Dr. Lencse Gábor

Mérési módszertan IPv6 áttérési technológiák teljesítőképességének vizsgálatára

SZE Távközlési Tanszék

2016.

1 Bevezetés

A publikus IPv4 címek kifogyása világszerte lendületet adott az IPv6 bevezetésének. A bevezetés soron következő fázisában várhatóan jellemző lesz az a szituáció, amikor a kliensek már nem jutnak publikus IPv4 címhez. Erre a problémára az előre tekintő megoldás az, hogy a szolgáltatók áttérnek IPv6-ra. Viszont a szerverek jelentős része még csak IPv4 címmel rendelkezik. Arra a problémára, hogy a csak IPv6 címmel rendelkező kliensek el tudják érni a csak IPv4 címmel rendelkező szervereket, a legjobb megoldás a DNS64 (RFC 6147) és az állapottartó NAT64 (RFC 6146) IPv6 áttérési technológiák együttes alkalmazása. Ebben az esetben a csak IPv6 címmel rendelkező kliensek natívan érik az el IPv6 címmel (is) rendelkező szervereket, a DNS64+NAT64 használatára kizárólag a csak IPv4 címmel rendelkező szerverek eléréséhez van szükségük. (A DNS64+NAT64 technológia működéséről, egyes szabad szoftver implementációinak teljesítményéről bővebb leírás található az IPv6 könyvben [1].) Egy további probléma, hogy vannak olyan elterjedten használt szoftverek (pl. Skype), amelyek nem képesek az IPv6 használatára. Erre a problémára nyújt megoldást a 464XLAT technológia (RFC 6877), ami kétszeres protokoll konverziót használ, a kliensnél egy ún. CLAT van, ami állapotmentes NAT64 (RFC 6145, illetve az azt felváltó RFC 7915), a szolgáltatónál pedig egy ún. PLAT, nem más, mint a korábban említett állapottartó NAT64 (RFC 6146). Megemlítjük egyes mobil Internet szolgáltatók gyakorlatát, mely szerint a hálózatukban az IPv4 szolgáltatást megszüntetik, és csak IPv6-tal foglalkoznak, a klienseknek pedig a 464XLAT segítségével nyújtanak (csak privát IPv4 címet adó) IPv4 szolgáltatást.

A fentiek ismeretében az állapotmentes és állapottartó NAT64, valamint a DNS64 technológiák egyes implementációinak vizsgálatát tartjuk a legfontosabbnak, ugyanakkor a továbbiakban a javasolt vizsgálati eljárásokat általánosan fogalmazzuk meg, a felhasznált releváns Internet Draft (leendő RFC) [2] kategóriáinak megfelelő csoportosításban.

A továbbiakban csak a teljesítőképesség-vizsgálat (benchmarking) kérdéseivel foglalkozunk. A bemutatásra kerülő mérések célja olyan objektív teljesítményjellemzők definiálása és mérési módszereik meghatározása, amelyek alapján egyértelműen eldönthető, hogy egy adott IPv6 áttérési technológiát implementáló eszköz milyen teljesítménnyel képes az adott szolgáltatást nyújtani.

Megjegyezzük, hogy a vizsgált teljesítményjellemzőkön kívül számos egyéb jellegű probléma is előfordulhat, például:

