OpenBSD hálózat és NAT64

Répás Sándor

2020.11.25.

Alapvető hálózati beállítások

- Az egyes interfészekhez tartozó beállítások:
  - /etc/hostname.* állományok
  - A * helyén a hálózati kártya típus (driver) azonosító
  - Az adott kártya/interfész IPv4 és IPv6 címet, valamint a szükséges routing bejegyzés(eke)t is tartalmazza
  - IPv4 cím: inet
  - IPv6 cím: inet6
  - Routing bejegyzés: !route add
- Pl:

```
inet 10.1.1.1 255.255.255.0
inet6 alias 2001:738:2c01:8001::1 64
!route add -inet6 default 2001:738:2c01:8000::99
```

Előzőek Debian Linuxon

- /etc/network/interfaces

```
iface eth0 inet static
address 10.1.1.1
netmask 255.255.255.0
iface eth0 inet6 static
address 2001:738:2c01:8001::1
netmask 64
gateway 2001:738:2c01:8000::99
```

Névfeloldás és interfészek

- /etc/resolv.conf

  ```
  search sze.hu
  nameserver 192.168.0.1
  lookup file bind
  ```
  (először a /etc/hosts állományban keresi, és ha ott nem találja, akkor DNS segítségével oldja fel a nevet)

- Nagyon hasonló, mint Debian Linuxon

Hálózat konfigurációs állományok (újra)beolvasása

OpenBSD

```
sh /etc/netstart
```

Debian Linux

```
/etc/init.d/networking restart
```

Parancsok

- ifconfig
  - Hálózati interfészek paramétereinek lekérdezése és beállítása
  - Hasonló, mint Linuxokon
- route
  - Routing tábla kiíratása, manipulálása
  - show: kiíratás
  - add: hozzáadás

Példák 1

```
bsd# ifconfig hvn1 185.143.48.39 255.255.255.0
bsd# route add 10.0.0.0/0 192.168.0.65
add net 10.0.0.0/0: gateway 192.168.0.65
bsd# route show
```

(eredménye a következő dián)

Példák 2

```
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use   Mtu  Prio Iface
default            192.168.0.1        UGS         5      280     -     8 hvn0
base-address.mcast localhost          URS         0      186 32768     8 lo0
10.0.0.0/32        192.168.0.65       UGS         0        0     -     8 hvn0
127/8              localhost          UGRS        0        0 32768     8 lo0
localhost          localhost          UHhl        1        2 32768     1 lo0
185.143/16         185.143.48.39      UCn         2    12080     -     4 hvn1
router0.ahol.com   cc:2d:e0:0b:79:af  UHLc        0     9551     -     3 hvn1
185.143.48.35      d8:cb:8a:39:06:5c  UHLc        1        2     -     3 hvn1
185.143.48.39      00:15:5d:c8:2f:2d  UHLl        0       53     -     1 hvn1
ip185-143-255-255. 185.143.48.39      UHb         0        0     -     1 hvn1
192.168.0/24       192.168.0.64       UCn         3        2     -     4 hvn0
192.168.0.1        00:15:5d:c8:2f:06  UHLch       1       22     -     3 hvn0
192.168.0.4        00:15:5d:c8:2f:04  UHLc        1       47     -     3 hvn0
192.168.0.64       00:15:5d:c8:2f:17  UHLl        0       26     -     1 hvn0
192.168.0.65       link#1             UHLch       1        4     -     3 hvn0
192.168.0.255      192.168.0.64       UHb         0        4     -     1 hvn0

Internet6:
Destination        Gateway            Flags    Refs      Use   Mtu  Prio Iface
::/96              localhost          UGRS        0        0 32768     8 lo0
localhost          localhost          UHhl       10       20 32768     1 lo0
::ffff:0.0.0.0/96  localhost          UGRS        0        0 32768     8 lo0
2002::/24          localhost          UGRS        0        0 32768     8 lo0
2002:7f00::/24     localhost          UGRS        0        0 32768     8 lo0
2002:e000::/20     localhost          UGRS        0        0 32768     8 lo0
2002:ff00::/24     localhost          UGRS        0        0 32768     8 lo0
fe80::/10          localhost          UGRS        0        0 32768     8 lo0
fec0::/10          localhost          UGRS        0        0 32768     8 lo0
fe80::1%lo0        fe80::1%lo0        UHl         0        0 32768     1 lo0
ff01::/16          localhost          UGRS        0        0 32768     8 lo0
ff01::%lo0/32      fe80::1%lo0        Um          0        1 32768     4 lo0
ff02::/16          localhost          UGRS        0        0 32768     8 lo0
ff02::%lo0/32      fe80::1%lo0        Um          0        1 32768     4 lo0
```

