

Kliensvezérelt mobilitás – mobilitásmenedzsment új nézőpontból

KOVÁCS BENEDEK, FÜLÖP PÉTER

Budapesti Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék
{bence, fpeti}@mcl.hu

Lektorált

Kulcsszavak: mobilitásmenedzsment, modellezés, kliensvezérelt, jelzésforgalom

Cikkünk egy új mobilitási algoritmust mutat be, melynek lényege, hogy a megszokott protokollokkal ellentétben nem a hálózat, hanem a mobil eszköz menedzseli önmaga mozgását. A hálózat ebben a megoldásban csupán alapvető szolgáltatásokat nyújt a mobil számára: a kapcsolatot és az adminisztrációt. Megalkottuk az új megközelítést kezelő kliensvezérelt mobilitási keretrendszert is. Bemutatjuk, hogy ebben milyen szerepet játszanak a hálózati eszközök és milyen funkciókat kell ellátniuk, illetve, hogy a mobil eszközön mit kell megvalósítani a helyes működéshez. Példát adunk néhány alapvető mobilitási stratégia; úgymint a centralizált, hierarchikus, cellás vagy éppen a tracking módszer implementációjára a keretrendszerben. Az új módszer működőképességét és létjogosultságát alátámasztjuk szimulációs környezetben elvégzett tesztekkel és a klasszikus megoldással összehasonlító elemzéssel.

1. Bevezetés

Az információmobilitás a modern világ egyik alapvető követelménye. Annak ellenére, hogy nagyon sok ezt megvalósító protokoll létezik, az IP-be ágyazott mobilitás még várat magára, pedig az egyre szélesebb körben terjedő, különböző kommunikációs eszközök alapvetően az IP-t használják.

Az IP statikusságának megkerülésére nagyon sok megoldás született, melyeknek megvannak a jó tulajdonságai és a hátrányai. Ha megnézzük őket közelebbről, akkor általában a komplexitás (vagy egyszerűség) áll szemben a költségoptimalizálással. Léteznek nagyon összetett hálózati struktúrával, és funkcionalitással rendelkező módszerek, melyek kevés jelzésforgalmat generálnak, de a másik oldalon ott van a régi, jól ismert Mobil IP, ahol a hálózat nagyon egyszerű, de a jelzés költsége magas.

A „komplexitás-optimalitás” dilemát mi sem oldjuk fel, de áthelyezzük egy másik dimenzióba, a hálózati szintről az individualitás szintjére, azaz a mobil eszközre.

A cikkben bemutatunk egy általunk kidolgozott, kliens által irányított mobilitási rendszert (CMFS – Client-based Mobility Frame System, kliensvezérelt mobilitási keretrendszer), amiben egyszerűen megvalósíthatóak a klasszikus, de akár az újabb, bonyolultabb mobilitási algoritmusok a mobil eszköz irányításával. Nem állítjuk, hogy megtaláltuk a legoptimálisabb IP alapú mobilitást, de bizonyítjuk, hogy ezzel a módszerrel sok esetben hatékonyabban lehetünk az eddigieknél.

Az alapvető ötlet, hogy a hagyományos protokollokkal, GSM-mel, Mobil IP-vel (IPv4 és IPv6) ellentétben a hálózati elemekben és a hálózati struktúrában nem kell semmiféle, az adott protokoll által megkívánt funkcionalitást implementálni. A hálózat maradhat egyszerű, csupán alapvető parancsokat kell értelmeznie és végrehajtania a csomópontoknak. Ilyen, a mobil által küldött egy-

szerű parancsok a csomagtovábbítás, vagy a regisztráció. A logika és irányítás, s így az egész menedzsment-algoritmus a mobilban dolgozik. Ebből következik a rendszer legfőbb előnye: minden mobil a számára optimális, legkedvezőbb és a hálózat számára legkisebb költséggel járó mobilitási stratégiát választhatja, sőt, akár működés közben változtathatja.

A későbbiekben bemutatjuk, hogyan lehet implementálni a klasszikus mobilitási stratégiákat, mint például a centralizált, hierarchikus, cellás vagy tracking módszert ebben az új környezetben, de elsőként egy protokoll-leírást és definíciót adunk a CMFS-hez.

2. CMFS – kliens-vezérelt mobilitási keretrendszer

E fejezetben definiáljuk a CMFS (Client-driven Mobility Frame System) által specifikált alapvető szerepeket és azt, hogy milyen igényeknek kell eleget tenni a hálózati eszközöknek, hogy kommunikálni tudjanak a mozgó entitással. Leírunk egy egyszerű módszert a hálózat felderítésére, mellyel megmutatjuk, hogy a mobil fel tudja építeni a saját logikai hálózatát. Ebben a fejezetben foglalkozunk továbbá a csomópontok adatbázisában megtalálható bejegyzések karbantartásával, elévülésével és törlésével.

2.1. A modell alapjai

A modellezés során a mobilitást általánosan kezeljük, a pontos, alkalmazott technológiára való tekintet nélkül. Az entitások, tulajdonságok alapjai a *mobilitásmenedzsment keretrendszerből* származnak [1].

A modellben általunk használt hálózati elemek és szerepük a következők:

- *Mobil állomás* (Mobile Node, MN): mozgó csomópont, amely más mobillal vagy fix állomással kommunikál.

- *Mobilitási hozzáférési pont* (Mobility Access Point, MAP): az egyetlen olyan típusú, fix entitás a hálózatban, amelyhez a mobil állomás képes kapcsolódni, és rajta keresztül kommunikálni. (Nem feltétlenül felel meg ez egy hozzáférési pontnak a valós hálózatban, hanem akár hálózatot, vagy hozzáférési pontok halmazát is jelentheti például abban az esetben, ha a mobil egyidejűleg képes több bázisállomással is kapcsolatban állni).
- *Mobil ügynök* (Mobile Agent, MA): fix csomópont, amin fut az adott mobilitásmenedzsment-algoritmus, de mobil nem tud csatlakozni hozzá.
- *Otthoni ügynök* (Home Agent, HA): speciális MA, a mobil entitás itt van regisztrálva, ez az otthoni hálózata. Ez az MA mindig ismeri az MN pontos vagy becsült helyét.
- *Egyéb csomópont* (Node, N): olyan csomópont a hálózatban, amely a mobilitásmenedzsment szemszögéből semmilyen funkcióval nem rendelkezik. Ide tartoznak a fix kommunikációs partnerek, routerek és egyéb hálózati elemek.
- *Hálózat* (Core Network, CN): az entitásokat összekötő kommunikációs (vezetékes, vagy vezeték nélküli) csatorna, amit egy gráffal adunk meg. A gráf csomópontjai a fentebb felsorolt hálózati entitások, az élei a közöttük fennálló kapcsolatok.

