

IPv6 TRANSITION

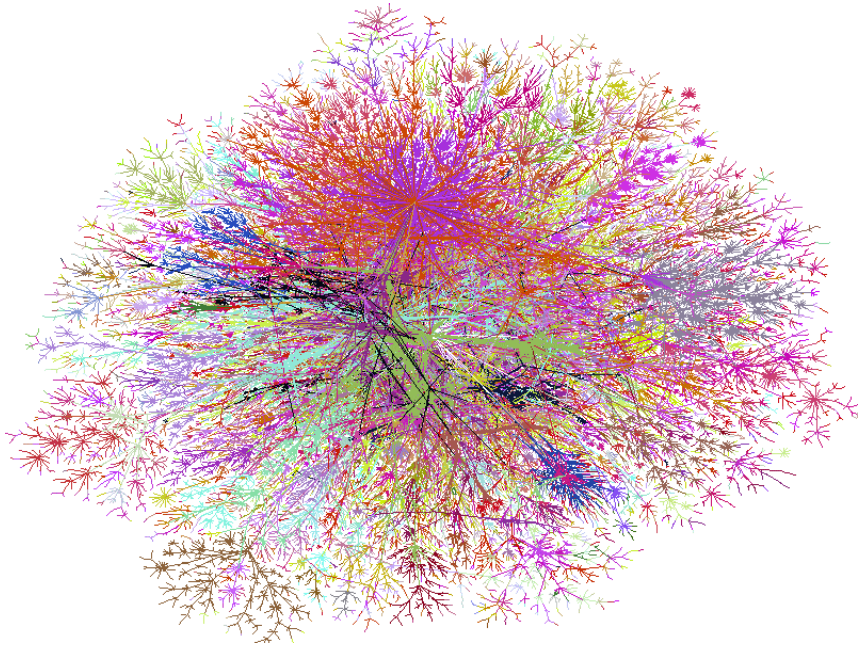
Kommunikációs hálózatok I. (BMEVIHAB01)
2016. évi fóliái alapján készült

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

2020. április 6.,
Budapest

Tartalom

- Az IPv4 → IPv6 átmenet jellemzői és eszközei
- Kitérő: hálózati címfordítás
- DNS64 + NAT64
- IPv4aaS (IPv4, mint szolgáltatás) megoldások
- 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! ☹
    - Megnézzük, mit lehet tenni: IPv4aaS megoldások
- IPv6 képes kliens IPv4-only környezetben és IPv6 szerver
  - A kliens képes IPv6-ra de az ISP csak IPv4-címet ad a kliensnek
  - Egy működő, de problémákkal járó megoldás: 6to4 használata
    - Idén már csak röviden foglalkozunk vele…
    - Van még a Teredo, illetve nem erre való a 6rd: röviden megemlítjük őket

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
    - Részletesen 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 lehetne (inkább: lehetett volna): DNS46 + NAT46
    - Az Internet Draftból nem lett RFC, de léteznek implementációk
    - Idén nem foglalkozunk vele, érdeklődőknek: IPv6 könyv

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
  - Mindkét fenti problémára megoldást nyújt, de sokkal többre képes az *MPT hálózati szintű többutas kommunikációs könyvtár* ☺
    - Az MPTCP-hez hasonlítható, de GRE-in-UDP alapú
    - Bármely IP verzió fölött bármely IP verzió szállítható
    - Idén még nem foglalkozunk vele, érdeklődőknek: IPv6 könyv

IPv4 – IPv6 együttműködése – 4

- Csak IPv4-re képes alkalmazások, DE az ISP csak IPv6-ot szeretne használni
  - Vannak IPv6-ra nem képes alkalmazások, amelyekhez az előfizetők mégis ragaszkodnak (például Skype, Spotify, stb.)
  - Az ISP a hálózatában csak IPv6-ot szeretne használni (kevesebb adminisztráció, mint a dual stack, különösen mobil esetben tipikus)
  - Számos IPv4aaS megoldás létezik, az öt legfontosabb:
    - 464XLAT
    - DS-Lite (Dual-Stack Lite)
    - Lw4o6 (Lightweight 4over6)
    - MAP-E (Mapping of Address and Port with Encapsulation)
    - MAP-T (Mapping of Address and Port using Translation)
- És még rengeteg más is van, érdeklődőknek
  - http://www.hit.bme.hu/~lencse/publications/e102-b_10_2021.pdf

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 privát IP-címmel rendelkező kliens küldje el az IP datagramot a publikus IP-címmel rendelkező szerver felé
    - Forráscímként csak a saját privát IP-címét tudja használni
    - Privát IP-cím használata az Interneten NEM megengedett!
  - A kimenő router cserélje ki a forrás privát 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