Csomagtovábbítás (forwarding) engedélyezése

OpenBSD

/etc/sysctl.conf állományban

```
net.inet.ip.forwarding=1
net.inet6.ip6.forwarding=1
```

Parancssorból

```
sysctl net.inet.ip.forwarding=1
sysctl net.inet6.ip6.forwarding=1
```

Debian Linux

/etc/sysctl.conf állományban

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
```

Parancssorból

```
sysctl net.ipv4.ip_forward=1
sysctl net.ipv6.conf.all.forwarding=1
```

Packet Filter (PF)

- 2001-ben jelent meg az OpenBSD 3.0-ában
- OpenBSD csomagszűrő és NAT eszköze, mely nagyon sokrétűen alkalmazható
- Különböző verziói megtalálhatók a NetBSD és FreeBSD rendszerekben is
- Képes (állapottartó) stateful és (állapotmentes) stateless működésre is
- Támogatja a NAT64-et is, mely az Ecdysis kódbázisán alapul

PF konfigurálás 1

- Konfigurációs állománya a /etc/pf.conf, melyet minden indításkor beolvas a rendszer (amennyiben engedélyezve van a PF)
- Konfiguráció (újra)beolvasása

  `pfctl -f /etc/pf.conf`
- Konfiguráció szintaxisának ellenőrzése

  `pfctl -nf /etc/pf.conf`
- PF letiltása/engedélyezése

  `pfctl -d/pfctl -e`
- Permanens PF letiltás/engedélyezés

  /etc/rc.conf-ban: `pf=NO/YES`

PF konfigurálás 2

- Aktuális szabályok kiíratása

  `pfctl -sr`
- Állapottábla kiíratása

  `pfctl -ss`
- Filter statisztikák és számlálók kiíratása

  `pfctl -si`
- Maximum értékek kiíratása

  `pfctl -sm`

/etc/pf.conf felépítése

- Öt rész:
  - Macros: Felhasználó által megadott változók. Pl: IP cím, interfész név
  - Tables: IP címeket tartalmazó struktúra
  - Options: PF működését befolyásoló beállítások
  - Queueing: Sávszélesség korlátozás és prioritizálás
  - Filter Rules: Csomagok szűrése, NAT

Példák

- Macro:

```
block out on fxp0 from { 192.168.0.1, 10.5.32.6 } to any
```

- Table:

```
table <goodguys> { 192.0.2.0/24 }
pass  in on fxp0 from <goodguys> to any
```

- Options:

```
set limit states 40000
```

Filter rule szintaxisa

action [direction] [log] [quick] [on interface] [af] \
[proto protocol] [from src_addr [port src_port]] \
[to dst_addr [port dst_port]] [flags tcp_flags] [state]

Filter rule példák

Próbáljuk önállóan értelmezni:

```
pass in on sk0

block in on sk1 from 192.168.0.1

pass in on sk1 proto udp from any to any no state

pass out on sk1 from sk0 to any nat-to sk1
```

Filter rule példák

```
pass in on sk0
```

sk0 interfészen érkező csomagokat elfogadja

```
block in on sk1 from 192.168.0.1
```

sk1 interfészen a 192.168.0.1 címről érkező csomagokat tiltja

```
pass in on sk1 proto udp from any to any no state
```

sk1 interfészen érkező UDP datagrammokat elfogadja/továbbengedi és nem tartja nyilván az állapottáblában

```
pass out on sk1 from sk0 to any nat-to sk1
```

Az sk1 IP címe mögé rejtve NAT az sk1 interfészen, minden célcím irányába kimenő forgalmat, amely az sk0 interfészen érkezett (masquerade, source NAT)

NAT64

- Ha egy IPv6-os kliens egy IPv4-es kiszolgálót szeretne elérni, akkor a kiszolgálót a DNS64-től kapott IPv4-embedded IPv6 címmel címzi meg.
- A routing az IPv4-embedded IPv6 címre küldött csomagokat a NAT64 gateway felé irányítja.
- A NAT64 gateway végzi az IPv6 és IPv4 protokollok közti átalakítást.

OpenBSD PF NAT64

- Az OpenBSD PF támogatja a NAT64-et „address family translation” néven
- Beállítása a /etc/pf.conf-ban:

