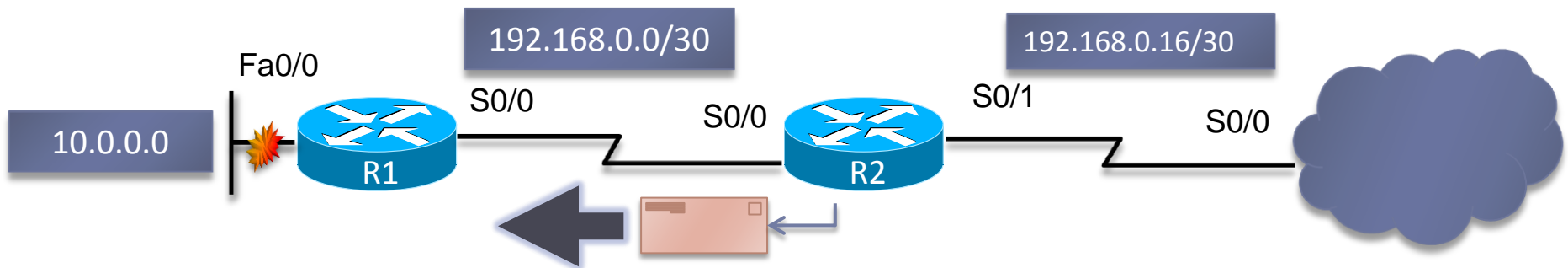


Rețea	Interfață	Număr de hopuri
10.0.0.0	Fa0/0	0
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

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”

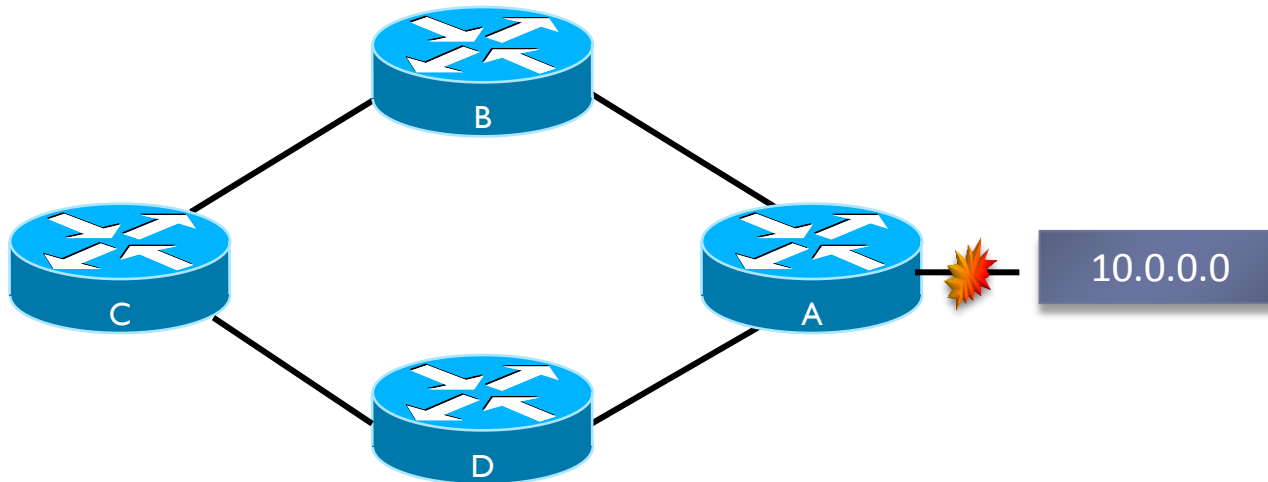


Rețea	Interfață	Număr de hopuri
10.0.0.0	S0/0	16
192.168.0.0	S0/0	0
192.168.0.16	S0/0	1

