



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Híradástechnikai Tanszék

Adaptív multihop adatterjesztő protokollok
ad hoc hálózatokban
(TDK dolgozat)

Készítette: Kőkuti András

Konzulens: Dr. Simon Vilmos

Tartalomjegyzék

1. Bevezetés	1
2. Információterjesztés ad hoc hálózatban	3
3. Motiváció	9
4. Új adaptív információterjesztő protokollok	10
4.1. Közös tulajdonságok	10
4.2. Distance Based Handshake Gossiping (DBHG): Távolság alapú kézfogásos Gossiping	16
4.3. Valency Based Handshake Gossiping (VBHG): Fokszám alapú kézfogásos Gossiping	17
4.4. Average Valency Based Handshake Gossiping (AVBHG): Átlagos fokszám alapú kézfogásos Gossiping	19
4.5. Összegzés	20
5. Eredmények	21
5.1. Szimulátor	21
5.2. A teljesítmény mutatók	24
5.3. Az eredmények kiértékelése	26
5.3.1. Együzenetes eset	26
5.3.2. Többüzenetes eset	32
6. Konklúzió, jövőbeli tervek	39

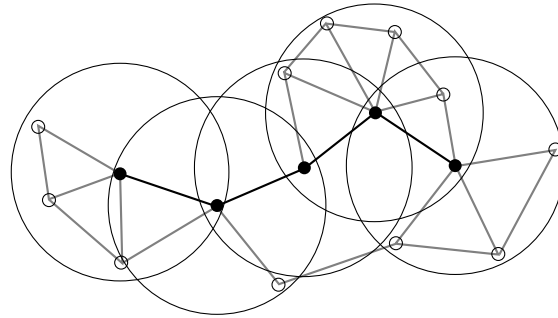
1. Bevezetés

Napjainkban egyre inkább felértékelődnek az infrastruktúra nélküli, azaz az önszerveződő, peer-to-peer kommunikációt lehetővé tévő hálózatok, köztük jelentős szerepet tölt be a mobil ad hoc hálózatok családja. A mobil ad hoc hálózat (az angol nyelvű szakirodalomban használatos rövidítése: MANET[1], azaz Mobile Ad-hoc Network) a rádiós csatornán keresztül kommunikáló, mobil eszközökből álló, önmagát konfigurálni képes hálózat. Az ilyen hálózatoknak számos felhasználási területük létezik, például vészhelyzeti MANET hálózatok, szenzor hálózatok a környezet megfigyeléshez[2], közlekedési ad hoc hálózatok (az angol nyelvű szakirodalomban VANET[3], Vehicular Ad-hoc Networks) és még számos esetben használják őket[4, 5]. Fontos különbség a hagyományos mobil hálózatok és az ad hoc hálózatok között, hogy utóbbinál nincs központi infrastruktúra, azaz nincs a csomópontoknak és hálózati elemeknek egy olyan kitüntetett halmaza, amelyek feladata a többiek kommunikációjának biztosítása, a forgalom szervezése. Ez a különbség a kulcsa az ad hoc hálózatok kívánatos tulajdonságainak és a legnagyobb fejtörést okozó nehézségeinek is. A hálózat mobilitásából adódóan topológiája, résztvevői, elvárt teljesítménye előre nem ismertek.

Pontosan ezen okok miatt rendkívül nehéz feladat a hálózat optimális kihasználása, vagyis, hogy minél kevesebb üzenetszórással mindegyik csomópont-hoz eljuttassuk a kívánt üzeneteket. Ezen hálózati körülmények között általában broadcast címzést (azaz üzenetszórást) használunk, ez egyaránt hordoz magával előnyöket és hátrányokat a többi címzéshez képest. Előnyt jelent, hogy nem igényel külön egyedi címzést, azaz minden egyes csomópont, amely a küldő csomópont rádiós sugarán belül helyezkedik el (feltételezve, hogy semmilyen akadályozó körülmény nem lép fel a közegben) azonos periódusban (nem feltétlenül ugyan abban az időpillanatban, hiszen különböző távolságra helyezkedhetnek el a küldő csomóponttól) kapja meg a kiküldött üzeneteket. Ugyanakkor pont ez jelenti az egyik hátrányát is, hiszen azonos időben túl sok üzenet szétküldése esetén, úgynevezett broadcast viharok (angol szakirodalom broadcast storm[6]) jelenhetnek meg a rádiós közegben. Ezen viharok a küldött üzenetek ütközését eredményezhetik, hozzájárulva a hálózat teljesítményének drasztikus romlásához, hiszen így számos csomópont csak későbbi fázisban kaphatja meg azon üzeneteket, amelyeket korábban már meg kellett volna kapnia. További hátránya, hogy az üzenetszórás hatására, nemcsak azon csomópontok szerzik meg az üzenetet, amelyekhez korábban még nem érkezett meg, hanem azok is, amelyek már rendelkeznek azokkal. Ennek következtében a hálózatban duplikátumok jelennek meg, pazarolva a rendszer erőforrásait, mint például a csomópontok memóriáját vagy a rádiós közeg átbocsátó képességét.

Az előző bekezdésben taglaltak okán, a rendszer optimális kihasználásához azon csomópontok legkisebb halmazát keressük, amelyek lefedik a teljes hálózatot és összeköttetésben vannak egymással. Összeköttetésben kell lenniük, hiszen csak így tud minden kijelölt csomópont a megfelelő üzenettel adást kezdeményezni (hiszen hozzájuk is el kell jutnia a tovább adni kívánt üzeneteknek), és természetesen a minimális számú halmazt keressük, hiszen ezáltal minima-

lizáljuk az üzenetküldések számát, így a duplikációk számát is minimalizáljuk a hálózatban, illetve felesleges erőforrást sem használunk fel. Feltételezve, hogy a mobil csomópontokat megfeleltetjük gráfbeli csomópontoknak, illetve ha a rádiós sugaruk folytán létezik közöttük kapcsolat, akkor élt húzunk közéjük, egy gráfot kapunk, amelyben az előbb taglalt halmazt minimális összefüggő fedésnek (a szakirodalomban Minimum Connected Dominating Set, MCDS) nevezzük.



1. ábra. Példa egy minimális összefüggő fedésre

Az 1.ábrán szemléltetve van egy példa egy minimális összefüggő fedésre az adott gráfban. Ezen fedés eléréséhez viszont a gráf teljes ismeretére szükségünk van, azaz a hálózat globális paramétereire (az egyes csomópontok koordinátái). Ha ez a globális információ a rendelkezésünkre is állna, ami ellentmond az ad hoc hálózatok definíciójával, akkor is a mobilitás miatt ezen információ hamarabb elavulna, minthogy felhasználhatnánk. Éppen ezért nem globálisan próbáljuk megoldani ezt a problémát, hanem az ad hoc hálózatokhoz híven lokális információk alapján. Az alacsony szinten történő lokális interakciók eredményeképpen a rendszer képes elérni egy globális célt. Különböző heurisztikákkal próbálják eldönteni az egyes csomópontok, hogy a hálózat eszközeit hatékonyan lefedő csomópontok halmazának részei vagy sem. Ezen heurisztikák maguk a protokollok, amelyeket az eszköz futtat. Ha a protokoll megfelelő lokális döntéseket hoz, akkor ez a hálózat hatékony működését eredményezi, azaz kevesebb duplikációt létrehozva hatékonyabb erőforrás kihasználtság mellett képes működni. Ellenkezőleg, ha rossz döntésekre vezet, az a hálózat nem megfelelő kihasználtságát, erőforrás pocsékolását eredményezi.

Így a dolgozat további része a megfelelő protokoll megtalálásáról szól, arról, amelyik képes lokális információk alapján, úgy mint például a csomópontok távolsága, illetve fokszáma, bizonyos fokig (hiszen a globális optimum az előzőek alapján nem érhető el) megfelelő döntéseket hozni a hatékonyság érdekében.

2. Információterjesztés ad hoc hálózatban

A bevezetőből már tudjuk, hogy az információ terjesztése a csomópontok által futtatott protokollok rendkívül nehéz feladata. A protokollokat csoportosítani tudjuk a protokollstack-ben elfoglalt helyük alapján, illetve a felhasznált információmennyiség szerint. Mivel a dolgozat későbbi fejezetében bemutatásra kerülő, általam tervezett protokollokat is a felhasznált információmennyiség szerint vizsgálom, így a többi protokollt is az utóbbi szempont alapján fogom nagyító alá venni. Ebben a szakaszban a szakirodalomban már fellelhető MANET protokollokat kívánom ismertetni.

A legegyszerűbb ezen szempont alapján természetesen az, amelyik semmiféle információt nem használ fel a csomagküldés során. Ez a protokoll az úgynevezett blind flood („vak elárasztás”), amelynek működése nagyon egyszerű, amint érkezik egy üzenet, azt azonnal továbbküldi[7]. Gondolhatnánk, hogy ez akár egy jó megoldást is kínál az adatterjesztésre, hiszen rendkívül egyszerű, s az összes csomópontot lefedi (feltételezve, hogy nem lépnek fel szeparációk), ám ez nem így van. Érezhető, hogy, ha minden csomópont megismétli a kapott üzenetet, akkor rengeteg duplikáció keletkezik, azaz rendkívül sok erőforrást emésztene fel (sőt véges erőforrást feltételezve, lehet, hogy pont emiatt lenne a csomópontoknak egy olyan halmaza, amely nem kapta meg az üzenetet), illetve a sok egyszerre adás következtében a bevezetésben ismertetett broadcast viharok jelenhetnek meg, amely akár az egész hálózatot térdre kényszerítheti. Tehát ez a megoldás rendkívül erőforrás pazarló, viszont ugyanennyire egyszerű is. A MANET világában nincs olyan protokoll, amely képes minden környezetben ugyan olyan hatékonyan működni, hiszen kompromisszumot kell kötni az egyszerűség és a hatékonyság között. A hatékonyabb működéshez (pl. kevesebb duplikáció) új jelzésüzenetek szükségesek (természetesen ezen jelzésüzenetek kisebb terheltséget okoznak a hálózatban, mint az adatüzenetek a méretkülönbségből adódóan).

Az előbbi egyszerű protokoll továbbfejlesztésével jött létre az APF (Adaptive Periodic Flood). Az algoritmus működése annyiban tér el az előzőekben ismertetett BF (blind flood) algoritmustól, hogy a csomagok ütközésének, illetve a broadcast viharok elkerülése érdekében[8] egy véletlenszerű időzítővel módosították a továbbküldés mechanizmusát. Azaz a csomópont nem azonnal ismétli meg a kapott üzeneteket, hanem sorsol magának egy véletlenszerű időtartamot (RAD, Random Assessment Delay), s csak abban az esetben küldi tovább, ha letelik az időtartam úgy, hogy közben nem kapott duplikációt. Ha az időzítő lejárt előtt már egy szomszédos csomóponttól megkapja a továbbadni kívánt üzenetet, akkor eldobja azt, feltételezve, hogy a két csomópont szomszédai között van fedés. A protokoll által alkalmazott heurisztika nagyon egyszerű, mégis jelentősen hatékonyabb tud lenni, mint az egyszerű elárasztás.

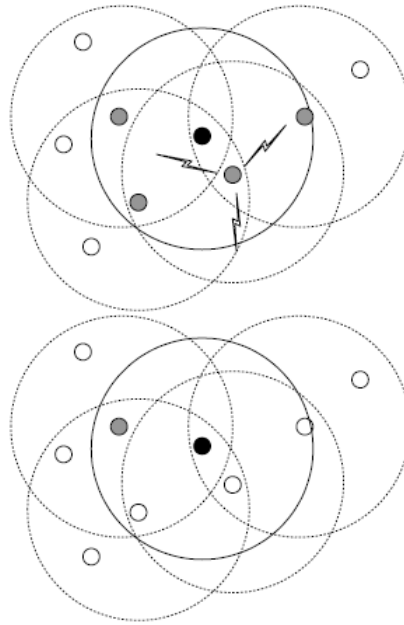
A Gossiping[8] is egy hasonlóan egyszerű algoritmus, mint a BF, ám működése teljesen eltér az előzőekben ismertetett algoritmusoktól. A protokoll előre meghatározott valószínűségeket rendel hozzá a csomópontokhoz, s ezen valószínűségek alapján indítanak új adást a csomópontok. Ez az algoritmus a legegyszerűbben működő valószínűség alapján döntő algoritmus. Egy statikus hálózat-

ban az optimális továbbadási valószínűség kiszámolható off-line[10] is, viszont mobil hálózatok esetében ezen valószínűségek összefüggésben vannak a mobil környezet állandóan változó paramétereivel, mint például a csomópontok fokszáma. Ezen hátrányos tulajdonság kiküszöbölésével jött létre a Hypergossiping [11]. A Gossiping - t ezen kívül még kiegészítették szomszédossági információkkal, illetve partíció detektálással. Ez különösen alkalmas olyan mobil hálózatokban, amelyekben gyakran jönnek létre szeparálódások, illetve ezen partíciók képesek egymáshoz csatlakozni, illetve kettéválni időre időre, azaz megoldást nyújt szeparált hálózatokban is, ellentétben az APF-al és a BF-al. A partíciókra esés detektálásához egy egyszerű heurisztikát használ: az egy partíción belüli csomópontok ugyanazokat az üzeneteket kapják meg. Így minden egyes csomópont listát tárol az általa birtokolt üzenetekről, s csatlakozáskor ezen listák kicserélődnek. Ha a listák metszete egy bizonyos határ alatt van, akkor az egyes üzenetek küldésre kerülnek. A partíciókon belül pedig az eszközök adaptív valószínűségeket kapnak, azaz a valószínűségek a környezet függvényében változnak (példa egy egyszerű heurisztikára: minél magasabb a csomópont fokszáma, annál nagyobb üzenettovábbküldési valószínűség tartozzon hozzá).