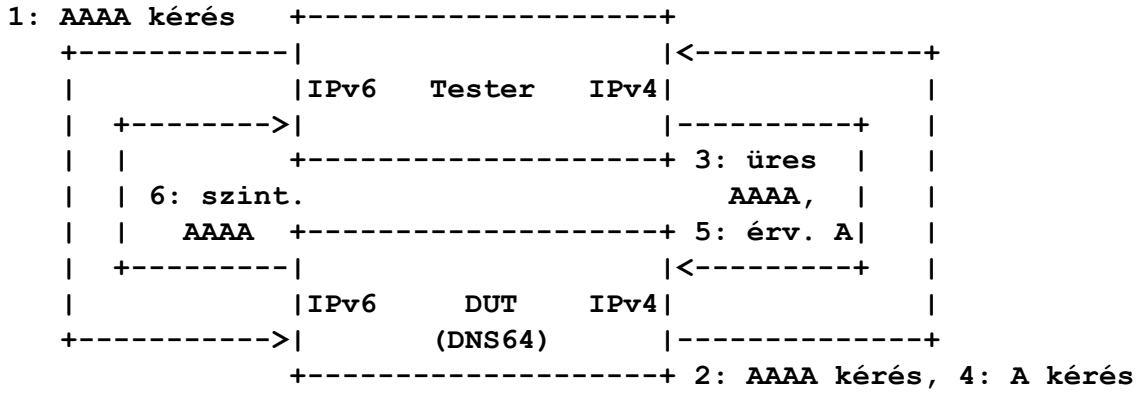
- A használt IPv6 áttérési technológiáknak (vagy adott implementációinak) a különféle alkalmazásokkal való kompatibilitásának problémája. Erről szintén további információ található az IPv6 könyvben [1].
- Bizonyos esetekben (web böngészés, FTP használata) a kliensek portszám igénye olyan magas lehet, ami beleütközhet egyes IPv6 áttérési technológiák (például A+P, RFC 6346) portszám korlátjába, ami szintén problémát okozhat. A portszám igények vizsgálatával kapcsolatos egyes eredményeink is megtalálhatók az IPv6 könyvben [1].
- Egyes IPv6 áttérési technológiák alkalmazásánál súlyos biztonsági problémák lépnek fel, melyekkel és az ellenük való védekezéssel számos RFC foglalkozik (például: RFC 3964, RFC 6343, RFC 7123).

2 Módszer DNS64 implementációk teljesítőképességének vizsgálatára

A DNS64 technológia jelentősen különbözik a többi (egyszeres vagy kétszeres fordítást, illetve beágyazást használó) technológiától, így vele a többitől elkülönítve foglalkozunk. A DNS64 szerverek teljesítőképességének mérésére kidolgozott módszert az alábbiakban mutatjuk be.

2.1 Mérési összeállítás és forgalom

A mérési összeállítás az RFC 2544 szerinti összeállítást követi, a méréshez használt forgalommal együtt az alábbi ábrán mutatjuk be.



Megjegyzés: A mérési összeállításban a Tester mind a DNS64 kliens, mind az Authoritative DNS szerver szerepét ellátja. A vizsgált eszköz (DUT: Device Under Test) pedig a DNS64 szerver szerepét látja el. (A Tester fizikailag megvalósítható egy vagy két eszközzel is.)

A mérés forgalma a következő lépéseket követi:

1. AAAA rekord (IPv6 cím) kérése egy szimbolikus névhez (domain name). (A klienstől a DNS64 szerver fele irányuló kérésnek felel meg.)
2. AAAA rekord kérése ugyanahhoz a szimbolikus névhez. (DNS64 szervertől az authoritative DNS szerver fele irányuló kérésnek felel meg.)
3. Üres AAAA rekord válasz. (Az authoritative DNS szervertől a DNS64 szerver fele küldött válasznak felel meg.)
4. A rekord (IPv4 cím) kérése ugyanahhoz a szimbolikus névhez. (DNS64 szervertől az authoritative DNS szerver fele irányuló kérésnek felel meg.)
5. Érvényes A rekord válasz. (Az authoritative DNS szervertől a DNS64 szerver fele küldött válasznak felel meg.)
6. Szintetizált AAAA rekord válasz. (IPv4 címet beágyazó IPv6 cím). (A DNS64 szervertől a kliens fele irányuló válasznak felel meg.)

Megjegyzések:

- Ha a DNS64 szerver cache-el és cache találat van, akkor a fenti üzenetek közül az 1. után a 6. következik, és a 2-5. üzenetek kimaradnak.
- Ha az adott szimbolikus névhez tartozik AAAA rekord, akkor a 3. lépésben az authoritative DNS szerver azt adja vissza, a 4. és az 5. lépés kimarad, és a DNS64 szerver nem szintetizál IPv6 címet, hanem a kapott AAAA rekordot adja vissza a kliensnek.

- Ami a Tester és a DUT között használt IP protokoll verzióját illeti, IPv6-ot kell használni a kliens és a DNS64 szerver között (hiszen a DNS64 szerver csak IPv6 címmel rendelkező kliensek számára nyújt szolgáltatást), míg a DNS64 szerver és az authoritative DNS szerver között akár IPv4 akár IPv6 használható.

