



# Managementul rețelelor

Proiectarea Rețelelor

# Cuprins

---

- ▶ Autentificare, Autorizare și Accounting
- ▶ Descoperirea rețelei
  - ▶ CDP
  - ▶ NBAR
- ▶ Monitorizarea rețelei
  - ▶ SNMP
  - ▶ NETFLOW
  - ▶ SMOKEPING



# Autentificare

---

- ▶ Tipuri de autentificare
  - ▶ Password – only
  - ▶ Local – database
  - ▶ Server – database

TACACS+	RADIUS
Cisco server version	Open Standard
TCP	UDP
Urmărește arhitectura AAA	Combină autentificarea cu autorizarea

- ▶ Autentificarea se poate face pe bază de utilizator și parolă sau folosind Kerberos 5

# Autorizare

- ▶ Implementată de obicei folosind un server de AAA
- ▶ Utilizatorul primește un set de atribute ce descrie nivelul său de acces în rețea

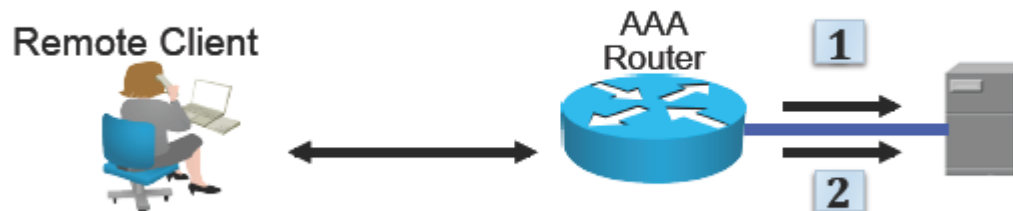


- ▶ Utilizatorul trimite o comandă către ruter
- ▶ Ruterul întreabă serverul dacă utilizatorul are dreptul să execute această comandă
- ▶ Serverul răspunde cu DA/NU

# Accounting

---

- ▶ Implementare folosind un server de AAA
- ▶ Menține evidența activităților individuale
- ▶ După autentificarea utilizatorului toate activitățile acestuia in rețea sunt salvate
- ▶ Foarte important pentru securitatea rețelei, dar și pentru rapoarte despre activitatea utilizatorilor



# Cisco Discovery Protocol

- ▶ protocol de nivel 2 proprietar Cisco
- ▶ folosit între două echipamente vecine pentru a anunța informații referitoare la:
  - ▶ platformă
  - ▶ sistemul de operare
  - ▶ adresa IP
  - ▶ interfețele direct conectate

```
R8# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID          Local Intrfce    Holdtme    Capability    Platform    Port ID
S1                 Fas 0/0         163        S I          WS-C2960-   Fas 0/4
R7                 Ser 0/2/1       131        R S I        2801        Ser 0/2/1
```

- ▶ Network Based Application Recognition
  - ▶ Recunoaște un număr mare de protocoale și poate fi extins prin folosirea de module (PDLM – Packet Description Language Modules)
  
- ▶ NBAR – protocol-discovery permite recunoașterea protocoalelor pentru o anumită interfață
  
- ▶ Folosirea lui poate duce la o utilizare excesivă a procesorului și a memoriei ruterului

## ► configurarea pe interfață

```
Aegis#config t
  Aegis(config)#interface FastEthernet0/0
  Aegis(config-if)#ip nbar protocol-discovery
```

## ► verificarea protocoalelor ce rulează

```
Aegis#sh ip nbar protocol-discovery
FastEthernet0/0      Input          Output
Protocol            Packet Count   Packet Count
                   Byte Count     Byte Count
                   5min Bit Rate (bps)   5min Bit Rate (bps)
                   5min Max Bit Rate (bps) 5min Max Bit Rate (bps)
-----
      ftp            617           606
                   792480        34749
                   34000         1000
                   34000         1000
      ospf           78            78
                   7356          7376
                   0             0
                   0             0
      Total          3898          4113
                   3045939        488008
                   59000         1000
                   78000         16000
```

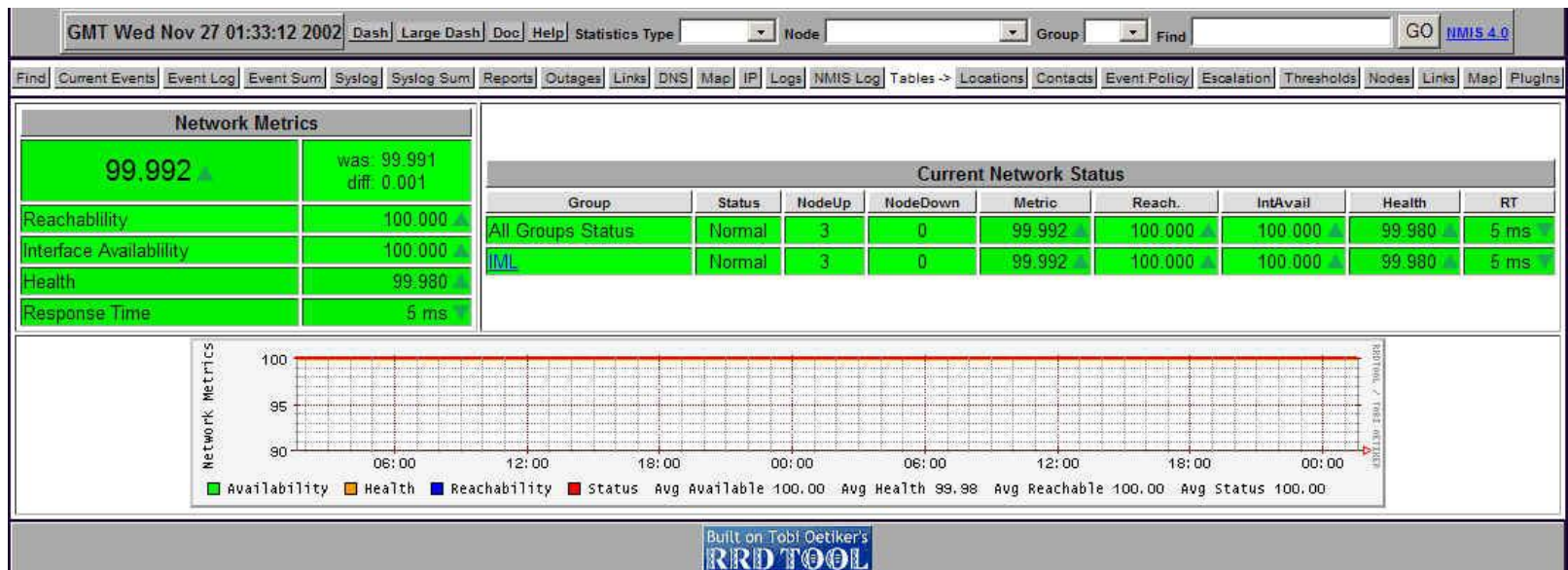


# SNMP

---

- ▶ Simple Network Management Protocol
- ▶ Protocol de Nivel Aplicație folosit pentru schimbarea de informații într-o rețea
- ▶ Componentele unei rețele ce folosește SNMP
  - ▶ Dispozitivul de monitorizat
  - ▶ Un software denumit agent instalat pe acest dispozitiv
  - ▶ O aplicație de monitorizare ce primește informații de la aceste dispozitive
- ▶ Prin SNMP se pot primi informații de la echipamente, existând și posibilitatea de trimitere de comenzi

- ▶ Network Management Information System
- ▶ Oferă informații despre disponibilitatea și încărcarea echipamentelor din rețea
- ▶ Folosește SNMP pentru colectarea datelor



# Netflow

---

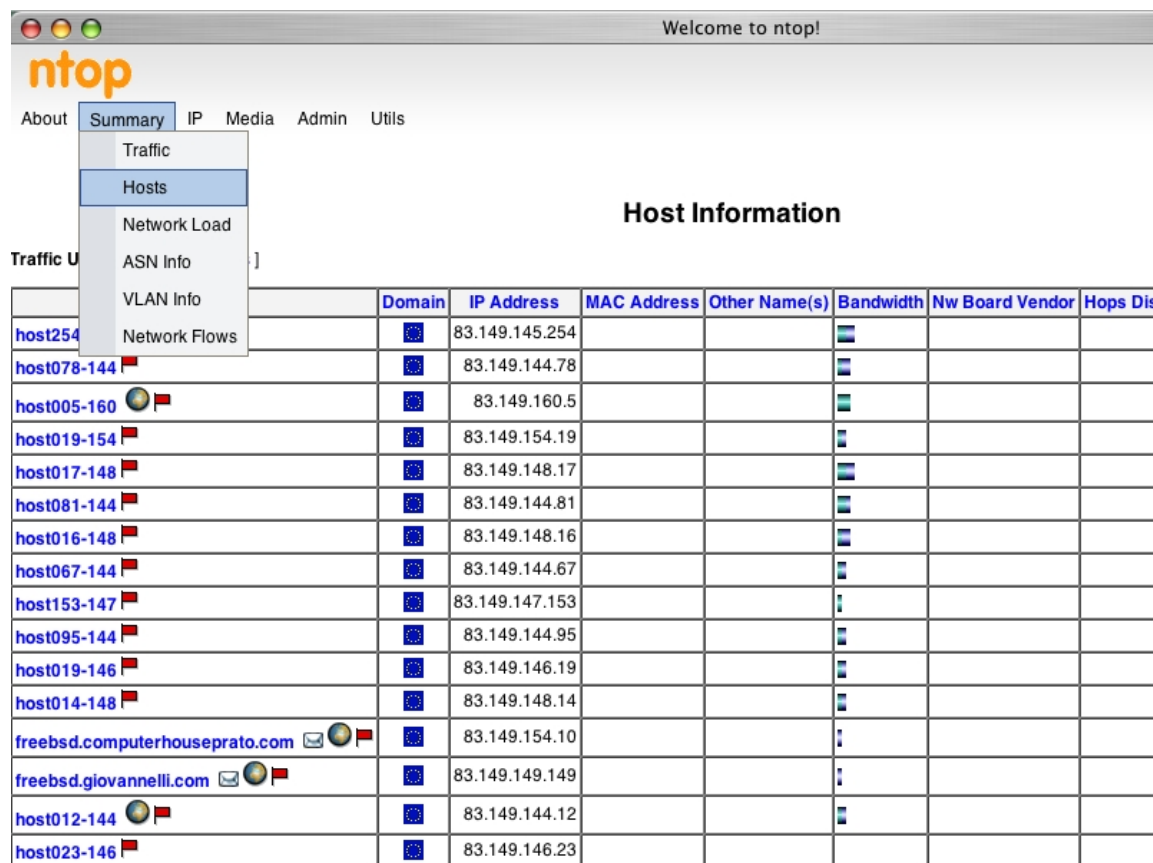
- ▶ Protocol implementat de Cisco pentru colectarea informațiilor despre trafic
- ▶ Arhitectura se bazează pe colectarea datelor de către un sistem separat, folosirea ruterului poate duce la suprasolicitarea acestuia
- ▶ Folosește mesaje sumarizate pentru transmiterea de informații referitoare la un anumit tip de trafic
- ▶ IPFIX este dezvoltat de IETF pentru îmbunătățirea și standardizarea protocolului

