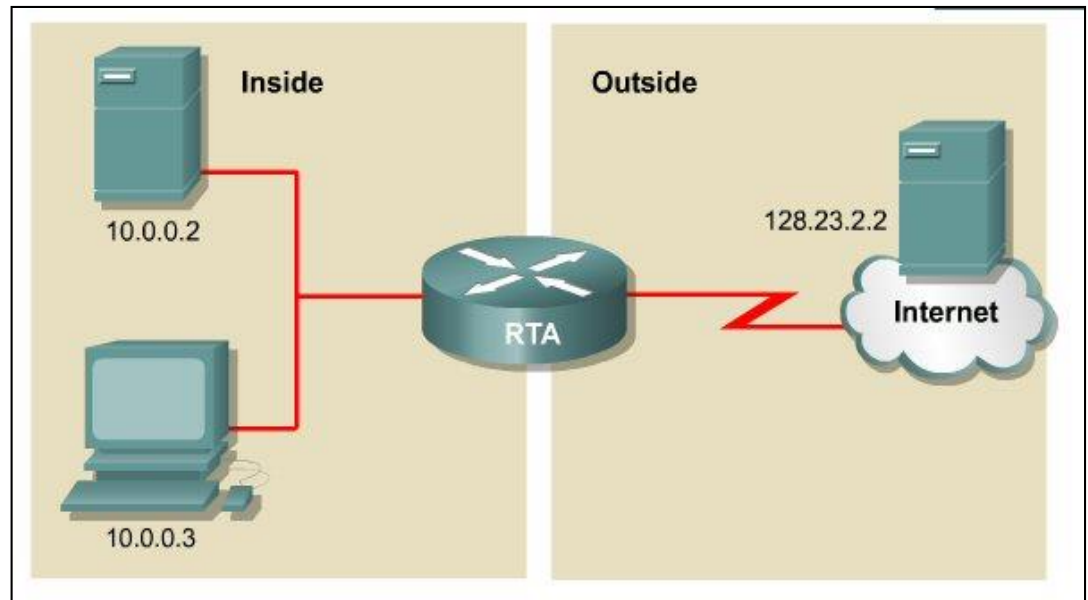


Curs 12: Fundamente ale rutării în rețea

NAT & tunneling

De ce translatarea adreselor?

- Problema epuizării adreselor IPv4
- NAT/PAT
- Configurare NAT cu iptables
- Dezavantajele translației



Epuizarea adreselor IPv4

- Problemă majoră IPv4
- Au fost introduse mecanisme pentru conservarea spațiului
- S-au alocat trei spații pentru adrese private:
 - 10.0.0.0/8
 - 172.16.0.0/12
 - 192.168.0.0/16
- Aceste adrese nu pot fi folosite în Internet
- Pentru ca o stație cu adresă privată să poată accesa Internetul adresa acesteia trebuie translatată

Procesul de translatare

- Atunci când un pachet trece printr-un ruter adresele IP sursă și destinație rămân neschimbate
- Procesul de translatare presupune schimbarea adresei IP sursă sau destinație a unui pachet la trecea printr-un ruter
- Procesul poartă numele de **NAT** (Network Address Translation)
- Pentru conectivitate translatarea trebuie să aibă loc în ambele direcții

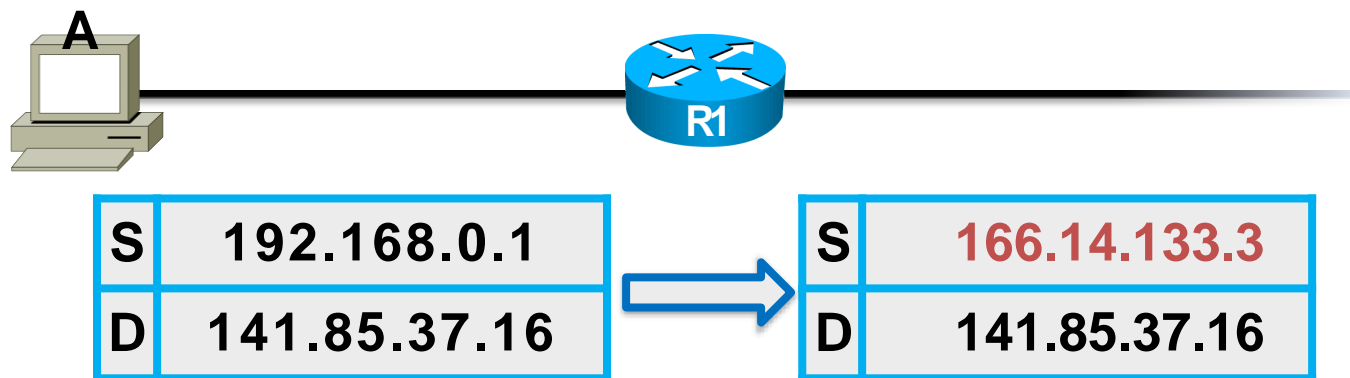
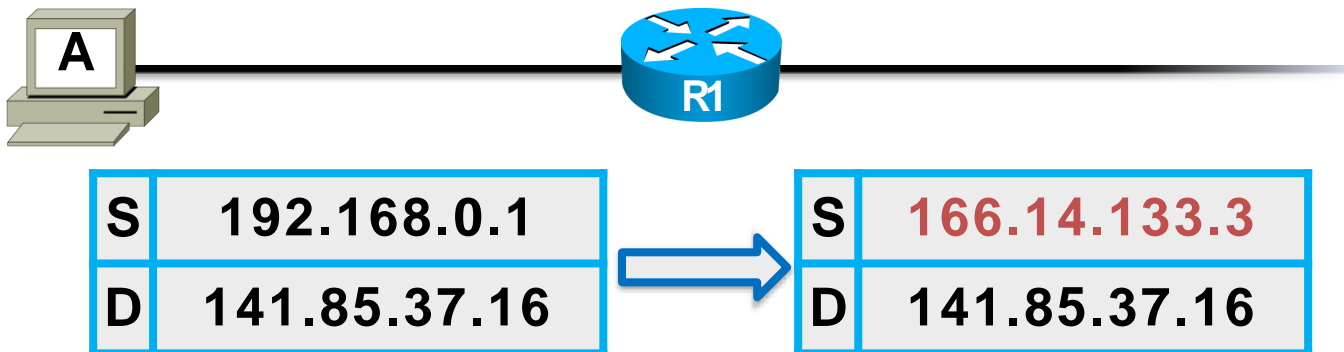


