

VIDEÓ FOLYAM SZELEKTÍV ÚJRAKÜLDÉSE IP HÁLÓZATOKBAN

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 egyre inkább tért hódító multimédiás alkalmazások Interneten való elterjedését a rendelkezésre álló sávszélesség korlátozza. Az újonnan megjelenő hozzáférési azonban képesek lesznek kielégíteni az új igényeket. Ilyen technológia például a nemrégben megjelent WiMAX (IEEE 802.16) [1], amely akár ötven kilométer sugarú körben, 70 Mbit/s sebességen kínál vezeték nélküli internetes hozzáférést. Ennek okán mindenképpen nagymértékű növekedésre kell számítani az IP alapú audio/vidéó alkalmazások terén.

Ugyanez a növekedési tendencia figyelhető meg a mobil eszközök elterjedése terén is. Jelenleg előrehaladott fejlesztések folynak adatszolgáltatások biztosítására vezeték nélküli hálózatokban, és a végcél a mobil kommunikáció és az Internet teljes konvergenciája, a mobil Internet kifejlesztése. 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-, vagy képátvitel. Természetesen az utóbbiaknál is törekedni kell arra, hogy minimális legyen a hibaarány. A kisebb hibákat az emberi érzékszervek nem 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.

A dolgozat következő fejezetében megismerhetjük a jelentősebb szállítási rétegbeli protokollokat, amelyek alkalmasak lehetnek az audio/videó folyam továbbítására. Ezt követően pedig az MPEG formátum jellemzőit mutatjuk be. A negyedik fejezetben az általunk kidolgozott algoritmust mutatunk be, mely egy új transzport protokoll (DCCP) alkalmazásával javítani képes az MPEG típusú multimédiás adatfolyam minőségén a szelektív újraküldési technika segítségével. Az azt követő fejezetben analitikus módszerekkel vizsgáljuk az algoritmus hatékonyságát. A hatodik fejezetben pedig levonjuk a következtetéseinket és ismertetjük a továbblépési lehetőségeket.

2. Transzport protokollok

A szállítási réteg két nagy protokollja a TCP (Transmission Control Protocol) [2] és az UDP (User Datagram Protocol) [3]. Ezek a protokollok a mai napig meghatározói a számítógépek valamint számítógép-hálózatok közötti adattovábbításnak, pedig immár 25 éves szabványok. Ez az oka annak, hogy mindkét szabványt vezetékes hálózatra dolgozták ki, azonban a ma egyre szélesebb körben használt vezeték nélküli hálózatok karakterisztikái jelentősen különböznek vezetékes hálózatok adatátviteli tulajdonságaitól. Emiatt a régi protokollokat felül kell vizsgálni, és megfelelően módosítani illetve kiterjeszteni.

A mai Internet protokollokat olyan vezetékes összeköttetésekre dolgozták melyeknek a jellemzőik a következők: nagy sáv szélesség, kis késleltetés, kis hibavalószínűség.

Az elmúlt időszakban megjelent és egyre népszerűbb vezeték nélküli hálózatok átvitelére azonban pont az ellenkező tulajdonságok jellemzőek: kisebb sáv szélesség, nagy késleltetés, nagy hibavalószínűség.

A két protokoll, a TCP és az UDP, valójában nagyon különböző célt szolgálnak. A TCP egy kapcsolatorientált és megbízható adatforgalmat biztosít a felhasználónak, míg az UDP megbízhatatlan. A TCP esetében a megbízhatóság azt jelenti, hogy az elküldött csomagok biztosan megérkeznek, de az esetleges újraküldések miatti késleltetésre nincs garancia, míg UDP esetén a küldő elküldi a csomagot és ezután a hálózaton múlik, hogy megérkezik-e. Újraküldésből adódó késleltetéssel ekkor nem kell számolni. Ezen jellemzői miatt az UDP-t kizárólag olyan esetekben alkalmazzák, ahol a küldött adatok egy részének elvesztése nem okoz működési problémát, sőt, esetleg kívánatos is (pl. torlódás esetén), mint pl. a műsorszórás; vagy ahol ezen hibák korrigálásáról egy magasabb szintű protokoll gondoskodik. Az 1. Táblázat tartalmazza a különböző alkalmazásokhoz leggyakrabban használt transzport protokollokat.

Alkalmazás	Alklam. réteg protokollja	Szállítási réteg
e-mail	SMTP	TCP
távoli hozzáférés	Telnet	TCP
Web	HTTP	TCP
file átvitel	FTP	TCP
távoli file server	NFS	UDP
multimédia	egyedi	UDP
IP telefónia	egyedi	UDP
hálózat menedzsment	SNMP	UDP
útvonalválasztás - routing	RIP	UDP

1. Táblázat Alkalmazások és protokolljaik

A közelmúltban több új transzport protokollt fejlesztettek ki, melyek megpróbálják kiküszöbölni a régebbi protokollok hibáit. Ilyen az UDP módosított változata az UDP Lite (Lightweight User Datagram Protocol) [4] és a DCCP (Datagram Congestion Control Protocol) [5].

2.1 User Datagram Protocol (UDP)

Az IP protokoll csak két gép közötti adattovábbítást biztosítja. Nem teszi lehetővé az alkalmazások vagy a felhasználók azonosítását. Az UDP szállítási protokoll biztosítja, hogy egy gépen egyidejűleg futó több alkalmazói program egymástól függetlenül küldhessen és fogadhasson csomagokat.

Az UDP sokkal gyorsabb protokoll, mint a TCP protokoll, viszont nem megbízható adatátvitel szempontjából. Nem kapcsolat orientált, nincs hibajavítás, nincs nyugtázás. Tulajdonképpen az IP szint által biztosított szolgáltatásokat nyújtja felfelé. Akkor szokták használni, ha az adatátvitel sebessége a legfontosabb, minden többi feladatot a felette elhelyezkedő réteg lát el. Tipikusan a DNS-ek (Domain Name Server), real-time alkalmazások, játékok szokták használni (egy játékban vagy real-time hang átvitel esetén, ha egy csomag rossz, akkor ott legfeljebb döccen egyet, de ez még mindig kisebb baj, mintha az adott pontnál megállna és onnantól elkezdené újra adni a csomagokat). A szegényesebb szolgáltatásból adódóan sokkal egyszerűbb az UDP fejléc.

