MŰEGYETEM 1782

IPv6 TRANSITION

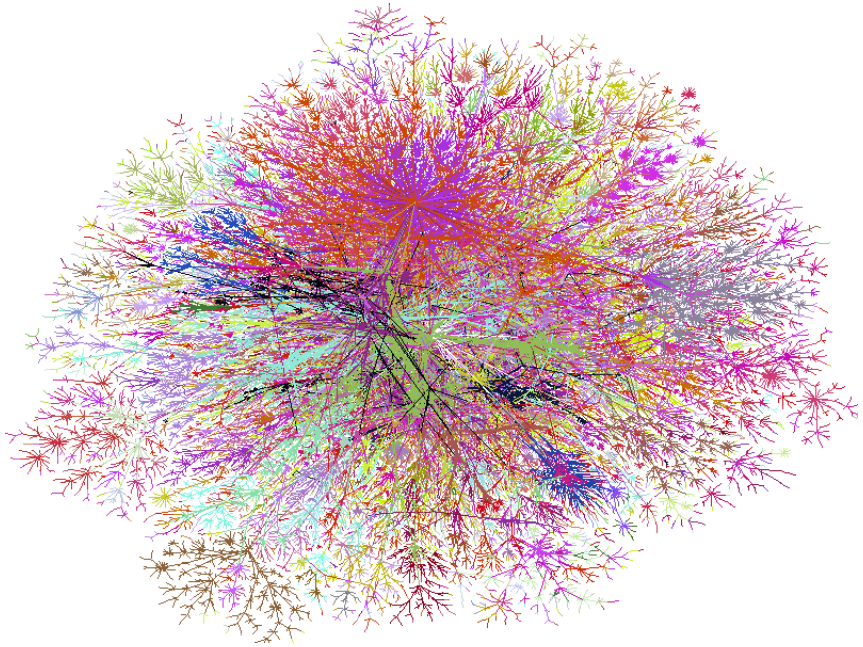
Számítógép-hálózatok (BMEVIHIA215)

Dr. Lencse Gábor
tudományos főmunkatárs
BME Hálózati Rendszerek és Szolgáltatások Tanszék
lencse@hit.bme.hu

2014. április 9.,
Budapest

Tartalom

- Az IPv4 → IPv6 átmenet jellemzői és eszközei
- Kitérő: hálózati címfordítás
- DNS64 + NAT64
- További megoldások



http://cheswick.com/ches/map/gallery/wired.gif

Az IPv4 → IPv6 átmenet jellemzői és eszközei

Az áttérés időzítése

- Áttérés „óraütésre”
  - Az Internet történelmében egyszer sikerült (RFC 801)
    - ARPANET, 1983. 01. 01. áttérés NCP-ről (Network Control Program) TCP/IP-re
  - Ma lehetetlen feladat
    - Több milliárd csomópont (lehetetlen egyszerre „átkapcsolni”)
    - Hibák biztos lennének (jelenleg rengeteg hardver és szoftver eleve alkalmatlan)
    - Hatalmas káosz lenne (pénzügyi, gazdasági,..)
- Hosszú idejű átmenet
  - A két protokoll tartós egymás mellett élésével
    - Bizonyos hardver és szoftver szállítók nem is fogják megoldani az IPv6 kompatibilitást
    - A régi eszközökhöz a felhasználók ragaszkodnak
  - Meg kell oldani az együttműködésüket

IPv4 – IPv6 együttműködése – 1

- Sok alkalmazásunk kliens-szerver konfigurációban működik
  - Mely protokollokra képes a kliens és a szerver?
  - Mely protokollokra képes a hálózat a kettő közötti úton?
- Az egyszerű eset
  - Ha a kliens és a szerver közül bármelyik is képes mindkét protokoll használatára (*dual stack*), akkor a „közös nyelv” használatával a kommunikáció megoldott – feltéve, hogy a hálózat is támogatja. ☺
  - De már nincs elég IPv4 cím! ☹
    - Kényszermegoldás: *Dual-Stack Lite* – bővebben nem foglalkozunk vele.
- IPv6 képes kliens IPv4-only környezetben és IPv6 szerver
  - A kliens képes IPv6-ra de az ISP csak IPv4-címet ad a kliensnek
  - Egy jó megoldás: 6to4 használata
    - Megismerjük

