



M Ű E G Y E T E M 1 7 8 2
TÁVKÖZLÉSI ÉS MÉDIAINFORMATIKAI TANSZÉK
BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM

SKÁLÁZHATÓ MEGOLDÁSOK EGYENRANGÚ
(PEER-TO-PEER) HÁLÓZATOK
KOMMUNIKÁCIÓJÁNAK JAVÍTÁSÁRA

Tézisfüzet

Kovács Balázs

Tudományos vezető:

Dr. Vida Rolland

Távközlési és Médiainformatikai Tanszék

BUDAPEST, MAGYARORSZÁG
2009. FEBRUÁR

1. Bevezető

A vezetékes és vezeték nélküli Internet terjedése, valamint az egyre változatosabb alkalmazások számos felhasználót ismertetnek meg a nagysebességű kommunikációban rejlő lehetőségekkel. Az Internet felhasználók és az általuk használt kommunikációs eszközök mennyiségének növekedése a kommunikációs algoritmusok újragondolását igényli, hogy biztosítsák a méretezhetőséget, skálázhatóságot.

Általánosságban a skálázhatóságon egy rendszer, hálózat, vagy folyamat azon képességét értjük, mely alapján növekvő mennyiségű munkát elegánsan tud kezelni [1]. Más szóval, egy protokollt akkor lehet skálázhatónak nevezni, ha jelentős terhelés alatt is megfelelően hatékony marad. A kommunikációs protokollok esetében ez azt jelenti, hogy a szolgáltatásminőség csak kismértékben változhat a csomópontszám, a mobilitás, a forgalmi paraméterek, vagy más hálózati jellemzők drasztikus módosításával. Ezt úgy tudjuk elérni, hogy a hálózati csomópontok által tárolt állapotok számát alacsonyán tartjuk; ez befolyásolja a memória felhasználást, a feldolgozási időt, és nem utolsósorban a hálózatban szétterjesztendő és összegyűjtendő információt, azaz a jelzésforgalmat.

A funkciók és felelősségek szétosztása egy olyan jellemző, mely egy rendszert skálázhatóvá tehet. Az elosztott rendszerek egy speciális formája az egyenrangú („peer-to-peer”, P2P) architektúra, egy rendszer melyet funkciókban egyenrangú egyedek alkotnak. A számítógépes hálózatokban a P2P jelleg a csomópontok egy olyan hálózataira utal, ahol minden csomópont szolgáltat és használja más csomópontok szolgáltatásait, azaz mindegyik csomópont lehet kiszolgáló és ügyfél. A P2P hálózatok egy megkülönböztető tulajdonsága az, hogy a partnerek hálózatba való belépésükkel hozzájárulnak a hálózati erőforrásokhoz. Egy másik előnye a P2P hálózatnak az egyszeres hibapontok kiküszöbölése; a hálózat képes tovább működni bármely résztvevőjének meghibásodása után. Vannak megoldások melyek ún. szuper-peer architektúrát alkalmaznak, mely kiemel partnereket valamilyen osztályozási kritérium alapján a P2P hálózat szervezésének feladata végett. A szuper-peer szerepet azonban bármely csomópont képes ellátni, ha egy megválasztott szuper-peer meghibásodik.

A P2P architektúrát általában az OSI alkalmazási rétegben definiálják. A P2P egy népszerű megoldás fájlmegosztáshoz, csevegéshez, telefonáláshoz, és média letöltő alkalmazásokhoz. Ugyanakkor a P2P kommunikációs paradigma jóval több ennél. Általánosan beszélve egy rendszer P2P természetű lehet, ha az előző paragrafusban említett jellemzők érvényesek rá. A vezeték nélküli ad hoc hálózatok útválasztása az IEEE 802.11 [4] technológiával egy speciális hálózati-rétegbeli P2P szolgáltatás. Annak érdekében, hogy többgrásos rádió kommunikációt biztosítsunk az ad hoc hálózatokban, minden csomópontnak képesnek kell lennie útválasztási szolgáltatást felajánlani mások számára. A 802.11 a Distributed Coordination Function (DCF) technológiával és a Carrier Sense Multiple Access Collision Avoidance (CSMA/CA)

eljárással [4] egy teljesen versengés alapú közegelési eljárás, amely a koordináció minden formáját mellőzi, ezzel lehetővé téve, hogy a 802.11 csomópontok útválasztókként funkcionáljanak.

A disszertációm kommunikációs protokollok skálázhatóságára fókuszál egyrészt a hálózati rétegben, ad hoc hálózatok számára, másrészt az alkalmazás rétegben olyan átfedő („overlay”) hálózatok számára, melyek speciális alkalmazási követelmények alapján formálódnak.

Egy ad hoc hálózatban a hálózattal kapcsolatos skálázhatósági problémák többnyire a címzési és útválasztási feladatokhoz kapcsolódnak, ha nagy a hálózat mérete, jelentős a mobilitás, vagy ezek kombinációja egyszerre érvényesül. Vannak proaktív, reaktív és hibrid/hierarchikus megközelítések a vezeték nélküli ad hoc útválasztás kezelésére. A [3, 5, 7] cikkek a legfontosabb ilyen jellegű megoldásokat mutatják be. A további részleteket az olvasó megtalálhatja a disszertációban [2].

A téziseimben javaslak egy hierarchikus címzési és útválasztási módszert, egy hibrid megoldást, mely arra a feltételezésre épít hogy a vezeték nélküli ad hoc hálózat dinamizmusa néhány relatívan fix, földrajzi ponthoz köthető a mobil csomópontok tartózkodási helyének közelében. Az architektúra melyet SCARF architektúrának neveztem el (SCalable Addressing and Routing architecture for Future networks) fix csomópontokat helyez el a tartózkodási helyen és bevonja őket hálózat-adminisztrációs feladatokba. Mindemellett bemutatom az architektúra egy kiterjesztését, mely hatékony többugrásos Internet elérést biztosít a SCARF hálózat számára.

A disszertációm alkalmazás-rétegbeli skálázhatósági problémákra is fókuszál, azon belül elosztott hash táblákra (Distributed Hash Tables, DHT), azaz strukturált P2P hálózatok átfedő hálózatának kialakítására és működésére. A strukturálás jelen esetben egy algoritmus alapján történő szervezést jelent, a hálózatban történő kikeresések felgyorsításának érdekében. A DHT megoldások erőforrás azonosítókat képeznek le egy hash tartományra, amely szét van osztva a hálózat csomópontjai között. Ha egy erőforrást egy csomópont hash alterébe képezünk le, akkor eme csomópont lesz felelős az adott erőforrásért, azaz a kikeresések erre az erőforrásra ezen a csomóponton végződnek. Általánosságban, a DHT megoldások segítségével strukturált P2P hálózatokat építhetünk, alacsonyan tartva a kikeresések költségét. Mivel ezek a DHT protokollok az alkalmazási rétegben működnek, bármilyen tetszőleges logikai vagy átfedő topológiát építhetnek.

A disszertációm egy javítást mutat be egy jól ismert, de nem túl népszerű DHT algoritmus számára, melyet tartalom alapján címezhető hálózatnak neveznek (Content Addressable Network, CAN) [17]. A CAN fő hátránya, hogy a kikeresési költség a hálózatban viszonylagosan magas, mivel $O(dN^{\frac{1}{d}})$ arányában növekszik, ahol N a csomópontok száma és d az azonosító tér dimenziószáma, egy olyan hálózati paraméter melyet nem lehet megváltoztatni a rendszer működése közben. A CAN megoldással szemben a modern DHT protokollok, mint a Chord [18] és a Kademia

[20], a kikeresések számára $O(\log N)$ lépésszámot biztosítanak. Ugyanakkor a CAN számos előnnyel rendelkezik, melyek az általam javasolt javítással karöltve $O(\log N)$ lépésszámú kikeresési költséghez vezethetnek úgy, hogy a továbbfejlesztett megoldás számos szempontból megelőzi a Chordot és a Kademliát.

Olyan DHT megoldásokban mint a Chord és a Kademlia a gyors kikeresések eszközei a hálózat „távoli” részeiben elhelyezkedő kapcsolatok. Minden DHT-ban egy távolság metrikát lehet definiálni az azonosító téren. Általában egy DHT csomópont legalább egy-egy kapcsolatot tárol a saját azonosítójától exponenciálisan növekvő távolságokra. Következésképpen, egy kikeresés egy DHT-ban egy logaritmikus keresés egy rendezett térben. A hatékony működés érdekében a DHT-k *hosszútávú kapcsolatokat* helyeznek el. Az egyetlen szerepe ezeknek a kapcsolatoknak a kikeresések felgyorsítása. Másrészt a DHT csomópontoknak pontos információra van szükségük a *rövidtávú kapcsolatokról*, melyek alapvetőek a konzisztens azonosítótér fenntartása érdekében. Ezen okokból tudunk különböző szempontokat érvényesíteni a rövid- és hosszútávú kapcsolatok fenntartása során.

A PhD munkámban a CAN számára egy algoritmikus javítás elkészítésére koncentráltam, melyet LR-CAN (Long-Range CAN) néven ismertetek majd. Ez egy adaptív megoldást mutat be a hosszútávú kapcsolatok elhelyezésére a DHT megoldásokban.

A tézisfüzetem a kutatói munkám legfontosabb eredményeit mutatja be két téziscsoportban.

2. Kutatási Célkitűzések

Az első téziscsoport célja a SCARF (SCalable Addressing and Routing architecture for Future Networks) protokoll kidolgozása volt ad hoc hálózatok skálázható kommunikációjára, ahol a mobil csomópontok véletlen mozgásai néhány adott, viszonylagosan fix pontok körül történnek. A kutatásom arra koncentrált, hogy ezek a fix pontok, mint hálózati csomópontok, hogyan tudják felgyorsítani a csomópontok helyének meghatározását, csökkenteni a üzenetszórt („broadcast”) forgalmat és növelni a mobilitás toleranciát. Másrészt, az első téziscsoport javasol egy architektúrát, mellyel a SCARF ad hoc hálózat számára többgrásos vezeték nélküli Internet elérés lehetséges.

A második téziscsoport által összefoglalt kutatói munka egy javítást dolgoz ki a CAN DHT protokoll számára. Ennek a kutatásnak a célja az, hogy alacsonyabb és sokkal kiegyensúlyozottabb kikeresési költséget biztosítson, alacsonyabb jelzésforgalmi költséggel, mint amit más elterjedt DHT protokollok, a Chord és a Kademlia nyújtani tudnak. Ezzel a javítással a CAN visszakerülhet a DHT hálózatokkal kapcsolatos kutatás fókuszába.

