



**Budapesti Műszaki Egyetem**  
**Távközlési és Telematikai Tanszék**  
**Nagysebességű Hálózatok Laboratóriuma (HSN Lab)**

# Internet mobilitás szabvány teljesítményvizsgálata kísérleti hálózaton

TDK Dolgozat

Félegyházi Márk, Szabó Csanád, Timár Veronika

IV. évf. villamosmérnök-hallgatók

Konzulensek: Dr. Halász Edit, egy. docens (Távközlési és Telematikai Tanszék),

Valkó András, Gaál Balázs, Zömbik László (Ericsson Traffic Lab).

**1999. október**

# Tartalomjegyzék

<b>1. BEVEZETÉS</b> .....	<b>2</b>
<b>2. IRODALMI ÁTTEKINTÉS</b> .....	<b>3</b>
<b>3. AZ INTERNET MOBILITÁS ALAPELVEI</b> .....	<b>4</b>
<b>4. A MOBILE IP PROTOKOLL</b> .....	<b>6</b>
4.1. ÚJ FOGALMAK ÉS ALKOTÓELEMEK .....	6
4.2. A MOBILE IP SZABVÁNY MŰKÖDÉSE .....	8
<b>5. A MOBILE IP SZABVÁNY MEGVALÓSÍTÁSAI</b> .....	<b>9</b>
5.1. MEGVALÓSÍTÁSOK ÖSSZEHASONLÍTÁSA.....	9
5.2. A SUN MICROSYSTEMS-FÉLE MEGVALÓSÍTÁS.....	11
<b>6. KUTATÁSI CÉLKITŰZÉS</b> .....	<b>12</b>
<b>7. MÉRÉSI EREDMÉNYEK</b> .....	<b>12</b>
7.1. MÉRÉSI ELRENDEZÉS .....	12
7.2. VALIDÁCIÓS MÉRÉSEK.....	14
7.3. ÁLTALÁNOS VIZSGÁLATOK A MOBILE IP TULAJDONSÁGÁVAL KAPCSOLATBAN ...	17
7.4. MÉRÉSEK HANDOVER ALATT.....	19
<b>8. ÖSSZEGZÉS</b> .....	<b>25</b>
<b>FELHASZNÁLT IRODALOM</b> .....	<b>26</b>
<b>FÜGGELÉK A</b> .....	<b>28</b>
<b>FÜGGELÉK B</b> .....	<b>29</b>
<b>FÜGGELÉK C</b> .....	<b>30</b>

# 1. Bevezetés

Az ezredvégre a számítástechnika a legdinamikusabban fejlődő tudományterületek egyikévé vált. Az Internet által nyújtott szolgáltatások és információk ma már szinte nélkülözhetetlenek úgy az átlagember, mint a szakemberek számára. A világháló legnagyobb előnye, hogy ezeket a szolgáltatásokat és információkat a világ bármely pontjáról, egyetlen számítógép segítségével elérhetjük. A számítógépek kapacitása rohamosan nő a felhasználói igényekhez igazodva. Mind többen és többen akarnak hozzáférni a világméretű hálózaton elérhető információkhoz.

Az Internet mellett a kommunikációs ipar másik dinamikusan fejlődő területe a mobil távközlés. A mozgás közbeni kommunikáció igénye egyre erősebb, hiszen a mindennapi életünk is egyre dinamikusabb. Ennek köszönhető, hogy a mobil hangkommunikáció népszerűsége egyre nő, és kiterjedtsége egyes országokban már megközelíti, vagy el is éri a hagyományos telefonhálózatét. A mobil Internet hozzáférés igénye ezzel szemben még csak most, az Internet jelentőségének növekedésével kezd jelentkezni.

A globális adatkommunikációs hálózat, az Internet szerepe várhatóan tovább fog nőni. Az Internethez való hozzáférést a jövőben akkor sem szeretnénk majd elveszíteni, ha éppen úton vagyunk. Már ma is sok felhasználó otthonában is dolgozik a számítógépén, majd munkahelyére érve tovább használja a gépet. Szeretné, ha az utazással töltött idő sem veszne kárba. Mindenhol és minden pillanatban hozzá akar férni a globális kommunikációs rendszerhez.

A mobilitást az Internet jelenlegi architektúrája nem támogatja. Az Internet Protokollt (IP) [RFC791], [STE94] eredetileg helyhez kötött számítógépek közötti kommunikációra találták ki, ezért nem tudja kezelni az alhálózatok között mozgó, de identitását megőrző gépet. Az alapvető problémát a címzés jelenti, hiszen az Internetben a számítógépek címzése hierarchikus. Mint a harmadik fejezetben részletesen meg fogjuk mutatni, a hierarchikus címzés nehézséget jelent mobil kommunikáció esetén, és az Internet mobilitás megoldásoknak elsősorban ezzel a nehézséggel kell megküzdeniük.

A mobilitás problémája valójában két kérdéskört takar. Az egyik a hordozhatóság, amikor az ember a számítógépét különböző helyeken kívánja használni, de míg egyik helyről a másikra utazik, addig nem kapcsolódik az Internethez. A másik a szó szoros értelmében vett, a cellás telefonokhoz hasonló mobilitás [MOU92], amikor a felhasználó utazás közben is fenntartja az Internetes kapcsolatot. Ehhez vezeték nélküli összeköttetésre van

szükség, ami miatt a hordozhatósághoz képest további problémák jelentkeznek. A vezeték nélküli környezet legnagyobb kihívása, hogy mozgás közben gyakori bázisállomásváltás történik, hiszen egy állomás csak viszonylag kis területet fed le. A bázisállomások közötti átlépést hívjuk handovernek vagy handoffnak.

Dolgozatunk célja az Internet mobilitásra 1996-ban kidolgozott szabvány [RFC2002] vizsgálata vezeték nélküli környezetben. Vizsgálatunk középpontjában a handover jelensége áll, illetve a handovernek a folyamatban lévő adatkapcsolatokra gyakorolt hatása. A vizsgálatok jelentőségét az adja, hogy a szabvány viszonylag új volta miatt még csak kevés vezeték nélküli környezetben is érvényes teljesítményvizsgálatot végeztek el. Az irodalomban elérhető mérési eredményeket a második fejezetben foglaljuk össze.

A dolgozat felépítése a következő. A bevezetés után a második fejezetben rövid irodalmi áttekintést adunk a mobil Internet eddigi teljesítményvizsgálatairól. A 3. fejezetben az Internet mobilitás problémáját és megoldásának elvét mutatjuk be, a 4. fejezetben a Mobile IP szabványt ismertetjük. A következő fejezetben hasonlítjuk össze a szabvány már működő megvalósításait és megmutatjuk az általunk használt megvalósítás előnyeit. Dolgozatunk középpontjában a 7. fejezet áll, ahol mérési eredményeinket ismertetjük. Végül 8. fejezetben összegezzük az általunk levont következtetéseket.

## **2. Irodalmi áttekintés**

A Mobile IP szabvány 1996-ban született meg. Eddig csak kísérleti megvalósításai születtek különböző kutatóintézetekben.

Az egyes megvalósításokat több alkalommal vizsgálták, de az eddigi vizsgálatok elsősorban a megvalósítások validációjára, és szabvány-konformanciájára vonatkoztak. Az Internet Engineering Task Force (IETF) gyakorlatának megfelelően a vizsgálatokat úgynevezett interoperability (együtműködési) tesztek keretében végezték el. Ennek lényege, hogy az egyes megvalósítások együtműködési képességét az összes lehetséges kombinációban megvizsgálják. Ez a vizsgálat lehetővé teszi, hogy a szabványosításban esetleg nem megfelelően rögzített, és különböző megvalósításokban másképpen értelmezett kérdéseket kiszűrjék, nem ad azonban lehetőséget a teljesítmény-paraméterek vizsgálatára.

A Mobile IP teljesítményvizsgálatával csak az utóbbi időben kezdtek el foglalkozni, ezek közül kettőt említenénk meg, mindkettő 1999-ben jelent meg.

A National University of Singapore munkatársai foglalkoznak a Mobile IP teljesítményvizsgálatával, de ők a tesztekét vezetékes környezetben végezték el [CHO99]. Megvizsgálták a szabvány késleltetési paramétereit és átviteli sebességét.

Cvetkovic és kollégái a University of Sheffielddről múlt hónap végén mutatták be munkájukat a New Orléans-i konferencián [FIK99]. Nevükhöz fűződik az első vezeték nélküli hálózatban elvégzett teljesítményvizsgálat. Méréseik során három különböző elgondolást teszteltek, melynek alapján a mobil számítógép észlelte, hogy új cellába lépett. Figyelmük a TCP és UDP késleltetésvizsgálatára terjed ki handover alatt.

Ellenőrzésképpen mi is elvégeztük az előbb felsorolt méréseket. Munkánk fő célja azonban az átviteli sebesség mérése volt handover esetén. Ez azért lényeges, mert a felhasználó számára ez a legfontosabb paramétere a kommunikációnak. Az átviteli sebesség változását több szempont alapján vizsgáltuk, erről részletesen a 7. fejezetben számolunk be.

A TCP protokoll viselkedését vezeték nélküli környezetben, de handover nélkül több cikk is részletesen tárgyalja. A problémát itt az jelenti, hogy a vezeték nélküli átvitel miatt bithibák jelentkeznek, amik csomagvesztésben nyilvánulnak meg. A TCP vezetékes környezetre optimalizált, így a csomagvesztést torlódásként értelmezi és csökkenti az átviteli sebességet. A problémára több megoldás javaslat is született, amelyeket [KAT97] foglal össze. Ez a probléma kívül esik jelenlegi munkánk tárgyán.

