

híradástechnika

1945 VOLUME LXI. 2006

hírközlés ■ informatika



Ambiens intelligencia

Szenzorhálózatok

Intelligens közlekedési rendszerek

Pervazív, helyzetfüggő szolgáltatások

2006/12

**A Hírközlési és Informatikai Tudományos Egyesület folyóirata
a Nemzeti Hírközlési és Informatikai Tanács együttműködésével**

Tartalom

<i>AMBIENS INTELLIGENCIA</i>	2
Ács Gergely, Buttyán Levente Útvonalválasztó protokollok vezeték nélküli szenzorhálózatokban	3
Vincze Zoltán, Vida Roland Mobil eszközök alkalmazása szenzorhálózatokban	12
Vass Dorottya, Vidács Attila Energiahatékony kommunikáció szenzorhálózatokban	18
Benkő Borbála, Katona Tamás, Schulcz Róbert CASCADAS – Autonóm kommunikáció és pervazív, helyzetfüggő szolgáltatások	23
Vajda Lóránt, Kardos Sándor, Gerháth Gábor, Mezny Balázs, Laborczi Péter, Gordos Géza Városi közlekedési rendszerek hatékonyságának javítása mobil ad-hoc hálózatok segítségével	29
Csák Bence Ambiens intelligencia a közutakon	35
Dévai István, Bányász Gábor, Levendovszky Tihamér Webszerver a mobiltelefonon – a 3G tartalommegosztás új formája	40
Varga László, Hosszú Gábor Kisfogyasztású integrált áramkörök tervezési kérdései	45
Kóczy T. László, Botzheim János, Sallai Richárd, Csányi Kornél Fuzzy következtető rendszerek alkalmazása mobil hálózatok felügyeletében	52
Kántor Csaba Könyvajánló – Dr. Lajtha György: Emberek és események	60
<i>ZTE: NG-DSLAM Solution Overcomes the Challenges of NGN DSLAM Design (x)</i>	62

Védnökök

SALLAI GYULA a HTE elnöke és DETREKŐI ÁKOS az NHIT elnöke

Főszerkesztő

SZABÓ CSABA ATTILA

Szerkesztőbizottság

Elnök: ZOMBORY LÁSZLÓ

BARTOLITS ISTVÁN
BÁRSONY ISTVÁN
BUTTYÁN LEVENTE
GYŐRI ERZSÉBET

IMRE SÁNDOR
KÁNTOR CSABA
LOIS LÁSZLÓ
NÉMETH GÉZA
PAKSY GÉZA

PRAZSÁK GERGŐ
TÉTÉNYI ISTVÁN
VESZELY GYULA
VONDERVISZT LAJOS

Ambiens intelligencia

gyori@tmit.bme.hu
imre@hit.bme.hu

Eszám cikkeinek döntő részét az „ambiens intelligencia” kulcskifejezés kapcsolja össze. Az ambiens intelligencia a számítástechnika, távközlés és szenzortechnológia kombinációjaként létrejött új „paradigma”. Olyan rendszerekről van szó, amelyek elemei általában nem láthatók környezetünkben, segítségükkel viszont számos területen válhat könnyebbé, hatékonyabbá az élet. Rokon fogalmak a pervazív számítástechnika, beágyazott rendszerek, szenzorhálózatok.

Kezdjük ez utóbbival. Ha az állatvilágból kéne példát keresnünk, akkor a szenzorhálózatokat talán a vadkacsához hasonlíthatnánk, aki úszik is, repül is, fut is, de egyikben sem ő viszi el a pálmát. A szenzorhálózatok hasonló szerepet töltenek be az infokommunikációs hálózatok körében: van memória kapacitásuk, van számítási kapacitásuk, rendelkeznek saját táplálással, de egyikből sem áll korlátlan mennyiség a rendelkezésükre. Ezért mindennel spórolni kell, hogy lehető leghosszabb ideig kitartson. A rendelkezésére álló energiával takarékoskodni kell, hogy minden funkcióra jusson belőle: az érzékelésre, az adatok feldolgozására, a kommunikációra, vagyis az adatok továbbítására illetve mozgásra.

Ács Gergely és Buttyán Levente cikkükben rendszerezik az ad-hoc a útvonalválasztó protokollokat, és minden családból bemutatnak egy prominens képviselőt. A cikk újdonsága a rendszerezéshez használt szempontrendszer, mely a protokollok eddigieknél részletesebb taxonómiáját eredményezi.

A szenzorhálózatok kialakításánál alkalmazott különböző energiatakarékos módszerekkel ismerkedhetünk meg Vincze Zoltán és Vida Rolland cikkében. A probléma ismertetésén túl olvashatunk különböző megoldási lehetőségekről is. Többek között az általuk javasolt és jelenleg a gyakorlati megvalósítás előtt álló módszerről, amikor egy olyan mozgó állomás gyűjti be szenzoroktól az adatokat, ami újratölthető energiaforrással rendelkezik.

A szenzorhálózatok energia hatékony kommunikációja is megnövelheti a hálózatot alkotó kis szenzorok élettartamát és ezzel az egész hálózat hosszabb működésre lesz képes. Egy nagyon sok elemből álló hálózat esetén problémát okozhat a rendelkezésre álló átviteli csatornák kiosztása, gondoskodni kell arról, hogy minél kevesebb legyen az ütközés, mert az adatok újra küldése jelentős energiát emészthet el. Ugyancsak új lehetőségeket kell feltárni a címezési eljárások során, mert a „hagyományos” IP címek használata nem célszerű ilyen sok, relatív rövid ideig alkalmazott hálózati elem esetén. Ilyen és ehhez hasonló problémákra ad lehetséges megoldásokat Vass Dorottya és Vidács Attila cikke. Mindkét cikkhez tartozik egy-egy remek irodalom jegyzék, ami segít a problémakör alaposabb megismerésében is.

Benkő Borbála, Katona Tamás, Schulcz Róbert a CASCADAS IST FET projekt keretében végzett kutató-fejlesztő munkáról számolnak be a cikkükben. A CASCADAS projekt alapvető célkitűzése, hogy autonóm és automatikus kommunikációra épülő, helyzetudatos/helyzETFüggő, önszerveződő alapegységekből felépített hálózatot valósítsanak meg.

A távközlő hálózatok és az úthálózat közötti analógiára világít rá a Vajda Lóránt és társainak cikke. A trendek azt mutatják, hogy egyre több autóra szerelnek Globális Helymeghatározó Rendszert (GPS). Ezek segítségével kialakítható olyan ad hoc hálózat, ami képes a közlekedési utakon fellépő torlódásokat felderíteni, előre jelezni és alternatív útvonalakat ajánlani a vezetőknél.

A mindenütt jelenlévő számítástechnika egyik remek példája lehet a biztonságos közlekedést segítő közlekedési rendszer, amihez a központi vezérlő egységet a Bay Zoltán Alkalmazott Kutatási Közalapítvány kutatói fejlesztették ki. A Hermes1 névre keresztelt egység alkalmazási lehetőségein túl annak a hardver felépítésével is megismerkedhetünk.

A már hazánkban is egyre szélesebb körben elérhető 3G hálózatok lehetővé tesznek olyan új hálózati alkalmazásokat, melyek kihasználják a mobiltelefon változó földrajzi pozícióját és mobiltelefonok adhoc létrejövő és megszűnő csoportjait. Dévai István, Bányász Gábor és Levendovszky Tihamér bemutatják, hogyan lehet egy 3G mobiltelefonon webszervert futtatni és milyen módon lehet elérhetővé tenni ezt a szolgáltatást az Internet közönsége részére.

Beküldött cikkeinkből e számunk fő témájához kapcsolódik Varga László és Hosszú Gábor munkája, amely kifizogasztású integrált áramkörök tervezési kérdéseivel foglalkozik, specifikusan az adiabatikus töltésvisszanyerő kapcsolásokkal, amelyek fő jellemzője, hogy az áramköri kapacitások feltöltése és kisütése Joule-vesztés nélkül történhet.

Kóczy T. László és társainak cikke mikrohullámú összeköttetések intelligens hálózatfelügyeletével foglalkozik, és egy olyan új elvi és rendszertechnikai megoldásról számol be, ahol a felügyeleti rendszer Fuzzy döntéshozatali alrendszer egészíti ki, amely a csillapítás mértékéből és időbeli változásából következtet az optimális beavatkozásra.

Imre Sándor,

BME Híradástechnikai tanszék

Győri Erzsébet,

BME Távközlési és Média-informatikai Tanszék

vendégszerkesztők

Útvonalválasztó protokollok vezeték nélküli szenzorhálózatokban

ÁCS GERGELY, BUTTYÁN LEVENTE

Budapesti Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék, CrySyS laboratórium
{acs, buttyan}@crysys.hu

Lektorált

Kulcsszavak: szenzorhálózatok, útvonalválasztás, protokollok, hálózati és működési modellek

A szenzorhálózatok változatos alkalmazásai különböző követelményeket támasztanak az útvonalválasztó protokollokkal szemben. A különböző követelményeknek köszönhetően igen sok javasolt protokoll található az irodalomban. Ebben a cikkben rendszerezük ezeket a útvonalválasztó protokollokat, és minden családból bemutatunk egy prominens képviselőt. A cikk újdonsága a rendszerezéshez használt szempontrendszer, mely a protokollok eddigieknél részletesebb taxonómiáját eredményezi.

1. Bevezetés

A szenzorhálózatokat egyrészt a mérést végző szenzorcsomópontok, másrészt az adatokat begyűjtő bázisállomások (báziscsomópontok) alkotják.

Míg a rendkívül nagy számú, és általában homogén szenzorcsomópontok tipikusan alacsony energiaellátottsággal és számítási kapacitással rendelkeznek, addig a kis számú bázisállomások erőforrása legtöbbször korlátlan. Az egyes szenzorcsomópontok egymással és a bázisállomásokkal egyaránt vezeték nélküli kapcsolatokon kommunikálnak, amelyeknek az energiaigénye jóval magasabb mint a csomópontok által végzett egyszerű számításoké, valamint egy üzenet elküldéséhez körülbelül kétszer annyi energia szükséges mind annak vételéhez.

Ezekből következik, hogy egy szenzorcsomópont által küldött üzenet útvonala egy bázis felé korántsem mellékes: a túl hosszú, illetve alacsony teleptöltöttségű csomópontokat tartalmazó útvonalak csökkentik a hálózat élettartamát, valamint a túl sok csomópontot tartalmazó útvonalak adott esetben növelhetik az üzenet késleltetését.

Sajnos ezen követelmények sokszor ellentmondanak egymásnak: ha mindig a legrövidebb útvonalon továbbít egy szenzorcsomópont a bázis felé, akkor az a köztes csomópontok rövid élettartamához, és bizonyos értelemben a hálózat élettartamának rövidüléséhez vezet, holott ez jelentené a legkisebb energiafogyasztást és késleltetést globális értelemben.

Összességében az útvonalválasztás célját maga az alkalmazás határozza meg; például valós idejű alkalmazásoknál a minimális késleltetés, míg statisztikai számításokat végző alkalmazásoknál a hosszú élettartam lehet elsődleges cél. A sokféle alkalmazásra így különböző útvonalválasztó protokollokat javasoltak [1], amelyek az útvonalválasztás céljában, valamint ezen célok elérésre használt technikákban különböznek, ahol a technikákat a hálózat által támasztott technológiai korlátok alakítják.

2. Osztályozási szempontrendszer

Az útvonalválasztó protokollok nagy száma a következő természetes kérdést vetheti fel egy alkalmazásfejlesztő számára: Melyik protokoll felel meg legjobban az alkalmazásom számára? E kérdés megválaszolásához az szükséges, hogy az összes útvonalválasztó protokollt egy olyan közös szempontrendszerbe helyezzük, ahol a protokollok összehasonlíthatóvá válnak.

Rendszerezésünk alapját egyrészt a hálózat technológiáját és a protokollok működését leíró rendszermodell, másrészt a protokoll által kitűzött cél alkotja. A rendszermodell tovább osztható a hálózati technológiát leíró hálózati modellre, valamint a működési tulajdonságokat leíró működési modellre.

2.1. A hálózati modell

2.1.1. Bázisállomás

A bázisállomások nagy számítási kapacitású és korlátlan energiaellátottsággal rendelkező hálózati eszközök. A következő tulajdonságok viszont már hálózatonként eltérőek lehetnek, és így az útvonalválasztó protokoll működését is befolyásolják.

- **Szám:** A bázisállomások száma lehet egy, illetve egynél több. A legtöbb gyakorlati esetben a tipikus számuk egy, de több bázisállomás növeli az adatbegyűjtés robusztusságát, valamint csökkenti annak késleltetési idejét. Az egy bázisállomást tartalmazó esetben (ha nincs egyéb igény a szenzorcsomópontok közötti explicit kommunikációra) a végcél minden üzenet számára ugyanaz, míg egynél több számú bázisállomás esetében, a cél lehet több, nem feltétlen minden üzenetre egyező bázisállomás.

- **Mobilitás:** Egy bázisállomás lehet fix (helyhezkött), korlátozott mértékben mobilis, illetve korlátlanul mobilis. Legtöbbször olyan hálózatokban mobilisak a bázisállomások, ahol a számuk kevés, mégis nagyobb robusz-

tusságra és kisebb késleltetésre van szükség az ugyanolyan számú de helyhez kötött bázisállomások esetéhez képest. A mobilitás miatt az útvonalválasztás által kialakított topológia időben nagy mértékben változhat, ami túlterheltséget jelenthet a hálózati rétegben.

- **Jelenlét:** Néhány alkalmazásnál a bázisállomások nincsenek folyamatosan jelen (például kikapcsolják őket karbantartási célból), míg más esetekben a jelenlét folyamatosan biztosított. Az időszakos leállásokat az útvonalválasztó protokollnak támogatnia kell, mivel a bázis jelenlétének hiánya nem feltétlen ebben az esetben nem hiba, és így az üzenetet nem kell eldobni vagy átírányítani, hanem azokat esetleg várakoztatni kell.

- **Lefedtettség:** A bázisállomás nagy erőforráskészlete lehetővé teszi egy erős antenna alkalmazását is, amivel esetleg az egész hálózat lefedhető, és így minden csomópontot közvetlenül elérhet a bázisállomás (ha nincs árnyékoló akadály). Ekkor nincsen szükség a bázis és a szenzorcsomópontok közötti útvonalválasztásra. Megjegyezzük azonban, hogy a legtöbb gyakorlati esetben a lefedettség nem teljes.

2.1.2. Szenzorcsomópont

Szenzor alatt a továbbiakban a vezérlő áramkörök és rádiós egység, az érzékelő egység, és a szükséges telep együttesét értjük. A szenzorok kis számítási kapacitású és korlátozott energiaellátottsággal rendelkező, legtöbbször helyhez kötött hálózati eszközök. A következő tulajdonságok viszont már hálózatonként eltérőek lehetnek, és így az útvonalválasztó protokoll működését is befolyásolják.

- **Elhelyezés:** A szenzorok elhelyezése lehet véletlenszerű, vagy determinisztikus. Sok esetben az útvonalválasztó protokollnak alkalmazkodnia kell a kötött hálózati szerkezethez (például út mentén szabályosan elhelyezett szenzorok), más esetben (helikopterről kiszórt szenzorok) pedig alapozhat arra a feltevésre, hogy a szenzorok egyenletesen véletlenül vannak elosztva.

- **Átviteli energia:** Az átviteli energia szintje lehet állítható, vagy pedig konstans, vagyis a szenzor minden üzenetet ugyanazon az energiaszinten sugároz. Ez elsősorban a szomszédos csomópontok energiában mért távolságának a meghatározásában játszik szerepet, hiszen sok protokoll ezt használja a szomszédokhoz rendelt költségek megállapításához.

- **Lefedtettség:** Szükséges esetben, ha egy szenzor nagy energiával ad, bármely más csomópontot közvetlenül elérhet a hálózatban. Ennek kihasználása azonban általában csak kis méretű hálózatokban lehet hatékony a nagy energiaigény és a megnövekedett interferencia miatt.

- **Címzés:** Az útvonalválasztás feladata a szenzorhálózatokban a bázistól (vagy ritkán más szenzortól) ér-

kező kérések eljuttatása ahhoz a csomópont(ok)hoz, amely rendelkezik a kért adattal, valamint ezen adat visszajuttatása a bázishoz. Ennek megfelelően beszélhetünk a kérés, illetve a válasz kezelése során alkalmazott címzésről:

- **Kérés:** A kérés címzése történhet a kért adat alapján (Mi az átlaghőmérséklet?), vagy a kért adat vagy szenzor pozíciója alapján (Mi a hőmérséklet az (x,y) helyen?).
- **Válasz:** A válasz címzése történhet a bázis vagy szenzor pozíciója alapján (Válasz az (x,y) helyen levő bázishoz), vagy a választ váró csomópont lokális, vagy globális azonosítója alapján. Utóbbi esetben, a kérés vétele során minden szenzor megjegyzi a kérést küldő szomszéd azonosítóját, és ennek a szomszédnak továbbítja a választ visszafelé.

- **A MAC réteg által nyújtott szolgáltatás:** A MAC réteg gondoskodhat a szomszédos csomópontok felderítéséről (ahol a szomszéd definíciója változó), illetve ezen felül, támogathatja a szomszédok költségeinek megállapítását (ahol a költség definíciója szintén protokollonként eltérő). Néhány útvonalválasztó protokoll integrálva van a MAC réteggel a gyorsabb és energiatakarékos működés végett (cross layer design). Viszont a legtöbb esetben a MAC réteg nem felel ezen feladatokért, és ezért magának az útvonalválasztó protokollnak a feladata a szomszédok és azok költségeinek megállapítása.

2.2. A működési modell

Az útvonalválasztó protokollok a következő ortogonális működési tulajdonságokkal jellemezhetőek:

- **Kommunikációs minta:** Az útvonalválasztó protokoll támogathatja a szenzorok közötti, a bázis és szenzorok közötti, valamint szenzorok és bázis közötti kommunikációt.

- **Szenzor–szenzor kommunikáció:**
Ezt a kommunikációs mintát elsősorban az ad-hoc hálózatok számára javasolt protokollok támogatják, szenzorhálózatokban ilyen kommunikációra ritkán van szükség.
- **Bázis–szenzor kommunikáció:**
A bázis felől érkező kérések irányítása során szükséges ennek a kommunikációs mintának a támogatása, mely azt a képességet jelenti, hogy a bázis közvetlenül vagy közvetve bármely szenzornak tud üzenetet küldeni.
- **Szenzor–bázis kommunikáció:**
A válaszok irányítása során szükséges ennek a mintának a támogatása, mely azt a képességet jelenti, hogy minden szenzor képes közvetlenül vagy közvetve bármely bázisnak üzenetet küldeni. Minden egyes minta esetén a kommunikáció típusa lehet unicast (egy-egy), multicast (egy-több), reverse-multicast (több-egy), illetve anycast (egy-

bármely). Anycast kommunikációra például akkor lehet szükség amikor a bázis adatot kérdez a hálózattól, és nem lényeges, hogy pontosan melyik szenzor válaszol a kérésre, hanem bárki válaszolhat, aki rendelkezik a kért adattal.

- *Hierarchia:* A hierarchikus protokollok esetében az egyes szenzorok (logikai) hierarchiaszinteken helyezkednek el. A szenzorok az alacsonyabb szinten levő szenzoroktól fogadnak üzeneteket, ezeket aggregálják saját adataikkal, és az aggregátumot továbbítják a magasabb hierarchiaszinten levő szenzoroknak. A hierarchia tetején a bázis található. A hierarchia kialakítása lehet statikus vagy dinamikus. Utóbbi esetben, a szenzorok dinamikusan választanak aggregátorokat, és ennek az aggregátor csomópontnak küldenek minden üzenetet. Az aggregátorok további aggregátorokat választanak, és így tovább. A hierarchia kialakításának célja a hálózat élettartamának növelése. A nem-hierarchikus protokollok esetén az egyes csomópontok bármely csomóponttól fogadnak üzenetet aggregálásra, így minden csomópont viselkedhet aggregátorként.

- *Kézbesítési módszer:* A legtöbb protokoll egyetlen útvonalat választ a bázis felé, s ezen minden üzenet egyetlen példányát továbbítja (egy/egy). Néhány protokoll azonban a robusztusság érdekében több útvonalat is választ, s vagy minden üzenetet ezen útvonalak egyikén továbbít (több/egy), vagy minden üzenetet minden útvonalon továbbít (több/több). Előbbi esetben, a továbbításra használt útvonal kiválasztása lehet determinisztikus vagy véletlenszerűen.

- *Számítás:* Az egyes szenzorok vagy maguk határozzák meg lokálisan a következő csomópontot a bázis felé (decentralizált), vagy pedig minden csomópont elküldi a szomszédossági listáját a bázisnak, amely globálisan meghatározza minden egyes szenzornak a következő csomópontot a bázis felé (centralizált). Az utóbbi optimális megoldást nyújt, viszont ehhez nagy mértékű hálózati kommunikáció szükséges, amely csak kevés számú csomóponttal rendelkező és fix topológiájú hálózatoknál használható hatékonyan. A szenzorok csak a szomszédokkal tartják a kapcsolatot, ahol a szomszéd definíciója változó lehet. Legegyszerűbb esetben, a szenzorok azt tekintik szomszédnak, akitől útvonalválasztó üzenetet kapnak. Más esetekben explicit HELLO üzenetek segítségével derítik fel a szomszédokat. Ekkor minden csomópont egy bizonyos energiaszinten küld egy HELLO üzenetet többesküldéssel (broadcast), és minden csomópont azt tekinti szomszédnak akinek HELLO üzenetét hallotta. Ez a mechanizmus lehet része az útvonalválasztó protokollnak, vagy pedig az alsóbb protokollrétegben (például MAC réteg) lehet megvalósítva.

- *Állapot:* Az egyes protokollok futtatáskor szükséges lehet minden csomópontnak valamilyen információt tárolni az aktuális állapotról (például a saját költség-

gét, ki a következő csomópont a bázis felé, annak mi a költsége stb.) Ezzel szemben néhány protokoll semmilyen, vagy csak elhanyagolható mennyiségű állapotinformációt tárol. Utóbbi esetben, minden csomópont az üzenetben elhelyezkedő információ, vagy minimális lokálisan tárolt állapotinformáció segítségével (például a szomszédos csomópontok pozíciói) képes meghatározni a következő csomópontot a bázis felé.

- *Következő szomszéd választása:* „Az összes protokoll közös tulajdonsága, hogy a csomópontok lokálisan, a saját információik alapján választják a következő csomópontot a továbbítandó csomag útján. A választás történhet: véletlenszerűen, az üzenet tartalma alapján, az üzenetben szereplő geometriai pozíció alapján, hierarchia szintek alapján, illetve néhány protokollban a csomópontok az üzeneteket többeszórással továbbítják, és a szomszédok maguk döntenek a továbbításról.”

- *Riportolási modell:* A riportolási modell három féle lehet, ahol a csoportosítás alapját az ok adja, ami a forrásszenzorokat üzenetküldésre készíti:

- *Idővezérelt:* A szenzorok szabályos időközönként, vagy egy bizonyos időpontban válaszolnak. Az idővezérelt protokoll támogathat folytonos (periodikus) vagy időben egyszeri jelentést, komplex (összetett típusú) vagy egyszerű (atomi típusú) adatok jelentését, aggregálható vagy nem aggregálható adatok jelentését, illetve replikált (több szenzornál megtalálható) vagy egyedi (egyetlen szenzornál megtalálható) adat jelentését.

- *Kérésvezérelt:* A szenzorok a bázis kérésére válaszolnak. A kérésvezérelt protokoll támogathatja komplex vagy egyszerű adatok jelentését, aggregálható vagy nem aggregálható adatok jelentését, illetve replikált vagy egyedi adat jelentését.

- *Eseményvezérelt:* A szenzorok egy bizonyos esemény hatására küldenek üzenetet a bázis felé. Az eseményvezérelt protokoll támogathatja komplex vagy egyszerű adatok jelentését, aggregálható vagy nem aggregálható adatok jelentését, illetve replikált vagy egyedi adat jelentését.

Néhány protokoll több csoportba is tartozhat, ha többféle riportolási modellt is támogat.

2.3. A protokoll célja

Minden útvonalválasztó protokoll alapvető célja az üzenet eljuttatása a forrástól a célig. Ez történhet valós idejű követelményekkel együtt, amikor az üzenetnek egy bizonyos időn belül el kell érnie a célt, vagy pedig valós idejű követelmények nélkül. Utóbbi esetben a sikeresség egy mérőszáma lehet a sikeresen kézbesített üzenetek száma, míg az előbbi esetben a sikerességet az adott időn belül sikeresen kézbesített üzenetek száma méri.

A protokolloknak az előbbivel párhuzamos célja lehet a hálózat élettartamának maximalizálása, aminek a

mérőszáma már nem egyértelmű. Ha minden csomópont egyenrangú, akkor a hálózat élettartama lehet az első csomópont működésképtelenségéig eltelt idő, vagy akár az utolsó csomópont működésképtelenségéig eltelt idő. Ritka esetekben, amikor csomópontok nem egyenrangúak, akkor a magasabb prioritású csomópontokra érvényesek ezek a megállapítások. Az alkalmas metrika megválasztása alkalmazásfüggő.

3. Protokollok

Amint látható, a protokollok csoportosítása többféleképpen lehetséges. A táblázatokban felsoroltuk a létező jelentősebb útvonalválasztó protokollokat, ahol az 1. táblázat tartalmazza az egyes protokollok rendszerezését a hálózati modell és a kitűzött cél tekintetében, a 2. táblázat pedig ugyanezen protokollok működési modell szerinti osztályozását.

A modellek szerint megegyező protokollokat egyként kezeltük, összefoglaló nevük végére csillagot tettünk. Az egyes cellák tartalma a fentebb leírtak szerint értelmezhető, az egyetlen csillagot tartalmazó cella jelentése, hogy a protokoll az adott tulajdonság minden értékével rendelkezik. A következőkben a protokollokat aszerint osztjuk családokra, hogy a csomópontok hogyan választja következő csomópontot a továbbítandó csomag útján, és minden így nyert családból bemutatunk egy reprezentáns protokollt, amelyeket részletesebben is ismertetünk.

3.1. Következő csomópont választása az üzenet tartalma alapján

Ezek a protokollok csupán a kérésben szereplő kért adat alapján döntenek arról, hogy melyik csomópontnak továbbítsák a kérést. Ez a modell illeszkedik legjobban a szenzorhálózatokhoz, hiszen a bázis többnyire nem egy konkrét szenzort akar lekérdezni, hanem egy bizonyos adat iránt érdeklődik. Ezen protokollok, paradigmák közül a Directed Diffusion [2] protokollt ismertetjük röviden. A Directed Diffusion több más protokoll alapjául szolgált (Energy Aware Routing, GBR stb.), így az itt leírtak részben azokra is érvényesek.

A bázis kezdetben elárasztja kérésével az egész hálózatot, amely tartalmazza a kért adatot leíró attribútumérték párokat. A kérés vételekor a csomópontok gradienseket állítanak be a kérést küldő csomópontra. A forrás így az üzeneteket ezen gradiensek mentén továbbítja a bázis felé, egy üzenetet akár több gradiens (több szomszédnak) küldve. A paradigma jól alkalmazható nyomkövetési alkalmazásoknál, és csak a közvetlen szomszédok egyértelmű címzését követeli meg. A gradiens minden csomópontnál definiálja a keresett adatot, amire illik a keresett attribútumérték pár, és a következő csomópontot, akinek az adatot majd tovább kell küldeni a bázis felé. Minden gradienshez egy súly van rendelve, amely súly egyenesen arányos a rajta küldött adat mennyiségével. Kezdetben a közbelső

csomópontok több szomszédtól is kaphatnak egyező kérést, így ennek megfelelően több gradienst is beállítanak a bázis felé. Idővel a legjobb minőségű út mentén a bázis növeli a gradiensek súlyát, amivel ezen úton növeli a forgalmat, míg másokon csökkenti. A közbelső csomópontok aggregálják a küldött adatot, és az aggregátumot továbbítják a megfelelő gradiensek mentén a gradiens súlyával arányos sebességgel. A bázis periodikusan újraküldi a kérést, amivel életben tartja az empirikusan megfelelő minőségű útvonalakat. Az egyes csomópontok különböző cache technikákkal növelik a protokoll robusztusságát és teljesítményét. A protokoll hátránya, hogy a kezdeti elárasztás költséges, és az empirikusan megfelelő minőségű utak kiválasztásáig a hálózat energiafogyasztása magas.

3.2. Véletlenszerűen választott következő csomópont

A véletlenszerűen választott következő csomópont mögötti szándék elsősorban a robusztusság és a egyenlő terheléelosztás elérése, illetve megközelítése. Ezek a protokollok nagyban alapoznak arra a tényre, hogy a szenzorok homogén, egyenletesen elhelyezett eszközök. Ezek közül az Energy Aware Routing [5] protokollt ismertetjük.

Az Energy Aware Routing protokoll fő célja a hálózat élettartamának növelése az egyenlő terheléelosztás megközelítésével. A protokoll futása során a cél csomópont (bázis) kezdeményezi a kommunikációt, amikor arra igény van. A szenzorok igyekeznek mindig különböző szomszédot választani a továbbításra. A választás véletlenszerű, ahol egy adott szomszéd választásának a valószínűsége fordítottan arányos annak költségével. A költség függ az adott szomszéd aktuális maradék energiaszintjétől, illetve a hozzá történő üzenet továbbításához szükséges energiától. A protokoll a szomszédok listáját, és az elérésükhöz szükséges energiát a MAC rétegtől kapja.

A protokoll három fázisból áll: inicializálás, üzenet továbbítás, útvonal karbantartása.

I. Inicializálás

Korlátozott elárasztás segítségével minden csomópont felépíti az útvonalválasztó tábláját, vagyis minden csomópont megállapítja a lehetséges útvonalakat a cél felé, és azok költségét.

1. A cél elárasztja a hálózatot a forrás irányába egy kérés üzenettel, aminek a költségmezejét nullára állítja: $Cost = 0$
2. Minden közbelső csomópont csak azon szomszédjainak továbbítja a kérést, amelyek a forráshoz közelebb, a céltól viszont távolabb helyezkednek el. Formálisan N_i csak akkor küldi el a kérést N_j szomszédjának, ha a következő egyenlőtlenségek teljesülnek: $d(N_i, N_S) \geq d(N_j, N_S)$, $d(N_i, N_D) \leq d(N_j, N_D)$, ahol $d(N_i, N_j)$ az N_i és N_j csomópontok távolságát jelenti, és N_S a forrás, N_D pedig a célcsoomópont azonosítója.

Rendszer modell													
No.	Protokoll	Bázisállomás						Szenzor					Cél
		Szám	Mobilitás	Jelenlét	Lefedtettség	Lehelyezés	Átv. energia	Lefedtettség	Címzés		MAC felület		
									Kérés	Válasz			
1	Rumor Routing	Egy	Fix	*	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Nem valós idejű
2	MCFA	Egy	Fix	*	Részleges	*	*	Részleges	X	X	X	*	Nem valós idejű
3	Energy Aware Routing	Több	Fix	*	Részleges	*	Allítható	Részleges	Adat	ID	ID	Kell	Nem valós idejű, élettartam
4	Directed Diffusion	Több	Korlátozott	Folyamatos	Részleges	*	*	Részleges	Adat	ID	ID	*	Nem valós idejű
5	GBR	Több	Korlátozott	Folyamatos	Részleges	*	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
6	LEACH*	Egy	Fix	Folyamatos	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
7	TEEN	Egy	Fix	Folyamatos	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
8	APTEEN	Egy	Fix	Folyamatos	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Valós idejű, élettartam
9	PEGASIS	Egy	Fix	Folyamatos	Részleges	Véletlen	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
10	ACQUIRE	Több	Korlátozott	Folyamatos	Részleges	*	*	Részleges	X	ID/Pozíció	ID/Pozíció	*	Nem valós idejű, élettartam
11	IDSQ/CADR	Több	Fix	Folyamatos	Részleges	*	*	Részleges	Adat	ID	ID	*	Nem valós idejű
12	Greedy Face Routing*	Több	Mobilis	*	Részleges	*	*	Részleges	Adat	ID/Pozíció	ID/Pozíció	*	Nem valós idejű, élettartam
13	DIR/GEDIR/MFR*	Több	Korlátozott	*	Részleges	*	*	Részleges	Adat/Pozíció	ID/Pozíció	ID/Pozíció	*	Nem valós idejű, élettartam
14	GEAR	Több	Korlátozott	*	Részleges	*	*	Részleges	Adat/Pozíció	ID/Pozíció	ID/Pozíció	*	Nem valós idejű, élettartam
15	MECN	Egy	Fix	*	Részleges	*	Allítható	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
16	TTDD	Több	Mobilis	*	Részleges	*	*	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
17	SAR (DAM)	Több	Korlátozott	*	Részleges	*	*	Részleges	*	ID	ID	*	Nem valós idejű, élettartam
18	HPAR	Egy	Fix	*	Részleges	*	Allítható	Részleges	Adat	ID	ID	*	Nem valós idejű, élettartam
19	SPEED	Több	Fix	Folyamatos	Részleges	*	*	Részleges	Adat/Pozíció	ID/Pozíció	ID/Pozíció	*	Valós idejű, élettartam

1. táblázat Hálózati modell és célkitűzés

Rendszer modell												
No.	Protokoll	Köv. szom.	Kommunikációs minta			Hierarchia	Kézbesítés	Számítás	Állapot	Riportolási modell		
			Sz-Sz	Sz-B	B-Sz					Idővezérelt	Kérésvezérelt	Eseményvezérelt
1	Rumor Routing	Véletlen	X	Rev. M.	Anycast	Nincs	Egy / Egy	Decentr.	van	X	Nem-agg.	X
2	MCFA	Összes	X	Rev. M.	X	Nincs	Egy / Egy	Decentr.	nincs	*	X	*
3	Energy Aware Routing	Véletlen	X	Rev. M.	Anycast	Nincs	Több / Egy	Decentr.	van	*	*	X
4	Directed Diffusion	Adat	X	Rev. M.	Anycast	Nincs	Több / Több	Decentr.	van	*	*	X
5	GBR	Adat	X	Rev. M.	Anycast	Nincs	Több / Több	Decentr.	van	*	*	X
6	LEACH*	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	*	van	*	*	X
7	TEEN	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	*	van	X	*	*
8	APTEEN	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	*	van	*	*	*
9	PEGASIS	Hierarchia	X	Unicast	Anycast	Van	Egy / Egy	*	van	*	*	X
10	ACQUIRE	Véletlen	X	Rev. M.	Anycast	Nincs	Egy / Egy	Decentr.	van	X	Nem-agg.	X
11	IDSQ/CADR	Pozíció	Unicast	Rev. M.	Anycast	*	Egy / Egy	Decentr.	van	X	*	X
12	Greedy Face Routing*	Pozíció		Unicast		Nincs	Egy / Egy	Decentr.	nincs	Nem-agg.	Nem-agg.	Nem-agg.
13	DIR/GEDIR/MFR*											
14	GEAR	Pozíció	Unicast	Rev. M., Uni	Anycast	Nincs	Egy / Egy	Decentr.	van	Nem-agg.	Nem-agg.	Nem-agg.
15	MECN	Pozíció	X	Rev. M., Uni	Anycast	Nincs	Egy / Egy	Decentr.	van	*	X	*
16	TTDD	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	Decentr.	van	Nem-agg.	Nem-agg.	X
17	SAR (DAM)	Hierarchia	X	Rev. M.	Anycast	Van	Egy / Egy	Decentr.	van	X	*	*
18	HPAR	Hierarchia	X	Rev. M.	*	Van	Egy / Egy	Centralizált	van	*	*	*
19	SPEED	Pozíció	Uni, Any, Mult	Uni, Any, Rev. M.	Uni, Any, Mult	Nincs	Egy / Egy	Decentr.	nincs	Nem-agg.	Nem-agg.	Nem-agg.

2. táblázat Működési modell

3. A kérés vétele során N_j kiszámítja az N_j -n keresztül haladó útvonal költségét a következőképpen:

$$C_{N_j, N_i} = Cost + Metric(N_j, N_i),$$

ahol $Metric(N_j, N_i)$ az N_j és N_i közötti metrikát jelenti (lásd később).

4. A magas költségű útvonalakat N_j nem veszi figyelembe és eldobja a megfelelő kérést, csak az alacsony költségű csomópontokat adja hozzá az útvonalválasztó táblákhoz:

$$FT_j = \{i \mid C_{N_j, N_i} \leq \alpha(\min_k C_{N_j, N_k})\}$$

5. Minden N_j csomópont egy valószínűséget rendel minden szomszédjához az FT_j táblában:

$$P_{N_j, N_i} = \frac{1/C_{N_j, N_i}}{\sum_{k \in FT_j} 1/C_{N_j, N_k}}$$

6. N_j kiszámolja a célhoz vezető utak átlagos költségét (amelyek az FT_j útvonalválasztó táblában szerepelnek):

$$Cost(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} \cdot C_{N_j, N_i}$$

7. Ez az átlagos költség lesz a továbbítandó kérés új költségértéke: $Cost = Cost(N_j)$, majd az új kérést N_j továbbítja a forrás fele a 2. lépés szerint.

II. Üzenet továbbítás

1. A forrás elküldi az üzenetet a táblájában szereplő egy szomszédjának, ahol a szomszédot a hozzá rendelt valószínűséggel választja.
2. Minden közbenső csomópont hasonlóképpen választja ki a következő csomópontot a cél felé, vagyis választ egy szomszédot a táblájából a hozzá rendelt valószínűséggel.
3. Ez folytatódik addig, ameddig az üzenet eléri a célt (bázist).

III. Útvonal karbantartása

Az útvonal karbantartása minimális; nem túl gyakran a cél (bázis) elárasztással indirekt frissíti a szenzorok útvonalválasztó tábláját.

A metrikát, amit az 1.3 pontban használunk a következőképpen számoljuk ki: $C_{i,j} = e_{i,j}^\alpha R_i^\beta$, ahol $C_{i,j}$ az N_i és N_j csomópontok közötti költségmetrika, $e_{i,j}$ az egy üzenet átviteléhez szükséges energia N_i -ből N_j -be, R_i pedig N_i csomópont maradék energiája (az aktuális teleptöltöttség) normalizálva a csomópont indulási energiájával (kezdeti teleptöltöttség). Az α és β változtatható paraméterek, attól függően, hogy a hálózat élettartamát vagy a globális energiafogyasztást akarjuk minimalizálni.

A protokoll a Directed Diffusion paradigmához képest átlagosan 21,5%-kal több energiát spórol, és 44%-kal növeli a hálózat élettartamát (ha a hálózat élettartamát az első szenzorcsomópont kimerüléséig eltelt idővel számítjuk). Hátránya az egyes csomópontok esetleges komplikált címzése, illetve az inicializálási fázis megnövekedett kommunikációs költsége a Directed Diffusion protokollhoz képest.

3.3. Következő csomópont választása geometriai információ alapján

Az útvonalválasztó protokollok ezen családja a szomszédok ismert geometriai pozíciói és a cél pozíciója alapján döntenek a következő csomópontról a cél felé, ami lehet egy régió vagy egy konkrét csomópont. Ezzel elkerülük az elárasztás okozta többletterhelést, viszont többletköltséget jelenthet a pozíciók megállapítása, ami történhet statikusan, előreprogramozott módon, vagy valamilyen pozíció-meghatározó rendszer segítségével (pl. GPS). Többnyire mind a kérés, mind pedig a válasz útvonalirányítására ugyanazt a technikát használják. A legtöbb decentralizált pozíció alapú protokoll közös problémája a lokális minimum feloldása, amikor minden szomszédos csomópont távolabb helyezkedik el a céltől mint maga a továbbító csomópont. Ezen probléma feloldására különböző technikákat használnak az egyes protokollok. Itt most a GEAR (Geographical and Energy Aware Routing) [6] protokollt ismertetjük vázlatosan.

A GEAR esetében a csomópontok csak bizonyos szomszédoknak továbbítják a kérést, így több energiát spórolnak meg mint a Directed Diffusion. A válasz továbbítása történhet vagy a kéréshez hasonló módon, vagy pedig a Directed Diffusion esetében leírt módon. Minden csomópont a célcsomópontra (forrás) nyilvántart egy becsült és egy tanult költségértéket. Az utóbbi felelős a lyukak megkerüléséért és a becsült költségérték finomításának tekinthető. Ha az útvonalban nincsenek lyukak (minden csomópontnak van a célhoz közelebb eső szomszédja), akkor a két költségérték megegyezik. A protokoll két fázisból áll: üzenet továbbítás a célterület felé, valamint üzenet továbbítás a célterületen belül.

I. Üzenet továbbítás a célterület felé

Egy közbenső N csomópont a célterülethez közelebb eső szomszédokból választja ki azt a következő csomópontot a cél felé, amelynek tanult költsége a legkisebb. Ha ilyen nem létezik, akkor azt a szomszédot választja az összes közül, amelynek becsült költsége a legkisebb. Kezdetben a tanult költség a célterületre megegyezik a becsült költséggel, ahol a becsült költség formulája a következő:

$$c(N_i, R) = \alpha d(N_i, R) + (1 - \alpha)e(N_i),$$

ahol $d(N_i, R)$ a szomszédos N_i csomópont távolsága a célterület középpontjától, $e(N_i)$ az N_i csomópont normalizált maradék energiája, α pedig egy állítható paraméter. Minden csomópont kiszámítja saját becsült költségét, majd elküldi minden szomszédjának, így minden csomópont értesül minden szomszédjának a becsült költségéről. Kezdetben a tanult költség minden N_i csomópontra egy adott célterületre $h(N_i, R) = c(N_i, R)$. Mivel egy N csomópont elküldte az üzenetet a minimális tanult költségű szomszédjának aminek az azonosítója N_{min} , frissíti a saját tanult költségét a következő módon:

$$h(N, R) = h(N_{min}, R) + C(N, N_{min}),$$

ahol $C(N, N_{min})$ az üzenet továbbításának költsége N -ből N_{min} -be (lehet a küldéshez használt energia mennyisége, a távolság, a normalizált maradék energia,

illetve ezek kombinációi). Így, ha az utat n csomópont alkotja, akkor az útvonal tanult költsége n lépésben konvergál az útvonal valódi költségéhez. A csomópontok a tanult költségük értékét bizonyos időközönként elküldik az összes szomszédjuknak. Így a tanult költségek használata a megfelelő frissítési technikával lehetőség adnak a lyukak kikerülésére.

A GEAR nem csupán hosszabb élettartamot biztosít a szenzorhálózatoknak mint más pozíció alapú útvonaválasztó protokollok, de a kézbesített csomagok száma akár 80%-kal több lehet, mint más pozíció alapú protokolloknál.

