

Elosztott hívásátadási eljárás vizsgálata újgenerációs mobil hálózatokban

TDK dolgozat

Elosztott hívásátadási eljárás vizsgálata újgenerációs mobil hálózatokban

TDK dolgozat

Reider Norbert
Temesváry András

Budapest, 2006. október 26.

Tartalomjegyzék

Tartalomjegyzék.....	3
Bevezetés.....	4
1. Hívásátadási eljárások bemutatása újgenerációs mobil hálózatokban.....	5
2. Hívásátadási folyamat részletes ismertetése a vizsgált hálózatban	7
2.1. A vizsgált hálózat felépítése és működése.....	7
2.2. Az elosztott hívásátadási eljárás bemutatása.....	10
3. A szimulátor bemutatása	12
3.1. A szimulátorban megvalósított hálózat felépítése	12
3.2. A szimulátorban szereplő modulok és tulajdonságaik.....	13
3.3. Elosztott hívásátadást megvalósító modul egységei és azok tulajdonságaik.....	15
4. A hívásátadási eljárás működése során vizsgált tervezési döntések értékelése szimuláció segítségével	17
4.1. A csomagok kezelésének kérdése a forrás bázisállomásnál a hívásátadási folyamat során.....	17
Vizsgált tesztesetek.....	17
A vizsgált tesztesetek szimulációs vizsgálata	18
Következtetés.....	22
4.2. TCP átvitel vizsgálata a szinkronizációs idő függvényében.....	23
Szinkronizációs idő hatása a TCP átvitelre csomagtovábbítással	23
Szinkronizációs idő hatása a TCP átvitelre csomagtovábbítás nélkül.....	25
4.3. A csomagok felcserélődésének problémája a cél bázisállomásnál	28
Csomagok keveredése a hívásátadási folyamat közben	31
Csomagok keveredése a hívásátadás befejezése után	33
A csomagok sorrendjének felcserélődésére egy lehetséges megoldás ismertetése	34
Szimulációs eredmények értékelése.....	34
4.4. A MAC réteg és a mobilitás kezelés közti együttműködés kérdései	39
A MAC működés részletes ismertetése.....	40
Hívásátadás hatása a MAC réteg adattovábbítási folyamatára.....	41
4.5. Transzport hálózati link túlcsoordulása.....	45
Transzport hálózati puffer mérete korlátozás nélküli esetben.....	47
Transzport hálózati puffer mérete korlátozás esetén.....	50
5. Összefoglalás	52
Irodalomjegyzék	53

Bevezetés

A jövő mobil hálózataival szemben egyre nagyobb elvárások mutatkoznak, mind a felhasználók szemszögéből, akik egyre gyorsabb és kényelmesebb szolgáltatásokat szeretnének igénybe venni, mind az operátorok részéről, akik a hálózat lehető leggazdaságosabb kihasználására törekednek. A mobil hálózatok jellegéből adódóan a hívásátadási eljárás (handover) kulcsfontosságú ebben a környezetben, és megvalósításának nagy hatása van a felhasználó által kapott szolgáltatás minőségére.

A dolgozatban először röviden áttekintjük, hogy milyen megoldásokat ismerünk a hívásátadási eljárás végrehajtására a mobil IP és a cellás mobil rendszerekben. Ezt követően az általunk vizsgált mobil hálózatban működő hívásátadási eljárás folyamatát és működését tekintjük át. Elemezzük a folyamat alatt felmerülő különböző kérdéseket, mint amilyen például a csomagvesztés elkerülésének lehetősége, vagy a hívásátadási idő hatása a TCP átvitelére, illetve vizsgálva a különböző tervezési szempontokat, mint például az adatkapcsolati és hálózati réteg közötti együttműködést. Az általunk vizsgált elosztott hívásátadási eljárás lényege, hogy a központi vezérlő egység helyett a bázisállomás irányítja a folyamatot. Egy ilyen általános eljárás használható lehet a jövőbeli cellás mobil hálózatokban.

A tervezési folyamat során felmerülő különböző alternatív megoldások részletes kiértékelésére szimulációs vizsgálatokat végeztünk, melyek segítenek a megfelelő tervezési döntések kiválasztásában. Ehhez röviden bemutatjuk a szimulátort és az általunk elkészített hívásátadási eljárást vezérlő modult, amelyet a vizsgálatokhoz használtunk.

Dolgozatunk kiemelt célja tehát, hogy az elosztott hívásátadási eljárást részletesen ismertessük, és a felmerülő problémákra szimulációs vizsgálatok segítségével egy-egy lehetséges megoldást adjunk.

1. Hívásátadási eljárások bemutatása a jövő mobil hálózataiban

Ebben a fejezetben különböző újgenerációs mobil hálózatokban alkalmazott hívásátadási eljárásokat mutatunk be, összehasonlítva azokat az általunk vizsgált hálózatban alkalmazott eljárással. A vizsgált hálózat a 3G WCDMA rendszer továbbfejlesztése, melynek részletes ismertetése a 2. fejezetben található. A vizsgált hálózat megfelel a 3GPP (3G Partnership Project) szabványosítási fórum által jelenleg is fejlesztett 3G LTE (Long-Term Evolution) hálózatnak [1]. Ez a fejezet azzal foglalkozik, hogy összehasonlítsa a *mobil IP* világban kialakult hívásátadási eljárásokat az *általunk vizsgált hálózatban* alkalmazott megoldással, kiemelve a köztük lévő legfontosabb különbségeket.

A mobilitás kezelés során azt a problémát kell megoldani, hogy a végpontok számára a hálózat transzparensen tudja követni a felhasználó földrajzi mozgását, tehát a mozgás miatt bekövetkezett hálózatbeli változások (mint például hozzáférési pont váltása) észrevétlenek legyenek a hálózatot használók számára.

Az IP szintű mobilitás kezelésnél két esetet különböztetnek meg, *mikro* és *makro* mobilitást. *Mikro mobilitás* esetén a mobil terminálnak egy előre definiált korlátozott földrajzi területen belül biztosítunk mobilitást. A *makro mobilitás* pedig ezen földrajzi területek között biztosítja az átjárást. A mikro mobilitási területen belüli cellaváltást intra-domain hívásátadásnak nevezzük, ezek kezelését végzik a mikro mobilitási protokollok. A makro mobilitási protokollok esetén a területek közötti váltásokat inter-domain hívásátadásnak nevezik. Például a *Mobil IP* ezen protokollok közé tartozik, mivel támogatja a *globális mobilitást*, és a felhasználószám növekedésével jól *skálázható* [3].

A felhasználók igényeinek kielégítése érdekében a mobil alkalmazások egyre gyorsabban fejlődnek, ami miatt növekszik a felhasználók által igényelt sávszélesség is. A cellás mobil hálózatokban sávszélesség növekedést többek között a rádiós cellák méretének csökkentésével lehet elérni, ami miatt megnő a cellaváltások száma. Ez azt jelenti, hogy a valóságban a mikro mobilitási hívásátadások gyakrabban fordulnak elő, mint a makro mobilitási területváltások. Mivel a gyakori cellaváltásokat a Mobil IP csak viszonylag nagy késleltetéssel tudja kezelni, ezért van szükség többek között a mobilitás hierarchikus felosztására.

A *mikro mobilitási* protokollok célja, hogy minél *gyorsabban* végrehajtsák az intra-domain cellaváltásokat, növelve ezzel a hálózat *kihasználtságát* és az *elérhető teljesítményt*. Ezek a protokollok képesek a Mobil IP-vel együttműködve egy jól definiált területre korlátozott mobilitási lehetőséget biztosítani. A mikro mobilitás protokollok a felhasználók számával *nem skálázhatóak* olyan jól, mint a Mobil IP, viszont lényegesen *jobb teljesítményt* nyújtanak a cellaváltások esetében [4]. A mikro mobilitás protokollok jelentősége egyre nagyobb az „All IP” megközelítés előretörésével a jövő mobil rendszereiben, amelynek jelentése, hogy ahol csak lehet, használjunk IP hálózatokat, azaz

az adat, jelzés, stb. mind IP csomagokban továbbítódnak. Ilyen mikro mobilitási protokoll például a Cellás IP, HAWAII és a Hierarchikus Mobil IP [2].

Hívásátadási eljárás szempontjából a mobil IP világban, és a cellás mobil rendszerekben (mint például a 3G WCDMA [6]) alkalmazott megoldások jó néhány ponton jelentősen különböznek egymástól. Az utóbbiakra jellemző az úgynevezett *előkészített* hívásátadás (prepared handover), ami azt jelenti, hogy a mobil terminál cellaváltását a hálózat részéről erőforrás foglalás előzi meg, ezzel biztosítva a cellaváltás sikerességét. Valamint ezekben a rendszerekben a mobilitást *az adatkapcsolati és hálózati réteg együttműködve* hajtja végre [10]. Ez a kooperáció főként abban mutatkozik meg, hogy ezekben a rendszerekben a protokoll rétegek integráltak, így optimalizálni lehet a különböző protokoll rétegek működését cellaváltáskor. Például a fizikai, illetve az adatkapcsolati réteg jelzéseket szolgáltatathat a hálózati rétegek számára hívásátadás kezdetekor, illetve végekor. Ilyen rétegek közötti együttműködés lehet például az a funkció, amikor hívásátadás során a hálózati réteg képes visszakérni az adatkapcsolati réteg pufferében lévő csomagokat, hogy továbbítsa azokat az új bázisállomáshoz.

Ellentétben az IP mobilitás alapú rendszerekkel, ahol *nincs együttműködés* a két réteg között, a hívásátadást elkülönítlen, egymástól függetlenül hajtják végre. Ezen kívül abban is különböznek, hogy mely réteg kezeli a mobilitást. A *cellás mobil rendszerekben* ezt a funkciót az *adatkapcsolati* réteg látja el. Például a WCDMA hálózatokban úgynevezett soft handover alkalmaznak, ami azt jelenti, hogy a mobil terminál egyszerre több bázisállomással is képes kommunikálni [5]. Az *IP mobilitás alapú rendszerekben* pedig a *hálózati* réteg látja el a mobilitás kezeléssel kapcsolatos feladatokat. Az IP mobilitást megvalósító protokollok általában nem garantálják a veszteségmentes csomagküldést, bár a Cellás IP és a HAWAII protokoll is törekszik erre. A Cellás IP esetén a mobil előre bejelentkezhet egy másik hozzáférési ponthoz, majd a mobil terminálnak küldött csomagokat duplikálják, hogy a régi és az új hozzáférési pont is megkapja. A HAWAII protokoll pedig a csomagok továbbítását támogatja a régi hozzáférési ponttól az újig [2]. A következő táblázat összehasonlítja az IP mobilitás alapú rendszerekben és a cellás mobil rendszerekben alkalmazott hívásátadási eljárásokat.

	IP mobilitás alapú rendszerek	Cellás mobil rendszerek
Előkészített	Egyes protokollok támogatják (Cellás IP semi-soft handovere)	Igen
Veszteségmentes	Nem, de törekvés a veszteség csökkentésére (Cellás IP semi-soft handovere és HAWAII)	Igen
Hálózat vezérelt	Nem	Igen
Adatkapcsolati és hálózati réteg közötti együttműködés	Nincs együttműködés	Integrált protokoll rétegek

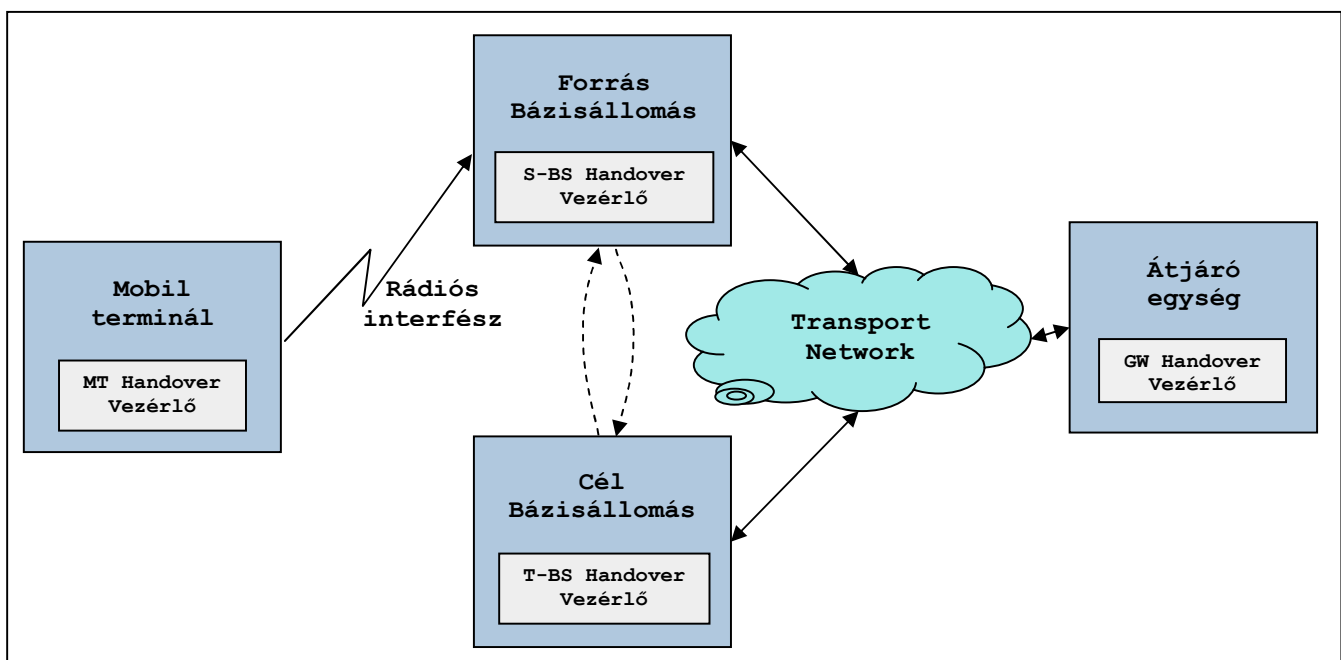
1.1. táblázat: Különböző újgenerációs mobil hálózatokban alkalmazott hívásátadási eljárások összehasonlítása

2. Hívásátadási folyamat részletes ismertetése a vizsgált hálózatban

Az *elosztott hívásátadási eljárás* részletes ismertetése előtt bemutatjuk az általunk vizsgált mobil hálózat felépítését és működését.

2.1. A vizsgált hálózat felépítése és működése

A vizsgált mobil hálózat egyszerűsített felépítését szemléltetjük a 2.1.1. ábrán, amelyen csak a hívásátadáshoz szükséges főbb elemeket vázoltuk fel.



2.1.1. ábra: A vizsgált mobil hálózat egyszerűsített felépítése

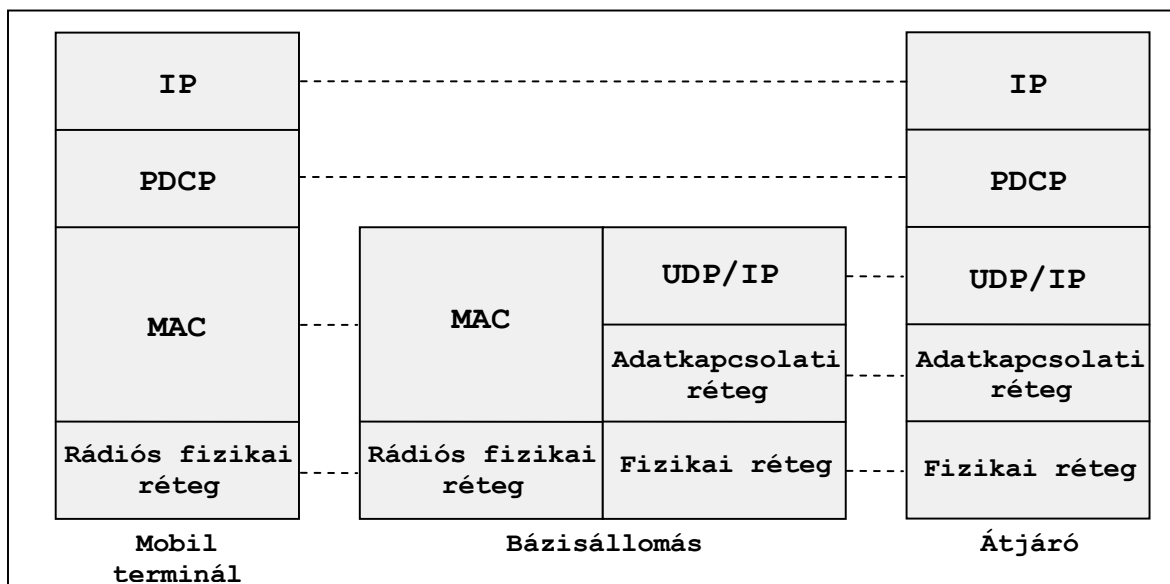
A 2.1.1. ábra alapján látható, hogy a mobil terminál (Mobile Terminal, MT) a rádiós interfészen keresztül kommunikál a forrás bázisállomással (Source Base Station, S-BS). A mobil készülék folyamatosan, meghatározott időközönként, vagy esemény triggerelt alapon *mérési riportokat küld* a bázisállomásnak, amelynél éppen tartózkodik. Ezek a mérési riportok tartalmazzák a készülék által mért rádiós jelerősséget. Ha a *forrás bázisállomás* a riportok alapján úgy dönt, hogy a rádiós link minősége már nem megfelelő, *hívásátadást kezdeményez*, és előkészíti a mobil terminál sikeres átjelentkezését más bázisállomáshoz. Minden hívásátadáskor egy új MAC (Medium Access Control) kapcsolat keletkezik, tehát a hálózat nem viszi át az előző MAC kapcsolat állapotát, azaz az új bázisállomás nem rendelkezik azzal az információval, hogy pontosan mely csomagok lettek már a mobil terminálnak elküldve. Ebből adódó

probléma lehet például a csomagok duplikációja. Az új MAC kapcsolat keletkezéséből adódó problémákat részletes megvizsgáljuk a 4.4. alfejezetben.

Szintén a forrás bázisállomás gondoskodik arról is, hogy a tőle távozó mobil terminálnak címzett csomagokat eljuttassa annak a bázisállomásnak (Target Base Station, T-BS), akihez a mobil átjelentkezett, a továbbiakban ezt a funkciót nevezzük *csomagtovábbításnak*. Ezenkívül a forrás bázisállomás nem csak a hálózaton érkező csomagokat továbbítja, hanem ezelőtt még a pufferében lévő csomagokat is kiveszi, és szintén továbbítja a cél bázisállomás fele. Ezt a funkciót a továbbiakban csomagvisszahívásnak nevezzük. Ha hálózat mind a két funkciót megvalósítja, akkor teljes csomagtovábbításról beszélünk. A 2.1.1. ábrán látható átjáró egység (Gateway, GW) pedig a megfelelő bázisállomás felé irányítja a hálózaton kívülről érkező csomagokat az útvonalirányítási táblája alapján, amelyben nyilvántartja, hogy melyik mobil terminál melyik bázisállomásnál tartózkodik jelenleg.

Továbbá minden egység tartalmaz egy hívásátadást vezérlő elemet, amely felelős a hívásátadás lebonyolításáért. Ezek közül kulcsfontosságú szerepet tölt be a forrás bázisállomás hívásátadás vezérlője, hiszen a forrás bázisállomásnál ismertetett hívásátadással kapcsolatos feladatokat ez az egység koordinálja. Azért nevezzük elosztott hívásátadási eljárásnak ezt a folyamatot, mert nem egy központi egység kezdeményezi és vezérli a hívásátadás egész folyamatát, hanem a forrás bázisállomás tartalmazza ezt a funkcionalitást.

A hálózat működésének pontosabb megismeréséhez a 2.1.2. ábrán felvázoljuk a fentiekben ismertetett egységek egyszerűsített protokollvermét a felsőbb rétegektől eltekintve.



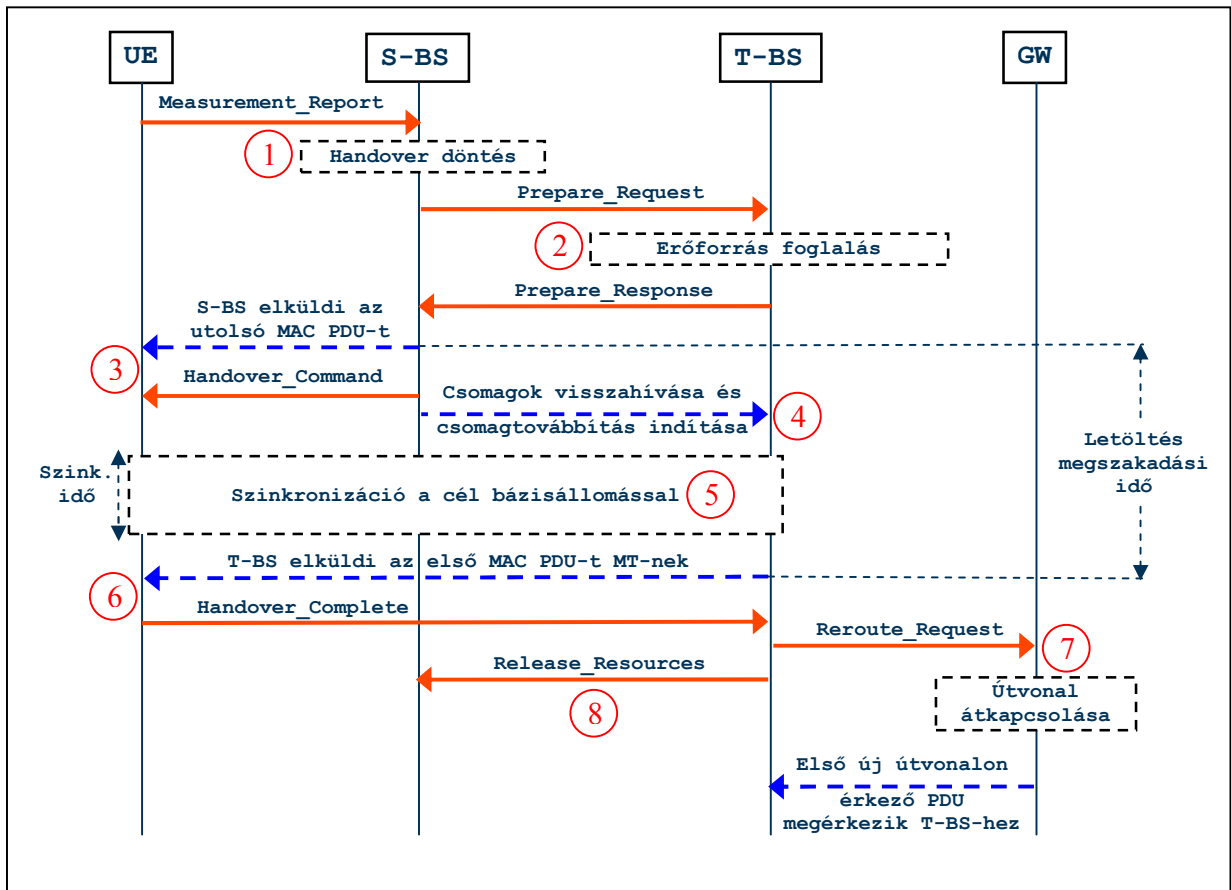
2.1.2. ábra: A vizsgált hálózat protokollverme

A 2.1.2. ábra szemlélteti, hogy az egyes protokoll rétegek hol végződnek [8]. A mobil terminál IP címe nem változik a hálózat során, tehát a felhasználó hálózati rétege nem érzi semmit a mobil hálózaton belüli útválasztásból. Ezt a feladatot az alsóbb réteg kezeli teljes egészében. A felhasználó csomagjait a bázisállomás beágyazza még egy UDP és IP csomagba, amiben a cél IP cím az átjáró egység IP címe, így a transzport hálózaton lévő útválasztók eljuttatják az átjáróig, ami pedig eltávolítja a csomagról ezeket a ráhelyezett fejléceket. Ezután már a felhasználó által összeállított IP csomagban lévő cím alapján halad a csomag a végső célállomásáig. Ezt az eljárást nevezzük *alagutazásnak* (tunneling). A forrás bázisállomás tehát alagutazással juttatja el a felhasználó adatát egy másik, hálózaton belüli egységhez [1]. Ennek az alagútnak az egyik vége a forrás bázisállomás, akinél a mobil terminál éppen tartózkodik, másik vége pedig lehet az átjáró, vagy hívásátadás esetén lehet egy másik bázisállomás is, akinek a csomagokat a forrás bázisállomás továbbítani akarja.