### **3. Az Internet mobilitás alapelvei**

Az IP-hálózatok [STE94] kialakításakor nem vették figyelembe, hogy a hálózatra csatlakozó gép megváltoztathatja a helyét. Az alapvető nehézséget a hierarchikus címzés jelenti. A hierarchikus címzésre azért van szükség, hogy a hálózat nagy méreteket érhesen el anélkül, hogy a benne működő útvonalválasztók (routerek) minden egyes számítógépet ismernének. A hierarchikus címzésnek azonban az a következménye, hogy a számítógépek címe egyszerre tölti be a globális azonosítót, és a gép helyzetét jelző paraméter szerepét. Ez konfliktust jelent, ha mobil számítógépekről van szó, hiszen az azonosító jellegből az kö-

vetkezik, hogy a címnek mozgás közben is állandónak kell maradnia, a helyzetmeghatározáshoz azonban minden mozgáskor változnia kellene.

- Ha a mobil gép új hálózathoz kapcsolódva új IP címet kap, akkor ezzel együtt a hozzátartozó TCP [RFC793] összekötés-azonosító is megváltozik. Ez viszont azt eredményezi, hogy megszakad az összes TCP összeköttetés.
- Ha a mobil gép megtartja az eredeti IP címét, akkor viszont elmozdulása után a rendszer nem tudja utána küldeni az érkező csomagokat.

Azért hogy a mobilitás problémáját kezelni tudják, logikai hozzárendeléseket társítottak az IP címekhez [BHA96]: ez az eljárás az úgynevezett “two-tier addressing”. Ez az elgondolás feloldja azt a nehézséget, amely egy Internet cím kettős szerepével függ össze.

A címzési eljárás a következőképp alakul. Minden mobil számítógéphez logikailag két Internet címet rendelünk. Az egyik cím betölti a globális azonosító szerepét, és a mobilitástól függetlenül mindig állandó marad. A másik cím ideiglenes jellegű, és minden mozgást követően megváltozik.

Ahhoz hogy egy mobil gépnek adatcsomagot küldjünk, csak az állandó címét kell ismernünk. A mobil gépek állandó címéhez egy adatbázis rendeli a mindenkori ideiglenes címet. Ennek az adatbázisnak a segítségével irányítja át a hálózat a csomagot a mobil gép aktuális címére. Minden csomag, melyet így a mobil gépnek küldenek, two-tier címzést használ, azaz mindkét cím szerepel benne. A csomag irányítása során csak az ideiglenes rész játszik fontos szerepet: ez alapján jut el a csomag a mobil géphez. Ettől kezdve, a cím első fele figyelmen kívül marad, már csak a második mezőt, az állandó (azaz otthoni) címet használják a további feldolgozás során. Ez a módszer lehetővé teszi, hogy kívülről úgy látszódjon, hogy a mobil gép mindvégig megtartotta a otthoni címét.

Abban az esetben, ha a mobil gépről szeretnénk adatot küldeni egy egyszerű hálózatba kötött gépnek, akkor nincs szükség a fent említett megoldásra, hisz ekkor az IP hálózatoknál ismert útválasztási eljárások végzik a csomagok irányítását. Ha viszont a címzett gép is mobil, akkor ismét a two-tier címzést kell alkalmaznunk.

A fentiekben ismertetett two-tier addressing elvet, melyet [BHA96] ismertet, több mobilitás megoldás is követi [TER95], [IOA91], [MYS97]. Ezt az elvet használja a mobilitásra elfogadott Internet szabvány, a Mobile IP [RFC 2002] is, amelyet a következő fejezetben ismertetünk.

## 4. A Mobile IP protokoll

### 4.1. Új fogalmak és alkotóelemek

Mielőtt behatóbban részleteznénk a Mobile IP szabványt, meg szeretnénk ismertetni az Olvasót azokkal az új kifejezésekkel, amelyeket a Mobile IP-vel kapcsolatos témákban használni fogunk. A fogalmak részletes magyarázata megtalálható a szabvány leírásában [RFC2002].

- Mobile Host vagy Node (MH vagy MN):

Az a számítógép, mely a csatlakozási pontját változtatja, anélkül hogy az IP-címét változtatná. Ezért a Mobile Node képes úgy kommunikálni az Internetre csatlakozó gépekkel, hogy közben egy másik hálózatra lép át és kapcsolata nem szakad meg a vele kommunikáló gépek felé.

- Home Agent (HA):

Minden Mobile Node-nak létezik egy alhálózata, melyben neki az IP-címet kiosztották. A Home Agent az a gép, amelyik észleli az adott alhálózaton (pl. Ethernet) a Mobile Node-nak küldött csomagokat és azt eljuttatja a Mobile Node aktuális tartózkodási helyére. A Home Agent mindig rendelkezik azzal az információval, hogy a Mobile Node a hálózatban éppen hol tartózkodik, azaz megfelel az előző fejezetben említett adatbázisnak.

- Foreign Agent (FA):

A Foreign Agent az a számítógép, amelyhez megérkeznek a Mobile Node-nak szánt csomagok a Home Agent-től. Mivel a Foreign Agent ahhoz az alhálózathoz csatlakozik, ahol a Mobile Node jelenleg tartózkodik, és regisztráció útján tudomást szerzett a Mobile Node-ról a neki szánt csomagokat képes annak továbbítani.

- Correspondent Node (CN):

A Mobile Node-al kommunikáló számítógépek gyűjtőneve, amely lehet helyhez kötve, de lehet egy másik Mobile Node is.

- Node:  
Egy számítógép vagy útválasztó (router).
- Tunnel:  
„Alagút”, amelyen keresztül a Home Agent elküldi a Mobile Node-nak címzett csomagokat. Az alagút lényege, hogy az IP csomagokat teljes egészében, tehát a fejléccel együtt beteszik egy másik IP csomagba, ezzel lehetővé téve, hogy a csomag, ne az eredeti címzés szerint haladjon a hálózatban. Ezt nevezzük “encapsulation”-nak. Az egyik ilyen módszer például az “IP in IP encapsulation” [RFC2003].
- Care-of Address:  
Ez a kifejezés a Mobile IP protokollban a mobil számítógép mindenkor ideiglenes, azaz aktuális címét jelöli. A feljebb említett alagút egyik végpontja a Home Agent, a másik végpontja a care-of address [RFC2002].
- Home Address:  
A Mobile Node állandó IP-címe, amelyre a neki küldött csomagokat címezik. Ezen a címen érhető el a Mobile Node, bárhol is tartózkodik a hálózatban.
- Home Network (HN):  
Az az alhálózat, amelyhez a Mobile Node látszólagosan mindig hozzá van csatolva. A vele kommunikáló gépek a Mobile Node állandó IP-címére küldik a csomagokat, amelyeket majd a Home Agent juttat el neki a pillanatnyi tartózkodási helyére.
- Foreign Network (FN):  
Minden egyéb részhálózat, mely nem egyezik meg a Home Networkkel.
- Registration:  
Az a folyamat, melynek során a Mobile Node jelzi a Home Agent-nek, hogy melyik alhálózatban tartózkodik. A bejelentkezést (Registration Request) először elküldi Foreign Agentnek, amennyiben az létezik, amely ezt az üzenetet továbbítja a Home Agent-nek. A Home Agent nyugtázza (Registration Reply) ezt a bejelentkezést. A negatív nyugtázás esetén a Home Agent a válaszban küldi a hiba okát is.

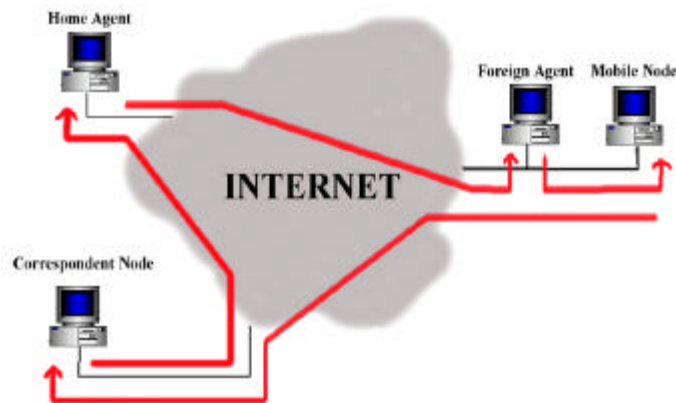


- Kereső üzenetek:

Agent Advertisement Message: az Agent-ek ezzel a periódusonként elküldött üzenetükkel jelzik jelenlétüket.

Agent Solicitation Message: az előbbi üzenethez hasonlóan ez is periódusonként jelenik meg a hálózaton. Ezzel a Mobil Node keresi a kapcsolatot az Agent-ekkel.

#### 4.2. A Mobile IP szabvány működése



4.1 ábra – Mobile IP elrendezése

A következőkben lépésről-lépésre, ám nem minden részletre kitérve mutatjuk be a Mobile IP protokoll [RFC2002] működését. Tekintsünk egy mobil gépet, amely új hálózathoz kíván csatlakozni. Első lépésben a mobil állomás Agent Solicitation üzeneteket küld a helyi hálózaton levő Foreign Agenteknek. A Foreign Agent-ek ugyanakkor periodikusan hirdetik jelenlétüket Agent Advertisement üzeneteken keresztül.

A mobil gép és az Agent-ek egy közös broadcast vagy multicast csatornára juttatják el az Agent Solicitation, illetve Agent Advertisement üzenetüket, ezáltal tudomást szereznek egymásról. A mobil gép a Foreign Agent segítségével kapja meg az ideiglenes címét (Care-of address). Ezt a címet közölnie kell a Home Agent-jével.