A Distance Adaptive Dissemination[12] protokoll, ahogy a nevéből is kitűnhet, a csomópontok távolságának ismeretét használja fel a hatékonyabb üzenet szórás érdekében. Alapvető célja, hogy a legtávolabbi szomszédoknak juttassa csak el az üzenetet, hiszen ezen csomópontok képesek az adó által ismeretlen területeket legnagyobb arányban lefedni. Ennek megvalósításához minden eszköz tárol egy úgy nevezett 1-hop (csak azok, amelyek egy „ugrásból” elérhetőek, azaz a szomszédok) szomszédossági rekordot. Ebbe a rekordba minden szomszédos csomópontra feljegyzi a távolságot a vett jelerősség (szakirodalomban RSS:Received Signal Strength) alapján, természetesen szükséges feltétele a protokoll működésének, hogy minden eszköz rendelkezzen jelerősség mérővel, ám ez a mai eszközöket feltételezve gyakorlatilag nem is szűkíti le az eszközök listáját (gondoljunk bele, ilyenrel a mai laptop-ok, okostelefonok mind-mind rendelkeznek). A protokoll szerzői két variánsát hozták létre, a DAD-NUM és a DAD-PER protokollokat. A DAD-NUM egy jelerősség határt ($S_{threshold}$) definiál a k darab legtávolabbi csomópont jelerőssége alapján (azaz a k darab legkisebb jelerősségből veszi a legnagyobbat). Egy üzenet érkezésekor a csomópont leellenőrzi, hogy a jelerősség kisebb-e, mint a $S_{threshold}$, ha igen akkor tovább küldi a csomagot, ellenkező esetben eldobásra kerül. Vegyük észre, hogy a jelerősség annál kisebb minél távolabb helyezkedik el a csomópont, de ez az arányosság természetesen nem lineáris, tehát, ha kisebb jelerősséget keresünk, az távolabbi csomópontnak felel meg, ezért kerülnek tovább küldésre az adott üzenetek. A DAD-PER működése rendkívül hasonló, különbség csak a jelerősség határ ($S_{threshold}$) megválasztásából adódik. Míg a DAD-NUM a csomópontok egy k elemű halmazát, addig a DAD-PER egy p százalékát választja a határ megválasztásához. A bekezdésben tárgyalt k és p paraméterek a teljesítmény növelése érdekében tetszőlegesen hangolhatóak.

A mobil ad hoc hálózatokban használt algoritmusok következő csoportja az úgy nevezett önmetsző algoritmusok, amelyek már több információt használnak fel a környezetből, s döntéseiket szomszédossági információkra (a hálózat

topológiájára) alapozzák. A csoport első általam bemutatott tagja, az SBA[13] (Scaleable Broadcast Algorithm). A működéséhez 2-hop listát tárol, azaz azon csomópontokat, amelyek két ugrásból érhetőek el (a szomszédok szomszédjai), illetve számon tartja az utolsó megszerzett üzenet küldőjének egy egyedi címét (például a valós életbe erre illeszkedik egy eszköz MAC címe). Amint egy csomópont - legyen ez v - kap egy üzenetet, ennek küldője legyen u , meghatározza az érdekelt csomópontokat (B) az $N(v) - N(u)$ halmazok különbségével, ahol $N(v)$ a v csomópont szomszédainak halmaza. Tehát azok a szomszédos csomópontok fognak kiesni a halmazból, amelyeket mindkettő csomópont lefed, azaz amelyekhez feltételezhetően (ha csak valami ütközés nem következett be) eljutottak már az u által küldött üzenetek. Ha a B halmaz számossága nagyobb mint nulla ($|B| > 0$), akkor az APF-nél már ismertetett véletlenszerűen kisorsolt időtartamot (RAD) várakozik, míg újra nem adja az üzenetet (persze, ha a halmaz üres, azaz számossága nulla, akkor csak duplikációt növelve, erőforrást pazarolna az újraküldés, ezért ebben az esetben nem hajtódik végre). Az intervallum, amelyből az időtartamot sorsolja, maximumát a $\left(\frac{d_v}{d_{max}}\right) * T_{max}$ formulával számolja. A képletben szereplő $d_v = |N(v)|$, azaz a v csomópont szomszédainak száma (gráf elméletben a v fokszáma), d_{max} pedig a v szomszédai közül a legtöbb szomszédos csomóponttal rendelkező csomópont fokszáma (ennek megállapításához szükség van a szomszédos csomópontok szomszédjaira is, de pontosan emiatt került kialakításra a 2-hop lista). A T_{max} paraméter segítségével megmarad az intervallum nagyságának állítási lehetősége, így hangolva ezt a hálózat eltérő körülményeihez. Lényegében a heurisztika alapja az, hogy a magasabb fokszámú csomópontok egy lépésben több csomópontnak képesek elküldeni az üzenetet, így a duplikáció csökkentése érdekében adjanak hamarabb, mint a kisebb számú szomszédal rendelkezők. Látható, hogy a protokoll teljesítménye az időzítőben rejlik. Várakozás közben a csomópont figyeli a környezetet, s ha üzenet küldést észlel, akkor a küldő csomópont szomszédjait kiveszi a saját 1-hop listájából. Ez a folyamat látható a 2. ábrán.



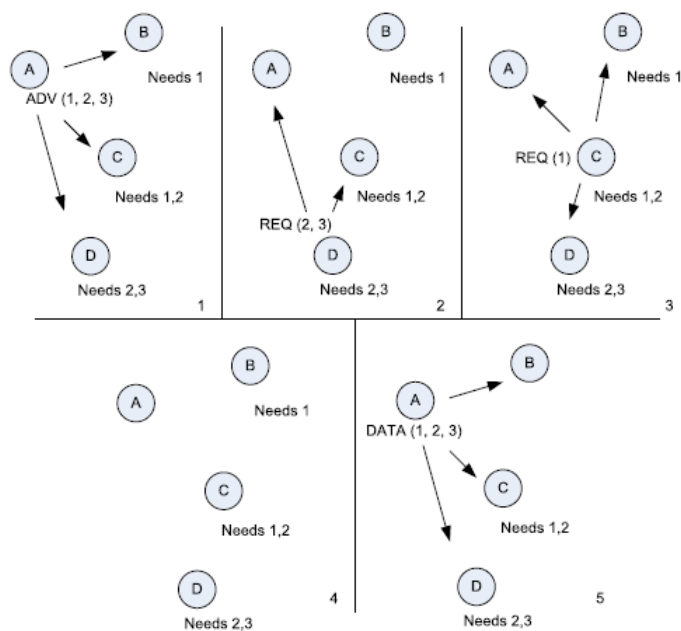
2. ábra. [7]Az SBA működése

A felső ábrán a fekete csomópont éppen az időzítő lejártát várja, miközben észleli, hogy az egyik szomszédja adást kezdeményezett. Ennek hatására az alsó ábrán a fekete csomópont frissíti a listáját, s amelyek már megkapták az üzenetet (az észlelt küldő szomszédjai) szürke színről fehérre váltanak át. Ha az időzítő lejárta előtt minden csomópont kikerült a listából (az ábrára gondolva, ha minden csomópont fehérre színeződik át), akkor az adás megszakad, kikerülve a felesleges erőforráshasználatot és duplikációkat.

Az SBA algoritmus módosításával, kiterjesztésével jött létre a Multi-Message Scalable Broadcast Algorithm (MMMSBA) nevű protokoll. Ez az algoritmus már több üzenetet képes kezelni egy időben. Az SBA-val ellentétben nem csak akkor indít RAD-ot, ha üzenetet vesz, hanem a szomszédinformáció változásakor is. Így, ha egy új, eddig ismeretlen csomópont jelenik meg, elindítja a RAD-ot, amelynek működése azonos az SBA részletezése során megismertekkel. Lehetőség, hogy az algoritmus által újnak nevezett csomópont már minden üzenetet birtokol (ez mobil környezet esetében igen egyszerű, például most került újra a rádiós sugáron belül), így az üzenetek újbóli küldése jelentős plusz terhelést okozna. A probléma megoldására a csomópontok periodikusan közlik szomszédjaikkal mely üzenetekkel rendelkeznek (a periodikusan elküldött HELLO üzeneteket kiegészítik a birtokolt üzenetek listájával). Így lehetőség van csak azokat az üzeneteket elküldeni, amelyekkel az újonnan csatlakozó eszköz még nem rendelkezik.

Egy egészen más megközelítés alapján próbál hatékony megoldást nyújtani az IOBIO algoritmus. A protokoll három fázisú kézfogást használ, hogy

felderítse azon csomópontokat, amelyek érdekeltek valamelyik üzenet iránt, az-
 zal a céllal, hogy csökkentse a szükségtelen terhelést a hálózatban a duplikációk
 elkerülésével. A három fázisú kézfogás megvalósításához a protokoll három tí-
 pusú üzenetet definiál. Az ADV (Advertisement) üzenet periodikusan küldésre
 kerül, tartalmazva a csomópont által birtokolt üzeneteket, amelyeket adott eset-
 ben tovább képes adni. A szomszédos csomópontok ez alapján kideríthetik, hogy
 nekik ezen üzenetekből melyekre van szükségük, azaz az alapvető feltételezése
 a protokollnak, hogy nem minden üzenet érdekel minden csomópontot. A kéz-
 fogás második fázisa, hogy a szomszédos csomópont visszaküldi, hogy neki mely
 üzenetek szükségesek, ezt REQ (Request) üzenet formájában hajtja végre. Ezek
 után az utolsó fázis az adat küldése DATA üzenetben. Az egyes csomópontok
 nem azonnal küldik ki a REQ üzenetet, amint kapnak egy ADV üzenetet, hanem
 itt is a korábbiakban már látott véletlen időtartam után. A késleltetés során a
 csomópontok folyamatosan figyelik a környezetet, s csak azokat az üzeneteket
 fogják kérni, amelyeket korábban még semelyik csomópont sem. Ez a folyamat
 kerül demonstrálásra a 3. ábrán.



3. ábra. [7]IOBIO működése

Az első lépésben (amelynek az 1-s jelzésű ábra felel meg) az A csomópont
 meghirdeti (ADV) az általa birtokolt 1-es, 2-es és 3-as számmal jelölt üzeneteket.
 A szomszédos csomópontok az ADV üzenet fogadásakor elindítják a véletlensz-
 erűen sorsolt időzítőt. A második lépésben (2-s jelzésű) D REQ üzenetet küld
 , jelezvén benne, hogy számára a 2-es és a 3-as számú üzenetek az érdekesek.
 A harmadik lépésben C is küld egy REQ üzenetet, viszont mivel hallotta, hogy

D-nek is kell a 2-s számú üzenet, így ő már csak az 1-s jelzésű üzenet iránti igényét jelezheti. A következő lépésben B időzítője is lejár, azonban az 1-es számú üzenet (amelyre neki szüksége volt) már valamelyik csomópontához tartozik, így B nem küld ki REQ típusú üzenetet. Az utolsó, ötödik lépésben pedig megtörténik az adatüzenetek szétküldése.

Ezennel végére értem az általam bemutatásra várt protokollokon, természetesen még számos protokoll létezik a szakirodalomban, amelyek a mobil ad hoc hálózatokra lettek kitalálva[9]. Ezek részletezése még jó pár oldalon át folytatódhatna, ám ezen dolgozatnak célja nem az irodalomban fellelhető összes protokoll ismertetése, hanem, egy új bevezetése. Új protokoll definiálásához persze elengedhetetlenek a legfőbb irányelvek, algoritmusok ismerete, ezért ennek a fejezetnek a tartalma is elengedhetetlen a továbblépéshez.

3. Motiváció

A 2. fejezetben láthattuk, hogy a mobil ad hoc hálózatokban már rendkívül sok protokoll került implementálásra, de egy sem képes minden környezetben egyaránt hatékonyan teljesíteni. Vannak olyan protokollok, mint például a már ismertetett Gossiping, vagy az APF[8], melyek a szétkapcsolódásra, és vannak olyanok is, mint a Hypergossiping[11], melyek a csomópontok sebességére érzékenyek. Bármely protokoll számára létezik olyan környezet, amelyben nem képes hatékonyan üzenetet továbbítani. Ezen alapelv természetesen az általam tervezett és implementált protokollokra is teljesül. Hiszen ahhoz, hogy kevesebb duplikációval kommunikáljanak a protokollok, bizonyos overhead-re van szükség, amelyek lehetnek jelzésüzenetek (HELLO, ADV, REQ), lehetnek pozíciós adatok (GPS koordináták), távolság mérések (RSS). Természetesen ezek többlet terhelést okoznak az adatüzeneteken felül, de lehet, hogy cserébe nagyságrendekkel kevesebb adatüzenet kerül forgalmazásra, amely összességében erőforrás takarékosabb működést eredményez. A kérdés ami felmerül, hogy mennyi többlet üzenet legyen a hálózatban, hol van az egészséges egyensúly, van-e ilyen egyáltalán? Sajnos nincs, illetve általánosságban nincsen, persze egy-egy kiragadott esetben megtalálható. Hasonlítsuk össze a bemutatott legegyszerűbb BF és legkomplexebb IOBIO protokollokat. Triviálisnak tűnik az a kijelentés, hogy az IOBIO „jobb”, de gondoljunk bele abba a helyzetbe, hogy kevés üzenetünk van, kis hálózatban. Ebben az esetben az IOBIO által forgalmazott ADV, REQ, és HELLO jelzésüzenetek akár nagyobb terhelést is eredményezhetnek, mint a feleslegesen elküldött adatüzenetek, és akkor még az eszközök háttér teljesítményét nem is vettük figyelembe (hiszen IOBIO esetén szükséges listát tárolni a memóriában). Ezen speciális esettől eltekintve természetesen kijelenthető, hogy az IOBIO protokoll átlagosan sokkal jobb teljesítményt képes nyújtani adott körülmények között.

