IP tuneli: ip tunnel naredba

IP tuneli su objekti mrežnog sloja OSI modela koji nastaju postupkom tuneliranja - omatanja IP paketa u novi paket na izvorišnom kraju tunela, te izdvajanja unutrašnjeg IP paketa iz primljenog IP paketa na odredišnom kraju tunela.

Primjer korištenja ovakvog tuneliranje je tvrtka koja radi na dvije udaljene lokacije. Računala na obje lokacije treba smjestiti unutar iste privatne mreže (s IP adresama iz nekog od privatnih raspona adresa). Drugi primjer bila bi tvrtka na istoj lokaciji koja koristi dvije privatne mreže, a mi želimo da nam računalo iz one druge privatne mreže bude vidljivo. U ovakvim slučajevima oblikujemo tunel i definiramo krajeve tunela - čvorove koji na originalni IP paket dodaju novo zaglavlje ili s primljenog paketa skidaju vanjsko IP zaglavlje što omogućuje putovanje paketa kroz nekompatibilni adresni ili protokolni prostor.

Protokoli za tuneliranje dijele se na nespojno-bazirane (datagram-based; IP in IP, GRE, SIT, IPsec, PPTP, MPLS i dr.) i spojno-bazirane (stream-based; SSH, TLS). U ovom ćemo članku objasniti IP in IP, GRE i SIT tunele.

IPIP tuneli su najjednostavniji tuneli i podrazumijevaju enkapsulaciju IPv4 paketa unutar IPv4 paketa. To je moguće isključivo za unicast promet. IPIP ne možemo koristiti u multicast okolini, primjerice kod korištenja OSPF, RIP, BGP i sličnih protokola.

GRE (Generic Routing Encapsulation) tuneli (originalno razvijeni i široko upotrebljavani od Cisca), između ostalog, omogućavaju i prenošenje multicast i IPv6 prometa. Nazivaju se još i IPv6/IPv4 over IPv4 tuneli. Tipična je primjena ovih tunela u mrežama s dinamičkim usmjeravanjem.

SIT (Simple Internet Transition) tuneli nazivaju se još i IPv6 over IPv4 tuneli. Nalaze primjenu u povezivanju izdvojenih IPv6 mreža preko IPv4 područja.

Prije stvaranja tunela na Linuxu treba uključiti sljedeće module jezgre: za IPIP modul ipip, za GRE modul ip_gre i za SIT modul ipv6. Pogledajmo sada ispis podataka za sučelja:

```
# modprobe -v ipip
# modprobe -v ipv6
# modprobe -v ip_gre
# ip addr sh

1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
link/ether 00:0c:29:68:5d:32 brd ff:ff:ff:ff:ff:ff
inet 192.168.122.130/24 scope global eth0
inet6 fe80::20c:29ff:fe68:5d32/64 scope link
valid_lft forever preferred_lft forever

3: sit0: <NOARP > mtu 1480 qdisc noqueue
link/sit 0.0.0.0 brd 0.0.0.0
inet6 ::127.0.0.1/96 scope host
valid_lft forever preferred_lft forever

4: tunl0: <NOARP> mtu 1480 qdisc noop
```

```
link/ipip 0.0.0.0 brd 0.0.0.0

5: gre0: <NOARP> mtu 1476 qdisc noop
link/gre 0.0.0.0 brd 0.0.0.0
```

Možemo uočiti da se učitavanjem svakog od modula pojavilo jedno pripadajuće sučelje. Uočimo iz ispisa overhead zbog dodavanja vanjskog zaglavlja: mtu vrijednost za eth0 sučelje pokazuje da su IP paketi veličine 1500 okteta. Kod IPIP i SIT tunela, maksimalna je veličina paketa 1480 okteta, dok je kod GRE tunela overhead najveći i iznosi 24 okteta.

Obratimo još samo pažnju na to da ukoliko konfiguriramo usmjernik uključimo odgovarajuće funkcije usmjernika:

```
#sysctl -w net.ipv4.conf.all.forwarding=1 # za GRE i IPIP
#sysctl -w net.ipv6.conf.all.forwarding=1 # za SIT
```

Nećemo koristiti generička sučelja sit0, tunl0 i gre0, nego ćemo stvoriti vlastiti tunel. Sintaksa naredbe za dodavanje tunela je:

```
# ip tunnel add TUN_NAME mode MODE [ local ADDR ] [ remote ADDR ]
```

TUN_NAME je proizvoljan string (ime tunela), MODE je ipip, gre ili sit, a ADDR su adrese krajeva tunela (čvorovi koji dodaju ili odvajaju dodatno IP zaglavlje). Vidimo da niti jedan od krajeva tunela ne mora biti definiran (ili može biti podešen na default vrijednost any). Tunel s nedefiniranim izvorišnim krajem koristi se kod mreža s dinamičkim pridjeljivanjem adresa.