IPv4 – IPv6 együttműködése – 2

- IPv6 kliens és IPv4 szerver
  - Csak IPv6-ra képes kliens (már csak IPv6 cím jutott neki)
  - Csak IPv4-re képes szerver (mert régi, az IPv6-ot nem támogatja)
  - Egy jó megoldás: DNS64 szolgáltatás + NAT64 átjáró használata
    - Megismerjük
- IPv4 kliens és IPv6 szerver
  - Csak IPv4-re képes kliens (régi hardver és/vagy szoftver)
  - Csak IPv6-ra képes szerver (ilyenek is vannak, és számuk nőni fog)
  - Egy megoldás lehet (talán): NAT46 + DNS46
    - Még nem kiforrott, bővebben nem foglalkozunk vele

IPv4 – IPv6 együttműködése – 3

- IPv6 kliens és IPv6 szerver DE útközben csak IPv4 van
  - Tipikus eset, az új IPv6 „szigeteket” össze kell kötni
  - Az IPv6 datagramok szállítása IPv4 fölötti „alagútban” (6in4 tunnel)
    - Röviden megnézzük az elvi megoldást
- IPv4 kliens és IPv4 szerver DE útközben csak IPv6 van
  - Ma még nálunk nem igazán jellemző, de majd lehet
  - Az IPv4 datagramok szállítása IPv6 fölötti „alagútban” (4in6 tunnel)
    - Bővebben nem foglalkozunk vele

KITÉRŐ: HÁLÓZATI CÍMFORDÍTÁS

Network Address Translation – 1

- A TCP/IP protokollcsaládot eredetileg végponttól végpontig való kommunikációra tervezték
- Az alkalmazások arra számítanak, hogy a címek és portszámok a hálózati átvitel során változatlanok
- Az IPv4-címek szűkössége miatt elterjedt megoldás:
  - egy szervezet hálózatában privát IP-címeket használnak
  - a külső kommunikációhoz *címfordítás*t használnak
- A külső kommunikáció feladata lehet:
  1. A privát IP című gépeknek el kell érniük az Internetet
  2. Az Internet felől el kell érni valamely privát IP című gépet
- A feladatok megoldásához rendelkezésünkre áll egy router, amely rendelkezik publikus IP-címmel

Network Address Translation – 2

- Alapötlet az Internet eléréséhez:
  - A célcím alapján küldjük el a kérést a publikus Internet felé
  - A kimenő router cserélje ki a forrás IP-címét a saját publikus IP-címére
  - A csomag megérkezik a címzetthez, és a válasz visszaér a routerhez.
  - A router továbbítsa a választ a forrásnak – DE HOGYAN?

Network Address Translation – 3

- Ahhoz, hogy a router a választ az eredeti feladónak vissza tudja küldeni, nyilván kell tartania, és fel kell ismernie, hogy ki volt az eredeti küldő.
  - Ennek érdekében az IP-címen túl mást is felhasznál. TCP és UDP esetén ezek a forrás portszámok
  - Nem elegendő ezeket megjegyezni, hiszen a forrás portszámok csak *gépenként egyediek*, a forrás IP-címet viszont a sajátjára cseréli
  - A forrás portszámokat kicseréli a *routeren egyedi portszámokra*
  - Az IP-címek és a célportszám mellett ezekkel már egyértelműen azonosítani tudja a kapcsolatokat
  - Kapcsolatonként nyilvántartja, hogy mit mire cserélt ki
  - A bejövő csomagoknál a cél IP-címen kívül a cél portszámot is vissza kell cserélnie ^(változott az irány)^

