Aktív tárolókezelő mechanizmusok hatása a TCP adaptivitására

SZABÓ ZOLTÁN[1] – KENESI ZSOLT[2] – MOLNÁR SÁNDOR[1]

szabo.z@alpha.ttt.bme.hu – Zsolt.Kenesi@eth.ericsson.se – molnar@ttt-atm.ttt.bme.hu

BME, Távközlés és Telematika Tanszék[1], Ericsson Magyarország Kft[2]

1. Bevezetés

Napjainkban az IP (Internet Protocol) alapú távközlés térhódítása és ezzel együtt a jelentősége is egyre meghatározóbb. A nem kapcsolat orientált IP-t egészíti ki a TCP (Transmission Control Protocol) szállítás rétegbeli protokoll [3], melyet megbízhatósága miatt leginkább a csomagvesztésre érzékeny adatfolyamok továbbítására alkalmaznak.

A TCP legfontosabb feladata, hogy a lehető leggyorsabban minden csomagot hibátlanul, helyes sorrendben továbbítson a hálózat két végpontja között, s ebből adódik az egyik legfontosabb tulajdonsága is. A nyugtázásokkal, a torlódás kezelő mechanizmusaival próbál minél hatékonyabban alkalmazkodni a link szabad kapacitásváltozásaihoz. Ezen mechanizmusok segítségével a TCP adaptálódik a háttérforgalomhoz és a maximális szabad linkkapacitást igyekszik elfoglalni.

Az adaptáció nem csak azért fontos, mert ez a teljesítményre is hatással van, hanem mert a TCP ezzel az adaptációval bizonyítottan [1] statisztikus jellemzőket vesz át a háttérforgalomtól. Tehát egy olyan érdekes jelenséget tapasztalhatunk, hogy a TCP forgalomra "átmásolódnak" a háttérforgalom jellemzői. Így például a háttérforgalom fraktális tulajdonságai átöröklődhetnek a TCP forgalomra. A TCP protokoll azonban egy végpontok közötti protokoll, így ezen forgalmi tulajdonságokat „elszállítja" a hálózat más pontjára is. Érdekes módon TCP forgalmak egymáshoz is tudnak adaptálódni [1], így az Internetet behálózó és egymást keresztező TCP folyamok egymásnak átadva a hálózat minden pontjára elterjeszthetik ezeket a forgalmi jellemzőket. A jelenség kulcsa a TCP adaptációja, így ennek megértése elengedhetetlen annak pontos magyarázatához.

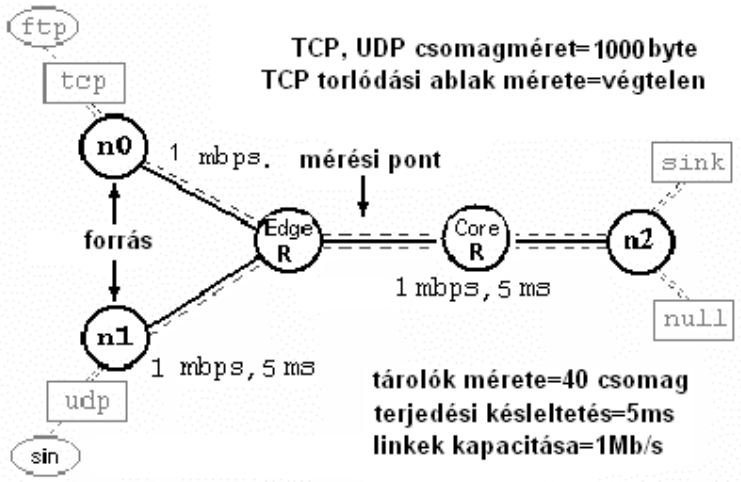
Az IP hálózatokkal szemben egyre nagyobb követelmény, hogy a hálózat képes legyen minőségi szolgáltatások biztosítására is. Az ilyen szolgáltatások kialakítása az IP hálózatokban leginkább a *Differentiated Service* [4] segítségével történik. A Diffserv lényege, hogy az adatfolyamokat osztályba sorolással különbözteti meg, és a különböző minőségi osztályba tartozó csomagokat eltérően kezeli.