Ovisno o tome da li je definirana adresa udaljenog kraja tunela (remote ADDR), tuneli se dijele na dvije skupine:

- pointtopoint tuneli: svi se paketi dostavljaju na definirano odredište,
- NBMA (Non-Broadcast Multi-Access) tuneli: remote adresa nije definirana (npr. generička sučelja sit0, tunl0 i gre0).

Prvi navedeni, pointtopoint tuneli, nakon stvaranja se konfiguriraju kao i svako fizičko sučelje - dodaje im se adresa i potrebne rute:

```
# ip addr add ADDR dev TUN_NAME
# ip link set TUN_NAME up
# ip route add DEST_ADDR dev TUN_NAME
```

Kod NBMA tunela sklopovlje ne zna kamo treba proslijediti paket te je zato potrebno unijeti rutu na drugačiji način (navodimo primjer za NBMA tunl0 sučelje):

```
# ip route add DEST_ADDR via REMOTE_ADDR dev tunl0 onlink
```

Nakon ključne riječi via navodimo adresu udaljenog kraja tunela. U članku o ip route naredbi objasnili smo da nakon ključne riječi via možemo navesti jedino adresu sučelja usmjernika koji je izravno dostupan preko našeg sučelja (u ovom slučaju tunl0). Kako u ovom slučaju udaljeni kraj tunela nije vidljiv, koristimo ključnu riječ onlink kako bi jezgra ipak stvorila rutu.

Dodatna opcionalna polja kod stvaranja tunela su primjerice ttl ili dsfiled i sl, a njihove podrazumijevane (default) vrijednosti podešene su na inherit (što znači da se vrijednosti nasljeđuju od mrežnog uređaja kojem je tunel pridružen). Tuneli tipa GRE koriste i neke dodatne opcije koje niti IPIP niti SIT tuneli nemaju: npr. podešavanje ključa (običan ili dotted-decimal broj) s opcijom key, te stvaranje, odnosno provjeravanje, jednosmjerne sume opcijom csum.

Pokažimo sada primjer stvaranja IPIP tunela (stvaranje preostalih tipova tunela potpuno je analogno). Zamislimo slijedeće mreže:

```
PRVA MREŽA

network 10.0.1.0/24
gateway 10.0.1.1
public IP 161.53.55.99

DRUGA MREŽA

network 10.0.2.0/24
gateway 10.0.2.1
public IP 161.53.66.88

Konfiguracija u prvoj mreži je:


# ip tunnel add prviIPIP mode ipip remote 161.53.66.88 local 161.53.55.99
# ip addr add 10.0.1.1 dev prviIPIP
# ip link set prviIPIP up
# ip route add 10.0.2.0/24 dev prviIPIP
# ip addr show dev prviIPIP

6: prviIPIP@NONE: <POINTOPOINT,NOARP,UP> mtu 1480 qdisc noqueue
link/ipip 161.53.55.99 peer 161.53.66.88
inet 10.0.1.1/24 scope global prviIPIP

# ip tunnel sh
tunl0: ip/ip remote any local any ttl inherit nopmtudisc
prviIPIP: ip/ip remote 161.53.66.88 local 161.53.55.99 ttl inherit

# ip route sh
10.0.2.0/24 dev prviIPIP scope link



Konfiguracija u drugoj mreži je:
```

```
# ip tunnel add drugiIPIP mode ipip remote 161.53.55.99 local 161.53.66.88
# ip addr add 10.0.2.1 dev drugiIPIP
# ip link set drugiIPIP up
# ip route add 10.0.1.0/24 dev drugiIPIP
# ip addr show dev prviIPIP

6: drugiIPIP@NONE>: <POINTOPOINT,NOARP,UP> mtu 1480 qdisc noqueue
link/ipip 161.53.66.88 peer 161.53.55.99
inet 10.0.2.1/24 scope global drugiIPIP

# ip tunnel sh
tunl0: ip/ip remote any local any ttl inherit nopmtudisc
drugiIPIP: ip/ip remote 161.53.55.99 local 161.53.66.88 ttl inherit

# ip route sh
10.0.1.0/24 dev drugiIPIP scope link
```

Pokušajte sada pingati ra?unalo 10.0.2.1/24 s ra?unala 10.0.1.1/24 ili obrnuto. Ukoliko ping uspješno prolazi, a trebao bi ako ste slijedili upute - stvorili ste i konfigurirali svoj IPIP tunel

- Logirajte [1] se za dodavanje komentara

uto, 2006-02-28 00:00 - Uredništvo**Vijesti:** Linux [2]
Kuharice: Linux [3]
Kategorije: Operacijski sustavi [4]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr/node/8

Links
[1] https://sysportal.carnet.hr/sysportallogin
[2] https://sysportal.carnet.hr/taxonomy/term/11
[3] https://sysportal.carnet.hr/taxonomy/term/17
[4] https://sysportal.carnet.hr/taxonomy/term/26