híradástechnika

1945 VOLUME LX. 2005

hírközlés ▪ informatika

Multimédia kommunikáció

Hálózati protokollok

„Ambient" hálózatok

2005/7

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

Tartalom

Beköszöntő

zombory@mht.bme.hu
szabo@hit.bme.hu

A jelen júliusi számmal az új szerkesztőbizottság és főszerkesztő átveszi a „stafétabotot” Dr. Lajtha György professzortól és szerkesztőbizottságától.

Szeretnénk bemutatni a most munkába álló szerkesztőbizottságot. A nevek remélhetőleg az olvasók széles körében ismertek, valamennyien szakterületük kiváló képviselői és ami fontos, lelkesen mondtak igent a felkérésre és vállalták az ezzel járó szakmai munkát. A nevek mellett feltüntettük az elsősorban gondozandó témakört is, rögtön hozzátéve, hogy ez utóbbiak nincsenek „kőbe vésve”, hiszen gyorsan változó szakmánkban gyakran új területek merülnek fel, mások pedig átalakulnak, illetve kevésbé hangsúlyosakká válnak.

- *Bartolits István (NHH)*
  – távközlés-menedzsment, gazdaság, szabályozás, jog
- *Bársony István (MTA-MFA)*
  – új híradástechnikai eszközök és technológiák
- *Buttyán Levente (BME-HT)* – informatikai biztonság
- *Győri Erzsébet (BME-TMIT)* – távközlési szoftverek
- *Imre Sándor (BME-HT)*
  – mobil kommunikáció és számítástechnika
- *Kántor Csaba (Magyar Telekom PKI)*
  – műholdas távközlés
- *Lois László (BME-HT)* – multimédia kommunikáció
- *Németh Géza (BME-TMIT)*
  – beszédfeldolgozás, szolgáltatás-automatizálás
- *Paksy Géza (BME-TMIT)* – optikai távközlés
- *Prazsák Gergő (NHIT)* – társadalmi vonatkozások
- *Tétényi István (MTA-SzTAKI)*
  – kutatói hálózati infrastruktúrák, tesztágyak
- *Veszely Gyula (BME-SzHVT)*
  – EM terek, hullámterjedés, antennák, RFI, EMC
- *Vonderviszt Lajos (NHH)* – internet és WWW

A szerkesztőbizottságban a folytonosságot Zombory László biztosítja, aki ezután a szerkesztőbizottság elnöke lesz. Lapunk egyik védnöke viszont változik: Dr. Sallai Gyula lép be a HTE nemrég megválasztott elnökeként.

A lap profilja szándékunk szerint továbbra is széles marad: egyfelől felöleli a HTE legfontosabb műszaki témaköreit, tehát a távközlés „klasszikus” témái mellett, az informatika távközléshez, kommunikációhoz kapcsolódó vonatkozásait, a média-technológiák és média-kommunikáció kérdéseit, ezzel is elősegítve a távközlés-informatika-média konvergenciájának folyamatát. Másfelől megtartjuk és erősítjük az interdiszciplináris jelleget is, helyet adva a távközléshez és média-kommunikációhoz kapcsolódó gazdasági, szabályozási, marketing, menedzsment témáknak és a távközlés-informatika-média társadalmi vonatkozásainak is.

Megpróbálunk a lehető legszélesebb szakmai körhöz szólni. Ezt a célt elsősorban azzal szeretnénk szolgálni, hogy egy lapszám első, általában nagyobbik részét meghívott, „tutorial” jellegű, közérthető, világos szaknyelven cikkekből állítjuk össze, amelyekből az olvasó megismerheti az adott téma helyzetét, újabb eredményeit, fejlődésének irányait. A tutoriális jelleg nem felszínességet, hanem érthető és ha lehet, élvezetes összefoglalókat kell jelentsen azok számára, akik nem ismerik közelebbről az adott szakterületet.

Az adott szám másik része publikációs fórumot fog biztosítani az új eredmények ellenőrzött közreadására. Itt tehát más követelménynek kívánunk eleget tenni: annak, hogy a kutató-fejlesztő szerző számára, bírált publikációs fórumként szolgáljunk.

A továbbiakban is tervezünk évi két angol nyelvű számot. A júniusi és decemberi lapszámainkat az év folyamán magyarul közölt legjobb beküldött-bírált cikkekből állítjuk össze.

Nem lenne teljes a beköszöntő, ha nem szólnánk végül a lap „lelkéről” az előző öt évben. Dr. Lajtha György, a hírközlés- és távközléstechnika örökké fiatal nagy öregje meghatározó szerepet játszott a folyóirat megújulásában és folyamatos szintentartásában. Páratlan szakmai felkészültsége és soha nem lankadó érdeklődése nagy munkabírással párosult. Előzetes lapszerkesztői tapasztalatait felhasználva szerkesztette a Híradástechnikát. Értékelte és lektorálta a cikkeket és ugyanakkor a teljes spektrum biztosítására fáradhatatlan energiával biztatta fiatal és kevésbé fiatal kollégáit a szakterületükre tartozó jól kiválasztott témák megírására. Tőle származik a 2-3 cikkből álló kis – cikkcsokrok – meghonosítása. Neki köszönhető, hogy a lap egyszerre tudta szolgálni a szakmai részletekben elmélyülni kívánó olvasót, az érdekes újdonságokra vadászót, a szakma társadalmi-gazdasági vonatkozásai iránt érdeklődőt. A lap tartalmát tömören összefoglaló és kitekintést adó vezércikkei pedig kedvet teremtettek a lapszám forgatásához.

Reméljük, hogy – bár visszavonul az aktív lapkészítéstől – nem válik el teljesen tőlünk. Számítunk Lajtha György hasznos tanácsaira, várjuk az észrevételeit és mindig örömmel fogadjuk a cikkeit a folyóiratban.

Dr. Zombory László,
a Szerkesztőbizottság elnöke

Dr. Szabó Csaba Attila,
főszerkesztő

Multimédiás szolgáltatások lokális adaptációja rétegek közötti kommunikáció felhasználásával

KOVÁCS ÁKOS, TAKÁCS ATTILA

Traffic Lab., Ericsson Magyarország Kft.
{akos.kovacs, attila.takacs}@ericsson.com

Kulcsszavak: multimédia, vezetéknélküli hálózatok, rétegek közötti kommunikáció

Vezeték nélküli hálózatokat egyre szélesebb körben és feladatokra használunk, sőt egyre dinamikusabban nő a vezeték nélküli multimédiás szolgáltatások iránti igény is. Számos kutatás és fejlesztés célozta meg a kihívásokkal teli valósidejű adatátvitel problémáit. Az egyik újra és újra napirendre kerülő téma a rétegek közötti kommunikáció. Az eredmények azt mutatják, hogy jelentős szolgáltatás-minőség javulás érhető el a szigorú rétegstruktúra fellazításával. Cikkünkben egy olyan lokális adaptáción alapuló eljárást mutatunk be, mely a skálázható audio/video kódolók adta lehetőségeket kihasználva a multimédia-forgalom átvitelének hatékony támogatására alkalmas. Ahhoz, hogy a vezeték nélküli csatorna képességeihez mérten a legjobb minőséget lehessen a felhasználónak biztositani, kiaknázzuk a rétegek közötti kommunikáció nyújtotta előnyöket.

1. Bevezetés [1]

A vezeték nélküli hálózatok és a multimédiás szolgáltatások gyors elterjedése miatt, az utóbbi időben egyre nagyobb az igény egy olyan átfogó koncepció kialakítására, amely a mai és a jövőbeni heterogén, vezetékes és vezeték nélküli, hálózatokon egyaránt hatékony, a felhasználók felé transzparens multimédia-átvitelt tesz lehetővé. Fontos tervezési feltétel, hogy amennyire lehet, a rendszer független legyen mindenféle média-típustól és hálózati architektúrától.

A multimédiás adatforgalom két lényeges pontban különbözik a szakirodalomban már sokat vizsgált TCP alapú szolgáltatásoktól. Az első különbség, hogy a „valósidejűség” követelménye miatt a késleltetést és a jittert szoros korlátok közé kell szorítani. A másik eltérés, hogy nincsen szükség adatveszteség mentes átvitelre, hiszen a multimédiás tartalom bizonyos veszteséget tolerálni képes. Alkalmas kódoló alkalmazása esetén lehetnek például olyan csomagok, amelyek csak minőséget javító információt hordoznak, ezért kisebb prioritásúak egy film visszajátszása szempontjából. Olyan kódolási eljárás is létezik, ahol még egy csomagon belül is megkülönböztethetőek különböző fontosságú bájtcsoportok.

Egyre inkább terjednek a vezeték nélküli hálózati megoldások. A vezeték nélküli csatorna sajátságos karakterisztikája (például a többutas terjedés okozta késleltetés és késleltetés-ingadozás) miatt a valós idejű forgalom átvitele sokkal kritikusabb, mint a hagyományos vezetékes hálózatokban. A jelentős hibázási valószínűség miatt a sűrűn használt adatkapcsolati rétegbeli újraküldési mechanizmus a késleltetést megnöveli. Ahhoz, hogy egy ilyen közegben a lehető legjobb videó-átviteli minőséget tudjuk biztosítani, nem elegendőek a rétegelt hálózati architektúra által nyújtott rétegek közötti kommunikációs lehetőségek. Az alkalmazási szintű minőségi igények az alsóbb (például hálózati vagy adatkapcsolati) rétegekben sérülhetnek, ami rontja az átvitt videó-folyam minőségét. Ennek kiküszöbölésére megoldást nyújthat egy részletesebb kommunikációt is megengedő, rétegek közötti (Cross-Layer) üzenetváltásokon alapú jelzésrendszer [5,6]. Ezzel a módszerrel alkalmazási szintű információt tudunk biztosítani az eltérő fontosságú videó keretekről hálózati és adatkapcsolati szintnek.

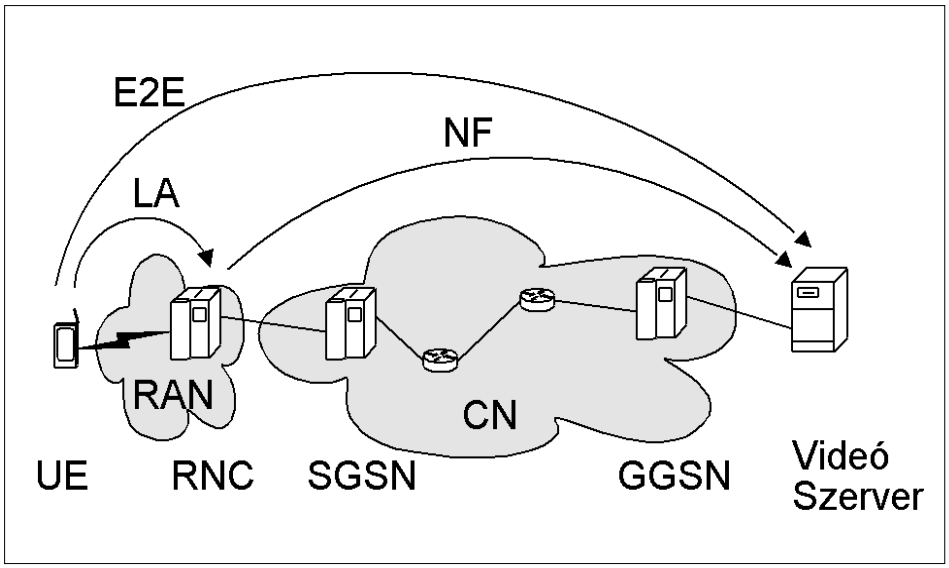
Megfelelő eljárásokkal és a rétegek közötti kommunikáció előnyeinek kiaknázásával megtervezhető egy olyan multimédia átvitelére alkalmas rendszer, amely dinamikus módon alkalmazkodni képes a vezeték nélküli hálózat okozta átviteli hibákhoz. Két hatékonyan alkalmazható eljárás a sávszélesség adaptáció (Rate

Rövidítések

AVC	Audio Video Coding
CL	Cross-Layer
E2E	End-to-End Feedback – *végpontok közötti visszacsatolás*
EEP	Equal Error Protection – *egyenlő hibavédelem*
LA	Local Adaptation – *lokális adaptáció*
L-Id	Layer ID
NALU	Network Abstraction Layer Unit
NF	Network Feedback – *hálózati visszacsatolás*
Q-Id	Quality ID
RA	Rate Adaptation – *sávszélesség adaptáció*
RFL	Réteg-Független Leíró
SVC	Scalable Video Coding – *skálázható videó kódolás*
T-Id	Temporal ID
UEP	Unequal Error Protection – *változó erősségű (egyenlőtlen) hibavédelem*

1 Munkánkat részben az EU IST M-Pipe projekt támogatta.

Adaptation) és a változó erősségű hibavédelem (Unequal Error Protection). A RA célja, hogy az átvitt adat mennyiségét csökkentsük a kevésbé fontos adatok eldobásával. Ezzel lehetőség nyílik a változó rádiós csatorna aktuális átviteli képességeihez való alkalmazkodásra. Az UEP egy hálózati szintű csomagon vagy adatkapcsolati szintű kereten belül alkalmazható hibavédelemi módszer. Az adat különböző fontosságú részeinek eltérő erősségű védelmére használják. Az UEP lehetőségeit például az AMR kódoló is kihasználja [1].



2. ábra
A legfontosabb adaptációs helyek egy UMTS hálózatban

Ahhoz, hogy a RA és UEP technikákat a vezeték nélküli csatornán használni tudjuk, szükség van alkalmazási szintű információk összegyűjtésére és a csomópontok között való szétterjesztésére. Cikkünkben erre adunk javaslatot. A következőkben bemutatjuk a különböző szolgáltatás-adaptációs lehetőségeket. A harmadik fejezetben áttekintést adunk a kódolási eljárásokról, kiemelve az adaptációt elősegítő skálázható kódolókat. A negyedik fejezet a lokális adaptációra általunk javasolt megoldás eszközeit ismerteti, míg az utolsó részben konkrét példán mutatjuk be működését.

2. Szolgáltatás-adaptációs lehetőségek áttekintése

Az adaptációs mechanizmusok két fő információfolyamra bonthatóak. Az első, az úgynevezett Visszajelzés (Backward Information), ami tulajdonképpen nem más, mint a visszacsatolás a manapság is működő megoldásokban. Az adaptáció szükségességének jelzésére szolgál, és legtöbbször a vevő alkalmazás küldi a forrásnak, például Real-Time Control Protocol (RTCP) [2] vagy Real-Time Streaming Protocol (RTSP) [3] üzenetek formájában.

A második típus az úgynevezett Szerver Oldali Információ (Forward Information), ami a forrás oldalon működő kódoló alkalmazás nyújtotta adaptációs lehetőségekről nyújt információt a hálózat számára. A kétféle jelzési mechanizmust mutatja az *1. ábra*.

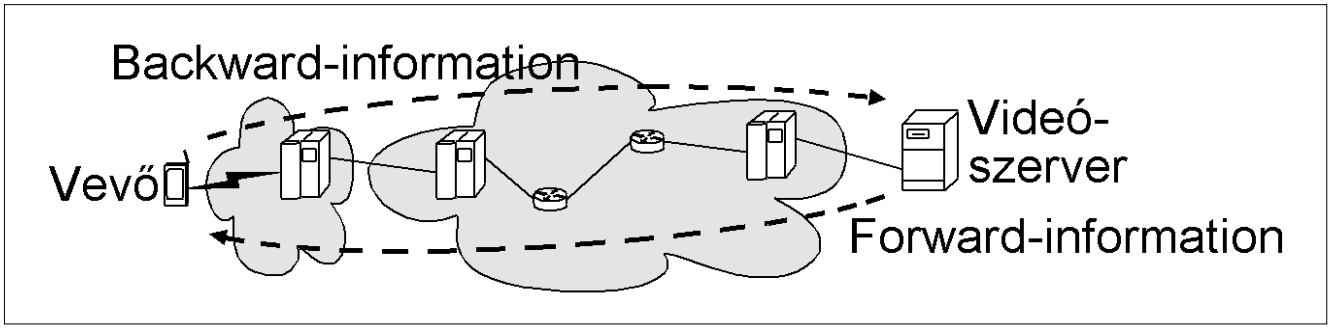
A jelzési mechanizmusok által szolgáltatott információt a hálózatban különböző helyeken használhatjuk fel. Alapvetően háromféle szabályozási kört különböztethetünk meg attól függően, hogy hol végezzük, illetve hol kezdeményezzük a szolgáltatás adaptációt.

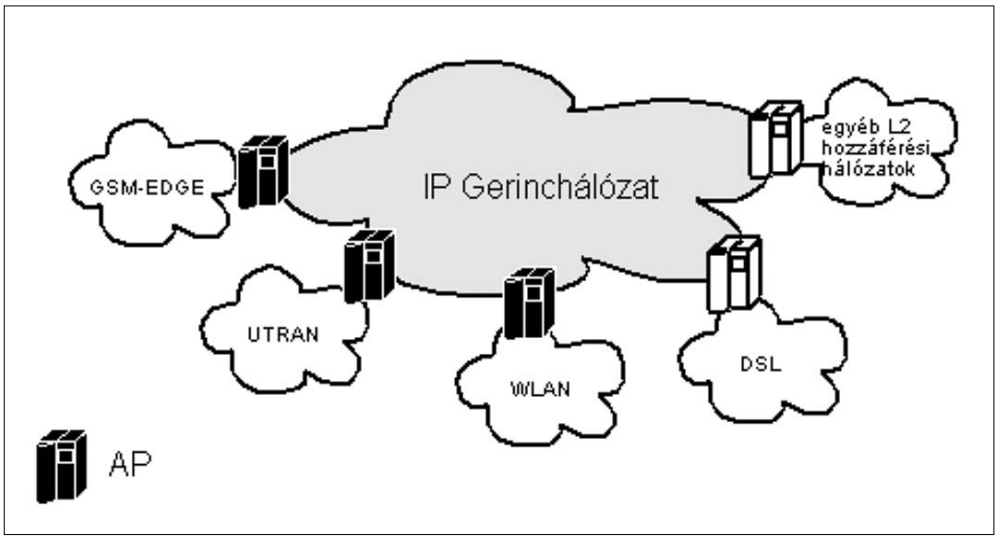
Az első típus a végpontok közötti visszacsatolás (E2E Feedback), a második a hálózati visszacsatolás (Network Feedback), a harmadik pedig, a lokális adaptáció (Local Adaptation). Ezek lehetséges helyét szemlélteti a *2. ábra* egy UMTS hálózatban.

Ezek az adaptációs mechanizmusok általános hálózati architektúrákban is azonosíthatóak. A telekommunikáció fejlesztése az „All-IP” szemléletet követi. Az IP protokoll segítségével a legkülönbözőbb átviteli technológiák összekapcsolása és ezzel egy globális telekommunikációs hálózat létrehozása valósulhat meg. Egy lehetséges hálózati architektúrára mutat példát a *3. ábra*.

Az E2E adaptáció természetesen itt is alkalmazható. A NF helye a hozzáférési hálózat elérését biztosító hálózati elem lehet, amit vezeték nélküli hálózatokban, az egyszerűség kedvéért, hozzáférési pontnak (Access Point) jelöltünk. Az LA pedig tipikusan a hozzáférési hálózatok hatékony eszköze lehet.

1. ábra
Adaptív szolgáltatásokhoz szükséges információfolyamok





3. ábra
A jövő hálózati architektúrájának egy lehetséges vázlata

Míg az E2E egy alkalmazási szintű visszacsatolás, a NF és a LA rétegek közötti információcserét igényel. E2E esetén nincs feltétlenül szükség rétegek közötti üzenetváltásokra, mivel egy ilyen megoldás szükségtelenül túlbonyolítaná a valós idejű szolgáltatásokhoz mérten amúgy is lassú üzenetváltásokat. Az alkalmazásokat futtató végpontok az alsóbb rétegek számára transzparens módon cserélnek információt. A vevő oldal figyeli a kapott minőséget, figyelembe véve többek között a sávszélességet, a puffer telítettségi szintet, csomagvesztést stb. Periodikusan, vagy ha a minőségromlás/javulás meghalad egy előre definiált szintet, üzenetet küld a forrásnak. A forrás a kapott információk alapján változtatja a küldési sebességet vagy a kódolást, a megfelelő mértékű adaptáció eléréséhez. A visszacsatoláshoz RTCP vagy RTSP üzenetek használhatók [2,3]. E2E adaptáció esetében előfordulhat, hogy a megfelelő adaptációhoz elvégzett mérések érvényességüket vesztik, mire azok az adaptálódó félhez kerülnek [7,8,9].

Röviden tehát, az E2E visszacsatolás egy alkalmazási rétegbeli jelzéseken alapuló módszer. Leginkább időben hosszabb távú adaptációhoz alkalmazható, a CL adaptációs módszerek kiegészítésére, illetve ahol a rétegek közötti kommunikáció nem megengedett.

2.1. Hálózati visszacsatolás – NF

A hálózati visszacsatolás kiküszöböli az E2E megoldás hiányosságait CL jelzések felhasználásával. A legtöbb esetben a hálózati elemek explicit információval rendelkeznek a felhasználó által érzékelhető minőséget befolyásoló hálózati körülményekről. Kiváltképp igaz ez a hozzáférési pontokra (Access Point), melyek pontosan tudják, hogy milyen változások zajlanak a vezeték nélküli csatornán: mennyi a felhasználók számára aktuálisan rendelkezésre álló sávszélesség, milyen az aktuális hiba karakterisztikája stb. Ezek alapján az AP jelzéseket tud küldeni a forrásnak, informálva azt a hálózat képességeiről.

Ilyen jelzések segítségével a folyam sávszélességét viszonylag gyorsan és pontosan a megváltozott erőforrás-viszonyokhoz lehet igazítani. Születtek már megoldások UMTS rendszerben NF alapú jelzésmegoldásra [10]. UMTS esetén a Radio Network Controller (RNC) feladata egyezik meg az említett AP-k feladatával és képességeivel. Az RNC felügyeli a rádiós csatornát, optimalizálja az adatátvitelt. Az RNC felhasználónkénti állapotokat használ, ami lehetővé teszi az összes, a vezeték nélküli csatornát használó alkalmazás nyomon követését. Ez által az RNC képes explicit jelzéseket adni az egyes felhasználók felé amikor alkalmazásaikat érintő változások történnek a rádiós interfészen. Ezeket a jelzéseket nevezik a szerzők Radio Network Feedbacknek (RNF) a [10]-es referenciában.

Általánosságban, az NF jelentős, közepesen gyors hálózati változások jelzésére alkalmas az AP-k és a forrás között, ahogy a 2. ábra is mutatta.

2.2. Lokális adaptáció – LA

Mivel a vezeték nélküli csatorna okozta gyors minőségbeli változásokat a NF típusú visszacsatolások képtelenek követni, szükség van egy gyorsabb mechanizmusra is. Ez a Lokális Adaptáció (LA), amit lokális hálózati elemek végeznek, hiszen ha a végpontokat is bevonnánk, az lelassítaná a reakcióidőt, lehetetlenné téve a csatorna gyors változásaihoz való igazodást.

A lokális adaptáció egyik eszköze, hogy átmeneti torlódás esetén a kevésbé fontos csomagokat az AP egyszerűen eldobja (Rate Adaptation). Ez a valós idejű forgalmak már korábban említett tulajdonságai miatt lehetséges. A csatorna minőségének romlásával erősebb hibavédelmet is lehet alkalmazni a visszajátszás szempontjából fontosabb csomagokon (Unequal Error Protection). Lokális adaptációnál lehet a legjobban mind a RA, mind pedig az UEP által nyújtott előnyöket kihasználni.

Létező megoldások lokális adaptációra kifinomult ütemező és buffer-menedzsment eljárásokat használnak. Ezek mellett kiemelt fontosságú az alkalmazási szintű QoS paraméterek hálózati, illetve alsóbb szintű leírókra történő leképzése [11,12,13]. Adatkapcsolati szinten bonyolult csatorna-adaptációs eljárásokra van szükség [14,15].

A jelenlegi megoldások csak az alkalmazási szintű információk igen szűk körét veszik figyelembe. Sokkal hatékonyabb működés érhetőel egy generikus megoldással, amely bővebb alkalmazási szintű információk figyelembevételét teszi lehetővé. Mivel a lokális mecha-

nizmusok nem működtethetők távolról, például a forrás által, ezért a működésük alapjául szolgáló alkalmazási szintű információkat el kell juttatni hozzájuk. Cikkünk egy ilyen megoldásra tesz javaslatot.

3. Kódolási eljárások

Lokális adaptációhoz olyan multimédia kódolásra van szükség, amely a lehető legjobban támogatja a RA-t. Bitsebesség szempontjából három kategóriába sorolhatók a kódolók. Az egyszeres bitsebességű (Single Rate) kodekek önmagukban képtelenek bármiféle adási sebesség változtatásra. Adaptáció csak a videó lejátszása közbeni kódoló-cserével lehetséges. A második típus a többszörös bitsebességű (Multi Rate) kódolók csoportja. Ezek a típusok működés közben is képesek a bitsebesség megváltoztatásra. Mivel a forrás (a kódoló) és a LA egymástól távoli hálózati elemekben található, kifinomult és gyors adaptációra nem ad lehetőséget egyik típusú kódoló sem. Az SR és MR megoldások az E2E és a NF megoldásokhoz használhatóak fel eredményesen, köztes hálózati csomópontokban azonban nem lehetséges az adaptáció.

Ahhoz, hogy a felhasználók igényeit megfelelően ki lehessen szolgálni, és az aktuális forgalmi igényekhez mérten a legjobb minőségű videó-folyamot lehessen átvinni, a legjobb megoldás a skálázható kódolók (Scalable Codecs) gyors fejlődése jelenti. Az ebbe a kategóriába tartozó kódolók előnye, hogy nem a forrás változtatja a bitsebességet, hanem olyan folyamot állít elő, mely a hálózati elemekben biztositja a skálázhatóságot. Skálázható kódolók alkalmazásával az AP a kevésbé fontos javító rétegeket eldobhatja, így érzékeny és gyors adaptáció érhető el. Ez az LA igényeinek is megfelel.

Egy ilyen kódoló szabványosítása folyik a Joint Video Team által az ITU-T keretében belül. A H.264-en alapuló MPEG-4 AVC szabvány skálázhatóvá való kiterjesztése [16,17] a cél. A H.264/MPEG4 SVC (Scalable Video Coding) lehetővé teszi, hogy bizonyos csomagok eldobásával kisebb jel-zaj viszonyú és/vagy kisebb felbontású és ezáltal kisebb sávszélesség-igényű folyamhoz jussunk. A videó keretek fontosságát és az általuk hordozott képminőséget három paraméter határozza meg.