# Ntop

▶ Este o unealtă de monitorizare a traficului prin protocolul Netflow/IPFIX

▶ Poate identifica

- ▶ tipurile de trafic
- ▶ dispozitivele
- ▶ lățimea de bandă



Welcome to ntop!

ntop

About Summary IP Media Admin Utils

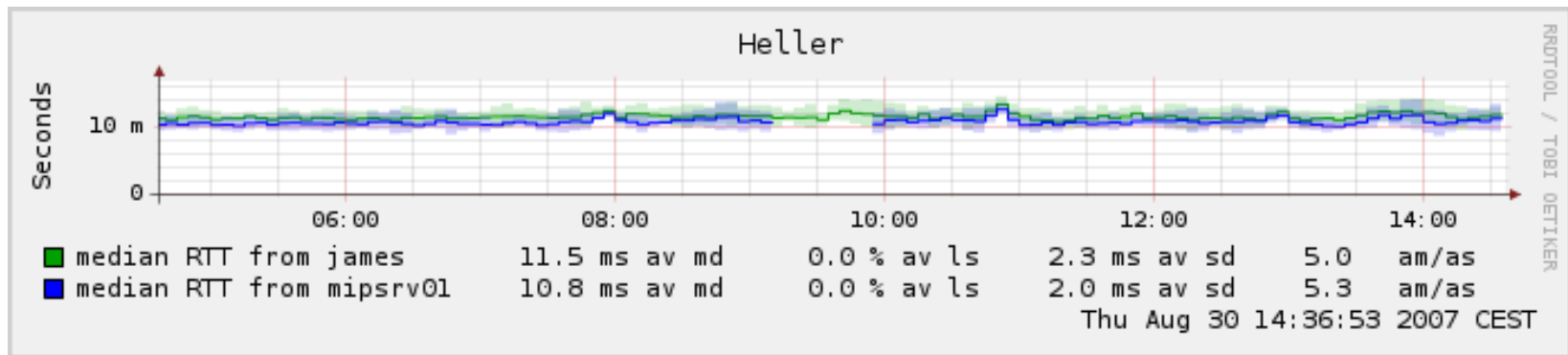
Traffic U

Host Information

	Domain	IP Address	MAC Address	Other Name(s)	Bandwidth	Nw Board	Vendor	Hops	Dis
host254		83.149.145.254							
host078-144		83.149.144.78							
host005-160		83.149.160.5							
host019-154		83.149.154.19							
host017-148		83.149.148.17							
host081-144		83.149.144.81							
host016-148		83.149.148.16							
host067-144		83.149.144.67							
host153-147		83.149.147.153							
host095-144		83.149.144.95							
host019-146		83.149.146.19							
host014-148		83.149.148.14							
frebsd.computerhouseprato.com		83.149.154.10							
frebsd.giovannelli.com		83.149.149.149							
host012-144		83.149.144.12							
host023-146		83.149.146.23							

# Smokeping

- ▶ Folosit pentru monitorizarea latenței în rețea
- ▶ Trimite pachete de ping către stațiile configurate, implicit 20 de pachete la fiecare 300 de secunde
- ▶ Pe baza răspunsurilor primite poate genera grafice cu disponibilitatea echipamentelor sau a rețelei

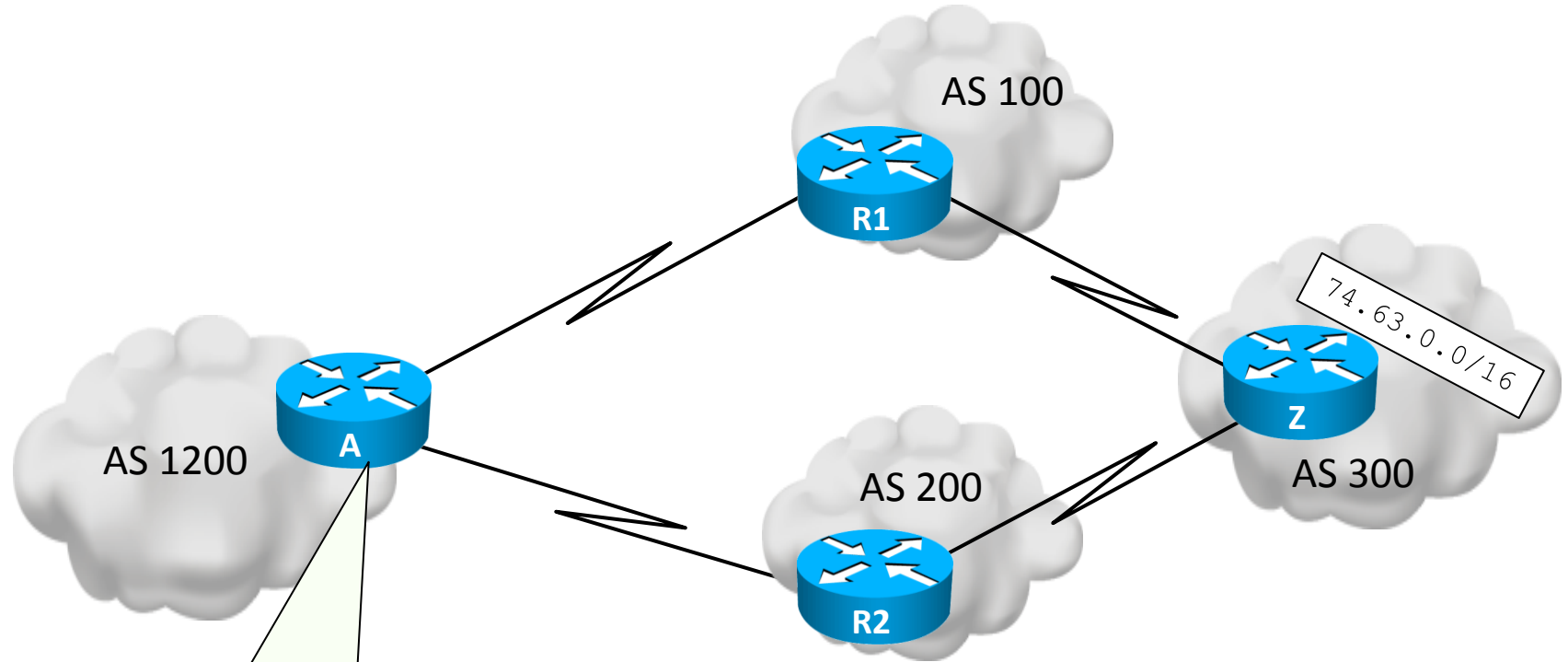


# IP SLA

---

- ▶ IP Service Level Agreement
- ▶ Folosit pentru monitorizarea resurselor
- ▶ Bazat pe crearea diverselor tipuri de pachete
  - ▶ TCP Connect
    - ▶ Folosit pentru simularea unui client, determinarea timpului de răspuns
  - ▶ FTP
  - ▶ ICMP Echo
  - ▶ HTTP
  - ▶ Poate seta și câmpul ToS din antetul IP

# IP SLA



```
ip sla monitor 11
  type echo protocol ipIcmpEcho 74.63.0.1
  frequency 5
ip sla monitor schedule 11 life forever start-time now
```

