

A TCP/HSDPA rendszer átvitelének analitikus modellje

Bodrog Levente* Horváth Gábor† Vulkán Csaba‡

Kivonat

E cikkben a TCP átvitelét adjuk meg mobil, adatforgalmat nyújtó, HSDPA környezetben a Padhye modell alapján, a TCP csomagvesztési valószínűsége és a körbefordulási ideje segítségével. E két paramétert meghatározandó megalkottuk a HSDPA-t leíró sorbanállási hálózatot, amely tartalmazza a torlódási pontokat és protokollrétegeket, amelyek hatással vannak a vesztesre és a késleltetésre. Ennek a sorbanállási hálózatnak a megoldását részletezzük.

1. Bevezetés

A HSDPA (High Speed Downlink Packet Access) nagysebességű (akár több megabit másodpercenként) csomagkapcsolt, letöltésirányú szolgáltatást nyújt UMTS (Universal Mobile Telecommunications System) felett. [6]

Hagyományos UMTS esetén az adatkapcsolati rétegbeli protokollok – mint például Radio Link Control (RLC) és Medium Access Control (MAC) – a rádióhálózat-vezérlőben (Radio Network Controller - RNC) végződnek. A rádiós interfészt megvalósító protokollok az RNC-vel az Iub interfészen kapcsolódó bázisállomásban (3. generációs mobilhálózatok esetén ez a Node B) vannak megvalósítva. Nyugtázott módban (AM) az RLC felelős a hibamentes, sorrendhelyes átvitelért, amelyet az ARQ (Automatic Repeat Request - automatikus újraadás) mechanizmussal érnek el, ami azonban növeli a második rétegbeli körbefordulási időt, így TCP időtúllépéshez vezethet.

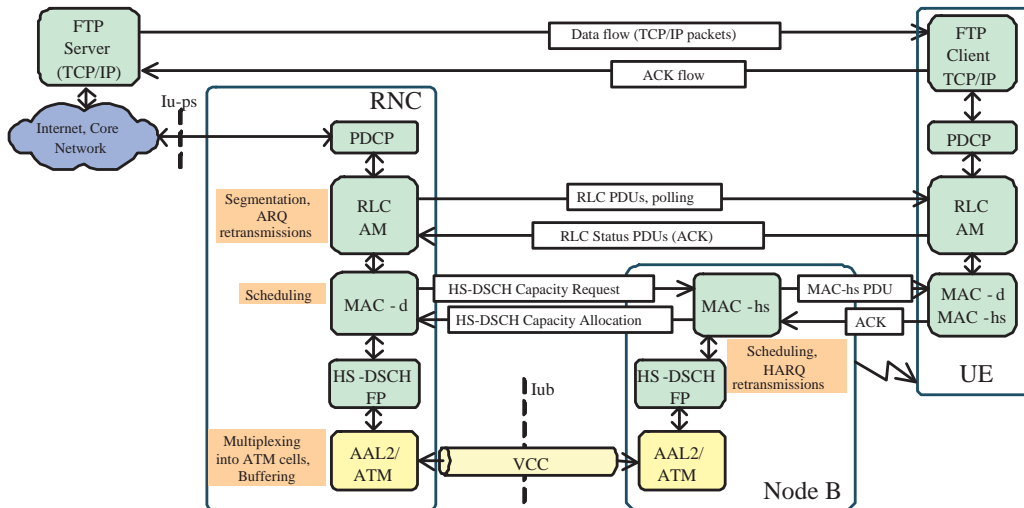
HSDPA esetén új protokollréteget – a MAC-hs réteget – vezettek be a bázisállomásban (lásd 1. ábra). Ennek segítségével a bázisállomás képes gyorsan alkalmazkodni a rádiós interfész aktuális állapotához modulációs és kódolási

*Budapesti Műszaki Egyetem, Híradástechnikai Tanszék, bodrog@hit.bme.hu

†Budapesti Műszaki Egyetem, Híradástechnikai Tanszék, ghorvath@hit.bme.hu

‡Nokia Siemens Networks, Budapest, csaba.vulkan@nsn.com

sémaváltással, gyors ütemezéssel és újraadással. Ez utóbbit a HARQ (Hybrid ARQ) mechanizmus valósítja meg. Ezek a megoldások segítik csökkenteni a második rétegbeli körbefordulási időt, ha az újraadást a hibás rádiós interfész feletti átvitel okozta. Habár ezzel a bázisállomás kezeli az újraadást, az RLC rétegbeli újraadás is megmaradt a Rel'99-es megoldásokkal való kompatibilitás, illetve a rendszeren belüli hívásátadás-vezérlés megtartása érdekében. Az RLC ugyanakkor továbbra is kezeli az újraadást, ha a MAC-hs újraadások száma elért egy megengedett legnagyobb számot, vagy ha a szállítási rétegben dobás volt. Így HSDPA esetén is növelheti az RLC a körbefordulási időt. Ezen megoldások azt is eredményezik, hogy a TCP képtelen megállapítani és kezelni a torlódást, csak ha már lejárt a TCP időzítője, vagy ha a csomag elérte az RLC újraadások legnagyobb megengedett számát is és azt az RLC eldobta.



1. ábra. A HSDPA protokollcsalád áttekintése

A rádiós interfész kezelésének az elosztása az RNC és a bázisállomás között egy áramlásvezérlési algoritmus beiktatását is maga után vont – HSDPA áramlásvezérlés [8]. Ennek az algoritmusnak a lényege, hogy a bázisállomás határozza meg az RNC által az egyes felhasználóknak küldött adat mennyiségét, úgy hogy a puffereket optimális szinten tartsa, azaz ne legyen sem a késleltetés túl nagy, de ne vesztegesse a rádiós interfész kapacitását sem. Ezt leggyakrabban a sorhossz mintavételezésével és az időegység alatt küldött csomagok (Packet Data Units - PDU) számának mérésével érik el.

Jól mutatja a HSDPA által nyújtott szolgáltatás színvonalát az elérhető TCP átvitel. Vizsgálták már a TCP teljesítményét HSDPA felett [3]-ban, [2]-

ben és [1]-ben, ahol a szerzők szimuláció alapú modell adtak. Ebben a cikkben mi az analitikus modelljét adjuk ugyanennek.