Ezzel az egyszerű leírással az alapvető funkciók leírhatók, de néhány praktikus megkötést még bevezetünk:

– A mobil hozzáférési pont (MAP) minden esetben mobil ügynök (MA) is. (Ahogy már említettük, a MAP jelenthet egy alhálózatot is, magában foglalva több hozzáférési pontot és kommunikációs partnereket. Ez teszi lehetővé, hogy szükség szerint mikromobilitás vagy éppen magasabb szintű mobilitás modellezésre alkalmazzuk a módszerünket).

– A mobil entitás (MN) legalább egygyel, de akár több MAP-pal is kapcsolatba kerülhet. Helyzetéről releváns információt mindig egy ügynök tárol, ezért a feladat, hogy megtaláljuk a mobilt, ekvivalens azzal, hogy megtaláljuk a MAP-ot, amihez kapcsolódik.

2.2. A modell ötlete

A kliensvezérelt mobilitás ötletét az a tény inspirálta, hogy a mobil felhasználók általában mindig meghatározott hozzáférési pontok között mozognak és nagyon ritkán hagyják el ezeket.

Annak érdekében, hogy a mobil jól tudja menedzselni a saját mozgását, ismernie és tárolnia kell hozzáférési pontokat (MAP) és mobil ügynököket (MA), akikkel már kapcsolatba lépett. Ezeket és kapcsolataikat hívjuk logikai hálózatnak (LN), amit a mobilnak mindig frissíteni kell, hogy megfelelő döntést tudjon hozni és aminek mérete függ attól, hogy milyen algoritmust futtat az MN.

A későbbiekben mutatunk példát a CMFS-ben implementált Mobil IPv4 típusú protokollra (3.1. fejezet) és ott látható lesz, hogy a mobil node-nak elég három (vagy éppen csak kettő) node-ból álló logikai hálózatot fenntartani, azaz egy kis kapacitású mobil eszköz is képes megvalósítani alapvető mobilitást a rendszerünkben.

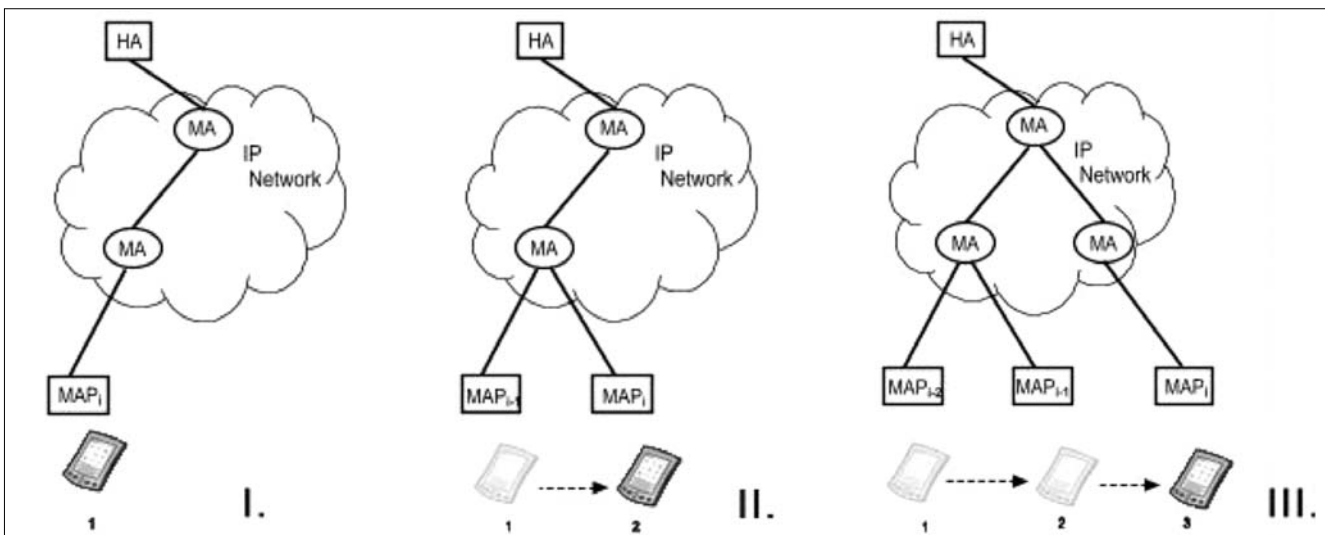
A legfontosabb előnye a megoldásunknak az, hogy a hálózati üzemeltetőknek nem kell választaniuk egy konkrét mobilitásprotokollt, hiszen nem biztos, hogy minden előfizetőjükre az a hatékony. A CMFS-t használó mobil ki tudja választani a számára legmegfelelőbb algoritmust, így a használt mobilitási stratégia mindig költség-hatékony lesz az adott pillanatban.

2.3. Hálózat felderítése

Akár egyszerűbb, akár komplexebb mobilitási stratégiát akar folytatni a mobil, logikai hálózatát fel kell építenie és fenn kell tartania. Az IP világban sok módszert használnak heterogén hálózatok felderítésére. Írásunk célja az, hogy bizonyítsuk az általunk fejlesztett rendszer működését, a hálózat felderítésének módszere nem tartozik a fókuszpontok közé. Ezért egy egyszerű, nem feltétlenül optimális eljárást mutatunk be a logikai hálózat kiépítésére.

A legegyszerűbb megoldás az IP csomag opciók mezőjének használata lenne hálózati információk szállítására, azonban ezt a routerek nagy része nem tudja értelmezni az implementáció hiánya miatt, ezért az ilyen

1. ábra A logikai hálózat felépítése



csomagokat eldobja. A másik kézenfekvő lehetőséget a *traceroute* által használt elv adja. Ekkor az IP csomag TTL (Time To Live) mezőjének értékét alacsonyra állítjuk, és így annál a csomópontnál, ahol ez lejár, ott egy ICMP (Internet Control Management Protocol) Time Exceeded üzenetet generál a feladóhoz. Szisztematikusan, a TTL mezőt egytől kezdve, folyamatosan növelve, felde-ríthető a célállomásig a hálózat.

Egy teljesen hasonló megoldást implementáltunk a CMFS-ben is, amit most egy példán keresztül bemutatunk. (A protokoll pontos implementációja a 2.5. fejezetben olvasható.)