A forgalom osztályozása mellett a minőségi osztályok kialakítása is a hálózati eszközökben (*router*) történik olyan aktív tároló kezelő mechanizmusok alkalmazásával, melyek dinamikusan képesek kiszolgálni a forgalmat és megelőzni a torlódás.

Ez a cikk kifejezetten arra koncentrál, hogy bemutassa a manapság legelterjedtebb QoS (Quality of Service) technikák hatását a TCP adaptivitására.

2. Adaptációs karakterisztika

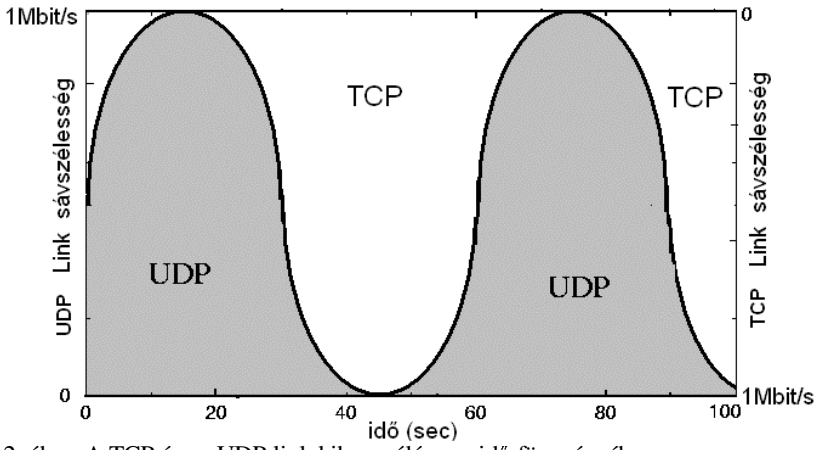
Az adaptáció méréséhez megfelelő módszernek bizonyult egy korábban publikált [2] karakterisztika, mely megalkotásakor arra törekedtek, hogy egyértelműen illusztrálja az adaptációt. Ezt tartva szem előtt, a szimuláció során a TCP-t egy alapvető háttérforgalom és hálózati topológia (1. ábra) mellett vizsgálták.



1. ábra A mérések során szimulált hálózat topológiája

A speciális háttérforgalom egy nem adaptív, UDP folyam, melynek sebességváltozását egy konstans amplitúdójú, adott *f* frekvenciájú szinuszhullámmal közelítettek, és amely így átlagban a link kapacitásának felét használja. Egy ilyen forgalom spektruma egyetlen egy frekvenciakomponens az *f* frekvencián.

Ha a TCP adaptálódik egy ilyen *f* frekvenciájú, szinuszosan változó sebességű háttérforgalomhoz, akkor egy idő után a TCP is egy *f* frekvenciájú szinuszhullámhoz fog hasonlítani, csak ellentétes fázisban (2. ábra). Ekkor a TCP spektrumában is szignifikánsan megjelenik az *f* frekvencián a tüske. Ismerve a két forgalom spektrumát az adaptáció mértéke kifejezhető a TCP és az UDP folyam *f* frekvencián lévő frekvencia komponensek hányadosaként, amely egy tökéletes adaptáció esetén 1.



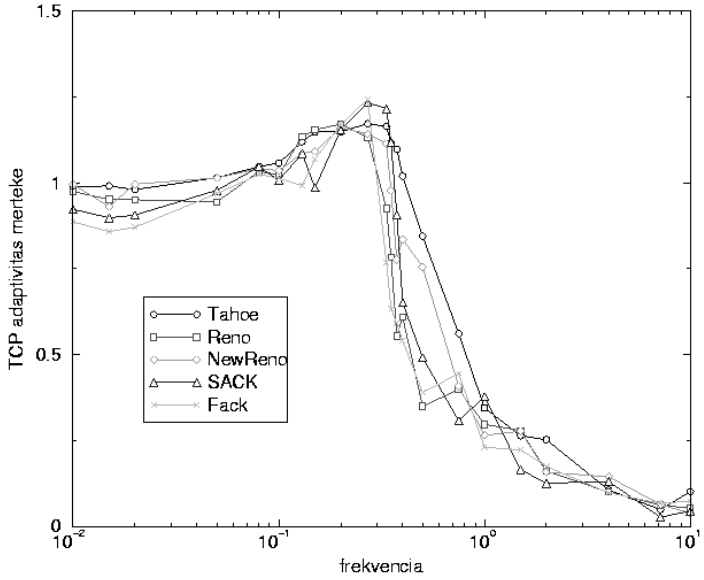
2. ábra A TCP és az UDP link kihasználása az idő függvényében

