

TFRC alapú szelektív újraküldő algoritmus

Huszák Árpád, Imre Sándor
huszak@hit.bme.hu, imre@hit.bme.hu
Budapesti Műszaki és Gazdaságtudományi Egyetem
Híradástechnikai Tanszék
Mobil Távközlési és Informatikai Laboratórium (MC²L)

1. Bevezetés

Az új vezeték nélküli technológiák ma már lehetővé teszik a multimédiás alkalmazások használatát, hiszen ezek már rendelkeznek azokkal a jellemzőkkel, melyek lehetővé teszik az ilyen típusú alkalmazások kiszolgálását. Ez az oka annak, hogy ezek az audió/videó szolgáltatások robbanásszerű fejlődésen mennek át és sok új és izgalmas problémát vetnek fel. Az újonnan megjelenő hozzáférési hálózatok már képesek kielégíteni az új igényeket. Ilyen technológia például a nemrégiben megjelent WiMAX (IEEE 802.16), amely akár ötven kilométer sugarú körben, 70 Mbit/s sebességen kínál vezeték nélküli internetes hozzáférést, vagy az UMTS. Ennek okán mindenképpen nagymértékű növekedésre kell számítani az IP alapú audió/videó alkalmazások terén.

Hasonló fejlődési tendencia mutatható ki a végberendezések fejlődésében is, hiszen ma már nem jelent technológiai problémát kép és hanganyag lejátszása a mobil terminálokon. Ugyanakkor ezek az eszközök nagymértékben már IP alapú kommunikációra alapszanak. Jelenleg előrehaladott fejlesztések folynak adat-szolgáltatások biztosítására vezeték nélküli hálózatokban, és a végcél a mobil kommunikáció és az Internet teljes konvergenciája, a mobil Internet kifejlesztésével. Az IP alapú hálózatot használva, a vezetékes hálózatokra írt jól bevált alkalmazások működhetnek a vezeték nélküli hálózatokon is. A vezeték nélküli hálózatok tulajdonságai nagymértékben eltérnek a vezetékes hálózatokétól, hiszen a rádiós csatorna sokkal érzékenyebb a zavarokra, a környezeti hatásokra vagy az időjárásra. Egy mozgó állomás esetén pedig a cellaváltás (handover) miatti késleltetések, adatvesztés is jelentősen ronthatja a minőséget. A feladat tehát egy változó tulajdonságú és jelentős hibavalószínűséggel rendelkező vezeték nélküli összeköttetésen, egy garantált minőségű adatfolyam átvitele, vagyis QoS (Quality of Service) biztosítása. Ez az adatfolyam lehet szigorúan érzékeny a hibákra, mint például FTP, WWW alkalmazásokhoz tartozó adatátvitel, de lehet a hibákra érzéketlenebb, mint a hang-, kép- illetve videó átvitel esetén. Ennek oka, hogy a kisebb hibákat az emberi érzékszervek nehezebben észlelik, ezért bizonyos mértékig megengedhető a hibázás, annak érdekében, hogy a vevő oldalon folyamatos legyen feldolgozás, más szóval folyamatos mozgóképet illetve hangot érzékeljen a felhasználó.

Ezek a multimédiás alkalmazások többnyire az UDP-t [1] használják szállítási rétegbeli protokollként, de újabb szabványok is megjelentek már, amelyek hatékonyabbnak bizonyulnak audió/videó folyamatok átvitele esetén (pl. UDPLite [2], SCTP [3], DCCP [4], stb.). A DCCP nagyon előnyös tulajdonsága, hogy torlódáskezelő algoritmust (TCPLike [5], TFRC [6]) is használ a hálózat túlterhelésének elkerülése érdekében.

A TDK dolgozatban egy olyan általunk kidolgozott eljárást mutatunk be, amely az MPEG típusú videó folyamat sérült illetve elveszett csomagjait szelektíven újraküldi a hálózat pillanatnyi állapotától függően. Ezáltal a hálózat szabad kapacitását kihasználva tudunk a multimédiás tartalom minőségén javítani. A szabad kapacitást a forrás sebessége és a DCCP torlódáskezelő algoritmus (TFRC – TCP Friendly Rate Control) által szolgáltatott paraméterek alapján tudjuk meghatározni.

A dolgozat következő fejezetében megismerhetjük a jelentősebb szállítási rétegbeli protokollokat, amelyek alkalmasak lehetnek az audió/videó folyamat továbbítására. Ezt követi az általunk kidolgozott algoritmust ismertetése, mely egy új transzport protokoll (DCCP) alkalmazásával javítani képes az MPEG típusú multimédiás adatfolyam minőségén a TFRC torlódáskezelő algoritmuson alapuló szelektív újraküldési technika segítségével. A negyedik fejezetben az algoritmus hatékonyságának vizsgálatát foglaljuk össze. Majd végül az ötödik fejezetben levonjuk a következtetéseinket és ismertetjük a továbblépési lehetőségeket.

2. Szállítási réteg protokolljai multimédiás alkalmazásokhoz

Az ISO/OSI rétegstruktúrában, a hálózati réteg (network layer) felett elhelyezkedő szállítási réteg feladata a sorrendhelyes, duplikációmentes és megbízható átvitel biztosítása a felhasználói alkalmazástól függően. Számos transzport protokoll áll rendelkezésünkre, amelyek közül az alkalmazás típusától függően választjuk ki a legalkalmasabbat. A szállítási réteg protokolljait alapvetően két osztályba sorolhatjuk: megbízható és megbízhatatlan.

A megbízható transzport protokoll legfőbb jellemzője, hogy minden csomagot eljuttat a vevőhöz. Ezt természetesen csak a megsérült vagy elveszett csomagok újraküldésével lehetséges. Az újraküldésnek hátrányai is vannak, mégpedig az, hogy a sérült illetve hiányzó csomagok újraküldését és a vevőhöz való megérkezését meg kell várni, és a többi csomagot csak ezt követően tudjuk a hálózatra küldeni. Ez időnként jelentős késleltetés okozhat. A megbízható protokollok legismertebb képviselője a TCP (Transmission Control Protocol) [7], de a nemrégiben kifejlesztett SCTP (Stream Control Transport Protocol) is ehhez a csoporthoz tartozik.