Network Address Translation – 4

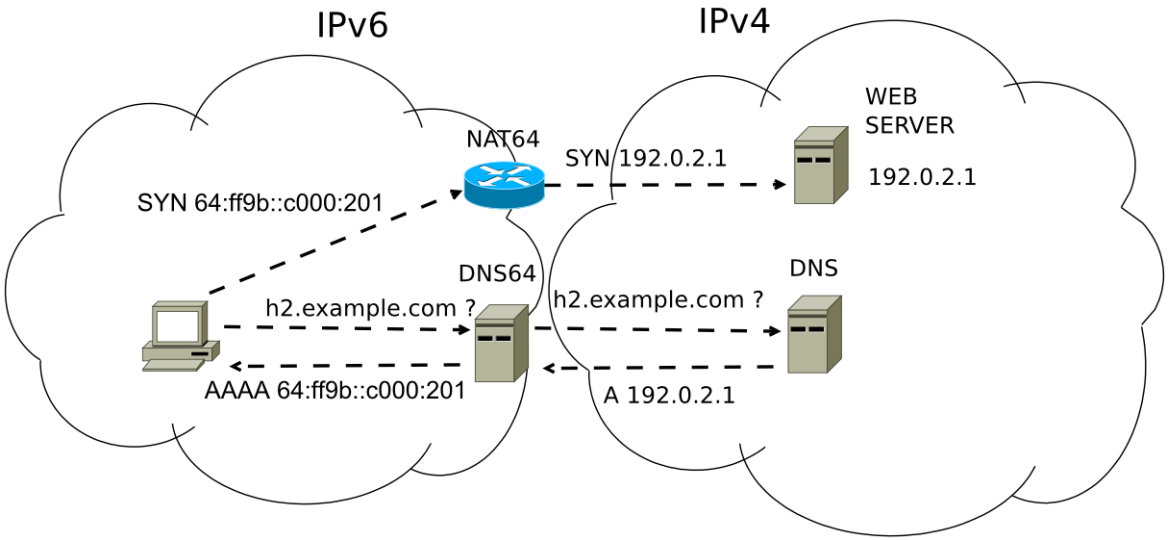
- Az ismertetett megoldást hívjuk *Source NAT*-nak (SNAT) akkor, ha a router publikus IP-címe fix, és Masquerade-nek, ha DHCP-vel kapta az interfésze a címet
- Megjegyzés: terminológia nem egységes
  - Eredetileg a NAT csak az IP-címek cseréjét jelentette, ezt hívják ma *basic NAT*-nak, vagy *one-to-one NAT*-nak.
  - A fenti megoldás precíz neve a *NAPT* (Network Address and Port Translation). Nevezik *many-to-one NAT*-nak is.

Network Address Translation – 5

- A másik irányú feladat az, hogy pusztán privát IP-címmel rendelkező gépeket elérhetővé tegyünk az Internet felől.
  - Erre a megoldás a *Destination NAT* (DNAT) vagy más néven port forwarding, ahol a router az adott portjára érkező datagramokat egy meghatározott privát IP című gépnek továbbítja úgy, hogy a célcímet kicseréli a csomagban.
    - Például a 80-as portra érkező datagramokat a 10.1.1.2 IP-című webszerver, a 25-ösre érkezőket pedig a 10.1.1.3 IP-című SMTP szerver felé továbbítja
- Mi helyzet az ICMP üzenetekkel?
  - ICMP esetén nincs portszám, de segíthet az, hogy egy hibaüzenetben benne van az azt kiváltó TCP vagy UDP adategység első 64 bitje a portszámokkal.
  - Amennyiben nem hibaüzenetről van szó, akkor is van megoldás, az ICMP üzenet valamilyen azonosító jellegű mezőjét használják fel.

Network Address Translation – 6

- Mindkét irányú megoldásnál gondot okozhat, ha az alkalmazások számítanak a címek és portok változatlanságára – ami részükről jogos elvárás.
  - Például egy FTP kliens aktív módban a vezérlő kapcsolaton keresztül megadja a szervernek, hogy mely portján várja, hogy a szerver felépítse az adatkapcsolatot.
  - Privát IP címmel várhatja – hacsak nem segít valaki: *protocol helper*
- Ez a probléma NAT64-nél is előjön.
- Érdeklődőknek:
  - A NAT-ról bővebben az RFC 3022-ben olvashatunk, az IP, TCP, UDP, ICMP fejrészek mezőinek módosításával a 4.1. rész, az ellenőrző összeg hatékony újraszámításával a 4.2. rész foglalkozik.

