

Útválasztó algoritmus fejlesztése műholdas ad-hoc hálózatokra

Jeney György András

Konzules: Dr. Hosszú Gábor, e. docens

October 28, 2011

Ez úton szeretném megköszöni kedves szüleimnek akik megteremtették a megfelelő anyagi helyzetet, hogy a, számomra külföldi, színvonalas Budapesti Műszaki egyetemen tanulhassak és még a szabad időmben kutatással is foglalkozhassak.

Megszeretném köszöni Dr. Hosszú Gábor egyetemi docensnek elsősorban azt a sok időt amit velem foglalkozott, hogy segített elmélyülni az ad-hoc hálózatok elméletében és a tévhiteim ellenére is a jó útra terelte a jelen kutatást. Nem utolsó sorban, meg szeretném köszöni azt a sok időt amit panasztalanul töltött el azzal, hogy a nyilvánvaló nyelvtani hibáimra rámutatott.

Tartalomjegyzék

| | |
|--|-----------|
| Tartalomjegyzék | 1 |
| Ábrák jegyzéke | 2 |
| Táblázatok jegyzéke | 3 |
| 1. Bevezetés | 4 |
| 2. Irodalmi Áttekintés | 5 |
| 2.1. A MAC réteg | 5 |
| 2.2. Ad-hoc útválasztási algoritmusok | 6 |
| 2.2.1. A DSR útválasztási algoritmus | 6 |
| 2.2.2. Az AODV útválasztási algoritmus | 7 |
| 2.2.3. A DREAM útválasztási algoritmus | 7 |
| 2.2.4. Az IBRR útválasztási algoritmus | 8 |
| 2.2.5. A BPR útválasztási algoritmus | 8 |
| 2.2.6. A PROPHET útválasztási algoritmus | 8 |
| 2.3. Kepleri pályák | 9 |
| 2.4. Hálózati szimulátor | 9 |
| 3. Felhasznált eszközök | 10 |
| 4. A kifejlesztett SatCom útválasztási algoritmus | 11 |
| 4.1. A műholdak találkozásai | 12 |
| 4.2. Az útválasztási algoritmus működése | 19 |
| 4.3. A távolsági metrika | 20 |
| 4.3.1. A dinamikus távolsági metrika | 21 |
| 4.3.2. A stacionárius távolsági metrika | 22 |
| 4.3.3. A útvonal hossz távolsági metrika | 22 |
| 5. Eredmények | 23 |
| 5.1. Értékelés | 25 |
| 6. Összefoglaló | 28 |
| Irodalomjegyzék | 29 |

Ábrák jegyzéke

| | | |
|------|--|----|
| 2.1. | A láthatatlan egység (hidden node) probléma bemutatása | 5 |
| 2.2. | A „kitett csomópont” (exposed node) probléma bemutatása | 6 |
| 4.1. | Háromszögek segítségével kiszámítva, hogy két műhold maximálisan milyen távolságban képes kommunikálni. | 12 |
| 4.2. | Két találkozás közti idő két véletlenszerűen kiválasztott műhold között | 13 |
| 4.3. | A 4.2. ábrán látható találkozás idősorhoz tartozó autokorrelációs függvény. | 14 |
| 4.4. | Hány találkozásnak van 1, 2, 3, stb. periódusa | 15 |
| 4.5. | Hány találkozásnak van 1, 2, 3, . . . 30 periódusa | 16 |
| 4.6. | Hány százalékát lehet a találkozásoknak lefedni a maximális periódus hossz függvényében | 17 |
| 4.7. | A találkozásokról egy olyan ábra amely jól mutatja, hogy több rövid időközben létrejött találkozás után két műhold sokáig nem fog találkozni | 18 |
| 4.8. | Hány találkozásonként vannak a találkozási szünetek két műhold találkozásaiiban | 19 |
| 4.9. | Hisztogram (eloszlás) a „találkozási holtpontok”-ról két műhold találkozásaiiban | 19 |
| 5.1. | A szerző által végzett szimuláció eredménye, abból a szemszögből, hogy egy adott időn belül hány százaléka a csomagoknak ér célba a CubeSat konstellációban. | 24 |
| 5.2. | Az átlag megérkezési idő annak a függvényében, hogy maximálisan hány párhuzamos útvonalon van egy csomag elküldve, a szorzásos útvonal metrikát alkalmazva. | 26 |

Táblázatok jegyzéke

| | |
|---|----|
| 5.1. A szerző által végzett szimuláció eredménye a kifejlesztett Sat-Com útválasztási algoritmus alkalmazásával, táblázat formában megadva. | 24 |
| 5.2. Statisztikák arról, hogy egy előrejelzés átlagban mennyi időre történik a jövőbe és, hogy az előrejelzések milyen régi adatokon van elvégezve. | 25 |
| 5.3. A különböző konstellációkban a találkozások jósági tényezők inverzének (azaz a rövid időn belüli találkozások valószínűségének) az átlaga és szórása | 25 |
| 5.4. Az átlag megérkezési idő maximálisan 1, 5, 10, 15 és 20 párhuzamos útvonalon való elküldésekor | 26 |

1. fejezet

Bevezetés

Az ad-hoc hálózatokról széleskörű szakirodalom létezik, viszont kevesen foglalkoznak a föld körüli pályán lévő műholdak között létesíthető ad-hoc hálózatokkal. Ugyanakkor ennek a lehetőségnek a jelentőségét mutatja, hogy az amerikai űrkutatási hivatal, a NASA is éppen az űrben létesített ad-hoc hálózatokban látja a jövőt a saját tudományos műhold flottája számára[1]. A kereskedelmi műholdas flottákban ezzel szemben ezt a kérdést eddig nem vizsgálták. Ennek az oka az, hogy ezekben az esetekben egyetlen szervezet felel a rendszer felállításáért, így a műholdas rendszer összes olyan paramétere ismert amely szükséges ahhoz, hogy a műholdak közötti kommunikáció számára elegendő legyen a statikus útválasztás. Továbbá még abban az esetben is, ha a műholdak egymáshoz képesti mozgása miatt az egymás közötti kommunikációt nem lehet teljesen statikus útválasztással szabályozni, akkor is olyan módon van a műholdak közötti kommunikációs rendszer kialakítva, hogy ahhoz egy egyszerűbb útválasztási algoritmus is megfeleljen. Ilyen megoldás például az Iridium és Globalstar[2].

A jelen dolgozatban a szerző egy olyan útválasztási algoritmust dolgozott ki, amely a CubeSat nevű kis műholdakban alkalmazható olyan módon, hogy a műholdakkal a keringési idejük során minél hosszabb ideig lehessen kapcsolatot létesíteni, ne csak akkor, amikor a földi állomás látóterében van a műhold. A kifejlesztett útválasztási algoritmus, ugyan a földi állomással kapcsolatos problémát hivatott megoldani, azonban használható olyan általános esetben is, mikor az egyik műhold egy másikkal szeretne kapcsolatot létesíteni.

A szerző által folytatott szimulációk a saját fejlesztésű szimulátorban megmutatja, hogy még annak ellenére, hogy a CubeSat műholdak véletlenszerű pályákon keringenek, a kifejlesztett SatCom útválasztási algoritmus érdembeli eredményeket ért el.

2. fejezet

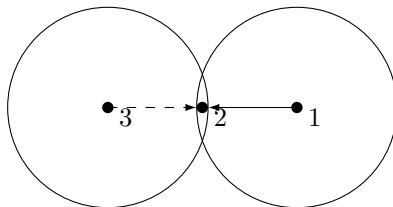
Irodalmi Áttekintés

Minden ad-hoc hálózatbeli kommunikációt befolyásolja a kommunikáló egyedek saját fizikai rétegének jellegzetes tulajdonságai, amelyek elkerülhetetlen adottságok. A hálózatban alkalmazandó útválasztási algoritmus szempontjából például elsődlegesen fontos, hogy a fizikai átvivő médiumhoz milyen módon történik a hozzáférés. Ezzel függ össze, hogy vezeték nélküli, mobil ad-hoc hálózatokban felléphetnek a „láthatatlan egység” (hidden node) illetve a „kitett csomópont” (exposed node) problémák.

Mivel a műholdak közötti kommunikációban — azok egymáshoz képesti helyzetének állandó változása miatt — nem használhatóak módosítás nélkül a klasszikus ad-hoc útválasztási algoritmusok, ezért érdemes azokat áttekinteni.

2.1. A MAC réteg

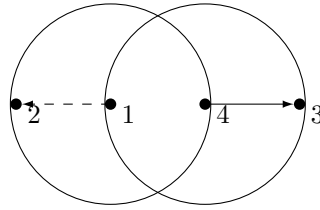
A MAC (Media Access Control) és a fizikai réteg befolyásolja az útválasztási algoritmust és annak hatékonyságát. Az ezzel kapcsolatban fellépő két legjellegzetesebb probléma a „láthatatlan egység” (hidden node) és a „kitett csomópont” (exposed node) problémák[3]. A láthatatlan egység probléma (lásd a 2.1. ábrát) akkor lép fel, amikor egy egység (az ábrán a 3-as egység) nem veszi észre, hogy egy másik egység (az ábrán az 1-es egység) már küld egy csomagot egy közbenső egységnek (az ábrán a 2-es egység) és ezért az is elkezd egy csomag küldését, ami ezért a közbenső egységnél ütközik a másik egység (az ábrán az 1-es egység) által elküldött csomaggal.