A megbízhatatlan protokollok, mit például az UDP (User Datagram Protocol) vagy a DCCP (Datagram Congestion Control Protocol) nem küldi újra a sérült adatsomagokat, emiatt nem fordulhat elő az sem, hogy egy újraküldött csomag megérkezésére kelljen várni. Ezeket a típusú protokollokat ezért olyan alkalmazások esetén használják, ahol a késleltetés alacsony szinten tartása a legfontosabb szempont és bizonyos számú csomag elvesztése elfogadható a felhasználó számára. Általában megbízhatatlan protokollokat használnak a multimédiás alkalmazások esetén, az ismertetett tulajdonságok alapján.

A szállítási réteg két régi nagy protokollja a TCP és az UDP. Ezek a protokollok a mai napig meghatározói a számítógépek valamint számítógép-hálózatok közötti adattovábbításnak, pedig immár 25 éves szabványok, hiszen a TCP-t és az UDP-t is az 1980-as évek elején fejlesztették ki. Ez az oka annak, hogy mindkét szabványt viszonylag kis hibavalószínűségű, vezetékes hálózatra dolgozták ki, azonban a ma egyre szélesebb körben használt vezeték nélküli hálózatok karakterisztikái jelentősen különböznek vezetékes hálózatok adatátviteli tulajdonságaitól. Az eltelt évek alatt a vezetékes számítógépes hálózatok is nagyon sokat fejlődtek. Emiatt a régi protokollokat felül kell vizsgálni, és az új igényeknek megfelelően módosítani kell azokat, illetve újabb szabványok kifejlesztésére van szükség. Így került sor a régi protokollok újabb változatainak megjelenésére, TCP esetén a TCP Reno, TCP New-Reno, TCP Vegas, TCP SACK, míg az UDP módosított változata az UDP Lite (Lightweight User Datagram Protocol). A közelmúltban több új transzport protokollt fejlesztettek ki, melyek megpróbálják kiküszöbölni a régebbi protokollok hibáit, mint például a multimédiás átvitelre szánt megbízható SCTP és a megbízhatatlan DCCP.

A továbbiakban a DCCP működését ismertetjük, mivel a TFRC torlódáskerülő mechanizmuson alapuló szelektív újraküldés ezt a protokollt használja számos előnye miatt.

2.1. Datagram Congestion Control Protocol

A DCCP egy megbízhatatlan transzport protokoll, amely torlódásszabályozási algoritmus használatára, valamint sorrendhelyes csomagtovábbításra is alkalmas a TCP-hez hasonlóan. UDP esetén a torlódás elleni védekezést az alkalmazásoknak kellett megoldaniuk, míg DCCP esetén ez már a protokoll szerves része. A torlódásszabályozó algoritmus feladata, hogy elkerülje a hálózat túlterhelését. Amennyiben a hálózat útvonalválasztóinak bufferei megtelnek a még nem továbbított csomagokkal, a továbbra is érkező csomagok elvesznek. A forrásnak tehát értesülnie kell arról, hogy a hálózatban torlódás alakult ki, ezért a küldési sebességet csökkentenie kell. A DCCP-t úgy próbálták alakítani, hogy a TCP és az UDP

előnyeit egy protokollként valósítsák meg. A DCCP fejlécében így ráismerhetünk az előbbi protokollokból ismert fejlécmezőkre.

A fejléc hossza minimálisan 12 byte, maximálisan pedig a 1024 byte-t is elérheti, ha az opcionális mezőket, és az egyes csomag típusok esetén használt pótlólagos mezőket is használjuk. Általános esetben az ellenőrzőösszeg (Checksum), az összes adatot lefedi, de az UDP Lite-hoz hasonlóan, a DCCP is lehetővé teszi az adatok részleges lefedését ellenőrzőösszeggel. Ez lehetőséget ad arra, hogy azok az alkalmazások, amelyek képesek kezelni a sérült adatokat, hatékonyabban működjenek.

A kapcsolatorientált DCCP a kapcsolat felépítése során megbízható protokollként működik. A torlódásszabályozással kapcsolatos üzenetek szintén megbízható adatfolyamként kerül továbbításra. A torlódásszabályozási algoritmust a kapcsolat felépítés során kerül meghatározásra. Az adatfolyam során azonban mind a vevő, mind az adó oldal kezdeményezheti a torlódásszabályozási algoritmus megváltoztatását. Jelenleg ugyan csak két ilyen algoritmus van specifikálva, de a DCCP protokollt felkészítették esetleges újabb torlódáskezelési módszerek bevezetésére is. A torlódásszabályozó algoritmusokat a fejléc CCID (Congestion Control Identifier) mezőjében különböztetjük meg.

- TCP-like Congestion Control [CCID 2]
- TFRC (TCP-Friendly Rate Control) Congestion Control [CCID 3]

A TCP-Like algoritmus a TCP torlódáselkerülési technikáját alkalmazza, amire jellemző a hirtelen sebességcsökkenés csomagvesztés esetén. A TFRC egy képlet (1) alapján határozza meg a küldési sebességet, aminek köszönhetően nincsenek drasztikus sebességváltozások.

$$T = \frac{s}{R \sqrt{\frac{2p}{3} + 4R \left(3 \sqrt{\frac{3p}{8}} \cdot p \cdot (1 + 32p^2) \right)}} \quad (1)$$

A képletben szereplő változók: T – küldési sebesség; s – csomagméret; R – körbefordulási idő (RTT) becült értéke; p – csomagvesztési arány.

A körbefordulási idő becslésére a következő képlet szolgál:

$$R = qR + (1-q)RTT \quad (2)$$

A képletben RTT az aktuális körbefordulási idő, míg az R a becslés, $q=0.9$ pedig állandó. A becslés célja, hogy a pillanatnyi ingadozás ne legyen hatással küldési sebességre. Ugyanígy módon kerül meghatározásra a csomagvesztési arány is hasonló módon számítható. A p számításához, a korábbi időintervallumokban mért csomagvesztési arány

értékét súlyozva használják. Ennek az ad értelmet, hogy az egyszeri hibák, amely lehet a torlódás következménye vagy a csatorna hibája, csak kisebb mértékben befolyásolja a küldési sebességet. A becslések teszik lehetővé, hogy a küldési sebesség ne változzon drasztikusan, hiszen pont ez a TFRC módszer lényege.