# Sumar

---

Autentificare  
Autorizare  
Accounting

Descoperirea  
Rețelei

Monitorizarea  
rețelei







# Test practic – Rezolvări

Proiectarea Rețelelor

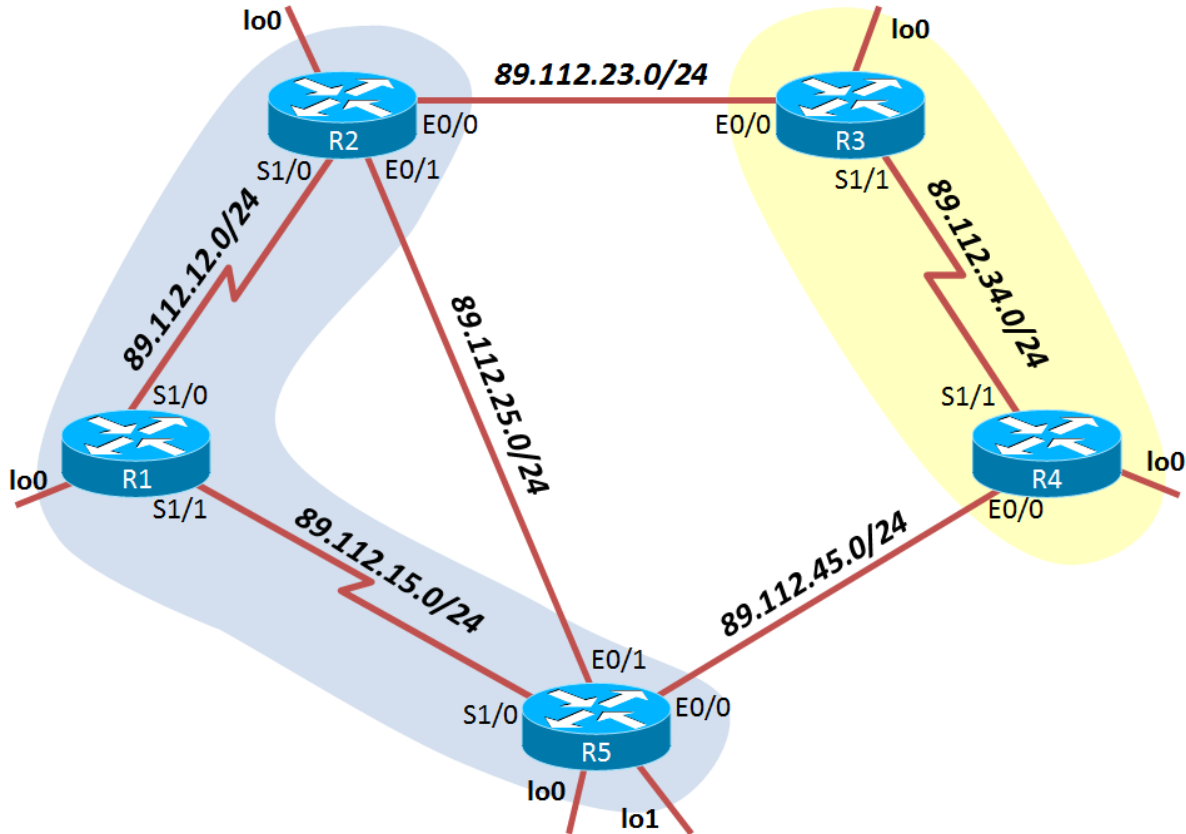
# Adresare IP

- Configurați adresele IP ale interfețelor de loopback conform tabelul de mai jos.

R1	Lo0	11.10.1.1 /24
R2	Lo0	12.14.14.1 /24
R3	Lo0	13.13.13.1 /24
R4	Lo0	14.14.14.1 /24
R5	Lo0	15.12.13.1 /26
	Lo1	15.12.13.65 /26

- 10 puncte

# Adresare IP



## R1

```
int 10
 ip add 11.10.1.1 255.255.255.0
int 11
 ip add 11.10.2.1 255.255.255.0
```

```
R1#sh ip int brief
```

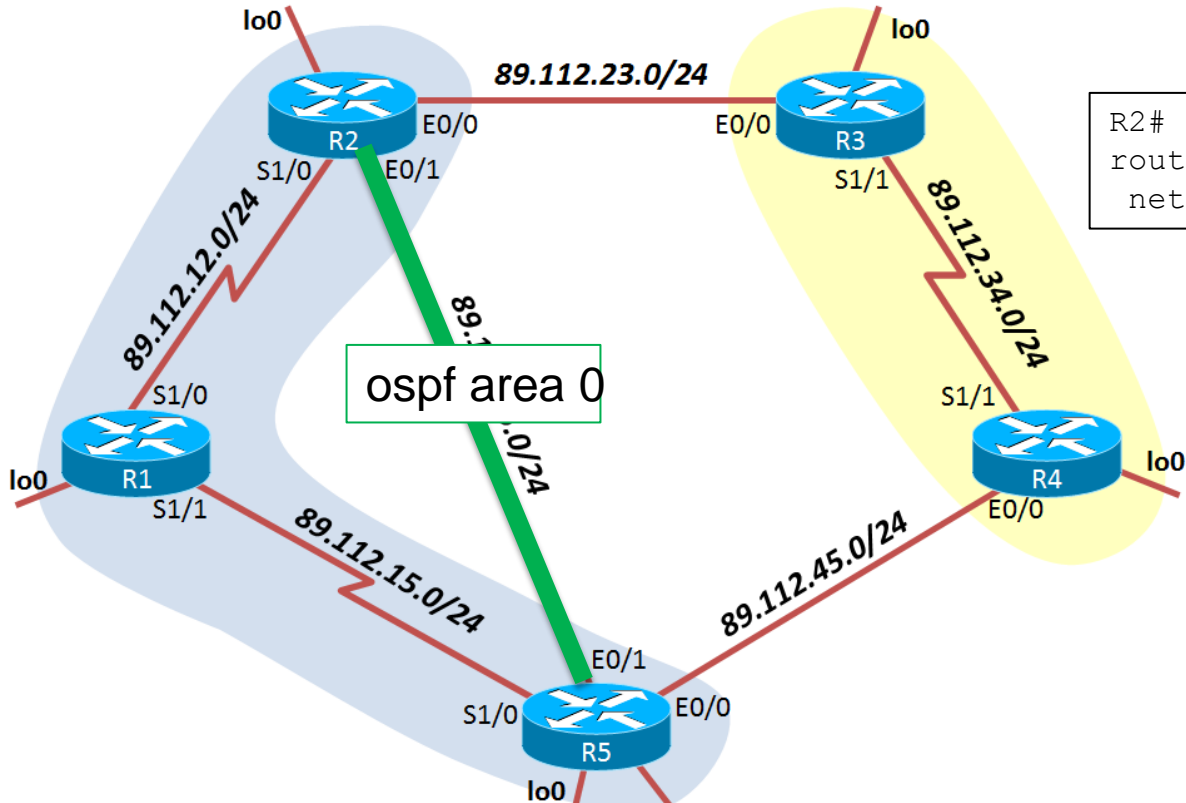
Interface	IP-Address	OK?	Method	Status	Protocol
[...]					
Loopback0	11.10.1.1	YES	manual	up	up
Loopback1	11.10.2.1	YES	manual	up	up

# OSPF

- a. Configurați OSPF aria 0 pe segmentul Ethernet dintre R2 și R5.
- b. Configurați OSPF aria 1 pe segmentul Serial dintre R1 și R2 și pe interfața de loopback lo0 a lui R2.
- c. Configurați OSPF aria 2 pe interfața lo0 a lui R1.
- d. Configure rețeaua OSPF astfel încât să aveți ping între R5 și interfața lo0 a lui R1
- e. Configurați rețeaua OSPF astfel încât R2 să fie mereu ales DR pe legătura dintre R2 și R5.
- f. Introduceți în OSPF, ca rute externe cu cost cumulativ, DOAR interfețele lo0 și lo1 ale lui R5.
- g. Introduceți în OSPF rețeaua lo2 a lui R2 ca rută internă în aria 0.
- h. Sumarizați rețele de pe lo0 și lo1 ale lui R5.
- i. Configurați aria 2 astfel încât această să nu accepte LSA-uri de tip 5. Verificați acest lucru.

► 35 puncte

# OSPF - a



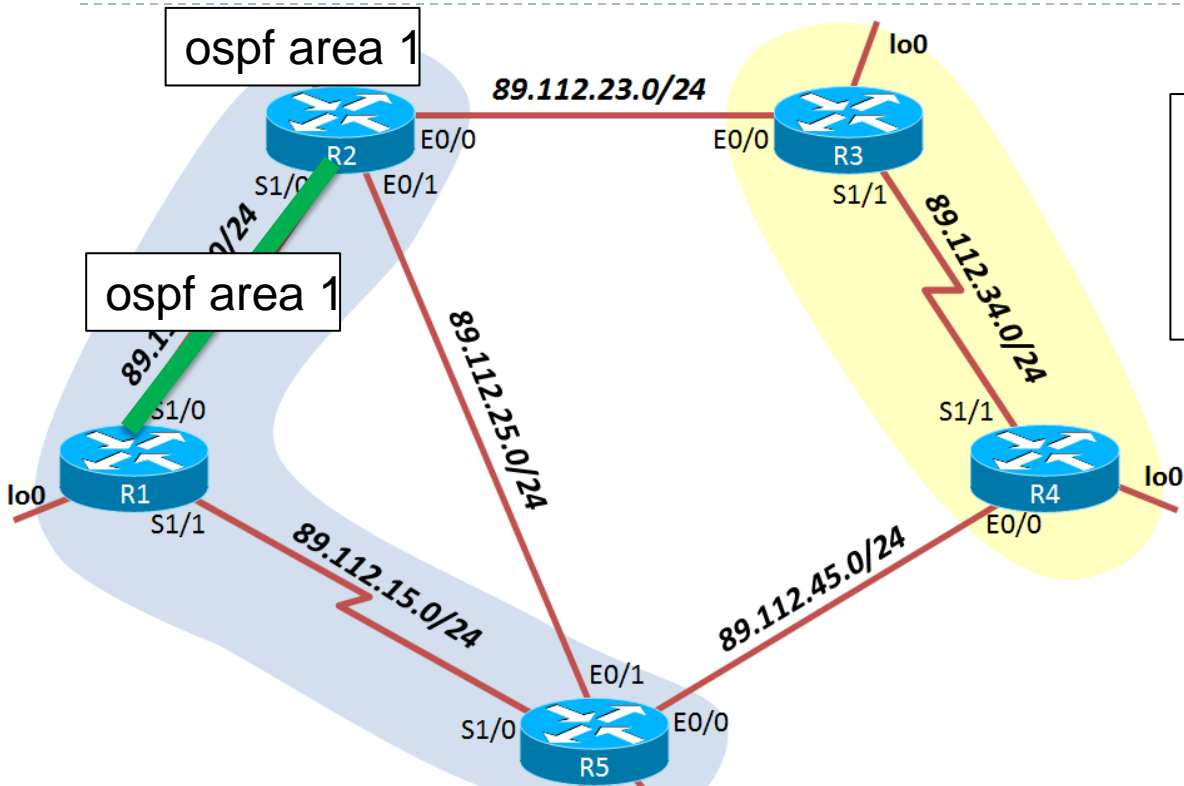
```
R2#  
router ospf 1  
network 89.112.25.0 0.0.0.255 area 0
```

```
R5#  
router ospf 1  
!  
int e0/1  
ip ospf 1 area 0
```

```
*Mar 1 00:46:06.519: %OSPF-5-ADJCHG: Process 1, Nbr 12.14.14.1 on Ethernet0/1  
rom LOADING to FULL, Loading Done  
R5#sh ip ospf nei
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
12.14.14.1	1	FULL/BDR	00:00:37	89.112.25.2	Ethernet0/1

# OSPF - b