Több különböző frekvenciájú szinuszosan változó háttérforgalommal kiszámolva ezt a hányadost, felrajzolható az adaptációs karakterisztika (3. ábra), mely a háttérforgalom frekvenciájának a függvényében ábrázolja az adaptáció mértékét.

Több TCP verzió esetén is megvizsgálták az adaptációt különböző frekvenciájú háttérforgalmak mellett, és a 3. ábrán látható, hogy adaptivitás tekintetében elhanyagolható különbség van a verziók között.

Az adaptációs karakterisztikán megfigyelhető, hogy alacsony frekvenciájú háttérforgalom esetén ($f_{\text{háttér}}$ <0. 1Hz) a TCP szinte tökéletesen adaptálódik, azaz a hányados közel 1. Magasabb frekvenciákon ($f_{\text{háttér}}$ >0. 3Hz) a TCP adaptivitása fokozatosan megszűnik, nem tudja követni a háttérforgalom egyre gyorsabb hullámzásait.

A karakterisztikán az is jól látható, hogy a köztes frekvencia tartományban (0. 1Hz <$f_{\text{háttér}}$ <0. 3Hz) az arány meghaladja az egyet. Ebből arra lehet következtetetni, hogy a TCP ezeken a frekvenciákon sokkal agresszívabb, és a háttérforgalmat részben ki is tudja szorítani a tárolóból. Ezen frekvencia tartományra a továbbiakban *kritikus frekvencia tartomány*-ként fogunk hivatkozni.



3. ábra Az adaptációs karakterisztika

A cikkben ezt a karakterisztikát fogjuk vizsgálni, és azért, hogy csak a QoS beállítások hatása legyen érezhető az adaptáción, a méréseink során is ezt az alap hálózati topológiát (1. ábra) és szinuszosan változó háttérforgalmat használtuk fel.

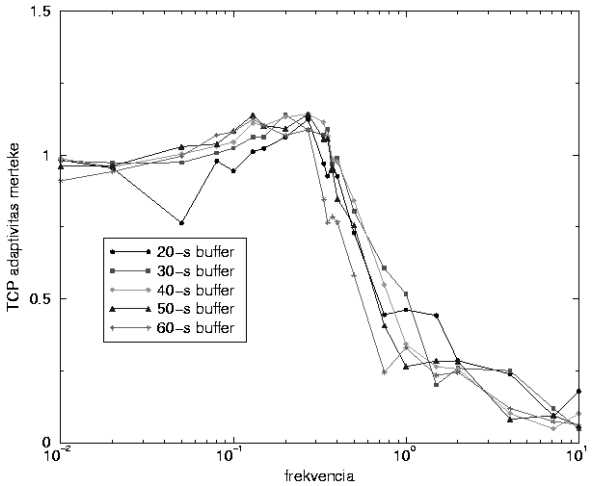
3. QoS technikák

Mint már az említettük minőségi szolgáltatást (QoS) IP alapú hálózatokban leginkább *Differentiated Service* [4] segítségével oldják meg, ahol a forgalmak osztályozása és a minőségi osztályok kialakítása a hálózati routerek feladata.

Méréseink elvégzésekor egyértelműen adódott, hogy a folyamokat a szállítás rétegbeli protokolljaik alapján fogjuk osztályozni, és ennek megfelelően két forgalmi osztályt fogunk kialakítani külön a TCP és külön az UDP forgalom számára.

Az [1] cikkben az adaptációs karakterisztika megalkotásakor a TCP és UDP folyam megkülönböztetés nélkül osztozik egy fix méretű, 40 csomag befogadásra képes, közös, tárolón, mely mindaddig fogadja a csomagokat, míg van szabad kapacitása. Az ilyen hagyományos tároló elnevezése az irodalomban: „Drop Tail”.