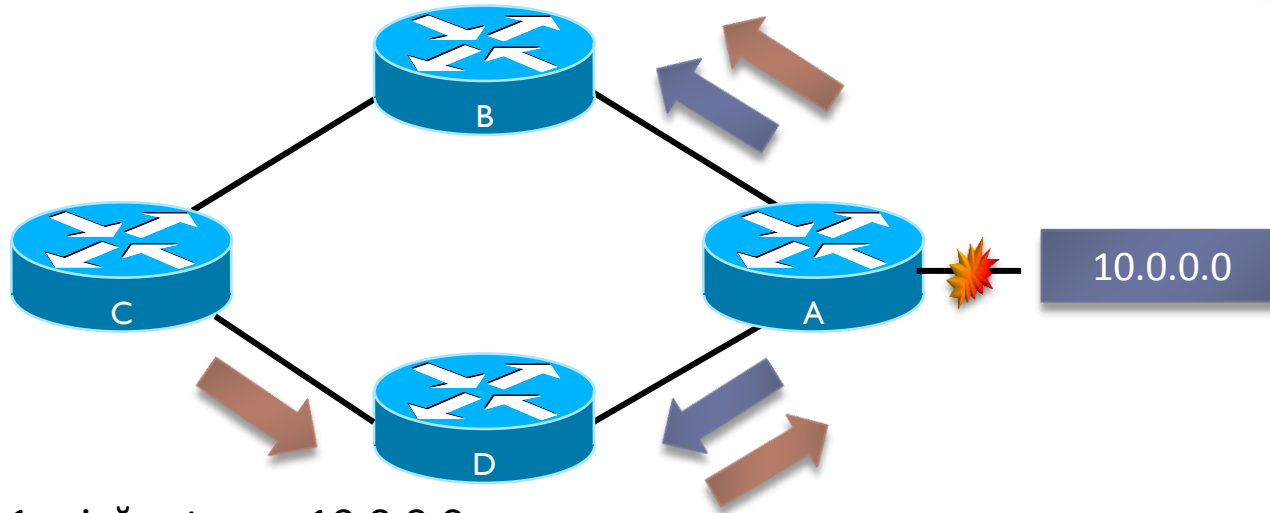
Rețea	Interfață	Număr de hopuri
192.168.0.0	S0/0	0
192.168.0.16	S0/1	0
10.0.0.0	S0/0	1

Este split-horizon îndeajuns?

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



Este split-horizon îndeajuns?