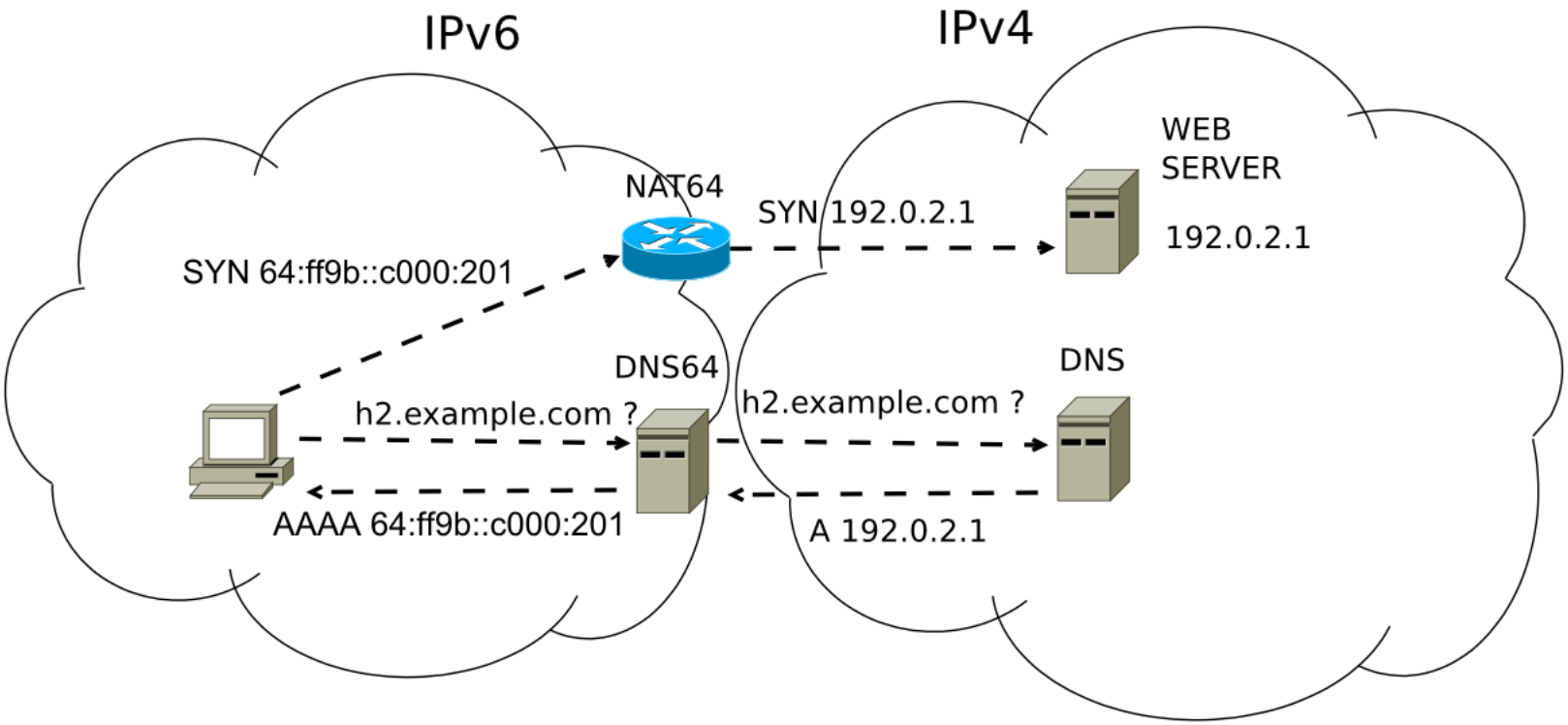


Forrás: http://en.wikipedia.org/wiki/File:NAT64.svg (javítva)

DNS64 + NAT64

IPv6-only kliens és IPv4-only szerver

- Az IPv4 címek kifogyása miatt várhatóan ez lesz az első tipikus megoldandó probléma
  - Az elvi megoldást az alábbi példán mutatjuk be
    forrás: http://en.wikipedia.org/wiki/File:NAT64.svg (javítva)



DNS64 szerver

- Mint egy caching-only name server, *Recursive query*kre válaszol
  - Ha van IPv6 cím, akkor továbbítja a válaszában
  - Ha nincs IPv6 cím, csak IPv4, akkor az IPv4 címből generál egy speciális IPv6 címet, és ezt adja vissza
- A speciális IPv6 cím a példában használatos well-known prefix esetén
  - A 64:ff9b::/96 prefix + az utolsó 32 biten a kapott IPv4 cím