Az adatfolyam ugyan megbízhatatlan, de az adó oldal értesül a vevő által fogadott csomagok helyes megérkezéséről. A vevő nyugtát küld az érkezett csomagokról. A torlódáskezelő algoritmus egyben azt is meghatározza, hogy milyen gyakran érkeznek nyugták. TCP-like (CCID 2) esetben körülbelül két elküldött csomag után érkezik nyugta, míg a TCP-Friendly Rate Control (CCID 3) esetén körbefordulási időnként (Round Trip Time) küld egy nyugtát a vevő.

A DCCP-t olyan alkalmazások számára fejlesztették ki, mint például a streaming médiaalkalmazások, amelyek ki tudják használni a DCCP beépített szabályozási módszereit. Annak érdekében, hogy a DCCP hatékonyan vegye fel a versenyt a gyors UDP-vel, a DCCP csomagok fejlécét próbálták a lehető legkisebbre méretezni. A protokoll bizonyos feladatok estén még így is túlságosan bonyolult, ezért kifejlesztettek egy egyszerűsített DCCP protokollt is, melynek neve DCCP-Lite.

3. TFRC alapú szelektív újraküldő algoritmus

A mobil eszközök teljesítménye ma már lehetővé teszi nagyobb erőforrás-igényű multimédiás alkalmazások használatát. Azonban az általuk használt rádiós csatorna rendkívül változékony, nagy hibavalószínűséggel rendelkezik, amely komoly gondok elé állítja a tervezőket. Emellett pedig a megnövekedett hálózati forgalom okozhatja a hálózat túlterhelését, ami csomagvesztéshez vezet.

Az MPEG típusú multimédiás adatfolyamok javítására már korábban kidolgoztunk egy eljárást, ami a csomagok fontosságának függvényében újraküldi a hibás vagy sérült csomagokat. Mivel az MPEG videó három különböző képtípus használ, melyeknek prioritása nem egyforma, a különböző típusú keretekhez tartozó információ elvesztése különböző hatással van a mozgókép minőségére.

I-képek: Intra frame coded - csak képkockán belül kódolt, ami azt jelenti, hogy a képkockákon belül JPEG tömörítést alkalmazva kódolja a teljes képet. Ez a kerettípus a legfontosabb, hiszen ez szolgál alapul a többi képtípushoz.

P-képek: a P (predicted) képkocka az őt megelőző "I" vagy "P" képen alapul, azokat használja referenciaként. Ezt nevezik "forward prediction"-nek. A P-keret a megelőző I- vagy P-keret

képrészleteinek elmozdulását, illetve a képtartalmak közti különbséget rögzíti. A P-képek tömörítési aránya nagyobb, mint az I-kereté.

B-képek: a B (bidirectional) kép a megelőző és rákövetkező "I" vagy "P" képkockákat is felhasználja referenciaként. A kódolás a mozgáskompenzációs technika felhasználásával, a P-keretekhez hasonló módon, de két irányból történik. Ezt az eljárást "bidirectional prediction", két irányból történő jóslásnak nevezik. A B-képeknek a tömörítési aránya a legnagyobb és nem szolgál referenciaként más képek kódolásakor, ezért nem terjeszt hibákat sem

Az általunk kidolgozott szelektív újraküldés átmenetet képez a megbízható és a megbízhatatlan alapprotokollok között, hiszen az elvesztett csomagok csak egy részét küldi újra. A korábbi munkákban [8] a csomag tartalma alapján döntöttünk a csomagok újraküldéséről, azonban ezt már szempontok alapján is megtehetjük. A csomag újraküldéséről tehát több feltétel alapján dönthetünk:

- Csomag tartalma alapján
- A hálózati terhelés alapján
- Az újraküldött csomag várható vételi ideje alapján

A szelektív újraküldés alapötlete az, hogy az MPEG formátum I-kereteit kiemelten kezeljük, hiszen az ezt követő keretek ezekre épülnek egészen a következő I-keretig. Amennyiben ezeket a keretek hibátlanul továbbítanánk a vevő oldalra, jelentősen javíthatnánk az audió/video minőséget. Algoritmusunk ezt oldja meg, vagyis amennyiben egy I-kép meghibásodik, újraküldjük azt. Ezt a problémát leghatékonyabban a DCCP protokoll segítségével tudjuk megoldani, hiszen sorszámozott csomagokkal rendelkezik, és nem alapfeladata a hibás csomagok újraküldése, valamint alkalmas a hibás csomagok továbbítására is, az UDPLite-hoz hasonlóan. Az egyes csomagok újraküldése már a mi feladatunk. Ebből a szempontból a DCCP összesíti a többi transzport protokoll előnyét, ezért választottuk ezt a protokollt a probléma megoldásához.

Ebben a dolgozatban a hálózati terhelés alapján határozzuk a csomagok újraküldéséről. Ehhez azonban ismernünk kell a hálózat aktuális állapotát. Tudnunk kell, mikor van lehetőség újabb csomagok beküldésére a hálózatba, anélkül hogy torlódást okoznánk, vagy a már túlterhelt hálózat terheltségét tovább fokozzuk. Annak megállapítására, hogy mikor tekinthető a hálózat túlterheltnek, vagy túlterheléshez közeli állapotban lévőnek, a DCCP TFRC (CCID3) torlódáskezelő algoritmus ad választ. Az TFRC algoritmus a hálózat jellemzőit használja fel, a torlódás elkerülése érdekében, melyeket az (1) képletben alkalmaz a küldési sebesség megállapítására. Abban az esetben, amikor a hálózat túlterheltség miatt csomagot

dob el, a küldő oldalra nem érkezik nyugta az adott csomagról. Ennek hatására a p csomagvesztési arány nő, ami a küldési sebesség csökkenéséhez vezet. Hasonló hatással van a küldési sebességre a körbefordulási idő (RTT) növekedése is. Ha ez az érték növekszik, abból arra következtethetünk, hogy az elküldött csomagok egyre tovább várnak a routerek bufferjeiben, mire továbbküldésre kerülnek, vagyis a hálózat közeledik a túlterhelt állapothoz.