Szükség lenne egy olyan protokollra, amely képes hasonló körülmények között nagyságrendekkel kevesebb duplikáció mellett helyesen kommunikálni és amely kevesebb üzenet szórással is képes teljes lefedettségre szert tenni. Ezen kihívásokra válaszul terveztem meg és implementáltam protokolljaimat, szám szerint hármat. Ezek ugyanazon az alap heurisztikán alapulnak, de, hasonlóan a DAD-NUM és a DAD-PER közötti különbségekhez, itt is vannak fontos különbségek az implementálásban. Mivel önszerveződő hálózati környezetben kell jól teljesíteniük, ezért az eszközökben futtatott protokollok lokálisan rendelkezésre álló információk alapján működnek, hasonlóan a fentiekben ismertettekkel.

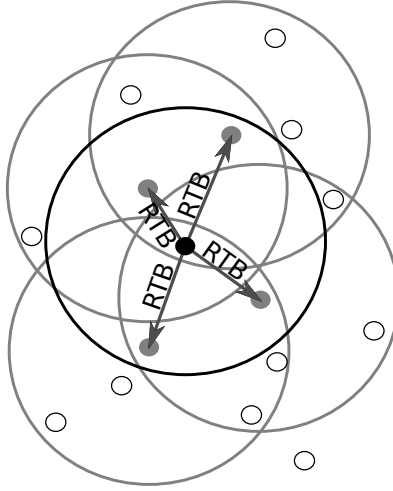
4. Új adaptív információterjesztő protokollok

4.1. Közös tulajdonságok

Az előző fejezetben már szó esett arról, hogy három protokollom kerül bemutatásra. Ezen alfejezet, ahogyan a címe is mutatja, a három protokoll közös tulajdonságait kívánja ismertetni. Hiszen ezek a protokollok mind ugyanazokból a fázisokból állnak, egyedül a valószínűségek meghatározásában különböznek, de ez majd a dolgozat későbbi anyagát képezi.

Mindegyik protokoll a mobil ad hoc környezetre lett tervezve, azaz feltételezi, hogy nincs központi infrastruktúra, amely a kommunikáció irányításáért felelne, hanem az eszközöknek saját maguknak kell ezt megoldani. Nincsen globális rálátásunk a hálózatra, hanem az eszközökben rejlő lokális információkra támaszkodva kell megfelelő döntéseket hoznunk, amelyek ha helyesek, akkor azok a teljes hálózat hatékonyságát eredményezhetik. A bevezetésben szó esett arról, hogy optimális esetben csak azon csomópontok továbbítanak üzeneteket, amelyek MCDS -t alkotnak. Erre a célra kerültek kialakításra az önmetsző (szakirodalomban self-pruning) algoritmusok, amelyek közé a második szakaszban ismertetett SBA algoritmus[13], illetve számos egyéb protokoll[15] is tartozik. Ezek jelzésüzenetek forgalmazásával érik el a teljesítmény javulást. Az algoritmusok egy másik csoportja az eszközökhöz rendelt üzenettovábbküldési valószínűségek segítségével kívánnak hatékonyabbak lenni. Ide tartozik a Gossiping, Hypergossiping[11], illetve sok egyéb protokoll[16] is. Az általam megtervezett protokollok ezen két csoport előnyeit kívánják egyesíteni. Tehát működésük során egyaránt történik jelzésüzenet alapú kommunikáció, illetve adaptív üzenettovábbküldési valószínűség meghatározás. A protokollok a valószínűségek számítási módjaiban térnek el, így ezekről csak későbbiekben, a konkrét protokollok ismertetésénél esik szó. Feltételezzük, hogy az eszközök rendelkeznek GPS adó-vevővel, mely feltételezés a mai okostelefonok, PDA-k és laptopok világában gyakorlatilag alapkövetelmény.

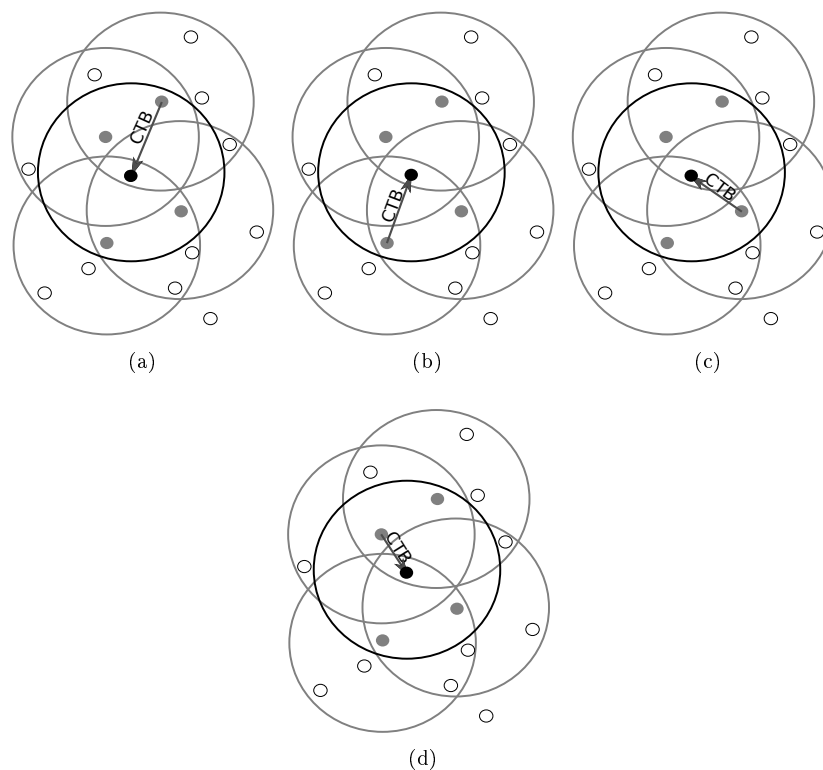
Mind három protokollom működése ugyanazokra a fázisokra bontható. Az első fázis az úgynevezett RTB[18] (Ready-To-Broadcast), amely a nevéből is adódóan, hasonlít a CSMA/CA RTS (Ready-To-Send) fázisára. Az az eszköz, amely rendelkezik a továbbadni kívánt üzenettel, RTB típusú jelzésüzeneteket szór szét a rádiós sugarán belül. Az üzenetbe a küldő elhelyezi a saját egyedi azonosítóját (például MAC címét), illetve többüzenetes kommunikáció esetén az egyes üzenetek azonosítóját, illetve típusát is.



4. ábra. Az első fázis

A 4. ábrán fekete ponttal az éppen adni szándékozó eszköz, szürkével a hatósugarán belül, fehérrel a kívül eső csomópontok vannak jelölve. A fekete kör a fekete pont, a szürke a szürke pontok rádiós sugarait kívánja szemléltetni. A feketével megjelölt eszköz adást kíván indítani, így belép az első fázisba és RTB üzeneteket szór szét. A szomszédos szürke színű csomópontok az üzenetszórás hatására megkapják az üzenetet. Mivel broadcast címezést használunk, ezért soha nem közvetlen a szomszédos csomópontoknak (amelyeket a mobilitásból adódóan explicit nem is ismerjük) küldjük (ez lenne a unicast), hanem a rádiós sugáron belül mindenkinek.

A következő fázisban minden RTB csomagot kapott csomópont indít egy véletlen hosszú időzítőt (RAD) a $[0; t_{CTB}]$ időtartományból sorsolva. Az időzítő felel azért, hogy a visszaküldött CTB (Clear-To-Broadcast) üzenetek ne okozzanak broadcast viharokat, illetve, hogy elkerüljük az ütközéseket. A CTB csomag tartalmazza a küldő és a „címezett” egyedi azonosítóját. Itt is üzenetszórás történik, de így az RTB-t küldött eszköz tudni fogja (hiszen összeegyezteti az üzenetben lévő és a saját azonosítóját), hogy az számára érkezett, vagy esetleg más (tőle 2-hop távolságra lévő) csomópont is kezdeményezett adást. A csomag ezen kívül tartalmazza még a küldő koordinátáit (amelyet a GPS adó vevő segítségével állapít meg) és a szomszédjainak a számát, azaz a fokszámát. A szomszédok felderítéséhez periodikusan küldött HELLO és HELLO-REPLY üzeneteket alkalmaznak a csomópontok. Minden eszköz megfelelő időközönként HELLO üzenetet küld, és figyel a környezetét. Ha HELLO-REPLY érkezett, akkor növeli a szomszédjai számát, ha HELLO, akkor HELLO-REPLY csomagot forgalmaz. A CTB üzenet ezeken kívül még tartalmazza azt is, hogy az RTB-ben helyett foglaló üzenetek közül (amelyeket azonosítójukkal, típusukkal jelöltünk meg) melyekre van szüksége a csomópontnak. A második fázis szemléltetésére szolgál az 5. ábra.



5. ábra. A második fázis

Először az 5.a ábrán látható csomópontnak jár le a véletlenszerűen sorsolt időzítője, így ő jut leghamarabb az adás lehetőségéhez. CTB jelzésüzenet áraszt szét a rádiós hatósugarán belül jelezvén benne, hogy hét szomszédja van, hiszen hét csomópont, négy fehér, két szürke és egy fekete, tartózkodik a szürke színnel ábrázolt rádiós sugarán belül. Majd a b, c, és d ábrán látható eszközök kerülnek sorra, az időzítők sorrendjében. A fekete színnel jelölt, korábban adást indító csomópont eközben várakozik. A csomópont nem azt vizsgálja, hogy beérkezett-e már az összes szomszédos eszköz által küldött válasz CTB üzenet, hiszen a periodikusan küldött topológiát felderítő HELLO üzenetek nem feltétlenül pontosak, mobilitásból adódóan elavulhatnak. Ehelyett egy t_{DATA} ideig várakozik, amely időtartam megadható a t_{CTB} és a maximális terjedési idő összegeként. Az RTB és a CTB jelzésüzenetek felépítését az 1. és a 2.táblázat prezentálja.

bit offset	0-47
0	Küldő egyedi azonosítója (például MAC címe)
48	A birtokolt üzenetek típusai, azonosítói

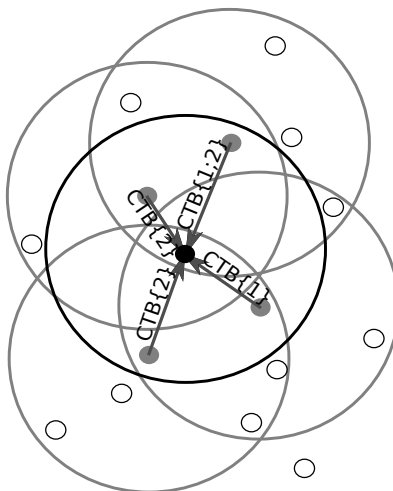
1. táblázat. Az RTB csomag felépítése

bit offset	0-40	41-47
0	Küldő egyedi azonosítója	
48	„Címzett” egyedi azonosítója	
96	Az átadásra kért üzenetek típusai, azonosítói	
144	A küldő csomópont koordinátái	fokszáma

2. táblázat. A CTB csomag struktúrája

Látható, hogy az RTB és CTB csomagok méretükben jelentősen elmaradnak az adatüzenetektől, így minimálisra csökkentve a jelzésüzenetek forgalmazásával járó terhelést.

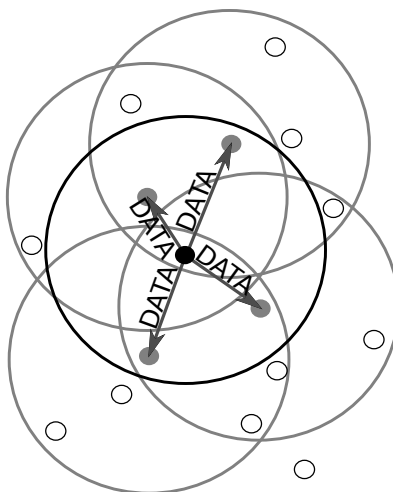
A végső fázis felel az információ továbbadásáért. Az adást indító csomópont (amely az RTB üzeneteket küldte) a t_{DATA} időzítő után megkezdte az adatüzenetek terjesztését. Először összegzi egy 1-hop listában, hogy mely azonosítójú csomópontok tartózkodnak pillanatnyilag a rádiós sugarán belül, illetve ezekhez a fokszámaikat és koordinátáikat is hozzárendeli, amelyek megtalálhatóak a CTB üzenet utolsó 48 bitjében. Ezek után minden egyes szomszédos eszköz számára kiszámít egy üzenettovábbadási valószínűséget. A kiszámítás módja különbözik a három protokoll esetén, így ennek ismertetésére későbbi alfejezetekben kerül sor. A válaszként érkezett CTB csomagokból a központi csomópont megállapítja, hogy mely általa birtokolt üzenetek kerüljenek továbbadásra. Ezt egyszerűen a kért üzenetek uniójaként számolja.



