

Útvonalválasztás késleltetés-toleráns hálózatokban

BABARCI PÉTER, TANAI FERENC, CSIKOR LEVENTE,
TAPOLCAI JÁNOS, HESZBERGER ZALÁN

Nagysebességű Hálózatok Laboratórium (HSNLab)

BME Távközlési és Médiainformatikai Tanszék

{babarci, tanai, csikor, tapolcai, heszi}@tmit.bme.hu

Lektorált

Kulcsszavak: késleltetés-toleráns hálózatok, útvonalválasztás, ad hoc járműhálózatok, időszakos összefüggőség

A késleltetés-toleráns hálózatok (DTN) olyan hálózatok, amelyekben az idő jelentős részében nem áll rendelkezésre végponttól végpontig terjedő útvonal a forrás és a célcsofópont között. Ilyen hálózatok körébe tartoznak többek között egyes szenzorhálózatok, mobil járműhálózatok, úrkommunikációs rendszerek vagy akár a katonai célra alkalmazott hálózatok.

A DTN környezet kétségtelenül legfontosabb kihívása az útvonalválasztás, hiszen a hagyományos technikák jellemzően stabil, folyamatosan rendelkezésre álló végponttól végpontig terjedő útvonalak keresésére alkalmasak.

Cikkünkben áttekintést nyújtunk a környezet legnagyobb kihívásairól és az útvonalválasztó módszerek főbb kategóriáiról, megkülönböztetve az egyetlen üzenetmásolatot alkalmazó, illetve az elárasztás jelleggel működő eljárásokat.

A különböző kategóriák néhány alapvető algoritmusának részletes ismertetése után áttekintésünk végén szimuláció segítségével hasonlítjuk össze a bemutatott módszerek teljesítményét nagyvárosi környezetben.

1. Bevezetés

Napjainkban az internet térhódításával a hálózati forgalom jelentős része IP-alapon továbbítódik a végpontok között. A hálózatok túlnyomó része az IP-alapú hálózati réteg felett TCP-t alkalmazva a szállítási rétegben megfelelő szolgáltatásminőségi szintet képes nyújtani az alkalmazások számára, ezzel biztosítva a felhasználók elégedettségét. Mindez ugyanakkor az alsóbb hálózati rétegekkel szemben implicit módon számos követelményt támaszt. A három legfontosabb elvárás az egyes linkektől, vagy akár a teljes hálózattól, hogy minden pillanatban létezzon végponttól végpontig terjedő útvonal az adat forrása és célcsofópontjai között, továbbá a maximális körülfordulási idő ne nőjön egy bizonyos határ fölé, illetve a végpontok közötti csomagvesztési valószínűség alacsony legyen [15].

Abban az esetben, ha ezen tulajdonságok közül bármelyik is sérül, nehézségekbe ütközünk a hagyományos protokollok alkalmazását illetően. Például végponttól végpontig terjedő útvonal hiányában a hálózatnak a *tárolás és továbbítás elvén* kell működnie, azaz a résztvevő csomópontoknak (vagy a forrásnak) akár órákra is el kell tárolnia egy csomagot, mielőtt továbbítani tudná azt. A jelenleg alkalmazott útvonalválasztók ugyan fel vannak készítve különböző adatkapcsolati rétegbeli protokollokkal való együttműködésre, de a csomagok huzamosabb tárolására nem alkalmasak az egyes linkek kiesése esetén. Hasonló a helyzet egy magas körülfordulási idővel rendelkező összeköttetés esetén, ahol a TCP-kapcsolatok rendre megszakadnának, valamint a magas csomagvesztés is érzékelhető minőségromlást eredményezne a felhasználók számára. Az olyan hálózatokra, ahol a három legfontosabb alapelv közül bármelyik sérül, összefoglaló néven késleltetés- vagy meg-

szakítás-toleráns hálózatokként (Delay or Disruption Tolerant Networks – DTN [9]) hivatkoznak.

Amint az előbbi példákban is láthattuk, DTN környezetben hagyományos protokollok alkalmazásával a minimális szolgáltatásminőségi szint nyújtása is kérdéses (pl. TCP, DNS [9]), ezért módosított hálózati architektúrák kidolgozására van szükség. Az igényelt szolgáltatásminőség, alkalmazási területtől függően vonatkozhat minimális időzítési követelmények betartására, vagy extrém esetben egyáltalán az adat célba juttatására. Az egyes hálózati rétegek funkcióinak késleltetés-toleráns környezetben való megvalósítására megoldások és ajánlások széles skálája létezik.

Az egyik legérdekesebb javaslat a fizikai rétegben postagalambok alkalmazása. Az IETF hagyományosan minden április 1-én tréfás RFC-et tesz közzé, ezek közé tartozott az 2549-es is, amely IP-csomagok küldését specifikálta postagalambokon. A módszert később implementálták is, melynek során kinyomtatott IP-csomagokat csatoltak a galambokra. A mért adatok alapján a hálózat körülfordulási ideje 3 óra volt, így megállapítást nyert, hogy TCP-kapcsolatok továbbítására nem alkalmas. Az adatkapcsolati rétegben más környezetből származó módszerek adaptálhatóak, például törlő kódolás (erasure coding [11]) vagy hálózati kódolás (network coding [17]), melyek alkalmazási területtől függően jelentősen javíthatnak a DTN-hálózatok teljesítményén.

Felsőbb rétegekben is találkozhatunk a DTN-hálózatok „tárolj és továbbíts” alapelvevel. Erre példa a hálózati rétegben a FIDONET [16], amely több aspektusból is hasonlít egy DTN-hálózatra és teljesíti a DTN-alkalmazások minimális megbízhatósági követelményeit. A hálózatot számítógépek közötti telefonos adatátvitelre tervezték az Amerikai Egyesült Államokban, ahol a helyi hívások díjazása nem percalapú volt, míg a távolsági

hívásoké igen. Így helyi hívás kezdeményezése napszaktól függetlenül megérte, míg a távolsági adathívásokat olcsóbb volt éjszaka bonyolítani. Ezért az egyes igényeket a két gép közötti közvetlen hívás helyett egy helyi központban gyűjtötték össze, és éjszaka továbbították őket a távolabbi célpontok felé. A bolygóközi kommunikációt biztosító DTN-típusú InterPlaNetary Internet szállítási rétegében a szigorú időzítést igénylő TCP helyett a TP-Planet [12] szállítási protokollt alkalmazzák. A TP-Planet egyszerűen a hagyományos TCP-protokollt terjeszti ki és teszi alkalmassá magas késleltetésű linkeken történő megbízható adattovábbításra.

A hagyományos alkalmazási rétegbeli protokollok leg többje megfelelő változtatásokkal könnyen illeszthető DTN-környezetbe. Számos olyan alkalmazási példa említhető, ahol a rendelkezésre álló hálózati infrastruktúra a fenti kihívásokkal rendelkezik. Szenzorhálózatok széles köre tartozhat a DTN-hálózatok közé, ahol mobil csomópontok is részt vesznek a mért adatok központba juttatásában. Ilyen alkalmazás például egy vízmonitorozó hálózat, ahol a víz minőségét bálnákra szerelt szenzorok mérik, majd amikor a bálna visszatér a táplálkozó helyére, átadja az adatokat a bázisnak. A magas körülfordulási idő miatt a víz alatti hanghullámokkal való kommunikációt használó hálózatok is a késleltetés toleráns hálózatok körébe sorolhatóak. További alkalmazási példa lehet egy vadállatok élőhelyének monitorozására telepített szenzorhálózat (pl. ZebraNet).