A hálózatban tehát két módon kerülhet sor csomagvesztésre vagy csomagsérülésre. Az egyik lehetőség, hogy az egyik a hálózati router bufferjének túlsordul, a másik pedig, hogy a csatorna hibázása miatt sérül a csomag. Az első esetben egy ideig nem küldhetjük újra a csomagot, hiszen az további terhelést okoz a hálózatnak, míg ha a routereknek vannak szabad kapacitásai, csupán a rossz minőségű csatorna miatt sérült a csomag, a csomagot újra lehet küldeni. A feladat tehát, hogy ezt a két eseményt szétválasszuk. Ez azonban nem egyértelmű, hiszen a vevő, illetve a küldő oldal csupán a csomagvesztés tényét és körbefordulási időt tudja észlelni.

Abban az esetben, ha ismerjük a forrásunk sebességét, ami jelen esetben egy MPEG videó folyam, el tudjuk dönteni, hogy egy adott pillanatban a TFRC által meghatározott küldési sebesség esetén, van-e még szabad kapacitás a hálózatban. Tehát abban az esetben, ha:

$$X_{TFRC}(t) > S_{MPEG}(t) \quad (3)$$

újraküldhetjük az elveszett csomagot, anélkül hogy túlterhelnénk a hálózatot. A képletben $X_{TFRC}(t)$ a TFRC torlódáselkerülő algoritmus által meghatározott pillanatnyi küldési sebességet jelenti, míg $S_{MPEG}(t)$ az MPEG folyam sávszélesség igényét jelenti.

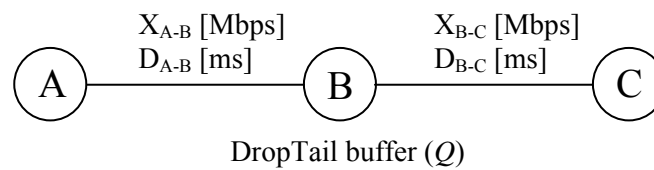
A TFRC alapú szelektív újraküldő algoritmus olyan szituációkban bizonyul hatékonyknak, amikor a TFRC küldési sebesség az MPEG folyam sebesség körül ingadozik. Ilyen torlódás közeli állapotban sikeresen meg tudjuk védeni a hálózatot az összeomlástól. A vizsgálatok során is az ilyen állapotú hálózatokat vizsgáltuk.

Abban az esetben, amikor a hálózat kapacitása sokkal nagyobb, mint a videó folyam sebessége, a TFRC algoritmus is nagy küldési sebességet engedélyez a küldő terminálnak: $X_{TFRC}(t) \gg S_{MPEG}(t)$. Ekkor az újraküldött csomag biztosan nem fog torlódást előidézni a hálózatban.

Előfordulhat, hogy a videó folyam sebessége túl nagy és emiatt a hálózat képtelen azt továbbítani, a szűkös erőforrásai miatt. Ilyenkor már maga az MPEG folyam is torlódást okoz, tehát nincs lehetőség a minőség javítására az újraküldés alkalmazásával. A Wakamiya és társai [9] által kidolgozott TFRC alapú MPEG kódolás megoldást nyújt arra, hogy mindig a felhasználható szabad hálózati kapacitáshoz alkalmazkodva kódoljuk a videó anyagot. Ez a

megoldás azt eredményezi, hogy a TFRC által meghatározott küldési sebesség és az MPEG folyam sebessége közel azonos. Ilyen helyzetekben pedig jól alkalmazható az általunk kidolgozott újraküldési mechanizmus.

A TFRC alapú szelektív újraküldést Ns2 szimulációs környezetben vizsgáltuk, valamint a szimulátor kimeneti adatai alapján kiegészítő méréseket végeztünk. A hálózatot egyszerűsített formában implementáltuk, ami azt jelenti, hogy három csomópontot helyeztünk el a következő módon:



1. ábra Topológia

Az ábrán az A csomópont jelöli az MPEG streaming szervert, amely a B routeren keresztül továbbítja a csomagokat a C felhasználóhoz. A szimulációban a B router reprezentálja a többi elemét is, így amikor a B router Q elemű bufferje túlsordul, a hálózat túlterheltté válik. A B router DropTail típusú buffert alkalmaz, melynek tulajdonsága, hogy azok a csomagok melyek már nem férnek el a tárolóban, elvesznek. A mérésekben $Q=10$. Az $A-B$ link sávszélessége (X_{A-B}) és késleltetés (D_{A-B}) a mérés során nem változik, míg a $B-C$ link sávszélessége (X_{B-C}) a hálózat terhelését szimulálva ingadozik.

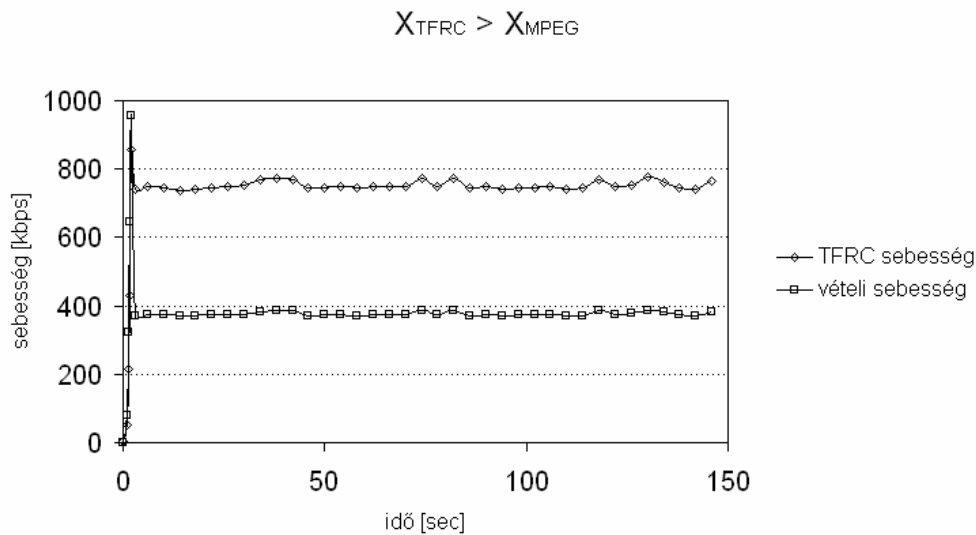
Az MPEG folyam sebessége is állandó, és mindegyik mérésben kisebb, mint az $A-B$ link kapacitása. Ezen a szakaszon tehát torlódás nem történhet. Azokban a mérési elrendezésekben, ahol a csatorna veszteségét is vizsgálni akarjuk, ezen a linken fogunk hibákat generálni.