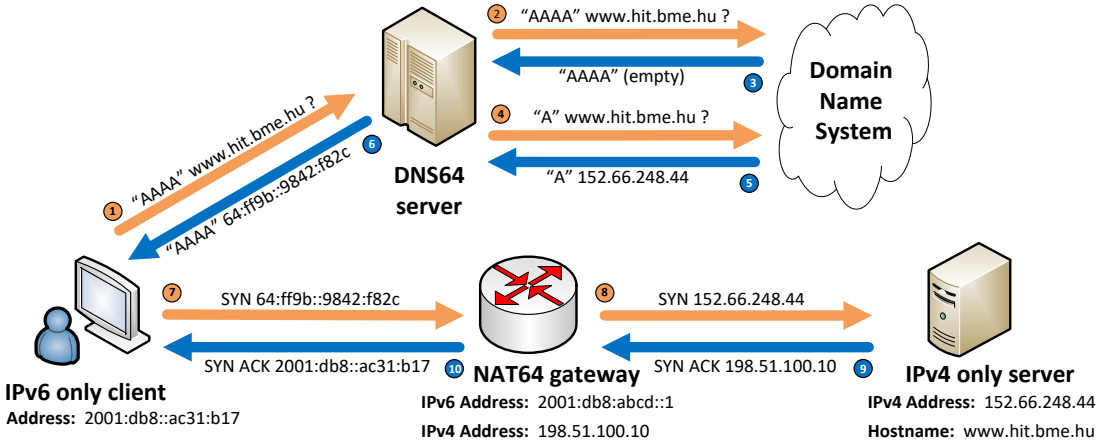
- Az ismertetett megoldást hívjuk *Source NAT*-nak (SNAT) akkor, ha a router publikus IP-címe fix, és Masquerade-nek, ha DHCP-vel kapta az interfésze a címet
- Megjegyzések:
  - A terminológia nem egységes
    - Eredetileg a NAT csak az IP-címek cseréjét jelentette, ezt hívják ma *basic NAT*-nak, vagy *one-to-one NAT*-nak.
    - A fenti megoldás precíz neve a *NAPT* (Network Address and Port Translation). Nevezik *many-to-one NAT*-nak is.
  - Milyen esetben kell a forrás portszámot kicserélni?
    - Traditional NAPT: mindig
    - Extended NAPT: csak akkor, ha a nyilvántartásban ütközés lenne valamely másik kapcsolattal.

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 ICMP 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.

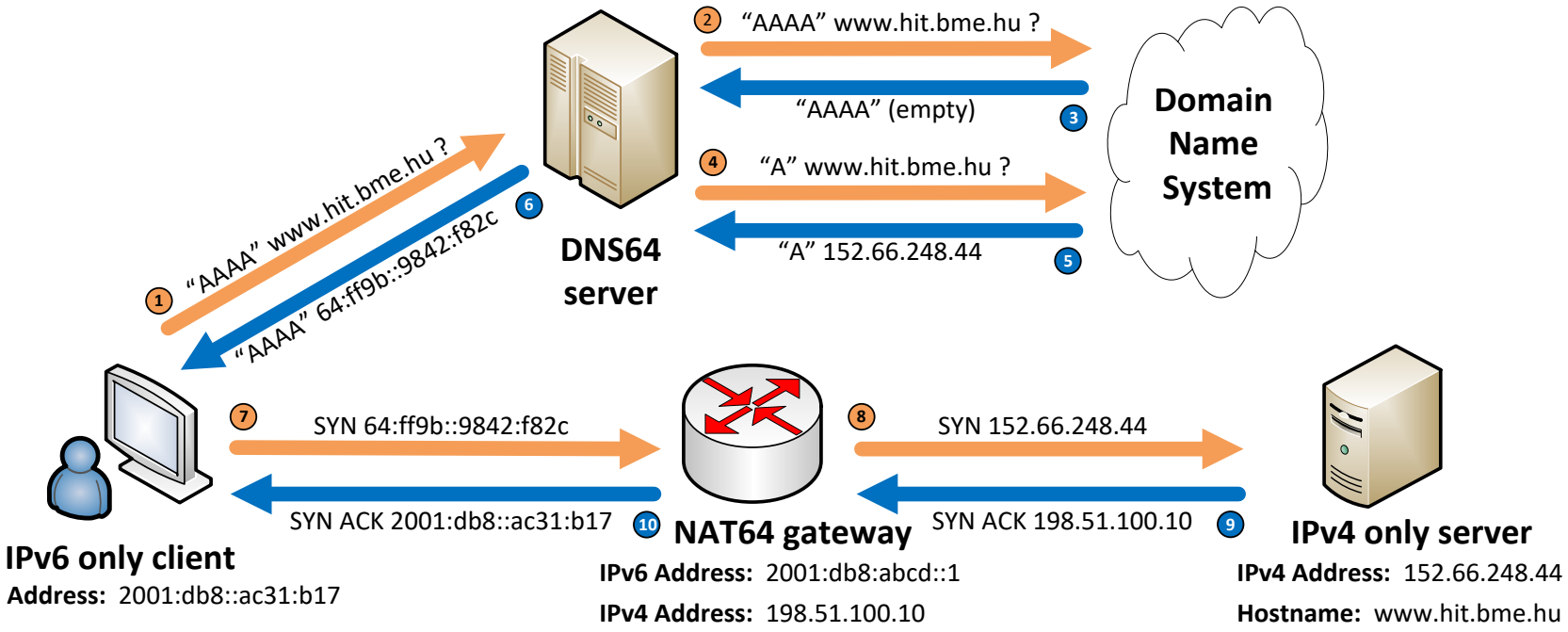


Forrás: http://www.hit.bme.hu/~lencse/publications/IJATES2-2016-MTD64-published.pdf

DNS64 + NAT64

IPv6-only kliens és IPv4-only szerver

- Az IPv4 címek kifogyása miatt ez egy tipikus megoldandó probléma
  - Az elvi megoldást az alábbi példán mutatjuk be
    forrás: DOI: 10.1109/TSP.2015.7296218



DNS64 szerver

- Mint egy caching-only name server, *Recursive query*kre válaszol
  - Lekéri a normál DNS rendszertől a kérdéses szimbolikus névhez tartozó IPv6 címet (AAAA record)
  - Ha van IPv6 cím, akkor továbbítja a válaszában
  - Ha nincs IPv6 cím, akkor IPv4 címet kér, és ha van, akkor abból 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 speciális (IPv4 címet tartalmazó) IPv6 címet úgy hívják, hogy *IPv4 címet beágyazó IPv6 cím* (IPv4-embedded IPv6 address)

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.)

