



Infokommunikációs hálózatok hatékonyságvizsgálata

Doktori (Ph.D.) értekezés

Szilágyi Szabolcs

Témavezetők:

Dr. Sztrik János

Dr. Almási Béla

Debreceni Egyetem
Természettudományi Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2015.

Ezen értekezést a Debreceni Egyetem Természettudományi Doktori Tanács Informatikai Tudományok Doktori Iskola Informatikai Rendszerek és Hálózatok programja keretében készítettem a Debreceni Egyetem természettudományi doktori (PhD) fokozatának elnyerése céljából.

Debrecen, 2015.

.....

Szilágyi Szabolcs
jelölt

Tanúsítom, hogy Szilágyi Szabolcs doktorjelölt 2009.- 2014 között a fent megnevezett Doktori Iskola Informatikai Rendszerek és Hálózatok programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Az értekezés elfogadását javaslom.

Debrecen, 2015.

.....

Dr. Sztrik János
témavezető

Tanúsítom, hogy Szilágyi Szabolcs doktorjelölt 2009.- 2014 között a fent megnevezett Doktori Iskola Informatikai Rendszerek és Hálózatok programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Az értekezés elfogadását javaslom.

Debrecen, 2015.

.....

Dr. Almási Béla
témavezető

Infokommunikációs hálózatok hatékonyságvizsgálata

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében
informatikai tudományok tudományágban

Írta: Szilágyi Szabolcs okleveles mérnök-informatikus

Készült a Debreceni Egyetem
Informatikai Tudományok Doktori Iskolája
Informatikai Rendszerek és Hálózatok programja keretében

Témavezetők:
Dr. Sztrik János, Dr. Almási Béla

A doktori szigorlati bizottság:

elnök:	Dr. Fazekas István
tagok:	Dr. Bíró József
	Dr. Kruppa András Tibor

A doktori szigorlat időpontja: 2014. július 11.

Az értekezés bírálói:

Dr.
Dr.

A bírálóbizottság:

elnök:	Dr.
tagok:	Dr.
	Dr.
	Dr.
	Dr.

Az értekezés védésének időpontja: 2015.

Köszönetnyilvánítás

Ezúton is meg szeretném köszönni mindazoknak, akik közvetve vagy közvetlen módon hozzájárultak a disszertációm elkészítéséhez.

Szüleimnek, akik felneveltek, támogattak és elindítottak az életben.

Feleségemnek, aki mindenben támogatott.

Témavezetőimnek, Dr. Sztrik Jánosnak és Dr. Almási Bélának, a rengeteg hasznos tanácsért, útmutatásért és bátorításért, melyek nélkül jelen dolgozat nem készülhetett volna el.

Tartalomjegyzék

Tartalomjegyzék	i
Ábrajegyzék	iii
Táblázatjegyzék	v
1. Bevezetés	1
2. A forgalomirányítók torlódáskezelési algoritmusainak hatékonyságelemzése	3
2.1. Bevezetés, alapfogalmak	3
2.1.1. A konvergált hálózatok kialakulása	3
2.1.2. Az IP csomagok megjelölése	7
2.1.3. A torlódáskezelési mechanizmusok szükségessége	9
2.1.4. A torlódáskezelési algoritmusok áttekintése	10
2.2. A forgalomirányítók torlódáskezelési algoritmusainak hatékonyságelemzése szimulációs szoftver segítségével	16
2.2.1. Irodalmi előzmények	16
2.2.2. Anyag és módszer	17
2.2.3. Eredmények	20
2.2.4. Az eredmények összefoglalása	24
2.3. A forgalomirányítók torlódáskezelési algoritmusainak VoIP-re gyakorolt hatékonyságelemzése valós mérési környezet kialakításával	24
2.3.1. Bevezetés	24
2.3.2. Mérési környezet	25
2.3.3. A forgalom generálása	26
2.3.4. A torlódáskezelési algoritmusok implementálása	27
2.3.5. Mérési eredmények	28
2.3.6. A mért eredmények összefoglalása	31
3. A Több utas Kommunikációs Könyvtár (MPT) hatékonyságelemzése	33
3.1. Bevezetés, alapfogalmak	33
3.1.1. Az MPT működési elve	34
3.2. Az MPT teljesítményelemzése két utas környezetben	37

3.2.1. Mérés környezet és beállítások	37
3.2.2. Mérés eredmények	40
3.2.3. Összefoglalás	44
3.3. Az MPT szoftverkörnyezet használatával történő több utas FTP és adatfolyam-átvitel hatékonyságelemzése	45
3.3.1. A mérési környezet	45
3.3.2. Az FTP-átvitel mérési eredményei	47
3.3.3. A multimédia adatfolyam átvitelének mérési eredményei	50
3.3.4. Összefoglalás	54
3.4. Az MPT több utas kommunikációs könyvtár átviteli teljesítményének hatékonyságelemzése IPv4 és IPv6 környezetben	54
3.4.1. Mérés környezet	54
3.4.2. Mérés eredmények – csak IPv4-es hálózat	56
3.4.3. Mérés eredmények – csak IPv6-os hálózat	61
3.4.4. Mérés eredmények – vegyes hálózat (IPv6 feletti IPv4)	65
3.4.5. Az eredmények összefoglalása	69
4. Összefoglalás	71
5. Summary	75
6. Irodalomjegyzék	77
6.1. Hivatkozások	77
6.2. Publikációim	81
A. Függelék: A torlódáskezelési algoritmusok szimuláció által nyert numerikus eredményei	83
B. Függelék: A torlódáskezelési algoritmusok fizikai eszközökön való implementálásához használt eszközök konfigurációs fájljai	98
C. Függelék: A torlódáskezelési algoritmusok fizikai eszközökön való implementálása	103
D. Függelék: Az MPT átviteli teljesítményének hatékonyságelemzése céljából használt IPv4 és IPv6 mérési környezetek	106

Ábrajegyzék

1. ábra. A globális Internet felhasználóinak száma	1
2. ábra. Hagyományos, nem konvergált hálózatok	3
3. ábra. Konvergált infokommunikációs hálózat	4
4. ábra. Túl nagy forgalom esetén a teljesítőképesség meredeken visszaesik	5
5. ábra. Sebesség eltérés példa	6
6. ábra. Aggregációs példa	6
7. ábra. Az IP fejrész ToS és DS mezőinek összehasonlítása	8
8. ábra. A sorkezelés komponensei	10
9. ábra. A FIFO ütemezési elv	10
10. ábra. A PQ ütemezési elv	11
11. ábra. A PQ ütemezés algoritmus [9]	11
12. ábra. A CQ ütemezési elv	12
13. ábra. A CQ ütemezés algoritmus [9]	12
14. ábra. A WFQ ütemezési elv	13
15. ábra. WFQ példa	14
16. ábra. CBWFQ ütemezési elv	15
17. ábra. LLQ ütemezési elv	16
18. ábra. Szimulációs hálózati topológia	18
19. ábra. Szimuláció: Végponttól-végpontig mért csomagvesztés	20
20. ábra. Szimuláció: Video: Fogadott csomagok száma	21
21. ábra. Szimuláció: Video: Csomag késleltetés	21
22. ábra. Szimuláció: Video: Dzsitter	21
23. ábra. Szimuláció: VoIP: Fogadott csomagok száma	22
24. ábra. Szimuláció: VoIP: Csomag késleltetés	22
25. ábra. Szimuláció: VoIP: Csomag késleltetés	22
26. ábra. Szimuláció: VoIP: Dzsitter	23
27. ábra. Szimuláció: VoIP: Dzsitter	23
28. ábra. Valós mérési laborkörnyezet a forgalomirányítók torlódáskezelési algoritmusainak VoIP-re gyakorolt hatékonyságelemzése céljából	26
29. ábra. Mérés: Az eldobott hangcsomagok száma	28
30. ábra. Mérés: PQ, WFQ, CBWFQ, LLQ: Az eldobott hangcsomagok száma	29
31. ábra. Mérés: A hangcsomagok késleltetése	29
32. ábra. Mérés: PQ, WFQ, CBWFQ, LLQ: A hangcsomagok késleltetése	30
33. ábra. Mérés: A hangcsomagok dzsitterje	30
34. ábra. Mérés: A hangcsomagok dzsitterje	31
35. ábra. A TCP és az MPTCP protokollkészlet	34
36. ábra. Az 50 Gb/s átviteli teljesítményre képes MPTCP környezet [32]	35
37. ábra. Az MPT rétegelt architektúrája [35]	36
38. ábra. Az MPT alapú kommunikáció PDU struktúrája	37
39. ábra. A két utas IPv4-es mérési környezet	38
40. ábra. 1. Eset: Adatméret: 10 MB, Órajel: 1.300.000 / 1.300.000	41

41. ábra. 2. Eset: Adatméret: 20 MB, Órajel: 1.300.000 / 1.300.000.....	42
42. ábra. 3. Eset: Adatméret: 50 MB, Órajel: 1.300.000 / 1.300.000.....	42
43. ábra. 4. Eset: Adatméret: 10 MB, Órajel: 2.000.000 / 2.000.000.....	42
44. ábra. 5. Eset: Adatméret: 20 MB, Órajel: 2.000.000 / 2.000.000.....	43
45. ábra. 6. Eset: Adatméret: 50 MB, Órajel: 2.000.000 / 2.000.000.....	43
46. ábra. 7. Eset: Adatméret: 10 MB, Órajel: 2.000.000 / 1.000.000.....	43
47. ábra. 8. Eset: Adatméret: 20 MB, Órajel: 2.000.000 / 1.000.000.....	44
48. ábra. 9. Eset: Adatméret: 50 MB, Órajel: 2.000.000 / 1.000.000.....	44
49. ábra. a), b), c) – A négy utas IPv4-es mérési környezet.....	47
50. ábra. 2. Eset: Adatméret: 20MB, Órajel: 1M/1M/1M/1M.....	49
51. ábra. 4. Eset: Adatméret: 20 MB, Órajel: 2M/2M/2M/2M.....	49
52. ábra. 6. Eset: Adatméret: 20 MB, Órajel: 2M/1M/1M/1M.....	49
53. ábra. 8. Eset: Adatméret: 20 MB, Órajel: 2M/2M/1M/1M.....	50
54. ábra. 10. Eset: Adatméret: 20 MB, Órajel: 2M/2M/2M/1M.....	50
55. ábra. 16. Eset. Multimédia adatfolyam dzsitter értékek négy utas IPv4-es környezetben	52
56. ábra. Multimédia adatfolyam – Csomagvesztési és csomag-újrarendezési ráta négy utas IPv4-es környezetben.....	53
57. ábra. A négy utas tiszta IPv6-os mérési laborkörnyezet.....	55
58. ábra. 2. IPv4 Teszt eset: Adatméret: 20 MB,.....	58
59. ábra. 3. IPv4 Teszt eset: Adatméret: 10 MB,.....	59
60. ábra. 4. IPv4 Teszt eset: Adatméret: 20 MB,.....	59
61. ábra. 5. IPv4 Teszt eset: Adatméret: 10 MB,.....	59
62. ábra. 6. IPv4 Teszt eset: Adatméret: 20 MB,.....	60
63. ábra. 8. IPv4 Teszt eset: Adatméret: 20 MB,.....	60
64. ábra. 9. IPv4 Teszt eset: Adatméret: 10 MB,.....	60
65. ábra. 10. IPv4 Teszt eset: Adatméret: 20 MB,.....	61
66. ábra. 2. IPv6 Teszt eset: Adatméret: 20 MB,.....	62
67. ábra. 3. IPv6 Teszt eset: Adatméret: 10 MB,.....	63
68. ábra. 4. IPv6 Teszt eset: Adatméret: 20 MB,.....	63
69. ábra. 5. IPv6 Teszt eset: Adatméret: 10 MB,.....	63
70. ábra. 6. IPv6 Teszt eset: Adatméret: 20 MB,.....	64
71. ábra. 8. IPv6 Teszt eset: Adatméret: 20 MB,.....	64
72. ábra. 9. IPv6 Teszt eset: Adatméret: 10 MB,.....	64
73. ábra. 10. IPv6 Teszt eset: Adatméret: 20 MB,.....	65
74. ábra. 2. Vegyes Teszt eset: Adatméret: 20 MB,.....	67
75. ábra. 3. Vegyes Teszt eset: Adatméret: 10 MB,.....	67
76. ábra. 4. Vegyes Teszt eset: Adatméret: 20 MB,.....	67
77. ábra. 5. Vegyes Teszt eset: Adatméret: 10 MB,.....	68
78. ábra. 6. Vegyes Teszt eset: Adatméret: 20 MB,.....	68
79. ábra. 8. Vegyes Teszt eset: Adatméret: 20 MB,.....	68
80. ábra. 9. Vegyes Teszt eset: Adatméret: 10 MB,.....	69
81. ábra. 10. Vegyes Teszt eset: Adatméret: 20 MB,.....	69

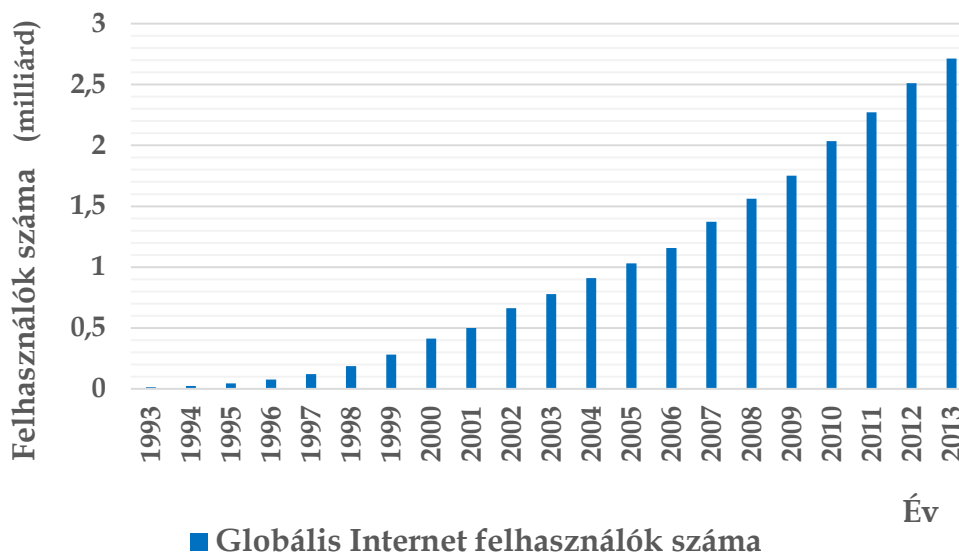
Táblázatjegyzék

1. táblázat. Alkalmazások szolgáltatásminőségi követelményeinek szigorúsága.....	7
2. táblázat. Az IP precedencia mező nevei és értékei	8
3. táblázat. A forgalomirányítók torlódáskezelési algoritmusainak főbb jellemzői [6]	16
4. táblázat. A szimulált alkalmazások beállításai.....	19
5. táblázat. A szimulált torlódáskezelési algoritmusok beállításai	19
6. táblázat. IP címzési tábla a két utas IPv4-es mérési környezet számára.....	39
7. táblázat. IPv4-es két utas mérési eredmények	41
8. táblázat. Az FTP átvitel mérési eredményei négy utas IPv4-es környezetben	48
9. táblázat. Adatfolyam-átvitel mérési eredmények négy utas IPv4-es környezetben	51
10. táblázat. IPv4 és IPv6 címzés	56
11. táblázat. FTP fájlátviteli eredmények – csak IPv4-es hálózat	57
12. táblázat. FTP fájlátviteli eredmények – csak IPv6-os hálózat.....	62
13. táblázat. Mérési eredmények – vegyes hálózat (IPv6 feletti IPv4)	66

1. Bevezetés

Az infokommunikációs hálózatok a rugalmas és hatékony információközlést és feldolgozást, sokrétű szolgáltatást és alkalmazások használatát teszik lehetővé a számítástechnikában, a távközlésben és az elosztott kiszolgáló rendszerek esetében. A jövőben a multimédia és az összetett információs társadalmi alkalmazások egy konvergált, hálózatok hálózatán (Internet) integrált szolgáltatási architektúrán jutnak el a felhasználókhoz. Ezen információs társadalmi technológiák gerincét a hálózatok és szolgáltatásaik adják.

Napjainkban az egyik legnagyobb és legkedveltebb infokommunikációs hálózat maga az Internet, melynek használata az elmúlt években robbanásszerűen növekedett. A felhasználók száma 2001-ben csupán 500 millió volt, egy évvel később ez a szám 662 millióra növekedett, majd 2013-ban elérte a 2,7 milliárdot, ami a Föld lakosság 37,9%-át jelenti. Figyelembe véve, hogy 1993-ban csupán 14 millióan használták az Internetet, belátható, hogy ez a növekedés igen nagymértékű, amint ezt az 1. ábra¹ is mutatja. A felhasználók számának növekedésével robbanásszerűen növekedett az Internet forgalma is. Mivel a felhasználók egyre nagyobb igényt



1. ábra. A globális Internet felhasználóinak száma

¹ Forrás: <http://www.internetlivestats.com/> Letöltve: 2014.07.25.

tartanak a gyors Internet elérésre és kiszolgálásra, az infokommunikációs hálózatok teljesítményelemzése igen fontos területe az informatikai kutatásoknak.

Jelen disszertációm keretében az infokommunikációs hálózatok hatékonyságának vizsgálatát² két nagyon érdekes és egyben fontos kutatási területére szűkítettem le, melyek:

- A forgalomirányítók torlódáskezelési algoritmusainak hatékonyságelemzése
 - a) szimulációs eszköz segítségével
 - b) valós mérési környezet kialakításával
- A Több utas Kommunikációs Könyvtár (MPT)³ hatékonyságelemzése.
 - a) az MPT teljesítményelemzése két utas környezetben
 - b) a több utas FTP és adatfolyam-átvitel hatékonyságelemzése
 - c) az MPT szoftver hatékonyságelemzése IPv4 és IPv6 környezetben

² A hatékonyság fogalma egy eléggé tág fogalom. Jelen dolgozatban az IP csomagvesztés, IP csomagkésleltetés, IP csomagkésleltetés ingadozás (jitter), valamint az átviteli teljesítmény fogalmaival hozható kapcsolatba.

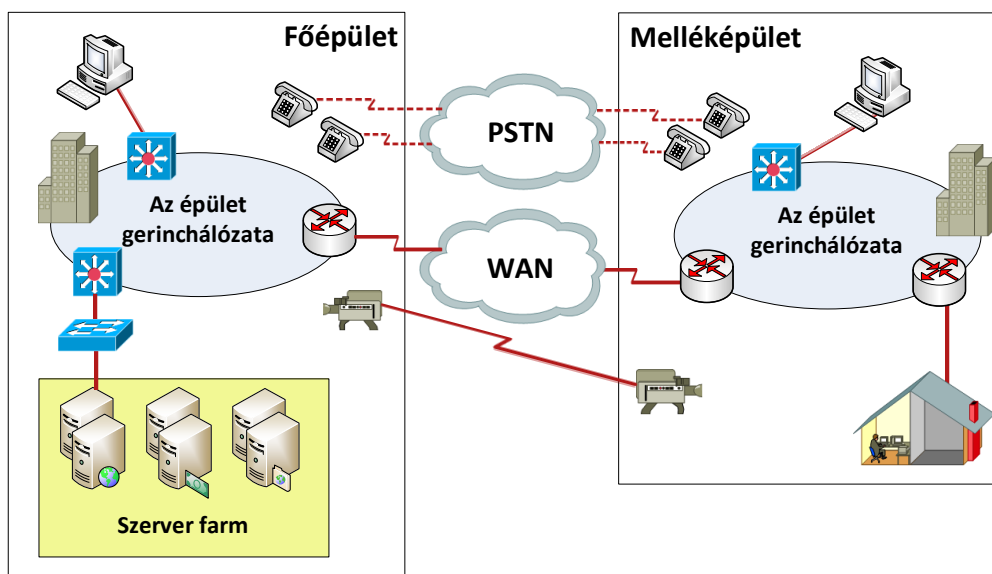
³ Az MPT szoftvert a Debreceni Egyetem Informatikai Karán fejlesztettük, melynek aktuális verziója, valamint dokumentációja a <http://irh.inf.unideb.hu/user/almasi/mpt/> oldalon érhető el.

2. A forgalomirányítók torlódáskezelési algoritmusainak hatékonyságelemzése

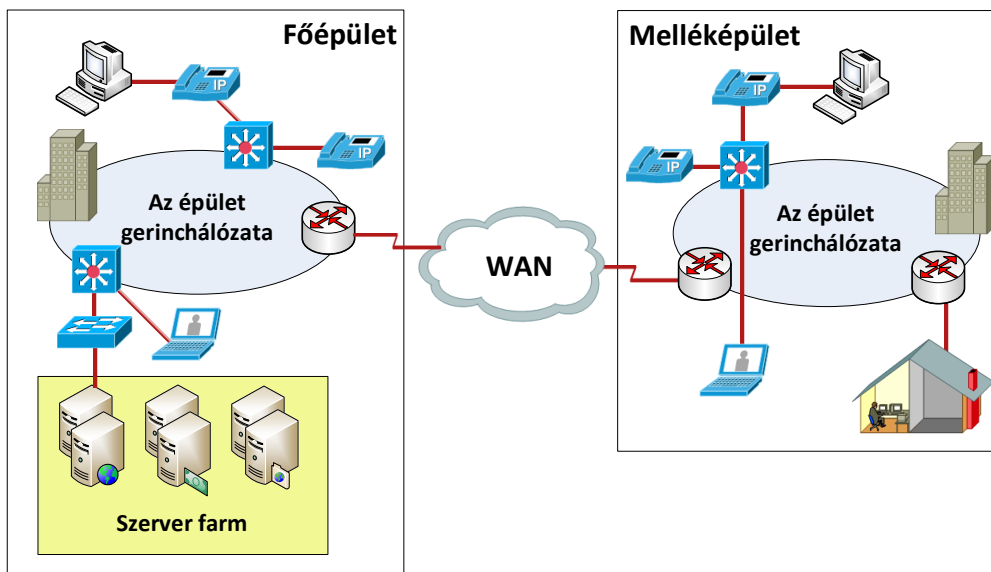
2.1. Bevezetés, alapfogalmak

2.1.1. A konvergált hálózatok kialakulása

Kezdetben az irodai, a vállalati hálózatok és nem utolsósorban, maga az Internet is csupán adatátvitelre (pl. FTP, e-mail) voltak tervezve, ahol az IP csomagkészlés lényegtelennek minősült. Legtöbb esetben a „legjobb szándékú” (Best-effort) kézbesítési szolgáltatás kielégítőnek bizonyult, adatvesztés esetén a TCP protokoll gondoskodott az ismételt adatátvitelről. Később, a multimédiás alkalmazások elterjedése következtében (hangátvitel, videokonferencia) ezek számára külön telefonhálózatokat, illetve videó kommunikációs hálózatokat hoztak létre (2. ábra). Napjainkban az irodai és a vállalati hálózatok átalakulóban vannak konvergált infokommunikációs hálózatokká [1], melyek egyetlen fizikai topológián keresztül biztosítják ugyanazon szolgáltatásokat (3. ábra).



2. ábra. Hagyományos, nem konvergált hálózatok

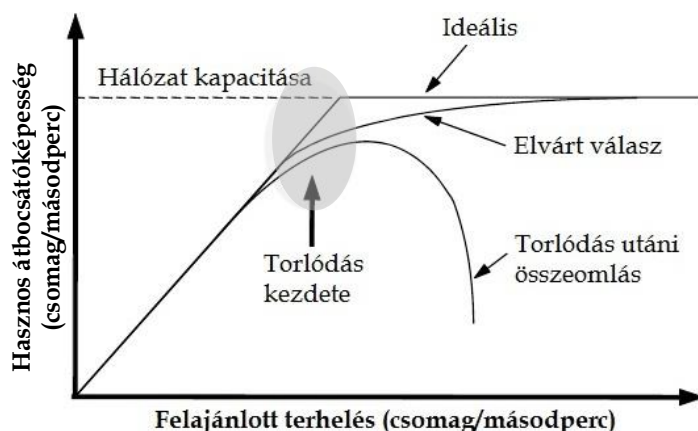


3. ábra. Konvergált infokommunikációs hálózat

A konvergált infokommunikációs hálózatok számos előnye mellett megjelenik egy óriási hátrány, mégpedig a közös hálózati erőforrásokért (pl. forgalomirányítók puffereit) való versengés, ami előbb vagy utóbb torlódáshoz (congestion) vezet. A torlódás tulajdonképpen az az állapot, amikor túl sok csomag van jelen a hálózatban (vagy annak egy részében), ami miatt a teljesítőképesség visszaesik, a csomagküldés pedig késleltetést szenved.

A 4. ábra a torlódás tüneteit mutatja be. Amikor a végpontok által a hálózatba küldött csomagok száma a hálózat átviteli kapacitásán belül marad, akkor minden csomag kézbesítésre kerül (kivéve egy néhányat, amely az átvitel során hibákat szenvedett), és a kézbesített csomagok száma arányos az elküldött csomagok számával (ld. 4. ábra). Ahogy azonban az átvitelre szánt (felajánlott) terhelés eléri a hálózat átviteli kapacitását, a forgalmi löket (burst) alkalmanként megtölti a forgalomirányítók puffereit, melynek következtében egy néhány csomag elvész. Az elveszett csomagok felemésztik a kapacitás egy részét, így a kézbesített csomagok száma az ideális szint alá csökken. A hálózaton torlódás alakul ki. Ha a hálózat nincs jól megtervezve, akkor a torlódás összeomlást okozhat, amelynek hatására a teljesítőképesség gyorsan csökken, amint a terhelés túlnő a kapacitáson. Ez azért történhet meg, mivel a csomagok késleltetése a hálózatban olyan nagy lehet, hogy azok már nem hasznosak, amikor elhagyják a hálózatot. Másféle hibamód jelentkezik akkor, amikor a feladók újraküldenek nagy késleltetésű csomagokat, mivel azt hiszik,

hogy azok elvesztek. Ebben az esetben ugyanannak a csomagnak több példányát továbbítja a hálózat, ami ismét a kapacitás pazarlásával jár. Ezen tényezők rögzítéséhez a 4. ábra y tengelye a hasznos átbocsátóképességet (goodput) ábrázolja. Gyakorlatilag ez az arány, amellyel a hálózat a hasznos csomagokat kézbesíti [2].

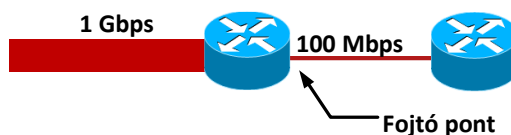
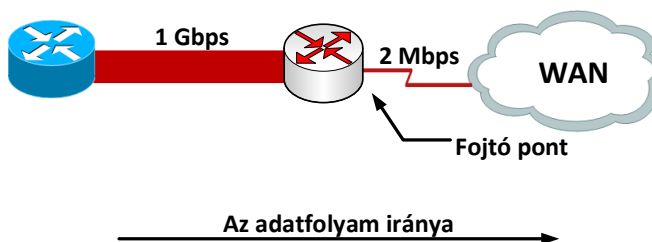


4. ábra. Túl nagy forgalom esetén a teljesítőképesség meredeken visszaesik⁴

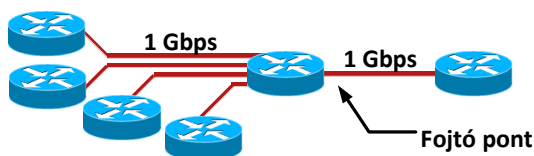
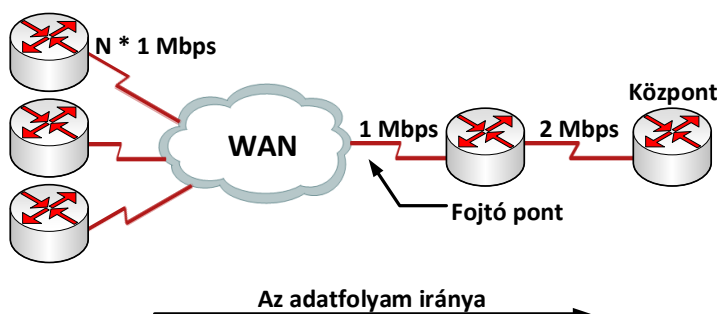
A rendszermérnökök célja olyan hálózatok kialakítása, amelyek lehetőleg elkerülik a torlódást, és nem történik a torlódás után összeomlás, amennyiben mégis torlódás alakulna ki.

A torlódás kialakulását két másik tényező is segítheti: a sebesség eltérések (speed mismatches – 5. ábra), valamint az útvonalösszegzés (aggregáció – 6. ábra) [1]. Sebesség eltérésekről akkor beszélünk, amikor egy adott LAN hálózati gyors adatforgalom egy lassú WAN kapcsolaton keresztül kell áthaladjon vagy amikor egy Gigabit-es hálózati forgalom egy nagyságrenddel alacsonyabb sebességű (100 Mbps) kapcsolaton kell továbbítódjon.

⁴ Forrás: Andrew S. Tanenbaum, David J. Wetherall – *Számítógép-hálózatok* – Harmadik, bővített, átdolgozott kiadás, Panem Könyvek, ISBN 978-963-545-529-4, 416. oldal, Budapest, 2013.



5. ábra. Sebesség eltérés példa



6. ábra. Aggregációs példa

Az útvonalösszegzés jelensége fellelhető úgy LAN, mint WAN környezetben. Torlódás akkor alakulhat ki, amikor több hálózati kapcsolat forgalma egyetlen közös kapcsolaton kell áthaladjon (ld. 6. ábra).

A torlódás következményei a csomagkésleltetés, a késleltetés ingadozás (jitter) és ami a legsúlyosabb, a csomageldobás. Nem minden alkalmazás érzékeny e problémákra. Például az FTP teljesen immunis a csomagkésleltetésre vagy a dzsitterre, míg a multimédiás alkalmazások (videó átvitel, hangátvitel) nagyon is érzékenyek (ld. 1. táblázat) [2].

Alkalmazás	Sávszélesség	Késleltetés	Dzsitter	Veszteség
E-mail	Kicsi	Kicsi	Kicsi	Közepes
Fájlmegosztás	Nagy	Kicsi	Kicsi	Közepes
Webes hozzáférés	Közepes	Közepes	Kicsi	Közepes
Távoli bejelentkezés	Kicsi	Közepes	Közepes	Közepes
Hálózati zenehallgatás	Kicsi	Kicsi	Nagy	Kicsi
Hálózati videózás	Nagy	Kicsi	Nagy	Kicsi
Telefon	Kicsi	Nagy	Nagy	Kicsi
Videokonferencia	Nagy	Nagy	Nagy	Kicsi

1. táblázat. Alkalmazások szolgáltatásminőségi követelményeinek szigorúsága

Ezen problémák kezelése végett vezették be a szolgáltatásminőséget (QoS), ami tulajdonképpen a hálózatok és hálózati eszközök azon képessége, amely által lehetőség nyílik az erőforrások meghatározott rend szerinti felosztására, és garantált paraméterek biztosítására. A QoS-t támogató hálózatokon a magas prioritású üzenetek előnyben részesíthetők alacsonyabb besorolású társaikkal szemben, és konkurencia-helyzetben előbbiek továbbítása utóbbiak feltartóztatásával garantált sebességen biztosítható [3], [4].

2.1.2. Az IP csomagok megjelölése

A hálózatok különböző szolgáltatásminőségi (QoS – Quality of Service) kategóriákat támogathatnak a különböző alkalmazások igényeinek kielégítése érdekében.

A különböző típusú IP csomagok megkülönböztetésére az IP csomagfejrész egyik mezőjét használják. Korábban, az RFC 791 által meghatározott ToS (Type of Service) bájttja szolgálta ezt a célt, majd később a DiffServ⁵ (Differentiated Services) megjelenése következtében átnevezték DS (Differentiated Services) mezőre, ezt követően újra strukturálták a ToS bájtot (ld. 7. ábra) [5].

⁵ A differenciált szolgáltatások IETF architektúra-szabványt többek között az RFC 2474 és RFC 2475 írja le.