Az UDP esetében is felmerül a mobilitásból adódó változó hibaarány, ami időnként a kapcsolat teljes megszakadásához vezet. A nyugtázásra ugyan nem kell várni, de a változó csatornaminőség és a cellaváltások itt is komoly gondot okoznak.

A hibaarány növekedéséből adódó csomagvesztésre legegyszerűbb módon úgy lehet védekezni, hogy a hatékonyabb FEC (Forward Error Connection) hibavédő kódolást alkalmazunk. Ezzel azonban növeljük az átvendő adatmennyiséget, nagyobb lesz az overhead, és a hálózati terhelés is növekszik.

2.2 Lightweight User Datagram Protocol (UDPLite)

Az UDP módosításával létrehoztak egy újabb protokollt az UDP-Lite-t [RFC 3828] amely valójában az új valósídejű multimédiás szolgáltatások kiszolgálására jött létre. Az új protokoll csupán annyiban változott az eredeti UDP-hez képest, hogy egy új részleges ellenőrző-összeget vezettek be. Amennyiben a csomagnak abban részében keletkezik bithiba, amelyet a részleges ellenőrző-összeg lefed, akkor a vevő érzékeli a hibát és eldobja a csomagot, míg ha a hiba olyan helyen van, amit a részleges ellenőrzőösszeg nem fed le, akkor nem dobja el. Ebben az esetben az alkalmazásnak kell kezelnie a hibás csomagot. Ezzel az eljárással kb. 40%-kal csökken a csomag eldobások száma. Abban az esetben, ha az UDP ellenőrzőösszege az egész csomagra kiterjed, az UDP Lite működése megegyezik a hagyományos UDP működésével.

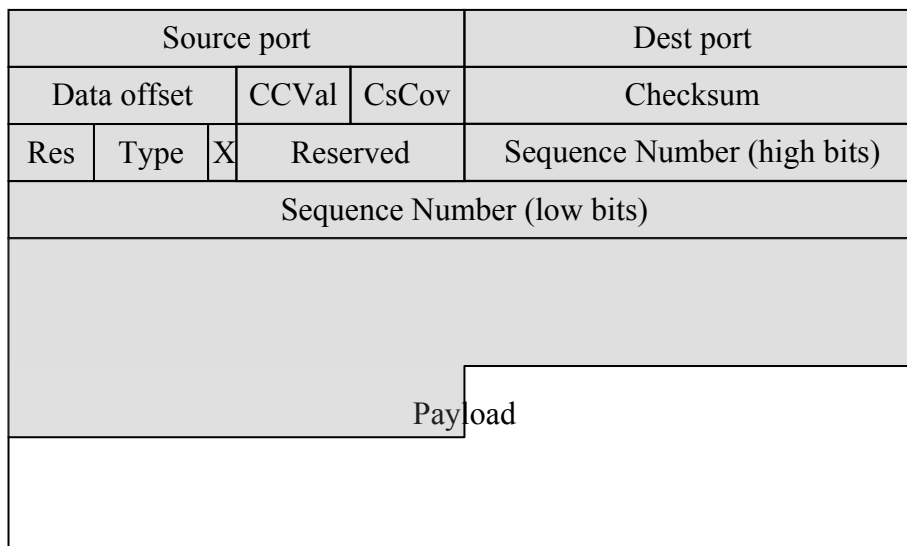
Egyik legjelentősebb ok, amiért létrehozták az UDP Lite protokollt, mert az alkalmazások egy csoportja kezelni tudja a hibás csomagokat is. A felhasználó által megfigyelhető minőség jobb lesz, ha a hibás csomagok nem kerülnek eldobásra, hanem az alkalmazásig eljutnak. Több hang és video codec is ehhez az alkalmazás csoportba tartozik: ITU-H.263, ITU-H.265, MPEG-4 video codec (ISO-14496). Ezek a kódolók jobb minőséget nyújtanak hibás csomagok kezelésével, mint ha egyáltalán nincs is csomag.

A csomagoknak nem minden byte-ja egyformán fontos, ezért a fontosabb byte-okat szükséges jobban védeni. Az Unequal Forward Error Correction is ezt a célt szolgálja. Egy rádiós csatornán, ahol a hibavalószínűség már jelentős, olyan szállítási rétegbeli protokollra van szükség, amely alkalmazkodik a csatorna tulajdonságaihoz éppúgy, mint az alkalmazásokhoz.

A szállítási réteg hibaérzékelő mechanizmusának, tehát védenie kell az alapvető információkat, mint a fejléc, de szükség szerint az adatok egy részét vagy egészét is. Annak eldöntése, hogy mely adatok fontosak, és melyek nem, azt az alkalmazásoknak kell eldönteniük. Annak meghatározása, hogy mely adatokat kell lefedni ellenőrző összeggel, a küldő oldali alkalmazás feladata.

2.3 Datagram Congestion Control Protocol (DCCP)

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 DCCP-t úgy próbálták alakítani, hogy a TCP és az UDP előnyeit egy protokollként valósítsák meg. A DCCP fejlécben így ráismerhetünk az előbbi protokollokból ismert fejlécmezőkre (1. ábra).



1. ábra A DCCP fejléce

A fejléc hossza minimálisan 12 byte, maximálisan pedig a 1024 byte-t is elérheti, ha az opcionális mezőket, és az egyes csomagtípusok 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. Az 1. ábrán egy olyan esetet mutattunk be, ahol nem a teljes csomag volt lefedve az ellenőrzőösszeg által, a lefedett területet szürkével jelöltük. Természetesen az is lehetséges, hogy az adatok egyáltalán ne legyenek lefedve, csupán a fejléc. Ez lehetőséget ad arra, hogy azok az alkalmazások, amelyek képesek kezelni a sérült adatokat, hatékonyabban működjenek. Arra azonban mindig ügyelni kell, hogy csupán a fejléct követő adatok fedhetőek le.

