



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

ÚJ MÓDSZEREK HATÉKONY ADATÁTVITEL BIZTOSÍTÁSÁRA A JÖVŐ HÁLÓZATAIBAN

Ph.D. tézisfüzet

Szerző:

Móczár Zoltán

Témavezető:

Dr. Molnár Sándor

egyetemi docens

Budapest

2015

1. Bevezetés

Az Internet megjelenése óta az adatforgalom torlódását egy olyan nemkívánatos jelenségként tartják számon, amelyet a hálózat stabil működése érdekében mindenképpen el kell kerülni. Torlódás akkor lép fel, amikor egy erőforrás iránt mutatkozó igények összessége meghaladja annak kapacitását, ami a kommunikációs hálózatokban gyakran jelentős teljesítménycsökkenéshez vezet. A probléma megoldásaként 1981-ben kidolgozták a *Transmission Control Protocol (TCP)* [1] adatátviteli eljárást, amely különböző megfigyelések alapján képes a küldési sebességet megfelelően beállítani, és ezáltal megelőzni a torlódás kialakulását. A TCP által alkalmazott zárthurkú torlódásszabályozás sikeres megközelítésnek bizonyult, az évtizedek során ugyanakkor számos verziója jelent meg, amelyekkel a hálózatok folyamatos változása által támasztott új követelményeknek igyekeztek megfelelni [2]. Az elmúlt években azonban nyilvánvalóvá vált, hogy a TCP inkrementális módosításának gyakorlata nem tudja követni a technológiák és alkalmazások rohamos fejlődését, valamint a növekvő felhasználói igényeket.

Egyértelműen látható, hogy az egyre népszerűbb paradigmák, mint például a cloud computing vagy a software-defined networking, továbbá a közeljövőben megjelenő 5G mobilhálózatok és az Internet of Things sokkal hatékonyabb, a napjainkban alkalmazottól *teljesen eltérő elven működő adatátviteli mechanizmusokat* igényelnek majd. Ezeket a trendeket figyelembe véve természetesen adódik a kérdés: a megbízható adatátvitel biztosításához valóban elengedhetetlen a torlódásszabályozás alkalmazása? Bár a kutatói közösség számára egyre sürgetőbbé válik megadni a választ, mégis alig néhány cikk foglalkozik ezzel a komoly kihívást jelentő feladattal. A [3] publikáció szerzői például amellett érvelnek, hogy a megfelelő teljesítmény eléréséhez nem feltétlenül szükséges elkerülni a torlódást, még egy ilyen értelemben nem konzervatív adatátviteli protokoll is teljesíthet jobban, mint a TCP. 2007-ben a GENI (Global Environment for Network Innovations) nevű szervezet közzétett egy kutatási tervet [4], amelyben a TCP torlódásszabályozási algoritmus helyett hatékony hibajavító kódok használatát javasolják a csomagvesztés kezelésére. A felvetés létjogosultságát erősíti meg az a nemrég megjelent tanulmány [5] is, amely szerint a legtöbb esetben akkor sem omlik össze a hálózat torlódás következtében, ha a végpontok nem szabályozzák a küldési sebességet. Annak feltárása azonban, hogy az Internet működhet-e hatékonyan a TCP kulcsfontosságú funkciója nélkül, alapos kutatómunkát igényel.

2. Kutatási célkitűzések

Értekezésemben egy olyan adatátviteli paradigmával foglalkozom, amely megbízható kommunikációt tesz lehetővé az Internet végpontjai között torlódásszabályozás alkalmazása nélkül. Kutatási célként új, hatékony adatátviteli módszerek kidolgozását és széleskörű vizsgálatát tűztem ki.

Főbb eredményeim az alábbiak szerint foglalhatók össze:

- Disszertációm első részében egy új, digitális szökőkútkódoláson alapuló kommunikációs paradigmát vizsgálok, amely egy adatátviteli mechanizmusra (*Digital Fountain based Communication Protocol, DFCP*) és egy olyan architektúrára épül, ahol az igazságos sáv szélesség-megosztást a végpontok helyett az útválasztók biztosítják. A tervezési és működési alapelveket részletesen ismertetem, valamint kiemelem a megközelítés főbb előnyeit.
- A második részben a DFCP és a TCP átfogó teljesítményelemzésével foglalkozom, amelynek eredményei egy többplatformos környezetben végzett csomagszintű szimulációkon és teszhálózati méréseken alapszanak. A vizsgálatok különböző hálózati topológiák és jellemzők mellett tárják fel a javasolt megoldás legfontosabb tulajdonságait, mint például az elérhető adatátviteli sebesség, a tárolóigény, a folyamatok átvitelének hatékonysága, az igazságosság, a hálózatkihasználtság és a skálázhatóság.
- Mivel a mai hálózatok forgalma rendkívül dinamikus, a harmadik részben a két adatátviteli paradigma működését folyamatosan változó forgalmi körülmények között tanulmányozom a stabilitás, a konvergencia, az érzékenység és a telítési idő jellemzőire fókuszálva.
- Végül javaslok egy mobilhálózatokra kidolgozott sáv szélességbecslő módszert, amely minimális hálózati terhelés mellett képes megbecsülni az elérhető sáv szélességet a felhasználó által generált adatforgalom kiaknázásával. A megoldás fő eleme egy, a maximális sebességű kiszolgálási periódusok meghatározására alkalmas algoritmus, amely a mobilhálózatokban gyakori és jelentős mértékű sáv szélesség-ingadozás ellenére pontos eredményt szolgáltat.

Úgy gondolom, hogy az értekezésemben bemutatott eredmények nagymértékben segíthetik az alternatív adatátviteli mechanizmusokkal foglalkozó kutatásokat, mivel rámutatnak arra a fontos tényre, hogy az adatkommunikáció torlódásszabályozás nélkül is lehetséges, és érdemes ebben az irányban további vizsgálatokat végezni.

3. Módszertan

A DFCP adatátviteli mechanizmusa a TCP-hez hasonlóan a Linux kernelben került megvalósításra. A javasolt paradigma alapos vizsgálatához méréseimet *szimulátorban* és *valós teszhálózatokon* is elvégeztem. *Szimulációs eszközként* a széles körben ismert ns-2 csomagszintű szimulátort [6] használtam a Network Simulation Cradle (NSC) [7] kiegészítő modullal, amelyet C++ nyelven átalakítottam úgy, hogy képes legyen a DFCP kernel implementációját futtatni. A mérésekhez különböző teszhálózatokat hoztam létre *laboratóriumi*, illetve a távoli *hálózati emulációt támogató Emulab környezetben* [8]. Az eredmények és a megállapítások helyességének biztosítása céljából az említett három platformot körültekintően validáltam, ami az irodalomban ritkának számít a szállítási protokollok területén. Az elérhető sávszélesség becslésére kidolgozott algoritmust 3G mobilhálózatból származó forgalmi adatokon értékeltem, amelyeket egy *realisztikus forgalomemulátor* [9] segítségével állítottam elő.

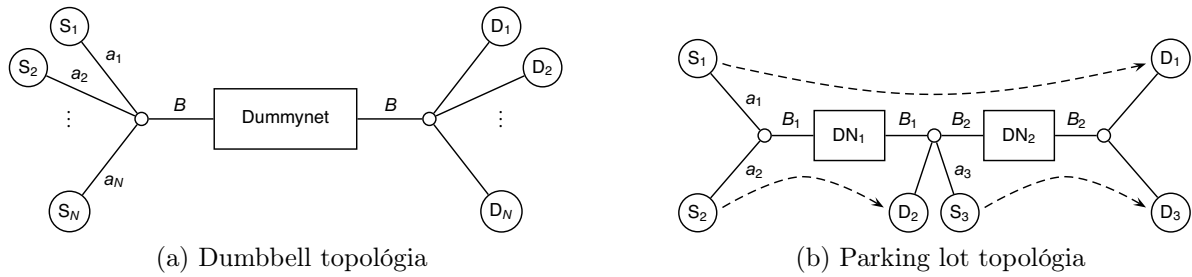
4. Új eredmények

4.1. Tesztkörnyezet új transzport protokollok vizsgálatához

Bár a hálózati protokollok, illetve algoritmusok elemzésével foglalkozó kutatások jelentős része még napjainkban is *kizárólag szimulációs vagy teszhálózati eredményekre* támaszkodik, számos tény igazolja ezek együttes meglétének szükségességét, különösen egy teljesen új koncepció alkalmazásánál. Az önmagukban szimulációs eredményekre alapozott megállapítások félrevezetőek lehetnek, hiszen a szimulációs környezetek ideális körülményeket biztosítanak a teszteléshez, amelyekben számos, valós hálózatokban előforduló hatás nem érvényesül [10, 11]. A teszhálózatokban végzett méréseknél ugyanakkor gondot jelent, hogy az eredményeket nagymértékben befolyásolhatják a felhasznált hardverkomponensek, a komplex hálózatok vizsgálata nem vagy nehezen kivitelezhető, valamint a mérési eredmények sok esetben nem megismételhetők [12, 13].