0	1	2	3	4	5	6	7
Precedence			D	T	R	0	0

0	1	2	3	4	5	6	7
DSCP						CU	

7. ábra. Az IP fejrész ToS és DS mezőinek összehasonlítása az RFC 791 és az RFC 2474 alapján

A ToS bájtt szerepe a csomagok megkülönböztetése (megjelölése), a különböző QoS (Quality of Service, szolgáltatásminőség) mechanizmusok alkalmazásának céljából (pl. torlódáskezelés). Kezdetben az RFC 791 négy részre tagolta a ToS bájtot: Precedence (IP Precedencia), D (Delay), T (Throughput), R (Reliability) flag-ekre, valamint az utolsó két bitet jövőbeli használatra tartotta fenn (ld. 7. ábra).

Mivel a D, T, R flag-eket nagyon ritkán használták, elmondható, hogy a ToS bájtt legfontosabb flag-je a Precedence flag, mely gyakorlatilag a ToS bájtt legelső 3 legmagasabb helyi értékű bitje, amely a 2. táblázatban feltüntetett értékeket veheti fel.

Név	Decimális érték	Bináris érték
Routine	Precedence 0	000
Priority	Precedence 1	001
Immediate	Precedence 2	010
Flash	Precedence 3	011
Flash Override	Precedence 4	100
Critic/Critical	Precedence 5	101
Internetwork Control	Precedence 6	110
Network Control	Precedence 7	111

2. táblázat. Az IP precedencia mező nevei és értékei

Később, mivel a DiffServ számára nem volt elégséges az Precedence flag 3 bitje, a ToS bájtt újradefiniálásra került. Maga a ToS bájtt is átnevezték DS-re (Differentiated Services), majd a Precedence flag-et átnevezték DSCP-re (Differentiated Services Code Point), valamint megnövelték ennek hosszát 6 bitre [6]. A gyakorlatban a QoS

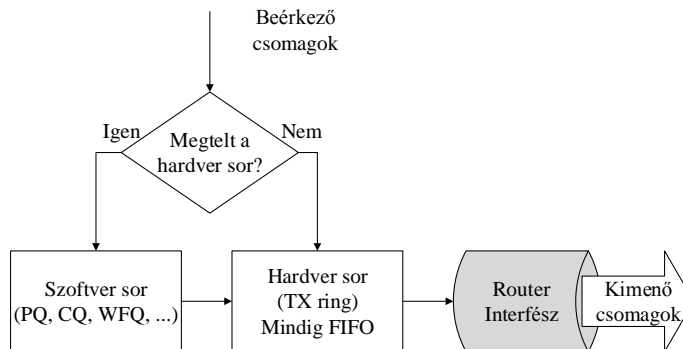
támogatja úgy az Precedence, mint a DSCP használatát, mivel a DSCP értékek kompatibilisek az IP Precedencia értékekkel. Az IP Precedencia értékei 0-7-ig terjednek, míg a DSCP értékek 0-63-ig.

2.1.3. A torlódáskezelési mechanizmusok szükségessége

A multimédiás alkalmazások, valamint a konvergált hálózatok robbanásszerű elterjedésével, egyre égetőbbé vált a hálózati torlódáskezelés⁶, mely különféle sorkezelési mechanizmusokból (csomagütemezés) áll. Ebben a fejezetben az egyik legnagyobb hálózati eszközgyártó (Cisco) forgalomirányítóinak sorkezelési mechanizmusaival foglalkozunk, melyek alapvetően két komponensből tevődnek össze: egy hardver és egy szoftver komponensből [1], amint azt az 8. ábra is mutatja. A működési alapelv a következő: amennyiben a hardver sorban, melyet gyakran „továbbítási sornak” (transmit queue vagy TxQ) is neveznek, nincs torlódás, az IP csomagok egyenesen a hardver sorba lesznek irányítva, kikerülve a szoftver sort, melyből FIFO elv szerint lesznek továbbítva az átviteli közeg felé. Ha a hardver sor betelik, a továbbításra várakozó IP csomagok a szoftver sorban lesznek tárolva és feldolgozva a különböző szoftveresen implementált sorkezelési mechanizmusok szerint, majd továbbítva lesznek a hardver sor felé [7]. A gyakorlatban legtöbbször használt sorkezelési mechanizmusok a következők: FIFO (First-In First-Out), PQ (Priority Queuing), CQ (Custom Queuing), WFQ (Weighted Fair Queuing), CBWFQ (Class Based Weighted Fair Queuing) és LLQ (Low Latency Queuing)⁷.

⁶ Az angol nyelvű szakirodalom különbséget tesz a torlódáskezelés (ide sorolva a csomagütemezést) és a torlódás-megelőzés (ide sorolva pl. a forgalomformálást) fogalma között. A magyar nyelvű szakirodalom viszont többnyire nem tesz különbséget. Jelen dolgozatban az első megközelítést vettem alapul.

⁷ Ezen mechanizmusok bemutatásával a későbbiekben részletesen foglalkozom. Lásd 2.1.4. fejezet

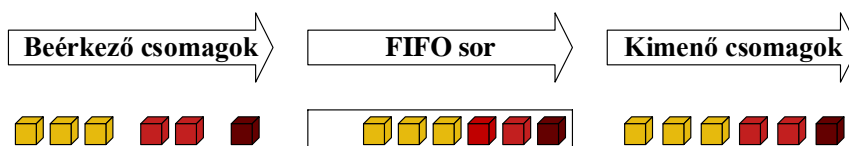


8. ábra. A sorkezelés komponensei

2.1.4. A torlódáskezelési algoritmusok áttekintése

FIFO

A FIFO (First In First Out – Elsőnek Be Elsőnek Ki) kiszolgálási elv a legegyszerűbb. A forgalomirányítókban lévő várakozási sorban a csomagok érkezési sorrendben lesznek kiszolgálva (9. ábra). Ha a FIFO sor megtelik, az újonnan érkező IP csomagok eldobásra kerülnek (tail drop – farok eldobás).

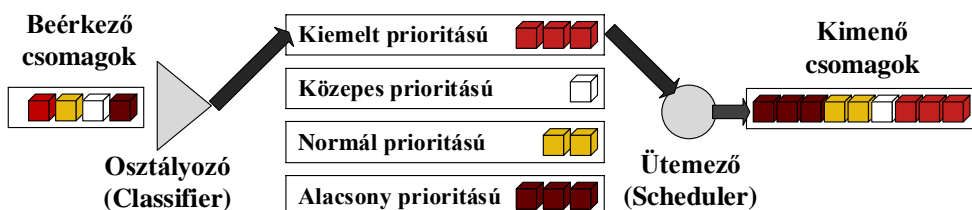


9. ábra. A FIFO ütemezési elv

Ennek a módszernek nagyon minimális az erőforrás igénye, ugyanakkor előre becsülhető a viselkedése, mivel például a csomagok késleltetése egyenesen arányos a FIFO sorhosszal. A Cisco hálózati eszközök operációs rendszerei alapértelmezetten a FIFO sorkezelést alkalmazzák a gyors interfészeiken (Fast Ethernet, Gigabit Ethernet) [8]. A FIFO ütemezési elv egyik óriási hátránya, hogy nem képes prioritást biztosítani a multimédiás alkalmazások csomagjai számára, egyetlen várakozási sorba sorolva minden csomagtypust, ami miatt azok késleltetést, dzsittert vagy csomagvesztést szenvedhetnek.

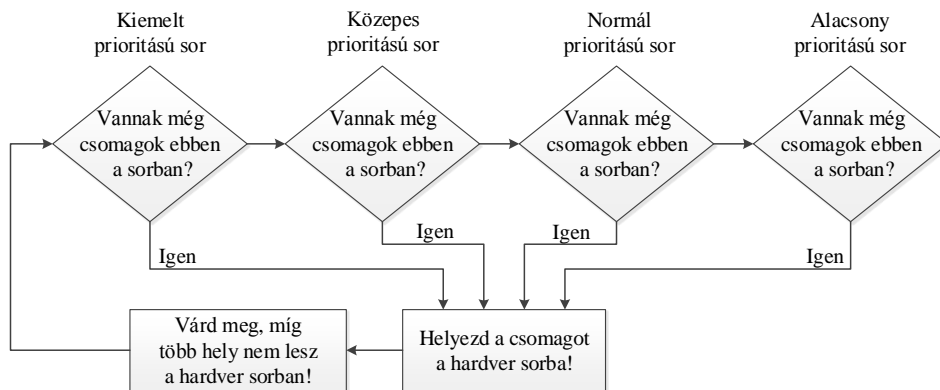
PQ

A PQ (Priority Queuing – Prioritásos Ütemezés) négy darab FIFO elven alapuló prioritásos sort tartalmaz, melyekre a már korábban említett farok eldobásos elvet alkalmazza (10. ábra) [9]. A legelső sor rendelkezik a legmagasabb prioritással, a többiek pedig csökkenő prioritásúak. Legelőször a legnagyobb prioritással rendelkező sorban lévő csomagok lesznek feldolgozva, majd az eggyel kisebb prioritású sorban levőké. Minden egyes csomag feldolgozása után a PQ ütemező leellenőrzi, hogy nem-e érkezett új csomag a legnagyobb prioritással rendelkező sorba. Amennyiben érkezett, az lesz feldolgozva (11. ábra).



10. ábra. A PQ ütemezési elv

A PQ algoritmus erőssége, hogy képes abszolút prioritást biztosítani a multimédiás alkalmazások számára, viszont ezzel könnyen kiéheztetheti a többi három prioritásos sorba sorolt csomagokat.

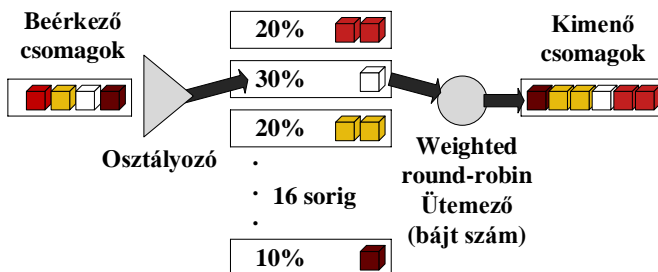


11. ábra. A PQ ütemezés algoritmus [9]

CQ

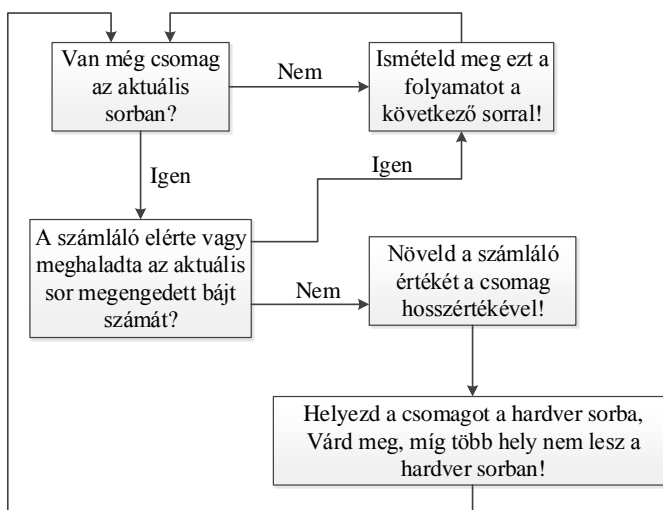
A CQ (Custom Queuing) segítségével a PQ fő hiányosságát próbálták meg kiküszöbölni, nevezetesen az alacsonyabb prioritással rendelkező sorok kiéheztetését. A rendszermérnököknek módjukban áll az alkalmazásokat 16 darab

FIFO sorba osztályozni (12. ábra). További lehetőség van mindenegyes sorhossz meghatározására, ami által beállítható, hogy egy bizonyos alkalmazás a teljes sáv szélesség hány százalékát használhatja. A sorokban levő csomagok kiszolgálása körbeforgásos elv (Round Robin – ld. [2]) szerint történik (13. ábra).



12. ábra. A CQ ütemezési elv

A CQ-val sikerülhet a csomagok kiéheztetésének megakadályozása, viszont a CQ nem képes a multimédiás csomagok számára prioritást biztosítani. Mindazonáltal a sorhosszak finomhangolásával viszonylag jó eredményeket lehet elérni [5].

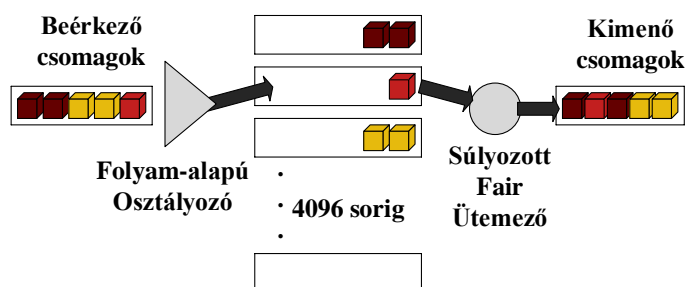


13. ábra. A CQ ütemezés algoritmus [9]

WFQ

A WFQ (Weighted Fair Queuing) egy súlyozott igazságos ütemezési elv. Maga az algoritmus teljesen automatizált, a rendszermérnököknek nincs lehetőségük a

finomhangolásra, csupán az algoritmus aktiválására. A WFQ adatfolyamokat használ, melyeket maximálisan 4096 sorba osztályoz [10] (14. ábra).



14. ábra. A WFQ ütemezési elv

Minden egyes beérkező IP csomag időbélyeggel lesz ellátva (sorszám), majd a neki megfelelő adatfolyam sorba lesz helyezve. A WFQ ütemező megvizsgálja a csomagok időbélyegeit, majd a legalacsonyabb időbélyeg-értékkel vagy sorszámmal rendelkező csomagot fogja feldolgozni (a kimenő hardver sorba helyezni). A gyakorlatban az is megtörténhet, hogy egy később érkezett csomag alacsonyabb időbélyeg-értékkel rendelkezzen, mint egy korábban érkező csomag.

Az adatfolyamok osztályozása az alábbi paramétereket veszi figyelembe: forrás IP cím, cél IP cím, szállítási protokoll (TCP vagy UDP), az IP csomagfejrész ToS mezője (IP_Precedencia), forrás TCP vagy UDP port szám, cél TCP vagy UDP port szám. A sor index (sorszám) kiszámításához a WFQ Hash algoritmust használ [9].

A WFQ ütemező az alábbi módon számítja ki egy IP csomag időbélyegét:

$$\text{Sorszám} = \text{Előző_sorszám} + (\text{súly} * \text{új_csomag_hossz}) \quad (1.)$$

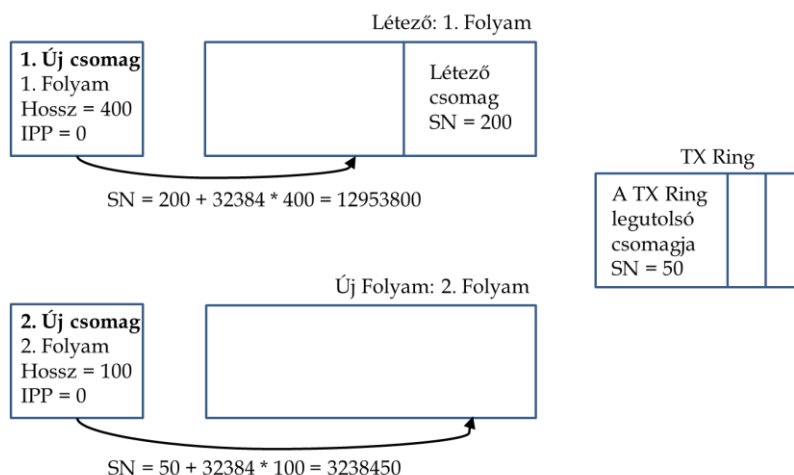
$$\text{súly} = 32384 / (\text{IP_Precedencia} + 1) \quad (2.)$$

Ahol a *Sorszám* a befejezési időt (időbélyeg), az *Előző_sorszám* az előzően kiszolgált csomag időbélyegét, az *új_csomag_hossz* az aktuális csomag hosszát, míg az *IP_Precedencia* az IP csomag fejrészében levő ToS (Type of Service) mező értékét jelöli.

Jól látható, hogy a csomaghossz arányos az időbélyegnek az előző időbélyeghez képest való növekményének értékével, azaz egy hosszú csomag nagyobb értékű időbélyeggel fog rendelkezni, mint egy rövidebb IP csomag. Az előző csomag időbélyegnek (*Előző_sorszám*) köszönhetően az a csomag, amelyik egy olyan folyamsorba lesz besorolva, amelybe már több csomag van, nagyobb időbélyeget fog kapni, mint egy olyan csomag, amelyik egy üres sorba lesz besorolva. Mivel az *IP_Precedencia+1* a képlet nevezőjébe kerül, ezért a nagyobb IP Precedenciával

(prioritással) rendelkező csomagok alacsonyabb értékű időbélyeggel lesznek ellátva, ily módon elsőbbséget élvezve az alacsonyabb prioritással rendelkező csomagokkal szemben.

Tekintsünk egy konkrét példát, amely segítségével érthetőbbé válik a WFQ ütemező által használt időbélyeg kiszámítási módja (15. ábra) [5].



15. ábra. WFQ példa

A hardver sor legutolsó csomagjának sorszáma (SN) 50. Kezdetben egyetlen szoftver sor (folyam) létezik, amelybe egyetlen IP csomag vár kiszolgálásra, melynek sorszáma 200.

Egy új csomag érkezésekor a rendszerbe a WFQ ütemező megvizsgálja az újonnan érkező csomagot. Ennek paramétereit figyelembe véve a csomagot az első folyamba sorolja, majd kiszámolja az új csomag sorszámát az (1.) és (2.) képleteket használva. Az egyszerűség kedvéért az IPP értéke jelen példában 0. Az így kapott sorszám értéke 12953800 lesz.

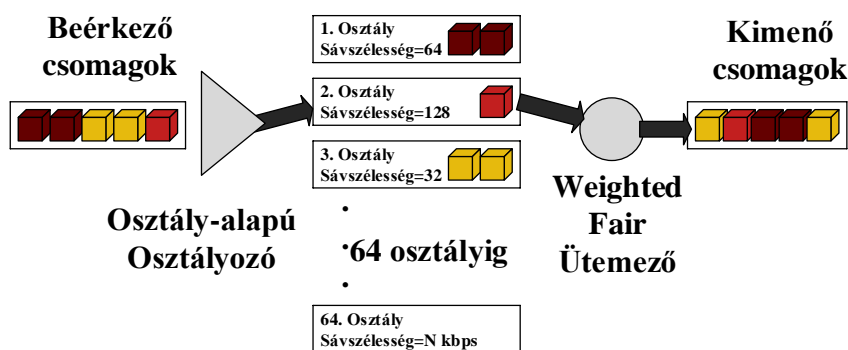
Kis idő elteltével egy másik csomag érkezik a rendszerbe. A WFQ ütemező a csomag paramétereit alapján látja, hogy a csomag nem sorolható a meglévő sorba, mivel egy másik folyam része. (Például különböző forrás IP címmel vagy port számmal rendelkezik.) Ez okból létrehoz a 2. folyam típus számára egy új sort, melybe belehelyezi az újonnan érkező csomagot. A WFQ algoritmus kiszámítja a csomag sorszámát, amely 3238450 lesz.

Jelen példában, a rendszer ebben a pillanatban négy csomagot tartalmaz (15. ábra). Mivel a WFQ ütemező a legalacsonyabb értékű sorszámmal rendelkező csomagot részesíti előnyben, ezért a kiszolgálás sorrendje a következő: a hardver sor csomagja

(SN=50), az első folyam első csomagja (SN=200), a második folyamsor csomagja (SN=3238450), míg legvégül az első folyam második csomagja lesz kiszolgálva (SN=12953800).

CBWFQ

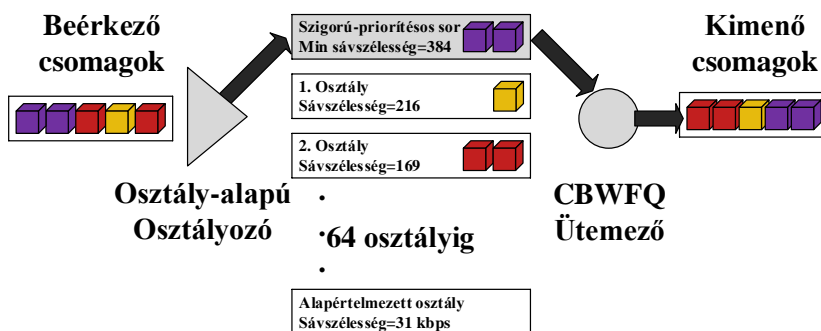
A CBWFQ (Class Based Weighted Fair Queuing – Osztály Alapú Súlyozott Igazságos Ütemező) a WFQ algoritmus működési elvén alapszik, azzal a különbséggel, hogy míg a WFQ algoritmus adatfolyam alapú, addig a CBWFQ adatosztályokat használ (16. ábra.) [6]. Az IP csomagok osztályozása nem történik automatikusan, mint a WFQ esetében, hanem lehetőség nyílik a rendszermérnökök általi manuális beállításra. A WFQ és a CBWFQ fő hátránya, hogy nem képes prioritást biztosítani a multimédiás alkalmazások számára [8].



16. ábra. CBWFQ ütemezési elv

LLQ

Az LLQ (Low Latency Queuing – Alacsony Késleltetésű Ütemező) a CBWFQ algoritmus működési elvén alapszik azzal a különbséggel, hogy tartalmaz egy szigorú-prioritásos sort (LLQ), melybe egy multimédiás alkalmazás sorolható (17. ábra.) [11], [12], [13]. Ily módon sikerül ötvözni a PQ és a WFQ előnyeit, ugyanakkor csökkenthetőek ezen algoritmusok hátrányai.



17. ábra. LLQ ütemezési elv

A következő táblázat összefoglalja a fentebb tárgyalt torlódáskezelési algoritmusok főbb jellemzőit:

Funkciók	FIFO	PQ	CQ	WFQ	CBWFQ	LLQ
Tartalmaz szigorú-prioritásos sort		Igen				Igen
Prioritásos sorok házirendje más sorok kiéheztesének elkerülése végett						Igen
Soronkénti sávszélesség foglalás			Igen		Igen	Igen
Folyamankénti osztályozás				Igen	Igen	Igen
Maximálisan engedélyezett sorok száma	1	4	16	4096	64	64

3. táblázat. A forgalomirányítók torlódáskezelési algoritmusainak főbb jellemzői [6]

2.2. A forgalomirányítók torlódáskezelési algoritmusainak hatékonyságelemzése szimulációs szoftver segítségével

2.2.1. Irodalmi előzmények

A torlódáskezelési algoritmusok vizsgálata és elemzése nagyon kurrens tudományterület. Az utóbbi években számos dolgozat jelent meg (ld. [14], [15], [16]) melyekben a torlódáskezelési algoritmusok hatékonyságvizsgálatával foglalkoztak, s

matematikai modelleken alapuló szimulációs szoftver segítségével (OPNET) határozták meg a legfontosabb paramétereket. Minden esetben arra a végkövetkeztetésre jutottak, mely szerint a WFQ algoritmus a leghatékonyabb sorkezelési mechanizmus a multimédiás alkalmazások, kiváltképpen a hangátvitel számára.

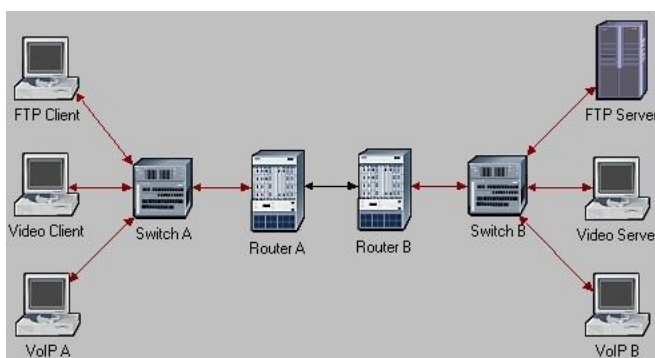
Korábbi tudományos munkáimban (ld. [17], [18], [19]) a mások által vizsgált torlódáskezelési algoritmusok mellett (FIFO, PQ, WFQ) még további három algoritmust tanulmányoztam (CQ, CBWFQ és LLQ) az OPNET IT Guru Academic Edition szimulációs szoftver segítségével.

2.2.2. Anyag és módszer

A szimuláció elvégzésére a jelen kutatási témában megjelent korábbi tudományos munkákban (ld. [14], [15], [16]) használt szimulációs szoftvert használtam, nevezetesen az OPNET IT Guru Academic Editiont⁸. A csomagütemezési algoritmusok teszteléséhez használt hálózati topológia a korábbi cikkekben szereplő topológián alapult, melyet tovább általánosítottam, egy bővíthető modell elnyerése érdekében (18. ábra.). A korábbi tudományos munkák szerzői mellőzték a hálózati kapcsolók (switchek) használatát, melynek következtében a szerver- és a kliens gépek közvetlenül a 2 forgalomirányítóhoz csatlakoztatták. Az így kapott hálózati topológia-modell kevésbé felelt meg a valóságnak, mivel a forgalomirányítók rendszerint kevés fizikai interfésszel rendelkeznek. Ebből kifolyólag válik szükségessé a gyakorlatban a kapcsolók használata. Emiatt döntöttem úgy, hogy a szimulációk elvégzésére egy kapcsolókat tartalmazó hálózati topológia-modellt fogok használni (ld. 18. ábra). Mi több, a korábbi cikkek csupán három torlódáskezelési algoritmust vizsgáltak: a FIFO-t, a PQ-t és a WFQ-t. Minden esetben azt a végkövetkeztetést vonták le, miszerint a WFQ a leghatékonyabb algoritmus a multimédiás alkalmazások számára. Ezzel szemben még további három algoritmust használtam: a CQ-t, a CBWFQ-t és az LLQ-t. Céлом volt megvizsgálni, hogy létezik-e a WFQ-nál hatékonyabb torlódáskezelési algoritmus a multimédiás alkalmazások számára (főleg a hangátvitel esetében).

Az OPNET IT Guru Academic Edition programban használt szimulációs környezetet a 18. ábra mutatja.

⁸ Jelen dolgozatban említett OPNET szimulációs szoftvert a korábbi tudományos munkáimban használtam. 2012. október 29.-én az OPNET-et felvásárolta a Riverbed cég, majd 2014 tavaszától az OPNET szimulációs szoftvert felváltotta a Riverbed Modeler [47].



18. ábra. Szimulációs hálózati topológia

A szimulációs környezet két forgalomirányítóból, két darab hálózati kapcsolóból és hat darab végpontból állt. A forgalomirányítók között pont-pont kapcsolatot hoztam létre *ppp-DS1* kapcsolati sebességgel. A többi eszköz *10BaseT* linkkel kapcsolódott egymáshoz. A szimulációk során, a korábbi tudományos munkáknak megfelelően, három adatforgalom típust vizsgáltam: a fájlátvitelt (FTP), a video átvitelt és az valós idejű hangátvitelt (VoIP). A *Switch A* kapcsolóhoz csatlakoztattam a klienseket, míg a *Switch B* kapcsolóhoz a kiszolgálók (szerverek) csatlakoztak. Így módon a három forgalomtípus a három kliens-szerver alapú végpont-páros között zajlott. A hálózat szűk keresztmetszetét, ahol a torlódás létrejöhet, a két forgalomirányító közötti rész képezi [20], [21]. Ezért, ebben a zónában aktiváltam a torlódáskezelési algoritmusokat.

Az alábbi táblázat foglalja össze a szimulációk során megvizsgált három alkalmazás beállítási értékeit. Jelen paraméterek segítségével megismételhetők az általam végzett szimulációk. Megjegyzendő, hogy a korábbi tudományos munkák szerzői által használt ugyanazon beállításokat és paramétereket használtam.

	FTP	Video	VoIP
Start Time Offset (seconds)	constant(5)		
Duration (seconds)	End of Profile		
Repeatability	Once at Start Time		
Operation Mode	Simultaneous		
Start Time (seconds)	constant(100)		
Duration (seconds)	End of Simulation		
Inter-Request Time	constant(1)	-	-
File Size	constant(1000000)	-	-
Traffic Type	High Load	Low Resolution Video	IP Telephony
Frame Interarrival Time	-	10 frames/sec	-
Frame Size	-	128x120 pixels	-
Encoder Scheme	-	-	G.729
Type of Service	Best Effort (0)	Streaming Multimedia (4)	Interactive Voice (6)

4. táblázat. A szimulált alkalmazások beállításai

A torlódáskezelési algoritmusok beállításait az alábbi táblázat tartalmazza (5. táblázat). Jól látható, hogy a három forgalomtípust három különböző sorba soroltam (a FIFO ütemezési elv kivételével). Az FTP csomagokat Best Effort (0), a Video forgalmat Streaming Multimedia (4), míg a valós idejű hangátvitel forgalmát Interactive Voice (6) ToS értékekkel jelöltem meg. Ez által biztosítva a hangcsomagok számára a legmagasabb kiszolgálási prioritást.

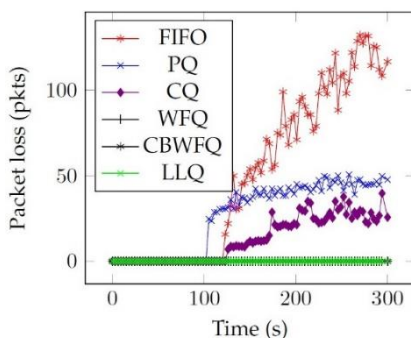
Algorithm	FIFO	PQ	CQ	WFQ/CBWfq	LLQ
Rows	1	4	8	8	8
ProfileName	-	ToS Based			
Maximum Queue size (pkts)	500	queue 0: 10 queue 2: 10 queue 3: 10	queue 0: 200 queue 4: 100 queue 6: 200	queue 0: 500 queue 4: 500 queue 6: 500	queue 0: 500 queue 4: 500 queue 6: 500
Classification Scheme	-	Best Effort (0) Streaming Multimedia (4) Interactive Voice (6)			
Byte Count (bytes)	-	- - -	1000 20000 850	- - -	- - -
Weight	-	- - -	- - -	1.0 40 60	1.0 40 60
Queue Category	-	Default Queue - -			Default Queue - Low Latency Queue

5. táblázat. A szimulált torlódáskezelési algoritmusok beállításai

2.2.3. Eredmények

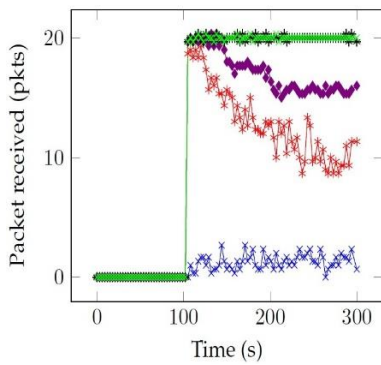
A szimuláció során lehetőség volt számos statisztikai vizsgálatra. Jelen munkámban csupán a legfontosabbakat vettem figyelembe, nevezetesen a globális csomageldobási rátát, a fogadott csomagok számát, a csomagok végponttól-végpontig történő késleltetését, valamint a késleltetés ingadozást (dzsittert). Minden esetben a szimuláció időtartama 5 perc volt, ami 2 perccel haladta meg a korábbi tudományos munkákban használt értéket, ezáltal részletesebb eredményeket nyerve. A torlódáskezelési algoritmusok csak torlódás esetén lépnek működésbe [8]. Ez a tény érhető tetten az alábbi ábrákon (19.-27. ábrák). Kezdetben a hálózat működését torlódásmentesség jellemezte, majd a 105. másodperctől kezdve hálózati torlódás lépett fel. A torlódásmentességi időtartam tetszőlegesen állítható be az OPNET szimulációs környezetben a *Start Time*, illetve a *Start Time Offset* paraméterek segítségével (ld. 4. táblázat). Emiatt a torlódáskezelési algoritmusok csak a 105. másodperctől aktiválódnak. A szimulációs eredmények a 19.-27. ábrákon láthatóak, illetve a numerikus eredményeket az *A. Függelék* tartalmazza.