2.2 Névfeloldás teljesítményének mérése

Cél: A DNS64 teljesítmény meghatározása a sikeresen kiszolgált DNS kérések másodpercenkénti számával.

Mérési eljárás: A Tester adott ideig, másodpercenkénti meghatározott számú DNS kérést küld a vizsgált eszköznek (DUT) és közben számolja az időben megérkező (a küldéstől adott timeout időn belül érkezik, melynek alapértelmezett értéke 1 másodperc) és érvényes (azaz AAAA rekordot tartalmazó) válaszokat. Ha a kérések és az időben érkező, érvényes válaszok száma megegyezik, akkor a kérések másodpercenkénti számát növeljük, és a tesztet újra elvégezzük. Ha az időben érkező, érvényes válaszok száma alacsonyabb a kérések számánál, akkor a kérések másodpercenkénti számát csökkentjük, és a mérést újra elvégezzük. Minden egyes mérésnek legalább 60 másodpercig kell tartania, azért, hogy csökkentsük azon DNS64 szerverek potenciális nyereségét, amelyek képesek a kéréseket tárolni, és a timeout idejét is felhasználni azok megválaszolására. Ugyanezen okból a timeout ideje nem lehet 1 másodpercnél hosszabb.

A sikeresen kiszolgált kérések másodpercenkénti maximális száma az a legnagyobb érték, amely esetén a Tester által elküldött kérések és a Testerhez időben érkező érvényes válaszok száma megegyezik.

Megjegyzés: a mérési eljárást az RFC 2544-nek megfelelően adtuk meg, a gyakorlatban bináris keresést célszerű használni.

A fenti mérést legalább 20-szor elvégezzük, az eredmények mediánját valamint minimumát és maximumát kiszámítjuk.

2.3 Részletek és paraméterek

2.3.1 Cache-elés

Először minden kérésnek különböző szimbolikus nevet (domain name) kell tartalmaznia. Utána végrehajthatók olyan merések is, amelyeknél a kérések 20%-a, 40%-a, 60%-a, 80%-a, illetve 100%-a olyan szimbolikus nevet tartalmaz, amelyek cache-elve vannak. Megjegyezzük, hogy a cache találat biztosításához két dolog szükséges: ezeket „elég későn” kell az első kérés után elküldeni ahhoz, hogy a feloldás megtörténte után már benne legyenek a cache-ben, és ugyanakkor „elég korán” is, hogy még benne legyenek a cache-ben.

2.3.2 AAAA rekord létezése

Először minden kérésnek olyan szimbolikus nevet (domain name) kell tartalmaznia, amelyhez nem létezik AAAA rekord, és létezik pontosan egy A rekord.

Utána végrehajthatók olyan merések is, amelyeknél a kérések 20%-a, 40%-a, 60%-a, 80%-a, illetve 100%-a olyan szimbolikus nevet tartalmaz, amelyekhez létezik AAAA rekord.

Megjegyezzük, hogy a fenti két feltétel ortogonális: bármely kombinációjuk vizsgálható. A 0% cache-elt és 0% AAAA rekordot tartalmazó kombináció vizsgálata kötelező, a többi opcionális. (Amikor minden név cache-elt, akkor az eredmény nem függ attól, hogy a nevek hány százalékához tartozik AAAA rekord, így ezeket a kombinációkat nincs értelme egyenként megvizsgálni.)

2.4 Eredmények megjelenítése

A DNS64 mérésnek az elsődleges eredménye a fent említett „0% +0%” kombinációval mért másodpercenként megválaszolt DNS kérések száma. A legalább 20 mérésből kiszámítjuk ezek mediánját, valamint a mérés stabilitásának tükrözése érdekében minimumát és maximumát. Ha opcionális méréseket is végeztünk, akkor azok eredményét (szintén: medián, minimum, maximum) megjeleníthetjük egy olyan két dimenziós táblázatban, ahol a két dimenzió a cache-elt nevek aránya, illetve az AAAA rekordot tartalmazó nevek aránya, melyeket a táblázat fejrészei százalékban megadva tartalmaznak. Alternatív megoldásként az eredmények megjeleníthetők egy kétdimenziós grafikon formájában is. Ebben az esetben a medián értékeknél hibasáv formájában feltüntetendők a minimum és maximum értékek. Akár a táblázatból, akár a grafikonból készíthetők egydimenziós kivonatok a másik dimenzió bármely fix százalékos értékénél, de ebben az esetben azt a fix értéket meg kell adni az egydimenziós táblázattal vagy grafikonnal együtt.