A DTN-koncepciót követő alkalmazások közül az egyik legfontosabb példa az interneteléréssel nem rendelkező távoli területek számára biztosítani a világhálóhoz való hozzáférést DTN-hálózat segítségével, ahol más technológia kiépítése költséges lenne. Ez történhet állatokra erősített adóvevők vagy motorkerékpárok segítségével, melyek folyamatosan ingáznak az Internet összeköttetéssel rendelkező nagyváros és a települések között, lehetővé téve az off-line böngészést. A késleltetés-toleráns hálózatok körébe sorolhatóak azok a közlekedési járművekkel megvalósított mobil hálózatok is (VANET, Vehicular Ad hoc NETWORKS), amelyekben időszakosan előfordulhat, hogy nem áll rendelkezésre útvonal a célhoz. VANET-hálózatok használhatóak státuszjelzések küldésére az út minőségét, a szabad parkolóhelyeket, vagy a dugókat illetően broadcast jelleggel, vagy akár videóteka szolgáltatás megvalósítására is alkalmazhatóak pont-pont kommunikációval.

Láthattuk, hogy hálózatok széles köre létezik, melyekben nincs folyamatosan végponttól végpontig terjedő útvonal például a csomópontok mozgása vagy az energiatakarékos működés miatt, illetve a tradicionális protokollok által nem tolerálható késleltetés tapasztalható a távolság vagy az átviteli közeg jellemzői miatt. Ilyen tulajdonságokkal rendelkező környezetben az útvonalválasztás jelenti a legnagyobb kihívást. Jogosan vetődik fel a kérdés: „Melyik csomópontnak adjuk tovább a csomagot (vagy DTN-terminológiával: üzenetet), ha még nincs is útvonal, sőt, még a cél helyzetéről rendelkezésre álló információ is bizonytalan?”. Ilyen környezetben a hagyományos útvonalválasztó algoritmusok (pl. RIP, OSPF),

illetve a mobil ad hoc környezetben alkalmazott módszerek (pl. DSR, AODV [4]) gyakran nem alkalmazhatóak, mivel sem a végponttól végpontig terjedő útvonal létezése, sem a célcsoomópont helyzetének vagy akár címének (pontos) ismerete nem garantálható. A hagyományos algoritmusok viszont éppen azon alapulnak, hogy a célig vezető lehetséges útvonalakat felderítsék, mielőtt bármilyen hasznos adatot küldenék a hálózatba.

Cikkünk a DTN útvonalválasztó algoritmusok alapvető tulajdonságairól és teljesítményéről próbál átfogó képet nyújtani. A következő szakaszban ismertetjük azokat a teljesítménymérőket és hálózati sajátosságokat, amelyek befolyásolják az alkalmazási környezethez legjobban illeszkedő algoritmus kiválasztását. Az ismertetett tulajdonságok alapján a 3. szakaszban különböző megközelítéseket alkalmazó útvonalválasztó algoritmusokat mutatunk be. Végül a 4. szakaszban nagyvárosi tömegközlekedést szimuláló környezetben hasonlítjuk össze a bemutatott algoritmusok teljesítményét a korábban bevezetett fontosabb teljesítménymutatók mentén.

2. DTN-hálózatok jellemzői

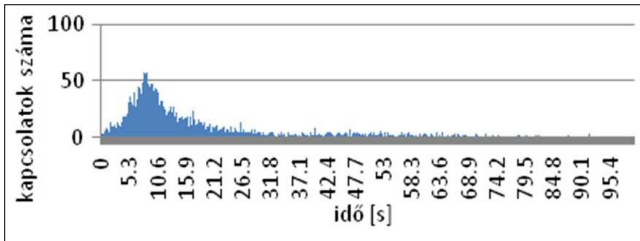
2.1. Az alkalmazási környezet tulajdonságai

A DTN-hálózatok tulajdonságai igen széles spektrumot ölelnek fel, gondoljunk csak a csomópontok sebességének hatására az átviteli paraméterekre, vagy a csomópontok számára rendelkezésre álló energiára (például szenzorhálózatok és járműhálózatok esetén). Ebben a szakaszban a DTN útvonalválasztó algoritmusok teljesítményének szempontjából lényeges három paramétert vizsgáljuk meg: a találkozások időpontjának ismeretét, a létrejövő összeköttetés sávszélességét, illetve a továbbítandó üzenetek tárolására szolgáló tároló méretét.

A *találkozások időpontjának ismerete* a konkrét DTN-hálózattól és alkalmazástól függ. Műholdakkal való kommunikáció során precíz információnk van arról, hogy a műhold mikor kerül a horizont fölé és mikor bukik le. Egy ad hoc hálózatban ezzel szemben semmilyen információnk nincs a csomópont jövőbeni helyzetéről és a jövőbeni összeköttetés paramétereiről. A két véglet közötti átmenet, ha valamilyen (pontatlan) információnk van a csomópontok helyzetéről vagy mozgásáról. Ilyen információ lehet például a menetrend egy tömegközlekedési hálózatban, amelytől a forgalom függvényében a találkozás időpontja eltérhet. A csomópontok mozgásáról ad hoc hálózatban is tudunk információt gyűjteni, például ha léteznek olyan régiók, ahol a csomópontok gyakrabban tartózkodnak, mint máshol (például állatok esetén az etető hely). Ezekben az esetekben statisztikai módszerekkel becsülni tudjuk a csomópontok jövőbeni helyét és az így nyert információt fel tudjuk használni az útvonalválasztási döntés meghozatalakor.

Az *összeköttetések sávszélessége* a maximálisan átvihető adatmennyiségen túl azt is meghatározza, hogy mekkora üzenetméretet érdemes alkalmazni. Túl kicsi üzenetek esetén nagy lesz az overhead, nagy üzenetek alacsony sávszélességű, vagy rövid ideig élő vezeték

nélküli összeköttetésen való továbbítása viszont csak többszöri próbálkozásra sikerülhet. A sikertelen próbálkozások ideje alatt viszont nem küldtünk hasznos adatot, ezzel is rontva a hálózat teljesítőképességét. A 4.1. szakaszban a videotéka alkalmazáshoz megadott átviteli jellemzőkkel az összeköttetés időtartamára azt kapjuk, hogy nagyvárosi környezetben két jármű várhatóan 8 másodpercig lesz összeköttetésben, mely idő alatt az adatcserére lehetőség van (1. ábra).



1. ábra

Járműhálózatban létrejövő összeköttetések hossza

A tárolóméret és az alkalmazott tárolóstratégia nagyban befolyásolja, hogy mennyi másolatot érdemes megengedni egyetlen üzenetből. Végtelen tároló alkalmazása esetén azok a módszerek lesznek előnyben, melyek minél több (potenciálisan az összes) útvonalon terjesztik az üzenetet, hiszen ekkor a sok útvonal között nagy valószínűséggel megtalálható az optimális (például leg-rövidebb, legkisebb késleltetésű stb.) is. Kis tároló esetén a sok üzenet nem kifizetődő, mivel a limitált hely miatt gyakran kell felülírunk olyan üzeneteket, melyek még nem értek célba, csökkentve ezzel a módszer hatékonyságát. A tárolóban a felülírandó üzenetek kiválasztásához alkalmazhatjuk például a megtett ugrásszámot, a hátralévő időt, vagy más valószínűségi alapú metrikát [18]. A tárolóban alkalmazott stratégia hatékonyságának javítása érdekében a már célba érkezett üzenetek törlésére nyugták is terjeszthetők a hálózatban.