A 4.1 pontban magyarázott fogalmak használatával bemutatjuk, milyen utat tesz meg egy csomag, miközben eljut a forrásgéptől a címzett Mobil Host-ig. A csomag címében

eredetileg a mobil gép állandó címe áll, hiszen a feladó semmit sem tud a Mobil Host jelenlegi tartózkodási helyéről (lásd 4.1 ábra). A Home Agent ezt a címet kiegészíti annak a Foreign Agent-nek a címével, ahová a Mobil Host utoljára bejelentkezett. Ezt a problémát oldja meg az encapsulation azzal az egyszerű módszerrel, hogy a Home Agent a beérkezett csomagot „becsomagolja”, és ellátja egy külső fejléccel, amelynek a címzettje már a Foreign Agent lesz. Ezután a csomag megérkezik a Foreign Agent-hez, ahol a külső fejléc eltávo-lításra kerül, hiszen a továbbiakban már nem lesz rá szükség. Ettől kezdve már csak egy egyszerű, saját hálózaton belüli routolás szükséges, hogy a csomag a címzett Mobil Host-hoz érjen. Az 4.1 ábra az egész csomagtovábbítási mechanizmust szemlélteti.

## **5. A Mobile IP szabvány megvalósításai**

### **5.1. Megvalósítások összehasonlítása**

A Mobile IP-t eddig csak kísérleti szinten valósították meg különböző egyetemeken és kutatóintézetekben, általánosan használható kereskedelmi változata még nem került forgalomba.

A Mobile IP protokollt megvalósító programokat Linux, FreeBSD, WindowsNT és Solaris/Unix operációs rendszerek alá írták. Idáig Linuxos környezet alatt dolgoztunk, ezért a Linuxon fejlesztett szoftverek ismerkedésével foglalkoztunk behatóbban. A különbség azonban nem az operációs rendszerekben, hanem az egyes szoftvercsaládokban rejlett. A rendelkezésre álló szoftverek közül az általuk nyújtott különböző szolgáltatások alapján a Sun Microsystems [SUN99] programja mellett döntöttünk. A döntésünk fő szempontjait az 5.1 táblázat tartalmazza.

Implementáció	Linux Kernel verzió	Szoftver milyen formában	Létezik szabad forráskód	Plusz szolgáltatások
Stanford University	2.0.36	Daemon		
National University of Singapore	2.0.34		Igen	Multicast, QoS
Hewlett Packard	Független			Biztonság
Sun Microsystems	2.2.x	Driver	Igen	Biztonság
Helsinki University of Technology	2.2.x	Daemon		Biztonság

**5.1 táblázat – Mobile IP megvalósítások jellemzői**

A Hewlett-Packard megvalósítás megfelelő volt a Linux kernel szempontjából, de a legjobbak a Sun és a Helsinki UT szoftverek voltak, mert azok támogatták a legújabb kernelt [LDP]. A Stanford University és a Helsinki UT programja daemon-ként futott, de számunkra itt is a driverként futó Sun program volt a legjobb, mert így az könnyen módosítható. A megvalósítások közül az egymás közötti teszteknel (“interoperability”) a National University of Singapore, a Sun és a Helsinki UT szoftverei hasonlóan megfelelő eredményt mutattak

Az implementációk abban különböznek egymástól, hogy más szinten valósítják meg a Mobile IP által megkövetelt funkciókat és különböző plusz szolgáltatásokat nyújtanak. A Sun Microsystems implementációjára tehát azért esett a választás, mert ez a leginkább rugalmas változtatás tekintetében, támogatja az új Linux kernelt, driverként fut, és szabadon hozzáférhető forráskódja rendelkezésre állt. Nem utolsósorban az együttműködést vizsgáló teszteknel is megfelelőnek bizonyult és programjukhoz a legrészletesebb dokumentációt kínálták fel.

## 5.2. A Sun Microsystems-féle megvalósítás

A Sun Microsystems implementációja [SUN99] két külön szoftverben valósítja meg a Mobile IP által megkövetelt funkciókat: a Home Agent és a Foreign Agent funkciókat a *mipagent* program futtatásával érhetjük el, míg a Mobile Node oldalán a *mipmn* program fut.

A *mipagent* programot a konfigurációs file megfelelő beállításával lehet Home Agent-ként vagy Foreign Agent-ként futtatni. Ennek a programnak fő funkciói a következők:

- az Agent Advertisement Message kiadása: ezzel jelzik az Agent funkciót betöltő gépek, hogy rendelkezésre állnak,
- az aktuális Mobile Node kapcsolatok nyilvántartása, azaz Foreign Agent esetén éppen melyik mobil számítógép csatlakozik hozzá, Home Agent esetén pedig hol van a hozzá tartozó Mobile Node,
- a Registration Request üzenetek megválaszolása: bejelentkezés kérésének kiértékelése és elfogadása vagy el nem fogadása.

A *mipmn* program feladatai:

- Agent Solicitation Message kiadása: így keres a Mobile Node bázisállomást,
- Az Agent Advertisement Message-k figyelése: a program figyel a lehetséges bázisállomások által adott üzenetekre,
- Movement detection: az a módszer, ahogy a Mobile Node felismeri azt, hogy bázisállomást kell váltani,
- Registration Request es Registartion Renewal Message-k küldése: a bejelentkezést kérő ill. egy meglévő kapcsolatot megújító üzenetek.

A programok a beállítható paramétereket konfigurációs file-okban tárolják. Agent oldalon lényeges az Agent Advertisement Messagek periódusának beállítása. Meg kell adni továbbá a mobilitást támogató interfészeket és Home Agent esetén a hozzá tartozó Mobile Node-okat. A Mobile Node részéről fontos az Agent Solicitation Message periódusának a beállítása és két másik időzítő meghatározása. Az egyik a Solicitation Threshold, amely megmondja, hogy a Mobile Node-nak mennyi idő után kell Agent Solicitation Message-t küldeni a Foreign Agentnek, ha arról nem hallott ezen időn belül. A másik az Agent Expire Threshold, ami meghatározza a mobilnak, hogyha az előző üzenetre sem kapott választ, akkor mennyi idő után tekintse a kapcsolatot semmisnek. Ezek az időzítések fontos szere-

pet kapnak a csomagvesztés kialakulásánál. Ha értéküket alacsonyra választjuk, azzal gyorsítható a Mobile IP. Ebben az esetben azonban több üzenettel terheljük a hálózatot, mert mind a mobil mind az Agent-ek gyakrabban küldenek Solicitation ill. Advertisement üzeneteket.

## **6. Kutatási célkitűzés**

Az általunk végzett tesztek a Mobile IP [RFC 2002] viselkedését kívánják bemutatni egy vezeték nélküli környezetben. Ebben a környezetben a Mobile IP speciális tulajdonságokat mutat, hiszen a protokollt alapvetően nem a mobil számítógéppel való szabad mozgás alatti kommunikációra tervezték, hanem arra, hogy lehetővé tegye a számítógép hordozhatóságát. A hordozhatóság azt jelenti, hogy a gép identitása a helyválttatás előtt és után ugyanaz, azonban e helyválttatás közben nem zajlik adatforgalom, míg a mobilitás magában foglalja ezt a kommunikációt is. Vezeték nélküli környezetben a Mobile IP várhatóan más késleltetési és csomagvesztési tulajdonságokat mutat, mint egyébként. Gondoljunk csak arra, hogy a rádiós interfész sokkal védtelenebb a fizikai rétegben keletkező bithibákra.

Különösen érdekes a csomagvesztés és a sáv szélesség változása handover esetén, mert a felhasználó csak a sebességet érzékeli az alkalmazás futtatásánál. A handoverek gyakorisága nagyban meghatározza a minőségi paramétereket.

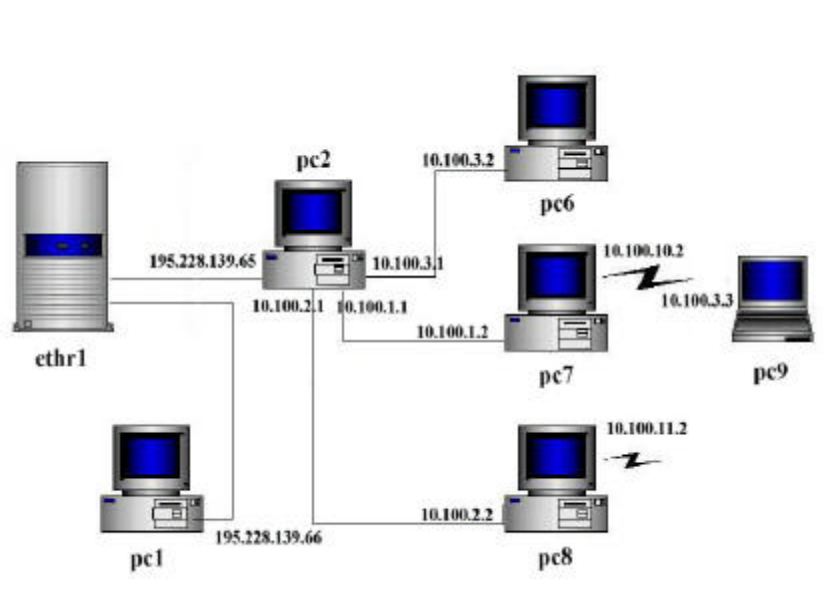
## **7. Mérési eredmények**

### **7.1. Mérési elrendezés**

A Mobile IP-vel kiegészített hálózat általános felépítése (4.1 ábra) jól modellezhető egy Home Agent-tel, egy Mobile Node-al és két bázisállomással, amelyek a Foreign Agent szerepét töltik be. Szükség van még egy Correspondent Node-ra vagyis egy csomagokat

küldő gépre. Az általunk összeállított elrendezést a 7.1 ábra mutatja. A bázisállomásokat és a Home Agent-et külön alhálózatra helyeztük el egy köztes útválasztó segítségével, amely az Internetet szimbolizálja. Ebben a konfigurációban már vizsgálhatóak a handover folyamán a Mobile IP tulajdonságai.

