

NAT și tunelare

IPv4 are patru miliarde de adrese, iar lumea are de mult mai multe ori mai multe dispozitive. Aritmetica aceasta ar fi trebuit să oprească Internetul prin anul 2000. Nu s-a întâmplat, iar cursul acesta explică de ce: translatarea adreselor a prelungit viața IPv4 cu peste două decenii, cu prețul unei încălcări deliberate a uneia dintre regulile fundamentale ale rețelilor. A doua jumătate a cursului este despre tunelare - arta de a face un protocol să circule printr-o infrastructură care nu îl înțelege, inclusiv IPv6 printr-un Internet care încă vorbește IPv4.

Din capitolul precedent.

- Adresele IP sursă și destinație rămân **constante** de la un capăt la altul. Doar cele MAC se rescriu.
- Adresele **private** din RFC 1918 nu sunt rutabile în Internet.
- În iptables, **lanțul** spune unde pe traseu, iar **tabela** ce fel de decizie.
- **PREROUTING** este înainte de decizia de rutare, **POSTROUTING** după ea.
- Un **socket** este perechea adresă + port; o conexiune e identificată de patru numere.

OBIECTIVELE CAPITOLULUI

- Să explicați de ce s-au epuizat adresele IPv4 și ce a amânat criza
- Să urmăriți un pachet prin translatare, dus și întors
- Să alegeți între NAT static, NAT dinamic și PAT
- Să folosiți corect termenii inside local, inside global, outside local, outside global
- Să configurați NAT pe Cisco și pe Linux
- Să enumerați ce strică NAT-ul și de ce
- Să explicați diferența dintre stratificare și tunelare
- Să scrieți și să citiți o adresă IPv6 și să explicați cum funcționează 6to4

Primul punct este cel pe care îl demolăm astăzi. Tot cursul 6 a insistat că adresele IP nu se schimbă pe traseu - și acum vom vedea un echipament care le schimbă intenționat, la fiecare pachet, în ambele direcții. Nu e o contradicție: e o excepție construită cu grijă, care are propriile consecințe neplăcute.

9.1 Problema: de ce s-au terminat adresele

IPv4 folosește adrese de 32 de biți, deci aproximativ 4,3 miliarde de valori - din care o parte considerabilă este rezervată și nefolosibilă. Ultimele blocuri libere au fost distribuite de IANA în **februarie 2011**.

Nu numărul a fost problema, ci risipa. Alocarea inițială se făcea **pe clase**, iar clasele veneau în trei mărimi: 256, 65 536 sau 16 milioane de adrese. Nimic între ele.

O organizație care avea nevoie de 300 de adrese nu încăpea într-o clasă C, deci primea o clasă **B** - și lăsa nefolosite 65 000 de adrese. Universități întregi au primit clase A, cu 16 milioane de adrese fiecare, în anii '80.

Deci: patru miliarde de adrese, dintre care poate un sfert au ajuns efectiv la un dispozitiv. Restul s-au pierdut în rotunjiri.

Trei mecanisme au amânat criza, iar toate trei sunt încă în uz:

Tabelul 9.1 Problema: de ce s-au terminat adresele.

Mecanism	Ce face	Unde l-am văzut
CIDR	elimină clasele; permite alocări de dimensiune potrivită (/22, /28...)	cursul 5
Adrese private (RFC 1918)	trei blocuri re folosibile de oricăre organizații, în paralel	cursul 5
NAT	face adresele private utilizabile în Internet	astăzi

Tabelul 9.2 Problema: de ce s-au terminat adresele.

Bloc privat	Prefix	Adrese	Unde apare tipic
10.0.0.0 – 10.255.255.255	/8	16,7 milioane	rețele mari de întreprindere
172.16.0.0 – 172.31.255.255	/12	1 milion	rețele medii, laboratoare
192.168.0.0 – 192.168.255.255	/16	65 536	rețele mici, ruterul de acasă

De ce funcționează re folosirea. Adresa 192.168.1.1 există, chiar acum, în milioane de locuințe simultan. Nu e nicio problemă, pentru că niciun ruter din Internet nu are rută spre ea: blocurile private sunt filtrate explicit la marginea oricărei rețele serioase.

Prețul: ca să iasă în Internet, aceste adrese trebuie **înlocuite** cu una publică. Exact asta face NAT.

9.2 Ce face, concret, traducerea

DEFINIȚIA 9.1 (NAT - NETWORK ADDRESS TRANSLATION). Schimbarea adresei IP sursă sau destinație a unui pachet la trecerea printr-un ruter. În mod normal, adresele IP rămân neschimbate de la un capăt la altul; NAT încalcă **deliberat** această regulă. Pentru ca legătura să funcționeze, traducerea trebuie să aibă loc **în ambele direcții** - altfel răspunsul nu ar mai nimeri stația inițitoare.