A cikk további része a következőképpen épül fel. A 2. szakaszban megadjuk a rendszer szűk keresztmetszeteit jelentő puffereleési pontokat és a belőlük felépített sorbanállási hálózatot, majd a 3. szakaszban összefoglaljuk a közelítő átvitelszámítást és megadjuk részletesen a sorbanállási hálózat megoldását, végül a 4. szakaszban összefoglaljuk az eredményeinket.

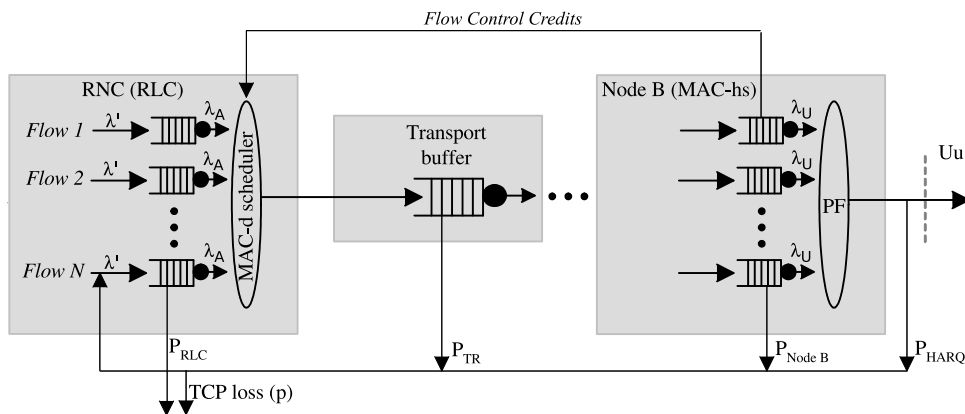
2. A rendszer áttekintése és annak sorbanállási hálózat modellje

A várakozási sorok lényeges alkotóelemei a HSDPA rendszernek, ezért természetesnek tűnik a TCP körbefordulási idő – mivel az nagy hatással van a TCP teljesítményére – modellezésére egy megfelelő sorbanállási modell alkalmazása. Ennek megfelelően a rendszer letöltési irányú késleltetését számottevően befolyásoló szűk keresztmetszeteit kell meghatározni (mobilszolgáltatások esetén a felhasználók jellemzően letöltenek, így leggyakrabban letöltés irányban szenvednek el nagyobb késleltetést). A kidolgozott modellben is a letöltési irány teljesítményére koncentráltunk, ahol a feltöltési késleltetést állandónak tekintettük. Csomagvesztés (p) épp ezeknél a várakozási soroknál, telített pufferek esetén fordulhat elő, vagy az újraadások legnagyobb számának elérésekor. Három ilyen pontja van a rendszernek:

- az RLC réteg pufferei, ahol a felhasználói csomagok részekre bontásával nyert RLC csomagokat tárolja a rendszer nyugta érkezéséig, vagy az újraadások legnagyobb számának elérése után eldobja azokat. A pufferbeli csomagok ütemezését a MAC-d réteg vezérli a bázisállomás MAC-hs rétege által biztosított kreditekre támaszkodva. A kreditek úgy vannak meghatározva, hogy a rádiós interfész átvitele a lehető legnagyobb legyen, nem feltétlenül figyelembe véve az Iub interfészen való torlódást, ezért is lehetséges, hogy az RLC túlterhelheti a szállítási réteget. E modellben azt feltételeztük, hogy feltöltés irányban nincs késleltetés,
- az AAL2/ATM szállítási hálózat pufferei. Minthogy a szállítási hálózaton a felhasználók osztoznak és véges kapacitású itt is előfordulhat torlódás, ami a csomagok késleltetéséhez, illetve azok eldobásához vezethet. A modellben ezt egy várakozási sorral vettük figyelembe, tekintettel a szűk keresztmetszetet jelentő ATM összeköttetésre.
- a bázisállomásbeli MAC-hs pufferek. Itt a rendszer szintén felhasználónként pufferele a csomagokat. A 2 ms alatt küldhető csomagok számát

a CQI (channel quality indicator) – a rádiós összeköttetést leíró mennyiség – határozza meg. Amennyiben egy csomag elvesz azt a HARQ mechanizmus működésének megfelelően a rendszer újraküldi, amíg el nem éri az újraküldések legnagyobb számát, amikor is az RLC ARQ veszi át a PDU kezelését.

A sorbanállási hálózati modell a 2. ábrán látható a különböző rétegekben elhelyezkedő pufferekkel. Az RLC felhasználónként egy-egy pufferben tárolja a csomagokat, amiket a bázisállomástól kapott kreditek alapján egymástól függetlenül ütemez. Egy PDU, akkor vész el, ha a puffer túlszordul, vagy ha elérte az újrAADÁSOK legnagyobb számát.



2. ábra. A rendszer sorbanállási hálózatmodellje

Az átviteli hálózatot egy, a szűk keresztmetszetet jelentő összeköttetést reprezentáló pufferrel modelleztük. Telített puffer esetén az ATM cellák elvesznek.

A bázisállomásban is van minden felhasználónak egy-egy puffere, amelyek közül egyet egy PF (proportional fair - arányosan igazságos) ütemező szolgál ki minden 2 ms hosszú időrésben. Dobás esetén a csomagot újra adja a bázisállomás MAC-hs rétege az újrAADÁSOK legnagyobb számának erejéig.