A hálózatunkban a számítógépek i486 gépek, 40 Mbyte memóriával, kivéve a routert, ami egy Pentium számítógép nagyobb memória- és tárolókapacitással. A bázisállomásokban és a mobil gépben egy-egy az IEEE 802.11 szabványnak [GEI99] megfelelő WaveLAN 2,4 GHz kártya [WAV98] biztosítja a vezeték nélküli összeköttetést. A kártya 13 csatornán képes kommunikálni, ezt a tulajdonságát handoveres méréseinknél használtuk ki. Kész drivert kaptunk hozzá Linux alá, amit a handover előidézéséhez egy saját programmal egészítettünk ki, mert a driver nem biztosította a kártya frekvenciájának működés közbeni változtatását. A *wper.c* programról a C függelékben írunk.



7.1 ábra - A mérési elrendezés

A hálózatban a **pc1** tölti be a Correspondent Node szerepét, a **pc7** és **pc8** a két bázisállomás, míg a **pc9** a Mobile Node. A Home Agent a **pc6**-os számítógép. A **pc2** útválasztóként (routerként) működik, ennek a segítségével valósítottuk meg a különböző alhálózatokat. Az **ethr1** egy a teszhálózattól független router, ami mindössze a **pc1** elérését teszi lehetővé.

A méréseink a Mobile IP tulajdonságait vizsgálták vezeték nélküli környezetben különös tekintettel a handover jelenségre, amelynek a hatása meghatározó egy mobil átvitel

esetén. A következő alfejezetek tartalmazzák a hálózatot először Mobile IP nélkül vizsgáló méréseinket, majd a Mobile IP tulajdonságait vizsgáljuk. A dolgozatunk legfontosabb részét a handover alatti mérések alkotják, melyeket a 7.4 fejezetben ismertettünk.

## 7.2. Validációs mérések

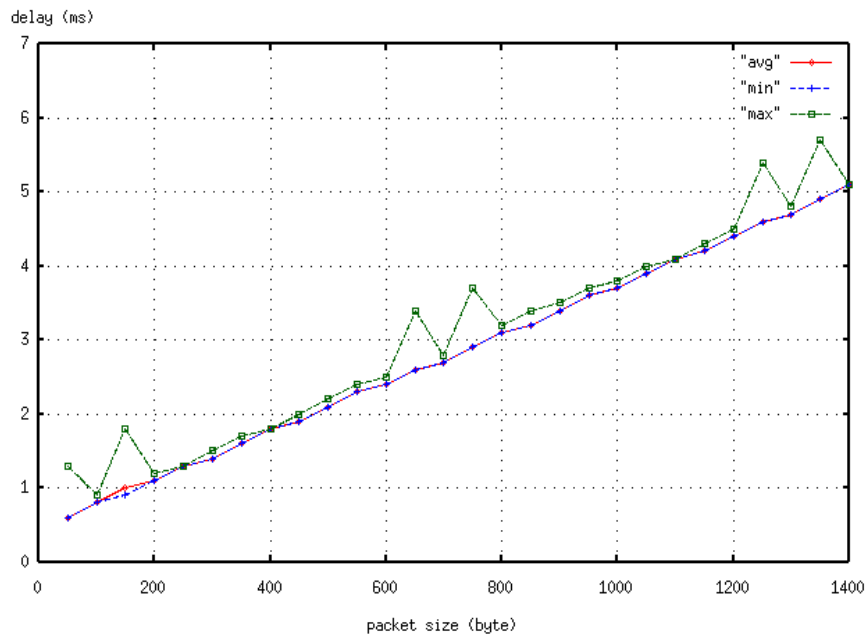
A hálózaton kezdetben érvényesítő (validációs) méréseket végeztünk a konfigurációs beállítások és a hardverparaméterek ellenőrzésére. Ezeket később összehasonlítottuk a Mobile IP futása alatt kapott eredményekkel. A validáció alatt azt kell érteni, hogy megvizsgáltuk a teszhálózat alapparamétereit, nehogy egy esetleges hardverhiba hamis mérési eredményeket okozzon.

### Routing beállítások ellenőrzése

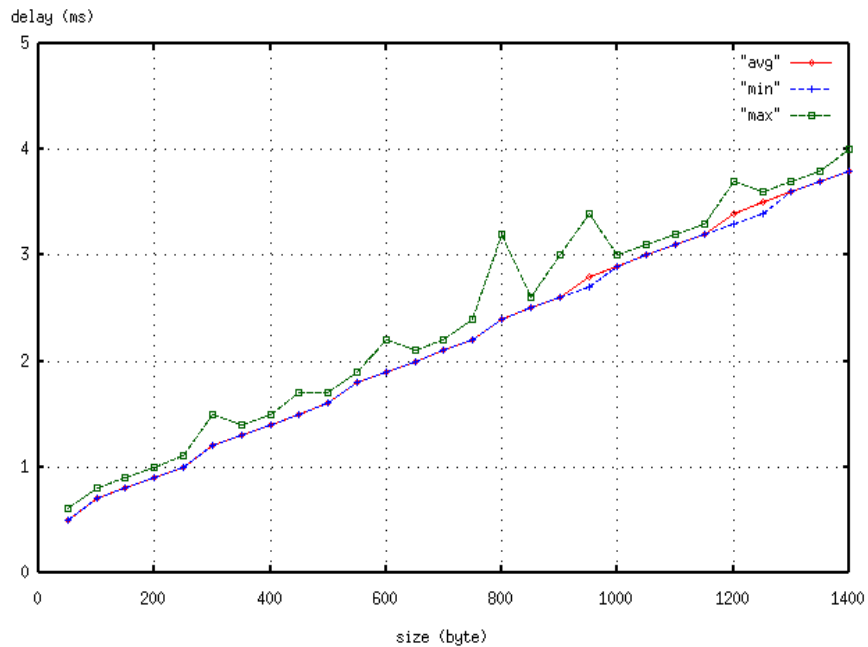
Először ellenőriztük a helyes útválasztási beállításokat, nehogy később a csomagküldésnél kerülő utak vagy hurkok alakuljanak ki. Az eredmények az A függelékben találhatóak. Ekkor ugyanis olyan késleltetések lépnének fel, amiket nem a Mobile IP okoz.

### Körülfordulási idő mérése

Elvégeztük az úgynevezett háromszögű csomagtovábbítás (triangular route) szakaszonkénti körülfordulási idejének vizsgálatát, hogy össze tudjuk hasonlítani a Mobile IP használatakor mért idővel. Erre szolgált a *packet.pl* nevű Perl szkriptnyelven írt program (vö. Függelék B). Szkriptünk a *ping [LDP]* program felhasználásával küldött másodpercenként egy csomagot, és minden 100. után 50 byte-al növelte annak méretét. Az alábbi ábrákon a csomagméret növekedésének függvényében vizsgáltuk a teszhálózatunk minden egyes összeköttetésének körülfordulási idejét. (lásd 7.2 – 7.5 ábra) Az ábrákon feltüntettük az egyes eredmények minimális, maximális és átlagos értékét. Az ábrákon jól látható, hogy az átlag és a minimum szinte egybeesik, vagyis az ettől való eltérés (amit a maximum görbe jellemez) csak ritka statisztikai jelenség. Azt vettük észre, hogy a körülfordulási idő egyenesen arányosan nő a csomagméret növekedésével. Az egyes szakaszok körülfordulási idejének összege körülbelül ugyanakkora lett, mint a Mobile IP háromszögútvonalválasztási (triangular route) teljes útra mért érték.