3. Módszertan

Szimulációkat használtam hogy bizonyítsam az életképességét és elemezzem a hatékonyságát az algoritmusoknak, protokolloknak melyeket az első tétiscsoportban javasoltam. A szimulációs kódot az ns-2 [13] diszkrét esemény alapú szimulátorban implementáltam.

Az alkalmazás-rétegbeli DHT javítás vizsgálata, melyet a második tétiscsoportban javaslok, analitikusan is lehetséges, mivel néhány, vagy az összes alacsonyabb-rétegbeli kapcsolattartási problémát mellőzhetjük a modelltől bizonyos mértékek értékelése során. Ugyanakkor az analitikus vizsgálat feltételezései miatt, és mivel a protokollt más megoldásokkal is szándékomban állt összehasonlítani, hálózatszimulátorban is kiértékeltem a protokollt.

4. Új eredmények

4.1. Skálázható útválasztás vezeték nélküli ad hoc hálózatokban

Napjainkban az IEEE 802.11 család az Internet elérésre és ad hoc hálózati kommunikációra használt legnépszerűbb vezeték nélküli technológia. A 802.11 rendszerek egy jelentős hátránya a „broadcast” üzenetek pazarló és megbízhatatlan továbbítása, egy szolgáltatás melyet a hálózat rétegben működő útválasztó protokollok gyakran használnak. A csomópontok információinak terjesztése és igénylése üzenetszórással igen költséges egy 802.11 technológiával működő vezeték nélküli ad hoc hálózatban.

Számos eset adódhat mikor ad hoc hálózatok épületek, termek, egyetemek, vagy más infrastruktúra közelében működnek. Olyan helyeken ahol a mozgó mobil csomópontokhoz viszonylagosan fix referencia pontokat határozhatunk meg. Abban az esetben ha elfogadjuk a kényszereket melyet ezek a referencia pontok jelentenek a hálózat dinamikájának számára, előnyre tehetünk szert hálózat-adminisztrációs feladatokban, mellyel csökkenthető a jelzésforgalmi terhelés.

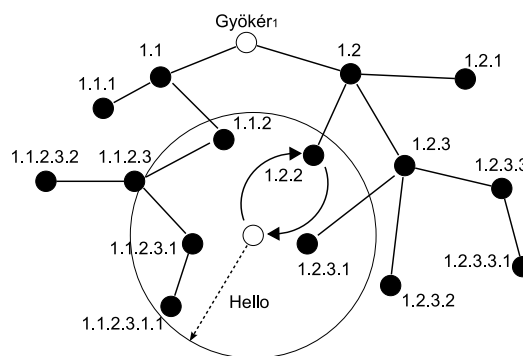
Ebben a tétiscsoportban javaslok egy hálózati architektúrát, mely megszünteti az üzenetszórás hullámokat a csomópont és útfelderítések esetén és kizárólag unicast lekérdezéseket használ. Ennek érdekében egy hierarchikus architektúrát építünk mely kevés jelzésforgalommal hozható létre és tartható fenn. Az architektúra jó alapot állít fel a hibátűrő útválasztáshoz.

Tétiscsoport 1: *Javasoltam egy címzési és útválasztási módszert vezeték nélküli ad hoc hálózatok számára, mely útválasztási táblák nélkül végzi az útválasztási feladatokat és ezáltal alacsonyban tartja a hálózat rétegben a jelzésforgalmat.*

4.1.1. Címzés

Tézis 1.1 (Cím kiosztás egy széles és kiegyensúlyozott fáért [J2, J6, C2, C3]): *Javasoltam egy hierarchikus cím kiosztási stratégiát vezeték nélküli ad hoc hálózatok számára. A mobil ad hoc csomópontok egy cím fát építenek, mely egy gyökérrel rendelkező fa egy statikus gyökér csomóponttal, az élei fizikai kapcsolatokat reprezentálnak. A csomópontok szülő-gyermek kapcsolatokat építenek a cím fában.*

A szülő-gyermek kapcsolatokat a cím kiválasztó algoritmus által jönnek létre. A belépő csomópont számára az algoritmus a gyökérhez legközelebbi és legkevesebb gyermekkel rendelkező csomópontot választja szülőnek.



1. ábra. Cím kiosztás

A fa struktúra az algoritmus legfontosabb eleme. Minden új csomópont a szomszédjait és azok hierarchikus azonosítóját, Local Node Address (LNA), periodikus Hello üzenetekből tanulja meg. Ezek az üzenetek elegendő információt hordoznak az új csomópont számára hogy kiválassza a megfelelő szülőt. A kiválasztott szülő kijelöl egy LNA-t az új csomópontnak, azaz egy utótaggal látja el a saját LNA címét. Például, az új csomópont az 1. ábrán az 1.2.2 című csomópontához csatlakozik és az 1.2.2.1 LNA-t kapja meg.

A cím kiosztási stratégia a cím kiválasztó algoritmussal a cím fát a gyökér köré kiegyensúlyozottra és szélesre építi, mely előnyös a csomagok útválasztása során.

4.1. Következmény. *A széles és kiegyensúlyozott fa a vezeték nélküli logikai topológiában a gyökeret a középpontba helyezi és ezzel biztosít vízszintes kiegyensúlyozást, azaz terhelési egyensúlyt a címfa ugyanazon szintjein elhelyezkedő csomópontjai számára.*

A szakirodalomban található megoldásokkal ellentétben ([6, 7, 8, 9, 10]) ez a hierarchikus címzési stratégia egy megbízható, statikus csomópontot feltételez mint gyökér. Egy ilyen csomópont elhelyezése kényszerrel állít fel a hálózat dinamizmusára, de megkönnyítheti a címzést és a csomópontok megtalálását amennyiben körültekintően tervezzük meg a hálózatot (részletesebben a 1.3 Tézis után). A címfa széles

és kiegyensúlyozott tulajdonsága nem volt célja a szakirodalomban létező korábbi megoldásoknak.

4.1.2. Útválasztás

A javasolt címzési eljárás egy útválasztási táblák nélküli útválasztási algoritmus működését teszi lehetővé. A következő definíciók segítenek megérteni az útválasztási algoritmus koncepcióját. A 1.2 Tézis megmutatja hogyan oldható meg az útválasztás a javasolt architektúrában.

4.1. Definíció. Egy csomópont *útválasztási táblája* egy adatbázis, melynek egy bejegyzése egy adott célponthoz vezető szomszédos csomópontot vagy csomópontoknak egy listáját nevezi meg.

A vezeték nélküli ad hoc hálózatokban a gyors útvonal eléréséért egy csomópontnak nemcsak a saját kommunikációja érdekében kell bejegyzéseket tárolnia, ezért egy útválasztási tábla a hálózat méretével arányosan $O(N)$, vagy legjobb esetben $O(\log N)$ mértékben nő, ahol N a csomópontok száma.

4.2. Definíció. Egy vezeték nélküli hálózatban egy csomópont *szomszédossági listája* egy adatbázis, melynek bejegyzései csomópontoknak felelnek meg egy bizonyos „távolságon” belül (általában rádiós hatótávolság).

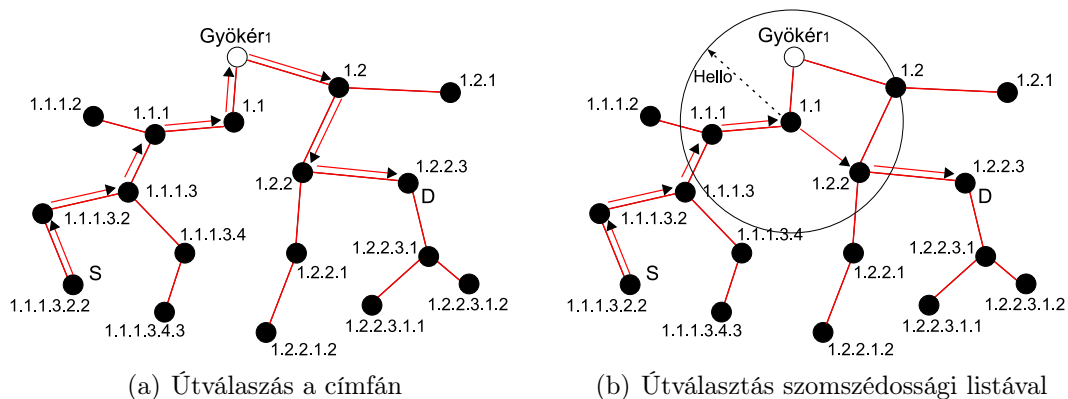
Általánosságban egy szomszédossági lista mérete $O(R^2\rho)$, ahol R a rádiós hatótávolság és ρ a hálózat sűrűsége, melynek értéke ésszerű esetekben konstansnak tekinthető. Mivel R szintén vehető konstansnak, a szomszédossági lista mérete is konstans.

Jelölje $S = \{m_i\}_{i=1}^M$ a szomszédossági listában elérhető M csomópont halmazát. Az a LNA cím hossza $|a|$. Ha egy i csomópont a_i címmel rendelkezik és a j csomópont címe a_j , a_i és a_j közös cím prefix hossza $|p_{ij}|$. A fa alapú távolság az i és j csomópont között $d_{ij} = |a_i| + |a_j| - 2 \cdot |p_{ij}|$.

Tézis 1.2 (Címfa alapú útválasztás [J2, J6, C2, C3]): *Javasoltam egy útválasztási megoldást az 1.1 Tézisben bemutatott hierarchikus címzési módszer alapján, amely kizárólag LNA címeket és szomszédossági listákat használ az útválasztáshoz, útválasztási táblákat mellőzve. Egy adott D célcsomópont felé a következő csomópont $\operatorname{argmin}_{m \in S} (d_{mD})$ lesz.*

Ha egy csomópont ismeri a szomszédjai és a célpont címeit, a továbbítás hurokmentes lesz.

A javasolt címalapú útválasztási módszer a szomszédossági listák segítségével nem terjeszt információt a hálózat távoli részeibe, és nem árasztja el a hálózatot üzenet-szórt útfelderítésekkel.



2. ábra. Alapvető útválasztási technika

A 2(a). ábrán egy példát láthatunk egy címfán történő útválasztásra. Egy csomagnak nem szükséges a címfa összeköttetésein keresztül utaznia. Ha a hálózati topológia megengedi, a csomópontok lerövidíthetik az utat a szomszédossági listák segítségével, ahogy a 2(b). ábra mutatja. Az utak rövidítése mellett ezek a levágások tehermentesíthetik a gyökér csomópontot a folyamatos csomagtovábbítás alól.