2.1. ábra. A láthatatlan egység (hidden node) probléma bemutatása

A „kitett csomópont” (exposed node) probléma abból adódik, hogy egymással kommunikációs kapcsolatban nem álló egyedek egymás hatókörén belül

kerülve zavarhatják egymás adását. A kitett csomópont probléma úgy léphet fel, hogy (lásd a 2.2. ábrát) egy egység (az ábrán az 1-es) egy csomagot kíván küldeni egy másik egységnek (az ábrán a 2-es), ugyanakkor egy másik egység-pár (az ábrán a 3-as és a 4-es egység) már használja ugyanazt a fizikai közeget (vagyis a két adó egymás hatókörén belül található). Ezt érzékeli az adni kívánó egység (az ábrán az 1-es), ezért mégsem kezdődik meg az adatátvitel amíg ez az egység egy másik adását érzékeli a hatókörén belül.



2.2. ábra. A „kitett csomópont” (exposed node) probléma bemutatása

E két probléma megoldására több tanulmány is született, viszont ezek közül sokan csak a „kitett csomópont” problémát vagy a „láthatatlan egység” problémát oldják meg, vagy csakis egy közelítő megoldást adnak mindkét problémára (ilyen például az IEEE 802.11 „WiFi” szabvány RTS/CTS szekvenciája). Ugyanekkor vannak megoldások amelyek teljesen megoldják e két problémát, például [4]. A jelen dolgozat a kitett csomópont és a láthatatlan egység kérdésével nem foglalkozik, mert a szimulációk során feltételezem, hogy e két probléma a műholdak igen ritka elhelyezkedése és ritkán kommunikációs kapcsolata miatt nem lép fel.

2.2. Ad-hoc útválasztási algoritmusok

Az ad-hoc útválasztási algoritmusokat két nagy csoportba tudjuk osztani: a proaktív és reaktív útválasztási algoritmusok. A proaktív útválasztási algoritmusban minden egységnek ismernie kell a hálózat egész topológiáját, továbbá, hogyha változás történik, akkor ezt az információt minden egységhez el kell juttatni. Ezzel szemben a reaktív protokoll csakis akkor ismeri meg a hálózat topológiájának egy részét, amikor egy csomag megérkezése miatt felmerül a konkrét igény egy útvonal megtalálására két egység között.

Bár a reaktív útválasztási algoritmusok esetén az adatátvitelt lassítja, hogy eltart egy ideig az az útvonal felállítása illetve kiszámítása, viszont jelentős előnyük a proaktív útválasztási algoritmusokkal szemben az, hogy nem kell sávszélességet pazarolniuk arra, hogy a későbbi tényleges adatátvitel szempontjából érdektelen topológiai változásokat továbbítsanak a kommunikációban résztvevő egységek egymásnak.

2.2.1. A DSR útválasztási algoritmus

A a Dinamikus Forrás Útválasztás (Dynamic Source Routing, DSR) nevű útválasztási algoritmus[5] egy reaktív útválasztó protokoll. A DSR a kiválasztott útvonalat az elküldendő csomag fejlécébe helyezi el. Ezt az előre kiválasztott útvonalat a kommunikáció során a csomópontok néhány kivételes esettől eltekintve

betartanak, így általánosságban kijelenthető, hogy az útvonal választásban kizárólag a forrás egység vesz részt. A DSR protokoll egy további tulajdonsága az, hogy egy adott útvonalon mentén lévő minden közbenső egység is eltárolja az útvonalat annak érdekében, hogy ne kelljen azt újra számolniuk ha például egy olyan egységnek szeretnének egy csomagot küldeni, amely szintén ugyanezen útvonal mentén helyezkedik el.

A DSR algoritmusnak a fentiekben leírt „forrás útválasztás” („source routing”) tulajdonsága miatt, az útválasztási algoritmust lokálisan meg lehet oldani, vagyis nem kell egy elosztott algoritmust tervezni), továbbá viszonylag könnyen megoldható az útvonal hurokmentesítése.

2.2.2. Az AODV útválasztási algoritmus

Az Ad hoc On-Demand Distance Vector (AODV) útválasztási algoritmus[6] hasonlóan a a DSR algoritmushoz egy reaktív útválasztási eljárás. Ugyanakkor eltér a DSR algoritmustól abban, hogy az útvonal elosztott jelleggel van eltárolva a kommunikációban résztvevő egységekben. Ezért fontos, hogy az algoritmus külön foglalkozzon az útvonal hurokmentesítésével. Hasonlóan a DSR algoritmushoz, az AODV útválasztási algoritmus feltételezi, hogy a kommunikáló egységek közti kapcsolat viszonylag stabil (azaz, a kapcsolatok időtartalma nagyságrendileg nagyobb mint amennyi ideig tart egy csomag továbbítása az útvonal mentén), továbbá, hogy az útvonal érvényessége alatt minden időpillanatban az útvonal összes kapcsolata érvényes. Sajnos a nem minden vizsgált műholdas konstellációban teljesül. Konstelláció alatt egy műholdflottában résztvevő műholdak egymáshoz képesti mozgását értjük.

2.2.3. A DREAM útválasztási algoritmus

A Distance Routing Effect Algorithm for Mobility (DREAM) útválasztási algoritmus[7] egy proaktív protokoll, amely a DSR és az AODV útválasztási algoritmusokkal szemben kihasználja azt, hogy minden egység ismeri a saját abszolút koordinátáit a térben és ezeket a koordinátákat az összes többi, a kommunikációban részt vevő egységnek továbbítja. A DREAM útválasztási algoritmus feltételezi, hogy az egyes egységek mobilitása ismert, azaz rendelkezésre áll, vagy az egységek maximális sebessége, vagy az egységek sebességének eloszlása az időben. Ez az algoritmus kihasználja azt a tulajdonságot, hogyha egy egységet minél távolabbról nézzünk, akkor a helyzete annál kisebb mértékben *látszik* változni.

A műholdak helyzete egy jellegzetes probléma, mert nem tételezhető fel, hogy a műholdak a térben egyenletesen oszlanak el, továbbá egy műholdnak a mobilitása (ebben az esetben a sebessége) túl nagy ahhoz, hogy a tér egy jól meghatározott részében pusztán egy múltbeli megfigyelés alapján megtalálható legyen. A műholdak egymásközi kommunikációjával kapcsolatos másik lényeges gond abból adódik, hogy a műholdak viszonylag ritkán találkoznak. Így, még ha ismert is, hogy egy műhold egy adott pillanatban hol található meg a térben, nem sokat segít, mivel általában nincs a közelében egy másik műhold, hogy annak továbbítsa a küldendő adatsomagot.

A műholdak ugyan elvben használhatnák a GPS rendszert, hogy meghatározzák az abszolút pozíciójukat, azonban a többlet súly, energiaigény, stb. költségei miatt a műholdakban általában nem alkalmaznak GPS rendszereket.

A jelen dolgozatban a továbbiakban feltételezem, hogy egy műhold nem képes a saját pozícióját meghatározni. Ennek a feltételezésnek az az előnye, hogy az általam kidolgozott műholdas ad-hoc hálózatokban használható útválasztási algoritmus nem korlátozódik a „nagy” műholdas hálózatokra (ahol az erőforrás a műholdakban nem annyira szűkös mint a kisméretű műholdak esetén), hanem akár pikóműholdak egymásközti kommunikációjában is alkalmazható.

2.2.4. Az IBRR útválasztási algoritmus

Az Interrogation Based Relay Routing (IBRR) útválasztási algoritmus[8] arra a műholdak kommunikációjában fellépő jellegzetes problémára koncentrálna, hogy a műholdak ritkán találkoznak és ezért az egy útvonalat képző kapcsolatok csak „szakaszosan” léteznek.

Az IBRR útválasztási algoritmus reaktív ugyan, de mégsem határozza meg az egész útvonalat, hanem csak azt vizsgálja meg, hogy mekkora valószínűséggel tudják a pillanatnyi szomszédok szomszédjai továbbítani a csomagokat a célállomáshoz. Vagyis ez az útválasztási algoritmus csak két „ugrásnyit” néz előre. Az IBRR által végzett valószínűségszámítás során figyelembe kell vennie a kommunikációban résztvevő műholdak pályáinak az adatait, ezek azonban nem állnak mindig a rendelkezésre.

2.2.5. A BPR útválasztási algoritmus

A Balanced Predictable Routing (BPR) útválasztási algoritmus[9] a DSR útválasztási algoritmuson alapul és azt használja ki, hogy a DSR a klasszikus ad-hoc útválasztási algoritmusok közül az, amelyik a legkevesebb memóriát és a legkevesebb sávszélességet használ el a csatornából. A DSR-el ellentétben viszont a BPR algoritmus több lehetséges útvonalat tárol és az útvonalválasztási metrikája egy adott módon számított jósági tényező. Ezen algoritmus számításában minden műhold párhoz hozzárendel egy jósági tényezőt, ami a kapcsolat minőségén alapul.

A műholdas kommunikáció szempontjából a BPR hátránya, hogy nem veszi számításba azt, hogy a véletlenszerű eloszlású CubeSat-ok esetén sok idő telhet el addig, amíg egy műhold találkozik egy másikkal, továbbá a BPR útválasztási algoritmus feltételezi, hogy viszonylag gyorsan meg lehet keresni egy útvonalat két műhold között.

2.2.6. A PROPHET útválasztási algoritmus

A Probabilistic ROuting using History of Encounters and Transitivity (PROPHET) útválasztó algoritmus[10] rámutat arra, hogy érdemes valószínűséggel modellezni, hogy két egységek mikor találkozik. Az útválasztó algoritmust eredetileg nem műholdas alkalmazásokra fejlesztették ki, tehát a találkozási valószínűség kiszámításában azt nem használja ki, hogy a találkozások periodikusak. A találkozási valószínűséget inkább úgy választja, hogyha két egység nem találkozott egy ideig akkor a valószínűség csökken, ezzel szemben amikor az egységek találkoznak akkor a találkozási valószínűség növekszik.

Műholdas alkalmazásokban, mivel nagyon ritkán találkoznak az egységek, a találkozás metrikája jó közelítésél konstans kell, hogy legyen, különben az érvénytelené válna amikor azt egy távoli egységnek fel kellene használnia.