Tabela NAT

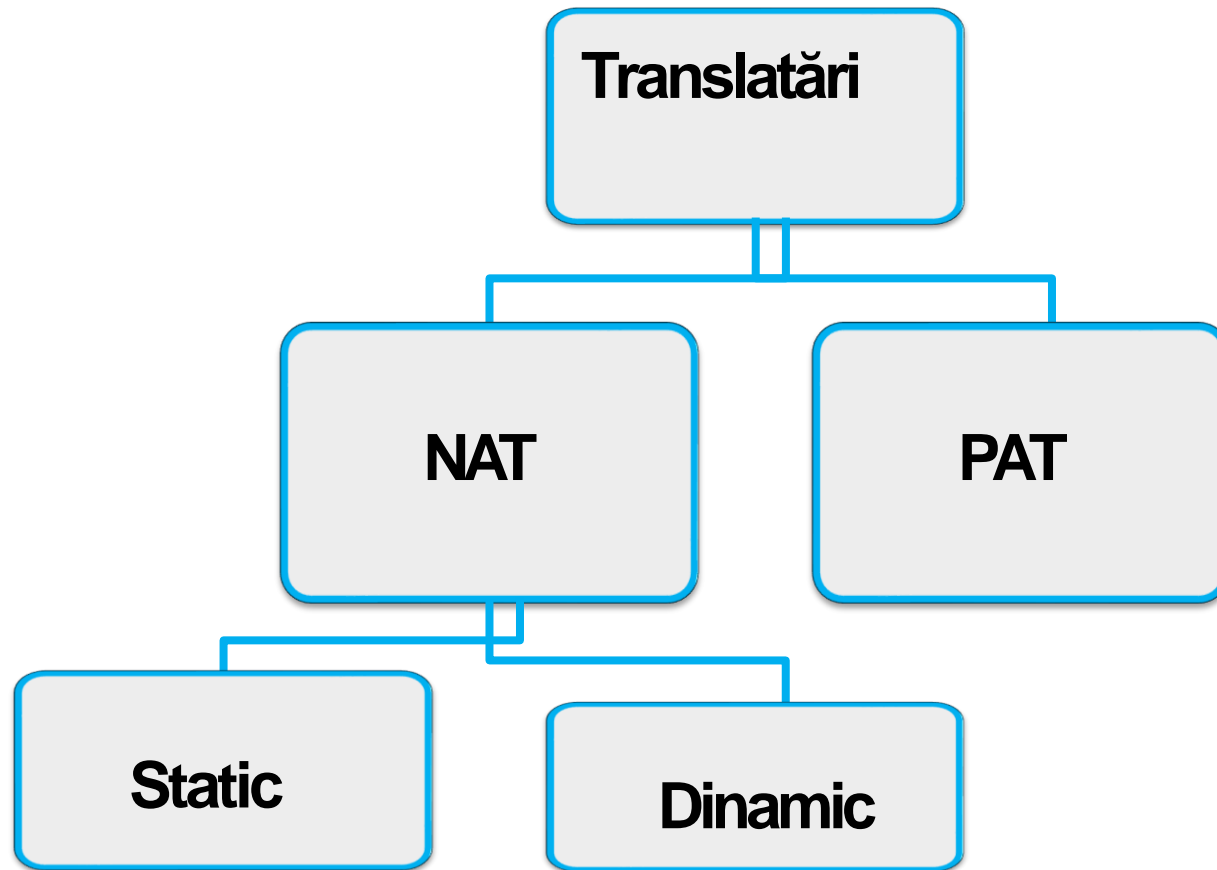
- Ruterul ține evidența translatărilor ce trebuie făcute în tabela de NAT
- Tabela NAT:
 - Poate fi construită static (de către administrator) sau dinamic (prin inspectarea traficului ce trece prin ruter)
 - Păstrează o listă de asocieri **adresă internă – adresă externă**

Tabela NAT:

192.168.0.1 – 166.14.133.3

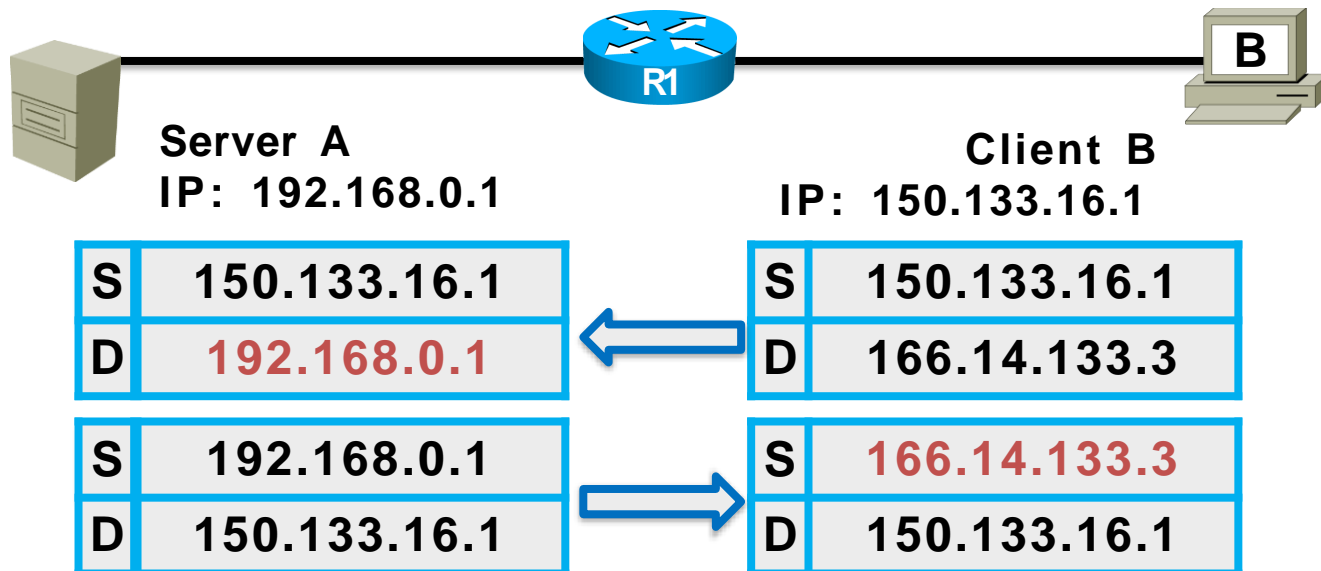


Procesul de translatare



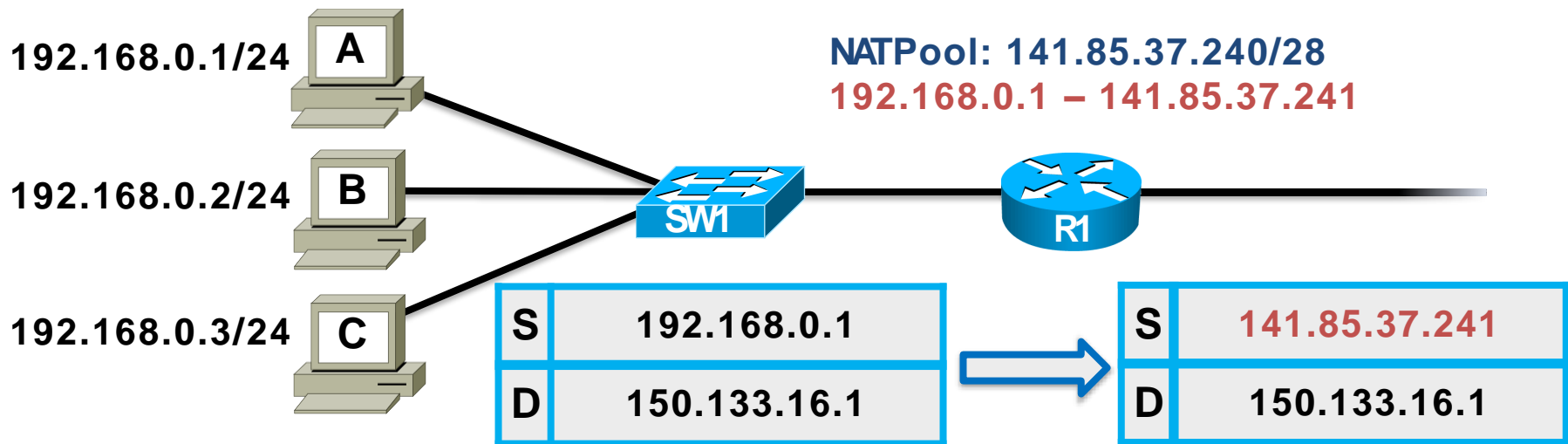
NAT Static

- **Problemă:** **Serverul A** are o adresă privată însă vrem să fie accesibil în exterior printr-o adresă publică unică și constantă
- **Soluție:** NAT Static
 - Adresa internă a serverului este mereu translatată la o adresă publică rezervată **192.168.0.1 – 166.14.133.3**



NAT Dinamic

- Problemă: Avem în rețeaua privată 40 de stații dar doar 20 de adrese publice
 - Soluție: NAT Dinamic
 - Stațiile care vor să comunice în Internet primesc temporar una din adresele publice disponibile (din NAT Pool), dacă mai există adrese nefolosite
- Ar putea fi o soluție NAT dinamic pentru problema anterioară a serverului?