A 2.1.2. ábrán látható még egy PDCP (Packet Data Convergence Protocol) réteg is, amelynek feladata a csomagok fejlécének tömörítése a gazdaságosabb átvitel érdekében, hiszen vannak olyan mezők a fejlécekben, amelyek nem változnak, így érdemes lehet ezeket tömöríteni. Valamint ez a réteg valósítja meg a titkosítást is [1]. A továbbiakban a fent ismertetett hálózatra fogunk hivatkozni vizsgált hálózatként.

2.2. Az elosztott hívásadási eljárás bemutatása

Ebben az alfejezetben részletesen ismertetjük az általunk vizsgált mobil hálózatban alkalmazott elosztott hívásadási eljárást. A 2.2.1. ábrán látható szekvencia diagram részletesen bemutatja, hogy a folyamat során az egyes hívásadást vezérlő egységek milyen jelzésüzenetekkel valósítják meg a kívánt működést.



2.2.1. ábra: Elosztott hívásadási folyamat szekvencia diagramja

A 2.2.1. ábrán tehát azt látjuk, hogy ebben a hálózatban hogyan is zajlik le valójában a hívásadási folyamat elosztott architektúrát feltételezve [9, Chapter 7.8.].

① Ahogy már korábban említettük a felhasználó készüléke (User Equipment, UE) *mérési riportokat* (*Measurement_Report*) *küld* a bázisállomásnak, amelynél tartózkodik. Ezt követően, ha a bázisállomás a riportok alapján úgy dönt, hogy a rádiós link minősége már nem megfelelő, hívásadást kezdeményez. Az új bázisállomást szintén a régi bázisállomás határozza meg a mobil által küldött riportok alapján. Így az lesz az új bázisállomás, amely cellájának jelét a mobil a legnagyobb jelerősséggel tudta venni.

② A forrás bázisállomás jelzési üzenet küld az új bázisállomásnak, hogy próbáljon *erőforrást foglalni* az érkező mobil számára (`Prepare_Request`). Ennek eredményét a `Prepare_Response` jelzésüzenetben küldi vissza a forrásnak.

③ Sikeres erőforrás foglalás után a forrás bázisállomás az utolsó adataegységet elküldi a mobil készülék felé (sikertelen erőforrás allokálás esetén a forrás bázisállomás a következő legnagyobb jelerősségű cellához tartozó bázisállomással próbálja felvenni a kapcsolatot). Innen kezdve a letöltési irányban már egyetlen csomag sem fog haladni a mobil terminál felé, ezt szemlélteti az ábrán a `Letöltés megszakadási idő`. Majd szintén a forrás bázisállomás `Handover_Command` jelzési üzenetet küld a mobil terminálnak, hogy *váltson át a másik cellára*.

④ Ezután a pufferében lévő, és az újonnan érkező, mobilkészüléknek címzett *csomagokat* elkezd *továbbítani az új bázisállomás felé*.

A 4. fejezetben ezt a részt részletesebben fogjuk elemezni, egyrészt összehasonlítjuk, hogy a forrás bázisállomáshoz érkező *csomagok továbbküldése, illetve eldobása* milyen hatással van a teljesítményre, másrészt pedig elemezzük, hogy milyen körülmények között fordulhat elő, hogy a *csomagok sorrendje felcserélődik*, amelynek szintén nagy hatása van az átvitelre. Valamint megvizsgáljuk azt is, hogy a cél bázisállomásnál az új MAC kapcsolat létrehozása során, a *régi MAC állapot elvesztésének hatása* milyen problémákat idézhet elő.

⑤ A szekvencia diagramon látható a *szinkronizációs idő* (`Synchronization time`) is, amely alatt a mobil terminál áthangolja a rádió adóvevőjét az új cella frekvenciájára. Szintén megvizsgáltuk ennek az időparaméter változásának *hatását a TCP átvitelre*, amit a 4.2. alfejezetben részletesen tárgyalunk.

⑥ A szinkronizációs idő letelte után a mobil már képes az új bázisállomással kommunikálni és egy `Handover_Complete` jelzésüzenetet küld neki, amely jelzi, hogy sikeresen megtörtént az átjelentkezés a forrásról a cél bázisállomásra. Közben a cél bázisállomás már elkezd kiküldeni a mobil terminálnak címzett csomagokat.

⑦ Az átjáró egységet is értesíti, hogy a továbbiakban a mobil terminálnak szóló csomagokat ne a forrás felé, hanem a cél bázisállomás felé küldje. Erre szolgál a `Reroute_Request` jelzésüzenet, melynek hatására az átjáró egység átállítja az útvonalirányítási tábláját, és továbbiakban a cél bázisállomás fele küldi a mobil terminálnak címzett csomagokat.

⑧ Végül a cél bázisállomás értesíti a forrást, hogy felszabadíthatja a készüléknek fenntartott erőforrásokat (`Release_Resources`), mivel az sikeresen átjelentkezett hozzá. Ezzel a hívásátadási eljárás folyamata lezárult.

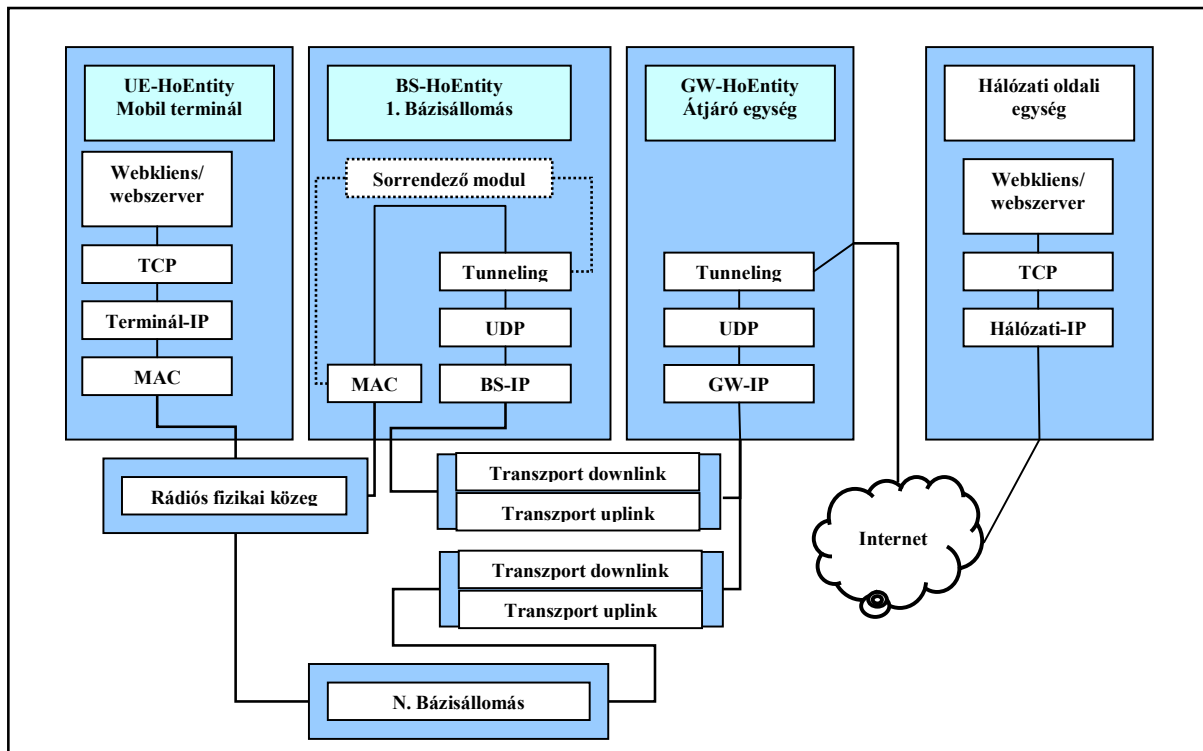
3. A szimulátor bemutatása

Általánosan elmondható, hogy a cellás mobil hálózatok vizsgálata során bonyolult protokollok működését és számos összetett valószínűségi folyamat hatását kell együttesen elemeznünk a teljesítőképesség meghatározásához. Ilyen esetekben célszerű szimulációs eszközöket használni, hogy megfigyelhető legyen, hogy egy valóságos rendszer hogyan reagálna a különböző vizsgálni kívánt esetekben.

Mi a szimulációs vizsgálatainkat egy az Ericsson Telecommunications által fejlesztett szimulátorral végeztük, melyhez a hívásadási folyamat modellezéséhez szükséges, nem létező funkciókat magunk valósítottuk meg. Ezen többlet funkciók integrálásához részletesen meg kellett ismerkednünk a létező szimulátor programmal, valamint számos, a hívásadási folyamat által érintett modult kellett adaptálnunk az új funkciók támogatása érdekében.

A használt program egy diszkrét idejű, esemény alapú szimulátor, amely a hálózati feladatok nagyon széles körének szimulációját és emulációját teszi lehetővé. A kommunikációs hálózatokban megtalálható *funkciókat és eszközöket modulonként valósítja meg*. A modulok közötti kapcsolattartáshoz jól definiált metódusokat biztosít. A fejlesztés és a tesztelés során, a szimulációs rendszer mélyreható ismeretében, az építőelemek és az összeköttetések megadásával összeállítható a vizsgálni kívánt hálózat.

3.1. A szimulátorban megvalósított hálózat felépítése



3.1.1. ábra: A vizsgált rendszer architektúrája

A 3.1.1. ábrán látható az általunk felállított modell, amelyben a felhasználók egy három bázisállomás által lefedett területen bolyonganak véletlenszerűen. Ebben a konfigurációban minden egyes bázisállomáshoz egy-egy cella tartozik. A hálózat által lefedett terület jelátvitel szempontból érdekes adatait egy az adott elrendezésre állandó jelterjedési térkép írja le. Ez a jelterjedési viszonyokat leíró térkép határozza meg tehát a lefedett terület tulajdonságait, például a természetben is előforduló domborzat, illetve más természetes és mesterséges akadály hatását a rádiós jelátvitelre.

A szimulált környezet tulajdonságait számos paraméter befolyásolja. Megadható a rendszerben bolyongó mobil felhasználók száma, valamint be lehet állítani a sebességüket is. Megadható továbbá egy a felhasználók érkezését és kiszolgálását leíró valószínűségi folyamat is, mindkét esetben exponenciális eloszlás szerint, mi azonban a szimulációs vizsgálatoknál az áttekinthetőség és kezelhetőség érdekében fix számú, a rendszer indulásakor már létező felhasználóval dolgoztunk, ami természetesen nem befolyásolja a hívásadási folyamatokat érintő vizsgálatot. Lehetőségünk nyílik a felhasználók fel- és lefele-irányú forgalmának szimulálására is. Bizonyos elméleti méretezési korlátok között lehetőség van a bázisállomások és a hozzájuk tartozó cellák számának megadására is. A felhasználói egység, a bázisállomások, a transzport hálózat és a kiszolgáló oldal funkcionalitásait az ezekre jellemző protokoll veremk határozzák meg, melyek az egyes protokoll rétegekből és az ezek között lévő kapcsolatokból állnak.

3.2. A legfontosabb rétegekben szereplő modulok funkciói és tulajdonságai

Fizikai réteg

A legfontosabb fizikai rétegben használt modul a fizikai átviteli csatornát modellező entitás. Több fajtája van, külön a *rádiós csatornákra*, illetve a *transzport hálózati összeköttetésekre*. A rádiós csatorna jellemzőit rádiós paraméterek, a vivőcsatornák sáv szélessége, a vivőcsatornák száma és az adási jel erőssége határozzák meg. A rádiós moduláció a szimulált hálózatban ortogonális frekvenciaosztásos multiplexálás (Orthogonal Frequency Division Multiplexing, OFDM). A közös rádiós erőforrást a bázisállomásokban megvalósított ütemezők osztják meg a felhasználók között időben és frekvenciában. A szimulátorban implementálták a rádiós jelterjedést véletlenszerűen befolyásoló fading hatásokat, a fast fading-et és a shadow fading-et, valamint számításba vették a rádiós jelek távolsággal arányos terjedési tulajdonságait is. A vezetékessé átvitelt modellező transzport csatornának állíthatjuk az adatátviteli sebességét, a hosszúságára jellemző fix késleltetését, valamint a csatornához tartozó puffer méretét és a puffer kezelését meghatározó sormenedzsment algoritmust. A transzport csatorna paramétereit a szimulációs vizsgálatok során mindig az egyes tesztesetekhez igazítottuk.

Adatkapcsolati réteg

A vizsgált hálózatban adatkapcsolati réteget a mobil terminál és a bázisállomás között szimuláltunk. A szimulátorban az adatkapcsolati rétegbeli funkciókat MAC (Medium Access Control) végzi el, amely megvalósítja többek között a felsőbb rétegtől kapott adategység (Service Data Unit, SDU) *szegegmentálását* és a MAC keretek *sorrendhelyes továbbítását* a felső rétegnek. A mobil hálózatokban a legtöbb *csomagvesztés*, illetve meghibásodás *a rádiós interfészen* következik be, ezért ilyenkor egy plusz hibajavító mechanizmust alkalmaznak annak érdekében, hogy a csomagvesztés hatása ne érjen el felsőbb réteget. Ezt Hibrid ARQ (HARQ) mechanizmusnak nevezik és a használt szimulátor MAC rétegében is megvalósították. Ez a hibajavító mechanizmus úgynevezett HARQ processzek segítségével valósítja meg a rádiós linken keletkezett hibázások kijavítását. A HARQ processzek működését részletesebben ismertetjük a 4.4. alfejezetben.

A szimulátorban a MAC működését két részre választották szét, *adó* és *vevő* részre. Ha letöltés van folyamatban, akkor a bázisállomás MAC adója küldi az adatot a mobil terminál MAC vevőjéhez, illetve feltöltés esetén pedig a mobil terminál MAC adója küldi a kereteket a bázisállomás MAC vevőjéhez. Az adó oldalon található egy SDU puffer, amibe a felsőbb rétegtől kapott adategységet helyezi el, majd az éppen küldésre kerülő SDU feldarabolva átkerül egy másik pufferbe, ahonnan csak akkor veszi ki az adó, ha azt sikerült a vevőnek sikeresen átjuttatni. A vevőnél is van egy puffer, hogy az érkező kereteket összeállítva és sorrendhelyesen tudja a felsőbb rétegnek feladni.

Ahogy a 2. fejezetben már említettük, hívásátadás során új MAC kapcsolat jön létre, tehát az aktuális *MAC réteg állapotát nem visszük át* az új kapcsolatba. Emiatt a MAC réteg küldési mechanizmusát ki kellett egészítenünk aszerint, hogy képes legyen ebben az esetben is biztosítani a veszteségmentes csomagtovábbítást.

Hálózati réteg

Mint a rendszer architektúrájából is látszik, a felhasználó (Terminál-IP) és a kiszolgáló (Hálózati-IP) oldalán, az átjáró egységben (GW-IP), valamint minden bázisállomásban (BS-IP) található egy-egy IP entitás, illetve a transzport hálózat egy IP gerinchálózat, ahol az egyes csomópontokat szintén IP entitások modellezik. A hálózati réteg feladata az IP entitások közötti *útvonalválasztás* megvalósítása, melyet szintén mi integráltunk a szimulátorba. A szimuláció során az *átjáró* egységnek speciális feladata is van, a csomagokat a forrás bázisállomás helyett már egy másik, a cél bázisállomás IP entitásához kell irányítani, melyet *az útvonal irányító táblázatának megváltoztatásával* ér el.

Az átjáró egységekben és a bázisállomásokban található IP entitások minden esetben egy UDP (User Datagram Protocol) szállítási rétegbeli entitáshoz [7, Chapter 11.] kapcsolódnak. Ezek az UDP/IP entitás párok valósítják meg az *alagútszerű átvitelt*, a tunneling-ot. Amennyiben egy bázisállomás, vagy az átjáró egység egy másik bázisállomásnak szeretne adatot küldeni, úgy azt beágyazza egy UDP szegmensbe, és így küldi tovább a hálózati rétegnek.

A 3.1.1. ábrán azt látjuk, hogy a bázisállomásokhoz tartozó IP entitások közvetlenül a transzport hálózati fizikai közeget modellező egységre kapcsolódnak. Ez a szimulátor

egyik sajátossága, ugyanis a megvalósított protokoll rétegek az absztrakciós szint mélységétől függhetnek. Ebben az esetben például, a transzport hálózaton nem valósítottuk meg a fizikai közeghez hozzáférő MAC protokollt, mert így a modell egyszerűbb lett. Ennek az elhanyagolásnak a szimulációk során nincs hatása.

Szállítási réteg

A vizsgált hálózatban a TCP (Transmission Control Protocol) szállítási rétegbeli entitás [7. Chapter 17-24.] szelektív nyugtázással (SACK – Selective ACKnowledgment) működő változatát használjuk a végpont-végpont közötti *megbízható kommunikációra*. A szimulációk szempontjából legfontosabb paramétere a TCP torlódási ablak (congestion window) mérete adó és vevő oldalon, melyeket minden esetben az adott tesztesetekhez igazítottunk.

A szintén szállítási rétegbeli UDP entitást csak az előző bekezdésben említett alagút technikához használjuk, ugyanakkor meg kell említeni, hogy amennyiben például VoIP forgalmat szeretnénk a felhasználó és a kiszolgáló között küldeni, úgy végpont-végpont között UDP entitást lehetne használni.

Mint a valóságban, a szállítási szintű kapcsolatok itt is függetlenek egymástól, így annyi TCP-TCP, illetve UDP-UDP entitás párra van szükség a szimuláció során, amennyi az egyes kapcsolatok száma. A szimulátor moduláris szerkezete jól alkalmazkodik tehát a hálózati funkciók megvalósításához.

Alkalmazási réteg

A szimulációs vizsgálatok során a kommunikáció minden esetben két alkalmazási rétegbeli protokoll egység, egy *webszerver* és egy *webkliens* között zajlik. Amennyiben lefele-irányú (downlink) forgalmat szeretnénk szimulálni, úgy a felhasználói egységben a webkliens, hálózati oldalon pedig a webszerver található. Felfele-irányú forgalom szimulációja esetén a szerepek felcserélődnek, ekkor a felhasználói készülék tölt fel a távoli, hálózati oldali egységre.

A webszerver entitás legfontosabb paramétere az átküldendő fájl mérete. Egy felhasználó addig tartózkodik a rendszerben, amíg a paraméterként megadott méretű fájlt fel-vagy letölti.

3.3. A hívásátadási folyamat szekvenciáját megvalósító modulok

s-BS-HoEntity - a forrás bázisállomás oldali hívásátadást vezérlő modul

Ez az egység a folyamat központi szereplője, hiszen a felhasználói egységtől kapott mérési adatok alapján a forrás bázisállomásnak kell eldönteni, hogy szükség van-e a hívásátadásra. Amennyiben igen, úgy a mobil készüléktől kapott adatok alapján meghatározza a cél bázisállomást, majd az erőforrások lefoglalását kezdeményezi ezen távoli oldalon. Handover parancsot küld a felhasználói készüléknek, vezérli az adatok visszahívását a MAC rétegből, és a rádiós kapcsolat bontása után még hozzá érkező csomagokat átirányítja a cél bázisállomás felé.

t-BS-HoEntity - a cél bázisállomás oldali hívásátadást vezérlő modul

A forrás bázisállomás által kezdeményezett kérésre erőforrásokat foglal le az új adatkapcsolat kiszolgálásához. Amennyiben például a hálózati terhelés miatt az erőforrás foglalás sikertelen, úgy negatív választ küld, jelezvén a forrásnak, hogy másik célt kell választania.

A mobil készüléktől érkező Handover_Complete parancs hatására kezdeményezi az útvonal irányítás megváltoztatását az átjáró egységben, majd az erre kapott pozitív válasz után utasítást küld a forrás bázisállomásnak az adott adatátvitel számára lefoglalt erőforrások felszabadítására.

A cél bázisállomás további feladata, hogy az ide érkező, a forrás bázisállomás által átirányított, illetve a már az átjáró egység által is errefelé küldött csomagok közötti sorrendet megőrizze. Ehhez egy külön sorrendező egységet kellett bevezetnünk, amit a 3.1.1. ábrán szereplő rendszerarchitektúrán is szerepeltettünk. A sorrendező egységnek csak a hívásátadási folyamat közben van szerepe, ezért ezt a modult és a kapcsolatait szaggatott vonallal jelöltük. Azt hogy egy csomag a cél, vagy a forrás bázisállomástól érkezett a 3.1.1. ábrán is szereplő Tunneling entitás képes meghatározni. A csomagok felcserélődésének problémáját a 4.3. alfejezetben részletesen is ismertetjük.