2.3. Kepleri pályák

Minden ad-hoc hálózat szimulációhoz nagyon fontos kellék egy megfelelő mobilitási modell. Mivel a jelen tanulmány a műholdak között felállított ad-hoc hálózatokkal foglalkozik, az alkalmazott mobilitási modell a Kepleri pálya. A kutatás során felhasználtam azt, hogy az amerikai NORAD ügynökség sok műholdnak a pályáját követi, beleértve a CubeSat műholdakat amelyekre a kutatásom során koncentráltam. A NORAD által szolgáltatott adatok felhasználhatóak arra, hogy a előre lehessen jelezni a műhold pályákon megjelenő perturbációkat. Ezek figyelembe vétele[11] pedig pontosabb és valóságosabb eredményeket adhat. Az eredeti egyenleteken pontosított és frissített a [12] tanulmány, amelyben közölteket a kutatásom során felhasználtam mobilitási modelljeként.

Egy műhold helyzetét az ellipszis alapú keringési pálya ismerete és a Kepler egyenlet adja meg:

$$M = E - \epsilon \sin E$$

Ahol M arányos az idővel és E az a szög amely meghatározza, hogy a test (esetünkben a műhold) hol található a keringési pályán. Tehát, hogyha az időfüggvényében szeretnénk kifejezni egy műhold jelenlegi helyzetét, akkor Kepler egyenletét kellene megoldani E -re, amire azonban nem létezik analitikus megoldás.

Két műhold relatív pályájának meghatározása érdekében a műholdak mobilitási modelljének célszerű a Kepler egyenletet megoldani általános esetben. A szakirodalomban két műhold relatív pályájának megoldására csak elvétve található leírás. Ezen kevés eljárás közül a legalkalmasabb a [13] tanulmányban leírtak szerinti eljárást találtam. Ez a tanulmány azzal a feltételezéssel él, hogy az egyik műhold egy körpályán kering, ami ugyan nem rossz közelítés a CubeSat pályák esetén, azonban a tanulmány azt is feltételezi, hogy ismert a műhold keringési pályájának a sugara, amit az általam végzet vizsgálatokban ismeretlennek tételeztem fel.

2.4. Hálózati szimulátor

A hálózat szimulációkban a legelterjedtebb az ns-3 szimulátor[14]. Ez a szimulátor a hálózat összes rétegét szimulálja, kezdve a médiumtól, a fizikai egységen át egészen az alkalmazásig. A jelen tanulmányban kizárólagosan az útválasztó algoritmusokkal foglalkoztam, ezért célszerűbbnek láttam egy teljesen új szimulátort fejleszteni minthogy az összes hálózati réteggel is foglalkozzak.

3. fejezet

Felhasznált eszközök

A felhasznált eszköz elsősorban az sgdp4 műhold keringési pályát számító rendszer, leírva [12]-ben és letöltve [15]-ről. Továbbá, a pálya-szimulátor számára szükségesek a pályaadatok. A pálya adatok a [16], [17] és [18]-ből kaphatók. Az OrbComm és az Iridium pályaadatokból eltekintettem azokat a műholdakat, amelyek már üzemben kívül vannak, vagy csak tartalékként szerepelnek a rendszerben.

4. fejezet

A kifejlesztett SatCom útválasztási algoritmus

A műholdak közötti útválasztási algoritmus kifejlesztésének nehézsége abból adódik, hogy a Föld kerülete viszonylag nagy ahhoz képest, hogy hány műhold áll rendelkezésre egy ilyen ad-hoc kommunikációs rendszerben. Az általam elvégzett vizsgálatok azonban bebizonyították, hogy még egy viszonylag kislétszámú műholdflotta esetén is adódik lehetőség egy érdemben alkalmazható útválasztási algoritmus megszerkesztésére. A műholdas kommunikációra kifejlesztett útválasztási algoritmust SatCom-nak (Satellite Communicator) neveztem el.

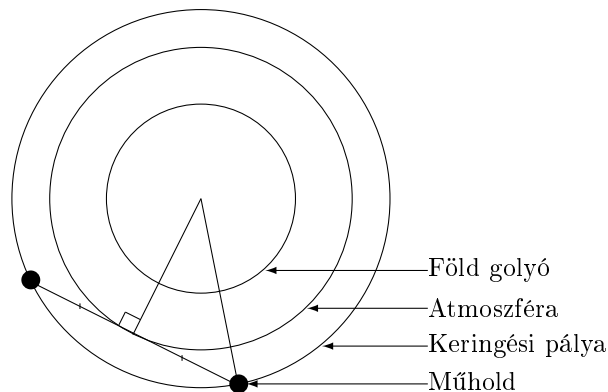
Az útválasztási algoritmus megtervezésekor — a korábban ismertetett szakirodalmi adatok alapján — a következő közelítéseket tétéleztem fel:

1. Minden műhold fel van szerelve egy irányfüggetlen antennával.
2. A fizikai réteg hozzáférése garantálja, hogy ne lépjen fel a láthatatlan és a kitett egység (hidden és exposed node) problémája.
3. Zajmentes a kommunikációs csatorna.
4. Két műhold akkor képes üzeneteket egymásnak továbbítani, amikor legalább 1600km távolságon belül vannak.
5. A fizikai réteg lehetővé teszi a „szórtadás” (broadcast) csomagok küldését, amelyeket minden olyan egység megkap, amely az 1600km-es kommunikációs távolságon belül található.
6. Az egyes egységekben működő óra szinkronizált az összes többi egységgel.
7. Egyik egység sem ismeri a saját keringési pályáját vagy pontos helyzetét.
8. Egy csomag elküldéséhez szükséges idő nagyságrendekkel kisebb mint amennyi ideig két műhold a kommunikációs távolságon belül van.

A zajmentes csatorna feltételezés, különösen az űrbeli alkalmazásoknál, ugyan nem teljesen jogos, viszont a csatornabeli jel/zaj viszony modellezhető a kommunikációs távolság változtatásával. Mivel e dolgot csak azon műholdakkal foglalkozik amelyek alacsonyan keringenek a föld felett (150km-1500km magasan) a kommunikációs távolság a maximálisnak van választva. Ez lehetséges abban

az esetben, ha két műhold 400km magasságban kering. Ennél alacsonyabban a légkörrel való fokozott sűrűdés miatt a műhold keringése gyakorlatilag nem lehetséges. Az elvégzett számítások során feltételeztem továbbá, hogy a föld feletti 350km-es atmoszférában túlságosan nagy a zaj ahhoz, hogy két műhold azon keresztül képes legyen kommunikálni. Ezekkel az adatokkal és a következő egyenlettel megkapható a maximális kommunikációs távolság (lásd a 4.1. ábrát):

$$\text{kom. táv} = 2\sqrt{(r_{\text{föld}} + \text{Műhold magasság})^2 - (r_{\text{föld}} + \text{Atmoszféra})^2}$$



4.1. ábra. Háromszögek segítségével kiszámítva, hogy két műhold maximálisan milyen távolságban képes kommunikálni.

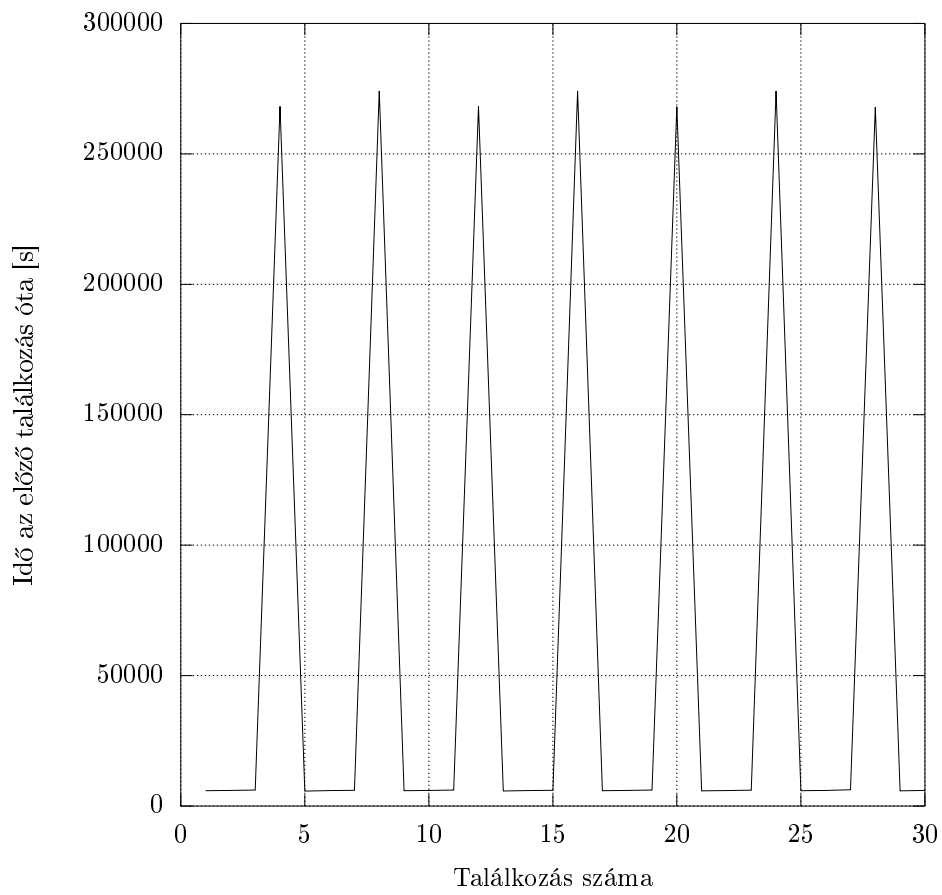
Az órák szinkronizáltságának egyébként jelentős feladatával a jelen dolgozat nem foglalkozik. Ezt a problémát számos tanulmányban vizsgálták és mostanra már több időszinkronizációs protokollt kidolgoztak, amelyek a műholdak közötti kommunikáció gyakorlatában alkalmazhatók. Ilyenek például a [19] és [20]. Az általam kifejlesztett ComSat útválasztási algoritmusban az abszolút időt kizárólag arra használom, hogy két műhold eldöntse azt, hogy melyiküknek van frissebb adata két műhold közti találkozásról. Tekintettel arra, hogy két műhold csak nagyon ritkán találkozik (leggyakrabban is csak 3000 másodpercenként) az időt sem szükséges ennél lényegesen jobban szinkronizálni. Az 1000 másodperc nagyságrendjébe eső szinkronizációt viszont a szakirodalomból ismert időszinkronizációs protokollokkal nem nehéz elérni.