Ruterul ține evidența traducerilor într-o **tabelă NAT**, construită fie static de administrator, fie dinamic prin inspectarea traficului care trece.



Figura 9.1 Un pachet prin NAT, dus și întors.

9.3 Cele trei tipuri de traducare

NAT static - unu la unu

Problema: un server are adresă privată, dar trebuie să fie accesibil din exterior printr-o adresă publică fixă.

Soluția: o asociere **permanentă** între adresa internă și una publică rezervată. Traducerea funcționează în ambele sensuri, deci conexiunile pot fi inițiate și dinspre Internet - este singurul tip la care asta merge fără configurări suplimentare.

NAT dinamic - n la m

Problema: 40 de stații interne, dar doar 20 de adrese publice disponibile.

Soluția: un **rezervor** (pool) de adrese publice, din care stațiile primesc temporar câte una, atât timp cât comunică. Când rezervorul se golește, stațiile următoare pur și simplu nu mai ies - comportament care se depanează greu, pentru că depinde de ora din zi. Conexiunile din exterior nu pot fi inițiate, pentru că nu se știe dinainte ce adresă are cine.

PAT - n la unu

Problema: 40 de stații interne și o singură adresă publică.

Soluția: **PAT** (Port Address Translation), numit și NAT overload la Cisco și **MASQUERADE** în Linux. Pe lângă adresă, ruterul translatează și **portul sursă**, folosindu-l ca identificator al conversației. Când răspunsul se întoarce, portul destinație spune rutelui cărei stații îi aparține.



Figura 9.2 Simulator PAT: o singură adresă publică pentru toată rețeaua.

Deschideți mai multe conexiuni de la stații diferite și observați coloana din mijloc: adresa publică este mereu aceeași, dar portul diferă. Aceasta este întreaga idee.



Figura 9.3 Câte stații poate ascunde o adresă publică?

Cu 16 biți de port, o singură adresă publică susține teoretic peste 64 000 de conversații - practic, câteva sute de stații, pentru că un browser modern deschide zeci de conexiuni pentru o singură pagină. Aceasta este, în cifre, explicația pentru care planeta a

rămas pe IPv4 mult mai mult decât se estima în 1995.

Tabelul 9.3 PAT - n la unu.

	NAT static	NAT dinamic	PAT
Raport adrese	1 : 1	n : m	n : 1
Asocierea	permanentă	temporară, cât durează sesiunea	temporară, per conexiune
Translatează portul	nu	nu	da
Inițiere din exterior	da	nu	nu, fără port forwarding
Utilizare tipică	servere publicate	rar; a fost înlocuit de PAT	practic orice rețea de acces

9.4 Cei patru termeni care încurcă pe toată lumea

Documentația Cisco folosește patru denumiri pentru adresele implicate. Sunt logice odată înțelese, dar până atunci produc o confuzie remarcabilă.

Cheia: două întrebări separate. Inside sau outside răspunde la: a cui este gazda? A rețelei noastre, sau a Internetului? Se referă la proprietar, nu la poziția adresei.

Local sau global răspunde la: *de unde privesc?* Din interiorul rețelei, sau din Internet?

Deci „inside local” înseamnă *gazda noastră, văzută dinăuntru*, iar „inside global” înseamnă *aceeași gazdă, văzută din Internet*. Sunt două nume pentru același calculator.

Tabelul 9.4 Cei patru termeni care încurcă pe toată lumea.

Termen	A cui gazdă	Văzută de unde	Exemplu
Inside local	a noastră	dinăuntru	192.168.1.10 - adresa privată reală
Inside global	a noastră	din Internet	86.120.5.7 - adresa publică translatată
Outside global	a lor	din Internet	93.184.216.34 - adresa reală a serverului
Outside local	a lor	dinăuntru	de obicei <i>aceeași</i> ; diferă doar la NAT dublu

Regula practică. În 95% din cazuri contează doar primele două, iar `show ip nat translations` le afișează exact în această ordine. Când citești tabela, coloana „inside local” este stația reală, iar „inside global” este masca ei publică.

9.5 NAT pe echipamente Cisco

Configurarea are întotdeauna **trei** componente, iar lipsa oricăreia produce un NAT care nu funcționează și nu dă nicio eroare.