II. Üzenetovábbítás a célterületen belül

Miután az üzenet elérte a célterületet, az ottani csomópontok korlátozott vagy pedig rekurzív elárasztásos technikát használnak az üzenet elterjesztésére a területen belül. Ha a csomópontok ritkán helyezkednek el, akkor a korlátozott elárasztás, míg sűrűn elhelyezett csomópontok esetén a rekurzív elárasztásos technika javasolt. Az utóbbi esetben a területet négy, körülbelül egyenlő alterületre osztjuk, és minden alterületre továbbítjuk az üzenet egy másolatát. Ez a felosztásos folyamat rekurzív módon addig folytatódik, amíg csak egy csomópont marad egy alterületen belül.

3.4. Hierarchia szint alapján választott következő csomópont

A hierarchikus protokollok esetében minden csomópont a hierarchiában fentebb elhelyezkedő csomópontnak (aggregátor) küldi az üzenetet, ahol a hierarchia csúcsán a bázis helyezkedik el. Az egyes csomópontok a bejövő üzeneteket aggregálás után küldik tovább, amivel jelentősen csökkenthetik az adatforgalmat, így energiát spórolnak meg, aminek eredménye a hálózat megnövekedett élettartama. A hierarchikus protokollok másik nagy előnye, hogy jól skálázhatóak.

Az egy aggregátornak továbbító csomópontok halmazát klaszternek, míg az aggregátort klasztervezetőnek is nevezik. Klasztervezetőknek mindig olyan csomópontokat választanak (statikusan vagy dinamikusan), amelyek nagyobb erőforrással (általában nagyobb teleptöltöttséggel) rendelkeznek, mivel a többi csomópont-hoz képest nagyobb forgalmat bonyolítanak le és több számítás végeznek. A hierarchikus útvonalválasztó protokollok alappillére az klaszterek kialakítása és a klasztervezető megválasztása, majd erre épülve az útvonalválasztás megvalósítása a klasztervezetők felé. A következőkben a *LEACH (Low Energy Adaptive Clustering Hierarchy)* [3] protokollt vázoljuk fel.

A LEACH protokollban a csomópontok dinamikusan, elosztott módon alakítják ki a klasztereket. A LEACH futásakor minden klasztervezető véletlenszerűen választódik meg, és ez a szerep dinamikusan rotálódik az egyenletes terheléselosztás végett. Minden klasztervezető a klaszterből érkező üzeneteket aggregálja, majd az aggregátumot közvetlenül elküldi a bázisnak. Így a LEACH feltételezi, hogy minden csomópont ké-

pes elérni a bázisállomást. A LEACH TDMA/CDMA csatornahozzáférési modellt követ a klaszteren belüli, illetve klaszterek közötti ütközések feloldására. Szimulációk szerint a csomópontok mintegy 5%-ának kell csak klasztervezetőnek lennie egy adott pillanatban a hálózat optimális energiafogyasztásához. A LEACH működése két fázisból áll, amelyek periodikusan ismétlődnek, a két fázis együtt alkot egy menetet.

I. Felépülési fázis

A felépülési fázis során létrejönnek a klaszterek és megválasztódnak a klasztervezetők. A csomópontok egy előre definiált p (pl. 0,05) hányada választja meg magát minden menet elején klasztervezetőnek a következőképpen. Egy n szenzor választ magának egy véletlen számot 0 és 1 között, és ha ez szám kisebb mint egy $T(n)$ küszöbérték, akkor a szenzor klasztervezetőnek deklarálja magát, ahol

$$T(n) = \frac{P}{1 - p \cdot (r \bmod 1 / p)} \quad \text{ha } n \in G.$$

Itt r jelöli az aktuális menetszámot, G pedig azon csomópontok halmaza, amelyek még nem voltak klasztervezetők az elmúlt $(1/p)$ menet során. Ezek után a magukat vezetőnek deklaráló csomópontok értesítik a körülöttük levő csomópontokat a választás eredményéről egy bizonyos energiaszinten küldött üzenetben. Minden csomópont azt a csomópontot választja saját klasztervezetőjének, akitől a legnagyobb jelerősséggel kapta az üzenetet, majd erről informálja a választott vezetőt is. Ezután a vezető létrehozza a megfelelő TDMA ütemezést, ahol minden hozzá tartozó csomópontnak lefoglal egy időrészt, majd az ütemezési információt elküldi a klasztertagoknak.

II. Működési fázis

A működési fázis során a klaszterek tagjai a mért adatot a vezetőnek továbbítják, amely, miután minden adatot megkapott, az aggregátumot továbbküldi a bázisnak. A menet végén a hálózat átvált felépülési fázisba és új klaszterek alakulnak.

A működési fázis célszerűen jóval nagyobb mint a felépülési fázis a túlterheltség csökkentése végett. A protokoll hátránya, hogy nem alkalmazható nagy területen elhelyezett hálózatnál, hiszen minden csomópontnak el kell érnie a bázist közvetlenül, továbbá feltételezi, hogy minden csomópont folytonosan küld adatot a vezetőnek. Továbbá nem feltételen igaz az, hogy a vezetők a hálózatban egyenletesen helyezkednek el, így néhány csomópontnak lehetséges, hogy nem lesz vezetője. A protokoll feltételezi, hogy minden csomópont egyenlő kezdeti energiával rendelkezik. A legnagyobb gondot mindezek ellenére talán mégis a felépülési fázis okozta többletterhelés jelentheti.

3.5. Minden szomszédnak történő továbbítás

Ezen protokollok működése rendkívül egyszerű. Minden csomópont a hálózatban önmaga dönt arról, hogy a kapott üzenetet továbbítania kell-e vagy sem. Ha

igen, akkor minden szomszédjának elküldi az üzenetet (egyetlen többszörös üzenetben), ha nem, akkor pedig eldobja az üzenetet. Ezen protokollok képviselője az *MCFA (Minimal Cost Forwarding Algorithm)* [4] protokoll. Az MCFA protokoll nagy előnye, hogy semmilyen információt sem tárolnak a csomópontok a szomszédjaikról, csak a saját maguk költségét a bázishoz viszonyítva.

A protokoll két részből áll. Az első részben minden csomópont megállapítja a saját költségét a bázishoz képest, aminek érdekében a bázis elárasztja a hálózatot egy kezdetben $C = 0$ értékű költséget tartalmazó üzenettel. Kezdetben minden csomópont költsége végtelen. Minden N_i csomópont, amely megkapja ezt az üzenetet egy N_j csomóponttól, vár $\alpha \cdot C_{N_i, N_j}$ ideig (α választásánál figyelembe kell venni a csomópontok közötti kapcsolatok késleltetését, hibáit, a csomópontok késleltetését stb.), ahol C_{N_i, N_j} az N_i és N_j csomópontok közötti kapcsolat költsége (energia, késleltetés stb.), majd frissíti az üzenetben szereplő C értéket: $C = C + C_{N_i, N_j}$, beállítja a saját költségét erre az új C értékre, és továbbítja a frissített üzenetet. Bebizonyítható, hogy ideális esetben (amikor α elég nagy) minden csomópont csak egyetlen ilyen üzenetet küld tovább, ami a csomópont minimális költségét tartalmazza a bázistól.

A protokoll második részében minden csomópont képes már üzenetet küldeni a bázis felé a következő módon. A N küldő szenzor elhelyezi a saját C_N minimális költségét a küldendő üzenetbe, és többesküldéssel elküldi azt szomszédainak. Minden M csomópont, amely hallja az üzenetet, ellenőrzi, hogy $M - C_{N, M} = C_M$. Ha igen, akkor M a minimális költségű útvonalon van, ezért többesküldéssel továbbküldi az üzenetet. Egyébként M eldobja az üzenetet.

A protokoll hátránya, hogy minden olyan csomópont, amely vesz egy üzenetet, azt mindenképpen értelmezi is, ami többletterhelést jelent.

4. Összegzés

Jelen írásunkban a vezeték nélküli szenzorhálózatokban alkalmazott útvonalválasztó protokollokat tekintettük át röviden. A cikk első felében bemutattuk a létező útvonalválasztó protokolloknak egy új osztályozási szempontrendszerét. A protokollokat rendszereztük hálózati modelljük, működési modelljük és célkitűzésük szerint. Táblázatokba rendezve felsoroltuk a létező jelentősebb útvonalválasztó protokollokat hálózati modell és cél, valamint működési modell szerint. A cikk második felében részletesen bemutattuk néhány konkrét protokoll működését. A bemutatott protokollok egy-egy protokollcsalád képviselői voltak, ahol a családot az határozta meg, hogy a csomópontok hogyan választják a következő csomópontot a cél felé vezető úton.

Köszönetnyilvánítás

Az itt bemutatott munkát részben támogatta a HSN Laboratórium, a UbiSec&Sens EU projekt (www.ist-ubisecsens.org), és az OTKA (T046664).

Irodalom

- [1] J. N. Al-Karaki, A. E. Kamal:
Routing techniques in wireless sensor networks: a survey.
In IEEE Wireless Communications, Vol.11, 2004.
pp.6–28.
- [2] C. Intanagonwiwat, R. Govindan, D. Estrin:
Directed Diffusion: a scalable and robust communication paradigm for sensor networks.
In Proceedings of ACM MobiCom 2000, Boston, MA, pp.56–67.
- [3] W. Heinzelman, A. Chandrakasan, H. Balakrishnan:
Energy-Efficient Communication Protocol for Wireless Microsensor Networks.
In Proc. of the 33rd Hawaii International Conference on System Sciences (HICSS 2000), January 2000.
- [4] F. Ye, A. Chen, S. Liu, L. Zhang:
A scalable solution to minimum cost forwarding in large sensor networks.
In Proc. of the 10th International Conf. on Computer Communications and Networks (ICCCN 2001), pp.304–309.
- [5] C. Rahul, J. Rabaey:
Energy Aware Routing for Low Energy Ad Hoc Sensor Networks.
In IEEE Wireless Communications and Networking Conference (WCNC), March 17-21, 2002, Orlando, FL, Vol.1, pp.350–355.
- [6] Y. Yu, D. Estrin, R. Govindan:
Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks.
UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.

Mobil eszközök alkalmazása szenzorhálózatokban

VINCZE ZOLTÁN, VIDA ROLLAND

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
{vincze,vida}@tmit.bme.hu

Kulcsszavak: vezeték nélküli szenzorhálózat, mobilitás, nyelőállomás, energiahatékonyság

A hálózati kommunikáció egyik jelenleg legtöbbet kutatott területe a vezeték nélküli szenzorhálózatok témaköre. A hálózat építőkövei a különböző fizikai paraméterek mérését végző szenzorok, és a mérési adatokat begyűjtő nyelőállomások. A szenzorok egyik legfontosabb tulajdonsága az erősen korlátozott energiatartalékuk. Ezt figyelembe véve, az eredeti sztatikus architektúráis elképzelések újragondolásával a hálózattervezők egyre több olyan megoldást javasolnak, amelyekben mind a szenzorok, mind a nyelők mobilitása megengedett. Az építőelemek mozgatása lehetővé teszi a hálózat energiahatékonyságának és egyéb funkcióinak további javítását. Ebben a cikkben bemutatjuk a legfontosabb eddigi javaslatokat a mobil eszközök szenzorhálózatokban való alkalmazására.

1. Bevezetés

A hálózati kommunikáció egyik jelenleg legtöbbet kutatott területe a vezeték nélküli szenzorhálózatok témaköre. A hálózat építőkövei a szenzorok, amelyek képesek fizikai jelenségek mérésére (például hőmérséklet, fény, páratartalom) és a mért adatok rádiós továbbítására egy speciális építőelem, a *nyelő* felé. Ez az eszköz kapcsolattal rendelkezik a külvilág felé, így az összegyűjtött adatokat hozzáférhetővé teszi a hálózat üzemeltetői számára. Mindemellett jóval nagyobb akkumulátorral rendelkezik mint a szenzorok, így működése során nem kell akkora figyelmet fordítani az energiafelhasználására (1. ábra).



1. ábra Mica2 és Mica2Dot típusú szenzorok

A szenzorhálózatok három kategóriába sorolhatók aszerint, hogy a szenzorok mikor küldik el mért adataikat. *Idővezérelt (time-driven)* hálózatokban minden szenzor periodikusan elküldi adatait, egy *eseményvezérelt (event-driven)* hálózatban csak azok küldenek adatot, akik egy eseményt érzékeltek, míg egy *lekérdezésvezérelt (query-driven)* hálózatban a nyelő kérésére küldenek csak adatot a szenzorok.

Az információ továbbításához felhasznált kommunikációs elv szerint is csoportosíthatjuk a hálózatokat, a következőképpen: *egyugrásos (single-hop)* esetben minden szenzor közvetlenül a nyelővel kommunikál, míg *többesugrásos (multi-hop)* esetben a nyelőtől távolabb lévő eszközök a társaikon keresztül juttatják el az adatokat a célhoz.

Kicsi méretük miatt a szenzorok energiatartaléka erősen korlátozott, ezért a hálózat tervezése és működése során különös figyelmet kell szentelni az energiafelhasználásra; a több ezer érzékelőből álló hálózatokban nem gazdaságos, illetve a telepítés helyszíne miatt esetleg nem is lehetséges az energia utánpótlása. Ez a tulajdonság motiválta a kutatókat arra, hogy számos olyan megoldást javasoljanak, melyek a hálózatok energiahatékony működését segítsék (például energiahatékony útvonalválasztás, topológia kontroll, alvászabályozás, klaszterezés, adataggregáció). A kezdeti elképzelések szerint mind a szenzorok, mind a nyelő sztatikus eszközök voltak, az utóbbi időben azonban egyre több olyan javaslat lát napvilágot, amelyek ezen eszközök mozgatásának bevezetését tanácsolják a működés hatékonyságának javítására. A következőkben összefoglaljuk az eddig kidolgozott legismertebb, mobilitást alkalmazó javaslatokat.

2. Szenzorok mobilitása

A szenzorok energia utánpótlása erősen korlátozott, az érzékelés–kommunikáció–mozgás hármas közül pedig az utóbbi igényli messze a legtöbb energiát, így a szenzorok mobilitása a hálózatban általában csak korlátozott számú érzékelőegységre terjed ki (ezek rendelkezhetnek nagyobb akkumulátorral is). Az érzékelők mozgatása során a cél sokszor nem a hálózat energiafelhasználásának a csökkentése, hanem a hálózat által felügyelt terület lefedettségének növelése, javítása.

A szenzorok elhelyezkedésének megváltoztatása két fő okból történhet: az egyik esetben a kezdeti telepítés után, vagy a működés közben lemerülő szenzorok miatt lefedetlenül maradt terület lefedése, a másik esetben pedig a kommunikációval jobban terhelt területeken elhelyezkedő társaik tehermentesítése a cél.

Az első cél elérésére javasolt protokollban a szenzorok közvetlen a telepítés után mozognak csak, a hálózat későbbi működése során már nem [1]. A mozgás célja a telepítés után lefedetlenül maradt területek lefedése. Feltételezik, hogy a szenzorok ismerik geográfiai koordinátáikat, például a GPS rendszernek, vagy más a szakirodalomban javasolt energiahatékonyabb lokalizációs algoritmusnak köszönhetően. A protokoll szerint a telepítés után minden szenzor elküldi szomszédainak saját koordinátáit, majd ezen információ alapján az érzékelők kiszámítják a hozzájuk tartozó Voronoi tartományokat (azon pontok halmaza, amelyek az adott érzékelőhöz vannak legközelebb [2]). Amennyiben a szenzor érzékelési sugara kisebb, mint a Voronoi tartománya valamely pontjának tőle mért távolsága, úgy ott a lefedettségben rés van, hiszen ő az ahhoz ponthoz legközelebbi érzékelő, de neki is a hatókörén kívül esik.

A szerzők három lehetséges iteratív mozgatási stratégiát javasolnak a rések lehető legjobb mértékű lefedésére: a *VEC (VECTor-based algorithm)* esetében a szenzorok elmozognak azokról a helyekről, ahol túl sűrűen vannak telepítve, így biztosítva az egyenletesebb eloszlást. A második stratégia a *VOR (VORonoi-based algorithm)*, amely során a szenzorok a lyukak felé mozognak; ugyanezt teszik a harmadik javaslatban, a *Minimax*-ban is, de itt konzervatívabb módon hagyják el korábbi helyüket, ügyelve arra, hogy az elmozdulással ne generáljanak újabb lefedettségi réseket. Mindegyik stratégia esetén a mozgatás körökre van osztva: egy körben egy szenzor virtuálisan vagy ténylegesen mozoghat. Az előbbi esetben a szenzorok nem mozdulnak el ténylegesen, csak a szomszédoknak küldött információt módosítják az új virtuális helyüknek megfelelően. Amennyiben azonban a virtuális helyük már olyan messze kerül a tényleges pozíciójuktól, hogy nem tudnak új szomszédokkal kommunikálni, akkor fizikailag is elmozdulnak. Az érzékelők mozgása akkor fejeződik be, amikor már nem növelhető tovább a lefedettség.

Sokszor előfordulhat, hogy a telepítés előtt még nem ismert az események várható eloszlása a hálózat területén, így célszerű kezdetben egyenletes térbeli eloszlás alapján lerakni a szenzorokat. A megfigyelt esemény lehet például egy állatcsorda vonulása, ilyenkor célszerű lenne, ha a vonulási útvonal környékén sűrűbben helyezkednének el a szenzorok, hiszen így részletesebb információt tudnak küldeni az eseményről, illetve az információ továbbításában több érzékelő tudna részt venni, ezáltal jobban elosztva a kommunikációs terhelést. E kihívás leküzdése is lehetséges mozgó szenzorok segítségével [3].

A javasolt algoritmus feltételezi, hogy minden szenzor ismeri saját térbeli elhelyezkedését, ezenfelül pedig

a bekövetkező események helyéről a hálózat összes szenzora értesül. Az érzékelők kétféle algoritmus alapján mozoghatnak, az első esetben a szenzor a saját korábbi helyeinek és az események jelenlegi helyének függvényében határozza meg, hogy merre lép. A második esetben a szenzor az előbb felsorolt adatokon kívül az esemény korábbi helyeit – amit egy eloszlás függvény segítségével tárol – is figyelembe veszi következő helyének megállapításánál. Amennyiben az összes szenzor az események gyakori felbukkanási helyére mozogna, akkor a hálózat bizonyos részei lefedetlenek maradnának. Ezt elkerülendő, miután a szenzorok elmozdultak korábbi helyükről, folyamatosan felderítő üzeneteket küldözgetnek szomszédaiknak. Figyelik a beérkező válaszüzenetek irányzatszokeiteit, és ha valamelyik térrészből nem kapnak ilyen üzenetet, akkor azon rész felé mozdulnak el, így biztosítva a lefedettséget ott is.

3. A nyelők mozgatása

A szenzorhálózatokban információt összegyűjtő nyelők speciális eszközök, amelyek a szenzorokhoz képest sokkal bővebb energiatartalékkal rendelkeznek, így mozgatásuk során általában nem kell törődni a mozgásra fordított energiával. Mozgatásuk sokféleképpen megoldható: felszerelhetőek robotokra, vezetől nélküli járművekre, repülőkre, állatokra. A mobilitás célja kettős: egyrészt ilyen módon csökkenthető a szenzorok által kommunikációra fordított energia nagysága, másrészt egyenletesebbé tehető az energiafogyasztás eloszlása a hálózatban.

Amennyiben egy vezetől nélküli kommunikációt használó eszköz d távolságra szeretne eljuttatni egy csomagot, akkor az ehhez szükséges energia nagysága arányos d^α -al, ahol α értéke 2 és 5 között mozoghat a jelterjedési viszonyoktól függően. Látható, hogy a nyelő közelítése az adatot küldő szenzorokhoz jelentős energiát képes megtakarítani a kommunikációs távolságok csökkentésével. Egyugrásos hálózatok esetén a nyelőtől távolabbi eszközöknek több energiát kell adattovábbításra fordítani, mint a közelebbi társaiknak, míg többesugrásos hálózatok esetében pont fordított az energiafogyasztás megoszlása, a nyelő körüli szenzorok terhelése jóval nagyobb a távolabbiakhoz képest, mivel rajtuk gyakorlatilag a hálózat teljes forgalma átfolyik. A nyelő mozgatása erre a problémára is gyógyír lehet.

Számos javaslat született már a nyelők mozgatására a tudományos szakirodalomban, ezek a használt mozgási modell alapján három fő kategóriába sorolhatók. A *véletlen (random) mozgás* során a nyelő egy véletlen útvonalat jár be, jövőbeli viselkedése nem jósolható meg; mindez alkalmazhatatlanná teszi olyan hálózatokban, ahol az információ továbbítására éles időbeli korlátok vannak. A *jósolható (predictable) mozgás* esetén a nyelő egy előre kijelölt pályán mozog, így korlát adható arra az időtartamra, amikor újra egy adott szen-

zor közelébe ér. A harmadik mozgás modell a *vezérelt (controlled) mozgás*; ezesetben a nyelő mozgását vezérlő protokoll a hálózat aktuális állapotát is figyelembe veszi.

3.1. Véletlen mozgás

MULE (Mobile Ubiquitous LAN Extension) eszköz bevezetését javasolták olyan hálózatok esetére, ahol a mért adatok továbbítása nem kell, hogy valós időben történjen [4]. Másrészt a szenzorok annyira ritkán vannak telepítve, hogy nem tudják egymáson keresztül eljuttatni a mért adatokat a nyelőhöz, mert vagy nincs egyetlen szomszédja sem egy szenzornak, vagy több független, egymással kommunikálni nem képes csoportra oszlanak az eszközök.

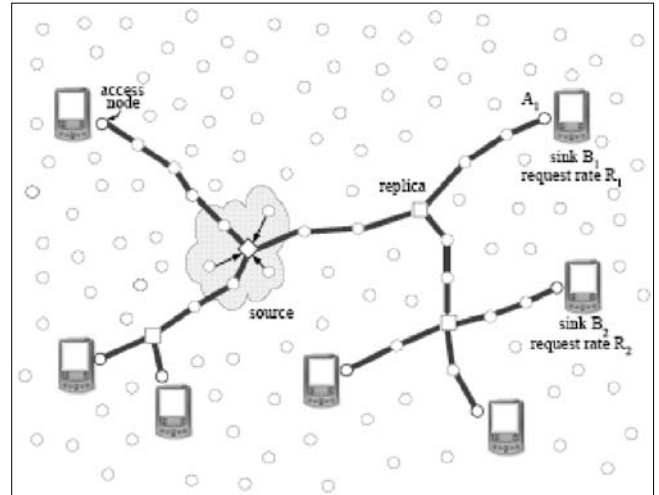
Ekkor az információk begyűjtéséhez több bázisállomást, vagy nagyobb mennyiségű szenzort kellene felhasználni, ami megnövelné a költségeket. Ez elkerülhető a *MULE* használatával, amely a nyelőt magával hordozó mobil eszköz, mozgása pedig egy véletlen bolyongást követ. Amikor egy szenzor közelébe ér, a szenzor átadja neki a mért adatokat, majd a mobil eszköz tovább folytatja útját. Az összegyűjtött információt fix hozzáférési pontoknak továbbítja, amint azok közelébe ér. Egy ilyen megoldás alkalmazható például arra, hogy egy adott területen a közlekedésmérnökök az egyes kereszteződések forgalmának nagyságát mérhessék. A teljes terület lefedése szenzorokkal felesleges lenne, elég helyette csak a kereszteződésekbe telepíteni szenzorokat, az adatokat begyűjtő nyelő pedig felszerelhető akár egy városi buszra is.

A *SENMA (SEnsor Networks with Mobile Agents)* architektúra is véletlenszerűen mozgó nyelőt alkalmaz az adatok begyűjtésére [5], de ellentétben az előző esettel itt a szerzők sűrűn telepített szenzorhálózatot feltételeznek. A nyelő a hálózat felett repül körbe-körbe, véletlen bolyongást követve; egy szenzor akkor kommunikál vele, amikor az éppen felette repül el. Ezesetben két okból is csökken az energiafelhasználás az adattovábbítás során. Egyrészt minden szenzor közvetlen a nyelővel kommunikál, így az adatnak nem kell több csomóponton áthaladnia. Másrészt a jel csillapítása a levegőben, függőleges irányban, kisebb mint a földi, vízszintes továbbítás esetén, ezért a kommunikáció folyamán a szenzoroknak elég kisebb erősségű jelet sugározniuk.

Azzal a kérdéskörrel, hogy hogyan lehet hatékonyan eljuttatni az információt egy adott szenzortól több mozgó nyelőhöz Kim és Abdelzaker foglalkozik [6]. A hálózatban a szenzorok ismerik a földrajzi helyüket és, ellentétben az eddigi feltételezett hálózatokkal, a nyelőkkel többesugrásos módon kommunikálnak. Egy esemény bekövetkezése után az azt érzékelő szenzornak a feladata elküldeni adatait az összes nyelőállomásnak. Ezen nyelők mindegyike véletlen bolyongás szerint mozog (2. ábra).

A javasolt *SEAD (Scalable Energy-efficient Asynchronous Dissemination)* protokoll a hálózat teljes eláraszta helyett kiépít egy adattovábbítási fát mindegyik

nyelő és a szenzor között. Kezdetben egy nyelőt a fában a fizikailag mellette elhelyezkedő szenzor reprezentál amennyiben a nyelő elmozdul, akkor értesíti az új helyéről ezt a szenzort, így az tudni fogja a nyelő új pozícióját, ezáltal képes továbbra is kézbesíteni az üzeneteket. Ennek következtében nem kell újraépíteni a fát a nyelő minden egyes elmozdulásánál. Amikor a kijelölt szenzor és a nyelő közötti távolság elért egy előre megadott küszöböt, akkor a fát újraépíti a protokoll.



2. ábra Adatok eljuttatása több mobil nyelőhöz

3.2. Jósolható mozgás

Ezen típusú mozgás esetén a nyelő egy előre meghatározott pályán mozog, kiszámíthatóvá téve jövőbeli viselkedését. Chakrabarti és társa véletlenszerűen telepített szenzorok alkotta hálózat minden elemét érintő útvonalon periodikusan körbe mozgó nyelő alkalmazását javasolják [7]. A nyelő állandó sebességgel halad úgy, hogy minden szenzor kommunikációs hatósugarát érintse az úton; egy szenzor csak ezen jól meghatározott időintervallum alatt küldheti el neki a mért adatokat. Ha a nyelő túl kevés ideig tartózkodik a szenzor szomszédságában, akkor az adatok egy része elveszik. Minden szenzor azonos sebességgel generálja az elküldendő adatokat, illetve azonos sebességgel képes továbbítani azokat.

A szerzők sorbanállási elmélet alkalmazásával vizsgálják, hogy adott adatvesztési valószínűséghez hogyan kell megválasztani a szenzorok adótávolságát, illetve az adattovábbítás sebességét. A vizsgálatokat két részre osztják. Az első esetben a szenzorok teljesen véletlenszerűen vannak telepítve. A második esetben is véletlenszerű a telepítés, de a szenzorok közötti távolság nem lehet kisebb egy küszöbértéknél. Míg az első esetben nehéz elkerülni az adatvesztést, a második esetben a küszöbérték megfelelő beállítása mellett garantálható az összes adat megérkezése a nyelőhöz.

Luo és társa egy sűrűn telepített, erősen összefüggő, kör alakú hálózatot feltételez, ahol a működés idővezérelt és a szenzorok többesugrásos módon kommunikálnak a nyelővel, egy legrövidebb út algoritmust al-

kalmazva [8]. Analitikus vizsgálatokon keresztül elemzik, hogy egy ilyen hálózatban mekkora a nyelő körüli szenzorok terhelése, a kapott eredmények pedig azt mutatják, hogy a nyelő környékén kialakuló nagy terhelés elosztása érdekében annak mozognia kell a hálózatban. A mozgás útvonalának olyan utat keresnek, amely periodikus és szimmetrikus a kör középpontjára. Az általuk használt analitikus modell segítségével bizonyítják, hogy az ilyen típusú utak közül a legjobb választás az, ha a nyelő a hálózat szélén mozog körbe. A bevezetett mobilitás jobb kihasználása érdekében új útvonalirányítási megoldást is javasolnak, melynél azt a ténytet használják ki, hogy hálózat szélén elhelyezkedő szenzorok kommunikációs terhelése jóval kisebb a középpontban lévőkhöz képest. A nyelő egy a középponthoz közelebbi körön mozog körbe, amelyen belül az adattovábbítás továbbra is legrövidebb út algoritmus alapján történik. A körön kívüli részre viszont a szerzők javaslata a „round routing”: a szenzor által küldött csomag egy köríven halad, párhuzamosan a nyelő pályájával, majd amikor a nyelővel egy magasságba ér, áttér a legrövidebb útra.

Erősen partícionált ad-hoc hálózatokbeli adattovábbításra ad megoldást Zhao és Ammar [9]. A partíciók között egy „komp” mozog előre megadott útvonalon, biztosítva az egyes részek közötti üzenetátvitelt. A szerzők feltételezik, hogy ismert a hálózat alkotóelemeinek elhelyezkedése és a köztük folyó forgalom nagysága. Az útvonal kialakításánál három szempontot vesznek figyelembe: minden eszköz tudjon közvetlenül kommunikálni a komppal, legyen minimális az üzenetek késleltetése, és legyen maximális az átvitt adatmennyiség. Ezen feltételek teljesítése NP nehéz, így heurisztikákat javasolnak az útvonal kialakításához: először az utazó ügynök problémájának közelítő megoldásával létrehoznak egy kezdeti útvonalat. Következő lépésként egy másik heurisztika segítségével a késleltetést csökkentendő módosítják a bejárás sorrendjét, majd az utolsó lépésben az adatátvitel mennyiségét maximalizálják egy lineáris programozási feladat segítségével. Bár ezen javaslat nem speciálisan szenzorhálózatok esetére született, jól használható ott is, amennyiben a hálózat erősen partícionált.

Az eddigi jószolható mozgások mind ciklikus útvonalat használtak. Ettől eltérően, Wang és társai idővezérelt, többesugrásos kommunikációt használó szenzorhálózatot feltételeznek, amelyben a szenzorok egy négyzetrács pontjaiba vannak telepítve és a nyelő is a rács mentén mozoghat csak [10]. Megbecsülik a hálózatban fellépő forgalom nagyságát és lineáris programozási modell segítségével meghatározzák, hogy a nyelő melyik rácspontra mennyi ideig tartózkodhat a hálózat lemerüléséig. Egy pontot azonban csak egyszer látogathat meg, a mozgása tehát nem lesz ciklikus.

3.3. Vezérelt mozgás

A három típusú mozgás közül ez a legintelligensebb: a nyelő mozgását vezérlő algoritmus a hálózat

aktuális állapotát figyelembe véve alakítja a bejárt útvonalat. Az algoritmusok általában két paraméter változtatásával módosítják a bejárást: fix útvonal esetén a mozgás sebességét módosítják, ellenkező esetben pedig a mozgás irányát és sebességét változtatják. Az első típusra példa az AIMMS (*Autonomous Intelligent Mobile Micro-server*) protokoll [11,12].

A véletlenszerűen telepített, többesugrásos kommunikációt használó hálózatban egy „mikroszerver” mozog körbe egy előre meghatározott útvonalon (3. ábra); feladata a mérési adatok begyűjtése a szenzoroktól. Az útvonal szomszédságában elhelyezkedő érzékelők közvetlenül a szerverrel kommunikálnak, míg a távolabbi szenzorok multi-hop kommunikációt használnak. A szerver t idő alatt ér körbe az útvonalán maximális sebességgel haladva. Amikor körbe ért, átadja a begyűjtött adatokat a nyelőnek. Az adatokat legfeljebb T idő késleltetéssel kell begyűjteni, ezért $t < T$ esetén lehetőség van rá, hogy a szerver bizonyos helyeken lelassítson. Erre szüksége lehet, hiszen előfordulhat, hogy egyes helyeken sűrűbben helyezkednek el a szenzorok, így ott a közeghozzáférési nehézségek miatt lassabb lehet a kommunikáció. Ugyancsak lassításra van szükség ott, ahol a rossz jelterjedési viszonyok miatt csak lassabb adatátvitel lehetséges.



3. ábra A „mikroszerver” pályája egy erdőben telepített szenzorhálózatban

Gandham és társai is több mozgó nyelő alkalmazását javasolják többesugrásos kommunikációt használó, idővezérelt hálózat esetén [13]. Minden szenzor azonos mennyiségű adatot küld, így a forgalmi minta ismert, az érzékelők elhelyezkedését pedig a MAC protokoll segítségével állapítja meg a javaslat. A hálózatban jelen lévő nyelők száma korlátozott és ezek csak a hálózat szélén, előre megadott helyeken tartózkodhatnak.

Ugyancsak ismert az egyes szenzorok aktuális energiaszintje, amely információ a cikkben javasolt routing algoritmus segítségével gyűjthető be. A hálózat élettartama körökre van osztva; mindig a kör elején kerül sor a nyelők új helyének megállapítására, mely helyekre a nyelők rögtön oda tudnak lépni. Ezen információk alapján a szerzők egy egészértékű programozási feladatként fogalmazzák meg a nyelők új helyének megállapítását.

Az eddigi javaslatok mind idővezérelt hálózatokat feltételeztek. Széles körben alkalmazhatóak azonban az eseményvezérelt hálózatok is, hiszen bizonyos feladatok esetén társaiknál energiatakarékosabbak. Egy behatolás-detektálási feladat esetén például nincs szükség arra, hogy periodikusan minden szenzor küldjön csomagot a nyelőnek; elég csak a behatolót észlelőknek megtenni azt. Az eseményvezérelt, valós idejű továbbítást igénylő szenzorhálózatok esetére is születtek a nyelő mobilitására vonatkozó javaslatok, úgy egyugrásos [14] mint többesugrásos [15] kommunikációt használó hálózatokra.

Egyugrásos esetben azt feltételeztük, hogy a hálózatban a szenzorok klaszterekben vannak szervezve. A szenzorok a klasztervezérlőnek (KV) küldik a mért adatokat, a KV szerep pedig periodikusan vándorol a szenzorok között. A KV-k aggregálás után továbbítják a nyelőnek a megkapott adatokat; csak azon KV-k kommunikálnak, amelyek területén esemény következett be. A nyelő ismeri a KV-k földrajzi helyét, a KV-k pedig ismerik a nyelő helyzetét. A KV-k képesek szabályozni adóteljesítményüket, így amennyiben a nyelő közelebb helyezkedik hozzájuk, úgy kisebb erővel sugározzák az adatot a nyelő felé. A mozgatás célja az, hogy csökkentse az aktív KV-k által kommunikációra fordított energiát.

Ennek megfelelően három mozgatási stratégiát javasoltunk. *Minavg* esetben a cél az összes aktív KV által felhasznált energia minimalizálása. A kommunikációhoz szükséges energia a felek közötti távolság 2 és 5 közötti hatványával arányos, így az energia minimalizálásához a nyelő és az aktív KV-k közötti távolságok hatvány összegét kell minimalizálni. Ez esetben előfordulhat, hogy a KV-k energiafogyasztása nagyon egyenetlen lesz, ami a másodikként javasolt *minmax* stratégia alkalmazásával kerülhető el.

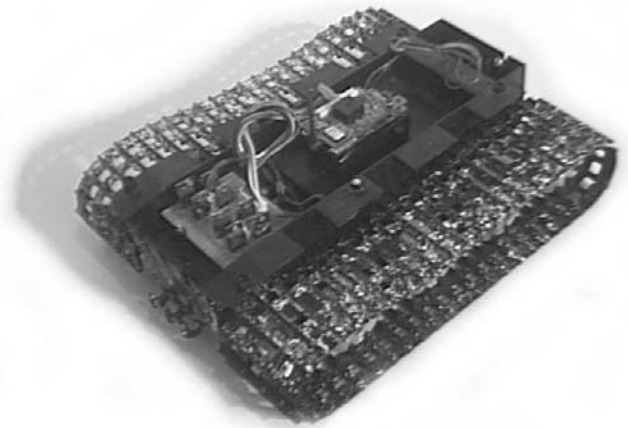
Ez a nyelőt úgy helyezi el a hálózatban, hogy a tőle legtávolabb eső aktív KV távolsága minimális legyen. Amennyiben a nyelő a klaszterek aktuális összenergia szintjét is ismeri, akkor a nyelő mozgatásának adaptivitása tovább növelhető; ezt a stratégiát *minrel*-nek nevezték el.

A stratégia a nyelő mozgatása során azon klaszterek energiafogyasztását igyekszik csökkenteni, amelyek kevesebb energiátartálékkal rendelkeznek. Ennek elérésére minden aktív KV esetére kiszámolja a kommunikációhoz szükséges energiatartalom és a klaszter energiátartalmának a hányadosát; a nyelő új helyének kiválasztásakor pedig ezen hányadosok maximumát minimalizálja.

Eseményvezérelt, többesugrásos kommunikációjú szenzorhálózatok esetén [15] analitikus vizsgálatokon keresztül előbb bemutattuk, hogyan alakul az egyes szenzorok terhelése, majd a kapott eredményeket felhasználva két fajta mozgatási algoritmust javasoltunk. Multihop kommunikáció esetén, az egy esemény jelentéséhez szükséges energia lineárisan arányos az esemény és a nyelő távolságával.

Ennek megfelelően, az első stratégiában a nyelő úgy mozog, hogy az események és a saját maga közötti távolságok összege minimális legyen; ez a *mintotal* algoritmus. A *minmax* algoritmus esetén a cél az események jelentése következtében fellépő terhelés egyenletesebb elosztása, ezért itt a nyelő a mozgásával a tőle legtávolabb eső esemény távolságát minimalizálja.

A BME Távközlési és Médiainformatikai Tanszékén már elkezdődött a mobil nyelő gyakorlati megvalósítása. A 4. ábrán látható lánctalpas mobil eszköz már képes közel mozdulni ahhoz a szenzorhoz, amelynek küldendő adata van.



4. ábra Mobil nyelőállomás

4. Összefoglalás

Amint azt bemutattuk, rengeteg olyan javaslat született a közelmúltban, mely a szenzorhálózatokbeli eszközök mobilitásának bevezetését szorgalmazta. Ezen javaslatok közös célja a hálózat működési minőségének javítása, illetve a hálózat használhatóságának meghosszabbítása.

Az elméleti és szimulációs eredmények azt mutatják, hogy ezek a megoldások tényleg képesek a kitűzött célok elérésére. Sajnálatos módon azonban a gyakorlati megvalósítások még váratnak magukra.

Irodalom

- [1] G. Wang, G. Cao, T. Porta,
„Movement-assisted sensor deployment”,
In Proc. of IEEE INFOCOM'04, Hong Kong, 2004.
- [2] F. Aurenhammer,
„Voronoi Diagrams – A Survey of a Fundamental
Geometric Data Structure”, ACM Computing Surveys,
1991, Vol. 23, pp.345–405.
- [3] Z. Butler, D. Rus,
„Event-based Motion Control for Mobile Sensor
Networks”, IEEE Pervasive Computing, October 2003.
Vol. 2, No.4, pp.34–42.
- [4] R. C. Shah, S. Roy, S. Jain, W. Brunette,
„Data MULEs: Modeling a three-tier architecture for
sparse sensor networks.”
In Proc. IEEE Workshop on Sensor Network Protocols
and Applications (SNPA), Anchorage, Alaska, 2003.
pp.30–41.
- [5] L. Tong, Q. Zhao, S. Adireddy,
„Sensor networks with mobile agents”,
In MILCOM 2003 – IEEE Military Communications
Conference, Boston, MA, October 2003.
Vol. 22, No.1, pp.688–693.
- [6] H. S. Kim, T. F. Abdelzaher,
„Minimum-energy asynchronous dissemination
to mobile sinks in wireless sensor networks”,
In ACM SenSys, Los Angeles, CA, November 2003.
- [7] A. Chakrabarti, A. Sabharwal, B. Aazhang,
„Using predictable observer mobility for
power efficient design of sensor networks”,
In Proc. of IPSN '03, 2nd International Workshop
on Information Processing in Sensor Networks,
Palo Alto, CA, USA, April 2003.
pp.129–145.
- [8] J. Luo, J.-P. Hubaux,
„Joint mobility and routing for lifetime elongation
in wireless sensor networks”,
In Proc. of IEEE INFOCOM '05, March 2005.
- [9] W. Zhao, M. Ammar,
„Message ferrying: Proactive routing in highly
partitioned wireless ad hoc networks”,
In Proc. of the 9th IEEE Workshop on Future Trends
in Distributed Computed Systems (FTDCS '03),
San Juan, Puerto Rico, May 2003,
pp.308–314.
- [10] Z. Wang, S. Basagni, E. Melachrinoudis, C. Petrioli,
„Exploiting sink mobility for
maximizing sensor networks lifetime”,
In Proc. of the 38th Hawaii International Conference
on System Sciences,
Big Island, Hawaii, Januar 2005.
- [11] A. Kansal, M. Rahimi, W.J. Kaiser, M.B. Srivastava,
G.J. Pottie, D. Estrin,
„Controlled mobility for sustainable wireless networks”,
In IEEE Sensor and Ad Hoc Communications and
Networks (SECON '04),
Santa Clara, CA, October 2004.
- [12] A. Kansal, A. Somasundara, D. Jea,
M.B. Srivastava, D. Estrin,
„Intelligent fluid infrastructure for embedded networks”,
In ACM MobiSys'04, Boston, MA, USA, June 2004.
pp.111–124.
- [13] S. R. Gandham, M. Dawande,
R. Prakash, S. Venkatesan,
„Energy efficient schemes for wireless sensor
networks with multiple mobile base stations”,
In Proc. of IEEE Globecom, SF, California, Dec. 2003,
Vol. 22, No.1, pp.377–381.
- [14] Z. Vincze, D. Vass, R. Vida, A. Vidács,
„Adaptive Sink Mobility in Event-driven Clustered
Single-hop Wireless Sensor Networks”,
accepted: 6th Int. Network Conference (INC 2006),
Plymouth, UK, 11-14 July, 2006.
- [15] Z. Vincze, D. Vass,
R. Vida, A. Vidács, A. Telcs,
„Sink Mobility in Event-driven Multi-hop Wireless
Sensor Networks”,
In Proc. 1st International Conference on Integrated
Internet Ad hoc and Sensor Networks (InterSense),
Nice, France, 30-31 May, 2006.
(Best Paper Award)

Energiahatékony kommunikáció szenzorhálózatokban

VASS DOROTTYA, VIDÁCS ATTILA

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
vass@tmit.bme.hu

Kulcsszavak: szenzorhálózat, energiahatékonyság, klaszterezés, útvonalválasztás, adataggregálás

A vezeték nélküli ad-hoc hálózatok egy speciális esetét képezik a szenzorhálózatok, ahol a csomópontokban apró, szerényebb számítási kapacitással rendelkező, különféle érzékelőkkel ellátott eszközök állnak, ám a hálózat nagy területet fed le és tipikusan rendkívül sok csomópontból áll. A hálózat tervezésekor egyik legfontosabb szempont a hálózat energiatakarékossága, mivel a szenzorok korlátozott energiával rendelkeznek, és vagy nem lehetséges, vagy nagyon költséges az újratöltésük. A következő oldalakon az energiahatékony kommunikáció megvalósításának lehetőségeit vizsgáljuk szenzorhálózatokban.