4.1. A műholdak találkozásai

Mivel a Kepler-egyenletet nem lehet analitikusan megoldani, ezért nem ennek a közelítő megoldását választottam, hanem a két műhold egymáshoz képesti mozgását statisztikai alapon vizsgáltam.

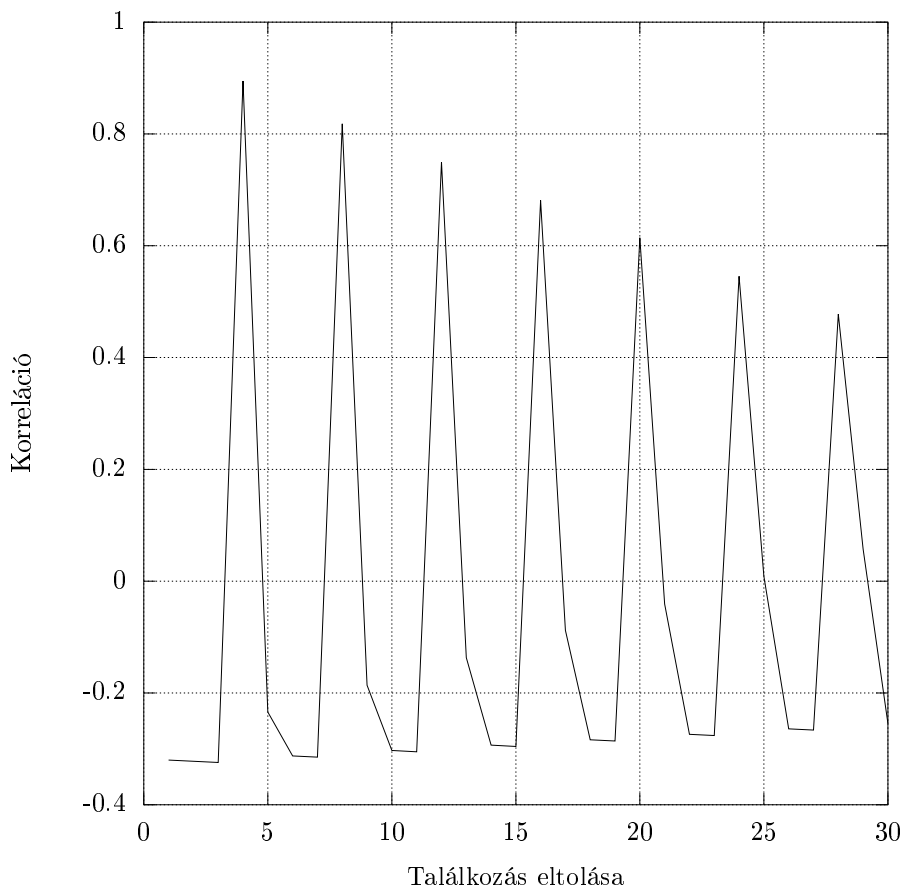
A statisztikai adatsokaság generálására felhasználtam a CelesTrak által kiadott összes CubeSat műhold pályadatait. Ez az adatbázis úgy keletkezett, hogy amikor két műhold közelebb került egymáshoz mint 1600km, akkor feljegyeztem, hogy mennyi idő telt el az előző találkozás óta. Vagyis azt, hogy két találkozás között mennyi idő telt el. Egy ilyen idősor látható a 4.2. ábrán.

Az ábrán jól látható, hogy a találkozások valamilyen periodicitást mutatnak. Az autokorrelációs függvényt alkalmaztam annak eldöntésére, hogy ez a



4.2. ábra. Két találkozás közti idő két véletlenszerűen kiválasztott műhold között

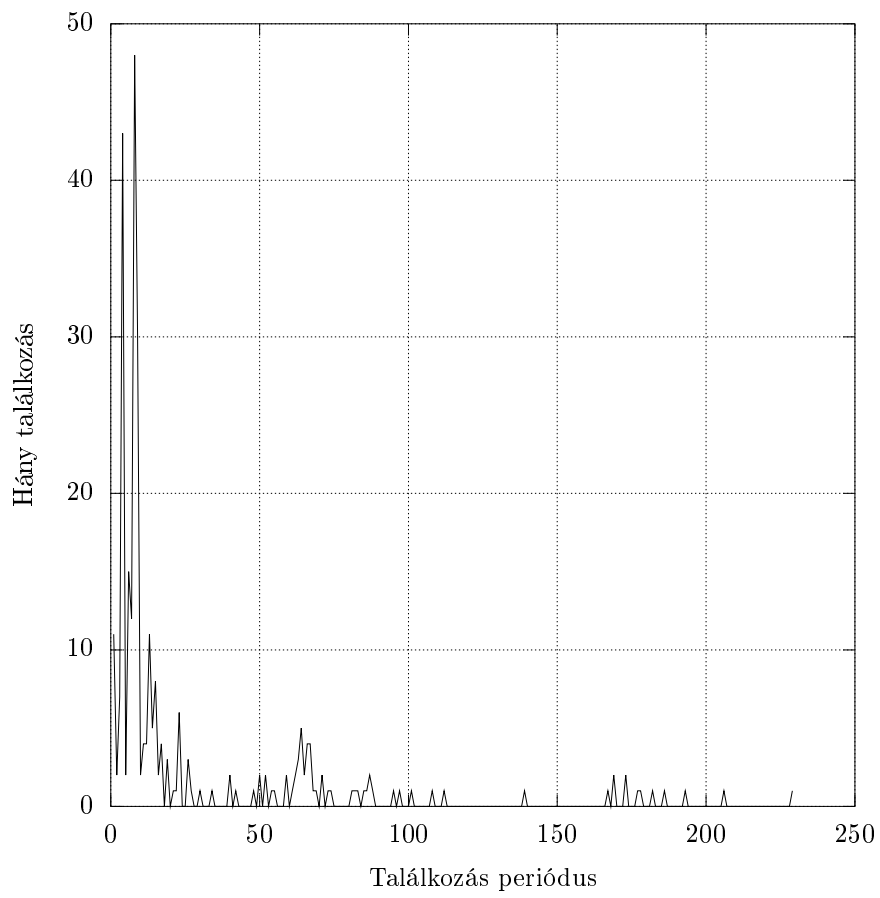
periodicitás menyire használható ki. A fenti idősor autokorrelációs függvénye a 4.3. ábrán látható. Az ábrán a csúcsok periodicitása jól mutatja, hogy a találkozások is mutatnak valamely periodicitást, továbbá az is jól látszik, hogy a csúcsok nagysága csökken. Ez azt jelenti, hogy minél tovább megyünk előre az időben, annál pontatlanban lehet előrejelezni a találkozásokat. Hogyha megnézzük az összes találkozás autokorrelációs függvényét és feljegyezzük, hogy hol van mindegyik maximuma, továbbá megnézzük az ebből eredő hisztogramot (hány találkozás periódusa 1, 2, 3, stb.) akkor a 4.4. ábrát kapjuk. Ennek az elejét kinagyítva a a 4.5. ábrát nyerhetjük.



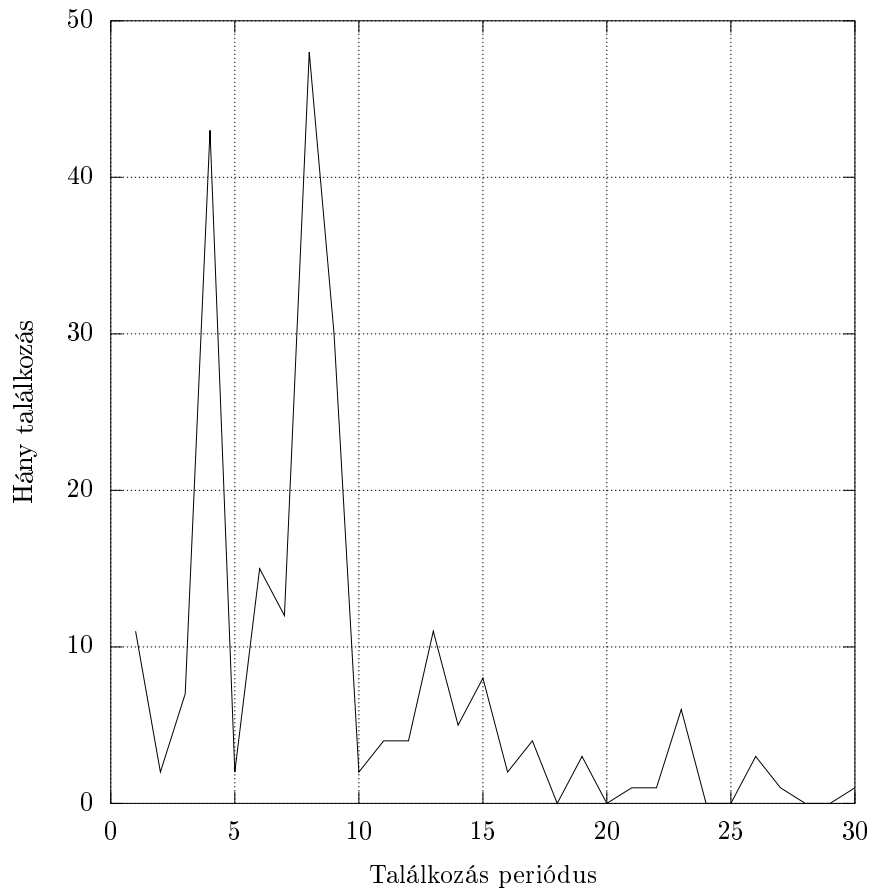
4.3. ábra. A 4.2. ábrán látható találkozás idősorhoz tartozó autokorrelációs függvény.