A 19. ábra alapján elmondható, hogy a globális csomagvesztés a FIFO esetében a legnagyobb, amint ez egyébként várható is volt. A PQ esetében mindamellett, hogy a hangcsomagok eldobási rátája nulla volt, a legnagyobb prioritású sor miatt, óriási csomagvesztés jelentkezett a videó, illetve az FTP csomagok esetében. Ennek a fő oka a legnagyobb prioritású sor (hangcsomag forgalom) abszolút prioritásából fakadó kiehéztetés (starvation) [2], [6]. A CQ csomagütemezési elv esetében egy kicsivel jobb eredményt lehetett elérni a csomagvesztés terén, viszont egy elfogadható működés elérése érdekében elég sok finomhangolás igényeltetik a rendszermérnökök részéről [9]. A globális csomagvesztés tekintetében a WFQ, CBWFQ, illetve az LLQ algoritmusok bizonyultak a leghatékonyabbaknak, mivel az ő esetükben a csomagvesztés minimális volt.

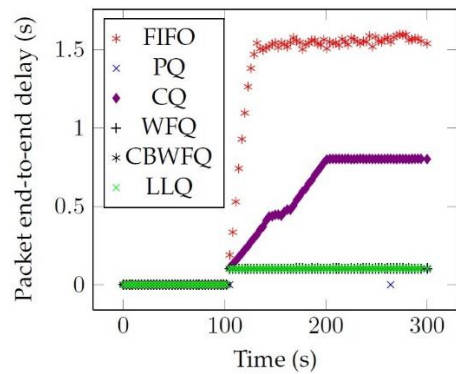


19. ábra. Szimuláció: Végponttól-végpontig mért csomagvesztés

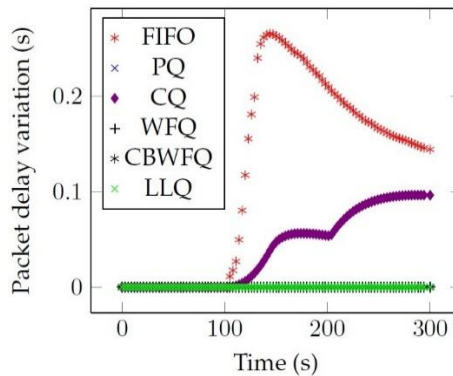
A következő ábrák a torlódáskezelési algoritmusok videó forgalomra gyakorolt hatását mutatják be (20.-22. ábrák). Három paramétert vizsgáltam: a fogadott videó csomagok számát, a csomagkésleltetést és a dzsittert. A fogadott csomagok száma fordítottan arányos a csomagvesztéssel, ezért gyakorlatilag a 20. ábra azonos a 19. ábrával. Ez alól csupán a PQ kivétel, a PQ működési elvből fakadó csomagkiéheztetés miatt (ahogyan azt már korábban említettem). A PQ esetében kaptuk a legrosszabb eredményt, mivel a hangcsomagok kiéheztették a náluk alacsonyabb prioritású videó csomagokat.



20. ábra. Szimuláció: Video: Fogadott csomagok száma



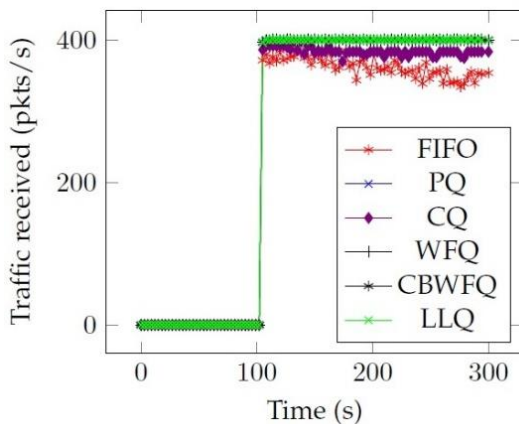
21. ábra. Szimuláció: Video: Csomag késleltetés



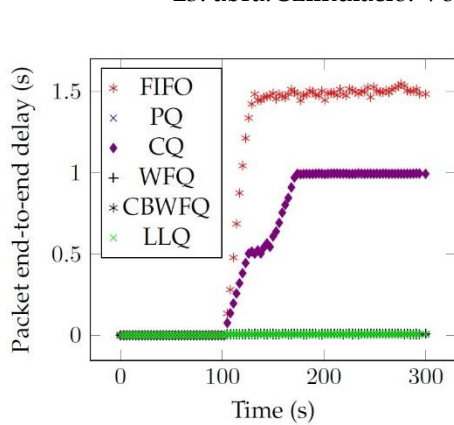
22. ábra. Szimuláció: Video: Dzsitter

A videó csomagok késleltetése tekintetében (21. ábra) a FIFO nyújtotta a leggyengébb eredményt, azt követően pedig a CQ. A többi négy algoritmus (PQ, WFQ, CBWFQ, LLQ) teljesítménye elfogadhatónak bizonyult. Ami a dzsittert illeti a videó fájlok esetében (ld. 22 ábra), a FIFO algoritmus bizonyult ismét a legkevesbé

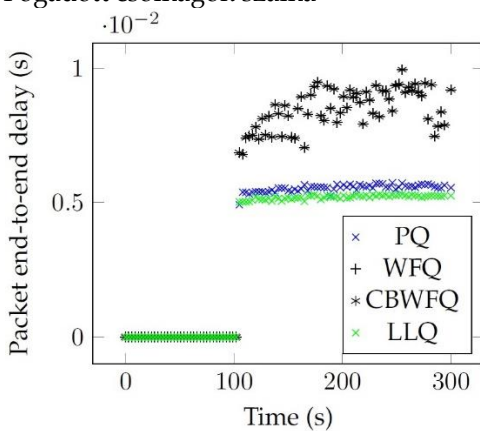
hatékonyak, a CQ viszont egy kicsivel jobban teljesített. A többi algoritmus (PQ, WFQ, CBWFQ, LLQ) ismét hatékonyak bizonyult.



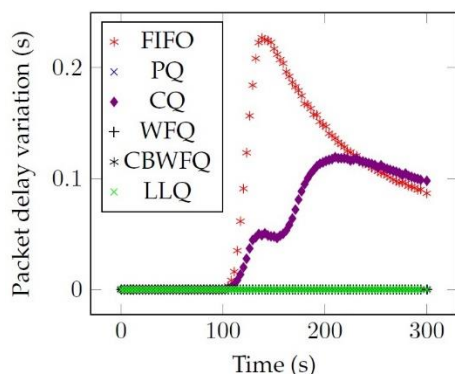
23. ábra. Szimuláció: VoIP: Fogadott csomagok száma



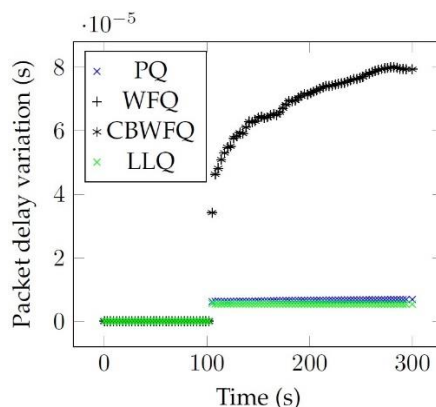
24. ábra. Szimuláció: VoIP: Csomag késleltetés



25. ábra. Szimuláció: VoIP: Csomag késleltetés



26. ábra. Szimuláció: VoIP: Dzsitter



27. ábra. Szimuláció: VoIP: Dzsitter

A hangátvitelnél is ugyanazokat a jellemzőket vizsgáltam, mint a videó átvitel esetében. Nevezetesen: a fogadott hangcsomagok számát, a hangcsomagok késleltetését, valamint a hangcsomagok késleltetés-ingadozását. A fogadott hangcsomagok tekintetében (23. ábra) a PQ, WFQ, CBWFQ és az LLQ algoritmusok bizonyultak a leghatékonyabbaknak. A CQ és a FIFO gyengébb eredményeket értek el, viszont még elfogadható volt a teljesítményük.

Ami az interaktív hangalapú kommunikációt illeti, a QoS előírja, hogy a hangcsomagok késleltetése nem haladhatja meg a 150 ms-ot [9]. Ennek a feltételnek sem a FIFO, sem a CQ nem tett eleget (ld. 24. ábra). A QoS szintén előírja, hogy a dzsitter nem haladhatja meg a 30 ms-ot [9]. A FIFO és a CQ algoritmusok ennek a feltételnek sem tettek eleget (ld. 26. ábra).

Végül a négy leghatékonyabb algoritmust hasonlítom össze a hangcsomagok késleltetése, illetve a dzsitter tekintetében. Amiatt esett a választás a hangcsomagokra, mivel ők a legérzékenyebbek úgy a csomagkésleltetésre, mint a késleltetés ingadozásra. Érdekes eredmények születtek (25. és 27. ábrák). A korábbi tudományos munkák szerzői azt a következtetést vonták le, miszerint a WFQ algoritmus a leghatékonyabb az interaktív hangátvitel számára. Jól látható azonban, hogy ami a csomagkésleltetést illeti, a WFQ és a CBWFQ algoritmusok szórása nagyobb az LLQ-énál. Sőt, a WFQ esetében a dzsitter értéke magasabb a PQ és az LLQ algoritmusokéinál. A szimulációs eredményeim alapján (25. és 27. ábrák) azt a következtetést vontam le, hogy a hangátvitel számára létezik a WFQ-nál hatékonyabb torlódáskezelési algoritmus, nevezetesen az LLQ.

2.2.4. Az eredmények összefoglalása

Ebben a fejezetben egy rövid áttekintést adtam a forgalomirányítók torlódáskezelési algoritmusairól. A korábbi tudományos munkákban tárgyalt algoritmusok mellett további hármat vizsgáltam meg. A szimulációs környezetet az OPNET IT Guru Academic Edition matematikai modelleken alapuló szimulációs szoftver biztosította [22]. Szimulációimhoz egy általánosabb, bővíthetőbb és valósághűbb hálózati topológiát használtam. Következtetésképpen elmondható, hogy hangátvitel számára az LLQ egy hatékonyabb torlódáskezelési algoritmus a WFQ-val szemben.

2.3. A forgalomirányítók torlódáskezelési algoritmusainak VoIP-re gyakorolt hatékonyságelemzése valós mérési környezet kialakításával

2.3.1. Bevezetés

Az előző fejezetben rávilágítottam az infokommunikációs hálózatokon gyakorta előforduló csomagtorlódás okaira, illetve az ezzel járó negatív következményekre. Bemutattam a gyakorlatban leggyakrabban használt hat darab torlódáskezelési algoritmus (FIFO, PQ, CQ, WFQ, CBWFQ, LLQ) működési elvét, majd megvizsgáltam ezeknek a fájlátvitelre, video átvitelre, illetve a hangátvitelre gyakorolt hatását az OPNET IT Guru Academic Edition szimulációs szoftver segítségével.

Az interaktív hangátvitel (VoIP) a legérzékenyebb multimédiás alkalmazás a késleltetésre, illetve a dzsitterre. A videó átvitel esetében egy kis késleltetés vagy késleltetés ingadozás a képkockák akadozását eredményezheti, ugyanakkor nem jelent különösebb gondot, mivel a hang alapján is ki lehet következtetni a hiányzó cselekményt. Ez az észrevétel fordítva is igaz. A VoIP esetében azonban nagyon zavaró lehet a késleltetés, a hangcsomagok elvesztése, illetve a dzsitter következtében kialakuló visszhang [2]. A QoS előírásoknak megfelelően a hangátviteli csomagvesztés nem lépheti túl az 1%-os küszöböt, a kéreltetés nem haladhatja meg a 150 ms-ot, míg a dzsitter a 30 ms-ot [9].

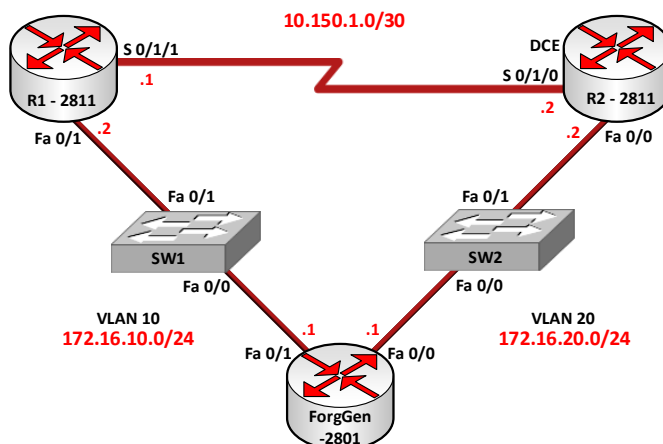
Korábbi tudományos munkák szerzői csupán szimulációs eredményeket közöltek, melyek elérésére csupán három ütemezési algoritmust használtak (FIFO, PQ, WFQ).

Ebben a fejezetben a korábban felsorolt hat darab torlódáskezelési algoritmusokkal fogok foglalkozni. Bemutatom ezen mechanizmusok az interaktív hangátvitelre gyakorolt hatását, azzal a különbséggel, hogy az algoritmusok implementációját és hatékonyságelemzését valós hálózati eszközök segítségével, valós laborkörnyezetben végzem⁹.

2.3.2. Mérési környezet

A mérési környezet fizikai topológiáját a 28. ábra mutatja. A mérési környezetet a Debreceni Egyetem Informatikai Karának Hálózati Laborjában állítottam fel. A hálózati topológia kialakításához három darab forgalomirányítót (egy darab Cisco 2801-es típusút és két darab 2811-es típusút) használtam. A 28. ábrán látható R1 és R2-es útválasztók IOS 12.4 verziójú operációs rendszerrel rendelkeztek. A *ForgGen* forgalomirányító valójában a kommunikációs végpont funkcionálisát látja el. Ezt forgalomgenerálás céljából használtam, melyen egy speciális operációs rendszer, nevezetesen a *c2801-tpgen+ipbase-mz.PAGENT.4.3.0.* futott [23], ami lehetővé tette a forgalomgenerálást, a kimenő csomagok időbélyeggel való ellátását, valamint a beérkező csomagok rátája alapján különböző statisztikai kimutatások elvégzését. A generált, illetve a bejövő forgalom szétválasztása céljából két darab Cisco 2960-as típusú kapcsolót használtam (SW1, SW2). Ezek segítségével két VLAN-t (Virtual LAN) hoztam létre (VLAN 10 és VLAN 20) [24]. Az R1 és R2-es forgalomirányítók között soros összeköttetés lett kiépítve, amely egy lassú WAN kapcsolatot szimulált. Három típusú forgalmat generáltam a korábbi tudományos munkákra támaszkodva, nevezetesen FTP, Video és VoIP forgalmakat. A forgalomirányítók konfigurációs fájljainak tartalmát az *B. Függelék* tartalmazza.

⁹ A kutatás a TÁMOP 4.2.4.A/2-11-1-2012-0001 azonosító számú Nemzeti Kiválóság Program – Hazai hallgatói, illetve kutatói személyi támogatást biztosító rendszer kidolgozása és működtetése konvergencia program című kiemelt projekt keretében zajlott. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.



28. ábra. Valós mérési laborkörnyezet a forgalomirányítók torlódáskezelési algoritmusainak VoIP-re gyakorolt hatékonyságelemzése céljából

2.3.3. A forgalom generálása

A forgalom generálásához a következő kódot használtam [25]:

wait-after-stop 1 !várakozási időtartam másodpercben kifejezve

!FTP forgalom generálása

```
fastethernet0/1 ! a ForcGen kimenő fizikai interfészének neve
add tcp ! a szállítási protokoll típusa
datalink ios-dependent fastethernet0/1 ! a kimenő fizikai interfész neve
l2-arp-for 172.16.10.2 ! Layer 2-es ARP üzenet a default gw felé
l3-src 172.16.10.1 ! Layer 3 forrás IP cím
l3-dest 172.16.20.1 ! Layer 3 cél IP cím
l3-tos 0x00 ! Layer 3 IP csomagfejrész ToS mezőjének hexadecimális értéke
l4-src 21 ! Szállítási rétegbeli forrás portszám
l4-dest 21 ! Transzport rétegbeli cél portszám
name FTP ! a generált forgalom neve
rate 20 !a csomaggenerálási ráta
length 1434 !csomaghossz bájtban kifejezve
delayed-start 0 !csomaggenerálási késleltetés másodpercben kifejezve
send 206 !átlagosan generált csomagok száma
fastethernet0/0 ios-dependent capture !bemenő fizikai interfész neve
```

!VIDEO forgalom generálása

```
fastethernet0/1
add tcp
datalink ios-dependent fastethernet0/1
l2-arp-for 172.16.10.2
l3-src 172.16.10.1
l3-dest 172.16.20.1
l3-tos 0x22
l4-src 4249
l4-dest 1720
name VIDEO
rate 50
length 1500
burst on ! börsztös forgalom generálása
burst duration off 1000 to 2000 !börsztösség beállítása
burst duration on 1000 to 3000 !börsztösség időtartama
delayed-start 0
send 333
fastethernet0/0 ios-dependent capture
```

!HANG forgalom generálása

```
fastethernet0/1
add udp
datalink ios-dependent fastethernet0/1
l2-arp-for 172.16.10.2
l3-src 172.16.10.1
l3-dest 172.16.20.1
l3-tos 0x2E
l4-src 44899
l4-dest 5060
name VOICE
rate 50
length 150
delayed-start 0
send 513
fastethernet0/0 ios-dependent capture
```

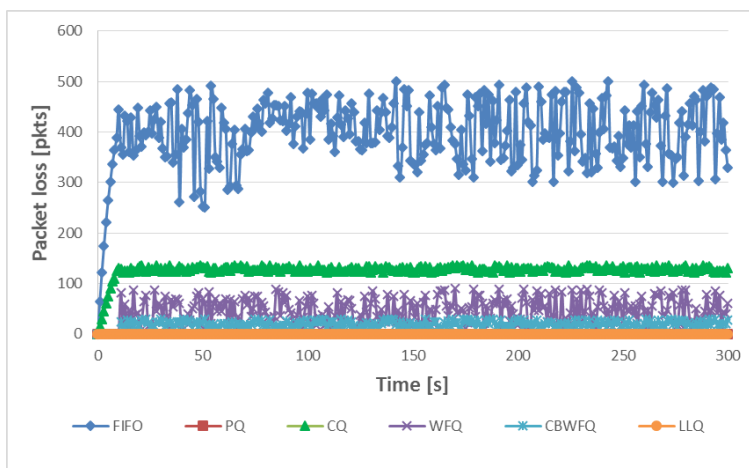
2.3.4. A torlódáskezelési algoritmusok implementálása

A mérési laborkörnyezet szűk keresztmetszetének hatására a forgalomirányítók között hálózati torlódás jött létre. Evégett, az R1 és R2 útválasztók soros interfészein (S0/1/1 és S0/1/0) beállítottam a különböző torlódáskezelési algoritmusokat (FIFO, PQ, CQ, WFQ, CBWFQ, LLQ), melyek kódjai megtalálhatóak a *C. Függelékben*.

2.3.5. Mérési eredmények

A korábbi tudományos munkáimhoz hasonlóan [18], [19], a mérési időtartam 5 percre tartott. A mérési eredményeket másodpercenként rögzítettem. A fenti forgalomgenerálási kódból látható, hogy másodpercenként átlagosan 513 hangcsomag lett generálva. A korábbi cikkekhez hasonlóan a következő területeket vizsgáltam: az eldobott hangcsomagok számát (csomagvesztési ráta), a torlódás következtében végponttól-végpontig terjedő hangcsomagok késleltetését, valamint a késleltetés ingadozást (dzsittert).

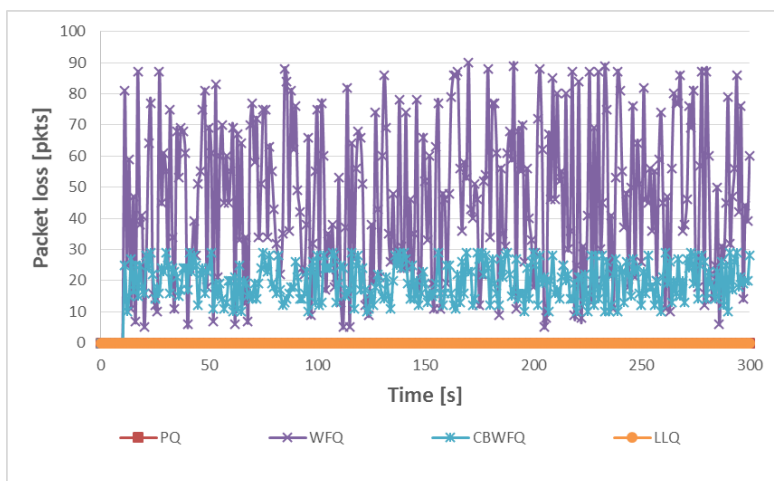
Az eldobott hangcsomagok száma tekintetében (29. ábra), az LLQ és a PQ algoritmusok bizonyultak a leghatékonyabbaknak, majd azt követően a CBWFQ. Megfigyelhető, hogy amíg a korábbi tudományos munkákban a szimulációs eredmények alapján a WFQ volt a leghatékonyabb torlódáskezelési algoritmus, addig a saját mérési tapasztalataim alapján a WFQ teljesítménye elmaradt az LLQ, PQ és a CBWFQ ütemezők teljesítménye mögött. A WFQ-nál a CQ nyújtott gyengébb teljesítményt, míg a legkevésbé hatékony algoritmus a FIFO volt.



29 ábra. Mérés: Az eldobott hangcsomagok száma

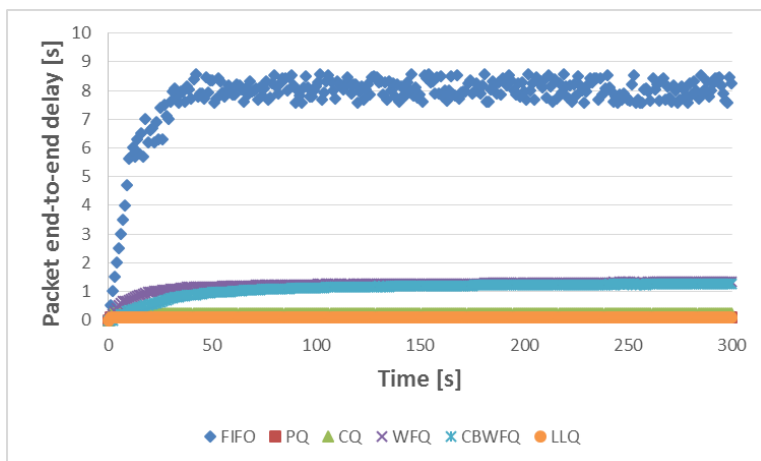
A 30. ábra tartalma azonos a 29. ábráéval, azzal a különbséggel, hogy az előbbi nem tartalmazza a két legkevésbé hatékony torlódáskezelési algoritmus által nyújtott hatékonyság ábrázolását. Így módon szembevetőbb a PQ, WFQ, CBWFQ, valamint az LLQ által a csomagvesztés terén nyújtott teljesítménykülönbség. Jól látható, hogy az LLQ és a PQ esetében nem volt csomagvesztés, az abszolút prioritásos sornak

köszönhetően, melybe a valósidejű hangforgalom lett besorolva. Ezt követően a CBWFQ teljesítménye volt a leghatékonyabb, majd legvégül a WFQ-é.

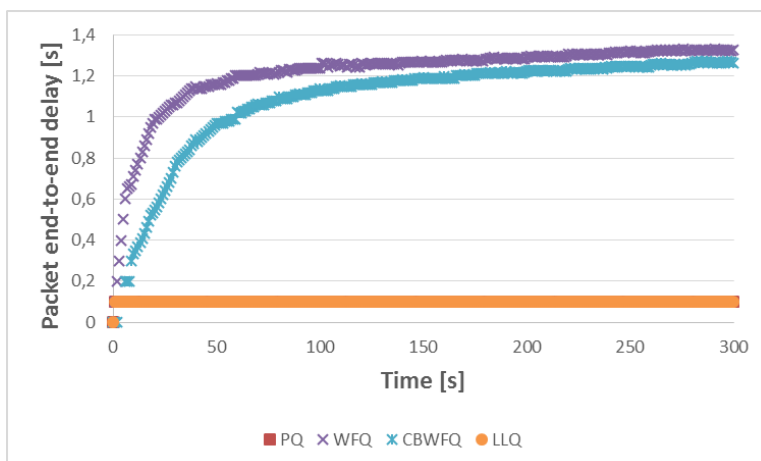


30. ábra. Mérés: PQ, WFQ, CBWFQ, LLQ: Az eldobott hangcsomagok száma

A hangcsomagok késleltetése tekintetében (31. és 32. ábrák), az LLQ és a PQ kiszolgálási elveknek sikerült 100 ms alá szorítani ezen értékeket, míg a CQ-nak 255 ms alá, ami már túllépi a QoS által előírt igényküszöböt [9]. Jól látható, hogy a WFQ és a CBWFQ esetében a késleltetés meghaladja egy kicsivel az 1 másodpercet, míg a FIFO-nál elérheti akár a 8.5 másodpercet is, mely értékek már elfogadhatatlanok a QoS által előírt igényeknek.

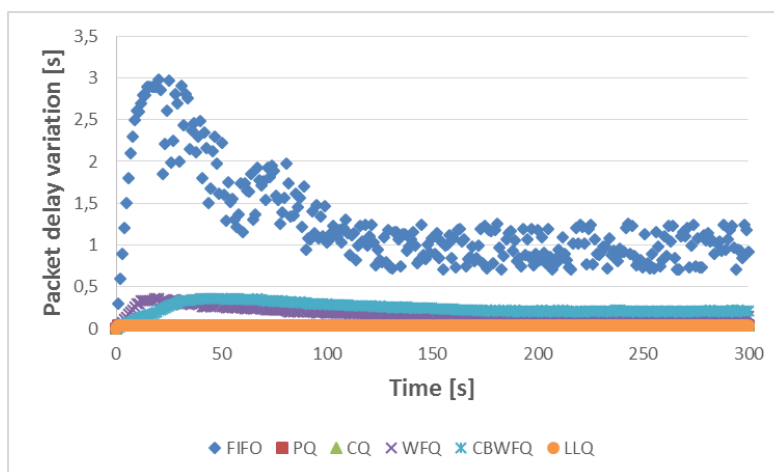


31. ábra. Mérés: A hangcsomagok késleltetése

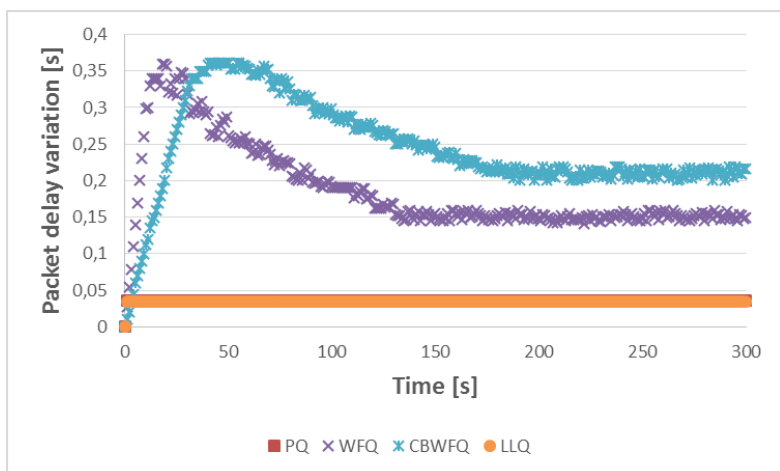


32. ábra. Mérés: PQ, WFQ, CBWFQ, LLQ: A hangcsomagok késleltetése

Ami a késleltetés ingadozást illeti (33. és 34. ábrák) az LLQ-nak, a PQ-nak és a CQ-nak sikerült 30 ms alatt tartani a mért értékeket. Ezt követően a WFQ, majd a CBWFQ biztosított 150 ms, illetve 210 ms körüli dzsittert, míg legvégül a FIFO-nak is sikerült stabilizálnia a késleltetés ingadozásának szórását 1 másodperc körül. Megjegyzendő, hogy dzsitter tekintetében a PQ és LLQ csomagütemezési diszciplínáknak sikerült a QoS által támasztott igényküszöb alatt teljesíteniük.



33. ábra. Mérés: A hangcsomagok dzsitterje



34. ábra. Mérés: A hangcsomagok dzsitterje

2.3.6. A mért eredmények összefoglalása

Minden esetben az volt tapasztalható, amint az egyébként várható is volt, hogy a FIFO ütemezési elv torlódás esetén a legalkalmatlanabb kiszolgálási elv a valós idejű hangcsomagok számára. A hangátvitel szempontjából az LLQ és a PQ volt a két legmegfelelőbb algoritmus, ami a csomageldobási rátát, a végponttól végpontig történő késleltetést és a dzsittert illeti. Ezt a két algoritmust alkalmazva egyetlen hangcsomag sem szenvedett csomageldobást. Azonban ismerve a PQ működési elvét, nevezetesen azt, hogy kiszolgálja ugyan maximális mértékig a legnagyobb prioritással rendelkező sort, viszont kiéhezteti a másik hármat, a szakirodalom alapján, illetve a mérési tapasztalataim alapján azt a következtetést vontam le, miszerint a valós idejű hangforgalom számára, minden szempontot figyelembe véve, az LLQ torlódáskezelési algoritmus a legmegfelelőbb.

3. A Több utas Kommunikációs Könyvtár (MPT) hatékonyságelemzése

3.1. Bevezetés, alapfogalmak

Napjainkban az Internet kommunikációs környezete csupán egyetlen adat utat tesz lehetővé az adat továbbítás számára egy adott kommunikációs viszonyon belül. Az egyetlen adatútvonal-megközelítés még elfogadható azon rendszerek esetében, melyek csupán egyetlen hálózati interfésszel rendelkeznek vagy egyetlen „kimeneti ponttal” kapcsolódnak az Internetre. Másfelől, a legtöbb napjainkban használt eszköz rendelkezik gyárilag beépített többes hálózati interfésszel, mint például RJ-45-ös csatlakozóval a vezetékes hálózat számára, rádiófrekvenciás interfésszel, amely a vezeték nélküli Wi-Fi hálózati csatlakozást teszi lehetővé, valamint mobil telefon adatátvitel csatlakozó interfésszel (pl. 3G, HSDPA vagy LTE).

Az egyetlen útvonalon alapuló kommunikációs technológia nem képes kihasználni az eszközök többinterfészes előnyeit. A kommunikációs teljesítmény (pl. adatátviteli teljesítmény) jelentősen növelhető, ha a hálózati környezetek egy adott kommunikációs viszonyon belül támogatják a többes adatútvonal használatát.

Az IETF RFC 6824 „TCP Extensions for Multipath Operation with Multiple Addresses – TCP Kiterjesztések a Több Címet használó Több utas Működés számára” (MPTCP, melyet 2013 januárjában publikáltak) [26], [27] napjainkban egy nagyon felkapott kutatási terület, amely a jelenlegi TCP implementációk kiterjesztésével foglalkozik a több utas kommunikáció támogatása céljából.

Ebben a fejezetben egy új architektúrát fogok bevezetni, amely egy nagyon könnyen használható kiterjesztést nyújt a jelenlegi TCP/IP protokollverem számára, lehetővé téve a kommunikációs végpontok közötti többes útvonalak használatát. Az új több utas környezetet egy MPT nevű szoftverkönyvtárban (és szoftver eszközben) implementáltuk. Az MPT¹⁰ modellezési háttere és architektúrája teljes mértékben különböző az MPTCP-jétől, ugyanis az MPT támogatást nyújt az alkalmazások számára úgy a TCP, mint az UDP szállítási protokollok felett, valamint a harmadik rétegben (Hálózati réteg) működik, míg az MPTCP a negyedik rétegben (Szállítási réteg).

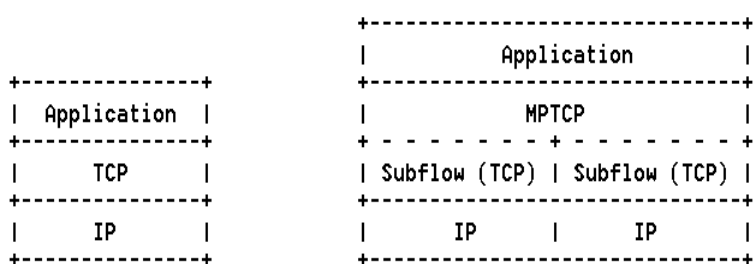
¹⁰ Az MPT szoftvert a Debreceni Egyetem Informatikai Karán fejlesztettük, melynek aktuális verziója, valamint dokumentációja a <http://irh.inf.unideb.hu/user/almasi/mpt/> oldalon érhető el.