4. Szimulációs eredmények

Az újraküldött csomagok, amennyiben időben érkeznek a vevő oldalra, hatékonyan javítják a videó folyam minőségét. A cél pedig, hogy feleslegesen ne küldjük olyan újraküldendő csomagokat a hálózatba, amelyek nagy valószínűséggel nem érkeznek meg, vagy más csomagok elvesztéséhez vezetnek. A hálózat terheltségét jól mutatja az 1. ábraán B -vel jelölt router bufferjének aktuális szintje.

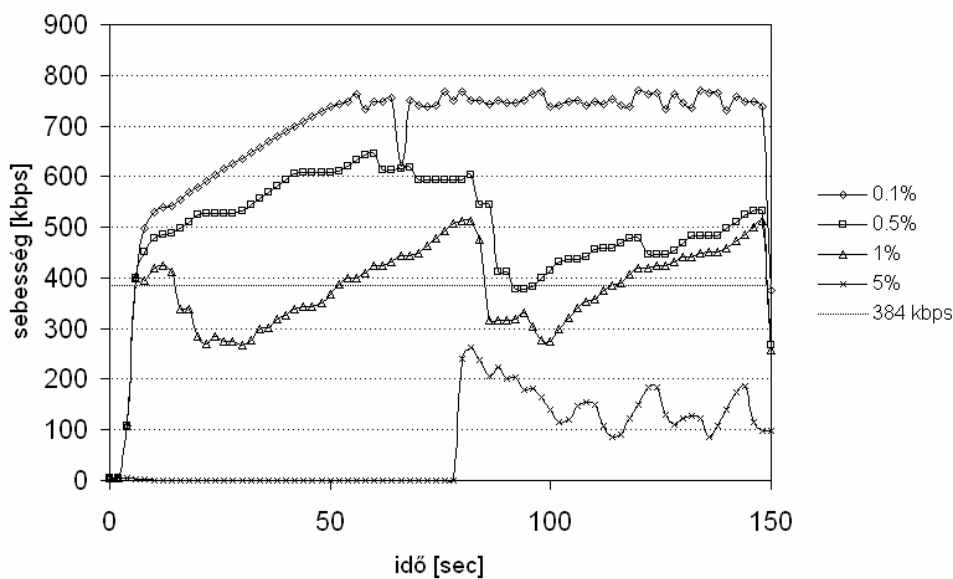
Abban az esetben, amikor az $A-B$ és $B-C$ link kapacitása is meghaladja az MPEG videó sebességét, az újraküldést a hálózati erőforrások szűkössége nem korlátozza. A 2. ábraán a

TFRC által meghatározott küldési sebesség és a vételi sebesség látható az idő függvényében, 384kbps kódolási sebességű MPEG videó esetén. A mérési elrendezésben a csomagvesztési valószínűség nulla, D_{A-B} és D_{B-C} egyaránt 30ms.



2. ábra A TFRC küldési sebesség és a vételi sebesség az idő függvényében

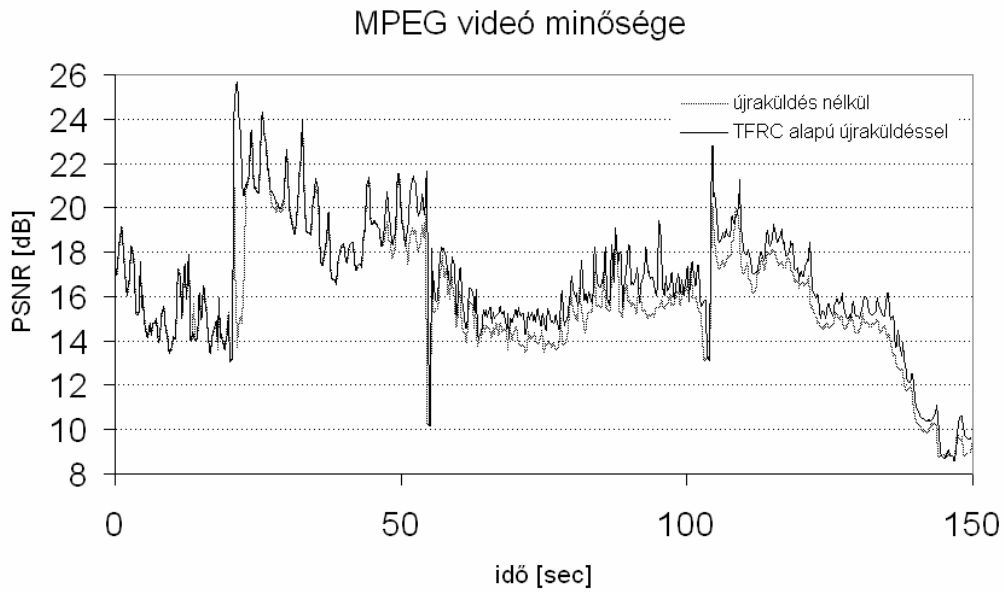
A TFRC küldési sebességét jelentősen befolyásolja a csomagvesztés alakulása. Az adó oldal nem tudja meghatározni, hogy mi a csomagvesztés oka, ezért egyformán reagál a torlódás miatti, és a rossz minőségű csatorna okozta csomagvesztésre. Ha a csomag nem érkezik meg a vevő oldalra, és a videó sebessége a link szabad kapacitása alatt van, a csomagot újra lehet küldeni ezzel javítva a videó minőségét. Megfigyelhető, hogy torlódás esetén az RTT értéke jelentősen megnő, mivel a csomagok a routerek tárolóiban várnak, míg a csatorna hibázása esetén az RTT értéke nem változik jelentősen. A TFRC algoritmus ezért nem a pillanatnyi RTT értékkel számol, hanem annak simított értékével, ahogy azt korábban bemutattuk. A következő ábrán a TFRC által meghatározott küldési sebesség alakulását ábrázoltuk, különböző csomagvesztési valószínűségek mellett. A linkek kapacitása (1Mbps) továbbra is nagyobb, mint a videó sebessége (384kbps), tehát van szabad kapacitása a csomagok újraküldésére.