2.2. Útvonalválasztási döntés meghozatalának időpontja

A hagyományos útvonalválasztó módszereknél alkalmazott két megközelítés, a forrás által vezérelt, illetve az ugrásonkénti útvonalválasztás (source routing, illetve hop-by-hop routing) DTN-környezetben is felmerül, mint lehetőség. A végponttól végpontig terjedő útvonal ismerete nélkül csak akkor tudunk hatékonyan *forrás által vezérelt útvonalválasztást* alkalmazni, ha a hálózatról kellően pontos információink vannak. Ez előfordulhat, ha van valamilyen orákulumunk, aki megmondja a pontos időzítéseket, a sorbaállási időket, jövőben érkező igényeket [10].

A gyakorlatban DTN-hálózatokban a lehető legkésőbbre kell halasztanunk az útvonalválasztási döntést. Az *ugrásonkénti útvonalválasztás* már közelebb visz ehhez a célhoz, de még mindig nem elég hatékony, mert már az üzenet beérkezésekor eldönti a következő csomópontot. Azaz még jobb, ha akkor döntünk az üzenet sorsáról, hogy átadjuk-e, vagy sem, amikor egy új összeköttetés létrejön, azaz *összeköttetésenkénti útvonalválasztást* alkalmazunk.

2.3. Algoritmusok hatékonyságának összehasonlítása

Mivel DTN-környezetben egy adott üzenet célbaérkezése nem garantált, az útvonalválasztó algoritmusok értékeléséhez meg kell határoznunk, hogy egy adott időpontig a keletkezett üzenetek hány százaléka érkezett célba. Ezt a hányadost *célbaérkezési aránynak* nevezük, és a DTN útvonalválasztó algoritmusok teljesítményének legfontosabb mérőszáma. Alkalmazástól függően akár már a küldött üzenetek felének célbaérkezése is jó teljesítménynek tekinthető, míg egy tömegközlekedési eszközökön megvalósított videotéka-szolgáltatás esetén akár egyetlen üzenet elvesztése is problémát jelent. Valamennyi üzenet célbaérkezését megkövetelő alkalmazások esetén az algoritmusok teljesítményéről a teljes megérkezésig eltelt *késleltetés* ad információt. Egy városi környezetben működő tartalomszolgáltatásnál kiemelt szerepet játszhat, ha a szolgáltató garantálja, hogy a terméke adott idő alatt megérkezik a felhasználóhoz. Ekkor olyan útvonalválasztási módszert célszerű alkalmazni, mely a megrendelés és a határidő közötti időtartamnál kisebb késleltetéssel képes a filmet célba juttatni.

Az útvonalválasztó algoritmusok teljesítményének vizsgálatakor nem mellékes információ a hálózatban keletkezett *üzenetmásolatok száma*, amely azt adja meg, hogy egyetlen üzenet hány különböző csomópontnál található egy adott pillanatban (végtelen tároló méretet feltételezve). A csomópontok véges erőforrásokkal rendelkeznek és az üzenetek tárolására is véges kapacitás áll rendelkezésre, így a sok üzenetmásolatot használó algoritmusok rosszabbul teljesíthetnek például kis tároló kapacitás esetén. Az üzenetmásolatok csökkentésének céljából alkalmazható az üzenetekben a *maximális ugrásszám* (vagy *élettartam*) paraméter. Ha az üzenet áthalad a paraméterben megadott csomópontszámon (vagy lejár az élettartama), akkor nem továbbítódik a következő átadáskor, és a tárolóban alkalmazott stratégiától függően vagy azonnal töröljük, vagy csak szükség esetén, szabad erőforrás hiányában írjuk majd felül.

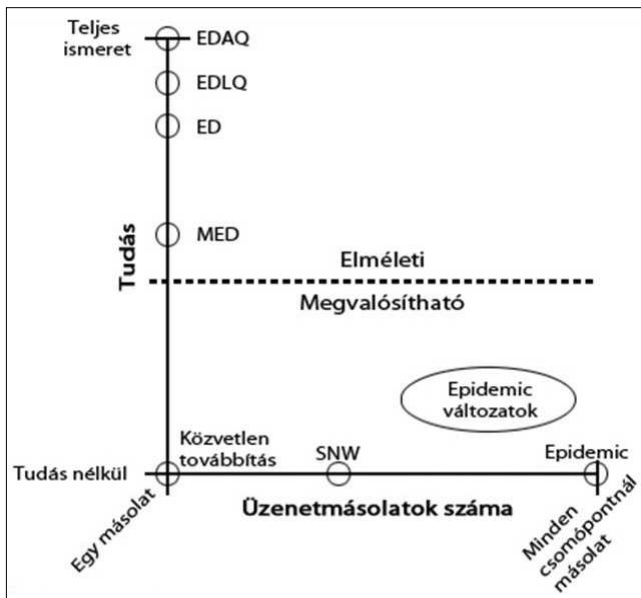
Általánosan elmondható, hogy az adott alkalmazási környezethez a célunk olyan útvonalválasztó stratégiát választani, amely maximalizálja a célbaérkezési arányt, tolerálható késleltetést biztosít és mindezek mellett a lehető legkevesebb üzenetmásolatot használja.

3. Útvonalválasztás

Az előző szakaszban láthattuk, hogy a DTN hálózatokban alkalmazott útvonalválasztó módszerekkel szemben támasztott legfontosabb követelmény, hogy maximalizálják a célbaérkezési arányt, figyelembe véve a csomópontoknál rendelkezésre álló erőforrásokat és az útvonalválasztási döntés meghozatalához rendelkezésre álló információ mennyiségét. Ebben a szakaszban rátérünk konkrét DTN útvonalválasztó módszerek ismertetésére.

A jelenleg alkalmazott útvonalválasztó algoritmusokat alapvetően két paraméterrel jellemezhetjük: a döntés meghozatalához felhasznált tudás, illetve az üzenet-

másolatok száma. A szakirodalom által feltérképezett megoldások ezek közül többnyire tisztán csak egyik vagy másik paramétert használják (2. ábra), egyedül a [6]-ban ismertetett néhány módszer és az epidemic változatok azok, amelyek törekvést mutatnak arra, hogy mind a több üzenetmásolat, mind a tudás előnyeit ötvözzék a magasabb hatékonyság elérése érdekében. A következő szakaszokban ismertetjük a tudás és másolatszám által kifeszített sík egymástól távoli pontjain található módszerek lényegesen eltérő megközelítéseit.