Egy alapprotokollban, mint például a Mobil IP-ben, az első lépés, hogy a mobil (MN) beregisztrálja magát az otthoni ügynökhöz (HA). Ez lesz az első pont a logikai hálózatban. Ezután az MN mozog és egy másik hozzáférési pontnál (MAP) kap egy ideiglenes IP címet. Ekkor a CMFS MN mondja meg pontosan, hogy mit kell tenni: regisztráció a HA-hoz, vagy például egyéb más MA-k

hoz. Ha közben, az MN szeretné megismerni, hogy az új MAP-tól, a HA-ig milyen a hálózat, akkor a CMFS *register* üzenetben bizonyos bitek beállításával felszólít speciális MA-kat, vagy akár az összes MA-t, hogy egy *reply* üzenet küldjenek a feladónak. A *reply* üzenet megérkezéséhez tartozó időbélyeget fel tudja használni az MN a link költségének meghatározásához. Az MN elmenti a megismert MA-kat a saját adatbázisába, így karbantartja logikai hálózatát. Ha közben valamelyik MAP-ra vagy MA-ra már nincs szüksége, akkor törli őket az adatbázisból. Ha egy bejegyzés már elévült, akkor automatikusan törölődik a MA/MAP-ok táblájából.

Ezekre az információkra támaszkodva a mobil entitás nagyon komplex döntési algoritmusokat is futtathat a leghatékonyabb stratégia meghatározásához.

2.4. Hálózati követelmények

Ahhoz, hogy az MN-t és az általa választott algoritmust ki tudjuk szolgálni, bizonyos követelményeknek eleget kell tenniük a hálózatban található MA-knak. Ha ezeknek a képességeknek birtokában vannak a node-ok, akkor a mobilok bármilyen mobilitásmenedzsment-algoritmust használhatnak, egymástól függetlenül is.

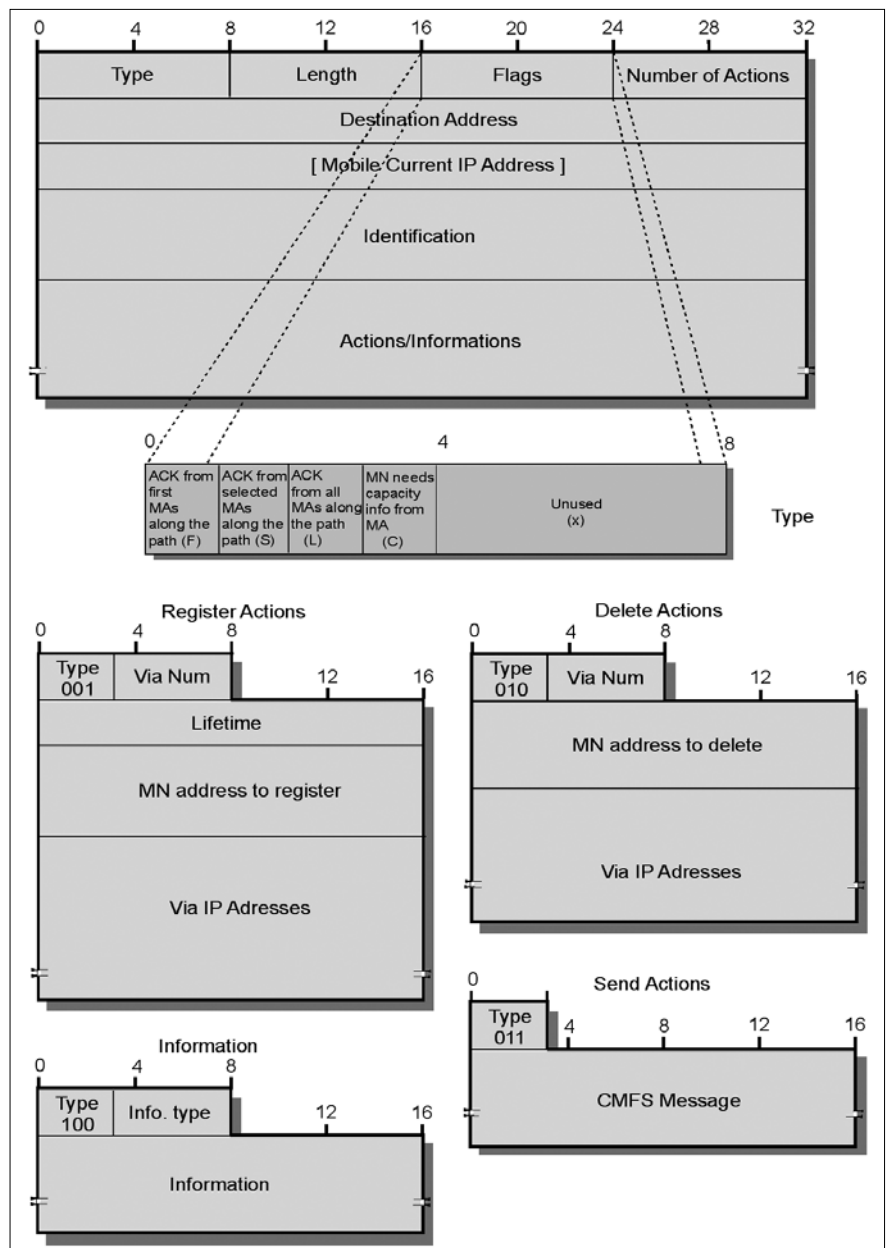
A HA-nak mindig tudnia kell, hogy merre route-olja a csomagokat a mobilnak. Egy adatbázisban tárolja a kötést a mobil generális IP címe és az aktuális IP címe között. A többi MA

is hasonló módon működik. Ha kap egy csomagot, amit a mobilnak címeztek, akkor megnézi a bejegyzései között, hogy melyik MA vagy MAP felé kell továbbítania. Ha nincs bejegyzés, akkor eldobja a kérést.

Hogyan regisztrál a mobil egy MA-hoz? Ha egy új MAP-hoz kapcsolódik egy sikeres handover után, természetesen oda regisztrál, de megadhatja, hogy a regisztrációs procedúrát folytatni kívánja további, magasabb szinten elhelyezkedő MA-khoz. Nézzünk egy ilyen összetett kérést és annak reprezentációját:

```
[Dst: MAPI, Src: MN, Actions: Register MN to MAPI via MN;
 [Dst: MAj , Src: MAPI, Actions: Register MN to MAj via MAPI,MAPIi,MAPIii;
 [Dst,Src,Actions: ; ;
 [ ...
 [Dst: HA, Src: MAn, Actions: Register MN to HA via MAn]
 ]
 ]
 ]
 ]
```

2. ábra
A CMFS üzenetstruktúrája



A struktúra értelmezése a következő: először is az MN regisztrációját kéri az MAP_i-nál (Register MN to MAPi via MN), és utasítja MAP_i-t hogy küldje tovább a belső struktúrát az MA_j-nek. Ezt megkapja MA_j és bejegyzi, hogy MN elérhető a MAP_i-, MAP_{ij}-, MAP_{iii}-ken keresztül (Register MN to MAj via MAPi,MAPii,MAPiii). Azaz egy cellás típusú (CIP) [2] stratégiát kért az MN. Ha legközelebb egy csomag érkezik MA_j-hez az MN-nek címezve, akkor továbbítani fogja azt MAP_i-, MAP_{ij}-, MAP_{iii}-hez egyaránt, azaz elárasztja a három MAP-ból álló *page*-et (3.4. fejezet). A regisztrációs struktúra legbelső utasítása a HA-nak szól. Ha ezt nem tartalmazná a struktúra, akkor egy Hierarchikus Mobil IP [3] típusú megoldást kér a mobil.