Az ábrákból jól látszik, hogy a legtöbb találkozás viszonylag rövid periódusokkal rendelkezik. A 4.6. ábra megmutatja, hogyha attól függően, hogy az útválasztási algoritmus mekkora maximális periodicitást feltételez, változik annak az aránya, hogy a találkozások hány százalékát lehet lefedni.

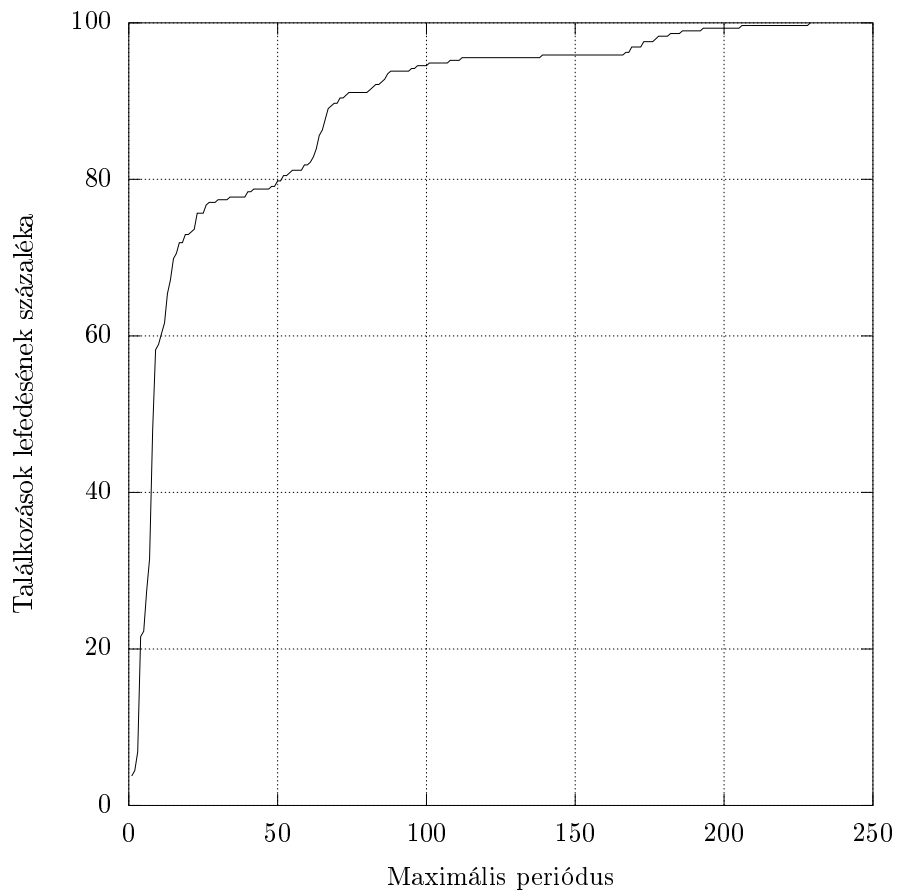
A találkozások másik lényeges tulajdonsága az, hogy minden műholdpár



4.4. ábra. Hány találkozásnak van 1, 2, 3, stb. periódusa

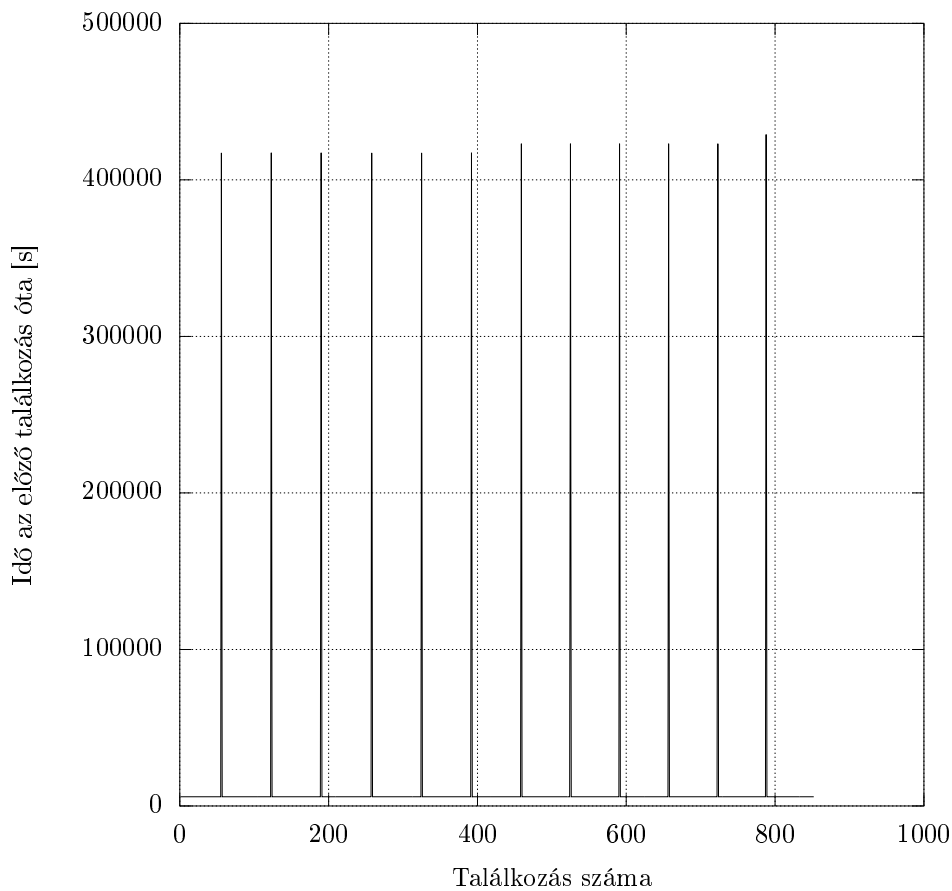


4.5. ábra. Hány találkozásnak van 1, 2, 3, ... 30 periódusa



4.6. ábra. Hány százalékát lehet a találkozóknak lefedni a maximális periódus hossz függvényében

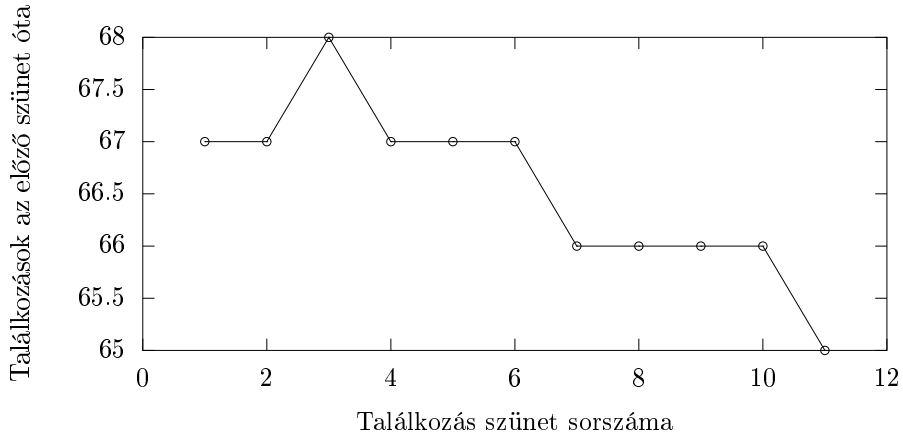
amely valamikor találkozik, mutat egy hosszútávú periodicitást, hogy több relatíve közeli találkozás után hosszú ideig nem találkoznak (példa a 4.7. ábrán). Ezeknek a nem találkozás időknak az a tulajdonsága, hogy nagyságrendekkel nagyobbak mint a csúcsok közti összes idő összege. Tehát, hogyha a csúcsok helyét nem lehet pontosan előrejelezni, akkor minél tovább próbálunk a jövőbe nézni, hogy két egység mikor fog majd találkozni, minél nagyobb a valószínűsége annak, hogy az előrejelzés hibája nagy lesz.



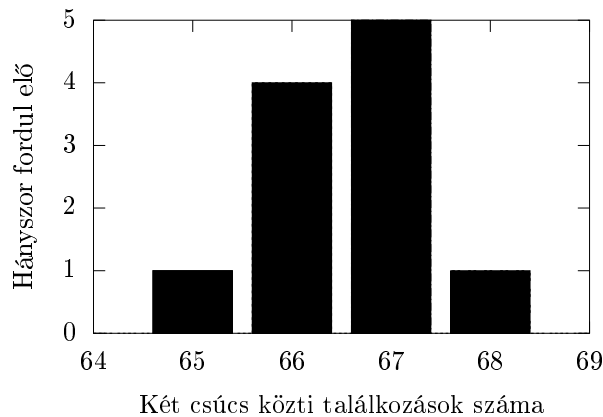
4.7. ábra. A találkozásokról egy olyan ábra amely jól mutatja, hogy több rövid időközben létrejött találkozás után két műhold sokáig nem fog találkozni

Annak érdekében, hogy lássuk, milyen pontosan lehet előrejelezni azt, hogy mikor vannak ezek a találkozásmentes időszakok, a 4.8. ábrán megvan adva, hogy két ilyen magas találkozási csúcs között hány találkozás történik. Az ábráról és az ábra hisztogramjáról (4.9. ábra) jól látszik (és az általam elvégzett szimulációk során numerikusan is igazoltam), hogy a csúcsok véletlenszerűen oszlanak el, vagyis nem jelezhetők kellő pontossággal előre. Az elvégzett szimulációkból az is kiderül, hogy a találkozásokat is a távoli jövőben kell előrejelezni

— 3 vagy még több nap előrejelzésre is szükség volt a szimulációk során. Ilyen feltételek mellett azonban a múltbeli adatokból inkább csak az dönthető el, hogy mely valószínűséggel kerül egy csomag legalább egy bizonyos időn belül továbbítva.