1. téziscsoport. *Kialakítottam egy többplatformos tesztkörnyezetet, amely lehetővé teszi a torlódásszabályozás, illetve a digitális szökőkútkódolás elvén működő transzport protokollok megbízható és konzisztens teljesítményelemzését [C3, J1, J2].*

1.1. tézis. *A transzport protokollok működésének és tulajdonságainak vizsgálatára laboratóriumi teszhálózatot építettem ki. Megterveztem és összeállítottam a protokollok jellemzőinek tanulmányozására legalkalmasabb hálózati topológiákat. Lehetővé tettem az adatátvi-*



1. ábra. Vizsgált hálózati topológiák

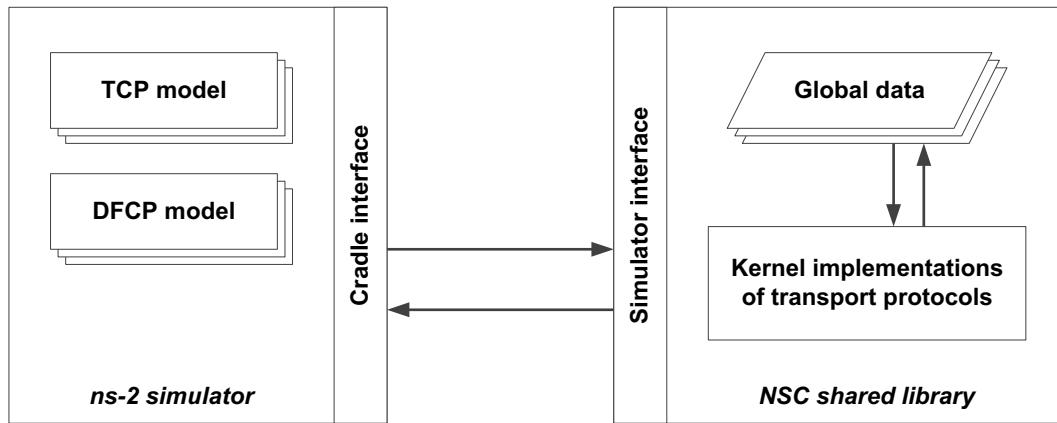
teli eljárások teljesítményelemzését eltérő késleltetésű, illetve különböző veszteségmodellek alapján működő környezetekben.

1.2. tézis. *Teszthálózatot állítottam össze a kutatási célokra létrehozott és fenntartott Emulab hálózatemulációs környezetben komplex hálózati topológiák vizsgálata céljából. Olyan több csomópontból és összeköttetésből álló hálózati topológiákat alakítottam ki, amelyek lehetővé teszik a protokollok működésének alaposabb megértését a valós hálózatokat jobban reprezentáló körülmények között.*

1.3. tézis. *Elkészítettem egy szimulációs környezetet a laboratóriumi körülmények között nem kivitelezhető mérési esettanulmányok megvalósítására. Lehetővé tettem az új, torlódásszabályozást nem alkalmazó protokoll elemzését ns-2 szimulációs környezetben az NSC modul módosításával, amely eddig kizárólag TCP alapú transzport protokollok vizsgálatára kínált megoldást.*

Az adatátviteli mechanizmusok teljesítményelemzését az 1. ábrán látható *hálózati topológiákon* hajtottam végre, amelyeket a szakirodalomban elterjedten használnak protokollok és algoritmusok vizsgálatára [14]. A dumbbell topológia (1a. ábra) lehetővé tette a transzport protokollok működésének és alapvető jellemzőinek tanulmányozását különböző hálózati beállítások (sávszélesség, késleltetés, csomagvesztés, tárolóméret, stb.) mellett. Az 1a. ábrán szereplő hálózati elrendezés továbbá lehetőséget kínált a sávszélességmegosztás igazságosságának (fairness), valamint az adatátviteli eljárások skálázhatóságának vizsgálatára versengő folyamatok esetén. A parking lot topológia (1b. ábra) segítségével a protokollok viselkedését több szűk keresztmetszeti összeköttetést (bottleneck link) tartalmazó hálózatokban elemeztem.

Az *integrált tesztkörnyezet* a különböző platformok előnyös tulajdonságainak kihasználásával egyrészt lehetőséget ad összetett vizsgálatok elvégzésre, másrészt megbízható és konzisztens eredményeket képes szolgáltatni, ami elengedhetetlen a helyes következtetések megfogalmazásához. A Network Simulation Cradle (NSC) az ns-2 szimulátor kiegészítő modulja, amely különböző operációs rendszerek (pl. BSD és Linux) hálózati funkcióinak



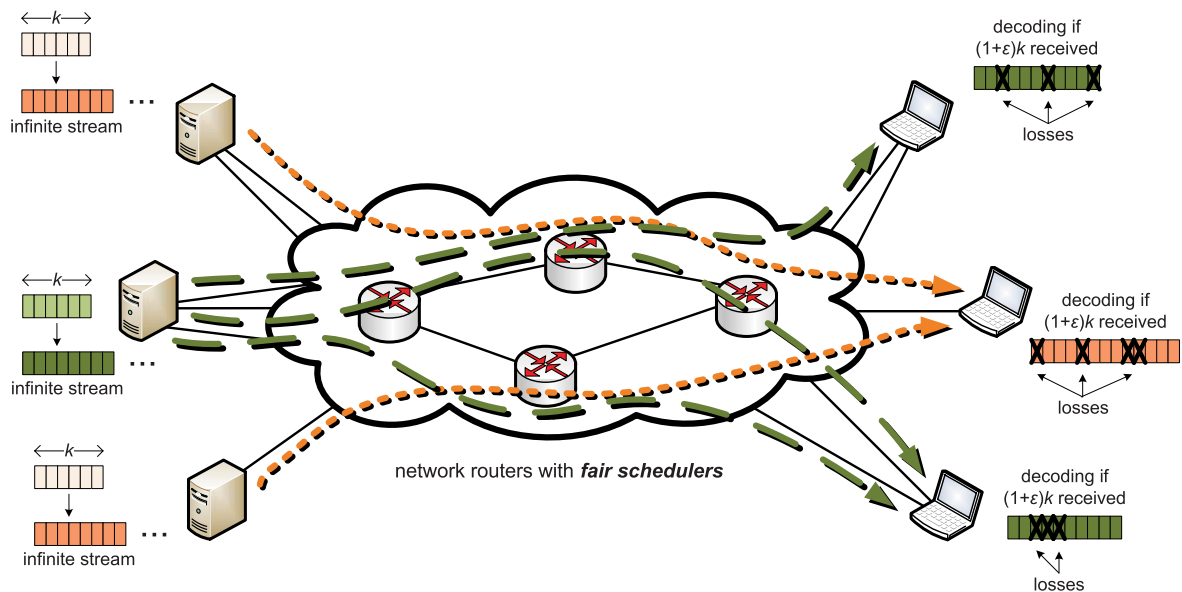
2. ábra. A DFCP-vel kompatibilis integrált szimulációs keretrendszer

szimulációját támogatja közvetlenül a kernel kódból. Az NSC nagy pontosságú eredményeket képes előállítani, de kizárólag TCP verziók és új TCP alapú transzport mechanizmusok vizsgálatát teszi lehetővé. Mivel a bemutatott adatátviteli eljárás a TCP-hez képest teljesen eltérő elven alapszik, számos protokoll-specifikus módosítást kellett elvégezni a DFCP forráskódjának integrációjához. A szimulációs keretrendszer főbb elemeit a 2. ábra szemlélteti. A transzport protokollok modelljei az ns-2 szimulátorban definiálhatók az összes kapcsolódó paraméterrel együtt. A szimulátor két komponense (ns-2 és NSC) egy közös interfészen keresztül kommunikál, amit egy C++ megosztott programkönyvtár reprezentál. Egy adott művelet végrehajtásakor az ns-2 először továbbítja a kérést az NSC-ben definiált protokoll-specifikus módszernek, amely végül meghívja a megfelelő kernel függvényt. A szimuláció során egy adott protokollból több példány is futtatható, mivel a hozzájuk tartozó adatstruktúrákat az NSC külön kezeli.