A DCCP működését tekintve összetettebb, mint az UDP, de valójában egy nagyon hatékony szállítási rétegbeli protokoll. A különböző működési módoknak köszönhetően a felhasználói alkalmazásokhoz kiválóan alkalmazkodó protokoll. Igen fontos tulajdonsága a

protokollnak, hogy a vevő oldal nyugtákat küld az adó oldalra, így a csomagvesztésre fény derül, hiszen e nélkül a torlódásszabályozó algoritmus sem működne.

A kapcsolatorientált DCCP a kapcsolat felépítése során megbízható protokollként működik. A torlódásszabályozással kapcsolatos üzenetek szintén megbízható adatfolyamként kerül továbbításra. Jelenleg két torlódásszabályozó algoritmust specifikáltak: TCP-like Congestion Control [CCID 2] és TFRC (TCP-Friendly Rate Control) Congestion Control [CCID 3]. 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. Az adatfolyam ugyan megbízhatatlan maradt, 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 képes annak meghatározására is, hogy milyen okból történt csomagvesztés. Ez az opció fontos lehet a torlódásszabályozó algoritmus számára, hiszen abban az esetben, ha például bithiba keletkezik, vagy a vevőoldali buffer túlsordulása miatt kerül sor csomageldobásra, nincs szükség torlódásszabályozó algoritmus beavatkozására.

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 [6].

3. MPEG formátum

Az MPEG (Moving Picture Experts Group) egy széles körben alkalmazott szabvány a videó (kép és hang) digitális adatfolyamának tömörítésére, az ehhez kapcsolódó specifikációkat tartalmazza. A különböző MPEG formátumok kidolgozása és fejlesztése folyamatos, az MPEG formátumokról kiforrott, de nem lezárt szabványok rendelkeznek.

Az MPEG tömörítés a videó/audió adatfolyam mértékét csökkenti, rendkívül számításigényes, veszteséges tömörítő eljárással (meghatározott keretek között skálázhatóan), kezelhető mértékűre. Az MPEG videó képváltási frekvenciája és az MPEG audió mintavételi frekvenciája szigorúan meghatározott és a különböző formátumok a különböző felhasználási területekhez igazodó adatátvitelt és minőséget célozzák meg:

- MPEG-1: CDROM alkalmazások, Video-CD, általában 1,5 Mbit/sec-ig
- MPEG-2: DVD, broadcast videó, telekommunikáció, többnyire 4-9 Mbit/sec
- MPEG-3: HDTV alkalmazások 1920x1080 felbontásig, 20-40 Mbit/sec
- MPEG-4: alacsony adatátviteli igényű alkalmazások; 176x144 felbontásig, 4,8-64 Kbit/s

Az MPEG videó képcsoportokból (GoP - Group Of Pictures) épül fel, amelyekben három különböző képtípus előre meghatározott sorrendben követi egymást (2. ábra GoP struktúra):

I-képek: Intra frame coded - csak képkockán belül kódolt. Csak olyan információt használ fel, amely a képkockán belül megtalálható. Az egyes képkockákon belül JPEG tömörítést alkalmaz és tömörítési arányuk viszonylag csekély.

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 videó egymást követő képkockáiban az objektumok alakja általában nem változik. Az MPEG a P (és B) képekben az objektumok elmozdulását mozgásvektorokkal írja le, a mozgáskompensációs eljárás segítségével. Az eltérő alak- és színinformációk meghatározása a megelőző (I, P) kép megfelelő információira épül, a különbséget, változást kódolja. A P-képek tömörítési aránya nagyobb, mint az I-kereté. Mivel a P-képek P-képből is származtathatók, előfordulhatnak közöttük hibásak is, és további hibaforrásul szolgálhatnak a hibás keretet referenciaként használó képkockák kódolásakor.

B-képek: a B (bidirectional) kép a megelőző és rákövetkező "I" vagy "P" képkockákat is felhasználja referenciaként. A kódolás a mozgáskompensációs 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.



2. ábra GoP struktúra

A GoP-n belül (2. ábra) az I és P képtípusok számát (amelyekből következik a B képek száma is) a kódoló határozza meg. Az I, P vagy B képek különböző aránya nagyban befolyásolja az elérhető képminőséget az adott adatátviteli sebesség mellett. A képcsoportban (GoP) az I képek frekvenciáját, másképpen fogalmazva a köztük lévő intervallumot jelölik N -nel. P illetve a megfelelő B képkockák frekvenciájának jele: M , a különféle tömörítők ezt nevezhetik *SubGoP*-nak is. Ha $N=1$, ekkor minden képkocka I-frame. Íme néhány példa:

GoP	N	M
IBPBPPBPBPI...	10	2
IPPPPPPPPPPPPI...	15	1
IBBPBBPBBPBBPBBPI...	15	3
IPPPPI...	5	1
IBPBI...	4	2
IBBI...	3	3
IPPI...	3	1

2. Táblázat Keretek gyakorisága

A következő fejezetekben a képcsoportok méretével fogunk dolgozni. A következő összefüggéssel számolható a GoP átlagos mérete az N és M valamint az I-, P-, B-keretek átlagos méreteinek felhasználásával.

$$size(GoP) = size(I) + \left(\frac{N}{M} - 1\right) \cdot size(P) + \left(N - \frac{N}{M}\right) \cdot size(B) \quad (1)$$

4. Szelektív újraküldési algoritmus

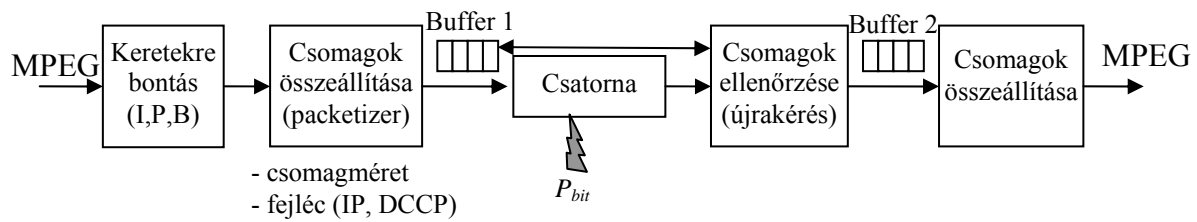
A mobil eszközök teljesítménye ma már lehetővé teszi nagyobb erőforrás-igényű multimédiás alkalmazások használatát. Azonban az általuk használt rádiós csatorna rendkívül változékony, nagy hibavalószínűséggel rendelkezik, amely komoly gondok elé állítja a tervezőket.