A NAT64 átjáró használata – 1

- A megoldás működéshez szükséges:
  - A kliensben névkiszolgálóként a DNS64 szerver van beállítva
  - Az útválasztási táblázatok szerint a well-known prefix felé az út egy NAT64 átjárón keresztül vezet (Anycast címzés használható)
- Kövessük végig a példát!
  - Az IPv6-only kliens csatlakozni szeretne az IPv4-only szerverhez
  - Lekéri a szerver IPv6 címét a szimbolikus neve alapján
  - Megkapja a szerver IPv4 címét tartalmazó speciális IPv6 címet
  - TCP SYN szegmenst tartalmazó IPv6 csomagot küld a kapott címre
  - Az IPv6 csomag megérkezik a NAT64 átjáróhoz
  - Az átjáró az IPv6 csomag alapján egy IPv4 csomagot készít, benne
    - A célcím az IPv6 cím utolsó 32 bitje
    - A forráscím a NAT64 átjáró IPv4 címe
  - Az átjáró a csomagot elküldi a címzettnek

A NAT64 átjáró használata – 2

- Tovább követjük a példát…
  - Az IPv4-only szerver megkapja a TCP SYN szegmenst tartalmazó IPv4 csomagot és a megszokott módon válaszol (SYN+ACK)
    - A kapott IPv4 csomagban a forráscím a NAT64 átjáró IPv4 címe volt
    - A válasz címzettje is a NAT64 átjáró IPv4 interfésze lesz
  - A választ megkapja a NAT64 átjáró és elkészíti a neki megfelelő IPv6 csomagot, melybe
    - Forráscímként az a speciális IPv6 cím kerül, amit a DNS64 szerver generált
    - Célcímként az IPv6-only kliens IPv6 címe kerül
    - Mindez a NAT által korábban is használt módon, kapcsolattábla alapján történik
  - Az IPv6 csomagot a NAT64 átjáró elküldi az IPv6-only kliensnek
  - Az IPv6-only kliens megkapja a csomagot
  - A kommunikáció a fentiek szerint tovább folytatódik…

A NAT64 átjáró használata – 3

- A fentiekben bemutatott megoldás az RFC 6052 szerinti 64:ff9b::/96 előre lefoglalt, ún. *Well-Known Prefix*et használja.
- A gyakorlatban számos hátránnyal járna, ha világszerte mindenki ezt a prefixet használná (RFC 6052 3.1 és 3.2)
- A NAT64 átjáró megvalósításakor a gyakorlatban az egyes IPv6 hálózatokból szoktak erre a célra lefoglalni egy részt, ezt *hálózat specifikus prefix*nek (network specific prefix) nevezik.
- Az ilyen IPv4 címeket tartalmazó IPv6 címeket úgy hívják, hogy *IPv4-Embedded IPv6 Addresses*

IPv4-Embedded IPv6 címek fajtái

- Az IPv4-címeket beágyazó IPv6-címeket még két további névvel illetik a felhasználásuk céljától függően, bár szerkezetük és előállításuk azonos.
  - Azokat az IPv6 címeket, amiket arra használunk, hogy IPv4 állomásokat képviseljenek IPv6 hálózatokban, úgy nevezik, hogy IPv4-Converted IPv6 Address. (Jelenleg pontosan erről van szó.)
  - Az IPv4-Translatable IPv6 Address nevet pedig akkor használjuk, ha a cím egy IPv6 állomáshoz tartozik, és a fordítás célja, hogy a csak IPv4-re képes eszközök is el tudják érni az IPv6 állomást. (Ezzel az esettel most nem foglalkozunk.)