4.2. Digitális szökőkútkódoláson alapuló kommunikáció elemzése

Az értekezésemben bemutatott adatátviteli paradigma kulcsfontosságú komponense a DFCP protokoll, amely az elveszett csomagokat ún. *szökőkútkódolás* [15] használatával állítja helyre. A szökőkútkódok különleges előnyét az adja, hogy segítségével az eredeti üzenetből potenciálisan végtelen hosszúságú kódfolyam készíthető, amelynek tetszőleges, az eredeti üzenetnél alig nagyobb méretű részéből a küldött információ visszafejthető. A DFCP implementációja Raptor kódolást [16] alkalmaz, amely lineáris kódolási és dekódolási komplexitásának köszönhetően a jelenleg ismert leghatékonyabb kódolási eljárás.

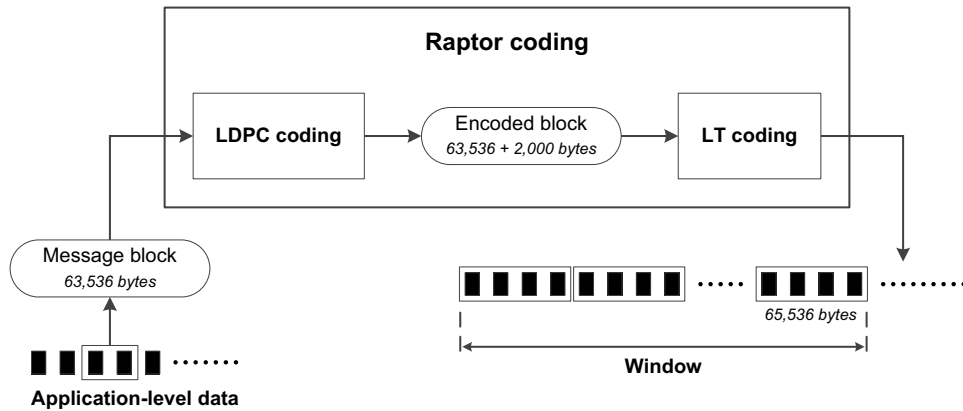
A *DFCP adatátviteli mechanizmusára épülő hálózati architektúra* a 3. ábrán látható. Tételezzük fel, hogy a küldő egy k hosszúságú üzenetet szeretne eljuttatni a vevőhöz, és a végpontokon az adatátvitel maximális sebességgel, torlódásszabályozás nélkül történik. Raptor kódolás segítségével a küldő $\epsilon > 0$ redundancia hozzáadásával egy elméletben



3. ábra. A DFCP adatátviteli mechanizmusára épülő hálózati architektúra

tetszőleges hosszúságú, kódolt adatfolyamot képes előállítani, a vevő oldalon pedig az eredeti üzenet nagy valószínűséggel sikeresen dekódolható, ha ennek bármely, összességében legalább $\lceil (1 + \epsilon)k \rceil$ méretű darabja megérkezik. Mivel valós alkalmazások esetén ϵ tipikus értéke néhány százalék, így ez a megoldás hatékony kommunikációt tesz lehetővé a csomópontok számára, miközben a hálózati erőforrások teljes kihasználtsága biztosított. A maximális sebességű adatküldés ugyanakkor könnyen egyes folyamatok kiéheztetéséhez vezethet, ezért az igazságos sáv szélesség-megosztást az útválasztókban működtetett *fair ütemezők* segítségével valósíthatjuk meg. A hálózati eszközökben számos ilyen ütemező algoritmus (pl. DRR [17]) jelenleg is megtalálható és konfigurálható, amelyek skálázható megoldást nyújtanak [18] az erőforrások megosztására.

Az adatok kódolásának és küldésének folyamatát a 4. ábra szemlélteti. A DFCP az alkalmazási rétegből érkező adatokat meghatározott méretű blokkokba szervezi, majd ezeken először LDPC (Low-Density Parity-Check) kódolást [19] hajt végre. Az LDPC kódolás az ún. előkódolás funkcióját tölti be, amely az eredeti adatot redundáns bájtokkal egészíti ki. Az ilyen módon kódolt blokkok képezik a következő lépés, az LT (Luby Transform) kódolás [20] bemenetét. Az LT kódoló elméletben végtelen hosszú, kódolt adatfolyamot képes előállítani, a gyakorlatban azonban ezt csak véges méretű egységekben lehet átküldeni a vevőnek. A DFCP erre a célra *csúszóablakos folyamvezérlést (flow control)* alkalmaz, amely meghatározza, hogy egyidejűleg maximum hány kódolt blokk kerülhet kiküldésre nyugtázás nélkül. Az alkalmazott mechanizmus elsődleges célja a fogadó fél elárasztásának elkerülése, amely hasonlít a TCP-nél használt megoldásra, azonban attól eltérő szerepet



4. ábra. Az adatok kódolásának és küldésének folyamata

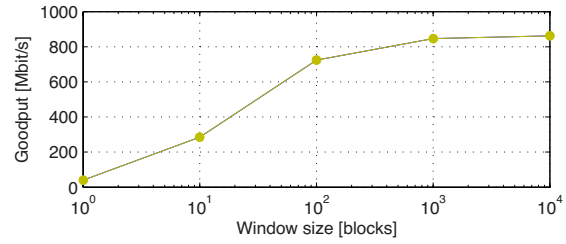
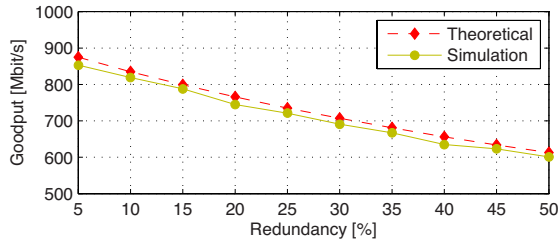
tölt be. A csúszóablak méretét a gyakorlatban olyan értékre célszerű választani, hogy az ne korlátozza a protokoll adatátviteli sebességét.

2. téziscsoport. *Egy új, torlódásszabályozást nem alkalmazó transzport protokoll, a DFCP (Digital Fountain based Communication Protocol) működését és alapvető tulajdonságait többplatformos tesztkörnyezetben tanulmányoztam [C3, C5, J1, J2].*

2.1. tézis. *Különböző hálózati topológiákon végzett vizsgálataimmal bebizonyítottam, hogy a DFCP prototípusa hatékony működésre képes olyan környezetben, ahol a csomagvesztési arány és a körülfordulási idő széles tartományban (0–50%, 0–500 ms) változhat. Megállapítottam, hogy a maximálisan elérhető hasznos adatátviteli sebesség független a csomagvesztés és a késleltetés mértékétől a valós hálózatokban gyakran előforduló értéktartományban (0–1%, 0–100 ms).*

2.2. tézis. *Megvizsgáltam, hogy a protokoll főbb paraméterei, a kódolási redundancia és az ablakméret hogyan befolyásolják az adatátviteli teljesítményt, valamint a generált forgalom jellemzőit. Megmutattam, hogy a hozzáadott redundancia arányát 50%-ig növelve az átviteli sebesség legfeljebb 33%-kal csökken. Igazoltam továbbá, hogy nagyobb ablakméret használata csomósabb forgalmat eredményez, ezért az optimális működés eléréséhez arra a legkisebb értékre kell beállítani, amelynél a folyamvezérlés korlátozó hatása már nem érvényesül.*