Az IP alapú csomagkapcsolt hálózatokban történő multimédiás adatátvitel esetén nem mindegyik szállítási rétegbeli protokoll felel meg. Rendkívül fontos ugyanis, hogy hiba esetén

az audio/videó folyam ne álljon le, várva a hibás csomagok újraküldésére. A nagy hibaarányal rendelkező rádiós csatornán ugyanis ez élvezhetetlenné tenné a szolgáltatást. Ez az oka annak, hogy a TCP nem alkalmas erre a feladatra, hiszen ennek a protokollnak az alapkövetelménye, hogy minden egyes csomagot hibamentesen eljuttasson a vevőhöz, nem törődve azzal, hogy az mennyi ideig fog tartani. A TCP sorszámokkal látja el a csomagokat, így téve lehetővé az újraküldést. Az UDP esetében már jobb a helyzet, hiszen itt pont az ellenkezője történik. A protokoll nem törődik azzal, hogy a csomag helyesen érkezik-e meg a vevőhöz, sokkal inkább az a fontos, hogy folyamatos legyen az adattovábbítás. Az UDPLite annyiban javít a helyzeten, hogy a sérült csomagok nem kerülnek eldobásra, ha csak nem a fejlécben történt a sérülés. Ezt a részleges ellenőrzőösszeg (partial checksum) használata teszi lehetővé. Az újraküldés sem az UDP, sem az UDPLite esetén nem lehetséges, hiszen a csomagokra nem lehet hivatkozni, ugyanis nincsenek ellátva a csomag azonosítására szolgáló sorszámmal.

A DCCP fejlécében megtalálható a csomag sorszám mező, így a csomagokra való hivatkozás is lehetséges. A sorszámozás a vevő oldalon történő sorba rendezéshez nélkülözhetetlen, hiszen nem garantált, hogy a csomagok sorrendben érkeznek meg. A DCCP nem írja elő a csomagok újraküldését, ugyanakkor értesül a csomagvesztésekről, a vevő által küldött nyugtákból. A szelektív újraküldési technika ezeket az előnyöket használja ki MPEG típusú multimédiás adatátvitel során.

Az alapötlet az volt, 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 audio/videó 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. Ebből a szempontból a DCCP összesíti a többi transzport protokoll előnyét, ezért választottuk ezt a protokollt a probléma megoldásához. Az általunk kidolgozott modell a 3. ábrán látható.



3. ábra A szelektív újraküldés modellje

Legelső modul az MPEG formátumú forrás adatait I-,P-,B-keretekre bontja, amelyeket a következő modul megadott méretű csomagokká állít össze. Itt történik a csomagok ellátása a fejlécekkel is (DCCP, IP), majd azok a csomagok, amelyek tartalmazzak I-kereteket, a *Buffer1* tárolóba kerülnek, hogy az esetleges újraküldés esetén innen lehessen kiolvasni azokat. A csomagokat a P_{bit} hibavalószínűségű csatornán továbbítjuk. A következő modul feladata a továbbított csomagok ellenőrzőösszegének vizsgálata és amennyiben hibát érzékel az I-keretet tartalmazó csomagban, újrakéri azt. A meghibásodás nélkül megérkezett csomagok a *Buffer2* tárolóba kerülnek. Az újraküldött csomagokat is ebbe a tárolóba illesztjük be, kihasználva a DCCP csomagok sorszámát. Mivel a sorszám egyértelműen azonosítja a csomagot, könnyedén tehetjük ezt meg. A következő modulban pedig összeállítjuk az MPEG folyamatot a *Buffer2*-ből kiolvasott csomagok alapján. Ezáltal egy jobb minőségű audio/videó folyamat kapunk, de cserébe körbefordulásnyi idővel (RTT – Round Trip Time) késleltetjük a multimédiás folyamatot. Természetesen semmilyen garancia nincs arra, hogy az újraküldött csomag hibátlanul érkezik meg, tehát e csomagok egy része ismét hibás lesz. A helyesen megérkezettek azonban javítják a médiafolyam minőségét.

Általános esetben egy csomag meghibásodásának valószínűségét a következő egyenlet adja meg, ahol P_{bit} a bithiba valószínűsége, amelyet a rádiós csatorna okoz, és az alsóbb rétegbeli protokollok nem tudnak javítani, CsM pedig a csomag mérete bitekben kifejezve:

$$P_{csomaghiba} = 1 - (1 - P_{bit})^{CsM} \quad (2)$$

A $P_{csomaghiba}$ UDP esetében a csomagvesztés, míg TCP esetében a csomag újraküldés valószínűségét jelenti. Az algoritmusunk használata esetén, DCCP protokoll alkalmazásával és annak részleges ellenőrzőösszeg (partial checksum) lehetőségét kihasználva, szintén ez lesz a csomag meghibásodásának valószínűsége, azonban a csomag újraküldésének valószínűségét a következő képlet adja meg, ahol a *Fejléc* a csomag fejlécének méretét jelenti (DCCP adatátvitel során 12 vagy 16 byte), míg az *Iframe* az I-kerethez tartozó információmennyiséget jelöli egy csomagon belül:

$$P_{\text{újra}k\ddot{u}ld\acute{e}s} = 1 - (1 - P_{\text{bit}})^{\text{Fejl\acute{e}c} + \text{Iframe}} \quad (3)$$