Amennyiben a 4.4. alfejezetben ismertetésre kerülő csomagduplikációt is kezelni szeretnénk, úgy a bázisállomásokat kellene felkészíteni ennek a problémának a megoldására egy duplikáció detektáló és kiszűrő modul segítségével.

UE-HoEntity - felhasználó oldali hívásátadást vezérlő modul

Periodikusan rádiós mérési eredményeket küld a bázisállomásnak, majd a handover parancs hatására megszünteti a rádiós kapcsolatát, melyet csak a szinkronizációs idő letelte után, a cél bázisállomással vesz fel újra.

GW-HoEntity - az átjáró egység hívásátadást vezérlő modulja

Egyetlen feladata, hogy a cél bázisállomás kérésére megváltoztassa az útvonalválasztó bejegyzéseit. A változtatás után a mobil készüléknek szóló csomagok mind a cél bázisállomás felé irányítódnak.

A 3.3. pontban bemutatott modulok implementálását és szimulátorba illesztését mi valósítottuk meg. Ennek eredményeképpen egy olyan eszköz készült, amellyel a 2. fejezetben bemutatott elosztott hívásátadási folyamat tulajdonságai, előnyei és korlátai elemezhetők.

4. A hívásadási eljárás működésének vizsgálata szimuláció segítségével

Ez a fejezetben az elosztott hívásadási eljárás során részletesen megvizsgált tervezői döntések értékeléséről szól. Összesen öt ilyen kérdést vizsgáltunk meg, amelyeket szimulációs eredmények alapján értékeltünk.

4.1. A csomagok kezelésének kérdése a forrás bázisállomásnál a hívásadási folyamat során

Az általunk vizsgált elosztott hívásadási folyamat egyik legfontosabb feladata a *csomagvesztés elkerülése*, hiszen a csomagvesztés a TCP torlódási ablak felezését okozza, csökkentve ezzel az átvitel teljesítményét. Ebben az alfejezetben két tervezési döntés hatását vizsgáljuk meg szimulációs módszerekkel, melyek a csomagvesztés elkerülését szolgálják, és melyek ezen hívásadási folyamat sajátosságai. Elsőként a hívásadási folyamat során, a felhasználói egység és a forrás bázisállomás közötti rádiós kapcsolat lezárását követően érkező csomagok cél bázisállomás felé történő továbbítását vizsgáljuk, majd ezt az esetet kombináljuk az adatkapcsolati és hálózati szintű együttműködés eredményeképpen, a *MAC rétegből visszahívott és továbbított csomagok* vizsgálatával. Az összehasonlítást mindkét esetben a csomagok átirányítását nem használó változathoz végezzük.

Vizsgált tesztesetek

Referencia

A forrás bázisállomás nem hívja vissza a csomagokat a MAC rétegből, illetve a további, a transzport hálózatból hozzá érkező csomagokkal sem foglalkozik, eldobja ezeket.

Részleges csomagtovábbítás

A forrás bázisállomás nem hívja vissza a csomagokat a MAC rétegből, de a további, a transzport hálózatból hozzá érkező csomagokat a cél bázisállomás felé irányítja.

Teljes csomagtovábbítás

A forrás bázisállomás mind a MAC rétegből visszahívott, mind a további, a transzport hálózatból hozzá érkező csomagot a cél bázisállomás felé irányítja.

Az esetek bemutatásához egy felhasználót szimuláltunk a rendszerben, aki 10 m/s-os sebességgel véletlenszerűen bolyong három bázisállomás által lefedett területen, és egy távoli, Internetre kapcsolt webszerverről tölt le egy 100 MB-os fájlt. A TCP torlódási ablak mérete adó és vevő oldalon 128 kB-át.

Hívásadások minden esetben a következő időpontokban vannak:

- 30,5 másodperc : 1. handover
- 31,0 másodperc : 2. handover

- 32,5 másodperc : 3. handover
- 33,5 másodperc : 4. handover
- 55,75 másodperc : 5. handover
- 94,0 másodperc : 6. handover

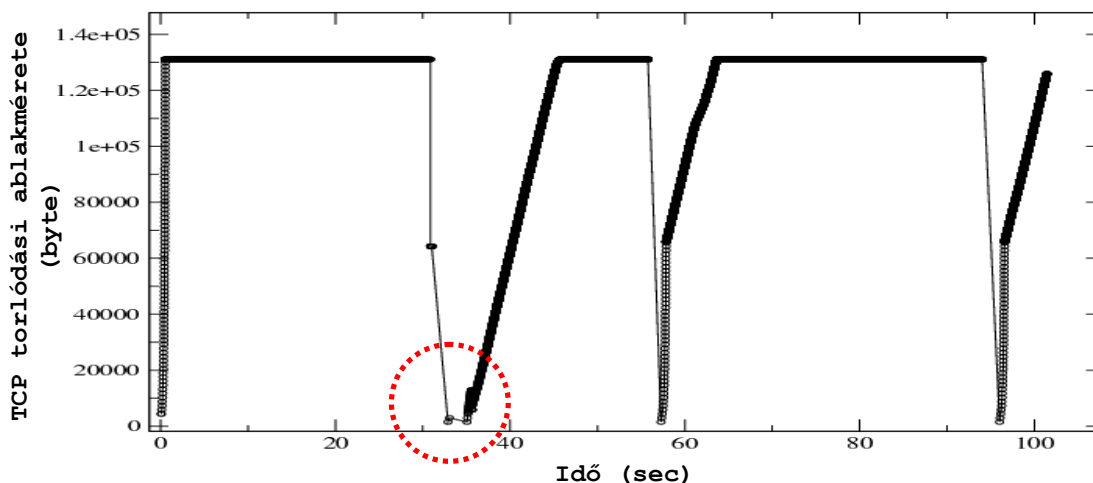
A hívásátadási időpontokat a mobil terminál véletlen mozgása, valamint a rádiós csatorna véletlenszerű változásai határozzák meg. A fenti listából látható, hogy a 30-ik másodperc környékén sűrűn vannak hívásátadások. Ez annak a következménye, hogy a lefedett cellák határán a rádiós jelerősség pillanatról pillanatra nagyon gyorsan megváltozhat, és ez elegendő a hívásátadás megindításához. Meg kell jegyeznünk, hogy egy adott felhasználó adatátviteli folyamatát tekintve, a szimuláció során használt késleltetési paraméterek garantálják, hogy egy hívásátadási folyamat mindig befejeződik, mielőtt egy újabb történhetne. Megszakított hívásátadással tehát nem kell számolnunk.

A szimulátorban használt véletlen szám generátort a tesztesetek összehasonlíthatóságának érdekében mindig azonos kezdőértékre állítottuk, így fordulhat elő az, hogy mindhárom vizsgált konfigurációban ugyanakkor történnek a hívásátadások.

A vizsgált tesztesetek szimulációs vizsgálata

A referencia eset szimulációs vizsgálata

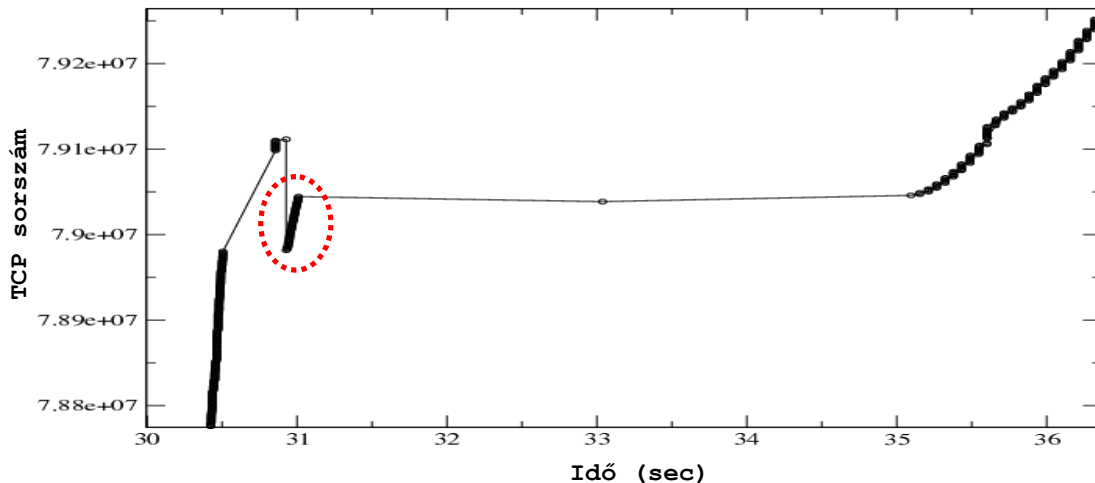
Mivel ebben az esetben a forrás bázisállomás nem végez csomag átirányítást, ezért mind a hívásátadási folyamat során a forrás bázisállomásnál ragadt csomagok, mind az útvonalválasztás megváltozása előtt még ide címzett csomagok elvesznek. Egymás utáni handover-ek esetén a sorozatos egymás utáni vesztesékre a TCP timeout-tal reagál, hiszen ekkor egy hosszabb időtartam alatt nem érkezik csomag, ami jelentősen csökkenti az átviteli minőséget.



4.1.1. ábra: A referencia eset szimulációja – a TCP torlódási ablak mérete

A 4.1.1. ábrán megfigyelhető a sorozatos csomagvesztések és egymás utáni handoverek miatti TCP timeout okozta kiesés. A bejelölt részen megfigyelhető, hogy a TCP torlódási

ablak mérete nulla közelébe ér, igazából minimális értéként éppen egy szegmens bájtnban kifejezett méretét veszi fel. A TCP torlódási ablak mérete majd csak a sorozatos hívásátadások után, másodpercek múlva fog ismét emelkedni.



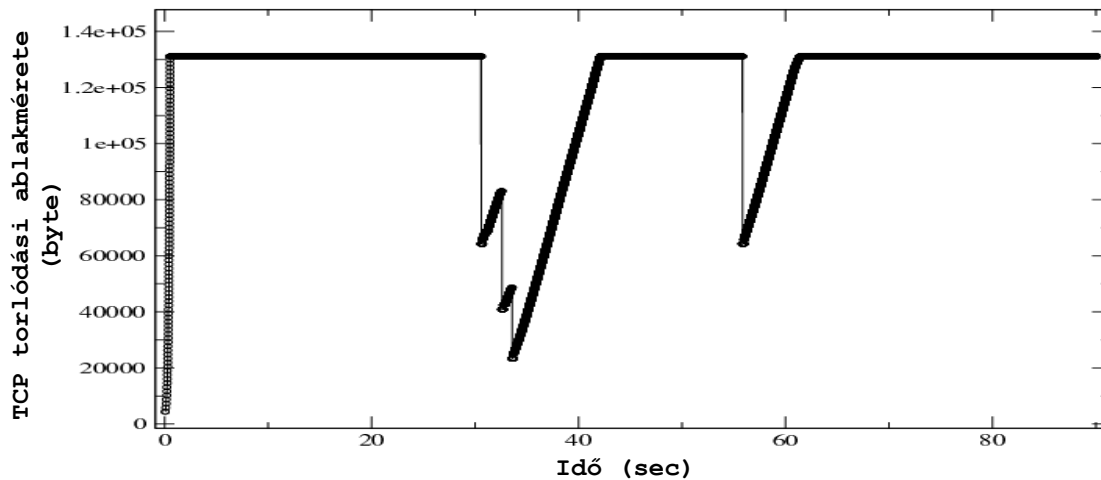
4.1.2. ábra: A referencia eset szimulációja – a TCP szintű átvitel handover során

A 4.1.2. ábráról jól látható, hogy a csomagok érkezése a 30,5-ik másodpercben bekövetkezett handover során hirtelen megszűnik, az elveszett csomagok majd egyben, a hívásátadási folyamat vége után érkehetnek meg. Ebben az esetben ráadásul egy a 31-ik másodpercben bekövetkező újabb handover *TCP timeout-ot okoz*, mivel az újraküldött csomagok sem érkeznek meg időben a felhasználói készülékhez, és így az ezekre a csomagokra küldött nyugták sem érkeznek meg időben az adó oldali TCP entitáshoz. A kapcsolat csak a 35-ik másodperc után áll helyre, a hívásátadások után. A 4.2. alfejezetben látni fogjuk, hogy a TCP timeout mechanizmus a mobil készülék rádiós csatorna váltási idejével is szoros összefüggésben van.

A 100 MB-os fájl átvitele a referencia esetben 101,4 másodpercig tartott.

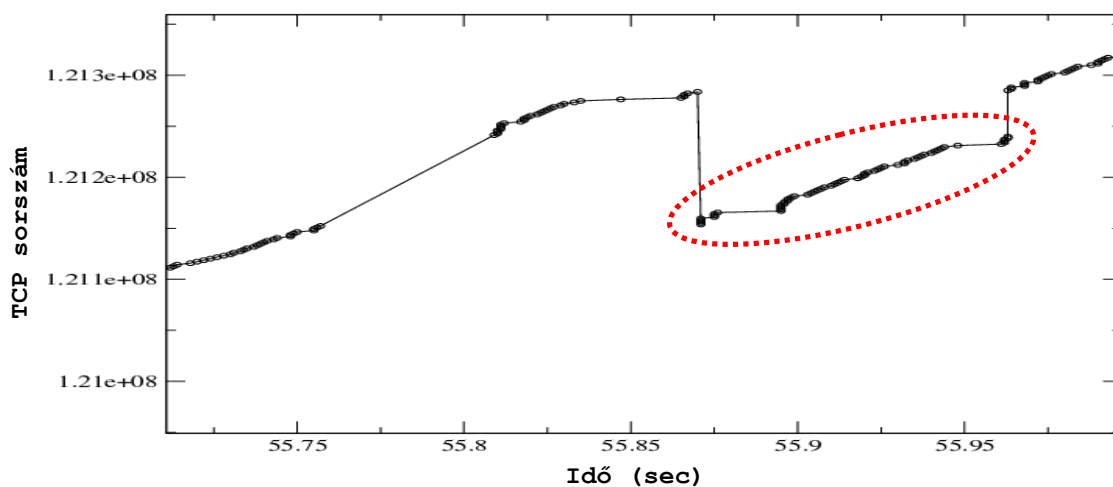
A részleges csomagtovábbítás szimulációs vizsgálata

Ebben az esetben ugyan elvesznek a forrás bázisállomásnál a MAC rétegben ragadt adatok, hiszen ekkor ezeket nem hívtuk vissza, de a rádiós kapcsolat bontása után a még ide érkező csomagok mind továbbirányítódnak a cél bázisállomás felé. Mivel a csomagvesztés csupán a MAC rétegben ragadt adatokat érinti, ezért a csomagok érkezése a cél bázisállomáshoz, majd a rádiós kapcsolat feléledése után a felhasználói készülékhez folyamatos, így TCP timeout sem történik, mindössze a TCP torlódási ablak mérete csökken a felére a hiányzó csomagok miatt. Ebben az esetben tehát jelentősen nő az átvitel minősége.



4.1.3. ábra: A részleges csomagtovábbítás szimulációja – a TCP torlódási ablak mérete

A 4.1.3. ábrán látszik, hogy a TCP a pufferben ragadt csomagok elvesztésére a TCP torlódási ablak felezésével reagál, de ebben az esetben már *nem történt TCP timeout*. A hívásátadási folyamatok után a TCP torlódási ablak mérete lineárisan növekszik.



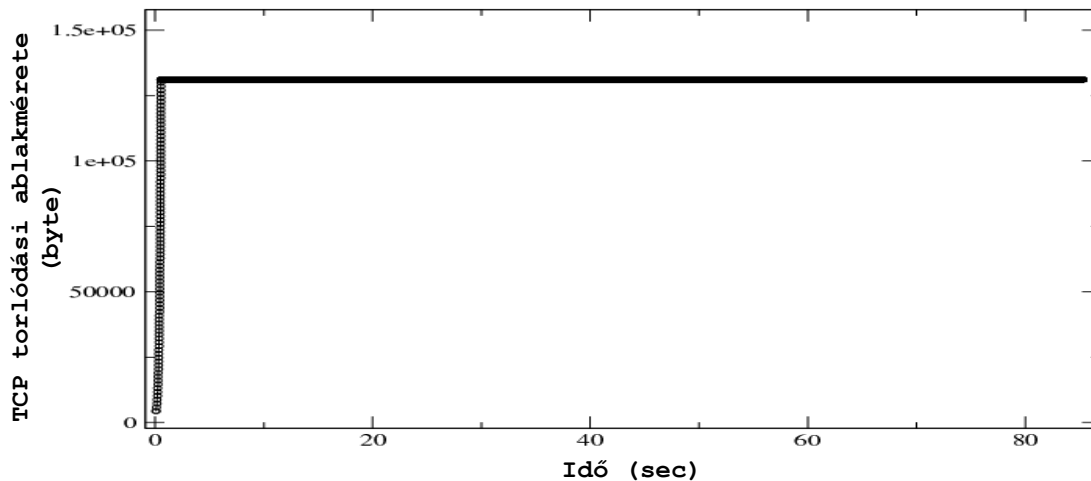
4.1.4. ábra: A részleges csomagtovábbítás szimulációja – a TCP szintű átvitel handover során

Az átvitelt szemléltető 4.1.4. ábrán bejelöltük, hogy a hívásátadási folyamat során az adatkapcsolati réteg pufferében ragadt csomagok mikor kerülnek újraküldésre. A szimulációs kimeneti adatok alapján az 55,75-ik másodpercben 60 SDU volt a MAC rétegben, amit nem hívtunk vissza, így a bejelölt részen pontosan 60 újraküldött szegmens van.

Ebben az esetben a 100 MB-os fájl átvitele 90 másodpercig tartott.

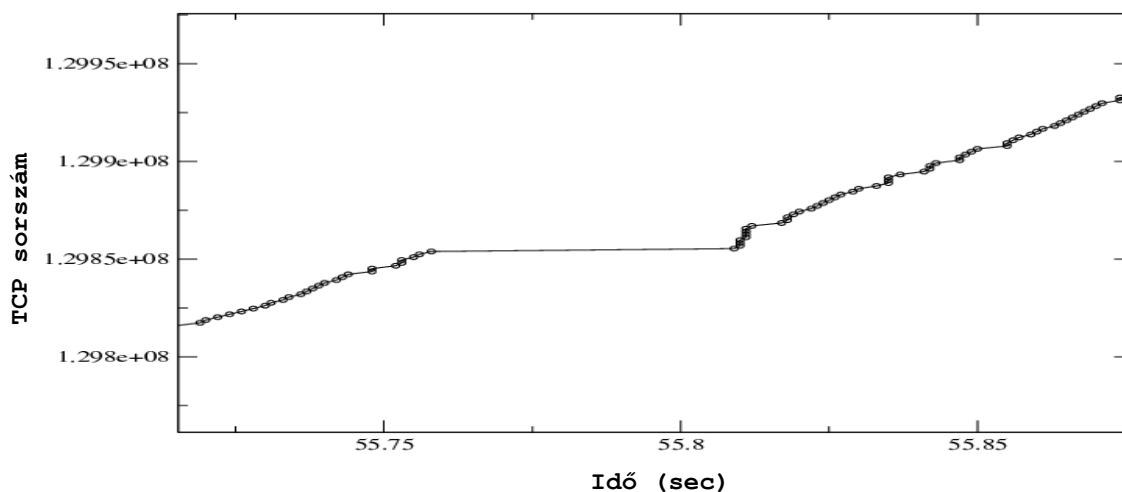
A teljes csomagtovábbítás szimulációs vizsgálata

A második megvizsgált esetben mind a forrás bázisállomásnál a MAC rétegben ragadt adatokat, mind a még ide címzett további csomagokat átirányítjuk a cél bázisállomás felé a veszteségmentesség elérésének céljából. Mivel ekkor minden elküldött adat időben megérkezik, *nem történik csomagvesztés*. A hívásátadási folyamatok miatt nem csökken le a TCP torlódási ablak mérete, az átviteli minőség ebben az esetben lesz a legjobb.



4.1.5. ábra: A teljes csomagtovábbítás szimulációja – a TCP torlódási ablak mérete

A 4.1.5. ábra mutatja, hogy a TCP torlódási ablak egyszer sem feleződött le az adatátvitel során.



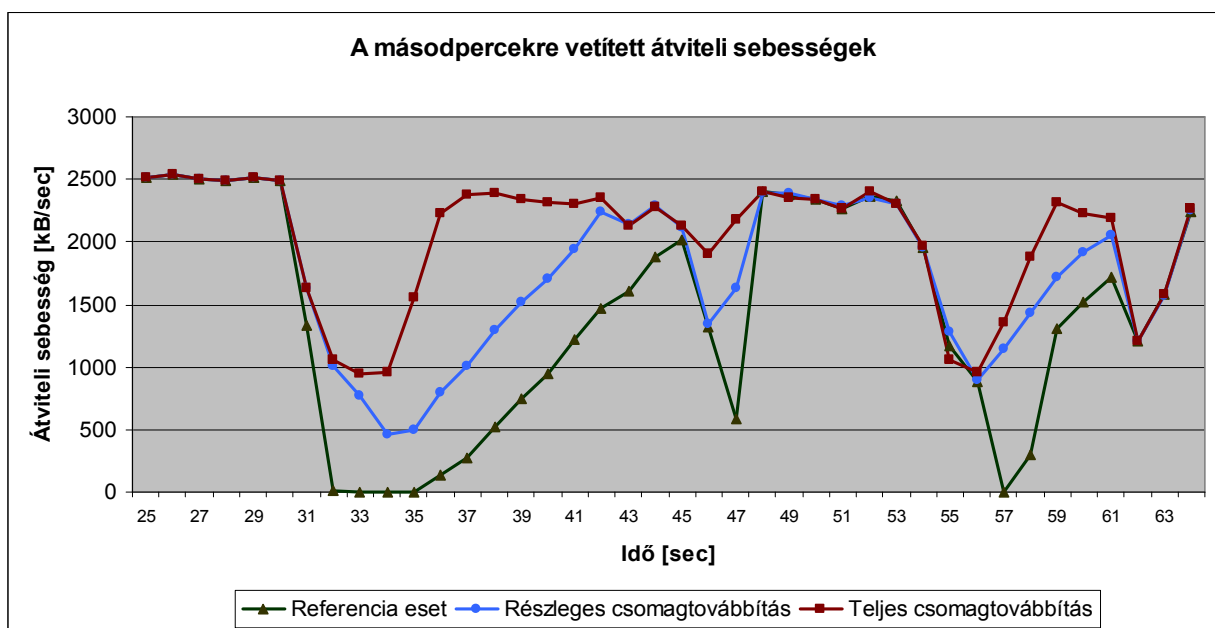
4.1.6. ábra: A teljes csomagtovábbítás szimulációja – a TCP szintű átvitel handover során

A 4.1.6. ábrán látszik, hogy ugyan kissé megnövekedett késleltetéssel, de minden csomag megérkezik a felhasználói mobil egységhez. Ezt a megnövekedett késleltetést főleg a mobil készülék rádiós csatorna váltási ideje határozza meg, mely időparaméter hatását szinkronizációs idő néven a 4.2. alfejezetben vizsgáljuk.