```
R1#  
router ospf 1  
!  
int se 1/0  
ip ospf 1 area 1
```

```
R2#  
int se 1/0  
ip ospf 1 area 1  
int lo  
ip ospf 1 area 1
```

```
*Mar 1 01:01:04.835: %OSPF-5-ADJCHG: Process 1, Nbr 12.14.14.1 on Serial1/0 fro  
m LOADING to FULL, Loading Done  
R1#sh ip route ospf  
89.0.0.0/24 is subnetted, 3 subnets  
O IA 89.112.25.0 [110/74] via 89.112.12.2, 00:00:06, Serial1/0  
12.0.0.0/32 is subnetted, 1 subnets  
O 12.14.14.1 [110/65] via 89.112.12.2, 00:00:06, Serial1/0
```

# OSPF - c

```
R1(config)#int l0
R1(config-if)#ip ospf 1 area 2
R1#sh ip ospf database

        OSPF Router with ID (11.10.2.1) (Process ID 1)

          Router Link States (Area 1)

Link ID        ADV Router    Age           Seq#           Checksum Link count
11.10.2.1     11.10.2.1    455           0x80000002    0x00CFCD  2
12.14.14.1    12.14.14.1   448           0x80000002    0x00341C  3

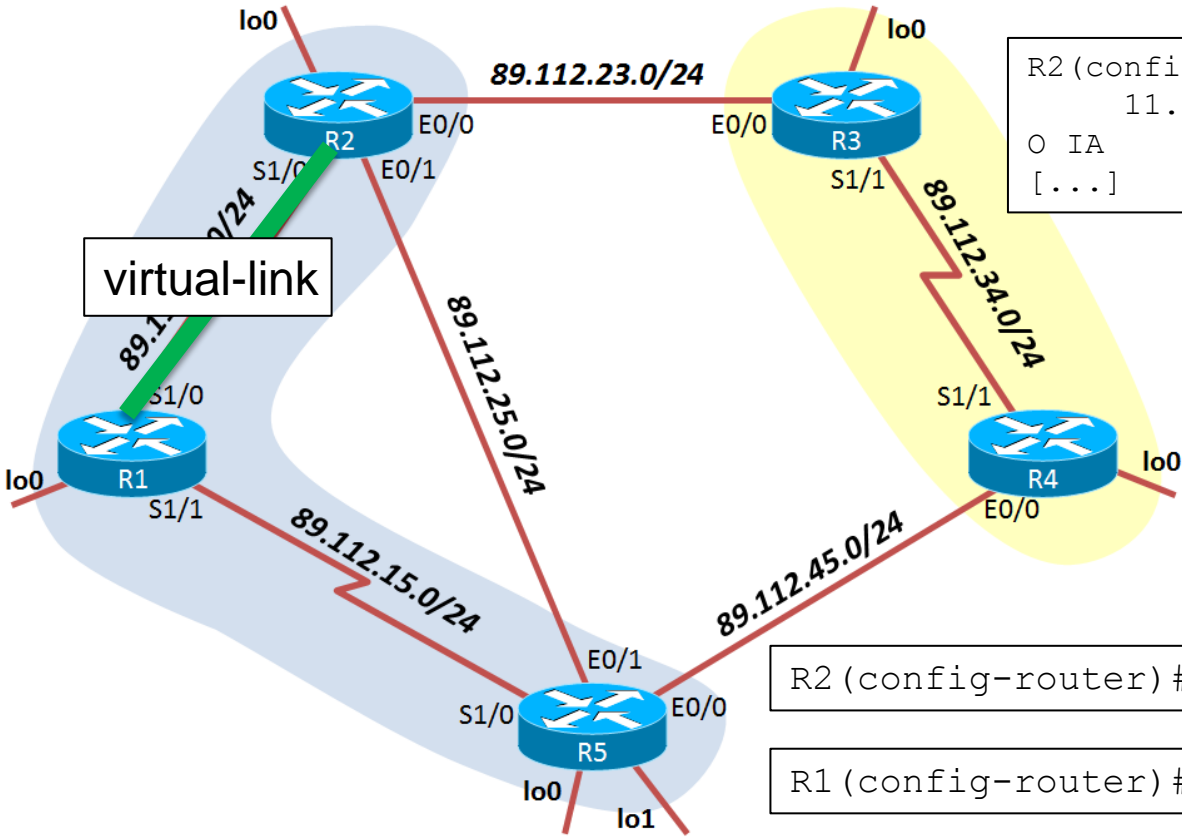
          Summary Net Link States (Area 1)

Link ID        ADV Router    Age           Seq#           Checksum
89.112.25.0    12.14.14.1   456           0x80000001    0x00C561

          Router Link States (Area 2)

Link ID        ADV Router    Age           Seq#           Checksum Link count
11.10.2.1     11.10.2.1    8             0x80000001    0x00BB2F  1
```

# OSPF - d



```
R2(config-router)#do sh ip route ospf
      11.0.0.0/32 is subnetted, 1 subnets
O IA   11.10.1.1 [110/65] via 89.112.12.1,
[...]
```

```
R2(config-router)#area 1 virtual-link 11.10.2.1
```

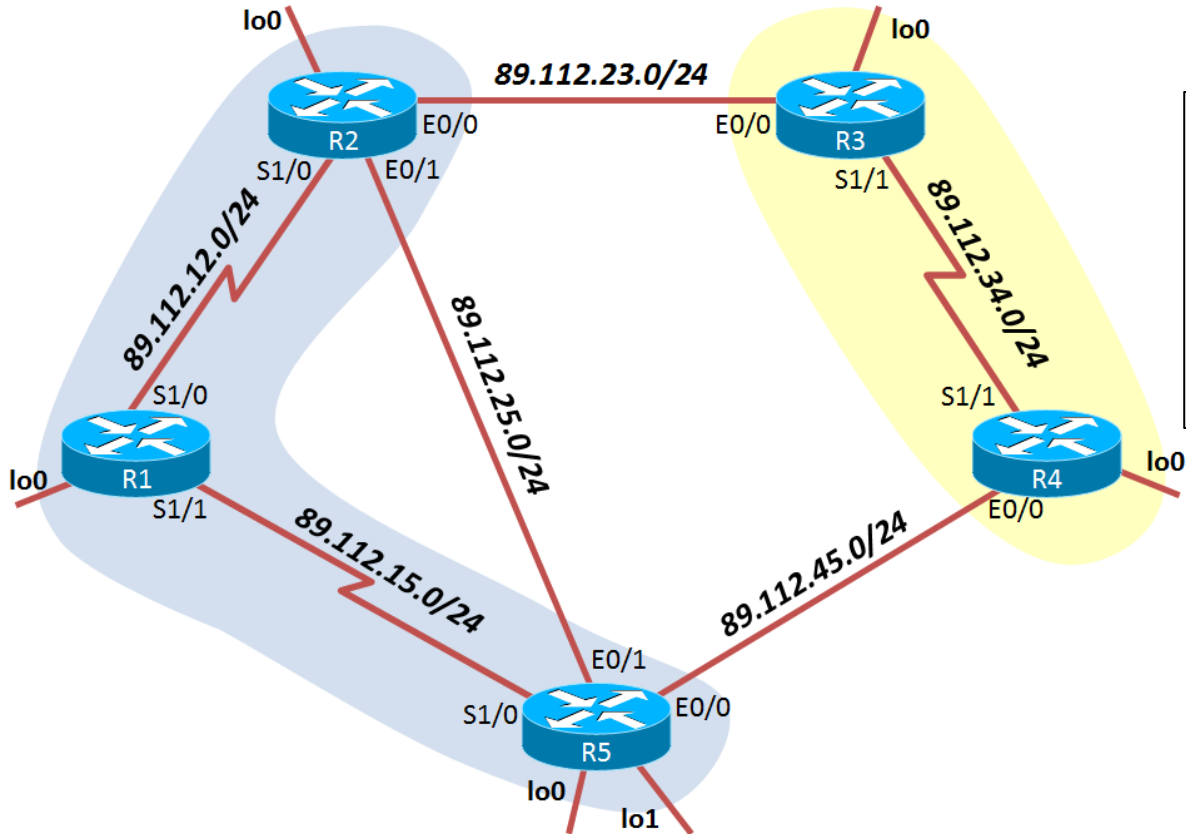
```
R1(config-router)#area 1 virtual-link 12.14.14.1
```

```
R1(config-router)#do sh ip ospf nei
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
12.14.14.1	0	FULL/ -	-	89.112.12.2	OSPF_VL0



