

IPv6-támogatás linuxos hálózati csomópontokon

Ez a cikk a Linux-kiszolgálók IPv6-beállításához és IPv6 alapú internetcsatlakoztatásukhoz nyújt technikai segítséget.

Az IPv4, az IP-protokoll jelenlegi változata bebizonyította, hogy hatékony, könnyen megvalósítható, széles körben használható, és kiállta a jelenkori Internet méretének próbáját, ahol nagyrészt még mindig IPv4 rendszert használnak, holott már közel húszéves módszer. Az IPv4 meglelt korának ellenére figyelemre méltóan rugalmas volt, de azért manapság már kezdenek gondok jelentkezni. Az IPv4 eredeti tervénél nem vettek figyelembe néhány, manapság már igen lényegesnek számító szempontot. Ezek közé tartozik például a nagy címtartomány kezelése, amely megoldást jelent a címbeszűkülés kérdésre, a hordozhatóság, a biztonság, az önműködő beállítási megoldások nyújtása, illetve a szolgáltatás minősége.

Hogy megoldást találjon e gondokra, az Internet Engineering Task Force (IETF) kifejlesztett egy protokoll- és szabványkészletet, amelyet IP version 6 (azaz IPv6) néven ismerünk, s amely már tartalmazza az IPv4 továbbfejlesztéséhez szükséges elképzeléseket és ajánlott módszereket. Az IPv6 képességei közé tartozik az új fejlécformátum, a nagyobb (128-bites) címtartomány, a hatékony és hierarchikus címzési és útkeresési háttér, az állapotfüggetlen (stateless) és állapotfüggő (statefull) címbiztonság, beépített biztonság, jobb hordozhatóságtámogatás, illetve egy új protokoll a szomszédos csomópontok kapcsolattartásához.

Az IPv6 esetében az állapotfüggetlenség és -függőség jelentése a következő:

- Az állapotfüggő önműködő beállítási módszer esetén befűzött csomóponti számítógép az összes szükséges adatot (cím, beállítási adatok, kapcsolók stb.) egy kiszolgálótól (annak adatbázisából) kapja meg. Az adatok szigorú ellenőrzés alatt vannak, s a vezérlést például a DHCPv6 protokoll végezheti.
- Az állapotfüggetlen önműködő beállítás esetén nem szükséges az előbbi feltétel. A befűzött számítógép a befűzéséhez szükséges adatokat a hálózatból nyeri ki (üzenetszórás segítségével). Az IPv6-címek csak egy meghatározott ideig érvényesek, azt követően más csomópontnak lesznek kiosztva. Látható, hogy az IPv6 nem csupán az IPv4 gondjait küszöböli ki, hanem sok újdonságot is bevezet. Az IPv6 várhatóan fokozatosan váltja fel az IPv4 protokollt, s a váltás során jó pár évig egymás mellett léteznek majd.

Linuxos IPv6-megvalósítások

Linux alá két fő IPv6 fejlesztési vonal létezik: a Linux-rendszermag részeként érkező megvalósítások, illetve az USAGI (UniverSAl playGround for IPv6, magyarul IPv6 egyetemleges játszótér) megoldások. Az USAGI Projekt azon munkálkodik, hogy termelési minőségű IPv6 protokollt állítson elő Linux alá, szorosan együttműködve a WIDE, KAME és TAHI projektekkel. Különböző szervezetekből érkező jelentkezők készítik, amelyek a IPv6-protokollverem létrehozásával járulnak hozzá a Linux és a IPv6-közösség munkájához. Jelenleg több különböző fejlesztés is folyik különféle terjesztési csoportokban; a USAGI ezeket a csoportokat igyekszik egyesíteni, hogy végül

minden Linux-terjesztésbe egyetlen, egységes IPv6-megvalósítás kerüljön.

E cikkhez a Linux-rendszermag <http://kernel.org>-ról letöltött 2.4.5 változatát használtuk, jelenleg azonban a 2.4.19-es rendszermag a legfrissebb. Először is megmutatjuk, hogyan lehet IPv6-támogatással lefordítani a rendszermagot, majd megnézzük, miképpen lehet az alapvető hálózati programokat rábírní az IPv6-támogatására, végül azt is megtudhatjuk, hogyan kapcsolhatjuk rá IPv6-támogatású kiszolgálónkat a IPv6 Internetre, a <http://www.freenet6.net> projekt szolgáltatását használva.