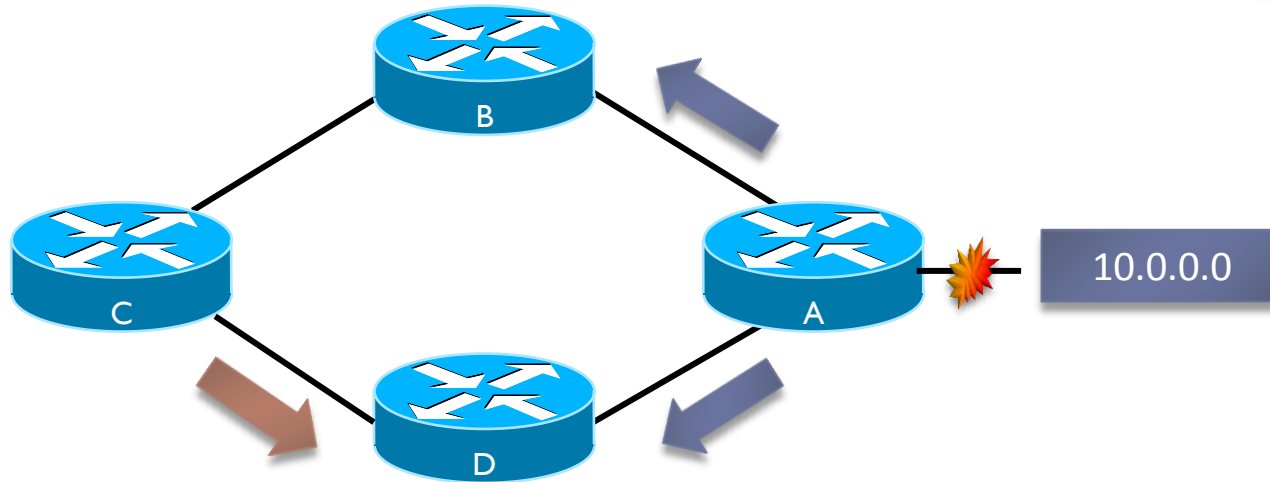


- ▶ Pasul 1: pică rețeaua 10.0.0.0
- ▶ Pasul 2: A trimite update-uri către B și D informându-i de acest aspect
- ▶ Pasul 3: C trimite un update către D spunând că el cunoaște rețeaua 10.0.0.0 cu o metrică de 3.
- ▶ Q: În trecut D i-a trimis ruta 10.0.0.0 lui C. De ce pasul 3 nu încalcă regula split-horizon?
- ▶ 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 ($3 > 1$)

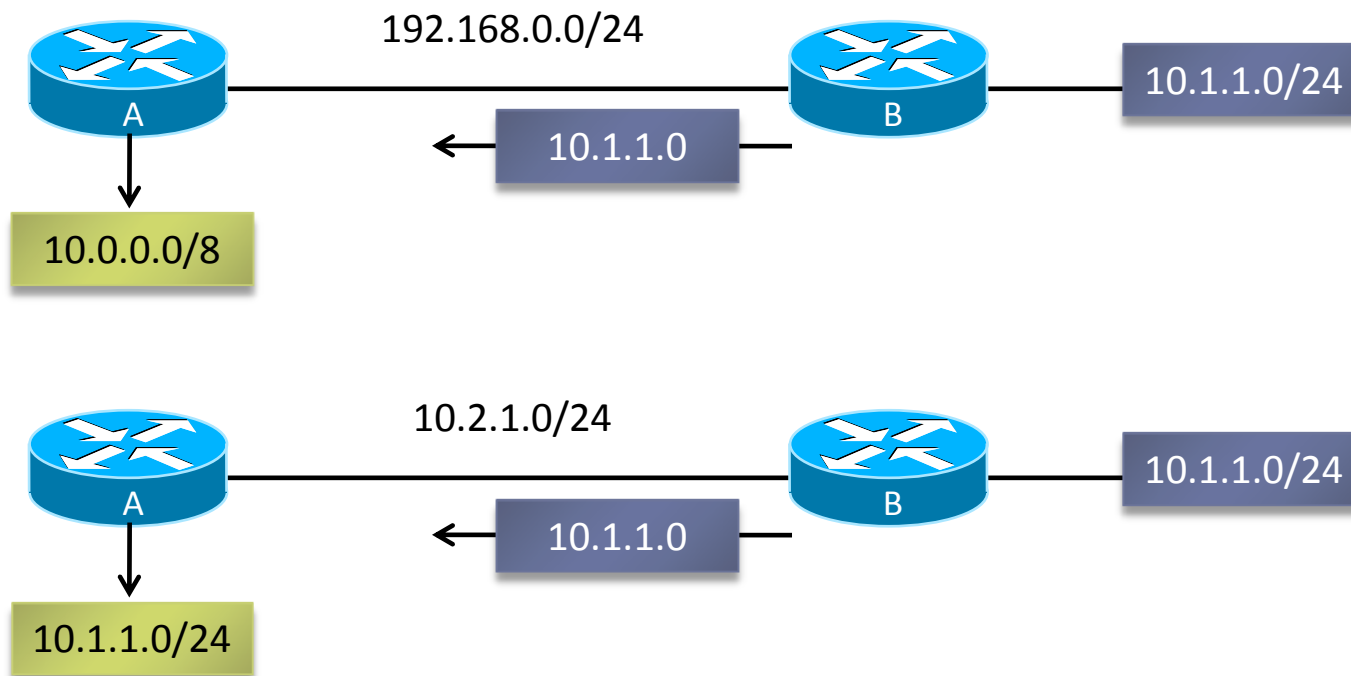


Protocolul RIP

- ▶ 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

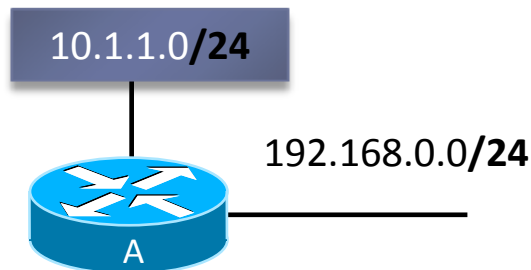
- ▶ Protocol de rutare classful
 - ▶ Nu trimite masca de rețea în update-uri
- ▶ Comportament RIPv1 la primirea update-urilor