# OSPF - e

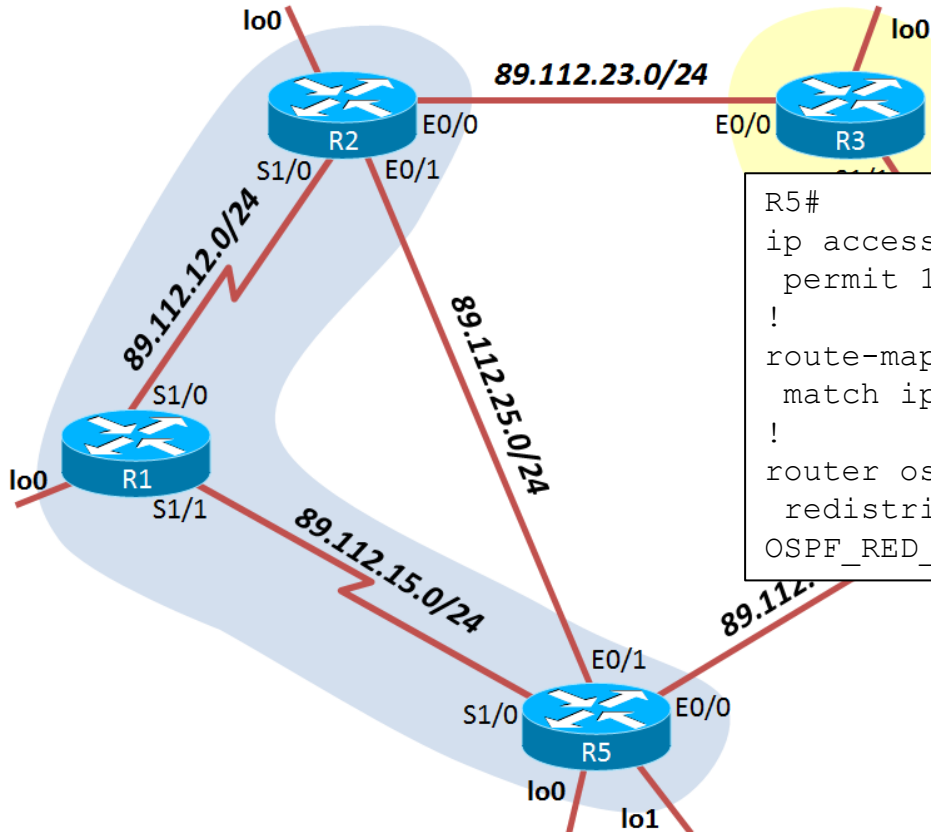


```
R2#  
int e 0/1  
 ip ospf priority 100  
!  
clear ip ospf 1 process  
Reset OSPF process? [no]:  
yes
```

```
R5#sh ip ospf nei
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
12.14.14.1	100	FULL/DR	00:00:35	89.112.25.2	Ethernet0/1

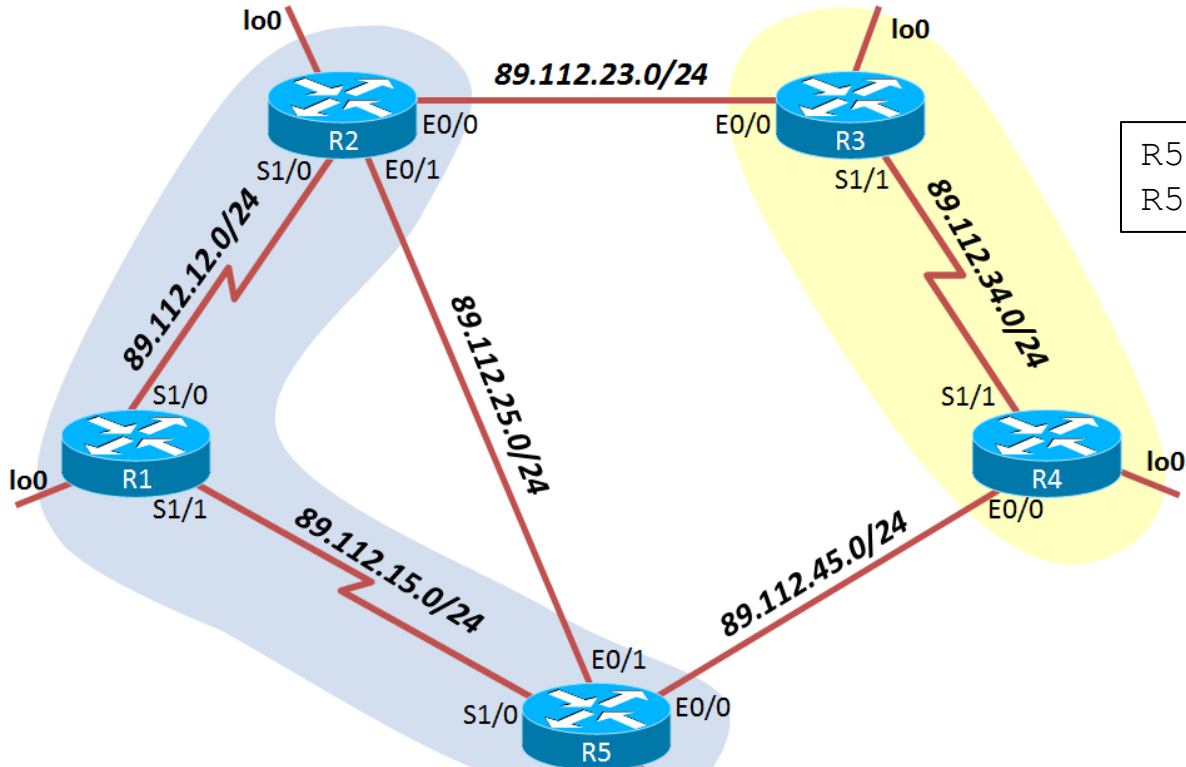
# OSPF - f



```
R5#  
ip access-list standard ACL_OSPF_RED_CONN  
 permit 15.12.13.0 0.0.0.127  
!  
route-map OSPF_RED_CONN  
 match ip address ACL_OSPF_RED_CONN  
!  
router ospf 1  
 redistribute connected subnets route-map  
 OSPF_RED_CONN metric-type 1
```

```
R1#sh ip route ospf  
[...]  
 15.0.0.0/26 is subnetted, 2 subnets  
O E1   15.12.13.0 [110/94] via 89.112.12.2, 00:00:01, Serial1/0  
O E1   15.12.13.64 [110/94] via 89.112.12.2, 00:00:01, Serial1/0
```

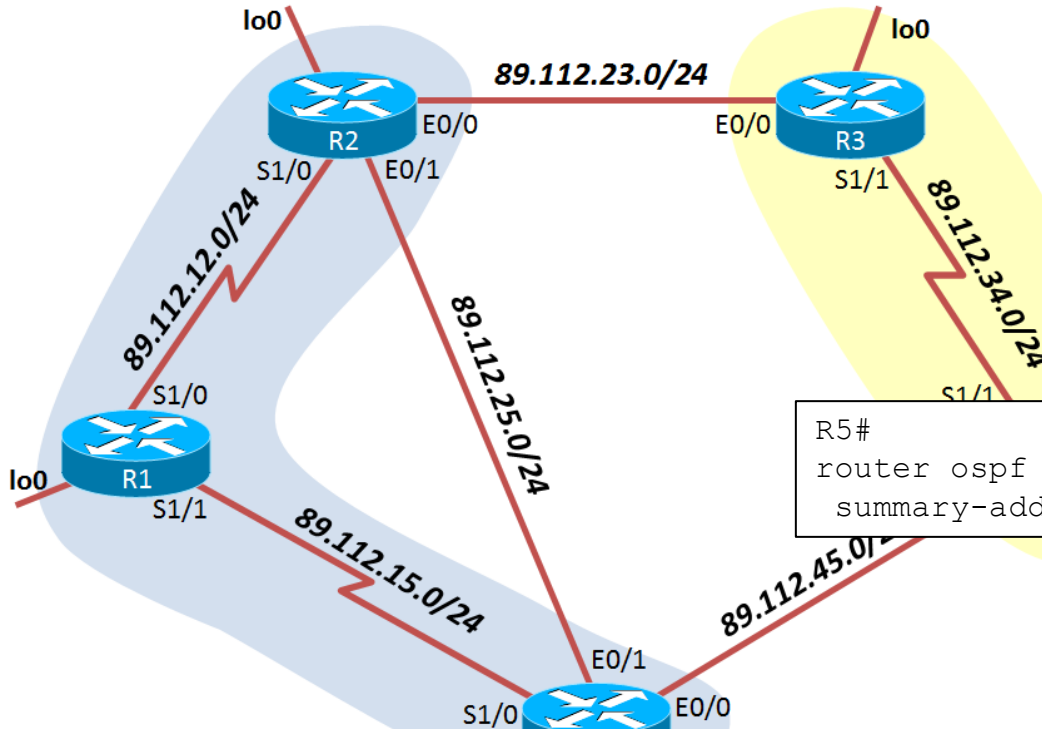
# OSPF - g



```
R5(config)#int 12
R5(config-if)#ip ospf 1 area 0
```

```
R1(config-router)#do sh ip route ospf
 89.0.0.0/24 is subnetted, 3 subnets
O    89.112.25.0 [110/74] via 89.112.12.2, 00:04:23, Serial1/0
 12.0.0.0/32 is subnetted, 1 subnets
O    12.14.14.1 [110/65] via 89.112.12.2, 00:32:23, Serial1/0
 15.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O    15.15.15.1/32 [110/75] via 89.112.12.2, 00:04:23, Serial1/0
```

# OSPF - h