Az első az L-Id (Layer ID). A magasabb L-Id számú keret nagyobb felbontást jelent. A második a T-Id (Temporal ID). A T-Id = 0 jelű keretek jelentik az I képkockákat, a magasabb sorszámúak jelölik a P és B kereteket, amelyek magasabb képváltási frekvenciát biztosítanak. Az utolsó paraméter a Q-Id (Quality ID), ami az egyes képkockákon belüli minőséget adja. Ha gyorsan, nagyobb mértékű sávszélesség-csökkentés szükséges, akkor a magasabb L-Id-vel rendelkező csomagokból eldobva ugyanolyan minőség mellett, kisebb felbontást kapunk. A T-Id keretekből dobva kisebb frame-rate-ű folyam kapható, ami minőségromlás nélkül, a mozgások folyamatosságának rovására csökkenti a sávszélességet. A Q-Id csomagok a legkisebb mértékű adaptáció eszközei lehetnek.

Ez az osztályozás lehetővé teszi, hogy megfelelő sorbanállási modellek alkalmazásával kontrolláltan, hirtelen nagymértékű minőségromlás nélkül alkalmazkodjon a folyam az aktuális sávszélességi lehetőségekhez.

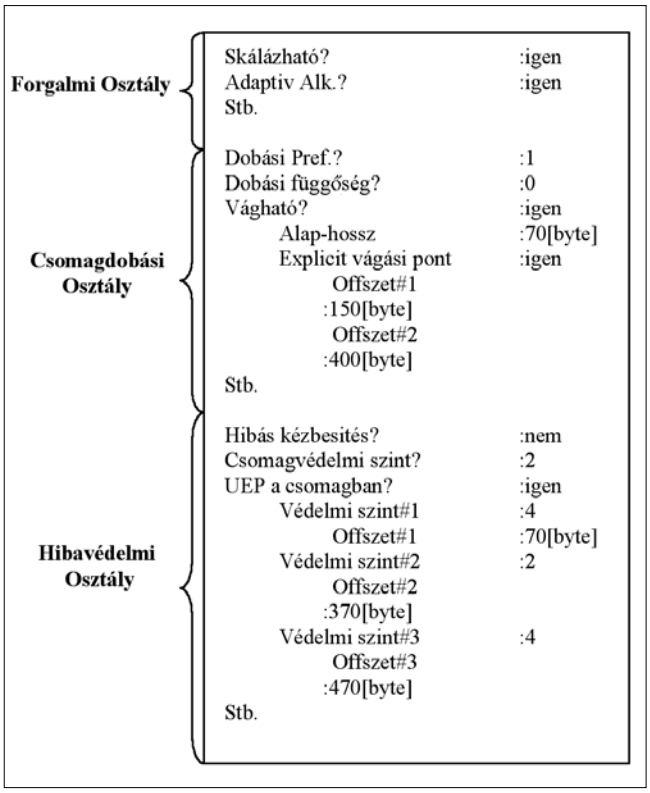
A skálázható kódolók által létrehozott folyamok dobható, és dobható vagy vágható csomagokat tartalmaznak, valamint olyanokat, amelyek a dekódolás szempontjából kiemelten fontosak: ezek nem dobhatók és nem is darabolhatók. Az előbbieknek megfelelően a csomagokhoz egy elsődleges prioritási sorrend rendelhető. Az egy védelmi szinten belüli, de különböző minőségjavítást biztosító kereteknek pedig, egy másodlagos prioritási sorrend adható. Ezeknek a fontosságoknak megfelelően kell az útvonalválasztóknak és az AP-nak kezelni az átvitt videó-kereteket, hogy a vevő oldalon megfelelő minőségű, dekódolható videót kapjunk.

Ahhoz viszont, hogy a kódoló nyújtotta előnyöket ki tudjuk használni a LA-hoz, szükséges a keretekre vonatkozó információt az AP-okhoz eljuttatni. Ez a már említett FI, amire még nem létezik általánosan elfogadott megoldási javaslat. Lehetséges eszköze lehet egy fejléc-kiterjesztés, amire egy példát mutatnak [4]-ban, vagy ennek egy általunk módosított változata, amit a következő fejezetben mutatunk be.

4. Szerver oldali információ biztosítása lokális adaptációhoz

Ahhoz, hogy a hálózati elemek hozzáférhessenek az alkalmazás szempontú csomagfontossági sorrendekhez, szükség van a FI-re. Egy hálózati pont nem tudja, hogy az egyes csomagokat hogyan kezelje a dekódoló szemszögéből. Természetesen lehetséges lenne az információ megszerzése az alkalmazásfüggő Real Time Protocol (RTP) fejlécekből is [2], de ez két ok miatt sem javasolható. Egyrészt nehézkes a rétegelt struktúrájú architektúrában a felsőbb szintű fejléc-tartalmak olvasása egy alsóbb (például a hálózati) rétegben. Másrészt pedig, az új alkalmazások megjelenésével új struktúrájú fejléc kiterjesztések jelenhetnek meg. Ez a fejrész értelmezésének módosulását jelenti, ami alsóbb rétegek működésére is kihatással lenne. Az említett problémák miatt szükség van egy általános megoldásra, ami alkalmazás-független módon segíti a Lokális Adaptációt applikációs szintű információk biztosításával.

Ahogy korábban már említettük, a skálázható kódolók sajátossága, hogy fokozatos adaptáció érhető el velük a megfelelő csomagok eldobásával vagy vágásával. Ez a valós idejű forgalom csomagvesztéssel szembeni toleranciája miatt lehetséges. A hálózati elemeknek tudnia kell tehát a kódoló által megkövetelt dobási/vágási precedenciáról, hogy torlódás esetén annak megfelelően dobja vagy vágja a torlódott csomagokat. Mindezen felül, a vezeték-nélküli csatorna előtt célsze-



4. ábra Példa RFL

rű alkalmazni a hibavédelmet is. Ez azért van így, hogy a hálózatot feleslegesen ne terheljük a védelem okozta redundanciával, és mert az AP tudja csak pontosan a csatorna jelenlegi állapotából, hogy milyen erősségű védelem szükséges a hatékony átvitelhez. Az említett UEP a rádiós csatornán igen hatékony védelmi megoldást jelent. Míg az egyenlő hibavédelem (Equal Error Protection) az egész csomagot védi, mindenhol egyformán, az UEP képes a csomagon belül, az applikációs szintű preferenciák figyelembe vételével eltérő mértékű védelmet biztosítani a különböző fontosságú részeknek.

Az általános minőségbiztosítási (Quality of Service) paraméterek mellett (adatsebesség, késleltetési korlátok, késleltetés ingadozás stb.) sokkal részletesebb, csomag-szintű információra is szükség van. Összegyűjtöttük ezért a hálózati réteg számára általunk fontosnak vélt alkalmazási szintű információkat. A fejezet további részében az ezeket leíró struktúrát ismertetjük. Bevezetjük a Réteg-független Leíró (RFL) fogalmát, mely a RA és UEP működéséhez szükséges információt tartalmazza a hálózati elemek számára. (Meg kell jegyezni, hogy bár lokális adaptációról beszélünk, NF-hez, vagy más, a hálózat többi részén előforduló adaptációs problémák kezelésére is felhasználható a RFL információtartalma.)

A RFL három fő osztályt tartalmaz: (1) Forgalmi Osztály, (2) Csomagdobási Osztály, (3) Hibavédelmi Osztály. A legfontosabb leíró mezőket a *4. ábra* mutatja.

A Forgalmi Osztály leírói jelzik, ha a forgalom skálázható illetve a végpontokban adaptív/skálázható alkalmazások helyezkednek el. A Csomagdobási Osztály mutatja a csomag dobási precedencia alapján való besorolását, illetve jelzi, ha vágható a csomag. Vághatóság esetén szükséges annak a minimális adathossznak a jelzése is, ami alatt a hordozott információ értelmét veszti, azaz aminél kisebbre nem lehet vágni. Ennek megfelelően tartalmaznia kell az összes olyan ofszetet is, mely a lehetséges vágási pontokat jelöli ki a csomagon belül.

Dobással érhető el a leggyorsabb reakció torlódás esetén, a vágás finomabb skálázhatóságot ad.

A Hibavédelmi Osztály írja le a csomagokra alkalmazható UEP, EEP típusát. Definiálható az egész csomagra (EEP) illetve csomagon belül is (UEP) a védelmi szint. Az UEP-hez meg kell adni az egyes csomag-darabok kezdeteit jelölő ofszeteket, valamint a hozzájuk tartozó védelmi szintet, ahogy azt a 4. ábra is mutatja.

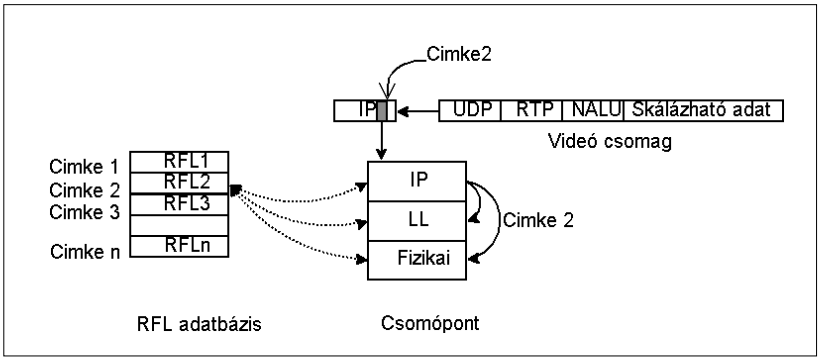
A következő fejezetben bemutatunk egy példát a RFL használatára.

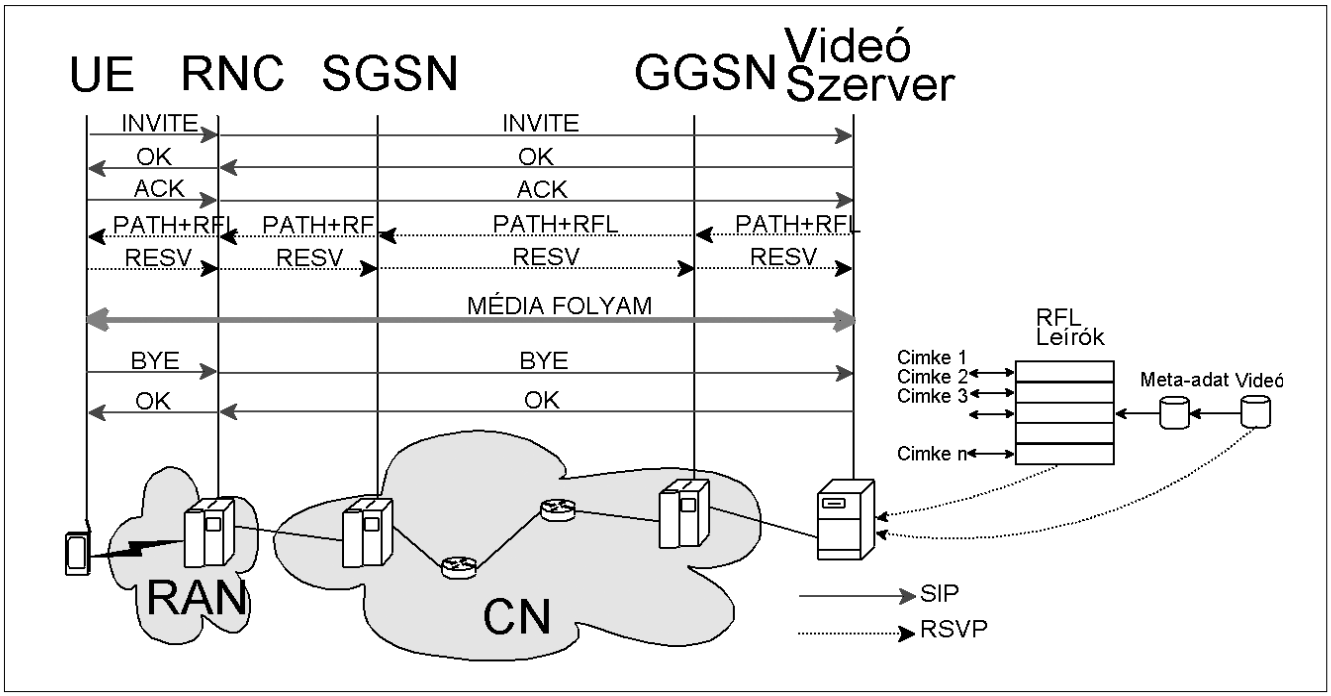
5. Kapcsolatfelépítés és a RFL terjesztése

Ahhoz, hogy a köztes hálózati csomópontok az alkalmazás által megkövetelt módon kezeljék a folyamot, minden csomagnak tartalmaznia kellene a rá vonatkozó RFL által leírt információt. Mivel ez egy elég robosztus megoldást jelentene, és egyúttal jelentős többletinformációt is, egy sokkal dinamikusabb módszerre van szükség. A RFL-eket elég csak a viszony kezdetén terjeszteni az adaptációban résztvevő csomópontok között, a csomópontok elraktározzák azokat a hozzájuk rendelt azonosítóikkal együtt. A videófolyam adása alatt az egyes RFL-ekhez rendelt azonosítókat kapják csak meg a csomagok, ezzel jelezve az útvonalválasztóknak és az AP-nak, hogy milyen forgalmi precedenciájú osztályba taroznak. A RFL-eket azonosító számok, mint Címkék (Label) átvihetők például az IP fejlécben, vagy egy kiegészítő fejlécben, vagy akár DiffServ CP-ben [18,12].

A hálózati csomópontokban a rétegek közötti kommunikáció az úgynevezett bróker-alapú megoldáson működhet. A bróker egy olyan egység, mely központosítottan tárol információkat, esetünkben a RFL-eket. A

5. ábra Bróker alapú Címke-RFL összerendelés





6. ábra Példa üzenetváltás

kommunikációban résztvevő többi elem ezeket rajta keresztül éri el. Ez a megoldás lehetővé teszi, hogy minden hálózati csomópont egyetlen adatbázist használjon és egyszerű adatbázis-hozzáférést biztosítson az összes réteg számára. Ezek alapján tehát egy csomópont IP rétege a kapott csomagból lekérdezi a megfelelő RFL-hez tartozó paramétereket, majd az így kinyert információnak megfelelően rendelkezik a csomaggal.

Ha – például AP esetében – szükség van a hálózati réteg alatt is a RFL-ek tartalmára, akkor az IP réteg egy előre definiált CL interfészen keresztül átadja a címkét, ami alapján az alsóbb, LL réteg is hozzáférhet a leírókhoz a brókeren keresztül. Így még hatékonyabb működést lehet elérni, mintha a RFL-eket rétegekhez kötve használnánk, hiszen az amúgy is szükséges címke segítségével a leírók halmazát rétegfüggetlenné tettük. A bróker alapú hozzáférést mutatja az *5. ábra*.

A következőkben egy példával illusztráljuk, hogyan terjeszthetők FI keretében a folyamra jellemző RFL-ek, és azok hogyan és hol alkalmazhatók a hálózatban adaptációs célokra.

Példánkban a felhasználó egy videó-szerverhez csatlakozik, ahonnan egy filmet igényel. A megfelelő kodek és annak paraméterei, valamint az egyéb képességegyeztetések SIP/RTSP [19,3] protokoll segítségével történik. Természetesen az RNC-n SIP proxy-nak kell futnia ahhoz, hogy a hívásfelépítés létrejöhessen. Az említett információkon felül, az általunk bevezetett és a kiválasztott filmhez tartozó RFL-ek a hozzájuk rendelt címkékkel együtt szintén szétterjesztésre kerülnek. Célszerű erőforrás-foglaló protokollt használni erre a célra. A feladatra alkalmas lehet az RSVP [20], ahol a PATH üzenetek vihetik át a videóra vonatkozó információkat. Ahhoz, hogy az RNC-n (SGSN, GGSN, MGW) adaptációt alkalmazhassunk, futnia kell az RSVP protokollnak. Az RSVP alkalmazása csomópontonkénti CAC-t (Call Admission Control) biztosít, és támogatja az alkalmazások által definiált objektumok átvitelét [21]. RSVP nélkül is megoldható a RFL-ek terjesztése. Ha SIP protokollt használunk a hívásfelépítéshez, akkor az OK üzenetben az SDP protokoll kiegészítéseként lehet elküldeni a RFL-eket.

A RFL-ek összeállítása a médiából kinyert metaadatok alapján történik. Például egy skálázható H.264/AVC kódolót használó alkalmazás által generált videó-keretek az úgynevezett NALU (Network Abstraction Layer Unit) fejléccel kezdődnek. Ezek a fejlécek hordozzák a prioritásra vonatkozó információt. Mivel a NALU struktúrája állandó, a keretre vonatkozó metaadat előállítására közvetlenül alkalmas, melyből végül egyszerűen fordítással kaphatjuk a videó-kerethez tartozó RFL-et. A RFL-ek terjesztése és a meta-adatok összeállítását mutatja a *6. ábra*.

A videó adás megkezdésével a kódoló csak az adatcsomaghoz rendelt megfelelő címkét adja át egy interfészen az IP-rétegnek. A csomag fontosságát és egyéb tulajdonságait ezek után egész útján ez az egy azonosító határozza meg: ez alapján működnek a csomópontokban a RA és UEP mechanizmusai. Például, ha torlódás van a hálózatban, vagy egyszerűen a börsztös videó-forgalom miatt, a várakozási sor hossza hirtelen megnő, akkor az IP szinten alkalmazhatjuk a RFL biztosította információkat hogy a kisebb dobási precedenciáju csomagokat dobjuk el a torlódás kezelésekor. Így lehetséges az, hogy a vevő a torlódás miatt csak kismértékű minőségromlást észlel. A RFL Hibavédelmi Osztályának információit az adatkapcsolati vagy fizikai réteg is felhasználhatja az AP-ban. Segítségükkel a csatorna jobb kihasználtsága érdekében más modulációs technikára válthat.

6. Összefoglalás

Vezeték nélküli hálózatokat egyre szélesebb körben és feladatokra használunk. A multimédiás szolgáltatások iránti igény dinamikus növekedésével számos kihívásnak kell megfelelnie a jövő kommunikációs hálózatainak. Jelentős mértékű szolgáltatás-minőség javulás érhető el, ha a szigorú rétegstruktúrát fellazítva kihasználjuk a rétegek közötti kommunikáció adta előnyöket. A multimédia tartalmú forgalom tulajdonságaiból adódóan a megfelelő minőségű videó-átvitelhez szükség is van CL információkra.

Cikkünkben a valós idejű videó forgalom esetében alkalmazható adaptációs lehetőségeket világítottuk meg. Irányelvet adtunk, hogyan javítható az alkalmazott technikák teljesítőképessége az átvitt videófolyam minősége szempontjából, ha alkalmazási szintű, rétegfüggetlen leirókat használunk. Összegyűjtöttük a fontosabb, RFL-ekhez tartozó információkat, melyek a RA es UEP, mint a lokális adaptációs eszközök működéséhez szükségesek. Definiáltunk egy lehetséges RFL formátumot, majd javaslatot tettünk alkalmazására. Végül konkrét példán keresztül bemutattuk a javasolt módszer működését, ismételten rávilágítva alkalmazásának lehetőségeire.

A cikkben a rétegek közötti komunikáció egy megvalósítási lehetőségét vázoltuk fel. A bemutatott módszer részletes kidolgozása, pontosítása jövőbeni kutatási témánk.

Irodalom

[1] Johan Sjoberg, Magnus Westerlund, Ari Lakaniemi, Stephan Wenger: RTP Payload Format for Extended AMR Wideband (AMR-WB+) Audio Codec. DRAFT, Februar 14, 2005.

[2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson: RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003.

[3] H. Schulzrinne, A. Rao, R. Lanphier: Real Time Streaming Protocol (RTSP). RFC 2326 (Proposed Standard), April 1998.

[4] L. Larzon, U. Bodin, O. Schelen: Hints and notifications. In: IEEE Wireless Communications and Networking Conference (WCNC), Orlando, Florida, USA, 2002.

[5] Q. Wang, M. A. Abu-Rgheff: Cross-layer signalling for next-generation wireless systems. In: IEEE Wireless Communications and Networking Conf. (WCNC), New Orleans, Louisiana, USA, 2003. Nr.1., pp.1084–1089.

[6] V. Kawadia, P. R. Kumar: A cautionary perspective on cross layer design. IEEE Wireless Communication Magazine, July 2003.

[7] P. M. Ruiz, E. García: Adaptive multimedia applications to improve user-perceived QoS in multihop wireless ad hoc networks. In: IEEE International Conf. on Wireless LANs and Home Networks (ICWLHN), Atlanta, USA, Aug. 2002, pp.673–684.

[8] J. Widmer, R. Denda, M. Mauve: A survey on TCP-friendly congestion control. IEEE Network, 15(3):28–37, 2001.

[9] J. W. Byers, G. Horn, M. Luby, M. Mitzenmacher, W. Shaver: FLID-DL: Congestion control for layered multicast. October 2002.

[10] S. Chemiakina, L. D'Antonio, F. Forti, R. Lalli, J. Petersson, A. Terzani: QoS enhancement for adaptive streaming services over WCDMA. IEEE Journal on Selected Areas in Communications, 21(10), December 2003. Recent Advances In Wireless Multimedia.

[11] H. Zhu, M. Li, I. Chlamtac, B. Prabhakaran: A survey of quality of service in IEEE 802.11 networks. IEEE Wireless Communications, pp.6–14, August 2004.

[12] W. Kumwilaisak, Y. T. Hou, Q. Zhang, W. Zhu, C.-C. Kuo, Y.-Q. Zhang: A cross-layer quality-of-service mapping architecture for video delivery in wireless networks. IEEE Journal on Selected Areas in Communications, 21(10):1685–1697, December 2003.

[13] A. Gurtov, R. Ludwig: Lifetime packet discard for efficient real-time transport over cellular links. (4):32–45, October 2003.

[14] R. Ludwig, A. Konrad, A. D. Joseph: Optimizing the end-to-end performance of reliable flows over wireless links. In: Proceedings of ACM/IEEE MobiCom'99, 1999.

[15] R. Ludwig, B. Rathonyi: Link layer enhancements for TCP/IP over GSM. In: Proceedings of IEEE INFOCOM'99, 1999.

[16] H. Schwarz, D. Marpe, T.Wiegand: MCTF and scalability extension of H.264/AVC. In: Proc. of PCS, San Francisco, USA, Dec. 2004.

[17] H. Schwarz, D. Marpe, T. Wiegand: Combined scalability extension of H.264/AVC, 2005.

[18] J. Shin, J. Kim, C.-C. J. Kuo: Quality-of-service mapping mechanism for packet video in differentiated services network. IEEE Transactions on Multimedia, 3(2), June 2001.

[19] R. Mahy, B. Biggs, R. Dean: The Session Initiation Protocol (SIP) "Replaces" Header. RFC 3891 (Proposed Standard), September 2004.

[20] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin: Resource ReSerVation Protocol (RSVP) – Version 1: Functional Specification. RFC 2205 (Proposed Standard), September 1997. Updated by RFCs 2750, 3936.

[21] J. Wroclawski: The use of RSVP with IETF Integrated Services. RFC 2210, September 1997.

Valós idejű beszélőnormalizációs eljárás és alkalmazása a „Beszédmester” beszédjavítás-terápiai rendszerben

KOCSOR ANDRÁS

Szegedi Tudományegyetem, MTA-SZTE Mesterséges Intelligencia Kutatócsoport
kocsor@inf.u-szeged.hu

Kulcsszavak: beszédfelismerés, beszélőnormalizáció, beszédjavítás

A különféle beszélőnormalizálási technikák alkalmazása jelentősen javíthatja a beszédfelismerés pontosságát. Ebbe a módszercsaládba tartoznak azok az eljárások is, amelyek az artikulációs csatorna hossznormalizálására (VTLN) törekednek. A kutatások tanúbizonysága szerint ezek a módszerek jól alkalmazhatók, amikor a beszédfelismerő rendszernek megbízhatóan kell működnie férfi, nő és gyermek beszélők esetén is. Elkészítettünk egy számítógéppel segített beszédjavítás-terápiára és olvasásfejlesztésre alkalmas eszközt; a Beszédmestert. A cikk kettős céllal készült. Egyrészt szeretnénk röviden bemutatni a Beszédmester szoftvert, rámutatni újdonságértékére, illetve betekintést adni a vele elért eredményekbe. Másrészt pedig a Beszédmester hátterében nyugvó beszédtechnológiai módszerek közül szertnénk ismertetni egy újszerű valós idejű VTLN eljárást. Nevezetesen megvizsgáljuk, hogy az irodalomból ismert lineáris diszkrimináns alapú VTLN modell, felépítés után, hogyan közelíthető valós idejű kiértékelést biztosító regressziós neuronhálózattal.

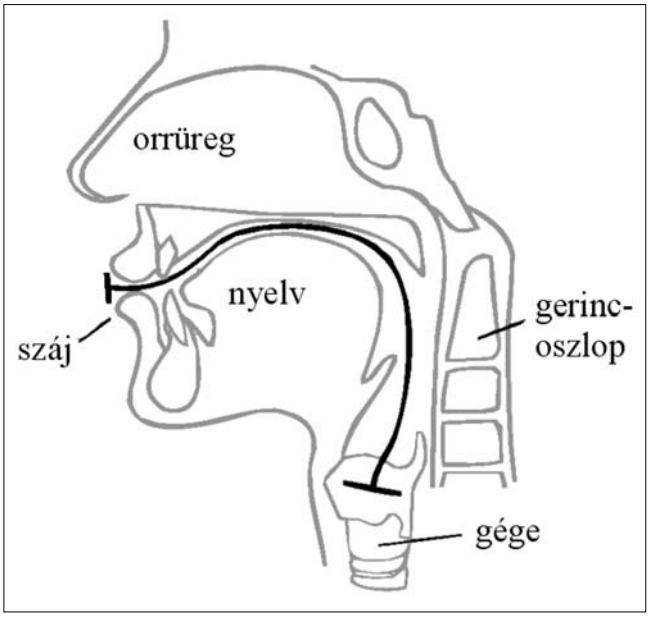
1. Bevezetés

A beszédfelismerési alkalmazások széleskörű elterjedésének előfeltétele a rendszerek beszélőfüggetlensége, azaz a különböző felhasználók esetén is megbízhatóan és pontosan működő felismerési mechanizmus. A rendszerek általánosítási képessége természetes módon fokozható a tanítási folyamat során felhasznált beszédadatbázisok nagy méretével és sokszínűségével, de ez hosszadalmas és költséges munkát követel meg a módszer korlátozott hatékonysága mellett. Egy másik megközelítés ezért az egyes beszélők hangját változtatja meg úgy, hogy az a lehető legjobban hasonlítson az alkalmazott beszédfelismerő rendszer „ideálisan” felismerhető hangjához.