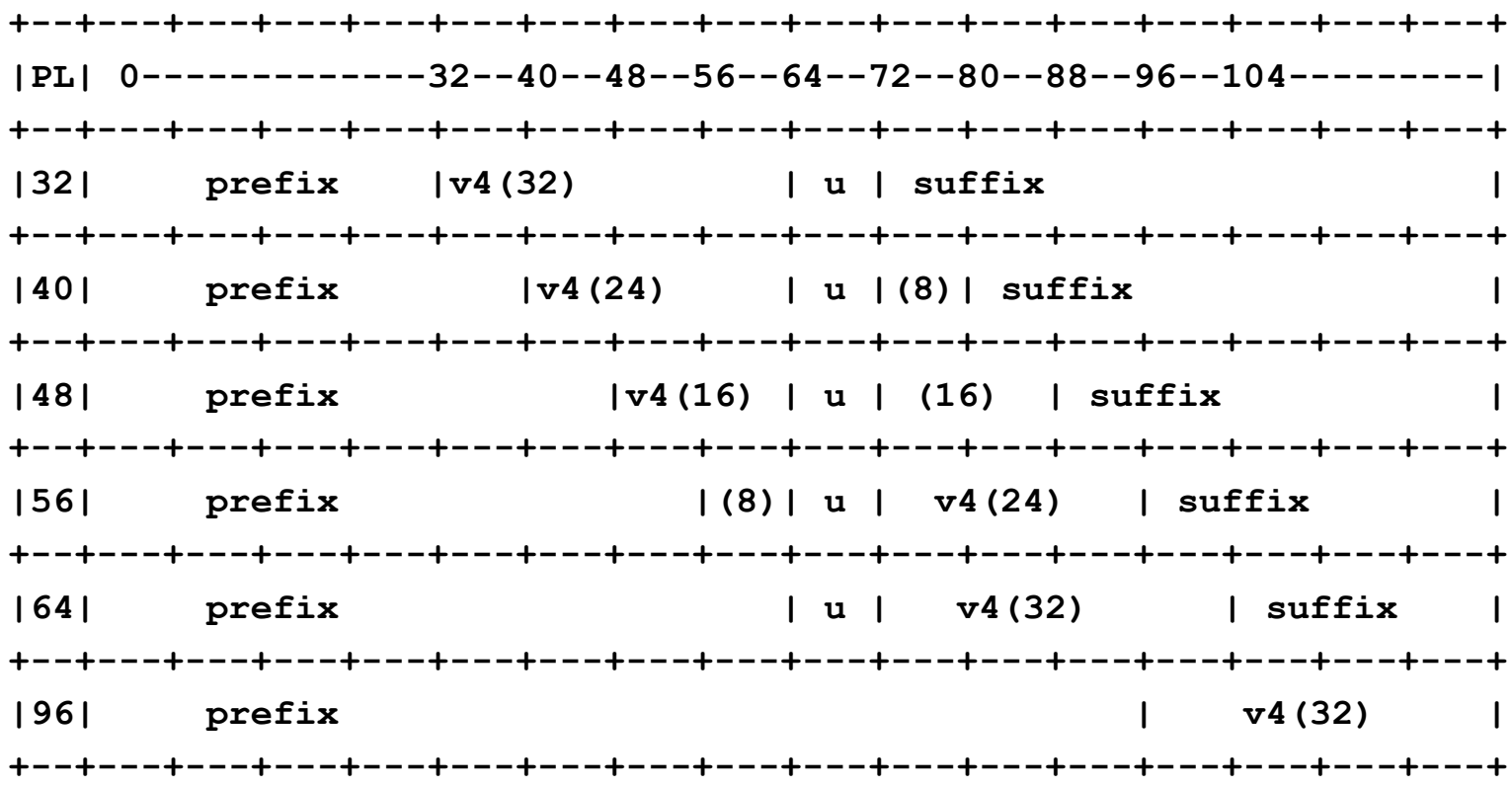
IPv4-Embedded IPv6 címek képzése – 1

- 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



Implementációk

- DNS64
  - BIND
  - TOTD https://github.com/fwdillema/totd (nem fejlesztik)
  - Unbound
  - PowerDNS
  - MTD64 https://github.com/Yoso89/MTD64 (csak kísérleti)
  - mtd64-ng https://github.com/bakaid/mtd64-ng (még kísérleti)
- NAT64 (állapottartó)
  - Ecdysis (nem nagyon fejlesztik, Linux kernel modul valaha fagyott)
  - TAYGA + iptables (TAYGA felhasználói címtérben fut)
  - OpenBSD PF
  - Jool