Egy forgalmi osztály kialakításakor egyáltalán nem mindegy, hogy mekkora a hozzá tartozó tárolóméret, ezért először megvizsgáltuk, hogy mennyire befolyásolja a TCP adaptivitásának a mértékét, ha az eredeti hálózati konfigurációban megváltoztatjuk a közös tároló befogadó képességét (4. ábra).



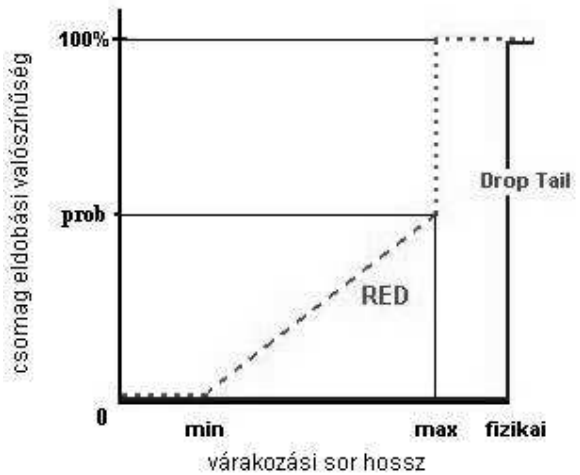
4. ábra A közös tárolóméret nagyságának hatása a TCP adaptivitására

Egyértelműen látható, hogy a TCP verziókhoz hasonlóan, az adaptáció ilyen hálózati konfiguráció mellett csaknem független a tároló méretétől is. Ennek tudatában a további méréseinket is az eredeti konfigurációhoz hasonlóan 40-es tároló mérettel végeztük.

Random Early Detection (RED)

Legyen a következő esetben is közös a két folyam tárolója, de most definiáljunk egy tároló kezelő mechanizmust is. A legismertebb és leginkább elterjedt tároló kezelő a RED, melyet 1990-es évek elején fejlesztett ki Sally Floyd és Van Jacobson [5], azért hogy a hálózatban kialakult torlódások kezelésére egy kevésbé reaktív, inkább megelőző módszert adjon.

A RED lényege, hogy egy hagyományos, „Drop Tail" tárolótól eltérően véletlenszerűen dobálja el a csomagokat a sorból, attól függően, hogy a tároló mennyire telített. A csomag eldobás intenzitásához definiálni kell egy RED profilt (5. ábra) a tárolóra.

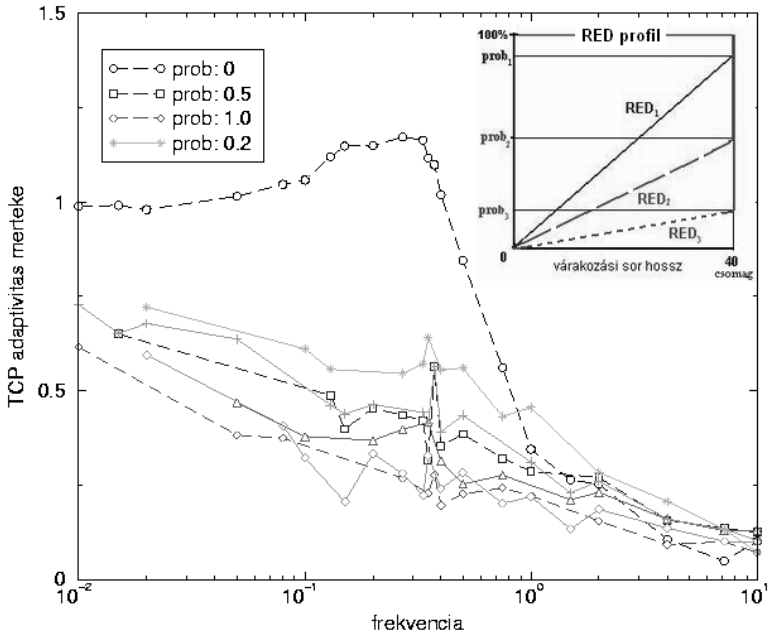


5. ábra Karakterisztika a RED agresszivitásának meghatározásához

A RED kifejezetten adaptív protokollokra alkalmazható, mivel a véletlenszerű csomageldobással – mely a nyugta elmaradását eredményezi – a forrás érzékeli, hogy a router kezd túlterhelődni, s hogy a torlódás kialakulását elkerülje, csökkenti az adási sebességét.