6. ábra. A továbbküldendő üzenetek meghatározása

A fenti ábrán látható a továbbküldendő üzenetek meghatározási módjára egy szemléletes példa. A szürkével jelölt szomszédos csomópontok a CTB válaszüzenetben megjelölik, hogy mely adatüzenetekre tartanak igényt. Van olyan eszköz, amelynek csak az 1-es, és van olyan is, amelynek az 1-s és a 2-s csomagra is szük-

sége van. Ezek alapján a központi, feketével ábrázolt, csomópont meghatározza, hogy neki az 1-es és a 2-es számmal megjelölt üzeneteket kell továbbküldenie (hiszen a visszaküldött halmazok uniója ezen számpár lesz). Tehát minden paraméter ismert (üzenetek, valószínűségek) ahhoz, hogy befejeződjön az információ terjesztése.



7. ábra. A harmadik fázis

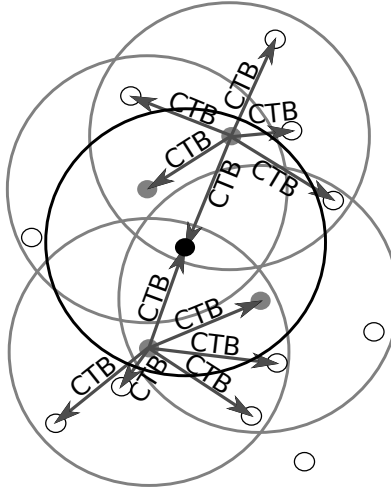
A 6.ábrán látható, hogy a feketével jelölt központi csomópont szétszórja az adatüzenetet a rádiós tartományában. Az adatüzenet (DATA) tartalmazza a korábban kiszámított továbbadási valószínűségeket, illetve hasznos teherként a kért üzenetek unióját. Egy DATA csomag felépítése látható a 3.táblázatban.

bitt offset	0-47
0	A küldő egyedi azonosítója
$48 - f_{\text{küldő}} * 96$	A szomszédok azonosítója, továbbadási valószínűségeik
$f_{\text{küldő}} * 96 + 48 -$	ADAT

3. táblázat. DATA felépítése

A küldő azonosítójára szükség van, hiszen előfordulhat, hogy egymástól 2-hop távolságra lévő csomópontok azonos időpillanatban kezdik meg az átadás fázisait, így a közös szomszédos csomópont ismeri, hogy mely DATA üzenet mely csomóponttól érkezett. A fejrész további részei alapján minden csomópont megtalálja a saját azonosítója mellett a továbbadási valószínűségét. Ezen valószínűségek alapján fognak az 5.ábrán szürkével jelölt csomópontok adást kezdeményezni. Például, ha az egyik csomópontnak 50% a továbbadási valószínűsége, az azt jelenti, hogy 50% eséllyel fog információterjesztést kezdeményezni.

Tételezzük fel, hogy az 5.ábrán szereplő szürkével jelölt eszközök közül kettő továbbadást sorsolt magának (a valószínűségek alapján), kettő pedig nem. Ezt az eshetőséget mutatja be a 8.ábra.



8. ábra. Szomszédos csomópontok információterjesztése

A továbbadás ismét három fázisból áll, így a két szürke csomópont CTB üzenetek küldésével kezdi meg az információ terjesztését. Természetesen a csomópontok helyzete a mobilitásból adódóan folyamatosan változik, így az ábra egy kicsit hamis képet mutat, ám így talán szemléletesebb.

Megismerkedtünk a protokollok alapvető mechanizmusaival, üzeneteivel, amelyek mind három protokoll esetén azonosak. A következő alfejezetek kívánják bemutatni az egymástól való eltérésüket. Az eltérés a valószínűségek kiszámításában rejlik, hiszen mind a három protokoll más és más képlettel, változókkal próbálja megfelelően adaptívvá és hatékonyvá tenni a valószínűségi változót.

4.2. Distance Based Handshake Gossiping (DBHG): Távol-ság alapú kézfogásos Gossiping

A protokollok bemutatása időrendi sorrendben fog történni, azaz először az elsőnek, utolsónak pedig az utolsóként megtervezett protokoll kerül ismertetésre.

A Distance Based Handshake Gossiping (DBHG) protokoll működéséhez nincs szükség a csomópontok fokszámára, így a CTB üzenet tartalmából az eszközök fokszáma kihagyható, a CTB üzenet mérete ebből következően csökken. Sőt az eszközök által periodikusan küldött topológia felderítő üzenetek (HELLO, HELLO-REPLY) is feleslegessé válnak, tehát ezek is elhagyhatóak. A valószínűségek kiszámításához csak a GPS koordinátákra van szükség, amelyeket az előző szakasz alapján a CTB csomag foglalja magában. A küldést indító csomópont a harmadik fázisában az 1-hop listájába beregisztrálja a szomszédos eszközöket, illetve valószínűségeiket, amelyeket majd tovább ad adatüzenet formájában. A valószínűségek kiszámításához a csomópontok távolságát veszi alapul, amely meghatározása az alábbi formula alapján történik.

$$d_i = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2} = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2}$$

A képletben szereplő x_0, y_0, z_0 változók az adást indító, míg az x_i, y_i, z_i az i . szomszédos csomópont koordinátái. A központi csomópont a távolságok közül kiválasztja a legtávolabbit, azaz a legnagyobb értékűt, ez lesz a d_{max} .

$$d_{max} = \max_{\forall i} \{d_i\}$$

A protokoll ezen paraméterek alapján az i . szomszédos csomóponthoz az alábbi valószínűséget rendel:

$$p_i = \frac{d_i}{d_{max}}$$

Azaz minél távolabb helyezkedik el az adótól, annál nagyobb lesz a továbbadási valószínűsége. Ennek oka, hogy így a közel lévő csomópontok okozta duplikációkat el lehet kerülni, illetve a távolabbi csomópontok által lefedett területek magasabb eséllyel tartalmaznak új csomópontokat[17] (amelyek nem a központi hatósugarán belül helyezkedtek el). Továbbá megállapítható, hogy annak a csomópontnak, amelyik éppen d_{max} távolságra helyezkedik el az adótól, a hozzárendelt valószínűségi változója 1 lesz, azaz 100% - os valószínűséggel fog adást kezdeményezni. Tehát a szomszédos csomópontok halmazából biztos lesz olyan (a legtávolabbi), amely továbbviszi az információ terjesztését. Ennek eredményeként az információterjesztés nem fog megállni (kivétel, amikor a kért üzenetek uniója üres, illetve ha nincs más csomópont a hatósugarán belül), így hamarabb ér el magasabb lefedettséget. A gyors terjedéssel járó duplikálódások viszont nem jelennek meg a rendszerben, hiszen a közelebb elhelyezkedő csomópontok továbbküldési valószínűsége alacsony, illetve a küldés mechanizmusával járó jelzésüzenetek segítségével csak a szükséges üzenetek kerülnek továbbadásra.

4.3. Valency Based Handshake Gossiping (VBHG): Fokszám alapú kézfogásos Gossiping

Az előző protokoll hiányossága, hogy nem veszi figyelembe a csomópontok fokszámát. Adott esetben előfordulhat olyan, hogy a legtávolabbi szomszédos eszköznek nincs az adón kívül más szomszédja. A helyzetéből adódóan a 100% - os továbbküldési valószínűség ennek ellenére őt illeti meg. Illetve a protokoll még mindig nem eléggé „bünteti” a közel lévő szomszédos csomópontokat. Ideális esetben távolabb elhelyezkedő, magas fokszámmal rendelkező csomópontoknak magas, míg a közelebb lévő magas fokszámmal rendelkező csomópontoknak alacsony valószínűségi változó járna. A protokoll ezen verziója ezt a célt próbálja elérni. Ehhez természetesen szükség van a topológiát felderítő üzenetekre, illetve hogy ezen információ belekerüljön a CTB csomagba, ahogyan azt az első alfejezetben láthattuk.

A protokoll működéséhez további paraméterekre van szükség. Jelölje f_i az i . szomszédos csomópont fokszámát. A szomszédok megkülönböztetését a CTB - ben visszaküldött egyedi azonosító teszi lehetővé, azaz az f_i ahhoz a csomóponthoz tartozik, amely i . - ként küldött CTB üzenetet. A d_{max} változó az előző protokoll alapján ismert, a legtávolabbi elhelyezkedő szomszédos csomópont távolsága (GPS koordináták alapján számolva). A $d_{\text{átlag}}$ változó a távolságok átlagát jelöli, kiszámítása a számtani közép alapján történik.

$$d_{\text{átlag}} = \frac{1}{n} * \sum_{i=1}^n d_i$$

A képletben szereplő n a szomszédos csomópontok száma, d_i az i . szomszéd távolsága. Ezen kívül még szükségünk van a szomszédos eszközök legnagyobb fokszámára, f_{max} - ra. Ez hasonlóan számítható, mint a d_{max} .

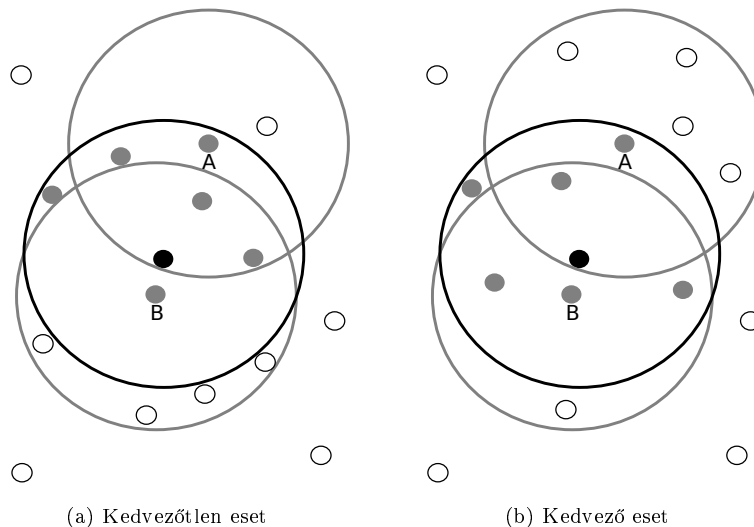
$$f_{max} = \max_{\forall i} \{f_i\}$$

Tehát a képlet egyszerűen a legmagasabb szomszédos fokszámot számolja ki. A szükséges változók ismertetése után most már következhet a valószínűség kiszámítási módja.

$$p_i = \begin{cases} \frac{d_i}{d_{max}} - \frac{d_i}{d_{max}} * \frac{f_i}{f_{max}}, & \text{ha } d_i < d_{\text{átlag}} \\ \frac{d_i}{d_{max}} + \frac{d_{max}-d_i}{d_{max}} * \frac{f_i}{f_{max}}, & \text{ha } d_i \geq d_{\text{átlag}} \end{cases}$$

Látható, hogy a hozzárendelés két diszjunkt esetre vált szét. Az egyik eset, amikor a csomópont a meghatározott átlag távolságon ($d_{\text{átlag}}$) belül helyezkedik el. Ebben esetben az a cél, hogy még drasztikusabban büntessük azokat, amelyeknek magas a fokszámuk, hiszen ezen csomópontok közül valószínűleg számos az eredeti sugáron belül helyezkedik el (hiszen a két lefedett területnek magas az átlapolódása), így a szétküldött információ nagy része felesleges duplikációként jelenne meg a rendszerben. A másik ágnak pont ellentétes a célja. A középponttól távol lévő csomópontok által lefedett terület magasabb valószínűséggel

tartalmaz új (amelyeket pillanatnyilag nem az adó hatótávolságán belül vannak) eszközöket. Tehát ezen eszközökhöz az információ eljuttatása szükséges, ezáltal a rendszer teljesítménye növekszik. Persze előfordulhatnak olyan topológia elrendezések, amelyekben a fenti valószínűségi hozzárendelés rosszabb megoldást nyújt, mint az első verzió.