3. A TCP átvitelének számítása

A TCP átvitelét a rendelkezésre álló lehetőségek közül a legnépszerűbb – Jitendra Padhye és társai által [9]-ben kidolgozott – modellel számoljuk. Ez lényegében egy egyszerű kifejezést ad a TCP átvitelére (B) a csomagvesztési

valószínűség (p) és a körbefordulási idő függvényében (RTT)

$$B(p, RTT) = \begin{cases} \frac{\frac{1-p}{p} + E[W] + \hat{Q}(E[W]) \frac{1}{1-p}}{RTT \left(\frac{b}{2} E[W_u] + 1 \right) + \hat{Q}(E[W]) T_0 \frac{f(p)}{1-p}}, & \text{if } E[W_u] < W_{\max} \\ \frac{\frac{1-p}{p} + W_{\max} + \hat{Q}(E[W]) \frac{1}{1-p}}{RTT \left(\frac{b}{8} W_{\max} + \frac{1-p}{p W_{\max}} + 2 \right) + \hat{Q}(W_{\max}) T_0 \frac{f(p)}{1-p}}, & \text{otherwise.} \end{cases} \quad (1)$$

A kifejezésben p a csomagvesztést, b az egyszerre nyugtázott csomagok számát (e cikkben végig $b = 1$ -et feltételezünk), T_0 a TCP időzítését (mi $T_0 = 1.5$ mp-et feltételeztünk), RTT a körbefordulási időt, W_{\max} a legnagyobb torlódási ablakméretet jelöli. $E[W_u]$ a korlátlan ablakméret várható értéke

$$E[W_u] = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2}.$$

$\hat{Q}(w)$ annak a valószínűsége, hogy w ablakméret esetén az időzítő lejárt okozta a veszteséget

$$\hat{Q}(w) = \min \left(1, \frac{(1 - (1-p)^3) (1 + (1-p)^3 (1 - (1-p)^{w-3}))}{1 - (1-p)^w} \right).$$

Végül $f(p)$ egy egyszerűsítés

$$f(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6.$$

Azaz a TCP átvitele két paramétertől függ, a körbefordulási időtől (RTT) és a vesztesési valószínűségtől (p). Az eddigiek alapján jogos a rendszert egy sorbanállási hálózattal modellezni, hiszen az RTT jelentős részét a különböző sorokban való késleltetés teszi ki, illetve csomagvesztés is vagy ezeknek a puffereknek a telítettsége miatt, vagy a rádiós interfész hibái miatt van. A cikkünkben a TCP forgalmat állandó intenzitású folyamként vesszük figyelembe ezzel is egyszerűsítve a modellt.

Azzal, hogy a TCP-t állandó intenzitású folyamnak tekintjük válunk képessé, hogy a Padhye modell paramétereit a következő alszakaszokban leírt sorbanállási hálózat segítségével számoljuk ki. Amint e két paraméter ismert az átvitel számolható, ami azonban nem feltétlenül felel meg a kezdeti feltételként megadott intenzitásnak. Ebben az esetben a kezdeti intenzitást korrigáljuk az eredménynek megfelelően és az átvitelt újra kiszámoljuk, addig amíg az egyensúlyi intenzitáshoz nem jutunk. Ez az a B^* átvitelnek megfelelő intenzitás, amelyre – ha ez a bemeneti intenzitás – pontosan olyan körbefordulási idő és vesztesési valószínűség jön ki, hogy a Padhye modell a $B^* = B(p, RTT)$ átvitelt adja.

3.1. A számítási algoritmus áttekintése

Ahogy azt a 3. szakaszban írtuk a TCP átvitelét HSDPA felett úgy számoljuk, hogy a hálózat terhelése (λ_{TCP}) épp olyan körbefordulási időt (RTT) és csomagvesztési valószínűséget (p) eredményez, amely paraméterekkel a Padhye modell épp ugyanekkora intenzitásnak megfelelő átvitelt ad, azaz $B(p, RTT) = \lambda_{\text{TCP}}$.

Ezt az egyensúlyi értéket például intervallumfelezéssel kaphatjuk. Ezt foglaltuk össze az 1. algoritmusban. Az intervallum alsó határának, természetes kezdeti értéke 0 és mivel az átvitel nem lehet nagyobb, mint a rádiós interfész átlagos átvitele, ezért az intervallum felső határának kezdeti értéke épp ez ($E[S_{\text{Node B}}]$) lesz. Kiszámoljuk a csomagvesztést és az átlagos körbefordulási időt minden lépésben a 2. ábrán látható sorbanállási hálózat segítségével. Beállítjuk az intervallum alsó és felső határát a legutóbbi TCP átvitel (λ_{TCP}) és az épp kiszámolt Padhye-átvitel ($\lambda_{\text{PADHYE}} = B(p, RTT)$) kapcsolatának függvényében.

algoritmus 1 A TCP átvitelének számítása

INPUT: `sysparam`//a rendszerparaméterek az 1. táblázatban találhatóak

OUTPUT: γ //a TCP átvitel

- 1: $a_1 = 0$ //az intenzitásintervallum alsó határa
 - 2: $a_2 = E[S_{\text{Node B}}]$ //a felső határ a rádiós interfész átvitelével egyezik
 - 3: **while** $|\lambda_{\text{TCP}} - \lambda_{\text{old}}| > \epsilon$ **do** //az intervallumfelező ciklus
 - 4: $\lambda_{\text{TCP}} = \frac{a_1 + a_2}{2}$
 - 5: $(p, RTT) = \text{QN Analysis}(\lambda_{\text{TCP}})$
 - 6: $\lambda_{\text{PADHYE}} = B(p, RTT) \cdot K f_T / f_M$ //A Padhye modell alkalmazása (1)
 - 7: **if** $\lambda_{\text{PADHYE}} > \lambda_{\text{TCP}}$ **then**
 - 8: $a_1 = \lambda_{\text{TCP}}$
 - 9: **else**
 - 10: $a_2 = \lambda_{\text{TCP}}$
 - 11: $\lambda_{\text{old}} = \lambda_{\text{TCP}}$
 - 12: **return** $\gamma = \lambda f_M$ //mértékegységváltás
-