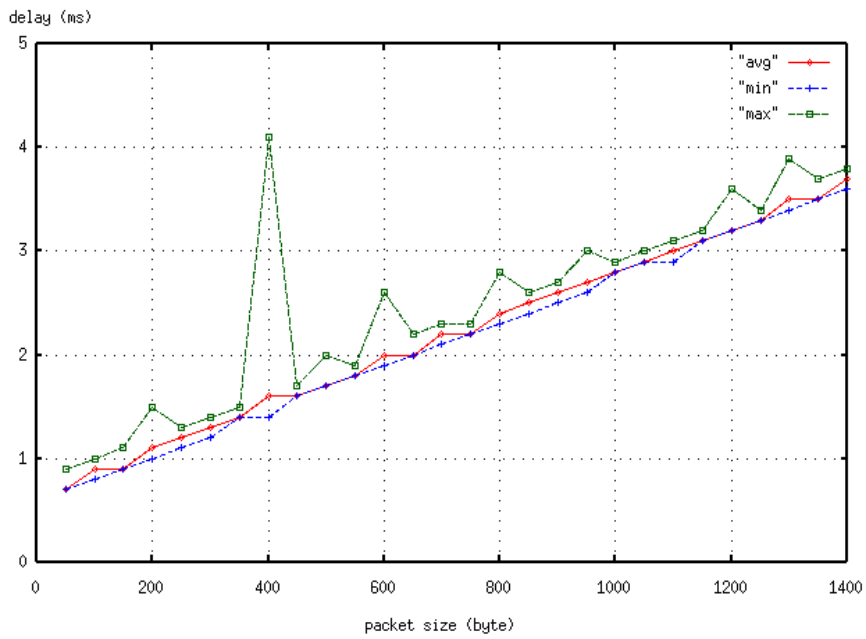


7.2 ábra Körülfordulási idő a Correspondent Node (pc1) és Router (pc2) között

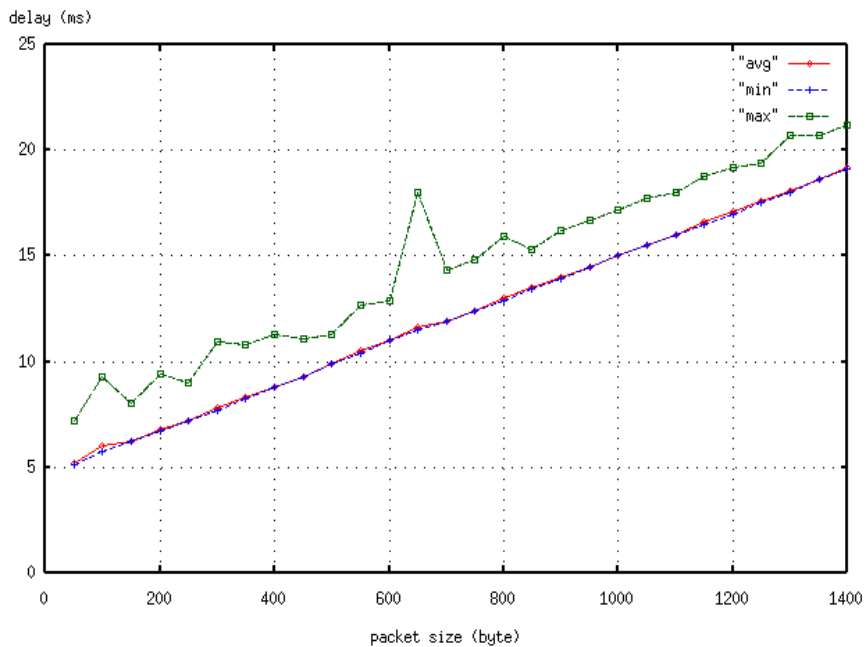


7.3 ábra Körülfordulási idő a Router (pc2) és Home Agent (pc6) között





7.4 ábra Körülfordulási idő a Router (pc2) és Base Station1 (pc7) között



7.5 ábra Körülfordulási idő a Base Station 1 (pc7) és Mobile Node (pc9) között

A fenti mérési eredmények alapján az értékek megfelelőek. A szakaszok értékeinek összegének ugyanakkorának kell lenni, mint a teljes háromszög útválasztás esetén mért körülfordulási időnek.

$T_{CN, R} + T_{R, HA} + T_{R, BS1} + T_{BS1, MN} = T_{CN, MN}$  ; ahol a  $T_{CN, MN}$  a Correspondent Node és Mobile Node között mért körülfordulási időt jelenti.

Nézzük meg a szakaszok összegét például 1000 byte csomagméretnél:

$$3.75 \text{ ms} + 2.9 \text{ ms} + 2.8 \text{ ms} + 15 \text{ ms} = 24.45 \text{ ms} \Leftrightarrow 23.8 \text{ ms}$$

Az összehasonlítás baloldalán a négy szakasz körülfordulási ideje áll, amiknek összege megegyezik a jobb oldalán álló CN – MN körülfordulási idővel Mobile IP futásakor.

A fenti eredmény igazolja, hogy a triangular route egyes szakaszainak késleltetése csak igen kis mértékben tér el a Correspondent Node és Mobile Node közötti (7.6 ábra) azonos csomagméretnél vett késleltetés eredményétől.

### **WaveLAN kapacitás**

Megvizsgáltuk a WaveLAN kártya kapacitását Mobile IP nélkül, ezáltal meg tudtuk becsülni a későbbi átvitel folyamán, hogy mekkora lesz a TCP forgalom sávszélessége. A TCP protokoll ugyanis az átvitelt a legkisebb keresztmetszetre méretezi (a nemzetközi irodalomban a “bottleneck” kifejezés használatos), ez esetünkben egyértelműen a WaveLAN általi rádiós átvitelt jelentette, hiszen a többi csatlakozás 10Mbps kapacitású. Az átvitelre átlagosan 1442.4 kbps értéket mértünk, ami körülbelül megfelel a WaveLAN rádiós interfész névleges átvitelének [WAV98].

A vizsgálatot a *ttcp* nevű TCP-csomagokat küldő és a forgalmat kiértékelő programmal végeztük. [TTCP]

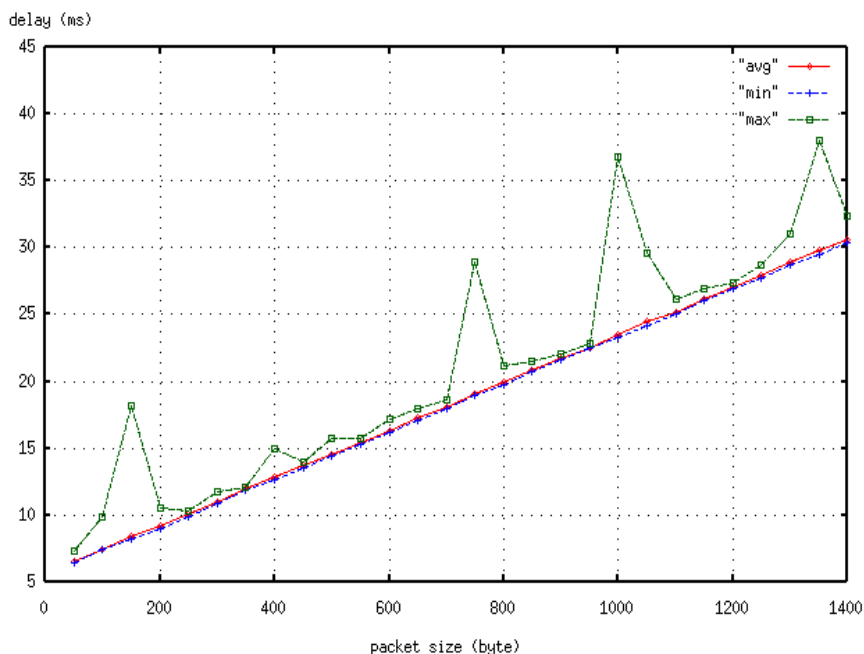
### **7.3. Általános vizsgálatok a Mobile IP tulajdonságával kapcsolatban**

Az alábbiakban azokat a mérési eredményeket ismertetjük, amelyeket a Mobile IP használatakor mértünk, de még handover nélkül. A handover teljesítmény vizsgálat eredményeit a következő alfejezetben fogjuk ismertetni.

## Triangular route késleltetés

Mobile IP használatakor lemértük a triangular routing körülfordulási idejét, hiszen a felhasználó által tapasztalt szolgáltatásminőség egyik legfontosabb paramétere a késleltetés. Az Interneten tapasztalt késleltetések Mobile IP használata esetén megnőnek, mert a két végpontú kapcsolat ebben az esetben háromszöggé egészül ki.

A 7.6 ábra a 7.2–7.5 ábrához hasonlóan a körülfordulási idő növekedését mutatja be a csomagméret növekedésének függvényében. A méréseket a *packet.pl* szkript segítségével végeztük. Az ábrán megfigyelhető, hogy a körülfordulási idő itt is lineárisan nő a csomagméret növekedésével. Megfigyelhető továbbá, hogy a Mobile IP esetén többször fordul elő nagyobb csomagkésés, mint Mobile IP használata nélkül, de az átlag most is szinte egybeesik a minimummal. Ez azt jelenti, hogy a Mobile IP használata során előfordul ideiglenes csomagtorlódás, ami egy-egy csomag késleltetését megnövelheti, de a néhány kiugróan nagy késleltetés egészében nem meghatározó.



7.6 ábra Körülfordulási idő a Correspondent Node (pc1) és Mobile Node (pc9) között

Ha az ábrán szemléltetett körülfordulási időt bármely csomagméretnél összevetjük a 7.2 – 7.5 ábrák megfelelő csomagméretnél vett körülfordulási idők összegével, akkor közel azonos eredményt kapunk. Ebből szintén azt a következtetést vonhatjuk le, hogy a Mobile IP szoftver csomagtovábbítása nem okoz lényeges késleltetést az átviteli utak késleltetéséhez képest. Feltételezhetjük tehát, hogy a későbbiekben, a handover vizsgálata során mért késleltetések valóban a handover jelenségével függnek össze.

### **Internet alkalmazások Mobile IP felett**

Általános méréseink utolsó lépéseként ellenőriztük, hogy Mobil IP felett működnek-e a hagyományos TCP kapcsolatok. Mind a *telnet*, mind az *ftp* programok hiba nélkül működtek Mobile IP felett is. Megmértük továbbá egy file átvitelekor a mérési hálózatban alkalmazott vezeték nélküli kapcsolat, tehát a WaveLAN interfész kapacitását. Ezt az eredményt összehasonlítottuk a Mobile IP nélkül mért eredménnyel és közel azonos eredményt kaptunk (1367.73 kbps), tehát a Mobile IP kihasználta a rendelkezésére álló átviteli szélességet.