A beszélők hangjai közti eltéréseket a garat-, száj- és orrüreg alkotta artikulációs csatorna *(1. ábra)* egyedekre jellemző különbségei okozzák. Ez a csatorna a gégéből kibocsátott hanghullámot, mint áramló levegőt közvetíti a külvilág felé. Működése során a kiinduló hang bizonyos frekvenciáit felerősíti, másokat elnyom, de a jelenlévő frekvenciák szerkezetét nem módosítja. Emiatt a csatorna jelfeldolgozási szempontból szűrőként viselkedik, azaz matematikailag egy átviteli függvénnyel jellemezhető (megadja az egyes frekvenciák erősítésének mértékét). Az átviteli függvény maximumhelyeit formánsoknak nevezzük. Fonetikai kísérletek alapján kijelenthető, hogy a hang első három formánsa a fonetikai minőséget, míg a többi a beszélő egyedi jellemzőit (például hangszín) határozza meg. Beszédfelismerés szempontjából az itt jelentkező különbségeket célszerű kompenzálni, létrehozva így egy „idealizált” beszédhangot.

A különböző beszélők által ejtett, de azonos fonetikai csoportba tartozó hangok formánsszerkezete hasonló, a frekvenciaértékek egymáshoz képest csak kis mértékben elcsúsztatottak. Minden fonetikai csoport jellemezhető egy átlagos formánsszerkezettel. A legelterjedtebb megközelítés szerint az egyes beszélők formánsainak átlagostól vett eltérése korrelál az artikulációs csatorna hosszával (Vocal Tract Length – VTL), ezért lehetőség nyílik a hangok normalizálására a beszélőre jellemző VTL érték szerint. Ezt a módszert a szakirodalom Vocal Tract Length Normalization-nek (VTLN) nevezi. Ennek során egy transzformációs függvény segítségével próbáljuk a beszélők formánsait az átlagos helyre leképezni [4,13]. A beszélőnormalizálás feladata egy megfelelő típusú transzformációs függvény kiválasztása, és a beszélőre jellemző optimális paraméterek beállítása.

1. ábra Az emberi artikulációs csatorna



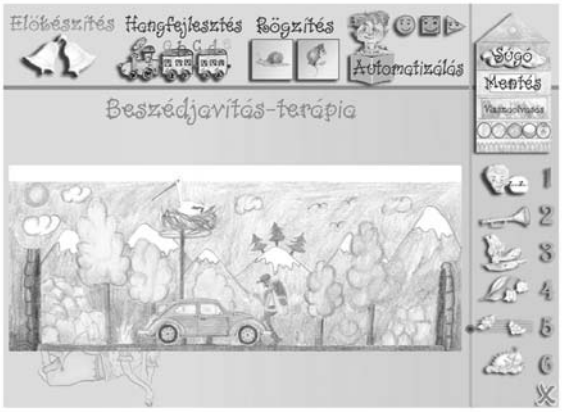
A VTLN során a transzformációs paraméterek beállítása történhet néhány másodperces bemondás alapján, vagy a teljes hanganyag segítségével is. Mindkét esetben számottevően csökkenthető a fonémafelismerés hibája.

Ezeknek a módszereknek az eredeti formában történő alkalmazására azonban nincs lehetőség az általunk kifejlesztett „Beszédmester” szoftver [6-11] működése során, hiszen ebben az esetben valós idejű fonémafelismerésre, és így valós időben elvégezhető normalizációs technikára van szükség. A probléma megoldására egy valós idejű VTLN eljárást ismertetünk, amely a bemondás pillanatában hatékonyan megbecsüli az alkalmazott transzformációs függvény paramétereit [10].

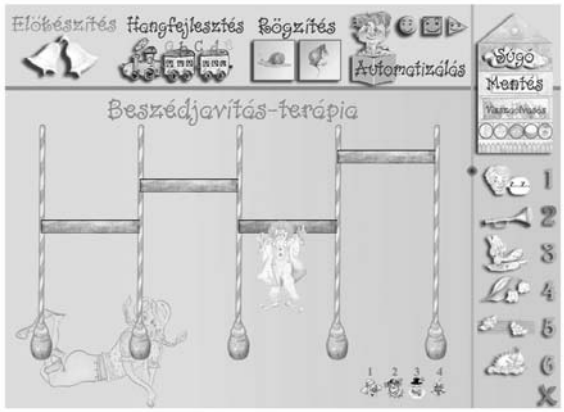
Jelen publikáció felépítése a következő. A második fejezetben, röviden bemutatjuk a „Beszédmester” szoftvert, amely funkcionalitásából adódóan felveti a valós idejű beszálőnormalizáció kérdését. A harmadikban ismertetjük a neuronháló-alapú normalizációs megoldásunkat, amit a módszer teszteléséről és a teszteredmények kiértékeléséről szóló negyedik fejezet követ. Végül levonjuk a cikk tanulságait és felvázoljuk a továbbfejlesztés lehetőségeit.

2. ábra Képek a beszédjavítás-terápia részből

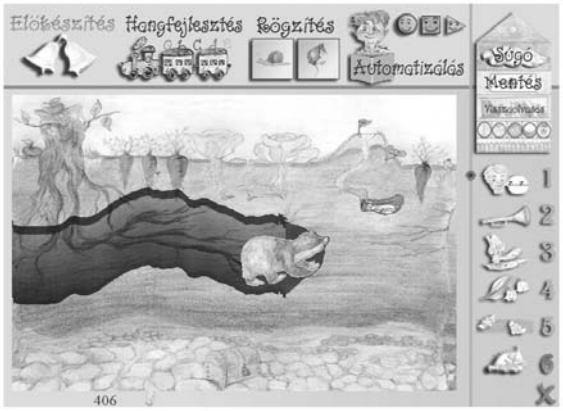
(A) zöngeadás gyakorlása



(B) hangerő változtatás berögzítése

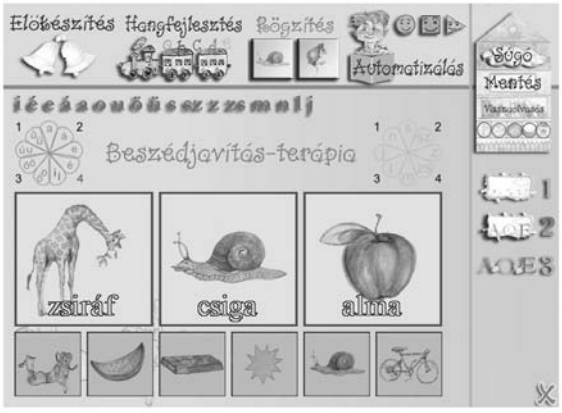


(C) játék a hangmagasság finom változtatásával

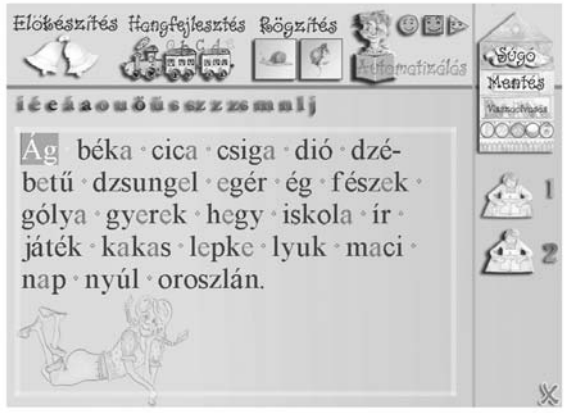


(D) hangfejlesztés valós idejű visszacsatolással

(E) hangok rögzítése szavakban



(F) hangok rögzítése mondatokban



2. A „Beszédmester”

A Beszédmester [6-11], egy komplex beszédjavítás-terápiai és olvasásfejlesztő számítógépes szoftver, amely ingyenesen letölthető a következő weboldalról:

http://www.inf.u-szeged.hu/beszedmester.

2.1. Beszédmester a beszédjavítás-terápiában

Óvodás és kisiskolás életkorban nagyon gyakori jelenség, hogy gyermekeink beszédében zavaróan súlyos hanghibák vannak, sőt előfordul az is, hogy alig, vagy egyáltalán nem értjük őket. Sok gyermeknél műtéti úton állítják elő a szinte teljes értékű, természetes minőségű hallást; ők a cochlea implantáltak. Számukra több éves folyamat a hallás olyan szintű „megtanulása”, ami a hangos beszéd elsajátításához szükséges. Vannak gyermekek, akik ép hallásuk mellett is súlyosan beszédhibásak. Ha szervi okok (szájpadhasadék stb.) akadályozzák a szépen hangzó beszéd kialakulását, akkor a műtét utáni rehabilitációban logopédus szakember segítségével megoldást lehet találni. Ép hallás és beszédszervek esetén is késhet a beszéd megindulása, vagy olyan sok hangzóhibát véthet a gyermek, ami zavaró, alig érthető kiejtéshez vezet. A Beszédmester szoftver *(2. ábra)* az érthető beszédet döntően befolyásoló magánhangzók felismerését végzi el az elhangzás pillanatában (2/D. ábra), ezzel egy olyan új eszközt adva a szakemberek kezébe, amellyel jelentősen lerövidülhet a remélt, kívánatos színvonalú beszédállapot eléréséhez szükséges gyakran több évi munka [3,5].

A Beszédmester szurdopedagógiai felhasználását, kipróbálását a program fejlesztésének kezdeti szakaszától a Kaposvári Siketek Iskolájában végezték. A kipróbálás eredményeit egyrészt az artikulációfejlesztő terápiát irányító szakemberek (szurdopedagógusok, szurdologopédusok, logopédusok), másrészt a terápiában részt vevő gyermekek megjegyzéseiből, kérdéseikből szűrték le, illetve hasznos kiegészítésül szolgáltak a tesztelés külső megfigyelőinek feljegyzései is. Ezek a megfigyelések adták a programot használó gyermekek metakommunikativ jelzéseinek legátfogóbb képet.

A foglalkozást követő elemzések során nagyon fontosak voltak a gyermekek reakcióiról összegyűjtött feljegyzések. Az olyan apróságnak tűnő megjegyzések, mint: *„...felcsillanó szemmel, mosolyogva nézett a monitorra...”, „...tapsikolt örömében, amikor meglátta a ...”*, pozitív minősítését adták az adott programrész felépítésének, működésének. A Beszédmester szoftver kipróbálását súlyos fokban hallássérült (siket) gyermekek kiejtésjavítására, nagyothalló és cochlea implantált gyermekek beszédérthetőségének fejlesztésére, és beszédfogyatékos gyermekek artikuláció fejlesztésére terápiás foglalkozásokon végezték. A nem beszelő siket gyermekek kiejtésjavítását öt hónapon keresztül végezték. A gyermekek magánhangzóinak ejtését a terápia előtt és után vizsgálták. Az eredmények azt mutatták, hogy a terápia előtt 1-3 magánhangzót a fejlesztés hatására pedig már 4-7 magánhangzót ejtettek helyesen a gyermekek.

A látványos növekedésben a Beszédmester szerepét bizonyítja, hogy valamennyi gyermek 1-2 éves szurdopedagógiai fejlesztésben vett részt, tehát a terápia előtti számadat 1-2 év kiejtésnevelésének a hozadéka. Már beszélő siket gyermekek és a logopédiai fejlesztés során az ajak- és szájpadhasadékos gyermekek „gépi” beszédjavító terápiája is látványos eredményt hozott a magánhangzók tisztaságában. A Beszédmesterrel „tanuló” gyermekek magánhangzói teljes mértékben orrhangzósságtól színezetmentessé váltak, és sokkal tisztábbak lettek.

A Beszédmester hatásfoknövelő szerepét a nagyothalló és a cochlea implantált gyermekek beszédérhetőségének fejlesztésében is vizsgálták. Mindkettő esetében a Beszédmester alkalmazásának legnagyobb hozadéka, hogy az állandó értékelés, visszajelzés (a számítógép automatikusan végzi el!) mellett tudják saját kiejtésüket javítani. A színes, motiváló, mozgó grafika azokból a gyermekekből váltott ki újra és újra hangos megnyilatkozást, akik eddig ritkán hallatták hangjukat. Az alacsony életkorú, figyelemzavaros gyermekeket tartós figyelmi helyzetbe hozta a változatos gyakorlási módok sokasága.

2.2. Beszédmester az olvasásfejlesztésben

A Beszédmester szoftver az olvasástanítás segítését, az olvasás terápiáját, fejlesztését is célul tűzte ki. A szoftver segítségével játékos úton, szinte észrevétlenül lehet gyakoroltatni az olvasást *(3. ábra)*. Használható az iskolai olvasástanítás során és egyéni gyakorlásra. A részképességükben sérült gyermekek fejlesztő terápiájában komplex készségfejlesztést biztosít: memória- és figyelemfejlesztés, irányfelismerés, iránytartás kialakítása, finommotorika fejlesztése, hallási diszkriminációs készség, hallási figyelem, vizuális differenciáló képesség fejlesztése [15]. Segítheti a diszlexiaterápiát, hiszen a betűk újratanításának feltételei biztosítottak, a fonémapárok feladatai pedig célzottan a kritikus párok gyakorlására, differenciálására készültek (3/F. ábra). A teszteredmények azt mutatják, hogy nemcsak az első osztályosok fejlesztésére, tanítására alkalmas a szoftver, hanem a 8-10 évesek terápiájában is kiválóan használható.

Az olvasásfejlesztés modul célja eredetileg egy programfüggetlen olvasástanítási szoftver előállítása volt, azonban az előzetes tesztek azt bizonyították, hogy az olvasási nehézségekkel küzdő gyermekek munkáját is hatékonyan segíti a Beszédmester. A munka eredményességét mutatja, hogy a gyermekek szívesen és hosszan dolgoznak a szoftverrel, s a feladatok megoldása segíti a fonológiai tudatosság kialakulását. A részképességükben sérült gyermekek, de még egészséges társaik is sokszor nehezen tájékozódnak a tankönyv feladatai között, figyelmüket könnyen elterelik a színes ábrák. A Beszédmester maximálisan biztosítja az egyéni tempóban való haladást. Az egyéni irányított munkaformában a gyermekek szívesen fogadták a mikrofonnal kiegészített szoftvert. Az egészséges óvodás és iskoláskorú gyermekek is könnyen dolgoztak vele. A magánhangzó-felismerése egyszerűbb feladatnak bi-

zonyult. A szófelismerés során azt tapasztaltuk, hogy a gyerekek egészen addig próbálkoznak az adott szó helyes kiejtésével, amíg fel nem villan a hívókép írásos változata, a szókép. Csoportos, irányított munkaformában a program hang- és szófelismerő része alkalmat ad a tehetséggondozásra és a felzárkóztatásra.

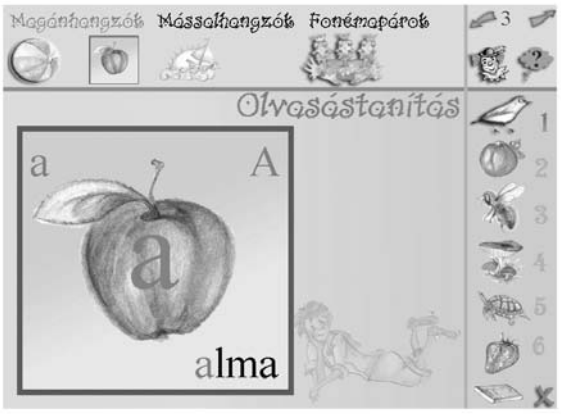
2.3. Beszédtechnológia a Beszédmesterben

A Beszédmester program, különösen a beszédorientált részeket tekintve, a számítógéppel segített oktatás területén innovatív jelentőségű, hiszen az interakció a beszédinterfész által a számítógép és a felhasználó között még emberibbé válik. A tanulás/terápia a tanuló/sérült gyermek és a számítógép manipulatív, „barátságos” interakciója alapján valósulhat meg.

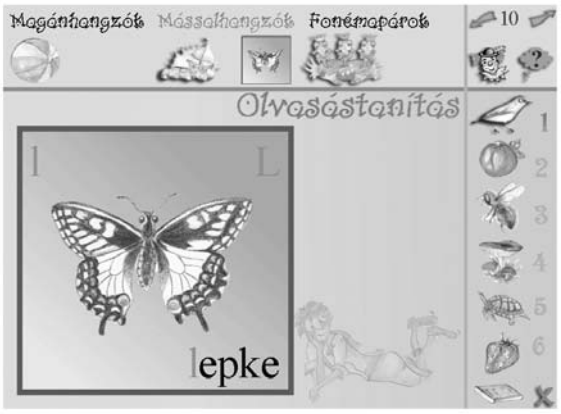
A szoftver kulcseleme a beszédtechnológiai modul, amely lehetővé teszi, hogy a rendszer a mikrofonba bemondott beszédhangokat, illetve szavakat valós időben visszajelezze. A Beszédmesterben alkalmazott technológia segítségével nem a hang oszcillogramja, vagy spektruma jelenik meg a gyakorlás során, hanem maga a hozzá kapcsolt betű jele tűnik fel.

3. ábra Képek az olvasásterápia részből

(A) magánhangzó tanítás valós idejű fonémafelismeréssel

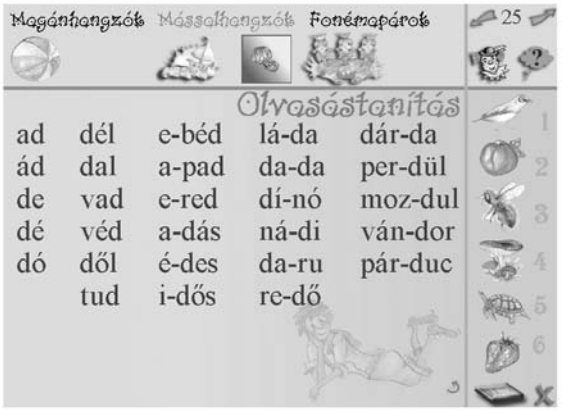


(B) mássalhangzó tanítás gépi szófelismeréssel



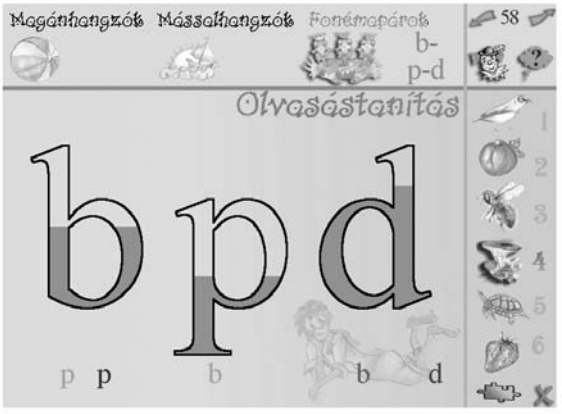
(C) fonémák helyének keresése

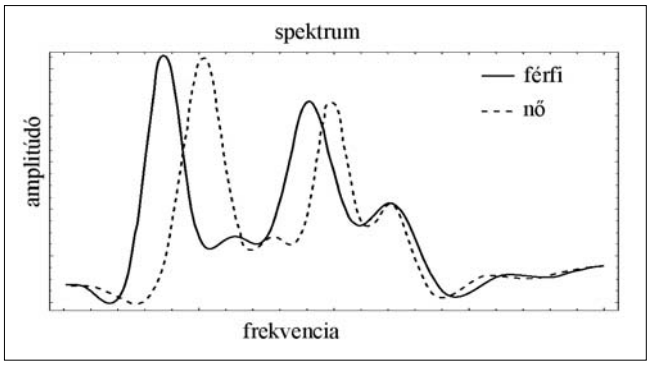
(D) olvasási gyakorlat



(E) kritikus párok újratanítása gépi szófelismeréssel

(F) betűfeltöltés





4. ábra
Az artikulációs csatorna hosszának frekvenciaeltoló hatása. Az ábrán egy férfi és egy nő által bemondott magánhangzó spektruma látható egy adott pillanatban.

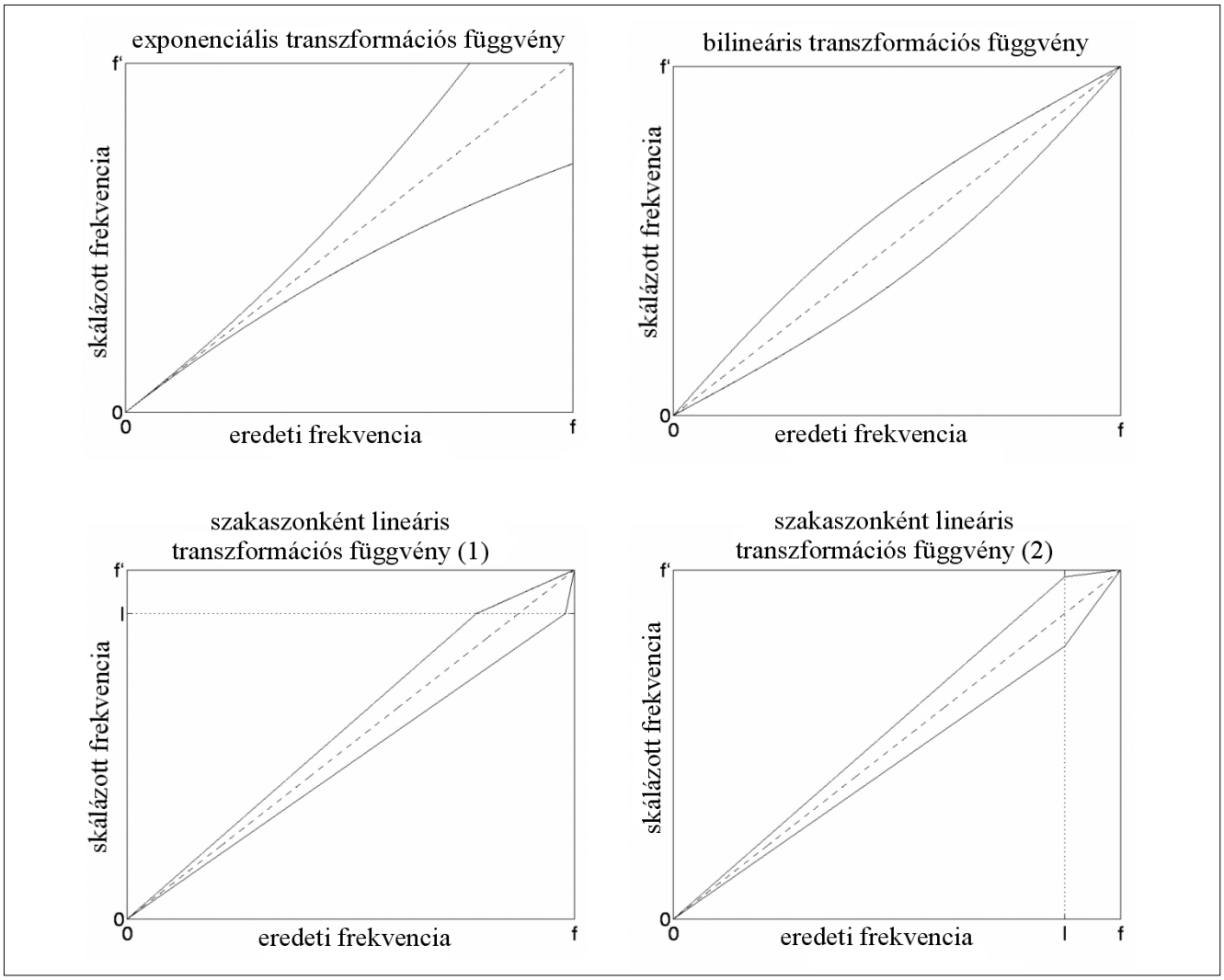
A vizuális kijelzés az elhangzással azonos időben történik. Mind a beszéd analízise, mind a beszédhangok azonos idejű és fonémaszintű feldolgozása önműködően megy végbe. Tehát ez a megoldás már nem a mikrofon előtt közvetlenül elhangzó egyedi beszédhangot jeleníti meg, hanem azt, amit „megért" a rendszer, azaz a hanghoz társított betűképet.

3. Beszélőnormalizáció az artikulációs csatorna modellezésével

Az emberi populáció egyedeinek beszédkarakterisztikája nagy különbözőséget mutat. Ennek egyik fő fiziológiai forrása a beszélők artikulációs csatornájának a hossza. A férfiak esetében az átlagos hossz 17, a nőknél 15, míg a gyermekeknél 14 cm [2]. Az artikulációs csatorna voltaképpen a beszélő formánsait eltolja a frekvencia tengelyen az adott fonetikai csoportra jellemző átlagtól [13]. Az eltolás azonban nem lineáris, az alsó és a felső frekvencia tartományokban különböző mértékű lehet. Az eltolási jelenség első fizikai leírása a Helmholtz rezonátor volt, amely az artikulációs csatornát egy csővel modellezte, így az eltolási mértékre exponenciális összefüggés adódott. A későbbi, finomabb modellek már figyelembe vették azt a tényt, hogy a magasabb frekvencia tartományokban csökken az eltolás mértéke, így keletkezett a bilineáris és a szakaszonként lineáris transzformációs függvény [12].

Az alkalmazott transzformációs függvények *(5. ábra)* általában kevés paraméterrel definiálhatóak, ez megkönnyíti mind az egyedre jellemző optimális paraméter meg-

5. ábra Lehetséges transzformációs függvények.
Az ábrák az eredeti f és a transzformált f' frekvenciaértékek közötti kapcsolatot mutatják.



határozását, mind a paraméteres modell kiértékelését: az exponenciális, a bilineáris és a szakaszonként lineáris függvények mindegyike egyetlen paraméterrel írható le.

Miután kiválasztottunk egy transzformációs függvénytípust, a beszédadatbázisban szereplő beszélőkre egyénenként meghatározzuk az optimális paraméter értéket egy iteratív eljárás segítségével. Bár a transzformációs függvény paraméterértéke folytonos, mégis célszerű a lehetőségeket egy kis elemszámú, véges halmazra korlátozni (általában 10-20 érték). Az egyedre jellemző optimális paraméterérték kiválasztásának stratégiája és az optimalizálás kritériuma módszerről módszerre változik [4,16-18]). Jelen dolgozat nem a létező módszerek összehasonlítására összpontosít, helyette vizsgálódásainkat csak a [17]-ben említett lineáris diszkrimináns (LD) alapú módszerre szűkítjük.

Ez az eljárás ugyanolyan hatékonynak, de stabilabbnak bizonyult, mint a legelterjettebb Maximum Likelihood módszer [4]. A Maximum Likelihood (ML) alapú paraméteroptimalizáció kezdetben minden beszélőhöz azt a paraméterértéket rendeli, amely nem változtatja meg a formánsait. Az így betanított beszédfelismerőt felhasználva, a lehetséges paraméterértékek egyénenkénti kipróbálásával minden beszélőre meghatározható az optimális érték, amely mellett a legnagyobb valószínűséggel ismerhetők fel a mintái. Ezután, az így meghatározott egyedi paraméterek felhasználásával transzformált beszédadatbázis mintáin újra betanítjuk a beszédfelismerő algoritmust. Az iterációt addig folytatjuk, amíg a változás meghalad egy bizonyos minimális mértéket.