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

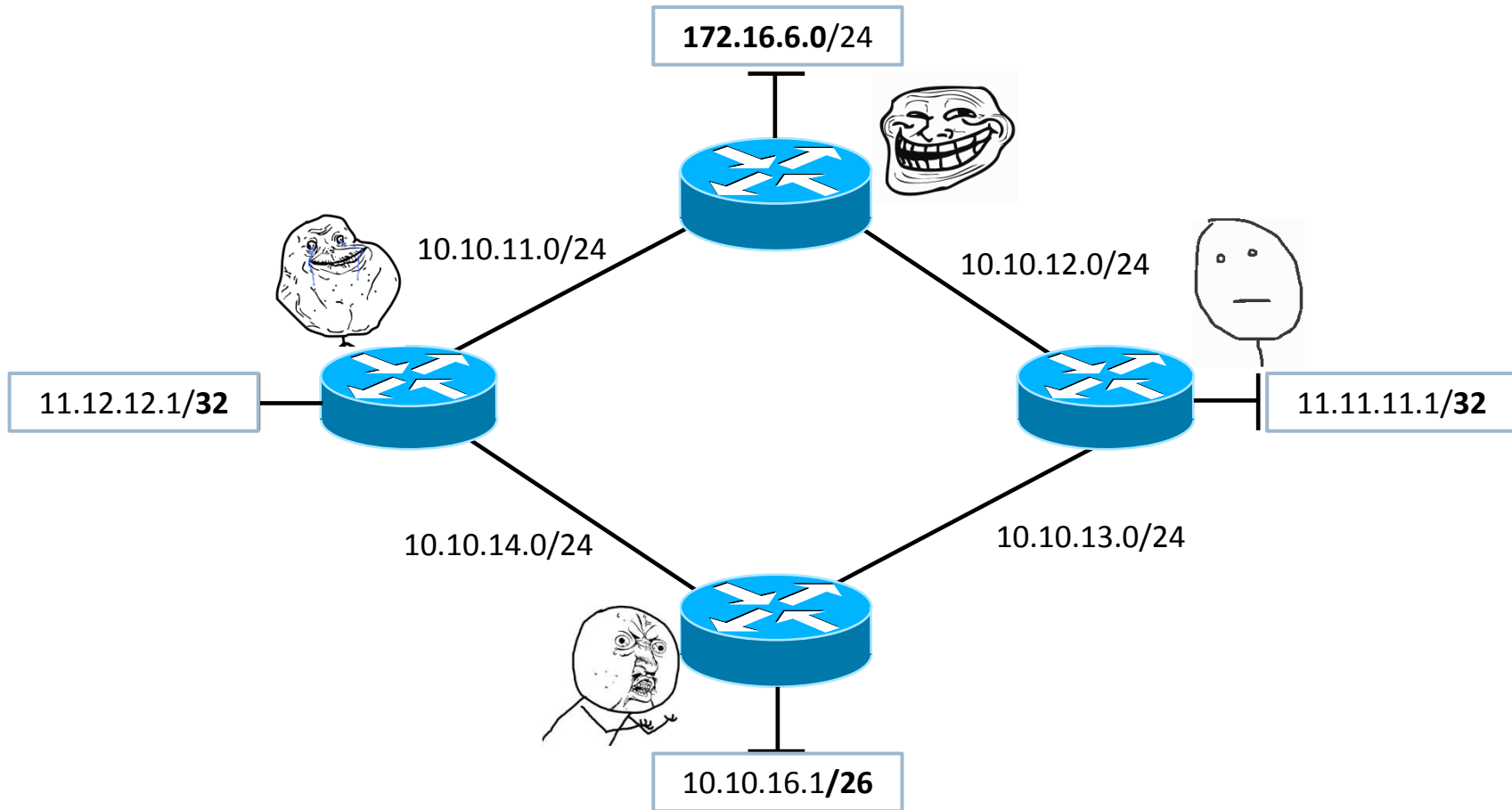
```
Batman#sh ip route
```

```
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
```