Ebben az esetben a 100 MB-os fájl átviteléhez 85,3 másodpercre volt szükség.

Következtetés

A 4.1.7. ábra egy diagramon szemlélteti a három különböző eset során kapott másodpercekre vetített átviteli sebességeket a letöltési folyamat során.



4.1.7. ábra: Másodpercekre vetített adatátviteli sebességek a különböző esetekben

Megállapíthatjuk tehát, hogy érdemes a bázisállomásokat felkészíteni a MAC rétegben ragadt adatok visszahívására, valamint az érkező csomagok átirányítására, hiszen ebben az esetben a hívásátadási folyamat *csomagvesztés nélkül* képes bázisállomást váltani, és ezáltal jelentősen nő az adatátvitel teljesítménye.

Az átviteli minőség növelése mellett, a csomagok átirányításának problémái is lehetnek, melyet a 4.3. alfejezetben vizsgálunk meg részletesen.

4.2. TCP átvitel vizsgálata a szinkronizációs idő függvényében

Ez az alfejezet foglalkozik a hívásadási eljárás egyik legfontosabb paraméterének, a szinkronizációs idő változásának a TCP átvitelre gyakorolt hatásával. A szimuláció során különválasztottuk azt az esetet, ha van teljes csomagtovábbítás, illetve ha nincs.

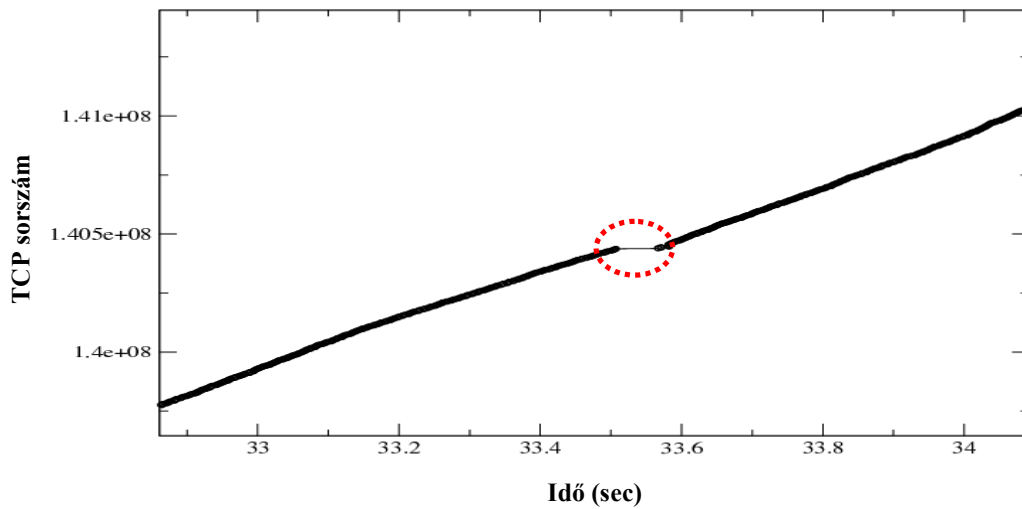
A szinkronizációs idő paraméter azt az időt modellezi, ami ahhoz szükséges, hogy a mobil terminál áthangolja a rádió adóvevőjét egyik cella frekvenciájáról a másikra. Ezalatt a készülék nem képes semmilyen csomagot sem venni, hiszen ezen idő alatt nincs kapcsolata a hálózattal.

A szimulációs eredményekben megmutatjuk egyrészt, hogy hívásadáskor a vevő által kapott TCP szegmensek érkezései között eltelt időt hogy befolyásolja a paraméter változása. Másrészt, hogy a pillanatnyi, illetve kumulált átviteli sebesség hogyan változik a szinkronizációs idő függvényében.

A rendszerben most egy felhasználót szimulálunk, aki 10 m/s sebességgel véletlenszerűen halad 3 bázisállomás között, és letöltést végez egy távoli, Interneten lévő webszerverről. A szimulációt teljes csomagtovábbítással és anélkül is megvizsgáltuk, melynek eredményét a következőkben mutatjuk be.

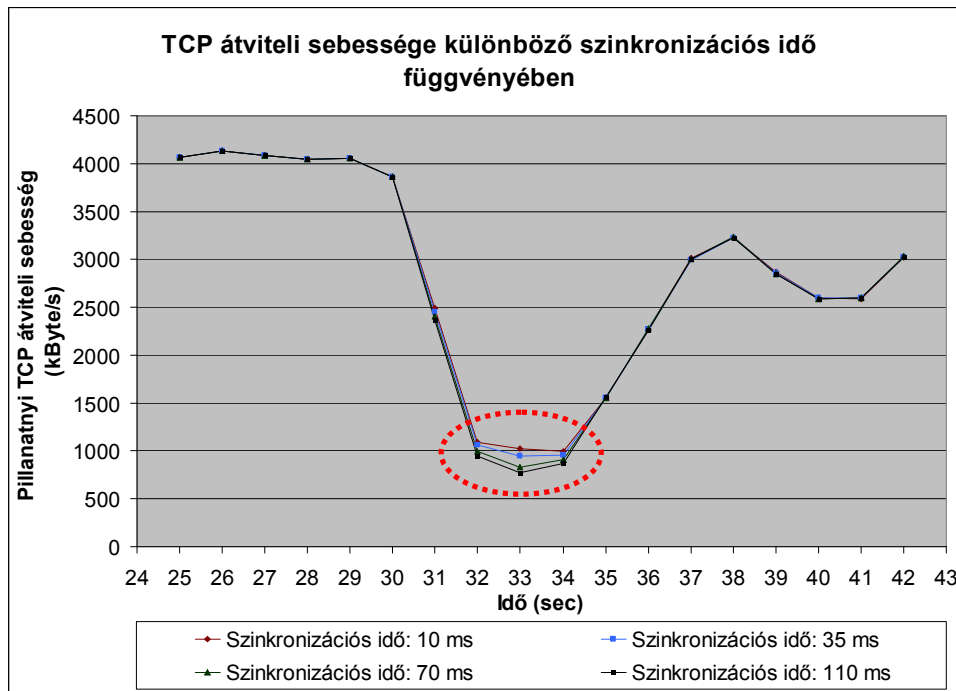
Szinkronizációs idő hatása a TCP átvitelre teljes csomagtovábbítással

A szimulációt 10, 35, 70 és 110 ms nagyságú szinkronizációs idővel futtattuk, majd megnéztük, hogy a vevő oldalon hogyan jelenik meg a szinkronizációs idő hatása. Az eredmények alapján a TCP sorszámokat (sequence number) ábrázoltuk az idő függvényében. A TCP sorszámozási mechanizmusából adódóan a TCP sorszám megegyezik a vevő által megkapott adatok byte-ban mért mennyiségével. A 4.2.1. ábrán jól látható, hogy a vevő körülbelül a szinkronizációs idő nagyságával megegyező ideig egyáltalán nem kap adatot. Ezt mutatja az ábrán bekarikázott rész, ahol a szinkronizációs idő 35 ms volt. Más szinkronizációs idő beállítás esetén is hasonló eredmény kaptunk, csak a csomag érkezése nélküli idő hossza változott a szinkronizációs időnek megfelelően.



4.2.1. ábra: Szinkronizációs idő: 35 ms

A 4.2.1. ábrán az is látható, hogy miután letelt a szinkronizációs idő a csomagok újból elkezdnek megérkezni a vevőhöz. Azt megjegyezzük, hogy a 35 ms-os szinkronizációs idő nem pontosan 35 ms-os kihagyást jelent a csomagok vételében, hanem egy rádiós link késleltetéssel többlet, mivel ennyi idő szükséges, míg az első csomag megérkezik a mobil terminálhoz miután a rádiós link szinkronizációja befejeződött, és az új bázisállomás elkezdte ütemezni a csomagok küldését. Ezért a 35 ms-os szinkronizációs idő körülbelül 40 ms-os kihagyást eredményez a csomagok vételénél.



4.2.2. ábra: Szinkronizációs idő hatása a TCP átvitelre

A 4.2.2. ábrán látható, hogy hívásátadások esetén (30 – 34 sec között) a pillanatnyi TCP átviteli sebesség kis mértékben egyre csökken, ahogy növeljük a szinkronizációs időt. Az ábrán az is észrevehető, hogy hívásátadások időpontja közelében jelentősen csökken a pillanatnyi TCP átviteli sebesség a szinkronizációs idő nagyságától függetlenül, annak ellenére, hogy csomag nem veszett el és így a TCP torlódási ablakának mérete egyszer sem csökkent le a hívásátadások során. Ennek a jelenségnek az oka az, hogy a csomagvisszahívás, továbbítás és sorrend visszaállítása miatt a vevőhöz jelentősen nagyobb késleltetéssel érkeznek meg a csomagok, ami a másodperce vetített TCP átviteli sebességet nagymértékben csökkenti.

A 4.2.1. táblázat alapján láthatjuk, hogy a kumulatív TCP átviteli sebesség az 50 másodperc letöltési idő alatt csak kis mértékben csökkent, tehát lényegesen nem változott a szinkronizációs idő növelése miatt.

Szink. idő (ms)	TCP átviteli sebesség (Kbyte/s)
10	3151.72
35	3144.22
70	3131.33
110	3122.52

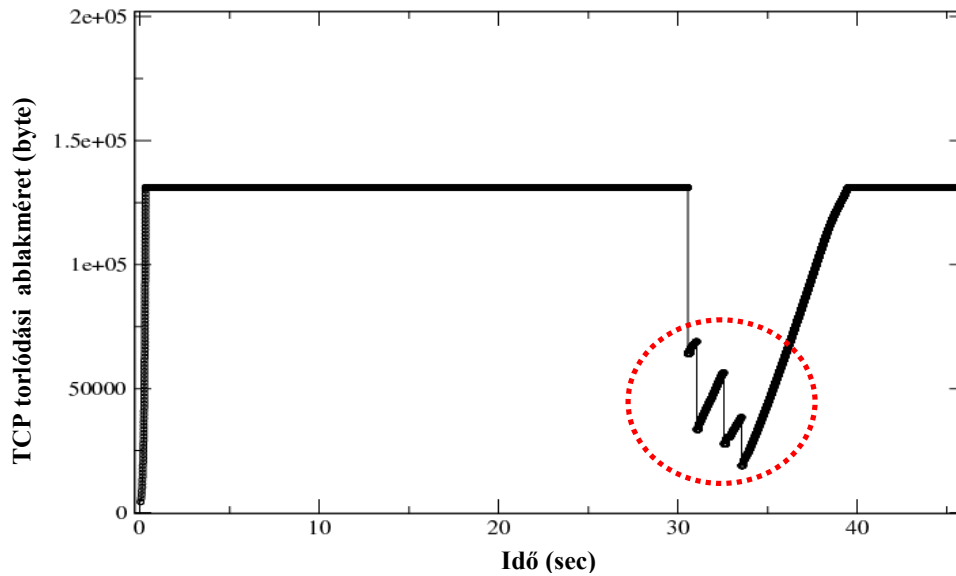
4.2.1. táblázat: Kumulált TCP átviteli sebességek 50 sec letöltési idő alatt

Összefoglalva azt mondhatjuk, hogy csak akkor van észrevehető hatása a szinkronizációs idő változásának, ha a letöltés rövid idejű (néhány másodperces) és hívásátadások pont ebben az időben következnek be. Ahogy a TCP kapcsolat ideje nő, egyre inkább elhanyagolhatónak tekinthető ennek a paraméternek az értéke.

Szinkronizációs idő hatása a TCP átvitelre teljes csomagtovábbítás nélkül

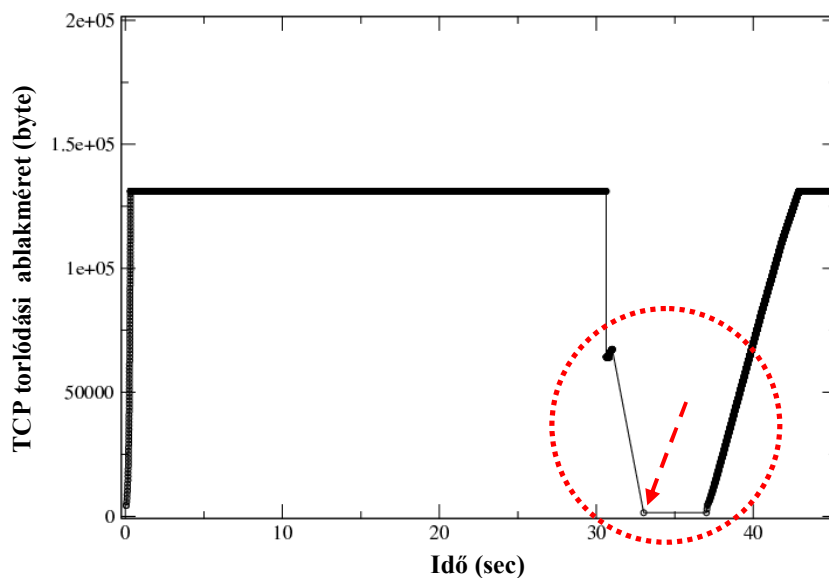
A fenti eredmények után megvizsgáltuk, hogy a szinkronizációs időnek milyen hatása van akkor, ha nem használjuk a teljes csomagtovábbítási funkciót. Ami azt jelenti, hogy a forrás bázisállomás hívásátadás bekövetkezte után nem továbbítja a pufferében ragadt és az ezután hozzá érkező csomagokat a cél bázisállomás felé, hanem egyszerűen eldobja azokat. Ebből következően a TCP torlódási ablakának mérete biztosan többször is feleződni fog, mivel több csomag elvész hívásátadásokor, ellentétben az előzőekben szimulált esettel, ahol egyáltalán nem csökkent az ablak mérete.

A veszteségmentes esettel egyező paraméter beállításokkal most is lefuttattuk a szimulációt 10, 35, 70, és 110 ms szinkronizációs időket beállítva. A 4.2.3 ábra szemlélteti a TCP torlódási ablakának változását 10 ms-os beállítás esetén.



4.2.3. ábra: A TCP torlódási ablakának mérete 10 ms-os szinkronizációs idő esetén

Látható a fenti ábrán, hogy minden hívásátadás esetén a TCP felére csökkenti a torlódási ablakának méretét, ami a forrás bázisállomásnál eldobott csomagok miatt következett be. A 35 ms-os szinkronizációs idő beállításával kapott eredményt a 4.2.4. ábra mutatja.

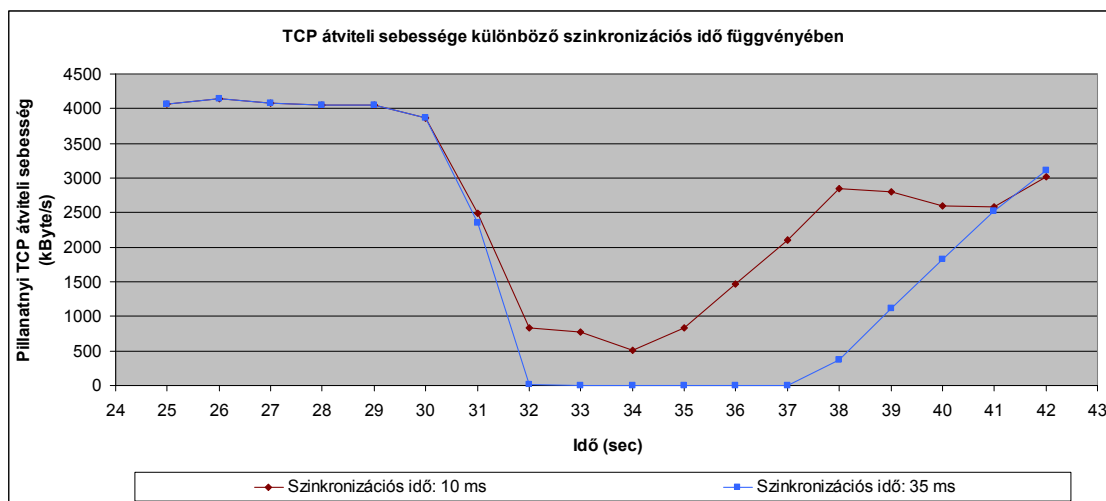


4.2.4. ábra: A TCP torlódási ablakának mérete 35 ms-os szinkronizációs idő esetén

A 4.2.4. ábrán láthatjuk, hogy az első hívásátadás után csak lefelezett a TCP torlódási ablaka, aztán a következő hívásátadás után pedig egy csomagméretnyre csökkent le. A TCP működéséből adódóan ez a viselkedés akkor fordul elő, ha a TCP küldésében timeout következik be, azaz a csomagokról sem pozitív, sem negatív nyugta nem érkezett meg a vevőhöz az RTO (Retransmission Timeout) idő lejártáig. Ekkor a TCP egy bizonyos ideig nem küld csomagot, majd megpróbál egy csomagot küldeni, ezt a csomagot mutatja a nyíl a 4.2.4. ábrán. Erről a csomagról sem érkezett pozitív vagy negatív nyugta a vevőhöz, így a TCP ismételt timeout miatt az előző várakozási idejének kétszerese ideig vár (exponenciális backoff). A várakozás leteltekor már megtörténtek a hívásátadások, ezért az újból leküldött csomag már sikeresen áttért a vevőhöz, mert az újabb csomagok generálását idézte elő. Az első RTO idő letelte után küldött csomag azért nem ért el a vevőig, mert egy handover következett be miközben a hálózaton utazott a felhasználó készüléke felé, ami miatt ez a csomag eldobásra került a forrás bázisállomásnál. A második próbacsomag már sikeresen áttért mobil terminálhoz, mert ekkor már nem volt hívásátadás, ami megzavarta volna az adatküldést.

Ezek alapján azt tapasztaljuk tehát, hogy ugyanakkora szinkronizációs idő változás esetén sokkal nagyobb mértékben romlott a TCP átvitel, mint a veszteségmentes esetben.

Összehasonlításként megmértük a pillanatnyi és a kumulált TCP átviteli sebességeket is 10 és 35 ms nagyságú szinkronizációs idő esetén, amiből a kapott eredmény a 4.2.5. ábrán látható.



4.2.5. ábra: Szinkronizációs idő hatása a TCP átvitelre csomagtovábbítás nélkül

Ha a szinkronizációs idő értékét 70 vagy 110 ms-ra állítottuk, akkor a 35 ms-os esethez teljesen hasonló eredményt kaptunk, tehát akkor is TCP timeout következett be.

A kumulatív TCP átviteli sebességet most is 50 másodperc hosszú letöltési idő alatt mértük meg, amelynek eredményét a 4.2.2. táblázat tartalmazza.

Szink. idő (ms)	TCP átviteli sebesség (Kbyte/s)
10	2983.35
35	2480.30

4.2.2. táblázat: Kumulált TCP átviteli sebességek 50 sec letöltési idő alatt

A kumulált átviteli sebesség is jóval nagyobb mértékben csökken, mint a veszteségmentes esetenél, ami szintén azt támasztja alá, hogy a szinkronizációs idő nagysága nagyobb jelentőséggel bír, mint az első esetben. A 4.2.1. és a 4.2.2. táblázatok alapján összehasonlítva a TCP átvitel sebességeket 10 és 35 ms-os szinkronizációs idő között, azt kapjuk, hogy *teljes csomagtovábbítás esetén csak 2,37 százalékkal*, míg *teljes csomagtovábbítás nélkül 16,86 százalékkal* csökkent az átviteli sebesség 10 ms-ról 35 ms-ra növelt szinkronizációs idő esetén.

Ezek alapján azt a következtetést vonhatjuk le, hogy ha nincs teljes csomagtovábbítás handover esetén, akkor a hálózat viselkedése sokkal érzékenyebb a szinkronizációs idő értékére, mint veszteségmentes hívásátadás esetén. Vagyis ha alkalmazzuk a teljes csomagtovábbítást a hálózatban, akkor nem csak az adatátviteli sebességet tudjuk javítani (ahogy a 4.1. alfejezetben tárgyaltuk), hanem annak az az előnye is megvan, hogy érzéketlenebb lesz a rendszer a szinkronizációs idő nagyságára, ami különösen fontos a mobil készülékekkel szemben támasztott igények szempontjából, hiszen ezt a paramétert a készülék működésének sebessége határozza meg. Ugyanakkor a valóságban is körülbelül a fent szimulált értékek közelében vannak a készülékek által meghatározott szinkronizációs idők (20 – 40 ms), így ekkor nem központi jelentőségű ennek az időnek a minimalizálása.

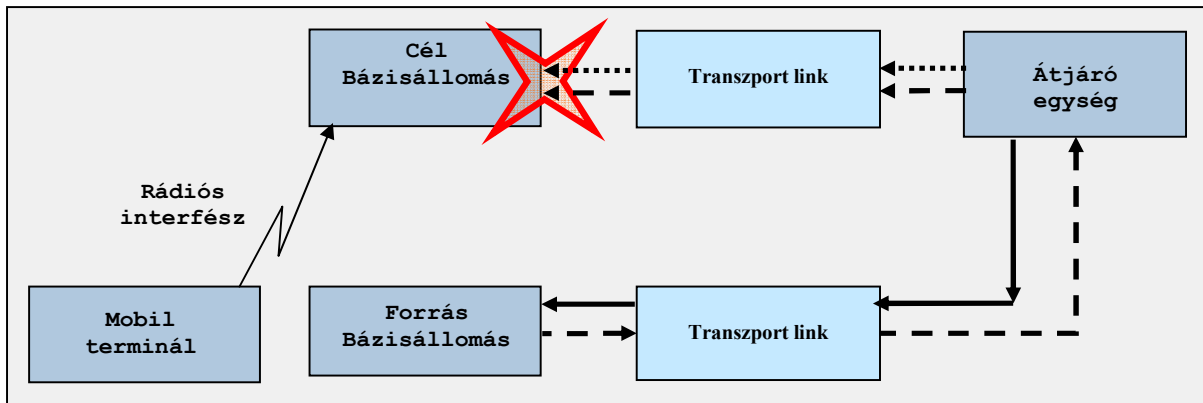
4.3. A csomagok felcserélődésének problémája a cél bázisállomásnál

A 4.1. alfejezetben már megemlítettük, hogy a hívásátadás során a csomagok forrás bázisállomásnál történő átirányítása nem várt mellékhatásokat okozhat a rendszer működésében. Most ezeket a problémás eseteket vizsgáljuk meg részletesen.