### **7.4. Mérések handover alatt**

Az eddig ismertetett mérési eredmények megmutatták, hogy teszhálózatunk megfelelően modellezi a Mobile IP-vel kiegészített Internet működését, és a vezeték nélküli átvitel is a specifikációnak megfelelően működik. A továbbiakban a handover minőségével kapcsolatos méréseink eredményeit ismertetjük. Ezek az eredmények jellemzik legjobban egy vezeték nélküli átviteli rendszert, hiszen a felhasználó mozgásakor a handover gyakori lesz. Nagyon fontos, hogy a tipikus Internet alkalmazások handoverek közben is folyamatosan működhessenek, hasonlóan ahhoz, ahogy a mobiltelefonokat is használhatjuk mozgás közben.

Mivel a kísérleti hálózatunkban a gépek fizikailag közel voltak egymáshoz és tesztelési szempontból nem volt kedvező a mobil gép mozgása, ezért helyhez kötött Mobile Node-al dolgoztunk és a handover-t frekvenciaváltással oldottuk meg. Ez azt jelenti, hogy a két bázisállomás (BS1 ill. BS2) különböző frekvencián működik. A Mobile Node frekvenciaváltását a *wfreq* program oldja meg, ami a *wper* programunk egyszerűsített változata. Esetünkben ez azt jelenti, hogy a Mobile Node számára az előző bázisállomás jele többé

nem fogható, mintha az kiesett volna. Ezt a megoldást azért választottuk, mert így pontosan tudtuk irányítani a handover gyakoriságot. Számunkra az volt érdekes, hogy a handover alatt hogyan viselkednek a már meglévő kapcsolatok.

Elsősorban a TCP csomagvesztést vizsgáltuk, mert az Internet alkalmazások túlnyomó többsége a TCP szállítási protokollra épül, így ez az adat fontos minőségi paramétere a protokollnak. A handover alatt csomagvesztés történik, mert egy bizonyos időtartamig a Mobile Node nem kapcsolódik egyik bázisállomáshoz sem, így nem kapja meg a neki szánt csomagokat. Ezt a TCP a torlódás jelének veszi, és lelassul. Így a rádiós kapcsolat nincs eléggé kihasználva.

### **Csomagvesztés**

Először megnéztük az átlagos csomagvesztést az alapbeállítások mellett, mert ez a csomagvesztés jellemző paramétere a Mobile IP handovernek. Az alapbeállítások a következőket jelentik:

Agent Advertisement Period = 5 s

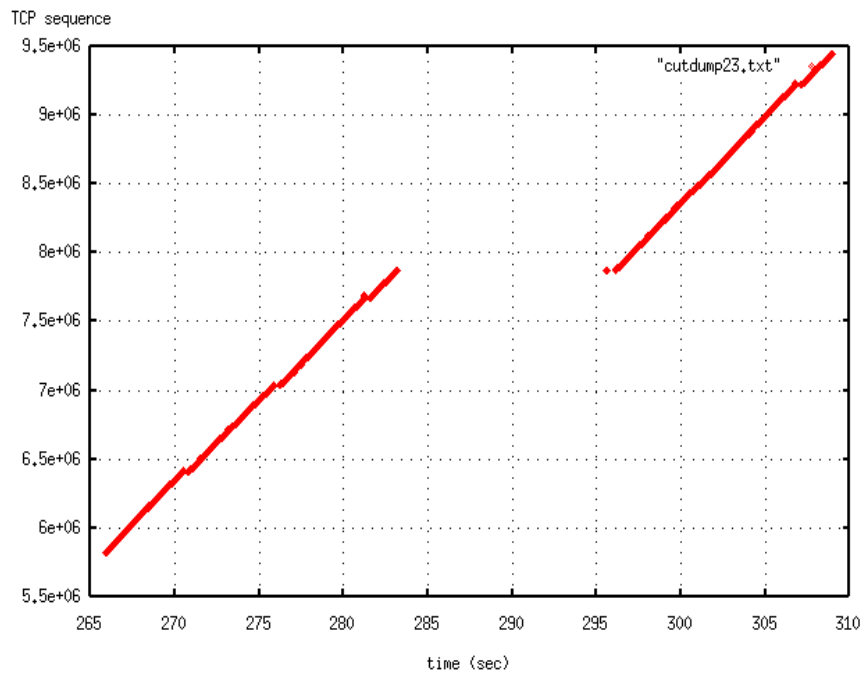
Agent Solicitation Threshold = 6 s

Agent Expire Threshold = 8 s

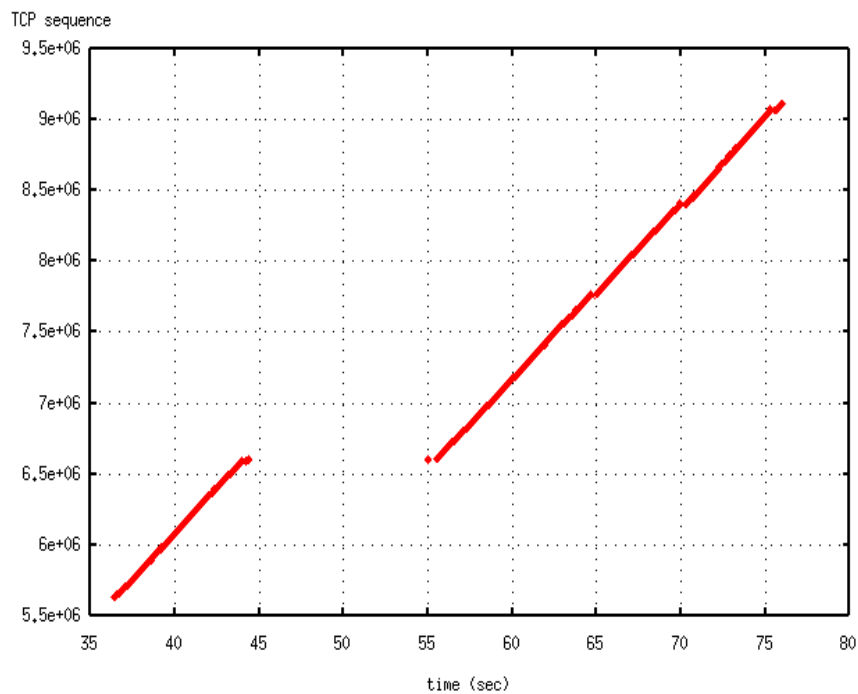
(a kifejezések magyarázatát lásd a 4.2 pontban)

A gyakorlati alkalmazásoknál fontos lesz, hogy a Mobile Node adatforgalma mennyi időre esik ki, hiszen ezt a felhasználó sebességcsökkenésként észleli majd.

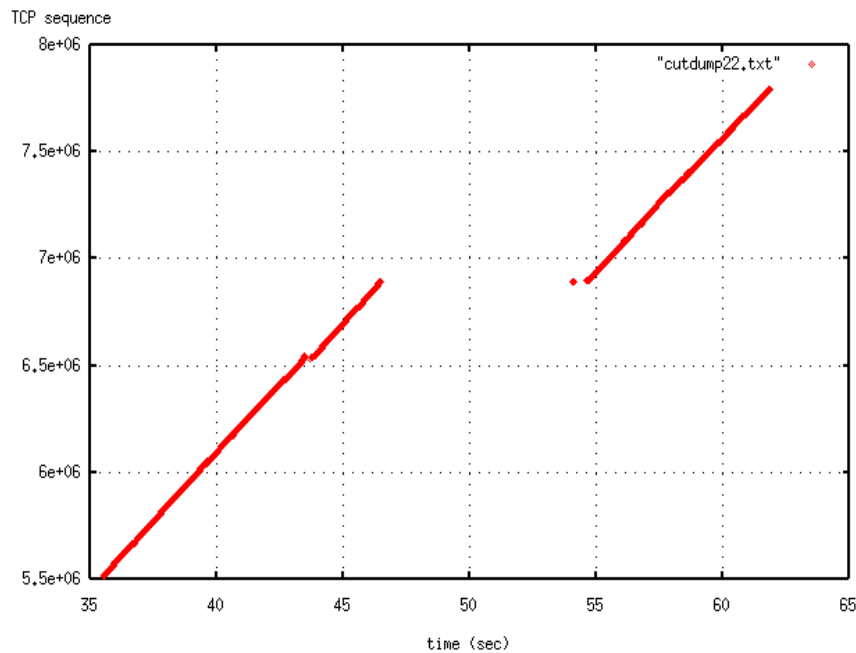
A 7.7 – 7.9 ábrán az idő függvényében ábrázoltuk a TCP csomagok sorszámának változását. Az ábrákon sorrendben a csomagvesztés maximum, átlag- és minimumértékét szemléltettük. A handover folyamán a Mobile Node először észreveszi, hogy 6 másodperc óta nem kapott Agent Advertisement-t a Base Station 1-től (üzenetek lásd 5.2 pont). Ekkor elküld egy Agent Solicitation-t, de erre sem kap választ így 8 másodperc múlva megszűntnek nyilvánítja a BS1-t. Elkezd figyelni és 5 másodpercen belül meghallja a BS2 Advertisement üzenetét. Mivel a 8 másodperc stabil késleltetés és átjelentkezés után 0-5 másodperc alatt fogja a Mobile Node a másik BS Advertisement Message-t, így összesen 8-13 másodperc adódik, amely eredményt az ábrák is alátámasztanak.



**7.7 ábra TCP csomagvesztés handover esetén (maximum)**



**7.8 ábra TCP csomagvesztés handover esetén (átlagos)**



**7.9 ábra TCP csomagvesztés handover esetén (minimum)**

Az ábrákon jól megfigyelhető, hogy a kezdetben növekvő TCP sorszám a handover alatt nem nő, azaz az adatforgalom leáll.