Transzport protokollok teljesítményének mérésére az irodalomban számos metrika ismert. Az egyik leggyakrabban alkalmazott jellemző a *throughput*, ami a küldőtől a fogadóig másodpercenként átvitt adatmennyiséget adja meg. Eltérő elven működő átviteli mechanizmusok hatékonyságának összehasonlítása esetén azonban a *throughput* helyett célszerűbb a *goodput* használata, mivel a *goodput* csak a hasznos adatokra vonatkozik,



(a) Átviteli sebesség a redundancia függvényében

(b) Átviteli sebesség az ablakméret függvényében

5. ábra. A DFCP paramétereinek hatása a teljesítményre

figyelman kívül hagyva a különböző protokollok fejléceit, az üzenethez adott redundanciát és a kódolásra fordított számítási kapacitást. A szökőkútkódoláson alapuló adatátviteli paradigma alkalmazásával elérhető hasznos adatsebesség az egyszerű dumbbell topológia (1a. ábra) esetén analitikusan meghatározható. Tegyük fel, hogy a szűk keresztmetszeti link sávszélessége c_B , amelyre N küldő csatlakozik rendre c_1, c_2, \dots, c_N kapacitású hozzáférési linkeken keresztül. Minden küldő pontosan egy folyamat hoz létre, azaz N konkurens folyamat verseng az elérhető sávszélességért. Feltételezve, hogy a hálózati csomópontokban fair ütemezők működnek, és a redundanciát ε jelöli, az i . folyamat sebessége a következőképpen számítható ki:

$$G_i = \begin{cases} \frac{c_B}{(1+\varepsilon_i)N} & \forall j : c_j \geq \frac{c_B}{N} \\ \frac{c_i}{1+\varepsilon_i} & c_i < \frac{c_B}{N} \\ \frac{c_B - \sum_{k=1}^N I_{\{c_k < \frac{c_B}{N}\}} c_k}{N} & \exists j : c_j < \frac{c_B}{N} \text{ and } c_i \geq \frac{c_B}{N} \\ (1+\varepsilon_i) \sum_{k=1}^N I_{\{c_k \geq \frac{c_B}{N}\}} & \end{cases}$$

A DFCP legfontosabb paramétereinek teljesítményre gyakorolt hatását az 5. ábra illusztrálja $\varepsilon = 0,05$ redundancia, illetve 1000 blokk méretű ablak használata esetén (ahol $N = 1$ és $c_B = c_1 = 1000$ Mbps). Megfigyelhető, hogy a hasznos adatátviteli sebesség a redundancia növelésével nem lineárisan csökken (5a. ábra). Napjainkban a leghatékonyabb Raptor kódok az eredeti adathoz már 5%-nál kevesebb redundancia hozzáadásával is nagy valószínűségű dekódolást tesznek lehetővé. Az ablakméret növelésével az átviteli sebesség addig nő, amíg el nem éri a maximális sebességet (5b. ábra). Nagyobb ablakméret használata mellett ugyanakkor a DFCP csomósabb forgalmat generál, így a gyakorlatban az ablakméretet arra a legkisebb értékre célszerű beállítani, amelyet tovább növelve már nem lehet gyorsabb adatátvitelt elérni. Más szóval azt a működési pontot kell megtalálni, ahol a folyamatvezérlési mechanizmus nem korlátozza a hasznos adatsebességet.

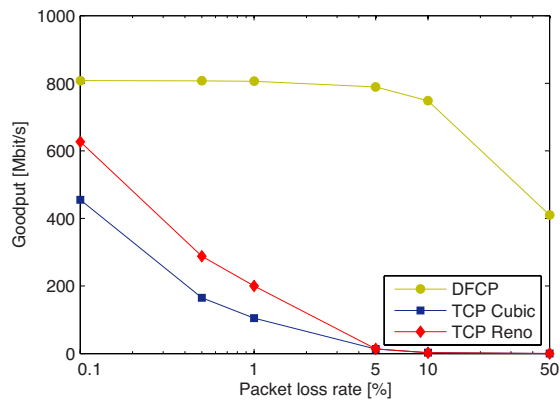
4.3. Adatátviteli paradigmák összehasonlító teljesítményelemzése

Napjainkban az Internet adatforgalmának több, mint 80%-át a TCP szállítja. A TCP egy kapcsolatorientált szállítási protokoll, amely megbízható adatátvitelt tesz lehetővé a hálózat végpontjai között. A TCP legfontosabb funkciója a torlódásszabályozás [21], amely egy adaptív mechanizmus révén képes az adatküldés sebességét megfelelően beállítani, és ezáltal megóvni a hálózatot az összeomlástól. A TCP gyökerei egészen 1981-ig nyúlnak vissza, amikor az IETF közzétette a hivatalos specifikációt a 793-as számú RFC-ben [1]. Az elmúlt mintegy három évtizedben a kutatói közösség komoly erőforrásokat fordított a TCP továbbfejlesztésére és optimalizálására, amelynek eredményeképpen számos, különböző környezethez kidolgozott verzió született [2].

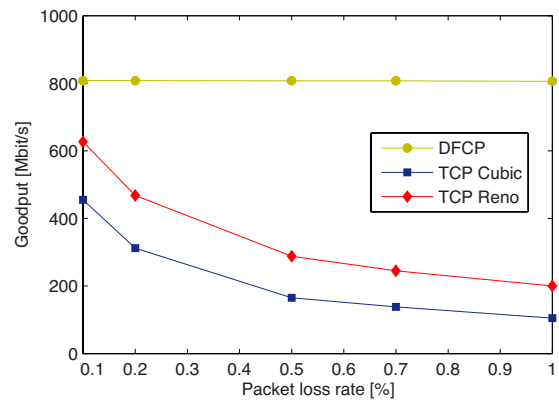
Disszertációmban a DFCP protokollt a legelterjedtebben használt *TCP Cubic* és a *TCP NewReno* verziókkal hasonlítom össze. A TCP Cubic [22] a Linux operációs rendszerek alapértelmezett transzport protokollja, a BIC TCP [23] továbbfejlesztett változata, amelyet eredetileg az ún. RTT unfairness probléma [24] megoldására terveztek. A TCP Cubic a torlódási ablak méretét egy köbös függvény szerint változtatja, amelynek köszönhetően képes stabil működést biztosítani és egyúttal skálázható megoldást nyújtani. A TCP NewReno [25] a TCP Reno [26] utódja, amelynek legfőbb hátránya, hogy nem képes hatékonyan kezelni a többszörös csomagvesztéseket, ezért ilyenkor jelentősen lecsökkenhet a teljesítménye. A NewReno úgy módosítja a Reno torlódásszabályozási algoritmusát, hogy a helyreállítás több csomag együttes elvesztése esetén is gyorsan megtörténhessen.

A szállítási protokollok működésének alapos megértéséhez elengedhetetlen átfogó teljesítményelemzésük. A *hálózati jellemzők*, mint például a csomagvesztés és a késleltetés értékei széles skálán mozoghatnak a különböző környezeteket tekintve, ezért teljesítményt befolyásoló hatásuk sokrétű vizsgálatot igényel. Az elmúlt évtizedben továbbá számos kutatás foglalkozott a *tárolók optimális méretezésének* kérdésével, amely a transzport protokollok hatékony működésében is kiemelt jelentőséggel bír. Napjainkban az útválasztók memóriáit jelentősen túlméretezik egy korábban lefektetett ökölszabály alapján [27], ami a végpontok közötti késleltetés növekedésének eredményeképpen gyakran a teljesítmény drasztikus csökkenéséhez vezet [28]. Teljesen optikai hálózatokban ugyanakkor technológiai korlátok miatt csak kisméretű tárolók használhatók [29], így különösen fontos, hogy az adatátviteli mechanizmus ilyen feltételek mellett is működőképes legyen.