A csomagok felcserélődésének problémája a cél bázisállomásnál akkor jelentkezik, ha a forrás bázisállomás által, a handover parancs kiadása után a transzport hálózatra küldött csomagok keverednek az új útvonalon érkező, már az átjáró egységnél is a cél bázisállomás felé irányított csomagokkal. A forrás bázisállomás által a transzport hálózatra küldött csomagok a MAC rétegből visszahívott, illetve a cél bázisállomás felé átirányított transzport hálózatból érkező csomagok lehetnek. Ez a probléma kiemelt fontosságú, ha az átvitel minőségét szeretnénk maximalizálni, hiszen a csomagok keveredése könnyen a TCP torlódási ablak felezéséhez vezethet, tekintve, hogy a TCP már három egymást követő, rossz sorrendben érkező csomag esetén lefelez. Nem elég

tehát arról gondoskodni, hogy minden elküldött csomag megérkezzen a cél bázisállomáshoz, figyelniük kell ezek helyes sorrendiségére is.

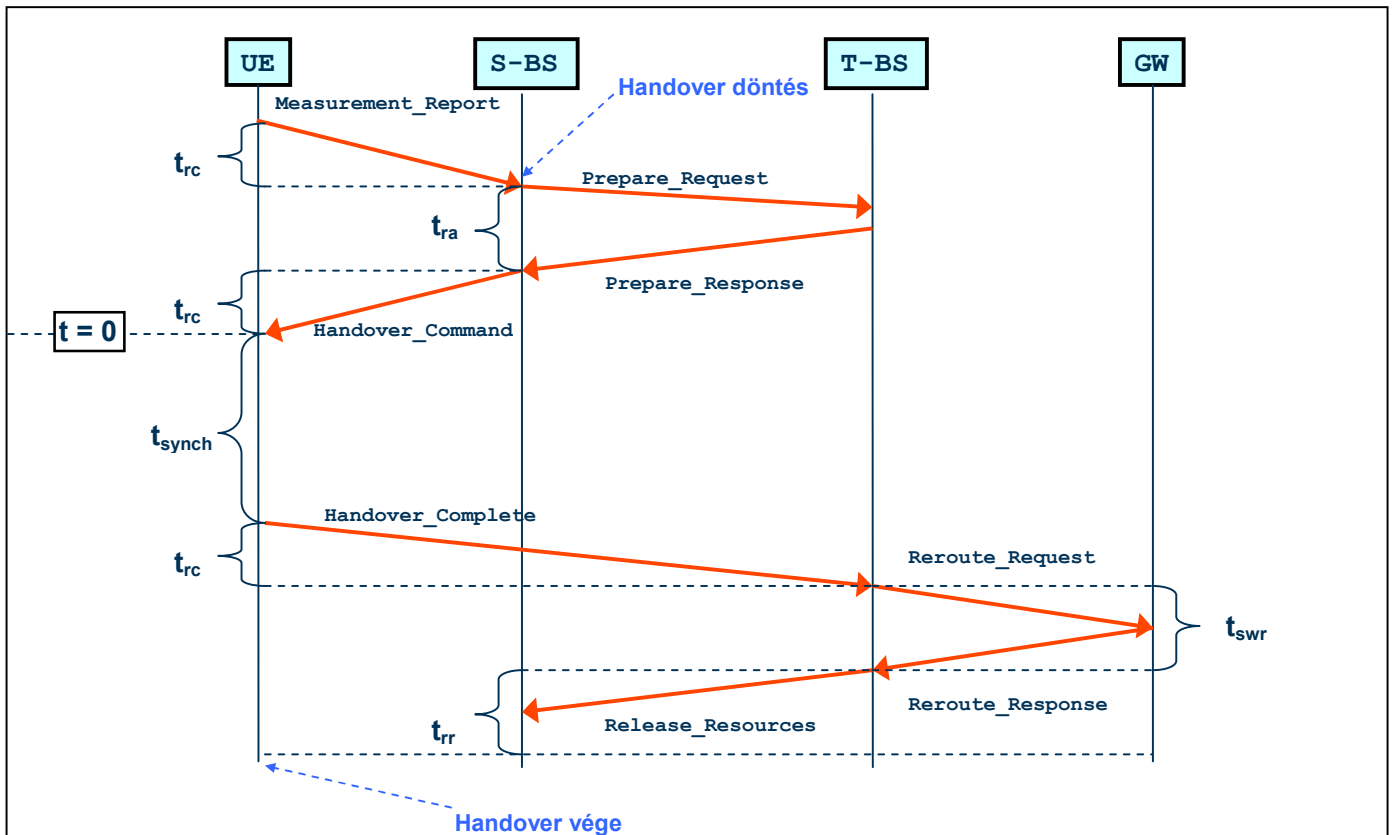
Ebben az alfejezetben azt vizsgáljuk meg, hogy a késleltetési paraméterek függvényében mikor következhet be a csomagok felcserélődése, illetve bemutatunk egy általunk megvalósított lehetséges közelítő megoldást is a problémára. A vizsgálatokat és az eredményeket szimuláció segítségével támasztjuk alá.



4.3.1. ábra Az csomagok keveredését szemléltető ábra

A 4.3.1. ábrán a szaggatott vonal szemlélteti a forrás bázisállomástól átirányított csomagok útját, míg a pontozott vonal jelzi a már az átjáró egységben is a cél bázisállomás felé irányított csomagok útvonalát. A két különböző úton érkező csomagok keveredhetnek egymással, így a sorrend felborulását okozhatják a cél bázisállomáshoz való érkezéskor. Az egyszerű vastag vonal mutatja a csomagok útját a forrás bázisállomás felé. Ezen az útvonalon érkező csomagok lesznek majd a cél bázisállomás felé továbbirányítva.

A hívásátadási modellben szereplő késleltetési paraméterek



4.3.2. ábra A hívásátadási modell késleltetési paramétereinek szemléltetése

A 4.3.2. ábrán látható jelzésüzenetek megfelelnek a 2. fejezetben használt, a hívásátadási eljárás folyamatát bemutató szekvencia diagramon használt üzenetekkel. A fenti 4.3.2. ábrán a modellben használt késleltetési paramétereket, és egymáshoz való viszonyukat, sorrendiségüket ábrázoltuk. Azt az időpontot, amikor a mobil terminál megkapja a handover parancsot, és lezárja a rádiós kapcsolatát a forrás bázisállomással, 0 időpontnak jelöltük, mert a számítások során ezt az időpontot referenciának véve könnyebben átlátható eredményeket kapunk. A modellben a következő késleltetési paramétereket használjuk:

t_{rc} : a rádiós kapcsolat késleltetési ideje (radio connection delay): modellezi, hogy a handover parancs jelzésüzenet (Handover_Command), valamint egy-egy felhasználói adategység mennyi idő alatt jut át a bázisállomás és a felhasználói egység közötti rádiós interfészen.

t_{ra} : erőforrás foglalási idő (resource allocation delay): az erőforrások lefoglalását kezdeményező kérés (Prepare_Request) kiadásától, az erre adott pozitív, vagy negatív válasz (Prepare_Response) megérkezéséig eltelt idő.

t_{synch} : szinkronizációs idő (synchronization delay): azt az időtartamot modellezi, ami a felhasználói egységnek szükséges ahhoz, hogy a rádiós interfészét az új bázisállomásra hangolja. Ez idő alatt nincs rádiós kapcsolat a felhasználói készülékkel.

t_{swr} : útvonal-átállítási idő (switch routing delay): az útvonalválasztás megváltoztatását kezdeményező kérés (Reroute_Request) kiadásától az átjáró egységbeli forgalomirányító tábla megváltoztatásáig eltelt idő.

t_{rr} : az erőforrások felszabadításához szükséges idő (release resources delay): a forrás bázisállomás lefoglalt erőforrásainak felszabadítását kezdeményező kérés (Release_Resources) kiadásától az erőforrások tényleges felszabadításáig eltelt idő.

További két paraméter, melyek a hálózat egyéb késleltetéseit írják le:

t_{tc} : a transzport hálózat egy-egy összekötő elemének egyirányú késleltetése, link késleltetés (transport channel delay). Az általunk vizsgált hálózatban ez a bázisállomások és az átjáró egység között lévő transzport hálózati összeköttetések egyirányú késleltetése.

t_I : a mobil gerinchálózatra kapcsolódó Internet hálózat késleltetése letöltési és feltöltési irányban (Internet delay). A szimulációk során a két érték megegyezik.

A számítások során egyszerűsítésként a transzport hálózatban a sorban állások okozta további véletlen késleltetéseket nem vettük figyelembe. Tekintve, hogy a szimulációkat ebben az alfejezetben is egyetlen rendszerben lévő felhasználóval végeztük, a sorban állási véletlen késleltetés itt valóban elhanyagolható volt.

A csomagok felcserélődésének részletes vizsgálata

A hívásátadási folyamatot tekintve első lépésben azt vizsgáljuk meg, hogy a késleltetési paraméterek mely értékeinél lehet olyan csomag, mely már az átjáró egységnél a cél bázisállomás felé irányítódva, a forrás bázisállomás által a cél bázisállomás felé átirányított csomagokkal keveredik. Megvizsgáljuk tehát, hogy mikor fordulhat elő a 4.3.1. ábrán szemléltetett csomagkeveredési jelenség.

A vizsgálatokat érdemes két részre bontani, mert a kapott eredmények így jobban kezelhetőek. Első esetben feltételezzük, hogy az első új útvonalon érkező csomag egy olyan nyugta csomag hatására generálódott, amely még a hívásátadás vége előtt ért el a kiszolgáló oldali egységhez, míg a második esetben feltesszük, hogy az első új útvonalon közlekedő csomag generálódását kiváltó nyugta a hívásátadás befejezése után ért csak el a kiszolgálóhoz. Az első esetben használt jelöléseket és a kapott eredményeket a második eset tárgyalásánál is felhasználjuk.

Csomagok keveredése a hívásátadási folyamat közben

TCP átvitelt feltételezve, legkedvezőbb esetben a handover parancs előtti pillanatban, egy a felhasználói készülék által küldött nyugta csomag még átjut a rádiós interfészen, és a transzport hálózaton, majd az Interneten haladva eljut a szerverhez. Erre a nyugtára a szerver az utolsó olyan csomagot generálja, amely a hívásátadás befejezése előtt a transzport hálózatra kerül. Ez a csomag a fent bevezetett jelöléseket használva a handover miatti rádiós kapcsolat lezárását követően $2 * t_I + t_{tc}$ múlva érkezik meg az átjáró egységhez. Ez az időtartam úgy adódik, hogy a nyugta csomag először egy transzport hálózatbeli fizikai csatorna késleltetését, majd az Internet hálózat késleltetését szenved

el, és ezen nyugta hatására generálódott csomag egy Internet hálózati késleltetéssel eljut az átjáró egységhez.

A számításoknál a könnyebb áttekinthetőség érdekében a hívásátadás miatti rádiós kapcsolatbontás időpontjához viszonyítunk, mint azt már korábban is jeleztük (ez volt a $t = 0$ időpont a 4.3.2. ábrán).

Ahhoz, hogy a csomagok felcserélődése lehetséges legyen a következő szükséges feltételnek kell teljesülnie:

$$I. 2 * t_I + t_{tc} > t_{synch} + t_{swr}$$

A jobb oldali kifejezés azt az időtartamot jelenti, ami a rádiókapcsolat lezárásától az átjáró egység átállításáig tart. Amennyiben ez az idő nagyobb lenne, mint az utolsó nyugtára generálódott, a rádiós kapcsolat bontását követően az átjáró egységhez érkező csomag ideje, úgy a hívásátadási folyamat során (tehát az első esetben) a sorrend felcserélődési probléma nem jelentkezik, hiszen ekkor újonnan érkező csomagok hiányában az új bázisállomáshoz csak átirányított adatok érkehetnek.

A következőkben a kérdés tehát az, hogy amennyiben elképzelhető az átirányított és az új útvonalon érkező csomagok keveredése, úgy ez milyen késleltetési paraméter értékeknél fordulhat elő. A kérdés megválaszolásához azt kell megvizsgálni, hogy mikor érkezik a cél bázisállomáshoz az utolsó, a forrás bázisállomástól átirányított csomag, illetve, hogy mikor érkezik az első új útvonalon közlekedő csomag. Amennyiben az első új útvonalon érkező csomag később érkezik, mint az utolsó átirányított csomag, úgy a csomagok sorrendje nem változik, egyébként biztosan keveredés történik.

A forrás bázisállomástól átirányított utolsó csomag megérkezési ideje a cél bázisállomáshoz

$$\min\{f = 2 * t_I + 4 * t_{tc}; g = t_{synch} + t_{swr} + 3 * t_{tc}\}$$

Az f függvény azt az időt fejezi ki, hogy a handover vége előtti *utolsó, szerver által generált csomag* még a forrás bázisállomás felé irányítódik az átjáró egységben ($2 * t_I + t_{tc}$ idő után), majd három fizikai átviteli közeg késleltetését elszenvedve ($3 * t_{tc}$) eléri a cél bázisállomáshoz. Ha a 4.3.1. ábrára tekintünk, akkor könnyen nyomon követhetjük a csomagok útvonalát. Először az átjárótól a forrás bázisállomáshoz ér el a csomag (t_{tc}), az egyszerű vastag vonallal jelzett útvonalon, majd az átirányítás során, a szaggatott vonallal jelzett útvonalon először visszajut az átjáróhoz (t_{tc}), majd innen eljut a cél bázisállomáshoz (t_{tc}). A g függvény azt az időt mutatja, hogy *egy tetszőleges csomag*, amely az átjáró átkapcsolása előtt még a forrás bázisállomás felé irányítódott, mennyi idő alatt ér el a cél bázisállomáshoz. Ekkor ugyanis $t_{synch} + t_{swr}$ idő után még három transzport hálózati összeköttetés késleltetését szenved el az átirányítás során (ismét a GW -> sBS -> GW -> tBS útvonalon haladva). Megmutatható, hogy a csomagok keveredése csak akkor fordul elő, ha $g < f$, hiszen egyébként a $2 * t_I + t_{tc} < t_{synch} + t_{swr}$ kifejezés teljesülne, ami ellentmondana az I. szükséges feltételnek.

$$II. f = 2 * t_I + 4 * t_{tc} < g = t_{synch} + t_{swr} + 3 * t_{tc} \text{ esete}$$

Ahonnán: $2 * t_I + t_{tc} < t_{synch} + t_{swr}$ ellentmond I.-nek

Az első új útvonalon érkező csomag megérkezési ideje a cél bázisállomáshoz

$$h = t_{synch} + t_{swr} + t_{tc}$$

A h függvény azt fejezi ki, hogy az útvonal irányítás megváltoztatása után ($t_{synch} + t_{swr}$) az új útvonalon közlekedő csomagnak már csak az átjáró egység és a cél bázisállomás közti transzport hálózati összeköttetés késleltetését modellező t_{tc} időre van szüksége, hogy eljusson a cél bázisállomáshoz.

Végkövetkeztetésként azt kapjuk, hogy mivel g minden esetben nagyobb, mint h ($t_{synch} + t_{swr} + 3 * t_{tc} > t_{synch} + t_{swr} + t_{tc}$), ezért a csomagok a hívásátadási folyamat során akkor és csak akkor keverednek, ha

$$I. 2 * t_I + t_{tc} > t_{synch} + t_{swr} \text{ áll fenn.}$$

Ekkor tehát az $I.$ kifejezés a csomagok keveredésének szükséges és elégséges feltétele.

Ha becslést szeretnénk kapni a felcserélődött csomagok számára, akkor először meg kell határozni azt az időintervallumot, amelyben csomagkeveredés lehetséges, majd ezt az időtartamot beszorozva a cél bázisállomáshoz érkező transzport hálózati linkek átlagos sebességével jó becslést kapunk az érintett csomagok számára. A felcserélődési időintervallum könnyen meghatározható, hiszen ez éppen a fent bevezetett g és h függvény különbsége lesz, hiszen ha a sorrend szükségképpen felborul, akkor pont az első új útvonalon érkező csomag (g), és az utolsó átirányított csomag (h) cél bázisállomáshoz való érkezési idejének különbsége alatt borulhat fel a csomagok sorrendje. A rossz sorrendben érkező csomagok száma tehát arányos

$$(t_{synch} + t_{swr} + 3 * t_{tc}) - (t_{synch} + t_{swr} + t_{tc}) = 2 * t_{tc} \text{ -vel.}$$

A cél bázisállomáshoz érkező forgalom sebességéről legegyszerűbben mérési úton tájékozódhatunk. Az általunk vizsgált esetben, ez a sebesség átlagosan 2500 kByte volt másodpercenként, amihez, ha hozzávesszük a későbbiekben meghatározásra kerülő $t_{tc} = 5$ ms-os értéket, úgy a két adat szorzatából, és az IP csomag méretének (1500 bájtt) figyelembevételével kijön, hogy körülbelül 17 csomagot érint a sorrendezés problémája.

Csomagok keveredése a hívásátadás befejezése után

Csomagsorrend felcserélődés ebben az esetben csak úgy fordulhatna elő, ha a hívásátadási folyamat befejeztével az első, a felhasználói egység által küldött nyugtára érkező, szerver által generált csomag megelőzi az utolsó, a forrás által átirányított csomagot. Erre az esetre a következő egyenlőtlenséget lehet felírni a már bevezetett jelölések felhasználásával:

$$g = t_{synch} + t_{swr} + 3 * t_{tc} > r = t_{synch} + t_{rc} + 2 * t_{tc} + 2 * t_I$$

A fenti kifejezés bal oldala a már említett g függvény, az utolsó átirányított csomag megérkezési ideje a cél bázisállomáshoz. A kifejezés jobb oldala, az r függvény azt fejezi ki, hogy a rádiós kapcsolat felélése után az első felhasználói egység által küldött nyugta először elérkezik a cél bázisállomáshoz (t_{rc}), majd átmegy a transzport hálózaton

(t_{tc}), eljut a távoli interneten lévő szerverig (t_I), a szerver csomagot generál, és ez a csomag eljut a cél bázisállomáshoz ($t_I + t_{tc}$).

Ekkor a következő szükséges és elégséges feltételt kapjuk a csomagok keveredésére a második esetben:

$$III. t_{swr} + t_{tc} > t_{rc} + 2 * t_I ,$$

ami a paraméterek valóságbeli értékeinél nem áll fenn, hiszen az Internet késleltetés sokkal nagyobb lehet, mint az átjáró átkapcsolási és a transzport hálózati link késleltetési ideje.

Egy általunk megvalósított megoldási javaslat a sorrendezés problémájára – prioritásos ütemező használata a cél bázisállomásnál

A csomagok felcserélődése könnyen a TCP torlódási ablak felezését okozhatja, ami jelentősen csökkenti az átvitel minőségét, ezért törekednünk kell ennek elkerülésére. Az általunk kiválasztott és megvalósított megoldási javaslat két pufferrel dolgozó prioritásos ütemezést használ a cél bázisállomásnál. Az egyik pufferbe a forrás bázisállomástól átirányított, míg a másikba az új útvonalon érkező csomagok kerülnek. A prioritásos kiszolgálás lényege, hogy amíg van csomag az átirányított adatokat tartalmazó pufferben, addig a MAC réteg innen kap adatokat. Ezt a puffert magasabb prioritással kezelni azért előnyös, mert amíg ez a sor ki nem ürül, addig biztosan nem történik keveredés. Amennyiben minden átirányított csomag úgy érkezik meg, hogy ez a puffer közben egyszer sem ürül ki, úgy a helyes sorrendezés garantált.

Ez a módszer sajnos nem ad tökéletes megoldást a problémára, hiszen ha a hálózati torlódások miatt az átirányított csomagokat tartalmazó puffer kiürül, miközben ilyen csomagok még vannak a transzport hálózatban, illetve az új útvonalon érkező csomagokat tartalmazó puffer nem üres, akkor a csomagok sorrendje felborul, hiszen ekkor ebből a pufferből történik a kiszolgálás. Elmondható azonban, hogy az esetek többségében, mint azt a szimulációs eredményeknél is megmutatjuk, ez a javaslat egész jó megoldást ad a problémára, a cél bázisállomás oldali prioritásos ütemezés használata az esetek többségében növeli az átvitel teljesítményét.

A szimulációs eredmények vizsgálata sorrendezést használó és sorrendezés nélküli esetben

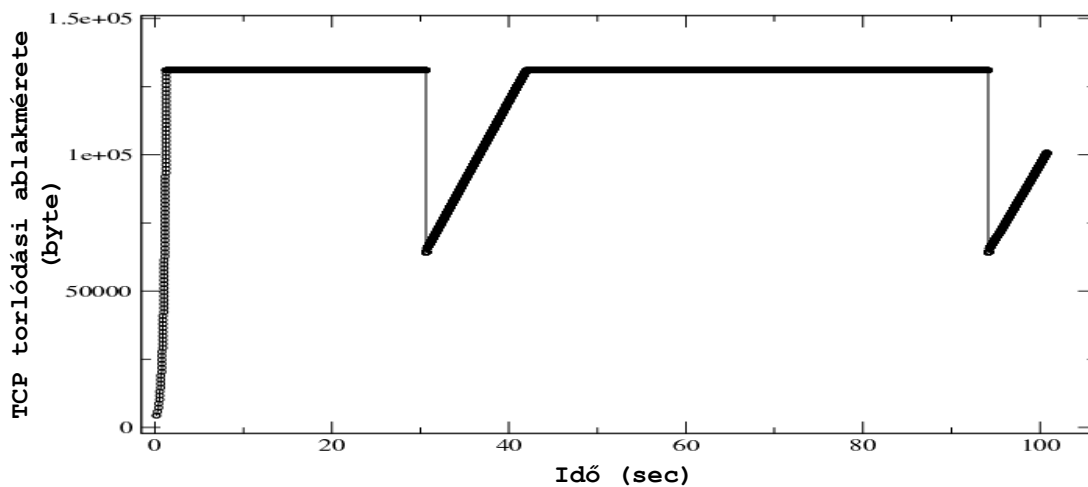
A vizsgált rendszerben egy 10 m/s sebességgel mozgó felhasználó tölt le egy 100 MB-os fájlt a távoli webszerverről. A torlódási ablak mérete mind adó, mind vevő oldalon 128 kB-ot.