9. ábra. Speciális topológiai példák

A 8. ábra példát nyújt egy kedvező és egy kedvezőtlen esetre. Az a, ábrán látható, hogy az A jelzésű csomópont az átlag távolságon kívül van, így a magas fokszáma miatt magas továbbadási valószínűség tartozik. Ám A-nak csupán egyetlen olyan szomszédja van, amelyet a feketével jelölt adást indító csomópont nem fed le. Így nagy valószínűséggel, hiszen magas valószínűségi változó tartozik hozzá, az általa terjesztett információnak magas hányada (pontosan négy üzenet az ötből) duplikációkat eredményez. Míg a B-vel jelölt eszközhez alacsony érték tartozik, hiszen az átlag távolságon belül helyezkedik el, illetve a fokszámai miatt még büntetést is szenved el. Pedig a B csomópont rádiós sugarán belül négy új eszköz is szerepel. A b, ábra ezzel szemben egy kedvező esetet kíván szemléltetni, hiszen a magas valószínűséggel bíró A csomópont hatósugarán belül több új és kevesebb korábban már lefedett eszköz helyezkedik el, illetve a B jelzésű csomópont szomszédai között az arány ezzel pont ellentétes. Persze az ábrák alapján is látható, hogy a két eset előfordulására nem ugyanakkora az esély. Hiszen a távolabb elhelyezkedő csomópontok átlapolódása kevesebb, így az itt elhelyezkedő csomópontok átlagos száma is kisebb. A közelebb elhelyezkedők között pedig magasabb metszet létezik, így a nagyobb területen magasabb valószínűséggel fordulnak elő közös szomszédos csomópontok.

4.4. Average Valency Based Handshake Gossiping (AVBHG): Átlagos fokszám alapú kézfogásos Gossiping

Végül elérkeztünk a harmadik, azaz a legutolsó változat ismertetéséhez. A protokoll ezen változatának a célja megegyezik a második változatéval, azaz a távol elhelyezkedő csomópontokat jutalmazzuk, a közeliakat büntessük a fokszámok arányában.

A valószínűség kiszámításához az előzőekben ismertetett változókon kívül szükségünk van még egyre. Ezen változó az $f_{\text{átlag}}$, mely az átlag fokszámot jelöli. Kiszámítási módja azonos, mint a $d_{\text{átlag}}$ esetén, azaz a szomszédok fokszámainak az átlaga.

$$f_{\text{átlag}} = \frac{1}{n} * \sum_{i=1}^n f_i$$

Ahol f_i az i . szomszédos csomópont fokszáma, míg n a szomszédok száma. A protokoll újítása, hogy csak azon csomópontok valószínűségeit változtatjuk az első verzióban látottakhoz képest, amelyeknek magas a fokszámuk. Ezek alapján a valószínűség meghatározása az alábbi képlet megoldásával történik.

$$p_i = \begin{cases} \frac{d_i}{d_{\max}} + \frac{d_{\max}-d_i}{d_{\max}} * \frac{f_i}{f_{\max}} & \text{ha } f_i \geq f_{\text{átlag}} \text{ és } d_i \geq d_{\text{átlag}} \\ \frac{d_i}{d_{\max}} + \frac{d_{\max}-d_i}{d_{\max}} * \frac{f_i}{f_{\max}} & \text{ha } f_i \geq f_{\text{átlag}} \text{ és } d_i < d_{\text{átlag}} \\ \frac{d_i}{d_{\max}} & \text{egyébként} \end{cases}$$

Látható, hogy az első két ág a második verzióhoz, a harmadik pedig az elsőhöz köthető. A távolabb elhelyezkedő, de kevesebb szomszédal rendelkező csomópontok valószínűségi változója kisebb, míg a közelebb lévő, kevesebb szomszédal rendelkező magasabb lesz, mint a második változatban. Természetesen ezen hozzárendelésre is léteznek kedvező és kedvezőtlen esetek hasonlóan, mint az előbb, de itt is elmondható, hogy nem azonos gyakorisággal fordul elő a két elrendezés.

4.5. Összegzés

A bemutatott protokollok működéséhez elengedhetetlenek a csomópontok távolságának ismerete. Ehhez elsődlegesen GPS adó-vevőt használnak, ám lehetőség van, hogy ehelyett a vett rádiós jel erősségből számolják (RSSI) a távolságot. Az utóbbi megoldásnál problémát okozhatnak a rádiós jel útjában álló akadályok, hiszen ekkor a vett jelerősség gyengébb lesz mint az a távolságból adódna. Elmondható továbbá, hogy mind a két megoldáshoz hardveres kiegészítés szükséges, de a mai eszközökben ezek igen nagy gyakorisággal fordulnak elő, így a pontosabb eredmények miatt esett a GPS adó-vevőre az elsődleges választás.

Mind a három protokollnak egyaránt létezik együzenetes és több üzenetes[19] (single - , multi message) megoldása is. Egyszerűen csak a jelzésüzenetek mérete (felépítése) változik, a működés fázisai, valószínűség hozzárendelése azonos mind a két esetben.

A protokollok célja az MCDS-hez hasonló lefedettség kiépítése adaptív továbbadási valószínűségek alapján. Azaz azoknak a csomópontoknak legyen magas a továbbadási valószínűsége, amelyek elemei lehetnek ennek a halmaznak, amelyek sok, eddig még nem lefedett csomópontot tartalmaznak. Az eszközök a birtokukban lévő lokális információk alapján próbálják megjósolni, hogy vajon mely csomópontnak kell magasabb valószínűségi változót biztosítani. Az első változatban láthattuk, hogy pusztán a távolságok alapján is lehet bizonyos fokig megfelelő döntéseket hozni. A következő kettő változat, már arra is ügyelt, hogy a szomszédos csomópontokat ne csak a távolság alapján, hanem a fokszámok alapján is vizsgálja. Természetesen mindig léteznek olyan csomóponti eloszlások, amelyek rendkívül kedvezőtlen eredményt produkálnak a felállított értékelés alapján. Viszont hosszabb időre, több eloszlásra átlagolva ezen speciális eloszlások szerepe nem jelentős. Hogy ezen kijelentések alátámasztást nyerjenek, a protokollok implementálásra és tesztelésre kerültek. A tesztelés eredményeit, illetve az eredmények összehasonlítását más, a szakirodalomban fellelhető, protokollokkal a következő fejezet tartalmazza.

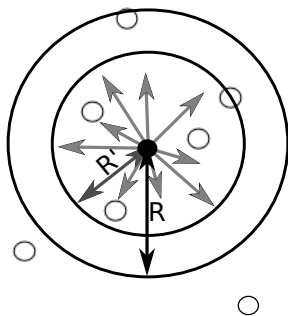
5. Eredmények

5.1. Szimulátor

A tesztelés elengedhetetlen tartozéka a környezet, amiben a tesztelést végrehajthatjuk. Ezen tesztkörnyezetet tartalmazza az általam implementálásra került szimulátor. A szimulátor és a tesztelni kívánt protokollok egyaránt C++ nyelven kerültek implementálásra.

A protokollok a mobil ad hoc hálózatok sajátos tulajdonságaira lettek megtervezve, így a szimulátor feladata ezen speciális környezet előállítás, változtatása, illetve az eredmények kimutatása. A szimulátor képes a folyamat grafikus megjelenítésére, illetve az eredmény szöveges fájlba való kiírására is. A grafikus megjelenítéshez a program az opengl metódusait használja.

A szimulátor egy $2 \times 2 e^2$ alapterületet definiál, amelyen a csomópontok mozognak, illetve kommunikálnak. Természetesen ezen terület mértéke, alakja állítható. A csomópontok ezen a területen véletlenszerű mozgást végeznek. A mozgási modell egy diszkrét függvény, amely megadja, hogy a következő időpillanatban hol bukkanhat fel az adott csomópont. A függvény egyszerűen kijelöl egy R' sugarú tartományt, ahol $R' \neq R$, azaz ez az adási sugártól egy független, továbbiakban állítható paraméter. Ezen tartományon belül pedig tetszőleges pozíciót sorsol magának.



10. ábra. A csomópontok mozgási modellje

A 9. ábrán látható, hogy a feketével jelölt csomópont a szürke R' sugáron belül bármilyen irányban bármekkora távolságot megtehet. Ez a mozgási modell megegyezik a szakirodalomban megtalálható random waypoint mobilitási modellel, amelyet számos MANET szimulátor[20] használ. Sajnos ez a modell sem felel meg teljesen a valós életben történő mozgásokat[21], de erre egyik modell sem képes[22]. Hiszen a valós életben az eszközöket hordozó emberek nem teljesen véletlenszerűen mozognak, vannak állandósult mozgások[23] (munkába menés), illetve vannak ad hoc jelleggel megjelenő útvonalak. Ezen komplex mozgást teljesen leíró modell viszont nem létezik, így hagyatkoznunk kell a bizonyos hiányosságokkal, hibákkal működő modellekre.

A mozgási modell tárgyalása után nézzük meg, hogy milyen paraméterekkel

bír a szimulátor. Az előzőek alapján a mozgási és a rádiós sugár is egyaránt állítható, a mozgási térrel egyetemben. Továbbá az alapterületen mozgó csomópontok száma is módosítható. A szimulátorba az általam megtervezett protokollokon kívül még másik, a szakirodalomban fellelhető, három protokoll is implementálásra került. Ezek a Gossiping, Adaptive Gossiping, és az utolsó, az általam kissé módosított (Modified) Adaptive Gossiping. A Gossiping egy előre meghatározott valószínűséggel adja tovább a beérkező üzeneteket, míg az Adaptive Gossiping a foksámok függvényében ($p_i \sim f_i$) adaptívan választja meg a valószínűségi változót. Az utolsó abban különbözik a sima Adaptive Gossiping-től, hogy a továbbadás nem azonnal történik meg, hanem végrehajtottik a három fázisú mechanizmus, mint az általam tervezett protokollok esetében. Ezzel kívánom szemléltetni, hogy egyrészt a jelzésüzenetek bevezetése, másrészt pedig a valószínűség megfelelő "belövése" is kulcs a protokollok hatékonyabb működéséhez. A szimuláció tetszőleges számban megismételhető, sőt lehetőség van előre megadni a tesztek számát, így a rendszer ezen eseteket átlagolja, hiszen az eredmények csak magasabb számú mérés elvégzése után mutatnak helyes képet. A 4. táblázatban összefoglalva szerepelnek az előzőekben felsorolt paraméterek.

A paraméter neve	megnevezése, leírása
N[db]	A szimulációban résztvevő mobil eszközök száma
R[e]	A szimulációban résztvevő mobil eszközök rádiós sugara
R'[e]	A szimulációban résztvevő mobil eszközök mozgási sugara
T[e ²]	A szimuláció alap területe, amelyen a csomópontok mozognak
Pid	A szimulálni, mérni kívánt protokoll azonosítója
t[db]	A mérések száma, amelyet a rendszer átlagol
mm	Egy bool típusú paraméter, amelynek hamis értéke esetén együzenetes, igaz értéke több üzenetes esetet jelent.

4. táblázat. A szimulátor bemenő paramétereit

Az mm paraméter bekapcsolásával (igazá tételével) a protokollok többüzenetes változata kerül a szimulátorba, míg - alapértelmezett - hamis állapotában együzenetes kommunikáció történik. A Pid egy mutató, amely segítségével hivatkozhatunk a használni kívánt protokollra. Értéke 1 és 6 közé eső egész szám, attól függően, hogy mely protokollt kívánjuk használni. A Pid és a protokollok közötti hozzárendelést ismerteti az alábbi táblázat.

Pid értéke	Protokoll megnevezése
1	DBHG
2	VBHG
3	AVBHG
4	Gossiping
5	Adaptive gossiping
6	Módosított adaptive gossiping

5. táblázat. A protokoll hozzárendelések

A szimulátor kimeneti változói, a mérési eredmények, amelyek alapján adott esetben összehasonlíthatók a protokollok. Ezen eredmények viszont nem egysíkúak, hanem egymástól szorosan függő paraméterek - pontosan négy darab - alkotják. A mérési eredmények összetételét, vizsgálati szempontjait tartalmazza a következő alszakasz, majd a konkrét eredmények bemutatása követi.

5.2. A teljesítmény mutatók

Az előző alfejezetben nyomon követhettük a szimulátor belső felépítését, illetve a bemenő paramétereket. A szimulátor, bizonyos tesztkörülmények között lefolytatott mérések eredményét teszi elérhetővé, így következzen a mérési eredmények kimeneti paramétereinek ismertetése, azaz, amely értékek a protokoll hatékonyságát teszik megítélhetővé.

Az első, és ebben a dolgozatban a legfajsúlyosabb mérőszám, a duplikáció. A duplikáció megadja, hogy hány darab feleslegesen fogadott üzenet létezik a rendszerben. Felesleges, hiszen a csomópontok már birtokolják korábbról az adott üzeneteket. A duplikációk száma kiszámolható, a rendszerben jelenlévő üzenetek és a csomópontok által kért üzenetek különbségeként.

$$\text{duplikáció} = \sum_{i=1}^N i_{\text{üzenet}_{\text{birtokol}}} - \sum_{i=1}^N i_{\text{üzenet}_{\text{szükséges}}}$$

A hálózatban lévő magas duplikáció szám a rendszer nem hatékony kihasználását jelenti, hiszen minden egyes duplikáció elküldéséhez, fogadásához felhasználjuk a csomópontok feldolgozási, illetve a rádiós közeg átbocsató képességét is. Optimális esetben ezen mérőszám értéke 0, azaz minden eszköz csak az általa szükséges üzeneteket birtokolja. Ezen optimum egy mobil ad hoc hálózati környezetben elérhetetlen, olyan okok miatt, mint a rádiós közeg hibavalószínűsége, csomópontok mobilitása, broadcast címzés stb. Természetesen a protokollok célja az optimum megközelítése, a duplikációk visszaszorítása.