A szolgáltatók többsége a hálózati erőforrások túlméretezésével igyekszik kiszolgálni a felhasználói igényeket a forgalmas órákat is figyelembe véve [30], ami azon a feltételezésen alapszik, hogy a minőséggel kapcsolatos problémák pusztán többletkapacitás biztosításával orvosolhatók. Ez a megoldás azonban sok esetben egyáltalán nem hatékony és a felhasználói élmény csökkenéséhez vezethet, vagy az adott környezetben egyszerűen



(a) Csomagvesztési arány tág intervallumban



(b) Realisztikus értéktartomány

6. ábra. A DFCP és a TCP működése veszteséges környezetben

nem kivitelezhető. A transzport protokollok által biztosított *kihasználtság* fontos metrika, amely segítséget nyújt a hálózatok megfelelő tervezésében és méretezésében. A hálózatkihasználtságot disszertációmban a szűk keresztmetszeti erőforrások (összeköttetések és tárolók) kapacitásának átlagos kihasználtságaként értelmezem.

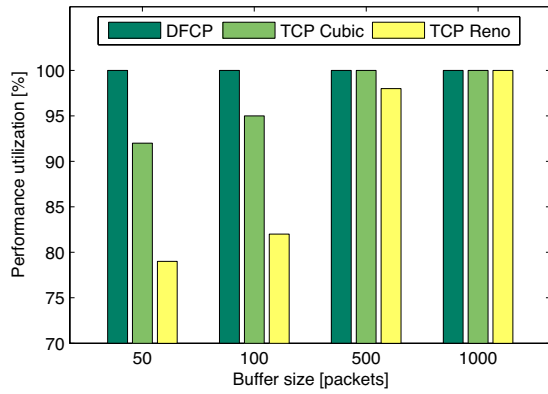
3. téziscsoport. *Összehasonlító elemzéssel feltártam, hogy a DFCP a TCP-hez viszonyítva milyen teljesítményt nyújt különböző hálózati környezetekben [C3, C4, C5, J1, J2].*

3.1. tézis. *Igazoltam, hogy a DFCP a veszteséges és nagy késleltetésű összeköttetések (>50 ms) esetén is magasabb adatátviteli sebességet biztosít, mint a TCP.*

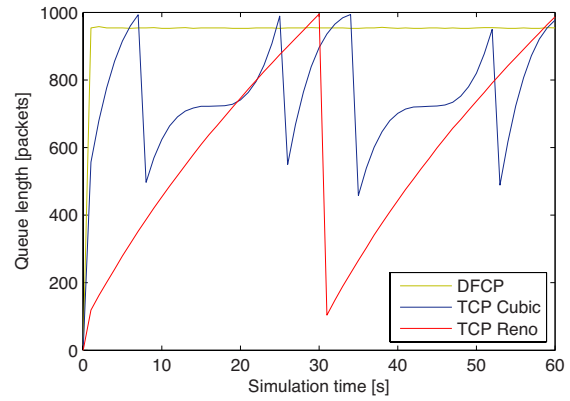
3.2. tézis. *Szimulációk segítségével bebizonyítottam, hogy a TCP-vel ellentétben a tároló mérete nem befolyásolja a DFCP teljesítményét, így kicsi tárolókkal (<100 csomag) is képes hatékonyan működni.*

3.3. tézis. *Megmutattam, hogy a DFCP átviteli mechanizmusa enyhe, a TCP torlódássa-bályozó algoritmusa viszont jelentős ingadozást eredményez a tárolókban kialakuló sorhossz tekintetében. Az ingadozás mértéke DFCP esetén tipikusan néhány százalék az átlagos sorhosszhoz viszonyítva, a TCP esetében azonban a sorban várakozó csomagok száma a tároló mérete által meghatározott teljes tartományban változhat.*

3.4. tézis. *Összetett hálózatokban végzett vizsgálataim alapján megállapítottam, hogy az esetek 90%-ában a DFCP 80%-nál nagyobb, míg a TCP 80%-nál kisebb hálózatkihasználtsággal működik. Rámutattam továbbá, hogy nagyon kicsi tárolók (0,01 BDP) alkalmazása mellett a DFCP-vel a TCP-hez képest legalább kétszer akkora kihasználtság érhető el.*



(a) A protokollok tárolóigénye



(b) A tárolók kihasználtsága az idő függvényében

7. ábra. A DFCP és a TCP tárolóigénye és a tárolók kihasználtsága

A DFCP és a TCP protokollok működését veszteséges környezetben a 6. ábra szemlélteti. A szökőkútkódoláson alapuló átviteli mechanizmus nagy előnye, hogy kismértékben érzékeny a csomagvesztésre még akkor is, ha az csomókban történik. A 6a. ábrán megfigyelhető a DFCP magas ellenálló képessége, amelynek eredményeképpen a teljesítmény alig csökken a csomagvesztési ráta tág intervallumában, míg a TCP verziók már néhány százalékos értéknél is közel működésképtelenné válnak. A 6b. ábra továbbá azt mutatja, hogy realisztikus arányt feltételezve a DFCP érzéketlenné válik a csomagvesztésre, így a TCP esetén tapasztalható sebesség-ingadozás elkerülhető, ami kiszámíthatóbb forgalmi jellemzőkhöz vezet.

A 7. ábra a DFCP és a TCP protokollok tárolóigényét, illetve a tárolók kihasználtságának változását mutatja 1 Gbps kapacitású, 10 ms késleltetésű link esetén. A 7a. ábra azt szemlélteti, hogy a protokollok teljesítménye hányad részére csökken egy adott tárolóméret használata mellett, ahhoz az esethez viszonyítva, amikor a tárolóméret nem lép fel korlátozó tényezőként. Láthatjuk, hogy 1000 csomag méretű tároló esetén minden protokoll maximális adatátviteli sebességet képes nyújtani, azonban a tárolóméret csökkentésével a TCP verziók teljesítménye jelentősen visszaesik. Ha például csak 50 csomag eltárolására van lehetőség, a TCP Cubic és a TCP Reno sebessége csupán a 92, illetve 79 százalékát éri el az adott hálózati linken biztosítható maximális értéknek. Ezzel szemben a DFCP esetén a kisméretű tárolók alkalmazása nem korlátozza a teljesítményt, ami jól illeszkedik a teljesen optikai hálózatok koncepciójához [29]. A 7b. ábrán a tárolók kihasználtsága figyelhető meg az idő függvényében. A sorhossz alakulását tekintve látható, hogy a DFCP a TCP verziókkal ellentétben nagyon rövid idő alatt feltölti a tárolót és átlagosan 95%-os kihasználtság mellett működik, míg a TCP Cubic és TCP Reno protokollok csupán a 75, illetve 58 százalékát használják ki a rendelkezésre álló erőforrásnak. Az is megfigyelhető,

hogy a sorhossz a TCP verziók esetén jelentős ingadozást mutat, a DFCP-nél azonban csak kismértékű oszcilláció tapasztalható.

4.4. Szállítási mechanizmusok működése dinamikus környezetben

Az utóbbi évtizedek során a hálózati forgalom jellemzői jelentősen átalakultak a technológiák fejlődésének és a különféle alkalmazások megjelenésének következtében. Napjaink Internete egy nagyméretű, rendkívül dinamikus hálózat, amelyben gyakran fordulnak elő a topológiát és az elérhető sáv szélességet befolyásoló hirtelen változások. Míg a TCP vizsgálatával foglalkozó tanulmányok túlnyomó része kizárólag statikus környezetben végzett teljesítményelemzésen alapszik, addig más kutatók fontosnak tartják a protokollok működésének feltárását gyorsan változó körülmények között is [31, 32]. Disszertáciomban *két alapjaiban eltérő adatátviteli megoldás dinamikus működését* vizsgálom: a TCP Cubic-ot FIFO sorkezelő algoritmus (torlódásszabályozás alapú architektúra, *Congestion Control based Architecture, CCA*), illetve a DFCP-t DRR ütemező (szökőkútkódolás alapú architektúra, *Digital Fountain based Architecture, DFA*) használatával. Kutatómunkám során csomagszintű szimulációkat végeztem a *stabilitás*, a *konvergencia*, az *érzékenység* és a *telítődési idő* jellemzőire fókuszálva.