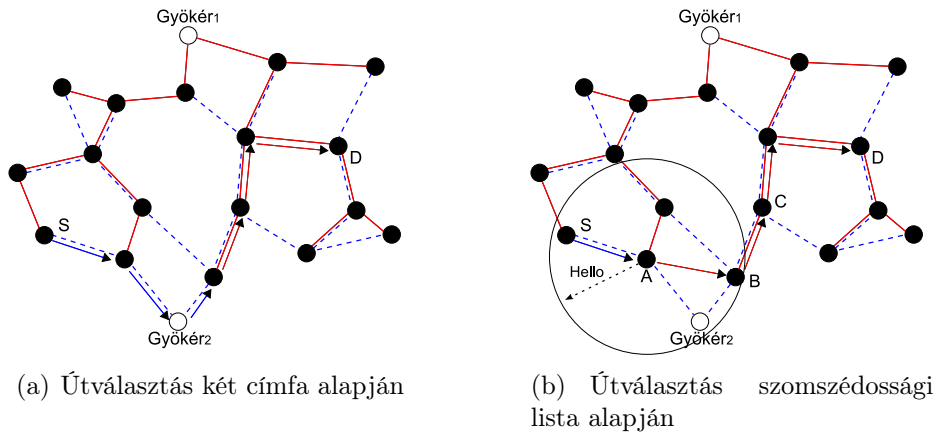
Léteznek más hierarchikus megoldások ([7, 8, 9, 10]), melyek csökkenthetik az útválasztási táblák méretét $O(\log N)$ -re. Azonban ezek a megoldások valamilyen proaktív vagy reaktív ad hoc útválasztási protokoll kiegészítő útválasztási információit is terjesztik. Mindemellett ezeknél a protokolloknál kompromisszumra van szükség a mobilitási események jelzésforgalma és a mobilitás tolerancia között, melyben nehéz jól dönteni.

Az útválasztási megoldás melyet az 1.2 Tézisben javasoltam leterhelheti a gyökérhez közeli csomópontokat. Ezek tehermentesítése érdekében van szükség függőleges kiegyensúlyozásra. A következő tétel erre ad megoldást.

Tétel 1.3 (Útválasztás több fával [J2, C2]): *Több fizikailag elkülönített statikus gyökér csomópontok elhelyezését és több hierarchikus címfa kifejlesztését javasoltam a vezeték nélküli hálózatban az útválasztási algoritmus hatékonyságának növelése és a függőleges terhelési egyensúly érdekében. A csomópontok mindegyik címfához csatlakoznak, hogy több LNA címmel rendelkezzenek, és ha lehetséges, mindegyik fában más szülő választanak maguknak. Ezek után a csomagtovábbítási folyamat a következő csomópontot bármely fa alapján kiválaszthatja, hogy a lehető legrövidebb utat biztosítsa a cél felé.*

Jelölje $T = \{t_k\}_{k=1}^K$ a hálózatban kifejlesztett K címfa halmazát. Egy adott D célcso-
mópont felé a következő csomópont $\arg\min_{m \in S} (\min_{t \in T} d_{mD}^t)$ lesz, ahol d_{mD}^t a távolság
az m és a D célcso-
mópont között a t fán.

Ahogy a 3(a). ábra mutatja, több alternatív és rövidebb útvonalat találhatunk a



3. ábra. Útválasztás több fával

két címfa kombinálásával ugyanazon forrás–cél pár között. Az 1-es gyökér címfáján megtett 8 ugrás helyett az S csomópont úgy találja hogy csak 7 ugrásra van D -től a 2-es gyökér címfáján, így arra továbbítja a csomagot. Az A csomópont a szomszédossági listáját használja az út levágására és rögtön a B csomópontra ugrik. A szintén észreveszi, hogy a csomagtovábbítás az 1-es gyökér fájára válthat a B csomópontnál, mivel D azon közelebb van, ezért B azon címét használja melyet az 1-es gyökértől kapott.

A gyökér csomópontok fizikai elkülönítése nagyon fontos az útválasztási teljesítmény szempontjából. Minél több fát használunk és minél jobban elkülönülő fákat tudunk kifeszíteni, annál jobb függőleges terhelésegysúlyt tapasztalhatunk, mivel több alternatív utat találunk. A gyökércsomópontok egyre kisebb valószínűséggel vesznek részt a csomagtovábbításban. A gyökércsomópontok számának és pozíciójának meghatározásáról a hálózatunk tervezése során kell gondolnunk. A több fa használata több jelzésforgalmat jelent: a csomagok mérete nő és a fa építésével kapcsolatos jelzésforgalom a fák számával többszöröződik. Fontos azonban megjegyezni, hogy a hello üzenetek gyakorisága, mely üzenetek a jelzésforgalom zömét képezik, nem változik.

A SCARF architektúra támogatja a gyökér csomópontok igény szerinti telepítését, hogy megkönnyítse a tervezési fázist. A disszertációmban nem vizsgálom ezen tulajdonságát.

A több fával végzett útválasztási mechanizmus képes továbbítási hibákat felfedni, vagy a logikai topológia segítségével—nem talál a jelenlegi csomópontnál a célhoz közelebbi csomópontot—vagy a fizikai hálózati mechanizmusok által—a következő csomópontot megtalálja de az adatkapcsolati réteg továbbítási hibát jelez.

4.2. Következmény. *Egy továbbítási hiba észlelésekor a több fával történő útválasztási módszer egy csomagot átirányíthat egy olyan csomópontra melyen már járt a csomag hurok létrehozása nélkül, mivel az adott csomag továbbításából az algoritmus kizárja azon fákat melyen hibát detektált.*

Ez a tulajdonság javítja az algoritmus kapcsolathibákhoz történő alkalmazkodóképességét, mivel több alternatív útvonalon is igyekszik továbbhaladni mielőtt útvonal hibát jelent. Miután elérjük a célt, az elavult címinformációk frissíthetők.

Fontos megjegyezni, hogy a megfelelő teljesítményhez egy k fával kifizetett vezeték nélküli hálózatnak legalább k összefüggőnek kell lennie.

4.1.3. Címtár szolgáltatás

Ahhoz hogy útválasztást tudjunk végezni az ad hoc hálózatban, először azonosítanunk kell a célcsomópontot. Mivel idővel egy csomópont LNA címei a mozgások során változhatnak, globális egyedi azonosítót kell minden csomóponthoz rendelni. Ezt az azonosítót én Global Node Identifier (GNI) névvel láttam el. Következésképpen az útválasztás megkezdése előtt a célcsomópont azonosítóját le kell képeznünk több, a címfáknak megfelelő LNA címekre.

Ezt a leképezést a címtár szolgáltatásban tároljuk, melynek neve Local Address Directory (LAD). A címtár szolgáltatás kizárólag a gyökér csomópontokon fut. Minden gyökér csomópont a saját címfájában résztvevő GNI és az ahhoz tartozó aktuális LNA címet tárolja. A LAD szolgáltatás lehetővé teszi, hogy bármely két csomópont kapcsolatba lépjen egymással egy egyszerű LAD lekérdezéssel.

Az útválasztási algoritmus mechanizmusai biztosítják, hogy a címtár szolgáltatással csak jelentős topológiaiaváltozás esetén kell kapcsolatba lépni, amit az útválasztás már nem tud javítani. Részleteket az olvasó megtalálhatja a disszertációban.

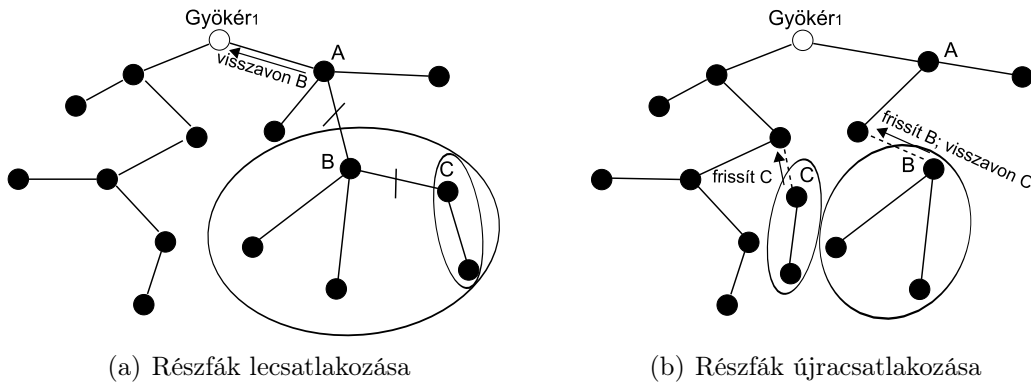
A javasolt címzési és útválasztási módszer megkívánja, hogy a LAD naprakész LNA címeket tároljon. Ennek érdekében a mobil csomópontoknak értesíteniük kell a LAD szolgáltatást ha új LNA címet kapnak. Továbbá a LAD-nak tudnia kell egy mobil csomópont jelenlétéről, vagy elévülőről, vagy fix bejegyzésekkel. Ez utóbbi csak akkor törlődik, ha a hálózat értesíti a LAD-ot egy csomópont távozásáról.

4.1.4. Tartományon belüli mobilitás

A csomópontok mobilitása kapcsolathibákat okoz a címfákon.

4.3. Következmény. *Az 1.1 Tézis által bemutatott cím kiosztási stratégia lehetővé teszi a topológia részfáinak együttes kezelését a mobilitási események során. Ennek következtében a részfák újracímzése alacsony költséggel megoldható.*

Egy részfa újracímzése két hálózati interakciót igényel. Először az elmozdult részfa i gyökerének értesítenie kell a LAD szolgáltatást az új szülőjétől szerzett címéről. A LAD lefuttatja a címcsereelő algoritmusát amely minden $j \in T_i$ csomópont a_j címét $\langle\langle a'_i \rangle, \langle id_j^{(i)} \rangle\rangle$ -re cseréli, ahol T_i az i csomópont részfája, a'_i az i csomópont új címe, és $id_j^{(i)}$ a j csomópont azonosítója T_i -n belül. Az i csomópont azonnal küld egy hello üzenetet hogy értesítse a gyermekeit a prefix változásról; a gyermekei is hasonlóan tesznek míg az értesítés el nem éri a részfa leveleit.



4. ábra. Mobilitáskezelés

A 4. ábrán láthatjuk ahogy a B csomópont T_B részfája elveszíti a szülő csomópontot mivel az elmozog. Ezzel egyidőben a C csomópont T_C részfája is lecsatlakozik a B csomóponttól. A B és a C is új címet kér. Ezáltal az összes csomópont amely tagja a B és C részfának új címet kap a LAD-ban. Ezt mindössze két unicast üzenet küldésével megtehető, az egyiket B a másikat C küldi. Így a mobilitási események miatt keletkező üzenetek száma a meghibásodott címfa kapcsolatok számával arányos.