Az MPT szoftver Linux rendszerben volt tervezve egy mérési laborkörnyezet létrehozásának céljából, amely segítségével az egy utas és a több utas kommunikáció teljesítményelemzését végeztük el. A mérési eredményeink azt a tényt támasztották alá, miszerint az MPT alapú több utas környezet képes hatékonyan összegezni az egyetlen útvonalon zajló kommunikációk adatátviteli teljesítményét.

3.1.1. Az MPT működési elve

A hagyományos IP kommunikációs infrastruktúra a kommunikációs végpontokon egyetlen IP cím használatára korlátozódik. Az IP címeket nemcsak a csomópontok (vagy egy bizonyos csomópont interfészeinek) azonosítására használják, hanem alkalmas a kommunikációs viszonyok beazonosítására is. Egy kommunikációs viszony különböző útvonalak közötti elosztása gyakorlatban egy nagyon érdekes téma, ami napjainkban egyik nagyon fontos kutatási területe.

A jelenlegi IP technológia az IP címeket és a TCP vagy UDP port-számokat használja a kommunikációs végpontok (socket) beazonosítására [28]. A hálózati forgalomirányítók ugyanezen IP címeket használják a csomagtovábbítási folyamatban az útvonalválasztás céljából. A hagyományos egy utas TCP/IP protokollkészlet nem képes megkülönböztetni a szoftveres végpontok azonosítására szolgáló IP címeket (socket ID) az útvonalválasztás céljára szolgáló címetől. A Több utas TCP architektúra (MPTCP) egy teljesen új rétegelt architektúrát használ (35. ábra) [26], [29].

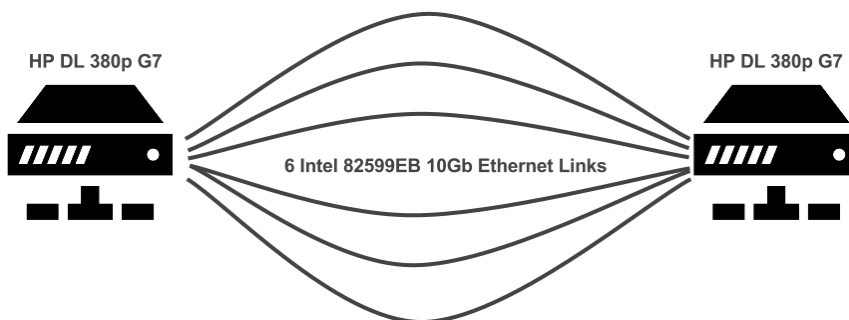


35. ábra. A TCP és az MPTCP protokollkészlet

Az új „MPTCP” alréteg feladata a kommunikációs interfész biztosítása az alkalmazások számára, míg a „TCP Subflow” alréteg, mely közvetlenül az MPTCP alréteg alatt helyezkedik el, a több utas kommunikációs környezet létrehozásáért

felel. A „Subflow” alrétegben használt TCP protokoll feladata a folyam ellenőrzés, valamint a megbízhatóság biztosítása [27].

A szakirodalomban elég sok olyan példát lehet találni, melyek az MPTCP átviteli teljesítményének hatékonyságát támasztják alá (ld. pl. [29], [30], [31]). Paasch és társai [32] egy olyan több utas mérési környezetet mutattak be, mely két végpont között 6 darab 10Gb-es linket tartalmazott, valamint az összegzett átviteli teljesítmény 50 Gb/s volt (ld. 36. ábra), ez az adott terület világszintű vezető teljesítményének számít.



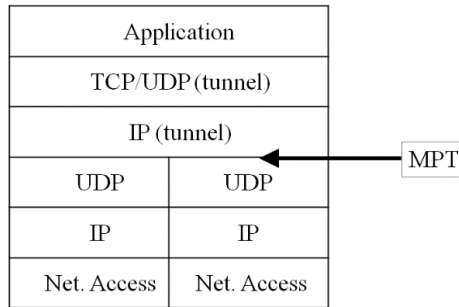
36. ábra. Az 50 Gb/s átviteli teljesítményre képes MPTCP környezet [32]

Az MPTCP számos előnye mellett rendelkezik egy néhány hátránnyal is:

- hatékony működés eléréséhez az Alkalmazási réteg vagy az operációs rendszer különböző hangolására lehet szükség
- a Szállítási rétegben működik
- csak a TCP szállítási protokollt támogatja, ami például a multimédiás alkalmazások használatakor jelenthet gondot

Ezen problémák készítették egy olyan több utas környezetet támogató szoftver fejlesztéséhez, amely kiküszöbölheti az előbb felsorolt hátrányokat.

A következőkben bemutatom azt a több utas kommunikációs környezetet, nevezetesen az MPT szoftver-könyvtárat, melyet az Debreceni Egyetem Informatikai Karán fejlesztettünk [33], [34]. Az MPT hálózati rendszer rétegtelt architektúrája a 37. ábrán látható [35].



37. ábra. Az MPT rétegelt architektúrája [35]

A klasszikus rétegelt architektúrához képest bevezettünk egy új logikai (tunnel) interfészt. Az IP (tunnel) alréteg feletti rész teljesen azonosan működik, mint a hagyományos környezet esetében, azzal a kivétellel, hogy az Alkalmazási rétegből érkező adategység nem egy fizikai, hanem egy logikai (tunnel) interfészre lesz kiküldve. A logikai interfész alatt lehetőség nyílik a kommunikációs viszony több fizikai interfészre való leképezésére. Gyakorlatilag e két rész között helyezkedik el az MPT szoftver, ami az egész működést hivatott vezérelni.

Az MPT eszköz működési elve az UDPTUN szoftver (ld. [36]) működési elvén alapszik. Az alapötlet a következő: a végpontokon létrehoztunk egy logikai (tunnel, alagút) interfészt, amely az alkalmazások által felhasználható a socketek azonosítására. Az MPT szoftver kiolvassa a küldő oldalon a tunnel interfészre érkező csomagot (IPv4 vagy IPv6 csomag). Ez a csomag egy új UDP szegmensbe lesz beágyazva, majd egy lehetséges útvonalon lesz kiküldve. Fogadó oldalon a beérkező UDP szegmens fejrésze el lesz távolítva, majd a beágyazott adat (amely gyakorlatilag a küldő fél tunnel interfészéről továbbított eredeti csomag) a fogadó hoszt tunnel interfészére lesz továbbítva. A kommunikációs felek tunnel interfészei közötti (logikai) kapcsolat egy közvetlen, pont-pont kapcsolat. Ebben az esetben egyáltalán nem szükséges az alkalmazás bármilyen módosítása, mivel a kommunikációs viszony egész ideje alatt az alkalmazás egyetlen logikai interfészt (tunnel interfészt) használ. Ugyanakkor, az alkalmazás használhatja az UDP szállítási protokollt is a tunnel interfész fölött, nincsen lekorlátozva a TCP protokoll használatára, mint az MPTCP esetében. Másrészt, ha a csomag-beágyazási folyamatban az MPT könyvtár az UDP protokollt használja, a tunnel interfész alatt nem lesznek újraküldési és folyamatvezérlő szolgáltatások biztosítva.

A továbbiakban az MPT alapú kommunikáció adategység-struktúráját mutatom be (ld. 38. ábra):

IP csomag v4/v6	UDP (Rögzített Portsztám)	Tunnel IP v4/v6	Tunnel TCP/UDP	Alkalmazás rétegbeli adat
--------------------	---------------------------------	--------------------	-------------------	------------------------------

38. ábra. Az MPT alapú kommunikáció PDU struktúrája

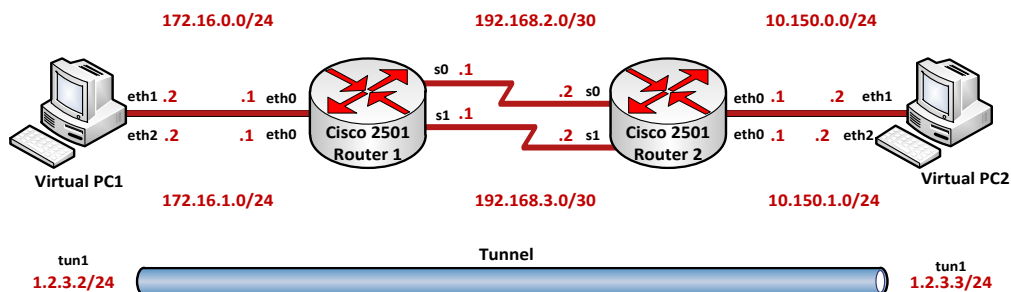
A 38. ábrán látható szürke rész az Alkalmazási rétegből érkezett csomagot jelöli, amely a logikai (tunnel) interfészhez kerül átadásra. A tunnel interfészre érkezett csomagot az MPT szoftver fogja kiolvasni, majd az egyik fizikai interfésznek megfelelő IP címmel fogja megcímezni az említett csomagot. Megjegyzendő, hogy az MPT könyvtárat dual stack-ben implementáltuk. A fizikai és a logikai IP réteg különállóságának köszönhetően az IP verziók (IPv4 vagy IPv6) tetszőleges használata lehetséges (pl. tunnel alatt IPv4, tunnel felett IPv6 vagy fordítva, vagy bármilyen lehetséges kombináció megengedett). A fogadó fél a *rögzített UDP portsztám*-on várakozik (figyel), majd az arra érkező csomagok fejrészeit leválasztva továbbítja a beágyazott adategységeket a logikai interfészre. A tunnel interfész fölött az Applikációs (Alkalmazási) réteg áll, ami le tudja kezelni a beérkező csomagokat.

Jóllehet, jelen disszertációmban csupán az MPT könyvtár adatátviteli teljesítményelemzésével foglalkozom, könnyen belátható ezen szoftver hasznossága más kommunikációs rendszerek esetében is, növelve ezek kommunikációs hatékonyságát (ld. pl. [30], [37], [38], [39], [40], [41]).

3.2. Az MPT teljesítményelemzése két utas környezetben

3.2.1. Mérési környezet és beállítások

Egy olyan mérési környezet létrehozása volt a cél, amely hűen tükrözi egy olyan két különböző hálózatban lévő csomópont között zajló kommunikációt, melyeket az Internet Felhő (Internet Cloud) választ el. A mérési környezet végső célja az MPT szoftver eszköz átviteli teljesítmény hatékonyságelemzésének lehetővé tétele.



39. ábra. A két utas IPv4-es mérési környezet

A mérésekhez a 39. ábrán látható hálózati topológiát használtam (ld. [42]), ami két csomópontból, két forgalomirányítóból, valamint a forgalomirányítók közötti két soros összeköttetésből állt. A PC-k 100 Mbps-os Ethernet kapcsolattal csatlakoztak a forgalomirányítókhoz, valamint a forgalomirányítók között két darab soros linket alakítottam ki. A fizikai Ethernet kapcsolat a hálózat közös része volt. Méréseimhez biztosítanom kellett, hogy a két soros kapcsolat sávszélessége kisebb legyen a végpontokat csatlakoztató Ethernet kapcsolatoknál (mivel a szűk keresztmetszet kialakulása nem engedhető meg a végpontoknál). A csomópontok linkjeit két logikai hálózatra bontottam, majd egy virtuális gép környezetet használtam a két számítógép végponton, két különböző hálózati interfész kialakítása céljából. A virtuális gépeket az Oracle Virtualbox 4.2.6 szoftver segítségével implementáltam. A virtuális gépek operációs rendszere a Linux OpenSuse 12.2 volt. A fizikai számítógépek, melyeken a virtuális gépeket hoztam létre, Intel Core i5 processzorokat (2.5GHz, 6MB cache) tartalmaztak, valamint 8GB memóriával voltak ellátva.

A laborkörnyezetet két darab Cisco 2501-es típusú forgalomirányító segítségével alakítottam ki, melyeken a Cisco IOS 12.2 operációs rendszer futott.

Mindkét virtuális gépen két darab fizikai (eth1 és eth2) és egy logikai interfész (tun1) lett felkonfigurálva. Mindkét gép számára alapértelmezett átjáróként a nekik megfelelő forgalomirányítók Ethernet portjai lettek beállítva, melyek két különböző IP címmel rendelkeztek, a két logikai hálózatnak megfelelően. Az útválasztók közötti két soros kapcsolat a végpontok közötti két különböző útvonalat ábrázolja. A két soros kapcsolat sávszélessége egymástól függetlenül volt beállítva, melyeket a forgalomirányítók DCE interfészén állítottam be a kívánt órajel megadásával. A két forgalomirányítón statikus útválasztást alkalmaztam, amely segítségével a végpontok között két különböző kommunikációs útvonalat hoztam létre (39. ábra). Ezáltal, a PC1-es csomópont eth1-es interfészéről érkező csomagok, melyek a PC2-es

hoszt eth1-es interfészének cél IP címét tartalmazták, a forgalomirányítók *serial 0* soros interfészei közötti kapcsolaton továbbítottak. Értelemszerűen, a két csomópont eth2-es interfészei közötti kommunikáció a forgalomirányítók közötti másik kapcsolaton zajlott. Az MPT szoftver által támogatott több utas kommunikáció a csomópontok tun1-es logikai interfészein lett kiépítve. A laborkörnyezet hálózati topológiája a 39. ábrán látható, valamint ennek részletes IP cím specifikációját a 6. táblázat tartalmazza.

A mérési környezetet a hálózatok átviteli teljesítményének tanulmányozására használtam, úgy szimmetrikus, mint aszimmetrikus kapcsolatok esetében. Szimmetrikus környezetben, a forgalomirányítók közötti mindkét soros összeköttetés sávszélessége ugyanazon értékre volt beállítva. Az órajel értékeinek beállítására két értéket használtam, 1.300.000 és 2.000.000 ciklusráta/másodperc értékeket, melyek gyakorlatilag a T1 és E1 bérelt vonalak kapcsolati sebességeit közelítik meg. A gyakorlati életben azonban két különböző kapcsolati útvonal sávszélessége jelentős mértékben különbözhet (aszimmetrikus útvonalak). Az aszimmetrikus útvonalakat a soros interfészek különböző átviteli rátáinak beállításával értem el: a serial 0 router interfészek közötti kapcsolat órajelét 2.000.000-ra állítottam, míg a serial 1-es interfészek közötti kapcsolati órajel az 1.300.000 ciklus/másodperc értéket vette fel.

Eszköz	Interfész	IP cím/prefix	Alapértelmezett átjáró
PC1	eth1	172.16.0.2/24	172.16.0.1
	eth2	172.16.1.2/24	172.16.1.1
	tun1	1.2.3.2/24	-
PC2	eth1	10.150.0.2/24	10.150.0.1
	eth2	10.150.1.2/24	10.150.1.1
	tun1	1.2.3.3/24	-
Router1	eth0	172.16.0.1/24	-
	eth0	172.16.1.1/24	-
	serial0	192.168.2.1/30	-
	serial1	192.168.3.1/30	-
Router2	eth0	10.150.0.1/24	-
	eth0	10.150.1.1/24	-
	serial0	192.168.2.2/30	-
	serial1	192.168.3.2/30	-

6. táblázat. IP címzési tábla a két utas IPv4-es mérési környezet számára

3.2.2. Mérési eredmények

Az általam kialakított hálózati rendszer teljesítményének elemzésére három típusú mérést végeztem el. Az első két esetben szimmetrikus útvonalakat használtam, 1.300.000 és 2.000.000 órajel/másodperc értékekkel. A harmadik típusú méréshez aszimmetrikus útvonalakat konfiguráltam fel, a serial 0 router interfészek közötti kapcsolat órajelét 2.000.000-ra állítottam be, míg a serial 1-es interfészek közötti kapcsolati órajelet 1.300.000-re. Minden egyes méréstípus három különböző esetet foglalt magába. Első esetben az FTP szerverről letöltött fájl mérete 10MB volt. A második és harmadik esetben 20, illetve 50MB-os fájlok kerültek letöltésre. A méréseket többször megismételtem, mindenegyes típusra és fájl méretre vonatkozóan. Elmondható, hogy a mérési eredmények állandóknak bizonyultak, minden esetben a mért átviteli teljesítmény-értékek közötti különbség 2% alatti volt.

Az FTP szervert a PC2-n állítottam be, melynek implementálásához az operációs rendszerbe beépített FTP szerverprogramot (vsftpd) használtam. A tesztfájlok letöltése céljából a PC1 beépített FTP kliensprogramját vettem igénybe. A mérési statisztikák kimutatására a *System Monitor* nevű alkalmazást használtam. A mérési eredményeket a 40. – 48. ábrák mutatják be.

Felhasználói szempontból nézve, az Alkalmazási rétegben mért átviteli teljesítmény sokkal érdekesebb a fizikai interfészekben mért teljesítménynél. Az Applikációs rétegben mért teljesítmény (pl. a továbbított adatmértetet elosztva az átviteli idővel) értékeket a 7. táblázat tartalmazza. Természetesen, a fizikai interfészekben mért átviteli teljesítmény egy kicsivel nagyobb az alkalmazási rétegben mérténél, a fizikai interfészen megjelenő fejrésztöbblet miatt.

Eset	Interfész	Idő (másodperc)	Az alkalmazási réteg teljesítménye (KiB/másodperc)
1. Eset: 10MB S0, S1:1.300.000	tun1	37	283.4
	eth1, eth2	72	145.6
2. Eset: 20MB S0, S1:1.300.000	tun1	74	283.4
	eth1, eth2	144	145.6
3. Eset: 50MB S0, S1:1.300.000	tun1	185	283.4
	eth1, eth2	362	144.8
4. Eset: 10MB S0, S1:2.000.000	tun1	23	455.9
	eth1, eth2	46	228.0
5. Eset: 20MB S0, S1:2.000.000	tun1	47	446.2
	eth1, eth2	92	228.0

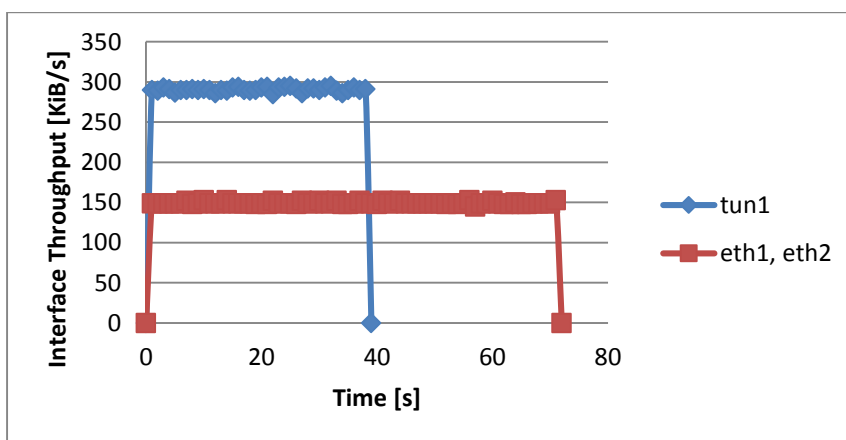
6. Eset: 50MB S0: 2.000.000 S1: 1.000.000	tun1	118	444.3
	eth1, eth2	231	227.0
7. Eset: 10MB S0: 2.000.000 S1: 1.000.000	tun1	31	338.3
	eth1	46	228.0
	eth2	89	117.8
8. Eset: 20MB S0: 2.000.000 S1: 1.000.000	tun1	62	338.3
	eth1	92	228.0
	eth2	179	117.2
9. Eset: 50MB S0: 2.000.000 S1: 1.000.000	tun1	157	333.9
	eth1	231	227.0
	eth2	449	116.8

7. táblázat. IPv4-es két utas mérési eredmények

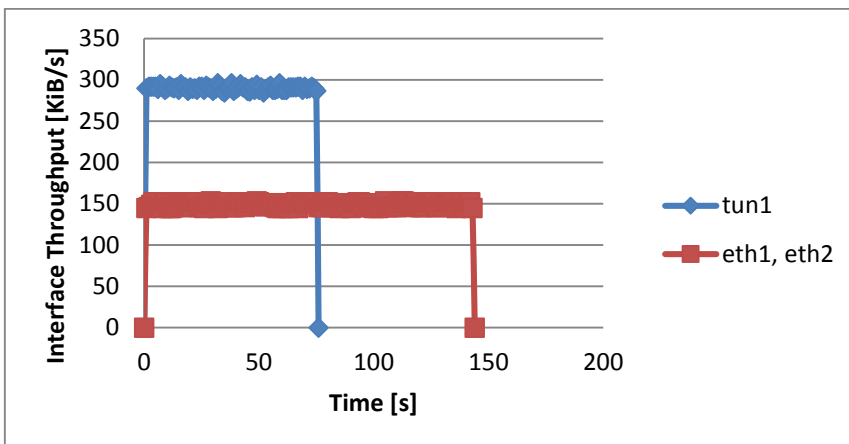
Egyszerű belátni, hogy az átviteli idő lineárisan növekszik a fizikai és a tunnel interfészen megjelenő adatmérettel. Pl. az átviteli teljesítmény nem függ össze a letöltött fájl mérettel. (A különbség minden esetben kevesebb, mint 3% volt.)

A mérési eredményekből jól látható, hogy a tunnel interfészen az MPT könyvtár hatékonyan összegezte a különböző útvonalak átviteli kapacitását. Ez a megállapítás igaz úgy az interfészeken mért teljesítményre, mint az alkalmazási réteg teljesítményére vonatkozóan is. Minden esetben a hatékonyság 90% feletti volt.

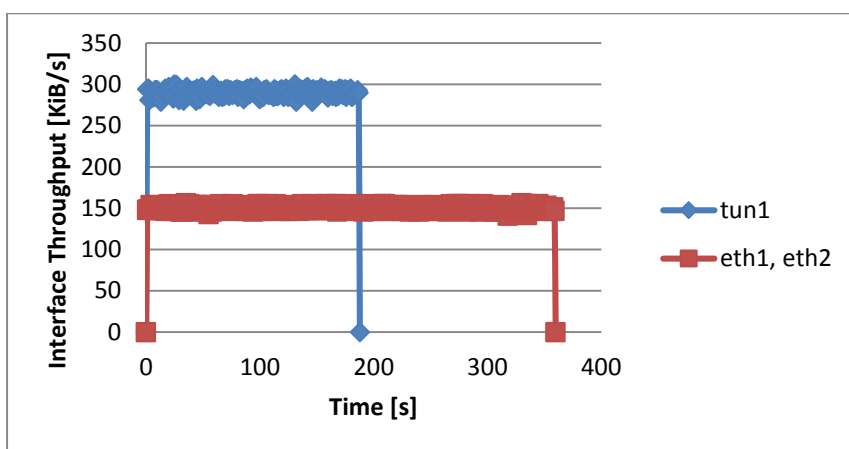
Ugyanakkor, az ábrák alapján jól látható, hogy aszimmetrikus esetben az interfész átviteli teljesítményvarianciája egy kicsivel nagyobb. Ezen jelenség tanulmányozása meghaladja jelen disszertáció célját.



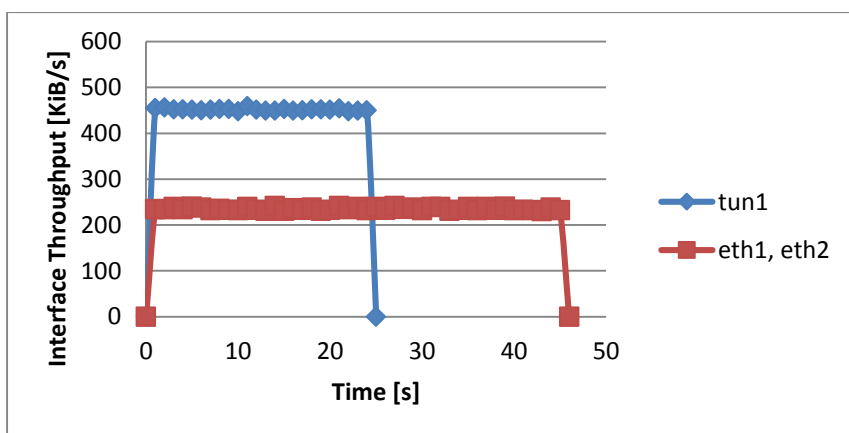
40. ábra. 1. Eset: Adatméret: 10 MB, Órajel: 1.300.000 / 1.300.000



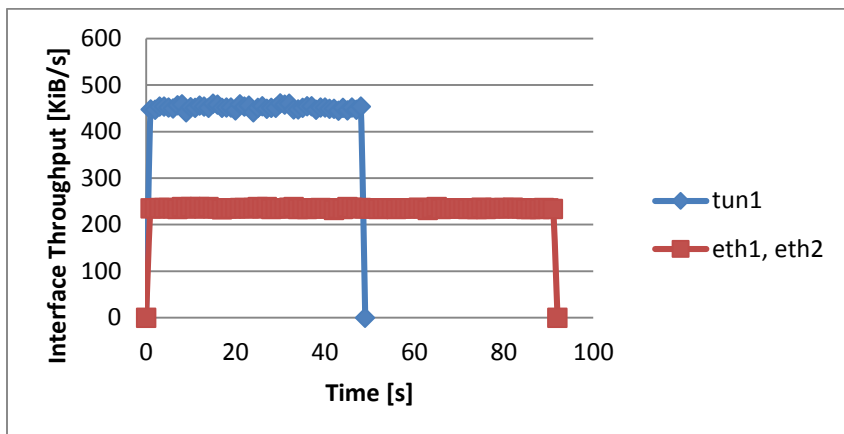
41. ábra. 2. Eset: Adatméret: 20 MB, Órajel: 1.300.000 / 1.300.000



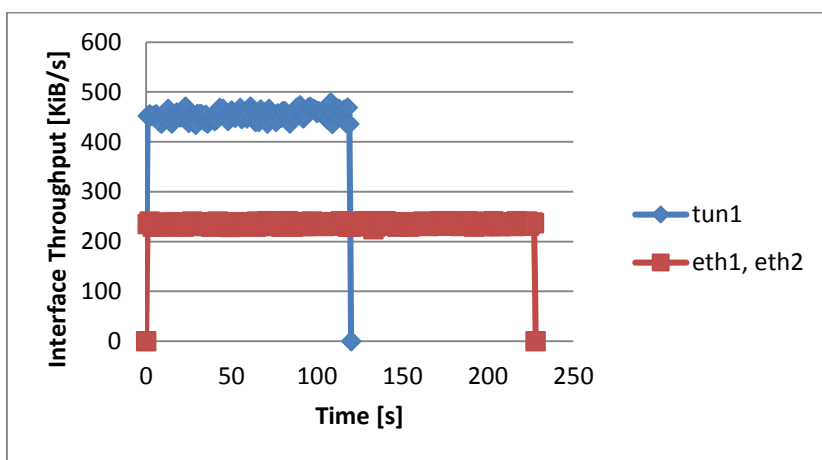
42. ábra. 3. Eset: Adatméret: 50 MB, Órajel: 1.300.000 / 1.300.000



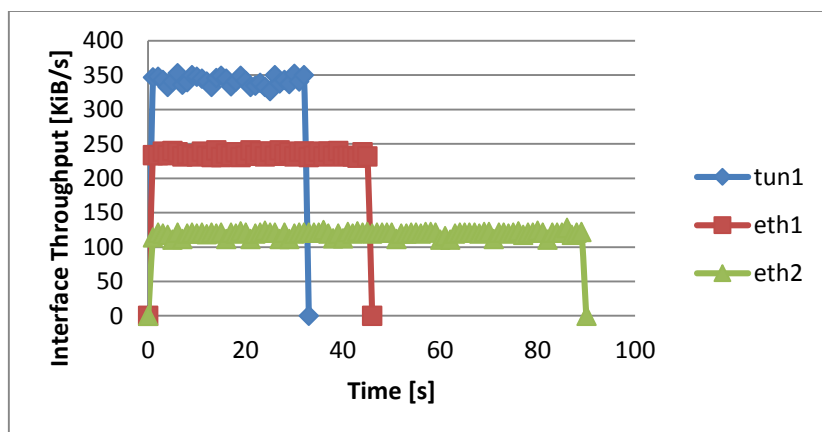
43. ábra. 4. Eset: Adatméret: 10 MB, Órajel: 2.000.000 / 2.000.000



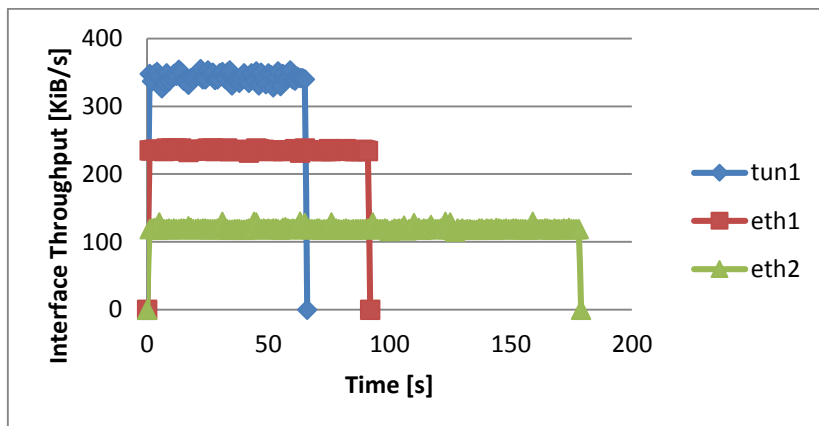
44. ábra. 5. Eset: Adatméret: 20 MB, Órajel: 2.000.000 / 2.000.000



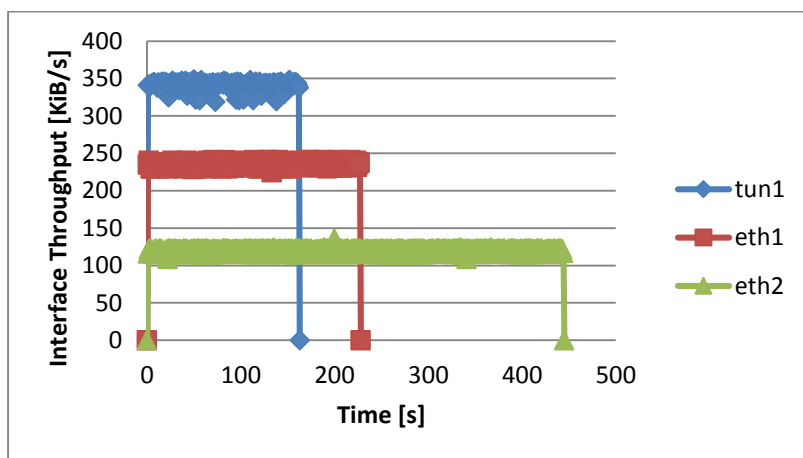
45. ábra. 6. Eset: Adatméret: 50 MB, Órajel: 2.000.000 / 2.000.000



46. ábra. 7. Eset: Adatméret: 10 MB, Órajel: 2.000.000 / 1.000.000



47. ábra. 8. Eset: Adatméret: 20 MB, Órajel: 2.000.000 / 1.000.000



48. ábra. 9. Eset: Adatméret: 50 MB, Órajel: 2.000.000 / 1.000.000

3.2.3. Összefoglalás

Ebben a fejezetben bevezettem az MPT szoftver-könyvtárat, mely lehetővé teszi a végpontok közötti több utas kommunikációt. Az MPT-eszköz fölött futó alkalmazások tetszőlegesen használhatják úgy a TCP, mint az UDP szállítási protokollokat. Jelen fejezet célkitűzése az MPT-eszköz hatékonyságelemzésének tanulmányozása volt. Ennek érdekében egy mérési környezetet hoztam létre, amely két különálló független útvonal-összeköttetést biztosított két kommunikációs végpont számára. Az átviteli teljesítményt szimmetrikus és aszimmetrikus sávszélességet alkalmazva és különböző továbbított adatméretet használva

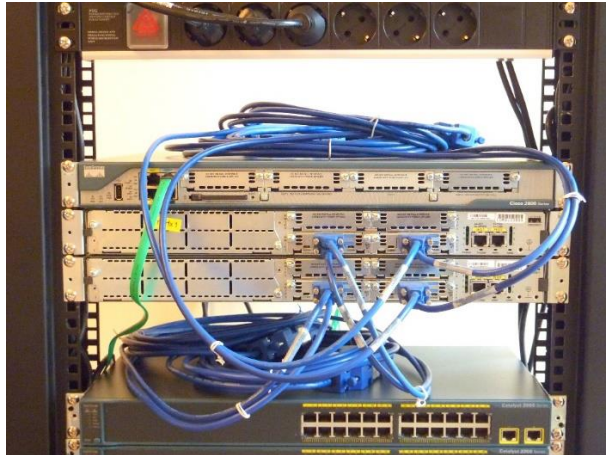
tanulmányoztam. Minden egyes mérési eredmény azt a tényt erősítette meg, hogy az MPT több utas környezet nagyon hatékonyan összegezi a fizikai útvonalak által nyújtott átviteli teljesítménykapacitást.

