

**Mérési utasítás****IPv6**

A Távközlés-informatika laborban natív IPv6 rendszer áll rendelkezésre. Először az ún. állapotmentes automatikus címhozzárendelést (SLAAC, stateless address autoconfiguration) vizsgáljuk meg, melynél előny, hogy kliens oldalon nem szükséges beállításokat végrehajtani. A mechanizmus alapja az ICMPv6-os protokoll Router Advertisement csomagjaira alapoz, melynek segítségével a kliens magától generál egy IPv6 címet saját MAC címéből.

**1. feladat**

Indítson csomagelkapást wireshark segítségével az eth0-es interfészen addig, amíg el nem kap egy „Router Advertisement” csomagot – az egyszerűség kedvéért alkalmazhatjuk az „icmpv6” szót a megjelenítési szűrőnél.

Válaszoljon az alábbi kérdésekre:

- Milyen IPv6-os címre küldi a router az ICMPv6-os csomagot?
- Milyen hosszúságú prefixet hirdet a router?
- Milyen ICMPv6-os flagek aktívak? (1)
- Hirdeti-e a helyi DNS kiszolgálókat a Router Advertisement csomag?

**2. feladat**

Ahhoz, hogy saját magunk konfiguráljunk IPv6 címet, meg kell adnunk, hogy kliensünk figyelmen kívül hagyja ezeket a csomagokat. Ehhez a következő parancsokat kell begépelnünk:

```
ifconfig eth0 down
echo 0 > /proc/sys/net/ipv6/conf/eth0/accept_ra
echo 0 > /proc/sys/net/ipv6/conf/eth0/autoconf
ifconfig eth0 0.0.0.0 up
```

Ezzel töröltük az aktív (eth0) interfész beállításait.



### 3. feladat

Állítson be egy statikus IPv6 címet az eth1 interfésznek. Annak érdekében, hogy érzékeltesük az IPv6 címtartományának nagyságát, az általunk beállítandó IPv6 címbe foglaljuk bele a mai dátumot és a gyakorlat időpontját, valamint a gépszámot is!

**Pl.:**

```
ifconfig eth0 inet6 add 2001:738:2c01:8001:2013:0325:1440:3/64
```

Ahol

- A dőlt betűs rész nem változtatható!
- A vastagon szedett rész a mai dátum és gyakorlat időpont!
- az aláhúzott rész a gépszám

Ezzel a helyi IPv6 hálózat elérhetővé vált. Ahhoz, hogy az IPv6 globális hálózata elérhetővé váljon, szükségünk van egy helyi átjáróra. Ezt a `route` parancs segítségével adhatjuk meg, mely parancs a Linux routing táblájának módosítására szolgál:

```
route -6 add default gw 2001:738:2c01:8001::1
```

Ezzel megadtuk, hogy „merre” kell a csomagot az internet felé irányítani. Most már elérjük az egész IPv6-os hálózatot, de ahhoz, hogy ne kelljen IPv6 címeket használni a különböző hálózati szolgáltatások eléréséhez, névfeloldást kell alkalmaznunk. Ehhez meg kell mondani, milyen IPv6 címen található a névkiszolgáló. Ezt a Linux alatt a `resolv.conf` fájlba egy „nameserver” bejegyzéssel tehetjük meg:

```
echo „nameserver 2001:738:2c01:8001::1” > /etc/resolv.conf
```

Most indítsuk el a `wireshark`-ot, és állítsunk be egy csomagvizsgálatot az eth0-re. Majd ezek után vegyük használatba az IPv6 hálózatot, első körben egy pingeléssel. Az IPv6 alatt, mivel más a hálózati protokoll, más programokat is kell használni, mint IPv4-nél. Jelen esetben ez a `ping6`:

```
ping6 -c 2 2001:738:2c01:8001::1
```

Az IPv6-ban nem található ARP lekérdezés. Az IPv6 tervezése közben a fejlesztők inkább az ICMP protokollba integrálták az IPv6 cím → MAC cím leképzést (ez az ICMPv6), valamint kiegészítették egy Neighbor Discovery Protokollal (NDP).

### 4. feladat

Keresse meg a `wireshark`-ban a Neighbor Discovery Protokoll csomagjait. (Neighbor solicitation Neighbor advertisement) valamint az ICMPv6 protokoll echo request, echo replay csomagjait!



5. feladat

Nyisson egy tetszőleges web böngészőt, majd nyissa meg a **http://www.kame.net** honlapot!  
Amennyiben minden megfelelően működik, a weblapon mozog a teknős.

6. feladat

Kérdezze le a `tracert` IPv6-os verziójával a **ipv6.google.com** szerverhez vezető utat!

```
tracert6 ipv6.google.com
```

**Mérési utasítás****DNS****Cél:**

A DNS (Domain Name System) rendszer működésének megismerése, használata.

A DNS egy hierarchikusan felépített rendszer mely a számítógépek, és bármely aktív hálózati eszköz elnevezhető egy szimbolikus névvel, ez által nem kell az eszközhöz tartozó IP címet megjegyezni.

Ahhoz, hogy a következő parancsok működjenek, ismét vissza kell térnünk a IPv4 világába!

Ezt a következő parancsokkal tehetjük meg:

```
dhclient -v eth0
```

**1. feladat**

Kérdezze le a root (.) névkiszolgálókat!

```
host -t NS .
```

Amennyiben ez a parancs hibaüzenetet ad, nagy valószínűséggel nincs telepítve a `host` parancs.

Ezt a következőképpen pótolhatjuk: `apt-get install host`

Ezzel a Linux kilistázza az elérhető root névkiszolgálókat. Ezek a névkiszolgálók felelősek a gTLD, és ccTLD domain-ekért. Ezek a szerverek mondják meg, hogy mondjuk a '.hu' domainért ki felel.

**2. feladat**

Kérdezze le a '.hu' domain névkiszolgálóit!

```
host -t NS hu.
```

A most kilistázott névkiszolgálók felelnek a '.hu' domainért. Egy domain-ért több névkiszolgáló felelhet.

A `host` parancs nemcsak az aktuális domain-hez tartozó névkiszolgálót tudja lekérdezni, hanem a domain-hez tartozó más rekordokat is. Ilyenek lehetnek: A, MX, SOA, CNAME, HINFO.

A – a tulajdonos IP címe

MX – Mail Exchange, a domainhez tartozó levélszolgáltató

SOA – State of authority, hitelességi specifikációk

CNAME – egy létező A rekordhoz kanonikus név hozzárendelése

HINFO – CPU és Operációs Rendszer meghatározása



### 3. feladat

Kérdezze le a tilb.sze.hu domain levélkiszolgálóit.

```
host -t MX tilb.sze.hu.
```

Egy adott domainben több levélkiszolgáló is lehet. Ezeket prioritásba kell állítani. Ezért az MX rekordhoz mindig tartozik egy számérték is, minél kisebb ez a szám annál nagyobb a prioritása a levélkiszolgálónak. A legkisebb számmal rendelkező gép a „root” levélkiszolgáló.