POC – RIPv1



POC – RIPv1

- ▶ Activați RIPv1 pe ruterul Troll pentru rețelele direct conectate și verificați activarea corectă a protocolului
- ▶ Activați RIPv1 pe toate ruterele pentru rețelele direct conectate
- ▶ Verificați tabela de rutare pe ruterul ForeverAlone. Ce observați legat de rutele de loopback?
- ▶ Configurați rețeaua astfel încât nici un ruter să nu mai cunoască rețeaua 172.16.0.0/16 fără a scoate această rețea din comenzile network

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`

- ▶ 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

Pachetul RIPv1 vs RIPv2

Rută RIPv1	Command (1 sau 2)	Version = 1	ZERO
	Address family identifier (2 pentru IP)		ZERO
	Adresa IP (adresa de rețea)		
	ZERO		
	ZERO		
	Metrica (nr de hopuri)		
	Rute multiple, până la 25 într-un singur update		

Rută RIPv2	Command (1 sau 2)	Version = 2	ZERO
	Address family identifier (2 pentru IP)		Route Tag (poate indica rutele externe)
	Adresa IP (adresa de rețea)		
	Mască de rețea		
	Suport pentru autentificare		
	Metrica (nr de hopuri)		
	Rute multiple, până la 25 într-un singur update, 23 cu autentificare		

Activarea RIPv2

- ▶ Implicit, rourele 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

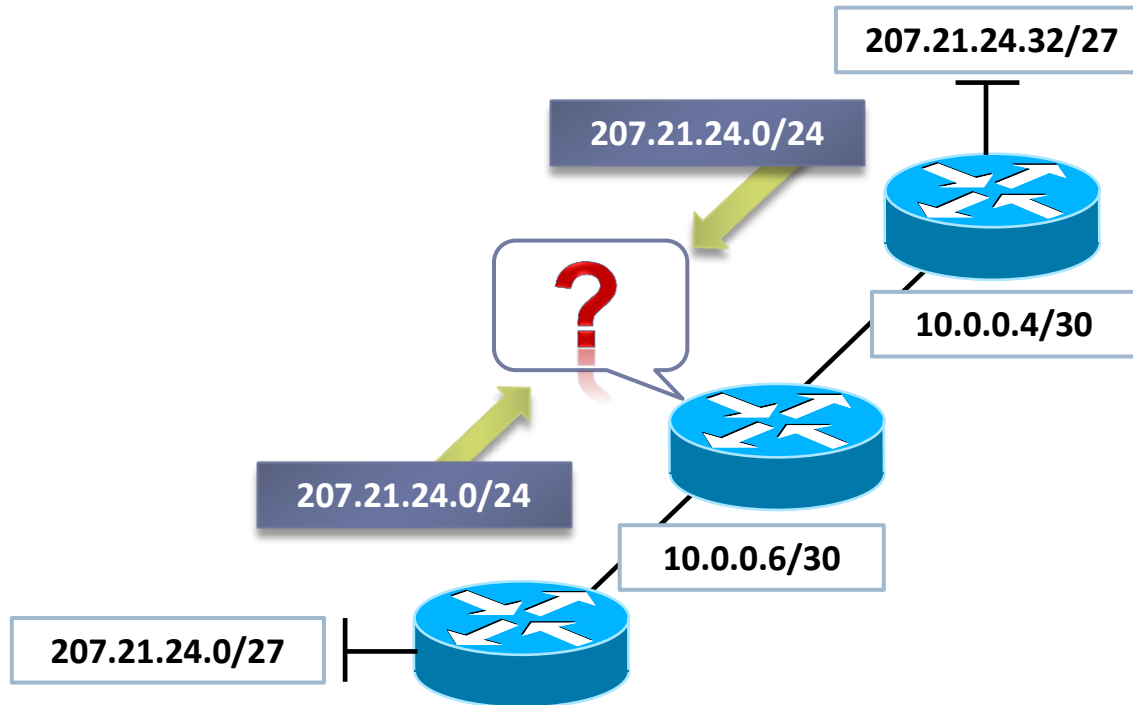
```
R1(config-router)#version ?
<1-2>  version
R1(config-router)#version 2
```