A 2. ábrán látható sorbanállási hálózat vizsgálatával kapjuk meg a csomagvesztést, illetve a körbefordulási időt. A felhasználókat azonosnak tekintjük és a számítást egy adott felhasználóra végezzük el. Ennek megfelelően a megjelölt felhasználó szempontjából a sorbanállási hálózat három várakozási sort tartalmaz: az RLC puffert, a többi felhasználóval közös szállítási (ATM) puffert és a bázisállomásban a MAC-d puffert. Ennek a sorbanállási hálózatnak nincs egzakt megoldása ezért a forgalom felbontásán alapuló, közelítő megoldását számoltuk [4]. A vizsgálat során minden sornak megadjuk a min-

	leírás	jelölés	érték
	HSDPA felhasználók száma	K	16
	RLC puffer mérete [PDU]		1000
	az átviteli hálózat pufferének mérete [ATM cella]	L	2000
	Node B puffer mérete [PDU]		1000
	RLC (újra)adások legnagyobb száma	R	6
	HARQ (újra)adások legnagyobb száma	M	3
	közösen nyugtázott TCP csomagok száma	b	1
	TCP időzítő hossza	T_0	1.5 mp
	legnagyobb TCP torlódásvezérlő ablak mérete	W_{\max}	48 KB
	blokkhiba-valószínűség a rádiós interfészen	P_e	0.01
	két egymást követő hibás küldés valószínűsége	P_s	0.001
	a rádiós interfész kiszolgálási eloszlása	$P(\hat{S} = k)$	file-ból
	TCP csomagméret	f_T	1500 byte
	MAC-d és RLC PDU mérete	f_M	336 bit
	pontosság	ϵ, ϵ'	1
	szállítási réteg csatornkapacitása	C	

1. táblázat. sysparam file tartalma

algoritmus 2 $(p, RTT) = \text{QN Analysis}(\lambda_{\text{in}})$

INPUT: λ_{in} //A TCP források jelentette terhelés

OUTPUT: (p, RTT) //csomagvesztés és átlagos körbefordulási idő

- 1: $\lambda' = \frac{\lambda_{\text{in}}}{K}$ //a megfigyelt TCP felhasználó átvitele
 - 2: **while** $|\lambda' - \lambda'_{\text{old}}| > \epsilon'$ **do** // λ' egyensúlyi értékét kereső ciklus
 - 3: $(P_{\text{RLC}}, E[T_{\text{RLC}}], D_{\text{RLC}}) = \text{solve rlc}(\lambda')$ //3.2. alszakasz
 - 4: $(P_{\text{Tr}}, E[T_{\text{Tr}}], D_{\text{Tr}}) = \text{solve tr}(C, D_{\text{RLC}})$ //3.3. alszakasz
 - 5: $(P_{\text{Node B}}, E[T_{\text{Node B}}], \lambda_U) = \text{solve node b}(S, D_{\text{Tr}})$ //3.4. alszakasz
 - 6: $p_L \leftarrow (P_{\text{Tr}}, P_{\text{Node B}}, P_{\text{HARQ}})$ //a visszacsatoló ág valószínűsége (15)
 - 7: $\hat{p} = \frac{\sum_{k=1}^R (1-p_L)^{k-1} p_L}{\sum_{k=1}^{R+1} (1-p_L)^{k-1} p_L}$ //RLC újraküldés valószínűsége
 - 8: $\lambda' = \frac{\lambda_{\text{in}}}{K} + \hat{p} \cdot p_L \cdot \lambda_A$
 - 9: $\lambda'_{\text{old}} = \lambda'$
 - 10: $(D_u, D_s) \leftarrow (P_{\text{Tr}}, P_{\text{Node B}}, P_{\text{HARQ}}, E[T_{\text{RLC}}], E[T_{\text{Tr}}], E[T_{\text{Node B}}])$ // (19)
 - 11: $RTT = \sum_{k=1}^R \frac{p_L^{k-1} (1-p_L)}{1-p_L^k} ((k-1) D_u + D_s)$ //a (20)-ban kapott RTT
 - 12: $p = 1 - \frac{\lambda_U}{\lambda}$ //a (21)-ban megadott TCP veszteségi valószínűség
 - 13: **return** (p, RTT)
-

ket érdeklő teljesítménymutatókon kívül a kimeneti folyamatát is, hiszen ez táplálja a következő sort. Az RLC-vesztést egy visszacsatoló ággal vettük figyelembe, mintha az elveszett, majd újraadott csomagok ismét a sorba érkeznének. Emiatt a sorbanállási hálózatot csak iteratíván lehet megoldani (lásd a 2. algoritmust): kezdetben azt feltételezzük, hogy nincs visszacsatolt forgalom és kiszámoljuk az elvesző csomagok számát, majd a következő lépésben ezt tekintjük a visszacsatoló ág forgalmának, majd ezt addig csináljuk így, amíg az utolsó két érték közti különbség meghalad egy előre meghatározott pontosságot.

3.2. Az RLC puffer

Az RLC réteg modelljének (a 2. algoritmus 3. sorának `solve rlc` függvénye) lényegét az a megfigyelés adja, hogy a távozó forgalmat (egyben a szállítási hálózat érkező forgalmát) a HSDPA áramlásvezérlési algoritmus szabályozza. A rádiós interfész hatékony használata érdekében a bázisállomás úgynevezett krediteket biztosít minden felhasználónak, a kredit értékét a csatornaminőség és az adott felhasználó átlagos átvitelének függvényében adja. A MAC-d ütemező minden kör során a krediteknek megfelelő mennyiségű csomagot visz át. Mi 10 ms-os köridőt feltételeztünk (ez egy szokásos érték), azaz az ütemező csomagokat $TTI_{\text{RLC}} = 10$ ms-onként küldi. A küldhető csomagok számának meghatározásakor azt feltételezzük, hogy a bázisállomás ismeri a rádiós interfész állapotát, azaz a 2 ms-os HSDPA TTI alatt küldhető csomagok számának eloszlása ismert (lásd a 3.4. alszakaszt).

E feltételezéssel a 10 ms alatt a MAC-d ütemező által átvitt csomagok száma

$$S_{\text{RLC}} = \sum_1^5 S_{\text{Node B}}.$$

Az RLC puffer *érkezési folyamata* két részből áll, a rendszerbe belépő (λ_{in}/K intenzitású) forgalom és az RLC által újraadott csomagok (ez a λ_{FB} intenzitású visszacsatoló ág, ahogy a csomagvesztést modellezzük).

$$\lambda' = \frac{\lambda_{\text{in}}}{K} + \lambda_{\text{FB}}.$$

Az RLC pufferbe 10 ms alatt érkező csomagok számának eloszlása így

$$P(A_{\text{RLC}} = k) = \frac{(\lambda' TTI_{\text{RLC}})^k}{k!} e^{-\lambda' TTI_{\text{RLC}}} \quad k = 0, 1, 2, \dots \quad (2)$$

Gyakorlatban az eloszlást úgy csonkoltuk (N -nél), hogy az eldobott farokrész valószínűsége már elhanyagolhatóan kicsi.