A Lineáris Diszkrimináns Analízis (LDA) alapú LD-VTLN módszer [17] szintén iteratív eljárás, amely az iteráció során az előző módszertől eltérően nem használ beszédfelismerő eljárást. A módszer az egyénenként megadott paraméterhalmaz kiértékelését a lineáris diszkrimináns (LD) kiszámításával végzi. Az LD érték meghatározása az egyénenként különböző módon transzformált mintákat tartalmazó beszédadatbázis alapján történik. A beszédadatbázisban a minták címkézettek, egy-egy fonetikai szimbólum és az adatközlő sorszáma, illetve adatai vannak hozzárendelve. A fonetikai szimbólumok alkotják az osztályokat, amelyek minnél pontosabb megkülönböztetése a cél ismeretlen mintára.

Először meghatározunk egy B mátrixot, amely az egyes osztályok mintái átlagának szórását reprezentálja. Majd előállítjuk a W mátrixot, ami az osztályokon belüli minták szórásának átlagára jellemző. Az LD értéke a két mátrix determinánsának hányadosa: LD=|B|/|W|. Ez az érték nagy, ha az egyes osztályok elemei kis szórást mutatnak, miközben a különböző osztályok átlagai távol esnek egymástól. Az LD érték növekedésével voltaképpen egyre nagyobb osztály-szeparabilitást (fonémaszeparációt) érhetünk el.

Az iteratív LD-VTLN algoritmus pszeudókódja a következő:

1. Válasszunk minden beszélőhöz egy iniciális paraméterértéket és végezzük el az általa bemondott minták frekvencia-transzformációját.
2. Minden beszélőre számoljuk ki az LD értéket úgy, hogy a hozzá társított egyedi paraméterértéket kicsit megnöveljük, illetve csökkentjük. Ezután válasszuk ki azt a paramétert, melyre a legjobb LD értéket kaptuk.
3. Transzformáljuk az adatbázist a kapott paraméterhalmaz figyelembevételével.
4. Ugorjunk a 2-es lépésre mindaddig, amíg az átlagos paraméterváltozás kisebb egy előre beállított küszöbszámnál. Különben az algoritmus végrehajtását megszakítjuk, és a kapott paraméterekkel transzformált adatbázison betanítjuk a beszédfelismerő módszert.

Ismeretlen mintára a spektrumot transzformáljuk a lehetséges transzformációs paraméterek mindegyikével, majd meghatározzuk a hozzájuk tartozó legvalószínűbb szimbólumsorozatot az LD-VTLN során betanított beszédfelismerő segítségével. A felismerés eredménye ezek közül a legkisebb költségű lesz. Sajnos ez az eljárás azonban nagyon költséges, valós időben nem kivitelezhető.

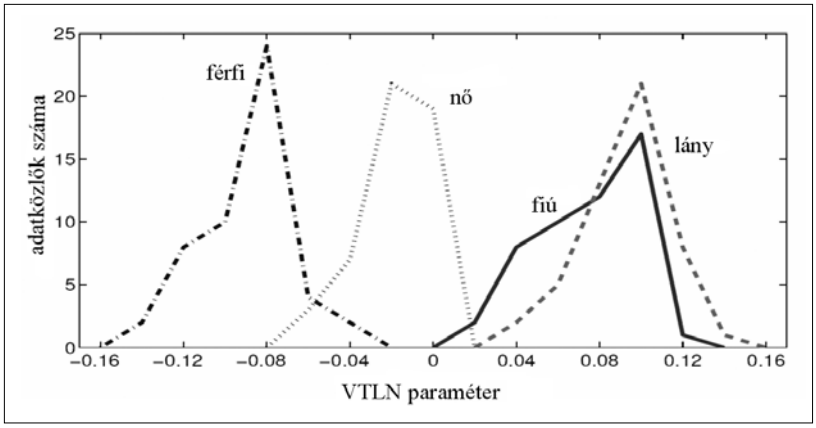
3.1. Valós idejű beszélőnormalizáció a Beszédmesterben

A BeszédMester beszédjavítás-terápia részében valós idejű fonéma-felismerési feladatot kell megoldani. A feladat során a tanár, illetve a diák felváltva használja a rendszert, ami tovább nehezíti a beszédfelismerőrendszer feladatát.

A rendszer tanításához bemondásokat rögzítettünk a célcsoportbeli felnőttek, illetve gyerekek segítségével. A gyerekek hangjához nehezebb megbízható felismerőt készíteni, mivel ebben a korban még sokat változik az artikulációs csatorna hossza és formája, ezért az elkészült beszédadatbázis sok különböző korú gyermek hangját tartalmazza. Első lépésben a tanítás során LD-VTLN alapú beszélő-normalizációt alkalmaztunk a fonémákból álló beszédadatbázison. A paraméterek eloszlása pontosan visszaadta a beszélők kor és nem szerinti eloszlását, amelyet megfigyelhetünk az *6. ábrán*.

Ezután a rendelkezésre álló optimális transzformációs paraméterek közelítésére egy regressziós algoritmust taníthatunk be az adatbázisban lévő bemondá-

6. ábra
A VTLN paraméterek eloszlása kor és nem szerint



sok alapján. A modell feladata, hogy a beszélő kilétének ismerete nélkül, mindössze a bemondásból kinyerhető akusztikai információk alapján becsülje meg az LD-VTLN algoritmus által kihozott optimális normalizációs paraméter értéket. Így a fonéma-felismerés során első lépésben a transzformációs paraméter becslése történik meg, majd ezután a transzformált hangon kell lefuttatni a beszédfelismerő rendszert. A felismerés így valós idejű marad, és teljesítménye alig marad el a globális keresés eredményétől. A tesztelés menetét és technikai részleteit a következő fejezetben írjuk le.

4. A módszer tesztelése

Ebben a fejezetben megvizsgáljuk, hogy az eredeti LD-VTLN algoritmus, illetve a valós idejű kiterjesztése milyen hatékonyságnövelésre képes a fonémaklasszifikáció feladatán. Először leírást adunk a teszteléshez használt beszédkorpuszról, majd számba veszzük az alkalmazott tulajdonságkinyerő módszert, illetve az osztályozó eljárást, végül bemutatjuk az eredeti LD-VTLN eljárás alkalmazásának részleteit és a teszt során használt regressziós algoritmust.

Beszédkorpusz. Tanítási és tesztelési céllal 200 beszélőtől rögzítettünk hangmintákat. A nemek közötti eloszlás a következő volt: 50 nő, 50 férfi, 50 lány és 50 fiú. A gyerekek a 6-9 éves korosztályból kerültek ki. A beszédjeleket 22050 Hz-es mintavételezéssel, 16-bit-es minőségben rögzítettük és tároltuk el. Minden beszélő rövid szünetekkel elválasztva az összes magyar magánhangzót kiejtette. Mivel úgy határoztunk, hogy nem teszünk különbséget a hosszú és rövid magánhangzók között, ezért összesen 9 magánhangzóval dolgoztunk.

Tulajdonságkinyerés. A jeleket 10 ms-os keretekben dolgoztuk fel, ezután 24 kritikus sáv energiáját számoltuk ki a logaritmikus skálán, az FFT és háromszög súlyozás felhasználásával [12]. Minden egyes keret energiáját külön-külön normalizáltuk, ami azt eredményezte, hogy csak a spektrális eloszlás alakját használtuk fel a fonémaosztályozáskor.

Osztályozók. A kísérletek során mesterséges neuronhálókat [1] használtunk fel a fonémaklasszifikációs feladat megoldására. A szokásosan alkalmazott háromrétegű, előrecsatolt többszintű neuronhálós modellt használtuk (feed-forward MLP), amelyet a backpropagation tanulóalgoritmussal tanítottunk. A rejtett neuronok száma minden esetben 16-ra lett állítva.

LD-VTLN. Az LD-VTLN algoritmust két különböző transzformációs függvény felhasználásával alkalmaztuk. A bilineáris és a szakaszonként lineáris transzformációs függvények, zárt alakban, rendre a következő formulákkal definiálhatók (1)(2):

$$f' = \arctan\frac{(1-\alpha^2)\sin f}{(1+\alpha^2)\sin f - 2\alpha} \qquad \alpha \in [-0.18, 0.18],$$

$$f' = \begin{cases} \alpha f & \text{if } 0 < f < 0.7/\alpha \\ \gamma f + (1-\gamma) & \text{egyébként.} \end{cases} \qquad \alpha \in [0.825, 1.175],$$

$$\gamma = \frac{0.3\alpha}{\alpha - 0.7}.$$

Az α transzformációs paraméter kezdőértékét 0-ra állítottuk az (1)-es és 1-re a (2)-es transzformációs függvény esetében. Az α lehetséges értékei rendre a formulák után láthatóak. Az optimalizáció elvégzéséhez α lehetséges értékeit kvantáltuk, az adott intervallumot ekvidisztáns módon 15 részre vágtuk. Az iterációt addig folytattuk, amíg a transzformációs paraméter átlagos változása 10^{-2} alá nem esett.

Regresszió. Az LD-VTLN módszer által kinyert beszélőkre jellemző paraméterek becslésére egy speciális MLP hálózatot hoztunk létre egy kimeneti neuronnal és két rejtett réteggel, amely 24-24 neuronból épült fel. A tanítást az átlagos négyzetes hiba minimalizálásával végeztük. A neuronháló inputját a 24 kritikus sáv energiája (ld. tulajdonságkinyerés) alkotta.

Tesztek. A kísérletek a következőképpen lettek elvégezve. Az összességében 200 ember bemondásaiból álló beszédkorpusz 3 db 50, 100 és 200 beszélőből álló részre lett felosztva, minden esetben egyenletes volt a fiúk, lányok, férfiak és nők aránya. Az így kapott beszédadatbázist 80/20% arányban bontottuk fel tanuló és tesztelő mintákra. A tanuló minták esetében mindkét transzformációs függvényre (bilineáris, szakaszonként lineáris) és minden lehetséges α értékre előállítottuk a transzformált mintát, majd kinyertük a 24 sáv energiaértékeit. Ezután végrehajtottuk az LD-VTLN algoritmust, amely beszélőnként kiválasztotta az optimális paraméter értéket. Ezek alkották a regressziós neuronháló kívánt outputját, míg az input adatokat az eredeti minta sávenergiái jelentették. A tesztek során neuronháló alkalmazásával a következő adatokon végeztünk osztályozást: transzformáció nélküli adatok (NO VTLN), LD-VTLN transzformálás utáni adatok (LD-VTLN), illetve a valós idejű transzformálással kapott minták (Realtime(RT)-VTLN).

Az eredményeket az alábbi *táblázat* foglalja össze:

Klasszifikációs hiba a magánhangzófelismerés feladatán. A sorok az adatközlők száma szerint, az oszlopok pedig a transzformációs függvény és a normalizációs algoritmusok szerint rendezettek.

Adatközlők száma	Bilineáris transzformáció			Szakaszonként lineáris transzformáció		
	NO-VTLN	LD-VTLN	RT-VTLN	NO-VTLN	LD-VTLN	RT-VTLN
50 adatközlő	18.52%	13.07%	14.09%	18.52%	13.86%	14.35%
100 adatközlő	15.36%	12.02%	13.02%	15.36%	12.17%	13.19%
200 adatközlő	14.33%	11.02%	11.52%	14.33%	10.87%	11.04%

Kiértékelés. Az eredményekből láthatjuk, hogy az LD-VTLN módszer a klasszifikációs hibát akár 21-29%-kal is csökkentheti. A regressziós becslést alkalmazva közelítőleg 14-24%-os hibacsökkenést kaptunk, amely megközelíti az LD-VTLN eredményeket. Az adatbázis méretének növelésével (50, 100, 200 adatközlő), a tényleges eltérés a két módszer között csökken. A két transzformációs függvény között nem tapasztaltunk jelentős különbséget.

5. Összefoglalás

A kifejlesztett Beszédmester szoftver egyrészt azzal a céllal készült, hogy segítse az iskolások olvasásfejlesztését, másrészt, hogy a hallássérült, siket és logopédiai kezelésben részesülő gyermekek számára kínáljon gyorsabb fejlődési lehetőséget. Előnye, hogy játékosan, színes képekkel, a számítógép motivációs erejét felhasználva próbálja meg a kisiskolásokat az olvasás rejtelmeire megtanítani, és a gyermekeket a tiszta, hangos beszéd birtokosává tenni. A szoftver célirányos, tudatos, oktató-fejlesztő program, mely figyelembe veszi az életkori sajátosságokat, a komplex készségfejlesztést és nagyfokú önállóságot biztosít.

A Beszédmester szoftver használati értékét a beszélőfüggetlen automatikus beszédfelismerő technológia nagymértékben megnöveli. Ennek a technológiának az egyik kulcseleme a beszélőnormalizáció. A beszédterápia során a tanár és a diák felváltva beszél, ezért gyorsan változhat az optimális normalizációs függvény, vagy ennak paramétere. A probléma megoldására egy valós idejű beszédnormalizációt lehetővé tévő VTLN eljárást ismertettünk, amely a bemondás pillanatában hatékonyan megbecsüli az alkalmazott transzformációs függvény paramétereit. Összegzésül, kijelenthetjük, hogy érdemes a VTLN eljárások – mint például az LD-VTLN – regresszió-alapú becslésével foglalkozni, mert az így kapott modell már valós időben kiértékelhető és majdnem olyan hatékony, mint a közelített eredeti eljárás.

Végezetül, meg kell említenünk, hogy fonológiai tudatosságra nevelő rendszerekben – mint a Beszédmester – nem csak a fonémák osztályozásának hatékonysága, hanem az egyes fonémaosztályok elkülönülésének milyensége is nagy jelentőséggel bír. Bizonyos beszélőnormalizációs módszerek, köztük az LD-VTLN és a bemutatott RT-VTLN, az egyes osztályok minél élesebb elválasztására törekszenek. Az e módszerek által létrehozott szeparációs modell vizsgálata további kihívást jelent és lehetőséget biztosít a kutatások folytatására.

Irodalom

[1] Bishop, C. M., (1995)
Neural Networks for Pattern Recognition, Oxford Uni. P.

[2] Claes, T., Dologlou, I., Bosch, L., Compernolle, D.(1998)
A Novel Feature Transformation for Vocal Tract Length Normalization in Automatic Speech Recognition,
IEEE Trans. on Speech and Audio Processing,
Vol. 6., pp.549–557.

[3] Csányi, Y. (1990) Hallás-beszédnevelés,
Tankönyvkiadó Budapest.

[4] Eide, E., Gish, H., (1997) A Parametric Approach to Vocal Tract Length Normalization, Proc. ICASSP'97,
Munich, Germany, pp.1039–1042.

[5] Farkas, M. (1996)
A hallássérültek kiejtés- és beszédfejlesztésének elmélete és gyakorlata, BGGYPF, Budapest.

[6] Kocsor, A., Tóth, L., Paczolay, D. (2001)
A Nonlinearized Discriminant Analysis and its Application to Speech Impediment Therapy, In:
V. Matousek, P. Mautner, R. Moucek, K. Tauser (eds):
Proc. of the 4th Int. Conf., Speech and Dialogue,
LNAI 2166, Springer Verlag, pp.249–257.

[7] Kocsor, A., Kovács, K. (2002)
Kernel Springy Discriminant Analysis and Its Applic. to a Phonological Awareness teaching System, In:
P. Sojka, I. Kopecek, K. Pala (eds.):
TSD 2002, LNAI 2448, Springer Verlag, pp.325–328.

[8] Kocsor, A., Tóth, L. (2004)
Kernel-Based Feature Extraction with a Speech Techn. Applic., IEEE Trans. on Signal Processing,
Vol. 52., No.8., pp.2250–2263.

[9] Paczolay, D., Kocsor, A., Sejtes, Gy., Hégely, G.(2004)
A „Beszédmester" csomag bemutatása: informatikai és nyelvi aspektusok. Alkalmazott Nyelvtudomány,
Veszprém, 4/1.szám, pp.57–79.

[10] Paczolay, D., Kocsor, A., Tóth, L. (2003)
Real-Time Vocal Tract Length Normalization in a Phonological Awareness Teaching System,
Matousek, V., Mautner, P. (eds.): TSD 2003,
LNCS 2807, Springer Verlag, pp.309–314.

[11] Paczolay, D., Tóth, L., Kocsor, A., Kerekes J. (2002)
Gépi tanulás alkalmazása egy fonológiai tudatosságfejlesztő rendszerben, Alkalmazott Nyelvtudomány,
2/2.szám, pp.55–67.

[12] Pitz, P., Molau, S., Schlter, R., Ney, H. (2001)
Vocal Tract Normalization Equals Linear Transform. in Cepstral Space, Proc. EUROSPEECH 2001,
Vol. 4., pp.2653–2656.

[13] Rabiner, L. R., Juang, B. H. (1993)
Fundamentals of Speech Recognition,
Englewood Cliffs, NJ, Prentice Hall.

[14] Uebel, L. F., Woodland, P. C. (1999)
Investigation into Vocal Tract Lenght Normalisation,
Proc. EUROSPEECH 99, Hungary,
Vol. 6., pp.2527–2530.

[15] Ványi, Á. (1998) Olvasástanítás a diszlexia-prevenciós módszerrel. Project-X. Budapest, pp.4–7.

[16] Wegmann, S., McAllaster, D., Orloff, J.,
Peskin, B. Speaker (1996)
Normalization on Conversational Telephone Speech,
Proc. ICASSP'96, Atlanta, Vol. 1., pp.339–341.

[17] Westphal, M., Schultz, T.,Waibel, A. (1998)
Linear Discriminant – A New Criterion for Speaker Normalization, Proc. ICSLP'98, pap.no.755, Sydney.

[18] Zhan, P., Westphal, M. (1997)
Speaker Normalization based on Frequency Warping,
Proc. ICASSP-97, Munich, Vol. 1, pp.1039–1042.

Új protokoll terve vezeték nélküli MIDI kapcsolatok megvalósítására Bluetooth rendszerben

HUSZTY CSABA, BALÁZS GÉZA

Távközlési és Médiainformatikai Tanszék, Budapesti Műszaki és Gazdaságtudományi Egyetem
info@midioverbt.com

Kulcsszavak: vezeték nélküli MIDI, Bluetooth

A cikk egy új protokoll terveit mutatja be vezeték nélküli MIDI kapcsolatok megvalósítására Bluetooth rendszerben. A gyakorlati felhasználást figyelembe véve a terv közel tetszőleges összeköttetési topológiát tesz lehetővé. Bemutatjuk egy általános Bluetooth-alapú MIDI rendszer tervét és adatátviteli módját, kiszámoljuk a késleltetését és megvizsgáljuk alkalmazhatóságának korlátait, egyúttal javaslatot téve a rendszer funkcióinak néhány lehetséges további bővítésére.

1. Bevezetés

A MIDI (Musical Instrument Digital Interface) legfőbb feladata a hangkeltéshez szükséges vezérlési információk átvitele mellett a rendszer szinkronizációja [1]. A MIDI protokollt használó berendezések többnyire három csatlakozási lehetőséggel vannak ellátva. Az IN csatlakozót a szomszédos egység OUT csatlakozójával fizikai kábellel kötik össze, a THRU kimenet segítségével pedig az összeköttetés láncszerkezetűvé alakítható úgy, hogy a THRU portra az IN-re érkező adatok kerülnek.

Ezzel a módszerrel új eszközök használata nélkül csak korlátozott bonyolultságú összeköttetéseket lehet létrehozni. Gyakori probléma több kimenet egy bemeneten való egyesítése is, amelyet szintén csak kiegészítő berendezéssel, úgynevezett MIDI Merger-rel lehet megvalósítani. Tekintettel az összekötő kábelek néhány méteres maximális hosszára, az összekötendő berendezések nem helyezhetők el akármilyen távolságban és sorrendben, valamint azt is számításba kell venni, hogy 15-20 egység esetében már az összeköttetést biztosító elosztó és összekapcsoló berendezések is számottevő helyet, árnövekedést és késleltetést vihetnek a rendszerbe. A szabványnak az implementációra vonatkozó korlátaival a továbbiakban nem foglalkozunk.

Az olcsó vezetéknélküli alkalmazások közül többek között a Bluetooth is ideális a MIDI összeköttetések helyettesítésére. Az egyes egységek kis teljesítményfelvétele, elegendő hatótávolsága és kedvező tulajdonságú zavarérzékenysége is alkalmazhatóvá teszi erre a célra. A Bluetooth-technikával kialakított MIDI összeköttetések nem pusztán a kábelt, hanem az összeköttetést segítő egyéb berendezéseket is ki tudják részben küszöbölni, illetve integrálhatóvá tenni már létező berendezésekbe. Bár más munkák már foglalkoztak a MIDI és a Bluetooth együttes alkalmazásával [4], e cikk új módszert mutat be a MIDI összeköttetések megvalósítására és továbbmegy a MIDI kábel helyettesítésénél.

2. MIDI összeköttetések megvalósítása Bluetooth rendszerben

2.1. Az alkalmas kapcsolati és csomagtípus megválasztása a MIDI-hez

A Bluetooth jelenlegi, 1.2 változata pikonetenként egy master egységet és legfeljebb hét aktív slave egységet támogat, bár parkolási állapotban hétnél több slave is lehet a pikonetben [2,3].

A csatorna-hozzáférést a master egység vezérli. Egy pikoneten belül a master egység órajeléhez igazodik a pikonet összes többi tagja. Minden egység időben és ugratásban (frekvenciában) is a masterhez szinkronizált. A master csak páros számú időszeletekben (slot) kezdeményezhet adást, páratlanokban vételt, a slave-ek pedig fordítva: csak páratlan számú időszeletben kezdeményezhetnek adást. A fentiek csak a kezdeményezésre vonatkoznak: a már megkezdett adás eltarthat több slot ideig is.

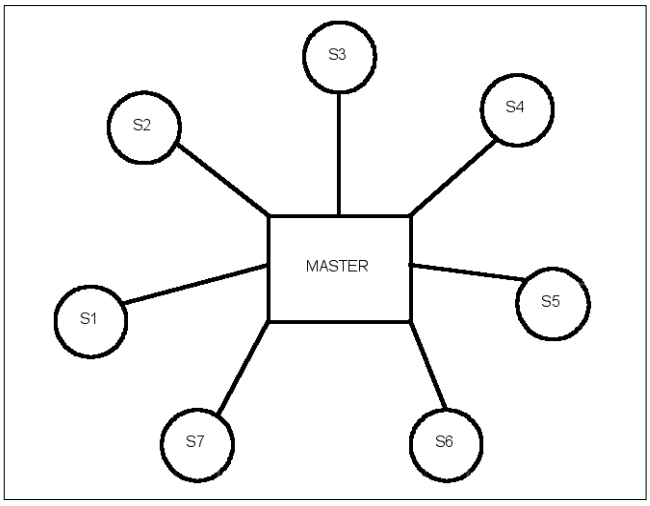
A MIDI alkalmazás szempontjából az ACL átvitelt választjuk. Ebből egy pikoneten belül egyidejűleg egy adatcsatorna működése támogatott. Az átvitelhez a szabványban hétféle csomagot definiáltak, az ezekkel átvihető byte-ok maximális számát a *táblázatban* tüntettük fel. Az átvitelre az M (Medium) csomagot választjuk, mert az M csomag esetén 2/3 FEC hibajavító kódolást alkalmaznak, míg a H (High) csomagok esetében egyáltalán nincs hibajavító kódolás.

1. táblázat
BT csomagok maximális byte-száma

csomag típus	max. átvitel
DM1	18
DH1	28
DM3	123
DH3	185
DM5	226
DH5	341
AUX1	30

2.2. A hub-alapú topológia

A MIDI-s alkalmazást nagyban megkönnyíti a csillagpontos elrendezésű Bluetooth. Broadcast jellegű üzenetek és megfelelő beállítások mellett az adatfolyamot módosító (pl. Merge) bármely összeköttetés megvalósítható.



1. ábra Hub-based topology

A kapcsolat beállításakor már tudjuk, hogy melyik egység működik MIDI Out-ként, melyik MIDI In-ként. A Bluetooth alapú megvalósítás járulékosan annak a lehetőségét is magával hozza, hogy egy egység kimenete önmagába visszahurkolódjon – ez a MIDI Echo funkció.

2.3. Az alapvető MIDI összeköttetések megvalósítása

A tetszőleges összeköttetési topológiához a MIDI In, Out, MIDI Thru Box (Hub), MIDI Merge, Echo és Patchbay eszközöket és funkciókat helyettesítő logikai összeköttetéseket kell megvalósítani *(2. ábra)*.

MIDI Out/In (MIDI kábel)

A slave egységekhez kapcsolódó MIDI eszközök adatai közvetetten, a master egységen keresztül jutnak el a másik eszközhöz. Az In és Out portokat a felhasználó jelöli ki. A master egység az első slave-et (S1) poll üzenettel lekérdezi, ennek hatására az beküldi MIDI adatait, majd a master broadcast üzenetben kiküldi az adatokat az összes slave-nek. A beállított topológiától függően az S1 egység figyelmen kívül hagyhatja a hozzá megérkezett adatokat.

MIDI Hub (Thru Box)

Egy bemenet több kimenettel történő összekötése logikai összeköttetésekkel szintén egyszerűen megvalósítható. A master egység poll üzenetére az S1 slave adatot küld. A rendszerben nincs az S1-en kívül további Out port, így a lekérdezés befejeződik, a master egység broadcast üzenetben továbbítja az adatokat az összes slave felé. Az S1 slave azonban nincs beállítva In-ként, így az nem fogja figyelembe venni a hozzá érkezett MIDI üzeneteket. Járulékos előny, hogy több slave esetén az adatok megérkezésének késletetése nem nő meg.