NAT64 kompatibilitása alkalmazásokkal

Alkalmazás protokoll	PF	Tayga	Ecdsys	
			[34]	[36]
HTTP	I	I	I	I
HTTPS	I	I	I	I
SMTP	I	I	I	I
POP3	I	I	I	I
IMAP4	I	I	I	I
Telnet	I	I	I	√
SSH	I	I	I	I
FTP passzív mód	I	I	?	?
FTP aktív mód	N	N	?	?
OpenVPN	I	I	N	N
RDP	I	I	I	√
Syslog	I	I	√	√
Skype	N	N	N	N
BitTorrent	N	N	feltételes	N
SIP	N	N	N	N

Forrás: IPv6 könyv, Hajas Tamás győri hallgatóm mérései alapján

IPv4, mint szolgáltatás (IPv4aaS) megoldások

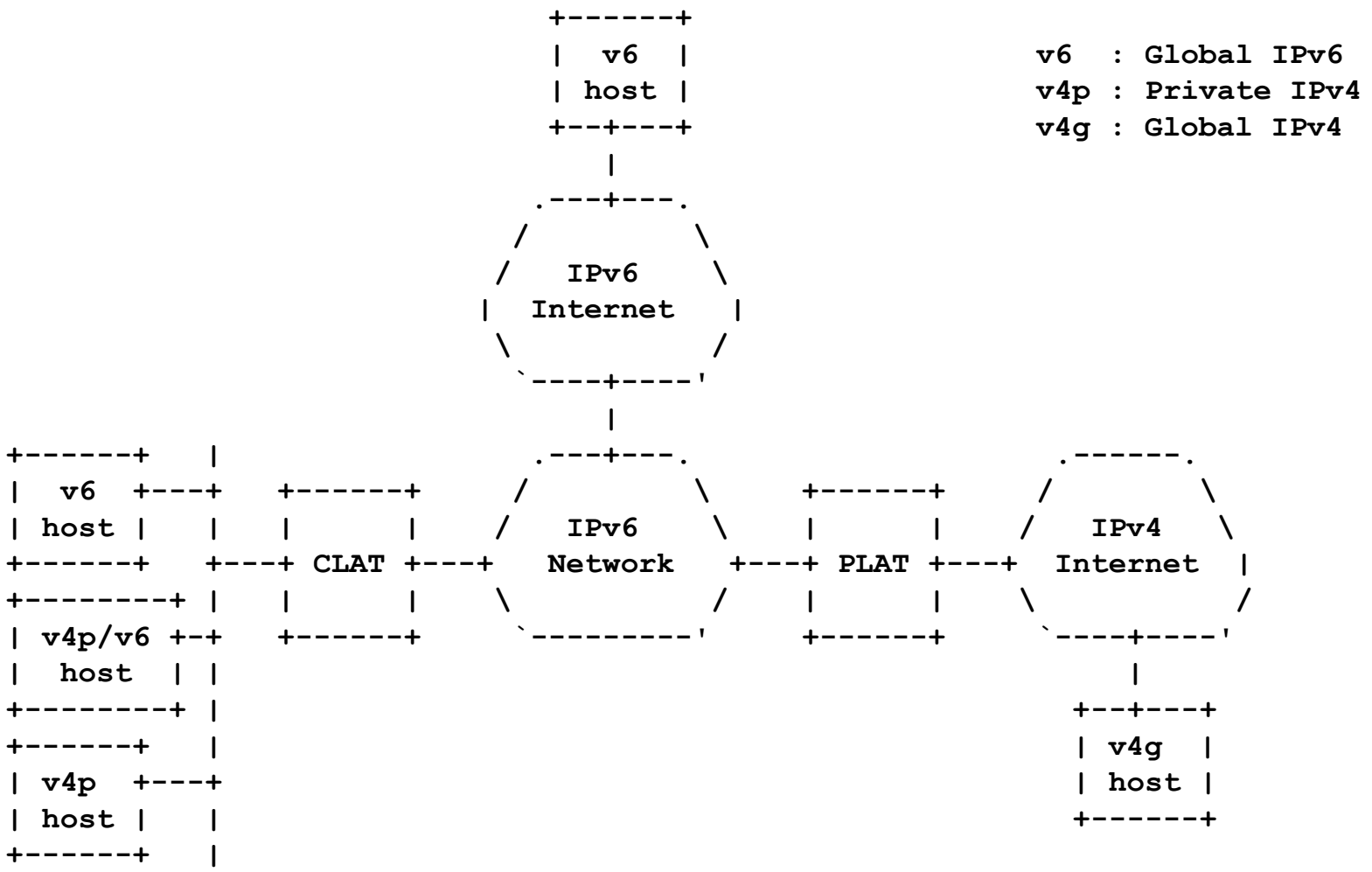
A feladat

- A kliens eszközök natív IPv6 elérést kapnak
- A szolgáltató hálózatában csak IPv6 van
  - A gerinchálózatban is (core network)
  - A hozzáférési hálózatban is (access network)
- Klienseknek vannak csak IPv4-re képes alkalmazásai, azok privát IPv4 címet kapnak
- Valamilyen eszköz, valamilyen technológiával megoldja, hogy az IPv4 forgalom a szolgáltató IPv6 hálózatában áthaladjon, erre lehetőségek:
  - Beágyazással
  - Kétszeres fordítással