```
pass in on sk1 inet6 from any to \
2001:738:2c01:8001:ffff:ffff::/96 af-to inet from \
193.225.151.75
```

DNS64

- A DNS szolgáltatás kiterjesztése.
- Ha a kért névhez tartozik IPv6-os cím, azaz AAAA rekord, normál működés történik.
- Ha a kért névhez csak IPv4-es cím, azaz csak A rekord van, akkor szintetizál egy AAAA rekordot, melynek utolsó 32 bitje az IPv4-es cím, majd válaszként ezt a szintetizált címet adja vissza. Ez az IPv4-embedded IPv6 cím.

DNS64+NAT64

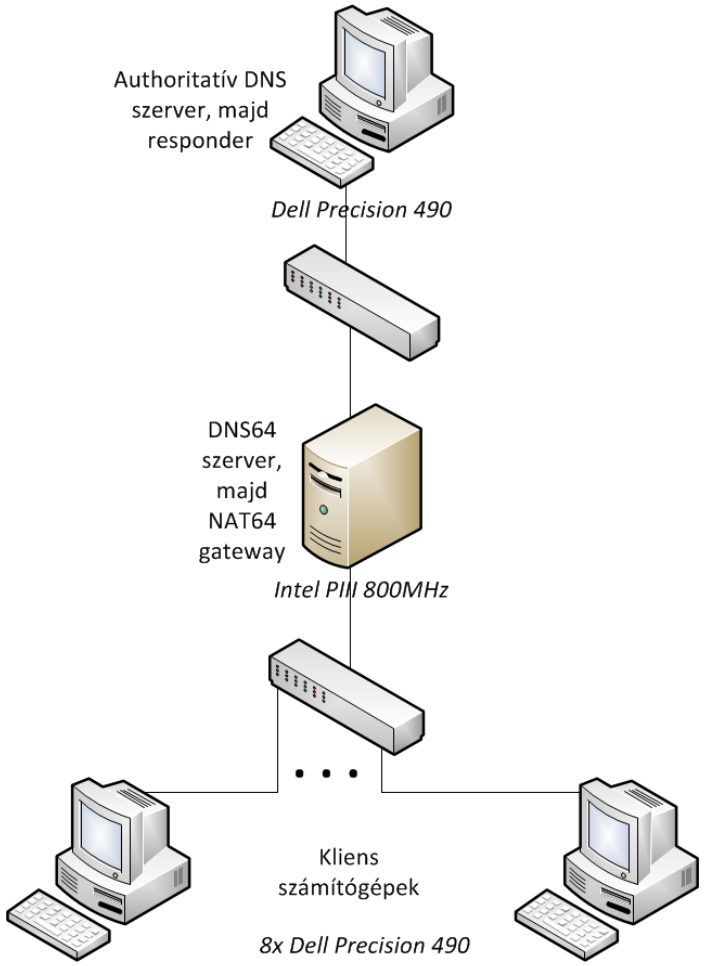
- A két szolgáltatás együttes alkalmazásával megoldható, hogy a csak IPv6-os IP címmel rendelkező kliens elérhesse a csak IPv4-es címmel rendelkező szervert.