Másik fontos kimeneti paraméter a lefedettség, amely megmutatja, hogy egy adott időpillanatban a csomópontok üzenet kéréseinek hány százaléka van kielégítve.

$$\text{lefedettség}[\%] = \frac{\sum_{i=1}^N i_{\text{üzenet}_{\text{birtokol}'}}} {\sum_{i=1}^N i_{\text{üzenet}_{\text{szükséges}}}} * 100\%$$

Ahol az $i_{\text{üzenet}_{\text{birtokol}'}} \neq i_{\text{üzenet}_{\text{birtokol}}}$, hiszen ebben az esetben nem kell többszörösen számolni a birtokolt üzeneteket, hanem csak azt kell vizsgálni, hogy birtokolja-e az általa szükségesnek tartott adatüzenetet. Együzenetes kommunikáció esetén az adatüzenetekkel bíró eszközök számát adja vissza. Több üzenetes esetben pedig a birtokolt üzenetek arányát. A protokollok próbálnak minél magasabb lefedettséget minél kevesebb idő alatt elérni.

A csomópontok broadcast címmel, azaz üzenetszórással juttatják el a rádiós sugáron belül lévő szomszédos csomópontokhoz az üzeneteket. Így a csomagküldések száma megadja, hogy hány üzenetszórás segítségével sikerül elérni egy adott lefedettséget. A rendszer csomagküldéseinek száma megkapható a hálózatot alkotó eszközök csomagküldési számainak összegeként.

$$\text{csomagküldések száma} = \sum_{i=1}^N i_{\text{csomagküldések}}$$

Az üzenetszórás pozitívuma, hogy gyorsan (egyéni címzés nélkül) magas lefedettséget képes elérni, viszont ebből adódóan, nem megfelelő helyeken alkalmazva magas számú duplikációkat okoz a hálózatban. A csomagküldés ugyanakkor teljesítmény felhasználásával jár (hiszen használja az eszközök energiáját, illetve a rádiós spektrumot, amely szűk keresztmetszet a rendszerben), így a protokollok célja ezen szám minimalizálása. Látható, hogy az eddig ismertetett paraméterek milyen szoros kapcsolatban vannak egymással.

Az eredmény negyedik - utolsó - paramétere, a fázisok száma. Megmutatja, hogy adott lefedettség eléréséhez a csomópontok hányszor hajtották végre a küldési mechanizmus három fázisát. Pontosabban a diszkrét idejű szimulátor időpillanatait adja meg. A szimulátor működése alapján minden időpillanatban azon csomópontok, amelyek birtokolnak üzeneteket, végrehajtják a küldési mechanizmus három fázisát. A protokollok a legkevesebb idő alatt kívánják a teljes lefedettséget elérni, hiszen adott esetben így juthat el a lehető leggyorsabban egy sürgős üzenet (például katasztrófa esetén) a legtávolabbi csomóponthoz.

Az alszakaszban ismertetett paraméterek szoros kapcsolatban, illetve ellentmondásban állnak egymással. Hiszen a gyors lefedettség elérése magas duplikációt von maga után, és fordítva. Viszont említésre került, hogy a protokollok céljai között egyaránt szerepel a gyorsaság, és a kevés duplikáció okozása. Természetesen nincs olyan protokoll, amely egyaránt képes a legkevesebb duplikációt okozva a leggyorsabb lefedettség elérésére, hiszen kompromisszumot kell kötni ezen két mérőszám között. Tehát a cél, az az arány megtalálása, amely viszonylag kevés duplikáció árán viszonylag gyors lefedettséget képes elérni. Hasonló párhuzam vonható a csomagküldések és a fázisok száma között. Így az eredmények kiértékelésénél a paraméterek összhatását kell figyelembe venni.

5.3. Az eredmények kiértékelése

A hálózat teljesítményét tehát a fentebb ismertetésre került négy paraméter fogja meghatározni, illetve csak három, mivel az egyik rögzítésre kerül. Azt vizsgáltam, hogy a protokollok milyen gyorsan (fázisok száma), mennyi duplikációt okozva és hány csomagküldés során érik el a 95% -os lefedettséget. Tehát a szimulátor egy tesztje akkor áll meg, ha a megadott protokoll eléri a 95% -os lefedettséget. Így az eredményeket ábrázoló oszlopdiagram három részből fog állni, hiszen a negyedik paraméter rögzített.

5.3.1. Együzenetes eset

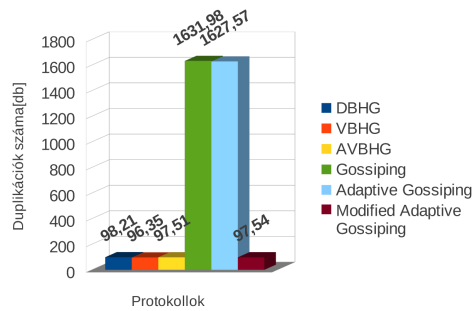
Vizsgáljuk meg először az együzenetes változatok eseményeit. Ekkor a rendszerben csak egy fajta adatüzenet létezik, és feltételezzük, hogy a hálózatban szereplő eszközök mindegyikének szüksége van erre az üzenetre. Erre a valós életből például valami vész jelzése jó példa, hiszen ez egy fajta üzenet (például SOS, vagy VIGYÁZZ), amelyet mindenkinek szeretnénk eljuttatni.

Az első mérési elrendezés bemenő paramétereit mutatja a 6. táblázat.

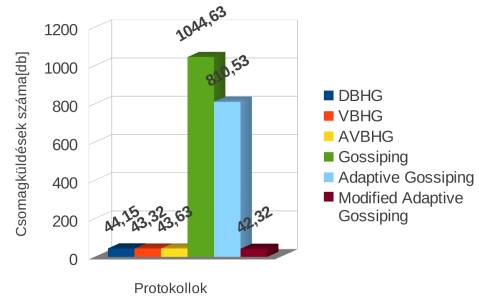
Paraméter neve	értéke
N	50db
R	0.2e
R'	0.1e
T	$4e^2$
Pid	1,2,3,4,5,6
t	100db
mm	hamis

6. táblázat. Az első együzenetes mérés bemenő paramétere

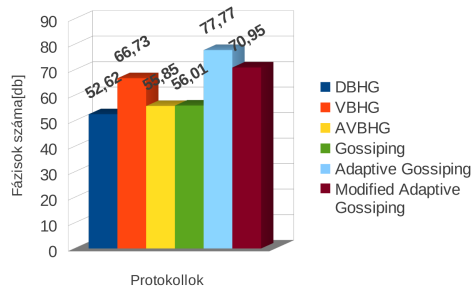
Tehát először egy 50 eszközből álló mobil ad hoc hálózatot vizsgálunk meg, amelyben a csomópontok adási sugara 0,2e, a mozgásuké pedig 0,1e (azaz negyed akkora területen mozoghatnak, mint amekkorán adhatnak), illetve az összes protokoll eredményeit szeretnénk összevetni, így a Pid minden értéket felvesz. A szimulációk szórásának csökkentése érdekében 100 mérést átlagolva kaptuk az eredményeket.



(a) A duplikációk száma



(b) A csomagküldések száma



(c) A fázisok száma

11. ábra. Az első eset eredményei

A 6. táblázat által beállított szimuláció eredménye látható a 11. ábrán. Az a, a duplikációk számát, a b, a csomagküldések számát, míg a c, ábrarész a fázisok számát jeleníti meg. A diagram oszlopain lévő számok jelölik az egyes mérőszámok felvett értékeit, amelyeket az alábbi táblázat összegyűjtve mutat be.

Protokoll	Duplikációk száma	Csomagküldések száma	Fázisok száma
DBHG	98,21	44,15	52,62
VBHG	96,35	43,32	66,73
AVBHG	97,51	43,63	55,85
Gossiping	1631,98	1044,63	56,01
Adaptive Gossiping	1627,57	810,53	77,77
Modified Adaptive Gossiping	97,54	42,32	70,95

7. táblázat. Az első mérés eredményei számokban

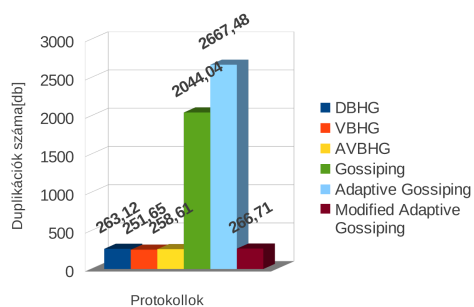
A Protokoll attribútum alatt találhatjuk az általam tervezett és implementált protokollokat (DBHG, VBHG és AVBHG), illetve a szakirodalomban fellelhető megoldásokat. A duplikációk vizsgálata esetén látható, hogy négy protokoll, az általam tervezett három, illetve a Modified Adaptive Gossiping azonosan teljesített, hiszen az 1-2% -os eltérés a mérés hibájának tekinthető. Viszont az is észrevehető, hogy ezen négy és a másik két (Gossiping, Adaptive Gossiping) protokoll között nagyságrendi eltérés van. Ennek oka, a három fázisból álló küldési mechanizmus, amely, ha a szomszédos eszközök mindegyike rendelkezik adatüzenettel, akkor nem teszi lehetővé a csomópont felesleges csomagküldését. Ebből adódnak a csomagküldések számánál található különbségek is. Hiszen elkerülve a felesleges üzenetszórást, kevesebb duplikációt okozva érhető el ugyanaz a lefedettség. A sebességek összehasonlításánál már azonos nagyságrendű számokat találunk. Itt három-három tagból álló protokoll csoportokra oszlik a mezőny. Az első hármas, a DBHG, az AVBHG és a Gossiping, amelyek közel 20-30% -kal jobban teljesítettek, mint a maradék három protokoll. A VBHG ezen a síkon elmaradt a többi változattól, hiszen az átlag távolságon belül lévő szomszédos csomópontok továbbadási valószínűségét nagy mértékben bünteti, ezáltal szűkítve az információterjesztés lehetőségét. Ám a későbbiekben látni fogjuk, hogy másrésről ezen ok miatt, sűrűbb környezetek esetén a duplikációk számát csökkenteni tudja. Az első elrendezésből levonható következtetés, hogy a jelzésüzenetek (illetve a küldési mechanizmus) segítségével képesek vagyunk drasztikusan csökkenteni a csomagküldések és ebből adódóan a duplikációk számát is. További konklúziók állapíthatók meg a többi mérési eredmények segítségével.

Az együzenetes változat következő két mérésének elrendezését a 8.táblázat tartalmazza.

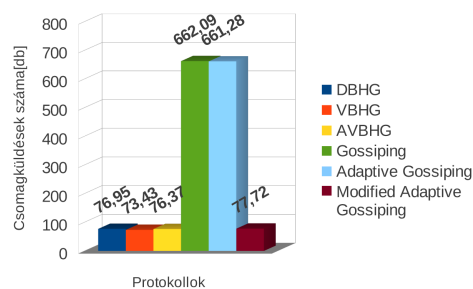
Paraméter neve	értéke
N	100,150db
R	0.2e
R'	0.1e
T	$4e^2$
Pid	1,2,3,4,5,6
t	100db
mm	hamis

8. táblázat. Az együzenetes mérés további bemenő paramétereit

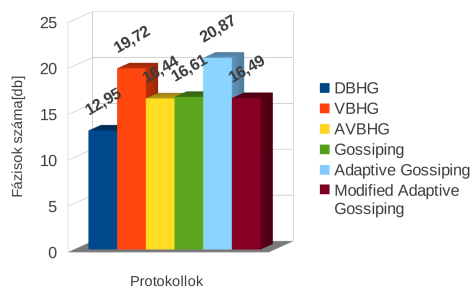
Tehát csak a csomópontok száma változik, először 100 db, majd az utolsó esetben 150 db eszköz fog részt venni a kommunikációban. A 100db csomóponttal végezett mérési eredményeket az alábbi ábra, illetve táblázat tartalmazza.



(a) A duplikációk száma



(b) A csomagküldések száma



(c) A fázisok száma

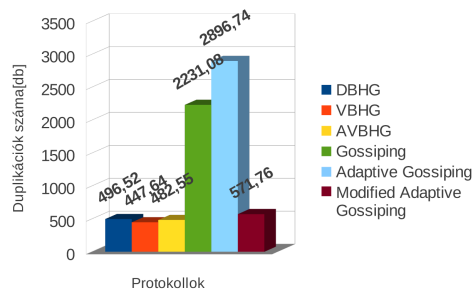
12. ábra. A második eset eredményei

Protokoll	Duplikációk száma	Csomagküldések száma	Fázisok száma
VBHG	263,12	76,95	12,95
DBHG	251,65	73,43	19,72
AVBHG	258,61	76,37	16,44
Gossiping	2044,04	662,09	16,61
Adaptive Gossiping	2667,48	661,28	20,87
Modified Adaptive Gossiping	266,71	77,72	16,49

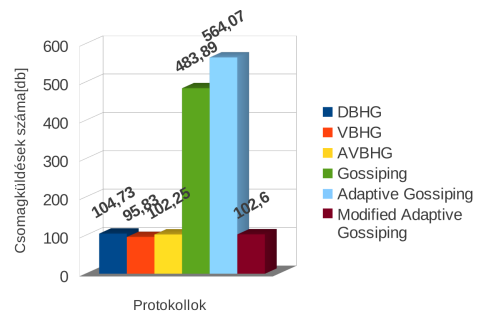
9. táblázat. A második mérés eredményei számokban

A duplikációk számában megmaradt a nagyságrendi különbség a fázisokat alkalmazó protokollok javára. Ami viszont új, hogy a DBHG gyorsaságát tekintve már mindegyik protokollt maga mögött hagyta. Ez a távolság alapú ($\frac{d_i}{d_{max}}$) valószínűség hozzárendelés pozitív vonzata, negatív viszont, hogy az általam tervezett protokollok közül már ez duplikál a legtöbbet, igaz csak kb. 5%-al többet. Itt már számokban is látható az, amit eddig nagyon sokat említettem, hogy kompromisszum kötendő a gyorsaság és a takarékoság között. A második változat okozza a legkevesebb plusz üzenetet a hálózatban, ám pontosan emiatt ez a leglassabb is. A harmadik verzió célja a középút megtalálása, amely kicsit többet duplikál mint a második, és lassabb mint az első, ám gyorsabb mint a második, és kevesebb plusz üzenetet terjeszt, mint az első. Ezen tulajdonság a valószínűségi változó három ágra bontásának köszönhető, hiszen csak a magas fokszámúakat ($f_i > f_{max}$) büntetjük, jutalmazzuk, a többi szomszédos eszköz, hasonlóan mint az első változatban, a távolság alapján kapja a valószínűségét.