- 1. **Ce se translatează** - un ACL sau o asociere statică.
- 2. **În ce se translatează** - un rezervor de adrese, sau o interfață.
- 3. **Care interfață e „inside” și care „outside”** - pasul uitat cel mai des.

```
! 1. ce se translateaza - un ACL obisnuit, dar aici permit inseamna
!   "acesta e traficul despre care vorbesc", nu "lasa-l sa treaca"
R1(config)# access-list 1 permit 192.168.1.0 0.0.0.255

! 2. in ce se translateaza - adresa interfetei de iesire, oricare ar fi ea
R1(config)# ip nat inside source list 1 interface gigabitEthernet 0/1 overload

! 3. cine e inauntru si cine e afara
R1(config)# interface gigabitEthernet 0/0
R1(config-if)# ip nat inside
R1(config)# interface gigabitEthernet 0/1
R1(config-if)# ip nat outside
```

Listingul 9.1 PAT: toată rețeaua, prin adresa interfeței de ieșire copiază

```
! serverul 192.168.1.100 este vizibil din Internet ca 86.120.5.8
R1(config)# ip nat inside source static 192.168.1.100 86.120.5.8

! doar portul 80, daca nu vrem sa expunem tot serverul
R1(config)# ip nat inside source static tcp 192.168.1.100 80 86.120.5.8 80
```

Listingul 9.2 NAT static: publicarea unui server intern copiază



Figura 9.4 Terminal: verificarea NAT.

Priviți ultima linie din prima comandă: are doar două coloane completate și niciun protocol. Este asocierea **statică** - permanentă, existentă chiar dacă nimeni nu comunică. Celelalte patru sunt dinamice și dispar la expirarea temporizatorului.

Simptomul clasic: „NAT-ul nu merge deloc”. Dacă `show ip nat translations` este **gol** deși traficul circulă, aproape sigur lipsește pasul 3: interfețele nu au fost marcate `ip nat inside / ip nat outside`.

Dacă tabela are intrări, dar ping-ul tot nu merge, problema nu mai e la NAT - verificați ruta implicită și dacă furnizorul chiar rutează adresa publică folosită.

9.6 NAT în Linux

Translatarea se configurează cu aceleași unelte ca filtrarea, dar în tabela `nat`. Regula de aur este să știți în ce lanț intrați - și motivul este pur logic:

Tabelul 9.5 NAT în Linux.

Ce rescriem	Lanț	Țintă	De ce acolo
adresa sursă	POSTROUTING	SNAT	după decizia de rutare - ruta se alege pe baza adresei reale
adresa destinație	PREROUTING	DNAT	înainte de decizia de rutare - altfel ruterul ar ruta spre adresa greșită
sursa, cu IP-ul interfeței	POSTROUTING	MASQUERADE	când adresa publică este obținută prin DHCP și se poate schimba

De ce DNAT trebuie să fie înainte de rutare. Pachetul sosește adresat lui `86.120.5.7` - adresa ruterului. Dacă ruterul ar decide ruta întâi, ar concluziona „e pentru mine” și l-ar da procesului local. Prea târziu.

Rescriind destinația **înainte**, pachetul ajunge la decizia de rutare deja adresat lui `192.168.1.100`, iar ruterul îl rutează normal spre LAN.

Simetric, SNAT trebuie să fie **după**: dacă am rescrie sursa mai devreme, nu ar schimba nimic la rutare (ruta depinde de destinație), dar am pierde informația despre interfața pe care va ieși pachetul - exact informația de care MASQUERADE are nevoie.

```
# iesire: rescrie sursa privata cu adresa publica rezervata
iptables -t nat -A POSTROUTING -s 192.168.1.100 -j SNAT --to-source 141.85.200.1

# intrare: rescrie destinatia publica cu adresa privata a serverului
iptables -t nat -A PREROUTING -d 141.85.200.1 -j DNAT --to-destination 192.168.1.100

# si, obligatoriu, activarea rutarii in nucleu
sysctl -w net.ipv4.ip_forward=1
```

Listingul 9.3 NAT static, în ambele sensuri copiază

SNAT nu înseamnă „static NAT”. Prescurtarea vine de la **Source** NAT. Este o confuzie care persistă în multe cursuri și în multe forumuri. Perechea ei este DNAT - **Destination** NAT. Un SNAT poate fi perfect dinamic.

```
# rezervor de adrese publice
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT \
    --to-source 141.85.200.2-141.85.200.6

# PAT cu adresa interfetei de iesire, oricare ar fi ea
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Listingul 9.4 NAT dinamic și PAT copiază

SNAT sau MASQUERADE? SNAT cere adresa publică scrisă explicit - e mai rapid, pentru că nucleul nu trebuie să o caute la fiecare pachet. Se folosește când adresa este fixă.

MASQUERADE citește adresa curentă a interfeței de ieșire. E puțin mai lent, dar supraviețuiește unei schimbări de adresă - și de aceea e standardul pe conexiunile casnice, unde furnizorul dă adresa prin DHCP.

Exercițiu rezolvat 9.1. Este vreo problemă cu următorul set de reguli?

reguli

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT --to-source 141.85.200.2-141.85.200.6
iptables -t nat -A POSTROUTING -s 192.168.1.100 -j SNAT --to-source 141.85.200.1
```