Ha mozgó entitás a következő üzenetet küldi:

```
[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;
 [Dst: MAPi1 //The former node//,
   Src: MN, Actions: Register MN to MAPi1 via MAPi
 ]
]
```

akkor HAWAII [8] típusú protokollt akar használni.

Vezetéknélküli tracking-stratégiához (pl. LTRACK [4]) a mobilnak a következő két különálló üzenet kell küldeni:

```
[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;]
[Dst: MAPi1 //The former node//,
 Src: MN, Actions: Register MN to MAPi1 via MAPi]
```

A bemutatott üzenetstruktúra és reprezentáció csak az utasítások egy lehetséges megjelenése, mely változtatható. A fő szempont, mint minden protokoll esetében, hogy egységesen használják a hálózati eszközök. A következőkben egy konkrét implementációt mutatunk be a CMFS-hez, amit szimulációban is megvalósítottunk.

2.5. CMFS Protokoll

A fentebb, példaként bemutatott üzenetstruktúrához fejlesztettünk egy alkalmazás-szintű protokollt, a CMFS Protokoll-t (CMFSP). CMFSP-t UDP szállítja, ahogy az megszokott a TCP helyett a mobilitási alkalmazásokban (természetesen implementálható bármelyik másik, újabb, mobilitásra kitalált szállítási protokollban is).

A CMFSP struktúrája a 2. ábrán látható.

A fejrész négy 1 bájtos (*Type, Length, Flags, Number*) és egy 4 bájtos (*Destination Address*) elemet tartalmaz. Két alapüzenetet különböztetünk meg, *Request*-et és *Reply*-t. A *Length* mező tartalmazza a teljes üzenet hosszát, a fejrészt is beleértve. A *Destination Address* a node, ahol fel kell dolgozni a CMFS kérést.

A *Flags* mezőben 4 különböző bitet lehet állítani, amivel az MN szabályozza, hogy milyen információt kapjon az MA-któl a logikai hálózatának felépítéséhez. A négy különböző bit a következő:

- F – az MN-től a HA-ig vezető úton az első MA-tól kér információt a LN felépítéséhez;
- S – az MN-től a HA-ig vezető úton az érintett MA-któl kér információt (azoktól akik feldolgozzák az MN CMFS üzenetét);
- L – az MN-től a HA-ig vezető úton az összes MA-tól kér információt;
- C – kapacitásról is kér információt az MN.

Az *Actions/Informations* mező tartalmazza az MA/ MAP-k számára az utasítást vagy MN számára információt a logikai hálózat felépítéséhez. Három különböző *Action* utasítást különböztetünk meg:

Register – Regisztráció az adott MA/MAP-hoz, MN elérhetőségéről

Delete – Bejegyzés törlése az adott MA/MAP-nál

Send – Üzenet továbbküldése a megadott cél MA/MAP-hoz.

Ebben a fejezetben egy konkrét alkalmazásréteg-beli megoldást mutattunk be a CMFS-hez. Természetesen ez csak egy példa, bármilyen más megvalósítás is működőképes lehet.

3. Példák CMFS-ben implementált mobilitásmenedzsment stratégiákra

A CMFS előnye nemcsak az, hogy az MN-ek egymástól függetlenül használhatnak különböző stratégiákat, hanem az egyes MN-ek is tudnak könnyedén váltani a közöttük. Csak annyi szükséges, hogy a mobil információt gyűjtsön a hálózatról és saját mozgásáról, majd meghozza a döntését a benne implementált algoritmus alapján és elküldje a parancsokat a megfelelő MA/MAP-oknak.

A következőkben a legismertebb mobilitási megoldások implementációját mutatjuk be a CMFS környezetben. Az implementációkhoz készített algoritmusokat a területi korlátok miatt nem prezentáljuk, ezeknek utána lehet olvasni a [12]-ben.

3.1. Personal Mobile IP – PMIP

A *Personal Mobile IP* működése nagyon egyszerű, gyakorlatilag a Mobil IP adaptációja CMFS környezetben. Amikor az MN csatlakozik egy új MAP-hoz, szimplán regisztrálja magát a HA-hoz. A következő üzenettel lehet végrehajtani ezt az akciót:

```
[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;
 [Dst: HA, Src: MAPi, Actions: Register MN to HA via MAPi];
 [Dst: MAPi1, Src: MAPi, Actions: Delete MN in MAPi1 via MN]
].
```

A törlésparancsot (*Delete MN in MAPi1 via MN*) hordozó üzenet nem szükséges, csak akkor, ha a felesleges bejegyzések kezelése az MN feladata. Általában ezek softstate módon tárolódnak, azaz egy bizonyos idő után automatikusan törlődnek, ha nem frissíti az MN.

Ezt az egyszerű PMIP működést nevezzük *Pure-PMIP*-nek (P-PMIP), ami a hagyományos Mobil IP protokollnak felel meg és hasonló terhelési karakterisztikával bír, mint ahogy az a 4. szakaszban látható. Ahogy már korábban említettük, itt az MN-nek elég csupán 2 node-ból álló logikai hálózatot fenntartani az adatbázisában.

Ha az MN szeretne QoS-t is biztosítani, akkor alkalmazhatja a PMIP kiterjesztését, az *Extended-PMIP*-et (E-PMIP). E-PMIP biztosítja a 0 csomagvesztést és nem tart fenn felesleges bejegyzéseket a hálózatban megtalálható MA/MAP-okban. Természetesen a használt üzenet ennek megfelelően jóval komplexebb:

```
[Dst: MAPi, Src: MN, Actions: Register MN to MAPi via MN;
 [Dst: MAPi-1, Src: MAPi, Actions: Register MN in MAPi-1 via MAPi;
 Delete MN in MAPi-1 via MN;
 [Dst: HA, Src: MAPi-1, Actions: Register MN to HA via MAPi;
 Delete MN in HA via MAPi-1;
 [Dst: MAPi-1, Src: HA, Actions: Delete MN in MAPi-1 via MAPi
 ]]]
```

A P-PMIP és E-PMIP működési elvét a 3. ábra mutatja, részletes teljesítmény-analízisük pedig a 4. szakaszban olvasható.