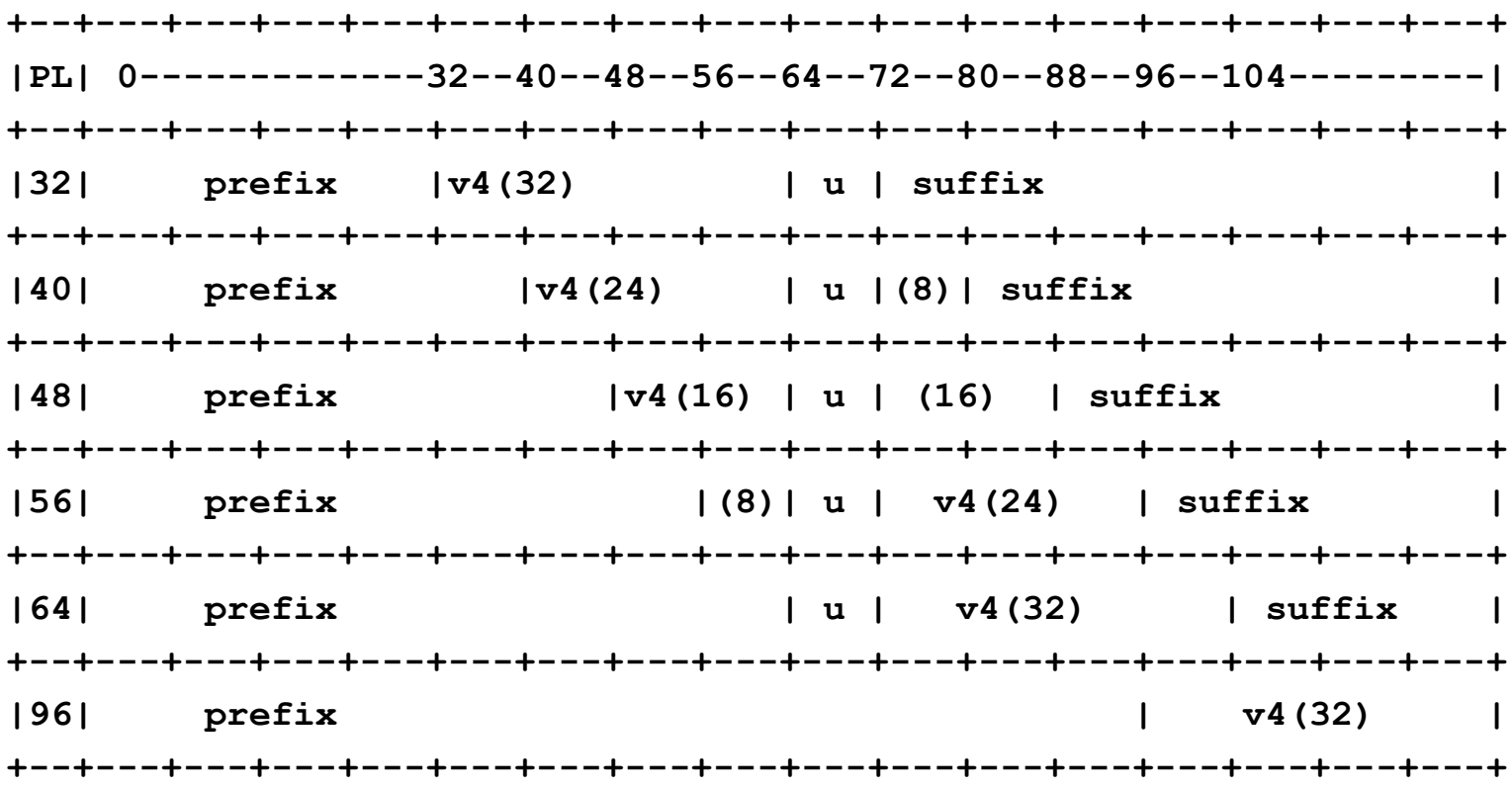
IPv4-Embedded IPv6 címek képzése – 1

- Az RCF 6052 definiálja, hogy a prefix méretétől függően hogyan kell az IPv4-címeket beágyazó IPv6-címeket képezni.
  - A prefix mérete szigorúan csak 32, 40, 48, 56, 64 vagy 96 lehet
  - Az IPv6 címben 64-71 biteknek 0-nak lenniük
  - Az IPv6 cím 32 bitjét általában a prefix után írjuk, de a fenti követelmény kielégítése érdekében a szigorúan 0 értékű bitek helyét „átugorjuk”
  - A cím végét szükség esetén ugyancsak 0 értékű bitekkel töltjük ki

(Ábra a következő oldalon)

IPv4-Embedded IPv6 címek képzése – 2

- A címek lehetséges formátuma
  - PL: prefix length, v4: IPv4 cím bitjei, u és suffix: 0 értékű bitek



TOVÁBBI MEGOLDÁSOK

A 6in4 tunnel elve

- A feladat:
  - IPv6 szigetek csak IPv4 hálózaton keresztül tudnak egymással kommunikálni
- IPv6-over-IPv4 (RFC 4213)
  - Az IPv6 csomagokat IPv4 csomagokba csomagolja be
  - Az IPv4-ben az 41-es protokoll azonosítót használja az IPv6 csomagok azonosítására
- A be- és kicsomagolást az IPv6 szigetek határán levő átjárók végzik