```
R5#
router ospf 1
summary-address 15.12.13.0 255.255.255.128
```

```
R1(config-router)#do sh ip route ospf
 89.0.0.0/24 is subnetted, 3 subnets
O   89.112.25.0 [110/74] via 89.112.12.2, 00:06:29, Serial1/0
 12.0.0.0/32 is subnetted, 1 subnets
O   12.14.14.1 [110/65] via 89.112.12.2, 00:34:29, Serial1/0
 15.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O   15.15.15.1/32 [110/75] via 89.112.12.2, 00:06:29, Serial1/0
O E1 15.12.13.0/25 [110/94] via 89.112.12.2, 00:00:04, Serial1/0
```

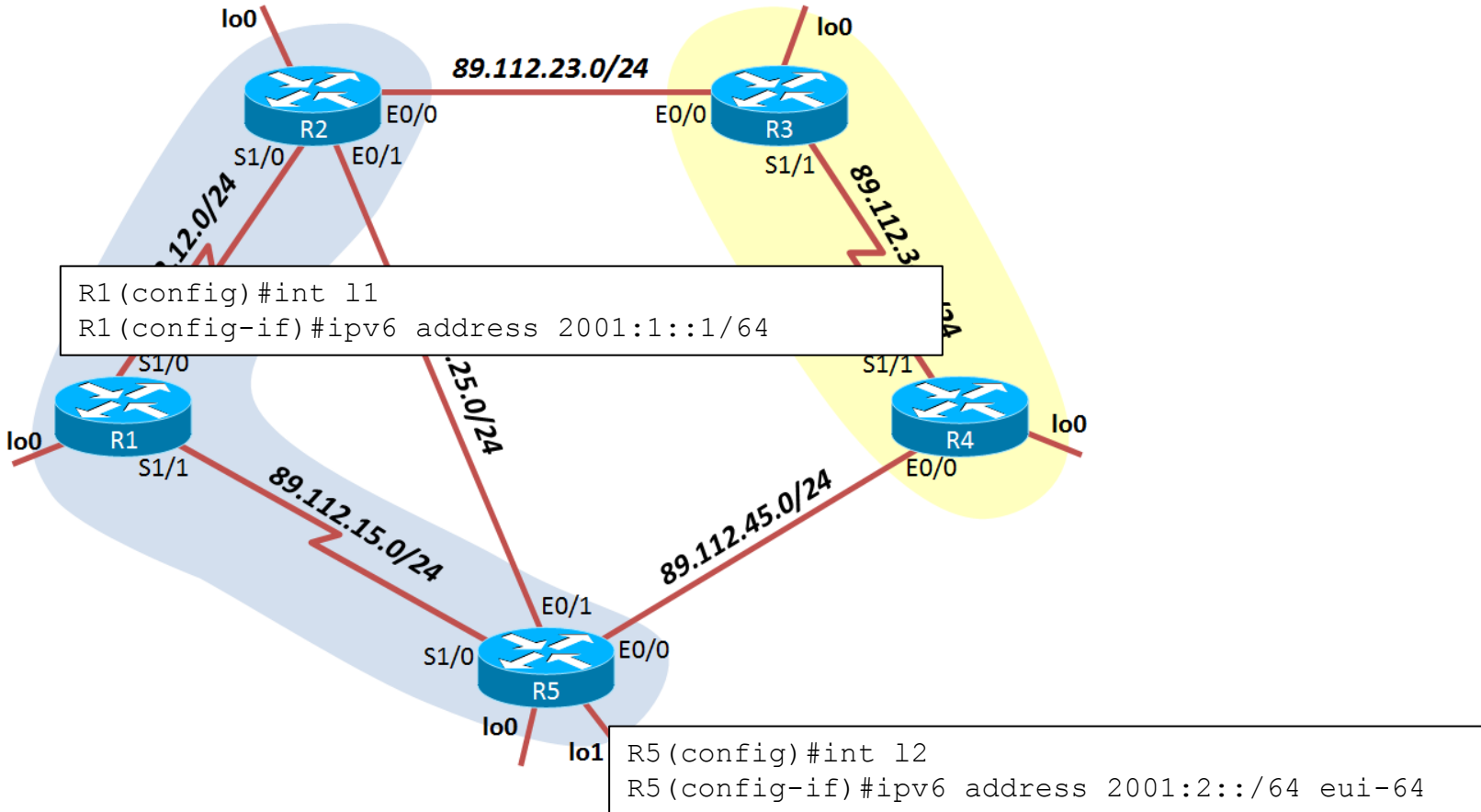
# OSPF - i

```
R1(config)#router ospf 1
R1(config-router)#area 2 stub
R1#sh ip ospf
[...]
  Area 2
    Number of interfaces in this area is 1 (1 loopback)
    It is a stub area
      generates stub default route with cost 1
    Area has no authentication
    SPF algorithm last executed 00:02:23.996 ago
    SPF algorithm executed 7 times
    Area ranges are
    Number of LSA 6. Checksum Sum 0x03ADC9
    Number of opaque link LSA 0. Checksum Sum 0x000000
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0
```

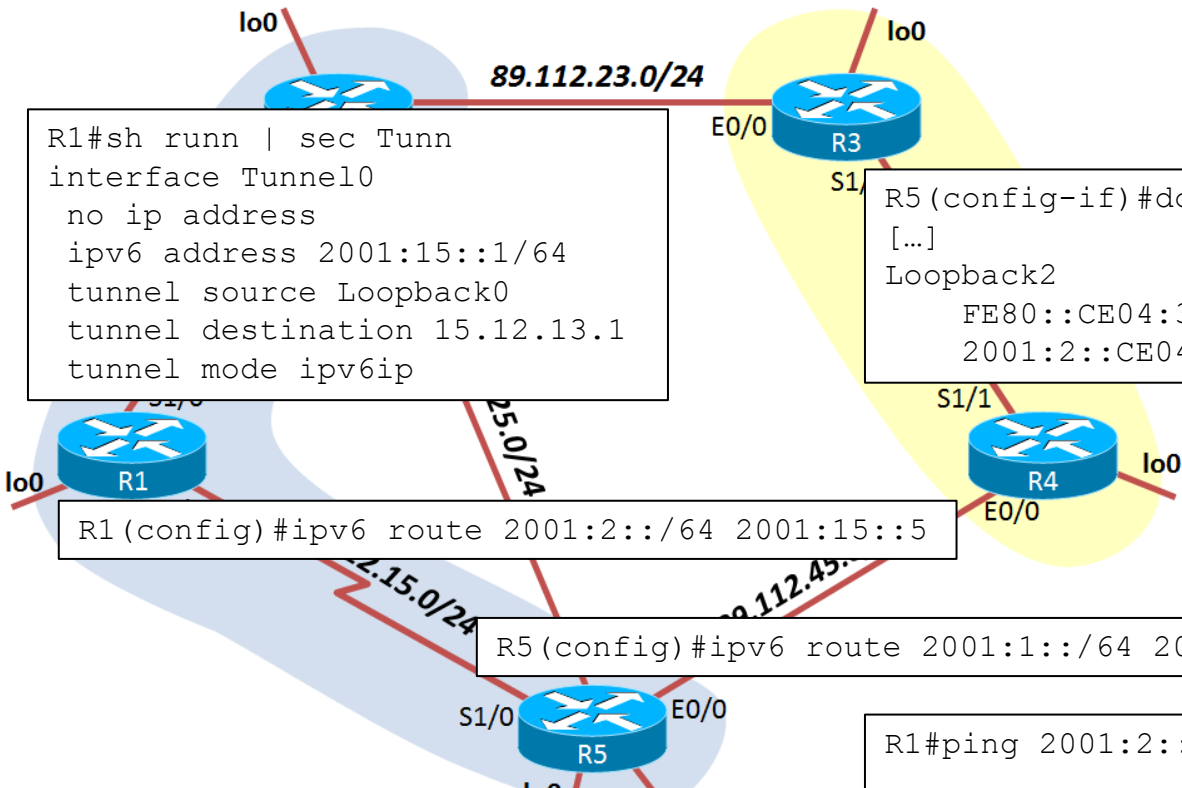
# IPv6

- ▶ a. Configurați adresa 2001:1::1/64 pe interfața lo1 a lui R1.
- ▶ b. Configurați adresa 2001:2::/64 pe interfața lo2 a lui R5. Ultimii 64 de biți ai adresei trebuie generați folosind metoda EUI-64.
- ▶ Configurați un tunel MCT între R1 și R5 astfel încât să existe conectivitate între lo1 a lui R1 și lo2 a lui R5. Pentru acest task este permisă folosirea rutelor statice.

# IPv6 – a,b



# IPv6 - c



```
R1#sh runn | sec Tunn
interface Tunnel0
no ip address
ipv6 address 2001:15::1/64
tunnel source Loopback0
tunnel destination 15.12.13.1
tunnel mode ipv6ip
```

```
R5(config-if)#do sh ipv6 int brief
[...]
Loopback2 [up/up]
FE80::CE04:3FF:FE8C:0
2001:2::CE04:3FF:FE8C:0
```

```
R1(config)#ipv6 route 2001:2::/64 2001:15::5
```

```
R5(config)#ipv6 route 2001:1::/64 2001:15::1
```

```
R5#sh runn | sec Tunn
interface Tunnel0
no ip address
ipv6 address 2001:15::5/64
tunnel source Loopback0
tunnel destination 11.10.1.1
tunnel mode ipv6ip
```