3.2. Personal Hierarchical Mobile IP – PHMIP

Hierarchikus Mobil IP esetén (kétszintű hierarchia) az első kérdés, ami felmerül, hogy melyik MA-t nevezzük ki a hierarchia pontnak a hálózatban. Az LN alapján az MN ezt is el tudja dönteni. Ezek után már adott a *Personal Hierarchical Mobile IP* működése és az üzenet is egyszerű (4. ábra).

Komplikáltabb már a helyzet, ha több szintű hierarchiáról beszélünk. Ez esetben az MN-nek már jóval bonyolultabb számításokat kell végeznie. Egy egyszerű eljárás a hierarchiapont meghatározására a következő lehet: az MN először a HA-hoz csatlakozik, majd átmozog egy idegen MAP-hoz. Itt eltérő a HA-hoz vezető úton megtalálható MA-kat. Majd amikor tovább lép egy új MAP-hoz, akkor ismét felderíti az utat a HA felé. Az első közös MA elem a két úton lesz a hierarchiapont az MN számára.

3.3. Personal Tracking Mobile IP – PTMIP

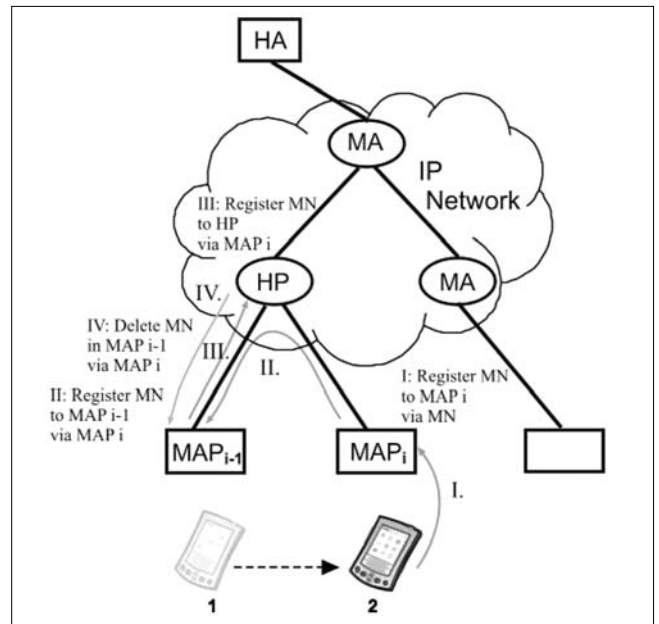
A tracking típusú mobilitás elve ismert már akár a HAWAII-ból [8], akár az LTRACK-ból [4]. Megvalósítása a CMFS környezetben szintén nem okoz túl nagy fejtörést.

A *tracking handover* esetén, amikor is az MN vezetékes vagy vezeték nélküli kapcsolaton csak az előző MAP-hoz jelenti be új kapcsolódási pontját (5. ábra) az üzenet a 2.4. fejezetben bemutatottak alapján konstruálható meg.

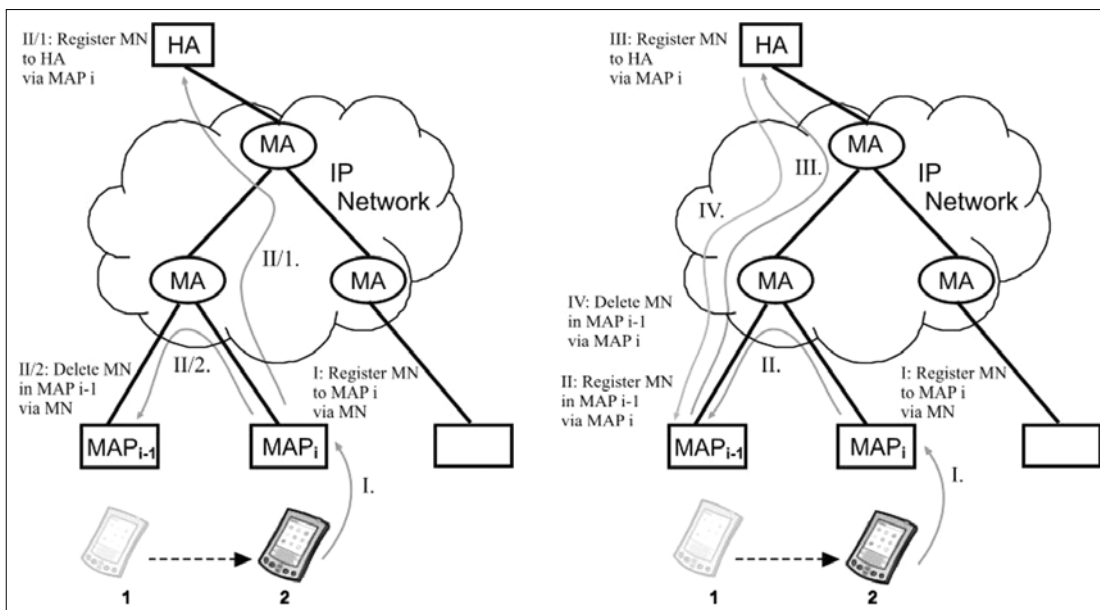
Bizonyos számú tracking handovert követően, *normal handover* kell a mobilnak végrehajtania, ami már teljes mértékben meggyezik a PMIP-nél megismert módszerrel. Hogy hány tracking handover után kell frissítenünk a HA-adatbázisát, arról több értekezés is született már az irodalomban [4].

3.4. Personal Cellular Mobile IP – PCMIP

A széles körben ismert GSM-nek köszönhetően, a mobilitási protokollokban is elterjedtek a cellás típusú megoldások. Az ötlet abban rejlik, hogy az MN a MAP-ok egy meghatározott halmaza között hajt végre handovereket, amiről nem értesít senkit. Így viszont amikor kérés érkezik hozzá, akkor az összes MAP-nál keresni kell az mobilt. Bizonyos esetekben ezzel a technikával sok jelzéseköltséget lehet megspórolni. A kulcskérdés ebben az esetben, hogy milyen MAP-okat teszünk egy halmazban, más néven *page*-be, úgy hogy a lehető leghatékonyabb legyen a protokoll működése.