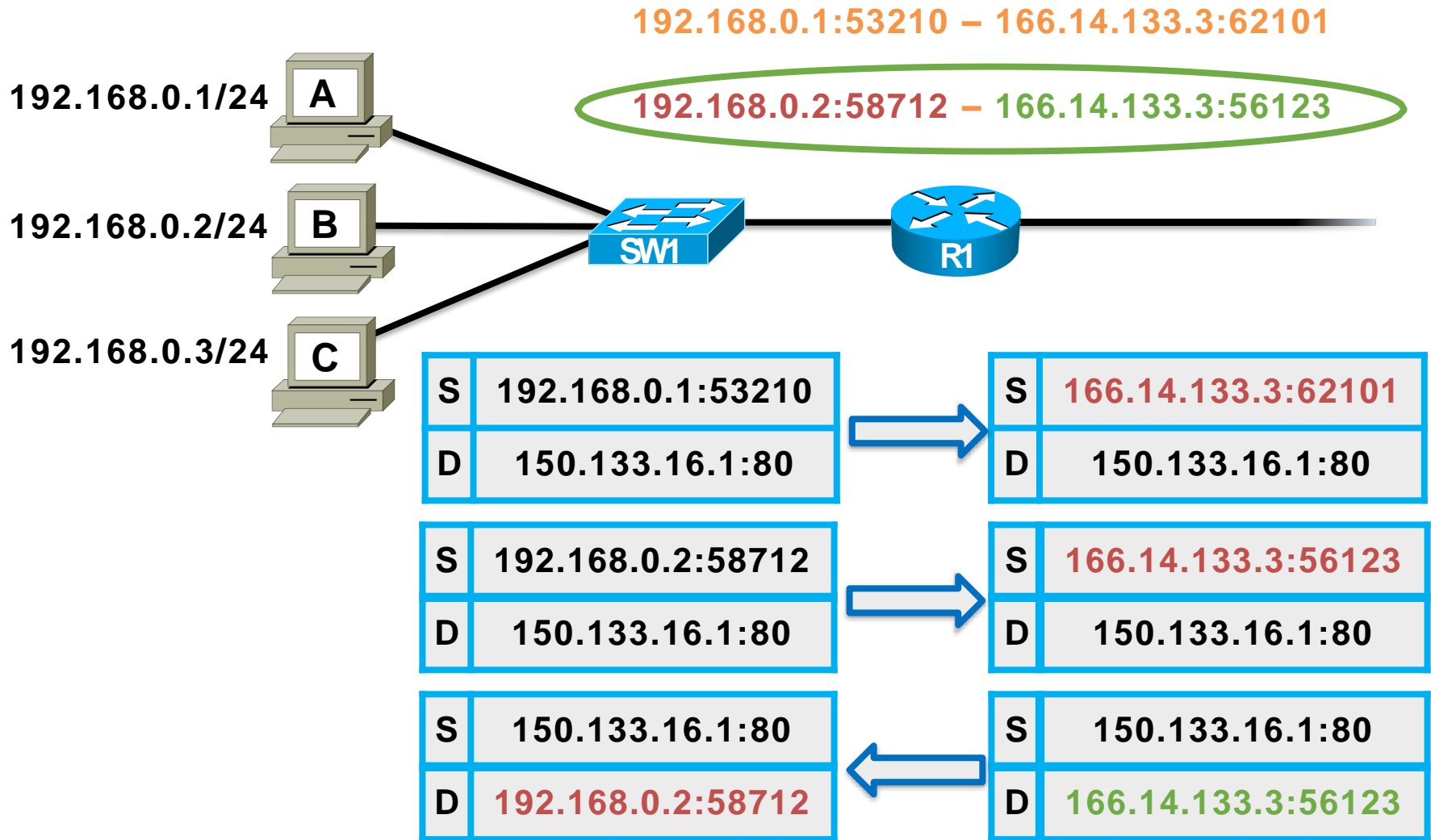


PAT

- Problemă: Avem în rețeaua privată 40 de stații dar o singură adresă publică
- Soluție: PAT (Port Address Translation)
 - Mai poartă și numele de masquerade sau NAT Overload
 - La traducere se asociază fiecărei comunicații și un port (un identificator de nivel transport ce indică programul sursă/destinație) pe ruter
 - Când răspunsul destinatarului ajunge la ruter, acesta citește portul din pachet și consultă tabela NAT pentru a vedea în ce să translateze

Tabela NAT		
192.168.0.1:80	–	166.14.133.3:62101
192.168.0.1:1614	–	166.14.133.3:62102
192.168.0.2:80	–	166.14.133.3:63105
192.168.0.3:1811	–	166.14.133.3:48231

PAT



NAT în Linux

- Se implementează folosind utilitarul iptables
- Se foloseşte tabela **nat**
- Lanţurile modificate de comenzile de nat sunt:
 - **PREROUTING** pentru rescrierea destinaţiei
 - **POSTROUTING** pentru rescrierea sursei

iptables

- Utilitar Linux
- Face parte din proiectul Netfilter
- Permite unei mașini Linux să:
 - Filtreze pachetele
 - Translateze adrese
 - Rescrie câmpurile unui pachet
- Configurat prin scrierea de **reguli**
- Regulile iptables sunt compuse din două secțiuni principale:
 - Șablon - ce valori trebuie să aibă câmpurile din pachet pentru a se acționa asupra lor
 - Acțiune - ce operație va efectua mașina Linux asupra pachetului

Tabele iptables

- **Filter**

- Conține reguli ce spun ce trafic poate să treacă și ce trafic trebuie aruncat
- Exemplu:
 - O adresă externă a eșuat în mod repetat să se conecteze la un server Linux prin SSH
 - Se adaugă o regulă de filtrare care blochează orice trafic de la adresa respectivă