IPv6-támogatás a Linux-rendszermagban

Első lépésünk a Linux-rendszermag letöltése a

<http://kernel.org>-ról, majd az állomány kicsomagolása:

```
tar -xzf linux-2.4.19.tar.gz
```

Ezáltal létrejött egy *linux* nevű könyvtárunk. Mozgassuk ezt a könyvtárat a */usr/src* könyvtár alá *linux-2.4.19* néven, jelezve a rendszermag változatszámát. Ezután készítenünk kell egy hivatkozást a 2.4.19 forráskönyvtárra:

```
ln -s /usr/src/linux-2.4.19 /usr/src/linux
```

Ha ezzel megvagyunk, az új rendszermagban be kell állítanunk az IPv6-támogatást:

```
cd /usr/src/linux
make xconfig
```

vagy karakteres felületen használhatjuk a `make menuconfig` parancsot.

A rendszermag-beállítás során ehhez két lehetőséget kell bekapcsolnunk. Először is lépünk a *Code Maturity Level* menüpontra, és kapcsoljuk be a *development/incomplete code/drivers* pontot:

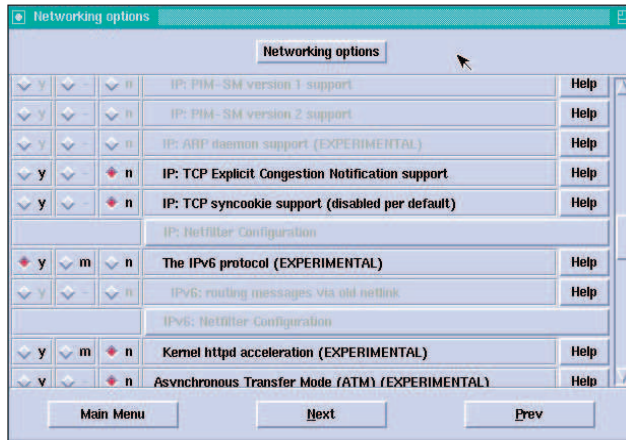
```
"Prompt for development and/or incomplete
code/drivers" YES
```

Majd menjünk a *Networking Options* menübe, itt most be tudjuk kapcsolni az IPv6-protokollt:

```
IPv6 Protocol (EXPERIMENTAL) YES
```

Rendszermagszinten mindössze ennyi beállításra volt szükségünk. Ezután beállításainkat menthetjük, majd a *Save and Exit* gombra kattintva kiléphetünk. Ezáltal a */usr/src/linux* könyvtárban létrejön egy *.config* nevű állomány, amely tulajdonképpen a rendszermag beállításfájlja. Most már készen állunk a rendszermag lefordítására, ehhez a következő lépéseket kell megtennünk:

```
make clean; make dep; make bzImage
```



1. kép A Linux-rendszermag IPv6-beállítás



2. kép A hurokeszköz

Az eredmény egy új rendszermagpéldány lesz a `/usr/src/linux/arch/i386/boot/` könyvtár alatt. Ha a további képességeket esetleg modulként állítottunk be, a modulokat is le kell fordítani, illetve telepíteni:

```
make modules; make modules_install
```

Ezek után az új, IPv6-támogatással rendelkező rendszermagunkat át kell másolnunk a `/boot` könyvtár alá:

```
cp /usr/src/linux/arch/i386/boot/bzImage
  ↪ /boot/bzImage.ipv6
```

és frissítenünk kell `System.map` fájlunkat:

```
cp /usr/src/linux/System.map
  ↪ /boot/System.map-2.4.5-ipv6
ln -fs /boot/System.map-2.4.5-ipv6
  ↪ /boot/System.map
```

Egyetlen lépés maradt csak hátra: frissítenünk kell a `/etc/lilo.conf` fájlt, hozzáadva az új IPv6-támogatású rendszermagbejegyzést. Nyissuk meg a `/etc/lilo.conf` fájlt, és adjuk hozzá a következő bejegyzést:

```
image=/boot/bzImage.ipv6
label=linux_ipv6
root=/dev/hda1 # ezt cserölj k ki a saját
               # lemezzrösz nkre
read-only
```