2. ábra DTN útvonalválasztó algoritmusok az üzenetmásolatok száma és a döntés meghozatalához felhasznált tudás függvényében [20]

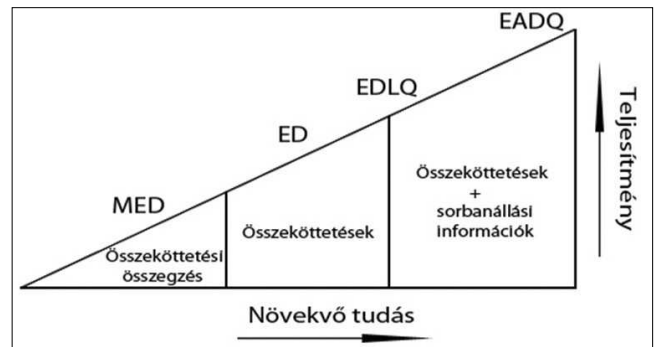
3.1. Egyetlen másolatot használó módszerek

A legegyszerűbb módszer a *közvetlen továbbítás*, azaz ha a forrás, vagy egy közbeiktatott „adathordozó öszvér” (Mobile Ubiquitous LAN Extensions, MULE) elviszi az üzenetet a célcsoóponthoz. Egy öszvér képes rövid távú vezeték nélküli összeköttetés létesítésére és adatok cseréjére más csomópontokkal (szenzorokkal, hozzáférési ponttal stb.), ha a közelükbe ér. Az alkalmazási területtől függően az adathordozó öszvér szerepét autók, buszok, de természetes élőhelyek monitorozás esetén ténylegesen állatok is betölthetik. Ilyen alkalmazás lehet például olyan szenzorhálózat, ahol a hálózatban mozgó öszvérek a szenzoroktól begyűjtik az adatot és elszállítják a gyűjtőpontba, ahol az adat feldolgozásra kerül [13]. A közvetlen továbbítás módszere az üzenet-továbbítások számában optimális, hiszen egyetlen (közbeiktatott öszvérrel két) átadás történik üzenetenként, a késleltetés viszont magas lehet a forrás vagy az öszvér mozgásától függően.

A megfelelő útvonalválasztási stratégia kiválasztása szempontjából érdemes lehet megvizsgálni, hogy milyen teljesítményre lenne képes a módszer, ha ismernénk a DTN hálózat bizonyos paramétereit, úgymint a létrejövő összeköttetések idejét és hosszát, a sorok állapotát, illetve a jövőben érkező igények paramétereit. Ezen in-

formációk nem nyerhetők ki a hálózatból, így csak elméleti szinten állnak rendelkezésre, úgynevezett orákulumok által. A feltételezett tudás függvényében több különböző orákulumot definiáltak [10], melyek a 3. ábrán láthatóak.

Az orákulumoktól szerzett részleges ismeret alapján az útvonalválasztási algoritmusok a rendelkezésre álló tudás függvényében megfelelően súlyozzák a gráfban az éleket és az így nyert segédgráfban egy *legrövidebb utat* (shortest path) számolnak.



3. ábra Részleges ismeretet felhasználó orákulum alapú technikák

A 3. ábrán látható, hogyan változik az algoritmusok teljesítménye a több tudás felhasználásán alapuló döntések esetén. A *legkisebb várható késleltetés* (Minimum Expected Delay, MED) algoritmus egy él súlyát az átlagos várakozási idő, jelterjedési késleltetés és az átviteli késleltetés összegeként értelmezi. A *legkorábbi célbaérkezés* (Earliest Delivery, ED) módszer jellemzője, hogy nem használ sorbanállási információkat, csak az összeköttetési orákulumtól kapott tudást veszi figyelembe a gráf súlyozásakor. Ebben az esetben forrás által vezérelt útvonalválasztásról beszélünk, ahol nem befolyásolja az útvonalak kiválasztását az eddigi üzenetek léte a rendszerben, ezáltal nem tudja elkerülni a torlódást.

A nevéből adódóan a *legkorábbi célbaérkezés helyi sorállási információk felhasználásával* (Earliest Delivery with Local Queuing, EDLQ) módszer az előző algoritmus-hoz képest figyelembe veszi a sorállási információkat is, de csak az adott csomópont esetén, ahol éppen az üzenet tartózkodik. Ennek a megközelítésnek komoly hátránya, hogy hurok alakulhat ki, és ezáltal az üzenetek örökké oszcillálhatnak két csomópont között. A *legkorábbi célbaérkezés globális sorállási információk felhasználásával* (Earliest Delivery with All Queues, EDAQ) a teljes topológia sorállási információit felhasználja az útvonalak kiszámításához. Ezen felül, ha a jövőben érkező igények pontos paramétereit is ismerjük, egy lineáris programozási (LP) feladat segítségével meghatározhatjuk az optimális útvonalat az egyetlen másolat számára.

3.2. Több üzenetmásolatot alkalmazó módszerek

3.2.1. Epidemic – elárasztás

Az előbbieken olyan megoldásokat ismertettünk, ahol az útválasztási döntések meghozatalához segítségünkre volt egy orákulum, ami rendelkezésünkre bocsá-

tott részleges vagy teljes információt a hálózatról. Az *epidemic routing* („járványszerű” elárasztás) [8] – ahogy a neve is sugallja – lényegesen eltérő megközelítést alkalmaz és hálózatról felhasznált tudás helyett az üzenet-másolatok számának növelésével próbál magasabb célbaérkezési arányt elérni a közvetlen továbbításnál (lásd a 2. ábrát).

Annak érdekében, hogy ne tisztán elárasztásról beszéljünk, hanem egy csomópont egy üzenetből legfeljebb egy másolatot tároljon, az eredeti epidemic algoritmusra [8] a következő megvalósítást javasolták. Minden csomópont tárol egy bitvektort (*summary vector*), ami mutatja, hogy milyen üzenetek találhatók a csomópontnál. Ha két csomópont találkozik, kicserélik a *summary vectorokat*, hogy megállapítsák, melyik üzenetből nem rendelkeznek másolattal, majd a második lépésként elkérnek egy másolatot az általuk nem birtokolt üzenetekből.

A cél a többi módszerhez hasonlóan maximalizálni a célbaérkezési arányt és minimalizálni a késleltetést azáltal, hogy a harmadik teljesítménymérőt, az üzenetek számát feláldozzuk (lásd 2.3. szakasz). Az epidemic útvonalválasztás több alkalmazási környezetben optimális megoldást biztosít. Végtelen sávszélességű összeköttetések és végtelen tároló alkalmazása esetén az epidemic minden üzenetet célba juttat, ami egyáltalán célba juttatható bármely útvonalválasztási módszer alkalmazásával. Ezen felül minimális késleltetéssel is rendelkezik, hiszen az összes útvonalon terjeszt egy üzenet-másolatot, közöttük a legkisebb késleltetésű útvonalon is. Azonban valós környezetben nem áll rendelkezésre végtelen sávszélesség és tárolókapacitás. Véges tároló esetén az üzenetek felülírásával, véges sávszélesség esetén pedig egy legrövidebb úton haladó üzenet átadásának meghiúsulásával az epidemic algoritmus teljesítménye jelentősen romolhat. Ugyanakkor még szűkös erőforrások esetén is érdemes alkalmazni olyan hálózatokban, amelyekben kicsi a csomópontok mobilitása, és a közvetlen továbbítás módszerét alkalmazva a forrás ritkán (vagy soha) nem találkozik a célcsoomóponttal.

Az egyetlen üzenetmásolatot alkalmazó módszerekhez képest megnövekedett erőforrás-használat miatt korlátozni kell a keletkező üzenetmásolatok számát, például az ugrásszám maximalizálásával, vagy valószínűségi alapú metrikák bevezetésével, ahogy ezt a következő szakaszban bemutatott epidemic változatok teszik.