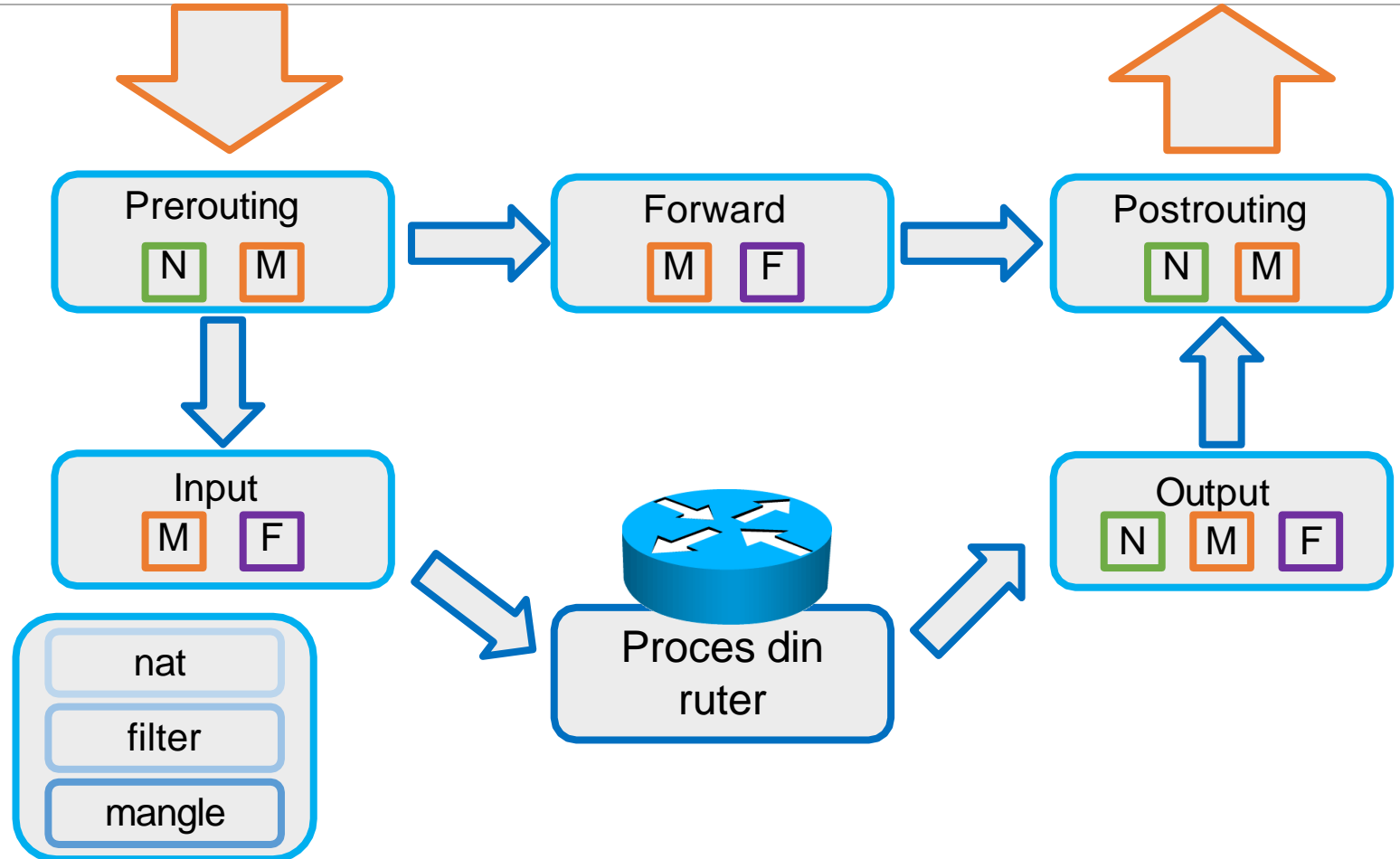
- **Nat**

- Conține reguli pentru translatarea adreselor în procesul de NAT
- Exemplu:
 - O adresă privată trebuie să acceseze un server din Internet
 - Se adaugă o regulă de NAT care rescrie adresa sursă privată cu o adresă publică
 - La întoarcere, pachetul va fi rescris invers

- **Mangle**

- Conține reguli pentru alterarea specializată a pachetelor (ex: TTL)

Reguli iptables

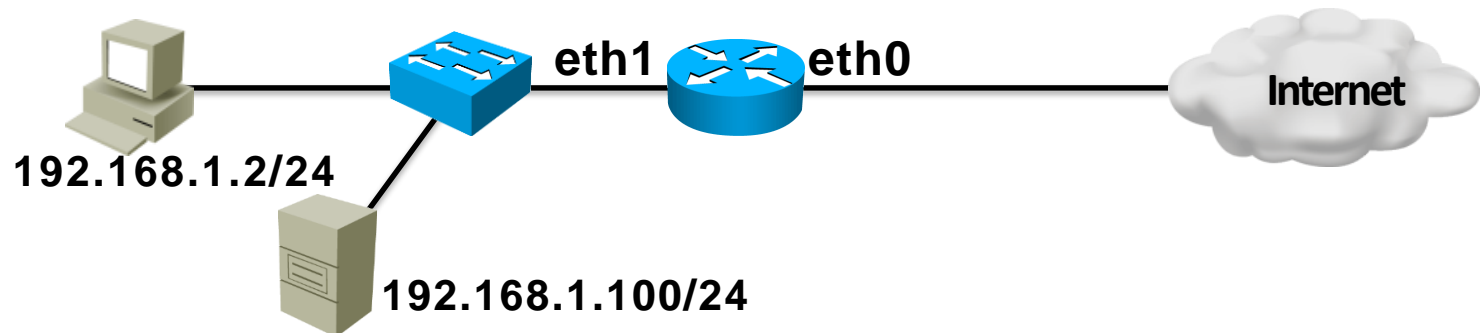


NAT static cu iptables

- Regulele sunt adăugate în tabela **nat** – lanțul **POSTROUTING**
- Este folosit target-ul **SNAT**:
 - Specifică în ce să fie rescrise IP-ul și portul sursă
 - Procesarea lanțului se încheie
- Pentru NAT static trebuie specificată sursa (-s)

```
linux# iptables -t nat -A POSTROUTING -s 192.168.1.100 -j SNAT -- to-source 141.85.200.1
```

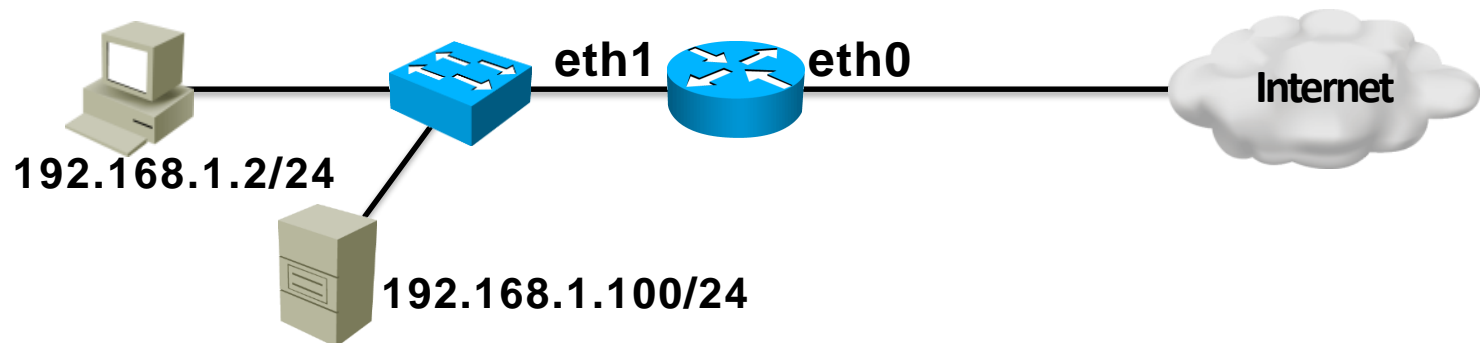
- Atenție: **SNAT** vine de la Source NAT (nu de la static NAT)



NAT static cu iptables

- Dacă este inițiată din exterior conexiunea, aceasta nu va ajunge la server
- Trebuie creată și regula inversă, care rescrie adresa destinație la trecerea prin ruter
- Rescrierea destinației se face cu target-ul **DNAT** (Destination NAT)
 - Se folosește lanțul de **PREROUTING** în acest caz. De ce?

```
linux# iptables -t nat -A PREROUTING -d 141.85.200.1 -j DNAT --to-destination 192.168.1.100
```

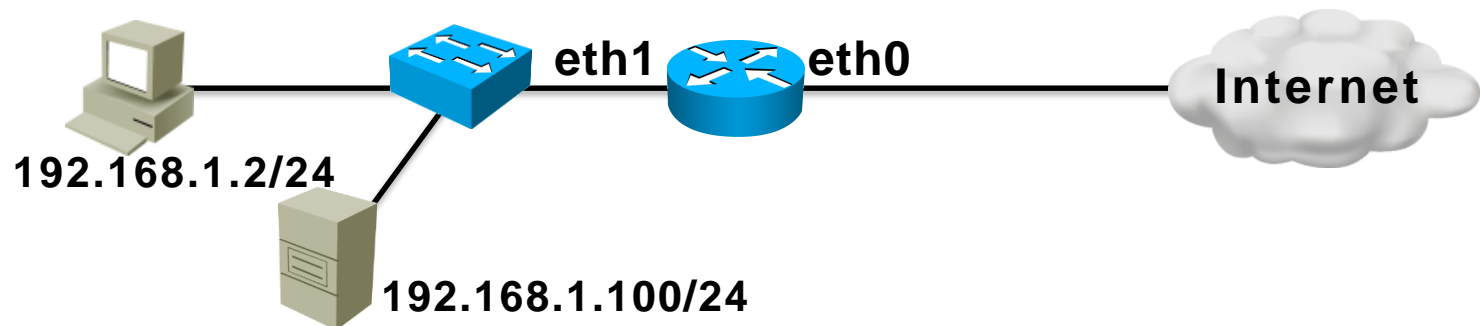


NAT dinamic/PAT cu iptables

- Regulile sunt adăugate în tabela **nat** – lanțul **POSTROUTING**
- Tot target-ul **SNAT** este folosit:
 - Pentru NAT dinamic se poate specifica un range de adrese IP
 - Ruterul nu mapează adrese unu la unu (se folosește de fapt o combinație de NAT dinamic cu PAT)

```
linux# iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT -to-source 141.85.200.2-141.85.200.6
```

- Vor putea fi inițiate conexiuni din exterior?