MIDI Merge modul

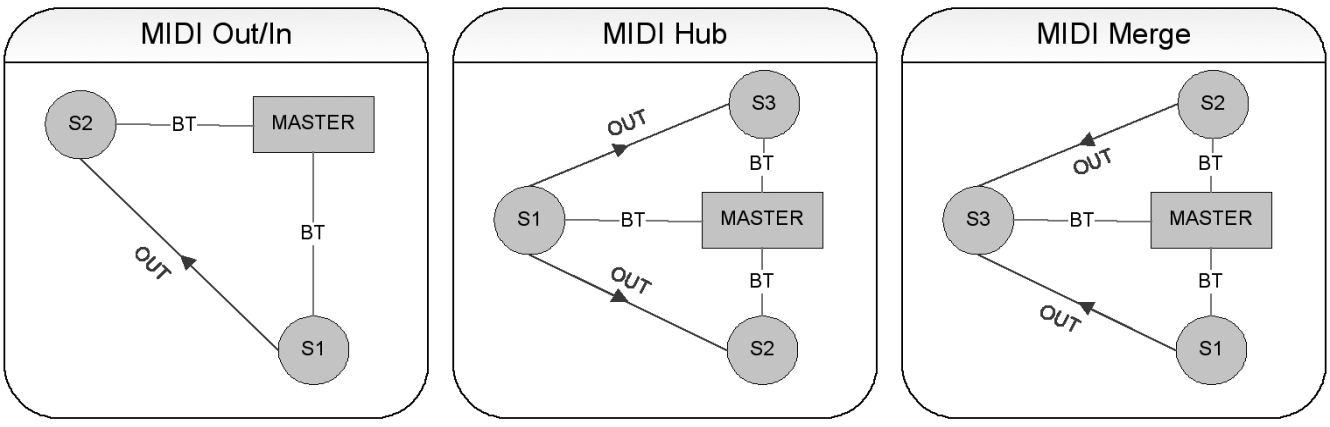
Két vagy több bemenet egy kimenetté történő egyesítésére a Master egység az S1 slave-nek egy Poll üzenetet küld, melyre az S1 slave válaszol, és visszaküldi az időközben összegyűlt MIDI adatait. Az adatok a master egységben tárolódnak, amelyek az S2 egység lekérdezése és adatbeküldése után egy többesküldés (broadcast) üzenetben egyszerre kerülnek elküldésre. Szükséges azonban a MIDI adatokat még kiküldésük előtt feldolgozni annak érdekében, hogy konzisztenciájuk megmaradjon, de ezt a feladatot az S3 egység lokálisan el tudja végezni még azelőtt, hogy a hozzá kapcsolt MIDI berendezésnek eljuttatná az adatokat.

3. A javasolt protokoll terve és időzítése

A MIDI adatok logikai egységeit, az üzeneteket a Bluetooth rendszerben csomagokba kell szervezni. Az időegység (MIDI slot) alatt átvitt csomagokban levő MIDI byte-okat mindig üzenethatáron tördeljük, ez alól egyetlen kivétel a System Exclusive Message (SysEx), amely tetszőlegesen hosszú lehet. Sorban, egymás után olvassuk be az előre definiált topológia szerinti MIDI Out portokról a beérkező MIDI byte-okat, ezeket minden Out számára konstans hosszúságú csomagokba foglaljuk, majd elküldjük a megfelelő In-ekre.

Egyes esetekben a küldés előtt fel kell dolgozni a csomagokat, hogy ne haladják meg a MIDI sávszélességét (például Merge). Egy ilyen ciklust nevezünk MIDI slot-nak, amelynek hosszúságát időben állandónak kell rögzíteni és tartani. Tekintettel arra, hogy változó számú byte érkezhet a MIDI slot ideje alatt, a csomag hosszát és a késleltetési időt maximális adatmennyiség esetére számoljuk.

2. ábra MIDI összeköttetések megvalósítása



A MIDI slot időzítése (DM3 válasz csomag esetében egyszeres küldéssel):

1. A Master Poll üzenete (DM1 csomag) a 0. Bluetooth slotban az első olyan slave-nek, amelyhez egy Out porttal rendelkező MIDI eszköz van csatlakoztatva [5].
2. A megcímzett Slave visszaküld egy állandó hosszúságú DM3 csomagot az addig felgyülemlett MIDI adataival az 1. BT slotban. Az átvitel a 3. BT slotban ér véget. Ha van még Out a rendszerben, a folyamat a soron következő slave-vel megismétlődik.
3. 2 darab üres BT slot következik, majd az Out-ok számától függően DM3, DM5 vagy DH5 típusú csomagban broadcast csomag az összes slave-nek. A master egység a slave-ek adatainak fogadásával egyidőben (a BT modulok UART portjai full-duplex üzeműek) elkezdheti összeállítani a küldendő broadcast csomagot, így csak az utolsó slave adatainak beillesztését kell a broadcast csomag kiküldése előtt megvárni. Mivel azonban az utolsó slave adatcsomag megérkezése után páros BT slot következik, a várakozás miatt meg kell várni a következő páros BT slot-ot, mielőtt a broadcast üzenetet kiküldenénk. Ez okozza a 2 BT slot idejű további késleltetést.
4. Végül ismét egy üres BT slot következik, mert a következő poll ciklus csak páros BT slot-ban kezdődhet és a broadcast csomagra nem válaszolhatnak az egységek. A ciklus ezután az első lépéstől ismétlődik.

Összesen tehát

$$N_{BT\_SLOTS} = 4*O + 2 + x + 1$$

BT slot-nak megfelelő idő telik el, ahol *O* az Out-ok száma, *x* pedig 3 DM3 vagy 5 DM5 csomag esetében.

Legyen a MIDI slot hossza byte-okban

$$B = S_{MIDI} \cdot T_{BT},$$

ahol a MIDI vonal átviteli sebessége S_{MIDI} = 3125 byte/s, T_{BT} pedig egy BT slot ideje: 625 µs.

Az időzítés tervezése érdekében kiszámítjuk, mekkora átviendő adatmennyiségre lehet igény. Egy MIDI slotban a maximálisan átvitt MIDI byte-ok száma egy aktív Out esetén:

$$B_{1\_OUT} = S_{MIDI} \cdot T_{BT} \cdot (1+1+2+1+1) = 6*S_{MIDI}T_{BT} = 11.71875 \Rightarrow 12 \text{ byte}.$$

Felfelé kerekítettünk a legrosszabb esettel számolva. Az összeg tagjai: 1 poll a mastertől a slave felé, 1 válasz a slave-től a master felé, 2 üres slot, 1 broadcast üzenet a mastertől a slave-ek felé és 1 üres slot, így összesen 6 slot hosszúságú egy MIDI slot.

A fent kiszámolt eredménynél a tényleges érték kettővel nagyobb, mert a MIDI csomagok általában kettő illetve három byte-os üzenetekből állnak, így előfordulhat, hogy a 11-ik byte után egy 3 byte-os üzenet kezdődik, amit még át kell vinni. A maximálisan átviendő MIDI byte-ok száma tehát

$$B_{ÁTVIENDŐ} = B_{1\_OUT} + 2 = 14 \text{ byte}.$$

Ahhoz, hogy az időzítést terhelt és terheletlen rendszer esetén egyaránt tartani tudjuk, válasszunk minden átvitt csomagot egyforma hosszúságúnak, függetlenül attól, hogy hány byte hasznos adat van benne. Az átvitelhez további 2 byte adminisztrációs információ szükséges (fejléc és lábléc). Az 1 byte-os fejléc az Out sorszámát fogja tartalmazni, amely alapján az esetlegesen implementált Merger programmodul azonosítja majd, melyik folyamba kell beillesztenie az érkező byte-okat, az ugyancsak 1 byte-os lábléc pedig a csomag végét jelző byte-ot, amit egy a MIDI szabványban nem definiált byte-tal valósítunk meg. Összesen tehát

$$B_{MERGER} = B_{ÁTVIENDŐ} + 2 = 16 \text{ byte}$$

kerül átvitelre egy MIDI slot-ban. Mivel a DM1 csomag pontosan 18 byte-ot képes átvinni és 1 BT slot ideig tart, ez ideális ennek az adatmennyiségnek az átvitelére.

Egyszerűsége és rugalmassága miatt kézenfekvő, hogy UART-típusú Host Controller Interface-szel (HCI) épített Bluetooth modult használjunk a realizációban, így a következőkben UART-alapú rendszerre számítjuk ki az időzítést.

A korrekt számításokhoz először is figyelembe kell venni, hogy az UART HCI BT moduloknál a csomagformázáshoz további 5 byte-ot át kell vinni (1 byte ACL azonosító, 2 byte connenction handle, a master-slave fizikai kapcsolat azonosítójával; valamint 2 byte flag a csomag hosszáról, amely az 5 fejléc byte nélkül veendő figyelembe). Ezek a byte-ok a küldés során a levegőbe nem kerülnek ki, így csomagtípust nem kell váltanunk miattuk. Összesen tehát 21 byte fog a BT modulhoz megérkezni, és a master egység is ugyanennyit kap a saját soros portjáról.

Ez egy 1 382 400 bit/s sebességű UART esetén 152 µs-ig tart (1 start bit + 8 adatbit + 1 stop bit = 10 bit. Az idő: T_{UART} = 10*21/1382400 = 152 µs), ami 24,3%-a a BT slot hosszának.

Ugyanez a számítás 2 Out-ot feltételezve már (DM3-as csomaggal) 506 µs-ra adódik, de még nem éri el a BT slot hosszát – vagyis az időzítés tartható.

3 Out esetében már más a helyzet: B_{3_OUT} = 20* $S_{MIDI}T_{BT}$ = 40 byte, mert a broadcast csomag már 5 slot hosszúságú. A broadcast hossza 132 byte – ez már DM5-ös csomagot igényel. Az átvitel a slave egységeknél 354 µs, a master egységnél 998 µs-is tart.

4 Out esetében a broadcast-ra 204 byte adódik. Az átviteli idő slave-eknél 405 µs, a masternél 1519 µs.

Végeredményben egy pikoneten belül még akár 5 Out is megvalósítható lenne, csak itt már a broadcasthoz nem lehet M csomagot használni, mert 295 byte-ot kellene elküldeni, ez pedig a DM5-be sem fér már bele. Az átvitel időtartama slave esetén 463 µs, master esetén pedig 2177 µs. Ez utóbbi érték BT slotokban kifejezve 3.48, tehát még mindig nem veszélyes az időzítésre.

Ennél több MIDI Out nem valósítható meg egy pikoneten belül, mivel ezt a BT sávszélessége nem teszi lehetővé.

3. ábra A broadcast csomag felépítése

Adatvédelem szempontjából sajnos a fenti megoldások meglehetősen rosszak – annak ellenére, hogy késleltetésük kétségkívül a legkisebb –, hiszen csak egyszer küldenek el minden adatot a címzettnek. A 2/3 FEC kódolás valamivel javítja a zavarérzékenységet, de az érték továbbra is függ az adó egységek térbeli elhelyezésétől és távolságától is. A legtriviálisabb módja a adatvesztés elkerülésének a többszörös küldés.

1 Out esetén akár háromszor is megtehetjük ezt, ebben az esetben a késleltetési idő 16 ms, de négyszeres újraküldés esetén is csak 21 ms.

2 Out-nál már csak kétszer küldhetjük újra az üzeneteket (mind a poll-okat, mind a válaszokat, mind pedig a broadcast-ot), ekkor a késleltetési idő 18,75 ms.

3 Out-nál DM5 csomag esetén még éppen meg lehet kétszer ismételni a lekérdezést, ekkor viszont a broadcast már csak egyszer fér bele a MIDI slot-ba. Ha a broadcast-hoz DH5-ös csomagot használunk, akkor megismételhető a küldés.

3-nál több Out-nál már sajnos semmilyen ismétlést nem tudunk megvalósítani a MIDI slot idejében a jelenlegi Bluetooth szabvány adatsebessége mellett.

Az előzőeket a *2. táblázatban* foglaltuk össze. (Megj.: 1 Out esetében a 14-szeres újraküldés a jelenlegi Bluetooth szabvány teljesítőképességének határa.

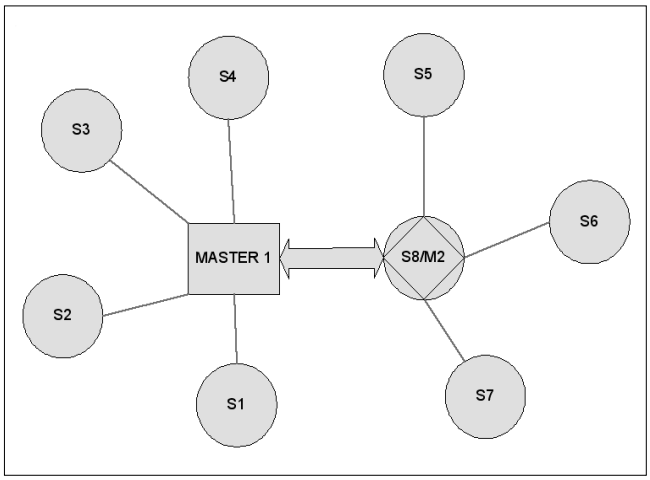
3.1. A MIDI csomagok felépítése

Maximálisan 3 Out-ot feltételezve a broadcast csomag a *3. ábra* szerint épül fel.

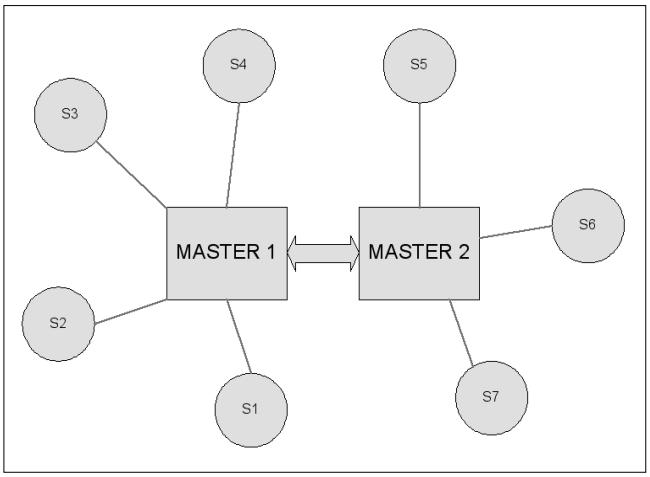
A csomag DATA jelzései az egyes Slave egységek MIDI adatait jelentik. Az adatcsomag lezárása a MIDI-ben nem definiált funkciójú 0xF9-cel történik, a broadcast csomag lezárása pedig a szintén nem definiált 0xFD-vel. Ha a MIDI berendezés a nem definiált 0xF4, 0xF5, 0xF9 vagy 0xFD MIDI byte-ok közül valamelyiket saját speciális működéséhez felhasználja, akkor a csomag struktúráját és az időzítést is módosítani kell, hogy ezeket a byte-okat is át tudjuk vinni.

2. táblázat
Az átvitel késleltetése – a kiemelt sorban levők szerint javasoljuk az összeköttetést megvalósítani.

	MIDI kimenetek száma	Válasz csomag típusa (Slave)	Broadcast csomag típusa (Master)	Késleltetés (slot)	Átvitt slot-ok (sorrendben)	**Késleltetés**	fizikailag átvitt MIDI byte-ok	logikailag átvitt MIDI byte-ok
1-szeres küldés	1	DM1	DM1	6	1 lekérdezés (poll), 1 válasz, 2 üres, 1 broadcast, 1 üres	**3,75 ms**	12	14
	2	DM3	DM3	14	2x1 lekérdezés (poll), 2x3 válasz, 2 üres, 3 broadcast, 1 üres	**8,75 ms**	28	30
	3	DM3	DM5	20	3x1 lekérdezés (poll), 3x3 válasz, 2 üres, 5 broadcast, 1 üres	**12,50 ms**	40	42
	4	DM3	DM5	24	4x1 lekérdezés (poll), 4x3 válasz, 2 üres, 5 broadcast, 1 üres	**15,00 ms**	47	49
	5	DM3	DH5	28	5x1 lekérdezés (poll), 5x3 válasz, 2 üres, 5 broadcast, 1 üres	**17,50 ms**	55	57
2-szeres küldés	1	DM3	DM3	18	2x1 lekérdezés (poll), 2x3 válasz, 2 üres, 2x3 broadcast, 2x1 üres	**11,25 ms**	36	38
	2	DM3	DM5	30	2x2x1 lekérdezés (poll), 2x2x3 válasz, 2 üres, 2x5 broadcast, 2x1 üres	**18,75 ms**	59	61
	3	DM3	DH5	38	2x3x1 lekérdezés (poll), 2x3x3 válasz, 2 üres, 2x5 broadcast, 2x1 üres	**23,75 ms**	75	77
3-szoros küldés	1	DM3	DM3	26	3x1 lekérdezés (poll), 3x3 válasz, 2 üres, 3x3 broadcast, 3x1 üres	**16,25 ms**	51	53
	2	DM3	DH5	44	3x2x1 lekérdezés (poll), 3x2x3 válasz, 2 üres, 3x5 broadcast, 3x1 üres	**27,50 ms**	86	88
4-szeres küldés	1	DM3	DM3	34	4x1 lekérdezés (poll), 4x3 válasz, 2 üres, 4x3 broadcast, 4x1 üres	**21,25 ms**	67	69
5-szörös küldés	1	DM3	DM3	42	5x1 lekérdezés (poll), 5x3 válasz, 2 üres, 5x3 broadcast, 5x1 üres	**26,25 ms**	83	85



4. ábra
Scatternet multifunkcionális (master-slave) egységgel



5. ábra
Scatternet nagysebességű kapcsolattal

A broadcast csomag maximális hossza 238 byte (DH5 csomag, 3 Out, kétszeres küldés). A lekérdezés (poll) üzenet tartalma közömbös.

3.2. A késleltetés csökkentése scatternet alkalmazásával

Ha egyszerre több master-t alkalmazunk, csökkenthető a késleltetési idő, mert az Out egységeket szét lehet osztani a master-ek között. Sajnos nem könnyű elkerülni, hogy a két master egység véletlenül se adjon ugyanazon a frekvencián, bár a Bluetooth 1.2 szabványával ez megoldható, ezért még inkább szükség lehet a többszörös újraküldésre, ami viszont megnöveli a késleltetést. A tetszőleges topológia ezek után a pikonetek összekapcsolásával valósítható meg.

Scatternet kialakítása a pikonetekből többféleképpen is lehetséges: megtehető, hogy egy slave egység egy másik pikonetben master-ként üzemel *(4. ábra)*, vagy hogy a master egységeket egyszerűen összehuzalozzuk egy nagysebességű busszal *(5. ábra)*. A MIDI megvalósítása szempontjából az első módszer semmiképpen sem megfelelő, hiszen a többfunkciós S8/M2 egység egyszerre csak egyik funkcióját láthatja el – a szinkronitás nem tartható. Felmerül viszont a kérdés, hogy mi történik akkor, ha kiegyenlítetlenül helyezkednek el az Out funkciójú slave egységek az egyes pikonetekben. Az időzítés tárgyalásakor már láttuk, hogy ahogyan nő az Out-ok száma, úgy nő a késleltetés, és úgy változnak az alkalmazandó protokoll paraméterei is, vagyis az újraküldések száma (ismételt adás) és a Bluetooth csomagok típusa.

Egy olyan rendszer megépítését tartjuk legcélszerűbbnek, melyben a felhasználónak nem kell az újabb master egységek üzembe helyezésekor az egész addigra meglevő és működő rendszerét újrakonfigurálni, és kényelmetlen, hosszadalmas úton az új összeköttetési topológiát a rendszerbe bevinni. Az OUT-ok elosztása során törekedni kell arra, hogy minden master a lehető legkevesebb Out-ot kapja meg, és hogy az ugyanazokat az Out-okat megkapó In-ek lehetőleg egy pikonetbe kerüljenek, vagyis minél kevesebb adat kerüljön a pikoneten kívülre.

4. Összefoglalás

A cikkben javaslatot tettünk egy Bluetooth protokollkoncepcióra és megvizsgáltuk annak megvalósíthatóságát vezetéknélküli MIDI kapcsolatok megvalósítására. Annak ellenére, hogy a Bluetooth segítségével történő megvalósítások eddig nem jártak túl nagy sikerrel, az itt felvázolt rendszer egészében kihasználja a Bluetooth egyre növekvő teljesítőképességét. Állandó késleltetési idő tartása mellett biztonságos adatforgalmat is lehetővé tesz. A cikkben ismertetettek alapján megépített rendszer bővíthető oly módon, hogy a felhasználónak semmit sem kell megváltoztatnia az addigra már működő régi rendszerében, csökkentheti a rendszer késleltetési idejét, és növeli a kiszolgálható egységek számát.

A legfontosabb konklúziók a rendszer megvalósíthatóságára: (1) legalább kétszeres újraküldést kell alkalmazni, (2) max. 3 Out lehet 1 pikonetben, amelyből a saját pikonetből származó Out-ok száma maximálisan 2 lehet. (3) Ha 1 Out sincs bekötve, nem alkalmazunk broadcast-ot, 1 és 2 bekötött Out esetén egyaránt DM3 csomagot alkalmazunk (a késleltetés 16,25 ms és 18,75 ms), 3 Out esetén pedig DH5-öt (23,75 ms). (4) Legfeljebb 3 master egységet kapcsolhatunk össze egy légtérben úgy, hogy minden egységhez nagy valószínűséggel megérkezzenek az adatok Bluetooth 1.1 szabvány alkalmazása esetén. Ez nem vonatkozik az 1.2 szabványra, ami lehetővé teszi, hogy egy pikonetben tetszés szerint osszuk el a rendelkezésre álló 79 frekvenciát.

Irodalom

[1] MMA MIDI Specifications, 1983-2003
[2] Specification of the Bluetooth System v1.1
[3] Specification of the Bluetooth System v1.2
[4] J. Keniston, S. Sturdivant (2003)
Wireless MIDI Network Implemented Via Bluetooth
[5] R. Mettala: Bluetooth Protocol Architecture; v 1.0
(Bluetooth White Paper Document # 1.C.120/1.0)

Kétosztályos WFQ kiszolgálás közelítő vizsgálata

HORVÁTH GÁBOR, TELEK MIKLÓS

Budapest Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék
{ghorvath, telek}@hit.bme.hu

Kulcsszavak: WFQ kiszolgálás, kétdimenziós Markov-lánc, várakozási idő várható értéke és szórása

A súlyozott igazságos kiszolgálási elvet (Weighted Fair Queueing – WFQ) régóta számos távközlési és számítástechnikai rendszerben alkalmazzák erőforrások megosztásának szabályozására. Látszólagos egyszerűsége ellenére a WFQ hatékony analitikus teljesítményvizsgálata még mindig nyitott kérdés. Az irodalomban megtalálható – numerikus, vagy komplex analízisen alapuló – algoritmusok gyakorlati alkalmazhatósága meglehetősen korlátozott. Ebben a cikkben egy egyszerű és gyors közelítő eljárást mutatunk be WFQ rendszerek vizsgálatára. Szimulációs eredményekkel igazoljuk, hogy egyszerűsége ellenére az ismertetett megközelítés megfelelően pontos.

1. Bevezetés

A súlyozott igazságos (WFQ) kiszolgálási elvet erőforrások megosztására alkalmazzák többosztályos környezetben, melyben az igények igényosztályokba sorolhatók. Minden igényosztályhoz tartozik egy súly. WFQ rendszerekben a teljes kiszolgálási kapacitás megosztását az igényosztályok között a pillanatnyilag jelenlévő igényosztályok súlyának aránya határozza meg. Ily módon az igények „fontossága” a súlyok segítségével szabályozható. Az igények torlódás esetén sorban állnak a kiszolgálóért, FCFS (first came first serve) elv szerint.

Ha az igények érkezési folyamatát Poisson folyamattal, a kiszolgálási időket pedig exponenciális eloszlással jellemezzük, akkor a WFQ rendszer egy kétdimenziós Markov-lánccal modellezhető.

Az irodalomban számos eljárás található ennek a kétdimenziós Markov-láncnak a megoldására. Először a numerikus eljárásokat vesszük sorra: [1,2]-ben a szerzők ugyanilyen működésű rendszer megoldásával foglalkoztak, bár *Coupled Processor Model*-nek hívták. A Markov-lánc egyensúlyi eloszlását a rendszer terhelésének a hatványsoraként fejezték ki, és adtak egy algoritmust a hatványsor együtthatóinak rekurzív kiszámolására. Ezzel a megközelítéssel csak néhány (2-3) igényosztályos rendszert tudtak kiszámolni, és ahogy a terhelés 1-hez közelít, túl sok együtthatót kell kiszámolni.

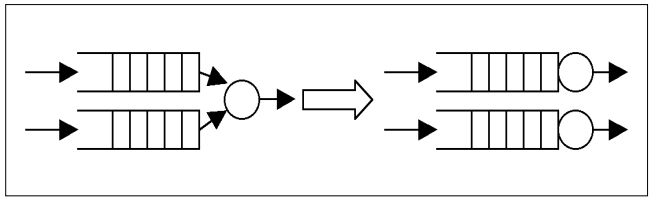
Egy másik megoldás ([3]) a végtelen Markov-láncot egy véges Markov-lánccal közelíti, és egyfajta Gauss-eliminációt használ az egyensúlyi eloszlás kiszámítására. A Gauss-elimináció közben kihasználja a mátrix speciális struktúrája adta gyorsítási lehetőségeket. De ebben az esetben is, nagy terhelés mellett a végesített Markov-láncnak is túl sok állapota lesz, és a megoldás drasztikusan lelassul.

[4]-ben a szerzők felírják az egyensúlyi eloszlás generátorát (Laplace transzformáltját). Az eredmény egy kétváltozós komplex (éppenséggel analitikus) függvény, ahol a problémát a peremeloszlások kiszámítása jelenti, ehhez ugyanis Wiener-Hopf faktorizációt kell alkalmazni.

Ebben a cikkben 2 osztályos WFQ rendszerrel foglalkozunk, de az ismertetett módszer egyszerűen kiterjeszthető többosztályos rendszerekre is. A fent ismertetett megoldásokkal ellentétben az igények érkezési és kiszolgálási idejét nem exponenciális eloszlásúnak tételezzük fel, hanem két momentumot veszünk figyelembe. Közelítést adunk az igények várakozási idejének várható értékére és szórására.

2. A közelítés elve

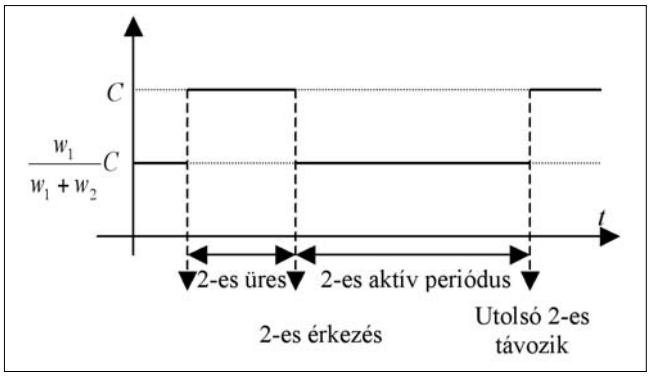
A közelítés lényege, hogy a két igényosztályt szeparáltan vizsgáljuk *(1. ábra)*. Egy olyan kiszolgálási folyamatot konstruálunk mindkét igényosztály számára, ami az eredeti kiszolgáló viselkedését „imitálja”.



1. ábra Az igényosztályok szétválasztása

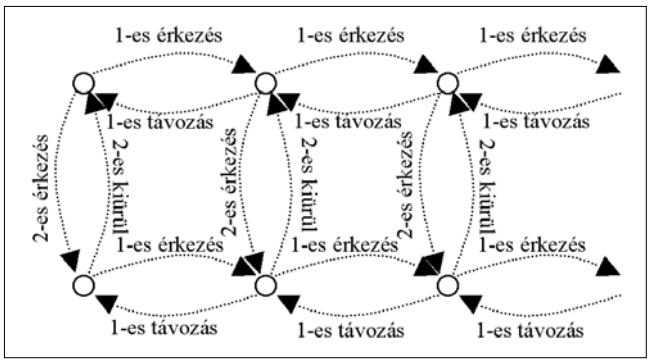
Például az 1. osztályt tekintve látható, hogy a kiszolgálási kapacitás a teljes kapacitás (*C*) és a súlyoknak megfelelően csökkentett kapacitás között változik attól függően, hogy van-e a rendszerben 2-es típusú igény.