Végül frissítsük a LILO-beállításokat a következő paranccsal:

```
/sbin/lilo
```

Észálta a `linux_ipv6` bejegyzés rendszerindításkor bekerül LILO-beállításaink közé. Most már készen állunk rá, hogy újraindítsuk a kiszolgálót. Amikor a LILO feljön, válasszuk a `linux_ipv6` indítást (a LILO nem feltétlenül a menüvel jön fel, ilyenkor egy CTRL vagy ALT lenyomása szükséges a LILO felirat megjelenésakor). Et voilà! Rendszerünk rendszermag-szintű IPv6-támogatással indul. Ellenőrzésképpen gépeljük be az `ifconfig` parancsot. Hurokeszköz-beállításunkban az IPv6 típusú `::1` helyi címet kell mutatnia (lásd a 2. képet).

IPv6-programok és -eszközök

Ha a rendszermag már támogatja az IPv6-ot, feltelepíthetünk néhány eszközt beállításaink kipróbálására, illetve a más rendszerekkel történő IPv6-átvitel kiépítésére. Először is szükségünk lesz az IPv6-támogatással bíró alaphálózati eszközökre, hogy beállíthassuk a csatolófelületet, illetve néhány más olyan IP-eszközre, mint a `ping6`, `telnet6` stb., végül képesnek kell lennünk néhány alapvető IPv6-próba végrehajtására.

Ezeket az eszközöket három csomagban érhetjük el: `net-tools`, `iputils` és `NetKit`-eszközök. Az első kettő feltétlenül szükséges, a `NetKit` elhagyható. Továbbá, ha az esetleg felmerülő lehetséges IPv6 hálózati hibákat fel szeretnénk tudni deríteni, szükségünk lesz az IPv6-támogatású `tcpdump` és `libpcap` csomagokra is. A következő részekben a fent említett csomagok telepítésével foglalkozunk.

net-tools

A `net-tools` a linuxos hálózatkezelés irányítására szánt programcsomag, amelyben olyan programokat találunk, mint az `arp`, a `hostname` (`domainname`, `dnsdomainname`, `nisdomainname`), az `ifconfig`, az `ipmaddr`, az `iptunnel`, a `netstat`, a `rarp`, a `route` és a `plipconfig`. Ezt a csomagot a <http://www.tazenda.demon.co.uk/phil/net-tools> címen lelhetjük fel.

A csomag telepítéséhez először is töltsük le a csomagokat közvetlenül a `/usr/src` könyvtárba, majd gépeljük be egyenként a következő parancsokat:

```
cd /usr/src
tar xIvf net-tools-1.60.tar.bz2
cd net-tools-1.60
./configure.sh config.in
```

A beállítás során néhány kérdésre kell válaszolnunk, hogy beállíthassuk a `net-tools`-t. Azt javasolom, a következő kérdésekre válaszoljunk igennel (y):

```
INET6 (IPv6) protocol family (HAVE_AF_INET6)
  ↪ [n] y
SIT (IPv6-in-IPv4) support (HAVE_HWSIT)
  ↪ [n] y
Build iptunnel and ipmaddr (HAVE_IP_TOOLS)
  ↪ [n] y
```

Ezek után le kell fordítanunk és telepítenünk kell a `net-tools` csomagot:

```
make
make install
```

A programok a `/sbin` és `/bin` könyvtárakba kerülnek, és máris használhatjuk őket, feltéve, hogy a gépet az IPv6-támogatású rendszermaggal már újraindítottuk.

iputils

Ez a csomag a következő eszközöket tartalmazza: ping, ping6, traceroute6, rdisc, clockdiff, tftpd, tracepath, tracepath6 és arping. Elképzelhető, hogy a rendszerünkön telepített IP-eszközök nem támogatják az IPv6-ot. Az első lépés annak vizsgálata, hogy jelenlegi változatunk vajon Ipv6-támogatású-e. Gépeljük be a következő utasítást a parancssorba:

```
rpm -q --qf "%{NAME}-%{VERSION}\n" iputils
```