2.5 Követelmények a Testerrel szemben

Mielőtt egy Testert valamely DNS64 szerver (DUT) teljesítményének vizsgálatára használnánk *r* kérés/másodperc sebesség és *t* timeout mellett, annak át kell esnie egy öntesztен, hogy kizárjuk annak a lehetőségét, hogy a Tester elégtelen teljesítménye befolyásolja az eredményeket. Az önteszt végrehajtásához a Testert (a DUT kihagyásával) visszahurkoljuk, és az authoritative DNS szerver alrendszerét úgy konfiguráljuk, hogy minden kérdéses szimbolikus névre AAAA rekord választ tudjon adni. A sikeres öntészthez $2*(r+\delta)$ sebességgel és $0.25*t$ timeouton belül kell AAAA rekordokat szolgáltatnia, hol $\delta \geq 0{,}1$.

Magyarázat: Amikor DNS64 teljesítményt mérünk, minden egyes AAAA rekord kérés következtében legfeljebb két kérést kell a DNS64 szervernek elküldenie; az elsőben egy AAAA rekordot, a másodikban egy A rekordot kér (akkor küldi el ténylegesen mindkettőt, ha nincs cache találat, és nem létezik AAAA rekord). A fenti paraméterek garantálják, hogy az authoritative DNS szerver alrendszer képes a megkívánt frekvenciával úgy megválaszolni a kéréseket, hogy ehhez összesen a timeout időnek legfeljebb a felét használja fel.

Megjegyzések:

- Létezik egy szabad szoftver mérőprogram, a dns64perf++, ami forráskódban elérhető [3], a program dokumentációja pedig megtalálható [4].
- A mérőprogram a Testernek csak a kliens részét valósítja meg, és valamely authoritative DNS szerver implementációval (pl. BIND, NSD, YADIFA) együtt használható.

3 Módszer más IPv6 áttérési technológiák (pl. NAT64, 464XLAT) implementációi teljesítőképességének vizsgálatára

3.1 Érintett IPv6 áttérési technológiák és javasolt vizsgálati eljárások

A következőkben bemutatjuk, hogy az egyes technológiák implementációi teljesítőképességének vizsgálatára milyen eljárásokat javaslunk. Megjegyezzük, hogy ezek elvileg sok esetben megegyeznek az RFC 2544 eljárásaival, azonban az RFC 2544 szerinti mérésre készített eszközök nem használhatók, mivel a csomagok formátumát az egyes eljárások megváltoztatják (például NAT64 esetén IPv4 és IPv6 közötti konverzió). A Széchenyi István Egyetemen a közeljövőben az alábbiak szerinti mérőprogram kísérleti verziójának kifejlesztését tervezzük.

- egyszeres fordítást használó technológiák esetén:
    - állapotmentes (SIIT/IVI, RFC 7915 állapotmentes NAT64, CLAT a 464XLAT-ban)
        - RFC2544-ben leírt throughput mérés
        - [2]-ben leírt késletetés (latency) mérés (https://tools.ietf.org/html/draft-ietf-bmwg-ipv6-tran-tech-benchmarking-03#section-7.2)
        - [2]-ben leírt PDV (Packet Delay Variation) mérés (https://tools.ietf.org/html/draft-ietf-bmwg-ipv6-tran-tech-benchmarking-03#section-7.3.1)
        - [2]-ben leírt IPDV (Inter Packet Delay Variation) mérés (https://tools.ietf.org/html/draft-ietf-bmwg-ipv6-tran-tech-benchmarking-03#section-7.3.2)
    - állapottartó (RFC 6146 állapottartó NAT64, =PLAT a 464XLAT-ban)
        - a fentiek mindegyike és az alábbiak:
        - RFC 3551 szerinti mérés az eszköz által egyidejűleg kezelni képes TCP kapcsolatok maximális számának meghatározására (https://tools.ietf.org/html/rfc3511#section-5.2)
        - RFC 3551 szerinti mérés az eszközön keresztül létesíthető TCP kapcsolatok másodperenkénti maximális számára (https://tools.ietf.org/html/rfc3511#section-5.3)
        - (A további állapottábla menedzsment lépések sebességének mérése a jövőben kiegészítésként szolgálhat.)
- kétszeres fordítást használó technológiák esetén (464XLAT, MAP-T, dIVI)
    - ugyanazok a módszerek, mint egyszeres fordítás esetén, de más összeállításban is lehet mérni (lásd lent: a kétféle fordítást végző eszközt