A képlet csak azokra a csomagokra vonatkozik, amelyben van I-kerethez tartozó adat. Természetesen a csomagon belül más kerethez tartozó adatok is lehetnek, azonban csak az I-kép adatait követően. A módszerünk alkalmazásához elengedhetetlen, hogy minden I-keret új csomagban kezdődjön, hiszen a DCCP protokoll részleges ellenőrzőösszege csak ebben az esetben használható. Annak a feladatnak az ellátását, hogy a fejléct követően I-kép adata következzen, a 3. ábraán ismertetett csomag-összeállító modul a végzi. A részleges ellenőrzőösszeg (partial checksum) alkalmazása csak ebben az esetben teszi lehetővé, hogy a fejléct és az azt követő I-keretekhez tartozó információkat védjük, a többi adattal pedig ne foglalkozunk. Ekkor az I-képet tartalmazó csomag újra küldési valószínűsége alulról és felülről is korlátozott:

$$1 - (1 - P_{\text{bit}})^{\text{CsM}} \geq P_{\text{újra}k\ddot{u}ld\acute{e}s} > 1 - (1 - P_{\text{bit}})^{\text{Fejl\acute{e}c}} \quad (4)$$

Ahol a CsM a csomagméretet jelöli, amelynek értéke a torlódáskezelő algoritmustól függ. A vizsgálataink során ennek értékét a számítások egyszerűsítése miatt állandónak tekintjük.

Az előbb leírtakból következik, hogy minél ritkábban vannak I-keretek, vagyis N (I-képek gyakorisága) minél nagyobb, annál kisebb valószínűséggel lesz szükség a csomag újra küldésére. Az N paraméter növelésével azonban a hang és kép minősége romlik, cserébe viszont nagyobb tömörítést érhetünk el.

A multimédiás folyamátvitel során a csomagok újra küldésére akkor kerül sor, ha a csomagban megsérül a fejléc vagy az I-kerethez tartozó adat, feltéve hogy I-keretet tartalmaz a csomag:

$$P_{\text{újra}k\ddot{u}ld\acute{e}s} = P_{\text{hiba}} \cdot P_{\text{Iframe}} \quad (5)$$

ahol a P_{hiba}

$$P_{\text{hiba}} = 1 - (1 - P_{\text{bit}})^{\frac{\text{CsM} - \text{Fejl\acute{e}c}}{\text{size}(\text{GoP})} \cdot \text{size}(\text{I}) + \text{Fejl\acute{e}c}} \quad (6)$$

Itt azzal a feltételezéssel élünk, hogy az egy csomagra jutó átlagos I-keret információ:

$$(\text{CsM} - \text{Fejl\acute{e}c}) \cdot \frac{\text{size}(\text{I})}{\text{size}(\text{GoP})} \quad (7)$$

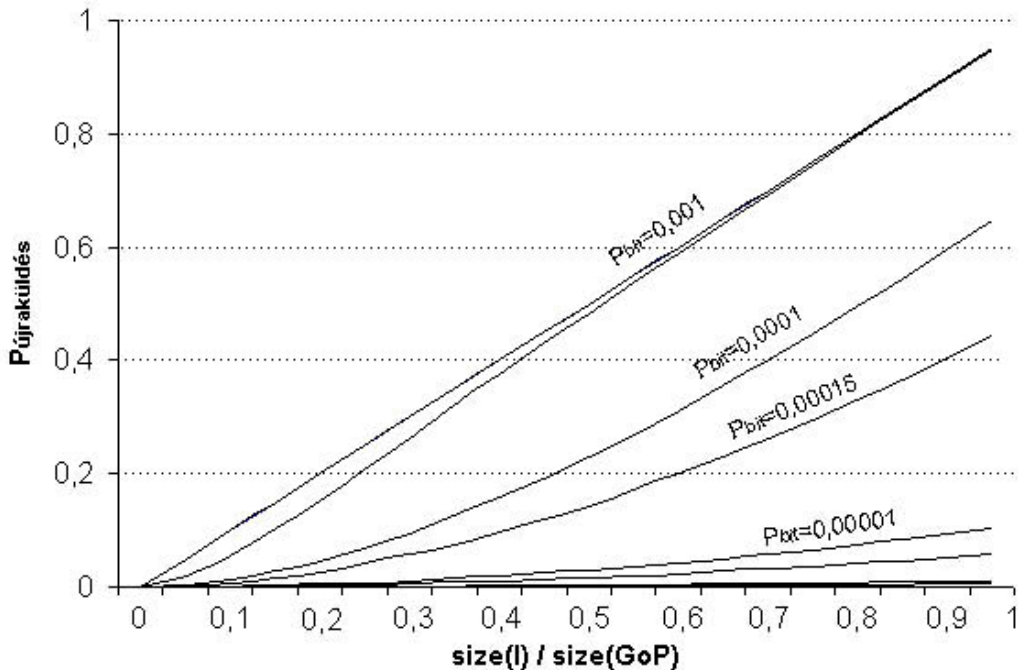
Annak valószínűsége, hogy I-képet tartalmaz a csomag:

$$P_{\text{Iframe}} = \frac{\text{size}(\text{I}) + \left[\frac{\text{size}(\text{I})}{\text{CsM} - \text{Fejl\acute{e}c}} \right] \cdot \text{Fejl\acute{e}c}}{\text{size}(\text{GoP}) + \left[\frac{\text{size}(\text{GoP})}{\text{CsM} - \text{Fejl\acute{e}c}} \right] \cdot \text{Fejl\acute{e}c}} \approx \frac{\text{size}(\text{I})}{\text{size}(\text{GoP})} \quad (8)$$

A fejléc és a csomag adatmezőjének aránya kicsi, ezért élhetünk az egyszerűsítéssel. Összevonva az egyenleteket, a következőt kapjuk:

$$P_{\text{újra}} = \frac{\text{size}(I)}{\text{size}(GoP)} \cdot (1 - (1 - P_{\text{bit}})^{\frac{CsM - \text{Fejléc}}{\text{size}(GoP)} \cdot \text{size}(I) + \text{Fejléc}}) \quad (9)$$

Az utolsó egyenlet felhasználásával kimutatható, hogy hogyan befolyásolja az I-kép és a teljes GoP méretének aránya az újraküldés valószínűségét, az algoritmusunk használata esetén (4. ábra).