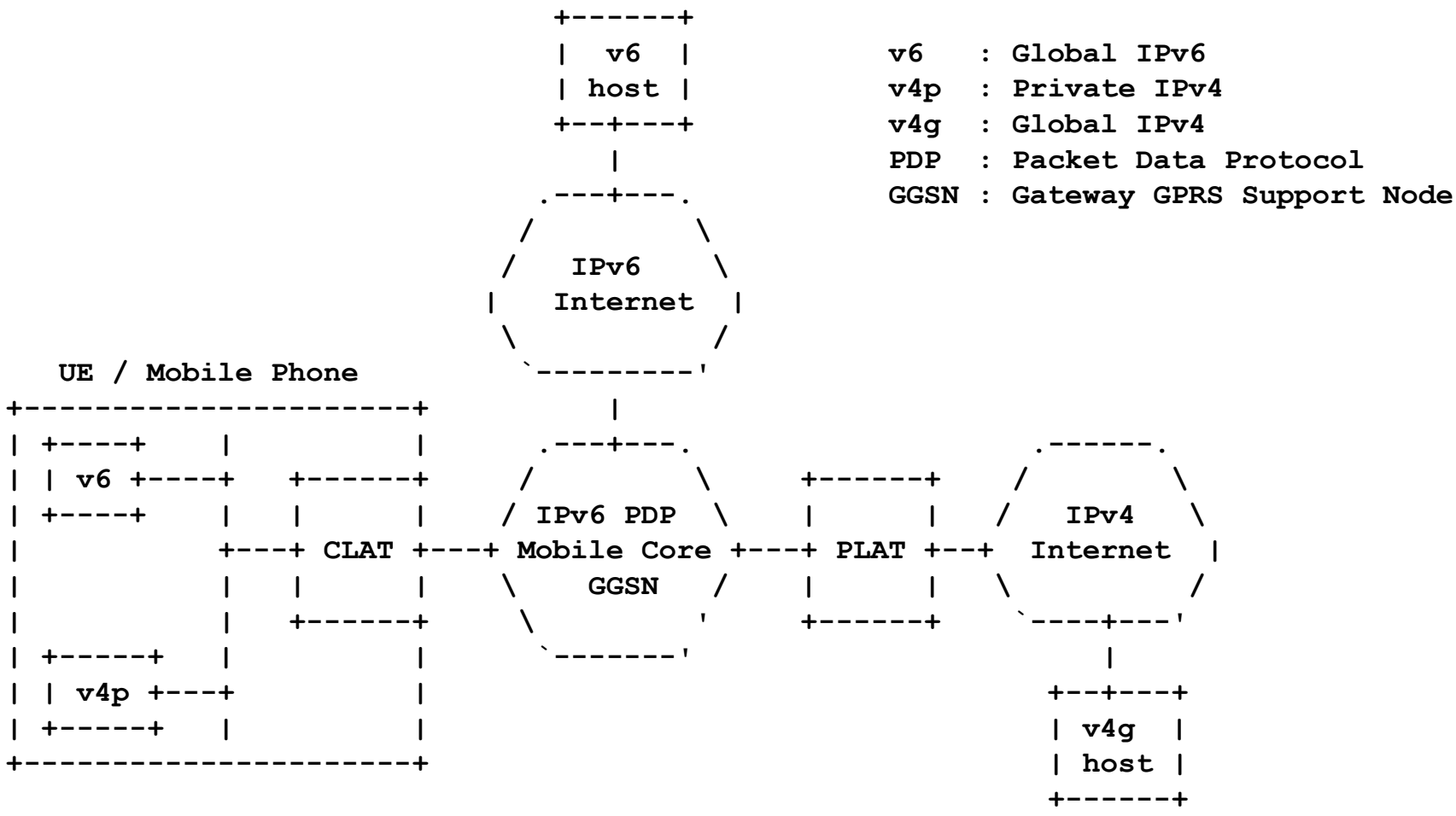
A 464XLAT megoldás

- Az IPv4 alkalmazások forgalmára kétszeres fordítást használ
  - CLAT: a kliens oldalon működik, állapotmentes IPv4/IPv6 fordítást végez a csak IPv4-re képes alkalmazás forgalma számára
  - PLAT: a szolgáltatónál működik, állapottartó IPv6/IPv4 fordítást végez a csak IPv4-re képes alkalmazás forgalma számára
    - Nem más, mint egy állapottartó NAT64 átjáró! ☺
  - A csak IPv4-re képes alkalmazásnak nem kell DNS64, de:
  - IPv6 kliensek és IPv4 szerverek esetén célszerűen DNS64
    - Ezt a forgalmat nem kell kétszer fordítani. ☺
    - A NAT64 átjáró pedig úgyis ott van. ☺
  - Mind vezetékes, mind mobil hálózatban jól használható, némileg eltérő topológiával (CLAT helye: ISP CPE eszköze / mobil eszköz)