A *sorhossz* alakulását minden TTI_{RLC} végén egy diszkrét idejű Markov láncsal (DTMC) modelleztük, amelynek sorhossza a következőképpen alakul

$$X_{n+1} = (X_n + A_{n+1} - S_{n+1})^+,$$

ahol X_{n+1} a sorhossz, A_{n+1} az érkező csomagok száma és S_{n+1} a kiszolgált csomagok száma az $n+1$ -s időrésben. $(\cdot)^+$ $\max(0, \cdot)$ -t jelöli.

A DTMC egylépéses állapotátmeneti mátrixának (\mathbf{P}) ij -dik eleme

$$p_{ij} = \begin{cases} \sum_{k=0}^{\infty} P(A_{\text{RLC}} = k) P(S_{\text{RLC}} = j - i + k) & i < L - N \\ \sum_{k=0}^{L-i} P(A_{\text{RLC}} = k) P(S_{\text{RLC}} = i - j + k) + \\ + P(S_{\text{RLC}} = L + 1 - j) \sum_{k=L-i+1}^N P(A_{\text{RLC}} = k) & i \geq L - N. \end{cases}$$

Az RLC puffer mérete L , az érkezési eloszlás tartója a $[0, N]$ intervallum. Az első esetben a kiindulási sorhossz olyan rövid, hogy az érkező csomagok nem veszhetnek el, azaz az állapotváltási valószínűség megegyezik annak a valószínűségével, hogy $j - i$ -vel több csomagot szolgáltat ki a rendszer, mint amennyi érkezett. A második esetben a kifejezésnek két tagja van, az első tag esetében nincs, a másodikéban van dobás.

A DTMC határeloszlását a következő lineáris egyenletrendszer megoldása adja

$$\begin{aligned} \boldsymbol{\pi} \mathbf{P} &= \boldsymbol{\pi} \\ \boldsymbol{\pi} \mathbf{h} &= 1, \end{aligned}$$

ahol \mathbf{h} a megfelelő méretű, csupa egyesekből álló oszlopvektor.

A határeloszlás ismeretében a csomagvesztési valószínűséget a $TTI_{\text{RLC}} = 10$ ms alatt elvesző és az ugyanezen idő alatt érkező csomagok átlagos számának hányadosaként számoljuk

$$P_{\text{RLC}} = \frac{\sum_{i=0}^L \pi_i \sum_{j=0}^N \max(0, i + j - L) P(A_{\text{RLC}} = j)}{\sum_{i=0}^L \pi_i \sum_{j=0}^N j P(A_{\text{RLC}} = j)}. \quad (3)$$

A csomagok rendszeridejét az RLC rétegben Little tételével számoljuk

$$E[T_{\text{RLC}}] = \frac{E[X_{\text{RLC}}]}{(1 - P_{\text{RLC}}) E[A_{\text{RLC}}]} TTI_{\text{RLC}} + \frac{1}{2} TTI_{\text{RLC}}, \quad (4)$$

ahol az átlagos sorhosszt $E[X_{\text{RLC}}]$ -vel jelöltük. Mivel a modell diszkrét idejű és a csomagok folyamatosan érkeznek, a modell nem tesz különbséget az időrés elején és végén érkező csomag között. Ezt a TTI alatt egyenletesen elosztott érkezési pillanatokkal vettük figyelembe, vagyis a DTMC-ből kiszámolt rendszeridőhöz hozzáadunk fél TTI-t – az érkezési pillanat várható értékét.

Az RLC puffer *távozási folyamatát* szintén megadjuk, mivel a sorbanállási hálózatban ez a szállítási puffer érkezési folyamata. Azt feltételezzük, hogy a távozások független azonos eloszlásúak, ahol a TTI_{RLC} alatt távozó csomagok számának eloszlása

$$\begin{aligned}
 P(D_{\text{RLC}} = k) &= \sum_{i=0}^L \pi_i \sum_{j=k+1-i}^{\infty} P(A_{\text{RLC}} = j) P(S_{\text{RLC}} = k) + \\
 &+ \sum_{i=0}^L \pi_i P(A_{\text{RLC}} = k-i) \sum_{j=k}^{\infty} P(S_{\text{RLC}} = j).
 \end{aligned} \tag{5}$$

A kifejezést két tag összege alkotja. Az első megfelel annak az esetnek, amikor van annyi csomag a pufferben, ahányat a kiszolgáló kiszolgálna, míg a második tag annak felel meg, amikor a kiszolgáló több csomagot szolgáltatna ki, mint ami a pufferben rendelkezésére áll.

3.3. A szállítási puffer

E cikkben AAL2/ATM szállítási réteget feltételezünk (ennek a modellje, illetve megoldása jelenik meg a 2. algoritmus 4. sorában). Az AAL2 réteg multiplexálja az egyes felhasználók forgalmát egy C kapacitású, állandó sebességű (Constant Bit Rate - CBR) VCC-be.

A MAC-d és a MAC-hs ütemezőkkal ellentétben az ATM kapcsoló folytonos időben működik, ennek ellenére úgy döntöttünk, hogy diszkrét idejű modellt dolgozunk ki, hogy elkerüljük a folytonos és a diszkrét idejű modellek keverését. Az RLC puffer $TTI_{\text{RLC}} = 10$ ms-onként küld, míg a bázis-állomásbeli PF ütemező $TTI_{\text{Node B}} = 2$ ms-onként. Ez utóbbi kisebb értékűt választottuk időegységül az ATM diszkrét idejű modelljében, mert így valamivel finomabb felbontását nyerjük a folytonos időnek. További egyszerűsítő feltételezés, hogy a szállítási puffer RLC csomagokat továbbít nem pedig ATM cellákat. Lévén, hogy az RLC PDU az adategység a hálózat többi részén ezzel is jelentősen egyszerűsödik a modell megoldása.