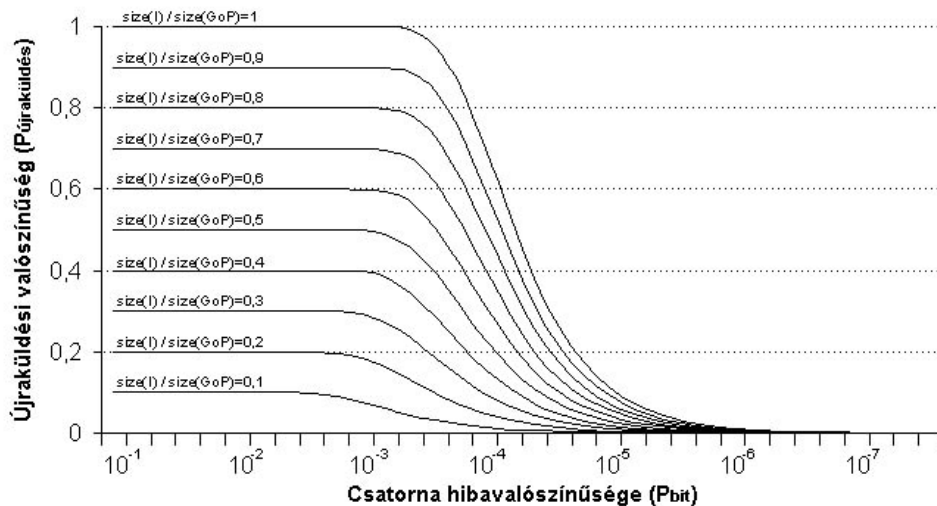


4. ábra Újra küldés valószínűsége az I-keret és a GoP méretének függvényében

A 4. ábra egyértelműen megmutatja, hogy ahogyan növeljük az I-keretek méretét, vagyis a videó folyam minőségét növeljük, úgy növekszik az újraküldés valószínűsége. Szélsőséges esetben, amikor csak I-képből áll a GoP az újraküldés valószínűsége:

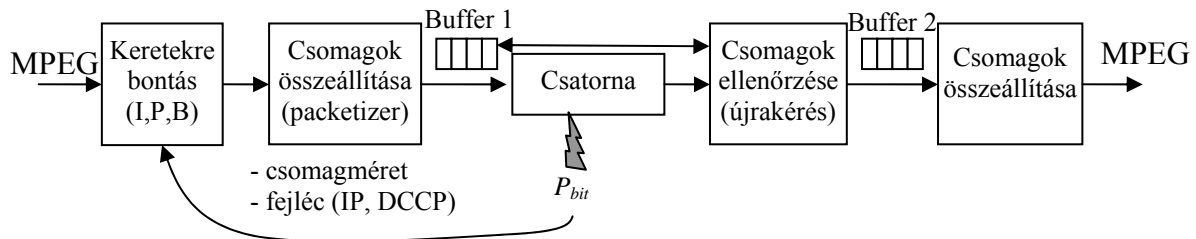
$$P_{\text{újra}} = 1 - (1 - P_{\text{bit}})^{CsM} \quad (10)$$

A (9) képlet alapján rá tudunk mutatni a csomagújra küldés valószínűsége és a bithiba valószínűsége közötti összefüggésre, amelyet az 5. ábrán mutatunk be. A bithiba csökkenésével természetesen az újraküldés valószínűsége is csökken, hiszen így kisebb valószínűséggel történik hiba az általunk fontosnak tekintett adatokban, vagyis az I-keretet tartalmazó csomagok, I-képhez tartozó adataiban illetve ezek fejlécében.



5. ábra Csomagújraküldés valószínűsége a bithiba valószínűségének függvényében

Ezen összefüggések felhasználásával akár fix újraküldési valószínűséget biztosíthatunk változó bithiba valószínűség mellett is, oly módon, hogy a fizikai rétegben mérhető P_{bit} paraméter átadásával módosítjuk az MPEG kódoló beállításait.



5. ábra A P_{bit} átadása az MPEG kódolónak

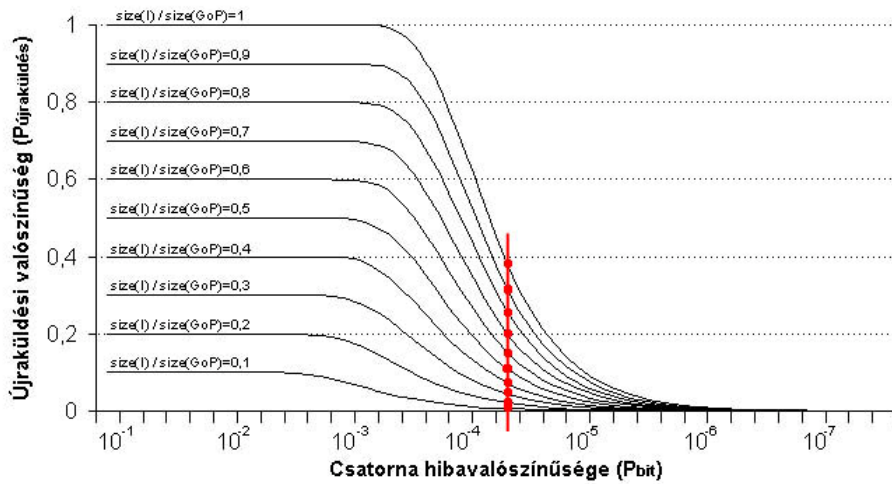
Ez a kódoló minőségére lesz hatással, hiszen a $\frac{size(I)}{size(GoP)}$ arány az N paraméterrel egyenes arányosságban változik. Az N paraméter az I-keretek gyakoriságát adja meg, ahogyan azt a 3. fejezetben már ismertettük. Természetesen nem tudunk tetszőleges újraküldési valószínűséget biztosítani, mivel egy felső korlátot jelent, amikor csak I-keretből áll a képcsoport, vagyis

$$\frac{size(I)}{size(GoP)} = 1, \quad (11)$$

így tehát minden P_{bit} mellett megadható egy maximális újraküldési valószínűség:

$$P_{újraküldés} \leq 1 - (1 - P_{bit})^{CsM}, \quad (12)$$

Ami a következő ábrából is jól látszik



6. ábra A maximális újraküldési valószínűség

5. Analitikus vizsgálat

Ebben a fejezetben analitikus módszerekkel vizsgáljuk meg a szelektív csomagújraküldés hatékonyságát. Megvizsgáljuk, hogy egy körbefordulási idő (RTT) késleltetés bevetésével, amelyre az egyszeri újraküldés miatt van szükség, hogyan változik a vevő oldali bithiba valószínűsége. Először a hagyományos UDP által továbbított multimédiás folyamatot vizsgálunk meg, majd pedig az UDP helyett UDP Lite-t vagy DCCP-t alkalmazunk. Legvégül pedig az általunk megtervezett algoritmust hasonlítjuk össze az előzőekkel.