3.2.2. Másolatok számának statisztika-alapú korlátozása

A *találkozási valószínűségekkel kontrollált elárasztás* [3] (Probabilistic ROuting Protocol using History of Encounters and Transitivity, PROPHET) algoritmus a hálózatról kinyert információ alapján csökkenti az üzenet-másolatok számát az epidemic módszerhez képest, és egy csomópont csak azoknak a csomópontoknak ad üzenet-másolatot, akiket az üzenet „jó” hordozójának ítél. Az órakulom alapú módszerekkel ellentétben a módszer a hálózatról kinyerhető információval dolgozik, és a múltbeli találkozásokról próbál következtetést levonni a hálózat jövőbeli állapotáról. Érezhetően a PROPHET

olyan környezetben fog hatékonyan teljesíteni, ahol szabályszerűség figyelhető meg a csomópontok mozgásában, míg teljesen véletlenszerű mozgás esetén a jóslás hasznavethetetlen.

A módszer eredeti változata [3] a következőképpen működik: A múltbeli találkozások alapján minden a csomópont kiszámol minden más b csomópontra egy úgynevezett *kézbizítési előreláthatóságot* (delivery predictability). A protokollt tekintve hasonlít az epidemic routingra olyan értelemben, hogy ha két csomópont találkozik, akkor kicserélik egymás közt a *summary vectorukat*, ami ebben az esetben az előbb említett valószínűségi értékeket is tartalmazza, ezáltal ezen információk is frissülnek a rendszerben. A kézbizítési előreláthatóság számítása során a The ONE szimulátorban ajánlott implementáció szerint [2] nulla kezdő értékek állnak rendelkezésre az egyes csomópontokhoz. Egy adott csomópont-ra vonatkozóan az első találkozáskor lesz nullától különböző kézbizítési előreláthatóság, majd a továbbiakban az (1)-(3) egyenletekben megadott szabály alapján változik ez az érték. Amikor két csomópont találkozik, akkor frissítik kézbizítési előreláthatóság értéküket az alábbiaknak megfelelően:

$$P_{(a,b)} = P_{(a,b)\text{régi}} + (1 - P_{(a,b)\text{régi}}) \cdot P_{\text{init}} \quad (1)$$

A múltat figyelembe véve, ha két csomópont már rég nem volt egymás hatósugarában, akkor a kézbizítési előreláthatóság öregszik egy úgynevezett öregedési konstanssal:

$$P_{(a,b)} = P_{(a,b)\text{régi}} \cdot \gamma^k, \quad (2)$$

ahol k az eltelt időegységek száma. A kézbizítési előreláthatóságnak van még egy tranzitív tulajdonsága is, ami azon a megállapításon alapul, hogy ha a pont gyakran találkozik b -vel és b gyakran találkozik c -vel, akkor c a metrika alapján jó továbbító pont azon üzeneteknek, amiknek a a célcsoomópontja, azaz

$$P_{(a,c)} = P_{(a,c)\text{régi}} + (1 - P_{(a,c)\text{régi}}) \cdot P_{(a,b)} \cdot P_{(b,c)} \cdot \beta. \quad (3)$$

Az (1)-(3) egyenletekben a P_{init} , γ , β , az alkalmazási környezetnek megfelelően választott konstans.

A frissített kézbizítési előreláthatóság értékek alapján akkor történik meg egy üzenetmásolat átadása, ha az üzenetmásolatot kérő csomópontnak magasabb a kézbizítési előreláthatóság értéke a célcsoomópont-ra vonatkozóan, mint az üzenet jelenlegi hordozójának.

A SARP [1] (*Self-Adaptive Routing Protocol*) fő célja a PROPHET-hez hasonlóan, hogy adaptálódjon a hálózatban a csomópontok viselkedéséhez, csökkentve ezzel a szükséges átvitelek számát és a késleltetést. Az eddigi technikák [3,14] a kézbizítési előreláthatóságot alapvetően a találkozások száma alapján számították. Ennek két problémája is jelentkezett. Az egyik a *hibásan detektált találkozások*, mely esetén, ha egy d csomópont egy c csomópont hatósugarában van, akkor d -t többször hibásan detektálhatja, ha d időközben ki- és bekapcsolt. Hasonló eset fordul elő akkor is, amikor d a c csomópont hatósugarának a határán mozog. A másik probléma az úgynevezett *hosszú távú szomszédság*, amikor két szomszédos

csomópont ugyanakkora sebességgel ugyanabba az irányba mozog. A problémák kiküszöbölésére a SARP bevezeti a *találkozás-közi időt*, mint további paramétert. A találkozás-közi idő két csomópont egymás hatásában való tartózkodásának időtartamára és találkozásuk között eltelt idejére alapszik. Ez a megközelítés nagyon hatékony lehet egy városi környezetben, ahol a csomópontok mozgása nagymértékben szabályozott az emberek mindennapi teendőjük végzésének, valamint a földrajzi és utazási korlátoknak köszönhetően.

3.2.3. Spray and wait/focus –

kétfázisú módszer az üzenetek szétterítésére

A Spray and Focus (SNF) és Spray and Wait (SNW) [6,7] algoritmusok célja, hogy jelentősen csökkentsék az elárasztás alapú megközelítésekhez képest az üzenetmások számát (legfeljebb L darab másolat létezhet a hálózatban) és alacsonyban tartásuk a késleltetést. Az üzenetmások számának korlátozása nyitva hagyja azt a kérdést, hogy az L darab másolatot mely csomópontok birtokolják. Két megoldást javasolnak az SNW és SNF módszerek közös első fázisának (*spray fázis*) megvalósításra [6]: Az egyszerűbb megoldás, hogy a forrás az első $L-1$ darab csomópontnak ad egy-egy másolatot, amelyekkel találkozik. Ezt a megoldást *forrás-alapú szétosztásnak* (source spray) nevezik, azonban a legtöbb alkalmazási környezet esetén hatékonyabbnak bizonyul az úgynevezett *bináris szétosztás* (binary spray), mely minimalizálja az üzenetmások szétosztásáig eltelt várható időt [6]. A bináris szétosztás lényege, hogy az üzenet forrása L darab másolat készítését engedélyezi. Viszont a forrás szétosztással ellentétben találkozás-kor minden olyan a csomópont, aminek több mint egy üzenetmásolat készítési joga van (először csak a forrás, később akár hordozók is), találkozik egy b csomóponttal, aminek nincs egy másolata sem az adott üzenetből, átad b -nek $\lfloor n/2 \rfloor$ darabot és megtartja a maradék jogot magának. Ökölszabályként elmondható, hogy az iniciális üzenetmások számát (L) a hálózatban lévő csomópontok 5-10%-ára érdemes választani.

Az SNW-módszer esetén a spray fázis után a *wait fázis* következik. Ha a spray fázisban nem sikerült kézbesíteni az üzenetet (nem volt a címzett az $L-1$ darab másik csomópont között), akkor mind az $L-1$ pont és a forrás is várakozik egészen addig, amíg nem találkozik a célcsoomóponttal, azaz közvetlen továbbításra váltanak. Az SNW-algoritmus erősen épít a csomópontok mozgására, és így próbálja meg elérni a célját. Egy olyan alkalmazási területen viszont, ahol az egyes csomópontok az idő nagy részében egy helyben vannak (pl. egyetemi kampusz), nem túl hatékony. Ennek a kiküszöbölésére ajánlották az SNF-algoritmust, amely megtartja a spray fázist, azonban az SNW-algoritmus utolsó, várakozási (közvetlen továbbítás) fázisa helyett az üzenetet egy egyetlen üzenetmásolat továbbítására ajánlott DTN-útvonalválasztási módszerrel [7] továbbítja a célcsoomópont felé egy előre (például eddigi találkozásuk száma és ideje, tranzitivitás alapján) definiált – a PROPHET kézbesítési előreláthatóságához hasonló – hasznosság függvény alapján.