1. Bevezetés

A hagyományos, huzalozott, központi irányítással ellátott, néhány csomópontból álló szenzorhálózat már a múlté. A mikroelektronikában és a vezeték nélküli kommunikációban az utóbbi években bekövetkezett óriási fejlődésnek köszönhetően lehetőség nyílt olyan kis fogyasztású, alacsony költségű szenzorok kifejlesztésére, melyek hatékonyan tudnak vezeték nélküli összeköttetés nélkül is kommunikálni egymással, vagy egy külső bázisállomással. A jövő emberének mindennapjaiban helyet követelnek (lásd intelligens otthon), de jelen vannak az egészségügy, a környezetvédelem, a környezet megfigyelése, a honvédelem, és az ipar számos területén is.

Az egyes szenzorok egy kis elemmel vannak telepítve, melyeknek nem megoldható, vagy nagyon költséges a cseréjük, újratöltésük. Amikor ez az energiaforrás kimerül, a szenzor többé nem képes sem érzékelni, sem adatokat továbbítani. Ezért a hálózat tervezés fő szempontja az energiahatékonyság, hiszen minél tovább élnek az egyes szenzorok, annál tovább életképes a hálózat. A továbbiakban bemutatjuk, hogy a hálózat működésének egyes részfeladataiban milyen energiahatékony megoldások születtek.

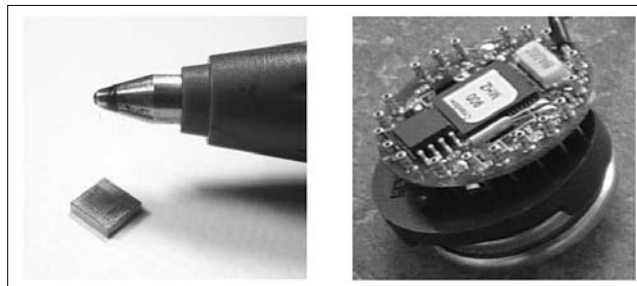
A második fejezetben áttekintjük a szenzorhálózatok sajátosságait. A harmadik fejezet témája a közeghosszú távú vezérlés, míg a negyedik fejezet a szenzorok klaszterekbe való összefogását vizsgálja. Az ötödik fejezet áttekinti az alkalmazható útvonalválasztási technikákat, a hatodik fejezet pedig az adatok aggregálásának lehetőségét vizsgálja, majd végül a hetedik fejezet egy rövid kitekintést ad a szenzorhálózatokban alkalmazható mobilitásról.

2. Szenzorhálózatok sajátosságai

Egy szenzorhálózat tipikusan nagyszámú, sok száz, vagy akár sok ezer, sűrűn elhelyezett szenzorból épül fel, a vizsgálandó jelenség vagy megfigyelt terület közvetlen

közelében, vagy a belsejében. Az egyes érzékelők elhelyezése nem feltétlenül előre megtervezett, lehetséges véletlenszerű telepítésük is. Egyes hálózatok például akkor is működőképesek, ha a szenzorokat egy repülőgépből szórják ki a vizsgálandó területre. Emiatt azonban a hálózat tervezőinek komoly nehézségekkel kell megküzdeniük, mivel olyan protokollokat és algoritmusokat kell kidolgozniuk, amelyek a hálózatot önszervező képességekkel ruházzák fel.

Egy tipikusnak mondható szenzorhálózat a következőképpen működik. A szenzorok feladata, hogy érzékeljék az érzékelési területükön bekövetkezett eseményeket és ezt jelentsék egy központi adatgyűjtő egységnek, az úgynevezett nyelőnek. Ez az esemény lehet egy behatoló detektálása, tűz felismerése, hőmérséklet, páratartalom, nedvességtartalom, esetleg sugárzás, szeizmikus aktivitás mérése, hang érzékelése vagy bármi más megfigyelése. A sokféle felhasználási lehetőséghez természetesen sokféle szenzor tartozik, de egy dolog közös ezekben a hálózatokban: a véges felhasználható energia. Az 1. ábra a hálózat alap építőkövét, a szenzort mutatja be.



1. ábra Szenzor csomópontok

A szenzorok energiafogyasztásának fő fázisai a következők. Energiát fogyasztanak érzékelés, az információ feldolgozása és adattovábbítás közben. Ezek közül a legjelentősebb a kommunikációra fordított energia. Az adatok átviteli költsége arányos a távolság magasabb hatványával, a hatványkitevő értéke általában

a környezettől függően 2-4 közé esik. Jelentős energiameennyiséget takaríthat meg a hálózat, ha az egyes szenzorok a rádiós adáskor teljesítményszabályozást alkalmaznak. Ekkor a csomópontok nem állandó nagyságú energiát használnak fel az információ továbbítására, hanem attól függően, hogy a szomszédai, vagy a célállomás milyen távolságra helyezkedik el, szabályozzák a rádiós adóteljesítményét, így a kommunikációs sugarat a szükséges mértékig lecsökkentik.

A következőkben arra összpontosítunk, hogy milyen módszerekkel csökkenthető a hálózat energiaszükséglete az adatok továbbítása során. (Egy alapos áttekintés található [1]-ben.)

3. Szenzorok közeghozzáférés-vezérlése

A szenzorok kommunikációja során számos probléma merül fel. Ha két, egymáshoz közeli szenzor adatküldéskor ugyanazt a csatornát használja, akkor két keret időben átlapolódik, a jelek összekeverednek és ütközés lép fel. Az újraküldés jelentős energia pazarlást jelent, ezért ennek az arányát a szenzorhálózatokban minimálisra kell csökkenteni, ugyanakkor el kell kerülni a túlzott jelzési és szinkronizációs költséget.

A közeghozzáférés vezérlés alapvető feladata a kommunikációs csatorna kiosztása, azaz hozzárendelése az egyes kommunikálni szándékozó szenzorokhoz.

A csatornakiosztás történhet statikus, vagy dinamikus módon. A megosztási módszer lehet:

- frekvenciaosztásos (FDM: Frequency Division Multiplexing),
 - időosztásos (TDM: Time Division Multiplexing),
 - kódosztásos (CDM: Code Division Multiplexing),
- esetleg e három módszer valamiféle kombinációja.

Sajnos a frekvenciagenerátorok (MEMS, olcsó kristály) pontossága csekély, így az időosztásos technikák nem hatékonyak.

A statikus kiosztás esetén a hálózat felépítésekor a csatornát fixen rendeljük hozzá az egyes szenzorokhoz, ezért az alkalmazás hátránya, hogy nagy állomásszám és/vagy nem egyenletes forgalom esetén a kihasználtság drasztikusan lecsökken.

Dinamikus csatornakiosztás esetén a változó igényeknek megfelelően oszthatjuk ki a csatornahozzáférés jogát. Igényeket figyelembe vevő vezeték nélküli hozzáférési technika például az ALOHA, a vivőérzékeléses többszörös hozzáférés (CSMA) és a lekérdezés (polling).

ALOHA az első véletlen hozzáféréseken alapuló vezeték nélküli MAC. Csillaghálózati topológiát feltételez, a központban egy vezérlővel. Külön csatornák vannak hozzárendelve a be- és kimenő forgalomnak, melyekhez az állomások aszinkron módon férnek hozzá. Ha ütközés lép fel, az állomások újra próbálkoznak egy véletlen várakozási idő után. A módszer problémája, hogy a csillagtopológia a mester csomóponttal a szenzorhálózati elvárásoknak nem megfelelő.

A CSMA alapötlete, hogy minden állomás az adás előtt belehallgat a csatornába, és csak akkor kezd el adni, ha a csatorna szabad. Lekérdezés (polling) esetén egy csomópont viszont csak akkor adhat, ha erre engedélyt kap egy mester csomóponttól.

A vezeték nélküli szenzorhálózatok speciális tulajdonságai miatt a közeghozzáférési technikáknak különleges követelményeknek kell megfelelniük. Energia-takarékossági szempontok miatt a csomópontok aktív részvétele csak az idő kis töredékében biztosítható. Mivel a szenzorok kiskapacitású, olcsó eszközök, egyszerűen implementálható (olcsó) megoldásokra van szükségünk. A vezeték nélküli szenzorhálózatok speciális igényeinek kielégítésére született meg az *S-MAC (Medium Access Control for Wireless Sensor Networks)* közeghozzáférés vezérlő protokoll.

S-MAC protokoll célja energiatakarékos, skálázható közeghozzáférés megvalósítása szenzorhálózatokban [2,3]. A protokoll három fő építőelemből épül fel:

- periódikus figyelés,
- ütközés és áthallás elkerülése,
- üzenet továbbadás.

A protokoll többgrásos (multihop) kommunikációt feltételez, tehát az üzenetváltás főként egyenrangú csomópontok között zajlik. Feltételezi továbbá, hogy az egész hálózat egyetlen alkalmazás céljából jött létre, tehát az egész rendszer teljesítményét vizsgálja, a csomópont szintű „fairness” másodlagos. Az alkalmazás során előfordulnak hosszú, tétlen időszakok. A következő alfejezetekben az S-MAC protokollt mutatjuk be.

3.1. Alvás-ébrenlét ütemezés

Az S-MAC megengedi az alvás-ébrenlét ütemezését, azaz két érzékelés közötti tétlen időszakban a szenzorok inaktív, úgynevezett alvó üzemmódba kapcsolhatnak. A szenzorok teljesítményfelvétele érzékelési szakaszban ~10 mW, míg alvó üzemmódban a teljesítményfelvétel ennek töredéke; ~10 μ W.

Általában az alvó üzemmód sokkal hosszabb, mint az ébrenléti, előfordulhat hogy a szenzorok az idő alig 1%-ában vannak ébren, esetleg még annyit sem. Ekkor az előre időzített ébredésig a szenzorok nem vesznek részt adattovábbításban, és nem érzékelik a környezetüket sem. Az alvó üzemmódba való átváltás történhet szinkronizáltan a többi csomóponttal, vagy véletlenszerű alvási és ébrenléti idők választásával (2. ábra).

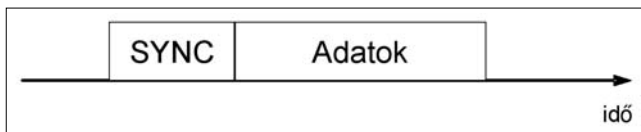


2. ábra Periódikus figyelés és alvás

Az alvás-ébrenlét ütemezés egy másik fajtája, amikor az egyes szenzorok azért váltanak alvó üzemmódba, mert a sűrűn telepített hálózat nélkülük is lefedett marad, így kímélik az energiaforrásaikat, amíg szükség nem lesz rájuk.

3.2. Ütemezés választása és karbantartása

Hogy egy állomás szinkronizációs (SYNC) csomagok mellett adatcsomagokat is fogadni tudjon, az aktív intervalluma két részre van osztva (3. ábra).



3. ábra S-MAC: Adatküldés és szinkronizáció

Minden csomag adása előtt az állomás egy véletlen ideig figyel a csatornát, hogy az szabad-e (vivőérzékelés – carrier sense). Egy szenzor aktív idejében küldhet egy SYNC és/vagy egy adatcsomagot is.

Minden csomópont karban tart egy ütemezési táblázatot a szomszédairól, amelynek felállításához a következő algoritmust használják.

1. Egy bizonyos ideig a szenzor figyel a csatornát. Ha nem hall ütemezési információt egyetlen állomástól sem, véletlenszerűen választ egy t időt az alváshoz, és közli ezt egy SYNC üzenetben a szomszédjaival. Ebben az esetben az állomás lesz a szinkronizáló.
2. Ha a csomópont a csatorna figyelése közben meghallja egy szomszédja ütemezését, akkor ugyanazt az értéket állítja be magának is, majd vár véletlen t_d ideig, és elküldi az ütemezését a szomszédainak. SYNC üzenet: $t - t_d$ másodperc múlva alszom.
3. Ha egy csomópont vesz egy sajátjától eltérő ütemezésű SYNC csomagot, akkor átveszi azt is, azaz felébreszti magát akkor is.

Egy szenzor akkor is küld időnként SYNC csomagot, ha nincsenek követői. Ez teszi lehetővé az esetleg újonnan érkező állomások szinkronizálódását is.

4. Szenzorok összefogása klaszterekbe

A fizikai összeköttetések véletlenszerűek a telepítés véletlenszerűségéből kifolyólag, ám szükség van logikai topológia kialakítására a skálázhatóság érdekében. A szenzorok összefogása csoportokba, úgynevezett klaszterekbe, széles körben elterjedt módszer a skálázhatóság növelésére és a kommunikációra fordított energia csökkentésére.

Minden szenzor tagja legalább egy klaszternek, és minden klaszterhez tartozik egy klasztervezérlő csomópont, amely vezérli a hozzá tartozó szenzorokat. A klasztervezérlő képes kommunikálni a többi klasztervezérlővel és a nyelővel is. Ebben a hálózati struktúrában a szenzorok nem egyenként, külön-külön küldik el az adataikat a nyelőnek, hanem először a klasztervezérlő kapja meg az összes információt a csoportjához tartozó szenzoroktól. Ez a csomópont összefogja az adatokat lehetőség szerint egyetlen csomaggá és csak ez az egyetlen csomag kerül továbbításra a nyelőnek.

A klasztervezérlők kiválasztása lehet determinisztikus, de számos protokoll szerint önszerveződen történik. Mivel a klasztervezérlőkre az átlagos szenzorokénál jóval nagyobb terhelés hárul, ezért ezt a szerepet időről időre váltogathatják is.

4.1. LEACH protokoll

A fentiekre példa a *LEACH (Low-Energy Adaptive Clustering Hierarhy)* protokoll, amelyben a szenzorok önszerveződen alkotnak cellákat [4]. Minden klaszterben egy csomópont klasztervezérlőként működik. A hálózathoz tartozik egy gyűjtőállomás, amely fixen telepített.

A LEACH protokoll működése közben az idő körökre oszlik fel. Minden kör a felépítési fázissal kezdődik, amelyet az adatátviteli fázis követ, ekkor juttatják el a szenzorok a jelentéseiket a nyelőhöz. A felépítési fázisban minden csomópont egy bizonyos valószínűséggel klasztervezérlővé válhat. Ez a valószínűség függ többek közt az adott hálózatra jellemző optimális cellaszámtól.

A klasztervezérlők minden szenzort értesítenek az állapotukról, majd a csomópontok eldöntik, hogy melyik klasztervezérlőhöz akarnak csatlakozni. Azt választják, amelyikkel a legkevesebb energia befektetésével tudnak kommunikálni, ez általában a legközelebb elhelyezkedő klasztervezérlő.

Amint az összes csomópont klaszterekbe szerveződött, minden klasztervezérlő kihirdeti a *TDMA (Time Division Multiple Access)* ütemezést. Ez az ütemezés definiálja a szenzoroknak, hogy mikor adhatnak és mikor kell lekapcsolniuk a rádiójukat. Elkerülendő az ütközést a szomszédos klaszterek szenzorjaival, minden cella kódosztásos többszörös hozzáférést (Code Division Multiple Access, CDMA) alkalmaz.

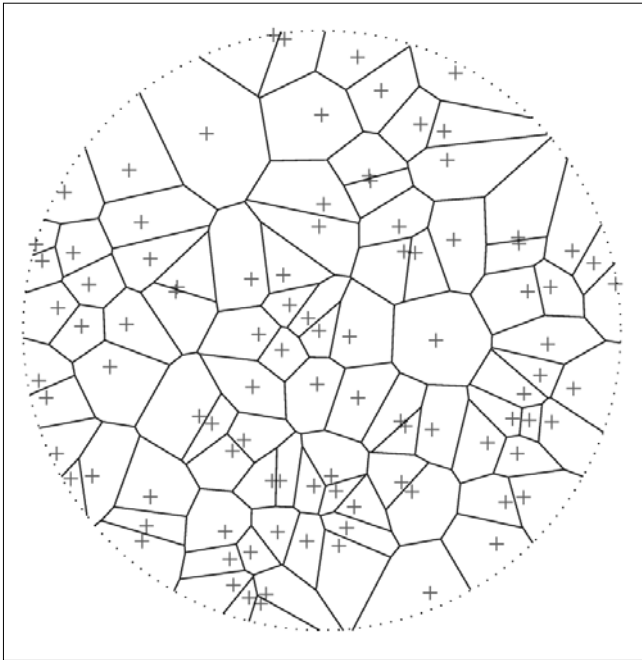
A klasztervezérlővé váló szenzor véletlenszerűen választ magának kódot a CDMA listából és meghirdeti a hozzá tartozó csomópontoknak. A klasztervezérlő az összes hozzá érkezett csomagot ezek után ennek a kódnak a segítségével szűri, így elkerülhető a cellák közötti ütközés.

A 4. ábrán egy szenzormezőben kialakított 100 klaszter látható.

Az adatgyűjtési fázisban a protokoll feltételezi, hogy minden szenzor adatokat gyűjt a környezetéből és beküldi azokat a klasztervezérlőnek. A klasztervezérlő végzi az adatok aggregálását és továbbküldését a nyelőnek.

Ha egy szenzor klasztervezérlőként funkcionál, az jelentősen igénybe veszi az energiaforrásait, ezek a csomópontok hamar kimerülnek. Ezt elkerülendő, a klasztervezérlői szerepet a szenzorok véletlenszerűen cseréltetik egymás között. Minden kör elején a csomópontok újraértékelik a klasztervezérlői szerepüket, s a döntésüket a hálózat többi részétől függetlenül hozzák meg. Amint változás történik a klasztervezérlők közt, a klaszterhatárok átalakulnak, minden csomópont újra döntést hoz, hogy melyik klasztervezérlőhöz is akar csatlakozni.

Néhány tanulmányban a szerzők az alap LEACH algoritmus továbbfejlesztését javasolják [5-7].



4. ábra Szenzormező 100 klaszterrel és vezérlőkkel

5. A hálózati réteg és az útvonalválasztás feladatai

A kommunikációs költségek csökkentésének egyik módja, ha az adatátviteli távolságokat csökkentjük. Ha a szenzorok a csomagokat nem közvetlenül a bázisállomásnak továbbítják, hanem az üzenet több lépésben, más szenzorokon keresztül jut el a célhoz, akkor a nagyobb távolságok sok kicsire oszlanak fel. Mivel a kommunikációs költségek a távolságnak egy magasabb hatványával egyenesen arányosak, ezért ezzel a módszerrel energiát takaríthatunk meg. Nagyobb kiterjedésű hálózatokban tehát érdemes többugrásos kommunikációt megvalósítani, ehhez azonban szükség van útvonalválasztásra.

A hálózati réteg feladata, hogy a hálózat bármely két szereplője között biztosítsa a hálózati kapcsolat létrejöttét, és karbantartsa azt. Szabályoznia kell a forgalmat, és bizonyos esetekben szolgáltatásminőség (QoS) biztosítási feladatai is lehetnek. A legfontosabb része azonban ezek mellett, és ezek részeként az útvonalválasztás.

A szenzorhálózatok több pontban eltérnek a hagyományos vezeték nélküli, vagy ad-hoc hálózatoktól, ezért eltérő útvonalválasztó eljárásokat kell használni. Az egyik ilyen probléma, hogy a szenzorok nagy száma miatt nem lehet egy olyan globális címzési sémát kialakítani, mint például az IP, mert túl nagy lenne az egyéni azonosító támogatás költsége. Nagy útvonalválasztó táblázatok fenntartása az egyes szenzorokban nem lehetséges, mert a szenzorok véges memóriakapacitással rendelkeznek, és a táblázatok periodikus frissítése is nagy terhet jelentene a hálózatnak. Így a hagyományos IP alapú útvonalválasztó módszerek nem használhatók.

Ezen kívül a szenzorhálózat működésének önszervezőnek kell lennie, mivel a szenzorok magára hagyottan működnek, önállóan kell a hálózatnak felépülni, és az esetleges topológia változásokra reagálni. Mivel a szenzorhálózatok általában adatcentrikusak, sok esetben nem is az a lényeges, hogy az adott adatot melyik szenzor küldte, hanem maga az adat, ami az érzékelés körülményeit is leírja.

Szenzorhálózatokban olyan útvonalválasztó algoritmusok alkalmazhatók, melyek szem előtt tartják az energia megkötések mellett a hálózat önszervező jellegét és az esetleges topológiaváltozásokat is. A szenzorhálózatokban alkalmazott útvonalválasztó megoldásokról bővebb áttekintést ad Al-Karaki és Kamal [8].

6. Adataggregáció

Az átviendő adatok lokális aggregálásával a többugrásos hálózatokban csökkenthetjük a továbbítandó adatok mennyiségét. Mivel a szenzorok rendelkeznek egy kis teljesítményű processzorral is, így lehetőség nyílik az adatok szelektálására, előzetes feldolgozására, esetleg tömörítésére is. A szenzorhálózatok általában sűrűn telepítettek, így egy eseményt több szenzor is érzékel. A legegyszerűbb példa, ha egy szenzor olyan eseményről kap jelentést egy másik csomóponttól, melyet maga is érzékelt, akkor elég csak egy üzenetet továbbítani.

Alkalmazástól függően lehetséges olyan algoritmusok használata, melyekkel jelentősen lecsökkenthető a továbbítandó csomagok száma anélkül, hogy ezáltal információt vesztenénk el a megfigyelt jelenségről. Egy lehetséges alkalmazás például egy terület maximális hőmérsékletének mérése.

Azok a szenzorok, melyek az adatok továbbításában vesznek részt a nyelő felé, valószínűleg több szenzor mérési eredményeit is továbbítják. Ha ezek a szenzorok egy ideig tárolják a beérkezett adatokat és ki tudják választani a legnagyobb hőmérsékletet tartalmazó adatcsomagot, akkor elegendő csak ezt az egy adatot továbbítani. Más alkalmazásokban lehetséges az adatok összekombinálása is. Ha egy átlagos értékre vagyunk kíváncsiak, akkor azoknak a csomópontoknak, amelyek több adatot is továbbítanak, elegendő egy átlagértéket és az adatok számát továbbítani.

Az adatok összefogása a hálózatban aggregációs pontokon történik. Aggregációs pont lehet a hálózat néhány kitüntetett csomópontja, de az is lehetséges, hogy az összes szenzor képes valamiféle adatösszegzésre. Fontos hogy az összegzés minél hamarabb megtörténjen, hogy fölösleges csomagok ne terheljék az amúgy is korlátos energiaforrásokat.

Sűrű hálózatokban előfordulhat, hogy néhány csomag továbbítása közeli, mégis éldiszjunkt utakon történik. Célszerű ilyenkor olyan útvonalválasztó algoritmust alkalmazni, amely az útvonalválasztási fában kerüli az éldiszjunkt utak létrehozását. Ez történhet minimális élszámú utak, vagy nagy foksámú aggregációs csomópontok létrehozásával.

7. Mobilitás szenzorhálózatokban

Néhány alkalmazás a kommunikációs távolság csökkentése érdekében mobil gyűjtőállomást feltételez. A kommunikációs távolság csökkentésével csökken a szenzor energiafogyasztása is.

Ekkor a nyelő mozgatása lehet véletlenszerű [9], vagy valamilyen energiaoptimalizáló stratégia szerinti [10,11].

8. Összefoglalás

Habár egy szenzorhálózat tervezésekor sok nehézséggel kell megküzdeni az önszerveződés, a véletlenszerűség és a korlátozott számítási kapacitás miatt, a legfontosabb kérdés a korlátos energiákkal rendelkező szenzorok energiahatékony működése.

A fentiekben összefoglaltuk, hogy a hálózat működésének egyes részfeladataiban milyen energiahatékony megoldások születtek. Ismertettünk energiahatékony közeghozzáférés vezérlést, valamint útvonalválasztó algoritmust. A kommunikációs távolságok csökkentésére a klaszterezés előnyeire mutattunk rá. Ismertettünk a forgalom csökkentésére adataggregációs módszereket is. Az adataggregáció megvalósítása egyszerű esetben a klasztervezérlőknél történik, míg többugrásos esetben az adatok összefogását kiemelt aggregációs pontok, vagy mindenegybes, az útvonalválasztásban résztvevő szenzor végzi.

Irodalom

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine, 40 (8) 2002. pp.102–114.
- [2] Wei Ye, John Heidemann, Deborah Estrin, An energy-efficient MAC protocol for wireless sensor networks. In Proc. of InfoCom 2002, NY, USA, pp.1567–1576.
- [3] Wei Ye, John Heidemann, Deborah Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Transactions on Networking, 12 (3) 2004. pp.493–506.
- [4] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks. In Proc. of 33rd Hawaii International Conference on System Sciences, 2000. pp.3005–3014.
- [5] M. J. Handy, M. Haase, D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In Proc. IEEE Int. Conference on Mobile and Wireless Communications Networks (MWCN), Stockholm, Sweden, 2002.
- [6] S. Lindsey, C. Raghavendra, Pegasus: Power efficient gathering in sensor information systems. In Proc. Aerospace Conf. 2002. Vol. 3, pp.1125–1130.
- [7] O. Younis, S. Fahmy, Heed. A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor network. IEEE Transactions on Mobile Computing, 3 (4), 2004.
- [8] J. N. Al-Karaki, A. E. Kamal, Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communication, Vol. 11, 2004. pp.6–28.
- [9] S. J. Rahul C. Shah, Sumit Roy, W. Brunette, Data MULEs: Modeling a three-tier architecture for sparse sensor networks. In Proc. IEEE Workshop on Sensor Network Protocols and Applications (SNPA), Anchorage, Alaska, USA, 2003. pp.30–41.
- [10] Z. Vincze, D. Vass, R. Vida, A. Vidács, „Adaptive Sink Mobility in Event-driven Clustered Single-hop Wireless Sensor Networks”, In Proc. 6th International Network Conference (INC 2006), Plymouth, UK, 2006. pp.315–322.
- [11] Z. Vincze, D. Vass, R. Vida, A. Vidács, A. Telcs, „Sink Mobility in Event-driven Multi-hop Wireless Sensor Networks”, In Proc. 1st Int. Conf. on Integrated Internet Ad hoc and Sensor Networks (InterSense), Article No.13, Nice, France, 2006.

CASCADAS – Autonóm kommunikáció és pervazív, helyzetfüggő szolgáltatások

BENKŐ BORBÁLA, KATONA TAMÁS, SCHULCZ RÓBERT

Budapesti Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék
{bbenko,tkatona,schulcz}@hit.bme.hu

Kulcsszavak: autonóm kommunikáció, pervazív szolgáltatások, önszerveződés, tudásháló, pervazív felügyelet

A CASCADAS¹ egy IST-FET² projekt, melynek célja a komplex, elosztott, pervazív³ szolgáltatások új generációjának meg-
alapozása mind elméleti síkon, mind mintaalkalmazásokon keresztül. Az autonóm és persze automatikus kommunikációra
épülő, helyzetudatos/helyzetfüggő, önszerveződő hálózat alapegysége – egyfajta absztrakcióként – az ACE⁴, az autonóm
kommunikációs elem. Az ACE szolgáltatásokat nyújt és használ, alkalmazkodik a szituációhoz, tervet készít, tudáshálót
épít, illetve autonóm döntései alapján mozog, önszerveződik. Külön érdekesség, hogy csak Layer2 hálózatot tételezünk fel,
így lehetőség nyílik egy új, a fizikai közelséget is alapul vevő Layer3 kommunikáció kidolgozása is.

1. Bevezetés

A CASCADAS [1] egyike annak a négy IST-FET projekt-
nek (ANA, HAGGLE, Bionets, Cascadas), melyek hely-
zetfüggő, autonóm technológiákat kutatnak különböző
szemszögekből [2]. A CASCADAS célkitűzése a jövő
összetett, nagyban elosztott, pervazív, kommunikáció-
intenzív szolgáltatásainak (menedzsment, fejlesztés, kon-
figuráció stb.) költségeit csökkenteni azáltal, hogy a há-
lózati autonóm építőelemekből áll, önszerveződik, tudást
menedzsel és alkalmazkodik a körülményekhez.

A CASCADAS célja több külön síkon is megfogal-
mazható. Elméleti síkon célunk egy közös absztrakció
kidolgozása (ez az ACE⁴) és erre épülve az önszerve-
ződéshez, tudásháló-építéshez, pervazív felügyelethez
és biztonsági vonatkozásokhoz modellek, algoritmusok,
és általános elvek megfogalmazása. Gyakorlati síkon,
– ahogy ez szokás – az elvek működőképességét min-
taalkalmazások készítésével demonstráljuk. Másodla-
gos cél emellett egy olyan kommunikációs modell kidol-
gozása is, mely csupán Layer2 hálózatot követel meg,
és hasznosítja a hálózati topológiában rejlő fizikai köz-
elség információt (mely egyébként IP használatával
elvész)⁵. Egy másik irányzat szerint az ACE-ek egy
CPN⁶, azaz kognitív csomaghálózat [3] felett futnak.

2. A CASCADAS vízió

Lássuk, mi a projekt célkitűzése, illetve milyen motiváló
erők, elvek húzódnak a háttérben!

2.1. A vízió

A CASCADAS vízió szerint a világ az autonóm, per-
vazív, helyzetfüggő szolgáltatások felé halad; cél a fel-
merülő problémák feltérképezése, modellek kidolgozá-
sa, és megoldásokat nyújtása [4]. Az absztrakció alap-
ja egy közös, pehelysúlyú modell, az ACE. A jövő szol-
galtatásai ACE-eken keresztül lesznek elérhetőek.

Az ACE-ek autonóm és automatikus módon – em-
beri beavatkozás nélkül – oldják meg a szolgáltatások-
kal kapcsolatos kommunikációs és menedzsment fela-
datokat, egyfajta optimalizációra törekedve.

- Az ACE-ek autonóm módon észlelik és
rendszerzik a tudást, mely a helyzet megértéséhez
szükséges, ideértve a fizikai, technológiai,
szociális, felhasználó-specifikus illetve kérés-
specifikus elemeket is.
- Az ACE önkonfiguráló és önadaptáló. A helyzethez
alkalmazkodva módosítja saját magát, illetve
paraméterezi az általa nyújtott szolgáltatást.
- Az ACE-ek (például szolgáltatás-optimalizálási
céllal vagy új szolgáltatás létrehozásához)
önszerveződő struktúrákba tömörülhetnek.
- Az ACE-ek ön-* (ejtsd: öncsillag) tulajdonságokkal
rendelkeznek (például öngyógyítás, önazonosság,
önkonfiguráció).

Az autonóm, önszerveződő kommunikációs eleme-
ket a projekt témakörökre bontva vizsgálja. (1) *Az ACE
modell.* Egy közös absztrakció definiálása, modellezés,
keretrendszer létrehozása. (2) *Szemantikus önszerve-
ződés.* Csoportalkotás, összeolvadás; ACE mobilitás.
(3) *Tudásháló.* A tudásháló felépítésének és a tudás

¹ CASCADAS: Component-ware for Autonomic, Situation-aware Communications, And Dynamically Adaptable Services
(CASCADAS: Komponenstechnológia autonóm, helyzetfüggő kommunikációhoz és dinamikusan alkalmazkodó szolgáltatásokhoz),
IST-FET EU 6. keretprogram projekt

² IST-FET: Information, Society, Technology – Future Emerging Technologies
(Információ, társadalom, technológia – A jövő előretörő technológiái), Európai Unió 6. keretprogram

³ pervazív: átható, mindenütt jelenlévő

⁴ ACE [ejtsd: ész]: Autonomic Communication Element (Autonóm Kommunikációs Elem)

⁵ Erre a megkötésre – amellyel, hogy elméletileg is érdekes téma –, praktikus okokból is szükség volt, hisz egyes felmerülő primitív operációs
rendszerek (például a TinyOS) kifejezetten csak Layer2-t támogatnak.

⁶ CPN – Cognitive Packet Network (Kognitív Csomaghálózat)

rendszerének kérdései. (4) *Átfogó felügyelet*. Az elosztott rendszer folyamatainak figyelése, és szükség esetén beavatkozás (pl. egyfajta rendszer-öngyógyítási képesség). (5) *Biztonsági vonatkozások*. A klasszikus feladatokon (azonosítás, hozzáférési jogok, titkosítás) kívül egy hírnév (reputation) alapú biztonsági modell is.



1. ábra Elvek és eszközök a CASCADAS vízióban

2.2. Kulcsszavak és a mögöttük rejlő elvek

A CASCADAS – csakúgy mint a másik három kapcsolódó projekt – számos, a telekommunikációs szektorban talán szokatlan fogalmat, kulcsszót vonultat fel. Sok köztük a hasonló vagy átfedő jelentésű, és akad olyan is, mely tulajdonképp csak egy új név egy már régóta ismert fogalomra.

Autonóm

Az „autonóm” szó elsőként 2001-ben került a figyelem középpontjába, amikor az IBM elindította az autonóm számítástechnika (autonomic computing) kezdeményezést, melynek célkitűzése egy – az emberi idegrendszer működéséhez hasonló – önmenedzselő rendszer volt. Az IBM által azonosított funkcionális elemek, melyek az autonóm működéshez szükségesek:

- Önkonfiguráció (self-configuration): az összetevők önálló, automatikus beállítása.
- Öngyógyítás (self-healing): az összetevők meghibásodásának érzékelése, hiba esetén a rendszer átrendezése, a hiba kijavítása, vagy legalábbis a negatív következmények csökkentése.
- Önoptimalizálás (self-optimizing): az erőforrások monitorozása és optimális használata az elvárásoknak megfelelően.
- Önvédelem (self-protection): a támadások azonosítása, proaktív védelem.

Szokás különbséget tenni az autonóm és az automatikus vonatkozások között. Az automatikusság jelentése, hogy önállóan, külső (például emberi) beavatkozás nélkül folyik. Az autonóm természetesen automatikus is egyben, de hangsúlyozni szokás a rendszer ön-^{*} aspektusait (mely egyfajta intelligencia).

A CASCADAS ACE autonóm módon működik, tehát rendelkezik a fenti négy tulajdonsággal.

Autonóm kommunikáció

Az autonóm kommunikáció lényege, hogy a kommunikáció ön-^{*} tulajdonságokkal bír. Egyrészt maga a kom-

munikációs technológia hibatűrő, önoptimalizáló, másrészt a kommunikáló elemek is autonóm módon döntenek (üzenetküldés, fogadás, az üzenet értelmezése, cselekvés az üzenet hatására).

Absztrakt szinten nézve az IBM autonóm rendszere egy viszonylag ismert környezethez optimalizálta a saját belső komponenseit. Autonóm kommunikációs rendszernél ez a környezet is nagyban változó, de kommunikáció révén lehetőség nyílik a környezet bizonyos fokú befolyásolására.

Az autonóm kommunikáció egyben azt is jelenti, hogy a rendszerben az intelligencia nem egy központi helyen, hanem az elemek között szétszórtva van jelen. Az autonóm kommunikáció a CASCADAS projekt alapja.

Pervazív szolgáltatások

Pervazív, mindenütt jelenlevő (pervasive, ubiquitous, everywhere) szolgáltatások alatt értjük, hogy a szolgáltatás forrása elsősorban (logikailag) a környezet, nem pedig egy konkrét számítógép. Ez többféle dolgot jelenthet, például szemantikus/tulajdonság alapú címezést; valamiféle hely/lokáltság fogalmát (például a szolgáltatást csak a közeli felhasználók érhetik el); nagyfokú logikai mobilitást (a szolgáltatás forrása szabadon mozog a neki legmegfelelőbb hely felé); illetve azt, hogy a környezet az igényeknek megfelelően átszervezi magát (például a népszerű szolgáltatást nyújtó elemek számát növeli). De olyan értelmezés is létezik, miszerint a szolgáltatások egyszerűen csak körülveszik a felhasználót (még a kenyérpírító is számítógép).

A pervazivitás – a ma szokásos értelmezés szerint – tipikusan párosul valamilyen intelligens vonatkozással, például önszerveződéssel, autonomiával. A pervazivitás a CASCADAS vízió egyik alapja, így a projektben elsősorban pervazív szolgáltatásokkal foglalkozunk.

Helyzettudatos, helyzetfüggő

A helyzetfüggőség (situation-, context-awareness, environment-aware) egyfajta újragondolása azoknak az elveknek, melyek már régóta jelen vannak a szabályozástechnikában és számos szoftveres területen (pl. ágensmodellek). A név beszédes, jelentése, hogy az elem megfigyeli a kontextust, és ennek megfelelően reagál. A feladat tehát négy lépésből áll: a kontextus megfigyelése, a kinyert információ értelmezése, a reakció kiszámítása és végül maga a reakció.

A helyzetfüggőségi modellek megjelenése után nagyon hamar nyilvánvalóvá vált két probléma. Az egyik a kommunikáció exponenciális robbanása: az információ – például a helyzetet leíró elemi üzenetek – mennyisége az elemszám növelésével exponenciálisan nő; így a modell, mely egy kis kísérleti rendszeren működött, nem skálázható a valóságos feladatmérethez. A másik probléma a megértés körül van; feltételezhetünk-e egy mindenki által ismert, háttérbeli ontológiát, amely garantálja, hogy ha a vett üzenetet az elem megérti, akkor helyesen érti meg? A CASCADAS megoldása szerint minden egyes ACE rendelkezik egy saját környezetmóddellel (mely nem fix, hanem például az észlelt új infor-

mációk hatására alakul, finomodik), és az üzenetek értelmezése a környezetmodell használatával történik. Az ACE helyzetudatos elem.

Helyfüggettség, lokalitás elve

A pervazivitás egyik értelmezése nagyon hasonló a lokalitás elvéhez. A lokalitás eszerint azt jelenti, hogy a rendszerben – önszervezés, mobilitás stb. eredményeként – a szolgáltatást nyújtó és használó fél egymáshoz közel vannak, így tulajdonképpen csak lokális kommunikációra van szükség (például az üzeneteket csak a hálózat közeli részein kell terjeszteni, hiszen úgyis ott van az, akit érdekel). Egy másik megvilágítás szerint a lokalitás azt jelenti, hogy egy információ értéke a keletkezés környékén a legnagyobb, időben és térben távolodva az információ értéke/pontossága/helyessége csökken.

A pervazív rendszerekben nagy szerepe van a lokalitásnak – értve ezalatt logikai vagy fizikai helyet –, hisz a szolgáltatás forrása lokális, a közvetlen környezetben található. A CASCADAS tudásháló támogatja a lokális működést (is).

Mobilitás

A CASCADAS vízió szerint az ACE mobil természetű. A félreértések elkerülése végett: az ACE mobilitás alatt nem fizikai mobilitást értünk – például nem azt, hogy egy ACE eltávolodik és kikerül a WLAN hatósugarából –, hanem logikai mobilitást, tehát hogy az ACE tetszése szerint mozoghat a fogadására alkalmas helyek között.

3. Eszközök

Vizsgáljuk meg közelebbről, milyen eszközökkel valósítja meg a CASCADAS a projekt vízióban vázoltakat!

3.1. Tudásháló

Alacsony szintről közelítve, az ACE a környezetre vonatkozó információt egy tudáshálóból nyeri. A tudásháló azonban több, mint a környezet pusztá absztrakciója; képes a tudást megszerezni, önmagát optimalizálni. A tudáshálót – az önazonosság paradigmáját követve – maguk az ACE-ek építik fel, és ha valamelyik ACE arra érdemes információval rendelkezik, elhelyezi a tudáshálóban.

A tudásháló támogatja a lokalitást, tehát azt, hogy a benne tárolt információk elsősorban a közvetlen környezetnek szánja, és az idő múlásával értékük/értelmük csökken. A lokalitás lehet fizikai vagy logikai természetű. Fizikai jellegű lokalitásnál a keletkezési hely a lényeges, logikainál a keletkezési ACE. A különbség akkor látható, ha az ACE elmozog egy másik fizikai helyre, hisz logikai lokalitásnál az információnak követnie kell (ezt egyébként a legegyszerűbb úgy megoldani, hogy az ACE a hozzá kötött információt saját magában tárolja).

A tudásháló emellett nemlokális, önrendező működésre is képes. Az önszerveződést hierarchikus fedőhálózatok (hierarchical overlays) segítségével képzeljük el. A skálázhatóság érdekében a jelenlegi álláspont szerint a hierarchiaszinten belüli kommunikáció lokális természetű; az emiatt keletkező korlátok pedig a hierarchiák közötti, vertikális irányú kommunikációval hidalhatók át.

A tudásháló tudásatomokból áll, ezek a tudásháló önrendezésének alapegységei.

3.2. Önszerveződés

Az önszerveződés célja többféle lehet, például a szolgáltatás és a kliens kapcsolódásának segítése (pl. fizikai közelítés), a szolgáltatás minőségének növelése (pl. elérhetőség növelése replikációval, azonos szolgáltatást nyújtó elemek csoportbafoglalása és terhelés-egyenlítés), vagy akár új szolgáltatás létrehozása.

Az önszerveződés alapművelete az csoportbafoglalás (aggregáció), mely lehet laza vagy szoros jellegű. A szoros aggregáció kizárólagos.

Az önszerveződés eredményeképpen létrejött struktúra technikai értelemben hierarchikus overlay hálózatnak felel meg. Viszont a felsőbb hierarchiaszinten ugyanúgy autonóm kommunikációs elemek vannak (és az autonómításból adódóan nem biztos, hogy a megszokott determinisztikus módon működnek).

3.3. Pervazív felügyelet

Az átfogó felügyelet (pervasive supervision) a rendszer öngyógyítását és önoptimalizációját segíti. A felügyelő szerv megfigyeli a rendszer – illetve a nem az egész rendszer, csak az ebbe beleegyező ACE-ek – működését, valamint a kontextust. Hibák vagy optimalizálható elemek után kutat és ha szükséges, beavatkozik: például utasítást ad az eddigi önszerveződés átalakítására, mozgásra utasítja az ACE-t, a hibás környezetmodell kijavítását kezdeményezi. A felügyelet szerződésen alapul, a szerződés a felügyelt ACE-t a kapott utasítások feltétlen végrehajtására kötelezi, tehát az ACE autonómiaja felügyelt módban korlátozott.

Mitől átfogó a felügyelet? A pervazív szó azért került bele az elnevezésbe, mert a felügyelet egyszerre működik minden szinten (hálózat, kommunikáció, tartalmi szint, önszerveződés, szociális vonatkozások).

Elvi nézőpontból a pervazív felügyelet az egyik legfontosabb elem, ezáltal kerülnek ki az ACE-ek a kvázi-véletlenszerűen önszerveződő, hibára érzékeny, kissé rugalmatlan állapotból, és kapnak esélyt az önjavításra és az önoptimalizációra.

3.4. Biztonság

A biztonsági rendszer alapfeladata a CASCADAS projektben a támadások elleni védelem (azonosítás, hitelesítés, sértetlenség, DoS elleni védelem). Emellett kiegészítő információként hozzájárul az ACE-ek környezetmodelljének alakulásához. Létrehoz egy hírnév (re-

putation) rendszert, melyben az egyes ACE-ek előléteinek kivonata található. A partnerrel való együttműködést segítheti az előélet ismerete (például a rossz hírnév óvatosságra int).