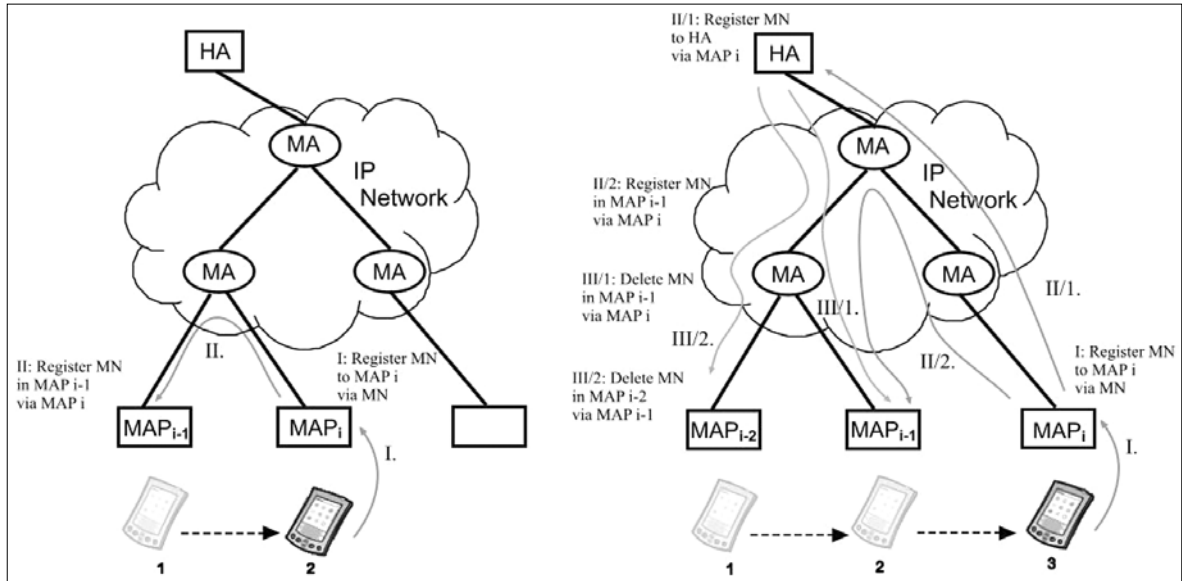


4. ábra PHMIP működési elve



3. ábra Baloldalt a P-PMIP, jobboldalt az E-PMIP működési elve

5. ábra
PTMIP
működési
elve.
Baloldalt
egy tracking
handover,
jobboldalt
egy normál
handover
folyamata.



A *Personal Cellular Mobile IP*-ben az MN-nek csak akkor kell regisztrációs üzenetet küldenie, hogyha átlépi a *page* határát. Ekkor az MN következőket teszi:

- kinevezi az új *page* vezető MA-ját;
- utasítja, a kinevezett MA-t, hogy regisztrálja be az MN-t felsőbb szinten, így már ebben az új *page*-ben lesz megtalálható;
- meghatározza az új *page*-be tartozó MAP-okat.

Az ehhez szükséges üzenet a következő:

```
[Dst: MAPleader //The leader of the paging area//,
Src: MN, Actions: Register MN to MAPleader via MAPi, MAPii, ... ,
  [Dst: HA, Src: MAPleader, Actions: Register MN to HA via MAPleader
  ]
]
```

Megoldásra vár a cellás rendszerekben a hatékony *page*-formálás, ami egy NP teljes probléma. A kutatásokban általában ezt aggregáltan, az összes mobil mozgását vizsgálva végzik. Természetesen, ha egyénileg, egy MN-re csináljuk, az szintén NP teljes problémához vezet. Szerencsére vannak alternatív lehetőségek, amik elég jól közelítik az optimális *page*-kiosztást. Egy egyszerű módszert mutattunk be erre [12]-ben.

4. Szimulációs eredmények

A fentebb bemutatott, sokoldalúan felhasználható CMFS mobilitási algoritmus létjogosultságának alátámasztására egy szimulációt készítettünk. A szimulációs környezetet a komponens alapú, nyílt forráskódú, C++ nyelvet használó OMNET++ biztosította.

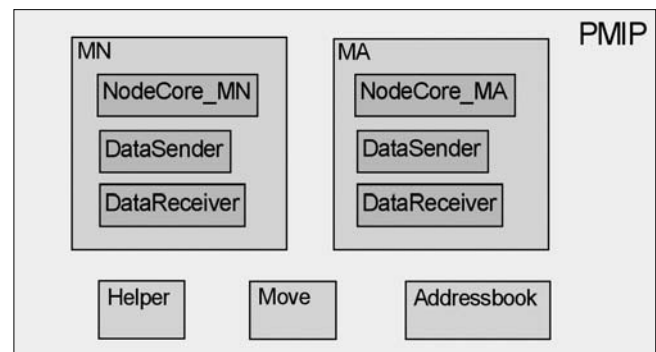
A programban két fő, összetett komponenst implementáltunk, az MN-t és a MA-t, illetve mellettük azonos hierarchiaszinten található egyszerű komponenseket. Az MN és az MA is nagyon hasonló belső struktúrával rendelkezik: mindegyikben van egy mag (*NodeCore_MN* és *NodeCore_MA*), amely alapjukat tekintve megegyezik és hálózati forgalom kezelésére szolgáló egyéb modulok (*DataReceiver* és *Datasender*) is megtalálhatók bennük (6. ábra).

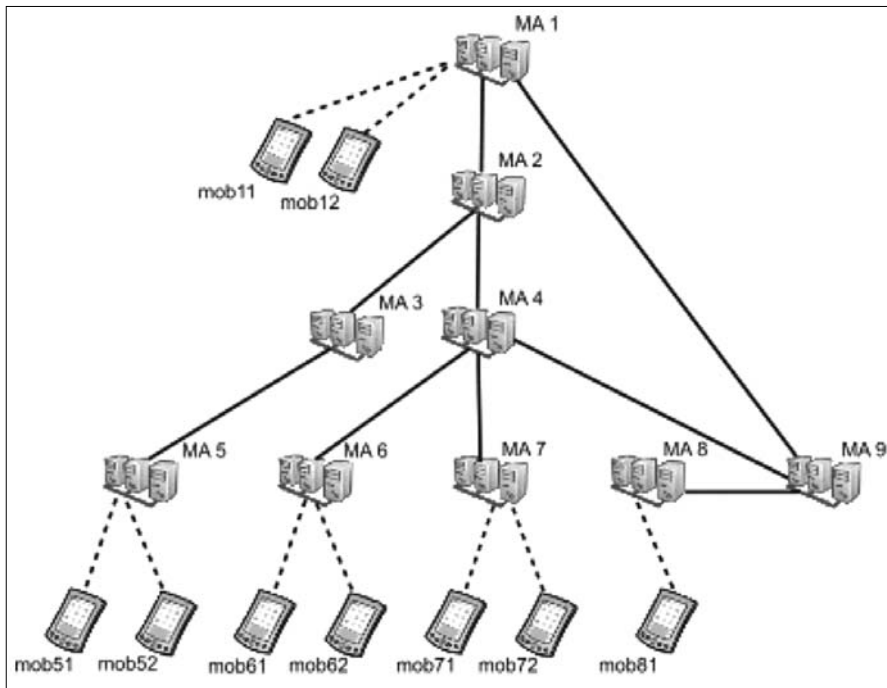
A 2.5. szakaszban pontosan definiált CMFS protokoll implementációját a *NodeCore* komponensekben valósítottuk meg. Az MN-hez tartozó *NodeCore_MN* komponens felel a CMFSP üzenetek összeállításáról, elküldéséről és a logikai hálózat felépítéséről, karbantartásáról. Az MA-ban megtalálható *NodeCore_MA* értelmezi a fogadott CMFSP üzeneteket, végrehajtja a benne található utasításokat, fenntartja az adatbázist és ez alapján routolja a hozzá beérkező adatcsomagokat.