Rezolvare. Da. A doua regulă **nu se va aplica niciodată**: adresa 192.168.1.100 face parte din 192.168.1.0/24, deci se potrivește cu prima regulă, care are ținută SNAT - o ținută terminală - și încheie procesarea lanțului.

Soluția este inversarea ordinii: regula mai specifică deasupra.

Este exact aceeași lecție ca la listele de acces din cursul 7, într-un context complet diferit: **ordinea regulilor este parte din semantica lor**. O veți întâlni a treia oară la rutare, sub numele de longest prefix match.

Port forwarding

Când un server intern trebuie expus doar pentru o anumită aplicație, se rescrie destinația pe baza portului. Este exact ce faceți în interfața ruterului de acasă când „deschideți un port”.

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
-j DNAT --to-destination 192.168.1.100

# regula de filtrare trebuie sa permita si ea traficul!
iptables -A FORWARD -i eth0 -o eth1 -d 192.168.1.100 -p tcp --dport 80 -j ACCEPT

# necesar doar daca in retea exista mai multe gateway-uri:
iptables -t nat -A POSTROUTING -o eth1 -p tcp --dport 80 -d 192.168.1.100 \
-j SNAT --to-source 192.168.1.1
```

Listingul 9.5 portul 80 al ruterului duce la serverul intern copiază

NAT nu înlocuiește filtrarea. Tabela nat și tabela filter sunt independente. Un DNAT corect duce pachetul la serverul intern - dar dacă politica lanțului FORWARD este DROP și nu există o regulă explicită, pachetul e aruncat imediat după traducere.

Simptom: `conntrack -L` arată conexiunea, dar serverul nu vede nimic.

9.7 Ce ne costă NAT

Dezavantajele translării

- În cazul PAT, o stație din Internet **nu poate iniția** comunicația - ceea ce a rupt modelul peer-to-peer al Internetului original
- Folosește informație de **nivel 4** (portul) pentru a controla **nivelul 3** - o încălcare directă a stratificării
- Îngreunează protocoalele care transportă adrese IP **în interiorul datelor**: FTP activ, SIP, unele jocuri. Ruterul trebuie să inspecțeze și să rescrie și încărcătura utilă, cu module dedicate (`nf_conntrack_ftp`)
- Gestionează dificil traficul UDP, care nu are noțiunea de conexiune, deci nici de sfârșit al ei - intrările expiră pe temporizator, uneori prea devreme
- **Rupe trasabilitatea**: în jurnalele unui server, sute de utilizatori apar cu aceeași adresă
- A **întârziat adoptarea IPv6**, oferind o soluție „suficient de bună” exact atunci când presiunea de migrare ar fi trebuit să crească

CGNAT - când și adresa publică e împărțită. Mulți furnizori nu mai dau nici măcar o adresă publică per abonat: pun un al doilea nivel de traducere în rețeaua lor (**Carrier-Grade NAT**), iar clientul primește o adresă din 100.64.0.0/10 - nici publică, nici privată clasică.

Consecința pentru dumneavoastră: port forwarding-ul de acasă **nu mai funcționează**, oricât l-ați configura corect. Ruterul dumneavoastră nu deține adresa publică pe care ar trebui să o expună.

Merită spus și partea bună, pentru că se invocă des ca argument: NAT oferă, ca efect secundar, o formă de protecție - nimic nu ajunge la stațiile interne fără să fi fost cerut. Nu este însă un firewall și nu înlocuiește unul: nu filtrează nimic din ce iese, nu inspectează conținutul și nu vă apără de o stație internă compromisă.

9.8 Conceptul de tunelare

DEFINIȚIA 9.2 (TUNELARE). Încapsularea datelor unui protocol (**payload protocol**) într-un alt protocol (**delivery protocol**), astfel încât primul să poată traversa o infrastructură care nu îl suportă.

Când e stratificare și când e tunelare. Faptul că IP încapsulează TCP, iar Ethernet încapsulează IP, nu este tunelare - este stratificarea normală, în care fiecare nivel îl poartă pe cel de deasupra.

Tunelarea apare când un protocol este încapsulat într-unul de **același nivel sau de nivel mai înalt**: IPv6 în IPv4 (3 în 3), Ethernet în IP (2 în 3), TCP în SSH (4 în 7).

Testul rapid: dacă în pachet apar **două antete de același tip** - două antete IP, de exemplu - este tunelare.

Analogie Vreți să trimiteți o scrisoare într-o limbă pe care poșta dintr-o anumită țară o refuză. O puneți într-un plic nou, cu adresa scrisă în limba acceptată, și o trimiteți unui prieten de acolo. El deschide plicul exterior și pune scrisoarea originală în poșta locală.

Poșta intermediară nu a știut niciodată ce transportă. Prietenul este capătul tunelului.

Tabelul 9.6 Conceptul de tunelare.

Tunel	Transportă	Prin	Nivel	Scop
GRE	protocoale de nivel 2 și 3	IPv4 sau IPv6	3	legături logice între rutere
SSH	protocoale de nivel 4	SSH	7	transport securizat
IPsec	IP	IP	3	VPN cu confidențialitate și integritate
L2TP	PPP, ATM, Frame Relay	UDP	2	legături PPP peste infrastructură IP
6to4, Teredo	IPv6	IPv4, respectiv UDP	3	tranziția spre IPv6

Tunel ≠ criptare. GRE nu criptează absolut nimic: oricine vede traficul poate citi pachetul din interior. Tunelul rezolvă problema **conectivității**, nu pe cea a confidențialității.

Când e nevoie de amândouă, se combină: GRE peste IPsec este configurația clasică de VPN între sedii.

9.9 Tunelul GRE

GRE (Generic Routing Encapsulation) creează o legătură logică între două rutere care nu sunt vecine fizic. Ruterele intermediare nu știu nimic despre conținutul tunelului: pentru ele, este un simplu pachet IP adresat capătului celălalt.

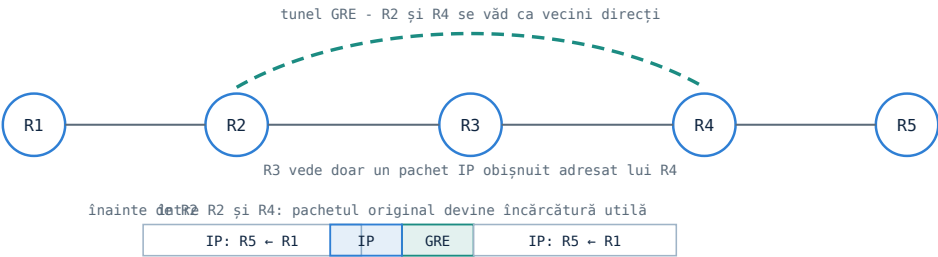


Figura 9.5 Încapsularea GRE. Pachetul original își păstrează adresele intacte; peste el se pun un antet GRE și un nou antet IP, cu adresele capetelor tunelului. Fiecare antet IP are propriul TTL, decrementat independent.