4. Az ACE modell

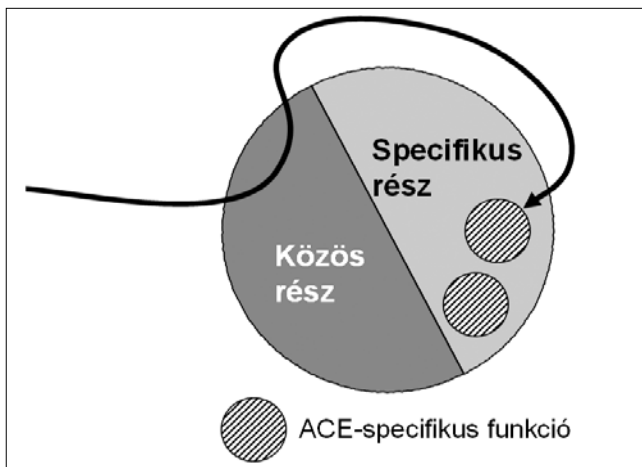
Az ACE az a közös absztrakció, mely körül a négy másik témakör felsorakozik. Az ACE egy komponensmodell, illetve ennek levetítése egy konkrét architektúrára (az architektúrára jelen cikkben nem térünk ki).

4.1. Az alap ACE modell

Az ACE különböző absztrakciós szinteken vizsgálható. Nézzük ezeket az általánosabb megfogalmazástól a konkrétumok felé haladva [5]!

Közös rész és specifikus rész

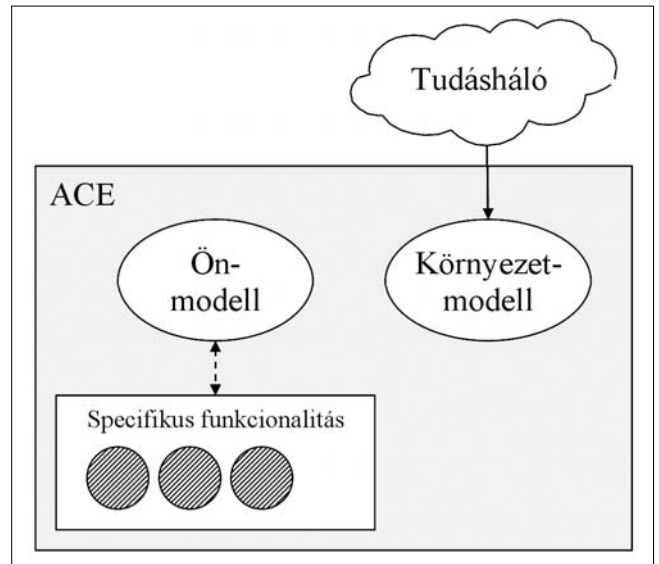
A CASCADAS ACE modellje két részből áll: egy közös és egy specifikus részből (2. ábra). A közös részben azok a funkciók vannak, melyek minden egyes ACE példányban megtalálhatók. A specifikus részben van minden ezen felüli dolog, például a szolgáltatást nyújtó képesség. A specifikus részben található elemek a közös résszel való kommunikáció által térképezhetők fel, érthetők el.



2. ábra Az ACE két része

Mi alapján működik az ACE?

Az ACE két modellel rendelkezik: egy önmodellel és egy környezetmodellel (3. ábra). Az önmodell a saját működését és céljait írja le, a környezetmodell a környezetet. Mindkét modell változhat, finomodhat. Az ACE ismeri saját céljait és lehetőségeit, ez alapján tervet készít és a terv alapján jár el. A cselekvések lehetnek reaktívak (például szolgáltatást nyújt egy bejövő kérésre válaszként) vagy proaktívak (szolgáltatás igénybevétele kezdeményezi). Az ACE belső „intelligens” része az önmodell és a környezetmodell alapján határozza meg a cselekvést (mely nem csak üzenetküldés jellegű lehet, hanem akár valamelyik modell átalakítása is).



3. ábra

Az ACE az ön- és a környezetmodell alapján működik

Elvi modell

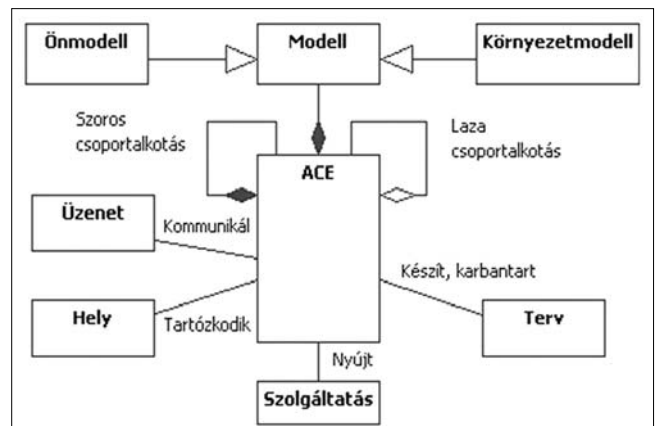
Az ACE-ek szolgáltatást nyújthatnak más ACE-eknek. Az ACE-ek üzenetekkel kommunikálnak. A kommunikáció három lépésből áll: *felfedezés* (a felek megtalálják egymást), *szereződéskötés* (megegyeznek az együttműködés feltételeiben), *együttműködés*. A lépések nem feltétlen különülnek el explicit módon (tehát egy szolgáltatást hirdető üzenet egyben magában foglalhatja az alapértelmezett szerződést is). Létezhetnek előre megkötött szerződések, melyeket már példányosítás pillanatában érvényesnek tekinthetünk (például az SEE ACE szolgáltatásainak használatára – bővebben lásd a következő szakaszban).

Minden ACE példány egyszerre egy helyen van (persze később elmozoghat).

4.2. Futtatási környezet

Biztonsági és egyéb megfontolásokból definiáltunk egy különleges ACE típust, az SEE-ACE-t (Service Execution Environment ACE – Szolgáltatásfuttatási környezet ACE). Az SEE kötelezően az első ACE az adott

4. ábra Az ACE koncepció



helyen (pl. számítógépen), explicit kell példányosítani, helyváltoztatásra képtelen. Minden, nem SEE ACE szabadon mozoghat azzal a megkötéssel, hogy a mozgás célhelyén már kell hogy legyen minimum egy ACE (ennek a feltételnek az SEE ACE nyilvánvalóan megfelel). Tehát más megfogalmazásban az ACE-ek szabadon mozognak az SEE ACE-ek között.

Mivel ezáltal garantált, hogy ahol ACE van, ott van SEE ACE is, lehetőséget kaptunk arra, hogy bizonyos általános „platform” funkcionalitást az SEE ACE specifikus részében helyezzünk el. Az új helyre mozgó vagy ott más okból keletkező ACE előbb feltérképezi a helyi SEE ACE-t, majd rajta keresztül használja a platform szolgáltatásait.

4.3. ACE modell és a CASCADAS részfeladatok

Tudásháló

ACE modell a tudáshálón keresztül szerez információt a környezetről (kontextusról), és ez alapján építi fel a környezetmodelljét. A tudásháló konkrét kérdésekre válaszol, illetve egy feliratkozásos rendszerű értesítő szolgáltatást nyújt.

Technikai oldalról nézve két modell lehetséges a tudásháló és az ACE-ek viszonyára: vagy minden ACE tagja a tudáshálónak, vagy létezhetnek olyan ACE-ek, melyek nem tartoznak a tudáshálóba (de esetleg használják azt). A jelenlegi álláspontunk a második változatot támogatja, hisz a tudásháló túlzott erőltetése elnehezítené az amúgy pehelysúlyúnak szánt komponenset. Ezért a jelenlegi modell szerint a tudáshálóban résztvevő ACE-ek specifikus részében található meg az a funkcionalitás, amely a tudás tárolásához, rendszerezéséhez, lekérdezéséhez, törléséhez szükséges.

Önszerveződés

Az ACE-ek szoros és laza aggregációt képesek létrehozni.

Laza csoportalkotás esetén az együttműködő felek továbbra is autonóm elemek maradnak, csak valamifajta együttműködésre születtek. A szerződés értelmében lehetséges, hogy az együttműködés idejére egymásról bizalmasabb információik is vannak (például ismerik egymás önmodelljének absztrakcióit), az együttműködés sikeresebbé tétele érdekében. Egyazon ACE több laza csoportban is részt vehet egyidejűleg.

Szoros csoportalkotás esetén megkülönböztetjük a bennfoglaló ACE-t (nevezzük főnöknek) és a bennfoglalt ACE-eket. Az együttműködés lényege, hogy a főnöknek teljes körű hozzáférése és irányítási joga van a bennfoglalt ACE-ek felett. Nem csupán a saját specifikus részéhez fér hozzá, hanem a bennfoglalt elemek specifikus részéhez is. Természetesen ahhoz, hogy ezeket a funkcionalitásokat használni tudja, előbb integrálnia kell a bennfoglalt ön- (és környezet-) modelljeit a saját ön- (és környezet-) modelljébe.

Jegyezzük meg, hogy ebben az együttműködésben a bennfoglalt elemek autonómiája tulajdonképpen elvész, minden döntés a főnök kezében van, a bennfoglalt elemek nem érhetők el többé közvetlenül. Viszont a szoros együttműködés olyan új dolgokat tesz lehetővé, mely laza esetben nem lett volna lehetséges. Ilyen például az új szolgáltatások megalkotása: a bennfoglalt önmodellek integrációja után a főnök szabadon kombinálhatja a bennfoglalt specifikus részekben megvalósított funkciókat.

4.4. Átfogó felügyelet

A felügyeletnek sokkal több információhoz kell hozzáférnie, mint bárki másnak. Szinte mindent monitorozhat: nem csak az üzeneteket, hanem az ACE belsejét is, az önmodellt, a környezetmodellt, a döntés folyamatát és a cselekvést (a kimenetet).

A felügyelt ACE a szerződésben szereplő információkat a felügyelő szerv rendelkezésére bocsátja. A felügyelet elméletileg csak a közös részben található elemeket képes teljesen megfigyelni, hiszen a specifikus részben található elemekkel kapcsolatban semmilyen előfeltevéssel nem élhet (a specifikus rész lehet gépi kódtól a Prolog gépen át egy áruház jósnőig bármi). A specifikus rész monitorozása elméletileg csak a ki/bemeneti csatornákon keresztül (üzenetküldés és -fogadás) lehetséges, az itt tapasztaltak pedig összevethetők az funkcionalitás önmodellbeli absztrakt leírásával. Lehetséges, hogy az ACE nem a teljes önmodelljét bocsátja a felügyelő szerv rendelkezésére, hanem csupán annak egy absztrakcióját – ilyenkor a felügyelet lehetőségei korlátozottabbak (például lehet hogy nem determinisztikusnak észlel egy valójában determinisztikus hibát?).

Természetesen – hogy a rendszer önazonos legyen – a felügyeletet végző elem is ACE, csak szigorúbb (biztonsági) előírásoknak felel meg, mint egy „normál” ACE.

5. Minta forgatókönyvek

Az eredményeket két közérthető mintaalkalmazáson keresztül fogjuk demonstrálni.

5.1. Pervazív tartalomszolgáltatás

A pervazív tartalomszolgáltatás több részből áll, pervazív reklámból, ismerőskeresésből és pervazív információs rendszerből.

A legérdekesebb – és más projektektől leginkább különböző – a pervazív reklám alkalmazás. A városban ACE-ek által irányított, adaptív reklámfelületek vannak elhelyezve (pl. képernyők). A reklámszolgáltatást üzemeltető ACE megfigyeli a környezetében található em-

⁷ Vegyünk alapul egy nagyon egyszerű együttműködési modellt. Normál esetben ha Jancsi ad Juliskának egy almát, Juliska örül. De ha közben Jancsi a haját is meghúzza Juliskának, akkor Juliska mégsem örül. Ha az absztrakt modellben az alma átadásának módja nem szerepel, akkor a felügyeleti szerv nem tudja a hiba okát feltárni, így nem tud javítási javaslatot sem adni.

berek preferenciáit (például a mobiltelefonokon futó, felhasználói preferenciát leíró ACE-eket használva), és a nézőközönséghez leginkább illeszkedő reklámot jeleníti meg. Például a fiatalok egy csoportja előtt a rockkoncert reklámját mutatja, menedzsereknek pedig a legújabb karóramodellt. Természetesen komoly etikai/biztonsági problémák merülnek fel: például hiába preferálná a többség a véres horrorfilm ajánlóját, ha kisgyerek van a közelben, nyilván nem etikus megjeleníteni.

A forgatókönyv kiválóan alkalmas a pervazív, helyzetfüggő elemek demonstrálására.

5.2. Elosztott árverés

Az elosztott árverés alkalmazásánál ha a felhasználó valamit venni vagy eladni akar, felparaméterez egy kereskedő ACE-t és kiküldi a hálózatba. Az ACE a hálózaton bolyong és aukciót indít vagy csatlakozik egy aukcióhoz. A rendszer igen nagyszámú kereskedőt feltételez, ezért az árverések mindig lokálisan folynak. A kereskedő alapvető célja ezért, hogy a számára lehető legideálisabb helyre kerüljön mind hálózati szempontból (gyors hálózati kapcsolat, kis késleltetés a kommunikációban), mind tartalmi szempontból (a lehetséges partnerek közelébe).

A forgatókönyv demonstrálja az önszerveződést, a tudásháló használatát (önszerveződési céllal vagy csak kideríteni, hogy mennyibe került mostanában az adott termék), az előélet-információ használatát (mennyire megbízható a partner, követett-e már el csalást) és az átfogó felügyeletet is.

6. A kommunikációs modell felé

Az ACE kommunikációs modell még kidolgozási fázisban van, jelen fejezet csupán bepillantás jelleggel mutatja be az alapokat.

Az ACE kommunikáció üzenetküldés-alapú. Az ACE tudja saját magáról, hogy mely üzenettípusokat ismeri (környezetmodell). Vannak olyan üzenettípusok, melyeket minden ACE ismer (a szolgáltatás felderítéséhez szükséges üzenetek, szívdobbanás üzenet a felügyeletnek).

Az üzenetek címzése háromféle sémán alapulhat.

(1) *Üzenetszórásos*. Az üzenet mindenkinek/bárkinek szól. Azonban figyelembe kell venni, hogy az üzenet továbbítást is ACE-ek végzik, tehát lehetséges, hogy nyomós ok esetén az üzenet végül mégsem jut el minden, a hálózatban megtalálható ACE-hez (például az elárasztásos túlterhelést elkerülendő).

(2) *Címzéses*. Az üzenet azoknak az ACE-eknek szól, melyek azonosítói egyeznek a címzésben megadottal. Az azonosító nem feltétlenül egyedi, így lehetőség van csoportos címzésre vagy esetleg tulajdonság alapján történő címzésre is. Lehetőség van kézbesítési visszajelzést kérni.

(3) *Közeli*. Az üzenet a közelben található (szomszédos vagy néhány ugrásnyi távolságra található) ACE-ek-

nek szól. Az üzenet küldője nem kell, hogy ismerje a címzettet, a kézbesítés a hálózat struktúráját használva történik. Ez a fajta címzés számos érdekes mellékhatással járhat. Például ACE X kiküld egy üzenetet a közeli elemeknek, a szomszédok közül ACE Y válaszol, ugyanígy, közeli címzést alkalmazva. Viszont a válaszüzenet címzettjei itt már nem biztos, hogy ugyanazok lesznek, mint X üzenetének a címzettjei. Ez megoldható úgy, hogy a közeli címzésnél a középpontot is megadjuk (pl. közeli(X)).

A közeli címzés erősen különbözik az IP feletti világban megszokottól, és láthatóan jól illeszkedik a pervazívítás által támasztott követelményekhez.

7. Összefoglalás

A cikk átfogó képet adott a CASCADAS projekt céljairól és rövid bepillantást engedett a jelenleg folyó munkába (forgatókönyvek, ACE modell, kommunikáció).

Mi az ambiens intelligencia a CASCADAS-ban?

A projekt célja egyfajta ambiens intelligencia modell kidolgozása. Intelligens elemek vannak a hálózat minden pontján és a felsőbb szinteken is; a rendszer elosztottan működik, nem központosított. Az intelligens elemek önálló döntései, együttműködése, csoportosulása és egymást befolyásoló hatása, felügyelete intelligens vonatkozásokat mutat. A rendszer több különböző szinten is autómmá válik (önoptimalizál, öngyógyít, önkonfigurál).

Irodalom

- [1] CASCADAS weboldal:
<http://www.cascadas-project.org>
- [2] F. Sestini,
Situating and Autonomic Communication an EC FET European initiative, ACM SIGCOMM Computer Communication, Review, Vol. 36, Issue 2, April 2006. pp.17–20.
- [3] E. Gelenbe, P. Liu,
QoS and routing in the cognitive packet network, Proc. of the 6th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks ISBN: 0-7695-2343-0, 2005. pp.517–521.
- [4] A. Manzalini, F. Zambonelli,
Towards Autonomic and Situation-Aware Communication Services: the CASCADAS Vision, IEEE Workshop on Distributed Intelligent Systems konferenciakiadványa, Prága, 2006.
- [5] E. Hoefig, B. Wuest, B. K. Benko, A. Mannella, M. Mamei, E. Di Nitto,
On Concepts for Autonomic Communication Elements, IEEE International Workshop on Modelling Autonomic Communications Environments konferenciakiadványa, Dublin, 2006.

Városi közlekedési rendszerek hatékonyságának javítása mobil ad-hoc hálózatok segítségével

VAJDA LÓRÁNT, KARDOS SÁNDOR, GERHÁTH GÁBOR, MEZNY BALÁZS,
LABORCZI PÉTER, GORDOS GÉZA

Bay Zoltán Alkalmazott Kutatási Közalapítvány, Ipari Kommunikációs Technológiai Intézet, Aml Csoport
vajda@ikti.hu

Lektorált

Kulcsszavak: intelligens közlekedési rendszerek, mobil ad-hoc kommunikáció, elosztott rendszerek, valóság-hű szimuláció

Napjainkban mind a járművezetők, mind utasaik részéről egyre nagyobb igény van az utazási időt csökkentő és komfortot növelő rendszerekre, megoldásokra. Lehet gyors és kényelmes autónk, ha éppen fontos találkozóra sietünk, de közben egy dugó kellős közepén veszteglünk. A legtöbb ember első reakciója tehetetlenségében az, hogy felsóhajt: „Bárcsak előre tudhattam volna, akkor biztosan nem erre jöttem volna!”. Munkánk erre a felkiáltásban megfogalmazott, lehetetlennek látszó kívánságra keresi és adja meg a választ. Ebben az esetben rendkívül hasznos lehet az olyan információ, amely alapján mintegy a „jövőbe látva” inkább egy kerülőutat választunk, hogy eljussunk a célunkig. Ezt az előrejelzést megkaphatjuk már indulásunkkor is, és az utunk során folyamatosan tájékozódhatunk az aktuális forgalmi viszonyokról. Fel lehet-e készülni időben egy baleset miatti torlódásra, egy épp aznap kezdődő útlezárásra, ráadásul, ha új helyen járunk, ott ismerjük-e a potenciális kerülőutakat?

1. Bevezetés

Budapesten ezer főre 354 autó jut [1], ami európai viszonylatban nem magas érték, ám azt is figyelembe kell vennünk, hogy 1990 óta 50%-kal nőtt meg a forgalom a Duna hidakon és a városba bejövő autók száma megduplázódott. Ez azt jelenti, hogy évente 30 ezerrel több autó járja Budapest utcáit. Ezen adatok ismeretében (melyeket személyes tapasztalataink csak tovább erősítenek) mind inkább szükséges a közlekedésben olyan rendszereket, megoldásokat alkalmazni, melyek képesek ezt az évről-évre folyamatosan növekvő autós áradatot kezelni. A helyzet egyik megoldását abban látjuk, hogy az autósok alternatív útvonalakat használnak akkor, amikor a főutakon való közlekedés már lassabb és aránytalanul sok időt töltenének el az eredeti útvonalunkon.

Ahhoz, hogy az autósok egyáltalán elkezdjenek alternatív útvonalakat keresni, rendelkezniük kell valamilyen információval a jelenlegi forgalmi helyzetről. Ennek az információnak az eléréséhez kommunikációs képességekkel kell a járműveket felruházni. Az autók közötti kommunikáció és az információk terjesztése kétfajta hálózati elrendezéssel oldható meg. Az egyik, hogy a járművek egy központi egységnek küldik el az információikat, amit bárki lekérdezhet. A másik megoldás, hogy közvetlenül egymással kommunikálnak egy alkalmi (ad-hoc) hálózaton keresztül.

Az autók egymás közti ad-hoc kommunikációjának rendkívül nagy előnye a központosított megoldással szemben, hogy a balesetekre, vészhelyzetekre való reagálás így sokkal gyorsabb. Ahhoz, hogy ezek a szolgáltatások (baleset-megelőzés, forgalomirányítás) elérhetőek legyenek az átlagemberek számára, szükség van mobil ad-hoc kommunikációra alkalmas eszközök-

re, a helymeghatározáshoz pedig a már működő globális helymeghatározó rendszerek (Global Positioning System, GPS) tökéletesen megfelelnek. Magyarországon ma nincsen pontos kimutatás arról, hogy hány darab készüléket használnak az országban. Egyes becslések szerint ez a szám nem haladja meg az 50.000-et, de a piac az elkövetkező 10-15 évben hatalmas ugrás előtt áll (figyelembe véve a nyugat-európai trendeket) [2]. Nem kell megvárunk, hogy minden autóba legyen beépítve ilyen eszköz, mivel már a gépkocsik 10%-ába telepített mobil kommunikációt biztosító készülékek alkalmazásával lehetőségünk nyílik a szolgáltatás megfelelő minőségű használatára [5].

Cikkünkben bemutatunk egy javaslatot az utakon keletkező torlódások enyhítésére. Megvizsgáljuk, hogy milyen formában lehet megvalósítani az információk elosztását, és ezeket a módszereket az általunk megvalósított szimulációs környezetben teszteljük.

2. Városi közlekedési rendszerek

A bevezetőben említett problémákra más országokban is elkezdtek megoldásokat keresni. Több nemzetközi együttműködés is született a témában, úgy az Európai Unió által finanszírozva, mint a nagy autógyárak saját kezdeményezéseként.

Ezen projektek száma és nagysága is mind azt bizonyítja, hogy mennyire fontos és kiemelt szerep jut az intelligens közúti alkalmazásoknak az elkövetkezendő évtizedekben. Így például a FleetNet-ben [3] felvázolták a jövőképet, és elkezdtek konkrét lépéseket is tenni annak érdekében, hogy bebizonyíthassák, hogy a kidolgozott rendszerük nem csupán egy utópia, hanem egy megvalósítható, életképes alkalmazás.

A NoW [7] projektben már piacképes termékek előállításán dolgoznak, amelyhez első lépésként a stabil, biztonságos kommunikáció felállításához és a szabványosításhoz szükséges feladatoknak már neki is álltak.

A Prevent [8] és a C2C [13] projektek leginkább a balesetmegelőzésre koncentrálnak, forgalommenedzsment kérdésekkel nem foglalkoznak. Az Invent projektnek [9] része a közlekedés menedzsment is, de ők csak konkrét, előre tudható útiránnyal rendelkező járműveket (például fuvarcégek autót) irányítanak. Ennek előnye az egyszerűsége és olcsó kiépíthetősége, hátránya, hogy igazából a közlekedési dugók problémáját nem oldja meg, csak valamelyest csökkenti azok kialakulásának valószínűségét.

Az előbb ismertetett projektek tehát a forgalommenedzsment kérdésekre nem helyeznek kellő hangsúlyt (fő kutatási területeik leginkább a balesetek elkerülésére, megelőzésére fókuszálódnak). Mi éppen ezért helyeztük előtérbe a forgalomirányítás módszereit a közlekedési dugók számának csökkentése érdekében. Alapötletünk az, hogy ha a járműveket a megfelelő útvonalakra tereljük, kihasználva a már meglévő úthálózatunk kapacitását, akkor esetenként kerülő úton, de mindenképpen gyorsabban juthatnak el céljukhoz az emberek.

3. Az általunk javasolt architektúra

A fent említett projektek nagyrészt a teljesen decentralizált megoldást részesítik előnyben a többi hálózati megvalósítással szemben. A mi javaslatunk szerint azonban egy hibridnek nevezett megoldás jobb alternatíva az autók közti kommunikáció biztosítására.

Annak érdekében, hogy kiderítsük melyik megvalósítás lesz a leghatékonyabb, megvizsgáltuk a lehetséges információterjesztési stratégiákat:

- **Centralizált:** egyetlen adó/vevő bázisállomás gyűjti az információkat a járművektől; a járművek mindig ettől a bázisállomástól kérdezik le az aktuális forgalmi adatokat.

- **Decentralizált:** a járművek csak egymással kommunikálnak ad-hoc módon, nincsenek nagyteljesítményű rádiós adótornyok, amelyek terjesztenék az információt, itt minden adat az autók közt terjed.

- **Hibrid megoldás:** a hierarchikus és elosztott hálózatok tulajdonságait egyesíti, melyben a járművek egymásnak is és egy központnak is továbbítják üzeneteiket, ami így egy hatékonyabb rendszert valósít meg.

3.1. Az alarendszer felépítése

Az általunk javasolt forgalmi információs rendszer két fő részből áll: az autóba szerelhető *Beépített Egységből* (1. ábra) és a *Központi Egységből* (2. ábra) [10].

A forgalmi információkat és az autó paramétereit a Beépített Egység gyűjti össze. Ilyen adatok például az autó sebessége, vagy GPS koordinátái. Az adatokat csomagok formájában vezeték nélküli hálózaton keresztül küldi el a többi autó felé. Mindemellett, a környező kocsiktól folyamatosan kapja az adatokat, amelyeket (a saját adataival együtt) elraktároz egy adatbázisban, amelyet a kapott üzeneteknek megfelelően frissít.

3.2. Helyi vezeték nélküli kommunikáció

A javaslatunk teljesítményének vizsgálatánál használt vezeték nélküli kommunikáció az IEEE 802.11 [4] protokoll családra alapszik.

3.3. Forgalmi információk terjesztése

Ahhoz, hogy meg tudjuk előzni a torlódások kialakulását, vagy vészhelyzeti információt tudjunk szórni, szükséges az információküldés stratégiájának meghatározása. Ezen algoritmusok hatékonysága nagyban befolyásolja a rendszer működőképességét. Az üzenetküldési stratégia tág határok között változhat.

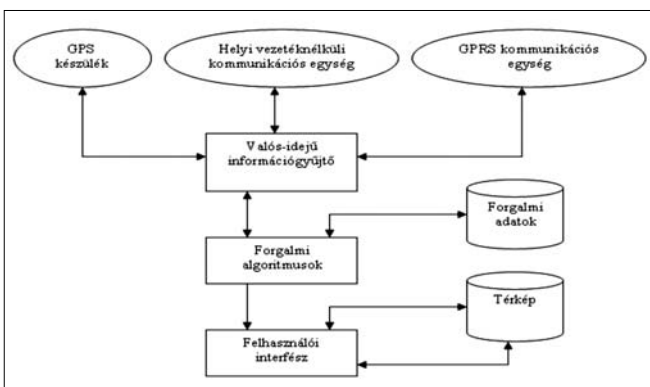
Küldhetünk üzeneteket *meghatározott időközönként*: ebben az esetben meg kell találnunk a középútát. Ha túl gyakran küldünk, nagyon leterheljük a hálózatot. Ezzel szemben, ha ritkán következik be a küldés, nem rendelkezünk elegendően friss információkkal.

Az üzeneteket küldhetjük *eseményhez kapcsolódóan* is: például most haladtunk át egy kereszteződésen, vagy ahhoz képest túl sok ideje vesztglünk egy útszakaszon, hogy a digitális térképünk alapján 50 km/h lenne a megengedett sebesség (dugó alakult ki).

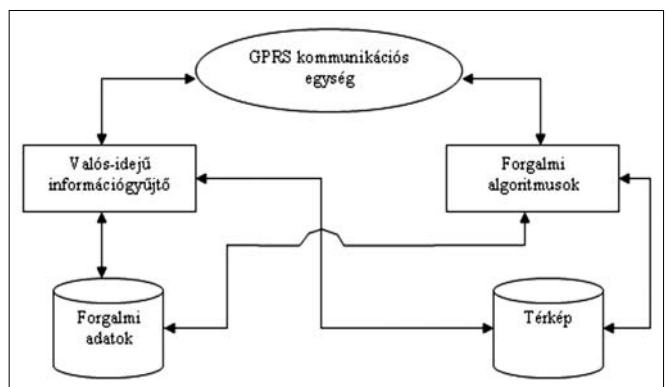
3.4. Az optimális útvonal kiválasztása a térképen

Az autók folyamatosan küldenek egymás között információkat a forgalmi adatok mellett az útviszonyokról,

1. ábra Beépített Egység



2. ábra Központi Egység



esetleges vészhelyzetekről. A forgalmi adatoknál azonban felvetődik a kérdés, hogyan tudjuk a kapott üzenetekből a megfelelő végkövetkeztetést levonni? A megoldás a mi esetünkben az, hogy mindig az adott útszakaszra vonatkozó, az utolsó órában történt eseményeket, adatokat tároljuk el a digitális térképünkben. Az utak közül is csak azokat figyeljük, melyek beletartoznak a kitűzött célunkhoz vezető lehetséges útirányokba. Az így megszerzett információkból kiszámítjuk az átlagos utazási időt az adott útvonalra.

3.5. Globális útoptimalizáció

A telekommunikációs és a közlekedési hálózatoknál hasonló algoritmusokat használhatunk útvonalválasztásra. E két hálózatban használt modellek analógiát mutatnak, mivel a hatékony forgalomelosztásban hasonló kérdések és feladatok merülnek fel. A távközlésben használt forgalommenedzsment kérdésekre (például: torlódás, útvonal-optimalizáció, új útvonalak keresése) a válaszkérés sokkal hangsúlyosabb volt, aminek következtében előrehaladottabb a híradástechnikai hálózatok karbantartása, mint a közlekedési hálózatoké. A forgalom irányításához szükséges alapötletek ezért a telekommunikációban alkalmazott módszerekhez hasonlóak.

4. Szimulációs környezet

Az említett architektúra megvalósításához egy szimulációs környezetet kellett felépítenünk. Szükség volt mind a kommunikációs hálózat, mind a járművek mozgásának szimulálására. Erre a feladatra nem érhető el egy-egy program, ezért saját szimulációs környezetet alakítottunk ki, amelyben egy telekommunikációs és egy közlekedési szimulátort kapcsolunk össze. A szimulátorprogramok kiválasztásában nagy hangsúlyt kapott a realiztikuság és a programozhatóság, hiszen a valós élethez hasonló, azt jól modellező környezetben kapunk reális értékeket.

Felismerve az előbb említett szempontok fontosságát, az autógyárak is használnak forgalomszimulátorokat bizonyos technológiák teszteléséhez, kifejlesztéséhez. A BMW Kutatási és Technológiai Részlege egyesítette a CARISMA nevű közlekedési és az NS-2 nevű távközlő hálózat szimulátor (Network Simulator-2) programokat, a Volkswagen AG pedig összekapcsolta az NS-2 programot a VISSIM közlekedési szimulátorral. A FleetNet projekt során a forgalomszimulátor a Daimler-Chrysler AG által kifejlesztett Videlio program volt.

A többféle kutatási terület és a hozzá kapcsolódó számos szimulációs környezet miatt létrehoztuk a többcélú általános szimulációs környezetet, amelyet *Vidéki és Városi elektronikusan támogatott – Utazás Hálózati Szimulátorának* (Rural & UrBan e-Travelling Network Simulator – *RUBeNS*) neveztünk el.

Sokféle közlekedési forgalommodellező programot vizsgáltunk meg (MITSIMLab, SUMO stb.), hogy a legmegfelelőbbet válasszuk ki, végül a realiztikuság és a

programozhatóság feltételeinek eleget tevő VISSIM elnevezésű forgalomszimulátor programra [12] esett a választásunk. A programkészítők célja az volt, hogy bármilyen forgalmi helyzet gyorsan és pontosan modellezhető, és a valósághoz leginkább hű legyen (3. ábra).



3. ábra VISSIM – működés közben

A RUBeNS információ-terjesztési és -feldolgozási részeit az NS-2 [11] általános hálózat szimulátorban valósítottuk meg. Az NS-2 egy diszkrét idejű szimulátor, esetünkben egymásodperces lépésekben számítja ki a hálózat adatait.

Az NS-2 egy eseményvezérelt, csomagszintű, ingyenes szimulátor, melynek segítségével lehetőség van TCP, útvonalválasztó, többesküldő (multicast), vezetékes, illetve vezeték nélküli hálózatok vizsgálatára.

4.1. A szimulációs modellünk alkotóelemei

A megvalósított szimulációs környezet funkcionális elemekre bontható, melyek egymástól függetlenül végzik feladataikat.

Forgalmi mátrix modul

A városi autóforgalom minél realiztikusabb modellezéséhez, a forgalom generálásához egy forgalmi mátrixot használunk. A mátrix azt írja le, hogy a város egyes kijelölt pontjaiból milyen gyakorisági jellemzők szerint indulnak járművek más kijelölt pontokba (például egy lakótelepről egyes munkahelyekre vagy bevásárlóközpontokba és viszont mennyi időnként indul egy-egy autó). A mátrix megadásához XML formátumú leírófájl állítunk elő, ami jelenleg egyenes vagy exponenciális eloszlású járműindítást tesz lehetővé.

Gráfkészítő modul

A blokk célja az, hogy matematikai háttérrel adjon az útkeresési, útvonalválasztási problémákhoz. A VISSIM úthálózatát felhasználva, létrehozuk az ennek tökéletesen megfelelő gráfot az NS-2-ben, melyben az útkecskésedéseket a gráf a csomópontjai, az utakat pedig a gráf élei reprezentálják. A legrövidebb út keresését az útszakaszokhoz tartozó átlagos utazási idő alapján számítjuk.

Vészhelyzet jelző modul

Ha hirtelen fékeznie kell egy autónak, akkor azonnal jelez a mögötte haladó többi járműnek is a balesetek elkerülése érdekében. A küldést az NS-2-ben található vezeték nélküli kommunikációs modul szimulálja realizáltikusan.

Ad-hoc kommunikációval megvalósított forgalmi információ kezelése

Ez a modul felelős a járművek által létrehozott alkalmi (ad-hoc) hálózatban az üzenetek küldéséért és fogadásáért. Ugyanúgy, mint az előbbi esetben, ezen kommunikációt is az NS-2 valósítja meg.

Térképkészítő modul

A valósághű modellezés céljából egy valós városrész elkészítésére van szükség. A térképek bevitele a VISSIM programba igen nehézkes. Ezt könnyítettük a térképkészítő modul kifejlesztésével, amely egy XML formátumú dokumentumból generál egy, a VISSIM által olvasható térképet.

5. Szimulációs vizsgálat

Szimulációink futtatásához választásunk a Budapesti Műszaki Egyetem (BME) környezetére esett, mivel két híd is van mellette és a hidakon alakul ki leginkább torlódás a viszonylag kicsi átbocsátó képesség és a kerülroutak hiánya miatt. Ezt a térképet fordítottuk le a VISSIM számára. Budapest ezen része a 4. ábrán, a területről elkészült térkép pedig az 5. ábrán látható.

Kiindulási állapotként az 5. ábrán látható útvonalakon haladó járműfolyamokat vizsgáltuk. Látható, hogy az 1. pontból a 2. pontba haladó járműfolyam a Szabadság-hídnál találkozik a 3. pontból a 4. pont felé ha-

ladó járműfolyammal. Itt dugó fog kialakulni. Az 1. és 2. pontok között létrehoztunk egy alternatív útvonalat is, melyet az ábrán fekete színnel jelöltünk. Az 1. pontban 25% eséllyel lép be másodpercenként egy új autó a hálózatba, majd a 2. pont felé halad. Az ábrán látható világos (háromas útvonal) és sötét (négyes útvonal) folyamatok nem nagy forgalmú útvonalak (15 másodpercenként indul egy-egy autó), mivel a célunk az volt, hogy a valós életet minél inkább megközelítsük a többféle forgalommal.

5.1. Szimulációs eredmények

A szimulációkat elvégeztük a kommunikációs architektúra három esetére, illetve arra az esetre is, amikor nem történik információcsere az autók között.

Hagyományos útválasztás

Ebben az esetben nem történik kommunikáció a járművek között. A kiválasztott pontok között a lehető leg-rövidebb útvonalon haladnak az autók, ahogy az az 5. ábrán látható. Az útvonalak meghatározásából várható, hogy torlódás fog kialakulni és jelentős utazási idő növekedésre számíthatunk a hálózatunkon.

Az elvégzett szimuláció eredménye a 6. ábrán látható. Leolvasható, hogy nagyjából tíz perc eltelte után kezdtek feltorlódni az autók és az utazási idő elkezdett emelkedni, míg a szimuláció elejétől számított ötvenedik percnél elérte a maximumát és az útszakaszok teljesen bedugultak.

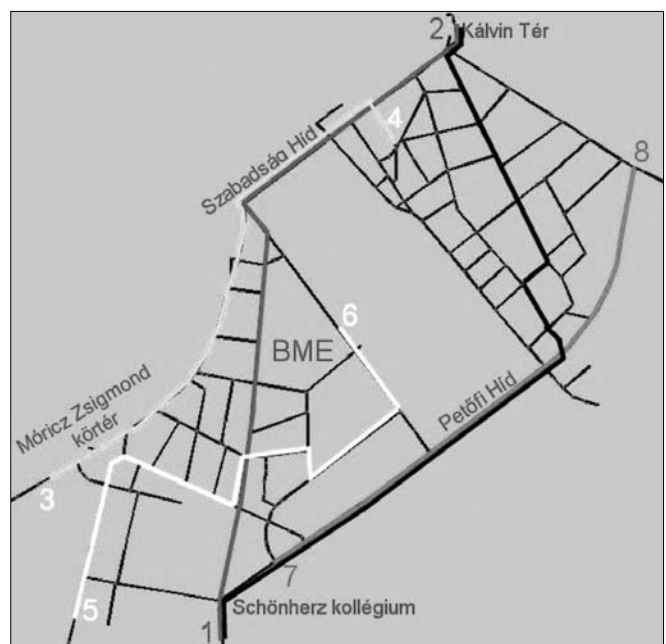
Központosított irányítás

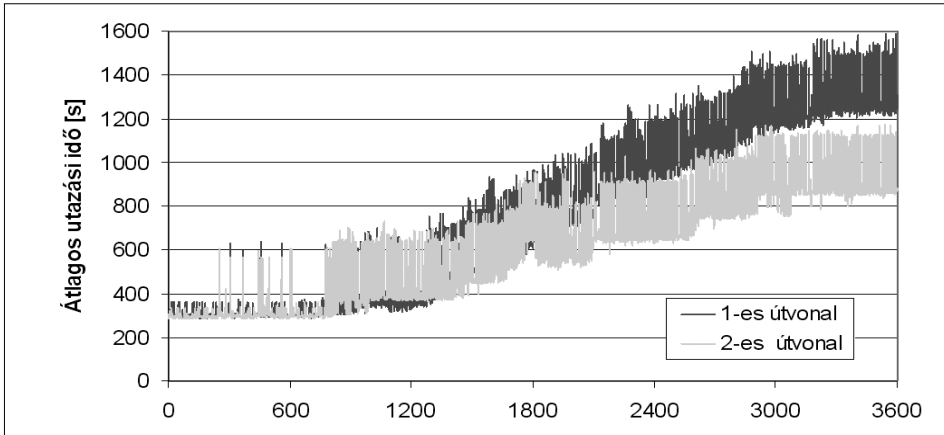
Ebben az esetben a lenti budapesti térképrészleten központosított útvonalválasztást alkalmaztunk. Amikor a Szabadság hídnál torlódás kezd kialakulni, akkor a központ a 2-es útvonalon haladókat a Petőfi híd felé irányítja.

4. ábra Budapest térképének részlete



5. ábra Kiindulási útvonalak



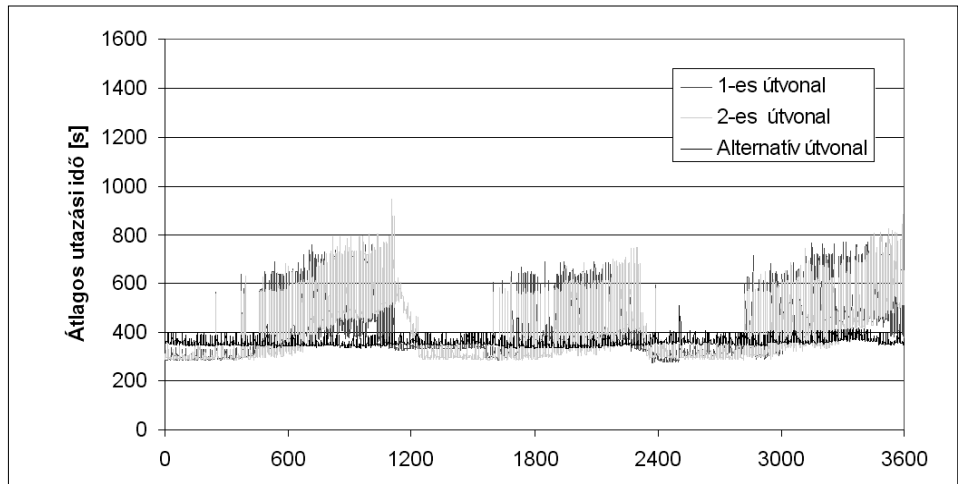


6-9. ábra

Hagyományos eset, központosított irányítás, ad-hoc irányítás és együttműködő útvonalválasztás a budapesti hálózaton

A 7. ábrán látható ennek a szimulációnak az eredménye, ahol az alternatív útvonal a Petőfi híd felé kerülő útvonalat jelzi.

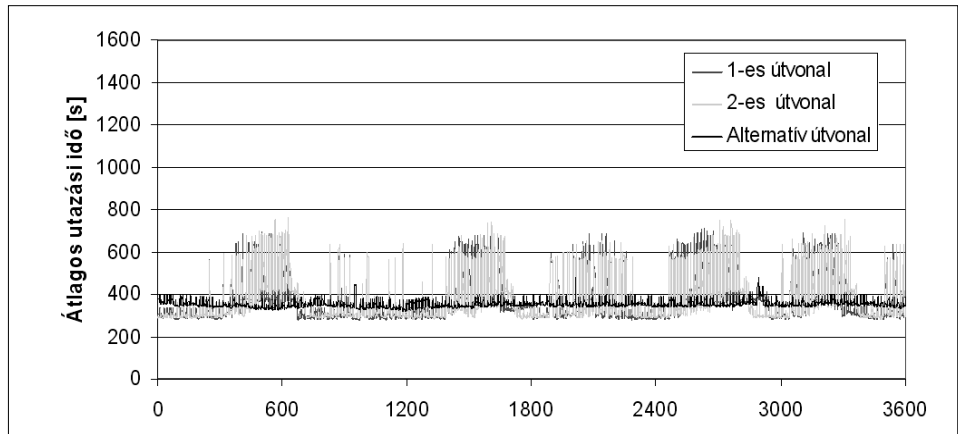
Az ábrán megfigyelhető, hogy az átlagos utazási idő megemelkedik, de ekkor az 1-es útvonalon haladókat az alternatív útvonalra irányítjuk, így nem alakul ki nagyobb torlódás. Az ábrán jelölt alternatív útvonalon a forgalom egyenletes és nincs dugó.