Auto-sumarizarea

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

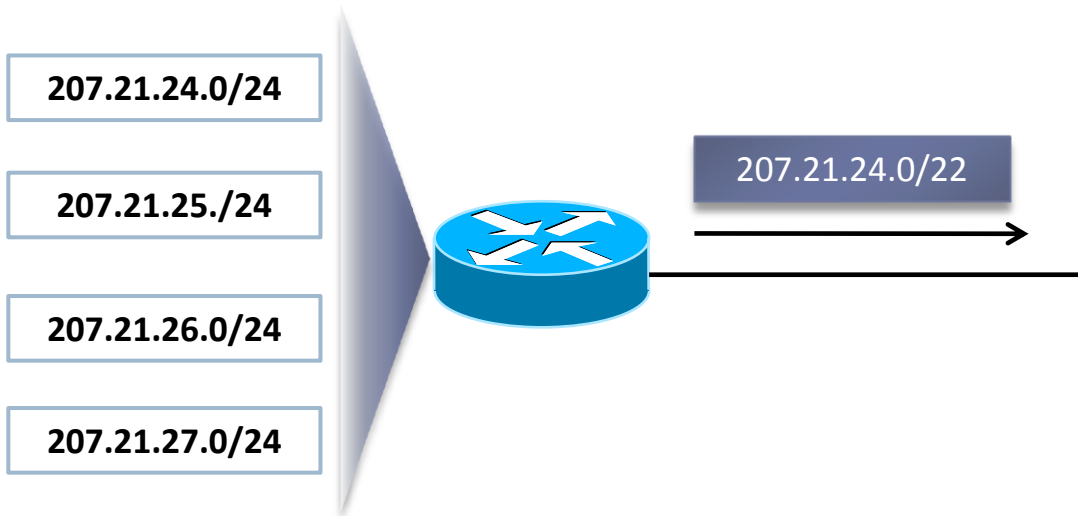
```
R1(config-router)#do show ip protocols
Routing Protocol is "rip"
[...]
  Default version control: send version 2, receive version 2
  Interface                Send  Recv  Triggered RIP  Key-chain
  Serial0/0/0              2     2
  Serial0/0/1              2     2
Automatic network summarization 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
```