4.8. ábra. Hány találkozónként vannak a találkozási szünetek két műhold találkozásaiban



4.9. ábra. Hisztogram (eloszlás) a „találkozási holtpontok”-ról két műhold találkozásaiban

4.2. Az útválasztási algoritmus működése

Az általam kifejlesztett SatCom útválasztási algoritmus egy proaktív útválasztási eljárás. A reaktív útválasztási algoritmusok ugyanis semmi képen nem alkalmas a műholdak kommunikációban mivel ekkor minden útvonalat kétszer kéne bejárni és mivel egy csomag továbbítása relatíve hosszú ideig tart (óra nagyságrendjébe esik) az útvonal felállítása ennek legalább a kétszerese lenne. Továbbá, a műholdak mobilitása nagy ahhoz képest, hogy hányszor találkoznak,

azaz, attól egy útvonal egyszer kiszámításra került, még egyáltalán nem biztos, hogy a jövőben is egy megfelelő útvonal lesz.

A proaktív protokollok legnagyobb kritikája, hogy sok sáv szélességet pazarolnak el arra, hogy az útvonal metrikákat széterjessék a hálózatban. A műholdas kommunikációban viszont, — mivel ritkán találkoznak a műholdak —, a táblázatcserét is csak ritkán kell elvégezni.

Annak érdekében, hogy az útválasztási algoritmus képes legyen egy jó döntést hozni afelől, hogy melyik egységnek küldje tovább a csomagot, ismerni kell, hogy a célállomás milyen messze van. Mivel, az általam vizsgált műholdas konstellációban az útválasztási algoritmus a legnagyobb mértékben a megérkezési időt befolyásolja, a távolság ebben a tanulmányban az időbeli távolságot jelenti. Ezzel szemben, ha az időt vesszük alapul, és elfogadjuk, hogy két műhold csak ritkán találkozik, akkor belátható, hogy a távolság két műhold között nem konstans, hanem erősen függ egyrészt attól, hogy milyen útvonalon kívánjuk továbbítani a csomagot, másrészt attól, hogy milyen lesz a műholdak jövőbeni helyzete. Ezért érdemes a műholdak kommunikációban egy proaktív útválasztási protokollt alkalmazni.

Ezen túlmenően, mivel a forrás egységnek ki kell számolnia az egész útvonalat, hogy tudja kinek kell a csomagot továbbítani, azt az információt továbbíthatja a csomaggal együtt (tehát „source routing”), annak érdekében, hogy a következő egységnek ne kelljen azzal foglalkozni. Az, hogy a forrás állomás számítja ki az útvonalat, sokat segít az útvonal hurokmentességének elérésében.

A műholdak ritkán találkoznak, tehát lehetséges, hogy arra sokat kell várni, hogy egy útvonalban a következő ugrásban szereplő műhoddal létre jöjjön a kapcsolat. Ezért, a SatCom útválasztási algoritmus, mikor egy csomagot megkap továbbításra, nem dobja el, mint sok hasonló algoritmusban történik, hanem addig tárolja, ameddig nem találkozik az adott műhold a következővel (ez a „store and forward” technika).

A műholdak, a kifejlesztett útvonal választási algoritmusban úgy tudják meg, hogy egy másik műhoddal találkoztak, hogy periodikusan küldenek egy „Helló” üzenetet azért, hogy minden közelben lévő műhold értesüljön arról, hogy közelében van egy másik műhold. Ekkor megkezdődik a találkozás információ cseréje, azaz a két találkozott műhold elküldi egymásnak a saját találkozás információs táblázatát és minden, a sajátjánál frissebb találkozási információt átír a saját táblázatában

Minden „Helló” üzenet után, egy bizonyos idő múlva egy „Helló 2” üzenet kerül elküldésre. Ennek az oka az a feltételezés, hogy minden üzenet vételekor a kommunikációt biztosító fizikai egység ki tudja számítani, hogy milyen messze van a küldő egység. A „Helló” és „Helló 2” üzenetek távolságából az egység ki tudja számolni a két műhold relatív sebességét, végül a kommunikációs távolság ismeretével kiszámítja, hogy a műholdak menyi ideig tudnak majd egymással kommunikálni:

$$\text{A kommunikáció idő tartalma} = \frac{\text{Kommunikációs távolság}}{\text{Heló táv} - \text{Heló 2 táv}} (\text{Heló, Heló 2 közötti idő})$$

4.3. A távolsági metrika

A vizsgálatok során három különböző távolsági metrikával számoltam. Az első, a dinamikus előrejelzés, megpróbálja pontosan előrejelezni, hogy két műhold

mikor fog következõleg találkozni és a második kettõ egy stacionárius előrejelzés. Ez utóbbiból az egyik kiszûri azt, hogy két egység milyen valószínűséggel találkozik nem több mint n másodperc múlva feltételezve, hogy a találkozás egy periódusán belül időben véletlenszerûen helyezkedik el. Az utolsó távolsági metrika egyszerűen az útvonalban az ugrások számát veszi figyelembe.

4.3.1. A dinamikus távolsági metrika

A dinamikus távolsági metrika számításában, minden egység veszi maximálisan a 40 legutóbbi találkozás intervallumot (τ_i) egy bizonyos egységgel és kiszámolja a normalizált autokorrelációs függvényét, 30 eltolással, azaz:

$$R_{\tau\tau}(j) = \frac{1}{10 \sum_{k=0}^9 \tau_k^2} \sum_{i=0}^9 \tau_i \tau_{i-j} ; j = 1, 2, 3, \dots, 30$$

Az algoritmus kiválasztja az $R_{\tau\tau}(j)$ eredmények közül a maximumot és a hozzá tartozó j_{\max} értéket. A j_{\max} értéket tekinti az algoritmus a periódus időnek, és az utolsó j_{\max} τ_i értéket, azaz $P(i) = \{\tau_{i-j_{\max}}, \tau_{j-j_{\max}} \dots, \tau_{i-1}\}$ tekinti az algoritmus a találkozások periódusának. A találkozás információhoz még az is hozzá tartozik, hogy a találkozás periódus mikor kezdődött (t_0).

Az algoritmus úgy jelezi előre a következõ találkozást, egy t idő után, hogy elõször kiszámítja a periódusban a fázist, azaz:

$$f = (t - t_0) \bmod \sum_{i=0}^{j_{\max}-1} P(i)$$

A következõ találkozás időbeni távolságot viszont a következõ algoritmus adja meg a „következõ_találkozás” változóban:

```

akumulatív_periódus_idő = 0;
i = 0;
míg(i < jmax) {
    akumulatív_periódus_idő += P(i);
    ha(f <= akumulatív_periódus_idő) {
        következõ_találkozás = akumulatív_periódus_idő - f;
        i = jmax;
    }
    i++;
}

```

Az útválasztási algoritmus ezen a ponton egy módosított Dijkstra algoritmust alkalmaz a legrövidebb útvonal megtalálására. A módosított Dijkstra algoritmus amiatt szükséges, mert az élek súlya (a továbbításra szükséges idő) nem konstans, hanem függ a kiinduló egység összesített súlyától (vagyis attól az időtől, hogy mikor ér oda a csomag). Az algoritmus még ilyen feltételek mellett is megtalálja a minimális súlyú útvonalat, mivel az egyes egységek összesített súlya, csak az elõzõ egységgel érvényes (azaz az útvonal), és mikor egy egységbõl továbblépünk, akkor annak a súlya már nem fog változni.

4.3.2. A stacionárius távolsági metrika

A stacionárius távolsági metrikában, maximálisan az utolsó 60 találkozási intervallum (τ_i) van figyelembe véve és a kommunikáló egység kiszámítja, hogy milyen valószínűséggel kerül egy csomag továbbítva 4000 másodpercen belül feltevé, hogy a csomagok érkezése egyenletesen oszlanak el az időben. Mivel a Dijkstra algoritmus egy minimum útvonalat keres, ezért a SatCom útválasztási algoritmus a valószínűség inverzét használja fel az élek súlyának. Hogyha a valószínűség nullának adódik, akkor a távolsági metrika 100-nak van véve.

Az általam elvégzett szimulációkból kiderült az is, hogy minél hosszabb a választott útvonal annál nagyobb a valószínűsége annak, hogy a csomag sokáig elakad mert két egység sokáig nem találkozik. Emiatt, érdemes a hosszú útvonalakat hátrányban részesíteni, azaz jobb eredményt lehet elérni hogyha a kiválasztott útvonal rövidebb még akkor is, ha rosszabb a jósági tényezője. Ezért, a stacionárius távolsági metrikával két vizsgálatot végeztem el, az egyikben a Dijkstra algoritmusban az egy egységnek a súlya a szokásos összeadásra alapul:

$$\text{Súly} = \sum \text{Az egységhez hozzávezető élek súlya}$$

és a másik vizsgálatban viszont a szorzásra:

$$\text{Súly} = \prod \text{Az egységhez hozzávezető élek súlya}$$

A szorzásnál az egységek távolsága az ugrások számának függvényében exponenciálisan növekszik, viszont az egyszerű összeadásban lineárisan.