Az ötlet egyszerű: jellemezzük a 2-es típusú igények aktív periódusának hosszát, és konstruáljunk egy olyan modulált kiszolgálási folyamatot, ahol a kapacitás modulációját a 2-es típusú igények aktív periódusa határozza meg *(2. ábra)*.



2. ábra A modulált kiszolgálási folyamat

A két igénytípus szeparálása után az így már egyosztályossá váló sorbanállási rendszert kvázi születés-halálozási (QBD) folyamattal modellezzük, és mátrix geometrikus technikával oldjuk meg. Az így keletkező (egy igényosztályt modellező) Markov-láncban az állapotokat duplikáljuk: az egyik csoport a 2-es igények aktív, a másik csoport a passzív állapotához tartozik. A Markov-lánc makró szintű struktúráját a *3. ábra* szemlélteti.



3. ábra A közelítő Markov-lánc szerkezete

A részletes tárgyalás előtt foglaljuk össze a megoldás menetét:

- Fázis-típusú eloszlást konstruálunk az igények érkezési és kiszolgálási idejének modellezésére. A fázistípusú eloszlások használata teszi lehetővé a sormodellek mátrix-geometrikus megoldását.
- Kiszámoljuk a két igényosztály aktív periódusának a hosszát. Ez a lépés négy eredményt fog adni, az aktív periódus hosszát a két osztályra a két lehetséges kiszolgálási kapacitás (teljes, illetve súlyoknak megfelelően csökkentett kapacitás) mellett.
- A 3. ábrának megfelelően összeállítjuk és megoldjuk az igényosztályok sorbanállását egyenként jellemző Markov-láncot. A teljesítményjellemzőket ennek a Markov-láncnak a segítségével kapjuk meg.

2.1. Az érkezési és kiszolgálási folyamat

A továbbiakban az *i* igényosztályhoz tartozó mennyiségeket és jeleket az $^{(i)}$ index jelöli (ebben a cikkben két igényosztállyal foglalkozunk).

Az érkezési folyamatot két mennyiség jellemzi: az érkezési intenzitás $\lambda^{(i)}$, valamint az érkezési időközök relatív szórásnégyzete $c_A^{2(i)}$.

Az így adott két momentum alapján egy olyan másodrendű, aciklikus fázis típusú eloszlást készítünk (PH, [6]), melynek ugyanez az első két momentuma. A PH eloszlásoknak három paramétere van: a tranziens átmenetek generátor mátrixa $D^{(i)}$, a nyelőbe vezető átmenetek rátavektora $d^{(i)}$, valamint a kezdőállapot-valószínűség vektor $\delta^{(i)}$. Könnyen ellenőrizhető, hogy a következő PH eloszlás első két momentuma megegyezik az érkezési időközeink első két momentumával:

$$D^{(i)} = \begin{bmatrix} -\frac{\lambda^{(i)}}{c_A^{2(i)}} & \frac{\lambda^{(i)}}{c_A^{2(i)}} \\ 0 & -2\lambda^{(i)} \end{bmatrix}, \quad d^{(i)} = \begin{bmatrix} 0 \\ 2\lambda^{(i)} \end{bmatrix}$$

$$\delta^{(i)} = \begin{bmatrix} \frac{1}{2c_A^{2(i)}} & 1-\frac{1}{2c_A^{2(i)}} \end{bmatrix}$$

Az igények által a rendszerbe hozott munka szintén két paraméterrel adott: a várható értékével $m_l^{(i)}$, és a relatív szórásnégyzetével $c_l^{(i)}$. Ezekből a paraméterekből kiszámoljuk a kiszolgálási időt teljes kiszolgálási kapacitást feltételezve:

$$\mu_f^{(i)} = \frac{C}{m_l^{(i)}}, \quad c_{Sf}^{2\ (i)} = c_l^{2(i)},$$

és csökkentett kiszolgálási kapacitást feltételezve:

$$\mu_r^{(i)} = \frac{w_i}{\sum_i w_i}\frac{C}{m_l^{(i)}}, \quad c_{Sr}^{2\ (i)} = c_l^{2(i)},$$

ahol C jelöli a kiszolgáló kapacitását, és w_i jelöli az i igényosztály súlyát. A fentiekhez hasonlóan PH eloszlást illesztünk a kiszolgálási időkre is:

$$S_f^{(i)} = \begin{bmatrix} -\frac{\mu_f^{(i)}}{c_{Sf}^{2\ (i)}} & \frac{\mu_f^{(i)}}{c_{Sf}^{2\ (i)}} \\ 0 & -2\mu_f^{(i)} \end{bmatrix}, \quad s_f^{(i)} = \begin{bmatrix} 0 \\ 2\mu_f^{(i)} \end{bmatrix}$$

$$\sigma_f^{(i)} = \begin{bmatrix} \frac{1}{2c_{Sf}^{2\ (i)}} & 1-\frac{1}{2c_{Sf}^{2\ (i)}} \end{bmatrix}.$$

A csökkentett kapacitáshoz tartozó kiszolgálási idők PH leírása ($S_r^{(i)}$, $s_r^{(i)}$, $\sigma_r^{(i)}$) hasonlóan számítható.

2.2. Az aktív periódus

Vessünk újra egy pillantást a 3. ábrára. Az érkezésekkel, illetve kiszolgálási időkkel kapcsolatos átmenetek az előző fejezet eredményei alapján már ismertek. Az egyetlen hiányzó átmenet a 2-es osztály kiürülési ideje, vagyis az aktív periódusának a hossza. Ebben a fejezetben az aktív periódus két momentumát számítjuk ki, teljes, valamint csökkentett kiszolgálási kapacitás mellett, a másik igényosztály hatását figyelmen kívül hagyva.

Mivel a sorok érkezési és kiszolgálási idejének eloszlása fázis-típusú, ezért az egyenként vizsgált sorok Markov-láncának szerkezete speciális mátrix-tridiagonál, vagyis kvázi születés-halálozási folyamat:

$$Q = \begin{bmatrix} A_1' & A_0' & & & \\ A_2' & A_1 & A_0 & & \\ & A_2 & A_1 & A_0 & \\ & & \ddots & \ddots & \ddots \end{bmatrix}$$

A generátor mátrix blokkjait az érkezési és kiszolgálási folyamat paramétereiből az alábbi módon kapjuk meg [6]:

$$A_{0f}^{(i)} = d^{(i)}\delta^{(i)} \otimes I_{2x2}$$
$$A_{1f}^{(i)} = D^{(i)} \oplus S_f^{(i)}$$
$$A_{2f}^{(i)} = I_{2x2} \otimes s_f^{(i)}\sigma_f^{(i)}$$
$$A_{0f}^{(i)}{}' = d^{(i)}\delta^{(i)} \otimes \sigma_f^{(i)}$$
$$A_{1f}^{(i)}{}' = D^{(i)}$$
$$A_{2f}^{(i)}{}' = I_{2x2} \otimes s_f^{(i)}$$

(A redukált kapacitás melletti mátrixok hasonlóképpen számíthatók.)

Egy érkező igény által indított aktív periódus *k*-adik momentuma ($m_{Bf}^{k\ (i)}$) a következőképpen kapható meg:

$$m_{Bf}^{k\ (i)} = (-1)^k (\delta^{(i)} \otimes \sigma_f^{(i)}) \frac{d^k}{ds^k} G_f^{(i)}(s)|_{s=0}\ h,$$

ahol $G_f^{(i)}(s)$mátrix kielégíti a következő mátrix egyenletet:

$$sG_f^{(i)}(s) = A_{2f}^{(i)} + A_{1f}^{(i)} G_f^{(i)} s + A_{0f}^{(i)} (G_f^{(i)}(s))^2 .$$

A $G_f^{(i)}(s)$mátrix 0-dik deriváltja az *s=0* helyen az úgynevezett *fundamental matrix,* melynek minél hatékonyabb megoldásával számos cikk és könyv foglalkozik [6]. Az első derivált kiszámításához fix-pont iterációt használtunk (csupa 0 elemű mátrixból kiindulva) az alábbi alakra hozott egyenlet segítségével:

$$\frac{d}{ds} G_f^{(i)}(s)|_{s=0} = \left(A_{1f}^{(i)} - A_{0f}^{(i)} G_f^{(i)}(0)\right)^{-1} \cdot \cdot \left(I - A_{0f}^{(i)} \frac{d}{ds} G_f^{(i)}(s)|_{s=0}\right) G_f^{(i)}(0)$$

Közelítésünkben az aktív periódus hosszának két momentumát fogjuk felhasználni. Az eddigi eredményeket összegezve ez a két momentum az alábbi módon számítható ki:

$$m_{Bf}^{1\ (i)} = -(\delta^{(i)} \otimes \sigma_f^{(i)}) \left(A_{2f}^{(i)} + A_{0f}^{(i)} G_f^{(i)}(0)\right)^{-1} h$$
$$m_{Bf}^{2\ (i)} = 2(\delta^{(i)} \otimes \sigma_f^{(i)}) \left(A_{2f}^{(i)} + A_{0f}^{(i)} G_f^{(i)}(0)\right)^{-1} \cdot \cdot \left(A_{0f}^{(i)} \frac{d}{ds} G_f^{(i)}(s)|_{s=0} - I\right) \cdot m_{Bf}^{1\ (i)}$$

Ebből a két momentumból kiindulva fázis-típusú eloszlást konstruálunk, hasonlóan, ahogy azt az érkezési és kiszolgálási idők esetében tettük. A kapott PH eloszlás paramétereit jelölje ($B_f^{(i)}$, $b_f^{(i)}$, $\beta_f^{(i)}$).

A 3. ábra Markov-láncának felépítéséhez még egy dolog hiányzik. Tudnunk kell, hogy az aktív periódus befejeztével milyen fázisban lesz az adott sor érkezési folyamata. Ez a következő egyenlet segítségével megkapható:

$$\alpha_f^{(i)} = (\delta^{(i)} \otimes \sigma_f^{(i)}) \cdot G_f^{(i)}(0) \cdot (h_2 \otimes I_{2x2}).$$

2.3. A sorbanállási modell

Ebben a fejezetben a 3. ábrán felvázolt, a rendszert az *i* igényosztály szemszögéből leíró Markov-láncot konstruáljuk meg. A másik igényosztályt *j*-vel fogjuk jelölni (vagyis ha *i*=1, akkor *j*=2, és fordítva). Amint az az ábrán látható, az állapottér két részre osztható, az alsó, és a felső állapotsorra.

A felső részben, ahol nincsen a rendszerben másik típusú igény, az *i* igényosztály a kiszolgáló teljes kapacitását használhatja. Az állapottérnek ezen a részén követni kell (1) az *i* igényosztály érkezési folyamatának a fázisát, (2) az *i* igényosztály kiszolgálási idejének a fázisát, végül (3) a *j* igényosztály érkezési folyamatának a fázisát. Mivel mindhárom esetben 2 fázisú eloszlásról van szó, ez eddig 8 állapotot jelent.

Ha megjelenik a *j* típusú igény a rendszerben, a kiszolgáló kapacitása arányosan megoszlik, és az *i* igényosztálynak csökkentett kapacitás jut. A *j* típusú igény megjelenése ezért a Markov-láncot az állapottér alsó részébe viszi, ahol a kiszolgálás lassabb sebességgel történik. Ebben a részben követni kell az (1)-es és (2)-es fázisokat, de a harmadik nyomon követendő dolog a *j* igényosztály aktív periódus eloszlásának a fázisa lesz. Ez ismét 8 állapotot jelent. Így a teljes Markov-láncnak összesen 16 állapota lesz.

Már a 3. ábrából is látható, hogy ismét kvázi születési-halálozási folyamatot kapunk, melynek mátrixblokkjai – az eddig leírtak alapján – eképpen állnak össze:

$$C_0^{(i)} = \begin{bmatrix} d^{(i)}\delta^{(i)} \otimes I \otimes I & 0 \\ 0 & d^{(i)}\delta^{(i)} \otimes I \otimes I \end{bmatrix}$$
$$C_1^{(i)} = \begin{bmatrix} D^{(i)} \oplus D^{(j)} \oplus S_f^{(i)} & I \otimes d^{(j)}\beta_r^{(j)} \otimes I \\ I \otimes b_r^{(j)}\alpha_r^{(j)} \otimes I & D^{(i)} \oplus B_r^{(j)} \oplus S_r^{(i)} \end{bmatrix}$$
$$C_2^{(i)} = \begin{bmatrix} I \otimes I \otimes s_f^{(i)}\sigma_f^{(i)} & 0 \\ 0 & I \otimes I \otimes s_r^{(i)}\sigma_r^{(i)} \end{bmatrix}$$

Az irreguláris 0. szint mátrixai a következők:

$$C_0^{(i)}{}' = \begin{bmatrix} d^{(i)}\delta^{(i)} \otimes I \otimes \sigma_f^{(i)} & 0 \\ 0 & d^{(i)}\delta^{(i)} \otimes I \otimes \sigma_r^{(i)} \end{bmatrix}$$
$$C_1^{(i)}{}' = \begin{bmatrix} D^{(i)} \oplus D^{(j)} & I \otimes d^{(j)}\beta_f^{(j)} \\ I \otimes b_f^{(j)}\alpha_f^{(j)} & D^{(i)} \oplus B_f^{(j)} \end{bmatrix}$$
$$C_2^{(i)}{}' = \begin{bmatrix} I \otimes I \otimes s_f^{(i)} & 0 \\ 0 & I \otimes I \otimes s_r^{(i)} \end{bmatrix}$$

Mivel a fázisok száma kicsi (szintenként 16 állapot), a klasszikus QBD megoldó algoritmusok nagyon gyorsan (1 másodpercen belül) képesek kiszámítani a teljesítményjellemzőket, közöttük az esetünkben fontos várakozási idő momentumokat [6].

3. Numerikus eredmények

Hogy az eredmények használhatóságát bemutassuk, két példát állítottunk össze. Az első esetben a két igényosztály azonos mennyiségű munkát hoz a rendszerbe,

míg a másodikban a 2-es típusú igények tízszer annyi kiszolgálási időt igényelnek, mint az 1-es típusúak.

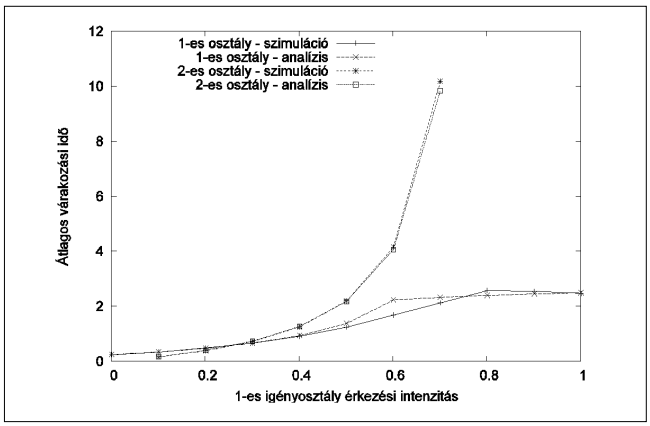
A példákban bemutatott görbék nem csak a cikkben ismertetett analitikus eljárás eredményeit tartalmazzák, hanem a szimulációval kapott eredményeket is, így lemérhető a közelítő eljárás pontossága.

1. példa

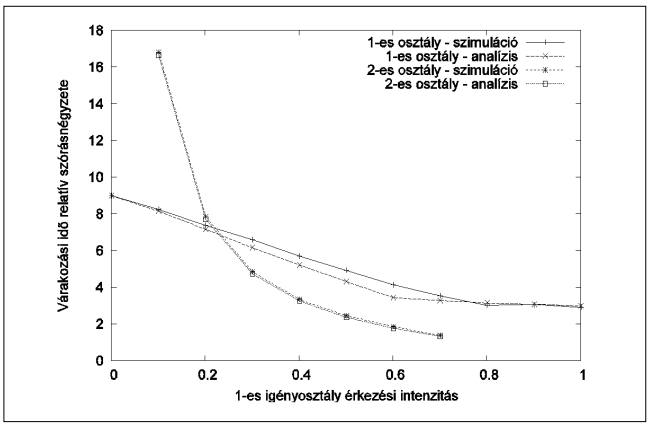
A *4. és 5. ábrán* a várakozási idő várható értékét és szórását ábrázoltuk, miközben az 1-es osztály terhelését (forgalmi intenzitását) növeltük. Amint várható volt, a terhelés növekedésével a várható várakozási idő nő, a relatív szórásnégyzet csökken.

A *6.-9. ábrákon* azt ábrázoltuk, hogy milyen hatással van a várakozási időre, ha a kiszolgálási idő, vagy az érkezési idő szórását növeljük.

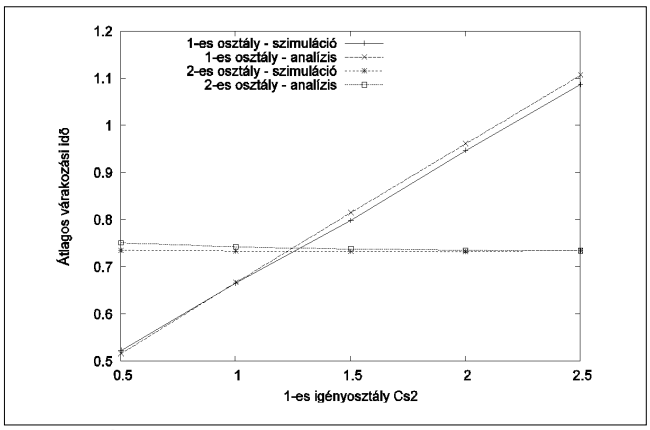
A bemutatott ábrák alapján elmondható, hogy a szórások növelése csak minimális mértékben van hatással a másik igényosztály vizsgált teljesítményjellemzőire.



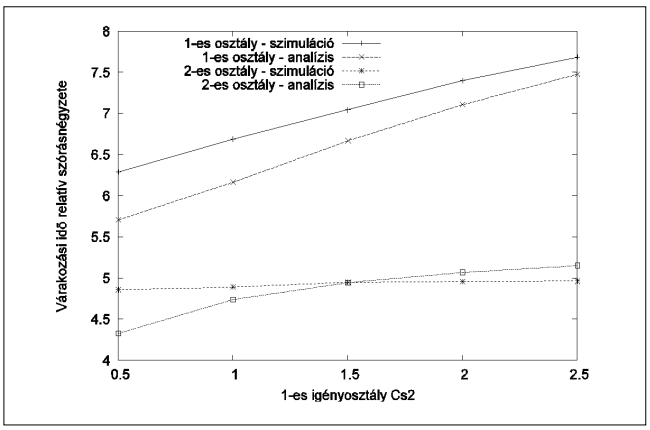
4. ábra Átlagos várakozási idő az érkezési intenzitás függvényében



5. ábra Várakozási idő relatív szórásnégyzete az érkezési intenzitás függvényében

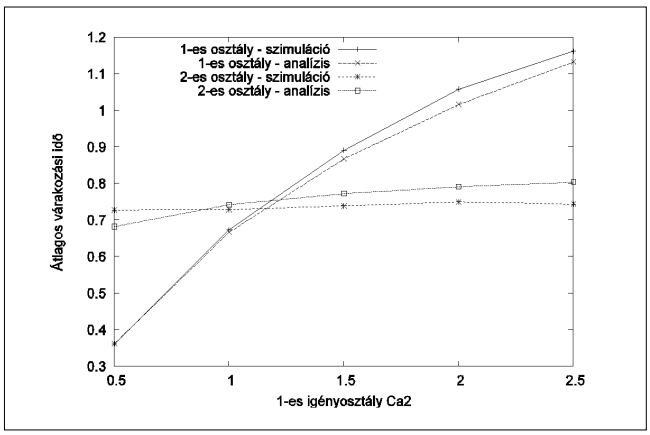


6. ábra Átlagos várakozási idő a kiszolgálási idő relatív szórásnégyzetének a függvényében

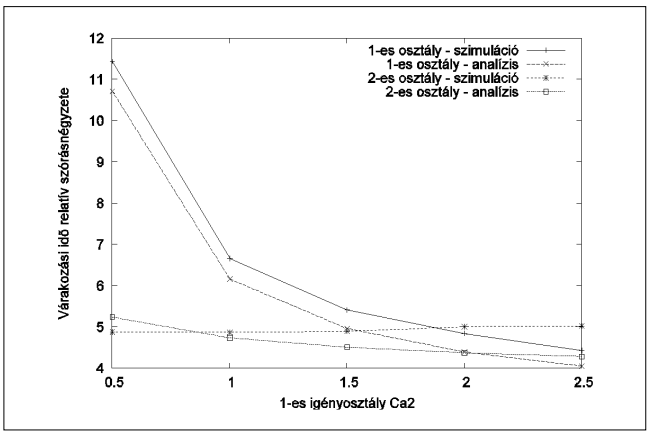


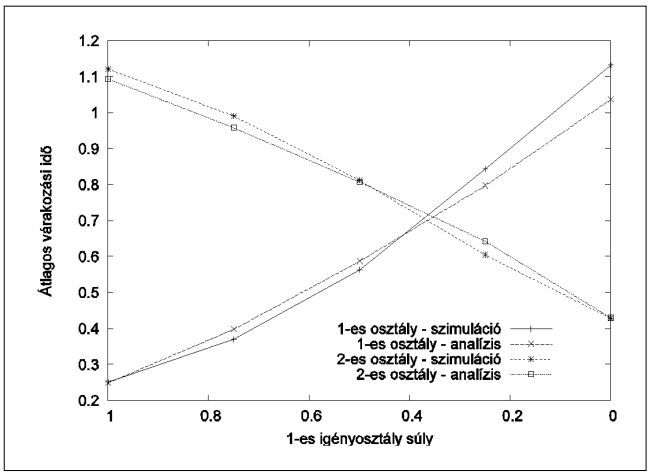
7. ábra Várakozási idő relatív szórásnégyzete a kiszolgálási idő relatív szórásnégyzetének függvényében

8. ábra Átlagos várakozási idő az érkezési időközök relatív szórásnégyzetének a függvényében

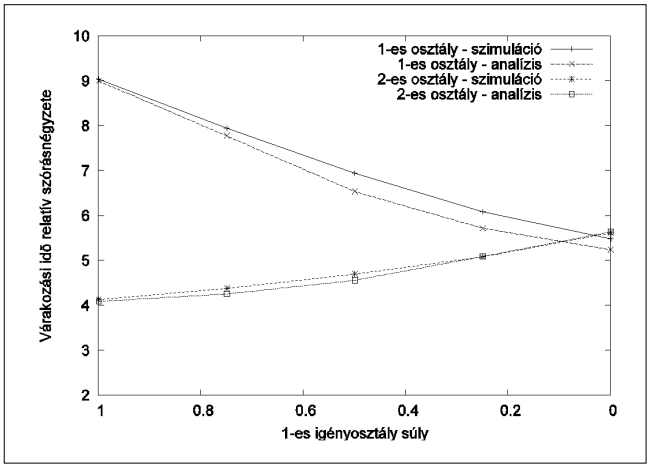


9. ábra Várakozási idő relatív szórásnégyzete az érkezési időközök relatív szórásnégyzetének a függvényében





10. ábra Átlagos várakozási idő a forgalmi osztály súlyának függvényében



11. ábra Várakozási idő relatív szórásnégyzete a forgalmi osztály súlyának függvényében

2. példa

Ebben a példában a 2-es típusú igények tízszer annyi kiszolgálási időt igényelnek, mint az 1-es típusúak.

A *12. és 13. ábra* a várakozási idő alakulását mutatja a terhelés függvényében. Látható, hogy még a 2-es osztály túlterhelése esetén is az 1-es típusú igények megkapják a garantált, súllyal csökkentett kiszolgáló kapacitást, így a várakozási idő nem nő jelentősen.

A *14.-17. ábrák* az érkezési és kiszolgálási idők szórásának hatását szemléltetik.

Ismét látható, hogy a 2-es típusú igények várakozási idejét nem befolyásolja az 1-es osztály érkezési és kiszolgálási idejének szórása. Igaz ugyan, hogy az analitikus megoldás mutat egy kis összefüggést, de a szimuláció görbéjétől való távolság még mindig elfogadható.

A *18. és 19. ábrán* a várakozási idő két paraméterét a súlyok beállításának függvényében ábrázoltuk.

A várható várakozási idő közelítése jó, de a relatív szórásnégyzet esetén itt tapasztaltuk a legnagyobb különbséget a szimuláció és az analízis között (kb. 20%).

4. Összefoglalás

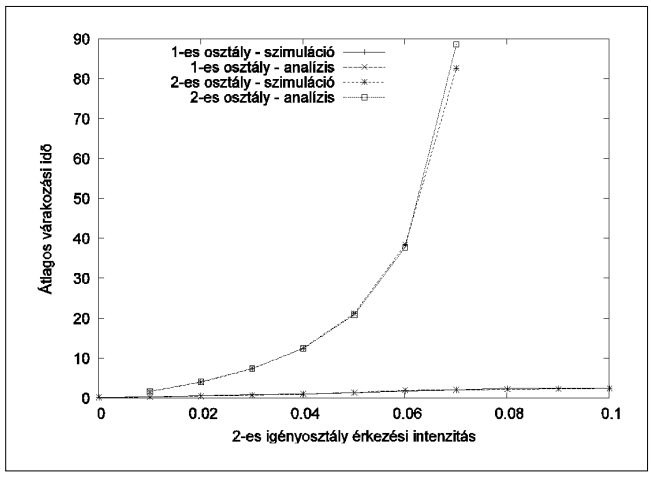
Ebben a cikkben egy közelítő eljárást ismertetünk WFQ rendszerek teljesítményvizsgálatára. A közelítés elvének bemutatása után klasszikus mátrix-geometriai eszközökkel oldottuk meg a felmerülő sorbanállás-elméleti problémákat.

Módszerünk előnye, hogy nagyon kicsi a számításigénye, továbbá hogy az irodalomban látott korábbi megoldásoknál általánosabb, mivel az érkezési és kiszolgálási időközöknek nem csak a várható értékét, hanem a szórását is figyelembe veszi.