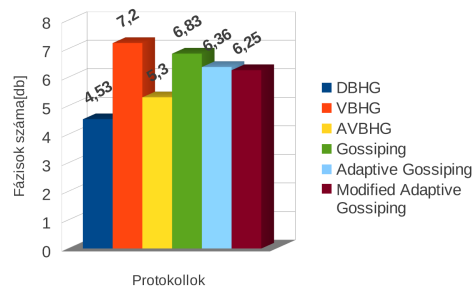
Az előbb említett észrevételekre még relevánsabb példát mutat a 150db eszközből álló hálózat mérési eredménye. Az elvégzett mérések eredményei a 13.ábrán illetve a 10.táblázatban láthatóak.



(a) A duplikációk száma



(b) A csomagküldések száma



(c) A fázisok száma

13. ábra. A harmadik eset eredményei

Protokoll	Duplikációk száma	Csomagküldések száma	Fázisok száma
DBHG	496,52	104,73	4,53
VBHG	447,64	95,83	7,2
AVBHG	482,55	102,25	5,3
Gossiping	2231,08	483,89	6,83
Adaptive Gossiping	2896,74	564,07	6,36
Modified Adaptive Gossiping	571,76	102,6	6,25

10. táblázat. A harmadik mérés eredményei számokban

Itt már jobban kivehető, hogy a második változat kevesebbet duplikál, hiszen a különbség már több mint 10%. Viszont már az is látható, hogy nem csak a jelzésüzenetek segítségével csökkenthető a feleslegesen elküldött csomagok

száma, hanem a megfelelő valószínűség meghatározásával. Hiszen a Modified Adaptive Gossiping, amely egy küldési fázisokkal felvértezett Adaptive Gossiping protokoll, már közel 30% -kal több (az előző esetben ez kb. 7% volt) feleslegesen elküldött üzenettel rendelkezik. Ahogy korábban említettem, ezen eset kitűnő példa, hogy lássuk a harmadik verzió köztes értékeit. Mind a duplikációk, mind a csomagküldések és mind a fázisok száma az első és a második verzió által kijelölt tartományba esik.

Együzenetes kommunikáció esetén ezek alapján elmondható, hogy a megadott környezetben az általam tervezett protokollok duplikációinak és csomag küldéseinek száma nagyságrenddel kevesebb, mint a szakirodalomban fellelhető Gossiping és Adaptive Gossiping protokollok értékei. Sőt létezik olyan változata, amely még gyorsabb kommunikációra is képes. Azt viszont nem szabad elfelejteni, hogy létezhet olyat környezet is, amelyben akár a Gossiping hatékonyabb működésre képes. A protokoll mind három verzióját célszerű telepíteni az eszközökbe, hiszen ezek csak a valószínűség kiszámításának módjában térnek el, ami nem okoz nagy terhelést. Így a csomópontok képesek adaptívan változtatni a valószínűség hozzárendelés függvényét, attól függően, hogy a rendszer mely paraméter(ek)re érzékeny. Ha a rendszer a gyors adatterjesztést követeli meg, akkor az eszköz az első verzióra vált, ellenben, ha magas költséggel jár az üzenet továbbítása, akkor a másodikra, köztes megállapodás esetén a harmadikra. Ezen hibrid protokoll (a három együttes alkalmazása) segítségével még hatékonyabb teljesítményt érthetünk el.

5.3.2. Többüzenetes eset

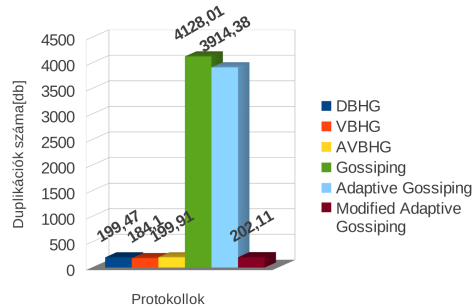
Többüzenetes kommunikáció esetén nem csak egy, hanem számos eltérő típusú adatüzenettel történik az információterjesztés. Feltételezzük, hogy az eszközök halmazokra oszthatóak aszerint, hogy mely üzenettípusok birtoklására van igényük. Azaz nem szükségeltetik minden adatüzenet eljuttatása minden eszközhöz, hanem csak azt kell figyelemmel kísérni, hogy a csomópontok által igényelt csomagok eljutottak-e már hozzájuk. Az általam végzett mérések során feltételezem, hogy három típusú adatüzenet van a rendszerben, illetve, hogy a csomópontok között egyaránt van olyan, amelynek csak az egyikre, illetve amelynek mindegyikre szüksége van. Azaz adott N csomópont esetén, hét $\approx \lfloor \frac{N}{7} \rfloor$ db eszközből álló halmazra osztom szét őket, amely halmazokban lévők mindegyike azonos igénnyel rendelkezik. Tehát a csomópontok halmazai megfeleltethetők az üzenetek részhalmazainak a számával (hiszen az üres halmazt kivéve, mert az nem nevezhető igénynek, $2^3 - 1 = 7$). Így például mondhatjuk azt, hogy a második halmaz elemeinek igénye legyen az első és harmadik üzenet, míg a hatodiknak csak a második. Az azonos halmazban lévő eszközök véletlenszerűen lettek kiválasztva, elkerülve az üzenetek csoportosulását.

A további mérési eredményekhez szükségem van elrendezéseket a 11. táblázat tartalmazza.

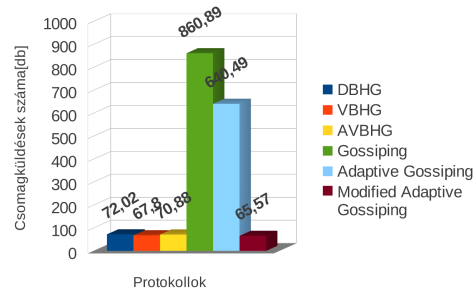
Paraméter neve	értéke
N	50,100,150db
R	0.2e
R'	0.1e
T	$4e^2$
Pid	1,2,3,4,5,6
t	100db
mm	igaz

11. táblázat. Az többüzenetes mérések bemenő paraméterei

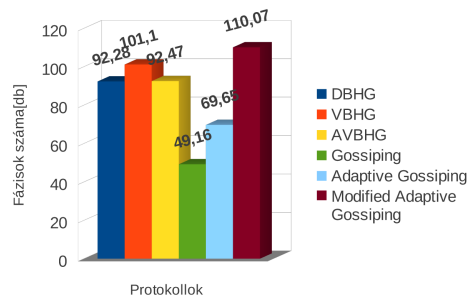
Hasonlóan, mint az együzenetes szimulációk során, itt is 50, 100 és 150db eszközből álló hálózatokat vizsgálunk. Illetve itt is ugyanazon protokollok többüzenetes változataival végezzük a méréseket. Különbség egyedül a táblázat utolsó sorában található, amely azt mutatja, hogy most többüzenetes módra kapcsolt a szimulátor. Azaz, ilyenkor a lefedettség nem pusztán az elért csomópontok száma, hanem az igényelt üzenetek kézbesítésének aránya. Az 50 eszközzel végzett szimuláció eredményeit tartalmazza az alábbi ábracsoport, és hasonlóan az együzenetes méréseknél, az ábra alatti táblázat foglalja össze a rész ábrák értékeit.



(a) A duplikációk száma



(b) A csomagküldések száma



(c) A fázisok száma

14. ábra. Az 50db eszközből álló, többüzemű szimuláció eredményei

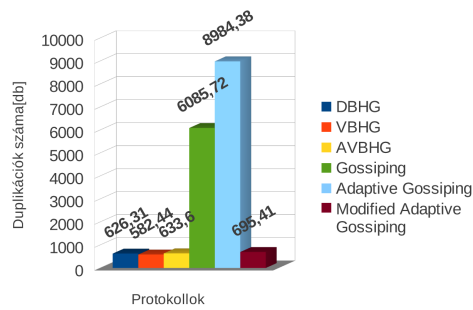
Protokoll	Duplikációk száma	Csomagküldések száma	Fázisok száma
DBHG	199,47	72,02	92,28
VBHG	184,1	67,8	101,1
AVBHG	199,91	70,88	92,47
Gossiping	4128,01	860,89	49,16
Adaptive Gossiping	3914,38	640,49	69,65
Modified Adaptive Gossiping	202,11	65,57	110,07

12. táblázat. Az első többüzemű mérés eredményei számokban

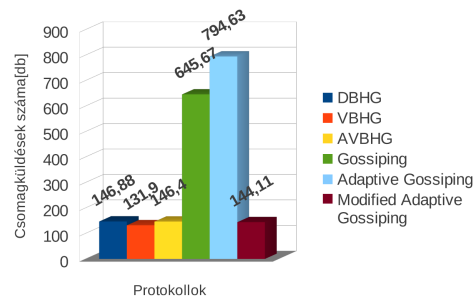
A több típusú adatüzenetek alkalmazásának eredményeképpen a duplikációk száma megnövekedett az együzemű esethez képest. Ez természetes, hiszen több csomag szétszórásával több duplikációhoz is jutunk. Viszont a küldési mechanizmust használó protokollok nem háromszor annyival duplikáltak többet, hanem

csak kb. kétszer annyival, ellentétben a Gossiping és az Adaptive Gossiping protokollokkal. Ennek oka, hogy a visszaküldött CTB jelzésű üzenetben a szomszédos csomópontok jelzik, hogy mely üzenetek birtoklására van igényük, ahogy azt korábban a 6. ábra szemléltette. Látható továbbra is, hogy a jelzésű üzenetek segítségével, egy nagyságrenddel csökken a feleslegesen kiküldött üzenetek, illetve a csomagküldések száma. Ami az együzenetes esetenél csak a 150 csomópont esetén, az jelen esetben már 50 eszköz esetén megmutatkozik, miszerint a második változat kevesebbet duplikálva lassabban végzi az információterjesztést. Különbözik viszont a protokollok sebessége, hiszen míg az előzőekben az első verzió, addig itt a Gossiping protokollok érik el leghamarabb a 95%-os lefedettséget. A protokollok lassulását az okozza, hogy ellenben a Gossiping-gel, csak abban az esetben továbbítódik mindhárom adatüzenet, ha a visszaküldött CTB üzenetek uniója tartalmazza azokat és a küldő csomópont is rendelkezik vele.

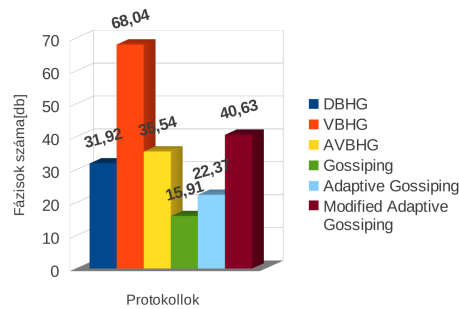
Ezek után vizsgáljuk meg a 100 eszközből álló rendszer teljesítményét.