3.3. Az MPT szoftverkörnyezet használatával történő több utas FTP és adatfolyam-átvitel hatékonyságelemzése

3.3.1. A mérési környezet

Az MPT környezet hatékonyságvizsgálata érdekében létrehoztam egy dedikált mérési laborkörnyezetet, ami a végpontok között négy darab útvonalat tartalmazott (ld. 49. ábra) [43]. Különböző méréseket végeztem, különböző paramétereket használva. Mindazonáltal a várt eredmény beigazolódott: az MPT szoftverkönyvtár hatékonyan összegezte a négy útvonal sávszélességét, még a nem azonos sávszélességgel rendelkező utak esetében is. A sávszélességek aggregálása a multimédia adatfolyam átvitelre is érvényes, viszont ebben az esetben, ha az útvonalak sávszélessége különböző, csomag újrendezés (reordering) léphet fel.

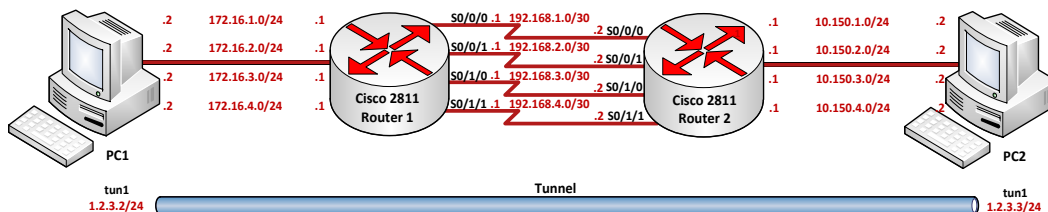
A mérési laborkörnyezet két számítógépet (8GB memóriával, Intel Core i5 2.5 GHz-es processzorral, 6MB cache-sel, Linux OpenSuse 12.0 operációs rendszerrel) tartalmazott. A PC-k négy hálózati interfésszel rendelkeztek (eth0, eth1, eth2 és eth3), melyek egymáshoz való csatlakoztatása négy különböző útvonal segítségével valósult meg. Az útvonalak kialakítására két darab Cisco 2811-es forgalomirányítót használtam. Az útválasztók egymáshoz való kapcsolása négy darab soros link segítségével történt, melynek következtében a routerek között négy különböző útvonal jött létre. A négy útvonal sávszélességének külön-külön való hangolására a DCE hálózati eszköz soros interfészein beállított órajelet használtam. A PC-k 100 Mbps-os Ethernet kapcsolattal csatlakoztak a forgalomirányítókhoz. A fizikai Ethernet kapcsolat a hálózat közös része volt. E közös kapcsolat magas sávszélessége (100 Mbps) garantálta azt a tényt, miszerint a hálózat szűk keresztmetszete ne ezen közös részen alakuljon ki. (A soros kapcsolatok maximális órajele 2.000.000 ciklus/másodpercre volt beállítva, melynek következtében a kommunikációs szűk keresztmetszet a forgalomirányítók közötti soros kapcsolatokon alakulhatott ki.). Statikus forgalomirányítást alkalmaztam úgy a PC-k, mint az útválasztók esetében.



a)



b)



c)

49. ábra. a), b), c) – A négy utas IPv4-es mérési környezet

3.3.2. Az FTP-átvitel mérési eredményei

Az hálózati rendszer teljesítményének elemzésére két típusú mérést végeztem el. Az első esetben szimmetrikus útvonalakat használtam, 1.000.000 és 2.000.000 órajel/másodperc értékekkel. A második típusú méréshez aszimmetrikus útvonalakat konfiguráltam. A routerek soros interfészei között lévő négy darab útvonal sávszélességét az 1.000.000 és 2.000.000 órajel-értékek összes lehetséges kombinációit alkalmazva állítottam be.

Minden egyes mérés-típus két különböző esetet foglalt magába, melyek csupán a továbbított fájl mérete tekintetében különböztek: első esetben az FTP szerverről letöltött fájl mérete 10MB volt, míg a második esetben 20MB-os fájlok kerültek letöltésre. A méréseket többször megismételtem, minden egyes mérési típusra és fájl méretre vonatkozóan. A mérési eredményeket másodpercenként rögzítettem. Elmondható, hogy a mérési eredmények állandóan azonosak voltak, minden esetben a mért átviteli teljesítmény-értékek közötti különbség 2% alatti volt.

Az FTP szervert a PC1-en implementáltam, melyhez az operációs rendszerbe beépített FTP szerver-programot (vsftpd) használtam. A tesztfájlok letöltésére a PC2 beépített FTP kliens programját vettem igénybe. A mérési statisztikák kimutatására a *System Monitor* alkalmazást használtam.

A mérési eredmények a 50. – 54. ábrákon láthatóak. Ezek az ábrák bemutatják az interfészek átviteli teljesítmény-értékeit, a 20MB-os letöltési fájlok használata esetében. A 10MB-os fájlok esetén is azonos eredmények születtek (ld. 8. táblázat).

Felhasználói szempontból nézve, az alkalmazási rétegben mért átviteli teljesítmény sokkal érdekesebb a fizikai interfészekben mért teljesítménynél. Az Alkalmazási rétegben mért teljesítmények (pl. a továbbított adatmennyiség elosztva az átviteli idővel) értékeit a 8. táblázat tartalmazza. Természetesen, a fizikai

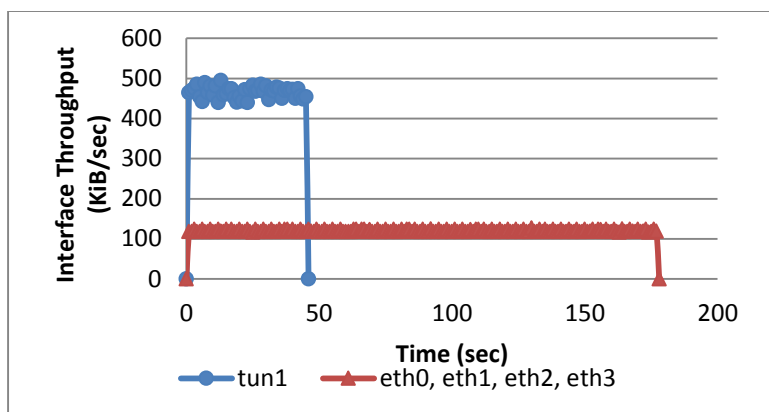
interfészeken mért átviteli teljesítmény egy kicsivel nagyobb az alkalmazási rétegben mértéknél, a fizikai interfészen megjelenő fejrésztöbbletnek köszönhetően.

Egyszerű belátni, hogy az átviteli idő lineárisan növekszik a fizikai és a tunnel interfészen megjelenő adatmérettel. Pl. az átviteli teljesítmény nem függ össze a letöltött fájl mérettel. (A különbség minden esetben kevesebb, mint 2% volt.)

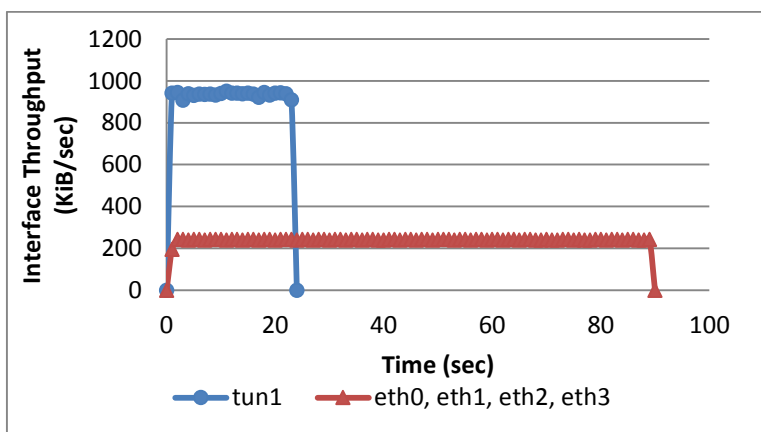
A mérési eredményekből jól látható, hogy a tunnel interfészen az MPT könyvtár hatékonyan összegezi a négy különböző útvonal átviteli kapacitását. Ez a tény megállapítható úgy az interfészeken mért teljesítményre vonatkozóan, mint az Alkalmazási réteg teljesítményének esetében is. Minden esetben a hatékonyság 96% feletti volt (8. táblázat).

Eset	Interfész	Idő (sec)	Teljesítmény (KiB/s)	Hatékonyság (%)
1. Eset: 10 MB 1 M / 1 M 1 M / 1 M	tun1	22	476,6	99,9
	eth0-3	88	119,2	
2. Eset: 20 MB 1 M / 1 M 1 M / 1 M	tun1	45	466,0	98,3
	eth0-3	177	118,5	
3. Eset: 10 MB 2 M / 2 M 2 M / 2 M	tun1	11	953,3	99,9
	eth0-3	44	238,3	
4. Eset: 20 MB 2 M / 2 M 2 M / 2 M	tun1	22	953,3	99,9
	eth0-3	88	238,3	
5. Eset: 10 MB 2 M / 1 M 1 M / 1 M	tun1	18	582,5	97,7
	eth0	44	238,3	
	eth1-3	88	119,2	
6. Eset: 20 MB 2 M / 1 M 1 M / 1 M	tun1	36	582,5	98,1
	eth0	88	238,3	
	eth1-3	177	118,5	
7. Eset: 10 MB 2 M / 2 M 1 M / 1 M	tun1	15	699,1	97,7
	eth0-1	44	238,3	
	eth2-3	88	119,2	
8. Eset: 20 MB 2 M / 2 M 1 M / 1 M	tun1	30	699,1	98,0
	eth0-1	88	238,3	
	eth2-3	177	118,5	
9. Eset: 10 MB 2 M / 2 M 2 M / 1 M	tun1	13	806,6	96,7
	eth0-2	44	238,3	
	eth3	88	119,2	
10. Eset: 20 MB 2 M / 2 M 2 M / 1 M	tun1	26	806,6	96,7
	eth0-2	88	238,3	
	eth3	177	118,5	

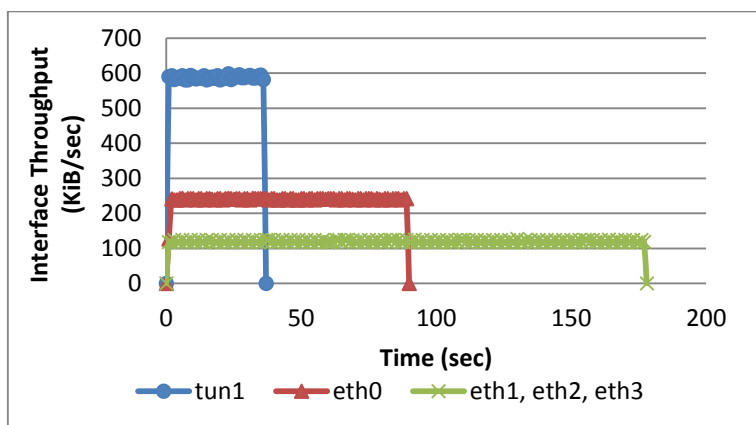
8. táblázat. Az FTP átvitel mérési eredményei négy utas IPv4-es környezetben



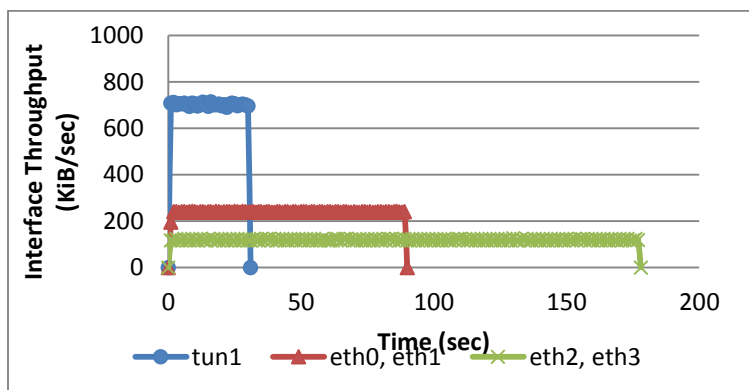
50. ábra. 2. Eset. Adatméret: 20MB, Órajel: 1M/1M/1M/1M



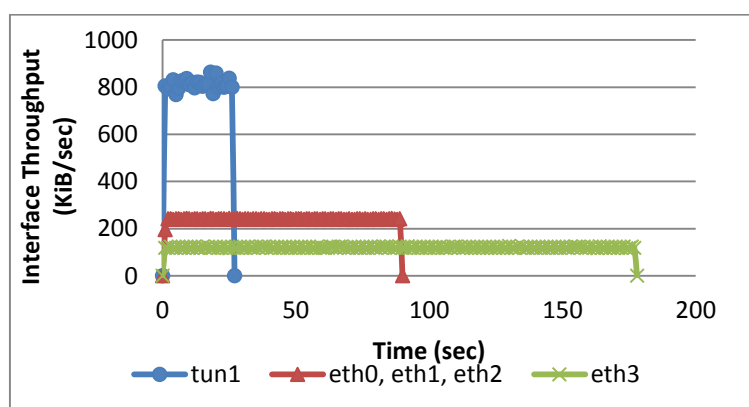
51. ábra. 4. Eset. Adatméret: 20 MB, Órajel: 2M/2M/2M/2M



52. ábra. 6. Eset. Adatméret: 20 MB, Órajel: 2M/1M/1M/1M



53. ábra. 8. Eset. Adatméret: 20 MB, Órajel: 2M/2M/1M/1M



54. ábra. 10. Eset. Adatméret: 20 MB, Órajel: 2M/2M/2M/1M

3.3.3. A multimédia adatfolyam átvitelének mérési eredményei

Az előző szakaszok alátámasztották azt a tényt, miszerint az MPT szoftverkönyvtár hatékonyan összegzi a négy különböző útvonal sávszélességét a TCP alapú fájlátvitel esetében. Az eredmények szintén bemutatták, hogy az összegzett útvonalak átviteli teljesítményvarianciája nagyobb az egyetlen útvonalon történő adatátvitellel szemben. Több utas környezetben a növekvő variancia csomagújrarendezéshez (packet reordering) vezethet [43]. Abban az esetben, ha a csomag sorrendezés egy kis ablakban történik az FTP átvitel ideje alatt, a TCP kijavítja ezt a problémát, az alkalmazás mit sem sejtve az egészeztől. Amennyiben UDP használunk a Szállítási rétegben (pl. a hang vagy video adatfolyamátvitel esetében), a rendszer viselkedése megváltozik: a csomagújrarendezési problémát az UDP nem kezeli, emiatt az

alkalmazás észlelni fogja azt, ezért találnia kell egy megoldást e jelenségre (pl. eldob néhány csomagot).

Teszt Eset	Útvonal szám	Órajel (1000 ciklus/másodperc)	Adatfolyam bitrátája (kbps)	Átlagosan mért teljesítmény (kbps)	Átlagos dzsitter (ms)	Csomagvesztési ráta (%)	Csomag újrendezés (%)
11. Eset	1	128	128	123,49	2,33	0	0
12. Eset	1	128	256	123,49	13,84	48,26	0
13. Eset	4	128-128-128-128	128	124,67	0,42	0	0
14. Eset	4	128-128-128-128	256	249,67	0,42	0	0
15. Eset	4	128-128-128-128	512	486,26	8,24	0	0
16. Eset	4	64-128-64-128	128	124,98	61,22	0	32,81
17. Eset	4	64-128-64-128	256	249,45	60,73	0	33,36

9. táblázat. Adatfolyam-átvitel mérési eredmények négy utas IPv4-es környezetben

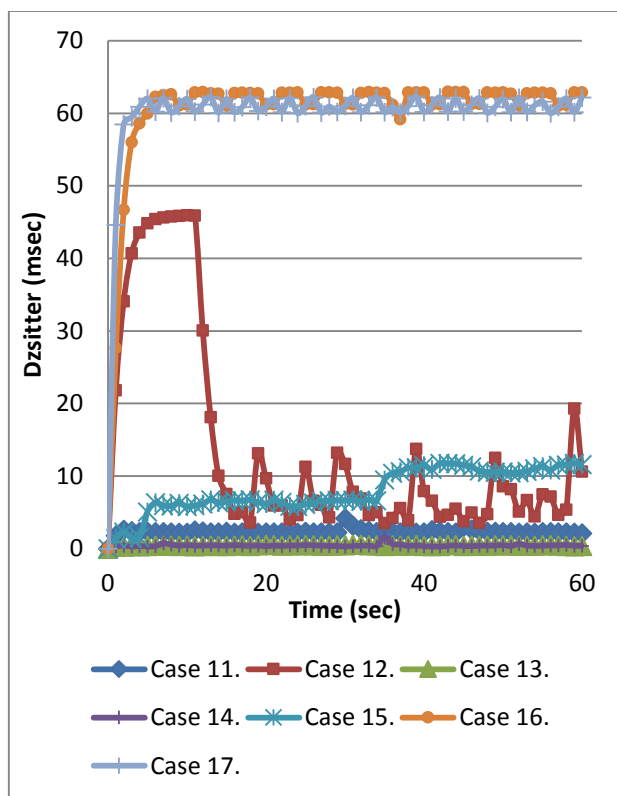
Ebben a fejezetben az MPT több utas környezet hatékonyságát vizsgáljuk meg, adatfolyam-átvitel esetében. Az adatfolyam kezdeményezéséhez az „iperf” forgalomgenerátort használtam. A legérdekesebb kérdés a több utas környezet viselkedésének tanulmányozása csomagtorlódás esetében. A hálózati torlódás kialakulása érdekében a forgalomirányítók közötti soros kapcsolatok órajeleit 64000 és 128000 ciklus/másodperc-re csökkentettem.

Az adatfolyam bitrátáinak értékei a következők voltak: 128, 256 és 512 kbps. Az adatfolyam hosszát 60 másodpercre állítottam. Az adatfolyam-átvitel mérési eredményeit a 9. táblázat tartalmazza.

Az egy utas átvitel esetében, melyben az órajel értéke 128000-re volt beállítva, a rendszer sikeresen továbbította 128 kbps bitrátával az adatfolyamot (nem volt csomagvesztés, sem csomag újrendezés, lásd a 11. esetet). Az adatfolyam bitrátájának 256 kbps értékre való növelésével nagymértékű torlódást idéztem elő: 10 másodperc eltelte után (a forgalomirányítók puffereinek túlszordulása következtében kialakult torlódás miatt) a csomagok 50%-a elveszett (lásd a 12. esetet).

Négy azonos órajellel (128000) beállított útvonal használata esetében, az MPT szoftver-környezet egy nagyon hatékony kommunikációs környezetnek bizonyult: A 128, illetve 256 kbps bitrátával rendelkező adatfolyamok továbbítása sikeres volt (nem volt csomagvesztés, sem csomag újrendezés). Továbbá a dzsitter értéke is nagyon alacsony maradt. Minden esetben kevesebb, mint 1 milliszekundum volt (lásd a 13. és 14. eseteket). Az adatfolyam bitrátájának 512 kbps-ra történő növelésével elértem az MPT több utas környezet (mely a szűk keresztmetszeten 4 darab 128000 ciklus/másodperc órajelű útvonalat tartalmazott) korláthatárát: a csomagátvitel

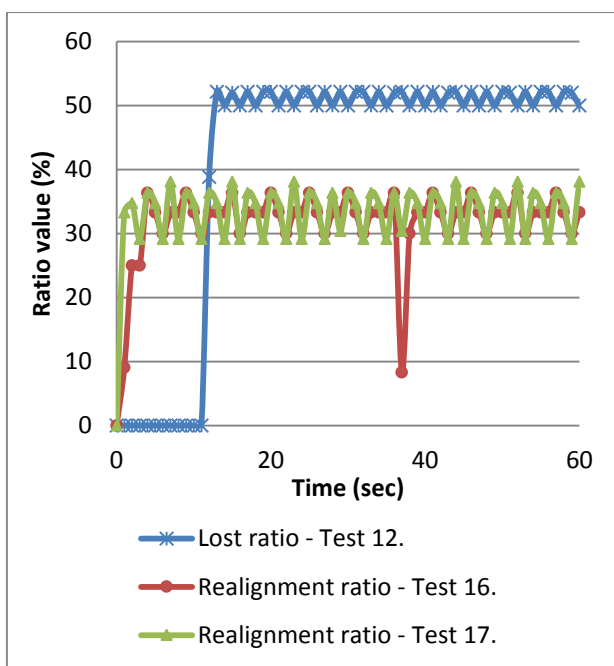
sikeresnek bizonyult (nem volt csomagvesztés, sem csomag újrendezés), viszont a dzsitter értéke 10 milliszekundumra növekedett, a 30 másodperc után kialakult torlódás következtében (ld. a 15. esetet).



55. ábra. 16. Eset. Multimédia adatfolyam dzsitter értékek négy utas IPv4-es környezetben

A „nem-egyenlő útvonalas” környezet (pl. 4 útvonal, melyeknek a szűk keresztmetszetén különböző átviteli teljesítmény mérhető) hatékonyságelemzése céljából, két útvonal órajelét 64000-re, míg a másik kettőét 128000-re állítottam. Mérési eredményeim alátámasztják a [42]-ben publikált egyszálas több utas kommunikációs szoftver eredményeket: a dzsitter és a variancia aszimmetrikus több utas környezetben növekvő tendenciát mutat. A legfontosabb probléma, ami az aszimmetrikus több utas környezetben történő adatfolyam-átvitelt érinti a következő: ha egy forrás állomás egy lassú kapcsolaton (melynek órajele 64000) küld egy csomagot, aztán hirtelen egy második csomagot is küld egy gyors útvonalon (melynek órajele 128000), ez utóbbi csomag korábban fog megérkezni a célállomásra,

mint az első csomag. Ebben az esetben biztosra vehető a csomag újrendezés megjelenése.



56. ábra. Multimédia adatfolyam – Csomagvesztési és csomag-újrendezési ráta négy utas IPv4-es környezetben

A küldendő csomagok a gyorsabb és a lassúbb útvonalak között lesznek szétosztva, az útvonalak órajel-értékeinek megfelelően (pl. a teszt esetemben, 66%-a a csomagoknak a gyors útvonalakra és 33%-a a csomagoknak a lassú útvonalakra került). Nyilvánvalóan, a lassú útvonalakon keresztül küldött csomagok nem lesznek sorrendtartóak. Ez az állítás igaz úgy a 128, mint a 256 kbps-os bitrátájú adatfolyam-átvitel esetén is (ld. a 16. és 17. esetet, valamint az 56. ábrát). Sikerként hatékonyan aggregálnom a különböző útvonalak átviteli teljesítményét az adatfolyam-átvitel esetében is. Mindazonáltal az aszimmetrikus esetben megjelenő csomagrendezés és a növekvő dzsitter nagymértékben befolyásolta a rendszer viselkedését.

3.3.4. Összefoglalás

Ebben a fejezetben az MPT több utas kommunikációs környezet teljesítményét elemeztem, figyelembe véve a fájl és multimédia adatfolyam-átvitelének lehetőségét. A mérések elvégzése céljából egy dedikált mérési környezetet hoztam létre.

A mérési eredmények azt a tényt bizonyították, miszerint az MPT környezet képes hatékonyan összegezni négy útvonal rendelkezésre álló kapacitását, még az eltérő útvonal kapacitások esetében is. Ez a megállapítás érvényes úgy a fájlátvitelre, mint a multimédia-átvitelre is.

Ami a multimédia adatfolyam-átvitel dzsitterjét illeti, azt tapasztaltam, hogy amennyiben az összegzendő útvonalak sebessége nagyjából azonos, a dzsitter értéke eléggé alacsony marad, viszont eltérő sebességű útvonalak esetén a dzsitter jelentősen megnő.

3.4. Az MPT több utas kommunikációs könyvtár átviteli teljesítményének hatékonyságelemzése IPv4 és IPv6 környezetben

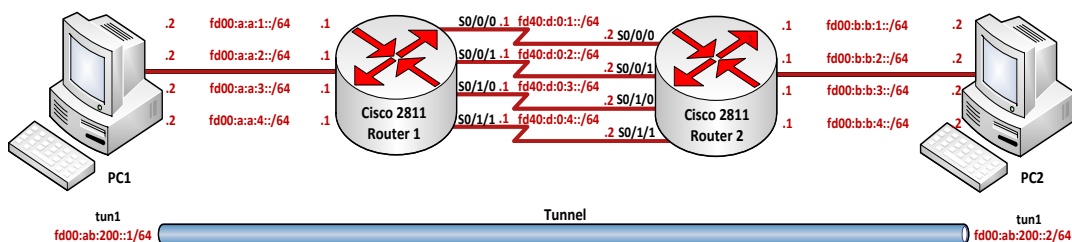
3.4.1. Mérési környezet

Az MPT több utas környezet rétegelt architektúrájából jól látható (37. ábra), hogy egy csomópont tunnel interfésze és fizikai interfésze különböző IP verziót használhat: az alkalmazás IPv4 címzést használhat a tunnel interfészen, míg a fizikai hálózati környezet IPv6-os címeket használhat a fizikai interfészekén. Az IP protokollváltás automatikusan történik az MPT szoftver által.

Az MPT könyvtár átviteli teljesítmény-összegzésének hatékonyságelemzése céljából egy négy utas mérési laborkörnyezetet alakítottam ki (ld. 49.a),b) és 55. ábrákat, valamint a D. Függelék), mely lehetővé tette úgy az IPv4, mint az IPv6 fölötti kommunikációt (ezáltal lehetőséget teremtve a kevert (mixed) protokollverziók tanulmányozására) [44].

A mérési laborkörnyezet két számítógépet (8GB memóriával, Intel Core i5 2.5 GHz-es processzorral, 6MB cache-sel, Linux Ubuntu 12.04 operációs rendszerrel) tartalmazott. A PC-k négy darab hálózati interfésszel rendelkeztek (eth0, eth1, eth2 és eth3), melyek egymáshoz való csatlakozása négy különböző útvonal segítségével valósult meg. Az útvonalak kialakítására két darab Cisco 2811-es típusú

forgalomirányítót használtam. Az útválasztók egymáshoz való csatlakoztatása négy soros link segítségével történt, melynek következtében a routerek között négy darab különálló útvonal jött létre. A PC-k 100 Mbps-os Ethernet kapcsolattal csatlakoztak a forgalomirányítókhoz. A fizikai Ethernet kapcsolat a hálózat közös része volt, viszont e közös rész nagy sávszélessége (100 Mbps) garantálta azt a tényt, miszerint a hálózat szűk keresztmetszete nem ezen a közös részen alakult ki. (A soros kapcsolatok maximális órajele 2.000.000 ciklus/másodpercre volt beállítva, melynek következtében a kommunikációs szűk keresztmetszet a forgalomirányítók közötti soros kapcsolatokon alakulhatott ki.). A négy útvonal sávszélességének külön-külön való hangolására a DCE hálózati eszköz soros interfészein beállított órajelet használtam. Statikus forgalomirányítást alkalmaztam úgy a PC-k, mint az útválasztók esetében. Az IP címzési sémát a 10. táblázat mutatja be.



57. ábra. A négy utas tiszta IPv6-os mérési laborkörnyezet

Eszköz neve	Interfész	IPv4/IPv6 cím/prefix	Alapértelmezett átjáró
PC1	eth0	172.16.1.2/24 fd00:a:a:1::2/64	172.16.1.1/24 fd00:a:a:1::1/64
	eth1	172.16.2.2/24 fd00:a:a:2::2/64	172.16.2.1/24 fd00:a:a:2::1/64
	eth2	172.16.3.2/24 fd00:a:a:3::2/64	172.16.3.1/24 fd00:a:a:3::1/64
	eth3	172.16.4.2/24 fd00:a:a:4::2/64	172.16.4.1/24 fd00:a:a:4::1/64
	tun1	1.2.3.2/24 fd00:ab:200::1/64	-
PC2	eth0	10.150.1.2/24 fd00:b:b:1::2/64	10.150.1.1/24 fd00:b:b:1::1/64
	eth1	10.150.2.2/24 fd00:b:b:2::2/64	10.150.2.1/24 fd00:b:b:2::1/64
	eth2	10.150.3.2/24 fd00:b:b:3::2/64	10.150.3.1/24 fd00:b:b:3::1/64
	eth3	10.150.4.2/24 fd00:b:b:4::2/64	10.150.4.1/24 fd00:b:b:4::1/64
	tun1	1.2.3.3/24 fd00:ab:200::1/64	-

R1	FE 0/0.1	172.16.1.1/24 fd00:a:a:1::1/64	-
	FE 0/0.2	172.16.2.1/24 fd00:a:a:2::1/64	-
	FE 0/0.3	172.16.3.1/24 fd00:a:a:3::1/64	-
	FE 0/0.4	172.16.4.1/24 fd00:a:a:4::1/64	-
	serial 0/0/0	192.168.1.1/30 fd40:d:0:1::1/64	-
	serial 0/0/1	192.168.2.1/30 fd40:d:0:2::1/64	-
	serial 0/1/0	192.168.3.1/30 fd40:d:0:3::1/64	-
	serial 0/1/1	192.168.4.1/30 fd40:d:0:4::1/64	-
R2	FE 0/0.1	10.150.1.1/24 fd00:b:b:1::1/64	-
	FE 0/0.2	10.150.2.1/24 fd00:b:b:2::1/64	-
	FE 0/0.3	10.150.3.1/24 fd00:b:b:3::1/64	-
	FE 0/0.4	10.150.4.1/24 fd00:b:b:4::1/64	-
	serial 0/0/0	192.168.1.2/30 fd40:d:0:1::2/64	-
	serial 0/0/1	192.168.2.2/30 fd40:d:0:2::2/64	-
	serial 0/1/0	192.168.3.2/30 fd40:d:0:3::2/64	-
	serial 0/1/1	192.168.4.2/30 fd40:d:0:4::2/64	-

10. táblázat. IPv4 és IPv6 címzés

3.4.2. Mérési eredmények – csak IPv4-es hálózat

A hálózati rendszer átviteli teljesítményének tesztelése érdekében, két típusú mérést hajtottam végre. Az első mérési típus szimmetrikus utakat használt, 1.000.000 és 2.000.000 ciklus/másodperc órajellel (ld. 11. táblázat, 5-10. esetek). A második típusú mérés aszimmetrikus útvonalakat használt: a soros interfészek közötti négy kapcsolat az órajel értékek (1.000.000 és 2.000.000 ciklus/másodperc) összes lehetséges kombinációját használta (ld. 11. táblázat, 5-10. esetek).

Eset	Interfész	Idő (másodperc)	Átviteli teljesítmény (KB/s)	Hatékonyság (%)
1. Eset: 10 MB 1 M / 1 M 1 M / 1 M	tun1	22	465,5	99,9
	eth0-3	88	116,4	
2. Eset: 20 MB 1 M / 1 M 1 M / 1 M	tun1	45	455,1	98,3
	eth0-3	177	115,7	
3. Eset: 10 MB 2 M / 2 M 2 M / 2 M	tun1	11	930,9	99,9
	eth0-3	44	232,8	
4. Eset: 20 MB 2 M / 2 M 2 M / 2 M	tun1	22	930,9	99,9
	eth0-3	88	232,8	
5. Eset: 10 MB 2 M / 1 M 1 M / 1 M	tun1	18	568,9	97,8
	eth0	44	232,8	
	eth1-3	88	116,4	
6. Eset: 20 MB 2 M / 1 M 1 M / 1 M	tun1	36	568,9	98,1
	eth0	88	232,8	
	eth1-3	177	115,7	
7. Eset: 10 MB 2 M / 2 M 1 M / 1 M	tun1	15	682,7	97,8
	eth0-1	44	232,8	
	eth2-3	88	116,4	
8. Eset: 20 MB 2 M / 2 M 1 M / 1 M	tun1	30	682,7	98,0
	eth0-1	88	232,8	
	eth2-3	177	115,7	
9. Eset: 10 MB 2 M / 2 M 2 M / 1 M	tun1	13	787,7	96,7
	eth0-2	44	232,8	
	eth3	88	116,4	
10. Eset: 20 MB 2 M / 2 M 2 M / 1 M	tun1	26	787,7	96,8
	eth0-2	88	232,8	
	eth3	177	115,7	

11. táblázat. FTP fájlátviteli eredmények – csak IPv4-es hálózat

Minden egyes mérés típus két különböző esetet foglalt magába, melyek csupán a továbbított fájl méret tekintetében különböztek: első esetben a továbbított fájl méret 10MB volt, míg a második esetben 20MB-os fájlok kerültek letöltésre. A méréseket többször megismételtem, minden egyes mérési típusra és fájl méretre vonatkozóan. A mérési eredmények állandóan azonosak voltak, minden esetben a mért átviteli teljesítmény-értékek közötti különbség 3% alatti volt.