4. Szimulációs eredmények

Szimulációinkban összehasonlítjuk az előző szakaszban bemutatott PROPHET, Epidemic és Spray and Wait algoritmusokat a The ONE [2] szimulátor segítségével, mely egy tömegközlekedési, jármű és gyalogos hálózatot modellezve végez térkép alapú szimulációt. A jelenlegi eredményeket Helsinkai térképe alapján készítettük.

4.1. Szimulációs beállítások

Az algoritmusok teljesítményének összehasonlítására két különböző szimulációs összeállítást használtunk. Az első összeállítással egy videotéka-szolgáltatás működését modelleztük. A város térképén két fix pont (szolgáltató és felhasználó) között zajlik a kommunikáció. A szolgáltatótól a városban közlekedő járművek tárolják és továbbítják elven juttatják el a hasznos információt a felhasználóig. A 802.11p szabványt [19] alapul véve átviteli sebességnek 24 Mbit/s-ot választottunk. A véges tárolókapacitás és sávszélesség nem teszi lehetővé egy nagyméretű (720 MB) videó egy darabban történő továbbítását, ezért az adatot kisebb üzenetekre (18 MB) kell darabolni a forrásnál.

A második összeállítás egy státusz üzeneteket küldő (szenzor-) hálózatot modellez (úthibák detektálása, dugók észlelése, szabad parkolóhelyek keresése stb.), ami lehetőséget ad az algoritmusok nagyobb adatforgalom esetén történő összehasonlítására. Ebben az összeállításban az adatátviteli sebesség nem korlátozza az algoritmusok teljesítményét, tehát a sávszélességet akkora választottuk meg, hogy egy találkozás alkalmával minden továbbításra szánt üzenetet át tudjon küldeni egymásnak a két csomópont.

1. táblázat A szimulációkban alkalmazott beállítások

Alkalmazási példa	Videotéka	Státusz
Csomópont szám	98 mozgó, 2 fix	100 mozgó
Sebesség	3 - 13,9 m/s	0,5 - 13,9 m/s
Átviteli sebesség	24 Mbit/s	„végtelen”
Hatósugár	50 - 150 m	50 m
Üzenet méret	18 MB	1 kB
Üzenet szám	40 db	200 db
Élettartam	5400 s (1,5 óra)	5400 s (1,5 óra)

Mindkét esetben valamennyi üzenet egyszerre generálódik 1000 másodperc felfutási fázis után, hogy a PROPHET kézbesítési előreláthatóság értékek kialakuljanak. Minden üzenet 1,5 óra (5400 másodperc) élettartammal rendelkezik, mely tekinthető a felhasználó által megadott határidőnek is, melyhez igazodnia kell a szolgáltatónak. Az élettartam lejártá után az üzenet törlődik a hálózatból.

4.2. Eredmények

Az ismertettett módszereken felül két további, az előzők kombinálásával illetve javításával nyert algoritmust ismertetünk. Az elsőt Spray and Prophet-nek [21] neveztük (SNP), ahol az SNW és SNF algoritmusok spray fá-

zisa után nem közvetlen továbbítást, illetve egyetlen üzenetmásolat továbbítást végzünk, hanem a PROPHET algoritmusnál megismert üzenetovábbítást és másolatkészítést alkalmazzuk. A spray fázisban a várhatóértékben optimális bináris szétszórást alkalmazzuk.

A másikat adaptív PROPHET (AP) [21] névre kereszteltük, ahol az útvonalválasztási döntés meghozatalakor figyelembe vesszük, hogy mennyi idő telt el az üzenet keletkezése óta. Az eredeti PROPHET nem volt figyelemmel arra, hogy az üzenetnek mennyi ideje van célbaérkezni és ettől független stratégiát alkalmazott. Az AP algoritmus esetén eleinte alacsony *hasznossági küszöböt* állítunk be az üzenetre, majd az idő előrehaladtával ezt növeljük. Ezzel egy agresszív üzenetterjesztés tapasztalható az üzenet keletkezése után, majd a végén szinte közvetlen átadásra váltunk. Az AP abban tér el a PROPHET-től, hogy két csomópont találkozása esetén nem a két, célcsomópontra vonatkozó kézbesítési előreláthatóságot hasonlítja össze, hanem az üzenet hordozójában a csomag hátralévő élettartama alapján számolt hasznossági küszöböt veszi alapul, míg a másolatot kérő csomópont esetén megtartja a célcsomópontra vonatkozó kézbesítési előreláthatóságot. A továbbiakban az átadásra vonatkozó döntés a PROPHET esetén megismertek szerint folytatódik.

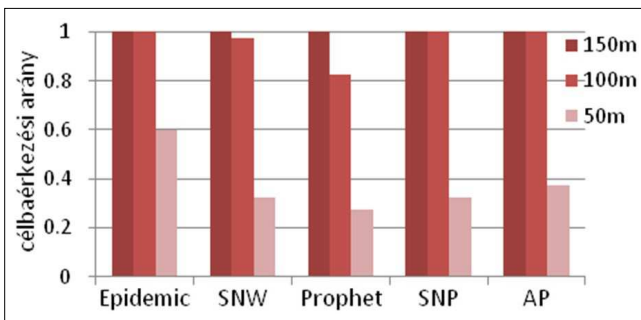
Az első szimulációs összeállításban először az összes üzenet tárolására alkalmas tároló mellett megvizsgáltuk, hogyan hat az adatátviteli eszköz hatósugara az egyes algoritmusok teljesítményére. A 4. ábrán jól látható, hogy a kisebb hatósugár kisebb célbaérkezési

arányt eredményez minden algoritmus esetén. Ez nem meglepő, hiszen kisebb hatósugár esetén a csomópontok kisebb valószínűséggel létesítenek összeköttetést a többi csomóponttal (kisebb a kommunikációs gráf összefüggősége), így az útvonalválasztás hatékonysága romlik. Megfigyelhető, hogy az SNP és az AP algoritmus a legnagyobb hatósugár esetén ugyanúgy (maximálisan), kisebb hatósugarak esetén pedig jobban teljesített, mint a Spray And Wait és a PROPHET algoritmusok.