(Debian alatt a fentebbi programok külön net-tools, traceroute és netkit-ping nevű csomagokban vannak). Ha a kimenet iputils-20000121 vagy ennél frissebb, akkor nem kell a legújabb iputils csomagra frissítenünk. Ha nem ez a helyzet, az eszközök feltelepítéséhez követnünk kell a következő lépéseket. Először is töltsük le a csomagot a <http://ftp.inr.ac.ru/ip-routing> címről. Mi az iputils-ss001110.tar.gz csomagot használtuk. Ezután csomagoljuk ki a */usr/src* könyvtárba:

```
tar -xzf iputils-ss001110.tar.gz
```

Végül fordítsuk le az iputils-t a make paranccsal. A csomag nem támogatja a make install utasítást, ezért aztán az elkészült programokat tetszés szerinti helyre tehetjük. Természetesen nem árt, ha meggyőződünk róla, hogy elérési útvonalunkban az eszközök régi változata nem ütközik-e az újabbakkal. Mentsük tehát az új programokat a */usr/local/iputils/bin/* könyvtárba. A csomag minden egyes eszközhöz egy-egy súgóoldalt is szolgáltat. Ezeket olyan könyvtárba kell helyezni, ahol a man parancs megtalálja őket; ha kíváncsiak vagyunk, milyen könyvtárak szerepelnek a man elérési útjában, gépeljük be a manpath parancsot.

NetKit-eszközök

Ezekre az alapvető eszközökre lesz szükségünk új IPv6-hálózatunk működtetéséhez és kipróbálásához. A NetKit a következő eszközöket tartalmazza: ping, finger, telnet, rwho és az ezeknek megfelelő démonok. Ezek igen hasznos eszközök, hiszen IPv6-támogatással is le tudjuk őket fordítani. A csomagot a következő címről tölthetjük le:

```
➔ http://freshmeat.net/projects/netkit.
```

Mi a nkit-0.5.1.tar.gz változatot próbáltuk ki. A következőkben azt mutatjuk be, milyen lépéseket kell megtenni az eszközök Linux-kiszolgálóra való telepítéséhez. Először is töltsük le a legfrissebb NetKit-csomagot a fenn megadott honlapról. Azután a letöltött fájlt helyezük a */usr/src* könyvtárba. Bontsuk ki a csomagot a következő paranccsal:

```
tar -xzf nkit-0.5.1.tar.gz
```

Ezután adjuk ki a ./configure utasítást. Fordítsunk a make clean és make parancsokkal, majd a programokat másoljuk a */usr/local/bin* könyvtárba:

```
cp telnet/telnet /usr/local/bin/telnet6
cp telnetd/in.telnetd
➔ /usr/local/sbin/in.telnetd6
cp finger/finger /usr/local/bin/finger6
cp ping/ping /usr/local/bin/ping6
cp fingerd/in.fingerd
➔ /usr/local/sbin/in.fingerd6
```

Mostantól rendelkezünk az alapvető szolgáltatásokkal, így például helyi IPv6 visszacsatolt (loopback) eszközünkön kipróbálhatjuk a ping6 parancsot.

Figyelem, a Red Hat 7.x-felhasználóknak a NetKit-csomagot foltozniuk kell! A folt a <http://ftp.bieringer.de/pub/linux/IPv6/netkit> címen érhető el.

Választható eszközök

Van néhány választható eszköz is, amit feltelepíthetünk a rendszerünkre, ha képességeit IPv6-protokollra is ki szeretnénk terjeszteni. Ebben a cikkben csak három ilyen csomagot említünk meg: libpcap, tcpdump és xinetd.

A libpcap és a tcpdump

Ha meg szeretnénk érteni, mi történik csomagszinten az IPv6-hálózatunkban, a libpcap és tcpdump csomagokat is fel kell készítenünk az IPv6-ra. A libpcap egy olyan rendszerfüggetlen felület felhasználói szintű csomagok elfogására, ami tulajdonképpen hordozható keretrendszer nyújt az alacsony szintű hálózatfigyeléshez. A másik a tcpdump nevű eszköz, ami hálózatfigyelési és adatgyűjtési feladatokat lát el. Amennyiben ilyen szolgáltatásokat szeretnénk, előbb le kell töltenünk a legfrissebb változatokat, majd telepítenünk szükséges őket. Az általunk kipróbált változatok a tcpdump 3.6.2 és a libpcap 0.6.2 voltak. Először is töltsük le a csomagokat a <http://www.tcpdump.org> oldalról, majd másoljuk a */usr/src*-be. Csomagoljuk ki a megfelelő parancsokkal:

```
tar -xzf libpcap-0.6.2.tar.gz
tar -xzf tcpdump-3.6.2.tar.gz
```