Ad-hoc információ terjesztés

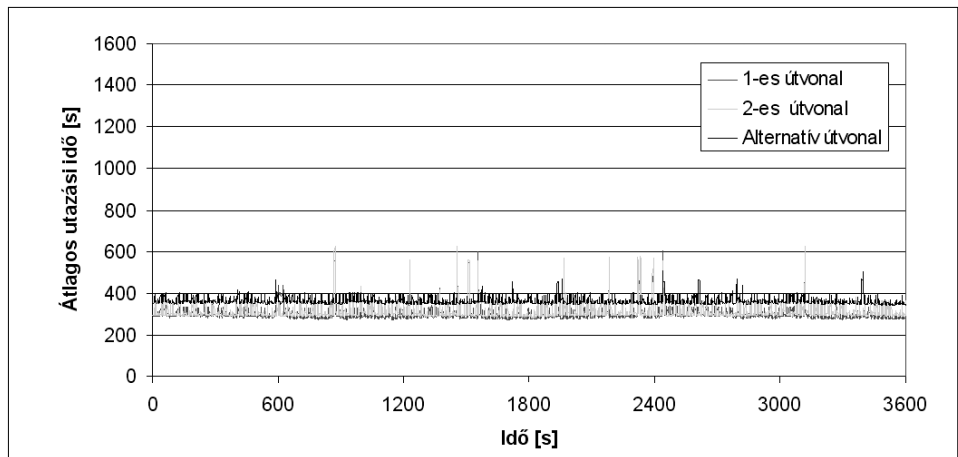
Az elosztott útválasztási döntésekkel végzett szimuláció átlagos utazási idő grafikonja a 8. ábrán található.

Ebben az esetben is elkezdett kialakulni a torlódás, de az autók gyorsabban tudtak alkalmazkodni, mint a központosított esetben, ahol öt perc késés volt az adatbázis frissítésében. Az alternatív útvonalon itt is zavartalanul tudtak közlekedni.



Együttműködő útvonalválasztás

A legmegfelelőbbnek a hibrid, együttműködő irányítási mód bizonyult, hiszen egyenletes forgalomelosztást eredményezett, ahogyan az a 9. ábrán látható. Nem alakult ki torlódás és a járművek utazási ideje is végig egyenletes volt.



	Hagyományos útvonalválasztás	Központosított irányítás	Ad-hoc információ alapján	Együttműködő útvonalválasztás
Átlagos utazási idő [s]	579,99	388,91	364	320,68

10. ábra A budapesti hálózat eredményei

5.2. Az eredmények összegzése

A négy irányítási módszer alkalmazásával kapott eredményeket *táblázatban* gyűjtöttük össze (10. ábra). Itt is látszik, hogy minden megoldás jobb, mint ha nem alkalmazunk intelligens útvonalválasztást.

A központosított és az ad-hoc irányítás közel azonos eredményt hozott. A vártnak megfelelően legjobban az együttműködő, hibrid útvonalválasztás teljesített, aminek segítségével az átlagos utazási idő közel a felére csökkent.

6. Összefoglalás

A nagy, európai projektek a biztonságos közlekedésre helyezik a hangsúlyt, a forgalmi információk ad-hoc terjesztésével, kezelésével kevésbé foglalkoznak. Pedig ha tudjuk előre, hogy előttünk néhány száz méterre dugó alakult ki, akkor még időben felkészülhetünk a fékezésre, illetve, amennyiben lehetséges, kikerülhetjük a kialakult dugókat.

A másik nagy előnye a torlódások elkerülésének az ilyen módon megtakarított idő, mivel sokkal több időt töltünk el egy dugóban, mint ha egy kicsit hosszabb úton (de folyamatosan) haladtunk volna.

A szimulációs eredményeinkkel bebizonyítottuk egy budapesti városrészben, hogy létjogosultsága van a különböző forgalomirányító módszereknek és elképzeléseink szerint minimális beruházással elérhetővé válik az emberek többsége számára. A legjobb eredményt nyilvánvalóan az hozná, ha az autók útvonalát előre tudhatnánk és megmondhatnánk a vezetőnek, hogy melyik utat válassza. De ettől még messze járunk, mivel ehhez szükség volna az emberek teljes együttműködésére, alkalmazkodóképességére.

A forgalom menedzselése természetesen nem csak a városi közlekedésben hoz számottevő javulást, hanem az elővárosi és autópályás közlekedés esetén is. Bár erre most nem végeztünk külön szimulációkat (jövőbeni terveink között szerepel), de általánosságban elmondható, hogy ha városi esetben javulást értünk el, akkor a többi helyen is elérhető bizonyos mértékű utazási idő csökkenés.

Szimulátorunkat annak szellemében építettük fel, hogy minél realiztikusabb, a valós életet leginkább megközelítő szimulációval biztosítsuk eredményeink hitelességét. Ennek érdekében használtuk a VISSIM-et is, és az NS-2-ben kialakított modulokat is úgy terveztük, hogy szimulálja úgy az élet véletlenszerűségét, mint a vezető „kiszámíthatatlanságát”.

Irodalom

- [1] Liberális város, 3.szám, 2006. augusztus, <http://www.demszky.hu/assets/doc/ujzag11.pdf>
- [2] Pricewaterhouse Coopers: Hatástanulmány Magyarország Galileo-programban történő részvételéről, 2005. június 30.
- [3] Dr. Walter Franz: <http://www.et2.tu-harburg.de/fleetnet/index.html>
- [4] <http://grouper.ieee.org/groups/802/11/>
- [5] Lars Wischhof, André Ebner, Hermann Rohling, Matthias Lott, Rüdiger Halfmann: SOTIS – A Self-Organizing Traffic Information System, 57th IEEE Semiannual Vehicular Technology Conf. (VTC 2003-Spring), Jeju, South Korea, April 2003.
- [6] Kendy Kutzner, Jean-Jacques Tchouto, Marc Bechler, Lars Wolf, Bernd Bochow, Thomas Luckenbach: Connecting Vehicle Scatternets by Internet-Connected Gateways, February 14, 2003.
- [7] <http://www.network-on-wheels.de>
- [8] <http://www.prevent-ip.org/>
- [9] Intelligent traffic and user-friendly technology: <http://www.invent-online.de/index.html>
- [10] Laborczi Péter, Török Attila, Vajda Lóránt, Kardos Sándor, Gerháth Gábor: „Vehicle-to-vehicle Traffic Information System with Cooperative Route Guidance”, Intelligent Traffic Systems Conference, Taiwan, 2006.
- [11] The Network simulator: <http://www.isi.edu/nsnam/ns/>
- [12] Ptv simulation – VISSIM: http://www.ptv.de/cgi-bin/traffic/traf_vissim.pl
- [13] Car2Car-Communication Consortium: <http://www.car-to-car.org/>

Ambiens intelligencia a közutakon

CSÁK BENCE

Bay Zoltán Alkalmazott Kutatási Közalapítvány, Ipari Kommunikációs Technológiai Intézet, Aml Csoport
csak@ikti.hu

Kulcsszavak: közlekedés, elektronikus vezérlő egység, ambiens intelligencia, WLAN

Egy ország gazdasági fejlettségének egyik fokmérője az egy főre jutó személy-, illetve haszongépjárművek száma. A személygépkocsi az itteni életforma egyik szimbóluma, míg a haszongépjárművek elterjedtsége jól mutatja, milyen gazdasági erő lüktet. Ezen országokban a munkaerő és az áruk mobilitása alapkövetelmény. Egy ország járműparkja azonban nem csak büszkeségre adhat okot, hanem komoly probléma is, mivel ezek nagy száma pont a saját mozgásukat akadályozza. A közúti balesetek döntő többségét emberi hiba okozza ugyan, de a „hibaforrást” nem lehet kiiktatni. Az Ambiens Intelligencia (Aml) eszköztára erre ad új megközelítést. Olyan rendszerek tartoznak ide, melyek az ember számára nem feltétlenül és közvetlenül érzékelhetőek, de segítségükkel egyszerűbbé, biztonságosabbá válik az élet.

1. Aml-fejlesztések a BZAKK-nál

A Bay Zoltán Alkalmazott Kutatási Közalapítványnál (BZAKK) 2004 decemberében egy új kutatócsoport jött létre azzal a céllal, hogy az Ambiens Intelligencia (Aml) témakörébe tartozó kutatásokat és fejlesztéseket végezzen autópálya-súlyponttal.

A csoport négy alkalmazást dolgoz ki 3 év alatt, melyek közvetve, vagy közvetlenül be kell kerülnenek a közlekedés mindennapjaiba a következő években:

- ad-hoc tempomat
- ráfutás elkerülő rendszer
- intelligens buszmegálló (busz és megálló egység)
- Aml-vel segített vezetés

A felsoroltak közül a második kivételével (ahol csak szimuláció készül) mindegyik gyakorlati alkalmazás, de az elkészülő apparátus segítségével a szimulációt segítő mérések is készülnek majd.

1.1. Ad-hoc tempomat

Az egyszerű tempomat egy olyan sebességszabályozó rendszer, melynek első verziói még a 60-as 70-es években jelentek meg főleg Észak-Amerikában és arra való, hogy a jármű sebességét a vezető által megadott értéken tartsa. A működésből adódóan ez csak egyenes forgalom mellett, főleg autópályán használható zavartalanul.

A tempomat következő változata az ACC (Adaptive Cruise Control), mely ugyan szintén egy, a vezető által meghatározott sebességet tart, de a hozzátartozó radar segítségével fel tudja térképezni az együtt közlekedő járműveket és ezekhez alkalmazkodva képes lassítani, gyorsítani. Természetesen a felelősség itt is a vezetőé, de egy magasabb szintű szolgáltatást kap.

Ezen rendszer nagyon jól megfelel az Európa zsúfolt útjain található vezetési körülményeknek, de a radar előállítási költségei miatt nem nagyon tud tért hódítani. Az ad-hoc tempomat szintén feltérképezi a környe-

zetét, de nem radar, hanem GPS és digitális rádió segítségével. Ezek az eszközök lényegesen olcsóbbak a radarnál és a tendencia egyértelműen látszik, hogy ezek – akárha más okból is, – megtalálhatók lesznek a közeli jövő járművein.

Az ad-hoc tempomat tehát méri saját helyzetét és sebességvektorát a GPS (vagy Galileo) segítségével és megosztja ezeket a rádiója hatósugarában található más járművekkel. Ilyen módon minden jármű „látja” a környezetét és képes lesz nem csak egyszerűen alkalmazkodni, de a rádió segítségével hatni is a körülötte közlekedőkre. Nem feltétlenül szükséges, hogy minden közlekedőnek legyen ilyen tempomatja, mivel a rendszer egyrészt a környező, felszerelt járművekhez viszonyítja magát, másrészt a nem felszerelt járművek a kialakuló rend miatt nagy valószínűséggel felveszik a környezetük tempóját.

Minden komfortelektronikára igaz, hogy a felelősség a jármű vezetőjénél marad. Az ad-hoc tempomat nem tökéletes megoldás, de a költségeihez képest elég jó, így létjogosultsága megalapozott.

1.2. Ráfutást elkerülő rendszer

A ráfutás elkerülő rendszer feladata, hogy figyelemzetéssel és esetleg akár beavatkozással segítse a vezetőt ráfutásos balesetek elkerülésében. Az ilyen balesetek lényege, hogy egy rosszul belátható útszakaszra valamilyen tárgy (lehullott rakomány, műszaki hibás vagy balesetes jármű) kerül, melyet az odaérkező járművek vezetői csak féktávolságon belül érzékelnek és beleütköznek.

Az ilyen helyzetek egy részét el lehet kerülni az ACC segítségével is, amennyiben a radar alapú érzékelés lehetséges (pl. köd, tarlóút füstje). Ha azonban a balesetet okozó tárgy valamilyen radarsugarak által áthatolhatatlan akadály mögött van (például beláthatatlan útkanyarulat, bukkanó), akkor más alapokra kell helyezni a működést.

Az előző szakaszban leírt ad-hoc tempomat apparátusa itt is megoldást nyújthat. Ha egy jármű lassulása a fékezéssel elérhetőnél magasabb értéket mutat, vagy aktivizálódik a légzsák, akkor a ráfutás elkerülő rendszer digitális rádióüzenetben közli a környezetével a jármű helyzetét és eredeti sebességvektorát. A közeledő járművek hasonló rendszerei figyelmeztetik a vezetőket, hogy a közelben baleset van, érdemes lassítani. További lehetőség, hogy ezen járművek továbbítják is az üzenetet a messzebb levőknek, így akár több kilométerre is eljuthat a figyelmeztetés.

1.3. Intelligens buszmegálló

Az intelligens buszmegálló feladata a dinamikus utastájékoztatás. A buszmegálló műholdas helymeghatározás, vagy betanítás útján ismeri pontos helyzetét és digitális rádiója segítségével kb. 1 km-es környezetében mozgó buszok üzeneteit figyeli. A járatok pár másodpercenként közlik helyzetüket és sebességüket, valamint a járatszámukat. Az érintett buszmegálló a hozzá közeledő buszok távolságából és sebességéből kiszámolja annak várható beérkezési idejét, majd ezt szintetizált emberi hangon közli a várakozó utasokkal. A beérkezési idő becslése koránt sem egyszerű feladat, így szükséges, hogy a buszmegálló képes legyen „tanulni” és annak alapján a becslését javítani.

1.4. Aml-vel segített vezetés

Az Aml-vel segített vezetés igen tág fogalom. Itt azt kell érteni, hogy a járművezető vizuális leterheltsége oly módon csökkenthető, hogy bizonyos jármű-, illetve vezetési paramétereikről nem jelzőfény, vagy mutatóállás segítségével kap értesítést, hanem szintetizált emberi hangon. Ez a hang/szóveg a paramétertől és a súlyosságtól függően változhat nemében, magasságában, tempójában vagy akár megfogalmazásában. Egy ilyen „megjelenítő rendszer” akkor igazán hasznos, ha bármilyen egyéb járműrendszer hozzá tud kapcsolódni és szöveget tud neki küldeni.

2. Hardver követelmények a vezérlőegységgel szemben

Az ismertett alkalmazások megvalósítására a Bay Zoltán Alkalmazott Kutatási Közalapítványnál sikerült közös megoldást találni egyetlen alkalmazás-orientált elektronikus vezérlőegység (Electronic Control Unit, ECU) specifikálásával.

A vezérlőegységgel szemben támasztott követelmények sorából az alábbiakban felsorolunk néhányat.

Az alkalmazási környezetből adódóan meg kell feleljen alapvető autóipari követelményeknek. Működési hőmérséklet tartomány: $-40^{\circ}\text{C} \dots +85^{\circ}\text{C}$, kényszerhűtés nem megengedett. Rázásállóság mind alkatrész, mind felépítés szinten és csatlakozók vonatkozásában is. Ez azt jelenti, hogy az ECU-n nem lehetnek gyenge mechanikai felépítésű alkatrészek, vagy olyanok, melyek tömege és súlypontjának helyzete a használat során

túlzott terhelést jelentene a forrasztásokra. A csatlakozók nem rázódnak ki a helyükről és a rázkódás nem okozhatja kopásukat, törésüket, meghibásodásukat. A több áramköri panelből, csatlakozók segítségével felépített egységek nagyon esendőek, így a monolitikus felépítés a megfelelő.

Az áramkör zavartűrése és zavarkibocsátása mind vezetett, mind sugárzott zavarok vonatkozásában szintén megfelelőségi kritérium. Az egységnek széles tápfeszültség tartományban kell működni ($8 \dots 15$ V személy-, illetve $14 \dots 30$ V haszongépjármű esetén, de optimálisan $8 \dots 30$ V). Ki kell bírnia a fordított polaritással való bekötést, az emberi érintésből vagy más tárgytól származó elektrosztatikus kisüléseket és stabilan működni kell az önindító által keltett feszültséglökések mellett.

Az ellátandó feladatokkal kapcsolatos architektúrais elvárásokkal kapcsolatban a legfontosabbak a következők:

A számításgényes feladatok (tanuló algoritmusok, gyors szabályzások, lebegőpontos számítások, beszédgenerálás stb.) megkívánják, hogy az ECU sok tíz MIPS számítási teljesítménnyel bírjon. Ez PC környezetben nem tűnik nagynak, de beágyazott rendszereknél azért már felső kategóriát jelent. A számítások és a programtárolás memóriáigénye 1 MB feletti RAM-ot és 5 MB feletti ROM-ot tesz szükségessé. Ezen felül kell még 2-64 kB nem felejtő, újraírható memória a paraméterek és a hibapló tárolásához.

A beszédgenerálás következménye, hogy legalább egy line-out kimenet álljon rendelkezésre, de célszerű egy 1 W-os audio erősítő felépítése a vezérlőegységen, hogy kis hangerő-igény esetén ne legyen szükség még egy külső erősítőre is.

Mivel az ECU-nak minden alkalmazásban kommunikálnia kell – néhányban akár több módon is – különböző kommunikációs képességekre van szükség. A legfontosabb a digitális rádiókommunikáció. Digitális adatokat kell minél nagyobb távolságra eljuttatni szabad felhasználású frekvencia tartományban. Ez a kritérium a 21. század elején komoly gondot okoz, mivel a szabad felhasználású sávok száma csekély, kapacitásuk korlátos és éppen a szabad felhasználás miatt a kisugárzott teljesítmény olyan csekély kell legyen, hogy 1 km-nél messzebb gyakorlatilag lehetetlen a forgalmazás. Ezen sávok az ISM nevet viselik (industrial, scientific and medical) és eredetileg ezen területek nemcsak rádiós felhasználására szánták, ám végül e sávokban (pl 433 MHz, 866 MHz, 2.4 GHz, 5 GHz) működnek a riasztó távirányítók, a zsinór nélküli telefonok, a drótnélküli számítógépes hálózatok és hasonlóak.

A különböző szabványok tulajdonságai alapján a feladatkörnek legjobban az IEEE 802.11b és g szabványok felelnek meg, melyek alkatrész támogatottsága is elfogadhatónak mondható és éppen ezen támogatottság miatt elterjedtségük is meggyőző. (A közlekedési alkalmazások rádiós támogatására a világ sajnálatosan eléggé felkészületlen, így a választék gyakorlatilag ezen két szabványra korlátozódik egyelőre, míg a 802.11p valóban gyakorlatba nem kerül.)

Az ECU-nak a vezetékes kommunikáció területén is fel kell mutatnia további erőnyeket. A más fedélzeti vezérlőkkel való kapcsolathoz nélkülözhetetlen az autóiparban igen elterjedt CAN kommunikáció, de általános célokra (diagnosztika, billentyűzet) elterjedtsége miatt az RS232 szabvány támogatása is kívánatos.

Beágyazott rendszereknél mindig „jól jön” néhány analóg, vagy digitális ki-, illetve bemeneti csatorna. A megrendelők gyakran kérnek még néhány előre nem látott kiegészítő funkciót, melyeket az ilyen tartákokkal le lehet fedni. A digitális kimenetek egy része legalább képes kell legyen 1-2 A-es terhelés folyamatos meghajtására. További elvárás, hogy ezek némelyikét a processzor PWM (Pulse Width Modulation) kimenetként is tudja használni.

3. Hardvervásárlás vagy -fejlesztés

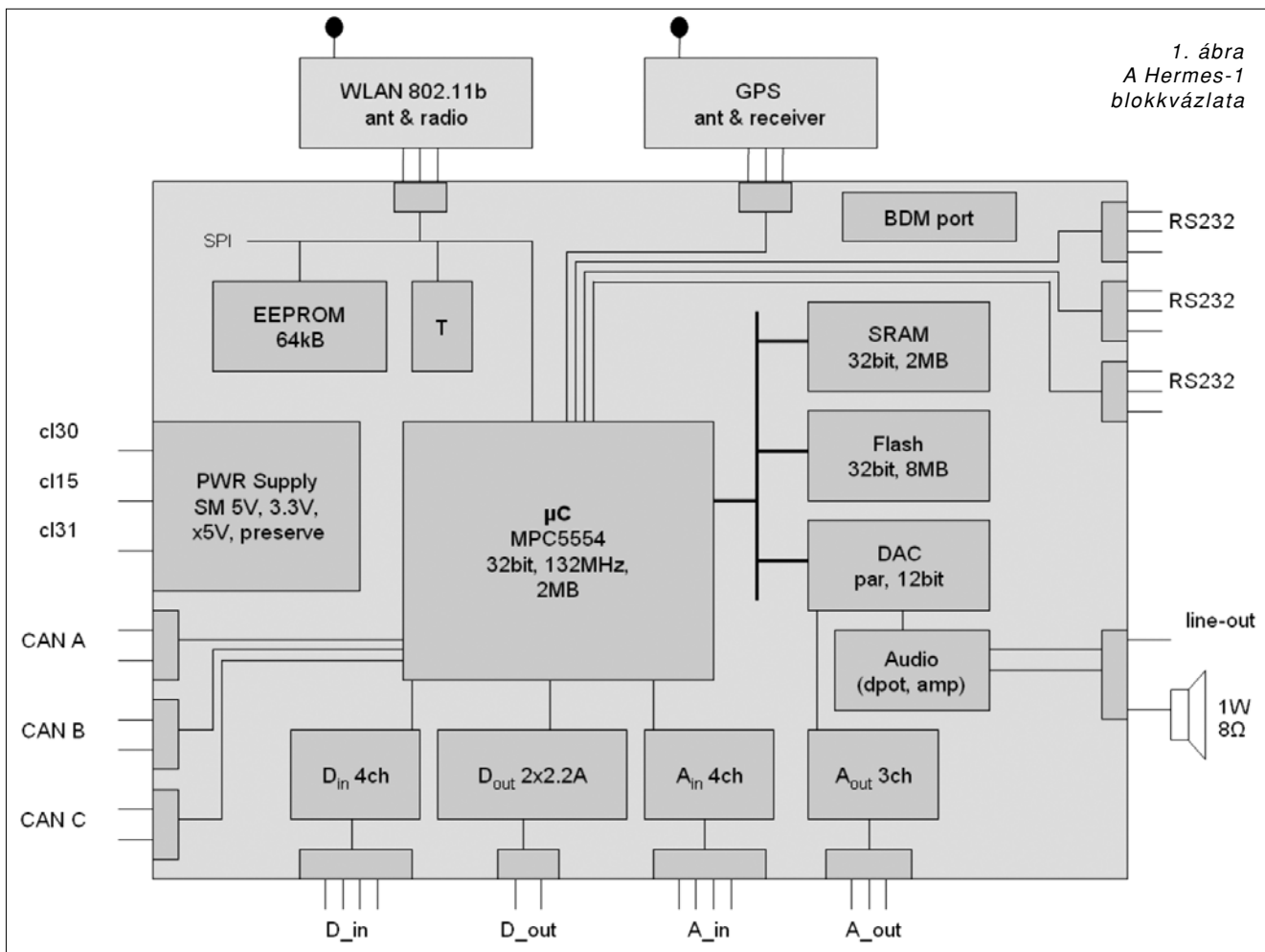
A felsorolt követelmények azt az alapkérdést is rögtön eldöntik, hogy piaci eszközzel, vagy saját fejlesztésű hardverrel kell megoldani a feladatot.

A piacok egy jelentős részét a PC-alapú megoldások jelentik. Ezek előnye a nagy számítási teljesítmény, a jól ismert és támogatott fejlesztői környezet és az illeszthető perifériák sokfélesége. A hátrány a gyakran erősen korlátozott működési hőmérsékleti tartomány (pl.

0°C...+40°C), esetenként a kötelező kényszerhűtés és a monolit felépítés lehetetlensége (az összes kívánt periféria nem található meg egy panelen és a tápegység is külön áll). Ha a piacon kapható megoldások közül a nem PC-alapúakat tekintjük (ezek főleg különböző mikrovezérlők fejlesztői próbakártyái csekély mennyiségű perifériával), akkor cserében a szélesebb hőmérséklet tartományért gyakran le kell mondani a komolyabb számítási teljesítményről, a kívánt méretű memóriáról, a választható perifériák sokféleségéről és ezek ipari kivitelű csatlakoztathóságáról. A tápegység kérdése itt is további probléma, mivel az gyakorlatilag nem kapható a kívánt paraméterekkel, s ha valahogy szert teszünk egy ilyenre, az akkor is egy különálló egység lesz.

Saját fejlesztésű hardver esetén – természetesen megfelelő munkaáldozat árán – az egész vezérlőegység minden tulajdonsága kézben tartható. Az alkatrészek kiválasztása, alkalmazása, elrendezése mind az adott feladatnak vannak alárendelve. A vezérlőegység egyetlen áramköri lapon valósul meg, beleértve a tápegységét is. Az eredmény egy optimális alakú, minimális méretű kompakt áramkör.

A Bay Zoltán Alkalmazott Kutatási Alapítványnál a saját fejlesztésű vezérlőegység használata lett a követett megoldás. A vezérlőegység a Hermes-1 nevet viseli. Nagyobb termetű öccse (Hermes-2) a képfeldolgozás terén fog komolyabb lehetőségekkel kiegészülni.



4. A Hermes-1 vezérlőegység

4.1 Követelmények

A saját fejlesztés melletti döntést követően került sor a részletesebb specifikálásra. Az előző oldali, 1. ábrán látható az ennek eredményeként elkészült blokkvázlat.

A követelmények alapján a vezérlőegység specifikált processzora a Freescale MPC 5554 lett. Ezt eleve autópári használatra szánták, ott is a nagy számítási igényű alkalmazásokhoz, mint amilyen a motorvezérlés, vagy a fékirányítás. Főbb paraméterei a 132 MHz maximális órafrekvencia, mely akár 132 MIPS számítási teljesítményt is jelenthet, a 32 bites adatszélesség, a 96 kB belső SRAM és a 2 MB belső flash memória. További előnye, hogy 3 különálló CAN protokoll-áramkört integráltak rá és 2 USART egységet, melyeket az RS232 kommunikációhoz lehet könnyen használni. Két külön SPI (Serial Peripheral Interface) busz segíti a szinkron soros kommunikációt és 64 (!) capture/compare egység nyújt széles szolgáltatásokat mindenféle időzítés alapú feladathoz. Ez utóbbiak a processzor sebességével társulva akár azt is lehetővé teszik, hogy egy-egy RS232 porttá alakuljanak.

A processzor belső memóriáit külső áramkörök egészítik ki. Így további 8 MB flash ROM és 2MB SRAM került az ECU-ba. Ezek a 8, 16 és 32 bites elérést is támogatják. Paraméter- és hiba-memóriaként a processzor egyik SPI buszára illesztett soros elérésű 64 kB kapacitású EEPROM került. Ugyanezen SPI buszra csatlakozik egy hőmérő áramkör, mely a vezérlőegység saját hőmérsékletét mérheti és a fejlesztések tesztelési fázisában szolgál fontos adatokkal.

A processzor-buszra került a nagy sebességű 12 bites AD átalakító, melynek 1usec beállási ideje bőven lehetővé teszi a legnagyobb mintavételezési frekvenciák használatát is. A 12 bites adatszélesség nem hi-fi minőség, de az alkalmazások igényeinek megfelel. Az AD átalakító után egy szűrőáramkör, egy digitális potencióméter, valamint egy 1 W-os audio erősítő következik.

A digitális rádió nem szerves része az ECU-nak. Ennek fő oka, hogy az antenna és a vezérlőegység általában külön helyezkedik el és sodort érpárt könnyebb vezetni, mint koaxiális kábelt. Ha az antenna közelében van az általa kiszolgált rádió modul, akkor onnan már sodort érpáron lehet az adatokat az ECU-ba vezetni. A vezérlő egység egy dedikált SPI portot használ a külső antenna modullal való kommunikációra differenciális érpár meghajtó közbeiktatásával. Ez a megoldás a zavarérzékenység csökkentéséhez szükséges.

A GPS modul választás során sikerült egy olyan eszközt találni, melynél az antennába van telepítve az egész modul, így szintén elkerülhető a koaxiális kábel használata. Az integrált GPS antenna modullal RS232 protokollon zajlik a kommunikáció a modul és a vezérlőegység között.

További három RS232 port került a specifikációba, hogy egy billentyűzet és egy intelligens kijelző illesztése után még egy diagnosztikai portja is maradhasson az ECU-nak.

Mivel a processzor három független CAN csatornát támogat, ésszerű ráfordítás mindhárom kihasználása és a vezérlőben így három hozzájuk tartozó fizikai illesztő feléptése.

Három analóg kimenet került specifikálásra, melyek a 0...5 V tartományban mozognak. A feszültségek előállítására a processzorból érkező nagyfrekvenciájú PWM jelek szűrésével történik az egyszerűség kedvéért. A későbbi kísérletek során ilyen analóg kimenettel sikerült jól érthető beszédhangot is létrehozni.

A négy analóg bemenet diszkrét elemekből felépített védőkapcsolásokból áll. A jelek digitalizálása a processzorban történik 12 biten.

A négy digitális bemenet az analóg bemenetekhez hasonlóan került meghatározásra, ám a jelek a processzor capture/compare egységeihez vannak kötve.

A négy digitális kimenet specifikálásánál fontos volt, hogy ne csak mint jelkimeneteket lehessen használni, hanem akár 2 A fogyasztású elektromos eszközök kibekapcsolásához is.

A vezérlőegység tápegységének specifikálásánál, majd tervezésénél számos szempontot kellett szem előtt tartani. A választott mikrovezérlő három tápfeszültséget (1.5 V, 3.3 V, 5 V) igényel, melyek be- és kikapcsoláskor adott rend szerint kell viselkedjenek. Mivel az MPC 5554 képes külső tranzisztor segítségével a 1.5 V tápfeszültség előállítására, csak a 3.3 V és az 5 V előállítása marad külső áramkörökre. Ezek a már említett 8...32 V tartományú bemenetről kell működjének, ami a vezérlőegység ~5 W fogyasztása mellett csak kapcsolóüzemű elvet enged meg.

További elvárás a tápegységgel szemben, hogy állandó bemeneti táplálás mellett, csak a gyújtás ráadásakor kapcsoljon be és a gyújtás elvételekor ne azonnal, hanem a processzor irányítása mellett kapcsoljon csak ki. A kikapcsolt állapothoz tartozó áramfelvétel nem haladhatja meg a 3...5 mA-t.

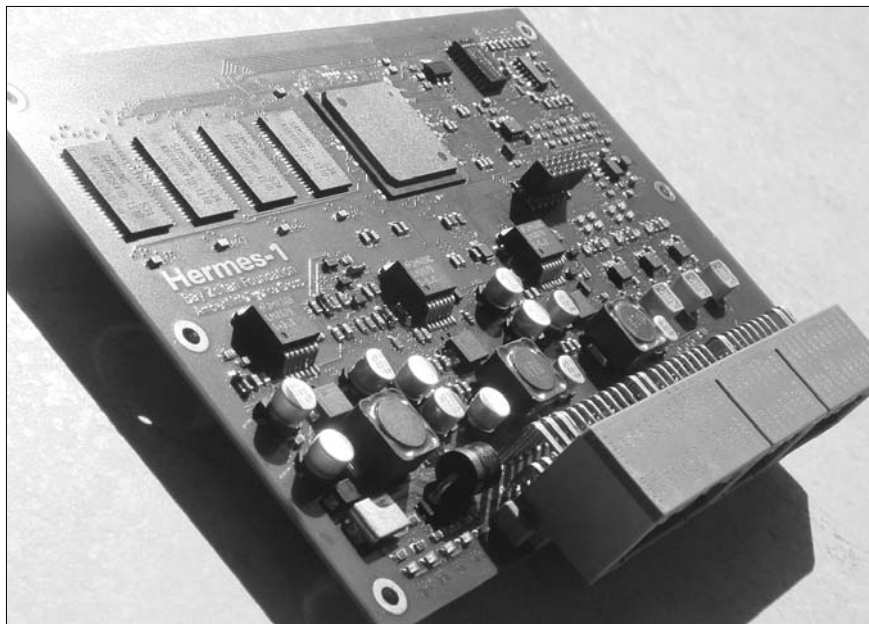
Sok alkalmazásnál van szükség külső modulok táplálására (mint esetünkben a WLAN antenna modul, illetve a GPS modul), így a specifikáció része lett még egy 5 V, 1 A kimenetű kapcsolóüzemű tápegység.

4.2. Megvalósulás

A Hermes-1 megtervezéséhez komolyabb CAD rendszerre volt szükség, mivel például az MPC5554 416 lábú BGA tokozásban kapható, melyben a forraszlabdák átmérője 0,5 mm és ezek 1 mm-es rácsba szerveződnek. A processzor tokozása már önmagában is indokoltá teszi a 4, de inkább 6 rétegű nyomtatott huzalozást, ám a 32 bites cím- és adatbuszok ezt csak megerősítik.

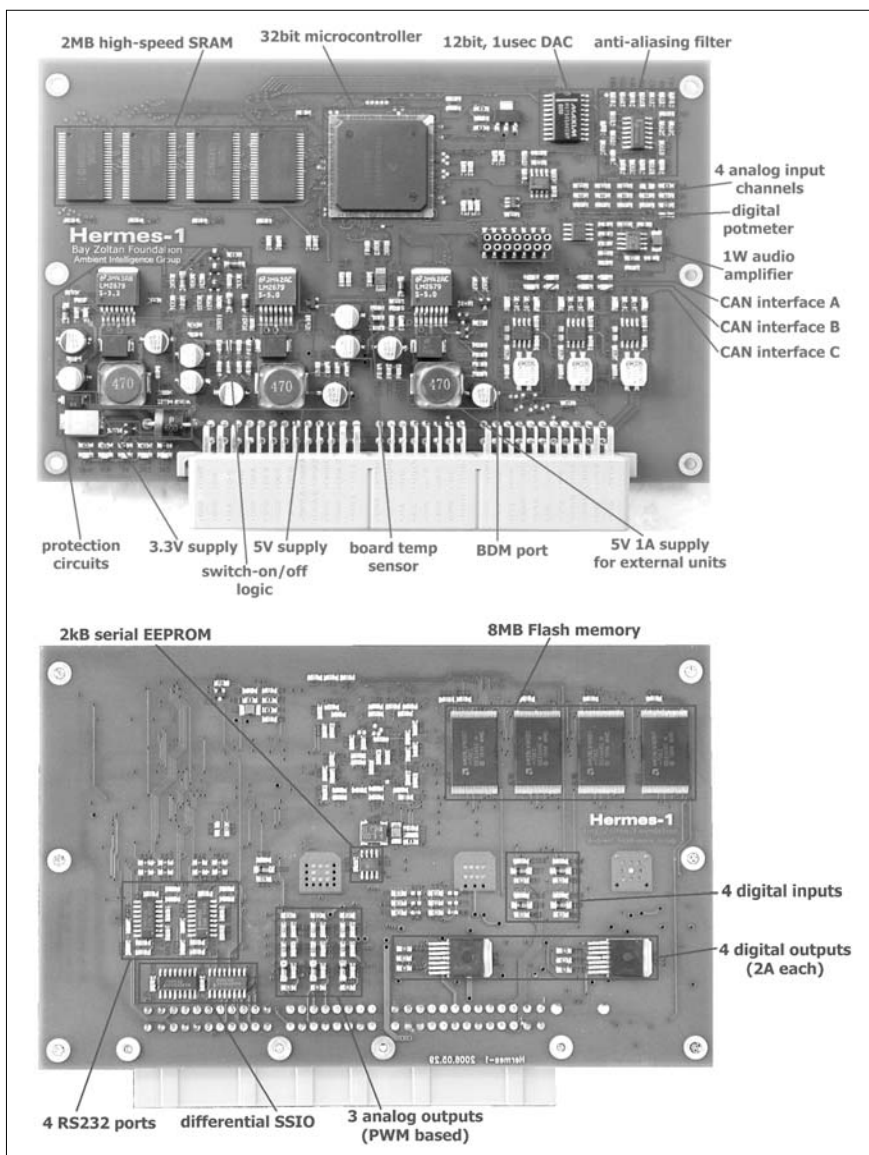
A 174 mm x 112 mm területen 291 alkatrész került az alkatrész- és 163 alkatrész a forrasztási oldalra. Ezek között 1364 összeköttetés van, melyet – a földelések nélkül – összesen 5 méternyi vezetékvezetés valósít meg. Összesen 1879 forrasztás van a vezérlőegységben, melyből 1803 felületszerelt alkatrészt köt be.

A Hermes-1 fényképe a 2. ábrán, az ECU alkatrész-és forrasztásoldala pedig a 3. ábrán látható.



2. ábra A Hermes-1 fényképe

3. ábra Az ECU alkatrész- és forrasztásoldala



5. További lépések

Időközben a Hermes-1 használatával lezajlottak az első mérések az intelligens buszmegállóval kapcsolatban, melyek beigazolták az előzetes méréseket és elképzeléseket. A XI. kerületi Fehérvári úton 200-500 m távolságból sikerült két személyautó között kapcsolatot létesíteni és a feladat szerinti működést produkálni. A végleges alkalmazásban az antennák magasabb és jobb rálátást lehetővé tevő elhelyezése várhatóan jobb eredményeket is ad majd. A vezérlőegység szoftvere pillanatnyilag már minden fő funkciót lehetővé tesz (felderítés, kapcsolatfelvétel, azonosítás, beérkezési idő becslése, beszédgenerálás), de még sok munka van a befejezésig.

Folyamatban van az első tíz darab gyártása, hogy az alkalmazások fejlesztése párhuzamosan és több szempontból (szoftver fejlesztés, rádiós mérések, járműves integráció) folyhasson.

A tervek és a vevőkkel kötött szerződések szerint a vevői tesztesre, illetve próbaüzemre 2007 második harmadában kerül sor.

Webszerver a mobiltelefonon – a 3G tartalommegosztás új formája

DÉVAI ISTVÁN, BÁNYÁSZ GÁBOR, LEVENDOVSKY TIHAMÉR

BME Automatizálási és Alkalmazott Informatikai Tanszék
{banyasz.gabor, tihamer}@aut.bme.hu

Kulcsszavak: 3G, mobil tartalommegosztás, mobil webszerver, mobil peer-to-peer alkalmazások

A 3G hálózatokhoz kapcsolódni képes mobiltelefonok elterjedésével egyre inkább jelentkezik az a felhasználói igény, hogy a készüléket ne csupán böngészésre, hanem tartalommegosztásra és más mobiltelefonokkal való IP alapú együttműködésre is lehessen használni. A fenti célok elérésére számos kutatási és fejlesztési irány létezik (peer-to-peer alkalmazások, azonnali üzenetküldők), azonban a vizsgálat tárgyát képező mobil webszerver koncepció rendkívül könnyű alkalmazásfejlesztést tesz lehetővé. Egy tesztkörnyezet összeállítása és teljesítménymérések elvégzése után a mobil webszerver életképes alternatívának látszik: szerver oldalon felhasználja a hatékony, objektumorientált Python szkriptnyelvet másrészt nem igényel speciális kliens oldali alkalmazást, csupán egy szabványos web böngészőt. Mindezek mellett a gyakorlatban is használható, akár 10-15 felhasználó egyidejű kiszolgálását lehetővé tevő teljesítményt nyújt. Ezért reményeink szerint a közeljövőben a mobil webszerver technológia számos új alkalmazással bővítheti a 3G készülékeket.

1. Bevezetés

A már hazánkban is széleskörűen elérhető 3G hálózatok nem egyszerűen gyorsabb internetelérést tesznek lehetővé, hanem teret engednek olyan új hálózati alkalmazások megszületésének, melyek kihasználják a mobiltelefon erőforrásait, a készülék változó földrajzi pozícióját és mobiltelefonok adhoc létrejövő és megszűnő csoportjait. Cikkünkben bemutatjuk, hogy miként lehet egy jelenleg is kapható 3G mobiltelefon-családon webszervert futtatni, és milyen módon lehet elérhetővé tenni ezt a szolgáltatást az internet nagyközönsége részére.

A rendszer technológiai ismertetése után részletezzük, milyen teljesítményre képes a jelenleg még fejlesztési stádiumban lévő szoftver, valamint milyen fejlesztési és tesztkörnyezet szükséges ahhoz, hogy új, web alapú mobiltelefon alkalmazásokat készítsünk. Végül bemutatjuk a jelenleg elérhető alkalmazásokat és valószínű jövőbeni fejlesztési irányokat.

2. Hardver és szoftver környezet

Az Apache alapítvány által fejlesztett webszerver [1] a szabadszoftver-mozgalom egyik jelentős eredménye, a

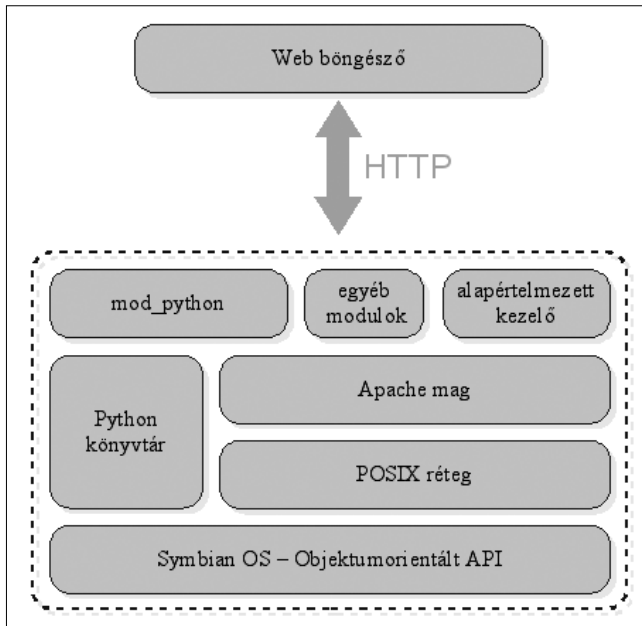
Netcraft adatai [2] szerint jelenleg az internetes oldalak közel 70%-át szolgálja ki. A C nyelvű forráskódot eddig is számos beágyazott rendszerre (többek közt ARM, PowerPC és MIPS) portolták, azonban erőforrásigénye valamint a mobil hálózati sávszéleség hiánya miatt mobiltelefonokon nem volt használható. Szintén probléma volt, hogy a telefonok beágyazott szoftvere nem nyújtotta az Apache számára szükséges POSIX [3] kompatibilis környezetet, ezért a portolás számos kód teljes újrainítását jelentett volna.

A Nokia 3G termékvonala azonban már mind memória, processzor teljesítmény valamint szoftver környezet tekintetében alkalmas arra, hogy az asztali gépeken elterjedt szerverszoftvert futtassuk rajta. Az 1. ábrán látható táblázatot tekintve láthatjuk a közelmúltban megjelent telefonok hardverspecifikációját.

További pozitívum a C és C++ nyelvű programozhatóság és a POSIX szabványt részben megvalósító API elérhetősége. Mivel mobil környezetben is szükséges dinamikus weboldalak generálása, ezért előnyös az is hogy, a Python szkriptnyelvnek [4] már létezik működő portja az S60-as sorozatú telefonokon.