Az időreseként *érkező* csomagok számának eloszlását az RLC távozási folyamatából (D_{RLC}) vezetjük le. Ez azonban 10 ms-onként adott, amíg az előzőeknek megfelelően a szállítási puffer időegysége 2 ms. Azaz első lépésként

végre kell hajtanunk az átalakítást a két eloszlás között, aholis a TTI_{RLC} ötször nagyobb TTI_{Tr} -nél. Binomiális feltételezéssel élve

$$P(D^{2\text{ms}_{\text{tr}}} = k) = \sum_{i=k}^{\infty} P(D_{\text{RLC}} = i) \binom{i}{k} \left(\frac{1}{5}\right)^k \left(1 - \frac{1}{5}\right)^{i-k},$$

ahol $P(D^{2\text{ms}_{\text{tr}}} = k)$ annak a valószínűsége, hogy TTI_{Tr} idő alatt k csomag érkezett, ha TTI_{RLC} alatt i , más szóval hogyan tudunk kiválasztani k -t i -ből $\frac{1}{5}$ valószínűséggel – tudniillik ez a két TTI aránya.

A szállítási puffer érkezési eloszlásának számításakor összegeznünk kell az összes felhasználó forgalmát, hiszen itt a teljes forgalmat egy VCC-be multiplexálja az ATM

$$A_{\text{Tr}} = \sum_1^K D^{2\text{ms}_{\text{tr}}}.$$

Itt K a HSDPA felhasználók száma.

Az RLC csomagok *kiszolgálási idejét* a szállítási pufferben így számoljuk

$$D = \frac{\text{RLC csomagméret fejlécekkel}}{C}.$$

A fejléceket a következőképpen vesszük figyelembe

$$\text{RLC csomagméret fejlécekkel} = f_M \cdot \underbrace{\left(\frac{53}{47} \frac{f_M + 24}{f_M} \frac{E[D_{\text{RLC}}]}{E[D_{\text{RLC}}]} \frac{f_M + 72}{f_M} \right)}_{\text{fejlécek}}.$$

A fejlécek az ATM fejlécből (40 bit) plusz a 8 bites CPS PDU kezdeti mezőből (Start Field - 53/47), a 24 bites CPS csomag fejlécből csomagonként $\left(\frac{f_M+24}{f_M}\right)$ és végül a 72 bites HS-DSCH FP keret fejlécből áll, ami $E[D_{\text{RLC}}]$ RLC csomagot szállít átlagosan.

Ebben a TTI_{Tr} időegységű diszkrét modellben a kiszolgáló vagy $F = \lfloor \frac{TTI_{\text{Tr}}}{D} \rfloor$ vagy $F + 1$ csomagot szolgál ki

$$P(S_{\text{Tr}} = F) = 1 - \left(\frac{TTI_{\text{Tr}}}{D} - F \right)$$

$$P(S_{\text{Tr}} = F + 1) = \frac{TTI_{\text{Tr}}}{D} - F$$

valószínűséggel.

A *sorhosszt* az RLC pufferéhez hasonló DTMC modellezi, azaz

$$X_{n+1} = (X_n + A_{n+1} - S_{n+1})^+, \quad (6)$$

ahol X_{n+1} a sorhossz, A_{n+1} az érkező és S_{n+1} a kiszolgált csomagok száma az $n+1$ -ső időrásben.

Az érkezési és a kiszolgálási eloszlás ismeretében az RLC-hez hasonlóan építhetjük fel a DTMC egylépéses állapotátmeneti mátrixát

$$p_{ij} = \begin{cases} \sum_{k=0}^{\infty} P(A_{\text{Tr}} = k) P(S_{\text{Tr}} = j - i + k) & i < L - (N - F) \\ \sum_{k=0}^{L-i} P(A_{\text{Tr}} = k) P(S_{\text{Tr}} = i - j + k) + \\ + P(S_{\text{Tr}} = L + 1 - j) \sum_{k=L-i+1}^N P(A_{\text{Tr}} = k) & i \geq L - (N - F). \end{cases}$$

A csomagvesztési valószínűséget is az RLC-éhez hasonlóan fejezhetjük ki

$$P_{\text{Tr}} = \frac{\sum_{i=0}^{L-F} \pi_i \sum_{j=0}^N \max(0, i + j - L) P(A_{\text{Tr}} = j)}{\sum_{i=0}^{L-F} \pi_i \sum_{j=0}^N j P(A_{\text{Tr}} = j)}. \quad (7)$$

A számláló az elvesző, a nevező pedig az érkező csomagok várható száma.

A szállítási puffer rendszeridejét a Little formula segítségével számolhatjuk (ugyanúgy, mint (4)-ben)

$$E[T_{\text{Tr}}] = \frac{E[X_{\text{Tr}}]}{(1 - P_{\text{Tr}}) E[A_{\text{Tr}}]} TTI_{\text{Tr}} + \frac{1}{2} TTI_{\text{Tr}}. \quad (8)$$

A távozási folyamat eloszlása szintén az RLC azonos paraméteréhez hasonlóan számolandó

$$P(D_{\text{Tr}} = k) = \sum_{i=0}^{L-F} \pi_i \sum_{j=k+1-i}^{\infty} P(A_{\text{Tr}} = j) P(S_{\text{Tr}} = k) + \\ + \sum_{i=0}^{L-F} \pi_i P(A_{\text{Tr}} = k - i) \sum_{j=k}^{\infty} P(S_{\text{Tr}} = j). \quad (9)$$

3.4. A MAC-hs puffer

E cikkben azt feltételeztük, hogy a MAC-hs pufferek tartalmát arányosan igazságos (proportional fair - PF) algoritmus alapján ütemezi az ütemező, amely a pillanatnyi csatornaminőség és a felhasználók átlagos átvitelének alapján, a lehető leghatékonyabb erőforráskihasználást szem előtt tartva

nyújt kiszolgálást a felhasználóknak. Az ütemező minden körben kiválaszt egy felhasználót, aki adhat (minden $TTI_{\text{Node B}} = 2 \text{ ms}$ -ban). A bázisállomás által meghatározott csatornaminőség-mutató (Channel Quality Indicator - CQI) meghatározza a kódolási sémát és ezzel együtt az egy TTI alatt küldhető csomagok számát. Mivel a csatornaminőség gyorsan változhat időlegesen előfordulhat puffer túlterhelés is a bázisállomásban. Az érkező csomagokat a felhasználónkénti MAC-hs pufferek tárolják.