4.3.3. A útvonal hossz távolsági metrika

Az útvonal hossz távolsági metrika abból áll, hogy minden élnek a súlya pontosan 1. Ez azt jelenti, hogy a választott útvonal minden a legkevesebb ugrásból fog állni. A probléma ezzel az útválasztási metrikával azonban az, hogy az összes lehetséges minimális hosszúságú útvonal közül nem biztos, hogy a legjobb kerül kiválasztásra, ami az eredményekből is látszik.

5. fejezet

Eredmények

A különböző metrikákkal rendelkező kifejlesztett SatCom útválasztási algoritmust az általam e célra kifejlesztett szimulátorban vizsgáltam meg több műholdas eloszlás esetén. A vizsgálat elsődleges célja az volt, hogy egy csomagnak átlagban hány másodperc szükséges, hogy célba érjen.

A szimulációk során három műhold konstellációval végeztem el a vizsgálatokat. Első sorban a 65 (elvben 66) műholdas Iridium és a 28 műholdas Orbcomm konstellációkat alkalmaztam a szimulációkban. E két konstellációban a műholdak keringési pályáit a tervező gondosan megtervezte, tehát ezekben a konstellációkban a műholdak pályái nem véletlenszerűek. A SatCom algoritmus hatékonyságát kipróbáltam a már meglévő 27 CubeSat műhold konstellációjában is. Erre az jellemző, hogy a műholdak lényegében véletlenszerűen oszlanak el. A műholdak pozíciójának meghatározása az sgdp4 programmal történt. A pályaadatok azonban a CelesTrak által nyilvánosságra hozott ún. „kétsoros” adatformátumban kerültek a szimulátor bemenetére.

Az elvégzett szimulációk során minden 300-adik idő egységben, minden egység véletlenszerűen választ egy számot (λ) a $[0, 1]$ intervallumban, és hogyha $\lambda < 0,8$ akkor az egység úgyszintén véletlenszerűen egyenletes eloszlással választ egy másik egységet aminek az útválasztási algoritmus segítségével küld egy csomagot.

Minden 14 időegységben, minden egység szórtad egy „Hello” üzenetet, hogy megismertesse az összes pillanatnyi szomszédjával, hogy kommunikációs távolságon belül van. Ugyanekkor minden szomszédjával kicseréli a találkozási táblázatokat.

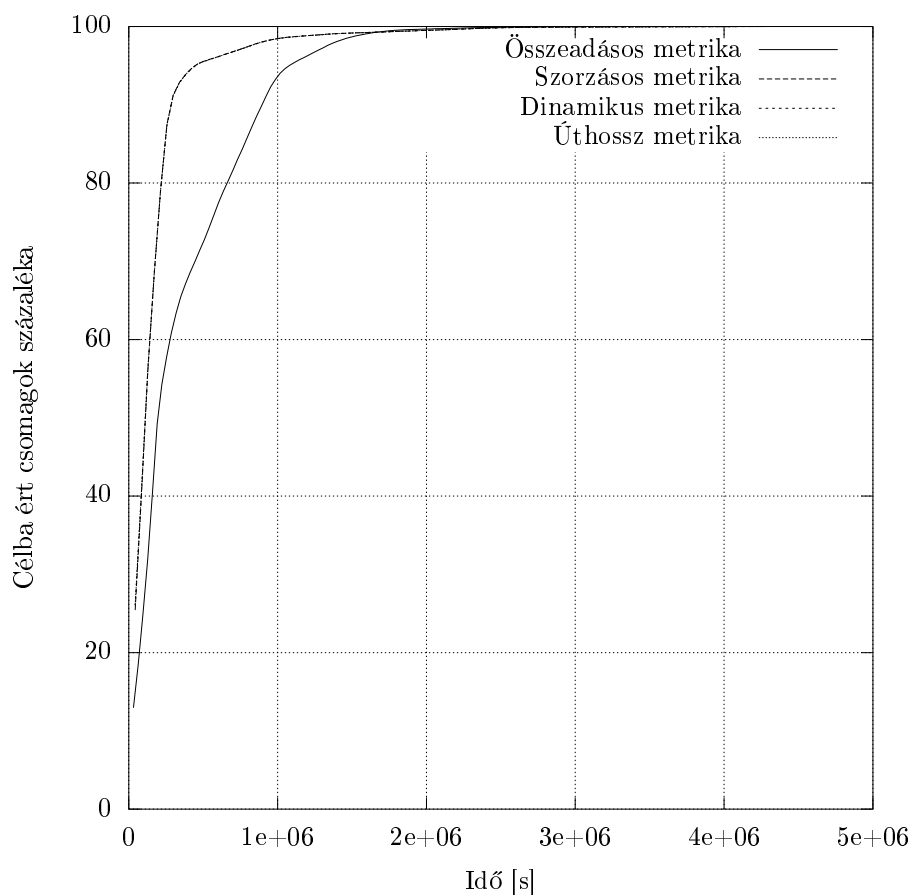
Minden szimuláció során az időegység egy másodperc volt, és az elvégzett szimuláció 5000000 másodpercet fogott át, ami körülbelül 58 napnak felel meg.

Az elvégzett első szimulációban azt vizsgáltam, hogy a három konstellációban mind a négyféle úthossz metrikával kiszámolva átlagban milyen hosszú ideig tart egy csomagot a célba juttatni. Az eredmények az 5.1. táblázatban láthatóak. Az 5.1. ábrán az van megadva, hogy a CubeSat konstellációban a négy útvonalhossz metrikával, hogyan oszlik el a csomag célba érkezési idejük.

A táblázatból jól látható, hogy nehezen lehet előrejelezni a műholdak pontos találkozásait. A legnagyobb probléma ezzel az útválasztási metrikával, hogy az előrejelezhető találkozási idő nagyságrendileg megegyezik azzal, hogy meddig tart az információt szét terjeszteni a hálózatban. Az 5.2. táblázatban meg van adva, hogy az útválasztási algoritmusban átlagban két műhold találkozása

| Útválasztó metrika | CubeSat | OrbComm | Iridium |
|------------------------------------|------------------------|-------------------------|-----------------------|
| Dinamikus útválasztási metrika | 161992s (44,98 óra) | 941780s (10,9 nap) | 2092s (34,87 nap) |
| Stacionárius metrika, összeadással | 355721s (4,12 nap) | 1052700s (12,18 nap) | 2097s (34,95 perc) |
| Stacionárius metrika, szorzással | 162384s (45,11 óra) | 943404s (10,92 nap) | 2091s (34,85 perc) |
| Útvonal hossz metrika | 161964s (44,99 óra) | 941668s (10,9 nap) | 2092s (34,87 perc) |

5.1. táblázat. A szerző által végzett szimuláció eredménye a kifejlesztett SatCom útválasztási algoritmus alkalmazásával, táblázat formában megadva.



5.1. ábra. A szerző által végzett szimuláció eredménye, abból a szempontból, hogy egy adott időn belül hány százaléka a csomagoknak ér célba a CubeSat konstellációban.

mennyi időre volt előrejelezve és az, hogy az előrejelzések átlagban milyen régi adatból születtek. Fontos azonban kihangsúlyozni, hogy a fentiekben közölt adatok az *összes* előrejelzést figyelembe véve kerültek megadásra és nem csak a végül kiválasztott útvonalra használt előrejelzést vettem számításba.

| Konstelláció | Előrejelzés idő | | Adatok öregsége | |
|--------------|-----------------|---------|-----------------|---------|
| | Átlag | Szórás | Átlag | Szórás |
| CubeSat | 230840s | 453010s | 229270s | 453030s |
| OrbComm | 785190s | 667060s | 783750s | 667000s |
| Iridium | 24292s | 141730s | 21596s | 141740s |

5.2. táblázat. Statisztikák arról, hogy egy előrejelzés átlagban mennyi időre történik a jövőbe és, hogy az előrejelzések milyen régi adatokon van elvégezve.

Az első eredményekből (az 5.1. táblázat) jól látható, hogy az Iridium konstelláció szorgos megtervezésének köszönhetően, a csomagok relatíve hamar célba érnek. Az OrbComm konstelláció viszont olyan tulajdonságokat mutat amik arra utalnak, hogy a tervezése közben nem volt figyelembe véve a műhold-műhold kommunikáció. A CubeSat konstellációban viszont a véletlenszerű műhold eloszlás ami rontja az útválasztási algoritmus hatékonyságát. Érdekesképpén megvizsgáltam, hogy a három konstellációban a stacionárius útválasztási algoritmusban hogyan oszlik el a találkozások mértékének az inverze (azaz, a vizsgálat a valószínűségeket nézte, nem a valószínűség inverzét). Az eredmények az 5.3. táblázatban láthatóak. A táblázatban közölt adatok jól rávilágítanak arra, hogy miért kell gondosan megtervezni egy műhold konstellációt (azaz a műhold flottában résztvevő műholdak egymáshoz képesti mozgását).