Az időzítési paraméterek a következők:

- $t_{synch} = 15$ ms
- $t_{swr} = 5$ ms
- $t_{rc} = 5$ ms
- $t_{tc} = 5$ ms
- $t_I = 50$ ms

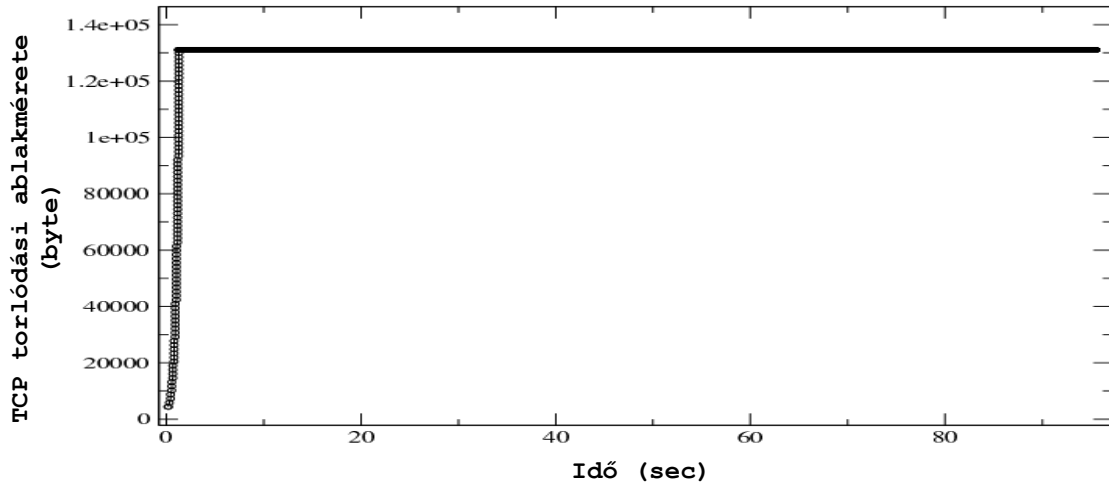
Hívásátadások mindkét esetben a következő időpontokban vannak:

- 30,5 másodperc : 1. handover
- 31,0 másodperc : 2. handover
- 32,5 másodperc : 3. handover
- 33,5 másodperc : 4. handover
- 55,75 másodperc : 5. handover
- 94,0 másodperc : 6. handover



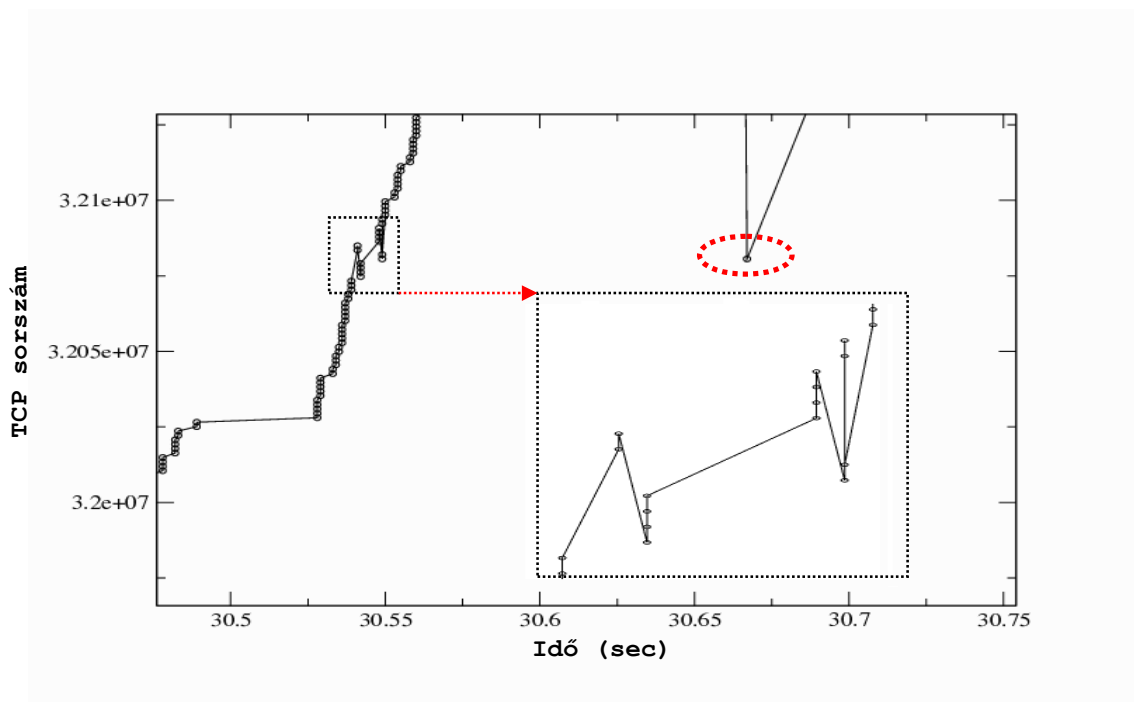
4.3.3. ábra A TCP torlódási ablak alakulása sorrendezés nélküli esetben

A 4.3.3. ábráról látszik, hogy két esetben is feleződött a TCP torlódási ablak mérete (a 30,5-ik és a 94-ik másodpercben). Az előző alfejezetekben bemutatott, átírányítást és csomagvisszahívást is használó szimulációk esetén nem tapasztaltuk a TCP torlódási ablak feleződését, ami azért volt, mert ezekben az esetekben az Internet késleltetést úgy választottuk meg, hogy a csomagok keveredését meghatározó szükséges és elégséges feltétel ($2 * t_I + t_{tc} > t_{synch} + t_{swr}$) ne teljesüljön. Most azonban a késleltetési paramétereket behelyettesítve látható ($2 * 50 + 5 > 15 + 5$), hogy a csomagok keveredésével számolni kell. A sorrendezést nem használó esetben tehát a szükségképpen bekövetkező csomagkeveredés a TCP torlódási ablak feleződését okozza.



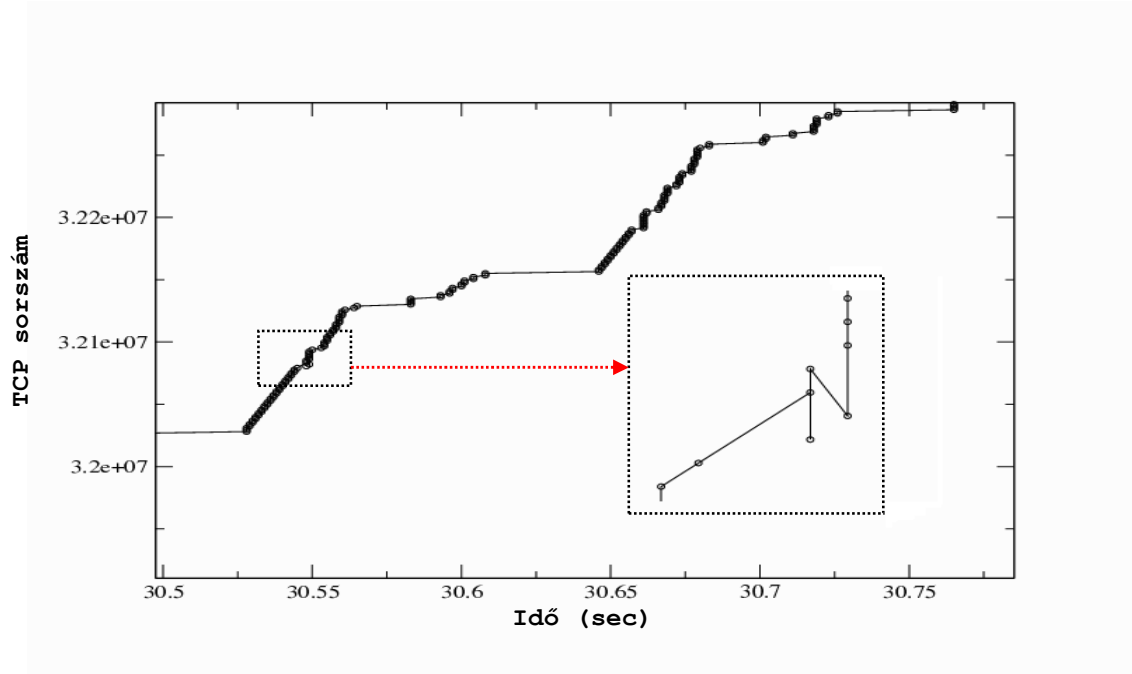
4.3.4. ábra Torlódási ablak alakulása sorrendezést használó esetben

A 4.3.4. ábra azt mutatja, hogy a sorrendezést használó esetben nem történt lefelezés, a TCP torlódási ablak mérete maximális maradt a hívásátadások során is.



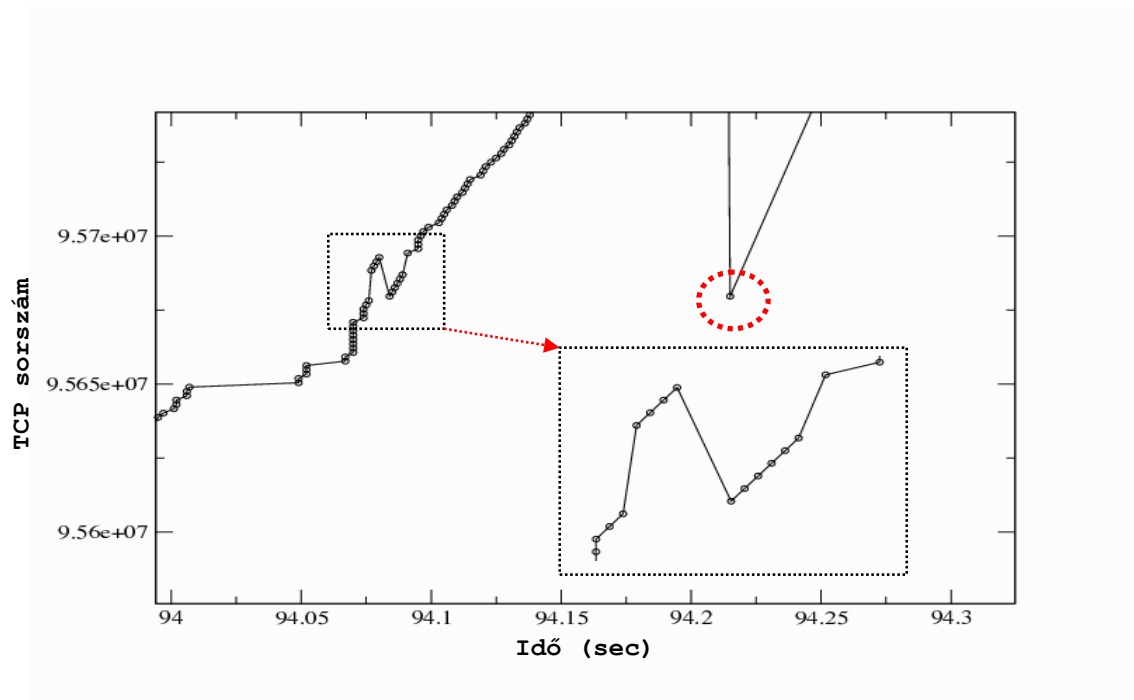
4.3.5. ábra A TCP szintű átvitel alakulása a 30,5-ik másodperc környékén sorrendezés nélküli esetben

A 4.3.5. ábráról látszik, hogy a csomagok keveredése okozta a torlódási ablak felezését. Kettőnél több rossz sorrendben érkező csomag újraküldést generál. Az egyetlen újraküldött csomagot bejelöltük az ábrán, a keveredés környezetét pedig kinagyítottuk.



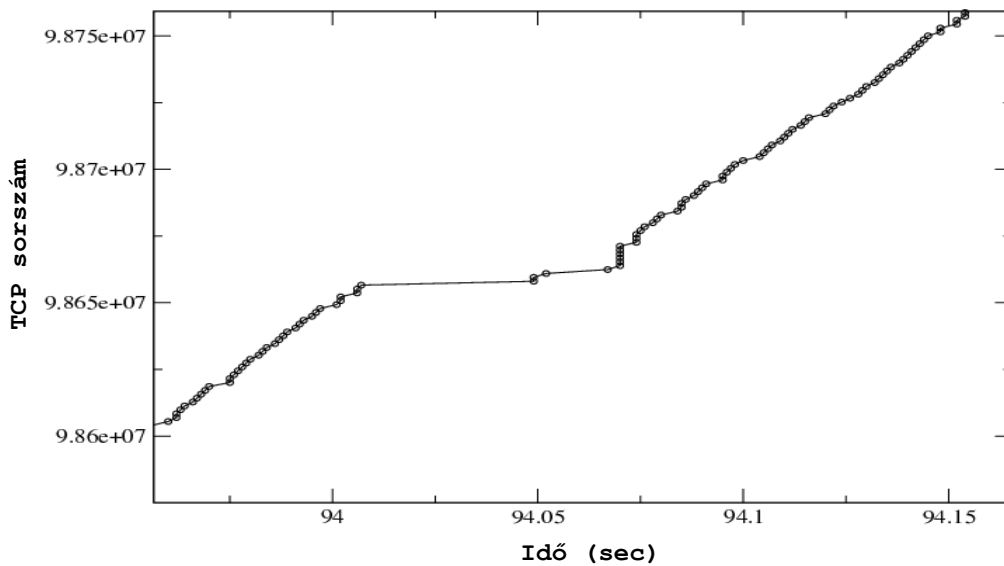
4.3.6. ábra A TCP szintű átvitel alakulása a 30,5-ik másodperc környékén sorrendezést használó esetben

A 4.3.6. ábra azt mutatja, hogy ugyan a csomagok sorrendje nem tökéletes, de nincs kettőnél több, egymás után rossz sorrendben érkező csomag, így a TCP torlódási ablak nem feleződik le.



4.3.7. ábra A TCP szintű átvitel alakulása a 94-ik másodperc környékén sorrendezés nélküli esetben

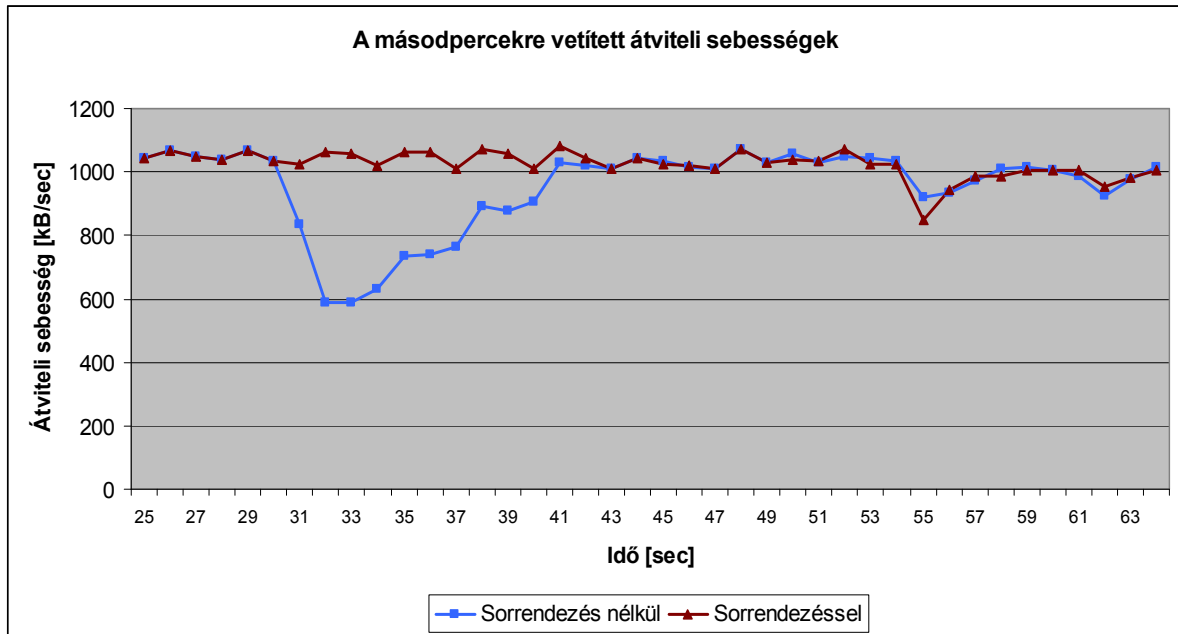
A 4.3.7. ábráról látszik, hogy a több mint kettő, egymás után rossz sorrendben érkező csomag TCP szintű újraküldést generál, így a torlódási ablak ismét a felére csökken.



4.3.8. ábra A TCP szintű átvitel alakulása a 94-ik másodperc környékén sorrendezést használó esetben

A 4.3.8. ábrán az látszik, hogy a sorrendezést használó esetben a csomagok sorrendje teljesen helyreáll.

Az átviteli sebességek alakulása a két esetben, értékelés



4.3.8. ábra Másodpercekre vetített átviteli sebességek alakulása a két esetben

A 4.3.8. ábráról jól látszik, hogy a sorrendezést használó esetben a másodpercekre vetített átviteli sebességek magasabbak a hívásátadási folyamatok környékén. A szimulációs eredmények tehát alátámasztották várakozásainkat. Amennyiben a hívásátadási eljárás tervezése során azt a döntést hozzuk, hogy a forrás bázisállomás legyen képes a hozzá érkező csomagok átirányítására, úgy a cél bázisállomást fel kell készíteni az ott esetlegesen kialakuló csomag felcserélődés megszüntetésére. Ellenkező esetben a csomagok átirányítása nem hozza meg a várt eredményeket, az átvitel minősége nem javul terveinknek megfelelően.

4.4. A MAC réteg és a mobilitás kezelés közti együttműködés kérdései

A 2. fejezetben már említettük, hogy hívásátadás során bázisállomás váltásakor új MAC kapcsolat jön létre a cél bázisállomással. Az előző MAC réteg aktuális csomagküldési állapota elvész. Ez azt jelenti, hogy elvész az információ, hogy egy küldésben lévő MAC SDU mely darabjai lettek már átküldve a vevő oldalra, ezért érdemes lehet azt megvizsgálni, hogy pontosan milyen problémák keletkezhetnek ebből. Olyan kérdések megválaszolásával foglalkozunk ebben az alfejezetben, hogy miért és milyen körülmények között fordulhat elő az, hogy a hívásátadás miatt egy csomag

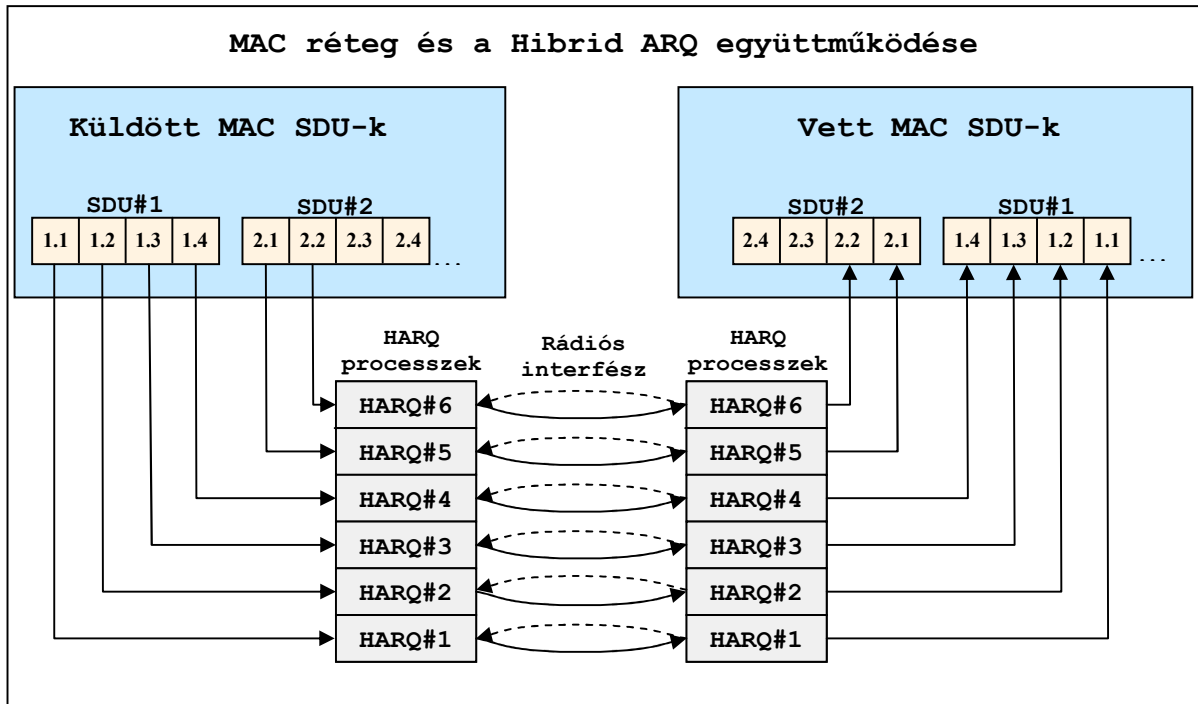
elveszik, vagy éppen duplikáltan érkeznek meg a vevőhöz. Ahhoz, hogy egzakt választ kapjunk ezekre a kérdésekre, részletesen kell vizsgálni a MAC réteget működését.

A MAC működés részletes ismertetése

Ahogy már említettük a 3. fejezetben, a MAC réteg legfőbb feladatai közé tartozik a szegmentálás és a sorrendhelyes továbbítás. Ezenkívül a rádiós linken történő hibázásokat javító mechanizmust is tartalmaz, mert a különböző rádiós terjedési viszonyok miatt a mobil hálózatoknál a legtöbb csomagvesztés, illetve meghibásodás általában a rádiós interfészen történik. Ezért ilyenkor egy plusz hibajavító mechanizmust érdemes alkalmazni annak érdekében, hogy a csomagvesztés hatása ne érjen el felsőbb réteget. Ugyanis például TCP protokoll használata esetén lecsökken az adattovábbítás sebessége, mivel a TCP ezt a veszteséget hálózati torlódásnak érzékeli és ezért az adási ablakának méretét lecsökkenti, pedig valójában nem torlódás okozta a csomagvesztést.

Ilyen hibajavító mechanizmus a Hibrid ARQ (HARQ), amely kijavítja a rádiós interfészen keletkezett hibák nagy részét úgy, hogy az adó oldal újraküldést kezdeményez, ha a vevő negatív nyugtát küld vissza egy adott PDU szegmensre. Ez láthatóan azokat a hibázásokat nem tudja kijavítani, amelyeknél a rádiós közegbeli zavarok miatt a vevő tévesen érzékeli pozitívnak egy negatív nyugtát. Így ezt az adategységet nem küldi újra, mert úgy tudja, hogy az megérkezett a vevőhöz, de valójában mégsem. Az ilyen hibák kijavítására egy második szintű ARQ protokollt alkalmaznak.