A kicsomagolást követően két könyvtárat kapunk, csomagonként egyet-egyet. Ezután mind a két csomag esetében ugyanazokat a lépéseket kell megtennünk; igaz, elsőként a libpcap csomagon kell végrehajtanunk és csak azután a tcpdump-on. Először is futtassuk le a beállító parancsfájlt IPv6-támogatás engedélyezéssel:

```
./configure --enable-ipv6
```

Majd fordítsuk le a csomagot a make clean és make parancsokkal. Végül a make install paranccsal telepítsük a programokat.

Következő lépésként módosítanunk kell az elérési utunkat, hogy az IPv6-támogatású programokat is elérhessük. Esetleg a */etc/profile* fájlt is átszerkeszthetjük, hogy a PATH környezeti változó tartalmazza a */local/sbin* és */usr/local/bin* könyvtárakat, majd a változtatások érvényesítéséhez töltsük újra a */etc/profile* fájlt:

```
source /etc/profile
```

A xinetd IPv6-támogatással

Ha telnet6 alatt is el szeretnénk érni a gépünket, a xinetd-t is le kell fordítanunk inet6-támogatással. Alap esetben a feltelepített démon nem képes IPv6-címeket kezelni. Ezért aztán kénytelenek vagyunk xinetd-re frissíteni. A xinetd legfrissebb változatát a <http://synack.net/xinetd> címen találjuk. A mi összeállításunkban xinetd-2.1.8.8p3.tar.gz szerepelt. Töltsük le a xinetd-2.1.8.8p3.tar.gz (vagy a legújabb) csomagot a */usr/src* könyvtárba, majd csomagoljuk ki:

```
tar -xzf xinetd-2.1.8.8p3.tar.gz
```

Futtassuk a beállító parancsfájlt:

```
./configure --with-inet6 --prefix=/usr/local/bin
```

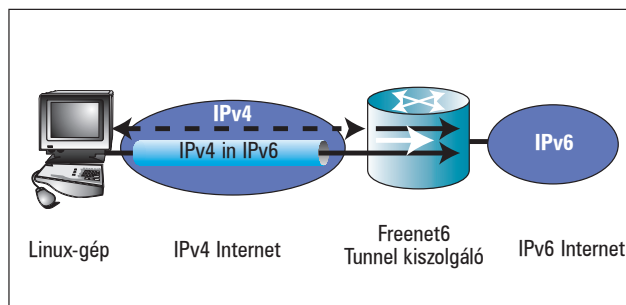
A `--prefix=/usr/local/bin` kapcsoló azt jelzi, hogy a létrejövő programok a `/usr/local/bin` könyvtár alá kerülnek. Ezután a fordítás és a telepítés következik:

```
make clean
make
make install
```

A következő lépés a régi `inetd.conf` átalakítása az új beállításfájl formátumára:

```
/usr/sbin/xconv.pl < /etc/inetd.conf >
➔ /etc/xinetd.conf
```

ahol a `/usr/sbin` a `xinetd` program könyvtárára mutató elérési út. Zárójelben jegyzem meg, nem árt meggyőződni róla, hogy a `xconv.pl` parancsfájl első sora valóban a helyes Perl programra mutat-e, illetve végre tudja-e hajtani egyáltalán. Ezután néhány igazán apró módosítást kell csak végezni a `/etc/xinetd.conf` állományban, jelezve, hogy `telnet6d`-t és `tfptp6d`-t használunk a szokásos IPv4 Telnet- és TFTP-



1. ábra Freenet6-szerkezet egyetlen gazdagéppel

démonok helyett. A fentiek elvégzése után rendszerünk Telnet és FTP szolgáltatása IPv6 rendszer alá került.