Abban az esetben, ha az UDP protokollt használjuk, az ellenőrzőösszeg a teljes csomagot fedni fogja, emiatt egyetlen hiba a csomagban a teljes csomag eldobását jelenti. Ebből egyértelműen látszik, hogy egy bithiba a csomagban, annyi bit elvesztését, vagyis sérülését okozza, amekkora a csomag mérete volt. Ez azt jelenti, hogy $P_{csomaghiba}$ valószínűséggel veszítünk csomagméretnyi bitet, és $1 - P_{csomaghiba}$ valószínűséggel nem sérül meg egy bit sem. Egy hosszú multimédiás folyamaton esetén a hibás bitek várható értéke csomagonként:

$$E(\text{hibás_bitszám})_{UDP} = P_{csomaghiba} \cdot CsM + (1 - P_{csomaghiba}) \cdot 0 = (1 - (1 - P_{bit})^{CsM}) \cdot CsM \quad (13)$$

Abban az esetben, amikor UDP Lite, illetve DCCP protokollt alkalmazunk, azok részleges ellenőrzőösszeg tulajdonságának kihasználásával, természetesen azt várjuk, hogy a hibás bitek számának várható értéke jelentősen kisebb legyen, hiszen a teljes csomag eldobására csak abban az esetben kerül sor, ha a fejléc megsérült. Az adatmezőben keletkező hibák egyszeres hibaként jelentkeznek. Figyelembe kell vennünk, hogy a fejléchihiba esetén már nem számít az adatmezőben keletkező hiba, hiszen ebben az esetben, az adatmezőben

történő hiba nem növeli a hibás bitek számát, mert a csomagot teljes egészében hibásnak tekintjük. Így tehát a hibás bitek számának várható értéke a csomagban:

$$E(\text{hibás_bitszám})_{UDPLite} = P_{\text{csomaghiba}} \cdot CsM + P_{\text{bit}} \cdot (CsM - Fejléc) - P_{\text{csomaghiba}} \cdot P_{\text{bit}} \cdot (CsM - Fejléc), \quad (14)$$

ahol a $P_{\text{csomaghiba}}$ a korábbi egyenletekből már ismert, így

$$E(\text{hibás_bitszám})_{UDPLite} = (1 - (1 - P_{\text{bit}})^{Fejléc}) \cdot CsM + P_{\text{bit}} \cdot (CsM - Fejléc) - (1 - (1 - P_{\text{bit}})^{Fejléc}) \cdot P_{\text{bit}} \cdot (CsM - Fejléc) \quad (15)$$

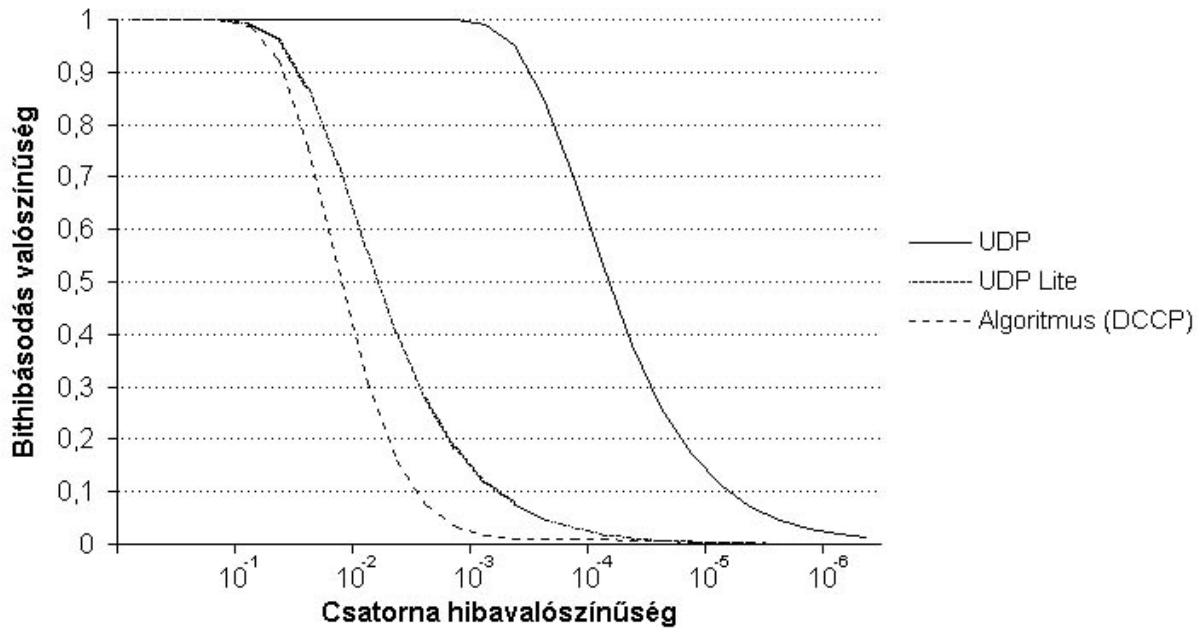
Ezenfelül azt is tudjuk, hogy a csomag mérete (CsM) UDP esetén 8 byte, míg DCCP esetén minimálisan 12 vagy 16 byte, attól függően, hogy milyen hosszú csomagsorszámokat használunk.