A HARQ mechanizmust több, úgynevezett HARQ processz valósítja meg, amelyek mindegyike időben egymáshoz képest elcsúsztatva egy-egy adategységet továbbítanak. A processzek megáll-és-vár (stop-and-wait) jelleg szerint működnek, azaz miután elküldtek egy adategységet, nyugtára várnak. Több processz alkalmazása esetén így jobban kihasználható a rádiós link. A MAC réteg és a HARQ processzek közötti együttműködést szemlélteti a következő ábra, melyen a rövidítések SDU (Service Data Unit), az adategység, amelyet a MAC réteg a felsőbb rétegtől kap. Valamint PDU-nak (Protocol Data Unit) nevezzük azt az adategységet, amelyet a MAC réteg hoz létre és ad tovább az alsóbb rétegnek.



4.4.1. ábra: A MAC réteg és a Hibrid ARQ együttműködése

Ha egy processz hibásnak talál egy adategységet a vételi oldalon, akkor azt eldobja, és negatív nyugtát küld az adó oldali processznek. Tehát könnyen előfordulhat az, hogy a vevő oldalon a hibátlanul megkapott PDU-k nem egymást követő darabjai az SDU-nak, azaz a vételnél a sorrend felcserélődhet. Ha a feldarabolt SDU utolsó darabja ugyan beérkezett már a vevőhöz, de van még hiányzó rész, akkor nem lehet az SDU-t felsőbb rétegnek feladni, csak akkor továbbítható, ha az SDU mindegyik darabja megérkezett a vevőhöz.

Hívásátadás hatása a MAC réteg adattovábbítási folyamatára

A 3. fejezetben már ismertettük, hogy a szimulált hálózatban a hívásátadási eljárást megvalósító implementációnk handover esetén a forrás bázisállomáshoz tartozó MAC réteg pufferének tartalmát a transzport hálózaton a cél bázisállomás felé továbbítja a veszteségmentes hívásátadás elérése érdekében. Ezt a funkciót nevezzük csomagvisszahívásnak. Ehhez a MAC réteg pufferét fel kell készíteni arra, hogy képes legyen egy adott pillanatban a megfelelő MAC SDU-kat visszaadni, amik a pufferben vannak, de még nem lettek átküldve a vevő oldalnak. Ehhez az adó oldalnak minden pillanatban a vevő oldallal egyező képet kell látnia a pufferben lévő csomagok állapotáról.

Ha megvizsgáljuk, hogy milyen lépései vannak a MAC küldési folyamatának, akkor láthatjuk a pufferben lévő SDU-k, illetve az éppen küldésben lévő SDU-khoz tartozó PDU-k megfelelő visszahívása nem triviális feladat. Először a MAC kivesz egy SDU

adategységet a várakozási sorából, amelyet szegmentál, annak függvényében, hogy az alsóbb réteg mekkora méretű PDU-kat kér. A feldarabolás során az egyes PDU-k sorszámmal (sequence number) lesznek ellátva, amely egy csúszóablakban (sliding window) körbefordul. A sorszám azért szükséges, hogy a vevő a sorrendhelyesen vissza tudja állítani az SDU-kat. Ezek után MAC PDU fejléccel ellátva továbbküldi az adategységet az alsóbb rétegnek.

Ha e küldési folyamat közben bekövetkezik egy hívásátadás, akkor előfordulhat, hogy az éppen szegmentált SDU darabjai közül a rádiós interfészen történt hibázás miatt nem mind érkezett meg helyesen a vevőhöz. Mivel hívásátadás során *új MAC kapcsolat keletkezik* az új bázisállomással, a régi kapcsolat alatt átjutott MAC PDU-k elvesznek, így ha akár csak 1 része is hiányzott az SDU-nak, újra kell azt küldeni.

Azonban a csomagvesztés mellett a csomagok duplikálódását is figyelemmel kell kísérni. A következőkben a csomagduplikálódás előfordulási lehetőségét fogjuk részletesen megvizsgálni. Ehhez azonban először ismertetjük a MAC réteg kétféle csomagvisszahívási mechanizmusát, a *kumulatív és szelektív működést*.

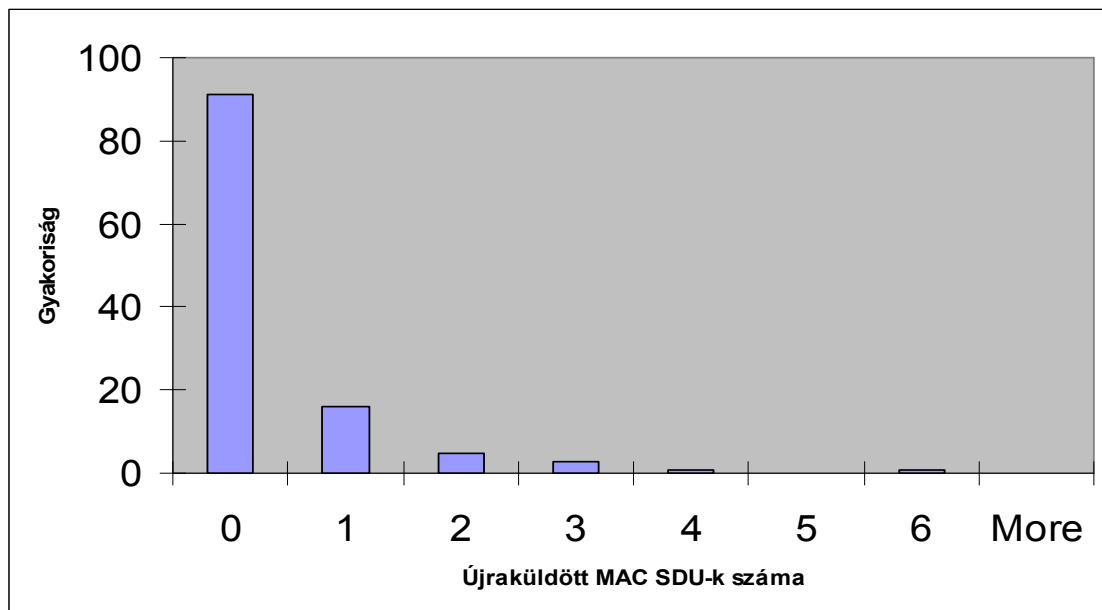
A 4.4.2. ábra egy lehetséges MAC állapotot szemléltet. Ezen jól látható, hogy a vételi oldalon a küldés alatt álló MAC SDU-k mely részei érkeztek meg helyesen a vevőhöz. A vevőtől érkező nyugták miatt az adó oldal is ugyanezt az állapotot látja. Az ábra alapján látszik, hogy az SDU#2-vel jelölt SDU minden darabja megérkezett már a vevőhöz. A *kumulatív működési mód* esetén a vevő oldal nem adja fel a felsőbb rétegnek ezt az SDU-t, mert az előtte lévő SDU#1-nek még nem minden darabja érkezett meg, és a sorrendhelyesség ezért felborulna, ha előbb továbbítaná az SDU#2 adategységet, mint az SDU#1-t. Ha ilyenkor következik be hívásátadás, akkor az adó oldal a csomagvisszahívás során az SDU#2-t is vissza fogja adni a hálózati rétegnek, mert az el fog veszni a vevő oldalon, mivel új MAC kapcsolat keletkezik a cél bázisállomással. A sikeresen átküldött SDU#2 tehát *elvész*, ezért azt ismételten át kell majd küldeni a rádiós interfészen. Ez a megoldás *biztosítja a sorrendhelyességet* a felsőbb réteg számára, de a *rádiós közeg kapacitását pazarolja* abban az értelemben, hogy a korábban sikeresen átküldött adatot még egyszer átküldi a rádiós linken.



4.4.2. ábra: Az adási, illetve a vételi oldalon a küldés alatt álló MAC SDU-k egy lehetséges állapota

Szelektív esetben egy handover bekövetkeztekor ugyanilyen MAC állapotnál az adó oldal a csomagvisszahívás során csak a nem teljesen vett SDU-kat (SDU#1, SDU#3) fogja visszaadni a hálózati rétegnek, tehát a sikeresen átküldött SDU#2 nem kerül még egyszer továbbításra. A vevő oldal pedig ez esetben a sikeresen vett SDU#2-t továbbadja a felsőbb rétegnek, annak ellenére, hogy az előtte lévő SDU#1 még nem érkezett át teljesen, felborítva ezzel a csomagok sorrendjét. Az előnye ennek tehát az, hogy az adónak már csak azokat az SDU-kat kell átküldenie a rádiós interfészen, amelyeknek nem minden része érkezett meg a vevőhöz. A *szelektív működés* tehát *nem biztosítja sorrendhelyességet*, viszont a *rádiós link kapacitását nem pazarolja*, azaz nem küld át még egyszer egy SDU-t, ha az korábban már sikeresen megérkezett a vevőhöz. Megvalósítás szempontjából a *szelektív megoldás bonyolultabb*, mert nyilván kell tartani, hogy pontosan mely SDU-k érkeztek meg teljesen a vevőhöz, hiszen ezeket nem kell újra elküldeni, míg kumulatív esetben az összes küldésben lévő SDU-t újra kell küldeni.

A szimulált hálózat MAC rétegében a *kumulatív módot valósítottuk meg*, tehát a MAC réteg működése miatt nem állhat elő csomagfelcserélődés hívásátadáskor, viszont a rádiós linken történő duplikált adatküldés előfordulhat. Ne feledjük el, hogy ez a duplikált küldés felsőbb rétegek számára nem látható, hiszen a vevő oldal a handover előtt egyszer már elküldött adategységet nem adta fel a felsőbb rétegnek, hanem az elveszett az új MAC kapcsolat létrejöttékor. A következő hisztogram mutatja, hogy hány hívásátadáskor történt felesleges küldés a rádiós interfészen és így hány csomagot kellett még egyszer átküldeni.

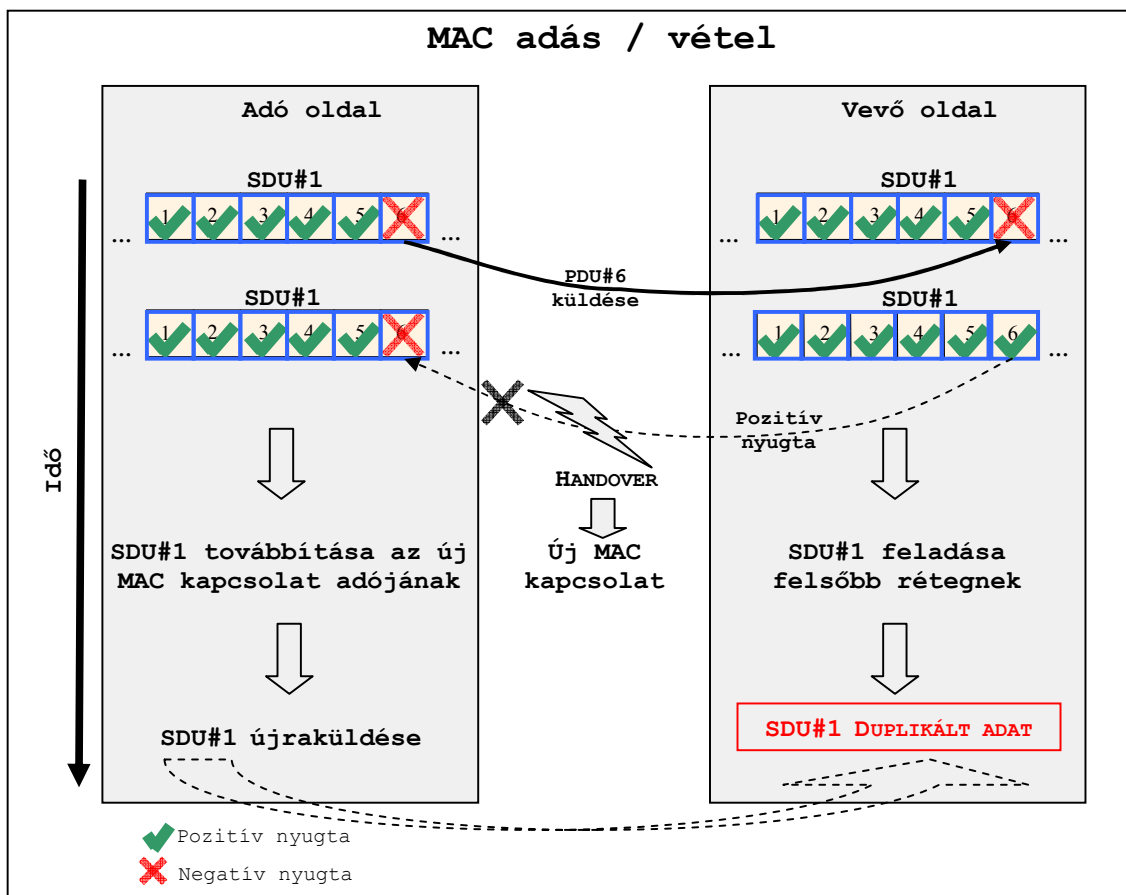


4.4.3. ábra: Kumulatív MAC működésből eredő duplikált csomagküldések száma

Látható a 4.4.3. ábra alapján, hogy nem pazaroltuk jelentősen a rádiós link kapacitását, mert a csomagok duplikált elküldésének száma elhanyagolható a hívásátadások számához képest. A kumulatív mód választása azért is kézenfekvőbb volt, mert általában véve a

küldés alatt álló MAC SDU-k mérete nagy, így csak kevés van egyszerre küldésben és viszonylag kicsi PDU keretekre daraboljuk őket, azaz annak a valószínűsége, hogy a bekövetkezett hívásátadáskor egy sikeresen vett SDU előtt még van ne teljesen vett SDU a vevőnél, meglehetősen kicsi, amit közvetve a kapott hisztogram is igazol.

Ha tovább vizsgáljuk a hívásátadás miatti lehetséges helyzeteket, akkor előfordulhat olyan eset, hogy a MAC réteg duplikáltan továbbít egy csomagot a felsőbb réteg felé. Tekintsük azt az állapotot, ahol egy küldésben lévő SDU utolsó PDU-ján kívül az összes rész sikeresen megérkezett a vevőhöz. A következő időpillanatban az utolsó rész is sikeresen megérkezik, az adó erről *pozitív nyugtát küld* a vevőnek, azonban ez már *nem ér vissza* a vevőhöz, mert hívásátadás következik be és a pufferben ragadt csomagok visszahívása után a régi MAC kapcsolat megszűnik. Ez azt eredményezi, hogy a vevő oldal feladja a sikeresen megérkezett SDU-t felsőbb rétegnek. A forrás bázisállomás viszont úgy látja, hogy nem sikerült teljesen átküldeni ezt az SDU-t, mert az utolsó PDU-járól nem kapott pozitív nyugtát. Ezért ezt az SDU-t is visszaadja csomagvisszahíváskor, hogy a cél bázisállomáshoz eljusson a csomagtovábbítás során, aki majd átküldi a vevőnek. A vevő oldal pedig sikeres vétel után újból feladja a felsőbb rétegnek, mert nem rendelkezik olyan információval amiből kiderülne számára, hogy ezt az adataegységet egyszer már felsőbb rétegnek továbbadta. Ezt a lehetséges helyzetet szemlélteti a 4.4.4. ábra.



4.4.4. ábra: Csomagduplikálást eredményező MAC küldési folyamat

Ahogy az ábrán is látszik az SDU utolsó darabja is sikeresen átért a vevőhöz, aki erről pozitív nyugtát küldött, amit a küldő már nem kapott meg, mert hívásátadás következett be. Emiatt ezt az adategységet újból meg fogja kapni a cél bázisállomás MAC rétegétől, és ismételten feladja felsőbb rétegnek.

A probléma, hogy mást lát az adó és vevő oldal, nem függ attól, hogy milyen csomagvisszahívási mechanizmust használunk a MAC rétegben. Azonban a *duplikált csomagok száma* már *függ a működési módtól*, mert ha kumulatív működésnél fordul elő ez az eset, akkor az összes küldésben lévő SDU-t újra fogja küldeni az adó oldal. Ez akkor jelent több mint egy csomagújraküldést, ha a sikeresen megkapott SDU előtt még van olyan SDU, amelynek hiányoznak részei. Ilyenkor tehát több csomag duplikálódása is előfordulhat, míg szelektív esetben csak az az SDU jelenhet meg duplikáltan, amelyet a vevő sikeresen vett, de az erre küldött pozitív nyugtát már nem kapta meg az adó.

4.5. Transzport hálózati link túlsordulása

A vizsgált hálózatban működő elosztott hívásátadási eljárás cellaváltáskor a forrás bázisállomás pufferében lévő csomagokat képes visszahívni és továbbítani azokat a cél bázisállomás felé. Amikor ez megtörténik a pufferben lévő csomagok egyszerre kerülnek rá a transzport hálózatra. Ahogy a 3. fejezetben már említettük, a szimulált hálózat felépítése olyan, hogy minden bázisállomást egy felfele és egy lefele irányuló csatorna köt össze az átjáró egységgel. Mivel a továbbított csomagok az átjáró egységen keresztül haladnak a cél bázisállomás felé, ezért a forrás bázisállomáshoz tartozó felfele-irányú fizikai csatorna pufferéhez érkeznek először. A fő célunk tehát az, hogy megvizsgáljuk, mennyire terheli ez a csomagtovábbítás egy adott bázisállomás felfele-irányú csatornájának pufferét. Az előző szimulációk során a transzport hálózat működése nem játszott fontos szerepet, így azokban az esetekben nem korlátoztuk a pufferek méretét, ezért akkor nem kellett ezzel a kérdéssel foglalkozni.

Mivel a felhasználók véletlenszerűen mozognak, ezért nem tudható előre, hogy egy adott időpillanatban mely felhasználók fognak hívásátadást kezdeményezni egy adott bázisállomásról. Ha viszont hívásátadáskor egy bázisállomás felfele-irányú csatornájának puffere túlsordul, akkor azt biztosan azon felhasználók MAC pufferében lévő csomagok okozzák, akik ugyanabban az időben, ugyanarról a forrás bázisállomásról kezdeményeznek hívásátadást, hiszen az összes ilyen puffereelt csomag a forrás bázisállomás felfele-irányú csatornájának pufferébe kerül.

Ha a hálózatban csak letöltő felhasználókat szimuláltunk, akkor a felfele-irányú csatorna pufferében hívásátadáson kívül csak nyugták váraakoznak, amelyek mérete az 1500 bytes-os IP adatcsomagok méretéhez képest elhanyagolható, tehát ha a puffer méretét vizsgáljuk az idő függvényében, akkor a hívásátadás során a pufferbe került csomagok mennyiségét látjuk. Az a döntés, hogy csak letöltő felhasználókat szimulálunk, nem befolyásolja a túlsordulási probléma létezésének tényét.

Egy egyszerű szimulációs mérésből kiderült, hogy a rádiós link sebessége letöltési irányban körülbelül 50 Mbit/s, feltöltési irányban pedig 30 Mbit/s. A transzport hálózaton található pufferek méretének meghatározásához a sávszélesség-késleltetés szorzatot (Bandwith-Delay Product) kell kiszámolni, ami megadja, hogy mennyi adat lehet kiküldve egyszerre a hálózaton. Ezek a csomagok vagy utaznak éppen a hálózaton, vagy valamely pufferben várnak. Ezen adatok mennyiségének meghatározásához ismerni kell a hálózat körbefordulási idejét (Round Trip Time, RTT), amit a 3. fejezetben ismertetett rendszer architektúrája alapján könnyen meghatározhatunk.

Jelölje t_I az Interneten, t_{tc} a transzport hálózaton, illetve t_{rc} a rádiós kapcsolat során elszenvedett késleltetést. A szimulációk során ezek a következő értékűek voltak:

$$\begin{aligned}t_I &= 50 \text{ ms} \\t_{tc} &= 5 \text{ ms} \\t_{rc} &\approx 5 \text{ ms}\end{aligned}$$

Ezek alapján a körbefordulási idő:

$$RTT = 2 * (t_I + t_{tc} + t_{rc}) = 120 \text{ ms}$$

A sávszélesség-késleltetés szorzat *letöltés-irányú sebességgel* számolva:

$$Bw-D_Product = 50 \text{ Mbit/s} * 120 \text{ ms} = 6000 \text{ Kbit} = 750 \text{ kByte}$$

A sávszélesség-késleltetés szorzat *feltöltés-irányú sebességgel* számolva:

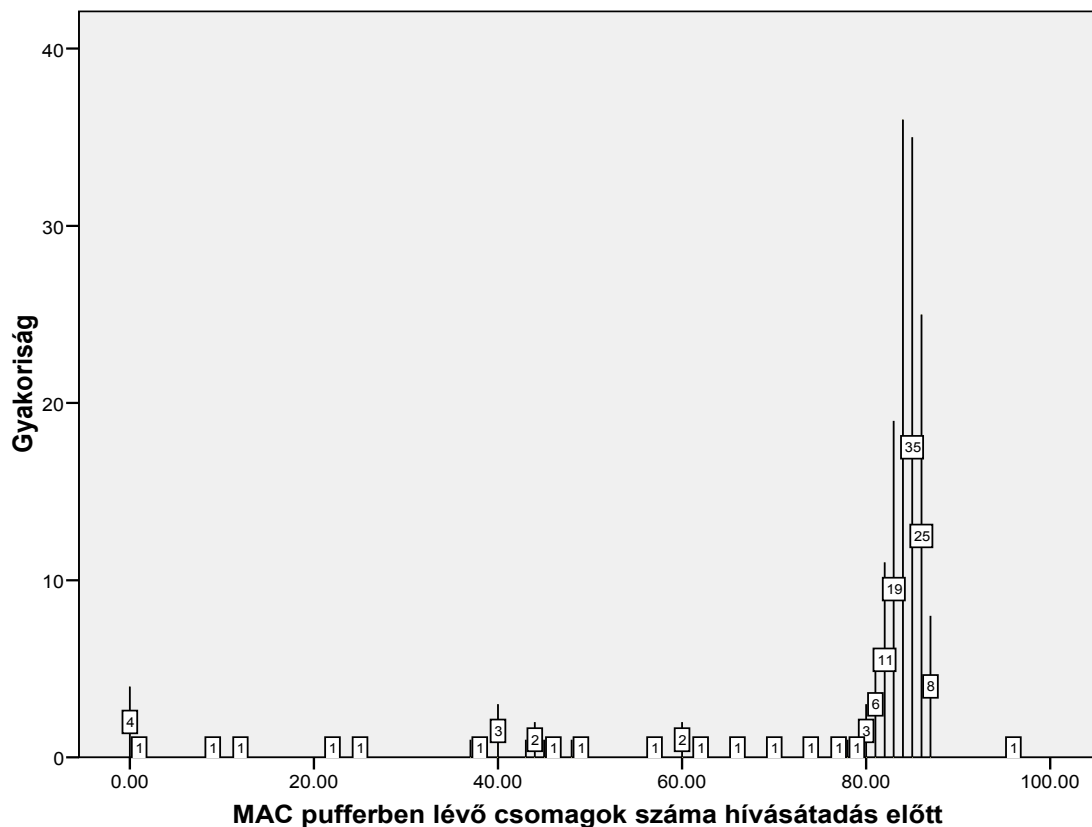
$$Bw-D_Product = 30 \text{ Mbit/s} * 120 \text{ ms} = 3600 \text{ Kbit} = 450 \text{ kByte}$$

A kapott értékekre állítottuk be a lefele-, illetve felfele-irányú transzport csatornák maximális pufferkapacitását, mert a felhasználók számát akkorára választottuk, hogy az adatforgalmuk kihasználja a transzport hálózat sebességét. Ez akkor lehetséges, ha minden bázisállomáson a letöltő felhasználók TCP torlódási ablakainak összmérete a kiszámolt 750 kByte kétszerese körüli érték, ami szimulációkban 60 felhasználó esetén teljesül. Ezzel a beállításokkal azt érjük el, hogy a transzport hálózat lefele-irányú pufferei telítődéshez közeli állapotban lesznek.