1. ábra A felhasznált készülékek specifikációja

	Nokia 6630	Nokia N80
Processzor	220Mhz ARM architektúra	Texas Instruments OMAP 220Mhz
RAM	10 Megabyte	40 Megabyte
Háttértár	MMC kártya	miniSD kártya
Kijelző	176*208 pixel	352*416 pixel
Hálózat	WCDMA, Bluetooth	WCDMA, Wifi, Bluetooth
Operációs rendszer	Symbian OS V8.0s	Symbian OS V9.1



2. ábra A mobil webszerver szoftverkönyvtára

A Nokia Research Labs által elkészített, „Racoon” [5] nevű, Symbian operációs rendszeren [6] futó Apache port a 2. ábrán láthatóak szerint épül be a mobil szoftverkönyvtárba.

Az ábrán látható szoftverek szerepe:

- *Apache mag*

A webszerver funkciókat ellátó kód, mely fogadja a webböngészők TCP/IP alapon érkező HTTP szabványú kéréseit, értelmezi azokat és visszaküldi az igényelt tartalmat (fájlt vagy valamilyen tartalmat előállító kód kimenetét).

- *Apache modulok*

Lehetséges az apache képességeit bővíteni különböző modulokkal. Például beállítható, hogy ha a `http://localhost/camera` oldalra hivatkoznak, akkor ezt ne az alapértelmezett kezelő válaszolja meg (amely a „camera” nevű fájl tartalmát próbálná elküldeni), hanem a `mod_camera` modul kezelje le. Ez a modul futáskor az apache processz részeként fog futni, és egyaránt elérheti az Apache belső API-ját és a Symbian API-t is. Konkrétan a fenti esetben a Symbian API segítségével elkészíthető egy fénykép a mobiltelefonba integrált kamerával, majd az Apache API segítségével továbbíthatjuk az elkészült képet a felhasználó felé.

- *Python csomag*

A Python kódot értelmezni és végrehajtani képes programkönyvtár valamint az azt parancssorban is elérhetővé tevő alkalmazás. A Python Symbian-változata meglepően sok szolgáltatást teszi elérhetővé a telefonnak: szkript fájlokból kezelhető a beépített kamera, készíthető grafikus kezelőfelület, kezdeményezhető hívások, elérhető a teljes telefonkönyv, naptár stb.

- *Mod_python*

Olyan Apache modul, amely felhasználja a Python programkönyvtárat olyan módon, hogyha a felhasználó egy Python szkript fájlra hivatkozik, akkor a `mod_python` azt előbb futtatja és a futás eredményét küldi a felhasználó felé.

náló felé. Hasonlóan a PHP-hoz, a Python is lehetséges HTML kódba beágyazni és dinamikus webalkalmazások készítésére használni (Python Server Pages).

- *POSIX réteg*

A Symbian OS felhasználói interfészét POSIX kompatibilis programkönyvtárként nyújtó réteg (mely nem a Racoon, hanem a Symbian SDK része). Segítségével például lehetővé válhat, hogy szabvány BSD socketek segítségével valósítsunk meg TCP/IP kapcsolatot stb.

3. Az átjárhatóság problémája

A fenti alkalmazásokat lefordítva, majd telepítve azonnal futásra kész a webszerver, elindítása után a telefonon található webböngészőbe a `http://127.0.0.1` oldalon máris láthatóak a kiszolgált oldalak.

Amennyiben 3G kapcsolattal rendelkező mobiltelefonon fut a szoftver, úgy elvárható, hogy az oldalakat az Internet felől is meg lehessen tekinteni, ennek azonban az vet gátat, hogy a szolgáltató a telefonokat az internetes támadásoktól védetten, egy saját belső hálózaton tartja.

Ennek jellemzői általában:

- Belső IP címek (a tűzfal hálózati címfordítást – Network Address Translation, NAT – végez)
- A belső IP címek nem tartoznak egy-egy telefonhoz vagy előfizetéshez, azok időben mindig más és más felhasználókhoz vannak hozzárendelve.
- Még ha ismerjük is egy telefon belső IP címét, a szolgáltató tűzfala nem ad arra lehetőséget (pl. port-forward), hogy a telefontal közvetlen kapcsolatba lépünk TCP/IP segítségével.

Hazai viszonylatban a T-Mobile szolgáltatását tesztelve megerősíthetjük a fenti állításokat hozzátéve, hogy az alkalmazott tűzfal az egyik legszigorúbb, teljesen szimmetrikus [7] típusú. Ennek következménye, hogy a kapcsolatot mindig a telefonnak kell kezdeményezni, hiszen a teljesen szimmetrikus tűzfal nem fog az Internet felől ismeretlen TCP/IP kapcsolatokat a telefonok felé továbbítani. Mivel egy szerveralkalmazásról beszélünk, ezért ebből az is következik, hogy szükség van egy olyan, a tűzfal Internet felőli oldalán található számítógépre, mely az „átjáróállomás” szerepét vállalja és az Internet felől érkező kéréseket egy folyamatosan fenn tartott, a mobiltelefon által kezdeményezett kapcsolaton keresztül továbbítja a telefonkészülék felé.

Az így felépülő rendszer látható a 3. ábrán.

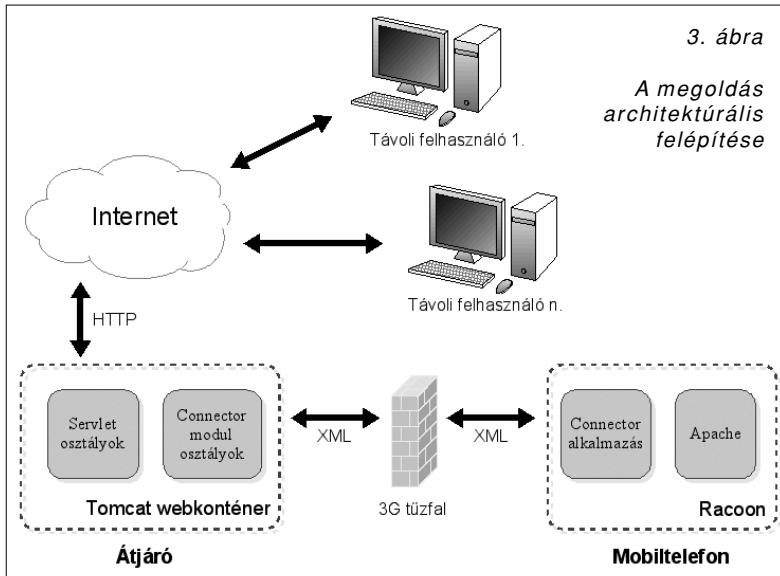
Az architektúra a következőkből áll:

- *Apache webszerver*

Az előzőekben bemutatott Symbianon futó Apache változat illetve a hozzá tartozó modulok, szkript interpreterok stb.

- *Connector Module*

Ez egy Symbian alkalmazás, mely indításkor TCP kapcsolatot kezdeményez az átjáróval majd ezt az átjáró segítségével a működés során folyamatosan fenn is tartja (az átjáró meghatározott időközönként életben tartó „ping” üzeneteket küld a telefon felé). Mivel ezen



3. ábra
A megoldás architektúráis felépítése

kapcsolat kétirányú, az átjáró bármikor értesítheti a Connector Module-t, miszerint egy Internet felhasználó bizonyos lapokat szeretne lekérni a webszervertől (így valósítható meg az átjárás a tűzfalon). A kommunikáció a Connector Module és az átjáró között egy speciális, XML alapú protokoll segítségével történik.

• **Átjáró**

Olyan Tomcat alkalmazáserveren [8] futó webalkalmazás, mely a felhasználó szempontjából reverse-proxy-ként működik, azaz a kért weboldalt elkéri a mobiltelefonon futó webszervertől, majd ezt a felhasználó számára transzparens módon továbbítja.

A megvalósítás egy kérés kiszolgálása esetén a 4. ábrán vázoltak szerint működik.

Amikor a mobiltelefonon elindítjuk a webszervert, a Connector modul egy vezérlőkapcsolatot épít ki az átjáróval (1) TCP protokoll segítségével, XML üzenetek felhasználásával. Az üzenet tartalmazza a felhasználónevet, jelszót, az életbentartó ping üzenet (2) gyakoriságát, valamint azt, hogy a webszerver maximum hány párhuzamos kérés kiszolgálására képes. Ez utóbbi paraméter jelentősége, hogy a Connector modul a vezérlő

socketen kívül még annyi párhuzamos TCP csatornát létesít (3) az átjáróval, amennyi párhuzamos letöltésre lehet számítani.

Ezután ha egy Internet felhasználó a `http://atjaro/~mobilfelhasznalo/` címet szeretné megtekinteni webböngészőjében, először egy szabványos HTTP kérést küld az átjáró felé, amelyet a Tomcat alkalmazáserverben található webkiszolgáló fog kezelni. Amennyiben „mobilfelhasznalo” névvel nincs mobiltelefon bejelentkezve, úgy egy „A felhasználó sajnos nincs online” tartalmú oldal jelenik meg a böngészőben.

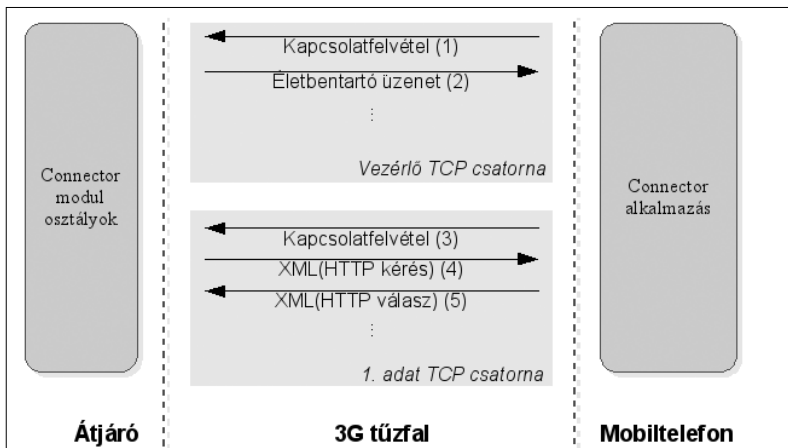
Amennyiben olyan oldalt kérnek le, amelyhez bejelentkezett felhasználó tartozik, úgy az átjáró a már említett XML alapú protokoll segítségével eljuttatja a kérést (4) a mobiltelefonon futó Connector modulhoz. A modul ezután az XML üzenetből előállítja az eredeti kérést, lefuttatja azt a webszerveren és a server választát szintén XML üzenetként az egyik adatcsatornán visszaküldi (5) az átjárónak, ami továbbítja az eredeti kérést feladó Internet felhasználónak.

Első pillantásra az átjáró csupán egy kényszermegoldásnak tűnhet, amivel a 3G szolgáltató tűzfalán át lehet jutni, azonban a gyakorlat azt mutatja, hogy számos esetben kifejezetten hasznos egy ilyen közvetítő a telefon és az Internet böngészői között:

- Mivel a telefon erőforrásai igen korlátozottak (lásd később a teljesítményértékeket), mindenképp szükséges a tartalom gyorsítótárazása (caching). Ez csökkenti a processzor és a hálózat terhelését a telefonon, s mivel természetesen a 3G hálózati forgalmat a telefon tulajdonosa fizeti, így csökkenti a mobil webszerver fenntartásának költségeit is.
- Az átjáró segítségével lehetséges a felhasználók földrajzi pozíció szerinti csoportosítása.
- Amennyiben a telefon pillanatnyilag nem kapcsolódik az átjáróhoz, a már említett weboldal jeleníthető meg (vagy a legutolsó tárolt állapot).

4. Mérési környezet és eredmények

4. ábra HTTP kérések és válaszok átvitele a 3G tűzfalon

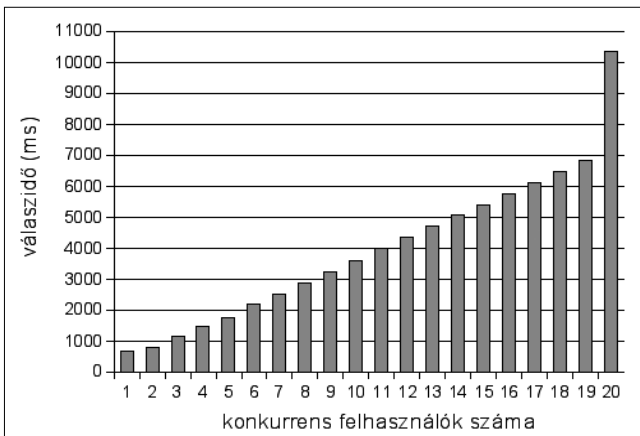


Olyan tesztkörnyezet létrehozása volt célunk, melynek segítségével megismerhető a rendszer működése, teljesítménymérések végezhetők rajta, valamint akár új mobil web alkalmazások fejlesztésére is lehetőség nyílik. Természetesen egy olyan élő SIM kártyával, amelyen engedélyezett a 3G szolgáltatás, azonnal használatba lehet venni a Racocon alkalmazás csomagot (akár saját átjárót üzemeltetve akár a Nokia által biztosítottat használva), azonban teszteléshez, alkalmazásfejlesztéshez jóval kényelmesebb közvetlen TCP/IP kapcsolatot kialakítani a telefon és a helyileg üzemeltetett átjáró között.

Az általunk használt Nokia 6630 és N80 típusok esetén ezt a következőképpen tudtuk megoldani:

- Nokia 6630-as esetén Bluetooth technológia segítségével lehetséges a közvetlen IP kapcsolat létrehozása. Mivel a gyári alapszoftverek nem támogatják ezt, egy közösségi fejlesztésű GnuBox nevű [9] alkalmazást kell telepíteni, ami a telefonon egy olyan speciális kapcsolatot hoz létre, mely a telefonon futó alkalmazások számára egy vonalkapcsolt internetelésnek látszik. Az internet vagy helyi hálózat felé egy Bluetooth adapterrel felszerelt Windowsos vagy Linuxos PC használható (Windows esetén a közvetlen kábelkapcsolat funkció használatával, Linux-nál PPP szervert futtatva).

- N80-as telefon esetén jóval egyszerűbb a helyzet, hiszen a készülék képes Wifi hálózatokra kapcsolódni, így egy helyileg üzemeltetett Access Point segítségével máris elérhetővé válik a helyi hálózat számára. (Mivel teljesítménymérést is szeretnénk volna végezni, nem egy külön hardveres AP-t, hanem egy PCMCIA WLAN kártya segítségével szoftveres AP-t üzemeltettünk, ezáltal elkerülve, hogy a helyi hálózati forgalom stb. befolyásolja a mérést. Ilyen szoftveres Access Point létrehozására Linux esetén a MadWifi, illetve HostAP driverek [10] segítségével van mód.)



5. ábra A webszerver teljesítménye 3G hálózaton, N80-as készülék esetén

A tesztkörnyezet összeállítása után mindkét telefont közvetlen, Bluetooth, illetve Wifi általi IP kapcsolat segítségével teszteltük. A teljesítménymérést a httpperf [11] illetve apachebenchmark [1] szoftverekkel végeztük. Mivel a helyi hálózat nem jelentett akadályt a sávszélességben, a mobiltelefon folyamatosan képes volt több, mint 10 kérés/sec teljesítményre is (ekkor egy kérés átlagos válaszára kb. 250-300 ms). Az igazi próbát természetesen a 3G hálózaton való tesztelés jelentette, mely esetén a mérési eredmény az 5. ábrán látható.

A mérés során fontos észrevétel volt, hogy a párhuzamos adatkapcsolatok számát célszerű 3-4 között tartani, mivel ez alatt a telefon még nem nyújt optimális teljesítményt (sokat kell a hálózati átvitelre, az átjáró válaszára stb. várni). Ezen érték felett pedig a lineárisnál némileg jobban nő a válaszára, ezáltal a párhuzamos kéréseket célszerű inkább az átjáró által sorosítani, mint újabb adatkapcsolatokat nyitni.

5. Jelenlegi alkalmazások

A Raccoon programcsomag már jelenleg is számos mobil webalkalmazást tartalmaz:

- *Képkészítés igény szerint*

Ha egy távoli böngésző a mobil weboldalon a megfelelő gombra kattint, a telefonon a 6. ábrán látható üzenetablak jelenik meg. Amennyiben engedélyezzük a képkészítést, akkor a távoli felhasználó azonnal láthatja az elkészült képet a böngészőben.



6. ábra

Képkészítés megerősítése a mobiltelefonon

- *Helyinformáció lekérdezése*

Mobile Network Code, Mobile Country Code, Local Area Code, Cell ID azonosítók.

- *Kik tartózkodnak a közelben?*

Bluetooth hálózaton a telefon képes észlelni, ha egy másik Bluetooth eszköz van a közelben. Amennyiben az eszköz szintén mobil webszervert futtató telefon, azonnal lekérdezhető a honlap címe és így oldalról oldalra vándorolva megismerhető a környék összes mobil weboldala.

- *Azonnali üzenet küldése*

A 7. ábrán látható üzenetablakban azonnal megjelenik az elküldött üzenet, hasonlóan egy SMS-hez. (Lehetőség van arra is, hogy a postafiókban hagyjunk üzenetet).



7. ábra

Azonnali üzenet fogadása

• *Hozzáférhető a telefonon tárolt adatok*

A weben megtekinthető (jelszó ellenében) a teljes telefonkönyv, rendelkezésre állnak a könyvjelzők, a naptár valamint a szöveges üzenetek. (Megléhetősen hasznos szolgáltatás abban az esetben, a ha otthon felejtjük a telefont, vagy ha kényelmesen szeretnénk új bejegyzéseket felvenni.)

6. A jövő alkalmazásai

Véleményünk szerint a mobil webalkalmazások egyik legfontosabb feladata, hogy elősegítsék az utóbbi években egyre erőteljesebben jelentkező igényt a mobil tartalomkezelésre. Egyrészt várhatóan számos szolgáltatás építhető a tartalom létrehozása, szűrése és csoportosítása köré: a felhasználók a mobilkészülék segítségével fogják létrehozni a tartalmat (képek, telefonkönyv, naptár, szöveges és hangüzenetek, napló, videoklipek stb.), és ezt mobil webtechnológia segítségével juttatják el az átjáró, majd a felhasználó felé. Másrészt mobil környezetben is megkezdődik az egyes webes tartalmak konvergenciája: a mobil weboldalunkon nincsenek szolgáltató által megszabott korlátozások, bárki küldhet nekünk szöveges, hang vagy videóüzeneteket és válaszképpen mi is bármilyen tartalmat szolgáltatathatunk az oldal segítségével.

Például egy sítúra után nem a felhasználó küldi el barátainak a mobiltelefonnal készített képeit, hanem azok automatikusan letölthetőek lesznek a telefon mobil weboldaláról (mobsite), akár még a sítúra alatt. Ha esetleg valaki elfelejt digitális kamerát vinni a sítúrára, akkor sincs gond, mivel lekérhető a közelben tartózkodó mobil webszervert futtató felhasználók (mobile user – muser) listája, és letölthetők képek a többiek oldaláról. Amennyiben a felhasználó eltéved a sítúráját keresve, beállítható, hogy ha a rendszer egy másik, magyarul beszélő síelőt észlel a közelben, akkor riasszon, így azonnali üzenetben segítség kérhető.

A 3GPP streaming (IP alapú mobil video) már jelenleg is realitás a legtöbb 3G telefon esetében, azonban nemsokára már az ellenkező irányban is lehetséges lesz mindez (hasonlóan a jelenleg is működő videotelefonáláshoz, azzal a különbséggel, hogy a nézőközönség akár a teljes világháló lehet). A lokalitás alkalmazása is számos lehetőséget rejt magában: lekérhető lesz a közelünkben tartózkodó érdekes pontok (Point of Interest, POI) listája és az azokhoz tartozó weboldal (például ha szükségünk van egy könyvre, az összes közelben levő könyvesbolt weboldala elérhető). Azonban nem csupán mi találhatunk meg másokat, hanem mások is minket: egy rendezvény helyszínén tartózkodó tömeg részére a rendezvény szervezője elküldheti a rendezvény képeit, szintén a helyinformációra alapozva.

Természetesen ilyen fokú földrajzi pontossághoz már nem elég csupán a mobiltelefontól lekérdezhető cellainformáció: vagy a szolgáltató oldaláról szükséges támogató vagy a földrajzi pozícióra más módon érdekes következtetni. Például ha mobilkészülékek egy cso-

portja eléri egymást Bluetooth segítségével, akkor az átjáró következtethet arra, hogy a telefonok egymástól maximum 10-100 méterre találhatók. A rendszernek ezenkívül számos ipari alkalmazása is lehetséges, például rendkívül könnyen fejleszthető felületet biztosít különböző mérési adatok gyors publikálására és az átjárón történő aggregálására.

Összefoglalásképpen a mobil weboldalak fő ismérve, hogy interaktív kommunikáció folytatható a weboldal látogatói és tulajdonosa között, kihasználható a földrajzi pozíció, a tartalom a látogatók igénye szerint folyamatosan változtatható. A mobil weboldalak hasznos tulajdonsága, hogy nem csupán explicit hivatkozások vannak az oldalak között, az átjáró számos szempont szerint automatikusan tudja csoportosítani azokat (például a két felhasználónak vannak közös ismerősei a készülék telefonkönyvében).

Szintén rendkívül érdekes alkalmazási területeket ad az, hogy a rendszer segítségével tetszőleges formátumú, szöveges, hang, kép, videóállományok cserélhetők a készülék és a világháló között. Ennek eredménye lehet, hogy számos, jelenleg a mobil hálózat által nyújtott szolgáltatás (pl. rövid szöveges üzenetek) a közeljövőben már IP alapon vehető igénybe, a cikkünkben bemutatott rendszer ehhez visz egy újabb lépéssel közelebb.

Irodalom

- [1] Apache webszerver
<http://httpd.apache.org/>
- [2] NetCraft – webszerver statisztika
http://news.netcraft.com/archives/web_server_survey.html
- [3] POSIX –
IEEE Portable Applications Standards Committee
<http://www.pasc.org/plato/>
- [4] Python szkriptnyelv
<http://www.python.org/>
- [5] Raccoon szoftvercsomag
<http://research.nokia.com/research/projects/mobile-web-server/index.html>
- [6] Jo Stichbury,
Symbian OS Explained: Effective C++ Programming for Smartphones (Symbian Press)
<http://www.symbian.com/>
- [7] Saikat Guha, Paul Francis,
Characterization and Measurement of TCP Traversal through NATs and Firewalls, Cornell University
<http://nutss.gforge.cis.cornell.edu/pub/imc05-tcpnat/>
- [8] Tomcat alkalmazáserver
<http://tomcat.apache.org/>
- [9] GnuBox
<http://gnubox.dnsalias.org/gnubox/>
- [10] MadWifi és HostAP eszközmeghajtók
<http://madwifi.org/>
- [11] Httpperf – webkiszolgálók teljesítménymérő szoftvere
<http://www.hpl.hp.com/research/linux/httpperf/>

Kisfogyasztású integrált áramkörök tervezési kérdései

VARGA LÁSZLÓ, HOSSZÚ GÁBOR

Budapesti Műszaki és Gazdaságtudományi Egyetem, Elektronikus Eszközök Tanszék
hosszu@nimrud.eet.bme.hu

Lektorált

Kulcsszavak: kisfogyasztású CMOS áramkör, adiabatikus töltésátvitel, töltésvisszanyerő kapcsolás

Napjaink elektronika alkalmazásainál a kis fogyasztás alapvető követelmény, különösen telepes készülékeknél. Ilyenek például az emberi implantátumok, az alkalmi (ad-hoc) érzékelő hálózatok, valamint egyéb, hordozható berendezések. Ezeknek a követelményeknek a kielégítésére lehetőséget nyújtanak az adiabatikus töltésvisszanyerő kapcsolások, amelyek fő jellemzője, hogy az áramköri kapacitások feltöltése és kisütése Joule-veszteség nélkül történhet. A cikk első része bemutatja az ismert adiabatikus áramkörök egy javított változatát, majd annak alkalmazását kisfogyasztású dinamikus tárolók megvalósítására. A második rész a kisfogyasztású áramkörök tervezésének különleges feladatát, a működés optimális ütemezését tárgyalja.

1. Bevezetés

A nagy bonyolultságú integrált áramkörök teljesítményfelvételének csökkentésére több ismert módszer létezik (telepfeszültség csökkentés, jelváltások számának minimalizálása), azonban ezek egyikével sem érhető el nagyságrendi csökkenés. Ha az áramkör sebessége nem túl kritikus (mint ahogy ez több alkalmazásban előfordul), akkor az adiabatikus kapcsolások nyújtják a legkedvezőbb megoldást [1]. A hőtanból átvett „adiabatikus” elnevezés arra utal, hogy a generátor által szolgáltatott energia veszteség nélkül alakítható át az általa feltöltött kapacitás energiájává. Továbbá az így feltöltött kapacitás töltését visszanyerve és valamilyen formában ideiglenesen tárolva, az a következő ciklusban újra felhasználható.

2. Az adiabatikus töltésvisszanyerő logikai kapcsolások működése

Ha egy kapacitást a szokásos ugrásfüggvénnyel töltünk fel, akkor a kapcsoló soros ellenállásán eső feszültség miatt jelentős (Joule) teljesítmény-veszteség lép fel. Ha a feltöltést egy olyan feszültségforrás végzi, amelynek feszültsége csak infinitezimálisan nagyobb a kapacitás aktuális feszültségénél, akkor a töltőáram, és ezzel a soros ellenálláson fellépő teljesítmény is a zérushoz tart. Ez tehát egy igen jó hatásfokú feltöltés, hátránya viszont, hogy a művelet ideje eközben a végtelenhez tart. Látható tehát, hogy így nem valósíthatók meg nagysebességű áramkörök. Egy közbülső megoldás az 1. ábrán látható rámpafeszültség alkalmazása, amely jelentős fogyasztáscsökkenés mellett még elfogadható működési sebességet enged meg.

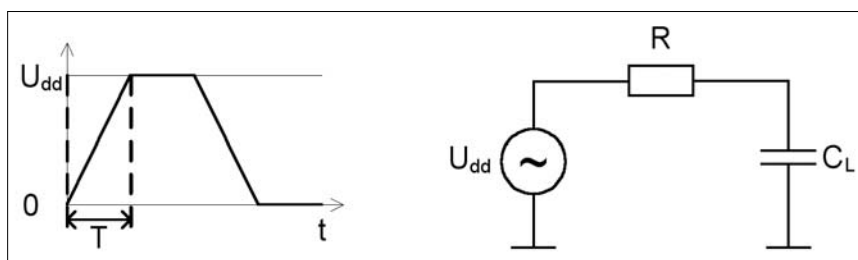
Ha az U_{dd} bemeneti feszültséget lassan emelve a C_L kapacitást T ideig töltjük, úgy, hogy T sokkal nagyobb, mint a kapcsolás RC időállandója, akkor a kapacitás feszültsége szorosan követi a bemeneti feszültséget és a kapacitásba befolyó áramot közel állandónak tekinthetjük: $I \approx (C_L \cdot U_{dd}) / T$. A töltési folyamat energiaveszteségét a következő képlet fejezi ki (1):

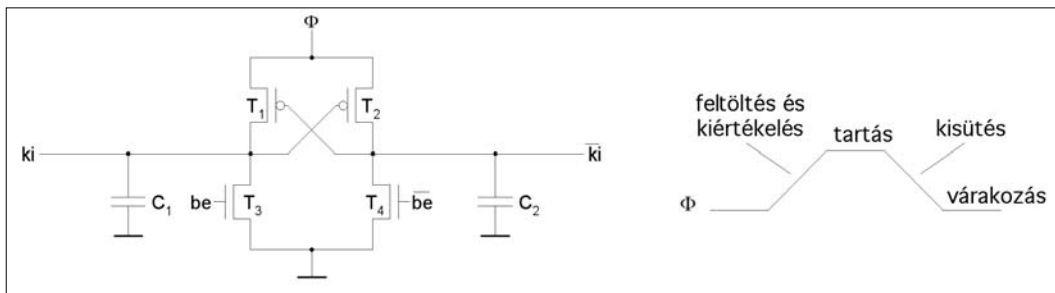
$$E_{\text{veszteség}} = P \cdot T = I^2 R \cdot T = \left(C_L \frac{U_{dd}}{T} \right)^2 R \cdot T = \frac{R \cdot C_L}{T} C_L \cdot U_{dd}^2$$

Az (1) képlet az úgynevezett adiabatikus veszteség értékét adja. A gyakorlatban a lassan emelkedő jellel történő táplálást rezonáns ütemező (fázis-) jelekkel valósítják meg, amelyeknél a kapacitásból visszanyert töltést átmenetileg egy induktivitás tárolja. A logikai magas szintet alkotó energia így mintegy „hintázik” az áramkör és az induktivitás között, az áramköri kapacitásnak és így a rezgés frekvenciájának állandóságához pedig mind a ponált, mind a negált logika megvalósítása szükséges.

Az irodalomban található adiabatikus töltésvisszanyerő megoldások közül egyszerűségében és működési sebességében kitűnik a $2N-2P$ kapcsolás [2], amelynek invertere a következő oldalon, a 2. ábrán látható. Az ábrán látható a működéséhez szükséges fázisjel elvi alakja is, amely a feltöltés és egyben kiértékelés, a tartás, a kisütés és a várakozás periódusokból áll.

1. ábra Adiabatikus töltés szemléltetése





2. ábra

A 2N-2P inverter és négy szakaszból álló fázisjele

A $be=1$ esetén T_3 kinyit és C_1 a földre kapcsolódik, míg T_4 lezár és így C_2 lebeg. Ha a felfutó Φ fázisjel értéke eléri a T_2 pMOS tranzisztor küszöb feszültségének abszolút értékét ($|U_{t,p}|$), akkor T_2 kinyit, és innentől C_2 adiabatikusan töltődik. A negált kimenet feszültsége követi a fázisjel értékét. T_2 kinyitásának pillanatában egy rövid időre T_1 is kinyit, azonban C_2 feszültségének emelkedésével azonnal le is zár, így C_1 feltöltetlen marad, mivel T_3 még mindig nyitott.

Amint Φ fázisjel eléri a legnagyobb értékét, a kimeneteken érvényes logikai szint van (tartási periódus). A kisütési periódusban C_2 a feltöltéshez is használt útvonalon keresztül adiabatikusan kisül, egészen addig, míg a lefutó Φ fázisjel értéke $|U_{t,p}|$ alá nem csökken, és T_2 le nem zár. Ekkor a C_2 kapacitásban $|U_{t,p}|$ feszültség marad, ami nem gond, ha a következő fázisjel-ciklusban ugyancsak a C_2 lesz feltöltve. Ha azonban a fázisjel következő felfutásakor a kapu logikai állapota megváltozik, akkor a C_2 töltése a T_4 -en keresztül nem-adiabatikusan sül ki, amelynek energiavesztesége $E=0.5 \cdot C \cdot (U_{t,p})^2$. Ugyanekkor nem-adiabatikus veszteséget okoz az is, hogy feltöltéskor a pMOS tranzisztor nem nyit ki azonnal.

3. Javított tulajdonságú adiabatikus kapcsolás

A javított tulajdonságú adiabatikus kapcsolás alapinvertere, valamint a vezérlését és egyben a tápellátását is biztosító fázisjele a 3. ábrán látható [3]. Működése a 2N-2P kapcsoláshoz hasonló, azzal kiegészítve, hogy feltöltéshez itt két áramút áll rendelkezésre. Az egyik a meglévő pMOS tranzisztoron halad keresztül, a másik pedig a hozzáadott T_3 , illetve T_4 tranzisztoron, amely által a feltöltendő kimenet kezdettől fogva követi a fázisjel értékét, így feltöltéskor nem jön létre nem-adiabatikus veszteség.

A javított tulajdonságú adiabatikus kapcsolás és a 2N-2P kapcsolás órajel-ciklusonkénti energiaveszteségét a frekvencia függvényében, különböző terhelőkapacitás értékek mellett a 4. ábrán láthatjuk. A szimuláció ideális fázisjel-generátort feltételezve trapéz alakú fázisjelekkel, egyenlő feltöltés, tartás, kisütés és várákozás idővel az AMS 0,8 μm -es technológiáján 100 darab láncba kapcsolt inverteren történt. Az inverterek láncba kapcsolásához négy, egymástól 90° -kal eltolt fázisjelre van szükség.

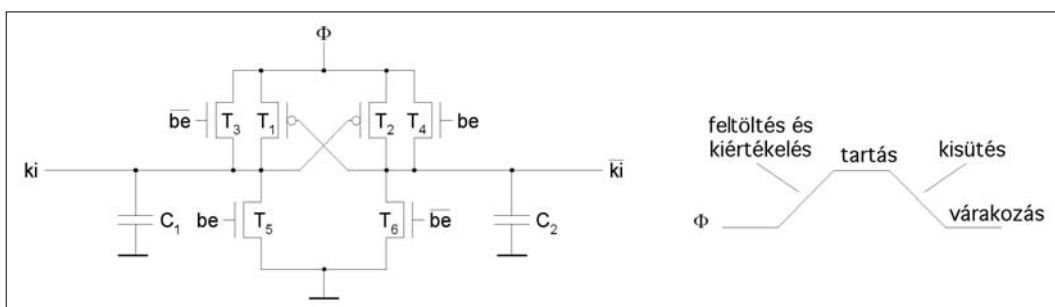
A 2N-2P kapcsolás energiavesztesége kis frekvenciákon igen jelentős, ami a lassú kapcsolási idő miatt a hosszabb ideig keresztbefolyó áramokból adódik. A javított tulajdonságú kapcsolás energiavesztesége a 2N-2P kapcsolásénál kedvezőbb, mindamellett, hogy a működési határfrekvenciája is magasabb. A működés felső határfrekvenciájához közeledve az energiaveszteségnek a működési frekvenciától való függése az addigi lineáris helyett négyzetes lesz, mivel az adiabatikus töltési feltétel már csak kismértékben teljesül.

Az energiaveszteség tovább javítható teljesen adiabatikus kisütéssel, amihez egy külön kisütő áramútra van szükség. Az ezt megvalósító nMOS tranzisztor fázishelyes vezérlését a következő logikai fokozatban elhelyezett inverter adja, csak a fázisjel kisütés szakaszában engedélyezve azt.

4. Adiabatikus elven működő tárolóelem kialakítása

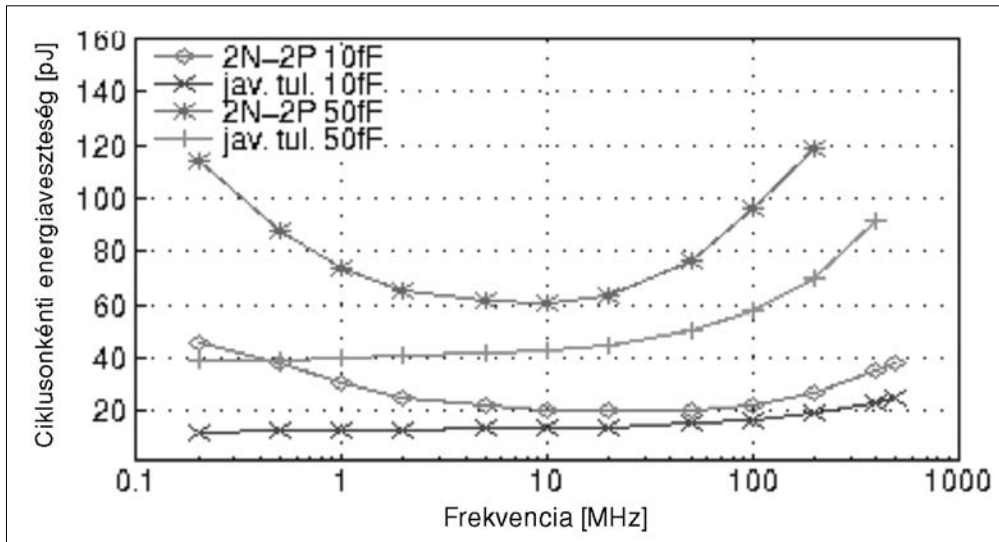
Adiabatikus elven működő kapcsolással statikus tárolóelem kialakítása nem lehetséges, mivel a vezérlő fázisjel egyben a kapcsolás tápvezetékét is jelenti, így ennek alacsony szintje alatt az áramkör nem táplált.

A bemutatott adiabatikus kapcsolások négyfázisú működésűek, ahol egy logikai kapu kimenete a bemenelethez képest negyed periódussal késik, ezáltal a ne-



3. ábra

A javított tulajdonságú adiabatikus kapcsolás invertere és négy szakaszból álló fázisjele

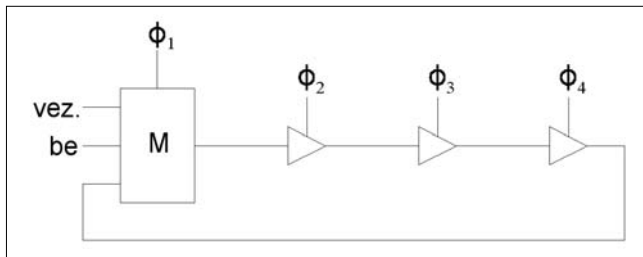


4. ábra

100 láncba kapcsolt inverter energiavesztesége 10fF és 50 fF terhelőkapacitások mellett

gyedik logikai fokozat kimenete fázishelyesen visszavezethető az első fokozat bemenetére. Invertereket így módon gyűrűbe fűzve az adattárolás a tárolt adat folyamatos másolásával megoldható.

Az ezen az elven működő tároló felépítése az 5. ábra szerinti, ahol a Φ_1 fázisjelhez kapcsolódó logikai kapu egy összetett kapu, míg a többi három kapu csak inverter (puffer).



5. ábra Adiabatus elven működő tárolóelem felépítése

Az „M”-el jelölt összetett logikai kapu egy kétbemenetű multiplexer, amely az új adat beírására szolgál, ami az ábrán látható elrendezésben a felfutó Φ_1 fázisjelre történhet meg.

5. Adiabatus működésű áramkörök ütemezése

Az önmagában lassú működésű, fázisjel-vezérelt adiabatus logika felépítéséből adódóan pipeline működésű, így átbocsátása a logikai mélységtől független. Ez előnyösen használható a digitális jelfeldolgozásban, ahol adiabatus logikával kétszintű pipeline feldolgozó egység építhető. Mindemelllett négyfázisú kapcsolással a lappangási idő is alacsonyan tartható, mivel egy órajel-ciklus alatt négy logikai fokozatot értékelünk ki.

Ismert, hogy a magas szintű logikai szintézis kulcslépése az ütemezés, amely megadja, hogy mely műveletek végrehajtása melyik vezérlési lépésben kezdődik. Az eddig használt ütemező algoritmusok azonban adiabatus áramkörökre nem alkalmazhatók, mivel nem

veszik figyelembe az adatútba az ütemezés után beépítendő multiplexerek műveletvégzési idejét.

Az adiabatus logika fázisjel-vezérelt működéséből következően minden egyes fázisjelre csak egyetlen logikai szint kiértékelése történik. Azáltal, hogy ütemezés-kor valamennyi művelet valamennyi logikai szintjét valamely vezérlési lépéshez rögzítjük, multiplexerek nem szűrhetők be az adatútba az ütemezés után, mivel ki nem használt vezérlési lépések jellemzően nem állnak rendelkezésre.

Az adiabatus ütemező eljárás alapja az azonos típusú műveletvégző egységek között a hozzájuk csatlakozó multiplexer bemenet darabszámának egyenletes elosztása [4].

Ehhez a következő adatokra van szükség:

- viselkedési leírás adott tervezési megkötésekkel történő megvalósításához szükséges műveletvégző egységek darabszáma minden egyes műveletvégző-egység típusból,
- az adatforrások forrástípusonkénti darabszáma, ahonnan az egyes típusú műveletvégző egységekhez bemenő adatok érkeznek.

Az ütemező eljárás először kiszámítja az előírt újraindítási idő eléréséhez szükséges műveletvégző egységek lehető legkisebb darabszámát, anélkül, hogy a tulajdonképpeni ütemezést végrehajtaná. A következő lépésben, felhasználva a műveletvégző egységek darabszámát és a viselkedési leírásban megadott műveletek közötti függőségeket, az azonos típusú műveletvégző egységekhez tartozó multiplexer-bemenetek darabszámának kiszámítása következik. A harmadik lépésben az azonos típusú műveletvégző egységek, és a rájuk jutó multiplexer-bemenetek darabszámából az algoritmus műveletvégző-egység típusonként meghatározza az egy műveletvégző egységre jutó multiplexer-bemenetek számát, és így az egyes multiplexer-műveletek végrehajtásához szükséges vezérlési lépések darabszámát.

A negyedik lépésben kerül sor a tényleges ütemezésre, ahol az algoritmus egy módosított lista ütemezéssel figyelembe veszi mind a multiplexer-műveletek vég-

rehajtásához szükséges vezérlési lépések darabszámát, mind a kétszintű pipeline működésből adódó művelet átlapolásokat.

A továbbiakban bevezetjük a következő jelöléseket:

- N_k – k típusú műveletek darabszáma a viselkedési leírásban
- M_k – k típusú műveletet végrehajtó műveletvégző egységek darabszáma
- DII – újraindítási idő vezérlési lépésben mérve
- L – felső korlát az ütemezés hosszúságára vezérlési lépésben mérve
- S_k – k típusú műveletvégző egységekhez csatlakozó multiplexerek bemeneteinek száma
- I_k – k típusú műveletvégző egység egy bemenetére jutó multiplexer-bemenetek darabszáma
- d_k – k típusú műveletvégző egység előtt elhelyezkedő multiplexerek műveletvégzési ideje vezérlési lépésben mérve
- t_k – k típusú műveletvégző egység műveletvégzési ideje vezérlési lépésekben mérve
- t'_k – k típusú műveletvégző egység ütemezéshez használt műveletvégzési ideje vezérlési lépésekben mérve

A fenti jelölésekkel a viselkedési megadás megvalósításához szükséges műveletvégző egységek lehető legkisebb darabszáma műveletvégző-egység típusonként a (2) összefüggés szerint számítható ki, ahol a „ $\lceil \cdot \rceil$ ” jel a következő egész számra való felfelé kerekítést jelenti.

$$M_k = \left\lceil \frac{4 \cdot N_k}{DII} \right\rceil \quad (2)$$

A következőkben ismertetésre kerülő eljárással a különböző típusú műveletvégző egységek darabszámának függvényében műveletvégző-egység típusonként meghatározhatjuk a hozzájuk csatlakozó multiplexerek bemeneteinek számát. A keresett S_k értékek a viselkedési leírásban megadott műveletek közötti függőségek vizsgálatával kaphatók meg, a következő eljárást minden egyes k -ra elvégezve.

Kezdetben S_k egyenlő nullával. A kiinduló leírás alapján egy listát készítünk, amelyben a k típusú műveletekre összegyűjtjük azon különböző jellegű forrásokat, ahonnan a k típusú műveletekhez bemenő adat érkezik, valamint a különböző jellegű források mellett feltüntetjük a hozzá tartozó bemenetek előfordulási darabszámát is, amelyet E_a -val jelölünk, ahol a különbözteti meg az egyes forrásokat.