Tézis 1.4 (A faegyensúly megőrzése[J2, J6, C2, C3]): *Az 4.3 Következmény módszere alapján javasoltam egy algoritmust, amely a logikai topológiát, azaz a címfát alakítja a hálózati dinamizmus által okozott változásokhoz, hogy az konvergáljon az 1.1 Tézisben bemutatott algoritmus által épített kiegyensúlyozott fához.*

Az algoritmus célja a kapcsolatminőség megőrzése; a mély címfák ebből a szempontból hátrányosak. Az algoritmus folyamatosan figyeli a csomópontok környezetét, hogy jobb csatlakozási pontokat találjon a címfákon, amelyek a hálózat-dinamizmusnak köszönhetően jelenhetnek meg. Az 4.3 Következmény alapján, egy csomópont újracsatolása a teljes részfájának kapcsolat-minőségét javítja. Azért hogy elkerüljük a szülő-gyerekek egyeztetésekből eredő hálózati túlterhelést, egy l hangolási paramétert alkalmaztam, ahol $1 < l \leq l_{max}$. A gyökér csomópont az 1. szinten helyezkedik el, és l_{max} egy fix paraméter, a maximális megengedett famélység. Az l paraméter aktuális értéke felett avagy a fában mélyebben elhelyezkedő csomópontoknak kötelező

új csatlakozási pontot keresniük.

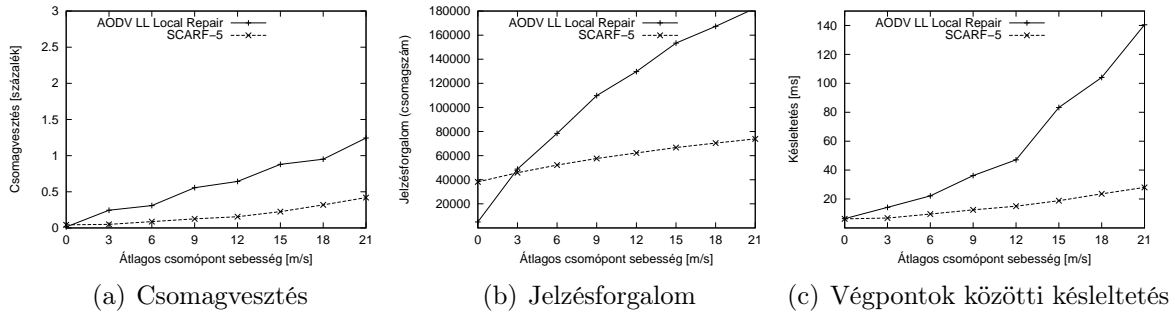
4.1.5. Szimulációk

Tézis 1.5 (Skálázhatósági szimulációk [J2]): *Alapos szimulációkkal megmutattam, hogy a javasolt SCARF címzési és útválasztási módszer ugyanazt a felajánlott felhasználói forgalmat jelentősen alacsonyabb jelzésforgalmi terheléssel vezeti el mint az AODV ad hoc útválasztó protokoll, abban az esetben, ha a hálózatban sok csomópont van, a mobile csomópontok dinamikusak, és számos párhuzamos felhasználói adatfolyamot kell fenntartani. Mindemellett az adatcsomagok alacsonyabb késleltetést szenvednek.*

A szimulációkat az ns-2 [13] hálózatszimulátorban futtattam. Jelentős számú forgatókönyvet elemeztem a mobilitás, hálózat terhelés és hálózatméret paraméterek minden egyes értéke mellett. A SCARF architektúrát egy, három és öt gyökér csomóponttal vizsgáltam. A SCARF hatékonyságát az AODV protokollal [11] hasonlítottam össze, az egyik leghatékonyabb és legjobban skálázható igény alapú ad hoc útválasztó protokollal. A szimulációimban az AODV egy hivatalosan RFC kompatibilis változatát használtam, melyet az Uppsala University [12] készített. Vizsgáltam az AODV számos különböző paramétereit és beállítási lehetőségét és arra következtettem, hogy az AODV-Link Layer Local Repair (AODV-LL-LR) a leghatékonyabb AODV variáns. Az AODV-LL a vezeték nélküli kapcsolatok elérhetőségét az adatkapcsolati réteg visszajelzéseiből ismeri az adatcsomagok elküldésekor, ezáltal nem használ Hello üzeneteket. Másrészt a „Local Repair” tulajdonság alapján egy köztes csomópont buffereli a csomagokat amíg javítást nem talál az útvonalhoz egy kapcsolathiba után. Az adatkapcsolati réteg visszajelzéseit a SCARF-ban is használtam a kapcsolathibák mihamarabbi észleléséhez. Az adatkapcsolati rétegben ugyanazon paramétereket használtam mindkét protokollnál.

A szimulációkban 30, 50 és 70 párhuzamos konstant bit rátájú UDP adatfolyamot generáltam, melyek a szimuláció 50. és 100. másodperce között indultak a 400 másodperces szimulációkban. Az adatfolyamoknak alacsony sebességük (1 kbit/sec, 2 packets/sec) volt azért, hogy ne terheljem az adatkapcsolati réteget és ezáltal lehetővé váljon sok adatfolyam felépítése különböző forrás-cél párokkal. Egy ilyen forgatókönyv jobban terheli az útválasztási algoritmust mint az adatkapcsolati réteget.

A tézisfüzetben csak a legmagasabb terheléssel szimulált ábrákat mutatom be. Ahogy a mobilitás, a forgalmi terhelés és a hálózatméret növekszik, a SCARF egyre hatékonyabbá válik, egyértelműen megelőzve az AODV-LL-LR-t. Az 5. ábra egy olyan forgatókönyvet mutat, ahol 80 csomópont mozgott egy 800 x 800 négyzetméteres szimulációs területen. A csomópontok elméleti hatótávolsága 250m. Az eredmények



5. ábra. ns-2 szimulációk 80 csomóponttal és 70 adatfolyammal

alapján 15 m/s átlagos mozgási sebességnél az én megoldásom öt gyöker csomóponttal akár 50 százalékkal kevesebb jelzésforgalmi üzenetet generál mint az AODV-LL-LR, miközben jobb csomagkézbesítési arányt és alacsonyabb csomagkésleltetéseket generál. Megjegyezném, hogy más ad hoc útválasztási protokollokkal szemben a csomagkézbesítési arány az AODV-LL-LR és a SCARF esetében is kimagaslóan jó erre a hálózatméretre és mobilitási értékekre.

4.1.6. Internet kapcsolat a SCARF számára

Az eddigiekben egy ad hoc útválasztási megoldást javasoltam, melynek alacsonyabb jelzésforgalmi terhelése, alacsonyabb késleltetése és magasabb csomagkézbesítési aránya van mint más ad hoc útválasztási megoldásoknak, ha a csomópontok mozgása a hálózatban viszonylagosan fix referencia pontok körül történnek. De miért ne lehetnének ezek a pontok különleges, dedikált csomópontok, például vezeték nélküli elérési pontok?

A 802.11 vezeték nélküli hálózatok rendszeresen ún. „szervezett” módban működnek, ahol egy Internet „Access Point” (AP) folyamatosan elérhető. A 802.11 a/b/g egyik fő problémája a rövid hatótávolság. A rádió jelek gyengülése sokkal kritikusabb épületekben, mivel a tereptárgyak visszaverik és elnyelik a őket. Erre a hátrányra egy megoldás lehet egy hálózat-rétegbeli kiterjesztése az AP hatóterületének, azaz a mobil csomópontok több ugráson keresztül érhetik el az AP-t. A SCARF architektúra kiválóan alkalmas erre a szolgáltatásra.

Tézis 1.6 (Architektúra [J2]): *Javasoltam a SCARF architektúrára egy kiterjesztést, mellyel a SCARF többugrásos Internet elérést képes biztosítani. Ehhez a következő lépéseket kell megtenni:*

- I. A SCARF címfák gyökér csomópontjait Internet elérési pontokhoz kell csatlakoztatni;

- II. Az elérési pontokat egy vezeték nélküli tartományba (*Wireless Domain, WD*) kell csoportosítani, hogy kihasználhassuk a több gyökércsomópont által nyújtott előnyöket;
- III. Egy *WD*-n belül az *AP*-k egy megbízható, alacsony késleltetésű, vezetékes hálózattal legyenek összekötve, mellyel tovább csökkenthető a vezeték nélküli hálózat által generált jelzésforgalom;
- IV. Egy tartomány átjárót (*Wireless Domain Gateway, WDG*) kell kineveznünk, amellyel elrejtjük a vezeték nélküli tartományt az Internettől, és amellyel transzparens és hatékony mobilitáskezelést tudunk biztosítani a tartományon belül.

Továbbá megmutattam, hogy a *SCARF* támogatja az Internet elérést, avval hogy megvalósítja az Internet átjáró felderítését, az IP címek konfigurációját, és az Internet kapcsolatok felépítését.

Az architektúra beilleszthető az IP mobilitás keretrendszerébe [16]. A *SCARF* segítségével a vezeték nélküli csomópontok többgrásos Internet elérése csak olyan módosításokat igényel, amely nem rontja azt a vezeték nélküli teljesítményt, melyet az 1.5 Tézisben mutattam be.

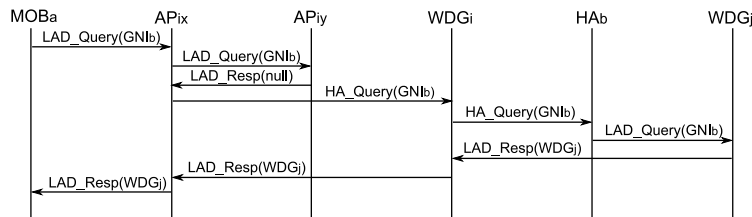
A vezetékes kapcsolat az *AP*-k között lehetővé teszi a gyökér csomópontok számára hogy egymással kommunikáljanak amennyiben a *LAD*-ot érintő jelzésüzenetet kapnak. Következésképpen a mobil csomópontoknak nem szükséges az összes *AP* megcímzése, amennyiben a *LAD*-ot el akarják érni. A címtár szolgáltatás jelzésüzeneteit bármely *AP*-nek címezhetik.