3. ábra TFRC sebesség a csomagvesztés függvényében

A TFRC működésére az RTT értéke is hatással van, ami az *A-B* és a *B-C* link esetén is 30ms. A csomagvesztés 0.1% és 0.5% értéke esetén a torlódáskerülő algoritmus által meghatározott küldési sebesség folyamatosan a videó sebessége felett van. Hangsúlyozni kell azonban, hogy nagyobb RTT esetén ez már nem állna fenn. Amikor a csomagvesztés 1%, a TFRC sebessége többször a videó folyam sebessége alá csökken. A TFRC alapú újraküldési algoritmus alapján, az adott mintahálózaton, 0.5%-nál kedvezőbb csomagvesztési arány esetén a sérült csomag újraküldése lehetséges, míg 5%-os csomagvesztés esetén az újraküldés nem engedélyezett. Látható, hogy ez az arány már a kapcsolat felépítésénél is gondot okoz, hiszen csak a 70. másodpercben sikerült azt létrehozni. A TFRC nagyon nagy csomagvesztési arány mellett nem tudja jól megadni a szabad csatornkapacitás mértékét, emiatt tapasztalhatjuk azt, hogy annak ellenére, hogy van szabad kapacitás, a TFRC már jelentősen visszafogja a küldési sebességet.

Az újraküldött csomagok jótékony hatással lesznek a videó folyam minőségére. Ha egy I-kerethez tartozó információt sikerül sikeresen eljuttatni a vevőhöz, a javulás jelentős lesz, hiszen erre a kerettípusra épül a többi. A videó minőségének mérésére a PSNR (Peak to Peak Signal to Noise Ratio) eljárást használjuk, ami egy objektív mérőszám. A TFRC alapú szelektív újraküldés hatását a videó minőségére a következő ábrán mutatjuk be.



4. ábra MPEG videó minősége (PSNR)

A TFRC alapú újraküldés alkalmazásával a jel-zaj arány növekedett, úgy hogy közben a forrás sebessége nem haladta meg a TFRC küldési sebességét. Újraküldés nélkül az átlagos jel-zaj arány 15.86dB míg az általunk kidolgozott módszerrel 16.6dB. A PSNR módszer az eredeti MPEG videóhoz hasonlítja sérült videó folyamat, emiatt az időbeni elcsúszást is természetesen hibának érzékeli. Érdeemes tehát más eljárásokkal együtt vizsgálni a videók minőségét, mivel egy adott pillanatban a kép minősége sokkal jobb is lehet, azonban az időbeni késés miatt a PSNR módszer csak kis különbséget érzékel. A következő ábra a 450. képkeretet mutatja, ahol PSNR csak kis különbséget mutat a (b) ábra javára.



(a) eredeti videó



(b) TFRC alapú újraküldéssel (PSNR=17.18dB)

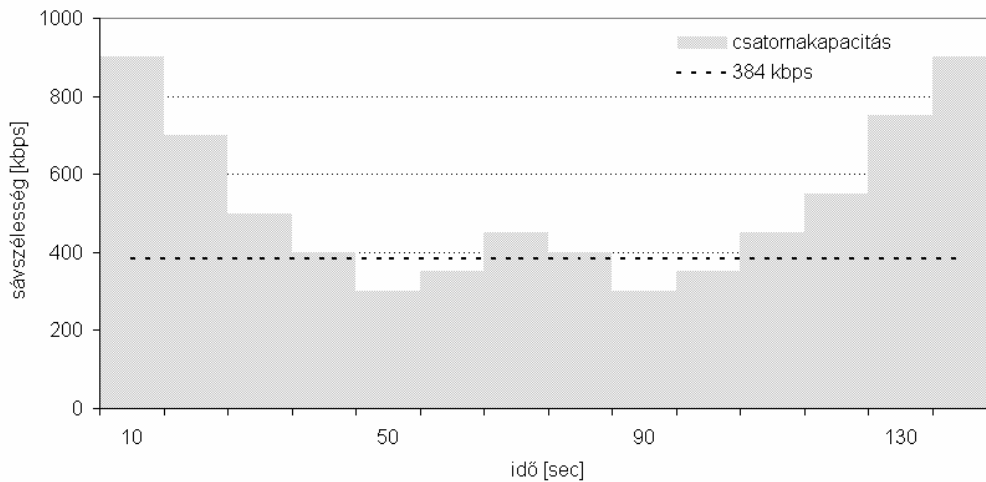


(c) újraküldés nélkül (PSNR=16.36dB)

5. ábra MPEG videó összehasonlítása

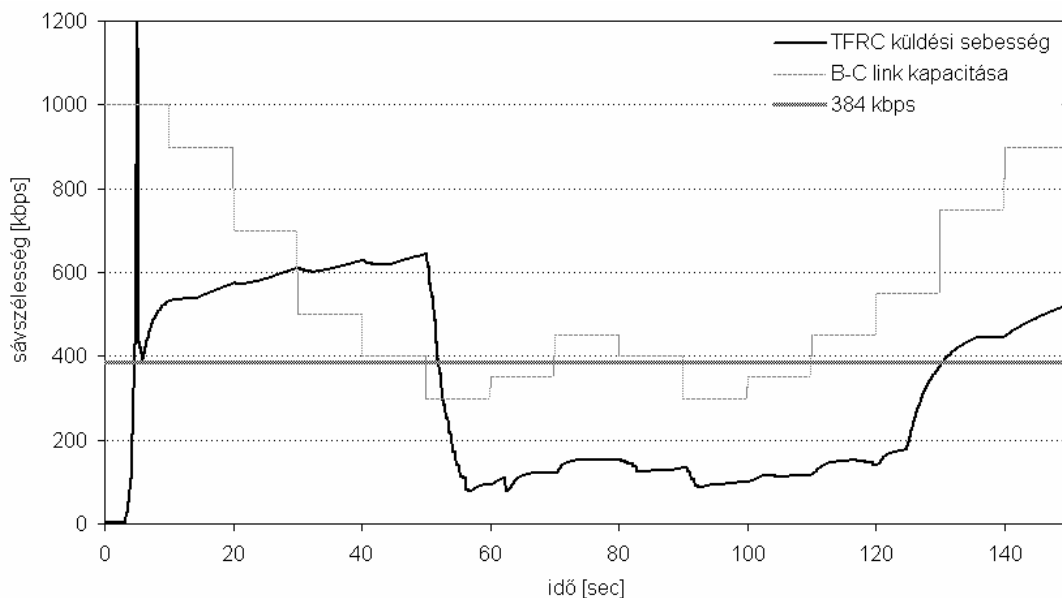
A következő vizsgálatban változó szabad kapacitás áll rendelkezésre, amely időnként 384kbps alá csökken, miközben a forrás pontosan ekkora sebességgel küldi a csomagokat a hálózatra. Ezekben a periódusokban tehát a B csomópont bufferje túlcsoordulhat, vagyis torlódás van a hálózatban. A TFRC pedig ennek megfelelően csökkenti a küldési sebességet. A vizsgálatban a $B-C$ link kapacitása a következőképp alakul:

A csatorna kapacitása



6. ábra A csatorna kapacitása

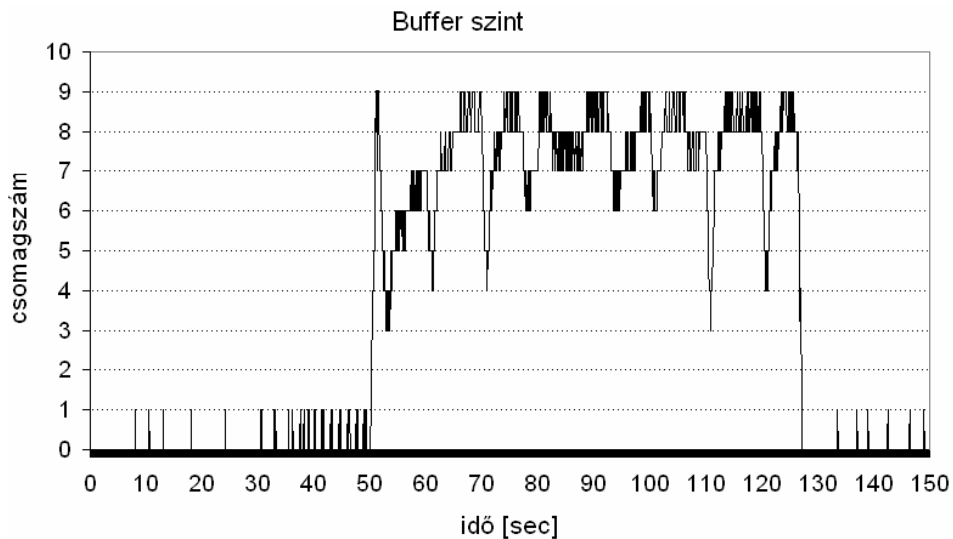
A csatorna hibavalószínűsége 0.1%, azonban a csomagvesztések nagy száma a buffer túlszordulása miatt történnek. Egy vizsgálat 150 másodpercig tart, amely alatt kb. 4800 felhasználói adatot tartalmazó csomag kerül továbbításra. A csatorna 0.1%-os hibavalószínűsége miatt közel 5 csomag veszik el, míg a buffer túlszordulás miatt több mint 50.



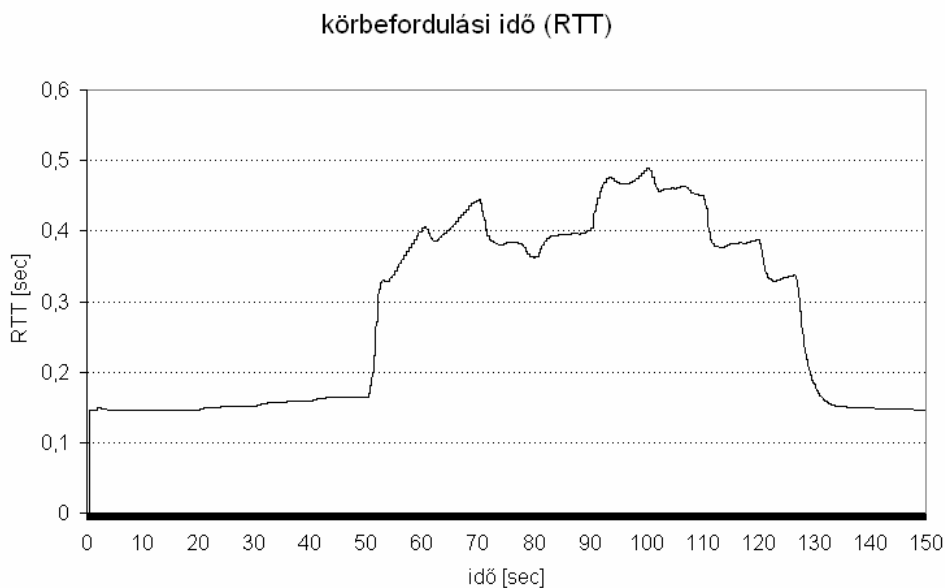
7. ábra Szabad sáv szélesség és TFRC sebesség

A TFRC küldési sebesség nem tér el jelentősen a 7. ábraán bemutatott eredményektől akkor sem, ha nincs csomagvesztés a csatornán. Szembetűnő, hogy a TFRC szinte azonnal a 384 kbps-os szint alá csökkentette a küldési sebességet, ahogy a szabad sáv szélesség kisebb

lett ennél a szintnél. A sebességnövelő szakaszon viszont sokkal visszafogottabban nő a küldési sebesség. Ennek az oka az, hogy a becsült körbefordulási idő értéke csak lassan csökken, részben a számítási módja miatt, másrészt mivel a routerek bufferjei lassan ürülnek ki. A következő ábra a *B* csomópont telítettségét mutatja, mely a 10. csomagot már nem tudja tárolni, így az elveszik. A 9. ábra jól mutatja, hogy a buffer telítettségi szintje és az RTT értéke szorosan összefügg.