```
! pe R2
R2(config)# interface tunnel 0
R2(config-if)# ip address 172.16.0.1 255.255.255.252
R2(config-if)# tunnel source 203.0.113.2
R2(config-if)# tunnel destination 198.51.100.4
R2(config-if)# tunnel mode gre ip

! pe R4 - oglinda exacta
R4(config)# interface tunnel 0
R4(config-if)# ip address 172.16.0.2 255.255.255.252
R4(config-if)# tunnel source 198.51.100.4
R4(config-if)# tunnel destination 203.0.113.2
```

Listingul 9.6 configurarea unui tunel GRE, pe ambele capete copiază

De aici înainte, `tunnel 0` se comportă ca orice interfață: are adresă IP, apare în tabela de rutare, poate participa la OSPF. Ruterele R2 și R4 sunt, din punctul de vedere al rutării, **vecini direcți** - deși între ele sunt kilometri de Internet.

Problema MTU, care apare mereu. Antetele suplimentare - 20 de octeți IP plus 4 sau 8 de GRE - **consumă** din cei 1500 de octeți disponibili. Un pachet de 1500 de octeți nu mai încape după încăpsulare.

Simptomul e caracteristic și înșelător: ping-ul merge, sesiunile SSH merg, dar transferurile mari și paginile web mari se blochează. Pachetele mici trec, cele mari nu.

Soluția uzuală: `ip mtu 1400` și `ip tcp adjust-mss 1360` pe interfața de tunel.

9.10 Tunelul SSH

Un tunel SSH redirectează un port local printr-o sesiune SSH deja criptată. Scenariu tipic: Alice are cont pe un ruter Linux accesibil din Internet și vrea să administreze prin Telnet un server vechi din spatele lui, fără să își trimită parola în clar prin Internet.