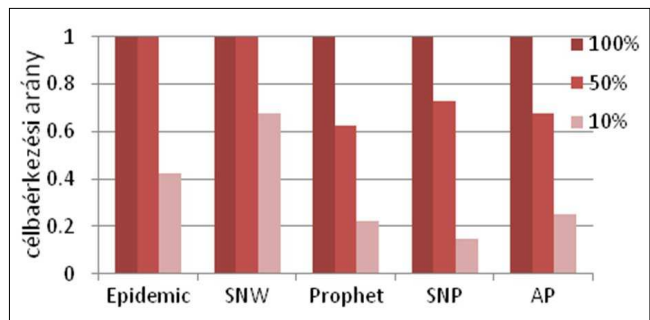
Szintén az első összeállításban az átviteli távolság változtatlansága mellett (150 m) megvizsgáltuk a tárolókapacitás méretének hatását az algoritmusok teljesítményére. A kisebb kapacitás általánosan rosszabb célbaérkezési arányt eredményez, mert ha nincs elegendő hely az új üzenetek számára, akkor az alacsonyabb prioritású üzenetek törlődnek, így nagy másolatszám esetén kisebb valószínűséggel jut célba egy üzenet, mint végtelen kapacitás (100%) esetén. A leghatékonyabbnak az Epidemic és a Spray And Wait algoritmus bizonyult, azonban látható, hogy az Adaptív PROPHET algoritmus korlátos kapacitás esetén jobb teljesítményt tud elérni, mint a PROPHET algoritmus (5. ábra).

A több üzenetet használó második összeállításban az algoritmusok hatékonyságát két különböző tárolókapacitás esetén (100% és 50%) elemeztük. A 6. ábra a célbaérkezési arányt mutatja egy adott szimulációs időpontig, maximális (korlátlan) tárolókapacitás mellett. 100%-os kapacitáson a legeredményesebben az Epidemic-algoritmusnak kell működnie, hiszen ebben az esetben minden útvonalon, azaz az optimálisan is terjed az

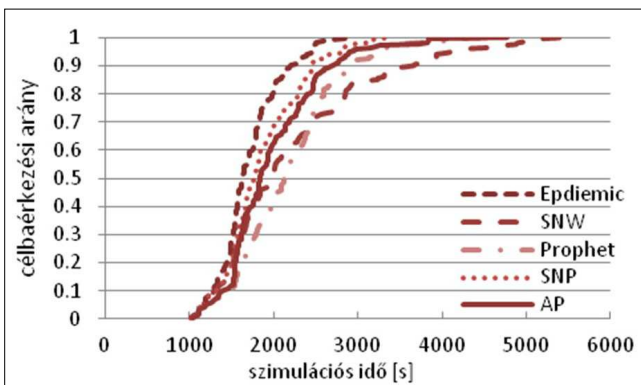
4. ábra
Célbaérkezési arány videotéka alkalmazás esetén különböző adatátviteli hatóságok mellett



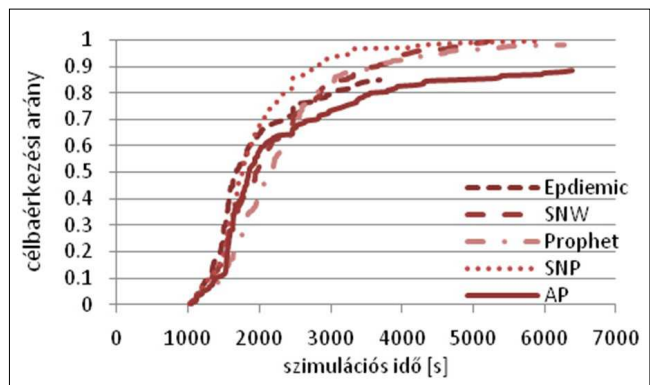
5. ábra
Célbaérkezési arány videotéka alkalmazás esetén különböző tárolókapacitások mellett



6. ábra
Célbaérkezési arány státusz üzenetek küldése esetén maximális tárolókapacitás mellett



7. ábra
Célbaérkezési arány státusz üzenetek küldése esetén 50%-os tárolókapacitás mellett



üzenet. Ezért az Epidemic-algoritmust *alsó korlátnak* tekinthetjük a késleltetésre, hiszen ennél gyorsabban ebben a szimulációs összeállításban egyik algoritmus sem képes az üzeneteket célba juttatni. Látható, hogy az összeállításban szereplő algoritmusok közül mindegyik kézbesítette az összes üzenetet, azonban az Epidemic után a leghatékonyabb a Spray And Prophet algoritmus volt. Az ábráról kitűnik a Spray And Wait és a Spray And Prophet algoritmusokat összehasonlítva, hogy mekkora teljesítményjavulást eredményez a spray fázis utáni másolatkészítés.

50%-os kapacitás esetén már nem jelent referenciát az Epidemic-algoritmus, mert a nagy másolatszám nagymértékben csökkenti a hatékonyságot. Az üzenetek élettartamának lejártáig nem is sikerült minden algoritmusnak az összes üzenetet kézbesíteni. Az ábrán kiemelkedik a Spray And Prophet kézbesítési sebessége. Míg a Spray And Wait és a PROPHET algoritmusok közel azonos sebességgel kézbesítik az üzeneteket, addig az Adaptív Prophet elmarad ettől a teljesítménytől a nagyobb számú üzenetmásolat használata miatt.

5. Összefoglalás

Cikkünkben különböző elven alapuló DTN útvonalválasztási módszereket tekintettünk át, kiemelt hangsúlyt adva az alkalmazási környezet legfőbb kihívásainak és a teljesítmény mutatóinak. A tárolj és továbbíts elv miatt a környezetből adódó egyik legfőbb korlát a csomópontokban rendelkezésre álló tároló, mely nagyban befolyásolja az adott alkalmazáshoz leginkább illeszkedő módszer kiválasztását.

Szimulációk segítségével nagyvárosi járműhálózaton keresztül összehasonlítottuk a módszereket, valamint bemutattunk két további új algoritmust (SNP, AP). Az eredmények alapján megfigyelhető a véges méretű tárolókapacitás összetett hatása a sok üzenetmásolatot használó algoritmusokra.

A szerzőkről



BABARCZI PÉTER 2008-ban szerzett műszaki informatikus oklevelet a Budapest Műszaki és Gazdaságtudományi Egyetemen, amely diplomamunka III. díjat nyert a Hírközlési és Informatikai Tudományos Egyesület diplomatervezési pályázatán. 2008-ban három hónapot töltött a kanadai Waterloo Egyetemen, mint vendégkutató. Jelenleg PhD hallgató a BME Távközlési és Médiainformatikai Tanszékén, ahol a HSN laboratórium tagja. Kutatási területei optikai hálózatok optimalizálása (hozzárendelt védelem, többszörös meghibásodások egyértelmű lokalizációja) és útvonalválasztás késleltetés toleráns hálózatokban. Több publikációját fogadták már el rangos nemzetközi szakmai konferenciákon, illetve folyóiratokban.



CSIKOR LEVENTE 2010-ben szerzett műszaki informatikus oklevelet a Budapest Műszaki és Gazdaságtudományi Egyetemen. Jelenleg PhD hallgató a BME Távközlési és Médiainformatikai Tanszékén, azon belül is a HSN laboratórium csapatában. Kutatási területei a gyors hibajavítás IP hálózatokban (lokális hibadetektálás és -javítás hurokmentes kerülőútvonalak és hálózati topológiai analízis segítségével) valamint útvonalválasztás és alkalmazhatóság késleltetés-toleráns hálózatokban.



HESZBERGER ZALÁN a BME-n kapott villamosmérnöki diplomát 1997-ben. Doktori fokozatát 2007-ben villamosmérnöki tudományok területén szerezte. 2000 óta a BME Távközlési és Médiainformatikai Tanszékének oktatója, ahol jelenleg egyetemi adjunktus. 2008 óta a Hírközlési és Informatikai Tudományos Egyesület Távközlési szakosztályának elnöke. Kutatási és oktatási területe a nagysebességű hálózatok menedzsmentje, illetve a neurális és önszerveződő hálózatok. Rendszeresen publikál nemzetközi szakmai konferenciákon illetve folyóiratokban. Számos nagy hazai és nemzetközi projekt résztvevője és vezetője.