8. ábra A *B* csomópont bufferjének telítettsége

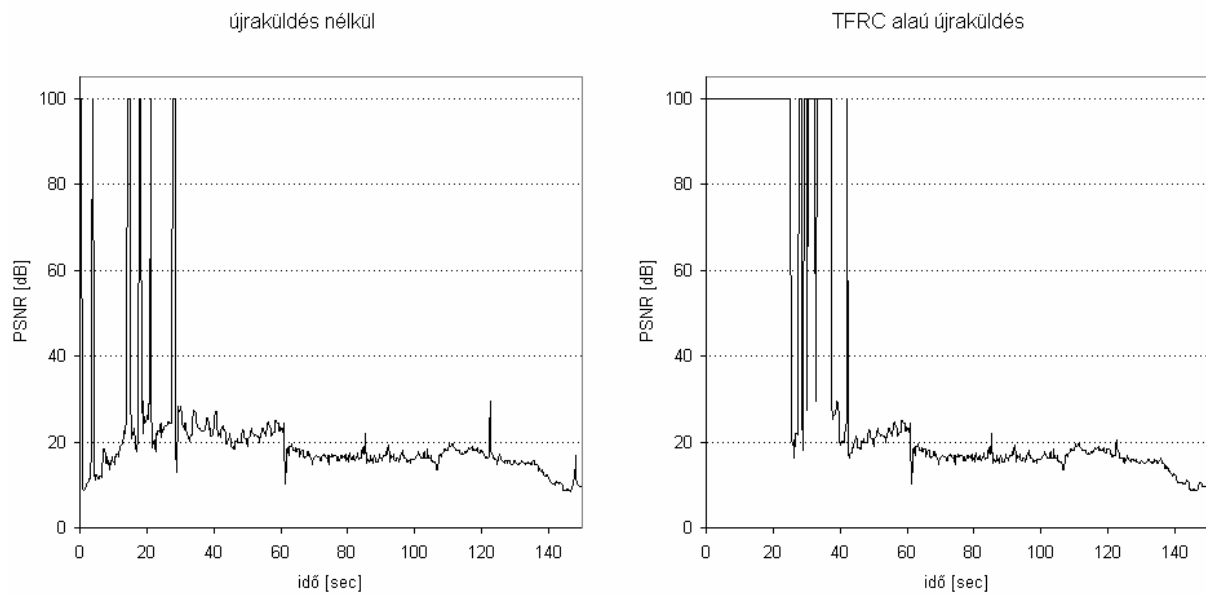


9. ábra Körbefordulási idő alakulása

A TFRC alapú újraküldés algoritmus a 7. ábra alapján az 50-130sec intervallum kivételével lehetőséget ad a hiányzó csomagok újraküldésére. Megfigyelhető, hogy az RTT értéke is az 50-130sec intervallumban tér el jelentősen a kezdeti értékektől. Ebben az

intervallumban hiába küldenék újra a csomagokat, azok csak tovább növelnék a hálózat terheltségét. Elképzelhető, hogy az újraküldött csomag kis valószínűséggel megérkezik a vevő oldalra, azonban nagy valószínűséggel másik csomag elvesztését okozza, így valójában nem javít a szolgáltatás minőségén.

A vizsgált változó kapacitású hálózatban ismét javítani tudtunk a videó minőségén. A PSNR módszerrel történt összehasonlítás eredménye a következő ábrán látható.



10. ábra Videó minőségének összehasonlítása

A minőség javítása az 50. másodperc előtti szakaszon nagyon hatékony, hiszen itt minden hibás csomag újraküldhető, ezáltal időbeni csúszásra sem kerül sor. A 130. másodperc után volt ismét lehetőség újraküldésre, azonban a PSNR ezt nem tudta kimutatni. Szemmel azonban jól érzékelhető a különbség. A TFRC alapú újraküldéssel az átlagos PSNR értéke 36dB, újraküldés nélkül pedig 19.9dB. A nagy különbség az 50. másodperc előtti szakasz hibátlan átvitelének köszönhető.

5. Összefoglalás

A TFRC alapú szelektív újraküldés lehetővé teszi jobb minőségű szolgáltatások nyújtását, olyan alkalmazások esetén, amelyek nem igényelnek megbízhatóságot, vagyis a csomagvesztést kezelni tudják. Vizsgálatainkat állandó sebességű MPEG típusú videó folyam esetére végeztük el, azonban az eljárás használható változó sebességű egyéb forrás esetén is. Az eredmények megmutatták, hogy jelentős javulást tudunk elérni a videó minőségében, a hálózati szabad kapacitás figyelembevételével. Ezáltal úgy javítunk a végleges

csomagvesztési arányon, hogy a megosztott csatorna többi felhasználóinak kommunikációját ez nem zavarja.

A jövőbeni tervek között szerepel az idő alapú szelektivitás vizsgálata. Valós idejű multimédiás szolgáltatás esetén fontos tényező a csomagok késleltetése, ezért az újraküldés csak akkor lesz eredményes, ha a csomag még időben megérkezik a vevőhöz.

6. Irodalom

- [1] J. Postel: "User Datagram Protocol", RFC-768, August 1980.
- [2] Larzon, Degermark, Pink: "The Lightweight User Datagram Protocol", RFC-3828, July 2004.
- [3] R. Stewart: "Stream Control Transmission Protocol", RFC-2960, October 2000.
- [4] Kohler, Handley, Floyd: "Datagram Congestion Control Protocol", RFC-, March 2006
- [5] S. Floyd, E. Kohler: "Profile for DCCP Congestion Control ID 2: TCP-like Congestion Control", draft-ietf-dccp-ccid2-10.txt, 10 March 2005
- [6] S. Floyd, E. Kohler, J. Padhye: "Profile for DCCP Congestion Control ID 3: TFRC Congestion Control", draft-ietf-dccp-ccid3-10.txt, 10 March 2005
- [7] J. Postel: "Transmission Control Protocol", RFC-793, September 1981.
- [8] Huszák Á, Imre S.: "Selective Retransmission of MPEG Video Streams over IP Networks", 5th International Symposium on Communication System Networks and Digital Signal Processing, CSNDSP 2006, Patras, Görögország, 2006. július 18-23., ISBN 960-89282-0-6
- [9] Naoki Wakamiya, Masaki Miyabayashi, Masayuki Murata, and HideoMiyahara: "MPEG-4 Video Transfer with TCP-Friendly Rate Control", LNCS 2216, p. 29 ff.
- [10] Ns2 – Network Simulator, <http://www.isi.edu/nsnam/ns/index.html>