```
ssh -N -L 0.0.0.0:5000:141.85.200.1:23 alice@141.85.200.19

# -N nu deschide shell pe masina intermediara
# -L tunel local: portul de intrare este pe masina locala
# 0.0.0.0:5000 adresa si portul pe care se asculta local
# 141.85.200.1:23 destinatia finala, vazuta din masina intermediara
```

Listingul 9.7 redirectare locală de port copiază

După comandă, orice conexiune la `127.0.0.1:5000` de pe mașina lui Alice iese criptată spre serverul SSH și, de acolo, spre `141.85.200.1:23`. Aplicația lui Alice crede că vorbește cu un serviciu local.

Tabelul 9.7 Tunelul SSH.

Formă	Cine ascultă	Scenariu
-L port:gazdă:port	mașina locală	ajung eu la un serviciu din rețeaua îndepărtată
-R port:gazdă:port	mașina îndepărtată	expun un serviciu al meu spre acea rețea
-D port	local, ca proxy SOCKS	tot traficul unei aplicații, prin tunel

Unde se termină criptarea. Traficul este protejat doar între mașina lui Alice și ruterul intermediar. De la ruter la serverul Telnet circulă în clar.

Un tunel securizează exact porțiunea pe care o traversează, niciun metru mai mult. Observația este valabilă pentru orice VPN - inclusiv pentru cele comerciale, care vă protejează până la serverul lor și nu mai departe.

9.11 IPv6, pe scurt

IPv6 nu este o versiune îmbunătățită a lui IPv4, ci un protocol nou, gândit să rezolve exact neajunsurile lui.

Ce nu merge la IPv4

- Adrese insuficiente

- Antet complicat, cu lungime variabilă
- Sumă de control recalculată la fiecare ruter
- Fragmentare făcută de rutere, costisitoare
- NAT introduce mai multe probleme decât rezolvă

Ce aduce IPv6.

- 128 de biți de adresă - $3,4 \times 10^{38}$ valori
- Antet fix de 40 de octeți, rapid de procesat
- Fără sumă de control în antet - o face nivelul 4
- Autoconfigurarea adreselor (**SLAAC**), fără DHCP
- Multicast simplificat; **broadcast-ul nu mai există**

Cât de mari sunt 128 de biți. $3,4 \times 10^{38}$ adrese înseamnă aproximativ 6×10^{23} adrese pentru **fiecare metru pătrat** al suprafeței Pământului. Nu este o cifră care se poate epuiza prin risipă.

De aceea IPv6 își permite ceva ce ar fi fost de neconceput la IPv4: fiecare segment de rețea primește un /64 - adică 18 miliarde de miliarde de adrese - indiferent dacă are două stații sau două mii. Subnetarea nu mai este un exercițiu de economie.

Scrierea adreselor

O adresă IPv6 se scrie ca opt grupuri de câte patru cifre hexazecimale. Există două reguli de prescurtare: zerourile din **fața** fiecărui grup se pot omite, iar **un singur** și continuu de grupuri nule se poate înlocui cu ::.



Figura 9.6 Comprimare și expandare de adrese IPv6.

Încercați fe80::1, ::1 și 2002:8d55:c813::1 - ultimul este o adresă 6to4, pe care o veți recunoaște imediat după explicația din secțiunea următoare.

De ce doar un singur . Dacă ar fi permise două, adresa ar deveni ambiguă: 2001::25::1 nu ar spune câte grupuri nule merg în stânga și câte în dreapta. Cu un singur ::, restul se deduce prin scădere din opt.

Tabelul 9.8 Scrierea adreselor.

Prefix	Tip	Rol
::1/128	loopback	testarea propriei stive - echivalentul lui 127.0.0.1
2000::/3	global unicast	adrese rutabile în Internet
FE80::/10	link-local	comunicație în același segment; generată automat pe fiecare interfață
FC00::/7	unique local	echivalentul adreselor private din RFC 1918
FF00::/8	multicast	transmisii către un grup
-	broadcast	nu există ; a fost înlocuit de multicast pe grupuri bine definite

Convenția practică este ca primii 64 de biți să fie partea de rețea și ultimii 64 partea de interfață. Subnetarea funcționează identic cu IPv4 la nivel de bit, doar că spațiul e atât de mare încât se lucrează cu prefixe fixe: /48 pentru un site, /64 pentru un segment.