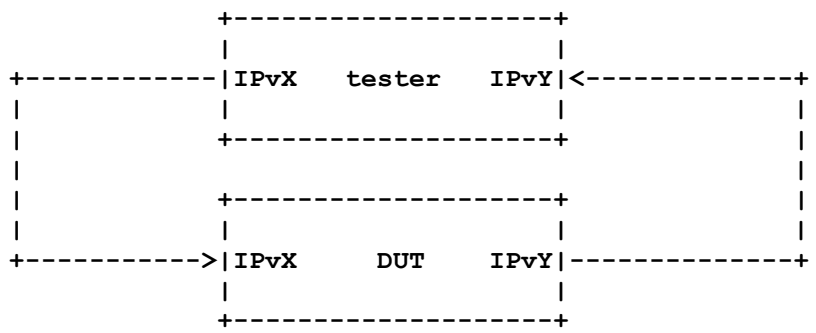
párban használva alkalmazható a két tesztelt eszközt tartalmazó összeállítás, de mindenképpen javasolt külön-külön az egy tesztelt eszközt tartalmazó összeállításban is mérni)

- beágyazást használó technológiák esetén (DSLite, MAP-E, Lightweight 4over6, 6to4, 6rd)
    - a beágyazást (encapsulation) és a kibontást (decapsulation/de-encapsulation) végző eszközöket párban használva ugyanazok a mérések, mint amiket az egyszeres fordítást használó technológiáknál felsoroltunk (lásd: két tesztelt eszközt tartalmazó mérési összeállítás)
    - csak az egyik irányú művelet elvégzésének a mérése speciálisan konfigurált forgalomgenerátorral (lásd: egy tesztelt eszközt tartalmazó mérési összeállítás)
        - ha a DUT a beágyazást végzi, akkor a mérő programnak képesnek kell lennie a csomagok kibontását elvégezni, és utána ellenőrizni a csomagokat
        - ha a DUT a kibontást végzi, akkor a mérő programnak képesnek kell lennie beágyazott forgalom küldésére (és aztán a DUT által kibontott csomagokat kell ellenőriznie).

3.2 Mérési összeállítás és forgalom

3.2.1 Egyszeres fordítást használó technológiák

A mérési összeállítás az RFC 2544 szerinti összeállítást követi. Az egyszeres fordításhoz az egy tesztelt eszközt (DUT) tartalmazó összeállítást kell használni:



Megjegyzés: bár az ábrán a nyilak egyirányúak, a teszteket mindkét irányban el lehet végezni!

A javasolt módszer az, hogy előbb az egyik irányban, majd a másik irányban teszteljünk.

Forgalom az első teszthez (az állapottartó NAT64 példáját használva):

1. A Tester küld IPv6 csomagot a NAT64 átjárónak.
2. A NAT64 átjáró veszi az IPv6 csomagot, lefordítja IPv4 csomagra és továbbítja a Testernek.