IPv6-alkalmazások

Számos alkalmazás létezik, ami támogatja az IPv6-ot. Ennek ellenére mi most csak egyetlen kiszolgálóalkalmazást említünk meg: az Apache webkiszolgálót. Az Apache az Internet legkedveltebb webkiszolgálója (forrás: <http://netcraft.com>). A legfrissebb kiadások már tartalmazzák az IPv6-támogatást, s ez alkalmassá teszi arra, hogy kipróbálhassuk rajta IPv6-beállításainkat. Miután letöltöttük és telepítettük az Apache webkiszolgáló legfrissebb változatát, IPv6 alatt is szolgáltatathatunk honlapokat. A kényelem kedvéért a `/etc/hosts` fájlt átszerkeszthetjük, hogy tartalmazza a következő sort:

```
:::1 ip6-localhost ip6-localhost
```

Ezek után a `:::1` használata helyett az `ip6-localhost` nevet is használhatjuk. Ne feledjük el ellenőrizni a `/etc/protocols`-t. Ha az alább említett bejegyzések itt nem találhatók meg, be kell őket szúrunk az IPv6-protokolltámogatási részhez:

```
ipv6 41 IPv6 # IPv6
ipv6-route 43 IPv6-Route # etvÆlasztÆsi fejlœc
# az IPv6-hoz
ipv6-frag 44 IPv6-Frag # t redetzetsœgi fejlœc
# az IPv6-hoz
ipv6-crypt 50 IPv6-Crypt # titoktartÆsi fejlœc
# az IPv6-hoz
ipv6-auth 51 IPv6-Auth # azonos tÆsi fejlœc
# az IPv6-hoz
ipv6-icmp 58 IPv6-ICMP icmpv6 icmp6M
# ICMP az IPv6-hoz
ipv6-nonxt 59 IPv6-NoNxt # nincs k vetkezi
# fejlœc az IPv6-hoz
ipv6-opts 60 IPv6-Opts # CœlbeÆll tÆsok
# az IPv6-hoz
```

Csatlakozás az IPv6 Internethez

Még 1996-ban, amikor az első IPv6 IETF szabvány elkészült, nagy érdeklődés mutatkozott egy IPv6-gerinc létrehozása iránt. Az 1996-os IETF-Montreal alatt megszületett a 6bone (IPv6 gerinc). Ez a kipróbálásra szánt (de még mindig érvényes) `3ffe::/16` tartományt használja. Kezdetben a gerinc nagy része a jelenlegi IPv4 Interneten, alagutakon keresztül működött. Ezáltal egy virtuális IPv6-hálózat jött létre az IPv4 Interneten keresztül. Manapság a 6bone egyaránt tartalmaz saját és alagutakon futó kapcsolatokat. A 6bone kipróbálásra készült, így aztán nincs is szolgáltatásszintű egyezmény az egyes szervezetek közt, természetesen ez nem azt jelenti, hogy megbízhatósági vagy hitelességi gondokkal küszködne. Minden, a `3ffe::/16` tartományon keresztülhaladó forgalom korlátozás nélkül hiteles.

Akár a 6bone-hoz, akár az IPv6 Internethez szeretnénk kapcsolódni, szükségünk lesz (akárcsak IPv4 alatt) egy szolgáltatóra. Ha közvetlenül illet nem találunk, vagy ha a jelenlegi ezt a szolgáltatást nem nyújtja, a legegyszerűbb megoldás csövezeteket készíteni egy olyan szolgáltatóhoz vagy géphez, ami hajlandó továbbító szolgáltatást felajánlani nekünk.

1999 júliusában a három helyi nyilvántartó, az amerikai ARIN, az európai és afrikai RIPE és az ázsiai illetőségű APNIC elkezdett szabályos, `2001::/16` tartományban kezdődő nem próbacímeket kiadni a szolgáltatóknak. Azok a gépek, amelyek ebben a tartományban kaptak címet, alkotják ma a termelési IPv6 Internetet.