A 6to4 megoldás – 1

- A megoldandó feladat:
  - IPv6 képes eszköz IPv4-only környezetben van
  - IPv6 protokollal egy másik IPv6-os eszközt szeretne elérni
    - akár az is lehet IPv4-only környezetben
- A 6to4 megoldás (RFC 3056) egy „automatikus” tunnel, ami az IPv6 csomagokat IPv4 csomagokba csomagolja be
  - A 6in4-hez hasonlóan a 41-es protokoll azonosítót használja
- Megvalósítás szempontjából kétféle konfiguráció lehetséges
  - Egyetlen host, amin IPv6 kliens fut, és a becsomagolást is a host végzi – a 6to4 pseudo-interface a hoston van (*6to4 host*)
  - Egy IPv6 hálózat, aminek a routere végzi a becsomagolást – a *6to4 (border) router* rendelkezik 6to4 pseudo-interface-szel.
- A kicsomagolás a *6to4 relay* feladata
  - Ilyen több is lehet, legközelebbi a 192.88.99.1 anycast címen

A 6to4 megoldás – 2

- A megoldás működésének lépései, ha a cél natív IPv6
  - A kliens a szerver felé IPv6 csomagot küld
  - A becsomagolás elvégzője (host vagy router) az IPv6 csomagot IPv4 csomagba ágyazza, és az IPv4 segítségével elküldi egy általa választott *6to4 relay*nek (saját döntése, hogy kinek címzi)
  - A 6to4 relay megkapja az IPv4-be ágyazott IPv6 csomagot, kicsomagolja és továbbítja a natív IPv6 hálózatba
  - A natív IPv6 hálózatban a csomag megérkezik a címzetthez
  - A címzett válaszol
  - A válasz visszaér vagy az előbbi vagy egy másik 6to4 relayhez
  - Visszafele a 6to4 relay az IPv6 csomagokat IPv4-be csomagolva küldi a korábban a kliens IPv6 csomagját IPv4-be csomagoló eszköznek (host vagy router). – Ennek publikus IPv4 címe van!
  - A host vagy router kicsomagolja az IPv4 csomagból az IPv6 csomagot és eljuttatja a kliensnek.