A heterogén alkalmazások által generált hálózati forgalom egymással versengő folyamatok sokaságát eredményezi, amelyek különböző útvonalakon haladnak keresztül a küldő és a fogadó fél között. Ezeknek a folyamatoknak az adatátviteli sebessége jelentős fluktuációt mutat, mivel a jelenleg alkalmazott, torlódásszabályozáson alapuló mechanizmus nem képes megfelelő gyorsasággal igazodni a változó hálózati viszonyokhoz. A *stabilitás* ezért fontos tulajdonság mind a forgalomtervezés, mind pedig a felhasználói élmény szempontjából [33]. Az átviteli sebesség folyamatos változása általában a sorhossz ingadozásához, végül a tároló túlcsoportolásához vezet. Ez a nemkívánatos jelenség gyakran a konkurens folyamatok közötti szinkronizáció megszűnését, a sáv szélesség alacsony kihasználtságát és gyengébb minőségű szolgáltatásokat eredményez. A hatékonyságot tekintve szintén kiemelt jelentőséggel bír annak vizsgálata, hogy egy adatfolyam milyen gyorsan képes elérni az állandósult állapotbeli sebességét vagy *konvergálni* az igazságos sáv szélesség-megosztás szerint biztosított maximális adatátviteli rátához dinamikus környezetben.

Transzport protokollok tervezésénél fontos szempont továbbá annak figyelembevétele, hogy azok milyen módon lesznek képesek reagálni a forgalmi viszonyokban bekövetkező hirtelen változásokra [33]. Valós hálózatokban a különböző adatátviteli mechanizmusok által szállított folyamatok gyakran vannak kitéve ilyen eseményeknek, elsősorban az útvonalak és az elérhető sáv szélesség megváltozása, vagy az esetleges torlódás miatt. Az *érzékenység*

egy olyan tulajdonság, amely megmutatja, hogy egy transzport protokoll milyen gyorsan és pontosan képes adaptálódni a különböző környezeti adottságokhoz.

A torlódásszabályozó algoritmusok működése két fő fázisra bontható. A kezdeti szakaszban a TCP fokozatosan növeli a küldési sebességet egészen addig, amíg a tároló fel nem töltődik. Ezt egy egyensúlyi állapot követi, amikor a protokoll eléri a maximális átviteli sebességet és próbálja azt stabilan tartani. Az első, átmeneti fázis hossza nagymértékben meghatározza a rövid folyamok letöltésének hatékonyságát, ezért számos alkalmazás esetén hatást gyakorolhat a felhasználói élményre is. Ezt a jelenséget az ún. *telítődési idővel* jellemzem [34], amely egy folyam átvitelének megkezdése és az első csomag tárolóból történő eldobása között eltelt időt adja meg. A tároló telítődési ideje mint metrika leírja, hogy egy szállítási protokoll milyen gyorsan képes elérni az állandósult állapotbeli teljesítményt.

4. téziscsoport. *Megvizsgáltam a digitális szökőkútkódoláson és a torlódásszabályozáson alapuló adatátviteli mechanizmusok működését dinamikusán változó hálózati és forgalmi viszonyok között [C1].*

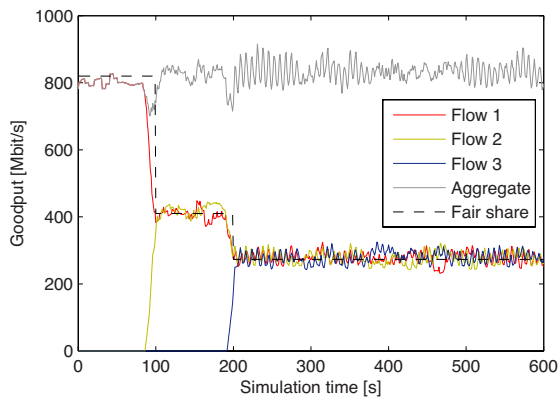
4.1. tézis. *Szimulációkkal igazoltam, hogy míg a CCA esetén a konkurens folyamatok hasznos adatátviteli sebességében rövid időskálákon kisebb mértékű ingadozás tapasztalható, addig a DFA stabilabb működést biztosít hosszú távon.*

4.2. tézis. *Megállapítottam, hogy a DFA használatával a versengő folyamatok adatátviteli sebessége átlagosan egy nagyságrenddel rövidebb idő alatt konvergál (tipikusan 1–2 mp) az igazságos sávzélesség-megosztás szerinti rátához, mint a CCA alkalmazása esetén.*

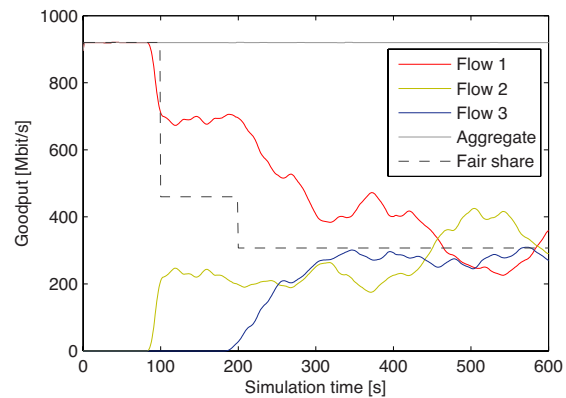
4.3. tézis. *Teljesítményelemzés segítségével megmutattam, hogy a DFA a CCA-hoz viszonyítva jóval gyorsabb alkalmazkodást tesz lehetővé az elérhető sávzélesség hirtelen megváltozásához mind a folyamonkénti, mind pedig az aggregált forgalmat tekintve. Az adaptívitás érzékenysége a CCA architektúrán kisebb tároló (<1 BDP) mellett csökken, míg DFA esetén független a tárolómérettől.*

4.4. tézis. *Igazoltam, hogy a CCA-val ellentétben a DFA esetén a tároló telítődési ideje független annak méretétől, valamint a folyamatok számától és a körülfordulási időtől is.*

A 8. ábra három, eltérő (0, 100 és 200 mp) késleltetéssel indított versengő folyam dinamikáját szemlélteti, amelyen a hasznos adatátviteli sebesség másodperces felbontással látható és a görbék értékei 10 mp hosszú ablakkal végzett mozgóátlagolásból származnak. Megfigyelhetjük, hogy a CCA esetén az adatfolyamok sebessége lassan konvergál az igazságos sávzélesség-megosztás szerinti rátához (fair share) és nagymértékben ingadozik



(a) Szökőkútkódolás alapú architektúra



(b) Torlódásszabályozás alapú architektúra

8. ábra. Eltérő késleltetéssel indított versengő folyamatok átviteli sebességének dinamikája és konvergenciája

körülötte. Ezzel szemben a DFA alacsony konvergenciaidőt biztosít, az igazságos megosztás értéke körül pedig csupán enyhe fluktuáció tapasztalható. A DFA mechanizmusa ugyanakkor jóval csomósabb átvitelt eredményez a CCA működéséhez viszonyítva, mivel az ablakméret befolyásolja a forgalom karakterisztikáját.

4.5. Sávszélességbecslés mobilhálózatokban

A sávszélesség becslésével az elmúlt évtizedekben számos kutatás foglalkozott a hálózatok világában történő széleskörű alkalmazási lehetőségeknek köszönhetően [35]. Ezeket a módszereket napjainkban több területen is hasznosítják a szállítási protokolloktól kezdve, a hálózatmenedzsmenten át egészen a multimédia átvitelig. Például az Internet forgalmának jelentős részét szállító TCP az elérhető sávszélességre vonatkozó becslés eredményét használja fel az adatküldési sebesség megfelelő beállítására, amely révén az erőforrások hatékony kihasználása válik lehetővé anélkül, hogy torlódás lépne fel a hálózatban. A szolgáltatók esetében szintén nagy szerep hárul a sávszélességbecslésre, mivel számukra kiemelt fontossággal bír a hálózatkihasználtság, valamint a felhasználói igények változásának folyamatos nyomon követése. Mobilhálózatokban komoly kihívást jelent *becslést adni az elérhető sávszélességre* [36], mivel az folyamatosan változik a különböző környezeti tényezők hatására, amilyen például a mobil eszköz pozíciója és sebessége, a cellában tartózkodó aktív felhasználók száma vagy a jelerősség.