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
```


Verificarea RIPv2

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

Troubleshooting RIP v2



debug ip rip - displays RIP routing updates sent and received

no debug all - turns off all debugging

undebug all



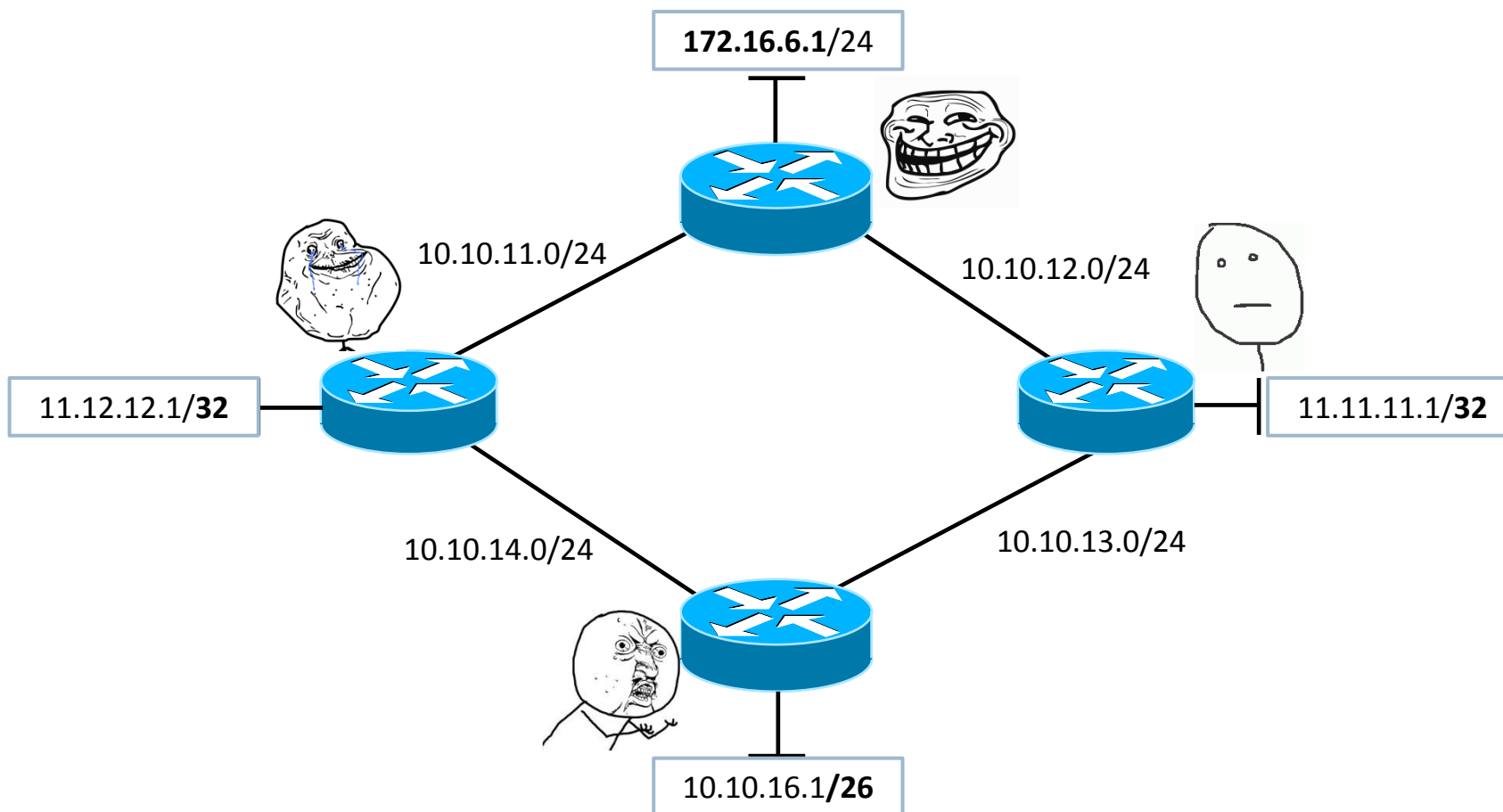
Erori frecvente

- ▶ Conflict de versiuni
 - ▶ routerele 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

Overview

- ▶ Rutarea
 - ▶ Rutare statică
 - ▶ Rutare dinamică
- ▶ Protocoale de rutare Distance Vector
 - ▶ Generalități
 - ▶ Probleme: bucle de rutare
 - ▶ Soluții pentru acestea
- ▶ Protocolul RIPv1 și RIPv2
 - ▶ Generalități
 - ▶ Configurare
 - ▶ Verificare

POC – RIPv2



POC – RIPv2

- ▶ Migrați întreaga infrastructură la RIPv2
- ▶ Analizați tabela de rutare a ruterelor. Ce diferență observați?
- ▶ Dezactivați sumarizarea automată și analizați din nou tabelele de rutare
- ▶ Publicați de pe ruterul Troll rețeaua 172.16.6.0/23 fără a schimba adresa de pe interfața de loopback
- ▶ Configurați autentificare între Troll și PokerFace