464XLAT vezetékes topológia



464XLAT 3GPP mobil topológia

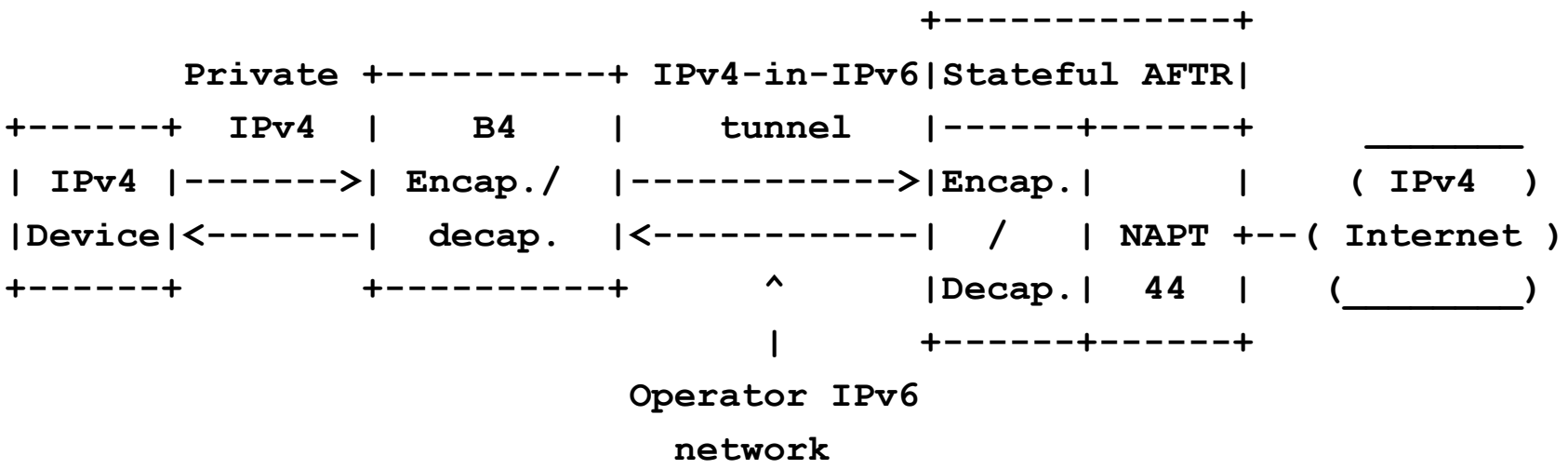


A 464XLAT megoldás működése

- A CLAT az IPv6 forgalom számára router
- Az IPv6-ra képes alkalmazások elérik:
  - az IPv6 fölött (is) elérhető szervereket natív IPv6-tal
  - a csak IPv4-en elérhető szervereket DNS64+NAT64 segítségével
- A csak IPv4-re képes alkalmazások számára a CLAT nyújt DNS szolgáltatást
  - DNS proxyként viselkedik, így a csak IPv4-re képes alkalmazások DNS forgalmát nem fordítják duplán
- A csak IPv4-re képes alkalmazások forgalmát az IPv4 szerverek felé
  - CLAT IPv6-ra fordítja (állapotmentes: 1 IPv4-címhez 1 IPv6 cím)
  - PLAT IPv4-re fordítja (állapottartó: forráscím: saját publikus IPv4)
  - (Visszafele irányban pedig minden fordított sorrendben történik.)

A Dual Stack Lite megoldás

- Ez volt az első IPv4aaS megoldás
- Az IPv4 alkalmazások forgalmára beágyazást használ
  - B4 (Basic Broadband Bridging): a kliens oldalon működik, a csak IPv4-re képes alkalmazás csomagjait IPv6-ba ágyazva küldi el az AFTR-nek
  - AFTR (Address Family Transition Router): a szolgáltatónál működik, az IPv4 kliens forgalmát IPv6-ból kibontja, és NAPT a publikus IPv4 Internet felé
  - Természetesen mindegyik működik a másik irányban is ☺