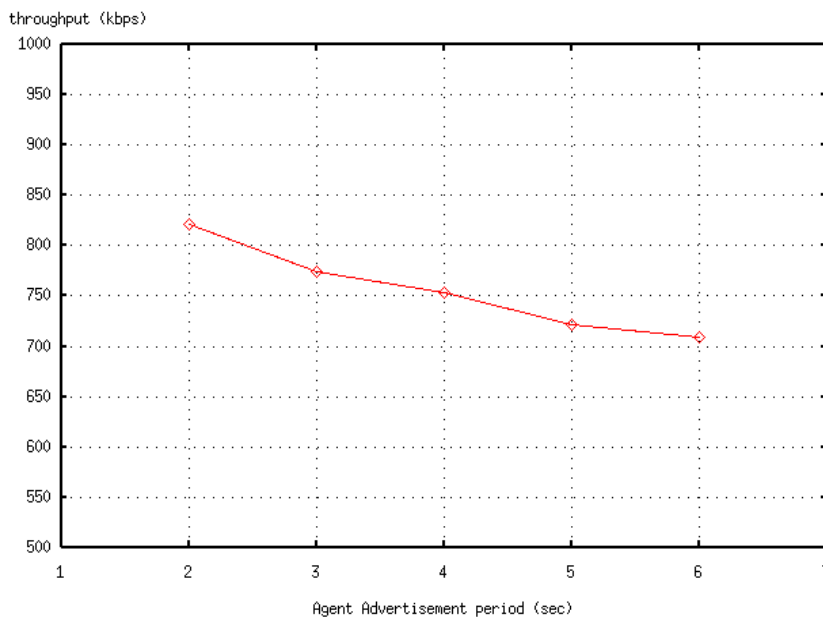
A fenti méréseink eredményei megegyeznek a Cvetkovic és csoportja [FIK99] által mért eredményekkel.

A következőkben azokat a mérési eredményeket ismertetjük, amelyekben az átviteli sebességet határoztuk meg handoverek közben.

Méréseinket folyamán eljutottunk a legfontosabb mérésekig, amelyekben megvizsgáltuk, hogy az átviteli sebesség hogyan függ a Mobile IP paramétereitől. Ezekben az esetekben nem lehet pontos kijelentést tenni az átviteli kapacitás változásáról, mert maga a TCP [RFC793] sem viselkedik a megszokott módon vezeték nélküli környezetben. A TCP rendhagyó viselkedése kívül esik a vizsgálódásunk körén, így csak a Mobile IP-től függő paraméterekről nyilatkozunk.

### **Csomagvesztés az Agent Advertisement Message-ek függvényében**

Az Agent Advertisement Message változtatása mellett vizsgáltuk az átviteli sebességét.



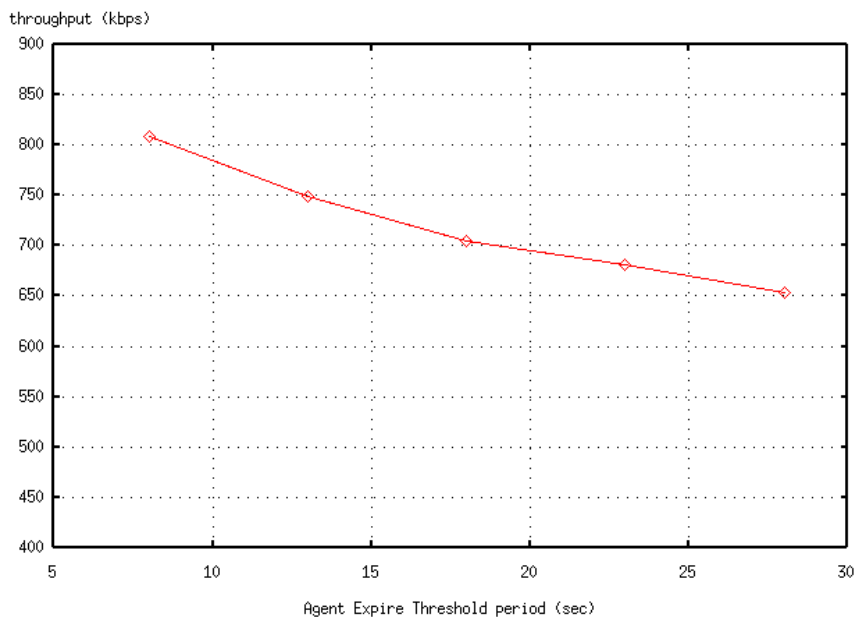
**7.10.ábra Átviteli sebesség az Advertisement Message-k függvényében**

Azt láthatjuk, hogy az Advertisement üzenet periódusát növelve csökken az átviteli sebesség. A 7.10 ábráról monoton csökkenés olvasható le. Ennek oka, hogy a nagyobb Advertisement esetén a mobilnak várhatóan több időre van szüksége, hogy az új Foreign Agenttel felvegye a kapcsolatot. Amíg a kapcsolat fel nem épül, a mobilnak küldött csomagok elvesznek. A kieső csomagok a TCP reakcióját váltják ki, így csökkentik az ablakméretet és ez eredményezi az átviteli sebesség visszaesését.

#### **Csomagvesztés az Agent Expire Threshold-ok függvényében**

Megvizsgáltuk, hogyan függ a késleltetési idő az Agent Expire Threshold-tól, illetve ez utóbbtól az átlagos átviteli sebesség. A várakozásoknak megfelelően, az Expire Threshold növelésével is csökken az átviteli sebesség, hiszen átlagosan több ideig vár egy hand-overnél az előző BS elévülésére. A hatás magyarázata az előzőhöz hasonló. Ezt szemlélteti a 7.11 ábra.



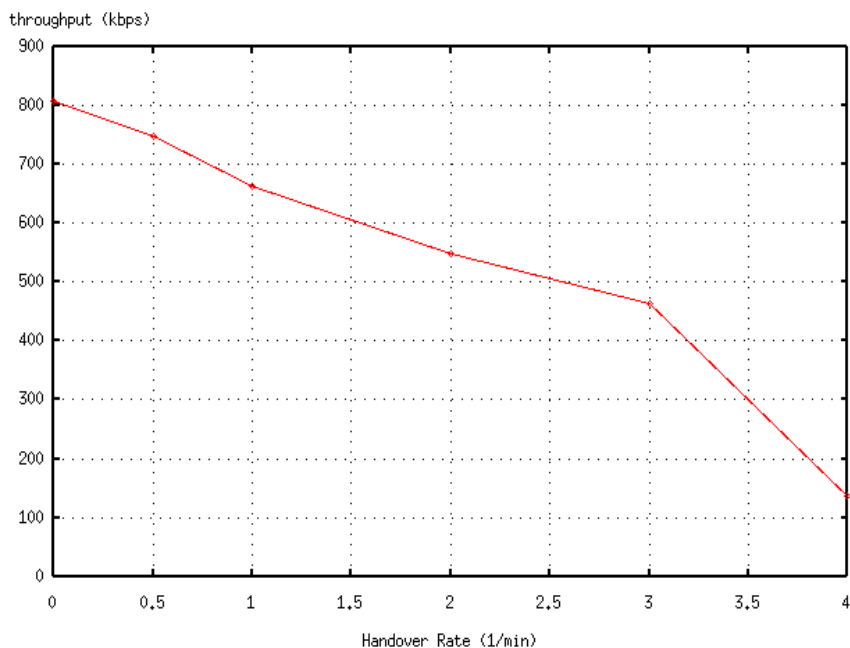


**7.11 ábra A sávszélesség változása az Agent Expire Treshold függvényében**

### **Átviteli sebesség a handover-gyakoriság függvényében**

Az előzőekben láttuk, hogy a handover csomagvesztést eredményezett. A csomagvesztés a felhasználó számára a sebesség csökkenésében nyilvánul meg. Ez azt jelenti, hogy gyakori handover esetén jelentősen romlik az alkalmazások minősége. Utolsó mérésünk ezt a jelenséget járja körül. Itt arra vagyunk kíváncsiak, hogy a handover gyakoriságának növekedésével hogyan változik a TCP átviteli sebessége.

Mérési eredményeinket a 7.12 ábrában foglaltuk össze. Itt a TCP-vel elérhető átviteli sebességet ábrázoltuk a handoverek gyakoriságának függvényében. Az ábrázolt pontok itt is több mérés átlagának felelnek meg. Jól látható, hogy a handoverek gyakoriságának növekedésével, erősen csökken az átviteli sebesség. Percenkénti négy handover esetén szinte használhatatlanok az alkalmazások. Ráadásul megfigyeltük, hogy az eredmények szórása ebben a tartományban jelentősen megnőtt. Ennek az az oka, hogy a TCP itt már nem tudja elérni normális működési tartományát, mert a csomagvesztések túl gyakran követik egymást. Az ábrán nem látható, de megfigyeltük, hogy ha ennél is jobban sűrítjük a handovereket, akkor a kapcsolat megszakad.



**7.12 ábra** Az átviteli sebesség változása a handovergyakoriság függvényében

## 8. Összegzés

Dolgozatunkban egy vezeték nélküli hálózaton mértük a Mobile IP szabvány egyes teljesítményjellemzőit. Elsősorban arra voltunk kíváncsiak, hogy a handoverek milyen hatással vannak a TCP-t használó alkalmazásokra. Az eredményekből azt a következtetést vonhatjuk le, hogy a Mobile IP főleg a lassú handover vagyis a hordozhatóság kezelésére alkalmas, mert belső késleltetései nagyok. Gyakori handoverek esetén a Mobile IP-t használó TCP alkalmazások átviteli sebessége a nagy csomagvesztés miatt rohamosan csökken.