A *WDG* célja, hogy a több gyökérrel rendelkező architektúrát az Internethez csatolja és emellett tartományok közötti mobilitáskezelést biztosítson. Az Internet elérés miatt praktikus együttműködést biztosítani a Mobile IP technológiával [14, 15]. Ezáltal azon csomópontok, melyek nem *SCARF* képesek, kommunikálni tudnak *SCARF* tartományon belüli mobil csomópontokkal. Ahhoz, hogy a *SCARF* csomópontokat az IP rendszerhez csatlakoztassuk, minden csomópont a *GNI* címét az otthoni IP címként konfigurálja fel. Az Internet elérés fő feladatait a *SCARF* a következőképpen oldja meg:

- *Internet átjáró felderítése*: mivel a *SCARF* hálózatban a gyökér csomópont információinak továbbítása a címfák karbantartásának velejárója, ezért a megoldásomnak nincs szüksége további mechanizmusokat megvalósítania, mint az átjáró meghirdetése vagy az átjáró lekérdezése;
- *Cím konfiguráció*: a *WDG* a saját IP címét mint helyfüggő cím („care-of-address”, CoA) rendeli hozzá a tartományán belüli mobil csomópontokhoz. Mikor egy bejövő csomag a *WDG*-hez kerül, a *SCARF* architektúra kezeli a csomag továbbítását a *LAD* szolgáltatás segítségével, amely leképezi a *GNI*-t (azaz az otthoni

IP címet) a friss LNA címekre;

- *Kapcsolatfelépítés az Internet felé:* a 6. ábrán látható a kapcsolatfelépítés címlekerdezés fázisa. A MOB_a mobil csomópont a WD_i tartományban kapcsolatot próbál létesíteni a b csomóponthoz, melynek azonosítója GNI_b . Mivel az i tartomány címtár szolgáltatásában b nem található, a kérést továbbítani kell az otthoni ügynöknek HA_b , mely a b csomópontért felelős. HA_b átirányítja a kérést a WD_j tartomány felé, ahol a b csomópont jelenleg tartózkodik. A LAD ebben a tartományban végrehajtja a címlekepezést, és a válasz a kezdeményezőhöz továbbítódik.



6. ábra. Tartományok közötti kommunikáció: egy SCARF csomópont megtalálása az Interneten keresztül

A tartományba beérkező kommunikáció és a tartományok közötti mobilitás részletes leírása a disszertációban olvasható.

4.2. Adaptív elosztott hash tábla

Téziscsoport 2: *Javasoltam egy továbbfejlesztést a CAN elosztott hash táblához, mely lehetővé teszi hogy modern P2P DHT protokolloknál kikeresési teljesítményben és hatékonyságban is jobban teljesítsen.*

A CAN (Content Addressable Network) az egyik elsőként publikált DHT algoritmus [17]. A CAN a csomópontokat és az erőforrásokat egy d -dimenziós azonosító-térre képezi le melyet egy d -tóruszon épít fel. Az azonosító teret d -dimenziós zónákra osztjuk fel, melyeket csomópontokhoz rendelünk.

A „Long-Range” CAN (LR-CAN) egy olyan algoritmus, mely a CAN megoldásban hosszútávú kapcsolatokat alkalmaz, egy népszerű megoldást skálázható DHT építésére. Emellett az algoritmus képes a hálózatmérethez egy költség-korlát függvényen keresztül alkalmazkodni, melyet $SR(N)$ jelöl a továbbiakban. A jelenlegi DHT algoritmusokban a hosszútávú kapcsolatok száma implicit módon változik a hálózatmérettel és arányos a hálózatméret logaritmusával. Ezzel szemben az LR-CAN a hosszútávú kapcsolatok számát az imént említett költség-korlát függvényen keresztül vezérli. A megfelelő függvénnyel az LR-CAN beállítható úgy, hogy a Chordnál és

a Kademliánál is jobb teljesítményt nyújtson. Mindemellett a CAN többdimenziós azonosító tere, mely minden dimenzió mentén kétirányú, az említett DHT protokollokkal összehasonlítva alacsonyabban tartja az LR-CAN karbantartási forgalmát.

Tézis 2.1 (Az adaptív LR-CAN algoritmus [J1, C1]): *Javasoltam egy algoritmust, LR-CAN néven, mely hosszútávú kapcsolatokat használ a CAN-ben a kikeresési költség csökkentése érdekében. Egy csomópont hosszútávú kapcsolatainak száma adaptívan változik egy költség-korlát függvény alapján (az összefüggéseket a 2.2 Tézis tárgyalja). Egy csomóponton az algoritmus a következő lépéseket hajtja végre:*

- I. a csomópont megbecsüli az átlagos kikeresési költséget (\widehat{R}_{avg}) és a csomópontok számát (\widehat{N}) a hálózatban;
- II. a csomópont meghatározza a költség-korlátot \widehat{N} behelyettesítésével ($SR(\widehat{N})$);
- III. ha $\widehat{R}_{avg} \geq SR(\widehat{N})$, a csomópont felépít egy új hosszútávú kapcsolatszintet a megfelelő számú kapcsolattal;
- IV. a csomópont az I. lépéssel folytatja amíg $\widehat{R}_{avg} < SR(\widehat{N})$ nem teljesül.

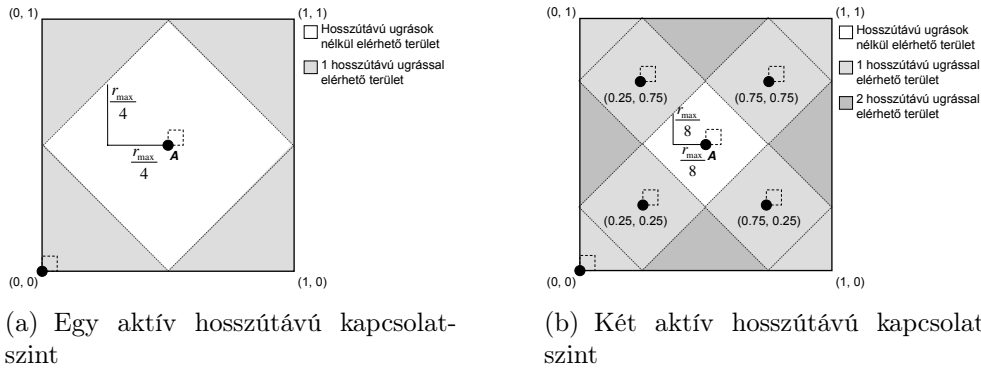
Mindegyik csomópont végrehajtja ezen lépéseket egy „stabilizációs” periódusban, mely fenntartja és visszavonja a kapcsolatokat amennyiben szükséges.

Az algoritmus alapelve, hogy egy csomópont képes megbecsülni a hálózat méretét egyénileg, bármilyen kiszolgáló segítsége nélkül. Ez a képesség elsősorban azért lehetséges, mert a csomópont-azonosító és a csomópont-hoz tartozó zóna függetlenek a CAN azonosító terében. Valójában egy csomópont a becsatlakozáskor sorsol egy véletlen d -dimenziós koordinátát, kikeresi, és felkéri a tulajdonosát, hogy két egyenlő részre ossza szét a saját zónáját. Valószínűségi alapon mindig a legnagyobb zónák osztódnak, így az azonosítótér felosztása konvergálni fog a teljesen egyenletes felosztáshoz.

E feltételezés alapján a hálózat méret (\widehat{N}) többféle módon megbecsülhető. Az egyik legpontosabb megoldást az azonosítótér legtávolabbi pontjához történő kikeresés adja, mely során megmérjük az ugrástávolságot (\widehat{R}_{max}); fontos, hogy az útválasztás csak vízszintesen és függőlegesen tud haladni és ennek megfelelően kell meghatározni a legtávolabbi pontot. Ha az átlagos kikeresési költség (\widehat{R}_{avg}) a költség-korlát függvény által megengedett értéknél magasabb, egy új hosszútávú kapcsolatszintet vezetünk be. Ezáltal az LR-CAN az átlagos keresési költséget a költség-korlát függvény értékei alatt tartja.

$$\widehat{R}_{avg} = \widehat{R}_{max} \cdot m_{\frac{avg}{max}} < SR(\widehat{N}), \quad (4.1)$$

ahol $m_{\frac{avg}{max}}$ a tapasztalati hányadosa az R_{avg} és az R_{max} változóknak (egy pontos becslés szimulációkkal igazolva). A hálózati csomópontok konzisztensen viselkednek,



7. ábra. Kétdimenziós azonosítótér hosszútávú kapcsolatokkal

így az egyéni döntéseik érvényesek lesznek az egész hálózatra nézve, mely a teljes hálózat teljesítményét javítja.

A továbbiakban a hagyományos CAN útválasztásra *rövidtávú útválasztás* néven utalok, mert kizárólag rövidtávú kapcsolatokat használ; ennek megfelelően *hosszútávú útválasztásnak* hívom ha hosszútávú kapcsolatok továbbítják a kikeresést. Az LR-CAN útválasztása két fázisra bontható: először egy adott kulcs értéket hosszútávú útválasztással közelítünk meg, majd rövidtávú útválasztással jutunk el a kulcs tulajdonosához. A 7. ábrán láthatjuk a hosszútávú kapcsolatok hatását az LR-CAN útválasztásban. A fehér négyzet mutatja az alteret, ahol csak rövidtávú útválasztás használunk, amennyiben az A csomópont indítja a lekérdezést. Ha a lekérdezés az azonosítótér egy távoli pontjába indítjuk, A a hosszútávú kapcsolatait alkalmazza először, és csak ezután a rövidtávú kapcsolatokon alapuló hagyományos CAN útválasztást, ezáltal csökkentvén annak hatáskörét. A 7(a). ábra mutatja az esetet, mikor csak egy hosszútávú kapcsolatszint van aktiválva, míg a 7(b). ábrán már két szintet aktiváltunk. Látható, hogy az első szinten csak egy kapcsolat van, míg a többi szinten 2^d . Ennek oka a tórusz tulajdonság: az azonosítótér „sarkai” egy pontnak felelnek meg a tóruszon.

Az eredeti CAN kikeresési költsége $O(dN^{\frac{1}{d}})$, melyet $O(d)$ kapcsolatszámmal biztosít. A következő tézisben az LR-CAN teljesítményét elemzem az eredeti algoritmusához viszonyítva.

Tézis 2.2 (LR-CAN kikeresési költségének analízise [J1, C1]): *Megmutatom, hogy az LR-CAN algoritmus a CAN kikeresési költségét $O(\log N)$ lépésszámra csökkenti $O(\log N)$ számú kapcsolat alkalmazásával.*

Habár a hagyományos CAN architektúrában a d paraméterrel biztosítható a $\log N$ lépésszámú kikeresési költség, egy adott N értékhez a megfelelő d értéket kell beállítani. Ha N jelentősen megváltozik, d értékét is módosítani kell, mely az azonosítótér

költséges ismételt leképezését vonja maga után. Az LR-CAN módszerben a hosszútávú szintek száma, melyet L jelöl, egy hatékony megoldást vezet be az $O(\log N)$ költség biztosítására. Ennek eredménye, hogy L bevezetése kiiktatja d jelentőségét az LR-CAN kikeresési költségében. Megjegyezném, hogy a többdimenziós azonosítótér építése továbbra is előnyös, mert jobb hibátűrést biztosít a rendszernek (több rövidtávú kapcsolattal); mindemellett a hosszútávú kapcsolatok esetleges pontatlansága kevésbé hátrányos a kikeresési teljesítmény számára mint egy egydimenziós azonosító-térben.