NAT cu iptables

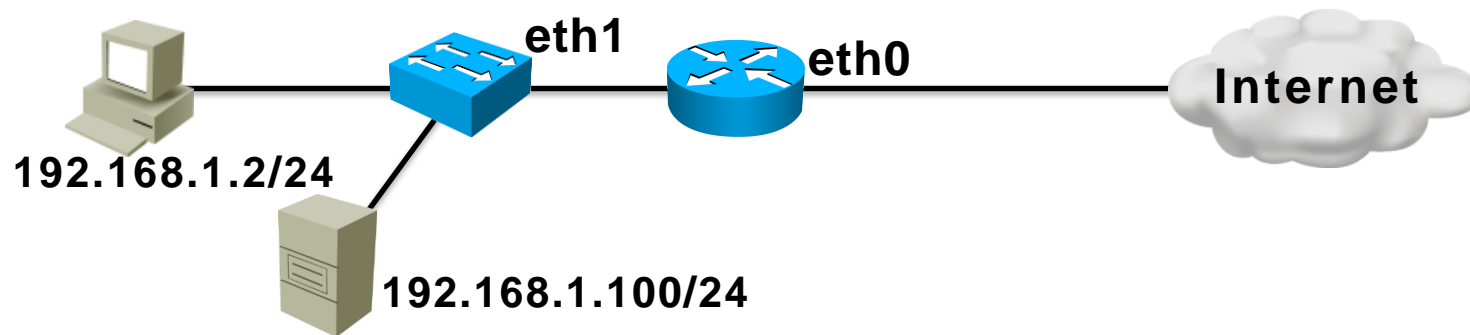
- Este vreo problemă cu setul de reguli de mai jos?
 - **R:** Da. Niciodată nu se va face match pe a doua regulă de NAT deoarece sursa 192.168.1.100 va face match pe prima regulă

```
linux# iptables -t nat -F
```

```
linux# iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT --  
to-source 141.85.200.2-141.85.200.6
```

```
linux# iptables -t nat -A POSTROUTING -s 192.168.1.100 -j SNAT --  
to-source 141.85.200.1
```

```
linux# iptables -t nat -A PREROUTING -d 141.85.200.1 -j DNAT --  
to-destination 192.168.1.100
```

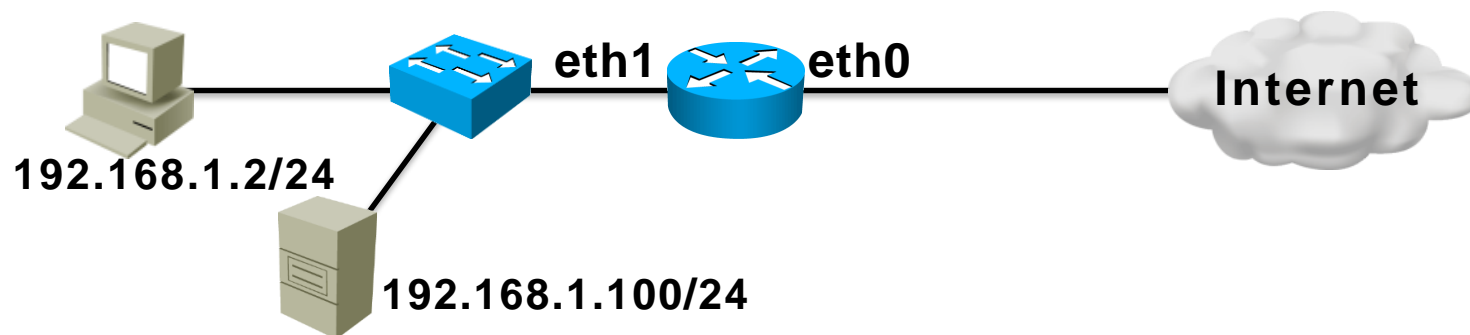


NAT dinamic/PAT cu iptables


- Target-ul **MASQUERADE** specifică faptul că se va folosi IP-ul interfeței de ieșire în traducere
- Utilă când interfața către Internet ia prin DHCP adresa
 - **MASQUERADE** face flush la mapări când interfața e repornită
- Se poate folosi pentru PAT doar un subset de porturi cu `--to-ports`
- Trebuie specificat tipul de trafic (UDP sau TCP):

```
linux# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
linux# iptables -t nat -A POSTROUTING -o eth0 -p tcp -j MASQUERADE --to-ports 50000-55000
```



Dezavantaje NAT



În cazul PAT comunicația nu poate fi inițiată de o stație din Internet

Folosește informații de nivel superior pentru a controla un nivel inferior

Întârzie adoptarea IPv6

Îngreunează configurarea tunelurilor

Are dificultăți în gestionarea traficului UDP

Avantaje – Dezavantaje translatate

Avantaje:

- conservarea adreselor
- păstrarea schemei de adresare în LAN, în cazul adresării publice
- libertate și consistență în schema de adresare privată
- securitate sporită în rețea

Dezavantaje:

- reducerea performanțelor rețelei
- conectivitate end-to-end întreruptă
- configurări mai complicate (tunelări)

Configurare NAT router

Mapare statică

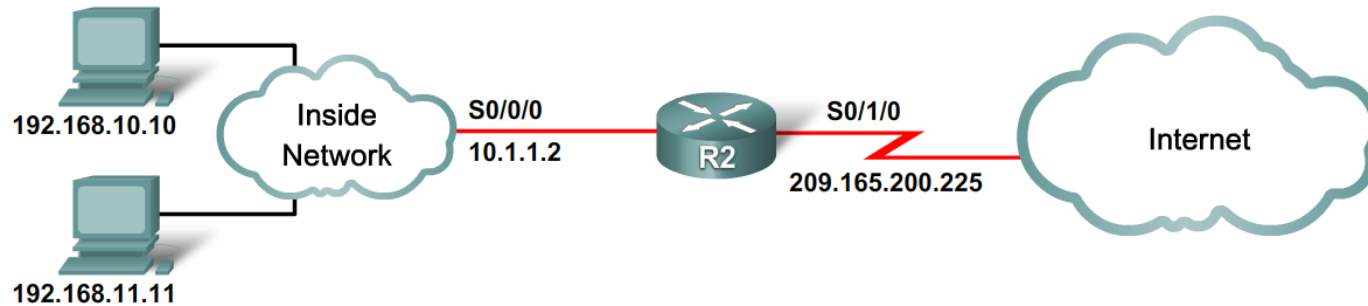
```
R(config)#ip nat inside source static <local-ip> <global-ip>
```

Mapare dinamică

```
R(config)#access-list <number> permit <source-ip> <wildcard>  
R(config)#ip nat pool <name> <start-ip> <end-ip>  
                {netmask <netmask> | prefix-length <prefix>}  
R(config)#ip nat inside source list <number> pool <name>
```