A jövőben várható a cellák méretének csökkenése. Átmérőjük egészen néhány tíz méterig lecsökkenhet [HAA98]. Vegyünk egy példát, ahol a cellák 30 méter átmérőjűek és a felhasználó 1 m/s sebességgel halad, közben laptopjával kapcsolódik az Internetre. Ekkor percenként két handover történik. Az átviteli sebesség kétharmadára csökken ahhoz képest, mintha egyhelyben állna, azaz megállapíthatjuk, hogy a Mobile IP-t nem a gyakori handover kezelésére tervezték.

Már léteznek javaslatok a szabvány kiegészítésére gyors cellaváltás esetén [RAM99], [MAL99], [MCA99]. A jövőben ezekkel fogunk foglalkozni.

## Felhasznált irodalom:

- [BHA96] P. Bhagwat, C. Perkins, S. Tripathi: “Network Layer Mobility: an Architecture and Survey”, Personal Communications Magazine, Vol. 3, No. 3, 1996. június
- [CHO99] F. C. Choong: “TCP Performance in Mobile-IP”, URL: [http://mip.ee.nus.edu.sg/paper/foo\\_tech.doc.gz](http://mip.ee.nus.edu.sg/paper/foo_tech.doc.gz)
- [CAC98] R. Cáceres, V. N. Pedmanabhan: “Fast and Scalable Wireless Handoffs in Support of Mobile Internet Audio”, ACM MONET Journal, Vol. 3., No. 4, 1998. december
- [DYN99] Dynamics Project at Helsinki Technical University, URL: <http://www.cs.hut.fi/Research/Dynamics>
- [FIK99] N. A. Fikouras, K. El Malki, S. R. Cvetkovic, C. Smythe: “Performance of TCP and UDP during Mobile IP Handoffs in Single-Agent Subnetworks”, Wireless Communication and Networking Conference, New Orléans, Louisiana, USA, 1999. szeptember 21.-24.
- [GEI99] J. Geier: “Wireless LANs. Implementing Interoperable Networks”, Macmillan Technical Publishing, 1999.
- [HAA98] J. Haartsen: “Bluetooth - The Universal Radio Interface for Ad Hoc Wireless Connectivity”, Ericsson Review, No. 3, 1998.
- [IOA91] J. Ioannidis, D. Duchamp, G. Q. Maguire Jr.: “IP-Based Protocols for Mobile Internetworking”, ACM SIGCOMM '91, 234-245. o., 1991. szeptember
- [KAT97] R. H. Katz, H. Balakrishnan, V. N. Padmanabhan, S. Seshan: “Comparison of Mechanisms for Improving TCP Performance over Wireless Links”, IEEE/ACM Trans. on Networking, 1997. december
- [LDP] Linux Documentation Project, URL: <http://metalab.unc.edu/LDP>
- [LIN96] “Running Linux”, O'Reilly and Associates, 1996.
- [MAL99] K. El Malki, N. A. Fikouras, S. R. Cvetkovic: “Fast Handoff Method for Real-Time Traffic over Scaleable Mobile IP Networks”, Internet Draft, 1999. június
- [MCA99] P. McCann, T. Hiller, J. Wang, A. Casati, C. Perkins, P. Calhoun: “Transparent Hierarchical Mobility Agents (THEMA)”, Internet Draft, 1999. március
- [MOU92] M. Mouly, M.-B. Pautet: “The GSM System for Mobile Communications”, ISBN 2-9507190-0-7, 1992.
- [MYS97] J. Mysore, V. Bhargavan: “A New Multicast-based Architecture for Internet Host Mobility”, ACN Mobicom, Budapest, 1997. október

- [RAM99] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan: “IP micro-mobility support using HAWAII”, Internet Draft, 1999. február
- [RFC768] J. Postel: “User Datagram Protocol”, RFC 768, Internet Engineering Task Force, 1980. augusztus
- [RFC791] J. Postel: “Internet Protocol”, RFC 791 Internet Engineering Task Force, 1981. szeptember
- [RFC793] J. Postel: “Transmission Control Protocol”, RFC 793, Internet Engineering Task Force, 1981. szeptember
- [RFC2002] C. E. Perkins: “IP Mobility Support”, RFC 2002, Internet Engineering Task Force, 1996. október
- [RFC2003] C. E. Perkins: “IP encapsulation within IP”, RFC 2003, Internet Engineering Task Force, 1996. október
- [STE94] R. W. Stevens: “TCP/IP Illustrated Volume 1: The Protocols”, Addison-Wesley, 1994.
- [SUN99] Sun Microsystems: “Solaris MobileIP”, <ftp://playground.sun.com/pub/mobile-ip>
- [TER95] F. Teraoka: “Virtual Internet Protocol version 2”, Internet Draft, 1995. december
- [TTCP] “ttcp: Test TCP”, U.S. Army Ballistics Research Lab, 1984. december
- [WAV98] “IEEE 802.11 WaveLAN. PC Card User’s Guide”, Lucent Technologies, Bell Labs Innovations, 1998. május

## Függelék A - Routingvizsgálat

CN (pc1) -> MN

- traceroute 10.100.3.3 (from 195.228.139.66)

```
traceroute to 10.100.3.3 (10.100.3.3), 30 hops max, 40 byte packets
 1 ethr1.ericsson.co.hu (195.228.139.65) 0.926 ms 0.742 ms 0.745 ms
 2 pc2.ericsson.co.hu (195.228.139.69) 0.615 ms 0.530 ms 0.546 ms
 3 10.100.3.2 (10.100.3.2) 1.050 ms 0.955 ms 0.927 ms
 4 * * *
 5 10.100.3.3 (10.100.3.3) 7.015 ms 6.649 ms 6.792 ms
```

## Függelék B - Körülfordulási időt értékelő program

### A *packet.pl* Perl szkript

```
#!/usr/bin/perl
# Round Trip Time measuring script based on command 'ping'
# (c) Csanad Szabo - 09/10/1999

sub bysize {
    (lc $a->{size}) > (lc $b->{size}) # function to sort lists
}

sub logging{
    $pack=$_[$nr-1]; # get last line of ping (statistics)
    if ($pack =~ /(\d{1,4}\.\d)\.(\d{1,4}\.\d)\.(\d{1,4}\.\d) ms$/) {
        my $rec = {};
        $rec->{size} = $i; # packet-size
        $rec->{min} = $1; # min. round-trip delay
        $rec->{avg} = $2; # avg. round-trip delay
        $rec->{max} = $3; # max. round-trip delay
        push @res, $rec; # put list in the array
    }
}

if ($ARGV[0] eq "") { # searching for parameters
    die("\t\tYou must give an IP-Address in the command-line!\n");
}
unless ($ARGV[0] =~ /^(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})$/) {
    die("\t\tIt does not look like an IP-Address\n");
} # is parameter an IP address?

$ip = $ARGV[0];
$first = `ping -c 20 -s 58 $ip`; # make the path for
# real measurement

for ($i=50;$i<=1400;$i+=50) { # sending periodically
    @tomb=`ping -c 100 -s $i-8 $ip`; # 100 packets increasing
    chomp(@tomb); # the size with 50 bytes
    $nr = @tomb; # after each period
    logging(@tomb);
} # creating log files:
open(LOG,">>/root/avg"); # average
foreach $f1(sort bysize @res) {
    printf LOG "%4d " . $f1->{avg} . "\n", $f1->{size};
}
close LOG;
open(LOH,">>/root/min"); # minimum
foreach $f2(sort bysize @res) {
    printf LOH "%4d " . $f2->{min} . "\n", $f2->{size};
}
close LOH;
open(LOI,">>/root/max"); # maximum
foreach $f3(sort bysize @res) {
    printf LOI "%4d " . $f3->{max} . "\n", $f3->{size};
}
close LOI;
```

## Függelék C - Frekvenciaváltó program

A *wper.c* a C nyelven írt program forráskódja. Lentebb programunk bemutatásánál csak a főprogramra szorítkozunk, annak is csak a lényegi része áll itt, szemléltetve, hogyan állítja a program az `ioctl()` Linux kernelhíváson keresztül a kártya paramétereit futás közben.

A *wfreq.c* program az alábbi program egy régebbi változata, mely csak egyszer módosítja a kártya frekvenciaparamétereit, ellenben a *wper* programmal, mely periodikusan teszi ezt.

```
/******          M A I N          *****/

int main(int argc, char **argv)
{
    int          i,j;
    int          skfd=-1;           // AF_INET raw socket desc.
    struct iwreq iwreq iwri;       // wireless extension
                                        // cf. wireless.h
    if (( skfd = socket(AF_INET,SOCK_DGRAM,0)) < 0) {
        perror("socket");
        exit(-1);                  // Creating channel to kernel
    }
    if (ioctl(skfd,SIOCGIWREQ,&iwri)<0) {
        fprintf(stderr, "SIOCGIWREQ: %s\n", strerror(errno));
        exit(-1);                  // Get current frequency;
    }
    for(i=0;i<nr;i++) {
        if (iwri.u.freq.m == 3) { iwri.u.freq.m = 5;
        }
        else { iwri.u.freq.m = 3;
        }
        if (ioctl(skfd,SIOCSIWREQ,&iwri)<0) {
            fprintf(stderr, "SIOCSIWREQ: %s\n", strerror(errno));
            exit(-1);              // Setting the frequency
        }
        printf("\t\t%d. handover (channel: %d)\n",i+1,iwri.u.freq);
        fflush(NULL);
        if (i == nr-1) {           // Do not wait after
            continue;              // last change
        }
        sleep(sec);                // Wait 'sec' seconds long
    }
    (void)close(skfd);            // Close channel to the kernel
    return 0;
}
```