Nem tartozik e cikk céljai közé a rádiós interfész modellezése, ezért az Eurane projectből (lásd [5]) vett MATLAB programmal állítottuk elő az egy TTI alatt átvihető csomagok számának eloszlását ($P(\hat{S} = k)$). Az eloszlás készítésekor telített puffereket feltételeztünk és nem vettük figyelembe a HARQ mechanizmust [7].

A MAC-hs puffer *kiszolgálási folyamat*ához először is a HARQ-ot vettük figyelembe. [1]-ben és [2]-ben a szerzők megadják annak az eloszlását, hogy j -dikre sikeres az átvitel

$$P_j = \begin{cases} 1 - P_e, & j = 1 \\ P_e^{j-1} P_s^{j-2} (1 - P_e P_s), & j > 1. \end{cases}$$

A két paraméter (P_e és P_s) jelentését az 1. táblázatban foglaltuk össze. Figyelembe véve, hogy az újraadások legnagyobb száma M , az (újra)adások várható száma

$$E[H] = \sum_{j=1}^M j P_j + M \left(1 - \sum_{j=1}^M P_j \right),$$

és annak a valószínűsége, hogy egy időrés elvész HARQ veszteség miatt

$$P_{\text{tl}} = 1 - \frac{1}{E[H]}.$$

Végül az egy TTI alatt átvihető csomagok számának az eloszlása (figyelembe véve a HARQ veszteségeket is)

$$P(S_{\text{Node B}} = k) = \begin{cases} (1 - P_{\text{tl}}) P(\hat{S} = k) + P_{\text{tl}} & k = 0, \\ (1 - P_{\text{tl}}) P(\hat{S} = k) & k \neq 0. \end{cases} \quad (10)$$

A MAC-hs pufferbe *érkező* csomagszám *eloszlásának* meghatározásakor azt feltételeztük, hogy a szállítási hálózatból érkező csomagok közül $1/K$ paraméterű binomiális eloszlás szerint tartoznak k -an a megfigyelt felhasználóhoz

$$P(A_{\text{Node B}} = k) = \sum_{i=k}^{\infty} P(D_{\text{Tr}} = i) \binom{i}{k} \left(\frac{1}{K}\right)^k \left(1 - \frac{1}{K}\right)^{i-k}.$$

Ellentétben a másik két csomóponttal a MAC-hs puffer *sorhosszának* alakulása

$$X_{n+1} = (X_n - S_{n+1})^+ + A_{n+1}.$$

Ez azt jelenti, hogy csak azokat a MAC-d csomagokat szolgálja ki a PF ütemező, amelyek a TTI kezdete előtt érkeztek, azaz az állapotátmeneti mátrix ij -dik eleme

$$p_{ij} = \begin{cases} \sum_{k=0}^{j-1} P(A_{\text{Node B}} = k) P(S_{\text{Node B}} = i - j + k) + & i < k_m \\ + P(A_{\text{Node B}} = j) \sum_{k=i}^{\infty} P(S_{\text{Node B}} = k) & \\ \sum_{k=0}^{\infty} P(A_{\text{Node B}} = k) P(S_{\text{Node B}} = j - i + k) & i \geq k_m. \end{cases}$$

A határeloszlás meghatározása után a csomagvesztési valószínűséget az elvesző és az összes érkező PDU számának hányadosaként kapjuk

$$P_{\text{Node B}} = \frac{\sum_{i=0}^L \pi_i \sum_{j=0}^{\infty} \max(0, i + j - L) P(A_{\text{Node B}} = j)}{\sum_{i=0}^L \pi_i \sum_{j=0}^{\infty} j P(A_{\text{Node B}} = j)}. \quad (11)$$

$P_{\text{Node B}}$ a puffer telítettsége miatt bekövetkezett dobási valószínűséget jelöli. Ugyanakkor nem ez az egyetlen módja a csomagvesztésnek a bázisállomásban. Ha a rádiós interfész rossz minőségű és a HARQ sem tudja már újraadni, a MAC-hs figyelmen kívül hagyja a csomagot és ha az újraadások száma eléri a legnagyobb megengedett értéket (M), akkor ismét az RLC réteg felelőssége lesz az újraadás. Ennek a valószínűsége

$$P_{\text{HARQ}} = 1 - \sum_{j=1}^M P_j. \quad (12)$$

A MAC-hs puffer rendszerideje a Little formula segítségével

$$E[T_{\text{Node B}}] = \frac{E[X_{\text{Node B}}]}{(1 - P_{\text{Node B}}) E[A_{\text{Node B}}]} TTI_{\text{Node B}} + \frac{1}{2} TTI_{\text{Node B}}, \quad (13)$$

ahol $E[X]$ az átlagos sorhosszt jelöli és a hozzáadott fél TTI magyarázata ugyanaz, mint az RLC és a szállítási puffer modelljeinek esetében.

A sorbanállási hálózat vizsgálatához szükség van még a bázisállomás pufferéből távozó csomagok intenzitására. A $TTI_{\text{Node B}}$ alatt távozó csomagok száma kiszolgálható, illetve a pufferben lévő csomagok számának minimumával egyenlő. Így a távozó csomagok intenzitása

$$\lambda_U = \frac{1}{TTI_{\text{Node B}}} \sum_{i=0}^L \pi_i \sum_{k=0}^{\infty} P(S_{\text{Node B}} = k) \min(i, k). \quad (14)$$

3.5. A visszacsatoló ág

Azt feltételeztük a sorbanállási hálózatmodellünkben, hogy a hálózat különböző pontjain elveszett csomagok az RLC pufferbe újraadásra ismét belépnek. A 2. ábra visszacsatoló ága ezeket a csomagokat „gyűjti” össze. Ebben az alszakaszban ennek az összeköttetésnek a forgalmát fogjuk kiszámolni. Ezt a (Poissonnak feltételezett [4]) forgalmat adjuk hozzá az RLC puffer bemeneti forgalmához a hálózat vizsgálata során.