Az elkészített listákban az egyes források jellege háromféle lehet, amelytől függően S_k a következőképpen változik: ha a forrás

- bemeneti kapu, akkor $S_k = S_k + E_{bemeneti\ kapu}$,
- i típusú művelet, akkor $S_k = S_k + \min(E_{i\ típusú\ művelet}, M_i)$,
- állandó érték, akkor $S_k = S_k + \min(E_{állandó\ érték}, 2 \cdot M_k)$ ha $S_k > M_k$, egyébként pedig S_k változatlan.

Ennek magyarázata a következő. A bemeneti kapukhoz szükséges multiplexer-bemenetek darabszáma a bemeneti kapuktól eredő bemenő csatlakozások számával egyenlő.

Ha egy k típusú művelethez bemenő adat egy i típusú művelet eredményeként érkezik, akkor az ehhez szükséges multiplexer-bemenetek száma attól függ, hogy az i típusú műveletektől a k típusú műveletekhez vezető csatlakozások száma és az i típusú műveletvégző egységek darabszáma hogyan viszonyul egymáshoz. Ha az előző a kisebb, a helyzet ugyanaz, mint a bemeneti kapuk esetén. Ha azonban az utóbbi a kisebb, akkor csak az i típusú műveletvégző egységek darabszámával növekszik a szükséges multiplexer-bemenetek száma, mivel ekkor az i típusú műveletvégző egységek megosztásra kerülnek az i típusú műveletek között.

Állandó érték kiválasztása érdekében egy műveletvégző egységhez elég csak egyetlen multiplexer-bemenetet figyelembe venni, mivel az már előre kiválasztható úgy, hogy a szükséges vezérlési lépésben rendelkezésre álljon. Ezt fejezi ki a $2 \cdot M_k$ tag, ahol a kettes szorzó a későbbi kettővel való osztás miatt szükséges. Ha nem jut minden műveletvégző egységre állandó érték, akkor a szükséges multiplexer-bemenetek darabszáma csak az állandó értékek darabszámával növekszik. Ha a bemeneti kapuktól és a többi típusú műveletvégző egységektől érkező bemenetek száma nem nagyobb mint a rendelkezésre álló műveletvégző egységek száma, akkor a multiplexer-bemeneteket egyáltalán nem kell figyelembe venni, mivel a nem állandó értékű bemenetek csak a műveletvégző egységek egyik bemenetéhez csatlakoznak.

A műveletvégző egységekhez csatlakozó multiplexerek bemeneteinek darabszámát műveletvégző-egység típusonként meghatározó eljárást a 6. ábra foglalja össze, amely eljárással az S_k értékek megkaphatóak minden k -ra.

Mivel minden műveletvégző egység két bemenettel rendelkezik, és a k típusú műveletvégző egységekből M_k darab van, ezért az egyes műveletvégző egységek egyes bemeneteire jutó multiplexer-bemenetek egyenletesen elosztott darabszámát a (3) összefüggés adja.

$$I_k = \left\lceil \frac{S_k}{2 \cdot M_k} \right\rceil \quad (3)$$

Egy k típusú műveletvégző egységekhez csatlakozó multiplexerek műveletvégzési ideje vezérlési lépésben mérve kiszámítható a (4) összefüggés ismeretében, ahol egy logikai szinten legfeljebb négy bemenetű multiplexereket engedélyezünk.

$$d_k = \left\lceil \frac{\log_2(I_k)}{2} \right\rceil \quad (4)$$

Ütemezéskor egy műveletvégző egységet és a hozzá csatlakozó multiplexert együtt úgy tekinthetjük, mint-ha a műveletvégző egység műveletvégzési ideje a hozzá csatlakozó multiplexer műveletvégzési idejével megnőtt volna.

Ezt fejezi ki az (5) összefüggés, ahol a t'_k jelenti a tényleges ütemezéshez használandó, vezérlési lépésben mért műveletvégzési időt.

$$t'_k = t_k + d_k \quad (5)$$

A tényleges ütemezés egy módosított lista ütemezés, amely képes a pipeline működésű műveletvégző egységek használatával kétszintű pipeline adatutatót ütemezni. Először meghatározzuk minden egyes művelet mozgékonyosságát az ASAP (olyan hamar, amint lehetséges) és az ALAP (olyan későn, amint lehetséges) indítási értékek kiszámolásával, műveletvégzési időnek a t'_k értékeket használva.

A módosított lista-ütemező a műveletvégző egységek minden típusára egy lefoglalási táblát tart fenn, mely táblának M_k darab sora van, oszlopainak száma pedig megegyezik a vezérlési lépésben megadott megengedhető legnagyobb ütemezési hosszúsággal.

Egy művelet kezdete csak akkor ütemezhető a t vezérlési lépésbe, ha az adott művelethez tartozó lefoglalási tábla t sorszámú oszlopában még van üres hely. Ha egy művelet kezdetét a t vezérlési lépésbe ütemezünk, akkor az ütemező egy jelet tesz az adott művelethez tartozó lefoglalási tábla mindazon oszlopaiba, amely-

nek sorszámára igaz a (6) összefüggés, ahol C_k jelöli a k típusú művelethez tartozó lefoglalási tábla oszlopának sorszámát, n pedig egész szám és $0 < n < L/DII$.

$$C_k = t + n \cdot DII \quad (6)$$

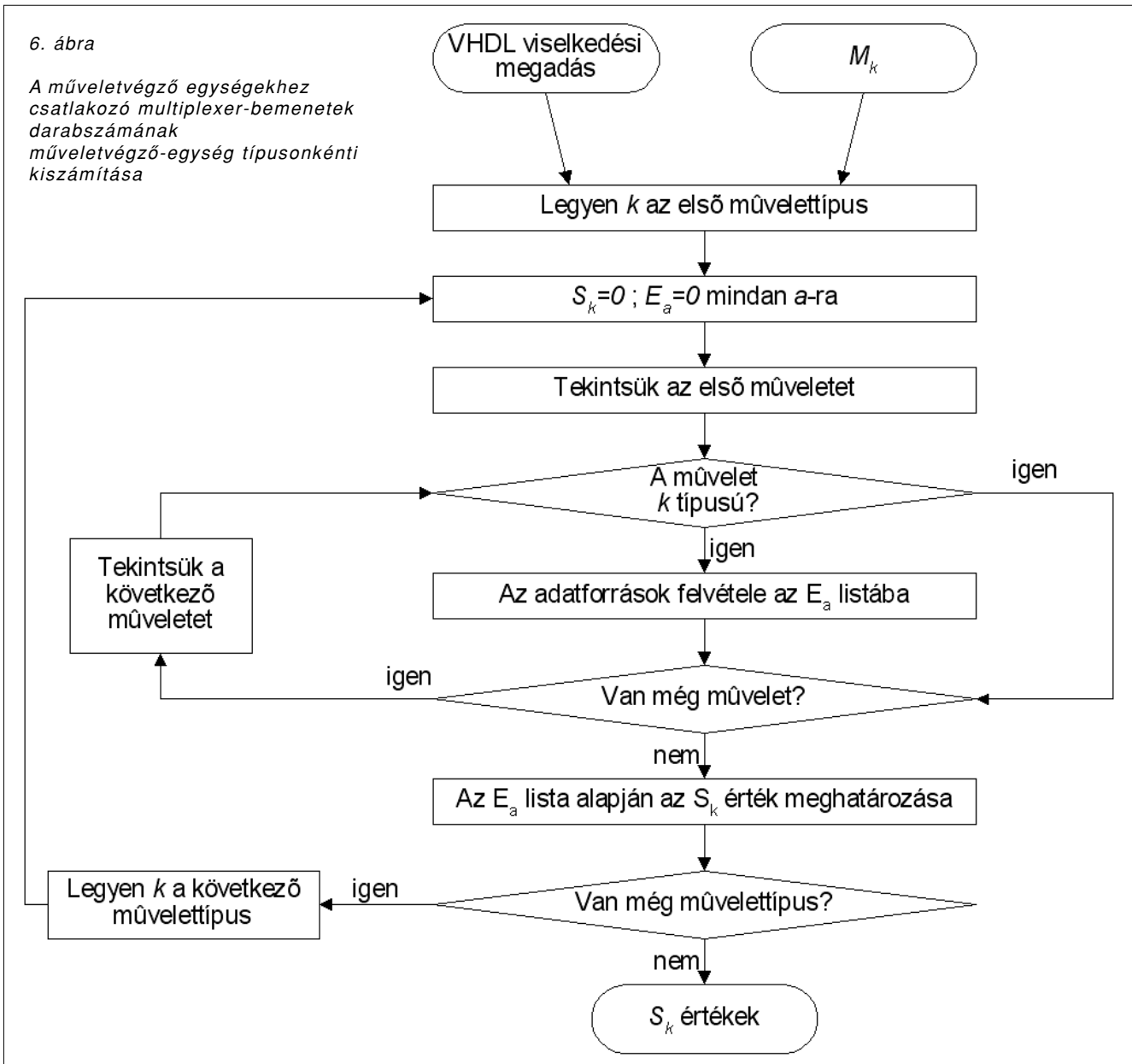
Ha egy k típusú művelet első alkalommal kerül ütemezésre, akkor az ütemező a k típusú művelethez tartozó lefoglalási tábla minden olyan oszlopának minden sorába egy-egy jelet tesz, amelyre a (7) összefüggés igaz.

$$C_k \neq t + 4 \cdot n \quad (7)$$

Az első eset a pipeline működés biztosításához szükséges, hiszen a lista ütemezés eredetileg erre nem alkalmas, míg az utóbbi a műveletvégző egységeknek a műveletek közötti megosztását teszi lehetővé, és ezzel biztosítja, hogy az ütemezés a (2) összefüggéssel meghatározott darabszámú műveletvégző egységekkel megvalósítható legyen.

6. ábra

A műveletvégző egységekhez csatlakozó multiplexer-bemenetek darabszámának műveletvégző-egység típusonkénti kiszámítása



Egy hagyományos ütemező algoritmusban az adott típusú műveletvégző egységek az ugyanolyan típusú műveletek között megoszthatók, ha ezen műveletek időben nem lapolják át egymást. Adiabikus műveletvégző egységek használatakor a második szintű pipeline lehetővé teszi az átlapolást, azonban az előbbi feltétel túl az erőforrás-megosztáshoz a négy fázisjeles vezérlés miatt még az is szükséges, hogy a megosztandó műveletvégző egységeken végrehajtandó műveletek azonos fázisjel alatt induljanak. A bemutatott ütemező algoritmus ezt úgy biztosítja, hogy az azonos típusú műveletvégző egységek ugyanazon logikai szintjeit ugyanazon fázisjelhez rendeli.

6. Egészértékű lineáris programozáson alapuló algoritmus négyfázisú adiabikus áramkörök ütemezéséhez

Az előzőekben közölt ütemező eljárás ugyan optimális eredményt ad a szükséges műveletvégző egységek darabszámára nézve, azonban az ütemezés vezérlési lépésekben mért hosszúsága általánosságban nem a legkedvezőbb, így a szükséges járulékos elemek, mint például a pufferek száma több lehet a kelleténél. Ezért bemutatunk egy egészértékű lineáris programozáson alapuló algoritmust is, amely tetszőleges szempontból optimális megoldást nyújt.

Az egészértékű lineáris programozáson alapuló ütemezés lineáris egyenlőségek és egyenlőtlenségek felállítását jelenti, amelyek leírják mind a viselkedési megadás műveleteinek egymástól való függését, mind a tervezési korlátokat. A tényleges ütemezés ezen egyenlőségek és egyenlőtlenségek megoldása azon feltétel mellett, hogy valamely célfüggvény értéke a legnagyobb vagy a legkisebb legyen.

A következőkben néhány újabb jelölést adunk meg, amelyeket – az előző alponban bevezetett jelöléseken felül – a továbbiakban használunk, valamint néhány korábban bevezetett jelölés most új értelmet kap:

- o_i – az i -dik művelet a kiinduló leírásban
- s_i – az o_i művelet legkorábbi (olyan hamar, amint lehetséges) indítási ideje
- l_i – az o_i művelet legkésőbbi (olyan későn, amint lehetséges) indítási ideje
- $x_{i,t}$ – értéke 1, ha az o_i művelet a t . vezérlési lépésben indul, egyébként nulla
- t_i – az o_i művelet végrehajtásához szükséges vezérlési lépések száma
- d_i – az o_i műveletet végrehajtó műveletvégző egység előtt elhelyezkedő multiplexer műveletvégzési ideje vezérlési lépésben mérve

A. Ütemezési korlátok

Valamennyi művelet végrehajtását a rá vonatkozó legkorábbi és a legkésőbbi indítási időpont között kell megkezdeni. Mivel a multiplexerek műveletvégzési idejét nem tudjuk előre megmondani, ezért az egyes mű-

veletek indítási ablakát csak a hagyományos esetre tudjuk felírni, azonban ezt is felhasználhatjuk arra, hogy elhagyhassunk nyilvánvalóan nulla értékű változókat. Így a (8) szerinti egyenletek írhatók:

$$\sum_{t=s_i}^{l_i} x_{i,t} = 1 \quad \text{minden egyes } i \text{ - re.} \quad (8)$$

B. Függőségi korlátok

Egy műveletvégző egység a bemenő adatait multiplexereken keresztül kapja, amelyek műveletvégzési idejét figyelembe kell venni. Azért, hogy a műveletek közötti függőségeket a kiinduló leírásnak megfelelően megtartsuk, egy művelet végrehajtása csak akkor kezdődhet, ha az azt megelőző műveletek már befejeződtek. A fentieknek megfelelő egyenletek a (9) szerinti:

$$\sum_{t=s_i}^{l_i} t * x_{i,t} + t_i + d_i - \sum_{t=s_j}^{l_j} t * x_{i,j} \leq 0$$

minden egyes i -re és j -re, ahol o_i közvetlenül megelőzi o_j -t,

$$\sum_{t=s_i}^{l_i} t * x_{i,t} + t_i + d_i - L \leq 1$$

minden egyes i -re, ahol o_i -t nem követi további művelet.

C. Pipeline átlapolási korlátok

Egy műveletvégző egység, amely egy számítást vezérlési lépésben kifejezve a t időpontban kezd meg, pipeline működésének köszönhetően a $t+4$ időpontban újabb számításba kezdhet anélkül, hogy az előző műveletet befejezte volna. Egy művelet, amely vezérlési lépésben kifejezve a T időpontban indul, lefoglal egy műveletvégző egységet $T \leq t < T+4$ ideig. Ez a (10) szerinti függvényekkel fejezhető ki, amelyet minden egyes műveletre (minden i -re) fel kell írni:

$$x'_{i,t} = 1 \quad \text{ha } T + d_i \leq t < T + d_i + 4 \\ = 0 \quad \text{egyébként.} \quad (10)$$

A párhuzamosan használt k típusú műveletvégző egységek darabszáma a vezérlési lépésben kifejezett t időpontban egyenlő azon műveletek darabszámával, amelyek ugyanezen t időpontban k típusú műveletvégző egységet lefoglalnak. A k típusú műveletvégző egységek szükséges darabszáma egyenlő a bármely időpontban párhuzamosan használt k típusú műveletvégző egységek darabszámának legnagyobbikával. Ezt fejezik ki a (11) függvények:

$$M_k = \max_{t=1}^{L/DII} \sum_{i=1}^{L/DII} x'_{i,j+t*DII} \quad (11)$$

minden i,j és k -ra, ahol $0 < j \leq DII$ és o_i k típusú

D. Multiplexer-korlátok

A multiplexer-korlátok felírásához meg kell határozni az egyes típusú műveletvégző egységek darabszámának függvényében a különböző típusú műveletvégző egységekhez csatlakozó multiplexerek bemeneteinek számát, amely a 6. ábrán ismertetett algoritmus szerinti

S_k értékek kiszámítását jelenti. Egy műveletvégző egység két bemenettel rendelkezik, így a szükséges multiplexer bemenetek száma megoszlik a két bemenet között, ezen kívül a k típusú műveletvégző egységből M_k darab van. Így a (12) egyenlőtlenségeknek kell teljesülniük:

$$\frac{S_k}{2} \leq \sum_{o_i} d_i \leq \sum_{k=1}^{M_k} 4^{d_k} \quad \text{ahol } o_i \text{ } k \text{ típusú} \quad (12)$$

7. Ütemezési algoritmusok eredményei

Az 1. táblázat a bemutatott lista-ütemező algoritmussal kapott ütemezési eredményeket mutatja a véges válaszú szűrő tervezési példára. Az adatút szélességét 16 bitre választva az adiabatikus összeadó műveletvégzősi ideje 6, a szorzóé pedig 9 vezérlési lépés. A táblázat oszlopai rendre az újraindítási idő vezérlési lépésben mérve, az összeadók darabszáma, a szorzók darabszáma, a multiplexer bemenetek száma, a pufferek száma és az ütemezési hosszúság vezérlési lépésben kifejezve. Az egészértékű lineáris programozáson alapuló ütemezési eredmények a 2. táblázatban láthatók

Mindkét ütemező algoritmus azonos és egyben a legkisebb szükséges számú műveletvégző egységet alkalmazza, azonban a lista ütemezés használatával ütemezési hosszúság nagyobb és a szükséges pufferek száma is több. Ennek az a magyarázata, hogy ez az ütemezés az azonos típusú műveletek kezdetét azonos fázisjelhez köti.

8. Értékelés, következtetések

A cikkben bemutatunk egy továbbfejlesztett adiabatikus töltésvisszanyerő kapcsolást, amely egy korábbi kapcsoláson alapul, azonban a terhelő kapacitás feltöltése alatt teljesen kiküszöböli a nem-adiabatikus veszteséget. A kapcsolat bonyolultságának kézben tartása érdekében a terhelő kapacitásnak csak a részleges kisütése megengedett, azonban lehetőség nyílik teljesen adiabatikus kisütésre is. A logikai kapcsolások mellett

szekvenciális hálózatok építhetők a bemutatott tároló elrendezés alkalmazásával.

Adiabatikus rendszerek automatizált magas szintű tervezésekor olyan problémát kell megoldani, amelyre a jelenleg használatos ütemező algoritmusok nem alkalmazhatók. A bemutatott ütemező algoritmus úgy teszi lehetővé nagy bonyolultságú adiabatikus rendszerek tervezését, hogy a tervezőnek az adiabatikus működéstől elvonatkoztatva, csak feladatköri szempontokra kell ügyelnie.

Köszönetnyilvánítás

A szerzők köszönetüket fejezik ki Dr. Kovács Ferenc egyetemi tanárnak, akinek szakmai segítsége meghatározó volt az egész kutatás kivitelezése során. Szintén köszönet illeti az Országos Tudományos és Kutatási Alapot a T023963 számú „Kisfogyasztású CMOS áramkörök tervezése” és a T029331 számú „Kisfogyasztású CMOS analóg áramkörök tervezési metodikájának kidolgozása” című OTKA kutatási támogatásokért.

Irodalom

- [1] J. Fischer, P. Teichmann, A.G. Stoffi, E. Amirante, D.S. Landseidel, „Scaling Trends in Adiabatic Circuits”, 1st Int. Workshop on Reversible Computing, May 2005.
- [2] E. Amirante, A.B. Stoffi, J. Fischer, G. Iannaccone, D.S. Landsiedel, „Variations of the Power Dissipation in Adiabatic Logic Gates”, 11th Int. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), September 2001., pp.9.1.1–10.
- [3] L. Varga, F. Kovács, G. Hosszú, „An Efficient Adiabatic Charge-Recovery Logic”, IEEE SoutheastCon 2001, Clemson, South Carolina, pp.17–20.
- [4] L. Varga, F. Kovács, G. Hosszú, „Approaches for Scheduling of Adiabatic Logic”, IEEE Int. Workshop on Logic & Synthesis, Granlibakken, CA, June 2001, pp.18–22.

1. táblázat A bemutatott lista szerinti ütemezést alkalmazó eljárás eredményei a véges válaszú szűrő tervezési példára, különböző újraindítási idők mellett.

<i>DII</i>	<i>M₊</i>	<i>M_*</i>	<i>Mplex. bem.</i>	<i>Puffer</i>	<i>L</i>
4	15	8	0	20	38
8	8	4	36	70	51
12	5	3	42	142	63

2. táblázat Az ILP-t alkalmazó ütemező eljárás eredményei különböző újraindítási idők mellett.

<i>DII</i>	<i>M₊</i>	<i>M_*</i>	<i>Mplex. bem.</i>	<i>Puffer</i>	<i>L</i>
4	15	8	0	20	38
8	8	4	36	34	43
12	5	3	42	81	57

Fuzzy következtető rendszerek alkalmazása mobil hálózatok felügyeletében

KÓCZY T. LÁSZLÓ, BOTZHEIM JÁNOS, SALLAI RICHÁRD, CSÁNYI KORNÉL
Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék

KUTI TAMÁS
Linecom Kft.

Lektorált

Kulcsszavak: mobil hálózatok, felügyeleti rendszerek, intelligens módszerek, fuzzy logika

Mobil szolgáltatást megvalósító mikrohullámú hálózatok esetében a vételi szintet befolyásolják az adó- és a vevőállomás közötti úgynevezett Fresnel-zónába bekerült akadályok. Ezek az akadályok lehetnek ideiglenesen vagy véglegesen az adott zónába került mesterséges vagy természeti tárgyak, vagy az éppen aktuális időjárás által meghatározott mennyiségű csapadék, illetve köd. A hálózatfelügyeleti rendszerbe célszerű olyan intelligens döntési alrendszer beépítése, amely a csillapítás mértékéből, illetve annak időfüggő viselkedéséből közvetlenül képes eldönteni, hogy a mi okozhatta a csillapítást. A cikk egy sikeresen befejezett NKFP K+F projekt keretében megvalósult hálózatfelügyeleti rendszer intelligens modulját ismerteti. Bemutatja az intelligens döntéstámogató rendszert és az ezt alkotó fuzzy logika alapjait és az időjárás helyzet automatikus identifikálására szolgáló alkalmazást.

1. Bevezetés

Mobil szolgáltatást megvalósító mikrohullámú hálózatok esetében közismert az a tény, hogy a vételi szintet erősen befolyásolják (csökkentik) az adó- és a vevőállomás között az úgynevezett Fresnel-zónába bekerült akadályok. Ezek lehetnek ideiglenesen vagy véglegesen az adott zónába került mesterséges vagy természeti tárgyak, vagy az éppen aktuális időjárás által meghatározott mennyiségű csapadék, illetve köd. A hálózatfelügyeleti rendszerbe célszerű olyan intelligens döntési alrendszer beépítése, amely a csillapítás mértékéből, illetve annak időfüggő viselkedéséből közvetlenül képes olyan következtetés meghozatalára, amely a csillapítás okát felismerve már a szükséges beavatkozásra ad javaslatot a hálózatfelügyeleti rendszer kezelői felületén. Jelen tanulmány egy sikeresen befejezett NKFP K+F projekt keretében megvalósult hálózatfelügyeleti rendszer modulját ismerteti. A modul keretében a felügyeleti rendszer által mért adási és vételi szint értékekből kiszámítható csillapítási értékek időbeli változása alapján a rendszer GUI felületén megjeleníti az intelligens döntési alrendszer által létrehozott következtetést. Írásunkban csupán egyetlen kérdésre fókuszálunk, a mérsékelt égövi kontinentális éghajlati zónában (nemzetközi szabványban rögzített módon) megjelenő különböző esőkategóriák intelligens felismerésére.

Csupán utalunk arra, hogy a következőkben konkrét példákkal bemutatott különböző riasztási szintek alapvetően kétféle jelenségcsoportra utalnak. Az első csoportba tartoznak a mikrohullámú állomások berendezéseinek hirtelen bekövetkező vagy a hosszabb idejű üzem során fokozatosan jelentkező hardver meghibásodásai vagy funkcióromlásai. Ezekben az esetekben természetesen karbantartói beavatkozást javasol a rendszer. A második csoportba tartoznak a Fresnel-zónába bekerült akadályok miatt bekövetkezett vételi szint csök-

kenések. Jellemző példa erre a városi környezetben egy háztetőre szerelt poszter felállítás, ami két állomás között az átvitel minőségét jelentős mértékben ronthatja. Ebben az esetben egészen más jellegű beavatkozásra van szükség, adott esetben akár az adó és/vagy a vevő helyének a megváltoztatására is szükség lehet. Másik példa erre a jelenségre rurál környezetben az adó és a vevő között elhelyezkedő erdősáv tavaszi kilombosodása. Ezt a jelenséget például az adóteljesítmény éves ciklus szerinti automatikus utánállításával lehet kompenzálni. E két példa egyébként a mesterséges és természetes terepakadályra is egy-egy illusztráció.

Az általunk vizsgált esetek azonban az időjárás jelenségek körébe tartoznak. Az esőzésből adódó fading elkülönítése a különböző terepakadályok okozta csillapítástól az egy körzetben egyidejűleg irányfüggetlenül jelentkező vételi szint csökkenés révén történhet meg. Megjegyezzük, hogy e két jelenségcsoport kombinált hatást is eredményezhet, az adó és a vevőállomás között elhelyezkedő lombos erdősáv esőzés idején, sőt azt követően még hosszabb ideig, igen erős anizotrop fadinget okozhat a leveleken található vízcseppek miatt. E levelek ugyanis megannyi apró hullám-visszaverő tükörként működnek.

A következőkben az intelligens döntés meghozatalára alkalmazott számítási intelligencia módszer háttéréről ejtünk néhány szót, majd a fuzzy rendszerek alapjait tárgyaljuk. Ezután röviden ismertetjük a hierarchikus fuzzy rendszereket, végül pedig a mobil hálózat felügyeletében alkalmazott fuzzy rendszert mutatjuk be.

2. A fuzzy rendszerek alapjai

Az emberi gondolkodásmód és bizonyos jelenségek nem írhatóak le pontosan a kétértékű logikával. Régebben felmerült már a kétértékű logika kiterjesztésé-

nek az igénye, hogy ne csak igaz és hamis logikai értékeket használjunk, hanem lehetőség legyen átmenetek definiálására. Sok olyan állítás van, amelyekről nem lehet élesen eldönteni, hogy igaz-e vagy hamis, hanem csak valamilyen mértéket lehet mondani az igazságtartalmáról. Ez a gondolat vezette el a '60-as években L. A. Zadeh-t a fuzzy logika megalkotásához [2].

Ma még a számítógépekben és az élet számos területén a kétértékű logika terjedt el. Nyilvánvalóvá vált azonban, hogy ha intelligensebb eszközöket szeretnénk létrehozni, akkor jobb eredményt érhetünk el úgy, ha az emberi logikához jobban közelítő módon írjuk le a rendszerek viselkedését.

A fuzzy logika a hagyományos logika kiterjesztése. A fuzzy logikai változó a 0 és az 1 érték között tetszőleges értéket felvehet, a 0 jelöli a „teljesen hamis” állítást, az 1 pedig a „teljesen igazat”. Ilyen értelemben a 0.5 körüli érték jelképezi a „félíg igazat”, például a 0.9 a „majdnem igazat”. A hagyományos logikai műveletek is kiterjeszthetők fuzzy logikára. A fuzzy logika segítségével definiálhatunk fuzzy halmazokat, fuzzy szabályokat és következtető rendszereket is létrehozhatunk. Az alábbiakban összefoglalt alapfogalmakkal kapcsolatban részletesebb leírást ad [1].

2.1. Fuzzy halmazok

Az A fuzzy halmazt az úgynevezett tagsági függvénnyel adhatjuk meg. A tagsági függvény minden egyes x alaphalmazbeli értékhez egy a $[0, 1]$ intervallumból vett értéket rendel aszerint, hogy az adott x érték mekkora mértékben eleme (tagja) az A fuzzy halmaznak:

$$\mu_A : X \rightarrow [0, 1],$$

ahol μ_A az A fuzzy halmaz tagsági függvénye, mely egyértelműen megadja a halmazt. Különböző alakú tagsági függvényeket szokás definiálni. Leginkább a háromszög, a trapéz és a Gauss-görbe alakú tagsági függvények terjedtek el.

Az 1. ábrán csillapítási értékeket láthatunk fuzzy halmazokkal leírva. Három kategóriát különböztethetünk meg, az „enyhe”, „közepes” és „nagy” csillapítást jellemző fuzzy halmazokat.

A tagsági függvények a példában trapéz alakúak.

A hagyományos halmazelméletben értelmezett három alapműveletet végtelen sokféleképpen lehet általánosítani a fuzzy halmazok elméletére. A legelterjedtebbek a klasszikusnak számító Zadeh-féle definíciók [2], de számos előnye van az úgynevezett algebrai műveleteknek is.

Egy X alaphalmazon értelmezett A fuzzy halmaz Zadeh-féle komplementese \bar{A} , ahol minden $x \in X$ értékre:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

A és B fuzzy halmazok Zadeh-metszete:

$$\mu_{A \cap B}^Z(x) = \min(\mu_A(x), \mu_B(x)),$$

Zadeh-uniója pedig:

$$\mu_{A \cup B}^Z(x) = \max(\mu_A(x), \mu_B(x)).$$

Az algebrai metszetet a

$$\mu_{A \cap B}^I(x) = \mu_A(x) \cdot \mu_B(x),$$

míg az unió a

$$\mu_{A \cup B}^I(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

összefüggés szerint számolható.

2.2. Fuzzy szabályok

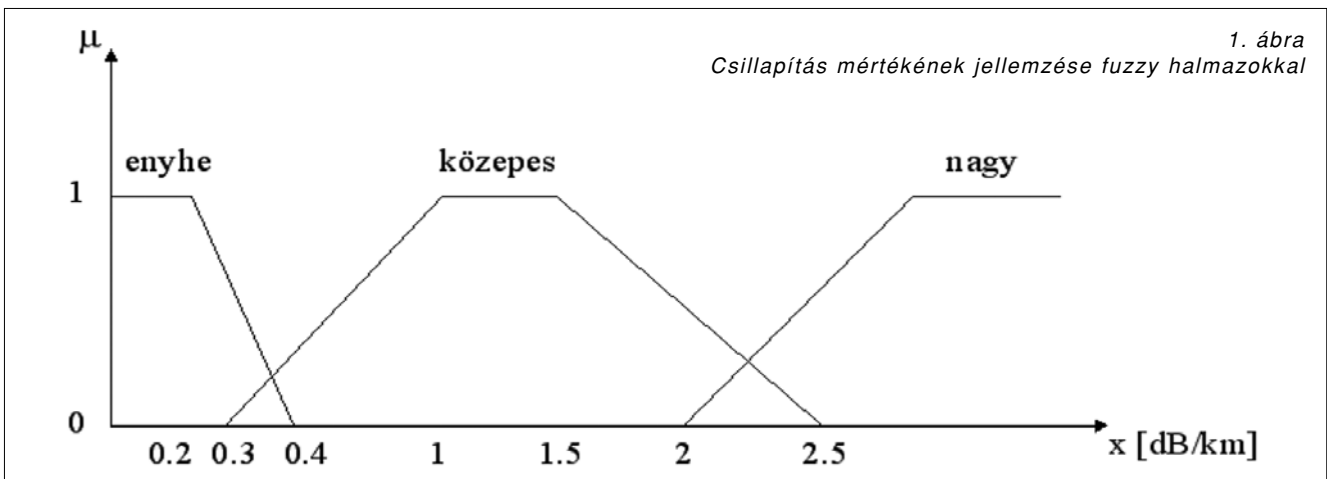
A fuzzy halmazok segítségével természetes emberi nyelven könnyen tudunk szabályokat megfogalmazni. Egy távfelügyeleti rendszerben például mondhatunk egy olyan szabályt, hogy „Ha a vételi szint csökkenés enyhe és az eltelt idő rövid, akkor gyenge esőzés van”. Ha ezután definiáljuk az „enyhe”, „rövid” és „gyenge” tagsági függvényeket, akkor fuzzy szabályhoz jutunk. Egy egybemenetű, egykimenetű egyszerű fuzzy szabály alakja:

$$R : \text{Ha } x = A \text{ akkor } y = B,$$

ahol $x \in X$ a bemeneti változó, $y \in Y$ a kimeneti változó, X a bemeneti változó alaphalmaza, Y a kimeneti változó alaphalmaza. A és B nyelvi változók. Az A az R szabály antecedense (premisszája), a B pedig az R szabály konzekvensze (konklúziója). Több bemenetű, egykimenetű fuzzy szabály általános, úgynevezett Mamdani-féle (ortogonálisan dekomponált) alakja [3]:

$$R : \text{Ha } x_1=A_1 \text{ és } \dots \text{ és } x_n=A_n \text{ akkor } y = B,$$

ahol $x=(x_1, \dots, x_n)$ a bemeneti értékek vektora, $x_j \in X_j$, $X=X_1 \times \dots \times X_n$ az n -dimenziós alaphalmaz, $A=(A_1, \dots, A_n)$ az antecedens halmazok vektora, $A \subset X$, $y \in Y$ a kimeneti változó, Y a kimeneti változó alaphalmaza, B a konzekvens halmaz, $B \subset Y$.

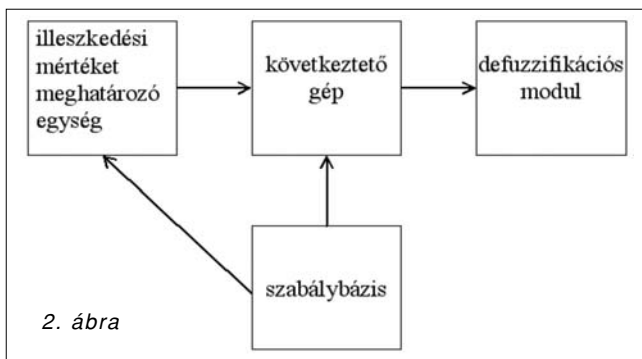


1. ábra
Csillapítás mértékének jellemzése fuzzy halmazokkal

A \tilde{C} jel fuzzy részhalmazt jelent. A szabály alkalmazásának feltétele, hogy az összes bemeneti változó pozitív mértékben essék a megfelelő antecedens halmazba. Több kimenetű szabályok esetén a kimenetek függetlenek egymástól, ezért az ilyen szabályok dekomponálhatók a fentivel megegyező alakú egykimenetűre, csökkentve ezzel a számítási igényt. Szokásos néha a fuzzy szabályt implikációként is interpretálni, például egybemenetű esetben : $R : A \rightarrow B$ formában, a következőkben azonban a Mamdani-féle függvényyszerű interpretációt fogjuk alkalmazni.

2.3. Fuzzy következtető rendszer

A fuzzy halmazok elméletét felhasználhatjuk bonyolult, analitikus módon nem modellezhető rendszerek kezelhető leírására. Fuzzy szabályok segítségével az emberi gondolkodáshoz hasonlító funkciót ellátó következtető rendszereket hozhatunk létre. A fuzzy rendszer vázlata a 2. ábrán látható.



2. ábra

Az illeszkedési mértéket meghatározó egység a megfigyelést hasonlítja össze a szabályok feltételrészeivel. Ennek alapján a következtető gép valamilyen következtetési algoritmussal meghatározza a kimeneti fuzzy halmazt. Többféle következtetési módszer ismert, gyakorlati alkalmazásokban legelterjedtebb a Mamdani-módszer [3]. A kimenetet defuzzifikációs egységgel alakítjuk át éles, úgynevezett „crisp” értékre. A 3. ábrán a Mamdani-féle következtetési algoritmust szemléltetjük.

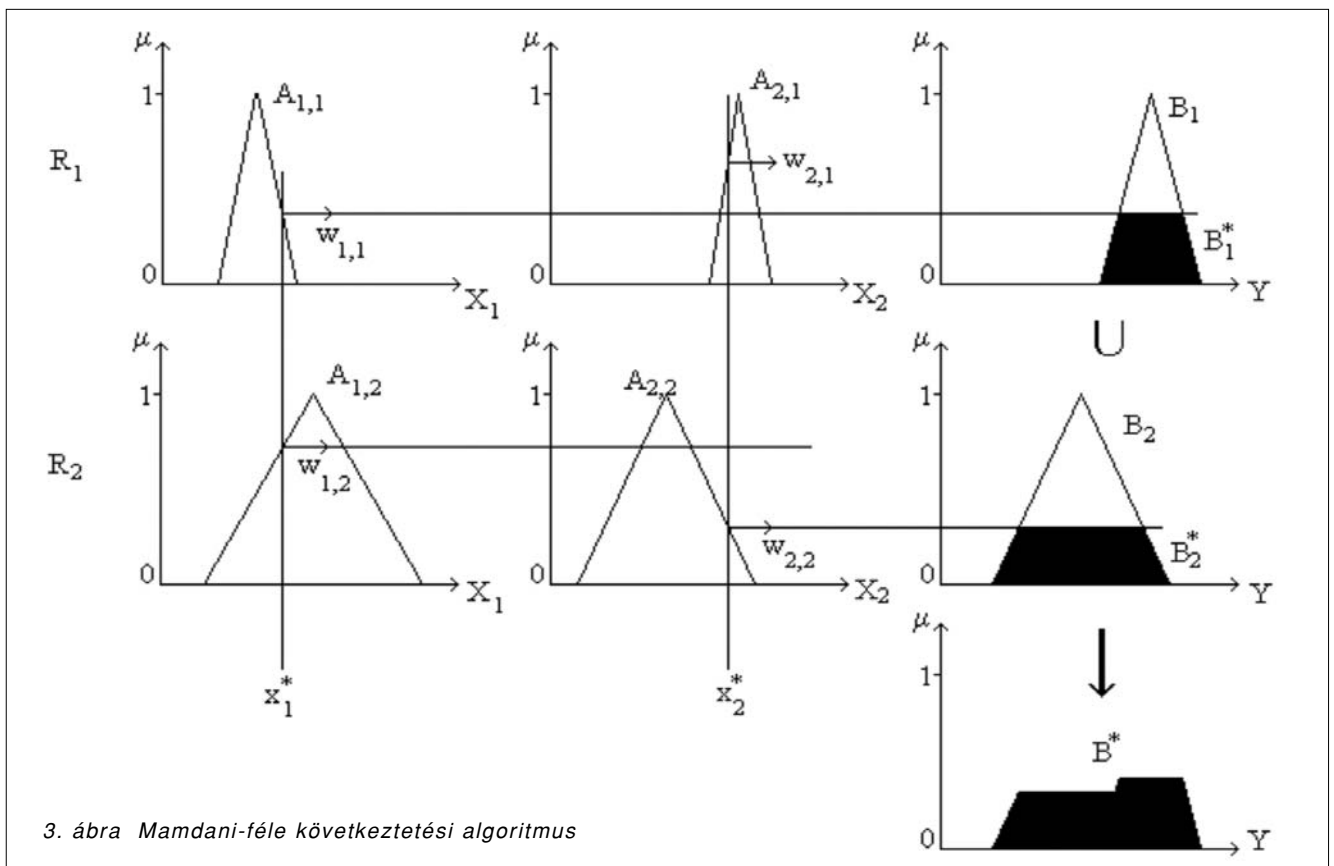
A következtetés elején meghatározzuk, hogy az adott bemenet mennyire illeszkedik a szabályokra. A megfigyelésvektor minden egyes elemét összehasonlítjuk mindegyik szabály feltételrészének ugyanezen komponensével. Legyen A^* az n -dimenziós megfigyelésvektor. Az illeszkedési (tüzelési) mérték a j . dimenzióban az i . szabályban:

$$w_{j,i} = \max_{x_j} \{ \min \{ A_{j,i}^*(x_j), A_{j,i}(x_j) \} \}$$

ahol $A_{j,i}$ az i . szabály j . dimenziójának tagsági függvénye. Ha a megfigyelés éles (crisp) vektor, akkor a fenti összefüggés a következőre egyszerűsödik: x^* állapot (helyzet-) vektor esetén, a j . dimenzióban az illeszkedés mértéke:

$$w_{j,i} = A_{j,i}(x_j^*).$$

Miután minden dimenzióban meghatároztuk az illeszkedés mértékét, kiszámítjuk az eredőt az egész antecedensre vonatkozóan is. Egy szabály alkalmazhatóságának (érvényességének) mértékét ugyanis antecedense minden egyes dimenziójának illeszkedési mértéke befolyásolja. Egy R_i szabály tüzelési mértéke a szabály antecedensei illeszkedési mértékeinek a minimuma lesz:



3. ábra Mamdani-féle következtetési algoritmus

$$w_i = \min_{j=1}^n w_{ji}$$

w_i értéke azt adja meg, hogy az R_i szabály mekkora mértékben befolyásolja a következmény előállítását az A^* megfigyelés esetén.

Miután egy megfigyelésre minden egyes R_i szabályhoz meghatároztuk annak w_i tüzelési súlyát, előállítjuk a szabályhoz tartozó következtetést. Ez a szabály konzekvensének w_i -vel való „csonkolásával” (azaz halmazmetszetével) történik.

A szabályhoz tartozó következtetés:

$$B_i^* = \min(w_i, B_i(y))$$

A teljes szabálybázishoz tartozó következtetést az így kapott B_i^* következtetések uniójaként állítjuk elő:

$$B^*(y) = \max_{i=1}^r B_i^*(y)$$

A következtetés eredőjeként a $B^*(y)$ fuzzy következményhalmazt kapjuk. Legtöbbször azonban egy rendszer kimeneteként nem fuzzy halmazt várunk, hanem éles értéket, mely konkrétan megadja a beavatkozás mértékét vagy módját. Tehát meghatározzuk azt az éles kimenetet, amely legjobban jellemzi a következtető gép által eredményezett fuzzy halmazt. Ez az eljárás a defuzzifikáció, melyre számos módszer ismert az irodalomban. A jelen alkalmazásban a súlypont alapú (COG) defuzzifikációs módszert választjuk, melynek alkalmazása a gyakorlatban a legelterjedtebb:

$$y_{COG} = \frac{\sum_{i=1}^r \int_{y \in B_i^*} B_i^*(y) y dy}{\sum_{i=1}^r \int_{y \in B_i^*} B_i^*(y) dy}$$

3. Hierarchikus fuzzy rendszerek

Az 1990-es évek elejétől különböző területeken vezettek sikerre a korábban alkalmazott „hagyományos” fuzzy rendszerekhez képest több hierarchikus szintbe strukturált újfajta fuzzy modellek.

Kiemelkedő ezek között Sugeno pilóta nélküli helikopter kísérlete, mely azóta tényleges alkalmazásba is került [4]. E rendszerek közös jellemzője, hogy a szükséges állapotváltozók száma viszonylag magas (mindenesetre 10 feletti), valamint az, hogy a vizsgált rendszer viselkedése értelemszerűen strukturálható, valamely alapváltozók szerint lokális modellek összességére bomlik, és e lokális modellek külön-külön csökkentett számú változót alkalmaznak. A helikopternél ezt a természetes struktúrát a különböző repülési manőverek összessége adja, melyeknél más és más állapotváltozó-részhalmaz a jellemző. A pilóta ennek megfelelően mindig csak a műszerek egy részhalmazát figyelve vezeti a gépet.