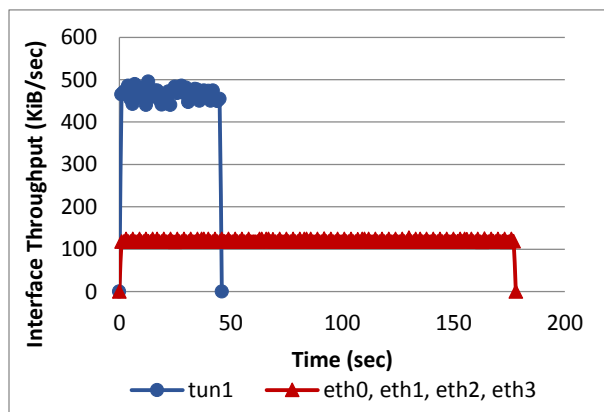
Az FTP szervert a PC1-en implementáltam, melyhez az operációs rendszerbe beépített FTP szerver programot (vsftpd) használtam. A tesztfájlok letöltésére a PC2 beépített FTP kliens programját vettem igénybe. A mérési statisztikák kimutatására a *System Monitor* alkalmazást használtam.

A részletes mérési eredmények megtekinthetők a 11. táblázatban, illetve az 58. – 65. ábrákon. Ezek az ábrák bemutatják az interfészek átviteli teljesítmény-értékeit, a 10, illetve 20MB-os letöltési fájlok használata esetében. Megfigyelhető, hogy aggregáció esetében az interfész átviteli teljesítményének varianciája enyhén növekszik.

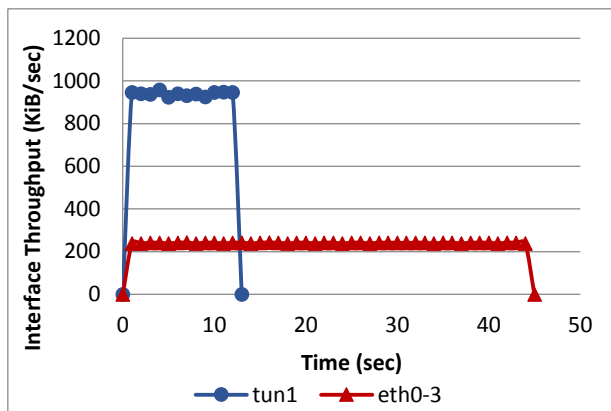
Azonos eredmények születtek mindkét fájlméret esetében. Egyszerű belátni, hogy az átviteli idő lineárisan növekszik a fizikai és a tunnel interfészen megjelenő adatmérettel. Pl. az átviteli teljesítmény nem függ össze a letöltött fájlmérettel. (A különbség minden esetben kevesebb, mint 3% volt.)

Felhasználói szempontból nézve, az Alkalmazási rétegben mért átviteli teljesítmény sokkal érdekesebb a fizikai interfészekben mért teljesítménynél. Az Applikációs rétegben mért teljesítmények (pl. a továbbított adatméretet elosztva az átviteli idővel, másodpercben mérve) értékeit a 11. táblázat tartalmazza (a továbbított adatméret 1024 bájt egységben van mérve). Természetesen, a fizikai interfészekben mért átviteli teljesítmény (58-65. ábrák) egy kicsivel magasabb az Alkalmazási rétegben mérténél, a fizikai interfészen megjelenő fejrész többletnek köszönhetően.

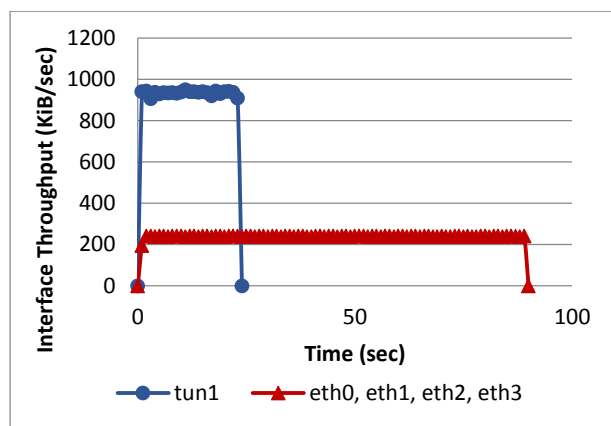
A mérési eredményekből jól látható, hogy a tunnel interfészen az MPT könyvtár hatékonyan összegezte a négy különböző útvonal átviteli kapacitását. Ez a tény igaz úgy az interfészekben mért teljesítményre, mint az Alkalmazási réteg teljesítményére is. Minden esetben a hatékonyság 96% feletti (11. táblázat).



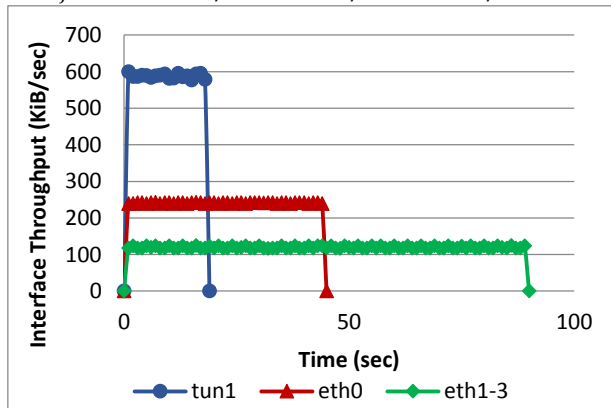
58. ábra. 2. IPv4 Teszt eset: Adatméret: 20 MB,
Órajel: 1.000.000 / 1.000.000 / 1.000.000 / 1.000.000



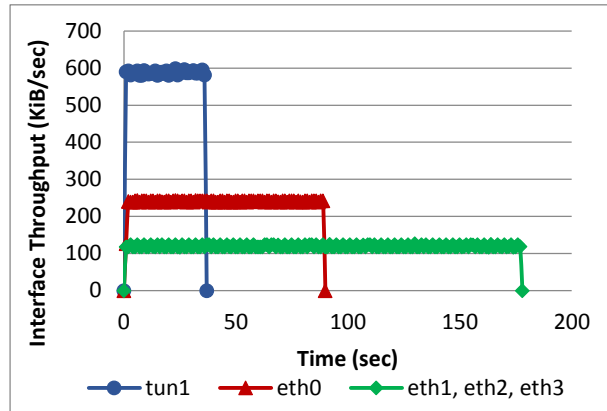
59. ábra. 3. IPv4 Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000



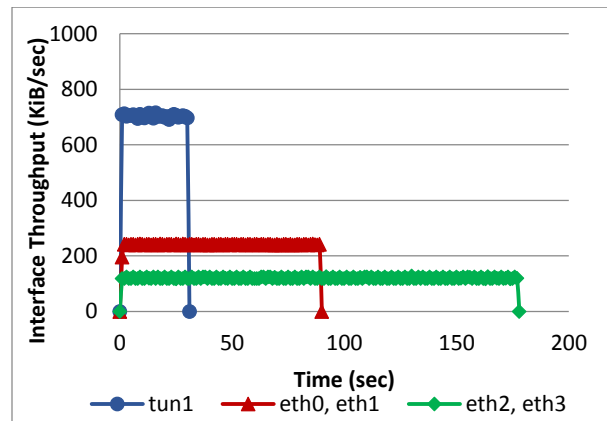
60. ábra. 4. IPv4 Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000



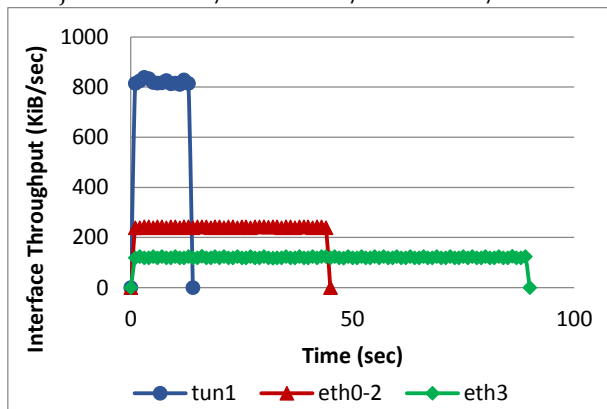
61. ábra. 5. IPv4 Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000



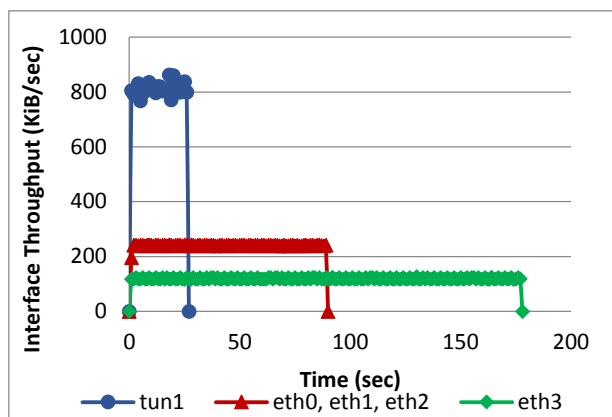
62. ábra. 6. IPv4 Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000



63. ábra. 8. IPv4 Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 1.000.000 / 1.000.000



64. ábra. 9. IPv4 Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000



65. ábra. 10. IPv4 Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000

Az ábrákon szintén jól látható, hogy az interfész átviteli teljesítményének varianciája aszimmetrikus esetekben egy kicsivel nagyobb.

3.4.3. Mérési eredmények – csak IPv6-os hálózat

Az IPv6 tunnel feletti és alatti használata megváltoztatja a mérési esetet: az újbóli IP beágyazás hosszabb IPv6-os fejrészt fog használni, ami miatt ez esetben hatékonyságcsökkenés lesz észlelhető.

Eset	Interfész	Idő (másodperc)	Átviteli teljesítmény (KB/s)	Hatékonyság (%)
1. Eset: 10 MB 1M / 1M 1M / 1M	tun1	23	445,2	97,8
	eth0-3	90	113,8	
2. Eset: 20 MB 1M / 1M 1M / 1M	tun1	47	435,7	95,7
	eth0-3	180	113,8	
3. Eset: 10 MB 2M / 2M 2M / 2M	tun1	12	853,3	93,7
	eth0-3	45	227,6	
4. Eset: 20 MB 2M / 2M 2M / 2M	tun1	23	890,4	97,8
	eth0-3	90	227,6	
5. Eset: 10 MB 2M / 1M 1M / 1M	tun1	19	538,9	94,7
	eth0	45	227,6	
	eth1-3	90	113,8	
6. Eset: 20 MB 2M / 1M 1M / 1M	tun1	38	538,9	94,7
	eth0	90	227,6	
	eth1-3	180	113,8	

7. Eset: 10 MB 2M / 2M 1M / 1M	tun1	16	640,0	93,7
	eth0-1	45	227,6	
	eth2-3	90	113,8	
8. Eset: 20 MB 2M / 2M 1M / 1M	tun1	32	640,0	93,7
	eth0-1	90	227,6	
	eth2-3	180	113,8	
9. Eset: 10 MB 2M / 2M 2M / 1M	tun1	14	731,4	91,8
	eth0-2	45	227,6	
	eth3	90	113,8	
10. Eset: 20 MB 2M / 2M 2M / 1M	tun1	27	758,5	95,2
	eth0-2	90	227,6	
	eth3	180	113,8	

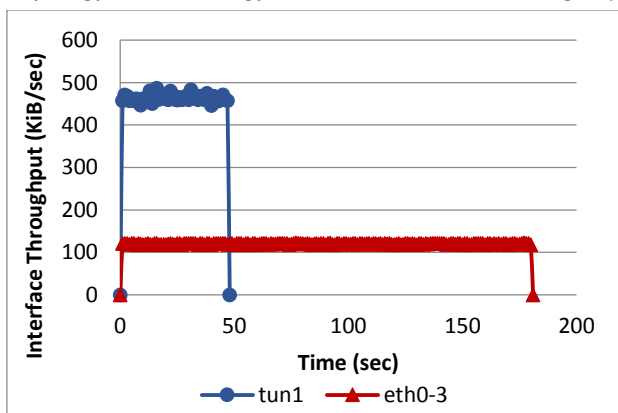
12. táblázat. FTP fájlátviteli eredmények – csak IPv6-os hálózat

A mérési környezet Fizikai és Adatkapcsolati réteg-tulajdonságai változatlanok maradtak az előző fejezetben bemutatottakhoz képest (pl. az órajelek, a fájlok méretei) (ld. D. Függelék). Egyedül a Hálózati réteg protokollja változott IPv4-ről IPv6-ra.

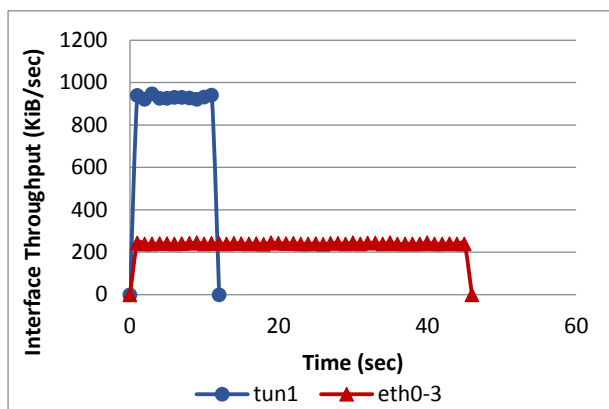
Az interfészek átviteli teljesítményének mérési eredményei a 66-73. ábrákon láthatóak. Az Alkalmazási rétegben mért átviteli teljesítmény numerikus értékeit a 12. táblázat foglalja össze (pl. felhasználói szemszögből nézve).

Jól látható, hogy az IPv6 átviteli teljesítmény-összegzésének hatékonysága egy kicsivel alacsonyabb, mint a korábban bemutatott IPv4 környezet esetében. Minden egyes hatékonyság-érték alacsonyabb, minden esetben a különbség 1-2% körüli. Amint az a 12. táblázatban is látható, minden egyes IPv6-ban mért hatékonysági szám meghaladja a 90%-ot, ezért megállapítható, hogy az aggregációs teljesítmény IPv6-os hálózati környezetben is megfelelő.

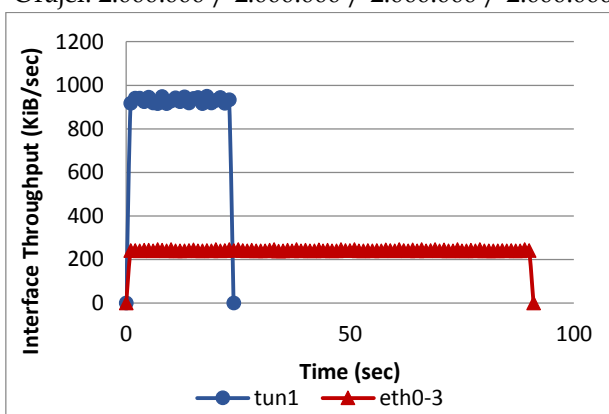
Hasonlóan az IPv4-es eredményekhez, az IPv6-os protokoll esetében is az interfész átviteli teljesítménye egy kicsivel nagyobb, az Alkalmazási réteg teljesítményénél.



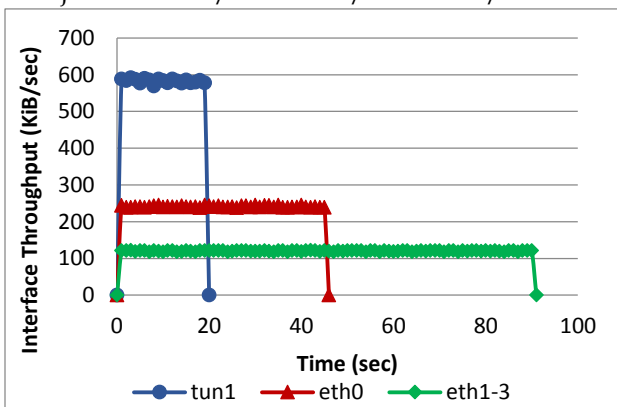
66. ábra. 2. IPv6 Teszt eset: Adatméret: 20 MB,
Órajel: 1.000.000 / 1.000.000 / 1.000.000 / 1.000.000



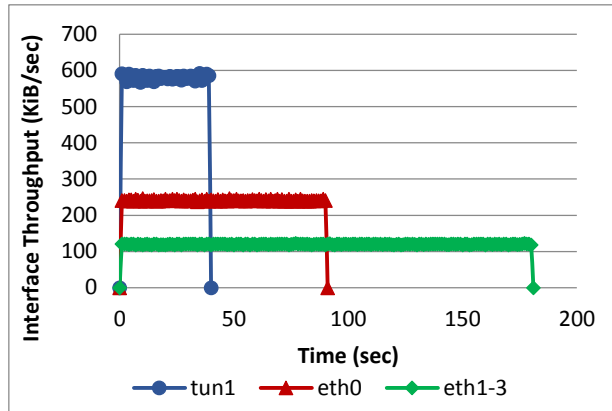
67. ábra. 3. IPv6 Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000



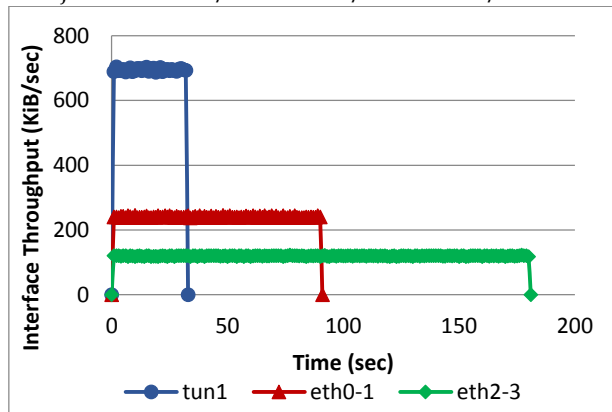
68. ábra. 4. IPv6 Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000



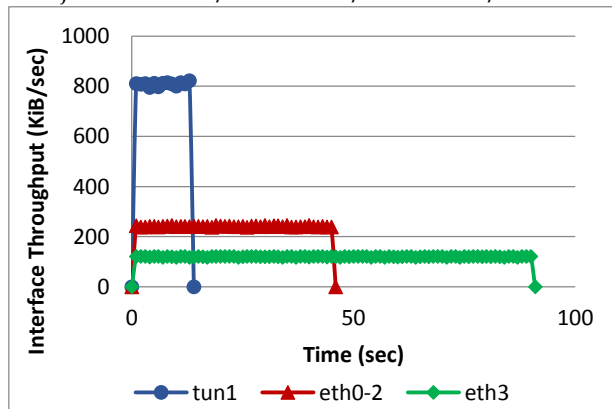
69. ábra. 5. IPv6 Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000



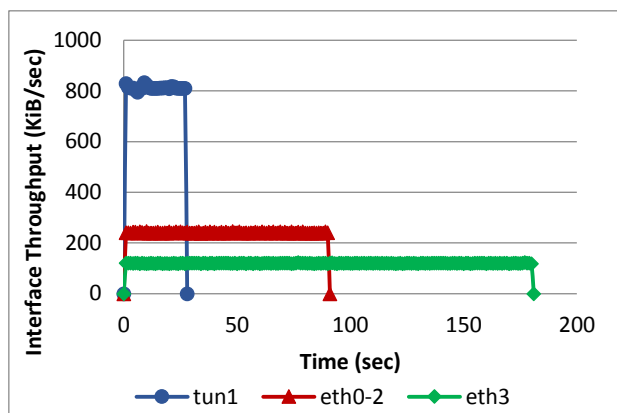
70. ábra. 6. IPv6 Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000



71. ábra. 8. IPv6 Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 1.000.000 / 1.000.000



72. ábra. 9. IPv6 Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000



73. ábra. 10. IPv6 Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000

3.4.4. Mérési eredmények – vegyes hálózat (IPv6 feletti IPv4)

Ebben a fejezetben egy speciális vegyes protokoll-környezetet fogok bemutatni: az alkalmazások a tunnel interfészen IPv4-et használtak, míg a fizikai interfészeken IPv6-ot implementáltam. Tudvalevő dolog, hogy az IPv4 címtartomány kifogyott 2012-ben. (Az IANA kiosztotta a RIR-eknek a központi készletből az utolsó /8 blokkokat. A RIR-ek közül a többségnek azóta már kimerült a készlete, így a legutolsó /8 blokkból oszt a megfelelő policy szerint (ld. [45], [46])). Ez a tény felgyorsíthatja az IPv6 használatát úgy a gerinchálózatokon, mint a végfelhasználói oldal esetében. A jövőben speciális hálózati környezetekben felmerülhet olyan speciális alkalmazások iránti igény, melyek nem támogatják az IPv6 használatát. Egy lehetséges megoldás ezen esetek kezelésére az MPT használata. Az MPT környezet lehetőséget teremt a logikai interfészeken történő IPv4-es protokoll használatára (pl. az alkalmazások IPv4-et használhatnak), míg a fizikai interfészek egy IPv6-os hálózathoz csatlakozhatnak.

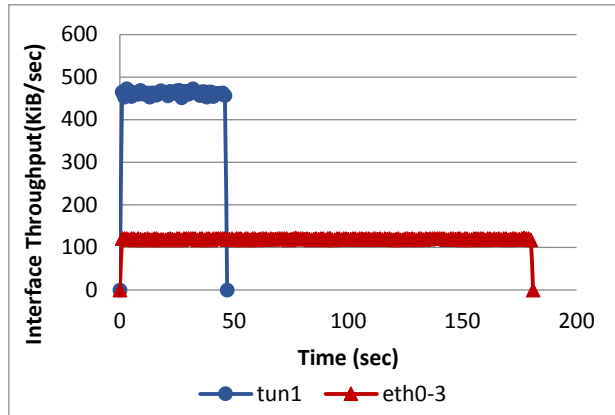
Ez a fejezet kiértékeli az MPT könyvtár teljesítményét vegyes protokoll-környezetben. A mérési környezet Fizikai és Adatkapcsolati réteg-tulajdonságai változatlanok maradtak az előző fejezetekben bemutatottakhoz képest (pl. az órajelek, a fájlok méretei) (ld. D. Függelék).

A mérési eredményeket a 74-81. ábrák mutatják be. Ezek az ábrák az interfészek átviteli teljesítményeinek értékeit ábrázolják azokban az esetekben, amelyekben a mérésekhez 10 és 20 MB-os fájl méreteket használtam. A 13. táblázat tartalmazza az Alkalmazási réteg átviteli teljesítmény-értékeit vegyes protokollkörnyezetben. Az eredmények csaknem azonosak a tiszta IPv4 és IPv6 hálózatok mérési eredményeivel:

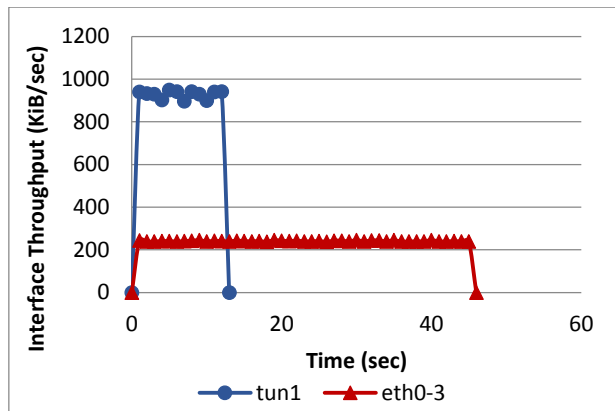
bizonyos esetekben az eredmények azonosak az IPv4 környezetével (ld. 8-9. esetek), máskor viszont a numerikus eredmények megegyeznek a tiszta IPv6 mérési eredményekkel (ld. 3-7. esetek). Tulajdonképpen ez azt jelenti, hogy a vegyes hálózati környezet numerikus eredményei gyakorlatilag a tiszta IPv4 és a tiszta IPv6 mérési eredményei közé illeszkednek (amint az egyébként várható is volt).

Eset	Interfész	Idő (másodperc)	Átviteli teljesítmény (KB/s)	Hatékonyság (%)
1. Eset: 10MB 1M / 1M 1M / 1M	tun1	23	445,2	97,8
	eth0-3	90	113,8	
2. Eset: 20MB 1M / 1M 1M / 1M	tun1	46	445,2	97,8
	eth0-3	180	113,8	
3. Eset: 10MB 2M / 2M 2M / 2M	tun1	12	853,3	93,7
	eth0-3	45	227,6	
4. Eset: 20MB 2M / 2M 2M / 2M	tun1	23	890,4	97,8
	eth0-3	90	227,6	
5. Eset: 10MB 2M / 1M 1M / 1M	tun1	19	538,9	94,7
	eth0	45	227,6	
	eth1-3	90	113,8	
6. Eset: 20MB 2M / 1M 1M / 1M	tun1	38	538,9	94,7
	eth0	90	227,6	
	eth1-3	180	113,8	
7. Eset: 10MB 2M / 2M 1M / 1M	tun1	16	640,0	93,7
	eth0-1	45	227,6	
	eth2-3	90	113,8	
8. Eset: 20MB 2M / 2M 1M / 1M	tun1	31	660,6	96,7
	eth0-1	90	227,6	
	eth2-3	180	113,8	
9. Eset: 10MB 2M / 2M 2M / 1M	tun1	13	787,7	98,8
	eth0-2	45	227,6	
	eth3	90	113,8	
10. Eset: 20MB 2M / 2M 2M / 1M	tun1	27	758,5	95,2
	eth0-2	90	227,6	
	eth3	180	113,8	

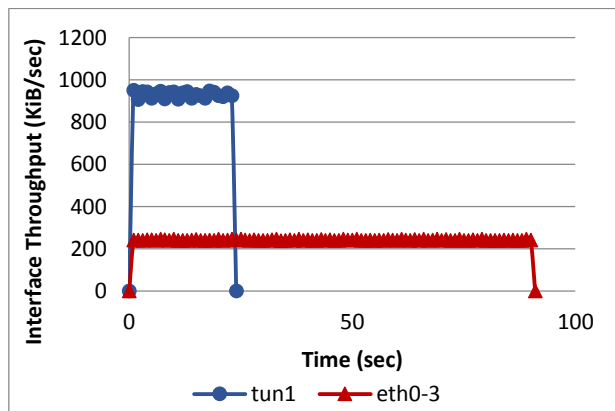
13. táblázat. Mérési eredmények – vegyes hálózat (IPv6 feletti IPv4)



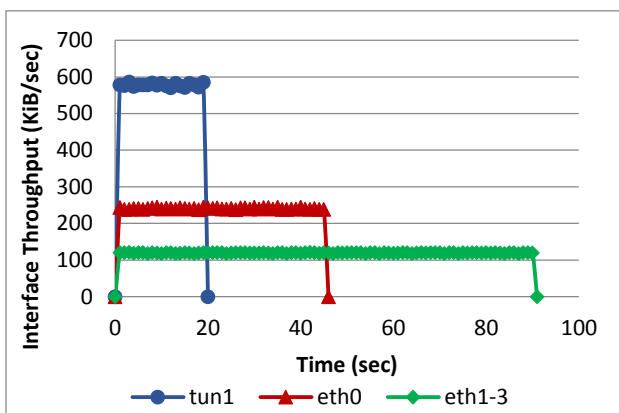
74. ábra. 2. Vegyes Teszt eset: Adatméret: 20 MB,
Órajel: 1.000.000 / 1.000.000 / 1.000.000 / 1.000.000



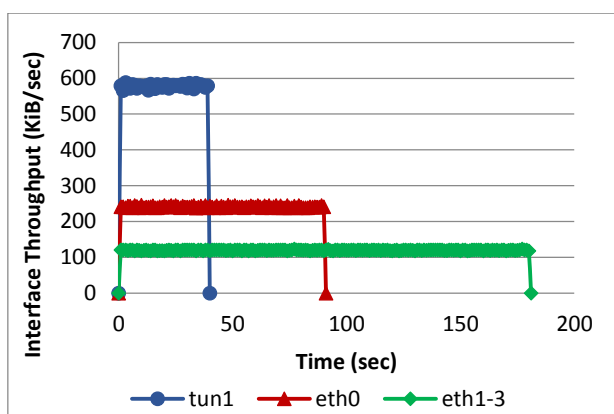
75. ábra. 3. Vegyes Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000



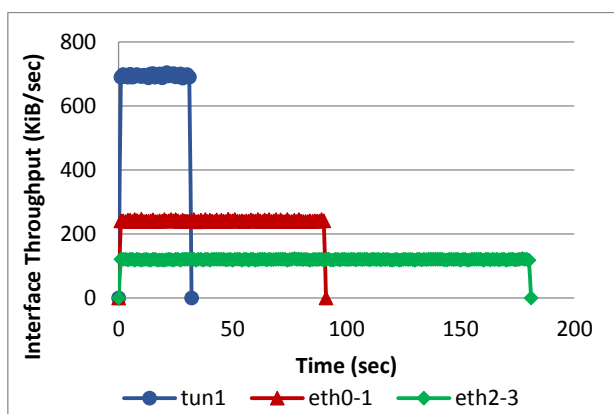
76. ábra. 4. Vegyes Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 2.000.000



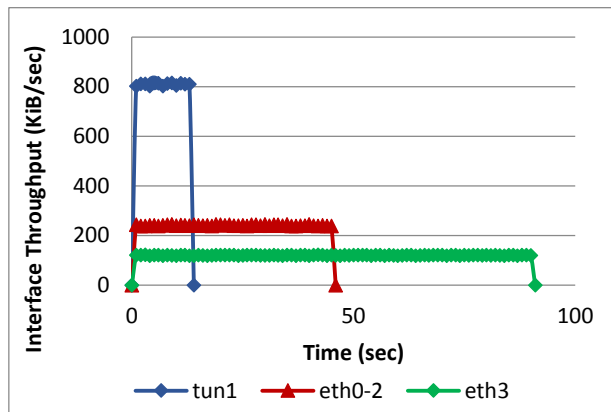
77. ábra. 5. Vegyes Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000



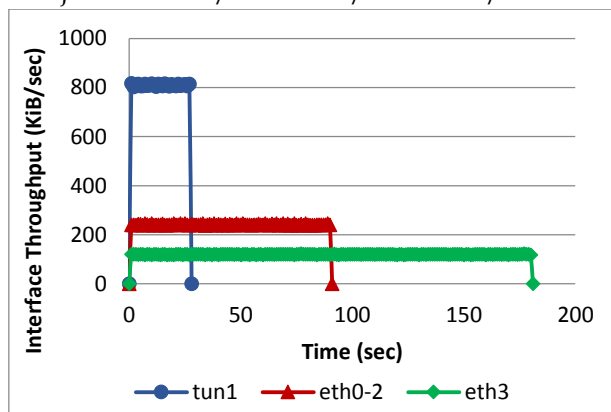
78. ábra. 6. Vegyes Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 1.000.000 / 1.000.000 / 1.000.000



79. ábra. 8. Vegyes Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 1.000.000 / 1.000.000



80. ábra. 9. Vegyes Teszt eset: Adatméret: 10 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000



81. ábra. 10. Vegyes Teszt eset: Adatméret: 20 MB,
Órajel: 2.000.000 / 2.000.000 / 2.000.000 / 1.000.000

3.4.5. Az eredmények összefoglalása

Ebben a fejezetben az MPT több utas szoftverkönyvtár átviteli teljesítmény összegzésének tulajdonságát vizsgáltam meg. Az MPT eszköz feletti alkalmazások tetszőlegesen használhatják a Szállítási réteg protokolljainak bármelyikét: úgy a TCP, mint az UDP használata engedélyezett. E fejezet célja az MPT eszköz átviteli teljesítményének hatékonyságvizsgálata volt, IPv4, IPv6 és vegyes hálózati környezetekben. Ennek érdekében létrehoztam egy mérési laborkörnyezetet, ami a kommunikációban résztvevő csomópontok számára négy darab független kapcsolati útvonalat biztosított. Az átviteli teljesítmény vizsgálata érdekében szimmetrikus és aszimmetrikus sávszélességi rátákat, valamint különböző átviteli adatméretek használtam. Annak ellenére, hogy bizonyos eredmények némi különbséget mutattak

különböző protokoll verziók esetében, a mérési eredmények azt mutatták, hogy az MPT több utas környezet hatékonyan összegezte a fizikai útvonalak átviteli kapacitását.