Legelőször is kiszámoljuk annak a valószínűségét, hogy a PDU az RLC puffer elhagyása után (bármilyen okból) elveszett. Ezt jelölje p_L .

$$p_L = P_{\text{Tr}} + (1 - P_{\text{Tr}}) P_{\text{Node B}} + (1 - P_{\text{Tr}}) (1 - P_{\text{Node B}}) P_{\text{HARQ}} \quad (15)$$

Az is megtörténhet, hogy egy újraadott PDU elvész. Egy adott számú újraadás után – ez az RLC újraadások legnagyobb száma (R) – az RLC réteg figyelmen kívül hagyja az adott csomagot, ami TCP-szinten veszteséget eredményez. Ezesetben a PDU nem lép be újra az RLC pufferbe (míg nem egy magasabb rétegbeli protokoll azt újra nem adja). Annak a valószínűsége, hogy egy elveszett PDU még nem érte el az újraadások legnagyobb számát, azaz növeli az RLC puffer terhelését:

$$\hat{p} = \frac{\sum_{k=1}^R (1 - p_L)^{k-1} p_L}{\sum_{k=1}^{R+1} (1 - p_L)^{k-1} p_L}, \quad (16)$$

ahol az újraadások számát csonkolt geometriai eloszlásúnak feltételeztük.

A fentieket figyelembe véve a visszacsatoló ág forgalma:

$$\lambda_{\text{FB}} = \hat{p} \cdot p_L \cdot \lambda_A, \quad (17)$$

ahol λ_A az RLC pufferből való átlagos távozási intenzitást jelöli.

3.6. A TCP vesztes és a körbefordulási idő

Ebben az alszakaszban a pufferenkénti teljesítményjellemzők (részletekért lásd (12), (3), (4), (7), (8), (11) és (13) egyenleteket) alapján kiszámoljuk a TCP teljesítményét.

Az a TCP csomag, amely nem vész el

$$D_s = E [T_{\text{RLC}}] + E [T_{\text{Tr}}] + E [T_{\text{Node B}}] \quad (18)$$

késleltetést szenved el. Ha azonban valahol elveszett, akkor az átlagos csomagkésleltetést

$$\begin{aligned} D_u = P_{\text{Tr}} E [T_{\text{RLC}}] + (1 - P_{\text{Tr}}) P_{\text{Node B}} (E [T_{\text{RLC}}] + E [T_{\text{Tr}}]) + \\ + (1 - P_{\text{Tr}}) (1 - P_{\text{Node B}}) P_{\text{HARQ}} \cdot \\ \cdot (E [T_{\text{RLC}}] + E [T_{\text{Tr}}] + E [T_{\text{Node B}}]) \end{aligned} \quad (19)$$

adja. Egy k -szor (újra)adott csomag átlagos körbefordulási ideje a $k - 1$ sikertelen és a sikeres küldés késleltetésének az összege. Geometriai eloszlású (újra)adásszámot feltételezve

$$RTT = D_{\text{UL}} + \sum_{k=1}^R \frac{p_L^{k-1} (1 - p_L)}{1 - p_L^R} ((k - 1) D_u + D_s), \quad (20)$$

ahol D_{UL} az állandónak feltételezett feltöltési irányú késleltetést jelöli, ahogy valóban UTRAN-ban jellemzően nincs torlódás ebben az irányban.

A TCP veszési valószínűsége egyszerűen egy mínusz a sikeresen átvitt csomagok hányada

$$p = 1 - \frac{\lambda_U}{\lambda}. \quad (21)$$

4. Összefoglalás

Ebben a cikkben egy közelítő modelljét adtuk a TCP-nek HSDPA felett. Azonosítottuk a rendszer lényeges torlódási pontjait, amelyek számottevően befolyásolják a TCP átvitelét és megadtuk ezek Markov-i modelljeit, hogy kiszámoljuk a rendszer teljesítményjellemzőit. A rendszer sorbanállási hálózatmodelljének egy iteratív megoldási módját adtuk.

Hivatkozások

- [1] Mohamad Assaad, Badi Jouaber, and Djamel Zeghlache. Effect of TCP on UMTS-HSDPA System Performance and Capacity. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, volume 6, pages 4104–4108, Dallas, TX, USA, November 2004.

- [2] Mohamad Assaad and Djamal Zeghlache. Cross-layer Design in HSDPA System to Reduce TCP Effect. *IEEE Journal on Selected Areas in Communications*, 24(3):614–625, March 2006.
- [3] Mohamad Assaad and Djamal Zeghlache. *TCP Performance Over UMTS-HSDPA Systems*. Auerbach Publications, Boston, MA, USA, 2006.
- [4] Gunter Bolch, Hermann de Meer, Stefan Greiner, and Kishor S. Trivedi. *Queueing Networks and Markov Chains : Modeling and Performance Evaluation With Computer Science Applications*. Wiley-Interscience, August 1998.
- [5] Eurane. The eurane project, 2004. <http://www.ti-wmc.nl/eurane/>.
- [6] H. Holma and A. Toscali. *HSDPA/HSUPA for UMTS*. John Wiley & Sons, 2006.
- [7] G. Horváth and Cs. Vulkán. Throughput Analysis of the Proportional Fair Scheduler in HSDPA. In Jan Sykora, editor, *Proceedings European Wireless 2008 (EW2008)*, 2008.
- [8] P.J. Legg. Optimised Iub Flow Control for UMTS HSDPA. *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 4:2389–2393 Vol. 4, 30 May-1 June 2005.
- [9] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling TCP Throughput: a Simple Model and its Empirical Validation. In *SIGCOMM '98: Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 303–314, New York, NY, USA, September 1998. ACM Press.