TANAI FERENC 2010-ben készítette szakdolgozatát a Budapesti Műszaki és Gazdaságtudományi Egyetem Távközlési és Média Informatikai Tanszékén. Késleltetés toleráns hálózatok útvonalválasztó algoritmusai témakörében írt TDK dolgozata III. díjat nyert a 2010. évi kari TDK konferencián.



TAPOLCAI JÁNOS a Budapesti Műszaki és Gazdaságtudományi Egyetemen szerzett villamosmérnöki diplomát 2000-ben. Doktori fokozatát 2005-ben informatikai tudományok területén szerezte. 2000 óta a BME Távközlési és Médiainformatikai Tanszékének oktatója, ahol jelenleg egyetemi adjunktus. Kutatási és oktatási területe a nagysebességű hálózatok címzéséhez és útvonalválasztásához, és a kombinatorikus optimalizáláshoz kapcsolódik. Rendszeresen publikál a legerősebb nemzetközi szakmai konferenciákon, illetve folyóiratokban. Több mint 60 cikk szerzője, összesített impact faktora 22 és elnyerte a legjobb cikk díját 2006-ban az IEEE ICC konferencián.

Irodalom

- [1] A. Elwhishi, P.-H. Ho, SARP A Novel Multi-Copy Routing Protocol for Intermittently Connected Mobile Networks. In GLOBECOM, 2009.
- [2] A. Keranen, J. Ott, T. Karkkainen, The ONE Simulator for DTN Protocol Evaluation. In SIMUTools '09: Proc. of the 2nd Int. Conference on Simulation Tools and Techniques, ICST. New York, NY, USA, 2009.
- [3] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks. Lecture notes in Computer Science, pp.239–254, 2004.
- [4] E. Royer, C. Toh, A review of current routing protocols for ad-hoc mobile wireless networks. IEEE personal communications, 1999.
- [5] T. Spyropoulos, K. Psounis, C. Raghavendra, Performance analysis of mobility-assisted routing. In Proc. of the 7th ACM Int. Symposium on Mobile ad hoc networking and computing. ACM, p.60, 2006.
- [6] T. Spyropoulos, K. Psounis, C. Raghavendra, Efficient routing in intermittently connected mobile networks: The multiple-copy case. IEEE/ACM Transactions on Networking (TON), 16(1):77–90, 2008.
- [7] T. Spyropoulos, K. Psounis, C. Raghavendra, Efficient routing in intermittently connected mobile networks: The single-copy case.

- IEEE/ACM Transactions on Networking (TON), 16(1):63–76, 2008.
- [8] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks. Duke University, 2000.
- [9] Delay Tolerant Networking Research Group, 2010 [Online], <http://www.dtnrg.org/wiki>
- [10] S. Jain, K. Fall, R. Patra, Routing in a delay tolerant network, SIGCOMM Comput. Com. Rev., 34(4):145–158, 2004.
- [11] Y. Wang, S. Jain, M. Martonosi, K. Fall, Erasure-coding based routing for opportunistic networks. In Proc. of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, p.236, 2005.
- [12] O. Akan, J. Fang, I. Akyildiz, TP-Planet: a reliable transport protocol for InterPlaNetary Internet. IEEE Journal on Selected Areas in Communications, 22(2):348–361, 2004.
- [13] R. Shah, S. Roy, S. Jain, Brunette, Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. In Elsevier Ad Hoc Networking Journal, Vol. 1, pp.215–233, 2003.
- [14] M. Musolesi, S. Hailes, C. Mascolo, „Adaptive routing for intermittently connected mobil ad hoc networks”, In Proc. of IEEE WoWMoM, pp.183–189, 2005.
- [15] K. Fall, A delay-tolerant network architecture for challenged internets. Proc. of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp.27–34, 2003.
- [16] S. Farrell, V. Cahill, Delay- and Disruption-Tolerant Networking, Artech House. Inc., Norwood, MA, 2006.
- [17] R. Ahlswede, N. Cai, S. Li, R. Yeung, Network information flow. IEEE Transactions on Information Theory, 46(4):1204–1216, 2000.
- [18] J. Burgess, B. Gallagher, D. Jensen, B. Levine, Maxprop: Routing for vehicle-based disruption-tolerant networks. In Proc. IEEE Infocom, Vol. 6. Citeseer, 2006.
- [19] W. Fisher, „Development of DSRC/WAVE Standards,” Szabvány: IEEE 802.11-07/2045r0, June 2007.
- [20] E. Jones, P. Ward, „Routing strategies for delay-tolerant networks,” Submitted to Computer Communication Review, <http://ccng.uwaterloo.ca/paswardPublications/>
- [21] Tanai F., „Útvonalválasztó algoritmusok tervezése és kiértékelése időszakosan összefüggő mobil hálózatokban,” TDK dolgozat, BME-VIK TMIT, 2010.

Felhívás

Ezúton is felhívjuk Olvasóink figyelmét a Híradástechnika magyar folyamában való publikálási lehetőségekre. Elsősorban közérthető, széles olvasóközönségnek szóló, színvonalas áttekintő cikkeket várunk, amelyek egy-egy szűkebb szakterület érdekességeit mutatják be azok számára is, akik nem ezen a területen dolgoznak. Célunk, hogy a szakma egyetlen magyar nyelvű, színvonalas ismeretterjesztő folyóirataként közvetítsük az egyes részterületek helyzetét, fejlődésének irányait és legújabb eredményeit a minél szélesebb olvasótábor számára és formáljuk, befolyásoljuk a magyar szaknyelvet.

Várjuk Olvasóink jelentkezését a fentiek szerint elkészített kéziratokkal, az infokommunikáció különböző részterületeiről és határterületeiről, többek között az alábbi témákban:

- Adat- és hálózatbiztonság
- Digitális műsorszórás
- Infokommunikációs szolgáltatások
- Internet-technológiák és alkalmazások
- Médiainformatica
- Multimédia-hálózatok és rendszerek
- Optikai kommunikáció
- Társadalmi vonatkozások
- Távközlés-gazdaság és -szabályozás
- Távközlési szoftverek
- Teszthálózatok és kutatási infrastruktúrák
- Úrhírközlés
- Vezetéknélküli és mobil távközlés

Rendszeresen jelentkező rovatainkhoz is várjuk beküldött anyagaikat, melyek közül a következőket szeretnénk kiemelni:

- hazai és nemzetközi projektek ismertetése,
- konferenciákról, fontos szakmai eseményekről szóló beszámolók,
- a HTE szakosztályainak tevékenységét bemutató cikkek,
- egyetemi és kutatóintézeti egységek bemutatkozása,
- könyvismertetések.

A kéziratosokat kérjük a főszerkesztőnek elektronikusán megküldeni a szabo@hit.bme.hu címre, akihez a témákkal és a cikkek elkészítésével kapcsolatos bármilyen kérdéssel is fordulhatnak a fenti e-mail-címen. A szerzőinknek szóló tájékoztató elektronikus változatát lapunk internetes portálján találhatják meg, a www.hiradastechnika.hu cím alatt.

A Szerkesztőség