A *DataReceiver* és *DataSender* modulok egyetlen feladata, hogy bizonyos, véletlenszerű időközönként adatforgalmat generáljanak egy szintén véletlenszerűen választott csomópont felé, illetve kezeljék a fogadott adatokat. Ez, a hálózati forgalom szimulálására szolgáló metódus széles skálán paramétrezhető a csomagok számával, méretével, küldések között eltelt idő eloszlásával, és választható kommunikációs partnerek csoportjával. Fogadó oldalon a modul méri a beérkezett/elvesztett csomagok számát, QoS paramétereket, késleltetést és jittert.

Az *Addressbook* modulban valósítottuk meg a különböző adatstruktúrákat és a hozzájuk tartozó metódusokat, melyek az MA adatbázisának az alapját képezik. A mobilok véletlenszerű mozgását, helyben maradását a *Move* komponens irányítja. A *Helper* részben található meg az összes, egyéb, szigorúan egy másik komponenshez sem tartozó függvény és objektum.

6. ábra
PMIP OMNET++-ban elkészített szimulációjának struktúrája





7. ábra
A szimulációban használt teszthálózat és a mobilok kezdeti eloszlása

8. ábra (középen)
MIPv4, PMIP pure és PMIP extended összehasonlítása a jelzés forgalom szempontjából

9. ábra (alul)
PTMIP 1, 3 és 5 tracking handoveres változatának összehasonlítása jelzésforgalom szempontjából

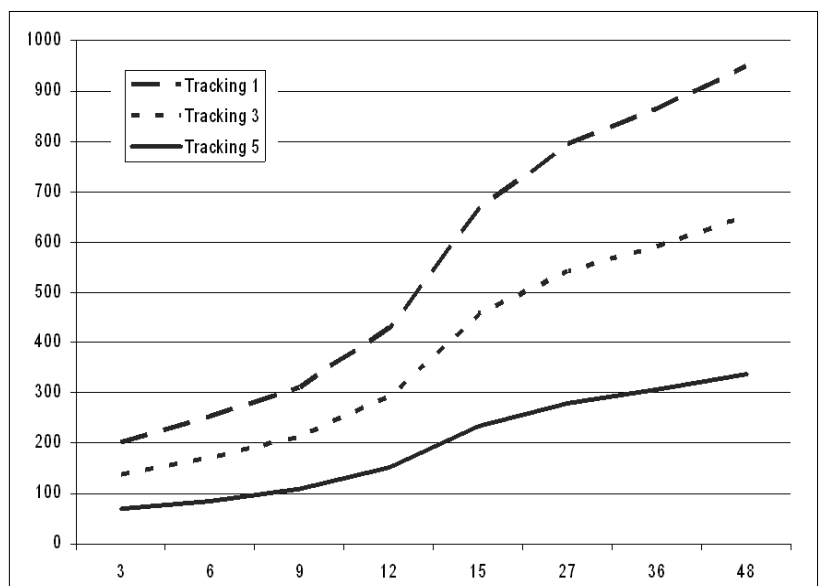
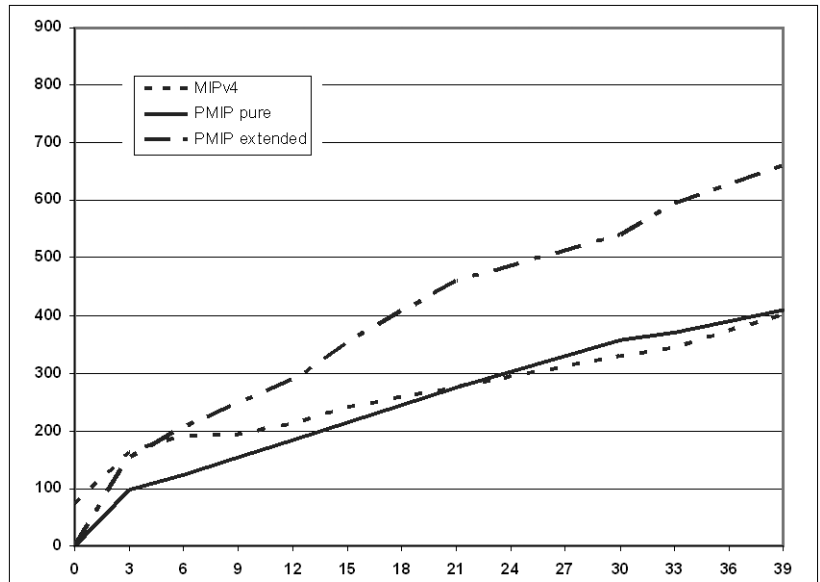
Létrehoztunk egy virtuális teszthálózatot (7. ábra) 9 MA-val és 9 MN-el, melyen a szimulációs programot futtattuk. PMIP két különböző változatát, PMIP pure (P-PMIP), PMIP extended (E-PMIP) és a hagyományos, korábban elkészített szimulációval, a Mobile IPv4-et vizsgáltuk a sok lehetőség közül. A PMIP-et, a korábban bemutatott CMFS protokoll implementációjával, a MIPv4pedig az RFC3220-ben definiált ajánlással teljesen megegyező szimulációs megvalósításával futtattuk.