Configurare PAT

NAT Overload Configuration Example



```
access-list 1 permit 192.168.0.0 0.0.255.255
ip nat inside source list 1 interface serial 0/1/0 overload
interface serial 0/0/0
  ip nat inside
interface serial 0/1/0
  ip nat outside
```

```
R(config)#access-list <acl-number> permit <source-ip> <wildcard>
R(config)#ip nat inside source list <acl-number>
interface <outside-interface> overload
```

Aplicarea pe interfețe

Configurarea interfețelor pentru inside:

```
R(config-if)#ip nat inside
```

Configurarea interfețelor pentru outside:

```
R(config-if)#ip nat outside
```


Depanare

Tabela cu translații NAT

```
R#show ip nat translations
```

Statistici despre translații NAT

```
R#show ip nat statistics
```

Ștergerea tabelului de translații dinamice

```
R#clear ip nat translations *
```

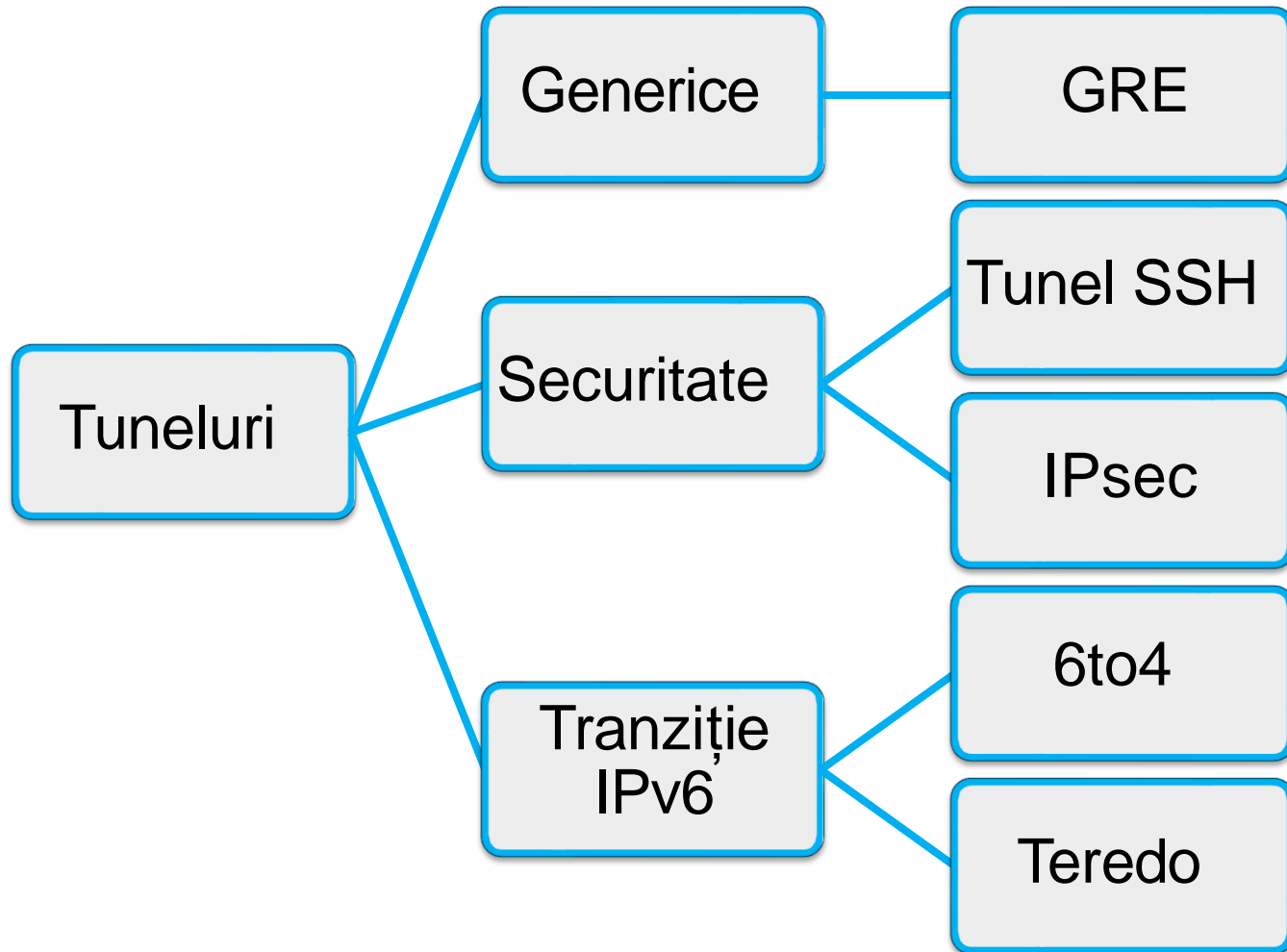
Informații despre fiecare pachet translatat

```
R#debug ip nat
```

Conceptul de tunelare

- Procesul de tunelare constă în încapsularea datelor unui protocol (**payload protocol**) într-un alt protocol (**delivery protocol**)
- **Observație:** Deși IP încapsulează datele TCP și Ethernet încapsulează datele IP, acestea nu sunt considerate exemple de tunelare

Exemple de tuneluri

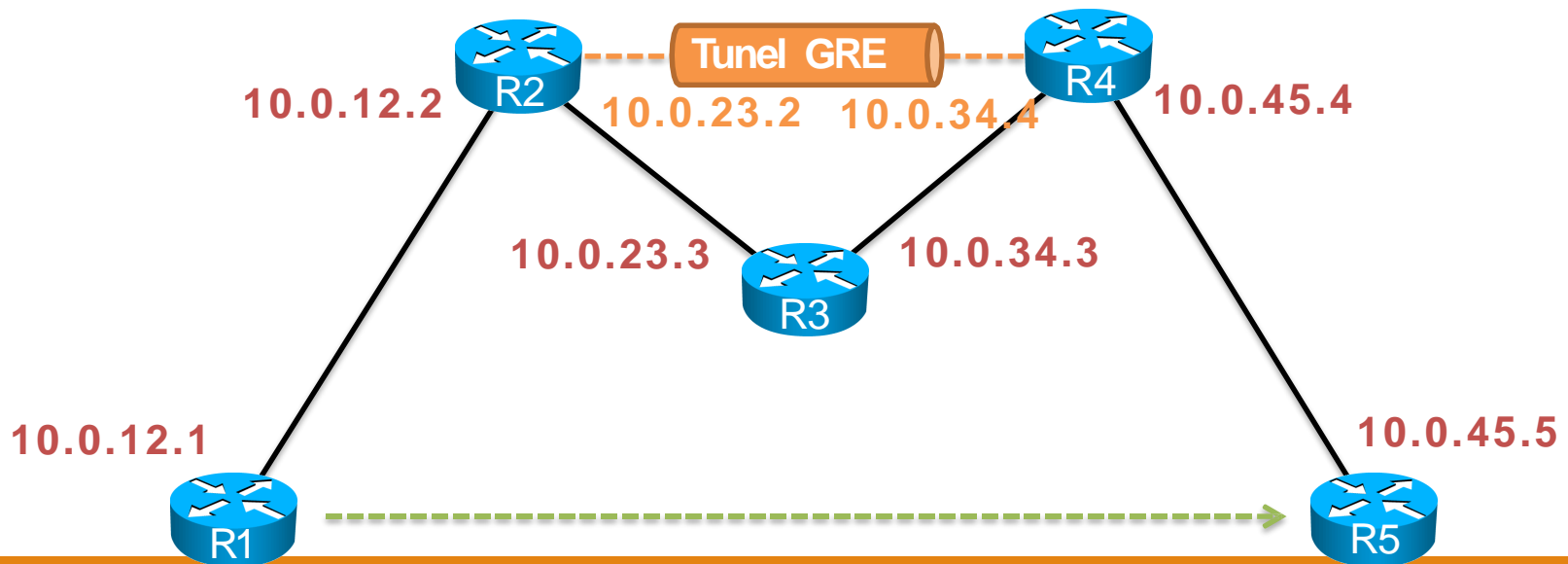


Tunel GRE (Generic Routing Encapsulation)