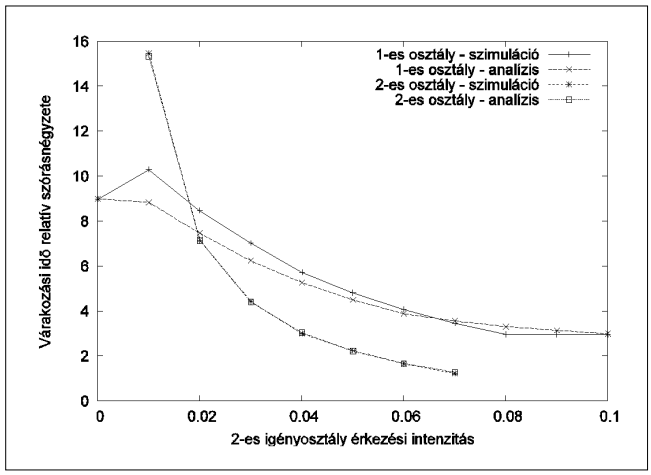
Két numerikus példán keresztül kimerítően vizsgáltuk az eljárás pontosságát, minden lehetséges paraméter függvényében. Az esetek nagy részében az eredmények nagyon jól közelítették a szimuláció eredményeit.

A legnagyobb eltérést (15-20%) a várakozási idő relatív szórásnégyzetében tapasztaltuk, a várakozási idő várható értékére azonban minden esetben jó közelítést kaptunk.

12. ábra Átlagos várakozási idő az érkezési intenzitás függvényében



13. ábra Várakozási idő relatív szórásnégyzete az érkezési intenzitás függvényében



Irodalom

[1] G. Koole,
"On the power series algorithm" (1994)
Tech. Rep., Centrum voor Wiskunde en Informatica

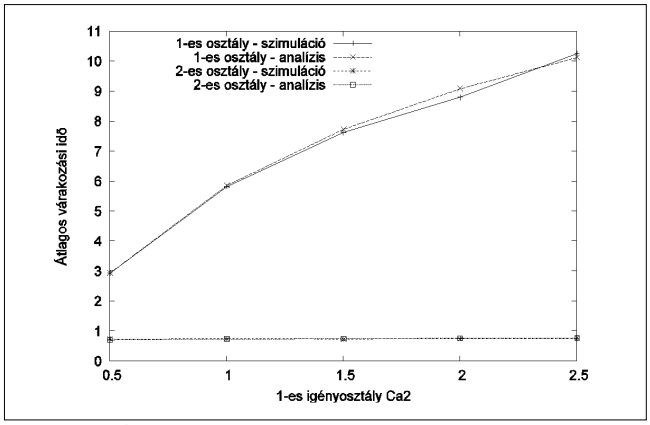
[2] J. P. C. Blanc (1988)
"A numerical study of the coupled processor model",
in Computer Performance and Reliability

[3] Leslie D. Servi (2002)
"Algorithmic solutions to two-dimensional birth-death processes with application to capacity planning",
Telecom. Systems, Vol. 21, No.2, pp.205–212.

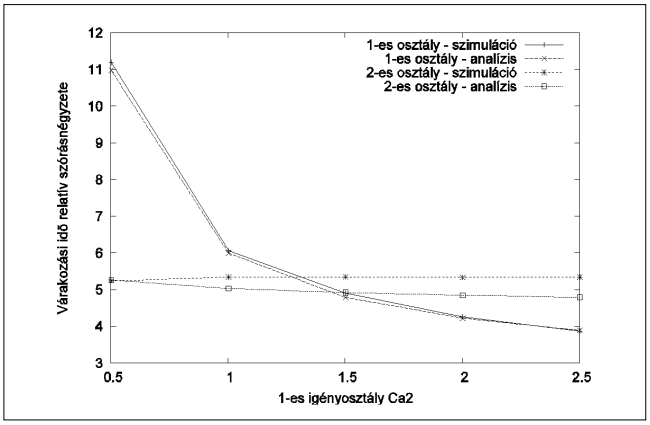
[4] F. Guillemin, R. Mazumdar, A. Dupuis, J. Boyer (2003)
"Analysis of the fluid weighted fair queueing system",
J. Appl. Probab., Vol. 40, No.1, pp.180–199.

[5] G. Fayolle, R. Iasnogorodski, V. Malyshev (1999)
Random Walks in the Quarter Plane,
Springer-Verlag New York

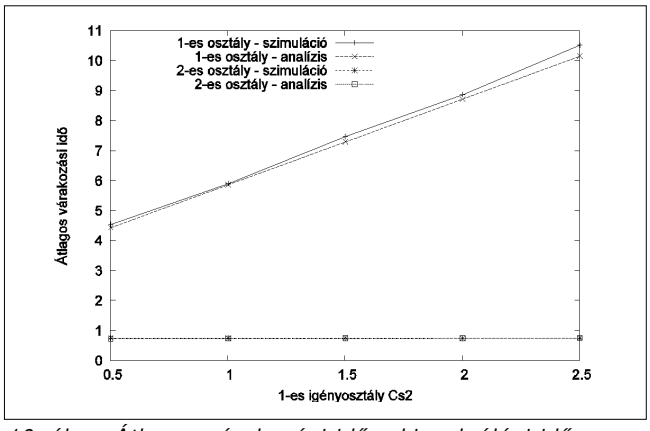
[6] G. Latouche, V. Ramaswami (1999)
Introduction to Matrix Analysic Methods in Stochastic Modeling,
American Statistical Association and the Society for Industrial and Applied Mathematics



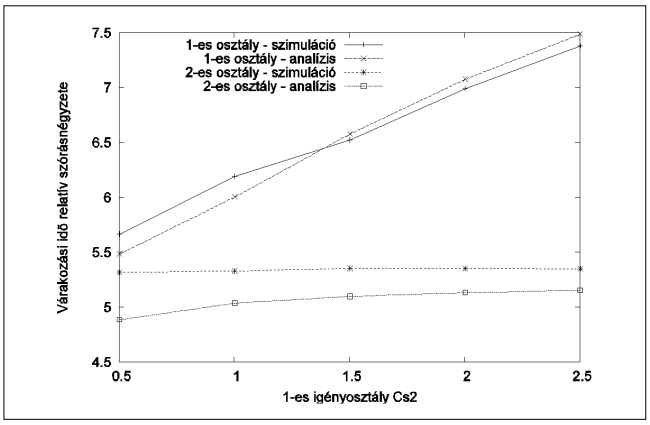
14. ábra Átlagos várakozási idő az érkezési időközök relatív szórásnégyzetének a függvényében



15. ábra Várakozási idő relatív szórásnégyzete az érkezési időközök relatív szórásnégyzetének a függvényében

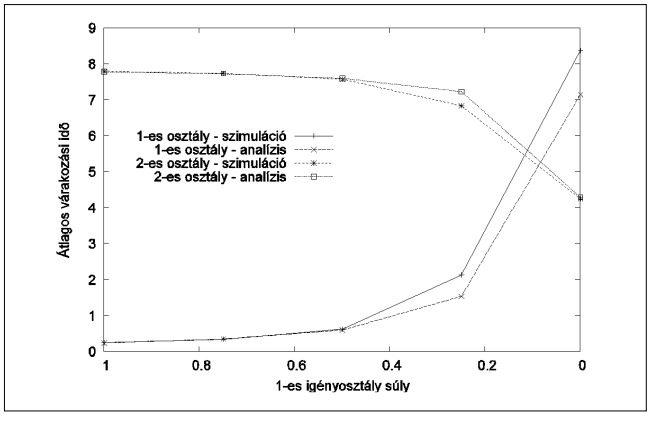


16. ábra Átlagos várakozási idő a kiszolgálási idő relatív szórásnégyzetének a függvényében

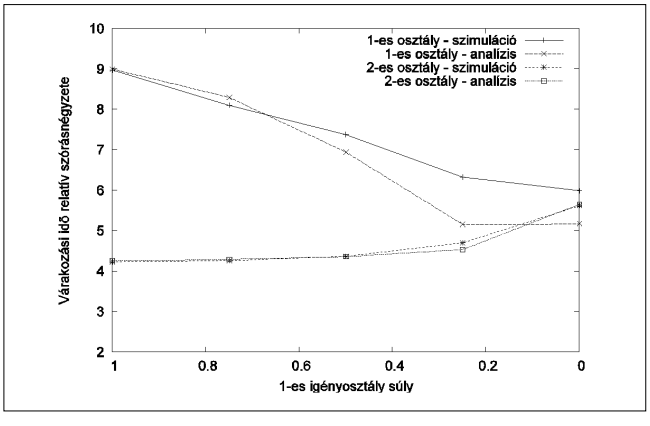


17. ábra Várakozási idő rel. szórásnégyzete a kiszolg. idő relatív szórásnégyzetének a függvényében

18. ábra Átlagos várakozási idő a forgalmi osztály súlyának függvényében



19. ábra Várakozási idő relatív szórásnégyzete a forgalmi osztály súlyának függvényében



A biztonságos információkezelés (secure processing) alapjai

SZŐLLŐSI LORÁND, MAROSITS TAMÁS

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
Nagysebességű Hálózatok Laboratórium
{szollosi, marosits}@ttt-atm.tmit.bme.hu

Kulcsszavak: informatikai biztonság, secure processing, titkosítás, doboz-struktúráltság

A megbízható, bizalmas feldolgozás technikai háttere kulcskérdéssé vált. Annál is inkább foglalkoznunk kell ezzel, mivel a digitális hordozó közegeken a szerzői jogok tarthatósága megbízható platform és eszközkészlet nélkül megkérdőjeleződik. Ugyanakkor problémát jelent a felhasználói jogok (anonimitás, biztonsági másolat, forrás hitelességének ellenőrizhetősége stb.) biztosítása is. A matematikai háttér adott, a kvantumprocesszorok korának beköszöntéig gyakorlati válasz adható az elméleti kérdéskörre. Cikkünkben magas szintről indulva szeretnénk összefoglalni a secure processing elméleti kérdéseit és útmutatást adni olyan konkrét rendszerek kidolgozásához, melyek alkalmazása esetén egyik fél érdekei sem sérülnek.

1. Bevezetés

A biztonságos feldolgozást nyújtó rendszerek kutatásának egyik legfontosabb mozgatórugóját a DRM (Digital Rigths Management) rendszerek jelentik. A témakör iránti érdeklődést jól mutatja, hogy a Conference on Communications and Multimedia Security [6] három vitaindító előadása közül az egyik a DRM rendszerek architektúrájával foglalkozik.

Tágabb és nehezebben kezelhető terület a PC alapú biztonságos környezet megteremtése, erre egy – sokat vitatott – megoldás a TCPA/TPM. [7] bemutatja a TPM egy, a felhasználók érdekeit előtérbe helyező lehetséges esetét: egy, a felhasználó személyi adatait és pénzügyi információit kezelő, a világhálóra kapcsolt szerver esetén kívánatos, hogy ez a szerver igazolni tudja a felhasználó felé az adatkezelési szabályzatban foglaltak betartását. Ugyanez a jelentés bemutat több elméletileg kivitelezhető támadást (többek között visszajátszásos támadást) a jelenlegi rendszer ellen. Arbaugh [8]-ban bemutatja a jelenlegi TCPA specifikáció olyan kibővítésének lehetőségét, mely a személyiségi jogokhoz kapcsolódó érdekeket is figyelembe veszi.

Ritter széleskörű gyűjteményét adja a felhasznált elméleti alapoknak, egyúttal közöl több, a témába vágó szabadalmi leírást [5].

2. Klasszikus információ és új követelmények

2.1. A klasszikus információ tulajdonságai

Témánk keretein belül információnak nevezünk minden olyan adatot, amely jelentéssel, jelentőséggel bír, így feldolgozásra alkalmas. A klasszikus információ teljesen más tulajdonságokkal rendelkezik, mint a klasszikus anyag. Így az arra vonatkozó törvényi szabályozásnak egészen másnak kell lennie, különben a jogbiztonság kérdésessé válik.

Az információ tulajdonságai:

1. *Oszthatatlan egységekre bontható,* melyek további darabolása vagy fizikailag lehetetlen, vagy az adat általa elveszti eredeti tartalmát, jelentését és így már nem tekinthető információnak. Minden általunk tekintett információ véges sok részre osztható.
2. Tökéletesen *replikálható* (lemásolható) anélkül, hogy az eredeti példány használhatósága romlana.
3. *A replikáció* ténye *nem bizonyítható* és *nem cáfolható*.
4. Önmagában nem azonosítja a forrást, a felhasználót, nem hitelesíti az adat integritását, nem hordozza a saját történetét; azaz a hordozó struktúra *keretezése memóriamentes*.

Fontos megjegyeznünk, hogy ezen tulajdonságok nem korlátozzák az informatikát – így nem jelentenek korlátokat az informatika felhasználhatóságára sem –, csak egy konkrét technológia ismérvei. Ezen jellegzetességek miatt azonban korunk informatikai rendszereiben a biztonság és az adatkezelés bizalmassága nem mindig tartható követelmények.

2.2. Alkalmazások és követelmények

Kényelem: Felhasználói szemmel nézve a legfontosabb döntési szempont, hogy az eszköz használata ne legyen bonyolult, a felhasználó ne ütközzön nehézségekbe. Nem tételezhető fel, hogy egy kényelmetlen rendszert bárki hosszú távon használni fog, legyen az bármennyire biztonságos.

Anonimitás: Két irányban vizsgálhatjuk: egyrészt bizonyos esetekben a felhasználó (például mint vásárló) rejtve akar maradni a tulajdonos/szolgáltató (például eladó) előtt. Másrészt sokkal általánosabb probléma, hogy harmadik fél ne szerezzen tudomást a tranzakció paramétereiről, különösen a felhasználóhoz kapcsolható bizalmas jellegű adatokról.

Tranzakciók letagadhatatlansága: Mindkét fél érdeke, hogy a tranzakciók letagadhatatlanok legyenek. Ehhez nem elég, hogy a felek kölcsönösen ellenőrizni

tudják, hogy a másik jóváhagyta-e a szóbanforgó tranzakciót, arra is szükség van, hogy ezt harmadik fél felé bizonyítani tudják titkos információ kiadása nélkül. A nyilvános kulcsú aláírás erre tökéletesen megfelelő eljárást ad.

Áteresztőképesség: A rendszer egyes elemeinek, például egy kliens-szerver modellben a szervergép(ek)nek jelentős terhelést kell elviselniük, ezáltal gyakran szűk keresztmetszetet jelenthetnek. Az áteresztőképesség a végrehajtható tranzakciók számát korlátozza, túllépése rosszabb esetben rendszerösszeomlást okozhat. Profitorientált szereplők ezért csak megfelelően nagy áteresztőképességű rendszerek alkalmazásában érdekeltek.

Költség: Elsősorban az egy tranzakcióra eső költség a döntő. A szolgáltató ezt vagy közvetlenül a felhasználóra terheli (növelve ezzel az árat, csökkentve a keresletet), vagy az előre megszabott árból próbálja kigazdálkodni a tranzakció költségét. A tranzakciós költség gazdasági megszorítást jelent egy rendszer alkalmazhatóságára és a technológia fejlődésével gyorsan csökken.

Tranzakciós adatok integritása: A információ sértetlenségének biztosítása nélkül a szolgáltató jogi kötelezettségeinek sem tud eleget tenni, például nem tud könyvelni. Szerencsére általában az integritás-problémák még a hibásan tervezett rendszerek esetén is a tranzakciók egy kis hányadát érintik; ezek kezelése azonban hosszú időt vehet igénybe és a rendszerbe vetett bizalmat is megingathatja (gondoljunk csak a bankkártyaszám-lopásokra vagy a szavazat-újraszámlálásra).

Az információ tulajdonságai adottak, azokat nem tudjuk megváltoztatni. Tudunk azonban olyan reprezentációt, olyan protokollokat kifejleszteni, amelyek során adott információcsomagok feldolgozása a kívánt működést mutatja, beleértve a biztonsági követelmények teljesítését is. Több ilyen rendszert láthatunk a gyakorlatban működni, melyek tulajdonságait a lenti *táblázat* foglalja össze.

A TCPA (Trusted Computing Platform Alliance) a Microsoft, Intel, IBM, HP és AMD szövetsége melynek célja a gyártók és fejlesztők számára biztonságosabb PC létrehozása. Ennek elérése érdekében egy erős kriptográfiát használó ellenőrző chipet építenének először az alaplapra, majd a processzorba. Sajnos a technológia felhasználók számára az eddiginél is kevesebb kontrollt biztosít a számítógépük felett. Bővebb információ a technológiáról az irodalomban [2] olvasható.

2.3. Az informatikával szemben támasztott új követelmények összegzése

Az előző szakasz alapján látható, hogy az informatikával szemben mind az új piacok, mind a környezet radikálisan új igényeket támasztanak. Ugyanakkor a jelenlegi informatikai megoldások a fenti problémákat csak külön-külön kezelik, vagyis minden, a biztonsággal vagy a bizalmassággal kapcsolatban felmerülő kérdésre egyedi válaszaink vannak. Szükség van egy olyan integrált megoldásra, amely egyszerre képes az összes felmerülő kérdésre megnyugtató választ adni. Feltételezhetjük, hogy egy ilyen integrált megoldás olcsóbb, egyszerűbben menedzselhető és kisebb erőforrásigényű lehet, mint az egyedi megoldások összessége.

Biztonsági követelmények teljesülése a különböző alkalmazásoknál

Rendszer	Használók érdekeinek teljesülése				Szolgáltatók vagy jogtulajdonosok érdekeinek teljesülése			
	Kényelem	Anonimitás tulajdonosok felé	Anonimitás harmadik fél felé	Tranzakciók letagadhatatlansága	Áteresztőképesség	Költség	Tranzakciós adatok integritása	Tranzakciók letagadhatatlansága
Home banking	+	–	–	+(?)	+	+	+	+
Internetes választások	–	–	+	–	+	–	+(?)	+/–
Jegyrendelés	+	–	+	–	+	–	–	–
Internetes vásárlás bankkártyával	+	–	+/–	–	+/–	–	+/–	+/–
TCP/TPM	+	–	–	–	–	+	+	+
Egyedi processzorazonosító	+	–	–	–	–	+	–	–

Jelmagyarázat:
+: Az adott érdek teljesülése a jelenlegi rendszerekben biztosított
–: Az adott érdek teljesülése a jelenlegi rendszerekben nem biztosított
+/–: Egyes rendszerekben nem, másokban biztosított az adott érdek teljesülése
+(?): Az adott érdeket a rendszer figyelembe veszi, de kérdéses vagy ellenőrízhetetlen a teljesülése

Ugyanakkor lehetséges, hogy bizonyos esetekben ennek az integrált rendszernek nem tudjuk az összes lehetőségét kihasználni és bevezetése nem gazdaságos.

Az új követelmények több kategóriába sorolhatóak. Egyrészt fontos egy új információfogalom (melyet a továbbiakban SPI-nek – secure processing information – nevezünk) kialakítása, amely valamilyen anyagszerűen kezelhető entitást kell fedjen. Ennek kapcsán az alábbi feltételeknek való megfelelést kell vizsgálnunk:

- Az információ véges sok, tovább nem osztható egységből kell álljon.
- Az információ *képes legyen magát azonosítani* és az *integritása* biztosított legyen. (Azaz „látszik", hogy mit tartalmaz.)

Másrészt olyan tulajdonságokat is megkövetelünk, melyeket az anyaggal kapcsolatban nem a fizika, hanem a jogrend biztosít a felhasználóknak, azaz

- Definiáljon egy *interfészt*, melyen át – és csak azon át – elérhető és feldolgozható az információ. (adatvédelemmel kapcsolatos törvények hétköznapokban)
- A *replikációjára tett kísérletek eredménytelenek* legyenek mindaddig, amíg ezt az interfész nem engedélyezi. (Lopás vagy visszajátszásos támadás elleni védelem.)
- *Forrása és összes módosítója azonosítható* legyen. (Legyen története, vagyis a keretszerkezet rendelkezzen memóriával. Hétköznapi példa erre a könyvelés.)

Az egyes szempontok persze különböző mértékben fontosak az informatika különböző szereplőinek, mivel azok különböző nézőpontokból vizsgálják a kérdéskört. Alapvetően három érdekcsoportot tudunk elkülöníteni:

- A *szellemi termékek* (szoftverek, tervek) *szerzőijog tulajdonosainak* érdekeit kell biztosítanunk. Erre megfelelő, ha egyrészt a *felhasználó bizonyítja jogosultságát* (de nem azonosítja saját magát) az SPI felé, másrészt csak meghatározott interfészen át érhetjük el a programokat.
  Program alatt a továbbiakban olyan információt értünk, melyet a felhasználónak joga van futtatni. Természetesen a felhasználó nem feltétlen pusztán a gép előtt ülő személy, hanem lehet egy másik program is (például adatbetöltés), vagy másik információ (például hivatkozás). Ezeknek a programoknak és információkereteknek is át kell esniük egy hasonló azonosítási procedúrán.
- A *rendszerüzemeltetők* (rendszergazdák, adminisztrátorok) célja a hatékony karbantartás lehetőségének a fenntartása. Szükségük van valamilyen eszközre, amivel a felhasználók jogosultságait állíthatják. Adott esetben szükséges lehet olyan jogosultságok kiadására is egyes felhasználók számára, amelyekkel a rendszergazda maga sem rendelkezik, tehát a rendszergazda nem csak a saját jogait engedheti át másoknak; továbbá a saját jogkörét nem tágíthatja.
  Fontos a felhasználók által végzett műveletek naplózása, követése.
- A *felhasználók* azt szeretnék, ha a rendszer iránt érzett – és a korábbi rendszerek használata során már megszokott – komfortérzetük nem csökkenne, ugyanakkor nem akarnak az információgyűjtés áldozatává sem válni. Ezt azért sem lenne szerencsés, mivel így monopóliumhelyzettel való visszaélés válik lehetségessé (információ programhoz kötése). Továbbá így – különösen üzleti felhasználók esetén – titkos adatok, információk megszerzése lenne lehetséges. Egyes felhasználócsoportok (például cégek, vagy azok osztályai) tovább szeretnék szűkíteni, személyre szabni a tagjaik számára elérhető funkciókat.

Látható, hogy a szereplők érdekei eltérőek. Ezt a helyzetet felfoghatjuk egy új evolúciós kalitkának. Evolúciós kalitkának nevezünk minden olyan környezetet, ahol a mintahalmaz (jelen esetben az informatikai megoldások halmaza) egy eleme el tud helyezkedni anélkül, hogy versenyhelyzetbe kerüljön valamely társával. Biológiai környezetünkben az új fajok kialakulását eredményezi egy evolúciós kalitka megnyílása, hasonlóan az informatikában új technológiák jelenhetnek meg. A jövő informatikájának tényleges szabályait az fogja meghatározni, hogy az informatika szereplőiből létrejövő érdekcsoportok mennyire tudják érvényesíteni elvárásaikat; egyáltalán: milyen hamar kapcsolódnak be ennek a szabályrendszernek a kialakításába.

3. A jelenlegi eszközök felhasználhatósága

A jelenleg rendelkezésünkre álló informatikai eszközkészlet (matematikai törvényszerűségek, algoritmusok, számítógépes technológiák, konkrét alkalmazások) a fentiekben vázolt problémák egy részét már képes megoldani. Ezek közül a legfontosabbak:

1. **Hashképzés (MD5, SHA1)**: Hashfüggvénynek nevezünk egy olyan dimenziószűkítő függvényt, mely könnyen számolható, de nehezen invertálható, és nehéz két azonos függvényértéket eredményező bemenetét találni. Az MD5 hash ennek tipikus példája [1,3]. A lenyomatkészítő függvényeket azért alkalmazzuk, hogy általuk az adat szándékos vagy véletlen módosítása felismerhető legyen. Azaz, ha az információnk MD5 hash-e rendelkezésre áll, ellenőrizhetjük annak *integritását*. Ez azonban nem nyújt módot a forrás azonosítására. Általában aláírás előtt alkalmazzák, hogy kisebb méretű szövegen kelljen végrehajtani a számításigényes RSA algoritmust.
2. **Nyilvános kulcsú aláírás (RSA, DSA)**: A nyilvános kulcsú aláírás lényege, hogy minden szereplő kap egy hitelesítő központtól egy kulcspárt: E (publikus) és D (titkos) kulcsokat, melyek – mint függvények – egymás inverzei, továbbá E-ből D nehezen számítható. Az E kulcsot nyilvánosságra hozza a résztvevő, míg a D kulcsot titokban tartja. Ez utóbbit használja aláírásra. (Az algoritmus pontos leírása meg-

található [3]-ban). Bárki, aki birtokában van az adott személy nyilvános kulcsának, ellenőrizni tudja, hogy egy információ tőle érkezett-e, tehát ez az eljárás lehetőséget nyújt a *hitelesség* biztosítására is. Az elektronikus aláírás egy további problémát is megold, nevezetesen a *letagadhatatlanságot*: azaz, ha valaki egyszer aláírt egy dokumentumot, akkor többé nem tudja letagadni az aláírás tényét, mert bárki könnyedén tudja ellenőrizni a dokumentumot. Felmerül, hogy hogyan tudjuk biztosítani az E kulcs hitelességét.

Erre megoldást nyújt egy mindenki által ismert CA, azaz hitelesítő központ, mely a kulcs eredetiségét és tulajdonosának kilétét saját aláírásával garantálja. Vegyük észre, hogy bár eddig személyekről beszéltünk, valójában egy processzor is részt vehet az aláírásban, amennyiben képes a szükséges műveleteket elvégezni és aláírását titokban tartani. Azaz ellenőrizheti például a felhasználó processzora (vagy egy alkalmazás, amire már most is számtalan példát mondhatnánk az internetes banki rendszerektől a különböző plug-inekig.), hogy az általa kapott programot olyan személy vagy cég írta-e, akinek a programjait a felhasználó engedi futtatni; így pedig kiküszöbölhetőek a vírusok.

3. **Nyilvános kulcsú titkosítás (RSA)** [3]: Hasonló az elve az aláíráséhoz, csak itt az adatok *titkosságának* biztosítása a cél. Azaz a partner nyilvános kulcsával titkosítjuk az adatot, melyet csak ő tud majd dekódolni, mert csak ő ismeri a saját titkos kulcsát. A titkosság egyúttal azt is jelenti, hogy a kommunikáció *lehallgathatatlanná* válik, mivel a titkos információ (cyphertext) az azt lehallgató személy számára nem szolgáltat információt. Szintén lehet egy processzor is szereplője az eljárásnak, ekkor a processzormagba úgy juttathatunk kódot és adatot, hogy az külső felek (lehallgatók) számára értelmetlen. Ezzel megoldhatjuk a *másolhatatlanságot* is, ami az anyag és a klasszikus információ közötti legszembetűnőbb eltérés: ugyanis, ha a programot a fentiekben vázolt módon juttatjuk a processzorba, akkor azt hiába másoljuk le, nem tudja egy másik processzor végrehajtani.