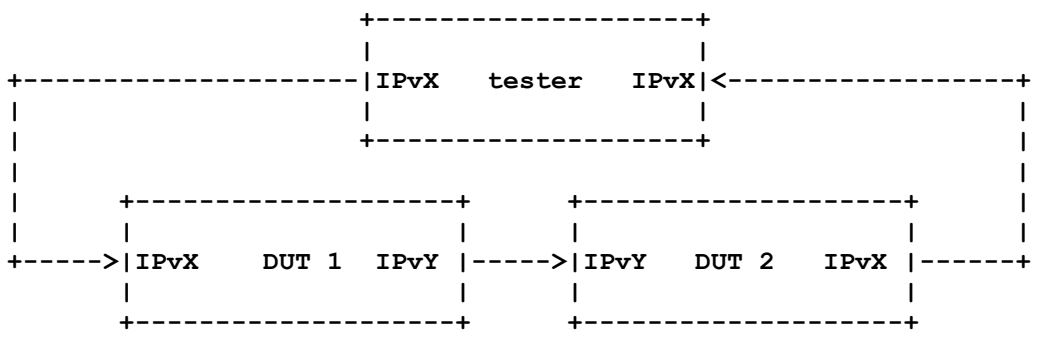
Forgalom a második teszthez (az állapottartó NAT64 példáját használva):

3. A Tester küld egy IPv4 csomagot (mint választ).
4. A NAT64 átjáró veszi az IPv4 csomagot, lefordítja IPv6 csomaggá, és elküldi a Testernek.

A legelső csomagnak a második teszt esetén is egy IPv6 csomagnak kell lennie, hogy létrehozza az állapottartó NAT64 átjáró állapottáblájában a megfelelő összerendeléseket, utána már nincs további megkötés, a Tester tetszőleges számú IPv4 csomagot küldhet.

3.2.2 Kétszeres fordítást, illetve beágyazást használó technológiák

Használhatjuk a két tesztelt eszközt (DUT) tartalmazó összeállítást:



Kétszeres fordítást használó technológiák esetén az első DUT az egyik irányú fordítást, a második DUT a másik irányú fordítást végzi. Beágyazást használó technológiák esetén az első DUT az beágyazást, a második a kibontást végzi.

A nyilak irányára vonatkozó fenti megjegyzést és a két irányban külön-külön végzett tesztelés módszerét itt is értelemszerűen alkalmazzuk, a forgalom leírását nem ismételjük meg.

A fenti összeállítás előnye (a mérési technológia szempontjából) az, hogy a két eszközt párban használva a Tester olyan mérendő rendszert lát, amely mindkét portján azonos IP verziót használ, ami lehetővé teszi a közönséges, RFC 2544 szerinti IPv4 vagy IPv6 mérőeszközök használatát.

A fenti összeállítást hátránya, hogy az eszközök párban való használata csak a két eszköz együtt történő vizsgálatát teszi lehetővé. Az esetleges aszimmetrikus viselkedés így rejtve marad. (Például 464XLAT esetén PLAT állapottartó, CLAT nem.) Ezért mindenképpen javasoljuk a kétféle DUT vizsgálatát külön-külön is elvégezni az egy tesztelt eszközt tartalmazó mérési összeállításban.

Amint fent említettük, a beágyazást használó technológiáknál a külön-külön történő vizsgálathoz speciális Testerre van szükség, ami elvégzi a vizsgált eszköz párjának a feladatát (a kibontást végző eszköz teszteléséhez triviális, hogy el kell végezni a beágyazást, de a beágyazást végző eszköz tesztelése esetén is szükség van a kibontás elvégzésére, hogy a kapott csomagot ellenőrizni/felismerni, megérkezését, időadatait helyesen adminisztrálni lehessen).

3.2.3 IP fölötti protokoll típusa

Az összes technológia esetén elvégzendő througput, latency, PDV, IPDV mérésekhez IP fölött UDP-t használunk. Az állapottartó technológiák estén elvégzendő RFC 3551 szerinti kiegészítő mérésekhez (egyidejűleg kezelni képes TCP kapcsolatok maximális számának meghatározása, az eszközön keresztül létesíthető TCP kapcsolatok másodperenkénti maximális számának meghatározása) pedig TCP-t (fölötte pedig HTTP-t).

3.2.4 Cél IP-címek

Jelenleg nem rendelkezünk tapasztalatokkal azt illetően, hogy a cél IP-címek azonos vagy különböző volta milyen mértékben befolyásolja a különféle IPv6 áttérési technológiák egyes implementációinak teljesítményét. (Ez a kérdés SOHO routerek esetén valószínűleg érdektelen, szolgáltatói eszközök esetén viszont fontos lehet.)