```
R1#ping 2001:2::CE04:3FF:FE8C:0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to
2001:2::CE04:3FF:FE8C:0, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip
min/avg/max = 200/735/1696 ms
```



# Route filtering

---

- ▶ Pe R2, filtrați rețeaua 15.12.13.0/25 astfel încât aceasta să nu fie instalată în tabela de rutare. R1 trebuie să aibă în continuare această rețea în tabela sa de rutare.
  - ▶ Hint: este posibil să trebuiască să restartați procesul OSPF pentru a vedea diferențele în tabela de rutare.

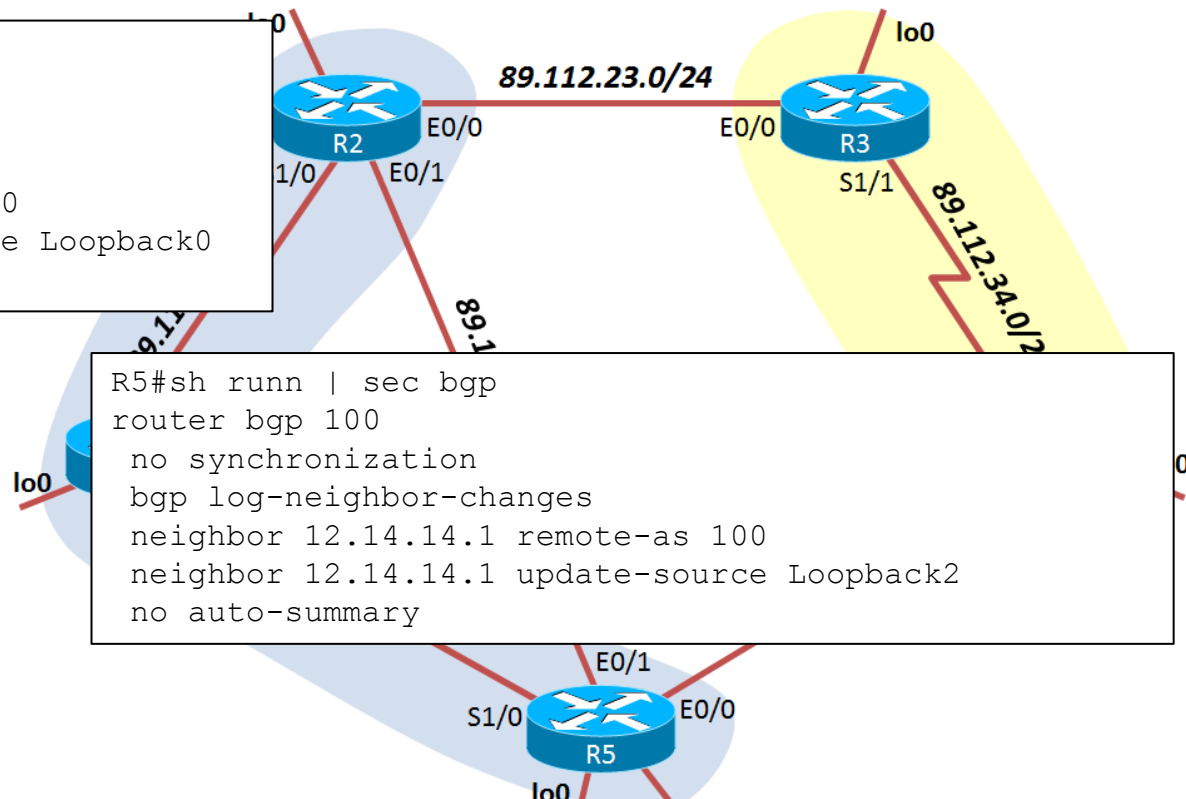
# Route filtering

```
R2(config)#router ospf 1
R2(config-router)#distribute-list OSPF_IN in
R2(config-router)#exit
R2(config)#ip access-list standard OSPF_IN
R2(config-std-nacl)#deny 15.12.13.0 0.0.0.127
R2(config-std-nacl)#permit any
!
R2(config-std-nacl)#do sh ip access-list
Standard IP access list OSPF_IN
  10 deny    15.12.13.0, wildcard bits 0.0.0.127 (2 matches)
  20 permit any (3 matches)
!
R2(config-std-nacl)#do sh ip route
[...]
  11.0.0.0/32 is subnetted, 1 subnets
O IA   11.10.1.1 [110/65] via 89.112.12.1, 00:00:05, Serial1/0
  89.0.0.0/24 is subnetted, 3 subnets
C     89.112.12.0 is directly connected, Serial1/0
C     89.112.25.0 is directly connected, Ethernet0/1
C     89.112.23.0 is directly connected, Ethernet0/0
  12.0.0.0/24 is subnetted, 1 subnets
C     12.14.14.0 is directly connected, Loopback0
  15.0.0.0/32 is subnetted, 1 subnets
O     15.15.15.1 [110/11] via 89.112.25.5, 00:00:05, Ethernet0/1
```

- ▶ a. Configurați următoarele adiacențe iBGP în AS-ul 100:
  - ▶ i. R2-R5 – adiacența trebuie realizată peste interfețele de loopback
  - ▶ ii. R1-R2 – adiacența trebuie realizată peste interfețele de loopback
  - ▶ iii. R3-R4 – adiacența nu trebuie realizată peste interfețele de loopback
- ▶ b. Configurați eBGP între R4-R5. Adiacența trebuie realizată direct peste interfețele fizice, fără a folosi interfețe de loopback.
- ▶ c. Configurați eBGP între R2 și R3 folosind interfețele de loopback pentru stabilirea adiacenței. Folosirea rutelor statice este permisă pentru acest task.
- ▶ d. Introduceți rețeaua lo1 a lui R4 în BGP cu codul de origine “?”
- ▶ e. Configurați rețeaua astfel încât R1 să poată da ping din interfața sa de loopback lo0 în interfața lo1 a lui R4. Nu este permisă folosirea rutelor statice.

# BGP – a

```
R2#sh runn | sec bgp
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 15.15.15.1 remote-as 100
neighbor 15.15.15.1 update-source Loopback0
no auto-summary
```



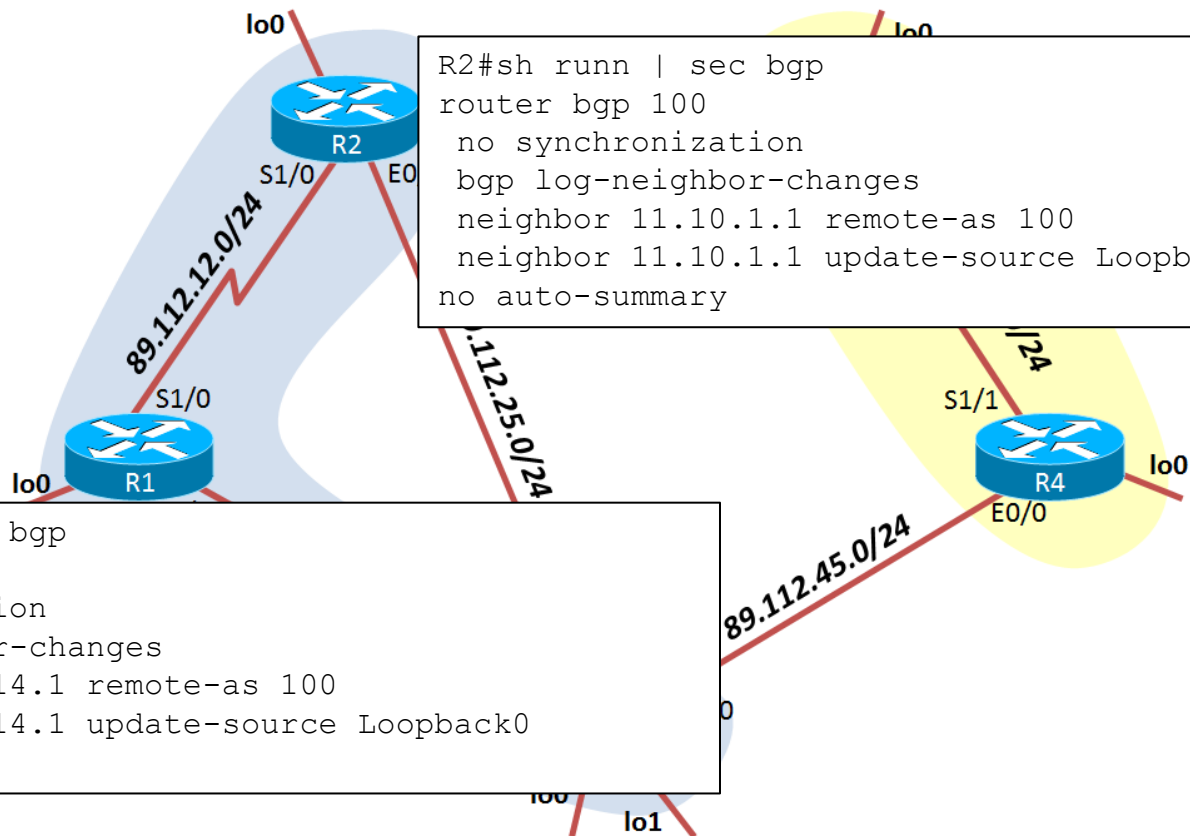
```
R5#sh runn | sec bgp
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 12.14.14.1 remote-as 100
neighbor 12.14.14.1 update-source Loopback2
no auto-summary
```

```
R5#sh ip bgp summ
BGP router identifier 15.15.15.1, local AS number 100
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
12.14.14.1	4	100	3	3	1	0	0	00:00:54	0



# BGP – a

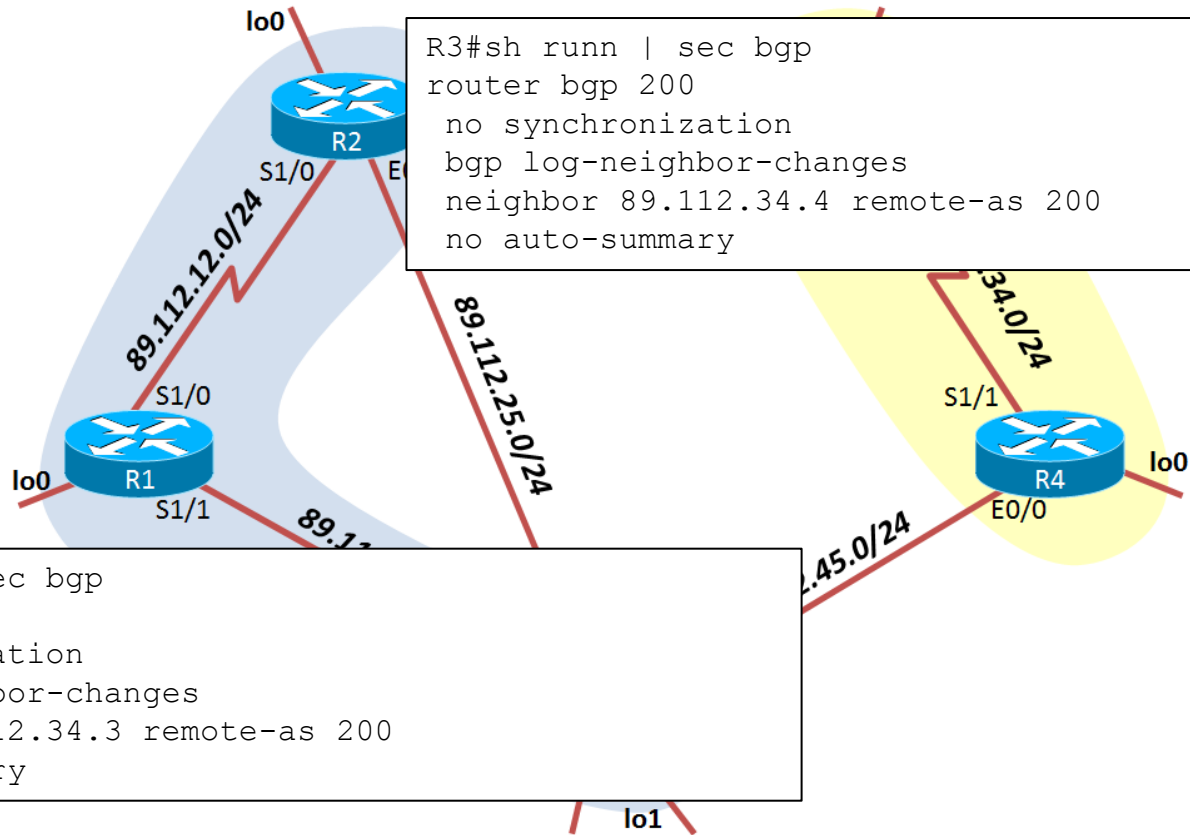


```
R1#sh runn | sec bgp
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 12.14.14.1 remote-as 100
neighbor 12.14.14.1 update-source Loopback0
no auto-summary
```

```
R2#sh runn | sec bgp
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 11.10.1.1 remote-as 100
neighbor 11.10.1.1 update-source Loopback0
no auto-summary
```

```
R1#sh ip bgp nei
BGP neighbor is 12.14.14.1, remote AS 100, internal link
BGP version 4, remote router ID 12.14.14.1
BGP state = Established, up for 00:03:39
```