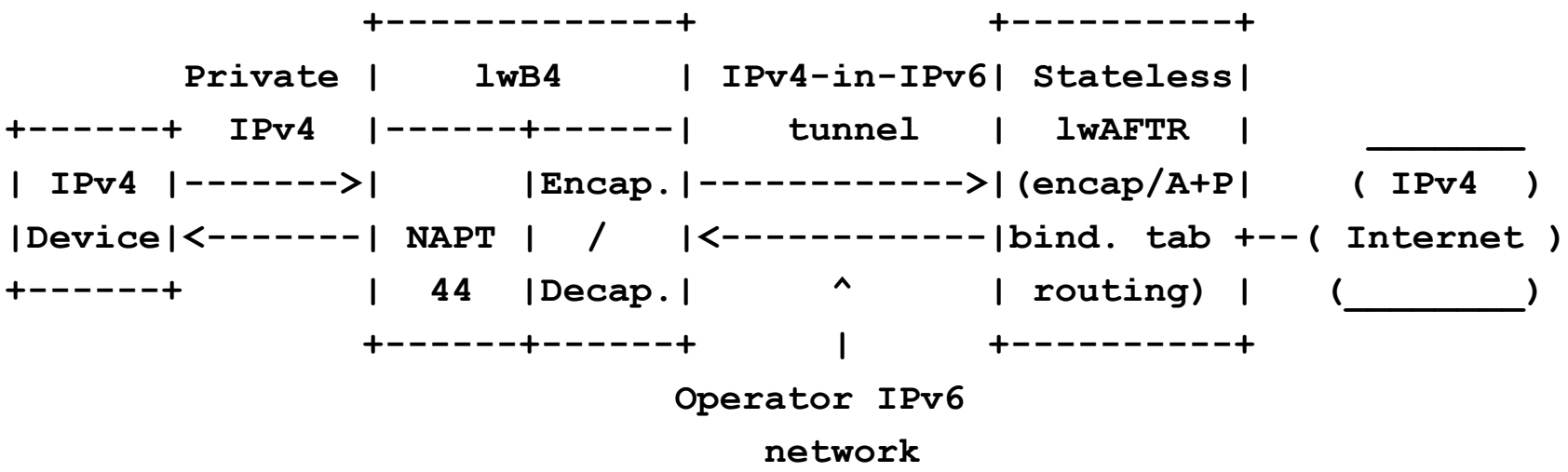


Kitérő: Address plus Port megközelítés

- A publikus IPv4 címek kifogyásának kezelésére többféle megoldás is létezik
- Tisztességes megoldás: áttérés IPv6-ra
- Gányolások:
  - Állapottal rendelkező (stateful): Privát IP-címek + NAPT
  - Állapotmentes (stateless): Address plus Port
    - Egy-egy előfizető nem kap meg egy teljes publikus IPv4-címet, csak annak egy részét (adott porttartományt)
    - A routing nemcsak az IP-címek, hanem IP-címek és portok alapján működik
    - Állapotmentesség → elméletileg jobban skálázódik
      - Megjegyzés: hasheléssel a NAPT is „elég jó” ☺
      - Én személy szerint nem lelkesedem az újabb gányolásokért. ☹
    - A további 3 IPv4aaS megoldás ezt használja ☹

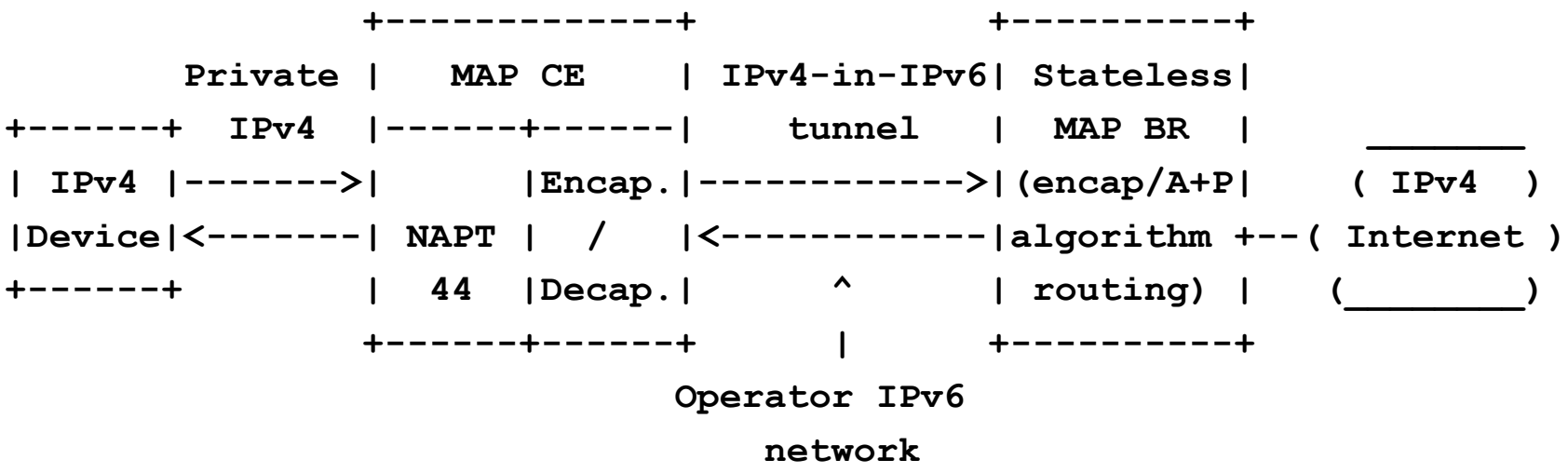
A Lightweight 4over6 megoldás

- A DS-Lite módosított változata
- Állapottartó NAPT áthelyezése a központból kliensekbe
  - lwB4: NAPT + beágyazás + IPv4-cím megosztása az A+P-vel
  - lwAFTR: kibontás + IPv4-cím megosztása az A+P megoldással
  - Természetesen mindegyik működik a másik irányban is ☺