5. tétiscsoport. *Megterveztem egy heurisztikus sávszélességbecslő eljárást mobilhálózatokra, majd valós forgalmi adatokon elemeztem a működését [C2].*

5.1. tézis. *Kidolgoztam egy sávszélességbecslő algoritmust, amely a sordinamika modellezésével minimális hálózati terhelés mellett képes megbecsülni a pillanatnyi elérhető sáv-*

szélességet mobilhálózatokban. A javasolt módszer működőképességét valós forgalmi adatok segítségével igazoltam.

5.2. tézis. Az algoritmus által szolgáltatott eredményeket elsődlegesen befolyásoló paraméter, a küszöbérték megfelelő beállítására egy általános szabályt adtam. Megmutattam, hogy a küszöbérték növelésével a becslésre felhasználható periódusok száma csökken, ugyanakkor a becslés pontossága egy töréspontig növekszik, és az optimális működés eléréséhez a küszöbparaméter értékét ezen a ponton kell rögzíteni.

5.3. tézis. Mobilhálózaton végzett mérések alapján megállapítottam, hogy webes forgalom esetén a maximális sebességű kiszolgálási periódusok döntő hányada ($\sim 75\%$) rövidebb egy másodpercnél.

5.4. tézis. Igazoltam, hogy a kidolgozott eljárás a becslésre alkalmas periódusok rövidegsége ellenére képes pontos eredményt adni.

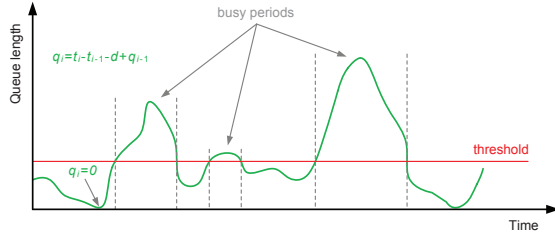
Sávszélességbecslő algoritmus

```

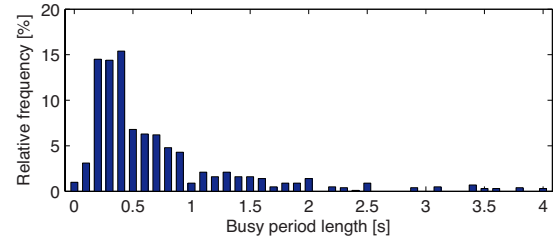
input  : trace, d, th, gap
output: bw
1  m ← false; b ← false;
2  for i ← 1 to n - 1 do
3      if  $t_{i+1} - t_i = d$  and m = false then
4          |   q ← 0;
5          |   m ← true;
6      else if  $t_{i+1} - t_i > gap$  and m = true then
7          |   m ← false;
8          |   b ← false;
9      else if m = true then
10         |   q ←  $t_{i+1} - t_i - d + q$ ;
11         |   if  $q \geq th$  and b = false then
12             |   s ←  $t_i$ ;
13             |   b ← true;
14         |   else if  $q < th$  and b = true then
15             |   rates ← Add( $\frac{\text{amount of traffic in } [s, t_i]}{t_i - s}$ );
16             |   b ← false;
17         |   end
18     end
19 end
20 return bw ← Mean(rates);

```

A megoldás lényege, hogy az aktív felhasználó által generált adatforgalomba azonos időközönként tesztcsomagok kerülnek injektálásra, amelyek mobil eszköznél észlelt ér-



(a) A sordinamika modellezése



(b) Maximális sebességű kiszolgálási periódusok előfordulási gyakorisága

9. ábra. A sáv szélességbecslő algoritmus működése

kezési ideje alapján modellezhető a szűk keresztmetszetet képező sor (bottleneck queue) dinamikája (9a. ábra). Az algoritmus a különböző hálózati tényezőket (pl. jitter) figyelembe véve egy pozitív küszöbérték definiálásával határozza meg azokat az intervallumokat, amikor a sorból maximális sebességgel történik a csomagok kiszolgálása. A részletes működést a fenti pszeudokód írja le, ahol d jelöli a tesztcsomagok generálása között eltelt időt, th a pozitív küszöbértéket, gap pedig a modellezési folyamat során elfogadható legnagyobb csomagközi időt. Az i . csomag érkezési idejét t_i , a tesztcsomagok számát a forgalmi mintában pedig n adja meg. Az m és b logikai változók a sordinamika modellezésének és a maximális kiszolgálási sebességű periódusok detektálásának fázisait azonosítják. A 9b. ábrán látható, hogy webes forgalom esetén az ilyen periódusok túlnyomó része rövidebb egy másodpercnél, ugyanakkor elegendően hosszú a pillanatnyi elérhető sáv szélesség nagy pontossággal történő megbecsléséhez. A módszer előnye, hogy számos megoldással ellentétben *nem igényel előzetes információt a hálózatról*, valamint a becsléshez kiaknázza a felhasználó által generált forgalmat, így *minimális többletterhelést ró a hálózatra*.

5. Az eredmények alkalmazhatósága

Az Interneten továbbított adatforgalom mennyisége évről-évre exponenciálisan nő az új technológiák megjelenésének és a növekvő felhasználói igényeknek köszönhetően. Napjaink folyamatosan változó, heterogén hálózati környezetében a torlódásszabályozás elvén működő TCP egyre kevésbé képes optimális teljesítményt nyújtani. Az értekezésemben bemutatott, hatékony hibajavító kódoláson alapuló adatátviteli paradigma előnyös tulajdonságai miatt számos alkalmazási területen jelenthet előrelépést. Mivel a DFCP mechanizmusa a valós hálózatokra jellemző tartományban gyakorlatilag érzéketlen a csomagvesztésre és a késleltetésre, így kiválóan alkalmazható *vezeték nélküli*, illetve olyan *veszteséges környezetben*, ahol a csomagvesztési arány nem elhanyagolható. A javasolt megoldás tá-

mogatja a *többutas kommunikációt*, amely lehetőséget biztosít többek között hatékonyabb terheléselosztás megvalósítására. A koncepció továbbá kielégíti az *adatközpontok* magas kihasználtságra vonatkozó követelményét, valamint kisméretű tárolók használata esetén is működőképes. Az utóbbi tulajdonság komoly előnnyel bír *optikai hálózatokban*, ahol technológiai korlátok miatt nincs lehetőség nagy tárolók alkalmazására.

Hivatkozások

- [1] J. Postel, “Transmission Control Protocol”, *RFC 793, IETF*, 1981.
- [2] A. Afanasyev, N. Tilley, P. Reiher, L. Kleinrock, “Host-to-Host Congestion Control for TCP”, *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, pp. 304–342, 2010.
- [3] B. Raghavan, A. C. Snoeren, “Decongestion Control”, *Proceedings of the 5th ACM Workshop on Hot Topics in Networks*, pp. 61–66, Irvine, CA, USA, 2006.
- [4] D. Clark, S. Shenker, A. Falk, “GENI Research Plan (Version 4.5)”, April 23, 2007.
- [5] T. Bonald, M. Feuillet, A. Proutiere, “Is the ‘Law of the Jungle’ Sustainable for the Internet?”, *Proceedings of the 28th IEEE Conference on Computer Communications*, pp. 28–36, Rio de Janeiro, Brazil, 2009.
- [6] ns-2 Network Simulator, <http://www.isi.edu/nsnam/ns/>
- [7] Network Simulation Cradle, <http://www.wand.net.nz/~stj2/nsc/>
- [8] Emulab Network Emulation Testbed, <http://www.emulab.net/>
- [9] S. Molnár, P. Megyesi, G. Szabó, “Multi-Functional Emulator for Traffic Analysis”, *Proceedings of the 48th IEEE International Conference on Communications*, pp. 2397–2402, Budapest, Hungary, 2013.
- [10] K. Pawlikowski, H.-D. Joshua Jeong, J.-S. Ruth Lee, “On Credibility of Simulation Studies of Telecommunication Networks”, *IEEE Communications Magazine*, vol. 40, no. 1, pp. 132–139, 2002.
- [11] M. Bateman, S. Bhatti, “TCP Testing: How Well Does ns2 Match Reality?”, *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 276–284, Perth, Australia, 2010.
- [12] S. Floyd, E. Kohler, “Tools for the Evaluation of Simulation and Testbed Scenarios”, *Technical Report, IETF*, 2006.
- [13] M. P. Fernandez, S. Wahle, T. Magedanz, “A New Approach to NGN Evaluation Integrating Simulation and Testbed Methodology”, *Proceedings of the 11th International Conference on Networks*, pp. 22–27, Saint-Gilles, Réunion Island, France, 2012.
- [14] D. X. Wei, P. Cao, S. H. Low, “Time for a TCP Benchmark Suite?”, *Technical Report, California Institute of Technology*, pp. 1–6, 2005.
- [15] D. J. C. MacKay, “Fountain Codes”, *IEE Proceedings – Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.
- [16] A. Shokrollahi, “Raptor Codes”, *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [17] M. Shreedhar, G. Varghese, “Efficient Fair Queuing Using Deficit Round-Robin”, *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 375–385, 1996.