- IPv4-es címet a szolgáltatók a viszonylag kevés új szervernek még tudnak biztosítani, azonban a sok új kliensnek már csak IPv6-os címet képesek allokálni.
- A meglévő szervereknek csak kis része érhető el IPv6 protokollal is.
- Így a DNS64/NAT64 ideális megoldás lehet az IPv6 bevezetésének jelenlegi szakaszában.

Miért érdemes OpenBSD-vel foglalkozni?

- Biztonságra kihegyezett operációs rendszer
- A PF nagyon gyors és sokrétűen alkalmazható
- Lényegesen gyorsabban továbbítja NAT64 segítségével a csomagokat, mint a Linux alapú TAYGA
- Lényegesen több csomagot továbbít nála, kisebb terhelés mellett

Mérési topológia

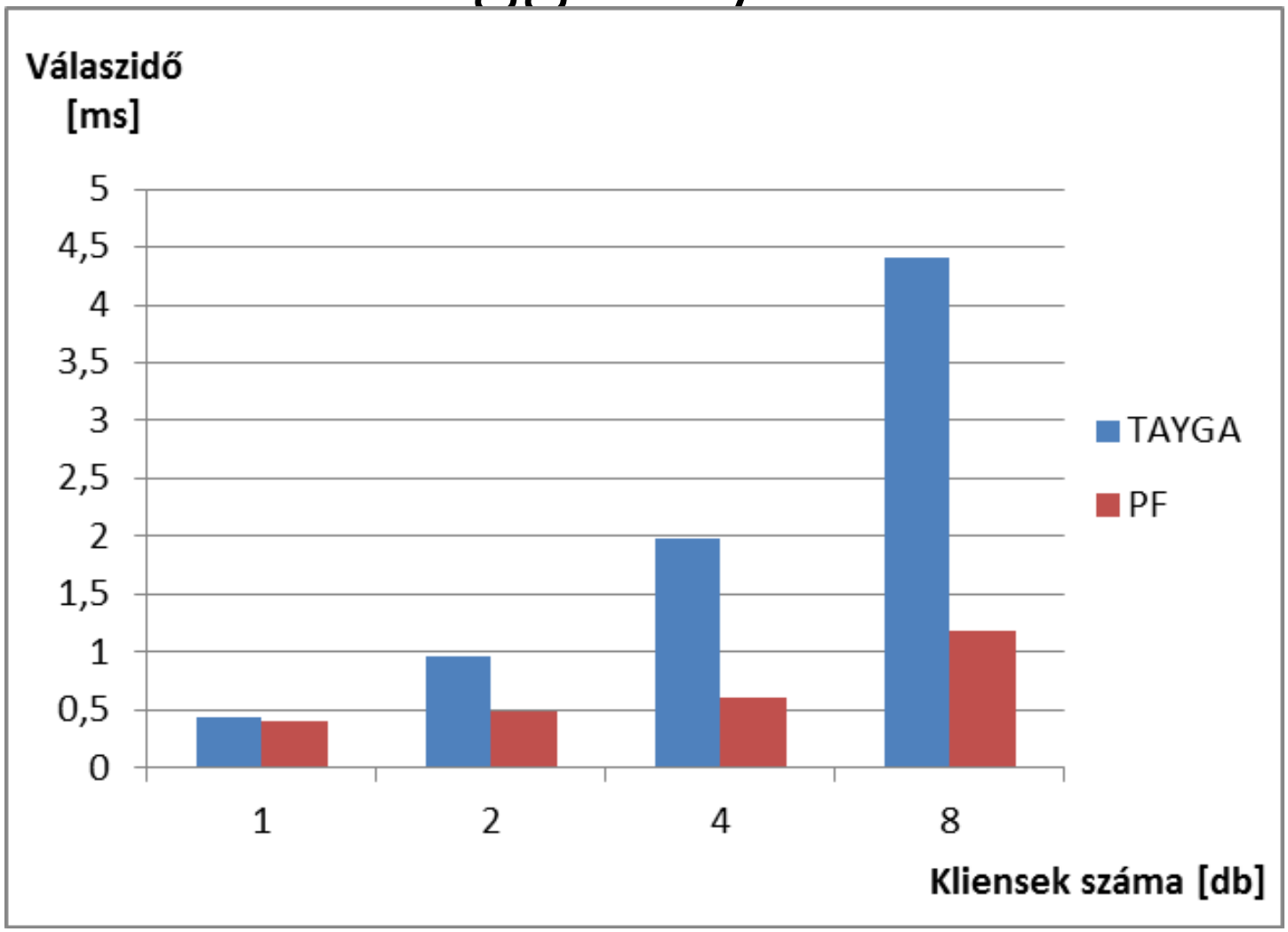