Akárcsak az Internet korai napjaiban, elindult egy projekt, aminek a célja az volt, hogy segítse az embereket elindulni az IPv6 világában, felajánlva nekik az ingyenes és önműködővé varázsolt csövezeték-szolgáltatást, ami bármilyen személyt vagy szervezetet az IPv6 Internethez tud kapcsolni. A Viagénie tanácsadó cég által vezetett projekt neve Freenet6.net, ami ingyenesen és jelentkezéses, illetve „run-on-a-best-effort” alapon működik. A szolgáltatás igen népszerű a közösségen belül, mivel könnyű és gyors elérést nyújt az Internethez. A Freenet6-ot a tunnel broker (RFC 3053) után mintázták, ahol a csomópont és a tunnel broker között IPv6-over-IPv4 alagút jön létre. A Freenet6 ennek a módszernek egy továbbfejlesztett változata, ahol a csomópont a kiszolgálóval való egyeztetéshez Tunnel Setup Protocolt (TSP) használ. Az ügyfélcso mpont lehet gazdagép vagy útvonalválasztó is. A Freenet6 TSP-kiszolgálója nemcsak csatornákat, hanem nagy címtartományokat is nyújthat bármilyen felhasználói szolgáltatásnak. A kiadott címtér /48 alakú, amely (16 biten) 65 536 alhálózatot tartalmazhat, melyek mindegyike 264 csomóponttal rendelkezhet (64 bit). Ez jóval több, mint a teljes mai Internet! Ez a címtér

© Kiskapu Kft. Minden jog fenntartva

a felhasználóhoz van rendelve, így az ügyfél gép IPv4-címének változása esetén is megmarad. A felhasználók és szervezetek számára óriási szabadságot nyújt, hogy címek millióit rendelhetik a kiszolgálókhoz és a szolgáltatókhoz – ezt egyáltalán nem volt könnyű megvalósítani az IPv4 alatti NAT-tal.

Az IPv6-over-IPv4 csatorna felépítéséhez mindkét oldali IPv4 és a másik oldalon IPv6 cím-beállításokat igényel. Amikor valamelyik oldal megváltoztatja az IPv4-címét, ennek megfelelően a csatorna mindkét oldalán meg kell változtatni a beállításokat. Ez különösen akkor nehézkes, amikor az IPv4-csomópont betárcsázásos alapon működik, vagy gyakran változtat címet. A Freenet6 szolgáltatásban megvalósított TSP beállítható úgy, hogy figyeljen ezekre a dolgokra.

Valahányszor a csatorna ügyfele megváltoztatja az IPv4-címét, például rendszerindításkor a DHCP-szolgáltatás segítségével, a TSP-ügyfél elküldi a módosított és hitelesített adatot a kiszolgálónak, így a csatorna továbbra is működőképes marad. A Freenet6 ügyfélszolgáltatás által támogatott rendszerek: Linux, FreeBSD, OpenBSD, NetBSD, Windows, Solaris és Cisco. Az 1. ábra a Freenet6 alapszerkezetét mutatja be.

Az IPv6 telepítését követően Linux alatt a következő lépéseket kell megtennünk, hogy használhassuk a Freenet6 szolgáltatást. Először is lépünk fel a <http://www.freenet6.net> oldalra, és jegyezzük be magunknak egy felhasználói nevet. Töltsük le a linuxos TSP-ügyfelet! Kövessük a fordításra és telepítésre vonatkozó utasításokat. Ezután állítsuk be a mellékelt `tspc.conf` fájlt. Írjuk be a felhasználói nevet és a jelszót, végül indítsuk el a TSPC-ügyfelet:

```
tspc -vf tspc.conf
```

A `tspc` ügyfélparancsot berakhatjuk a rendszerindító folyamatba is, így a csatorna minden egyes rendszerindítás után önműködően újra felépül, mégha az IPv4-címünk meg is változott.

A Freenet6 egyaránt adhat nekünk egyetlen IPv6-címet (ha géphez kérjük), vagy egy teljes /48 tartományt (amennyiben útválasztóhoz kérjük). A Freenet6 úgy állítja be a Linuxot, hogy megfeleljen a feladatnak.