- [18] A. Kortebe, L. Muscariello, S. Oueslati, J. Roberts, “On the Scalability of Fair Queuing”, *Proceedings of the 3rd ACM Workshop on Hot Topics in Networks*, pp. 1–6, San Diego, CA, USA, 2004.
- [19] A. Shokrollahi, “LDPC Codes: An Introduction”, *Technical Report, Digital Fountain Inc.*, pp. 1–34, 2003.
- [20] M. Luby, “LT Codes”, *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pp. 271–280, Vancouver, BC, Canada, 2002.
- [21] V. Jacobson, “Congestion Avoidance and Control”, *Proceedings of the 1988 ACM SIGCOMM Symposium*, pp. 314–329, Stanford, CA, USA, 1988.
- [22] I. Rhee, L. Xu, “CUBIC: A New TCP-Friendly High-Speed TCP Variant”, *Proceedings of the 3rd International Workshop on Protocols for Fast Long-Distance Networks*, pp. 1–6, Lyon, France, 2005.
- [23] L. Xu, K. Harfoush, I. Rhee, “Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks”, *Proceedings of the 23rd IEEE International Conference on Computer Communications*, vol. 4, pp. 2514–2524, Hong Kong, China, 2004.
- [24] T. V. Lakshman, U. Madhow, “The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss”, *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, 1997.
- [25] S. Floyd, T. Henderson, A. Gurtov, “The NewReno Modification to TCP’s Fast Recovery Algorithm”, *RFC 3782, IETF*, 2004.
- [26] K. Fall, S. Floyd, “Simulation-based Comparisons of Tahoe, Reno, and SACK TCP”, *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 3, pp. 5–21, 1996.
- [27] G. Appenzeller, I. Keslassy, N. McKeown, “Sizing Router Buffers”, *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 281–292, 2004.
- [28] J. Gettys, K. Nichols, “Bufferbloat: Dark Buffers in the Internet”, *Communications of the ACM*, vol. 55, no. 1, pp. 57–65, 2012.
- [29] H. Park, E. F. Burmeister, S. Bjorlin, J. E. Bowers, “40-Gb/s Optical Buffer Design and Simulation”, *Proceedings of the 4th International Conference on Numerical Simulation of Optoelectronic Devices*, pp. 19–20, Santa Barbara, CA, USA, 2004.
- [30] Y. Huang, R. Guérin, “Does Over-Provisioning Become More or Less Efficient as Networks Grow Larger?”, *Proceedings of the 13th IEEE International Conference on Network Protocols*, pp. 225–235, Boston, MA, USA, 2005.
- [31] D. Bansal, H. Balakrishnan, S. Floyd, S. Shenker, “Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms”, *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 263–274, 2001.
- [32] Y. R. Yang, M. S. Kim, S. S. Lam, “Transient Behaviors of TCP-friendly Congestion Control Protocols”, *Proceedings of the 20th IEEE International Conference on Computer Communications*, vol. 3, pp. 1716–1725, Anchorage, AK, USA, 2001.

- [33] S. Floyd, “Metrics for the Evaluation of Congestion Control Mechanisms”, *RFC 5166, IETF*, 2008.
- [34] S. Molnár, B. Sonkoly, T. A. Trinh, “A Comprehensive TCP Fairness Analysis in High Speed Networks”, *Elsevier Computer Communications*, vol. 32, no. 13–14, pp. 1460–1484, 2009.
- [35] C. D. Guerrero, M. A. Labrador, “On the Applicability of Available Bandwidth Estimation Techniques and Tools”, *Elsevier Computer Communications*, vol. 33, no. 1, pp. 11–22, 2010.
- [36] U. Hentschel, A. Schmidt, A. Polze, “Predictable Communication for Mobile Systems”, *Proceedings of the 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 24–28, Newport Beach, CA, USA, 2011.

Publikációk

Folyóiratcikkek

- [J1] S. Molnár, **Z. Móczár**, B. Sonkoly, “Living with Congestion: Digital Fountain based Communication Protocol”, accepted to *Elsevier Computer Communications*, 2016.
- [J2] **Z. Móczár**, S. Molnár, “Towards the Transport Protocols of Future Internet”, *Infocommunications Journal*, vol. 6, no. 3, pp. 3–9, 2014.
- [J3] G. Szűcs, **Z. Móczár**, “Iterative Text Clustering of Search Results”, *International Journal of Computers*, vol. 7, no. 4, pp. 127–134, 2013.

Konferenciatickek

- [C1] **Z. Móczár**, S. Molnár, “On the Dynamic Behavior of Digital Fountain Based Communication”, *Proceedings of the 58th IEEE Global Communications Conference, Exhibition and Industry Forum (GLOBECOM 2015)*, pp. 1–6, San Diego, CA, USA, 2015.
- [C2] **Z. Móczár**, S. Molnár, “Bandwidth Estimation in Mobile Networks by Busy Period Detection”, *Proceedings of the 25th IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2014)*, pp. 1354–1358, Washington D.C., USA, 2014.
- [C3] **Z. Móczár**, S. Molnár, B. Sonkoly, “Multi-Platform Performance Evaluation of Digital Fountain Based Transport”, *Proceedings of the Science and Information Conference 2014 (SAI 2014)*, pp. 690–697, London, UK, 2014.
- [C4] S. Molnár, **Z. Móczár**, B. Sonkoly, “How to Transfer Flows Efficiently via the Internet?”, *Proceedings of the 3rd IEEE International Conference on Computing, Networking and Communications (ICNC 2014)*, pp. 462–466, Honolulu, HI, USA, 2014.
- [C5] S. Molnár, **Z. Móczár**, A. Temesváry, B. Sonkoly, Sz. Solymos, T. Csicsics, “Data Transfer Paradigms for Future Networks: Fountain Coding or Congestion Control?”, *Proceedings of the 12th IFIP International Conference on Networking (Networking 2013)*, pp. 1–9, New York, NY, USA, 2013.
- [C6] **Z. Móczár**, S. Molnár, “Comparative Traffic Analysis Study of Popular Applications”, *Proceedings of the 17th EUNICE Workshop on Energy-Aware Communications (EUNICE 2011)*, pp. 124–133, Dresden, Germany, 2011.
- [C7] S. Molnár, **Z. Móczár**, “Three-dimensional Characterization of Internet Flows”, *Proceedings of the 46th IEEE International Conference on Communications (ICC 2011)*, pp. 1–6, Kyoto, Japan, 2011.

- [C8] **Z. Móczár**, S. Molnár, “Characterization of BitTorrent Traffic in a Broadband Access Network”, *Proceedings of the 5th International Conference on Access Networks (AccessNets 2010)*, pp. 1–8, Budapest, Hungary, 2010.

Egyéb tudományos közlemények

- [O1] **Z. Móczár**, “Modeling and Analysis of TCP Algorithms Based on Optimization Theory”, *Technical Report*, Budapest, Hungary, 2012.
- [O2] S. Molnár, **Z. Móczár**, B. Sonkoly, Sz. Solymos, T. Csicsics, “Design and Performance Evaluation of the Digital Fountain based Communication Protocol”, *Technical Report*, Budapest, Hungary, 2012.