Amennyiben az útválasztás csak vízszintesen vagy függőlegesen haladhat, két \mathbf{p} és \mathbf{q} pont között, ahol $(\mathbf{p}, \mathbf{q} \in \{\mathbb{R}^d \mid [0,1]\})$, a távolság az azonosító-téren $\frac{1}{2} \cdot d$ dimenzióban a maximális távolság $r_{max} = \frac{d}{2}$ (lásd a 7. ábra). Ahhoz hogy ezt a távolságot ugrásszám-ban kapjuk meg, a maximális távolságot meg kell szoroznunk az azonosítótér „felbontásával”, amely $N^{\frac{1}{d}}$. Mindegyik új hosszútávú kapcsolatszint felezi a maximális távolságot. Így a rövidtávú útválasztás átlagos úthossza (mely nem tartalmazza a hosszútávú útválasztás lépéseit):

$$R_{avg} = \frac{r_{max}}{2^{L+1}} \cdot N^{\frac{1}{d}} \cdot m_{\frac{avg}{max}}. \quad (4.2)$$

Mivel az LR-CAN megoldásban $O(\log N)$ kikeresési költséget akarok biztosítani, más DHT megoldásokhoz hasonlóan, $SR(N) = \frac{1}{c} \cdot \log_2 N$ függvényt javaslom mint a rövidtávú útválasztás felső korlátja, ahol c a költség-korlát faktor, egy tetszőleges pozitív valós szám. A kikeresési költség kifejezéséhez először le kell vezetnünk a kapcsolatszintek számát N függvényében. Az 4.2 egyenletben az R_{avg} változó $\frac{1}{c} \log_2 N$ kifejezésre történő helyettesítésével következőt kapjuk:

$$\begin{aligned} L &= \log_2 \frac{r_{max} N^{\frac{1}{d}} m_{\frac{avg}{max}} c}{2 \log_2 N} = \log_2 r_{max} N^{\frac{1}{d}} m_{\frac{avg}{max}} c - \log_2 2 \log_2 N \\ &= O(\log N) - O(\log \log N) = O(\log N) \end{aligned} \quad (4.3)$$

mivel d konstansnak tekinthető az algoritmusomban. A kikeresések átlagos üzenetköltsége a hosszú- és rövidtávú útválasztási költség összegével fejezhető ki:

$$LR_{avg} + R_{avg} = O(L) + O(\log N) = O(\log N) \quad (4.4)$$

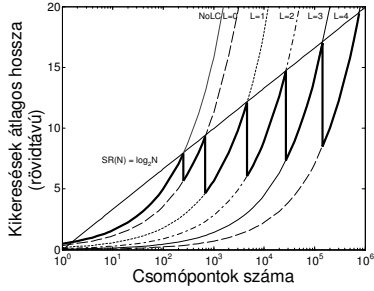
Az LR-CAN megoldásban a csökkentett kikeresési költséget a CAN algoritmushoz képest magasabb kapcsolat-fenntartási költség árán érjük el:

$$LC_{avg} + SC_{avg} = 2d + 1 + L2^d = O(d) + O(L2^d) = O(L) = O(\log N) \quad (4.5)$$

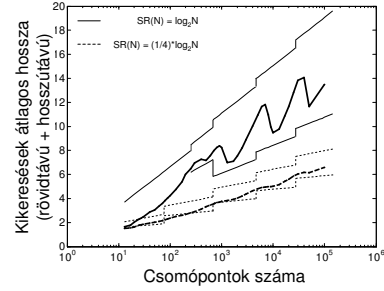
Természetesen, L változónak van egy implicit felső korlátja, mely az azonosítótér „felbontásától” függ. Nincs értelme új hosszútávú koordinátákat meghatározni, ha csak ugyanazon csomópontokra tudjuk leképezni őket. Következésképp L értékeit a

$(-1 \leq L \leq \lfloor \log_2 N^{\frac{1}{d}}/2 \rfloor, L \in \mathbb{Z})$ tartományon határozom meg; -1 azt az esetet jelöli mikor nem használunk hosszútávú kapcsolatokat. A kívánt költség-korlát függvényt a c paraméteren keresztül fejezzük ki, habár L értéktartománya impliciten meghatározza c értéktartományát, azaz túl magas c értéket sincs értelme megadnunk. A 2.1 Tézisben bemutatott megoldás biztonságosan kezeli a tetszőlegesen magas c értékeket és csak akkor helyez el hosszútávú kapcsolatokat ha előnyös.

A 8(a). ábrán láthatjuk a rövidtávú úthossz növekedését a hálózat függvényében egy kétdimenziós azonosító-térben a bemutatott matematikai modell alapján. Egy új szintet akkor helyezünk el, ha a $f(N) = dN^{\frac{1}{d}}$ görbék metszik a rövidtávú útválasztás kívánt átlagát, azaz $SR(N) = \frac{1}{c} \cdot \log_2 N$ függvényt. A 8(a). ábrán $c = 1$. $SR(N)$ görbét elérve, az úthossz lecsökken, míg a kapcsolatok száma növekszik. A különböző görbéknek megfelelő szintek számát L jelöli. Az átlagos elméleti rövidtávú útválasztási úthossz a vastag vonal mentén változik.



(a) Analitikus eredmények R_{avg} változására N függvényében



(b) Szimulációs eredmények az átlagos úthosszra különböző SR függvények mellett

8. ábra. Analitikus és szimulációs eredmények összehasonlítása

A matematikai elemzés tartalmaz olyan feltételezéseket, melyek szimulációkban vagy a való világban csak megközelítőleg teljesülnek. A szimulációs eredményeket a vastag vonalú görbe mutatja a 8(b). ábrán két különböző c érték mellett ($c_1 = 1$ és $c_2 = 4$), melyek két költség-korlát függvényt határoznak meg. Ezen az ábrán összehasonlíthatjuk a szimulációs és analitikus eredményeket, mivel a vékony vonalak az elméleti alsó és felső korlátot mutatják a szimulációk számára, körbezárják a megfelelő vastag görbét. A vékony görbéken megfigyelhető ugrások a hosszútávú útválasztási költségnek (LR_{avg}) köszönhető, amely minden új hosszútávú kapcsolatszint bevezetésénél változik. Ez a költség csak az L és d változóktól függ (lásd 2.4 Tézis). Az elméleti görbék azonos méretű kétdimenziós zónákat és ugyanazon L értékeket feltételeznek minden csomópontra, melyek természetesen nem minden esetben érvényesek valós környezetben.

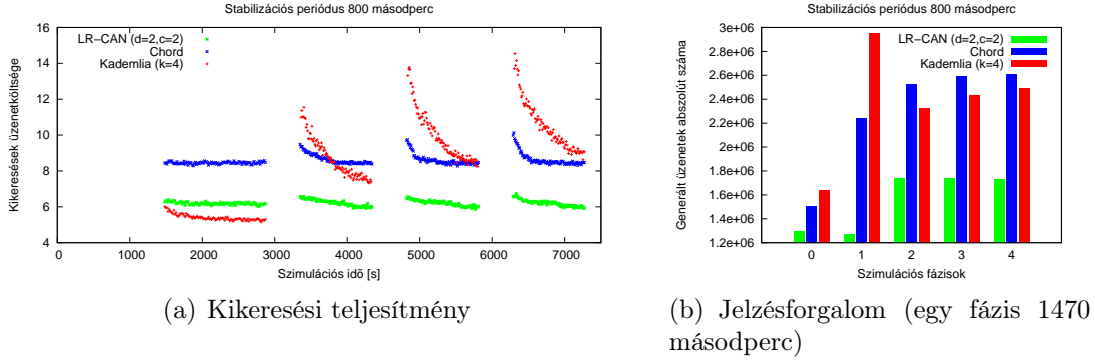
Tézis 2.3 (LR-CAN, Chord és Kademia teljesítmény-összehasonlítás [J1]): *Alapos szimulációkkal megmutattam, hogy elismert modern P2P kikereső algoritmusokhoz képest (mint a Chord és a Kademia) az LR-CAN gyorsabb kikereséseket és a kikeresések számára alacsonyabb szórást biztosít dinamikus hálózatokban, és mindezt alacsonyabb jelzésforgalommal teszi.*

Több dimenzióban a kikeresések gyorsíthatóak. Egy egydimenziós térben történő lineáris keresés $O(N)$ kikeresési költséggel bír, ezzel szemben a CAN $O(dN^{1/d})$ költséget biztosít egy d -dimenziós térben; magasabb d értékek alacsonyabb lineáris kikeresési költséget adnak. Következésképp ha egy többdimenziós megoldás, mint az LR-CAN egy adott kikeresési költséget biztosítana, kevésbé szigorú követelményei lehetnek a hosszútávú kapcsolatok pontosságára, mint például az egydimenziós Chord megoldásban. Az LR-CAN így kevesebb jelzésforgalmat generálhat a hosszútávú kapcsolatok fenntartásához.

A szimulációimban különös figyelmet fordítottam a DHT megoldások *igazságos összehasonlítására*. A szimulációk célja annak vizsgálata, hogy a dinamizmus hogyan befolyásolja a kikeresések üzenetköltségét, amelyet alapvetően a hosszútávú kapcsolatok pontossága határoz meg. Mivel a meghibásodások és a csomópontok elegáns kilépései a hosszútávú kapcsolatok karbantartásának szempontjából egyenértékű, ezért nem implementáltam a Chord és az LR-CAN meghibásodást lekezelő szolgáltatásait. Mivel a Kademia igény alapú megoldása a meghibásodásokat kiegészítő szolgáltatások implementálása nélkül is megoldja, ezért a 9. ábrán található összehasonlítás csak részben érdelemes. A hibatűrés vizsgálata a DHT megoldásokkal kapcsolatos kutatások egy másik fontos témája [19].

A szimulációkat egy eseményalapú P2P kikereső algoritmusok számára készített szimulátorral végeztem [21, 22]. A szimuláció kezdetén 10.000 csomópont lépett be a hálózatba és tárolt el véletlenszerű tartalmat. A hátralévő szimulációs idő négy egyenlő hosszú 1470 másodperces részre lett osztva. Az első részben minden csomópont 10 kikeresést indított az eltárolt tartalmak valamelyikére. A hátralévő három részben a csomópontok 32 százalékát kicseréltem egy időablakban, majd ismét 10 kikeresést indított minden csomópont. A szimulációs szakaszok egyenlő hossza és ugyanolyan terhelése lehetővé teszi, hogy megfigyeljük és összehasonlítsuk a különböző szakaszokban a dinamizmus hatását a rendszer-teljesítményre.