4. Összefoglalás

Ebben az értekezésben az infokommunikációs hálózatok hatékonyságelemzésével foglalkoztam, érintve két érdekes és egyben fontos kutatási területet, nevezetesen a forgalomirányítók torlódáskezelési algoritmusainak hatékonyságvizsgálatát, valamint a Debreceni Egyetem Informatikai Karán fejlesztett Több utas Kommunikációs Könyvtár (MPT) (ld. [35]) hatékonyságelemzését.

A második fejezetben felhívtam a figyelmet a forgalomirányítók torlódáskezelési algoritmusainak fontosságára és szükségességére, majd egy rövid áttekintést adtam az egyik legnagyobb hálózati eszközgyártó (Cisco) által a gyakorlatban leggyakrabban használt torlódáskezelési algoritmusok (FIFO, CQ, PQ, WFQ, CBWFQ, LLQ) működési elvéről. A korábbi tudományos munkákban (ld. [14], [15], [16]) tárgyalt algoritmusok mellett (FIFO, PQ, WFQ) további hármat sikerült megvizsgálnom (CQ, CBWFQ LLQ). A szimulációs környezetet az OPNET IT Guru Academic Edition¹¹ matematikai modelleken alapuló szimulációs szoftver biztosította. Szimulációimhoz a korábbi tudományos munkákhoz viszonyítva egy általánosabb, bővíthetőbb és valósághűbb hálózati topológiát használtam. A szimulációs tapasztalataim alapján azt a következtetést vontam le, miszerint a hangátvitel számára az LLQ egy hatékonyabb torlódáskezelési algoritmus a WFQ-val szemben.

Ezt követően disszertációm a korábban említett 6 darab torlódáskezelési algoritmus hatékonyságvizsgálatával folytattam, melynek alapjául egy valós fizikai mérési laborkörnyezetet használtam. Minden esetben az volt tapasztalható, amint az egyébként várható is volt, hogy a FIFO ütemezési elv torlódás esetén a legalkalmatlanabb kiszolgálási elv a valósidejű hangcsomagok számára. A hangátvitel szempontjából az LLQ és a PQ volt a két legmegfelelőbb algoritmus, ami a csomageldobási rátát, a végponttól-végpontig történő késleltetést és a dzsittert illeti. Ezt a két algoritmust alkalmazva egyetlen hangcsomag sem szenvedett csomageldobást. Azonban ismerve a PQ működési elvét, nevezetesen azt, hogy kiszolgálja ugyan maximális mértékig a legnagyobb prioritással rendelkező sort, míg a többi kiéhezteti, a szakirodalom alapján, illetve a mérési tapasztalataim alapján azt

¹¹ Jelen dolgozatban említett OPNET szimulációs szoftvert a korábbi tudományos munkáimban használtam. 2012. október 29.-én az OPNET-et felvásárolta a Riverbed cég, majd 2014 tavaszától az OPNET szimulációs szoftvert felváltotta a Riverbed Modeler [47].

a következtetést vontam le, miszerint a valósidejű hangforgalom számára, minden szempontot figyelembe véve, az LLQ torlódáskezelési algoritmus a legmegfelelőbb.

A 3. fejezetben a Debreceni Egyetem Informatikai Karán fejlesztett Több utas Kommunikációs Könyvtár (MPT) hatékonyságelemzésével foglalkoztam. Rámutattam a napjainkban használt Internet kommunikációs környezet egyik hiányosságára, valamint azokra az előnyökre és lehetőségekre, melyek a korszerű eszközökben rejlenek, mint például a többes adatútvonal támogatásából fakadó kommunikációs teljesítmény növelhetősége.

Ezt követően egy rövid áttekintést adtam az IETF RFC 6824-ben publikált MPTCP működési elvéről, valamint bemutattam egy példát, amely az MPTCP átviteli teljesítményének hatékonyságát támasztotta alá. Majd felhívtam a figyelmet az MPTCP számos előnye mellett megjelenő egy néhány hátrányára is:

- egy hatékony működés eléréséhez az Alkalmazási réteg vagy az operációs rendszer különböző hangolására lehet szükség
- a Szállítási rétegben működik
- csak a TCP szállítási protokollt támogatja, ami például a multimédiás alkalmazások használatakor jelenthet gondot

Bevezettem egy teljesen új architektúrát, melyen a Debreceni Egyetem Informatikai Karán fejlesztett MPT szoftver működési elve alapszik, majd ismertettem ennek működési elvét. A MPT szoftver Linux rendszerben volt tervezve egy mérési laborkörnyezet létrehozásának céljából, amely segítségével az egy utas és a több utas kommunikáció teljesítményelemzését végeztük el.

Megvizsgáltam az MPT szoftver eszköz hatékonyságát két utas-, négy utas-, valamint IPv4-es, IPv6-os, illetve vegyes (mixed) környezetben. Az átviteli teljesítményt szimmetrikus és aszimmetrikus sávszélességet alkalmazva és különböző továbbított adatméretet használva tanulmányoztam.

A mérési eredmények azt a tényt bizonyították, miszerint az MPT környezet képes hatékonyan összegezni több útvonal rendelkezésre álló kapacitását, még az eltérő útvonal kapacitások esetében is. Ez a megállapítás érvényes úgy a fájlvitelre, mint a multimédia-átvitelre is.

Ami a multimédia adatfolyam-átvitel dzsitterjét illeti, azt tapasztaltam, hogy amennyiben az összegzendő útvonalak sebessége nagyjából azonos, a dzsitter értéke eléggé alacsony marad, viszont eltérő sebességű útvonalak esetén a dzsitter jelentősen megnő.

Az MPT eszköz átviteli teljesítménye IPv4, IPv6, illetve vegyes környezetben hatékornak bizonyult. Annak ellenére, hogy bizonyos eredmények némi különbséget mutattak különböző protokoll verziók esetében, a mérési tapasztalatok alapján elmondható, hogy az MPT több utas környezet hatékornan összegzi a fizikai útvonalak átviteli kapacitását.

5. Summary

The present dissertation deals with the performance analysis of infocommunication networks, touching two interesting and important research areas, namely the efficiency of congestion management algorithms, and the performance analysis of Multipath Communication Library (MPT) developed by the Faculty of Informatics, University of Debrecen.

The second chapter emphasizes the importance and usefulness of the routers' congestion management algorithms, presenting a short summary of the congestion management algorithms (FIFO, CQ, PQ, WFQ, CBWFQ, LLQ) which are most frequently used by one of the main routers manufacturers, Cisco. In addition to the previous works, where was examined FIFO, PQ, WFQ, this work deals with another three algorithms (CQ, CBWFQ, LLQ). The simulation environment is ensured by OPNET IT Guru Academic Edition¹², which is a tool based on mathematical models. The network topology of the simulation is a more general, scalable and realistic one compared to the previous works. According to my simulation results, LLQ is a more efficient congestion management algorithm for voice transfer compared to WFQ.

The dissertation continues with the performance analysis of the above mentioned six congestion management algorithms, where I used a real laboratory environment for measurement. In all cases, it was proven that the FIFO principle is the least suitable to serve real time voice packets in case of congestion. In the case of voice packets, LLQ and PQ proved to be the most appropriate, taking into consideration the jitter, the delay and packet lost rate. Using the two algorithms no voice packets suffered dropping. But knowing that PQ serves the highest priority row, and starves the others, based on the literature and my measurement experience, I drew the conclusion that taking into consideration all criteria, LLQ is the most appropriate algorithm.

The third chapter deals with the performance analysis of Multipath Communication Library (MPT) developed by the Faculty of Informatics, University of Debrecen. I highlighted the drawbacks of the current Internet communication environment, and also the advantages and opportunities that lie in the modern tools such as the performance increase based on multipath support.

¹² The OPNET simulation software mentioned in this dissertation I used in my previous works. On October, 29, 2012, Riverbed acquired OPNET to build on Riverbed's strong heritage and experience in delivering solutions that improve the performance of technology for business. By the spring of 2014 the OPNET simulator was replaced by Riverbed Modeler [47].

Afterwards I presented shortly the working principles of MPTCP published in IETF RFC 684 and I showed an example, which proved the transfer performance of MPTCP. Then I highlighted some of the drawbacks of MPTCP, which exist besides of the many advantages:

- in order to function efficiently, a tuning of the application layer or operating system is necessary
- it works in the transport layer
- it supports only the TCP transfer protocol, which can be an issue for the multimedia applications

I introduced a completely new architecture, the same on which the MPT software is based upon, and the working principle of this is presented. The MPT was designed in Linux operating system, in order to create a measurement laboratory environment, used for analysis of single and multipath way performance.

The dissertation examined the MPT tools' efficiency in case of two-, four-, IPv4-, IPv6- and mixed environment. The transfer performance was studied using symmetrical and asymmetrical bandwidth, and also with different transferred data sizes.

The measurement results proved the fact that the MPT environment is able to efficiently summarize the capacity of several pathways, even if these are different from another. The results are valid for file transfer and multimedia transfer as well.

Concerning the jitter of multimedia transfer, I concluded that, if the speed of summarized pathways are largely similar, the amount of jitter remains relatively small, but if the speeds are quite different, the jitter will increase significantly.

The transfer performance of the MPT tool in case of IPv4, IPv6 and mixed environment proved to be efficient. Despite of the differences in some results in case of different protocol version, based on the measurement results, we can state that the MPT multipath environment summarizes efficiently the transfer capacity physical pathways.

6. Irodalomjegyzék

6.1. Hivatkozások

- [1] QOS, Implementing Cisco Quality of Service, Student Guide, Ver. 2.2, vol. 2, Cisco Systems Inc., 2006.
- [2] A. S. Tanenbaum and D. J. Wetherall, Számítógép-hálózatok. Harmadik, bővített, átdolgozott kiadás, Budapest: Panem Könyvek, 2013.
- [3] T. Svensson and A. Popescu, Development of laboratory exercises based on OPNET Modeler, Blekinge Institute of Technology, Department of Telecommunications and Signal Processing, 2003.
- [4] C. Mancas and M. Mocanu, "QoS Optimization in Congested Multimedia Networks," *Proceedings of the 36th TSP Conference*, pp. 38-42, 2013.
- [5] W. Odom, J. Geier and N. Mehta, CCIE Routing and Switching Official Exam Certification Guide, 2nd Edition, Cisco Press, 2006.
- [6] W. Odom, R. Healy and D. Donohue, CCIE Routing and Switching Certification Guide, 4th Edition, Cisco Press, 2010.
- [7] A. S. Ranjbar, CCNP ONT Official Exam Certification Guide, Cisco Press, 2007.
- [8] B. Durand, J. Sommerville, M. Buchmann and R. Fuller, Administering Cisco QoS in IP Networks, Syngress Publishing, Inc., 2001.
- [9] Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T, Cisco Systems Inc., 2008.
- [10] M. Barreiros and P. Lundqvist, QoS-Enabled Networks - Tools and Foundations, A John Wiley & Sons, Ltd., Publication, 2011.
- [11] K. Wallance, Authorized Self-Study Guide - Cisco Voice over IP (CVOICE), 3rd Edition, Cisco Press, 2009.
- [12] P. K. Verma and L. Wang, "Voice over IP Networks - Quality of Service, Pricing and Security," *Lecture Notes in Electrical Engineering*, vol. 71, 2011.
- [13] S. Kashihara, VoIP Technologies, INTECH Open Access Publisher, 2011.

- [14] M. M. G. Rashed and M. Kabir, "A Comparative Study of Different Queuing Techniques in VoIP, Video Conferencing and FTP," *Daffodil International University Journal of Science and Technology*, vol. 5, no. 1, 2010.
- [15] T. Velmurugan, H. Chandra and S. Balaji, "Comparison of Queuing Disciplines for Differentiated Services using OPNET," in *ARTCom '09*, Kottayam, Kerala, 2009.
- [16] S. Farhangi, A. Rostami and S. Golmohammadi, "A comparative study between combination of PQ and MWRR Queuing techniques in IP network based on OPNET," *Middle-East Journal of Scientific Research*, vol. 13, no. 8, 2013.
- [17] S. Szilágyi and B. Almási, "A Review of Congestion Management Algorithms on Cisco Routers," *Journal of Computer Science and Control Systems*, vol. 5, no. 1, 2012.
- [18] S. Szilágyi, "The Effects of Different Queuing Techniques over VoIP Performance: A Simulation Approach," in *Abstracts & Pre-Proceedings of the 9th International Conference on Applied Mathematics*, Baia Mare, Romania, 2013.
- [19] S. Szilágyi, "Analysis of the Algorithms for Congestion Management in Computer Networks," *Carpathian Journal of Electronic and Computer Engineering*, vol. 6, no. 1, pp. 3-7, 2013.
- [20] L. L. Peterson and B. S. Davie, *Computer Networks: A System Approach - Network Simulation Experiments Manual*, 3rd Edition, Elsevier Inc., 2011.
- [21] A. S. Sethi and V. Y. Hnatyshin, *The Practical User Guide for Computer Network Simulation*, NW: CRC Press Taylor & Francis Group, 2012.
- [22] A. Kuki, "Tools for Performance Evaluation and Modeling in Higher Education of Informatics," *Informatics in the Hungarian Higher Education*, pp. 70-77, 2010.
- [23] A. Cisco Networking, „Pagent IOS Tutorial,” Cisco Systems Inc., 2009.
- [24] A. Cisco, "CCNP: Optimizing Converged Networks v5.0 - Instructor Lab Manual," Cisco Systems, Inc., 2007.
- [25] C. Cisco, „TGN (Traffic GeNerator) User Manual,” Cisco Systems Inc., 2007.
- [26] A. Ford, C. Raiciu, M. Handley and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF RFC-6824, 2013.
- [27] M. Handley and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Address," IETF Internet-Draft, 2013.

- [28] J. F. Kurose és K. W. Ross, Számítógép-hálózatok működése - Alkalmazásorientált megközelítés, Panem Kiadó, 2008.
- [29] A. Singh, C. Görg, A. Timmel-Giel, M. Scharf and T. R. Banniza, "Performance Evaluation of Multipath TCP Linux Implementations," 11th Würzburg Workshop on IP: Joint ITG, ITC and Euro-NF Workshop "Visions of Future Generation Networks", Würzburg, Germany, 2011.
- [30] S. Mohan, S. Sekar, D. Chinnasamy and L. Krishnasamy, "Multipath Routing using Reinforcement Learning Algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 9, pp. 3404-3410, 2013.
- [31] M. Scharf, T. R. Banniza, P. Schefczik, A. Singht and A. Timm-Giel, "Evaluation and Prototyping of Multipath Protocol Mechanisms," 10th Würzburg Workshop on IP: Joint ITG, ITC and Euro-NF Workshop "Visions of Future Generation Networks", Würzburg, Germany, 2010.
- [32] C. Paasch, G. Detal, S. Barre, F. Duchene and O. Bonaventure, "The fastest TCP connection with Multipath TCP," ICTEAM, UCLouvain, Louvain-la-Neuve, Belgium, 2013.
- [33] B. Almási, "Multipath Communication - a new basis for the Future Internet Cognitive Infocommunication," in *Proceedings of the CogInfoCom 2013 Conference*, Budapest, 2013.
- [34] B. Almási és A. Harman, „An Overview of the Multipath Communication Technologies,” in *Proceedings of the Conference on Advances on Wireless Sensor Networks 2013*, 2013.
- [35] B. Almási, "MPT Userguide," [Online]. Available: <http://irh.inf.unideb.hu/user/almasi/mpt>.
- [36] B. Almási, "UDPTUN - Direct TCP connection between 'NAT behind' hosts," *Proceedings of the 8th International Conference on Applied Informatics*, pp. 325-332, 2010.
- [37] B. Almási, "A Simple Solution for Wireless Network Layer Roaming Problems," *Carpathian Journal of Electronic and Computer Engineering*, vol. 5, no. 1, pp. 5-8, 2012.
- [38] B. Almási, "A Solution for Changing the Communication Interfaces between WiFi and 3G without Packet Loss," in *Proceedings of the 37th International Conference on Telecommunications and Signal Processing*, Berlin, Germany, 2014.

- [39] R. Avataram, D. B. Prabhakara and B. A. S. Roopa Devi, "A Hybrid Routing Mechanism for Fast and Secure Data Transmission in MANET," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 8, pp. 3058-3063, 2013.
- [40] T. Singh, B. Kaur and S. K. Dhandu, "Performance Evaluation of Routing Protocols in VANETs," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 9, pp. 3590-3594, 2013.
- [41] G. Lencse and Á. Kovács, "Testing the Channel Aggregation Capability of the MPT Multipath Communication Library," in *World Symposium on Computer networks and Information Security 2014*, Hammamet, Tunisia, 2014.
- [42] B. Almási and S. Szilágyi, "Throughput Performance Analysis of the Multipath Communication Library MPT," in *Proceedings of the 36th International Conference on Telecommunications and Signal Processing*, Rome, Italy, 2013.
- [43] B. Almási and S. Szilágyi, "Multipath FTP and Stream Transmission Analysis using the MPT Software Environment," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 11, pp. 4267-4272, 2013.
- [44] B. Almási and S. Szilágyi, "Investigating the Throughput Performance of the MPT Multipath Communication Library in IPv4 and IPv6," *Journal of Applied Research and Technology*, (accepted) 2015.
- [45] G. Goth, "The End of IPv4 is Nearly Here," *Internet Computing*, vol. 16, no. 2, pp. 7-11, 2012.
- [46] "Global Policy for Post Exhaustion IPv4 Allocation Mechanisms by the IANA," Ratified 6 May 2012.
- [47] "OPNET official homepage," [Online]. Available: www.opnet.com. [Accessed 09. 09. 2013.].

6.2. Publikációim

6.2.1. Folyóíratcikkek:

1. Daniela E. Popescu, Daniel Filipaş, **Szabolcs Szilágyi**, *Some Aspects about a Self-Testing Solution for Implementing the TCP/IP Protocol*, Journal of Computer Science and Control Systems, Vol. 1, No. 1, pp. 82-87, ISSN 1844-6043, University of Oradea Publisher, Oradea, 2008.
2. **Szabolcs Szilágyi**, Béla Almási, *A Review of Congestion Management Algorithms on Cisco Routers*, Journal of Computer Science and Control Systems, Vol. 5, No. 1, pp. 103-107, ISSN 1844 - 6043, University of Oradea Publisher, Oradea, 2012.
3. **Szabolcs Szilágyi**, *Analysis of the Algorithms for Congestion Management in Computer Networks*, Carpathian Journal of Electronic and Computer Engineering, Vol. 6, No. 1, pp. 3-7, ISSN 1844 - 9689, North University of Baia Mare, Faculty of Engineering, Electrical Engineering Department, Baia Mare, 2013.
4. Béla Almási, **Szabolcs Szilágyi**, *Multipath FTP and Stream Transmission Analysis using the MPT Software Environment*, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 11, pp. 4267-4272, ISSN 2278 - 1021 (Online Version), ISBN 2319 - 5940 (Print), November, 2013.
5. **Szabolcs Szilágyi**, *The Effects of Different Congestion Management Algorithms over VoIP Performance*, International Journal of Advanced Computer Science and Applications, Vol. 6, No. 2, pp. 66-70, ISSN 2156-5570 (Online Version), ISSN 2158-107X (Print), Digital Object Identifier: 10.14569/IJACSA.2015.060210, 2015.
6. Béla Almási, **Szabolcs Szilágyi**, *Investigating the Throughput Performance of the MPT Multipath Communication Library in IPv4 and IPv6*, "Journal of Applied Research and Technology, (accepted) 2015.

6.2.2. Konferencia-kiadványok

1. Béla Almási, **Szabolcs Szilágyi**, *Throughput Performance Analysis of the Multipath Communication Library MPT*, TSP 2013 - The 36th International Conference on Telecommunications and Signal Processing, pp. 86-90, ISBN 978-1-4799-0402-0, Digital Object Identifier: 10.1109/TSP.2013.6613897, 2-4 July, 2013, Rome, Italy.
2. **Szabolcs Szilágyi**, *The Effects of Different Queuing Techniques over VoIP Performance: A Simulation Approach*, Abstracts & Pre-Proceedings of the 9th International Conference on Applied Mathematics, pp. 99-103, ISBN 978-606-93094-8-3, BiblioPhil Publisher, 25-28 September, 2013, Baia Mare, Romania.

6.2.3. Konferencia előadások

1. *A Review of Congestion Management Algorithms on Cisco Routers*, EMES'12 - 11th International Conference on Engineering of Modern Electric Systems, Faculty of Electrical Engineering and Information Technology, 2012, University of Oradea, 24-25 May, 2012, Oradea, Romania.
2. *A Review of Congestion Management Algorithms on Routers - Forgalmirányítók Torlódáskezelési Algoritmusainak Áttekintése*, amk2012 - BJMT Conference of Applied Mathematics 2012, Széchenyi István University, 21-23 June, 2012, Győr, Hungary.
3. *The Effects of Different Queuing Techniques over VoIP Performance: A Simulation Approach*, ICAM9 - The 9th International Conference on Applied Mathematics, Technical University of Cluj-Napoca, North University Center at Baia Mare, Faculty of Science, Department of Mathematics and Computer Science, 25-28 September, 2013, Baia Mare, Romania.

A. Függelék: A torlódáskezelési algoritmusok szimuláció által nyert numerikus eredményei

Végponttól-végpontig mért csomagvesztés:

Idő (s)	Csomagvesztés (csomagszám)					
	FIFO	PQ	CQ	WFQ	CBWFQ	LLQ
0	0	0	0	0	0	0
.
.
.
102	0	0	0	0	0	0
105	0	24,66667	0	0	0	0
108	0	23,66667	0	0	0	0
111	0	28,66667	0	0	0	0
114	0	28,66667	0	0	0	0
117	0	30,33333	0	0	0	0
120	0	30,66667	0	0	0	0
123	15,66667	30,66667	0	0	0	0
126	22	31,33333	7	0	0	0
129	32	36	8,666667	0	0	0
132	50	30,66667	8,333333	0	0	0
135	30,33333	40,33333	9	0	0	0
138	31	32,66667	8,666667	0	0	0
141	45	37,33333	8,666667	0	0	0
144	45,66667	37	8,333333	0	0	0
147	50,33333	34,33333	10,33333	0	0	0
150	54	36,33333	11,66667	0	0	0
153	47,66667	37,66667	10,66667	0	0	0
156	54,66667	42,33333	11,66667	0	0	0
159	57,66667	39	12	0	0	0
162	51,66667	38,66667	11,66667	0	0	0
165	58,66667	42,33333	12	0	0	0
168	71,33333	38,66667	12,33333	0	0	0
171	68,66667	36,66667	15	0	0	0
174	53,66667	42,33333	28,66667	0	0	0
177	56,33333	38,33333	21,33333	0	0	0

180	75	41	20	0	0	0
183	73,66667	42,33333	20,33333	0	0	0
186	99	38,33333	21,33333	0	0	0
189	79	38,33333	21,33333	0	0	0
192	68	43	20,33333	0	0	0
195	82,66667	41,66667	20,33333	0	0	0
198	85,66667	39,33333	22	0	0	0
201	71	45	21,33333	0	0	0
204	93,33333	43	31	0	0	0
207	96	43,33333	29,66667	0	0	0
210	90,33333	44	29,33333	0	0	0
213	85,66667	43	35	0	0	0
216	85,66667	40,33333	34	0	0	0
219	76	42	25	0	0	0
222	79	44,33333	24,33333	0	0	0
225	98	47,33333	22,33333	0	0	0
228	110	41	22,66667	0	0	0
231	101,6667	49,66667	23,33333	0	0	0
234	97,66667	49	25,66667	0	0	0
237	112	43	28,33333	0	0	0
240	104,3333	47,33333	25,33333	0	0	0
243	121,6667	46,33333	35	0	0	0
246	88,33333	50	30	0	0	0
249	107,6667	46	31,66667	0	0	0
252	110	41,33333	37,66667	0	0	0
255	98	45	27,33333	0	0	0
258	105,3333	50,66667	34,33333	0	0	0
261	122	45	25	0	0	0
264	113,6667	39	28,66667	0	0	0
267	128,6667	47,33333	26,66667	0	0	0
270	132,3333	47,66667	29,66667	0	0	0
273	127,3333	45	29	0	0	0
276	131,6667	44,33333	23	0	0	0
279	131,6667	44,66667	22	0	0	0
282	114,3333	45	28,33333	0	0	0
285	125,6667	45	24,66667	0	0	0
288	125	46,66667	23,33333	0	0	0

291	112	44,66667	26,66667	0	0	0
294	108,3333	49,66667	39,66667	0	0	0
297	116,6667	47,66667	25,66667	0	0	0
300	116,6667	47,66667	25,66667	0	0	0

Video: Fogadott csomagok száma:

Idő (s)	Fogadott csomagszám					
	FIFO	PQ	CQ	WFQ	CBWFQ	LLQ
0	0	0	0	0	0	0
.
.
.
102	0	0	0	0	0	0
105	18,66667	0	19,66667	19,66667	19,66667	19,66667
108	19	1	20	20	20	20
111	18,33333	0,333333	19,66667	19,66667	19,66667	19,66667
114	19	0,333333	20	20	20	20
117	18,33333	1,333333	19,66667	20,33333	20,33333	20,33333
120	19,33333	1,666667	20	20	20	20
123	18,33333	1,666667	19,66667	19,66667	19,66667	19,66667
126	17,33333	1	20	20,33333	20,33333	20,33333
129	15,66667	1,666667	19,33333	20	20	20
132	16,66667	0,333333	20,33333	20	20	20
135	16	0,666667	19,33333	19,66667	19,66667	19,66667
138	16,66667	0,666667	20,33333	20,33333	20,33333	20,33333
141	15,33333	1	19,33333	20	20	20
144	15,33333	2,666667	20	20	20	20
147	14,33333	1,333333	19	19,66667	19,66667	19,66667
150	15,66667	0,666667	18	20	20	20
153	15,33333	1	18	20	20	20
156	15	0,666667	17,66667	20	20	20
159	13	0,333333	17	20	20	20
162	14,33333	1,333333	17,66667	20,33333	20,33333	20,33333
165	13,33333	0,666667	17,66667	19,66667	19,66667	19,66667
168	12,33333	1,333333	17,66667	20,33333	20,33333	20

171	14	2,666667	17,66667	19,66667	19,66667	20
174	12,66667	1,333333	18	20	20	20
177	15	1,666667	17,66667	20	20	20,33333
180	13,33333	2,333333	17,33333	20	20	19,66667
183	12,33333	1	17,33333	20,33333	20,33333	20
186	12,33333	0,666667	17,33333	19,66667	19,66667	20
189	12	0,666667	17,33333	20	20	20
192	12	2,333333	17,66667	20	20	20
195	13	1	16,66667	20,33333	20,33333	20
198	12,66667	1,666667	17,33333	19,66667	19,66667	20
201	13	1,333333	16,33333	20	20	20
204	11,66667	0,666667	15,66667	20	20	20
207	10	2	15,33333	20	20	20,33333
210	13	1	16	20	20	19,66667
213	12	0,666667	15	20	20	20
216	12,66667	1,333333	15,33333	20,33333	20,33333	20
219	10,33333	1,666667	15,66667	19,66667	19,66667	20
222	13	1	15,66667	20	20	20
225	11,33333	1,333333	16	20	20	20
228	10,66667	1	16	20	20	20
231	11,66667	1,666667	15,66667	20	20	20
234	10	2,333333	15,33333	20	20	20
237	8,666667	1,666667	15,33333	20	20	20
240	9,666667	1,666667	15,33333	20	20	20
243	13,33333	2	15,66667	20	20	20
246	12,66667	2,333333	15,66667	20	20	20
249	9,666667	1,333333	16	20	20	20
252	10	1,333333	15,66667	20	20	20
255	11,33333	1	15,66667	20	20	20
258	9,666667	2,333333	15,33333	20	20	20
261	10,66667	1,666667	15,33333	20	20	20
264	9	0	16	20	20	20
267	8,666667	0,666667	15,33333	20	20	20
270	10	1	15,66667	20	20	20
273	8,666667	1	15,66667	20	20	20
276	9,666667	1,666667	16	20	20	20
279	8,666667	1	16	20	20	20

282	9,333333	2	15,66667	20	20	20
285	9,333333	1,666667	15,33333	20	20	20
288	11	1,666667	15,33333	20,33333	20,33333	20
291	9,333333	1,333333	15,33333	19,66667	19,66667	20
294	11,33333	2,333333	15,66667	20,33333	20,33333	20
297	11,33333	0,666667	16	19,66667	19,66667	20
300	11,33333	0,666667	16	19,66667	19,66667	20

Video: Csomag késleltetés:

Idő (s)	Késleltetés (s)					
	FIFO	PQ	CQ	WFQ	CBWFQ	LLQ
0	N/A	N/A	N/A	N/A	N/A	N/A
.
.
.
102	N/A	N/A	N/A	N/A	N/A	N/A
105	0,189176	0,099234	0,109592	0,101317	0,101317	0,101317
108	0,334443	0,099234	0,131453	0,101388	0,101388	0,101388
111	0,528631	0,099654	0,152045	0,101504	0,101504	0,101504
114	0,740926	0,09929	0,175963	0,101743	0,101743	0,101743
117	0,927494	0,099426	0,198536	0,101701	0,101701	0,101701
120	1,094648	0,099454	0,223746	0,102643	0,102643	0,102643
123	1,261995	0,10077	0,245057	0,101981	0,101981	0,101981
126	1,379451	0,102067	0,27279	0,102369	0,102369	0,102369
129	1,468037	0,100297	0,296846	0,101961	0,101961	0,101961
132	1,53999	0,099234	0,319288	0,102389	0,102389	0,102389
135	1,504772	0,099823	0,350144	0,10182	0,10182	0,10182
138	1,495828	0,101556	0,370146	0,103091	0,103091	0,103081
141	1,510048	0,100512	0,408946	0,102835	0,102835	0,102835
144	1,518967	0,101624	0,434453	0,101847	0,101847	0,101847
147	1,533305	0,099571	0,436677	0,102811	0,102811	0,102567
150	1,542234	0,101372	0,445277	0,10262	0,10262	0,102679
153	1,512261	0,099435	0,446347	0,102374	0,102374	0,102366
156	1,54063	0,100016	0,439847	0,102006	0,102006	0,102162
159	1,532137	0,103934	0,464652	0,102795	0,102795	0,102662