# BGP – a



```
R3#sh runn | sec bgp
router bgp 200
no synchronization
bgp log-neighbor-changes
neighbor 89.112.34.4 remote-as 200
no auto-summary
```

```
R4#sh runn | sec bgp
router bgp 200
no synchronization
bgp log-neighbor-changes
neighbor 89.112.34.3 remote-as 200
no auto-summary
```

```
R4#sh ip bgp
```

```
R4#
```

# BGP – b,c

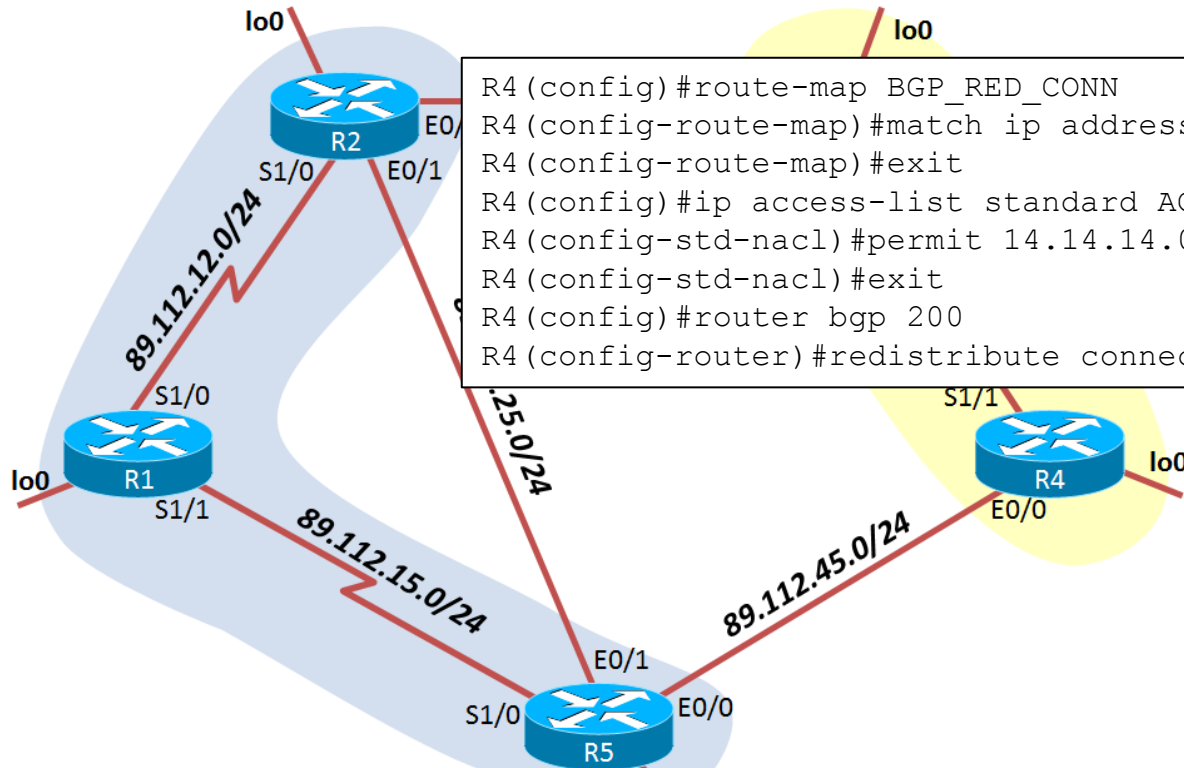
```
R5(config)#router bgp 100
R5(config-router)#neigh 89.112.45.4 remote-as 200
```

```
R4(config)#router bgp 200
R4(config-router)#neigh 89.112.45.5 remote-as 100
```

```
R2(config)#ip route 13.13.13.0 255.255.255.0 89.112.23.3
R2(config)#router bgp 100
R2(config-router)#neighbor 13.13.13.1 remote-as 200
R2(config-router)#neighbor 13.13.13.1 update-source 10
R2(config-router)#neighbor 13.13.13.1 ebgp-multihop 2
```

```
R3(config)#ip route 12.14.14.0 255.255.255.0 89.112.23.2
R3(config)#router bgp 200
R3(config-router)#neighbor 12.14.14.1 remote-as 100
R3(config-router)#neighbor 12.14.14.1 update-source 10
R3(config-router)#neighbor 12.14.14.1 ebgp-multihop 2
```

# BGP – d



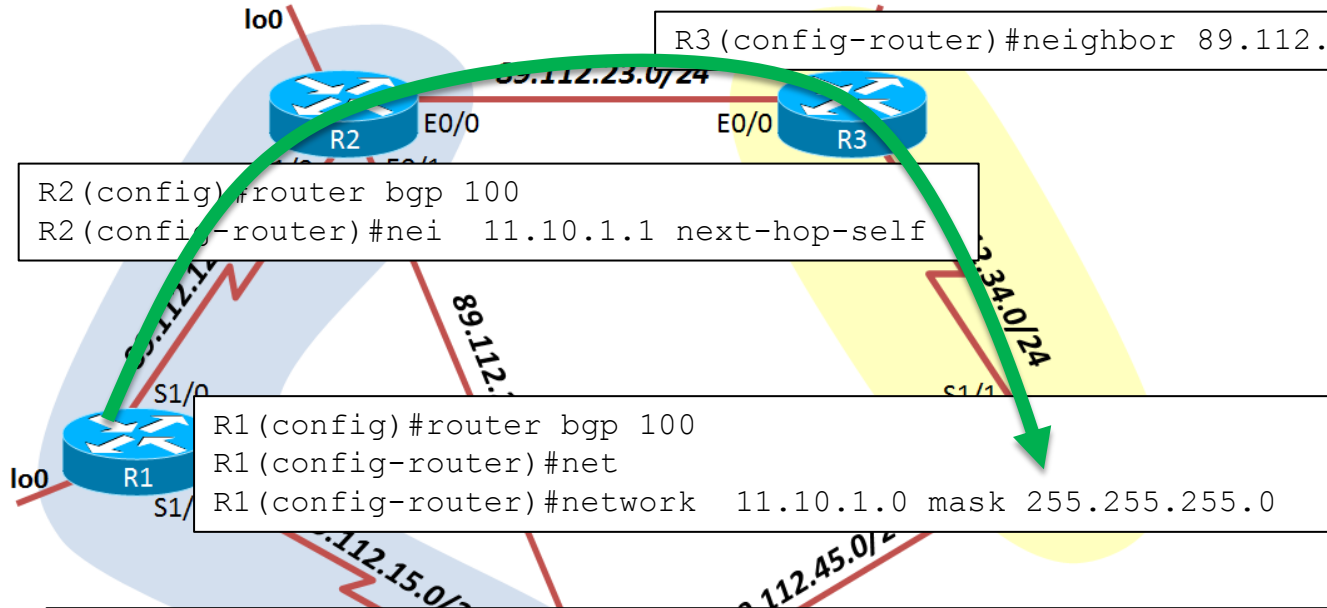
```
R4(config)#route-map BGP_RED_CONN
R4(config-route-map)#match ip address ACL_BGP_CONN
R4(config-route-map)#exit
R4(config)#ip access-list standard ACL_BGP_CONN
R4(config-std-nacl)#permit 14.14.14.0 0.0.0.255
R4(config-std-nacl)#exit
R4(config)#router bgp 200
R4(config-router)#redistribute connected route-map BGP_RED_CONN
```

```
R5#sh ip bgp
[...]
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
* i14.14.14.0/24  13.13.13.1        0      100     0 200 ?
*>                89.112.45.4        0              0 200 ?
```



# BGP – e



```
R3(config-router)#neighbor 89.112.34.4 next-hop-self
```

```
R2(config)#router bgp 100
R2(config-router)#nei 11.10.1.1 next-hop-self
```

```
R1(config)#router bgp 100
R1(config-router)#net
R1(config-router)#network 11.10.1.0 mask 255.255.255.0
```

```
R4#sh ip bgp
[...]
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i11.10.1.0/24	89.112.34.3	0	100	0	100 i
*> 14.14.14.0/24	0.0.0.0	0		32768	?

```
R4#ping 11.10.1.1 source 11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.10.1.1, timeout is 2 seconds:
Packet sent with a source address of 14.14.14.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/134/252 ms
```



**Succes!**