Az 6. ábra szemlélteti, hogy mi történik a TCP adaptációjával, ha RED-et definiálunk a közös tárolóra. A RED profil az ábra jobb felső sarkában illusztrálja, hogy az egyes mérések során egyre meredekebb eldobási valószínűség profilt definiáltunk a sor teljes hosszára.

Látható, hogy már a legenyhébb RED profil is lerontja a TCP azon képességét, hogy illeszkedni tudjon a háttérforgalomhoz.



6. ábra RED hatása a TCP adaptivitására

Magában a RED mechanizmust nem vizsgáljuk tovább, mert az amúgy sincs hatással a nem adaptív, UDP háttérforgalomra, valamint ha minőségi szolgáltatást akarunk biztosítani, akkor az adatot szállító TCP és a valós idejű UDP folyam – még ha egy közös sorba kerül is – nem fog ugyanolyan elbánásmódban részesülni.

Weighted Random Early Detection (WRED)

Tehát, ha minőségi szolgáltatást akarunk nyújtani, mindenképp meg kell különböztetni a TCP forgalmat a valós idejű háttérforgalomtól. Erre jól alkalmazható a RED egy kibővített változata a WRED [6], mely segítségével ugyanazon tárolóban különböző RED profilokat rendelhetünk hozzá különböző folyamokhoz, így egy közös tárolóban minden egyes folyamnak más lehet a csomagvesztése azonos torlódási szinten.

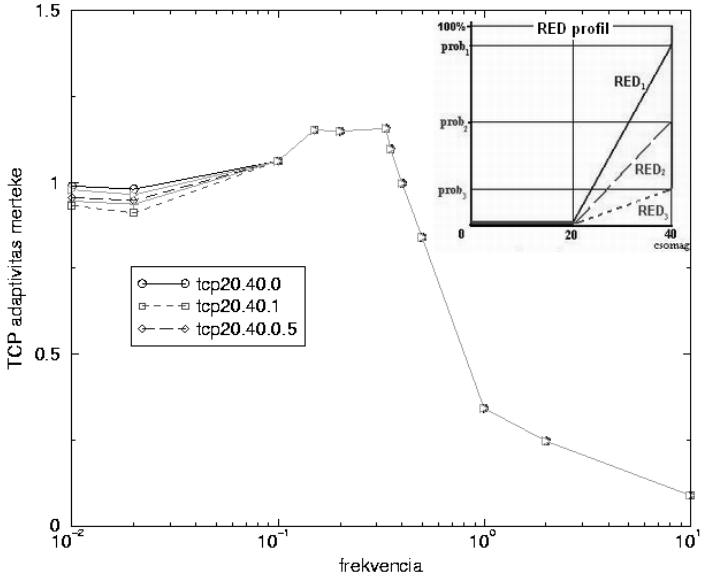
A WRED alkalmazásával sikerült úgy módosítani az alapkonfiguráción, hogy a két forgalom ugyan egy közös tárolóba került, de mégis meg van különböztetve, azaz a kiszolgálásuk különböző minőségi paraméterekkel történik.

Ezt a megkülönböztetést felhasználva, úgy állítjuk be a buffer kezelőt, hogy az UDP kiszolgálása a hagyományos „Drop Tail” tároló szerint történjen, és csak a TCP folyamra legyen értelmezve mindig más és más RED profil.

A továbbiakban szemléltetünk két esetet a WRED alkalmazására.