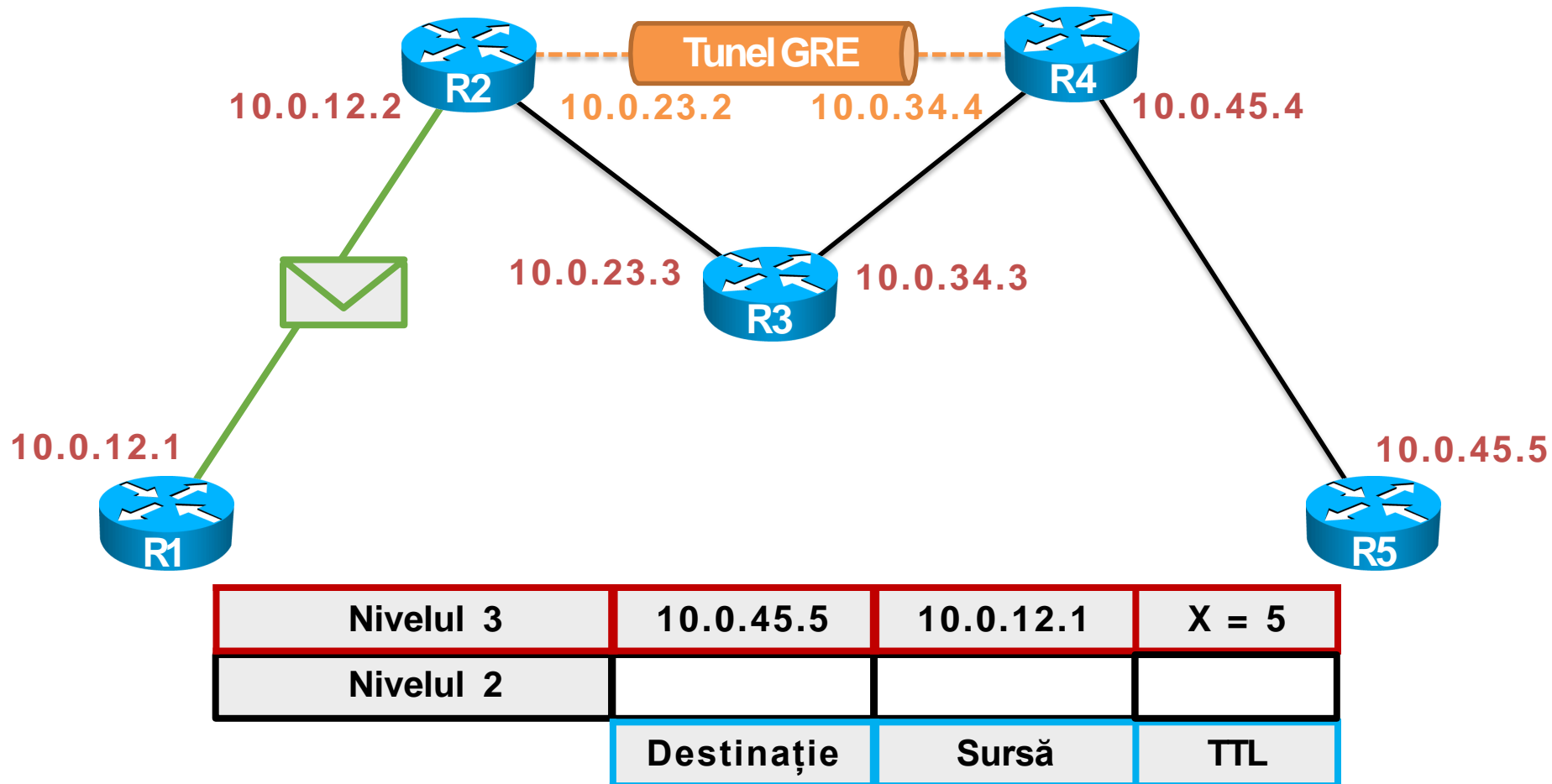
Tunel GRE	
Delivery protocol:	IPv4, IPv6
Payload protocol:	Protocoale de nivel 3
Nivel OSI:	3
Funcție:	Folosit pentru transport de pachete IP fără a fi procesate de ruterele intermediare

Tunel GRE (Generic Routing Encapsulation)

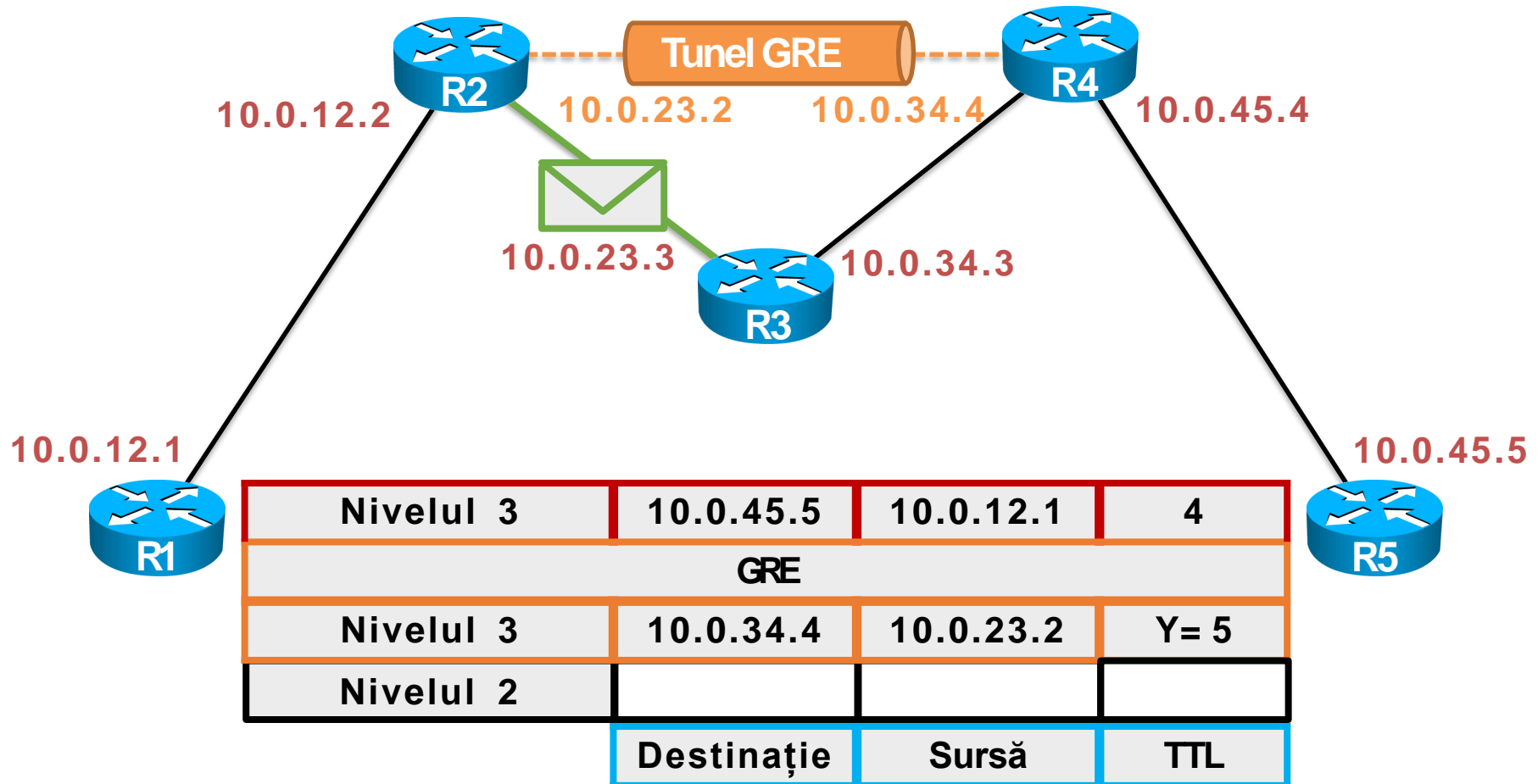
- R1 trimite un pachet către R5
- Între R2 și R4 este configurat un tunel GRE (nu este o legătură fizică)
 - Capetele tunelului sunt reprezentate de IP-urile 10.0.23.2 și 10.0.34.4 de pe interfețele fizice