- Vizsgált eszközök:
  - DNS64 szerver és NAT64 gateway implementációk
  - egy kis teljesítményű számítógépen futtatva
- További eszközök:
  - 8+1 darab nagyteljesítményű Dell Precision munkaállomás
  - 2 darab 1000BASE-TX switch



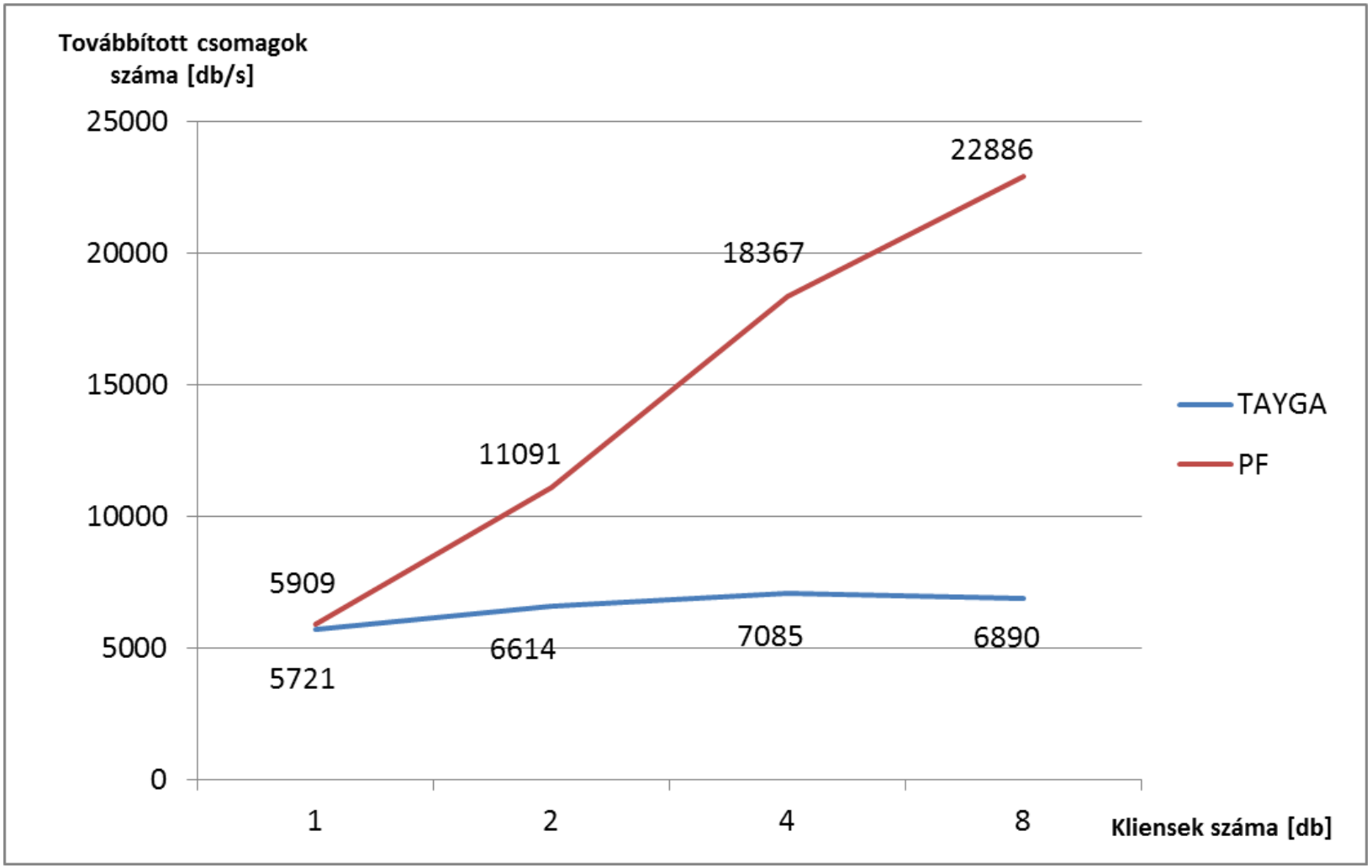
Mérés menete

- Előző ábra hálózati topológiája.
- Erre a célra készült scriptek segítségével történt.
- Minden mérési sorozat 1, 2, 4 és 8 klienssel került kivitelezésre, a terhelés mértékének megbízható beállításához.

Válaszidő [ms] a kliensek számának függvényében



Másodpercenként továbbított csomagok száma a kliensek számának függvényében



Kérdések

repas.sandor@sze.hu