Ce dispare odată cu IPv6. Fără broadcast, **ARP nu mai există** - este înlocuit de Neighbor Discovery, care folosește multicast și deci nu deranjează toate stațiile din segment. Fără penurie de adrese, **NAT-ul nu mai are rost**, iar modelul peer-to-peer redevine posibil.

Ironia: tocmai NAT-ul, care a salvat IPv4, este motivul principal pentru care IPv6 a fost adoptat atât de încet.

9.12 Tunelul 6to4

Migrarea la IPv6 se face treptat: apar insule IPv6 într-o magistrală care rămâne IPv4. Tunelurile statice GRE funcționează, dar trebuie configurate manual, capăt cu capăt. Tunelul **6to4** se construiește automat, printr-un truc elegant de adresare.

1. Adresele insulei IPv6 trebuie să facă parte din 2002::/16.

2. Următorii 32 de biți ai adresei sunt **exact adresa IPv4 publică** a ruterului de la ieșirea insulei, scrisă hexazecimal. Pentru 141.85.200.19 aceasta înseamnă 8D:55:C8:13, deci prefixul 2002:8D55:C813::/48.
3. Când un ruter de capăt primește un pachet IPv6 spre o astfel de adresă, **extrage din ea** adresa IPv4 a destinației și încapsulează pachetul într-un pachet IPv4 spre acea adresă.
4. Ruterele intermediare rutează un pachet IPv4 obișnuit; câmpul Protocol are valoarea 41, adică „IPv6 încapsulat”.
5. Ruterul de la celălalt capăt recunoaște valoarea 41, decapsulează și livrează pachetul IPv6 în insula lui.

Exercițiu rezolvat 9.2. Care este prefixul 6to4 al unui sit cu adresa publică 192.0.2.129?

Rezolvare. Convertim fiecare octet în hexazecimal:

192 = C0 · 0 = 00 · 2 = 02 · 129 = 81

Grupăm câte doi octeți și lipim după 2002::

2002:C000:0281::/48

Verificare: prefixul are 16 biți de 2002 plus 32 de biți de adresă IPv4 = 48. Rămân 16 biți pentru subrețelele sitului, adică 65 536 de segmente /64. Suficient.

Ideea de reținut. 6to4 nu are nevoie de nicio configurare a capătului celălalt, pentru că adresa IPv4 a destinației este **conținută în adresa IPv6**. Este un exemplu excelent de proiectare în care informația de rutare este codificată chiar în identificator.

În practică, 6to4 a fost în mare parte abandonat (RFC 7526) în favoarea stivei duale - dar ideea rămâne instructivă.

9.13 Greșeli frecvente

Greșeli frecvente.

„NAT-ul e configurat, dar tabela e goală”

Lipsește marcasele ip nat inside / ip nat outside de pe interfețe.

Fără ele, ruterul nu știe în ce direcție să translateze. Sunt al treilea pas, cel uitat.

„Am pus regula specifică după cea generală”

SNAT și DNAT sunt ținte terminale: prima potrivire câștigă și lanțul se oprește.

Specificul deasupra generalului. Aceeași regulă ca la ACL-uri.

„Port forwarding-ul nu merge, deși DNAT-ul e corect”

Tabela nat a mutat pachetul, dar tabela filter l-a aruncat imediat după.

Adăugați regula pe FORWARD. Cele două tabele nu se cunosc între ele.

„Tunelul GRE e sus, ping-ul merge, dar site-urile nu se încarcă”

Problemă de MTU: pachetele mici trec, cele mari nu încap după încapsulare.

ip mtu 1400 și ip tcp adjust-mss 1360 pe interfața de tunel.

„Am crezut că tunelul îmi criptează traficul”

GRE nu criptează nimic. Nici 6to4, nici L2TP simplu.

Pentru confidențialitate: IPsec, SSH sau WireGuard. Tunelul rezolvă conectivitatea, nu secretul.

„Am scris două :: într-o adresă IPv6”

Adresa devine ambiguă și e respinsă de orice implementare.

Un singur ::, pentru cel mai lung șir de grupuri nule.

„Port forwarding-ul de acasă nu merge deloc”

Furnizorul folosește CGNAT: adresa dumneavoastră „publică” e din 100.64.0.0/10.

Verificați ce adresă are interfața WAN a ruterului. Dacă e din acel bloc, soluția e o adresă publică de la furnizor sau un tunel către un server extern.

9.14 Rezumat și glosar

Ce trebuie să rămână.

- Criza IPv4 a venit din **risipa** alocării pe clase, nu doar din numărul de adrese.
- NAT rescrie adresele **în ambele direcții** și ține evidența într-o **tabelă**.
- **PAT** translatează și portul, deci n stații pot împărți o adresă publică.