162	1,516269	0,099908	0,478664	0,103096	0,103096	0,102863
165	1,54458	0,099781	0,477842	0,101691	0,101691	0,101755
168	1,574798	0,099577	0,503664	0,102849	0,102849	0,102849
171	1,553647	0,101344	0,538868	0,103416	0,103416	0,102884
174	1,544047	0,101334	0,563523	0,103403	0,103403	0,10378
177	1,500934	0,102381	0,593836	0,103576	0,103576	0,103174
180	1,531855	0,10094	0,611735	0,102824	0,102824	0,102199
183	1,539706	0,102174	0,64376	0,10243	0,10243	0,102761
186	1,55774	0,101005	0,666147	0,103481	0,103481	0,103288
189	1,558051	0,099882	0,698345	0,10287	0,10287	0,102733
192	1,53394	0,101154	0,723807	0,103453	0,103453	0,103148
195	1,529771	0,10381	0,749745	0,10253	0,10253	0,102437
198	1,56212	0,101589	0,777181	0,102861	0,102861	0,102482
201	1,505149	0,102444	0,798121	0,103226	0,103226	0,103182
204	1,547758	0,103162	0,80155	0,103014	0,103014	0,102545
207	1,537073	0,102327	0,801983	0,103358	0,103358	0,102936
210	1,544182	0,100961	0,802052	0,103275	0,103275	0,103593
213	1,526672	0,100436	0,802118	0,103459	0,103459	0,103353
216	1,565861	0,103655	0,801621	0,103172	0,103172	0,103434
219	1,548274	0,101174	0,80149	0,102389	0,102389	0,102857
222	1,530301	0,102354	0,803052	0,103198	0,103198	0,103246
225	1,572749	0,102001	0,799926	0,103374	0,103374	0,102692
228	1,565363	0,100489	0,800374	0,10275	0,10275	0,103535
231	1,549587	0,104271	0,801393	0,103295	0,103295	0,103178
234	1,537246	0,1025	0,802725	0,102718	0,102718	0,103376
237	1,553292	0,10289	0,802	0,103435	0,103435	0,103613
240	1,548509	0,101511	0,800693	0,103446	0,103446	0,103514
243	1,585968	0,102283	0,802568	0,10333	0,10333	0,103268
246	1,517829	0,102047	0,800784	0,103082	0,103082	0,103105
249	1,580305	0,103207	0,800929	0,103522	0,103522	0,103669
252	1,552705	0,101906	0,801841	0,103422	0,103422	0,10352
255	1,546393	0,102681	0,800717	0,104037	0,104037	0,103597
258	1,566565	0,101564	0,801405	0,103252	0,103252	0,103682
261	1,574505	0,100554	0,80175	0,103536	0,103536	0,103313
264	1,547603	0,104513	0,802369	0,103525	0,103525	0,103605
267	1,576599	0,104513	0,80068	0,103643	0,103643	0,103351
270	1,587391	0,102058	0,80027	0,103317	0,103317	0,103247

273	1,595388	0,105799	0,802595	0,103331	0,103331	0,103492
276	1,5915	0,10145	0,80115	0,103737	0,103737	0,103373
279	1,593687	0,101756	0,801633	0,102791	0,102791	0,10329
282	1,559986	0,102251	0,80032	0,103464	0,103464	0,103452
285	1,562597	0,100797	0,801824	0,101933	0,101933	0,103634
288	1,569155	0,101804	0,802156	0,102386	0,102386	0,103237
291	1,567829	0,101966	0,80242	0,102551	0,102551	0,103255
294	1,551569	0,10222	0,802433	0,102608	0,102608	0,103486
297	1,53715	0,099336	0,800331	0,103324	0,103324	0,103545
300	1,53715	0,099336	0,800331	0,103324	0,103324	0,103545

Video: Dzsitter:

Idő (s)	Késleltetés ingadozás (s)				
	FIFO	PQ	CQ	WFQ/ CBWFQ	LLQ
0	N/A	N/A	N/A	N/A	N/A
.
.
.
102	N/A	N/A	N/A	N/A	N/A
105	0,01132478310	0,00000000000	0,00019271010	0,00001110481	0,00001110481
108	0,01766575557	0,00000000000	0,00064718560	0,00000673894	0,00000673894
111	0,02700692701	0,00000000000	0,00140585330	0,00000635232	0,00000635232
114	0,04968267561	0,00000000058	0,00260053755	0,00000600060	0,00000600060
117	0,08044410844	0,00000002955	0,00420436343	0,00000594601	0,00000594601
120	0,11735516833	0,00000010010	0,00619356091	0,00000594960	0,00000594960
123	0,15543194867	0,00000169626	0,00853019324	0,00000595315	0,00000595315
126	0,18092328179	0,00000270183	0,01149642841	0,00000595110	0,00000595110
129	0,19896568170	0,00000253279	0,01468468533	0,00000597684	0,00000597684
132	0,23995774178	0,00000222391	0,01849163370	0,00000598189	0,00000598189
135	0,25464799768	0,00000208731	0,02267325926	0,00000596827	0,00000596827
138	0,26126901817	0,00000270671	0,02755725779	0,00000597509	0,00000597509
141	0,26446163100	0,00000290590	0,03269926766	0,00000594091	0,00000594091
144	0,26534172052	0,00000601622	0,03827019202	0,00000593385	0,00000593385
147	0,26483562057	0,00000569924	0,04346546184	0,00000595268	0,00000593601
150	0,26308681592	0,00000574486	0,04726570207	0,00000596336	0,00000595031

153	0,26120891065	0,00000560679	0,05008522093	0,00000596497	0,00000595656
156	0,25842057972	0,00000531646	0,05213605748	0,00000593683	0,00000592844
159	0,25523643190	0,00000547976	0,05364176593	0,00000594076	0,00000593727
162	0,25256025140	0,00000095907	0,05472382075	0,00000594748	0,00000595393
165	0,24912687920	0,00000538018	0,05555134380	0,00000595018	0,00000596103
168	0,24578980451	0,00000522922	0,05603716834	0,00000594473	0,00000596225
171	0,24449350332	0,00000493366	0,05627607461	0,00000602231	0,00000597678
174	0,24203738209	0,00000459947	0,05636381844	0,00000626793	0,00000625562
177	0,23916428897	0,00000459826	0,05636311171	0,00000650376	0,00000642350
180	0,23505250086	0,00000437066	0,05627055884	0,00000654998	0,00000640759
183	0,23071018260	0,00000427385	0,05613756024	0,00000656952	0,00000648463
186	0,22759867532	0,00000416542	0,05586074994	0,00000665633	0,00000660657
189	0,22403056196	0,00000397802	0,05554148760	0,00000672546	0,00000664047
192	0,22123481747	0,00000417953	0,05517901323	0,00000684703	0,00000671866
195	0,21685830053	0,00000479986	0,05474557863	0,00000690786	0,00000675735
198	0,21282501339	0,00000473129	0,05432392407	0,00000692247	0,00000673067
201	0,20943512710	0,00000513374	0,05385258456	0,00000699732	0,00000681990
204	0,20541124793	0,00000506358	0,05503823395	0,00000701650	0,00000684743
207	0,20246402338	0,00000553870	0,05958431276	0,00000705903	0,00000685747
210	0,19968542984	0,00000541872	0,06383976635	0,00000708252	0,00000692864
213	0,19620217149	0,00000527709	0,06749133052	0,00000715288	0,00000700160
216	0,19296858012	0,00000539494	0,07093662432	0,00000719266	0,00000707544
219	0,19020618101	0,00000576998	0,07402432005	0,00000718607	0,00000712006
222	0,18793036008	0,00000590962	0,07684583640	0,00000720515	0,00000715501
225	0,18489857781	0,00000576214	0,07934779296	0,00000728127	0,00000716433
228	0,18189635675	0,00000573138	0,08155445141	0,00000730176	0,00000722608
231	0,17966891625	0,00000639986	0,08348239233	0,00000732908	0,00000726101
234	0,17660625347	0,00000669178	0,08521475107	0,00000732586	0,00000730330
237	0,17434789303	0,00000668087	0,08677881584	0,00000734499	0,00000734371
240	0,17286989356	0,00000651227	0,08814857542	0,00000736274	0,00000735867
243	0,17126860778	0,00000629973	0,08945405928	0,00000736564	0,00000737434
246	0,16892399179	0,00000616484	0,09053716213	0,00000737714	0,00000737497
249	0,16647670427	0,00000617587	0,09157971012	0,00000741764	0,00000739272
252	0,16502199063	0,00000615755	0,09242893347	0,00000742994	0,00000741110
255	0,16323405496	0,00000641442	0,09317473312	0,00000748289	0,00000742667
258	0,16153710601	0,00000637574	0,09382933730	0,00000748576	0,00000745448
261	0,15984258782	0,00000624212	0,09438368698	0,00000752569	0,00000747184

264	0,15834048066	0,00000656399	0,09488915486	0,00000754830	0,00000750222
267	0,15698639722	0,00000656399	0,09527145121	0,00000755530	0,00000750743
270	0,15577516742	0,00000652617	0,09558942881	0,00000757247	0,00000751302
273	0,15473873002	0,00000697605	0,09587583823	0,00000757598	0,00000752729
276	0,15388359386	0,00000693072	0,09607804745	0,00000758515	0,00000753089
279	0,15225406084	0,00000697934	0,09623872733	0,00000759301	0,00000751878
282	0,15091593892	0,00000708321	0,09634072425	0,00000758443	0,00000751986
285	0,14958088064	0,00000701814	0,09639121350	0,00000756940	0,00000752810
288	0,14836010390	0,00000701309	0,09640773990	0,00000755580	0,00000752900
291	0,14673228192	0,00000709177	0,09640957279	0,00000754704	0,00000751664
294	0,14580018189	0,00000717815	0,09637766120	0,00000753931	0,00000750539
297	0,14418408690	0,00000546059	0,09631399960	0,00000753239	0,00000750890
300	0,14418408690	0,00000546059	0,09631399960	0,00000753239	0,00000750890

VoIP: Fogadott csomagok száma:

Idő (s)	Fogadott csomagszám					
	FIFO	PQ	CQ	WFQ	CBWFQ	LLQ
0	0	0	0	0	0	0
.
.
.
102	0	0	0	0	0	0
105	371,3333	396,6667	386	396,6667	396,6667	396,6667
108	380	399,6667	391,6667	400	400	400
111	367,3333	400,3333	391,6667	399,3333	399,3333	399,3333
114	376,3333	399,3333	391,6667	400,6667	400,6667	400,6667
117	370,3333	400,6667	391,6667	400,3333	400,3333	400,3333
120	385,3333	399,3333	391,6667	399,6667	399,6667	399,6667
123	371,3333	400,6667	391,6667	399,3333	399,3333	399,3333
126	373,6667	399,6667	390	400,6667	400,6667	400,6667
129	376,6667	400,3333	386,6667	400,3333	400,3333	400,3333
132	376,6667	399,6667	398,3333	399,6667	399,6667	399,6667
135	387,3333	400,3333	385,3333	400	400	400
138	381	400	398	400	400	400
141	374,3333	399,3333	380,3333	400	400	400

144	376	401,3333	391	400,6667	400,6667	400,6667
147	365	399,3333	391	399	399	399
150	376	400	382	400,6667	400,6667	400,6667
153	378,6667	400	384,6667	399,6667	399,6667	399,6667
156	370,6667	400	383,3333	399,3333	399,3333	400,3333
159	364	399,3333	383,3333	400,6667	400,6667	399
162	384	400	391	400	400	401
165	368,3333	400	383,3333	399,3333	399,3333	399,3333
168	357,6667	400	383,3333	401	401	400
171	361,3333	400	383,3333	399	399	401
174	369,3333	400,3333	369,6667	400,6667	400,6667	399,3333
177	384,6667	400,3333	381,6667	400	400	400,6667
180	361,6667	400,3333	383,3333	399,3333	399,3333	399,6667
183	367,3333	400	383,3333	401,3333	401,3333	399
186	343,3333	399,3333	383,3333	399	399	400,6667
189	368,6667	400,3333	383,3333	400	400	399,3333
192	373,6667	399,6667	379,3333	400,3333	400,3333	400
195	361	401	379,6667	400	400	401
198	351	398,6667	383,3333	400	400	399
201	380,6667	400	383,3333	400	400	401
204	360,6667	400	383,3333	400	400	400
207	357,3333	400	383,3333	399,3333	399,3333	400
210	358,3333	400	375,6667	400,3333	400,3333	399,6667
213	363,3333	400	383,3333	399,6667	399,6667	400
216	354	400	383,3333	400,6667	400,6667	400,3333
219	371,3333	400	383,3333	399,3333	399,3333	399
222	365	400	383,3333	400	400	400,6667
225	355,6667	400,6667	375,6667	400,3333	400,3333	399,3333
228	353	399,3333	383,3333	400,3333	400,3333	400,3333
231	355,3333	400,3333	375,6667	400	400	399,6667
234	357,3333	400,3333	383,3333	399,6667	399,6667	400,6667
237	345,3333	399,3333	383,3333	400,3333	400,3333	399,6667
240	360,6667	400,3333	383,3333	399,3333	399,3333	400,3333
243	338,6667	400,3333	383,3333	400	400	399,3333
246	368,3333	399,3333	383,3333	400,6667	400,6667	401
249	345,3333	400	383,3333	399,6667	399,6667	399,6667
252	354,3333	400,3333	375,6667	400,3333	400,3333	399,3333

255	354,3333	400	375,6667	400	400	401,3333
258	357	401	383,3333	400,3333	400,3333	399,3333
261	338,6667	398,6667	383,3333	399,6667	399,6667	400,3333
264	358,6667	400,3333	383,3333	400,3333	400,3333	400
267	339,3333	399,6667	383,3333	399	399	399
270	340,3333	400,3333	383,3333	400,3333	400,3333	400
273	341,3333	399,6667	383,3333	400,6667	400,6667	400,6667
276	334	400	376,6667	399,6667	399,6667	400
279	338,6667	401	374,6667	400	400	400,6667
282	347,6667	399,6667	383,3333	400,3333	400,3333	399
285	355,3333	399,3333	383,3333	399,3333	399,3333	401
288	339,3333	400	383,3333	400,6667	400,6667	398,6667
291	353,3333	400,6667	383,3333	399,6667	399,6667	401,3333
294	352	400	383,3333	400,3333	400,3333	399,3333
297	353,6667	399,3333	383,3333	400	400	399,3333
300	353,6667	399,3333	383,3333	400	400	399,3333

VoIP: Csomag késleltetés:

Idő (s)	Késleltetés (s)					
	FIFO	PQ	CQ	WFQ	CBWFQ	LLQ
0	N/A	N/A	N/A	N/A	N/A	N/A
.
.
.
102	N/A	N/A	N/A	N/A	N/A	N/A
105	0,133916	0,004921	0,075757	0,006857	0,006857	0,005028
108	0,280686	0,005387	0,135738	0,006791	0,006791	0,005024
111	0,478025	0,005378	0,195628	0,007406	0,007406	0,005028
114	0,685419	0,005326	0,257103	0,007474	0,007474	0,005043
117	0,874903	0,005378	0,319982	0,007504	0,007504	0,005089
120	1,043283	0,005405	0,381825	0,00781	0,00781	0,005211
123	1,212673	0,005401	0,44426	0,007355	0,007355	0,005134
126	1,337156	0,005405	0,50236	0,008118	0,008118	0,005119
129	1,422517	0,005372	0,514948	0,007514	0,007514	0,005096
132	1,483374	0,005401	0,501835	0,008207	0,008207	0,005158

135	1,447602	0,005465	0,522243	0,007433	0,007433	0,005056
138	1,444723	0,005366	0,502956	0,008637	0,008637	0,005228
141	1,465131	0,005542	0,540606	0,008304	0,008304	0,005151
144	1,468688	0,005537	0,565552	0,007454	0,007454	0,005061
147	1,473145	0,005538	0,543752	0,008607	0,008607	0,005154
150	1,48881	0,005434	0,607961	0,008219	0,008219	0,005183
153	1,454859	0,0054	0,638212	0,007418	0,007418	0,005127
156	1,483961	0,005495	0,691771	0,007394	0,007394	0,00517
159	1,470885	0,00543	0,754391	0,008494	0,008494	0,005174
162	1,463152	0,005511	0,802531	0,00893	0,00893	0,005192
165	1,485344	0,005659	0,845127	0,007045	0,007045	0,005046
168	1,512259	0,005439	0,909758	0,008285	0,008285	0,005257
171	1,49591	0,005598	0,969952	0,008995	0,008995	0,005252
174	1,485517	0,005585	0,994109	0,009301	0,009301	0,005297
177	1,44457	0,005572	0,990612	0,009473	0,009473	0,005249
180	1,476949	0,005582	0,992371	0,008227	0,008227	0,005119
183	1,479695	0,005563	0,993523	0,008048	0,008048	0,005183
186	1,503658	0,005554	0,992343	0,009333	0,009333	0,005259
189	1,499065	0,00525	0,992734	0,008501	0,008501	0,005214
192	1,478096	0,005499	0,993889	0,009215	0,009215	0,005205
195	1,475264	0,005676	0,99475	0,007975	0,007975	0,005221
198	1,502854	0,005561	0,99368	0,008332	0,008332	0,00515
201	1,446275	0,005656	0,991378	0,008939	0,008939	0,005267
204	1,488848	0,005578	0,994072	0,008523	0,008523	0,00517
207	1,489567	0,005676	0,994634	0,009191	0,009191	0,005185
210	1,493966	0,005607	0,995266	0,008915	0,008915	0,005248
213	1,475761	0,005469	0,995173	0,009066	0,009066	0,005298
216	1,512567	0,005665	0,993996	0,008721	0,008721	0,005241
219	1,483207	0,005603	0,99432	0,007921	0,007921	0,005231
222	1,474438	0,005583	0,994592	0,009112	0,009112	0,005273
225	1,515589	0,005631	0,992432	0,008809	0,008809	0,00519
228	1,507106	0,005545	0,992955	0,008314	0,008314	0,005297
231	1,49878	0,005717	0,993331	0,00935	0,00935	0,005238
234	1,485625	0,005705	0,995746	0,008185	0,008185	0,005303
237	1,500819	0,005696	0,994283	0,009152	0,009152	0,00526
240	1,497774	0,005581	0,99338	0,009147	0,009147	0,005288
243	1,524315	0,005561	0,994294	0,008849	0,008849	0,005227

246	1,465635	0,005735	0,994345	0,008404	0,008404	0,005279
249	1,528093	0,00549	0,993247	0,009342	0,009342	0,005262
252	1,501966	0,005571	0,994769	0,009398	0,009398	0,005294
255	1,497646	0,00572	0,993296	0,009938	0,009938	0,005309
258	1,494222	0,005619	0,995034	0,009096	0,009096	0,005264
261	1,507321	0,005591	0,994926	0,009294	0,009294	0,005219
264	1,49367	0,005608	0,993133	0,009161	0,009161	0,005289
267	1,513511	0,005569	0,994018	0,009413	0,009413	0,005267
270	1,521791	0,00564	0,993378	0,009103	0,009103	0,005245
273	1,532547	0,00568	0,99417	0,008963	0,008963	0,005195
276	1,545864	0,005663	0,993671	0,009442	0,009442	0,005226
279	1,534244	0,00569	0,993627	0,008107	0,008107	0,005247
282	1,505137	0,00565	0,993795	0,009382	0,009382	0,005264
285	1,501346	0,00559	0,993842	0,007462	0,007462	0,005251
288	1,501878	0,005559	0,994812	0,007837	0,007837	0,005244
291	1,513124	0,005541	0,99556	0,008369	0,008369	0,005234
294	1,486199	0,005644	0,995343	0,007879	0,007879	0,005261
297	1,484575	0,005554	0,993369	0,009193	0,009193	0,005256
300	1,484575	0,005554	0,993369	0,009193	0,009193	0,005256

VoIP: Dzsitter:

Idő (s)	Késleltetés ingadozás (s)				
	FIFO	PQ	CQ	WFQ/ CBWFQ	LLQ
0	N/A	N/A	N/A	N/A	N/A
.
.
.
102	N/A	N/A	N/A	N/A	N/A
105	0,00316256819	0,00000615604	0,00108501526	0,00003420547	0,00000556957
108	0,00831077922	0,00000628252	0,00234577624	0,00004631302	0,00000540104
111	0,01614200153	0,00000634153	0,00476321748	0,00004814738	0,00000539674
114	0,03509034992	0,00000626745	0,00847810833	0,00005091906	0,00000538196
117	0,06157604213	0,00000623650	0,01359752336	0,00005303920	0,00000540558
120	0,09101418632	0,00000622856	0,02006586662	0,00005479527	0,00000541923
123	0,12335369788	0,00000626274	0,02781465144	0,00005513507	0,00000541529

126	0,15664681383	0,00000629700	0,03704980964	0,00005767964	0,00000540534
129	0,18428130110	0,00000633131	0,04472284455	0,00005832211	0,00000540780
132	0,20838638894	0,00000634727	0,04700757645	0,00005939783	0,00000540807
135	0,22272370791	0,00000632569	0,05014901665	0,00005918878	0,00000540144
138	0,22702658533	0,00000631186	0,04894442528	0,00006111505	0,00000540168
141	0,22531378579	0,00000631375	0,05079472666	0,00006288041	0,00000539997
144	0,22476735684	0,00000633767	0,04877415987	0,00006277616	0,00000539733
147	0,22283776257	0,00000637507	0,04796800764	0,00006305551	0,00000539811
150	0,21703659309	0,00000639848	0,04791945447	0,00006422463	0,00000539698
153	0,21201504505	0,00000639958	0,04647412689	0,00006447457	0,00000539650
156	0,20734665652	0,00000640938	0,04855375002	0,00006405230	0,00000540097
159	0,20390197345	0,00000642184	0,05030340302	0,00006422630	0,00000539834
162	0,19630818023	0,00000644925	0,05325176244	0,00006492397	0,00000539592
165	0,19225050782	0,00000647297	0,05825265123	0,00006527176	0,00000539184
168	0,18687401857	0,00000647560	0,06411307188	0,00006503144	0,00000539102
171	0,18289844358	0,00000649572	0,07209306118	0,00006587303	0,00000539133
174	0,17936362988	0,00000651278	0,08107206245	0,00006716614	0,00000538702
177	0,17474794687	0,00000651723	0,08861656722	0,00006826559	0,00000538970
180	0,16760723020	0,00000652140	0,09502760075	0,00006945347	0,00000539061
183	0,16681229261	0,00000654358	0,10026504351	0,00006945979	0,00000538669
186	0,16317147757	0,00000654737	0,10453042060	0,00006957792	0,00000538787
189	0,15778944667	0,00000655285	0,10792013979	0,00007042673	0,00000538690
192	0,15750869395	0,00000656239	0,11119363764	0,00007104082	0,00000538115
195	0,15148964784	0,00000656868	0,11343076488	0,00007166810	0,00000538192
198	0,14826710834	0,00000657767	0,11468128029	0,00007143021	0,00000538339
201	0,14696079776	0,00000658725	0,11596426565	0,00007197919	0,00000538209
204	0,14075673471	0,00000659964	0,11694668060	0,00007234509	0,00000537794
207	0,13901154859	0,00000660854	0,11765274378	0,00007280349	0,00000537575
210	0,13627886656	0,00000661866	0,11940277370	0,00007329547	0,00000537431
213	0,13274772797	0,00000662613	0,11835649068	0,00007385774	0,00000537526
216	0,13093212979	0,00000663323	0,11841297737	0,00007409848	0,00000537357
219	0,12846376074	0,00000664291	0,11829615652	0,00007412185	0,00000537246
222	0,12481644577	0,00000665887	0,11803798255	0,00007426000	0,00000537169
225	0,12402546926	0,00000666426	0,11630759864	0,00007461878	0,00000537097
228	0,12107035563	0,00000667947	0,11722170505	0,00007486777	0,00000536928
231	0,12081784124	0,00000668821	0,11809800712	0,00007501518	0,00000536875
234	0,11751160550	0,00000670332	0,11610101295	0,00007520444	0,00000536793

237	0,11368927841	0,00000671820	0,11544674379	0,00007549905	0,00000536643
240	0,11418493349	0,00000671722	0,11473228795	0,00007606066	0,00000536615
243	0,11173175568	0,00000672285	0,11397511717	0,00007626990	0,00000536435
246	0,10975305591	0,00000673766	0,11318440203	0,00007637435	0,00000536403
249	0,10889491360	0,00000674563	0,11234450464	0,00007659922	0,00000536391
252	0,10655829128	0,00000675822	0,11008334026	0,00007704719	0,00000536264
255	0,10508579333	0,00000676794	0,11210348311	0,00007777714	0,00000536052
258	0,10467504651	0,00000677309	0,10979495722	0,00007804452	0,00000535657
261	0,10255515203	0,00000678277	0,10890652929	0,00007851581	0,00000535301
264	0,10095533678	0,00000678325	0,10800667024	0,00007885753	0,00000535024
267	0,09900439149	0,00000679436	0,10709229816	0,00007923446	0,00000535154
270	0,09788011671	0,00000679597	0,10617860846	0,00007946116	0,00000535465
273	0,09637049567	0,00000680515	0,10524603457	0,00007969352	0,00000535089
276	0,09667527723	0,00000681356	0,10316467505	0,00007979009	0,00000534711
279	0,09563459544	0,00000682201	0,10463818640	0,00007997773	0,00000534690
282	0,09324110398	0,00000682660	0,10254378776	0,00008005687	0,00000534412
285	0,09271183451	0,00000683636	0,10164281356	0,00007989838	0,00000534248
288	0,09193314771	0,00000683773	0,10070657444	0,00007960824	0,00000534457
291	0,08983238177	0,00000683954	0,09982461476	0,00007947042	0,00000534540
294	0,09023265234	0,00000684288	0,09893528462	0,00007943857	0,00000534510
297	0,08677896255	0,00000684448	0,09806279508	0,00007939587	0,00000534296
300	0,08677896255	0,00000684448	0,09806279508	0,00007939587	0,00000534296

B. Függelék: A torlódáskezelési algoritmusok fizikai eszközökön való implementálásához használt eszközök konfigurációs fájljai

ForgGEN:

```
!  
version 12.4  
service timestamps debug datetime msec  
service timestamps log datetime msec  
service password-encryption  
!  
hostname ForgGen  
!  
boot-start-marker  
boot system flash c2801-tpgen+ipbase-mz.PAGENT.4.3.0.bin  
boot-end-marker  
!  
enable secret 5 $1$p5rJ$HtAnkccacJRowf4PNSVVu0  
!  
no aaa new-model  
memory-size iomem 25  
ip cef  
!  
!  
!  
!  
no ip domain lookup  
ip host PAGENT-SECURITY-V3 27.92.92.33 78.72.0.0  
multilink bundle-name authenticated  
!  
!  
!  
!  
!  
!  
!  
interface FastEthernet0/0  
 ip address 172.16.20.1 255.255.255.0  
 duplex auto  
 speed 10  
!  
interface FastEthernet0/1  
 ip address 172.16.10.1 255.255.255.0  
 duplex auto  
 speed 10  
!  
interface Serial0/1/0  
 no ip address
```

```

shutdown
clock rate 2000000
!
interface Serial0/1/1
no ip address
shutdown
!
!
!
ip http server
!
no cdp run
!
!
control-plane
!
!
line con 0
exec-timeout 0 0
logging synchronous
line aux 0
line vty 0 4
password 7 13061E010803
logging synchronous
login
!
scheduler allocate 20000 1000
end

```

R1:

```

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$IYVW$GZMbaSQX90e3Jo9d5iUTs.
!
no aaa new-model
memory-size iomem 25
ip cef
!
!
!

```

```

!
no ip domain lookup
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
!
voice-card 0
!
!
!
!
!
!
!

interface FastEthernet0/0
  no ip address
  shutdown
  duplex auto
  speed auto
!
interface FastEthernet0/1
  ip address 172.16.10.2 255.255.255.0
  duplex auto
  speed 10
!
interface Serial0/1/0
  no ip address
  shutdown
  clock rate 2000000
!
interface Serial0/1/1
  bandwidth 2048
  ip address 10.150.1.1 255.255.255.252
  fair-queue
!
ip forward-protocol nd
ip route 172.16.20.0 255.255.255.0 Serial0/1/1
!
!
!
ip http server
no ip http secure-server
!
no cdp run
!
!
!
control-plane
!
!
!

```



```

line con 0
  logging synchronous
line aux 0
line vty 0 4
  password 7 0822455D0A16
  logging synchronous
  login
!
scheduler allocate 20000 1000
end

```

R2:

```

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$VvTk$JQgDv3V3GtuYDAjy8MQjt1
!
no aaa new-model
memory-size iomem 25
ip cef
!
!
!
!
no ip domain lookup
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
!
voice-card 0
!
!
!
!
!
!
!
interface FastEthernet0/0
  ip address 172.16.20.2 255.255.255.0
  duplex auto
  speed 10
!

```

```

interface FastEthernet0/1
  no ip address
  shutdown
  duplex auto
  speed auto
!
interface Serial0/1/0
  ip address 10.150.1.2 255.255.255.252
  clock rate 128000
!
interface Serial0/1/1
  no ip address
  shutdown
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
!
no cdp run
!
!
!
control-plane
!
!
!
!
!
!
line con 0
  logging synchronous
line aux 0
line vty 0 4
  password 7 060506324F41
  logging synchronous
  login
!
scheduler allocate 20000 1000
end

```

C. Függelék: A torlódáskezelési algoritmusok fizikai eszközökön való implementálása

FIFO

```
int s0/1/1
no fair-queue
end
```

PQ

```
access-list 101 permit tcp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 21
access-list 102 permit tcp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 1720
access-list 103 permit udp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 5060
```

```
priority-list 1 protocol ip high list 103
priority-list 1 protocol ip medium list 102
priority-list 1 protocol ip normal list 101
priority-list 1 default low
```

```
priority-list 1 queue-limit 20 40 60 80
```

```
int s0/1/1
priority-group 1
end
```

CQ

```
access-list 101 permit tcp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 21
access-list 102 permit tcp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 1720
access-list 103 permit udp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 5060
```

```
queue-list 1 protocol ip 2 list 103
queue-list 1 protocol ip 3 list 102
queue-list 1 protocol ip 4 list 101
queue-list 1 default 1
```

```
queue-list 1 queue 1 limit 4
queue-list 1 queue 2 limit 10
queue-list 1 queue 3 limit 10
queue-list 1 queue 4 limit 4
```

```
int s0/1/1
custom-queue-list 1
end
```

WFQ

```
int s0/1/1
fair-queue
end
```

CBWFQ

```
access-list 101 permit tcp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 21
access-list 102 permit tcp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 1720
access-list 103 permit udp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 5060
```

```
class-map VOICE
match access-group 103
exit
class-map VIDEO
match access-group 102
exit
class-map FTP
match access-group 101
exit
```

```
policy-map R1-Serial
class VOICE
bandwidth percent 30
exit
class VIDEO
bandwidth percent 30
exit
class FTP
bandwidth percent 10
exit
class class-default
bandwidth percent 5
exit
exit
```

```
int s0/1/1
no fair-queue
service-policy output R1-Serial
end
```

LLQ

```
access-list 101 permit tcp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 21
access-list 102 permit tcp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 1720
access-list 103 permit udp 172.16.10.0 0.0.0.255 172.16.20.0
0.0.0.255 eq 5060
```

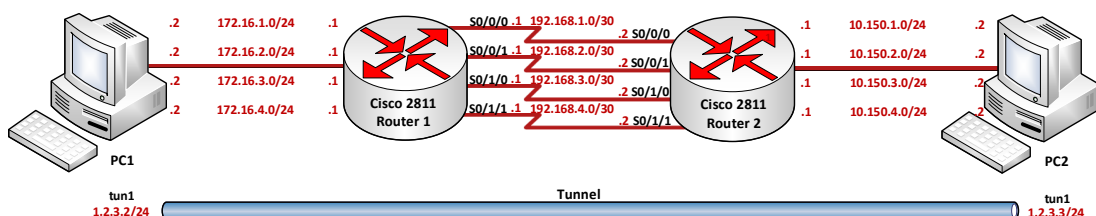
```
class-map VOICE
match access-group 103
exit
class-map VIDEO
match access-group 102
exit
class-map FTP
match access-group 101
exit
```

```
policy-map R1-Serial
class VOICE
priority 384
exit
class VIDEO
bandwidth percent 30
exit
class FTP
bandwidth percent 10
exit
class class-default
bandwidth percent 5
exit
exit
```

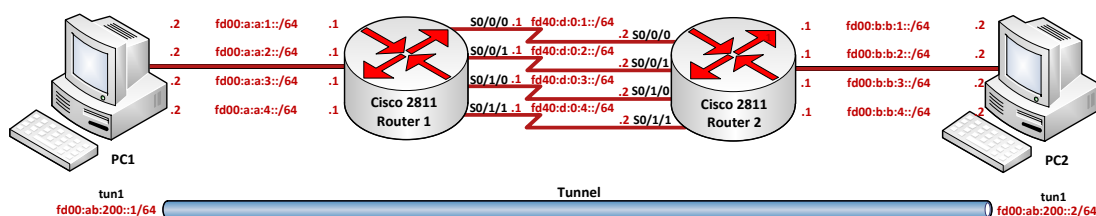
```
int s0/1/1
no fair-queue
service-policy output R1-Serial
end
```

D. Függelék: Az MPT átviteli teljesítményének hatékonyságelemzése céljából használt IPv4 és IPv6 mérési környezetek

Csak IPv4-es hálózat



Csak IPv6-os hálózat



Vegyes hálózat (IPv6 feletti IPv4)