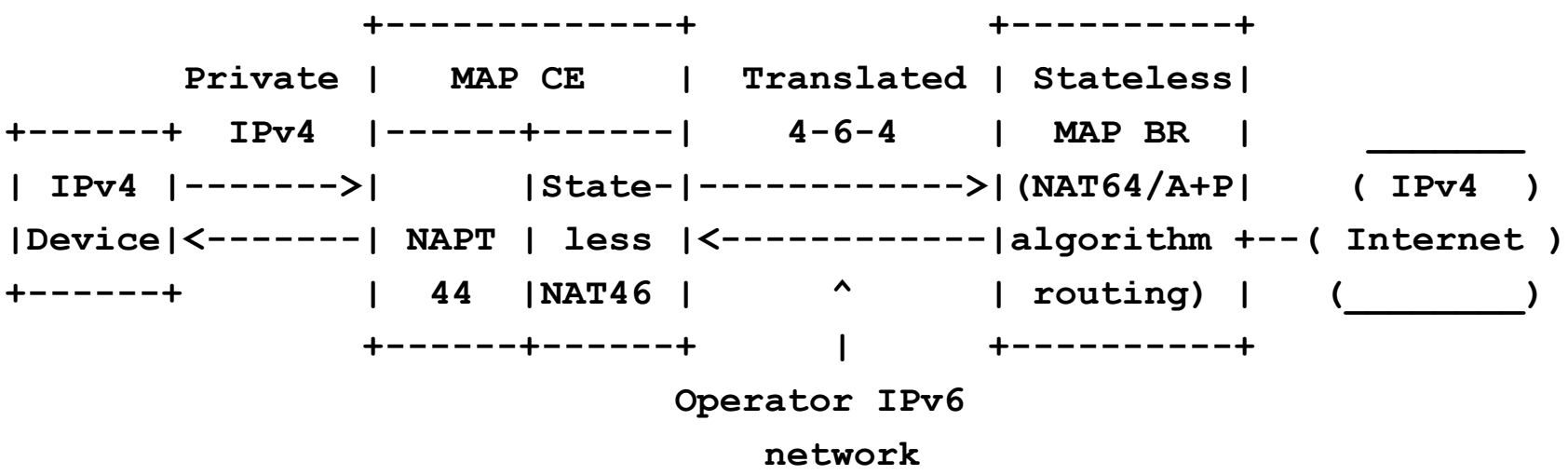
A MAP-E megoldás

- Nagyon közeli rokonságban van az lw4o6-tal, de vannak eltérések
  - MAP CE: NAPT + beágyazás + A+P leképzési szabályok
  - MAP BR: kibontás + A+P leképzési szabályok
  - Természetesen mindegyik működik a másik irányban is ☺
  - Részleteit idén nem tanuljuk ;-)



A MAP-T megoldás

- A MAP-E-hez hasonlít, de beágyazás helyett kétszeres fordítást használ
  - MAP CE: NAPT + állapotmentes NAT46
  - MAP BR: állapotmentes NAT64 + A+P leképzési szabályok
  - Természetesen mindegyik működik a másik irányban is ☺
  - Részleteit idén nem tanuljuk ;-)



Az 5 megoldás összevetése (A)

- 2x2 lehetőség
  - Szolgáltató hálózatán való áthaladáshoz
    - beágyazás
    - kétszeres fordítás
  - A szolgáltató hálózatában állapot
    - van (stateful)
    - nincs (stateless)

State in the operator network	Service provider network traversal technology: single/double translation	encapsulation/ de-encapsulation
Present	464XLAT	DS-Lite
Absent	MAP-T	MAP-E, lw4o6

https://tools.ietf.org/html/draft-lmhp-v6ops-transition-comparison-01

Az 5 megoldás összevetése (B)

- 2x2 lehetőség
  - Szolgáltató hálózatán való áthaladáshoz
    - beágyazás
    - kétszeres fordítás
  - A szolgáltató hálózatában állapot
    - van (stateful)
    - nincs (stateless)

	464XLAT	DS-Lite	lw4o6	MAP-E	MAP-T
4-6-4 trans.	X				X
4-in-4 encap.		X	X	X	
Per-flow state in op. network	X	X			

https://tools.ietf.org/html/draft-lmhp-v6ops-transition-comparison-02

TOVÁBBI MEGOLDÁSOK

SIIT (stateless „NAT64”)

- SIIT: Stateless IP/ICMP Translation
  - Más technológiák részeként már láttuk
  - Önállóan is használható, de mire?
    - Hozzáférést nyújt a másik fajta protokollal működő szolgáltatáshoz
- Fontos alkalmazás
  - Valamilyen szolgáltatás megvalósítása belül: csak IPvX
  - Hozzáférés mégis: dual stack

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

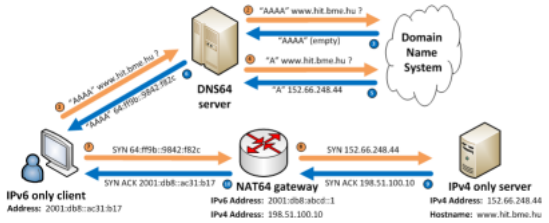
A 6to4 megoldás

- 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
- Problémák
  - Minőségi, sőt működési
    - Anycast, idegen relayek használata, esetleg nem továbbítja…
  - Biztonsági: számos lehetőséget ad a visszaélésre
    - Érdeklődőknek:

      http://tdk.bme.hu/VIK/Halozat1/IPv6-atteresi-technologiak-vizsgalata
- RFC 7526: nagyobb részben eltemették

Összefoglalás

- Az IPv4 → IPv6 átmenet jellemzői és eszközei
- Kitérő: hálózati címfordítás
- DNS64 + NAT64
- IPv4aaS megoldások
  - 464XLAT, DS-Lite, lw4o6, MAP-E, MAP-T
- További megoldások
  - SIIT, 6in4, 6to4 (teredo, 6rd)
- Irodalom:
  - Lencse Gábor, Répás Sándor, Arató András: „IPv6 és bevezetését támogató technológiák”, HunNet-Média Kft. 2015. Budapest, ISBN: 978-963-12-3272-1, DOI: 10.18660/ipv6-b1.2015.9.1 [Online] elérhető: http://ipv6ready.hu/konyv/
  - G. Lencse et. al., “Pros and Cons of IPv6 Transition Technologies for IPv4aaS”, Internet Draft, January 6, 2020, [Online], available: https://tools.ietf.org/html/draft-lmhp-v6ops-transition-comparison-04



Kérdések?

KÖSZÖNÖM A FIGYELMET!

Hálózati Rendszerek és Szolgáltatások Tanszék

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