- **Inside/outside** = a cui e gazda. **Local/global** = de unde privim.
- DNAT în **PREROUTING**, SNAT în **POSTROUTING** - din motive de ordine față de rutare.
- NAT rupe peer-to-peer, încurcă protocoalele cu adrese în date și nu este un firewall.
- **Tunelare** = încapsulare într-un protocol de același nivel sau mai înalt.
- Un tunel **nu criptează** automat; și securizează doar porțiunea pe care o traversează.
- IPv6 are 128 de biți, antet fix, fără broadcast și fără nevoie de NAT.
- **6to4** codifică adresa IPv4 chiar în adresa IPv6, deci tunelul se construiește singur.

Exerciții

1. De ce PAT poate deservi zeci de mii de conexiuni cu o singura adresa publica?
 - a. Pentru ca adresele private sunt nelimitate
 - b. Pentru ca portul, pe 16 biți, devine identificator al fiecărei conversații
 - c. Pentru ca ruterul comprimă pachetele
 - d. Pentru ca folosește IPv6 în fundal
2. În ce lanț iptables se rescrie adresa destinație și de ce?
 - a. POSTROUTING, pentru că este ultimul
 - b. PREROUTING, pentru că decizia de rutare trebuie luată pe baza adresei finale
 - c. INPUT, pentru că pachetul intră în mașină
 - d. FORWARD, pentru că traficul tranzitează
3. Ce înseamnă inside global?
 - a. Adresa privată a unei stații din rețeaua noastră
 - b. Adresa publică sub care apare în Internet o stație din rețeaua noastră
 - c. Adresa unui server din Internet
 - d. Adresa de loopback a ruterului
4. show ip nat translations este gol, deși traficul circulă. Ce verificați întâi?
 - a. Dacă ruterul are licența de NAT
 - b. Dacă interfețele au fost marcate cu ip nat inside și ip nat outside
 - c. Dacă ACL-ul are deny la final
 - d. Dacă există ruta implicită
5. Care este diferența esențială dintre stratificare și tunelare?
 - a. Nu există nicio diferență
 - b. Tunelarea încapsulează un protocol într-unul de același nivel sau de nivel mai înalt, pentru a traversa o infrastructură care nu îl suportă
 - c. Tunelarea folosește doar UDP
 - d. Stratificarea presupune criptare
6. Un tunel GRE este sus, ping-ul funcționează, dar transferurile mari se blochează. Cauza probabilă?
 - a. Ruta implicită lipsește
 - b. MTU: antetele suplimentare nu mai lasă loc pachetelor de 1500 de octeți
 - c. Tunelul nu este criptat
 - d. Adresa de tunel este gresită

7. Ce prefix 6to4 corespunde adresei IPv4 publice 141.85.200.19?
 - a. 2001:8D55:C813::/48
 - b. 2002:8D55:C813::/48
 - c. 2002:141:85:200::/48
 - d. FE80:8D55:C813::/48
8. Un tunel SSH între stația A și routerul R protejează traficul dintre R și serverul final?
 - a. Da, tot traseul este criptat
 - b. Nu, criptarea există doar între A și R
 - c. Da, dacă serverul are IPv6
 - d. Doar dacă se folosește UDP
9. De ce nu se poate scrie o adresă IPv6 cu două grupuri :: ?
 - a. Pentru că ar fi prea lungă
 - b. Pentru că adresa ar deveni ambiguă: nu s-ar ști câte grupuri nule merg de fiecare parte
 - c. Pentru că hexazecimalul nu permite
 - d. Se poate, dar doar în adrese link-local

Note bibliografice

Cursul următor revine la rutare, dar la varianta care se întreține singură: protocoalele **link-state** și OSPF. Veți vedea de ce, într-o rețea cu douăzeci de routere, rutele statice devin imposibil de administrat - și cum reușesc ruterele să își construiască singure o hartă completă a topologiei.

Laboratorul 6 configurează ACL-uri, NAT și PAT pe routerul de margine al firmei - exact combinația din ultimele două cursuri.

- RFC 3022 - NAT tradițional
- RFC 2784 - Generic Routing Encapsulation
- RFC 3056 - conectarea domeniilor IPv6 prin nori IPv4 (6to4)
- RFC 7526 - retragerea 6to4 din practica recomandată
- RFC 8200 - Internet Protocol, versiunea 6
- RFC 6598 - spațiul de adrese pentru CGNAT

Extras din volumul „Rețele Locale - Note de curs și lucrări de laborator”. Varianta online a acestui capitol conține figurile în formă interactivă.