A szimulációt különböző mobilitási paraméterek mellett, szeparáltan futtattuk. Minden csomópont Poisson-eloszlás szerint generálta a kéréseket egy véletlenszerűen kiválasztott csomópont felé, aminek 80%-a mobil állomás volt. A mobilitási rátát (a handoverok száma a beérkezett kérések között) növeltük és megvizsgáltuk a protokoll teljesítőképességét a jelzési üzenetként elküldött kbyte-ok függvényében.

A szimuláció eredményét a 8. ábra mutatja. Alacsony mobilitási értéknél az E-PMIP jobban teljesít még a MIPv4-nél is, de a mobilitási ráta növekedésével egyre növekszik a hozzá kapcsolódó jelzésforgalom. Ez annak köszönhető, hogy a QoS, és azon belül legfőképpen a nulla csomagvesztés biztosításához több küldési műveletet kell elvégeznie az E-PMIP-nek.

MIPv4-et helyettesítő P-PMIP nagyszámú rendileg megegyező terhelést jelent a hálózatban.

A szimulációban megvalósítottuk a korábban bemutatott PTMIP-et is és különböző tracking handover számokkal vizs-



gáltuk. A 9. ábrán jól látható, hogy a mobilitási paraméter növekedésével egyre nő a különbség a változatok között. A legtöbb tracking-et végrehajtó megoldás okozza a legkisebb jelzéseköltség-növekedést a handoverek gyakoriságának növekedése esetében.

Az „ízeltől” bemutatott szimulációk mellett további érdekes méréseket lehetne végezni sokféleképpen, sokféle protokollra az általunk fejlesztett OMNET++ programcsomagban. Legfontosabb konklúzió az eredmények alapján azonban ennyiből is levonható, azaz a protokoll működőképes, CMFS-ben a PMIP megvalósítás nem teljesít rosszabbul a MIPv4nél (PMIP pure), illetve minimális jelzéseköltség-többlettel opcionálisan QoS is biztosítható IP alapú mobilitás felett.

5. Összefoglalás

Ebben a munkában egy olyan IP mobilitást megvalósító protokollt mutattunk be, mely teljes mértékben szakít a korábbi elképzelésekkel és architektúrákkal. A teljes mobilitáskezelést és algoritmusokat maga a mozgó entitás határozza meg. A legnagyobb vívmány, hogy az általunk bemutatott módszer lehetővé teszi többféle IP mobilitást kezelő protokoll megvalósítását – a hálózat újrakonfigurálása nélkül. Mivel minden mobil saját mozgásához adaptált rendszert tud fenntartani, így úgy gondoljuk, hogy megfelelő algoritmussal a jelzésforgalom az eddigi megoldásoknál hatékonyabban minimalizálható.

Specifikáltunk egy lehetséges protokollüzenetet, illetve szimulációs környezetben bemutattuk algoritmusunk működőképességét IP alapú hálózatban. Megjegyezzük, hogy más kommunikációs hálózatokban is alkalmazható a koncepció. Az alapvető klasszikus mobilitáskezelő protokollok kliensbázisú megfelelőjére példákat mutatunk be, kaput nyitva a megoldásunkban rejlő lehetőségekre.

Az elkészült szimulációban további vizsgálatokat tervezünk, komplex, matematikailag kidolgozott algoritmusok tesztelésével, a későbbiekben pedig egy valós teszt-környezetet is szeretnénk felépíteni. Mivel a kliensbázisú mobilitás, szemben a klasszikus mobilitáskezelő rendszerekkel, nem igényel komplex hálózati struktúrát és kliens oldalról fejleszthető, úgy gondoljuk, hogy elterjedhet szélesebb körben.

A szerzőkről

KOVÁCS BENEDEK M.Sc. diplomáját 2006-ban szerezte meg a Budapesti Műszaki és Gazdaságtudományi Egyetemen, mérnök-informatikus szakon. Jelenleg a Matematika és Számítástudományok doktori iskola doktorandusza a Matematikai Analízis Tanszéken és tagja a High Speed Network (HSN) Laboratóriumnak. Kutatási területe az IP mobilitás, túlterhelés-védelem és intenzitásbecslés a telekommunikációban, valamint dinamikai rendszerek paramétereinek becslése.

FÜLÖP PÉTER 2005-ban szerezte meg M.Sc. diplomáját a BME mérnök-informatikus szakán. Jelenleg az Informatikai Tudományok doktori iskola doktorandusza a Híradástechnikai Tanszéken. Kutatási területei az IP mobilitás, heterogén mobil hálózatok együttműködése, valamint a mozgásmodellezés cellás mobil hálózatokban.

Irodalom

- [1] Kovács B., Fülöp P., Imre S., “Extended Mobility Management Framework”, Proc. of 6th Computer Information Systems and Industrial Management Applications Int. Conference (CISIM 2007), pp.191–196.
- [2] A.T. Campbell, J. Gomez, A.G. Valkó, “An Overview of Cellular IP”, IEEE, 1999, pp.29–34.
- [3] C. Castelluccia, “A Hierarchical Mobile IP Proposal”, Inria Technical Report, 1998.
- [4] Kovács B., Szalay M., Imre S., “Modelling and Quantitative Analysis of LTRACK – A Novel Mobility Management Algorithm”, Mobile Information Systems, Vol. 2, No.1, 2006, pp.21–50.
- [5] W. Ma, Y. Fang, “Dynamic Hierarchical Mobility Management Strategy for Mobile IP Networks”, IEEE Journal of Selected Areas In Comm., 2004.
- [6] C.E. Perkins, “Mobile IP”, IEEE Communications Magazine, 1997.
- [7] W. Fritsche, F. Heissenhuber, “Mobile IPv6 – Mobility Support for the Next Generation Internet”, IAB GmbH, 2000.
- [8] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, L. Salgarelli, “A Hierarchical Mobile IP Proposal”, Inria Technical Report, 1998.
- [9] S. Das, A. Misra, P. Agrawal, S.K. Das, “TeleMIP: Telecommunications-Enhanced Mobile IP Architecture for Fast Intradomain Mobility”, IEEE Personal Communications, 2000, pp.50–58.
- [10] Szalay M., Imre S., “Hierarchical Paging – A novel location management algorithm”, ICLAN’2006 International Conference on Late Advances in Networks, Paris, 6-8 Dec. 2006.
- [11] K.D. Wong, W.W. Lee, “Likelihood of Lost Binding Updates when Mobile Nodes Move Simultaneously”, 3rd Int. Conf. on Advances in Mobile Multimedia, MOMM 2005, pp.9–19.
- [12] Kovács B., Fülöp P., Imre S., “Mobility Management Algorithms for the Client-driven Mobility Frame System – Mobility from a Brand New Point of View”, MOMM 2008 (submitted).
- [13] <http://www.omnetpp.org/>