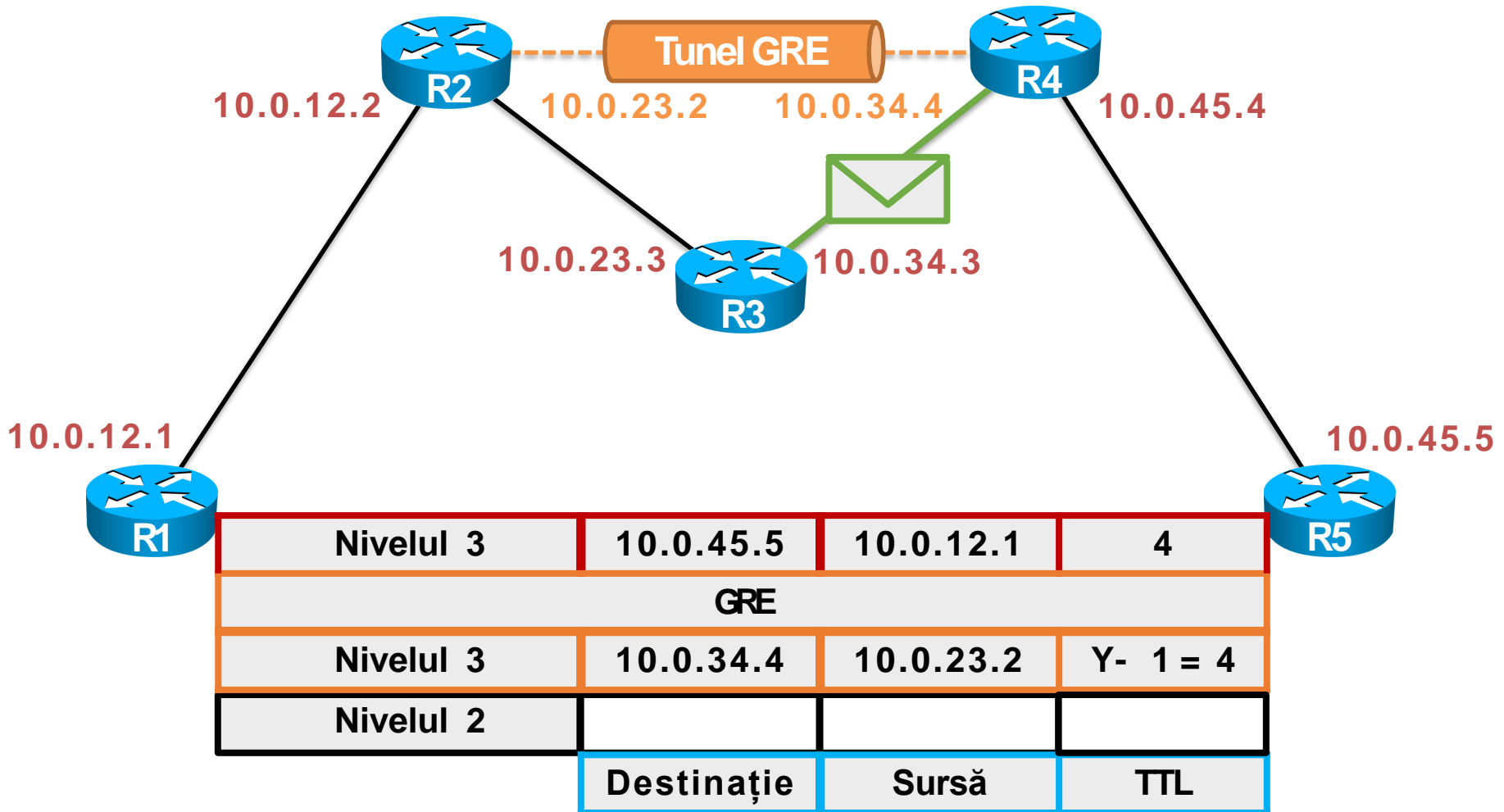
Tunel GRE (Generic Routing Encapsulation)



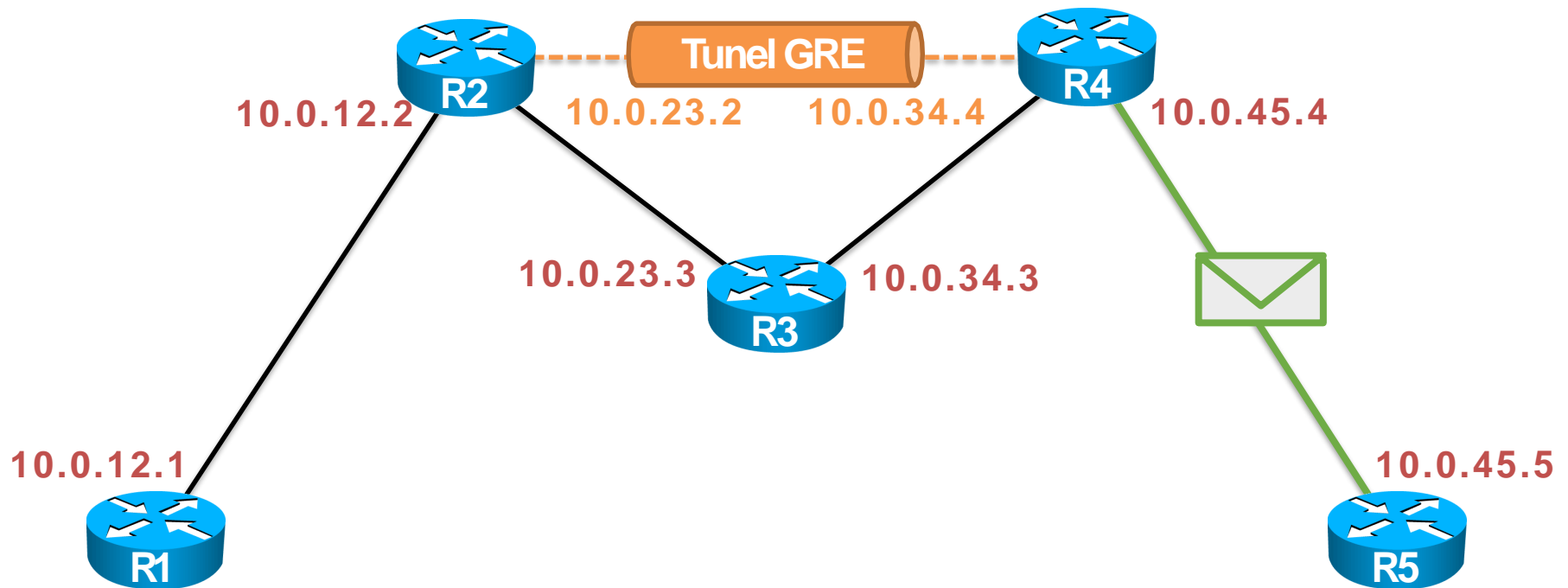
Tunel GRE (Generic Routing Encapsulation)



Tunel GRE (Generic Routing Encapsulation)



Tunel GRE (Generic Routing Encapsulation)



Nivelul 3	10.0.45.5	10.0.12.1	4
Nivelul 2			
	Destinație	Sursă	TTL