A következőkben megvizsgáljuk a transzport hálózat felfele-irányú puffereinek viselkedését kapacitáskorlát nélkül és kapacitáskorlattal is. Mivel a hálózatban 3 bázisállomást szimulálunk, ezért mindhárom felfele-irányú puffert külön kell vizsgálni. A továbbiakban mi csak egy bázisállomáshoz (1-es bázisállomáshoz) tartozó felfele-irányú puffer viselkedését vizsgáljuk, a többi bázisállomás esetén is ehhez teljesen hasonló eredményeket kapunk.

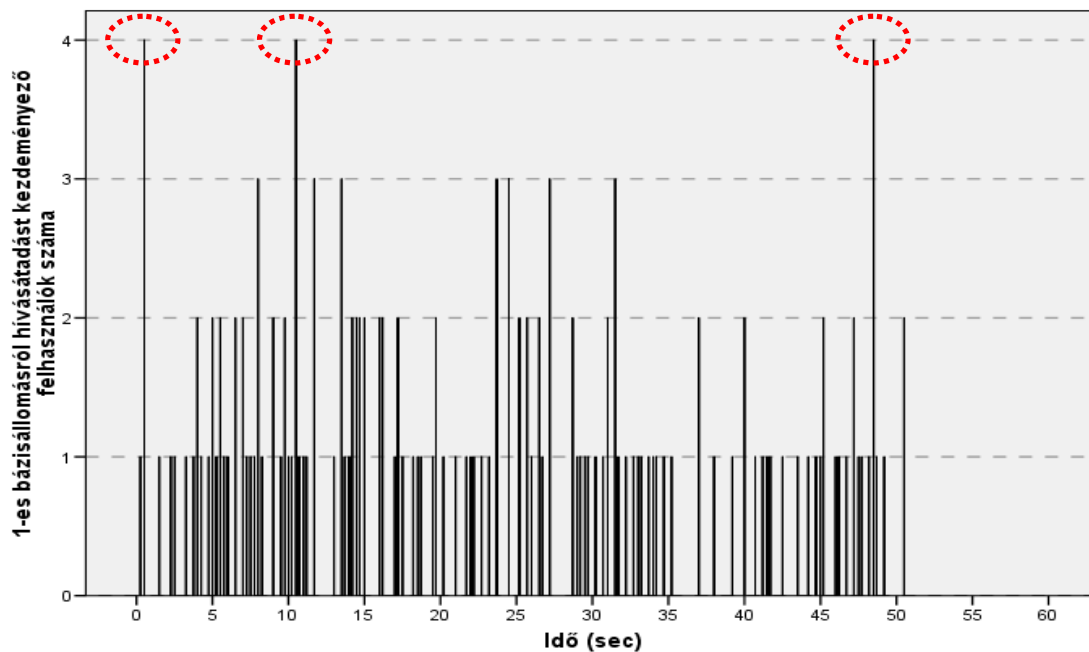
Transzport hálózati puffer mérete korlátozás nélküli esetben

Az ismertett beállításokkal szimulált eredményből származik a 4.5.1. ábra, amely egy hisztogramot ábrázol azon felhasználók lefele-irányú MAC pufferben lévő csomagok számáról, amelyek 1 bázisállomáson tartózkodnak, hiszen ezek a csomagok fognak majd a felfele-irányú pufferre kerülni. Az ábra az 1-es bázisállomásra vonatkozik, és közvetlen hívásátadás előtti eredményeket mutat. A kapott ábra a többi bázisállomáson is hasonlóan alakult.



4.5.1. ábra: Hisztogram az 1-es bázisállomástól távozó felhasználók lefele-irányú MAC pufferében lévő csomagok száma hívásátadás előtt

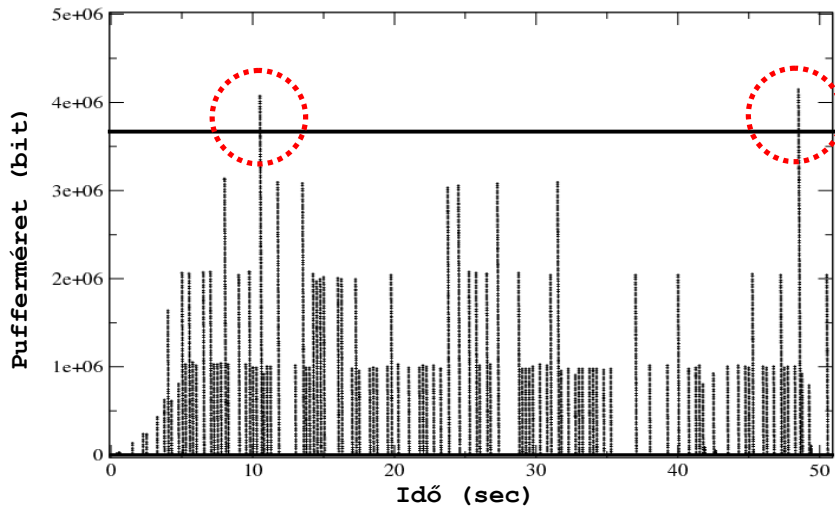
Az ábra alapján látható, hogy az összesen 175 hívásátadásból az esetek többségében az egy felhasználó lefele-irányú MAC pufferében lévő csomagok száma 80 és 86 csomag között volt, ami 1500 byte-os IP csomagok esetén körülbelül 120 kByte adatot jelent. Ha megvizsgáljuk azt is, hogy egy időben hány felhasználó kezdeményez hívásátadást, akkor látható lesz, hogy hol várható a puffer túlcserélése.



4.5.2. Az 1-es bázisállomásról egy időben hívásátadást kezdeményező felhasználók száma

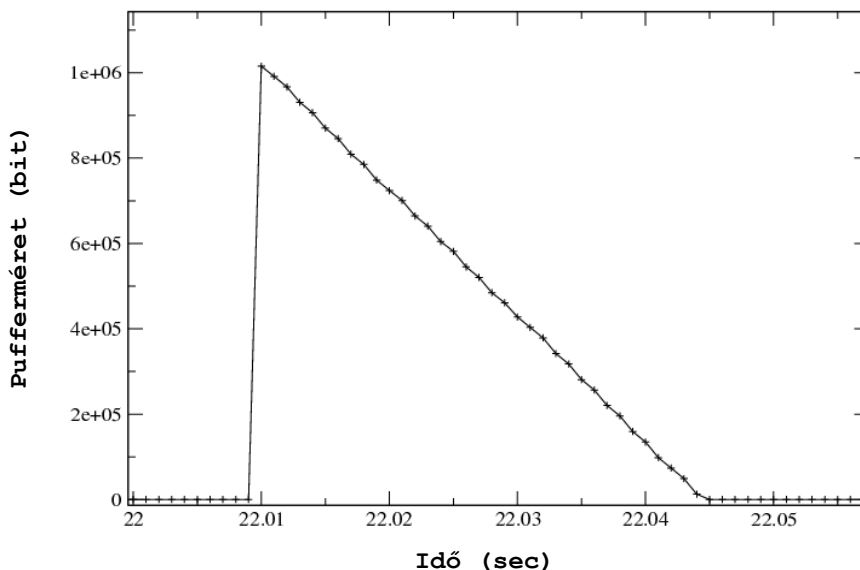
Az ábrán látható, hogy például a 10. másodperc környékén 4 felhasználó is hívásátadást kezdeményezett az 1-es bázisállomástól. A 4.5.1. ábra alapján ez a 4 felhasználó körülbelül $4 \cdot 120$ kByte (= 480 kByte) adatot tesz a felfele-irányú pufferbe, ami várhatóan el fogja érni a 450 kByte-os kapacitáskorlátot, hasonlóan 50. másodperc előtt is. A 0. másodperc utáni 4 felhasználó esetében várhatóan nem lesz túlsordulás, mert a TCP működéséből adódóan még nincs annyi adat a MAC pufferben, hogy túltöltse a transzport hálózati puffert. A 4.5.1. ábrán is látható, hogy vannak esetek, amikor nulla, vagy csak néhány csomag van a MAC pufferben. Ezek az értékek a szimuláció futásának elejéről származnak.

A 4.5.3. ábra mutatja, hogy ha a forrás bázisállomás pufferében lévő csomagokat egyszerre továbbítjuk a transzport hálózatra, akkor hívásátadásnál előfordul, hogy néhány csomagot el kell dobni. Az ábrán vízszintes vonallal jelöltük, hogy hol lépi túl a későbbiekben beállított 450 kByte-os méretkorlátot. A pufferek méretének mintavételezését 1 ms-onként végeztük el.



4.5.3. ábra: Az 1-es bázisállomás felfele irányú pufferének mérete az idő függvényében

Látható az ábrán, hogy az 1-es bázisállomáshoz tartozó felfele-irányú csatorna puffere túlsordulna 2 helyen is, ha a korlátozást beállítanánk. Mint már említettük korábban, ennek oka főleg az, hogy az adott bázisállomásnál lévő, hívásátadást kezdeményező felhasználók MAC puffereiben lévő csomagokat *egyszerre* továbbítja a felfelé irányuló csatornába, hogy azokat az átjáró egységhez eljuttassa. Ez jól látható, ha ráközelítünk az ábra egy részletére, amely eredményét a 4.5.4. ábra mutatja.

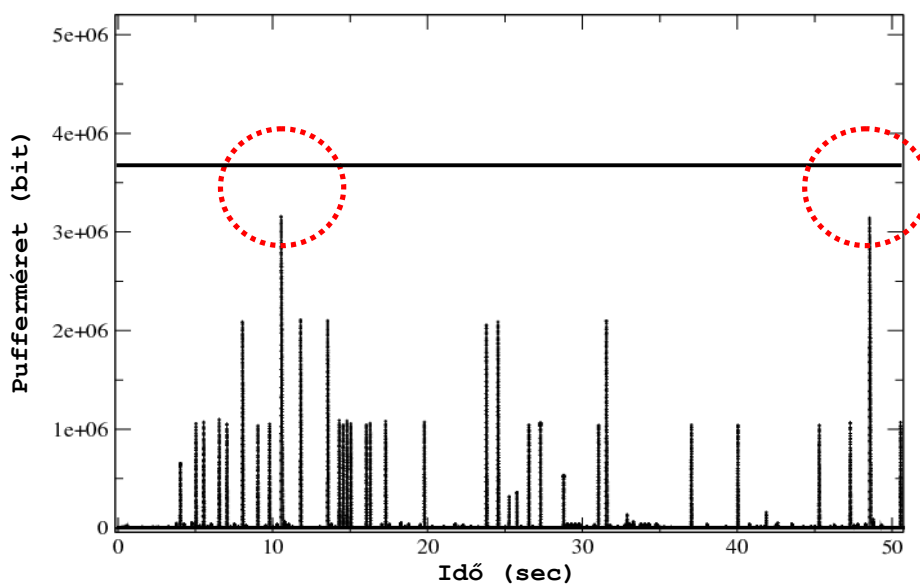


4.5.4. ábra: Puffer méretének változása egy időben történő csomagtovábbítás esetén hívásátadáskor

A 4.5.4. ábrán látható, hogy hívásátadás után hirtelen, 1 ms alatt nagyra nő a felfele-irányú puffer mérete, ami a csomagvisszahívás miatt hozzá egyszerre érkező csomagok miatt következett be. Ezt a problémát küszöböltük ki a következőkben bemutatott megoldással.

Transzport hálózati puffer mérete korlátozás esetén

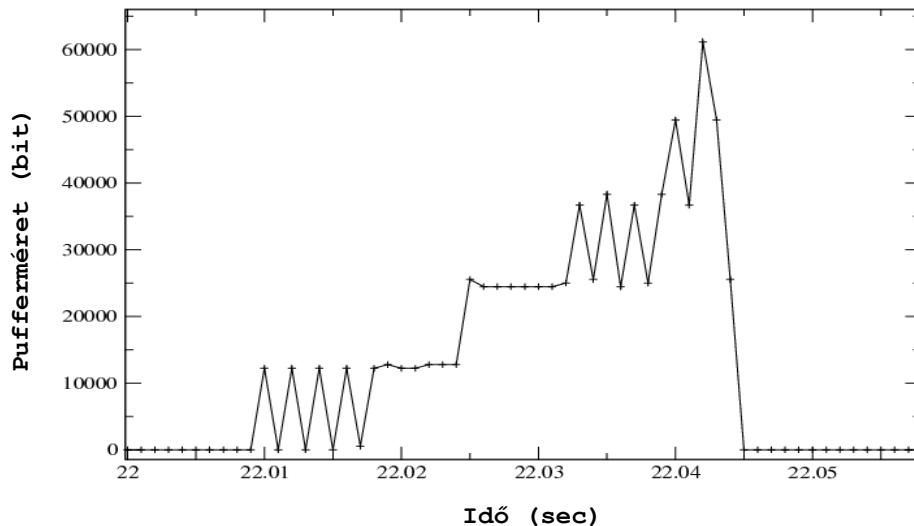
A fent bemutatott problémára egy megoldást jelenthet az, ha a forrás bázisállomás nem egyszerre küldi rá a puffertelt csomagokat a hálózatra, hanem bizonyos sebességgel továbbítaná azokat. Tehát egy egyszerű forgalomformálást kellene végeznie, de ez a formálás csak hívásátadáskor a pufferben lévő csomagok továbbítására értendő. Ezért a bázisállomást képessé tettük arra, hogy a csomagokat ne egyszerre továbbítsa, hanem egy kontrollált rátával. Kézenfekvőnek tűnhet azt a sebességet választani, ami a feltöltési link sebessége a transzport hálózaton, hiszen ha körülbelül ezzel a rátával adjuk a csomagokat a puffernek, akkor nagy valószínűséggel nem töltjük túl, mert nagyjából ugyanekkora rátával ürül is a puffer. A szimulációs eredmények alapján azt mondhatjuk, ez a kontrollált küldés sikeres megoldásnak bizonyult. Az alábbi ábrák tanulsága alapján ekkor nem történt hívásátadás miatti puffer túltöltés.



4.5.5. ábra: Az 1-es bázisállomás felfele irányú pufferének mérete az idő függvényében, kontrollált véges rátával való csomagtovábbítás esetén

Az ábra mutatja, hogy ha kontrollált sebességgel küldjük a transzport hálózat felfele-irányú csatornájának pufferéhez a csomagokat, akkor a puffer mérete nem éri el a

beállított kapacitáskorlátot, amit most is a vízszintes vonal jelöl. Ha itt is ráközelítünk ugyanarra a rövid időintervallumra, amelyre a fentiekben is, látható, hogy fokozatosan nőtt a puffer mérete, és mivel eközben csomagok szolgálták ki, így a maximális mérete nem haladta meg azt az értéket, mint az első esetben. Célunkat ezzel a megoldással tehát elértük.



4.5.6. ábra: Puffer méretének változása kontrollált véges rátával való csomagtovábbítás esetén hívásátadáskor

Az ábra jól mutatja, hogy a puffer mérete nem hirtelen ugrik, hanem *időben elcsúsztatva érkeznek a csomagok*, ami elegendő a csatorna ütemezőjének, hogy közben ki is tudjon szolgálni csomagokat, amit az ábra tüskés jellege igazol. Az is megfigyelhető, hogy valószínűleg valamelyest gyorsabban töltődik a puffer, mint ahogyan ürül, mert a mérete nő, ami a 4.5.5. ábrán is látható. Ha pontosan egyező sebességgel töltenénk a puffert, akkor ez a megoldás minden esetben kielégítő lenne, hiszen ekkor sohasem fordulhatna elő, hogy megnő a puffer mérete hívásátadáskor, mert amennyi csomag a pufferbe kerül adott idő alatt, ugyanannyi távozik is. Ha kisebb sebességgel továbbítanánk a csomagokat, mint transzport hálózati link sebessége, akkor viszont tovább tarthat a csomagtovábbítás folyamata, mint eredetileg, ami viszont olyan problémákhoz vezethet, mint a 4.3. alfejezetben ismertetett csomagfelcserélődési probléma. Viszont a növekvő pufferméret, azt is jelentheti, hogy ha jóval több visszahívott csomag lenne, akkor előfordulhatna túltöltés még ezzel a kontrollált rátával is.

A szimuláció során kapott eredményt megfigyelve, mindkét esetben ugyanolyan hosszú ideig tart kiszolgálni a csomagokat, mert egyrészt ugyanakkora mennyiségű adat érkezett, csak időben egyenletesen elosztva, így a puffer mérete közel sem ér el akkora értéket, mint a kontrollálatlan esetben (1 Mbit helyett csak 60 Kbit méretűre nőtt a puffer), másrészt pedig nem volt lassabb a továbbítás sebessége, mint a transzport hálózati sebesség pedig.

5. Összefoglalás

A dolgozatunkban egy, a jövő mobil hálózataihoz tervezett elosztott hívásátadási eljárást ismertettünk, és részletesen vizsgáltunk szimuláció segítségével.

A vizsgálódások során megmutattuk, hogy érdemes felkészíteni a bázisállomásokat a hívásátadási folyamat során az adatkapcsolati rétegben ragadt adatok visszahívására, valamint azok és az újonnan érkező adatok továbbítására. Ha ezt a funkciót beépítjük a hálózatba, az adatátviteli sebesség növelésén kívül még például a mobil készülék is sokkal érzékenyebb a rádiós csatorna áthangolásának idejére.

Rávilágítottunk arra is, hogy a teljes csomagtovábbítás a csomagok sorrendjének keveredését okozhatja a cél bázisállomásnál. Szükséges és elégséges feltételt adtunk a késleltetési paraméterek függvényében, hogy mikor fordulhat elő csomagsorrend keveredés a hívásátadási folyamat közben, illetve annak befejezte után. A felmerülő sorrend felborulási problémára adtunk egy lehetséges megoldást, a cél bázisállomásban használt prioritásos ütemezőt, melynek hatását szintén szimulációs vizsgálatokkal ellenőriztük. Megállapítottuk, hogy az általunk javasolt módszer az esetek többségében leküzdí a sorrend felecserelődési problémát.

A hívásátadási folyamat hatásait tovább kutatva megmutattuk, hogy az adatkapcsolati rétegben is felmerülhetnek problémák abból adódóan, hogy hívásátadáskor új MAC kapcsolat jön létre a cél bázisállomással. Vizsgáltuk, hogy milyen esetekben fordulhat elő nem sorrend helyes továbbítás, illetve csomagduplikálás. Kielemeztük a MAC réteg szelektív és kumulatív csomagvisszahívási mechanizmusát.

Végül teljes csomagtovábbítást feltételezve megmutattuk, hogy a MAC rétegből visszahívott, és hálózatra küldött csomagok több felhasználó együttes hívásátadása esetén túltölthetik a transzport hálózati linkek puffereit. Ezért megvalósítottunk egy forgalom formálót, mely egy véges, kontrollált sebességgel engedi a visszahívott csomagokat a hálózatra.

Irodalomjegyzék

- [1] Hannes Ekström, Anders Furuskär, Jonas Karlsson, Michael Meyer, Stefan Parkvall, Johan Torsner, Mattias Wahlqvist, Ericsson, „*Technical Solution for the 3G Long-Term Evolution*”, IEEE Communications Magazine, 2006. március
- [2] Andrew T. Campbell, Javier Gomez, Sanghyo Kim, Chieh-Yih Wan, Columbia University, Zoltan R. Turanyi, Andras Valko, Ericsson Research, „*Comparision of IP Micromobility Protocols*”, IEEE Wireless Communications, 2002. február
- [3] D. Johnson, Rice University, C. Perkins, Nokia Research Center, J. Arkko, Ericsson, „*Mobility Support in IPv6*”, 2004. június
- [4] A. T. Campbell and J. Gomez, „*IP Micro-Mobility Protocols*”, ACM SIGMOBILE, Mobile Comp. and Commun. Rev., vol. 4, no. 4, 2001. október
- [5] Ericsson White Paper, „*Basic Concepts of WCDMA Radio Access Network*”, 2001. november
- [6] Stefan Parkvall, Eva Englund, Peter Malm, Tomas Hedberg, Magnus Persson, Janne Peisa, „*WCDMA evolved—High-speed packet-data services*”, 2004. december
- [7] W. Richard Stevens, „*TCP/IP Illustrated, Volume 1, The Protocols*”, Addison-Wesley Publishing Company, 1995. október
- [8] 3GPP TR 25.813, „*Evolved UTRA and Evolved UTRAN: Radio Interface Protocol Aspects*” (nem végleges), http://www.3gpp.org/ftp/Specs/archive/25_series/25.813/25813-710.zip, 2006. szeptember
- [9] 3GPP TR 23.882, „*3GPP System Architecture Evolution: Report on Technical Options and Conclusions*” (nem végleges), http://www.3gpp.org/ftp/Specs/archive/23_series/23.882/23882-142.zip, 2006. szeptember
- [10] Harri Holma, Antti Toskala, „*WCDMA for UMTS*”, 3rd edition, Wiley&Sons, 2004