(a) A duplikációk száma



(b) A csomagküldések száma



(c) A fázisok száma

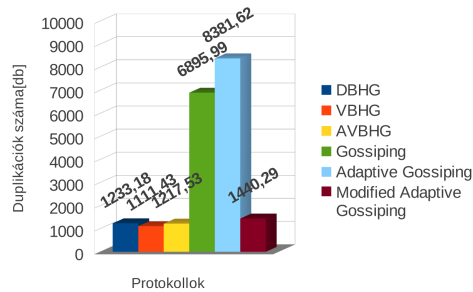
15. ábra. A 100db eszközből álló, többüzenetes szimuláció eredményei

Protokoll	Duplikációk száma	Csomagküldések száma	Fázisok száma
DBHG	626,31	146,88	31,92
VBHG	582,44	131,9	68,04
AVBHG	633,6	146,4	35,54
Gossiping	6085,72	645,67	15,91
Adaptive Gossiping	8984,38	794,63	22,37
Modified Adaptive Gossiping	695,41	144,11	40,63

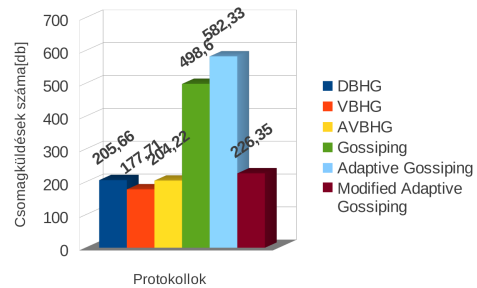
13. táblázat. A második többüzemes mérés eredményei számokban

Az átlag sebességek csökkentek, hiszen sűrűbben helyezkednek el a csomópontok, de a különbségek megmaradtak a Gossiping javára. Észrevehető, hogy a fázisokat használó Modified Adaptive Gossiping magasabb duplikációt (kb. 15%-al) ér el, ennek oka, hogy az általam implementált protokollok a valószínűség kiszámítását más módon, komolyabb összefüggésekre alapozva végzik. Ezen különbséget láthattuk az együzemes kommunikáció során is. Ami viszont újdonság, hogy a harmadik változat nem az első és a második által meghatározott tartományba esik (ez alól kivétel a fázisok száma), hanem gyakorlatilag az első változatával azonos eredményeket mutat.

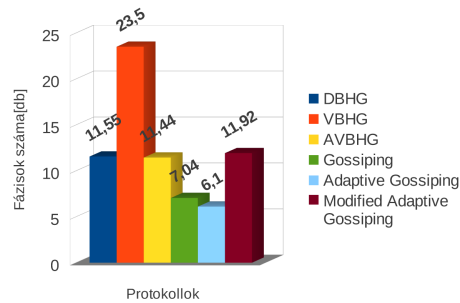
Végül vizsgáljuk meg az utolsó elrendezés (amely 150 eszközből áll) mérési eredményeit.



(a) A duplikációk száma



(b) A csomagküldések száma



(c) A fázisok száma

16. ábra. A 150db eszközből álló, többüzemű szimuláció eredményei

Protokoll	Duplikációk száma	Csomagküldések száma	Fázisok száma
DBHG	1233,18	205,66	11,55
VBHG	1111,43	177,71	23,5
AVBHG	1217,53	204,22	11,44
Gossiping	6895,99	498,6	7,04
Adaptive Gossiping	8381,62	582,33	6,1
Modified Adaptive Gossiping	1440,29	226,35	11,92

14. táblázat. A harmadik többüzemű mérés eredményei számokban

A táblázat értékei még inkább rámutatnak azokra a tendenciákra, amelyeket a 100 csomópontos esetben vettünk észre. Azaz a Modified Adaptive Gossiping már több mint 20% -kal duplikál többet, mint a három változat. Illetve a

második verzió már több mint 10% -kal kevesebb adatüzenetet küldött ki. A sebesség tekintetében pedig még mindig a küldési fázisokat mellőző protokollok a leggyorsabbak.

Összefoglalva a többüzenetes eset eredményeit, azt mondhatjuk, hogy továbbra is egy nagyságrenddel sikerült a feleslegesen elküldött üzenetek, illetve a csomagküldések számát csökkenteni. Ezeket a küldés fázisokra bontásával értem el, viszont jelen esetben ennek meg van a hátránya is, hiszen sebesség tekintetében a jelzésüzeneteket nem használó protokollok jobb teljesítmény nyújtására képesek. Hiszen a távolság alapú valószínűséggel nem érünk el akkora gyorsulást, mint amekkora lassulást elszenvedünk, hogy csak azon üzeneteket továbbítjuk, amelyekre a szomszédos csomópontok igényt tartanak. Így itt is megjelent az, a dolgozat során már sokat említett tézis, amely szerint a duplikációk csökkentése a gyors terjesztés, és az egyszerűség rovására történik. Természetesen itt is a hibrid megoldással érhetjük el a környezet leghatékonyabb kihasználását, azaz mindegyik verziót egyaránt telepítve, adaptívan használva azokat az eszközökben.

6. Konklúzió, jövőbeli tervek

A dolgozat során az olvasó megismerkedhetett az infrastruktúra nélküli mobil hálózatok nehézségeivel. Miképp ebben a környezetben nincsen a csomópontoknak egy olyan kitüntetett halmaza, amely a kommunikáció irányítását végezné. Az ad hoc hálózat miatt nem támaszkodhatunk globális információkra, sőt az eszközök mobilitása miatt, még lokális esetben is ügyelni kell annak érvényességére.

A szakirodalomban számos protokoll létezik az ad hoc hálózati kommunikáció valamilyen szintű megvalósítására, kezdve a legegyszerűbb BF protokolltól eljutva a számos jelzésüzenetet használó IOBIO[14] protokollig. Mindegyik protokoll - az általam tervezett és implementált protokollok sem kivételek ez alól - rendelkezik valamilyen hátrányos tulajdonsággal, hiszen kompromisszumot kell kötni a gyorsaság, egyszerűség és a komplexitás, hatékonyság között. Hiszen a feleslegesen elküldött üzenetek elkerüléséhez jelzésüzenetekre van szükség, viszont ezen jelzésüzenetek overhead-ként (plusz információkként, műveletekként) jelennek meg a rendszerben. A cél tehát az optimum megtalálása, amikor még viszonylag kis overhead árán vagyunk képesek hatékonyan kommunikálni. Ezen célok vezérelték a dolgozatban bemutatott három, név szerint a DBHG, a VBHG és az AVBHG protokollok megalkotását általam.

A protokollok küldési mechanizmusa három fázisra bontható. A fázisokat a bennük használt jelzésüzenetek különböztetik meg egymástól. Az elsőben az adni szándékozó eszköz küld egy RTB üzenetet felszólításként a szomszédos csomópontoknak. A szomszédos eszközök pedig egy CTB jelzésüzenetben válaszolnak a felszólításra, amelyben tudatják a koordinátájukat, illetve az általuk igényelt üzeneteket. Ezek után a megfelelő adatüzenetek halmazával megtörténhet az információterjesztés és a továbbküldési valószínűség hozzárendelés is. Ezen működésnek köszönhetően a protokollok együzenetes és többüzenetes esetben egyaránt hatékonyabb működésre képesek, mint a szakirodalomban megtalálható Gossiping és Adaptive Gossiping protokollok. A duplikációk számát sikerült egy nagyságrenddel visszaszorítani, sőt megfelelően alkalmazva a protokollok verzióit gyorsabb információterjesztés is elérhető vele együzenetes kommunikáció esetén. Hiszen a leghatékonyabb megoldást a három protokoll szimultán használata nyújtja. Ennek feltétele, hogy az eszköz rendelkezzen mindhárom protokollal, viszont ez megoldható egy darab hibrid protokoll telepítésével, amely az érzékeny paraméterek hatására változtatja a valószínűség kiszámítási függvényét. Természetesen ezen eredmények nem azt jelentik, hogy ezek az ideális protokollok, hanem hogy bizonyos környezetben sokkal hatékonyabban képesek kommunikálni, mint a létező protokollok.

Jövőbeli cél a szimulátor fejlesztése, hogy képes legyen más mozgási modellek alkalmazására, illetve többféle mobilitási környezet előállítására is. Továbbá a protokollok valószínűség hozzárendelésének magasabb szintű analízisét kívánom elvégezni, hogy a lokális információkra alapozva még pontosabb eredményekhez juthassunk. Így eljutva újabb verziókhoz, amelyek képesek lehetnek felülmúlni az elődeiket, például gyorsabb információterjesztést elérni még többüzenetes kommunikáció esetén is.

Hivatkozások

- [1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovi. *Mobile Ad Hoc Networking*. IEEE Press John Wiley, 2004.
- [2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *Proc. of ASPLOS-X*, San Jose, CA, Oct. 2002.
- [3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Max-Prop: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. of IEEE INFOCOM*, April 2006.
- [4] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41:pp. 128–136, June 2003.
- [5] K. Fall. A delay tolerant network architecture for challenged internets. In *Proc. of ACM Sigcomm*, Karlsruhe, Germany, Aug. 2003.
- [6] Ozan K. Tonguz, Nawaporn Wisitpongphan, Jayendra S. Parikh, Fan Bai, Priyantha Mudalige, Varsha K. Sadekar: On the Broadcast Storm Problem in Ad hoc Wireless Networks. *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*. S. Basagni, M. Conti, S. Giordano, and I. Stojmenovi. *Mobile Ad Hoc Networking*. IEEE Press John Wiley, 2004.
- [7] Ahmad Al Hanbali, Mouhamed Ibrahim, Vilmos Simon, Endre Varga, Iacopo Carreras: A Survey of Message Diffusion Protocols in Mobile Ad Hoc Networks. In *Proceedings of InterPerf, 2009*. Athen, Greece.
- [8] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proc. of MobiCom*, pages 151–162, New York, NY, USA, 1999. ACM.
- [9] F. Dai and J. Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *IEEE Trans. Parallel Distrib. Syst.*, 15(11):1027–1040, 2004.
- [10] J. Carle, J. Cartigny, and D. Simplot. Stochastic flooding broadcast protocols in mobile wireless networks. Technical Report 2002-03, LIFL Univ. Lille 1, France, May 2002.

- [11] A. Khelil, P. J. Marrón, C. Becker, and K. Rothermel. Hyper-gossiping: A generalized broadcast strategy for mobile ad hoc networks. *Ad Hoc Netw.*, 5(5):531–546, 2007.
- [12] S. Krishnamurthy, X. Chen, and M. Faloutsos. Distance adaptive (dad) broadcasting for ad hoc networks. In *Proc. of MILCOM*, Oct. 2002.
- [13] X. Lu and W. Peng. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proc. of Mobihoc*, 2000.
- [14] E. Varga, T. Csvarics, L. Bacsárdi, M. Berces, V. Simon, and S. Szabó. Novel information dissemination solutions in biologically inspired networks. In *Proc. of ConTEL*, Zagreb Croatia, Jun. 2007.
- [15] F. Dai and J. Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *IEEE Trans. Parallel Distrib. Syst.*, 15(11):1027–1040, 2004.
- [16] Z. Haas, J. Halpern, and L. Li, “Gossip-based Ad Hoc Routing,” *Proceedings of IEEE INFOCOM 2002*.
- [17] Jae-soo Kim, Qi Zhang and Agrawal, D.P. Probabilistic broadcasting based on coverage area and neighbor confirmation in mobile ad hoc networks. *IEEE Global Telecommunications Conference Workshops*, 2004.
- [18] Ahmad Al Hanbali, Mouhamed Ibrahim, Vilmos Simon, Endre Varga, Iacopo Carreras: A Survey of Message Diffusion Protocols in Mobile Ad Hoc Networks. In *Proceedings of InterPerf*, 2009. Athen, Greece.
- [19] Yuxin Chen, Sanjay Shakottai and Jeffrey G. Andrews. Sharing Multiple Messages over Mobile Networks. *IEEE INFOCOM 2011*.
- [20] David Cavin, Yoav Sasson, André Schiper. On the accuracy of MANET simulators. *Proceedings of the second ACM international workshop on Principles of mobile computing*. 2002
- [21] Yoon Jungkeun, Liu Mingyan, Noble Brian. Random waypoint considered harmful. *IEEE INFOCOM 2003*
- [22] Anand D. Sarwate, Alexandros G. Dimakis. The Impact of Mobility on Gossip Algorithms. *Proceedings of IEEE INFOCOM 2009*.
- [23] Wei-Jen Hsu, Thrasyvoulos Spyropoulos, Konstantinos Psounis, Ahmed Helmy. Modeling Spatial and Temporal Dependencies of User Mobility in Wireless Mobile Networks. *IEEE/ACM TRANSACTIONS ON NETWORKING*, VOL. 17, NO. 5, PAGE: 1564-1577, OCTOBER 2009

Ábrák jegyzéke

1.	Példa egy minimális összefüggő fedésre	2
2.	[7]Az SBA működése	6
3.	[7]IOBIO működése	7
4.	Az első fázis	11
5.	A második fázis	12
6.	A továbbküldendő üzenetek meghatározása	13
7.	A harmadik fázis	14
8.	Szomszédos csomópontok információterjesztése	15
9.	Speciális topológiai példák	18
10.	A csomópontok mozgási modellje	21
11.	Az első eset eredményei	27
12.	A második eset eredményei	29
13.	A harmadik eset eredményei	31
14.	Az 50db eszközből álló, többüzenetes szimuláció eredményei . . .	34
15.	A 100db eszközből álló, többüzenetes szimuláció eredményei . . .	35
16.	A 150db eszközből álló, többüzenetes szimuláció eredményei . . .	37

Táblázatok jegyzéke

1.	Az RTB csomag felépítése	12
2.	A CTB csomag struktúrája	13
3.	DATA felépítése	14
4.	A szimulátor bemenő paraméterei	22
5.	A protokoll hozzárendelések	23
6.	Az első együzenetes mérés bemenő paraméterei	26
7.	Az első mérés eredményei számokban	28
8.	Az együzenetes mérés további bemenő paraméterei	29
9.	A második mérés eredményei számokban	30
10.	A harmadik mérés eredményei számokban	31
11.	Az többüzenetes mérések bemenő paraméterei	33
12.	Az első többüzenetes mérés eredményei számokban	34
13.	A második többüzenetes mérés eredményei számokban	36
14.	A harmadik többüzenetes mérés eredményei számokban	37