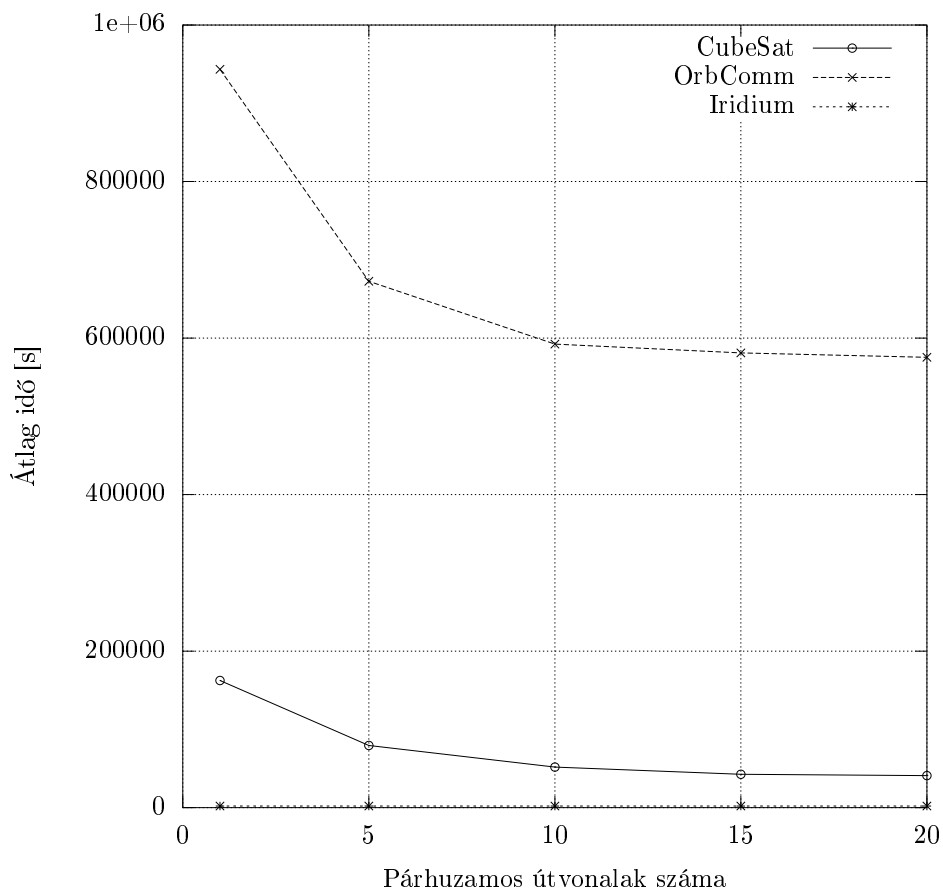
| Konstelláció | Átlag valószínűség | Szórás |
|--------------|--------------------|---------|
| CubeSat | 0,27444 | 0,36601 |
| OrbComm | 0,31698 | 0,37528 |
| Iridium | 0,99885 | 0,00621 |

5.3. táblázat. A különböző konstellációkban a találkozások jósági tényezők inverzének (azaz a rövid időn belüli találkozások valószínűségének) az átlaga és szórása

Mivel a stabil találkozás valószínűsége kicsi (az 5.3. táblázat) megvizsgáltam, hogy mi történik akkor, hogyha egy csomagot $n = \{1, 2, 3, \dots, 20\}$ párhuzamos útvonalon küldünk el. A párhuzamos útválasztás úgy történt, hogy a legjobbnak minősülő útvonal került kiszámításra és ezen az útvonalon a legrosszabb élt kitöröltem a gráfból. Ezután a második legjobb útvonal került kiszámításra és így tovább ameddig vagy el nem lett érve a maximális útvonalak száma, vagy már nincs útvonal a forrás és a célállomás között. Az eredmények több útvonalon elküldött csomagok esetén az 5.2. ábrán és az 5.4. táblázatban látható.

5.1. Értékelés

Az 5.1. táblázatból jól látható, hogy a Stacionárius szorzásos úthossz metrika teljesen összemérhető a Dinamikus útválasztási metrikával. A dinamikus útválasztási metrika sokkal több matematikai műveletet igényel mint bármelyik



5.2. ábra. Az átlag megérkezési idő annak a függvényében, hogy maximálisan hány párhuzamos útvonalon van egy csomag elküldve, a szorzásos útvonal metrikát alkalmazva.

| Max. párhuzamos útvonal | CubeSat | OrbComm | Iridium |
|-------------------------|------------------------|------------------------|-----------------------|
| 1 | 162384s (45,11 óra) | 943404s (10,92 nap) | 2091s (34,85 perc) |
| 5 | 79509s (22,09 óra) | 672552s (7,78 nap) | 1977s (32,95 perc) |
| 10 | 51761s (14,38 óra) | 592341s (6,86 nap) | 1973s (32,88 perc) |
| 15 | 42527s (11,81 óra) | 581011s (6,73 nap) | 1973s (32,88 perc) |
| 20 | 40984s (11,38 óra) | 575347s (6,66 nap) | 1973s (32,88 perc) |

5.4. táblázat. Az átlag megérkezési idő maximálisan 1, 5, 10, 15 és 20 párhuzamos útvonalon való elküldésekor

stacionárius útválaszási metrika, ami egyenesen azt jelenti, hogy a Dinamikus úthossz metrika számítása sokkal több energiát használ fel a műhold már szűkös készletéből, mint a stacionárius úthossz metrikával számolt útvonal. Vagyis, nem érdemes a Dinamikus úthossz metrikát alkalmazni a szorzásos stacionárius úthossz helyett.

Az különböző konstellációkra nyert eredmény jól rávilágítanak arra, hogy a műhold konstelláció felépítése milyen nagy mértékben befolyásolja az eredményeket. Az Iridium konstellációban, ahol tervezési szempont volt, hogy a műholdak kommunikáljanak egymással, ottan még a legegyszerűbb metrikával is nagyon jó eredményt lehet elérni, viszont az OrbComm konstelláció, ahol nem volt szempont a műholdak közti kommunikáció alig ha lehet megfelelő eredményt elérni.

A véletlen eloszlású CubeSat konstellációban az egy úton való csomag továbbítás egyik vizsgált útvonal hossz metrikával sem lehet úgy megoldani, hogy a csomag célba érési ideje jó valószínűséggel kézben tartható legyen. Csak a több úton továbbított csomagok érnek célba ész szerű időn belül, amivel viszont tűrhető eredményeket ért el ahhoz képest, hogy a konstellációban semmilyen tervezés nem történt.

6. fejezet

Összefoglaló

A tanulmány elején áttekintést adtam a már meglévő ad-hoc útvonal választási algoritmusokról és, aztán a legjobb tulajdonságait mindegyiknek kihasználva megterveztem a SatCom útvonal választás algoritmust. Az algoritmusban négy féle távolsági metrikát vizsgáltam: a dinamikus távolsági metrikát, a stacionárius távolsági metrika (összeadással és szorzással) és az egyszerű útvonal hossz távolsági metrikát.

Mind a négy metrikát három műhold konstelláción vizsgáltam, amelyek keringési pályái különböző tulajdonságokat mutatnak. Az eredményekből jól látszik, hogy a műhold konstelláció formája (a műholdak keringési pályái) nagy szerepet játszik. Viszont az útválasztó algoritmus amit kifejlesztettem a minimálisat tételezi fel a műholdak képességeiről, érdembeli eredményt ért el a több utas stacionárius (szorzással) úthossz metrikával a véletlenszerűen elosztó CubeSat konstelláción. A minimális feltételezésnek köszönhetően a CubeSat műholdak fedélzetén is alkalmazható.

Irodalomjegyzék

- [1] G.E. Prescott, S.A. Smith, K. Moe: Real-Time Information System Technology Challenges for NASA's Earth Science Enterprise, in Proceedings of The 20th IEEE Real-Time Systems Symposium, Phoenix Arizona, 1999.
- [2] L. Wood: Satellite Constellation Networks, 2002.
- [3] Z.J. Haas, J. Deng, B. Liang, P. Papadimitratos, S. Sajama: Wireless Ad Hoc Networks, 2002.
- [4] Z.J. Haas and J. Deng: Dual Busy Tone Multiple Access (DBTMA): A Multiple Access Control Scheme for Ad Hoc Networks," IEEE Transactions of Communications, 2002, pp. 975-985.
- [5] D. Johnson, Y. Hu, D.Maltz, The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, IETF RFC 4728, 2007.
- [6] C. Perkins, E. Belding-Royer, S.Das, Ad hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561, 2003.
- [7] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, A distance routing effect algorithm for mobility (DREAM), in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom), 1998, pp. 76-84.
- [8] C-C. Shen, G. Borkar, S. Rajagopalan, C. Jaikao: Interrogation-Based Relay Routing for Ad hoc Satellite Networks, IEEE Globecom, 2002, pp. 17-21
- [9] Z. Chang, G. Gaydadjiev: Cross-Layer Designs Architecture for LEO Satellite Ad Hoc Network. In: J. Harju et al. (Eds.): WWIC 2008, LNCS 5031, Springer-Verlag Berlin Heidelberg, 2008, pp. 164-176
- [10] A. Lindgren, A. Doria, O. Schelén. Probabilistic routing in intermittently connected networks, in Proceedings of SAPIR, 2004.
- [11] F.R. Hoots, R.L. Roehrich: Spacetrack report no. 3: Models for propagation of NORAD Element Sets, 1980.
- [12] D.A. Vallado, P. Crawford, R. Hujsak, T.S. Kelso: Revisiting Spacetrack Report #3, 2006.
- [13] Orbital mechanics with MATLAB: Relative Motion Between Two Earth Satellites.

- [14] <http://www.nsnam.org/>
- [15] http://www.sat.dundee.ac.uk/psc/dundee_sgp4.zip, letöltve 2011, Szeptember 11.
- [16] <http://www.celestrak.com/NORAD/elements/cubesat.txt>, letöltve 2011 október 4.
- [17] <http://www.celestrak.com/NORAD/elements/orbcomm.txt>, letöltve 2011 október 25.
- [18] <http://www.celestrak.com/NORAD/elements/iridium.txt>, letöltve 2011 október 22.
- [19] IEEE Standard for a Precision Clock Synchronization Protocol for networked Measurements and Control Systems, IEEE Std 1588, 2008.
- [20] D. Mills, J. Martin, J. Burbank, W. Kasch, Network Time Protocol Version 4: Protocol and Algorithms Specification, IETF RFC 5905, 2010.