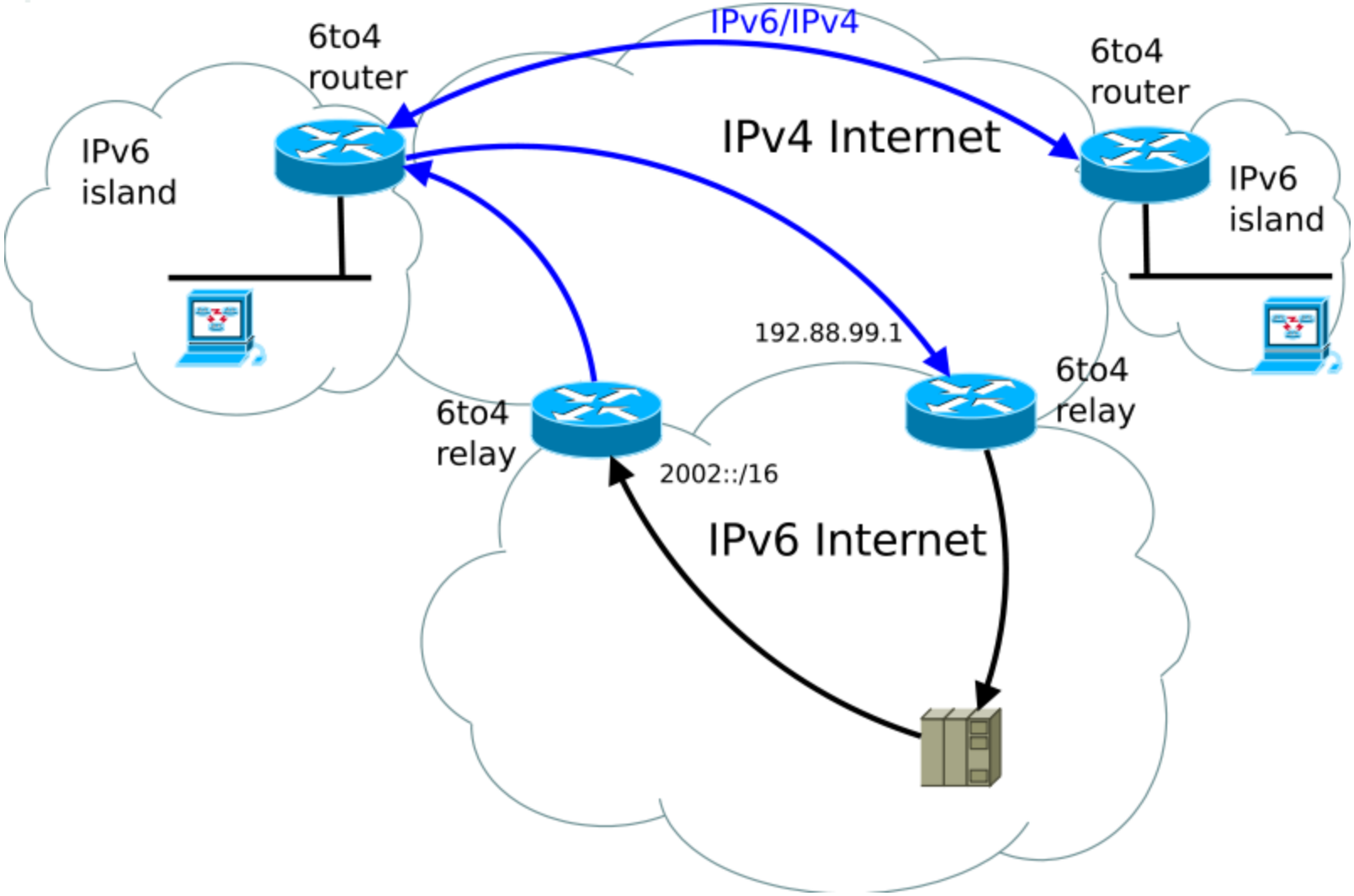
A 6to4 megoldás – 3

- Működéséhez szükséges feltétel: a kliensnek legyen publikus IPv4-címe
  - Ha nincs neki, akkor *Teredo*t kell helyette használni. (RFC 4380)
  - A 6to4 továbbfejlesztett változata a *6rd* (RFC 5969)
- Példánkban egy IPv4 környezetben működő IPv6 host natív IPv6 Interneten levő szervert ért el.
- A 6to4 segítségével IPv6 szigeteket is összeköthetünk IPv4 fölött.
  - Ebben az esetben a becsomagolást végző eszköz a cél IPv6-címéből (6to4 cím) tudja meg, hogy nem egy IPv6 relaynek, hanem a megfelelő IPv4 című pseudo-interfésszel rendelkező eszköznek kell IPv4 szinten címeznie és küldenie a csomagot.

A 6to4 megoldás címzése

- A 6to4 címzéshez 2002::/16 prefixet foglalták le
- A címek képzése
  - Hálózatcím:
    - 2002::/16 prefix + publikus IPv4 cím 32 bitje + 16 bit subnet ID
    - Host esetén a subnet ID egy generált véletlenszám
    - Ha a 6to4 mechanizmust router használja, akkor akár több IPv6 hálózat is lehet mögötte, ekkor hasznos a subnet ID.
  - Gépcím:
    - A szabványos módosított EUI-64 azonosító
- Ilyen módon a 6to4 minden publikus IPv4 címhez egy 2002::/16 kezdetű, /48 méretű IPv6 címtartományt rendel
  - Mindegyik „mögött” elérhető lehet egy ilyen méretű IPv6 hálózat

A 6to4 megoldás lehetőségei



Forrás: http://en.wikipedia.org/wiki/File:6to4.svg

Összefoglalás

- Az IPv4 és IPv6 még sokáig együtt fog élni.
  - Az együttműködésnek számos fajtája van/lesz.
  - Még nagyobb a megoldások száma.
- Megismertük:
  - DNS64+NAT64: IPv6-only kliens eléri az IPv4-only szervert
  - 6to4: IPv4 környezetben levő IPv6 eszköz eléri a natív IPv6 Internetet vagy egy másik hasonló eszközt
  - 6in4 tunnel (csak elvi szinten): IPv6 szigetek IPv4 fölött képesek egymással kommunikálni

Kérdések?

KÖSZÖNÖM A FIGYELMET!

Hálózati Rendszerek és Szolgáltatások Tanszék

Dr. Lencse Gábor
tudományos főmunkatárs
BME Hálózati Rendszerek és Szolgáltatások Tanszék
lencse@hit.bme.hu