A bonyolult rendszerek nagy többségénél azonban egy ilyen természetes struktúra nem adott, vagy tiszta formában nem is létezik. Gyakran megjelölhetők azonban olyan részmodellek, amelyek lokálisan csökkentett változószámmal jellemezhetők, nem alkotnak azonban teljes partíciót. Természetesen ez a részmodell az állapotér különböző régióiban más és más lehet. Ha a lo-

kális változórészhalmazok ismertek, a teljes állapotteret partitionáljuk, és minden egyes résztérhez lokális modellt készítünk. Szerencsés esetben a lokális rendszerek lényegesen kevesebb változót használnak, és így az összesített szabályszám is jelentősen csökkenhet. A helikopteres példánál maradva, más változók dominálnak az „emelkedés”, és megint mások például az „előre repülés” vagy a „helyben lebegés” művelete közben [4].

Minden lokális modellhez egy *alszabálybázis* tartozik. A felső, úgynevezett *metaszinten* először – a megfigyelés környezete vagy a rendszer előírt reakciója alapján – a megfelelő alszabálybázis kiválasztására kerül sor. Ezt a lépést úgynevezett *metaszabályok* határozzák meg, amelyek bizonyos, a lokális modelleket lényegében elkülönítő változók értéke alapján, vagy speciálisan a rendszer lokális működését szabályozó változók értéke alapján választják ki a megfelelő lokális modellt.

Lehetőség van több metaszint alkalmazására is, ilyenkor az egyes metaszintek a modell egyre pontosabb finomítását végzik és a kiválasztott állapotváltozókhoz tartozó lokális szabálybázis szerinti következtetés meghatározása a legalsó szinten történik.

Egy nagyon egyszerű hierarchikus szabálybázis például a következő lehet:

Metaszint (R_0):

Ha $x_1 = A_{11}$ és $x_2 = A_{21}$ akkor vegyük a D_1 tartományt

Ha $x_1 = A_{12}$ és $x_2 = A_{22}$ akkor vegyük a D_2 tartományt,

R_1 szabálybázis (a D_1 tartományhoz):

Ha $x_3 = A_{31}$ és $x_4 = A_{41}$ akkor $Y=B_1$

Ha $x_3 = A_{32}$ és $x_4 = A_{42}$ akkor $Y=B_2$,

R_2 szabálybázis (a D_2 tartományhoz):

Ha $x_5 = A_{51}$ akkor $Y=B_3$

Ha $x_5 = A_{52}$ akkor $Y=B_4$.

Ahogy ezen a példán is látszik, az eredetileg 5 változós szabályrendszer a változók lokális csoportosításával kisebb dimenziós alszabálybázisokra bontható, így az eredő „mélység” csak 4 változónyi ($\max\{2+2, 2+1\}$), ily módon az eredő komplexitás csökkent.

4. A fuzzy logika alkalmazása

Mint említettük, ebben a tanulmányban a fuzzy logikát egy projekt keretében megvalósult hálózatfelügyeleti rendszer egy moduljaként alkalmazzuk [5].

Olyan intelligens döntéstámogatási rendszert hoztunk létre, amelyet egy mobil hálózat felügyeletében alkalmazunk. A rendszer csapadék-jellemzőket határoz meg adott területen, a felügyelt távközlési hálózatban rendelkezésre álló adási/vételi jelszintek alapján. Ezen információk segítségével az operátor pontosabb képet kap a mikrohullámú távközlési hálózat tényleges állapotáról, és célszerű döntést hozhat a legjobb beavatkozásról.

A rendszer a döntéseket két bemenő paraméter alapján hozza meg, vételi jelszint csökkenése és eltelt idő alapján.

Az alkalmazás képes adatok fogadására a bázisállomásokról, amelyeket egy adatbázisban tárol a hozzátartozó időbélyeggel együtt. Meghatározott időnként végez egy körlekérdezést, és az új adatok régiekkel való összevetéséből meghatározza a jelszint változást és a hozzá tartozó időt, majd ezen információk alapján következtetést hoz létre.

A következtetés alapvetően kétféle csoportba sorolható. Az első kategóriába az időjárással kapcsolatos információk tartoznak, míg a másik esetben valamilyen egyéb riasztási következtetés történik. Az időjárási viszonyok ismerete több, a távközlési felügyeleti rendszer működésében bekövetkező változásra is magyarázatot adhat.

Ilyen esetekben az operátor erőforrásokat szabadíthat fel. Például, ha az operátor jelszintcsökkenést észlel és az intelligens döntéstámogató alrendszer képes meghatározni, hogy a riasztást időjárásváltozás vagy egyéb, beavatkozást igénylő probléma okozta. Az első esetben szükségtelen a műszaki személyzetet a helyszínre küldeni, így bevezethetőek maradnak más technikai problémák megoldásában.

A jelszintváltozás és a csapadékmennyiség közötti összefüggés matematikailag a következőképpen számolható:

$$\gamma = kR^\alpha$$

ahol γ a jelszint csökkenése, R a csapadékmennyiség, k és α pedig különböző, elsősorban a használt frekvenciától függő paraméterek. Ennek az összefüggésnek az inverzét szeretnénk egy fuzzy modellel helyettesíteni, vagyis a γ jelszintcsökkenés alapján, figyelembe véve az azalatt eltelt időt, meghatározni az R csapadékmennyiséget.

Fuzzy szabálybázist használva finomabb módon tudjuk leírni a kapcsolatot a különböző változók között, mivel a szabályokban alkalmazott fuzzy halmazok nem éles határ szerint különítik el az egyes változók lehetséges értékeit, hanem az emberi gondolkodásmóddhoz hasonlóan, finom, fokozatos „fuzzy” átmenetek segítségével.

4.1. A fuzzy halmazok meghatározása

A fuzzy szabályrendszer állomáspáronként két bemeneti változót használ: a jelszint csökkenést és az eltelt időt.

A jelszint csökkenést hat kategóriába soroljuk:

- nagyon enyhe vételi szint csökkenés:
0 - 0.05 dB/km
- enyhe vételi szint csökkenés:
0.03 - 0.18 dB/km
- mérsékelt vételi szint csökkenés:
0.15 - 0.7 dB/km
- számottevő vételi szint csökkenés:
0.5 - 2.5 dB/km
- nagy vételi szint csökkenés:
1.8 - 5.5 dB/km
- nagyon nagy vételi szint csökkenés:
3.3 - 18 dB/km

A fuzzy halmazokat a fenti intervallumok felhasználásával határozzuk meg. Az intervallumok a fuzzy halmazok tartóit (pozitív tagsági értékű pontjait) jelentik.

A fuzzy halmazokat a 4. ábrán szemléltetjük. Megfigyelhető, hogy a fuzzy halmazok tartói részben átfednek, az egyes jelszint csökkenési osztályok között nem éles a határ.

Az eltelt időt négy kategóriába osztjuk: rövid, közepes, hosszú, nagyon hosszú. Ennek megfelelően a tartományok:

- rövid: 0 - 1 óra
- közepes: 0,5 - 4 óra
- hosszú: 3 óra - 4 nap
- nagyon hosszú: 3 nap - 1 év

A fuzzy halmazokat az előző változóhoz hasonlóan képezzük a tartományok alapján.

Az eltelt idő fuzzy halmazai az 5. ábrán láthatók. Itt is átlapolnak a tartók.

A fuzzy rendszer kimenete a csapadékmennyiséget tartalmazza. Ezeket az alábbi kategóriákba soroljuk:

- szitálás: 0 - 0,5 mm/óra
- gyenge esőzés: 0,25 - 1,75 mm/óra
- közepes esőzés: 1 - 7 mm/óra
- erős esőzés: 4 - 28 mm/óra
- záporosó: 16 - 54 mm/óra
- felhőszakadás: 35 - 150 mm/óra

Az ezek alapján meghatározott fuzzy halmazokat a 6. ábrán láthatjuk.

4.2. A fuzzy szabályok

A rendszer első bemeneti változóját hat fuzzy halmazzal írtuk le, a másodikat pedig négygel, ezért az összes lehetséges kombinációt 24 fuzzy szabállyal fedhetjük le. A szabályok kimenetében nem csak csapadékmennyiségek, hanem különféle nem időjárási hatásra utaló riasztástípusok is szerepelnek.

Az első bemenetet hatféle címkével jellemezhetjük. A jelszint csökkenés fokozatai A_i , ahol i értékei a következők:

- nagyon enyhe vételi szint csökkenés: 1
- enyhe vételi szint csökkenés: 2
- mérsékelt vételi szint csökkenés: 3
- számottevő vételi szint csökkenés: 4
- nagy vételi szint csökkenés: 5
- nagyon nagy vételi szint csökkenés: 6

Az eltelt idő fokozatai T_i címkéssel jelölhetők, ahol i értékei a következők:

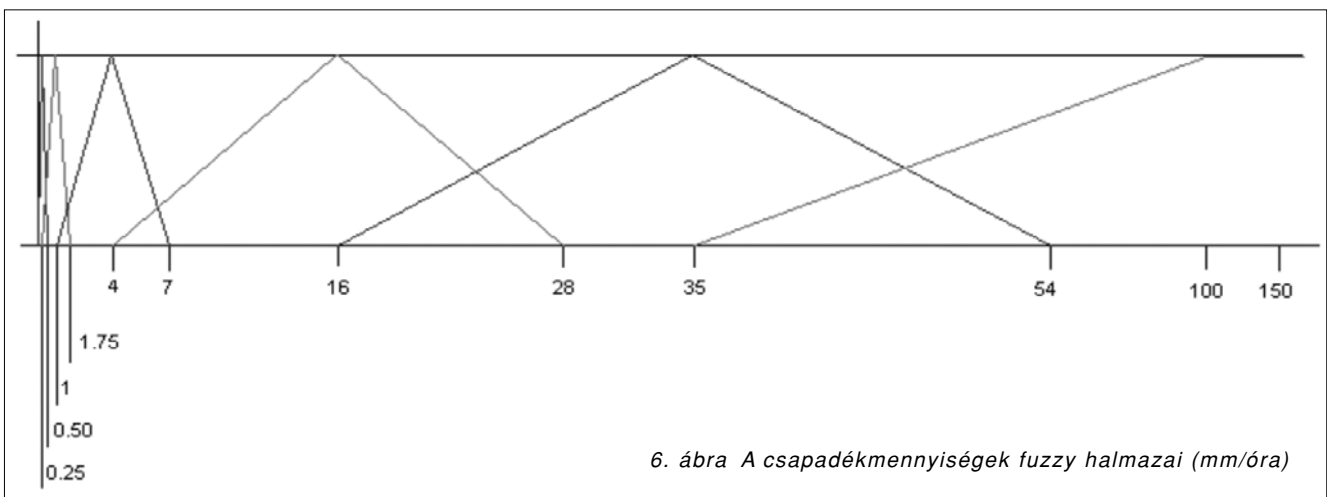
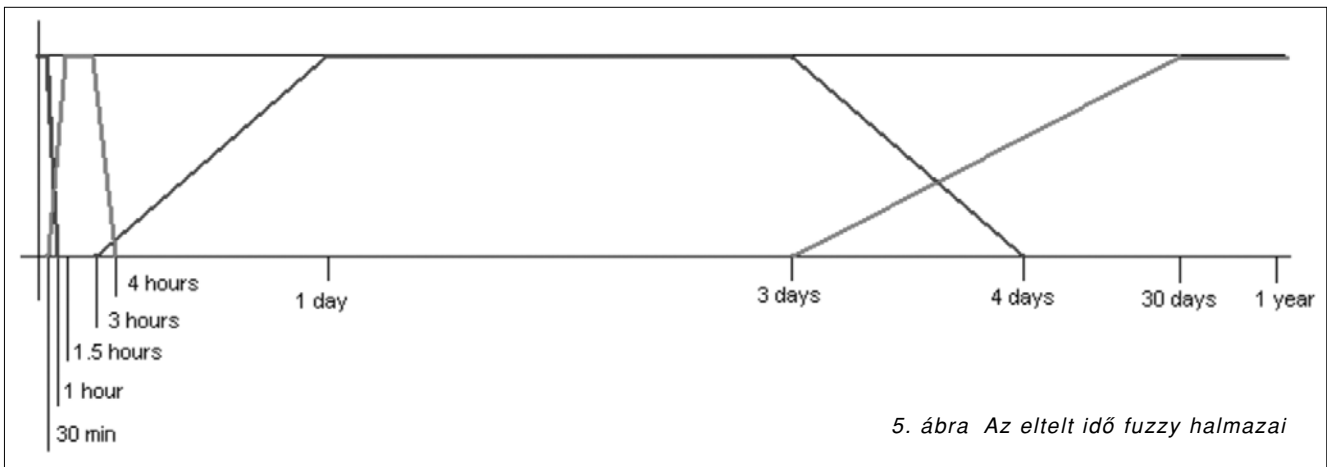
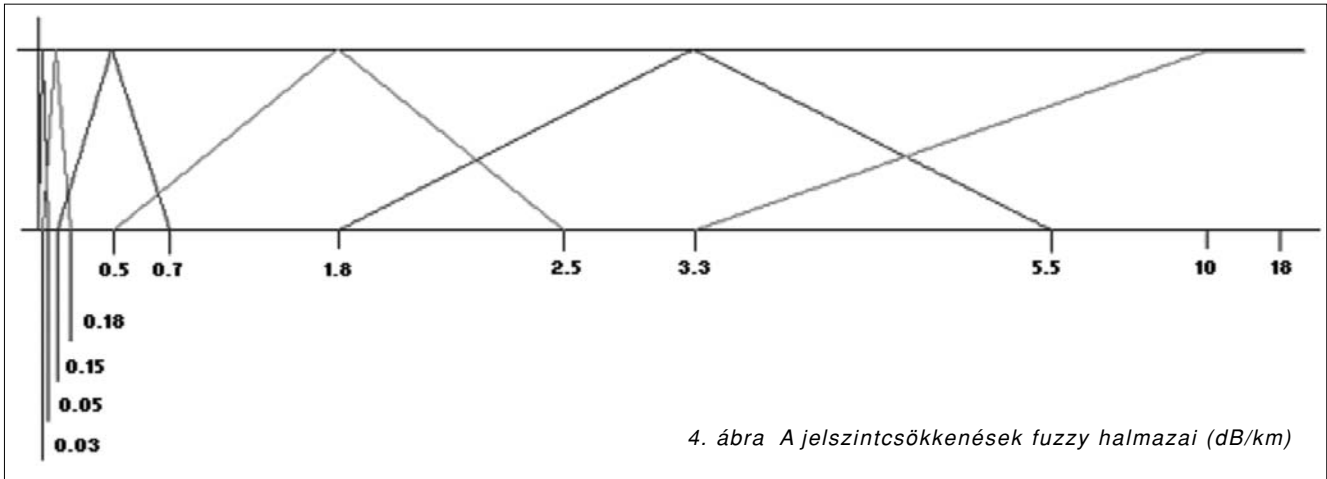
- rövid: 1
- közepes: 2
- hosszú: 3
- nagyon hosszú: 4

Fentiek alapján a következő szabályokat építettük fel:

- R1: Ha A_1 és T_1 akkor Szitálás
- R2: Ha A_1 és T_2 akkor Szitálás

R3: Ha A_1 és T_3 akkor Szitálás
 R4: Ha A_1 és T_4 akkor Figyelmeztetés
 R5: Ha A_2 és T_1 akkor Gyenge esőzés
 R6: Ha A_2 és T_2 akkor Gyenge esőzés
 R7: Ha A_2 és T_3 akkor Gyenge esőzés
 R8: Ha A_2 és T_4 akkor Figyelmeztetés
 R9: Ha A_3 és T_1 akkor Közepes esőzés
 R10: Ha A_3 és T_2 akkor Közepes esőzés
 R11: Ha A_3 és T_3 akkor Közepes esőzés
 R12: Ha A_3 és T_4 akkor Kis riasztás
 R13: Ha A_4 és T_1 akkor Erős esőzés

R14: Ha A_4 és T_2 akkor Erős esőzés
 R15: Ha A_4 és T_3 akkor Kis riasztás
 R16: Ha A_4 és T_4 akkor Kis riasztás
 R17: Ha A_5 és T_1 akkor Záporosó
 R18: Ha A_5 és T_2 akkor Záporosó
 R19: Ha A_5 és T_3 akkor Nagy riasztás
 R20: Ha A_5 és T_4 akkor Nagy riasztás
 R21: Ha A_6 és T_1 akkor Felhőszakadás
 R22: Ha A_6 és T_2 akkor Nagy riasztás
 R23: Ha A_6 és T_3 akkor Nagy riasztás
 R24: Ha A_6 és T_4 akkor Nagy riasztás



4.3. A következtetés

A következtetés során először a szabályok illeszkedési mértékét határozzuk meg az aktuális megfigyelés és a szabályok antecedensrésze alapján. A szabályok két csoportra oszthatók a kimeneteik szerint. Amennyiben az illeszkedés mértéke olyan szabály(ok)nak a legnagyobb, amely(ek) riasztás(oka)t tartalmaz(nak), akkor a fuzzy rendszer olyan következtetést hoz, mely a legnagyobb illeszkedési mértékű riasztásos szabály kimenetével egyezik meg.

Ellenkező esetben, amikor a csapadék vonatkozó sű következtetést tartalmazó kimenetű szabályok illeszkedési mértéke nagyobb, akkor ezen szabályok illeszkedési mértékeikkel arányosan csonkolt konzekvens fuzzy halmazainak unióján COG defuzzifikációt hajtunk végre, vagyis a Mamdani-féle következtetést alkalmazzuk.

A fuzzy rendszer által adott következtetést térképen is megjeleníthetjük. A felrajzolt négyzetek színei utalnak az esőzés mértékére. A színek és az esőzés összerendelését láthatjuk az 1. táblázatban.

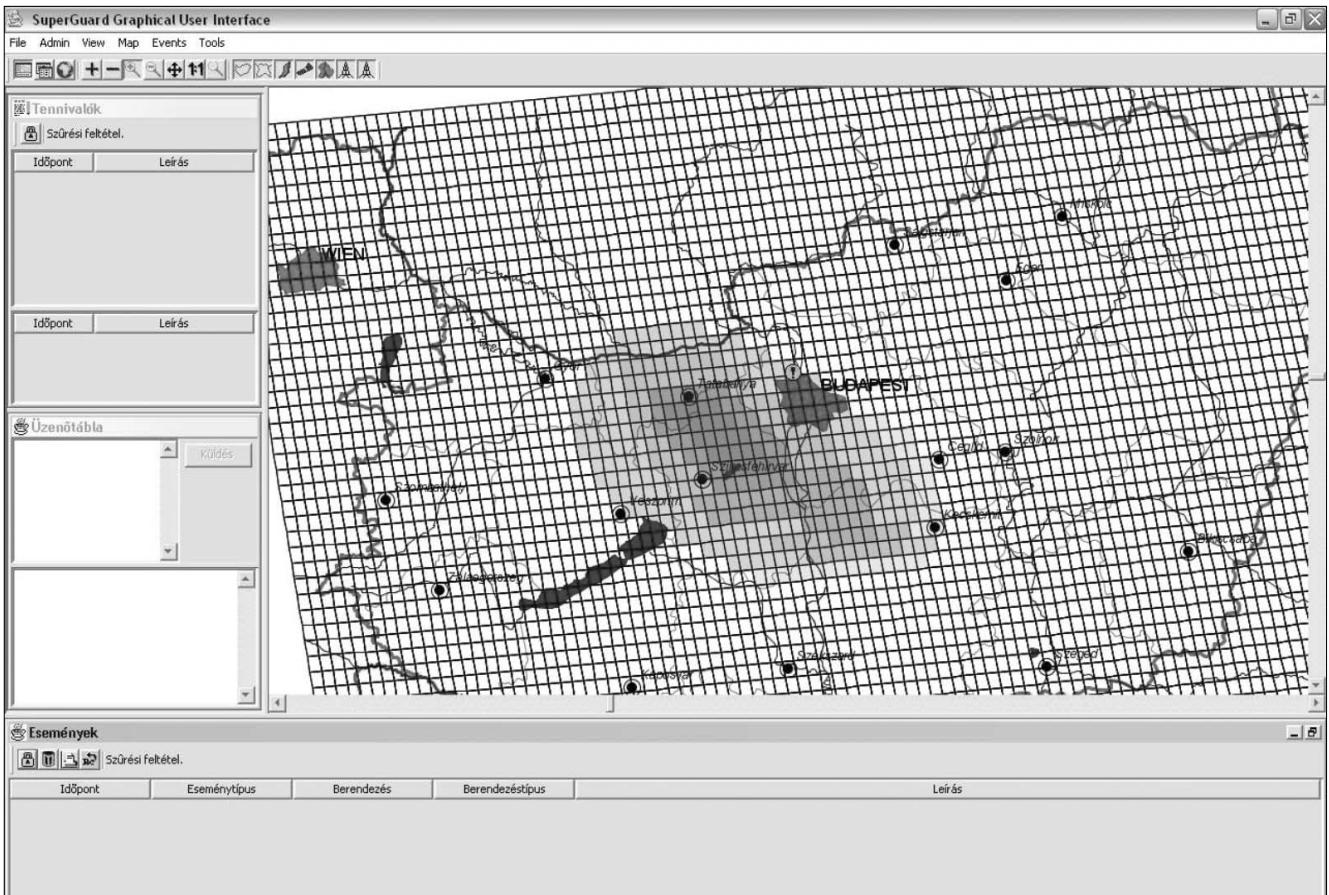
A hat csapadékkategórián kívül a riasztásokat is jellemezhetjük színekkel. Ezt látjuk a 2. táblázatban.




Végül a 7. ábrán a rendszer működését szemléltetjük egy térképes kezelői felületi képpel, melyet szimulált csapadék segítségével állítottunk elő. Magyarország térképét raszterhálózattal fedtük le, ily módon természetes, hierarchikus struktúrát adva a teljes rendszernek. Olyan területeken, ahol egy raszterelemen belül több állomás is található, átlagos viselkedést számí-

Szín	Csapadék típus	Szín kód
	Szitalás	1
	Gyenge esőzés	2
	Közepes esőzés	3
	Erős esőzés	4
	Zivatar	5
	Felhőszakadás	6

1. táblázat
Csapadék típusok és a hozzárendelt színek

tunk és jelenítünk meg, ahol pedig egy raszterelem esetleg üresen marad, interpolációval állítjuk elő a csapadék becsült értékét. A kék és lila színek a szimulációs adatok alapján a feltételezett csapadékmennyiséget mutatják.



Szín	Riasztás	Szín kód
	Figyelmeztetés (kék)	7
	Kis riasztás (sárga)	8
	Nagy riasztás (piros)	9

2. táblázat Riasztások és színeik

5. Összefoglalás és további tervek

A jelen tanulmányban a fuzzy rendszerek alapjait tekintettük át, kitértünk a hierarchikus fuzzy rendszerek megemlítésére és egy alkalmazási példát is bemutatunk. A fuzzy rendszer jól használható döntéstámogató eszközként olyan alkalmazásokban, amelyekben az emberi tudás fuzzy szabályok formájában egyszerűen reprezentálható. Jövőbeli tervünk, hogy kiterjesszük az intelligens döntéstámogató rendszert valódi hierarchikus fuzzy szabálybázisokat használó rendszerré, még egyszerűbben használhatóvá és finomabbá téve a modellt ezáltal.

Ugyanakkor megkeresést kaptunk egy hazai hálózat üzemeltetőtől az intelligens döntéstámogatási alrendszer saját felügyeleti rendszerükkel történő integrálása tárgyában. Így a jövőben tervezzük az éles üzembehelyezést is.

Irodalom

- [1] Kóczy T. L., Tikk D.:
Fuzzy rendszerek, Typotex, Budapest, 2000.
- [2] L. A. Zadeh:
Fuzzy sets. Information and Control, 8(3):338–353, 1965.
- [3] E. H. Mamdani and S. Assilian:
An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller.
Int. J. Man-Mach. Stud. Vol. 7:1–13, 1975.
- [4] M. Sugeno, M. F. Griffin, and A. Bastian:
Fuzzy hierarchical control of an unmanned helicopter.
In Proc. of the 5th IFSA World Congress (IFSA'93), pp.1262–1265., Seoul, 1993.
- [5] Üzemfelügyeleti rendszer mesterséges intelligenciával.
Projekt zárójelentés.
Projekt azonosító: NKFP-2/0015/2002.
Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék,
konzorciumvezető, szakmai vezető: Dr. Kóczy T. László,
Linecom Kft., szakmai vezető: Veres Lajos.
2005. szeptember

Könyvet ajánlunk

Emberk és események

Szubbjektív visszaemlékezések a távközlés elmúlt 50 évére

Olasva Lajtha György professzor könyvét, aki hosszú éveken át a Posta Kísérleti Intézet tudományos igazgatóhelyettese volt, – és azóta is elkötelezett feladatának tekinti az Intézet tudományos tevékenységének istápolását –, az első érzés az irigységé...

Szinte el sem tudja képzelni az ember, hogy miként tud egy sokféle elfoglalt ember – igaz, nyugdíjas, de akik ezt az állapotot elérték váltig állítják, hogy ez életük legelfoglaltabb időszaka, s ez Lajtha professzorra sokszorosan igaz – energiát és időt fordítani egy ilyen hosszú és szerteágazó tudományos életpályára való visszatekintésre.

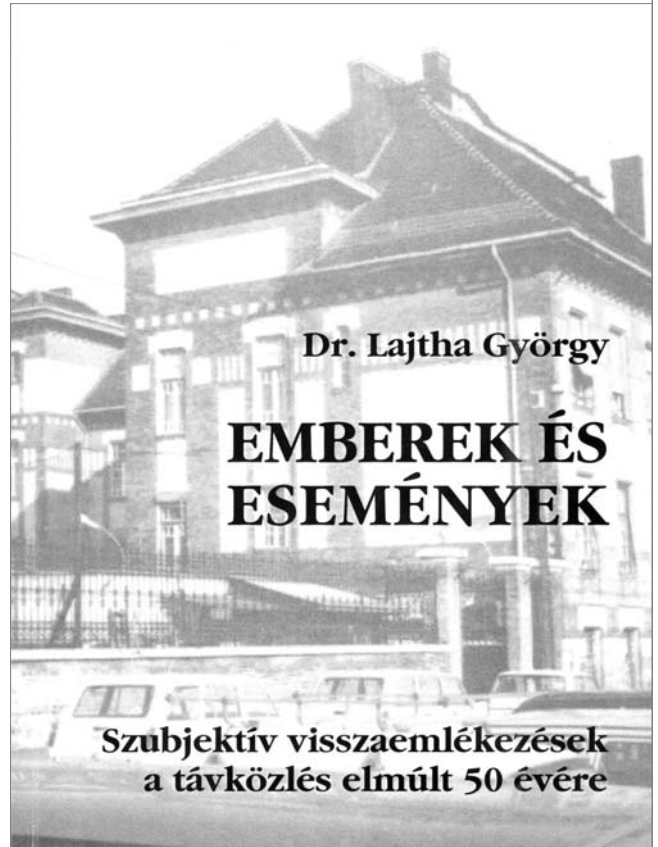
Hogyan tudta és tudja az agyában elraktározni, rendszerezni, olvasmányos formában közreadni és a memóriájából mindig a legmegfelelőbb információt előhívni...?

A második: a könyv a „szubbjektív visszaemlékezések a távközlés elmúlt 50 évére” alcímet viseli. Én, mint ennek az időszaknak aktív részese, szívemhez is közel állóan és szintén szubbjektíven tudom ezt a kötetet olvasni. Határozott érzésem, hogy könyvét olyan érdekes formában írta meg, – a szakmai olvasóközönség által érthető részeket általános érvényű megállapításával ötvözve –, hogy ezzel mindenki számára élvezhetővé tette.

A könyv iskolapéldája a szakmai ihletésű visszaemlékezéseknek. De nem csak a szakmát szerető, hanem a szélesebb érdeklődésű olvasóközönség is példát kaphat arra, hogyan alakulhat ki egy sikeres tudományos életpálya.

A szerző szokásos könnyed „lediktált” stílusban tanít, emlékeztet és vezet be a távközlés elmúlt időszakának eseményeibe, majd mindezek melléktermékeként észrevétlenül nevel, támpontot ad a jövő generációjának, hogyan lehet egy szakmát tudományosan, eredményesen művelni, szeretni.

Lajtha György visszaemlékezéseit tizenegy fejezetben meséli el. Az első fejezetekben tiszteleg a pályáját elindító és segítő személyek, intézmények emléke előtt, majd a következő fejezetekben saját és a távközlés fejlődésének legfontosabb állomásait veszi sorra. A visszatekintés döntő mértékben a Posta Kísérleti Intézet (PKI) szemszögéből idézi fel az eseményeket.



Emlékeit rendszerezve, többnyire időrendi sorrendben, a távközlés fejlődési folyamata mentén tárgyalja, bemutatva azokat a személyeket, akiknek munkássága egy-egy időszakban a távközlés fejlődésére és a PKI hírnevére hosszabb időre kihatott.

Kiemelten foglalkozik a hazai fejlődést erőteljesen meghatározó nemzetközi együttműködésekkel és a nemzetközi szervezetekben való eredményes részvételünkkel.

A könyv nyelvezete már csak azért is olvasmányos és közérthető, mivel szerzője a szakmai publikációknál is mindig hivatásának tekintette a magyar nyelv igényes használatát.

Végül egy személyes élményem, hogy a kézirat elolvasása a karácsonyi szabadságom idejére esett, így a már betervezett könyvek elolvasására szánt időt „rabolta” el tőlem. Nem tudtam letenni. Igazi szakmai, történelmi csemege.

Kántor Csaba

Routing protocols in wireless sensor networks

Keywords: sensor networks, routing

Different applications of sensor networks pose different requirements toward routing protocols. Due to the variety of requirements, many protocols have been proposed in the literature. In this paper, we provide a systematic overview of routing protocols and present a prominent representative of each class. Novelty of the paper is the set of characteristics which resulted in a more detailed taxonomy of protocols.

Mobile devices in sensor networks

Keywords: wireless sensor networks, mobility, sink nodes

Wireless sensor networks (WSNs) have recently become one of the most investigated topics in the field of communication networks. The building blocks of these networks are the sensors, sensing various physical phenomena, and the sink nodes, gathering the sensor data. At the beginning, the WSNs were supposed to be static; however, more and more researchers today are proposing WSNs with mobile sensors and sinks. The mobility of the elements makes it possible to enhance energy efficiency, and improve other features of the network as well. In this paper we present a survey of the most important proposals dealing with mobile elements in wireless sensor networks.

Energy efficient communication in sensor networks

Keywords: sensor network, energy efficiency, clustering, routing, data aggregation

A sensor network can be seen as a special case of wireless ad-hoc networks where the nodes are tiny devices with low computational power and different sensing capabilities. Typically, the network covers a large area and consists of a huge number of devices. One of the most important factors in network planning is its energy efficiency, since sensors have limited energy supply, and it is not economical (if possible at all) to replace or recharge the batteries. In this paper the implementation issues of energy efficient communication techniques in sensor networks are investigated.

CASCADAS – Autonomic communication and pervasive situated services

Keywords: autonomous communications, self-organization, knowledge network, pervasive control

CASCADAS is one of the four IST-FET projects that aim to provide both theoretical and practical background for a new generation of complex, distributed, pervasive services. The basic building block of the situation-aware, self-organizing, autonomic communication based network is a common abstraction called ACE (Autonomic Communication Element). ACEs provide and use services, adapt to the situation, create and manage plans, build up knowledge networks, and move and self-organize based on their own autonomic decisions. The paper gives detailed introduction to the project vision, describes the concepts behind, and offers insight into the ongoing work.

Intelligent Transportation System's performance with the help of MANETs

Keywords: Intelligent Transportation System, traffic jam avoidance, ad-hoc communication, realistic simulations

Nowadays, there is an increasing demand from passengers and drivers for comfortable and secure travelling. Even if we have expensive, secure and fast cars, we may still not be able to reach our destinations because of the traffic jams possibly formed. The first reaction of every person in this case is „I wish I knew about this jam, I would have chosen another route!”. In this paper we propose a solution, which gives a response to this sigh.

Ambient Intelligence on Public Roads

Keywords: traffic, electronic control unit, ambient intelligence, WLAN

One metric of economical development of a country is the number of personal and commercial vehicles per person. The personal car is a symbol of the lifestyle here, while incidence of commercial vehicles shows how the economy is beating. Mobility of labour and goods in such countries is a basic requirement. Rolling goods of a country however is not only a cause of pride, but a real problem, because their high amount hinders their move. The instruments of Ambient Intelligence give a new approach. Systems belonging here are not crucially and directly sensible for humans, but by their help life gets easier and safer.

Webserver on mobile phones – a new way of 3G content sharing

Keywords: 3G, content sharing, mobile content, mobile web server, mobile peer to peer

Due to the growing adoption of 3G infrastructure and mobile devices, customer requirements are now beyond simple web browsing: a new way of mobile content sharing and IP-based cooperation of mobile devices are necessary. We have assembled a test environment for benchmarking, and our results show that the mobile web server concept is a viable alternative: using efficient, object-oriented Python scripts on the server side and a standard web browser as a client. It is also surprisingly efficient: it can serve up to 10-15 concurrent users on a 3G connection with reasonable speed. We also mention several new application areas, however, we expect that mobile web server technology will lead to many new applications on 3G mobile devices in addition to the discussed case studies.

Design considerations of low power integrated circuits

Keywords: low-power CMOS circuit, adiabatic charging, charge recovery circuit, dynamic storage element

Nowadays the low-power consumption of the electronic circuits is essential. Adiabatic charge-recovery circuits can satisfy such requirement. Their main benefit is that internal capacitances of their circuits can be charged and discharged without Joule-loss. The paper presents an improved adiabatic charge-recovery circuit and its application to build low-power dynamic storage elements and describes a particular task of the large scale adiabatic circuit design, namely the optimal scheduling of operations.

Application of Fuzzy decision support systems for supervision of mobile networks

Keywords: mobile networks, supervision systems, intelligent techniques, fuzzy logic

In mobile telecommunication networks the transmission level is affected by objects located between transmitter and receiver stations in the so-called Fresnel-zone. These obstacles may get temporarily or permanently into the zone, they can be artificial or natural objects, or they can be of meteorological origin, too, like rain or fog. It is reasonable to use intelligent decision making subsystems which can decide from the degree of attenuation and from its time dependent behaviour what the reason of the attenuation could be. The paper demonstrates the intelligent module of a network supervision system created in the framework of a successfully completed National R&D Programme project. Intelligent decision support systems and the basics of fuzzy logics are introduced. In the next an application for automatically identifying the weather situation is discussed in some detail.

ZTE NG-DSLAM

Solution Overcomes the Challenges of NGN DSLAM Design

Digital subscriber line access multiplexers (DSLAM) are a leading broadband access technology, delivering high-speed data transmission over existing copper telephone lines. The next generation of DSLAM platforms has to deliver greater density, higher bandwidth and lower cost per port. Any solution must be able to handle the diversity and inter-operation of networking technologies such as IP-to-ATM switching. The first challenge in the future evolution of the DSLAM market though is to enable migration from centralised ATM-based systems to a decentralised IP-based architecture requiring much greater data processing capability. ZTE NG-DSLAM technology is the ideal solution to enable this migration from ATM to the IP-based structure of the next generation platform.

1. Design challenges for next generation DSLAM

The ability to deliver voice, video and data is the goal of every telecom operator, and next generation DSLAM offers the potential to deliver these multiple revenue streams from a single IP packet network.

There are many design challenges for a successful DSLAM line-card, including:

- The ability to support the transition and eventual convergence of ATM and IP networks
- Increased port density - typically 24 to 96 ports
- Increased bandwidth from high performance services and standards, such as ADSL-> ADSL2+ and VDSL->VDSL2/+ for "triple play"
- Intelligence moving to the edge of the network as a traditional DSLAM assumes some edge router functionality
- Maximising the existing infrastructure to accommodate more users
- Maximising Return On Investment (ROI) by reducing cost per port or user
- Supporting required protocols such as ATM, IP, VLAN, RTP, PPP, PPPoE/oA, L2TP, Diffserv, ML-PPP, MPLS
- POS, Routing, Bridging, QoS, etc.
- Managing QoS to ensure prioritisation of latency sensitive traffic to meet service level agreements
- Configurable bandwidth based on policy configuration across ATM, PPP, Ethernet, and IP technologies

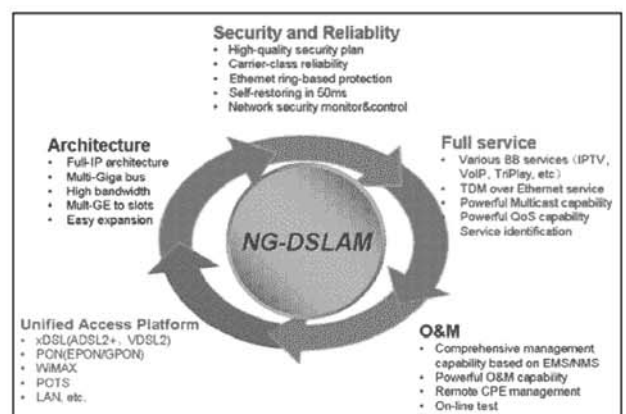
- Multiple QoS dictates the need for different hierarchical IP or ATM traffic scheduling and shaping capability
- Ensuring the security/confidentiality of subscriber information
- Providing the ability to add new features and functions through software based on market and subscriber demand

A robust DSLAM line-card design must also combine the best density, throughput functionality, scalability and cost-efficiency to provide equipment vendors with a distinct competitive advantage.

2. ZTE NGN-oriented NG DSLAM

To accommodate these requirements for the development of next generation networks and services, broadband access layer equipment is evolving to a platform - NG DSLAM.

ZTE NG DSLAM is designed with a multi-Giga, multiple-bus architecture to enable an extensive range of ultra bandwidth services. It features flexible networking capabilities with carrier-class reliability and full-service protection functions. This is a new type of access layer equipment with wired (xDSL, POTS, LAN), wireless (WiMAX, WLAN), and fibre (EPON/GPON) access capabilities. As a unified service access platform, it provides network management, QoS management, security management, and operation and maintenance management oriented to the various services provided. Its full service open architecture supports completely seamless integration with the NGN. Effectively, it is an NGN-ready super IP-DSLAM access platform. As it has evolved from the ATM/IP DSLAM, the NG DSLAM is compatible with the functions of the earlier equipment.



3. Features of ZTE NGN-oriented NG-DSLAM technology

Compared with earlier DSLAM access equipment, the significant characteristics of ZTE's NG DSLAM technology are:

- Multi-Giga bus, and multiple Gigabit service slots
- Full-service open architecture supporting full service access, bearer and transmission, to implement full service platform-based access
- Extremely powerful access capability supporting multiple access modes
- Flexible networking capabilities with carrier-class reliability and full service protection
- Full IP architecture, multi-Giga bus and easy expansion
- High-quality security solution, high reliability Ethernet-ring protection, 50ms self-recovery and network security monitoring
- Full range of broadband services (IPTV, VoIP, and triple play), TDM over Ethernet, powerful multicast capability, service identification, service perception and multi-service edge
- Sound EMS/NMS capability and powerful operation & maintenance capability
- Unified Access supports xDSL access, PON optical access, wireless access, POTS access and LAN access

4. ZTE NG DSLAM solution - FSAP (Full Service Access Platform)

FSAP is the full service access platform for ZTE's NG DSLAM solution. Building on its long experience in broadband access R&D, ZTE has launched the new-generation ZTE FSAP to address the needs of large-capacity, high-bandwidth and high-quality video, audio, data, and interactive services. It also addresses the need for the access layer to evolve to the NGN.

The ZTE FSAP is a new multi-service access platform that provides broadband services such as XDSL, LAN and PON. It allows carriers to build access layer networks with full service bearer capability.

The ZTE FSAP family consists of the following: FSAP 9800 with large capacity and active/standby redundancy; FSAP 9803 with medium capacity; FSAP 9805 with medium to small capacity and high/low temperature resistance of -40°C ~ +65°C outdoors; FSAP 9806 with small capacity

Pizza box; and FSAP 9807 with small capacity and high/low temperature resistance of -40°C ~ +65°C outdoors.

The large capacity FSAP 9800, Macro FSAP 9803 and Pizza box 9806 have been already put into commercial application.

5. Typical application areas of ZTE NG DSLAM technology

ZTE NG DSLAM can implement the following networking applications:

- Mixed insertion of EPON/GPON cards, enabling the concurrent delivery of services
- Mixed insertion of Wimax cards, meeting pilot and small-scale service needs
- Smooth upgrade to support access of VoIP and POTS, providing voice services

The ZTE NG-DSLAM system has fully optical access equipment PON with powerful broadband functionality and meets the optical access networking requirements of TDM access by using GPON. It also provides VoIP voice functions and supports Gigabit Ethernet ring protection enabling reliable, full-service, high-bandwidth transmission. It also supports the extremely powerful broadband functionality that can be achieved by adding DSL boards.

6. Commercial application of ZTE NG-DSLAM

Many ZTE NG DSLAM systems are in commercial applications around the world. To date, ZTE has deployed over 15 million DSL products in more than 40 countries including Romania, Greece, Egypt, India and Pakistan. In 2005, Gartner reported ZTE as one of the top three DSL providers in the world. Also in 2005, ZTE became a global supplier of ADSL equipment to France Telecom. At the 2004 Olympics ZTE's DSL successfully provided athletes, referees, journalists and spectators from all over the world with quality broadband services in 16 venues and news centres. Other regions like Hong Kong, and the mainland provinces such as Guang Xi and Shang Hai have deployed ZTE DSLAM systems already.

With over 10% of its revenues committed to R&D every year, ZTE is developing the very latest DSL technologies, including ADSL2/2+ and VDSL/VDSL2. In addition, ZTE also offers new technologies such as SNOOPING ++, on-line usage monitoring software and M+1 terminal protection to provide operators with next-generation broadband products.

Contents

<i>AMBIENT INTELLIGENCE</i>	2
Gergely Ács, Levente Buttyán Routing protocols in wireless sensor networks	3
Zoltán Vincze, Roland Vida Mobile devices in sensor networks	12
Dorottya Vass, Attila Vidács Energy efficient communication in sensor networks	18
Borbála Benkő, Tamás Katona, Róbert Schulcz CASCADAS – Autonomic communication and pervasive situated services	23
Lóránt Vajda, Sándor Kardos, Gábor Gerháth, Balázs Mezny, Péter Laborczi, Géza Gordos Intelligent transportation system's performance with the help of MANETs	29
Bence Csák Ambient Intelligence on public roads	35
István Dévai, Gábor Bányász, Tihamér Levendovszky Webserver on mobile phones – a new way of 3G content sharing	40
László Varga, Gábor Hosszú Design considerations of low power integrated circuits	45
László Kóczy T., János Botzheim, Richárd Sallai, Kornél Csányi Application of fuzzy decision support systems for supervision of mobile networks	52
Csaba Kántor Book review – Dr. György Lajtha: People and events	60
<i>ZTE: NG-DSLAM Solution Overcomes the Challenges of NGN DSLAM Design (x)</i>	62