Bár az állapottartó implementációk vizsgálatára fent már javasoltunk két további vizsgálatot, véleményünk szerint azok nem feltétlenül tárják fel a probléma összes lehetséges aspektusát, ugyanis a kapcsolattábla elemeinek száma befolyásolhatja a benne való keresés, így áttételesen a csomagtovábbítás sebességét is. (Az most teljesen mindegy, hogy TCP vagy UDP „kapcsolatok”-ról van szó, egy hálózati folyam azonosítása mindkét esetben ugyanúgy a következő 5 szám segítségével történik: forrás IP-cím, forrásport, cél IP-cím, célport, valamint az IP fölötti protokoll típusa TCP/UDP.)

Tehát alap esetben az összes korábban említett mérést egyetlen egy adott cél IP-címmel végezzük (ami a Tester megfelelő interfészének a címe), és további mérésként javasoljuk n különböző cél IP-cím esetén elvégezni az összes korábban említett mérést, ahol n=1, 10, 100, 1000, 10 000, 100 000, 1 000 000, … (nyitott kérdés, hogy meddig érdemes növelni).

Érdemes lehet megvizsgálni és összehasonlítani például különböző állapottartó NAT64 implementációk viselkedését (például néhány szabad szoftver: TAYGA+iptables, OpenBSD PF, Jool), hogy romlik-e (és milyen mértékben) a teljesítményük a használt cél IP-címek számának függvényében.

Technológiai megjegyzés: különféle cél IP-címek esetén a DUT-ban az útválasztást (routing) úgy állíjuk be, hogy az alapértelmezett átjáró (default gateway) a Tester megfelelő interfésze legyen, a Testeren pedig magunkra irányítjuk az összes címet. Ez állapottartó NAT64 átjáró vizsgálata esetén pusztán NAT44-et, de például 6to4 relay vizsgálata esetén NAT66-ot igényel. Az utóbbi sem probléma, az OpenBSD PF régóta képes rá, és használtuk is. Létezik Netfilterhez is NAT66 implementáció, de azzal nincs még tapasztalatunk.

3.2.5 Mérőprogram fejlesztésére vonatkozó terveink

Kiindulásként a dns64perf++ program kódját tervezzük felhasználni, ami megoldást nyújt a következő feladatokra:

- másodpercenként megadott számú csomag küldése pontos időzítéssel
- a válaszcsomagok időben történt megérkezésének ellenőrzése
- a csomagnyilvántartás kiértékelése, CSV formában történt elmentése

Munkanévként a „6transperf” megnevezés használatát tervezzük, ami később változhat.

A programot több lépcsőben tervezzük kifejleszteni:

1. mérföldkő: az egyszeres fordítást használó technológiák UDP mérései
2. mérföldkő: az egyszeres fordítást használó technológiák TCP mérései
3. mérföldkő: kiegészítés a kétszeres fordítást használó technológiák méréseihez
4. mérföldkő: a beágyazást használó technológiák mérései

4 Hivatkozások

[1] Lencse Gábor, Répás Sándor, Arató András "IPv6 és bevezetését támogató technológiák", HunNet-Média Kft., Budapest, 2015, ISBN: 978-963-12-3272-1, DOI: 10.18660/ipv6-b1.2015.9.1, http://ipv6ready.hu/konyv/

[2] M. Georgescu, L. Pislaru, and G. Lencse, "Benchmarking Methodology for IPv6 Transition Technologies", *IETF Benchmarking Working Group*, Internet Draft, Oct. 27, 2016, https://tools.ietf.org/html/draft-ietf-bmwg-ipv6-tran-tech-benchmarking-03

[3] D. Bakai, "A C++11 DNS64 performance tester", source code, https://github.com/bakaid/dns64perfpp/

[4] G. Lencse, D. Bakai, "Design and implementation of a test program for benchmarking DNS64 servers", a cikket feltételesen elfogadták az *IEICE Transactions on Communications* folyóiratba, és a cikk javított verziója elérhető a következő helyen: http://www.hit.bme.hu/~lencse/publications/IEICE-2016-dns64perfpp-revised.pdf