Ezen DHT megoldások teljesítménye számos paraméterrel hangolható. A stabilizációs periódus az egyik leglényegesebb: meghatározza milyen gyakran kell egy csomópontnak ellenőriznie kapcsolatainak pontosságát. Egy hosszú stabilizációs periódus a dinamizmus gyenge toleranciáját eredményezi, ugyanakkor az alacsony periódus nagyon magas jelzésforgalmat eredményez. Az itt bemutatott ábrák egy 800 másodperces stabilizációs periódussal futtatott szimuláció eredményeit mutatják, mely a ki- és belépések időablakának kétszerese. A Kademia k paramétere szintén jelentős hatással bír a dinamika szempontjából. Minél nagyobb a k érték, annál jobb a



9. ábra. Keresési teljesítmény és az ennek megfelelő jelzésforgalom dinamikus hálózatban

dinamizmus toleranciája és egyben annál nagyobb a jelzésforgalom.

A 9. ábrán láthatjuk, hogy a dinamizmusnak a kikeresési költség szórásában az LR-CAN esetén van a legenyhébb hatása. Mindemellett az LR-CAN a leggyorsabb kikereséseket tudja biztosítani eme protokollok között, azonban a kikeresési-költség függvény kicserélésével (azaz c értékének módosításával) a kikeresési teljesítmény tovább hangolható. Az LR-CAN mindeközben a legkevesebb jelzésforgalmat generálja, mivel a hosszútávú kapcsolatok szervezése sokkal hatékonyabb a speciális CAN azonosítótérnek köszönhetően.

A rendszer-paraméterek további hatásait a disszertáció vizsgálja részletesebben.

Tézis 2.4 (Jelzésforgalom minimalizálás [J1]): *Az 2.2 Tézis eredményei alapján meghatároztam az L változó optimumát a teljes hálózat számára, a kikeresési forgalom és a kapcsolat-karbantartási jelzésforgalom összegzett értékének minimalizálásához.*

A javasolt számításoknak információra van szüksége a hosszútávú kapcsolatok karbantartásának gyakoriságáról (λ_p), a hálózatt méretről (N) és az átlagos kikeresési rátáról (λ_l) az egész hálózatra nézve. Ezen bemeneti paramétereket használva modellezhető a karbantartási és a kikeresési üzenetköltség minden időegységben:

$$M_{cost}(L) = \begin{cases} 2 \cdot N \cdot \lambda_p \cdot (1 + L), & \text{ha } -1 \leq L < 0 \\ 2 \cdot N \cdot \lambda_p \cdot (1 + L \cdot 2^d), & \text{ha } L \geq 0 \end{cases} \quad (4.6)$$

$$L_{cost}(L) = N \cdot \lambda_l \cdot (LR_{avg}(L) + \frac{d}{2} N^{\frac{1}{d}} \cdot m_{max}^{avg}), \quad \text{ha } L \geq -1 \quad (4.7)$$

$$LR_{avg}(L) \approx \begin{cases} 0.5 \cdot (1 + L), & \text{ha } -1 \leq L < 0 \\ 0.5 + d \cdot 0.343 \cdot L, & \text{ha } L \geq 0, d = 1, 2 \end{cases} \quad (4.8)$$

L optimuma:

$$L_{opt} \approx \arg \min_{\left(\left[-1, \log_2 N^{\frac{1}{d}} / 2 \right], L \in \mathbb{R} \right)} (M_{cost}(L) + L_{cost}(L)) \quad (4.9)$$

A gyors kikeresések fontosak egy DHT megoldásban, de biztosításuk jelentős jelzésforgalommal járhat. Lehetnek esetek amikor az alacsony kikeresési forgalom miatt bizonyos hosszútávú kapcsolatok fenntartása felesleges, mivel az többbe kerül mint amennyi hasznot hoz. Az eredeti LR-CAN algoritmus néhány feltételezése és módosítása mellett létrehozhatunk egy olyan megoldást (SIGMIN) ami a hosszútávú kapcsolatok számát nemcsak a hálózatmérethez (N), hanem a kikeresési rátához (λ_l) is igazítja a 2.4 Tézis módszere alapján, L_{opt} meghatározásával.

A probléma így a hosszútávú kapcsolatszintek (L) optimális számának megtalálása a hálózatméret, a hosszútávú kapcsolatok karbantartási gyakorisága (λ_p) és a kikeresési ráta (λ_l) függvényében. N meghatározásának módszerét korábban a 2.1 Tézisben bemutattem. A hosszútávú kapcsolatok karbantartásának gyakorisága, azaz a stabilizációs periódus (λ_p) egy globális hálózati paraméter. A λ_l számára azonban meg kell valósítani egy adatgyűjtő technikát, amely a kikeresési rátáról gyűjt adatokat az egész hálózatból, vagy legalább egy jelentős részéből. Az én megoldásomban az adatgyűjtést a baloldali legalsó csomópont kezdeményezi, amely egy dedikált őscsomópont lehet a CAN hálózatban amennyiben a hálózat indításától kezdve jelen van. Az adatgyűjtésnek $O(N)$ költsége van. Az adatgyűjtés befejeztével ez a csomópont kiértékeli a bemenő paramétereket és meghatározza L optimális értékét, mely aztán üzenetszórással az egész hálózatba eljut. Következésképp, itt meg kell változtatnunk L korábbi meghatározási módszerét, melyet a 2.1 Tézis mutatott be és amely a csomópontok egyéni döntésein alapszik egy olyan megoldásra, ahol egy csomópont dönti el L értékét összegyűjtött globális információk alapján.

$M_{cost}(L)$ magában foglalja a hosszútávú kapcsolatok ellenőrzésének költségét, a hálózatméretet (N), a karbantartási gyakoriságot (λ_p) és a hosszútávú kapcsolatok számát ha L szintet és d dimenziót használunk. $L_{cost}(L)$ tartalmazza a hálózatméretet, a kikeresési rátát (λ_l) és a hosszú- és rövidtávú útválasztás átlagos költségét (ez utóbbi a 2.2 Tézis tárgyalja). A hosszútávú költségre ($LR_{avg}(L)$) egy pontos közelítést adtam a 4.8 egyenletben, melyet a következő formulákkal vezettem le:

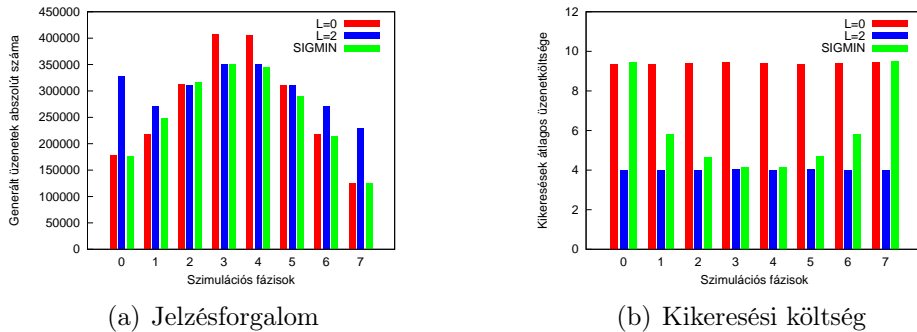
$$c(L) = 2^L \cdot \left(\frac{1}{2} + \frac{L}{2} - \sum_{i=2}^L \left(\frac{1}{2^i} + \sum_{j=1}^{\lfloor \log_2(i-1) \rfloor} \frac{1}{8^j} \right) \right), \quad \text{ha } L \geq 1 \quad (4.10)$$

$$LR_{avg}(L) = \begin{cases} \frac{1}{2}, & \text{ha } L = 0 \\ \frac{(c(L)-1)}{2^L} + \frac{1}{2^{L+1}}, & \text{ha } d = 1, L \geq 1 \\ \frac{(c(L)-1)}{2^{L-1}} + \frac{1}{2^{2L+1}} - 4 \cdot \sum_{i=1}^{\lfloor \log_2(L+1) \rfloor} \frac{1}{4^i} \cdot \sum_{i=1}^{\lfloor \log_2(L) \rfloor} \frac{1}{4^i} & \text{ha } d = 2, L \geq 1 \end{cases} \quad (4.11)$$

A képleteket az útválasztási zónák súlyozott átlagával vezettem le a 7. ábra alapján. A 4.10 és 4.11 egyenletek a 4.8 közelítés érvényességét bizonyítják maximum két dimenzióra. Az LR-CAN és a SIGMIN számára két dimenzió elegendő, hiszen a keresési teljesítmény nem függ a dimenziószámtól, ahogy a 2.2 Tézis is említette.

Az 4.9 egyenlet megoldása megadja L optimális értékét, mely egy valós szám lesz. Azonban egy csomóponton L egész értékű lehet csak. Emiatt egy csomópont egy egyszerű véletlen módszer alapján az optimális L érték alatti vagy fölötti egész számot választja, ezáltal L globális átlaga az optimálishoz közeli értéket vesz fel.

L optimális értékének megválasztása a P2P hálózatban egy jövőre vonatkozó döntés, azaz a hálózat teljesítményét a jövőben kell optimalizálnunk a múlt adatai alapján. Ezért a csomópontoknak részletes információval kell szolgálniuk a kikeresési aktivitásukról rendszeresen (pl. naponta), nem elég egy hosszú időintervallum egyszerű átlagát figyelembe venni.



10. ábra. A kikeresési ráta hatásának szimulációja a SIGMIN algoritmusban

A szimulációk mellyel a 10. ábra grafikonjai születtek egyenlő időszakokból épült fel konstans csomópont populációval. Csak a kikeresési ráta változott az időszakokban folyamatosan növekedve a 0-tól a 3-as időszakig, a 3-as és 4-es időszakban nem változott, míg 4-től 7-ig lineárisan csökkent. Az egyik esetben L statikusan 0 értékre volt állítva az egész hálózatban, egy másik esetben 2 volt L értéke, míg a harmadik esetben a SIGMIN algoritmus futott és határozta meg L értékét dinamikusan. A

SIGMIN a szimuláció elején nem használt hosszútávú kapcsolatokat. A hosszútávú kapcsolatok felállítása igényel némi nem elhanyagolható jelzésforgalmat, mely tisztán látszik az $L = 2$ változat esetén az első időszakban. Az $L = 0$ esetben kevesebb kapcsolatot kell felállítani és karbantartani. A SIGMIN teljesítményében ugyanúgy megmutatkozik ez a járulékos költség, mivel a növekvő kikeresési ráta újabb hosszútávú kapcsolatokat iktat be az 4.6 – 4.9 egyenletek alapján. Ezt láthatjuk az 1-es és 2-es időszakban, ezért a 2-es fázisban a SIGMIN kevéssel több jelzésforgalmat generál mint a másik két megoldás (ezt az alkalmi költséget a 4.6 – 4.9 egyenletek nem kezelik). A SIGMIN algoritmus kimagasló tulajdonsága, hogy jelzésforgalom szempontjából mindig a hatékonyabb megoldáshoz tart.