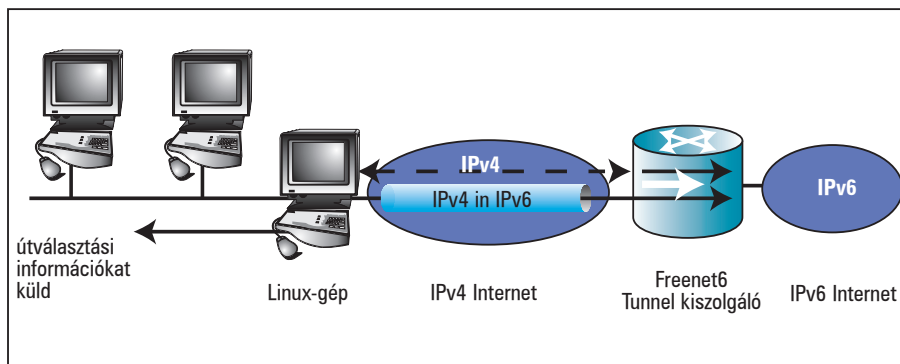
Útvonalválasztó esetében /48 formájú címet kapunk, ahol az útvonalválasztónk első alhálózatának beállítása útvonalválasztó üzenetszórásos típusú lesz. Ez azt jelenti, hogy az ezen az alhálózaton található gépek megkapják az előtagot (prefix), és önműködően beállíthatják magukat, ahogyan azt a 2. ábrán láthatjuk.

Az IPv6 Internethez sokféle módon kapcsolódhatunk. A Freenet6 a TSP-protokollal együtt könnyű, IPv6-csatornán alapuló kapcsolatot nyújt nekünk, állandó címtérrel, ahol ha IPv4-címünk meg is változik, IPv6-címünk és a kapcsolatunk változatlan marad.

Összefoglalás

Tevékenységünk részeként az Ericsson Research Canada Open Architecture Research részlegénél néhány IPv6-tal kapcsolatos projektet futtatunk, többek között foglalkoztunk telecom-fokozatú kiszolgáló-csomópontjaink IPv6-támogatásával, alkalmazáskiszolgálók IPv6 alapú működésének megvalósításával, illetve kutatási projekteket hozunk létre különféle IPv6-területeken.

Az egyik ilyen érdekes művelet a jelenleg elérhető Linux IPv6-megvalósítások kipróbálása volt, illetve annak eldöntése, hogy melyik változatot érdemes Linux-gépeinkre átültetni. A javaslatokat IPv6-megvalósítások karakterisztikája alapján tettük,



2. ábra Freenet6-szerkezet útválasztóval és több géppel

ideértve a fejlesztés ütemét, a szabványoknak való megfelelést, illetve a teljesítményüket más megvalósításokhoz képest.

Az eredményeket az ottawai Linux Symposiumon mutattuk be Kanadában (http://www.linuxsymposium.org/2002/view_txt.php?text=abstract&talk=93).

2002-ben tovább folytatjuk IPv6-kutatásainkat Linux-telepeinken. Jelenleg erőfeszítéseinket IPv6 alapú linuxos SIP- és SCTP-megvalósításokra összpontosítjuk, illetve szeretnénk egy IPv6 teljesítménymérő környezetet felállítani, amivel meghatározhatjuk az egyes IPv6 alatt futó gépek és alkalmazások teljesítményét és méretezhetőségét.

Végezetül rámutatnék arra, hogy az IPv6 legfőbb ereje a címekben rejlik. Bárki kaphat egy akkora tartományt, mint a teljes jelenlegi IPv4-címtér. A felhasználók ezáltal tökéletes szabadságot kapnak a kiszolgálók és szolgáltatások telepítése terén, anélkül, hogy NAT-gondokkal vagy a korlátozott címtérrel kellene küszködniük. Üdvözöllek az internetes szolgáltatások telepítésének szabad világában!

Köszönetnyilvánítás

Szeretnénk köszönetet mondani az Ericsson Open Architecture Researchnek e cikk megjelenésének támogatásáért, a Canarie, Inc.-nek (<http://www.canarie.ca>) a Freenet6 Projekt támogatásáért, illetve David Gordon-nak (David.Gordon@Ericsson.ca) azért a segítségért, amit a ECUR Lab IPv6 csomópontjainak beállításában nyújtott, illetve a cikk szerkesztéséért és átnézéséért.

Linux Journal 2002. augusztus, 100. szám



Ibrahim Haddad

(Ibrahim.Haddad@Ericsson.com) jelenleg kutatóként dolgozik a montreali Ericsson Corporate Research Unit részlegénél Kanadában.



Marc Blanchet

Viágénie-ben dolgozik egy IP, IPv6 módszerekre és hálózati biztonságra szakosodott tanácsadó cégnél. 1995 óta foglalkozik az IPv6-tal és számos IETF dokumentumot írt már a témában.