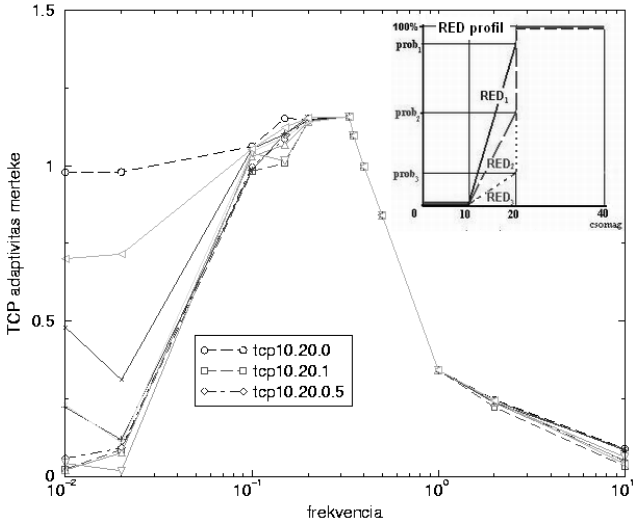
Elsőször rendeljük a TCP szolgáltatási osztályához egy *enyhe* RED profilt, mely az első csomagot csak akkor dobja el, ha a tároló már félig megtelt (7. ábra).

A 7. ábrán látható, hogy a TCP adaptivitása robosztus, nem hat rá ez a WRED beállítás sem.



7. ábra WRED hatása a TCP adaptivitására (enyhe RED profil a TCP számára)

Ezek után megnéztük, hogy mennyire tud adaptálódni a TCP a hullámzó háttér forgalomhoz, ha közben még egy agresszív RED profil is definiálva van a TCP forgalomra (8. ábra).



8. ábra WRED hatása a TCP adaptivitására (agresszív RED profil a TCP számára)

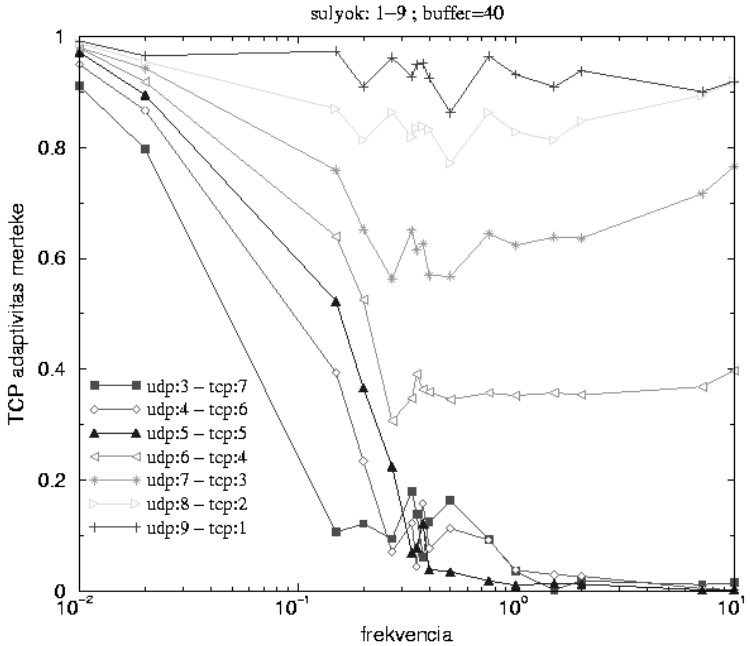
A karakterisztikából kiderül, hogy ez az agresszív RED profil is csak ott tudott hatást gyakorolni a TCP adaptivitására, ahol eredeti esetben is tökéletes volt az adaptáció. Meglepő, hogy ez a beállítás sem hat egyáltalán a TCP adaptációjára a kritikus frekvencia tartományon, mivel ugyanúgy kiszorítja az UDP-t a linkről. Magasabb frekvencia eseten pedig a RED semmilyen hatást nem tud kifejteni az amúgy is adaptálódni képtelen TCP-re.

Weighted Fair Queuing (WFQ)

Eddig olyan mérési eredményeket mutattunk be, ahol a TCP és az UDP háttérforgalom közös várakozási soron osztozott. QoS kialakításakor – még ha WRED-el meg is vannak különböztetve a forgalmak – ez csak egy kényszermegoldás. Garantált minőségi szolgáltatás biztosításakor tipikusan külön sorba rakják a TCP adatforgalmat és az UDP-t használó valós idejű forgalmat, s tároló ütemező algoritmussal határozzák meg a kiszolgálás sorrendjét.