5. Az eredmények alkalmazhatósága

A tézisfüzet első részében bemutattam egy útválasztási megoldást, amely az ad hoc hálózat helyének körzetében elhelyezkedő fix csomópontokra épít. Az algoritmus robotus ad hoc kommunikációt, alacsony csomagkésleltetést, és alacsony jelzésforgalmat biztosít ha a hálózatot sok csomópont alkotja, a mobilitás jelentős, és számos kommunikációs igény kerül a rendszerbe. A fix csomópontok Internet elérési pontokhoz csatlakoztathatóak és vezetékes összeköttetés segítségével tovább javíthatják a vezeték nélküli hálózat útválasztási hatékonyságát. Ennek eredményeként, egy WiFi technológiával ellátott épület Internet eléréssel teljesen lefedetté tehető kevesebb elérési pont telepítésével. Mindemellett, a csomópontok helyi kommunikációja P2P jelleggel megoldható az elérési pontok kikerülésével.

A második tézis csoport az LR-CAN algoritmust mutatja be, mely az eredeti CAN DHT megoldást több szempontból is feljavítja. Először is képes a hálózatmérethez alkalmazkodni. Az eredeti CAN algoritmussal a jelentősen változó hálózatméret kezelése nagyon költséges. Az LR-CAN adaptivitása biztosítja, hogy a csomópontok mindig a megfelelő számú kapcsolatot tartják fenn a hálózat méretének változásával. Ez a tulajdonság gyorsítja a kikereséseket és csökkenti a költségüket. Másodszor az LR-CAN megoldásban különböző költség-korlát függvényeket használhatunk és a hálózati igényeknek megfelelően azt könnyedén kicserélhetjük, például ha két hálózatban különböző szolgáltatási minőséget biztosítanánk a kikeresésekre. Azonban ha a hálózatban kevés keresés történik, felesleges sok kapcsolatot fenntartani, ekkor is kicserélhetjük a költség-korlát függvényt. Erről a döntés vagy a csomópontok egy kritikus tömege alapján hozható meg, vagy egy őscsomópont segítségével, melynek adminisztrációs jogai vannak. Harmadszor pedig, néhány ésszerű módosítással és feltétellel, az LR-CAN képes meghatározni a kapcsolatok optimális számát az egész hálózatra a kikeresési intenzitás mérésével, és dinamikusan változtatni a kapcsolatokat a kikeresési és jelzésforgalom csökkentése érdekében.

Hivatkozások

- [1] A.B. Bondi. Characteristics of Scalability and Their Impact on Performance. In *Proceedings of the 2nd International Workshop on Software and Performance*, pp.195–203, Ottawa, Ontario, Canada, 2000.
- [2] Kovács Balázs. Skálázható módszerek egyenrangú (peer-to-peer) hálózatok kommunikációjának javítására *Ph.D. Disszertáció*, Budapesti Műszaki és Gazdaságtudományi Egyetem, Budapest, Magyarország, 2009.
- [3] C.E. Perkins and P. Bhagwat. Highly Dynamic Destination–Sequenced Distance–Vector Routing (DSDV) for Mobile Computers. In *Proceedings of ACM SIGCOMM*, pp.234–244, London, England, 1994.
- [4] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11–1997, The Institute of Electrical and Electronics Engineers, New York, NY, 1997.
- [5] C.E. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, July, 2003.
- [6] G. Pei, M. Gerla, X. Hong, and C. Chiang. A Wireless Hierarchical Routing Protocol with Group Mobility. In *Proceedings of IEEE Wireless Communications and Networking Conference*, pp.1538–1542, Boston, MA, 1999.
- [7] P.F. Tsuchiya. The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks. In *Proceedings of the ACM SIGCOMM*, pp.35–42, Stanford, CA, 1988.
- [8] B. Chen and R. Morris. L+: Scalable Landmark Routing and Address Lookup for Multi–Hop Wireless Networks. *Technical Report 837*, Massachusetts Institute of Technology Laboratory for Computer Science, 2002.
- [9] S. Du, A. Khan, S. PalChaudhuri, A. Post, A. K. Saha, P. Druschel, D. B. Johnson, and R. Riedi. Self-Organizing Hierarchical Routing for Scalable Ad Hoc Networking *Technical Report TR04-433*, Department of Computer Science, Rice University, Houston, TX, USA, 2004.
- [10] J. Eriksson, M. Faloutsos, and S. KrishnaMurthy. Peernet: Pushing Peer–to–Peer Down the Stack. In *Peer-to-peer Systems II*, Vol.2735, pp.268–277, ISBN.978-3-540-40724-9, Springer-Berlin, Heidelberg, 2003.
- [11] R.D. Samir, C.E. Perkins, E.M. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. In *Proceedings of INFOCOM*, pp.3–12, Tel Aviv, Israel, 2000.
- [12] Uppsala University. Ad Hoc On Demand Distance Vector Routing - For Real World and Simulation. URL: <http://core.it.uu.se/AdHoc/AodvUUImpl>.

- [13] NS-2. URL: <http://www.isi.edu/nsnam/ns/>
- [14] C.E. Perkins. IP Mobility Support for IPv4. RFC 3344, August, 2002.
- [15] D. Johnson, C.E. Perkins, and J. Arkko. IP Mobility Support in IPv6. RFC 3775, June, 2004.
- [16] Máté Miklós. Hierarchikus mobilitás-kezelés skálázható vezeték nélküli hálózati architektúra felhasználásával. MSc diploma dolgozat, Budapest, 2006.
- [17] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proceedings of the ACM SIGCOMM*, pp.161–172, San Diego, CA, 2001.
- [18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Proceedings of the ACM SIGCOMM*, pp.149–160, San Diego, CA, 2001.
- [19] D. Liben-Nowell, H. Balakrishnan, and D. R. Karger. Analysis of the Evolution of P2P Systems. In *Proceedings of the 21st ACM Symp. Principles of Distributed Computing (PODC)*, pp.233–242, Monterey, CA, 2002.
- [20] P. Maymounkov and D. Mazières. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, pp.53–65, Cambridge, MA, 2002.
- [21] V. Darlagiannis, A. Mauthe, N. Liebau, and R. Steinmetz. An Adaptable, Role-Based Simulator for P2P Networks. In *Proceedings of International Conference on Modeling, Simulation, and Visualization Methods*, pp.52–59. Las Vegas, NV, 2004.
- [22] Peerfactsim.kom. [Online]. Available: <http://www.peerfact.org>

Publikációk

[J] Folyóirat publikációk

- [J1] **Balázs Kovács**, Rolland Vida. Long-Range CAN: to Enhance the Performance of Content-Addressable Networks. *Int. Journal on Advances in Networks and Services, IARIA Journals*, accepted, 2009.
- [J2] **Balázs Kovács**, Rolland Vida. Hierarchical, Multi-spanning Architecture for Managed Wireless Networks. *Int. Journal of Wireless and Mobile Computing, Inderscience Publisher*, accepted, 2007.
- [J3] **Kovács Balázs**, Simon Csaba. „Ambient” hálózatok - áttekintés. *Híradástechnika*, Vol. LX., pp. 39–44, Budapest, July 2005.
- [J4] Biczók Gergely, Égi Norbert, Fodor Péter, **Kovács Balázs**, Vida Rolland. Skálázható útválasztás mobil környezetben. *Híradástechnika*, Vol. LIX., pp. 26–31, Budapest, September 2004.
- [J5] **Kovács Balázs**, Vida Rolland. A Zigbee technológia. *Híradástechnika*, Vol. LVIII., pp. 9–12, Budapest, November 2004.
- [J6] Gergely Biczók, Norbert Égi, Péter Fodor, **Balázs Kovács**, and Rolland Vida. Scalable Addressing and Routing in Large-Scale Wireless Networks. *Production Systems and Information Engineering*, Volume 2 (2004), pp. 143–158, HU ISSN 1785–1270, 2004.
- [J7] Gergely Biczók, Kristóf Fodor, and **Balázs Kovács**. Handover Latencies in BCMP Networks. *Communications: Scientific Letters of the University of Zilina*, No.1, pp.34–37, Zilina, Slovakia, 2004.
- [J8] Gergely Biczók, Kristóf Fodor, **Balázs Kovács**, and Ágoston Szabó. Pervasive computing - An Overview. *Híradástechnika (selected papers)*, Vol. LVIII., pp. 2–7, Budapest, June 2003.

[C] Konferencia publikációk

- [C1] **Balázs Kovács**, Rolland Vida. An Adaptive Approach to Enhance the Performance of Content-Addressable Networks. In *Proceedings of the IEEE 3rd International Conference on Networking and Services ICNS 2007*, Vol. 1, pp.93–98, Athens, Greece, June 19–24 2007. Available electronically on: *IEEEExplore*, **Best Paper Award**.
- [C2] **Balázs Kovács**, Rolland Vida. Hierarchical, Multi-spanning Architecture for Managed Wireless Networks. In *Proceedings of the IEEE International Workshop on Pervasive Computing and Ad hoc Communications (PCAC-AINA'06)*, Vol. 2, pp.111–115, Vienna, Austria, April 18–20 2006. Available electronically on: *IEEEExplore*

- [C3] Gergely Biczók, Norbert Égi, Péter Fodor, **Balázs Kovács**, Rolland Vida. Scalable Addressing and Routing Architecture for Future Networks. In *Proceedings of the Polish-Czech-Hungarian Workshop*, Volume 1, pp.19–28, Budapest, September 2004.
- [C4] Róbert Szabó, Péter Kersch, **Balázs Kovács**, Csaba Simon, Márk Erdei, Ambrus Wagner Dynamic Network Composition for Ambient Networks: a Management View. In *Proceedings of Eurescom Summit 2005*, pp.35–42, Heidelberg, Germany, April 27–29 2005.
- [C5] **Balázs Kovács**, Rolland Vida, Gergely Biczók. Hierarchical Architecture for Managed Wireless Networks. In *Proceedings of Student Workshop Conext 2005*, pp.260–261, Toulouse, France, October 24–27 2005.
- [C6] Fodor Kristóf, **Kovács Balázs**. Blown-up rendszer megvalósítása. In *Proceedings of HTE-BME 2003*, Budapest, Hungary, 2003.
- [O] **Egyéb publikációk**
- [O1] Biczók Gergely, Fodor Kristóf, **Kovács Balázs**, Szabó Ágoston. Blown-up rendszer tervezése és megvalósítása. *TDK Dolgozat*, Budapest, Hungary, 2002.