Vizsgáljuk meg mekkora lesz a hibás bitek várható értéke a szelektív újraküldés alkalmazása esetén. Ez annyiban tér el az előző esettől, hogy a kiemelt jelentőségű helyen keletkezett bithiba a csomag újraküldését eredményezi. Természetesen az újraküldött csomag is sérülhet, illetve elveszhet a fejléc sérülése miatt. A vizsgált algoritmusban csak egyszeri újraküldéssel számolunk, a többszörös újraküldéssel most nem foglalkozunk. A hibás bitek száma hasonlóan számolható, mint azt UDPLite esetében, vagyis az előző esetben számolt hibás bitek várható értékéhez, hozzá kell adni az újraküldés után továbbra is sérült bitek számát, és ki kell vonni azokon a bitek számát, amelyeket újraküldtünk:

$$E(\text{hibás_bitszám})_{SzelÚjrakAlg} = E(\text{hibás_bitszám}) + P_{\text{újraküldés}} \cdot E(\text{hibás_bitszám}) \cdot E(\text{hibás_bitszám}) - P_{\text{újraküldés}} \cdot E(\text{hibás_bitszám}) \quad (16)$$

Abban az esetben, amikor a nincs I-képhez tartozó adat a csomagban, a képlet alapján kiadódik az UDPLite esetben számolt érték. Megfigyelhető az is, hogy minél nagyobb arányban szerepelnek I-keretek a csomagokban, annál kisebb lesz a hibás bitek várható értéke. Ez azzal magyarázható, hogy ekkor növekszik az újraküldés valószínűsége, és az újraküldött csomagoknak egy része helyes érkezik meg, melyek előzőleg mind hibásak voltak. A következő ábrán a dekódolóba kerülő bitek hibaarányát mutatja, abban az esetben, amikor a teljes képcsoport méretének felét I-keret teszi ki:

$$\frac{\text{size}(I)}{\text{size}(GoP)} = 0,5 \quad (17)$$



7. ábra Bithiba valószínűség a dekódoló bemenetén

Az 7. ábra jól mutatja, hogy a legtöbb hiba az eredeti UDP esetén keletkezik, ami a teljes csomag eldobása miatt jelentkezik, egyetlen bithiba miatt. Az UDPLite sokkal jobban teljesít, hiszen csak a fejléc hibája okozza a teljes csomag elvesztését. A szelektív újraküldés alkalmazásával tovább csökkenthetjük a hibás bitek számát, kétszeres körbefordulási idő késleltetés árán. Ez a késleltetés azonban elfogadható, a vevő oldali videó minősége azonban jelentősen javul, hiszen nem csupán a hibás bitek számát csökkentettük, hanem az I-keretekben keletkezett hibákat is visszaszorítottuk, amelyek a teljes képcsoport (GoP) minőségére hatással van. Egyetlen bithiba a kiemelt keretben végiggyűrűzik az összes kereten, egészen a következő I-képig. Ez tehát azt jelenti, hogy egyetlen bit helyreállítása ebben a keretben, N -szeres bithiba csökkenést jelent a képcsoportban, ami már nagyon jelentős minőségi javulást jelent. Mivel az I-kereten belüli hibajavítás valószínűsége egy csomagra:

$$P_{javít} = P_{újraküldés} \cdot P_{hibásUDPLite}^2 - P_{újraküldés} \cdot P_{hibásUDPLite} = P_{újraküldés} \cdot P_{hibásUDPLite} (P_{hibásUDPLite} - 1), \quad (18)$$

így a teljes képcsoportra vetítve a kijavított bitek várható értéke:

$$E(jav_bit)_{GoP} = \frac{size(I)}{size(GoP)} \cdot P_{javít} \cdot N \quad (19)$$

A javított bitek okozta minőségjavulás számottevő, ami élvezhetőbbé teszi a multimédiás szolgáltatásokat a mobil eszközökben.

6. Összefoglalás

A szelektív újraküldés jelentősen emelheti a multimédiás szolgáltatások minőségét MPEG audio/videó átvitele esetén. Az algoritmus kifejezetten mobil környezetben hatékony, ahol a változó rádiós csatornán való adatátvitel csak magas hibaarány mellett valósítható meg. A MPEG formátum kiemelt I-kereteinek újraküldésével, jelentősen csökkenthető nemcsak e keretek bitjeinek hibavalószínűsége, hanem a teljes képcsoport hibavalószínűsége is, az I-képek hibája továbbgyűrűzik a többi kereten is.

Az analitikus vizsgálatot a szimulációs vizsgálatok fogják követni. Néhány DCCP implementáció már elérhető, ezért a közeljövőben egy valós teszhálózaton kívánjuk analizálni az algoritmust.

Ugyanakkor további kutatási területet jelent annak megvizsgálása, hogy az I-kereteken kívül a P-képeket is újraküldjük, hiszen e keretek hibái a B-képekben is megjelennek. Ezenkívül érdekesnek tűnik egy adaptív MPEG kódoló alkalmazása, annak érdekében, hogy az újraküldési valószínűséget állandó szinten tartsuk, változó rádiós csatornajellemzők mellett. Meg szeretnénk vizsgálni azt is, hogy mekkora késleltetést tudunk elfogadhatónak tekinteni, esetleges többszörös újraküldés esetén.

Az szelektív újraküldés alkalmazása egy jó módszer a mobil hálózaton történő multimédiás forgalom továbbítására, amely a DCCP rendkívüli rugalmasságát használja ki az MPEG folyam minőségének jelentős növelésére.

7. Irodalom

[1] <http://www.wimaxforum.org>

[2] J. Postel: "Transmission Control Protocol", RFC-793, September 1981.

[3] J. Postel: "User Datagram Protocol", RFC-768, August 1980.

- [4] Larzon, Degermark, Pink: "The Lightweight User Datagram Protocol", RFC-3828, July 2004.
- [5] Kohler, Handley, Floyd: "Datagram Congestion Control Protocol", draft-ietf-dccp-spec-11.txt, March 2005
- [6] Phelan: "Datagram Congestion Control Protocol - Lite (DCCP-Lite)", draft-phelan-dccp-lite-00.txt, August 2003