Egy ilyen elterjedten alkalmazott tároló ütemező a WFQ [7], mely megakadályozza, hogy az alacsonyabb prioritású sorok kiéhezzenek, mivel a minden várakozási sor kap kiszolgálási időt a hozzá rendelt súly függvényében. A párhuzamos kiszolgálás miatt a WFQ a felszabadult szabad kapacitást szétosztja a többi tároló között.

Végezetül megvizsgáljuk, milyen hatással van TCP adaptivitására, ha a két forgalom külön tárolót kap, és a WFQ-t alkalmazva más-más súllyal kerülnek kiszolgálásra (9. ábra).



9. ábra WFQ hatása TCP adaptivitására

A 9. ábrán látható, hogy az UDP háttérforgalom minél nagyobb súllyal kerül kiszolgálásra, a TCP annál jobban adaptálódik. Az egész frekvencia tartományon a TCP szinte tökéletesen képes adaptálódni a háttérforgalomhoz, ha a kiszolgálási arány 9:1 az UDP javára.

A WFQ-val nem tettünk semmi mást, csak egy közös tároló helyett két külön sorba helyeztük a két forgalmat, s mégis az adaptáció teljesen megváltozott. Addig, míg egy tárolós esetben

azt tapasztaltuk, hogy bármilyen mechanizmust is használtunk, nagyobb frekvenciájú háttérforgalmak esetén a TCP képtelen volt adaptálódni, addig WFQ-val sikerült ezt elérni.

Az adaptációban lévő különbség abból adódik, hogy egy várakozási sor esetén a TCP a gyors háttérforgalom változások hatására egyre kevésbé tud két UDP csomag közé beékelődni. Két tároló esetén pedig mindig van TCP csomag a sor kimenetén, amit el tud küldeni, mikor nincs UDP csomag.

4. Összegzés

Ebben a cikkben megvizsgáltuk, hogy a TCP adaptációs tulajdonságai milyen mértékben függnek a minőségi osztályokat biztosító IP hálózatokban a legelterjedtebben használt aktív tárolókezelő eljárásoktól. Az eredményekből látható, hogy közös tároló használata esetén az eredeti "Drop Tail" megoldáshoz képest a RED vagy WRED esetén a beállításoktól függően módosulhat az adaptációs karakterisztika. A külön tárolót alkalmazó WFQ esetén az eredetinél is jobb, szinte minden frekvencián ideálisan adaptálódó TCP forgalom beállítása is lehetséges.

A kutatásaink eredménye és a további vizsgálataink alapján a célunk az, hogy pontosan megértsük a TCP adaptációs mechanizmusát és ennek birtokában forgalomelméleti és teljesítményanalízisbeli összefüggéseit feltárjuk. Továbbá célunk az is, hogy pontos választ adhassunk a régóta megválaszolatlan kérdésre: miért mérünk az Internet minden pontján fraktális jellemzőket?

Irodalom jegyzék

[1] **A. Veres, Zs. Kenesi, S. Molnár, G. Vattay,** TCP's Role in the Propagation of Self-Similarity in the Internet, appear in Computer Communications, Special issue on Performance Evaluation of IP Networks and Services.

[2] **A. Veres, Zs. Kenesi, S. Molnár, G. Vattay,** On the Propagation of Long-Range Dependence in the Internet, ACM Computer Communication Review, Vol. 30, No. 4, pp. 243-254, October, 2000.

[3] **Gary R. Wright, W. Richard Stevens**, TCP/IP Illustrated, Volume 2 The Implementation, Addison-Wesley Publishing Company 1995

[4] **S. Blake, D. Black, M. Carlson,** An Architecture for Differentiated Services, Request for Comments 2475, December, 1998

[5] **Sally Floyd**, **Van Jacobson**, Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, 1993

[6] **Chuck Semeria,** Supporting Differentiated Service Classes: Active Queue Memory Management, Juniper Networks, White Paper, 2002, http://www.juniper.net

[7] **H. Zhang,** Service Disciplines For Guaranteed Performance Service in Packet Switching Networks, Proceedings of the IEEE, 83(10), Oct 1995.