   Természetesen a processzorunk lemásolhatja azt, illetve ha kiadja saját titkos kulcsát, akkor sem garantálható tovább a másolhatatlanság. Azonban, ha csak azon processzorok kaphatnak megfelelő minősítést, melyek ezen szabályokat betartják, és az információ előállítója kéri a processzortól ezt a megfelelőségi tanusítványt, mielőtt számára lekódolná az adatokat, akkor garantálható a zavartalan működés. Vegyük észre, hogy ezzel sem a felhasználó, sem a processzora nem kellett, hogy azonosítsa magát; pusztán a hozzáférés jogosságát igazolta a processzor.

Van azonban egy Achilles-sarka a jelenlegi technológiák csokránának: ez pedig a rosszindulatú programokkal szembeni védtelenség. Vírusok ugyan nem juthatnak a felhasználó gépére, de csak azt tudhatja jelenleg, hogy *kitől származik* a kód, amit futtat, és nem azt, hogy *mit csinál,* vagy hogy *mihez fér hozzá*. Olyan környezetet kell tehát alkotnunk, amelyben eleve lehetetlen a jogosultsági határokon adatokat átszivárogtató programok írása. Erre nyújt megoldást a javasolt hierarchikus jogosultsági szinteken alapuló, általunk „doboz-struktúrált”-nak nevezett információfeldolgozási környezet.

4. Doboz-struktúrált feldolgozási környezet

A hagyományos informatikai környezetekben megszoktuk, hogy a gyermek-folyamatok a szülőjüktől viszonylag független életet élnek: saját maguk foglalhatnak memóriát, írhatnak és olvashatnak állományokat, foglalhatnak le processzoridőt (rosszabb esetekben ez utóbbi automatikusan és megállíthatatlanul történik, működésképtelenné téve a rendszert).

Ha teljes biztonságot szeretnénk a felhasználó számára nyújtani, akkor ezt a programozó-barát megközelítést fel kell, hogy adjuk. Cserébe egyrészt jól lokalizálhatóvá válnak a nem megfelelően működő modulok, azaz amelyek nem felelnek meg az interfészüknek, vagy nem komformak a felhasználó szándékával, másrészt jól menedzselhetővé válnak a rendszer erőforrásai.

4.1. A doboz-struktúráltság feltételei

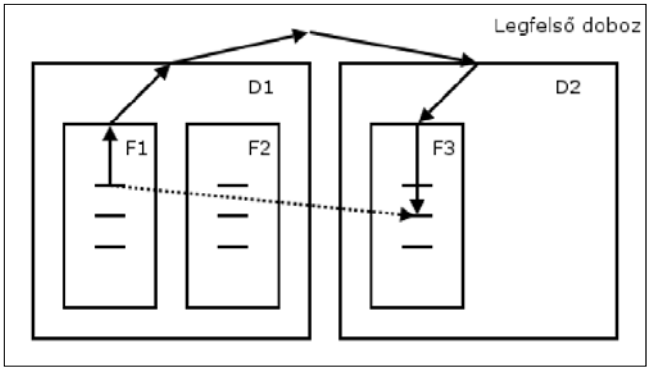
Tisztán doboz-struktúráltnak vagy SP-képesnek nevezünk egy rendszert, ha a következő feltételeket teljesíti:

1. A legfelső szinten egyetlen doboz áll.
2. Dobozai további, hierarchikusan egyenrangú dobozokra bonthatók, vagy végső dobozok, melyek a hierarchia alján állnak.
3. Egy doboz csak önmagán változtathat közvetlenül, az azt tartalmazó (szülője) és az általa tartalmazott (gyermeke) dobozokat specifikált interfészen át éri el.
4. Az interfészek tartalmazzák a függvények nevén, input és output paraméterein kívül a függvény gyártóját azonosító aláírást, továbbá a megvalósított szabvány (ha van ilyen) tanusítványát (ezáltal biztosítva, hogy a függvény valóban azt a funkciót látja el, amit a neve takar).
5. Alapértelmezésben a szülő minden lehetőségét öröklik a gyermekek, de a szülő ezt szűrheti.
6. Ha az interfész-specifikációban ellentmondás van a szülő és a gyerek között, akkor annak a szava a döntő, aki a feldolgozást végezni fogja. Tehát, ha a szülő hívja a gyerek egyik függvényét, akkor a gyereké, ha a gyermek a szülő egy függvényét, akkor a szülőé.

Ha a legfelső szinten SP-képes egység áll, akkor a rendszer SP-képes, tehát képes SPI kezelésére. Ehhez nemcsak az egyes gyártóknak kell a processzorok

kulcsainak védelmét biztosítaniuk, hanem gyártók közötti (és fölötti) összefogásra is szükség van a központ(ok) által kiadott tanusítványok karbantartása miatt. Ez nem jelenti természetesen a fejlesztési vagy technológiai információk cseréjét, az informatikai monokultúra kialakulását, csak a közös érdekek mentén történő egyeztetést (például szabványok, ajánlások megfogalmazása és betartásának ellenőrzése).

Lehetőség van akár több különböző CA használatára is, ha ezek között megfelelő szerződések jönnek létre, melyeknek természetesen anyagi vonatkozásai is lehetnek. Ekkor tulajdonképpen a pénzügyi-gazdasági életben megszokott viszontbiztosítást alkalmazzuk az informatikában. A jelenlegi CA-k esetében ezt keresztanusítványok kiállításával oldják meg.



Az *ábrán* látható, hogy a doboz-struktúrált folyamatok hívási modellje a hálózati rétegek egymásba ágyazásával analóg [4]. Az F1 folyamat látszólag közvetlenül üzen F3-nak, míg a háttérben ezt ellenőrzi F1, D1, a legfelső doboz, D2 és F3 interfésze. Ezért mondhatjuk, hogy az üzenetküldés biztonságos. Az SP-képes rendszert felfoghatjuk úgy is, hogy benne a folyamatok és a hozzájuk rendelt erőforrások egy hierarchikus fába szervezhetőek és az egyes folyamok csak ennek a fának az őket összekötő élein haladva érhetik el másik folyamatok erőforrásait. A fa gyökere a rendszer összes erőforrása, a levelei a folyamatok. Ahhoz, hogy a folyamatok erőforrásigényeikkel a fa ágaira kilépjenek, igazolniuk kell ehhez való jogukat.

4.2. A doboz-struktúráltság előnyei

A doboz struktúrában az összes erőforrás foglalásának lehetősége felülről lefelé adható tovább, tehát az SP-képes processzortól a processzek felé. Így az összes futó folyamat ellenőrzés alatt tartható. Legmagasabb szinten ezt az ellenőrzési lehetőséget az SP-képes processzor nyújtja, közvetlenül fölötte áll azonban a felhasználó, aki tilthatja, illetrve korlátozhatja egyes erőforrások elérését.

Fontos, hogy a dobozok között nincs átfedés, tehát egy erőforrás egy oszthatatlan atomja egyszerre csak egyetlen doboznál lehet. Ha mégis több blokkban van szükség a feldolgozására, akkor az információt a közös ősön át érhetik el. Ezáltal lehetővé válik az erőforrások korlátozott foglalása és felhasználása. Ez a mai rendszerekben még részlegesen van jelen és főleg a programok (illetve felhasználók) tárterületének egymástól való védelmét, valamint a kernel memória védelmét tartalmazza.

Nincs megoldva viszont általában például az időkeretek (processzoridő) védelme. Ennek köszönhetően egy féreg-program képes lehet pusztán osztódással a processzoridő legnagyobb részét lefoglalni, ami egy biztonságos rendszerben megengedhetetlen. Az általunk javasolt rendszerekben az időkeret (processzoridő) is ugyanolyan erőforrás, mint a többi, ugyanúgy kell foglalni, felszabadítani, és minden doboz csak az ő szülőjétől kérheti.

Ahhoz, hogy ne kelljen minden egyes szálnak processzoridőt kérnie magától a felhasználótól, nem kell más, mint a kontroll felületen belül egy újabb doboz, amely megkapja a rendszer szabad processzoridejét. Ha valamely processz többet akar foglalni, mint amennyi processzoridő még rendelkezésre áll, akkor a rendszernek a felhasználóhoz kell fordulnia a kontroll felületen keresztül, hogy döntse el, hogy a kérdéses folyamat kaphat-e a többi processz kárára többleterőforrást. Ezáltal zavartalan működés mellett biztosítható az összes erőforrás felügyelete és indokolt esetben elvétele.

5. Processzor és periféria az SP környezetben

Az informatikában a feldolgozásban résztvevő eszközöket két csoportba szokás sorolni: processzorok és perifériák. A perifériák szolgáltatják, tárolják és jelenítik meg az adatokat, a processzor(ok) végzi(k) a tényleges feldolgozást. Az SP környezetben ez a szétválasztás sokkal jelentősebbé válik, mivel biztosítani kell az információhoz való korlátozott hozzáférést. Erre csak az SP-képes processzorok és SP-képes perifériák képesek, a nem SP-képes eszközökkel való kommunikáció tehát nem minden esetben engedélyezett.

5.1. Processzor

Minden olyan egység, amely információ feldolgozására képes (azaz Turing-gépnek minősül), és részt vesz a feldolgozásban, szükségképpen SP-képes kell legyen. Azaz egyetlen SP-képes processzor sem adhat át SP igényű információt nem SP-képes processzornak, különben rés keletkezik a biztonsági hálón. A processzoroknak ezért igazolniuk kell egymás felé (a központtól kapott tanusítványukkal), hogy ők képesek SP-re, illetve az információnak biztonságosan (titkosítva) kell mozognia a két processzor között. A titkosítás történhet például nyilvános kulcsos rendszerben, vagy bármilyen más, azonosítást és titkosságot garantáló rendszerben. Nem SP-képes processzor csak olyan információt kaphat, amely nem igényel SP-et.

5.2. Periféria

Periféria minden egység, mely a processzorral kapcsolatba kerülhet, annak információt szolgáltat, vagy attól információt kér illetve fogad. A perifériák lehetnek

SP-képesek vagy nem SP-képesek; ameddig az SPI információ *feldolgozása* SP-képes processzorokban történik, addig nem jelent problémát egy vagy több nem SP-képes eszköz. Fontos azonban, hogy az információ *kódoltan* kerüljön a nem SP képes perifériákra (például winchester, hálózati eszköz), és ezt a kódot csak a címzett tudja dekódolni.

Ha a periféria tárolja az adatot, vagy abból bármilyen módon kinyerhető az, akkor nem elég, hogy a periféria interfésze SP-képes, a tárolt adat is titkosított kell legyen, hogy az SP környezet határfelületén belül maradjunk. Összefoglalva: SP-képes perifériából csak a címzett SP-képes eszköz nyerheti ki az információt még fizikai eszközök felhasználásával is, azonban nem SP-képes perifériák is jelen lehetnek a rendszerben mindaddig, amíg ezekben nem történik feldolgozás.

5.3. Amit a dobozstruktúrált feldolgozás sem képes garantálni

Mint minden technológia, az SP is csak egy adott környezetben, adott feltételezések fennállása mellett nyújt garanciákat; ezért meg kell határozni azokat a kockázati tényezőket, amelyek kívül esnek a SP által vizsgált és megoldott problémák körén.

Ezek a következők:

– A központ kulcsának visszafejtése (minimális kockázat);
– A központ hitelességének megkérdőjelezhetővé válása (minimális kockázat);
– Kvantumprocesszorok vagy bármilyen más, NP-teljes problémákat belátható idő alatt megoldó eszközök elterjedése (egyelőre nem várható, jól előrejelezhető az áttörés);
– Magát SP-képesnek mondó, de valójában nem SP-képes processzorok tanusítványszerzése (kis kockázat megfelelő processzor-ellenőrzés mellett);
– Magát SP-képesnek mondó, de valójában nem SP-képes perifériák tanusítványszerzése (nagyon kis kockázat megfelelő periféria-ellenőrzés mellett);
– Interfész és jogi lehetőségek (licensz) eltérése (ez esetben kívánatos olyan állami szintű jogi szabályozás, mely a licensznek megfelelő interfész készítését írja elő);
– A megjelenítési rétegből (például képernyőről, hangkábelről) nyert információ újradigitalizálása, és visszahelyezése az informatikai környezetbe, ezáltal SPI nem SP-képes környezetbe juttatása titkosítatlanul (egyedi megítélést igényel, de az információ minősége az újradigitalizálás következtében valószínűleg romlik, információ vész el).

6. Összefoglalás

Cikkünk arra vállalkozott, hogy feltárja az információkezelés és -feldolgozás jelenlegi módszerei és az informatika fejlődése következtében megjelenő új követelmények közötti ellentmondásokat. Megvizsgáltuk az új kihívásokra adott válaszokat, illetve a rendelkezésünkre álló eszköztárat.

A mai, csak egyes kiválasztott részproblémák megoldására használatos rendszerek helyett egy új paradigma, a *biztonságos információfeldolgozás* (secure processing) alkalmazását tartjuk célravezetőnek, mivel ezáltal az információforgalom valamennyi szereplőjének érdekei biztonságosan és gazdaságosan teljesíthetőek. A doboz-struktúrált feldolgozási környezet cikkünkben bemutatott jellegzetességei lehetővé teszik, hogy a következő évtizedek infokommunikációs rendszereinek meghatározó eleme lehessen.

Meg kell még jegyeznünk, hogy bár az SP széleskörű elterjedése sem tudja teljesen kizárni az emberi tényezőben megjelenő kockázatot és azokat az egyéb veszélyforrásokat, amelyek az informatikai technológián kívülről fenyegetik a biztonságos információfeldolgozást (például fizikai sebezhetőség vagy biztonsági szempontból nem megfelelő ügyviteli-rendszabályi eljárások), de ezek némelyikének a kockázatát csökkenti, illetve korlátozza az ezen támadások által okozható kár mértékét.

Irodalom

[1] RFC 1321, http://www.faqs.org/rfcs/rfc1321.html
[2] Trusted Computing FAQ, http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html
[3] Buttyán Levente, Vajda István: Kriptográfia és alkalmazásai; Typotex, 2004
[4] Andrew S. Tannenbaum: Computer Networks, 3rd Ed.; Prentice-Hall Inc, 1996.
[5] Terry Ritter: Ciphers By Ritter, http://www.ciphersbyritter.com/
[6] Conference on Communications and Multimedia Security 2005, http://cms2005.sbg.ac.at/
[7] J. Marchesini, S. Smith, O. Wild, R. MacDonald: Experimenting with TCPA/TCG Hardware, Computer Science Technical Report TR2003-476
[8] B. Arbaugh: Improving the TCPA Specification, IEEE Trans. on Computer Science, 2002 August (Vol. 35, Issue 8), http://ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=1023792

Kisfogyasztású érzékelők tervezése

NAGY GERGELY

Budapesti Műszaki és Gazdaságtudományi Egyetem, Elektronikus Eszközök Tanszék
gregn@freemail.hu

Kulcsszavak: integrált hőmérséklet-érzékelők, kisfogyasztású áramkörök, áramreferencia-áramkör

Az áramkörök fogyasztásának csökkentésére alkalmas módszer a használaton kívüli részegységek kikapcsolhatóvá tétele. Lényeges áramfelvétel-csökkenés érhető el, ha mindig csak azok az áramköri elemek fogyasztanak, amelyek éppen valamilyen műveletet végeznek, vagy értéket tárolnak. Analóg áramköröknél bizonyos helyzetekben a kikapcsolás és az újraindítás nem triviális feladat. Jelen munkában egy hőmérsékletfüggő áramreferencia kikapcsolhatóságának és indításának megoldásáról lesz szó. Két, az áramfelvétel minimalizálását segítő megoldást mutatunk be. Az első esetben a kiindulópont egy már létező, de tökéletlenül működő kapcsolás, a másik teljesen önálló fejlesztés.

1. Az integrált hőmérséklet-érzékelők szerepe

Az integrált áramkörök méretcsökkenésével, és bonyolultságuk valamint sebességük növekedésével a hőmérséklet-érzékelők szerepe egyre nő. A további sebességnövelés útjában álló egyik legnagyobb probléma a növekvő disszipáció, amely az áramkör túlmelegedéséhez, és így tönkremeneteléhez vezethet. A modern mikroprocesszorokban elterjedten alkalmaznak hőmérséklet-érzékelőket, amelyek túlmelegedés esetén csökkentik a működési frekvenciát, és így az áramfelvételt is.

Nagy szükség van tehát olyan áramkörökre, amelyek képesek érzékelni és elektromos jellé alakítani környezetük hőmérsékletének értékét. A hagyományos hőmérsékletérzékelők áram- vagy feszültség-kimenetűek. Ilyen például egy dióda, amelyen ha állandó áramot engedünk át, a rajta eső feszültség (nyitófeszültség) csökken, ha a hőmérséklet növekszik.

Az analóg kimeneti jellel az a probléma, hogy azt mindenképpen digitalizálni kell ahhoz, hogy egy logikai áramkör döntéseket hozhasson a mért értékek alapján, ráadásul a zavarérzékenysége is jelentős.

A hőmérséklettel arányos analóg jelet tehát érdemes az érzékeléshez lehető legközelebb átalakítani valamilyen digitális jellé. A Budapesti Műszaki és Gazdaságtudományi Egyetem Elektronikus Eszközök Tanszékén megtervezett hőmérséklet-érzékelő [1] frekvencia-kimenetű. Egy hőmérsékletfüggő áramgenerátor áramával arányos frekvenciájú jelet szolgáltat a kimenetén, amely így könnyen feldolgozható digitálisan *(1. ábra).*

A szenzor kimenete egy számláló órajele, amely tehát a jel frekvenciájának ütemére számol. Egy párhuzamos betöltésű, sorosan kiléptethető regiszterbe megfelelő periódusonként beolvasva a számláló értékét, majd az egymás után érkező jelek különbségét véve, ismerve a mintavételi időt, kiszámolható a szenzor frekvenciája:

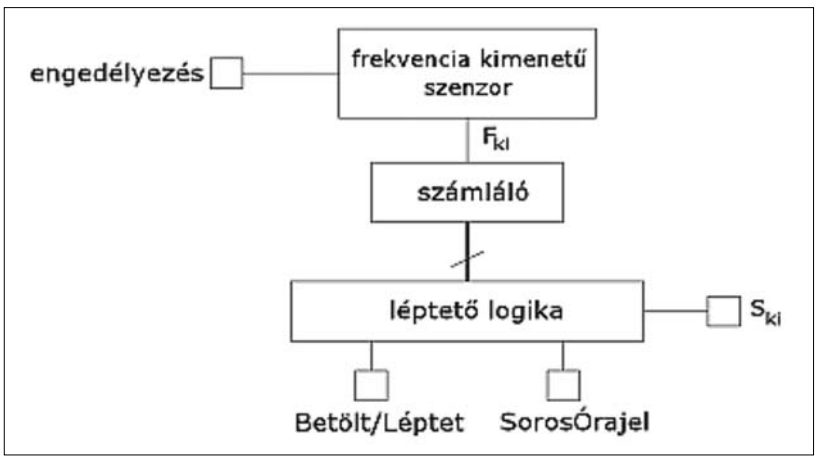
$$f = \frac{S_{ki}(t_1) - S_{ki}(t_0)}{t_1 - t_0}$$

ahol S_{ki} a számlálóból kiolvasott érték, t_0 és t_1 pedig két egymást követő mintavételi időpillanat.

Ez az architektúra tehát különösen alkalmas arra, hogy digitális áramkörökhöz kapcsolódva a környezet hőmérsékletéről adjon azok számára könnyen feldolgozható információt.

2. Az érzékelő működése

A Elektronikus Eszközök Tanszékén nagy hagyományai vannak az integrált áramkörök hőmérsékleti viselkedését elemző kutatásoknak. Ezen munkák során készült el egy disszipátorokból és érzékelőkből álló, mát-



1. ábra
A frekvencia kimenetű szenzor jelének digitális feldolgozása

rix elrendezésű áramkör. Ebben a disszipáló elemek tetszőlegesen be- és kikapcsolhatóak. A szenzorok segítségével tanulmányozható a hőterjedés a chip felületén. Mivel egyszerre egy adott érzékelő kimenetét figyeljük, adódik az igény, hogy a többi kikapcsolható legyen.

Az áramkör a hőmérséklet érzékelésére egy, a tanszéken kifejlesztett eljárást alkalmaz [2]. A hőmérsékletfüggő elem egy áramforrás, amely szándékosan úgy lett kialakítva, hogy felerősítse a hőmérsékletváltozás kiváltotta munkaponti eltolódásokat. Az áramforrás jeléből ezután az áramával arányos frekvenciájú kimeneti jelet állítunk elő [3]. A forrás tranzisztoros kapcsolók által vezérelt módon felváltva tölt fel, illetve süt ki egy kondenzátort. A kondenzátor feszültsége egy komparátor bemeneti jelét szolgáltatja. A komparátor másik bemenetén, szintén kapcsolók által vezérelve, két referencia feszültség van.

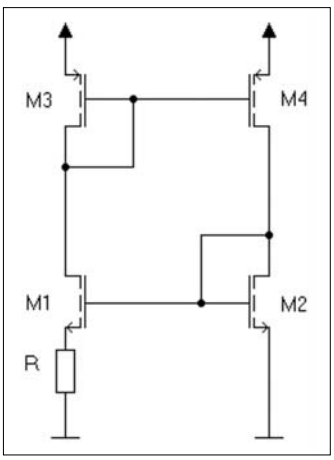
A kapcsolókat úgy vezéreljük, hogy amikor a töltő áram a kondenzátort a magasabbik referenciafeszültség szintjéig töltötte, akkor átkapcsolnak, és onnantól kezdve a hőmérsékletfüggő áram elkezdi kisütni a kondenzátort. Ekkor a komparátorra már a másik, alacsonyabb értékű referenciafeszültség van kapcsolva, így a kondenzátor kisütése abba fog maradni, amikor annak feszültsége a referenciát eléri. Amikor ez bekövetkezik, az áram újra tölteni kezdi a kondenzátort. Így alakul ki egy oszcilláció, amelynek frekvenciája a töltő árammal megegyező irányban változik, hiszen annál gyorsabban töltődik fel, illetve sül ki a kondenzátor, minél nagyobb az áram.

A komparátor kimenete egyben a teljes kapcsolás kimenete is, és ez a potenciál, illetve az invertáltja vezérlik a kapcsolókat is.

3. Az áramforrás

Egy áramforrás alapeleme az önbeálló áramreferencia. Ennek alapkapcsolása *(2. ábra)* egy ellenállásból és két áramtükörből áll. A kapcsolás két stabil állapottal rendelkezik: egyik a zérus feszültséghez és áramhoz tartozik, a másik ahhoz az áramhoz, amelyen működtetni szeretnénk a kapcsolást.

Amikor tápfeszültséget kapcsolunk az áramkörre, akkor – amennyiben a tranzisztor gatejére nem adunk vezérlést – a két elem a zérus értékhez tartozó munkapontba kerül. Amennyiben a másik munkapontba szeretnénk helyezni őket, indító áramkörre van szükség.



2. ábra
Önbeálló áramreferencia

Az R ellenállás tulajdonképpen egy negatív áram visszacsatolást eredményez. Ha – például a hőmérséklet változása miatt eltolódott munkapont következtében – megnő M2 árama, akkor megnő M1-é is, hiszen áttükröződik. Azonban ekkor megnő az R ellenálláson eső feszültség is, amitől lecsökken M1 gate-source feszültsége, így M1 árama is. Ez felül visszatükröződik M2 ágába is, így az áram lecsökken, stabil értéken marad.

Az általunk vizsgált szenzorban a hőmérsékletfüggés felerősítése érdekében az ellenállás a referencia másik ágába kerül, és egy „diódának kapcsolt" MOS tranzisztor valósítja meg. A referencia kimeneti ellenállása meg lett növelve úgy, hogy stabil munkapontba állított tranzisztorok kerültek a felső áramtükrök alá *(3. ábra).*

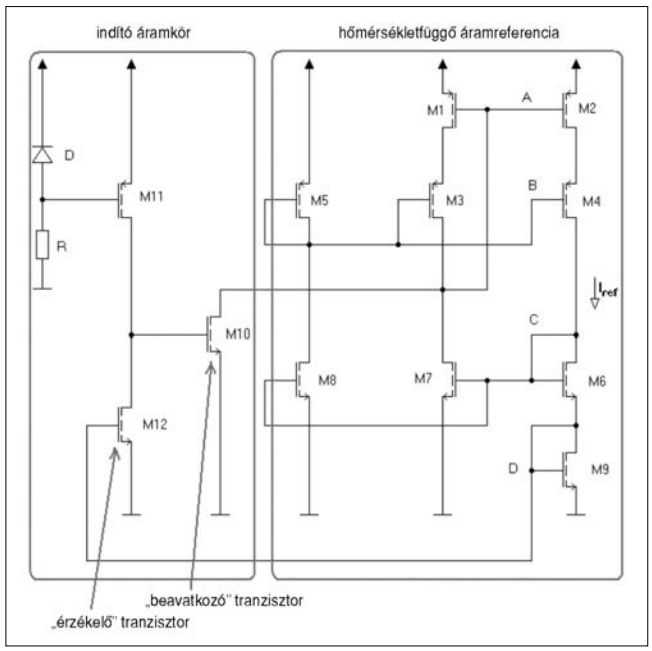
Az áramreferencia egy kétállapotú áramkör, ezért indítóáramkörre van szüksége, hogy a munkapontjába bekerüljön. Egy indítóáramkör lehet statikus, vagy dinamikus. Az előbbi azt jelenti, hogy egy „érzékelő" tranzisztor folyamatosan figyeli azt, hogy a referencia munkapontban van-e. Amíg nincs, úgy vezérel egy vagy több „beavatkozó" tranzisztort, hogy az segítsen az áramkörnek elindulni. Amikor ez megtörtént, a vezérlés kikapcsolja a beavatkozó tranzisztort.

Egy dinamikus indítóáramkör általában egy egy-időállandós dinamikus tag, amely a tápfeszültség megjelenése után lejátszódó ugrásválaszát követően állandósult állapotba kerül. Ezt az időállandónyi késleltetést lehet felhasználni.

Az általam vizsgált szenzorban *(3. ábra)* statikus indítóáramkör szerepel, amelynek érzékelő tranzisztora az M12-es, beavatkozó tranzisztora pedig az M10-es.

A statikus indítóáramkör hátránya, hogy kikapcsolt állapotban (amikor nincs szükség beavatkozásra) van áramfelvétele. Ennek oka az, hogy ilyen esetben a beavatkozó tranzisztor vezérlő elektródájára kapcsolódó

3. ábra Az áramreferencia kapcsolása



mindkét tranzisztor nyitva van, és a pont potenciálját a két tranzisztor csatorna méreteinek aránya dönti el. Az M11-es felhúzó tranzisztor csatorna-ellenállása nagy, az M12-esé kicsi, így a beavatkozó tranzisztor vezérlő elektródája közel földpotenciálra kerül. Eközben a két tranzisztoron keresztül áram folyik.

4. A fejlesztési feladat

A fejlesztés célja a korábbiakban ismertetett szenzor fogyasztásának minimalizálása. Ennek egyik lehetősége a szenzor kikapcsolása, amelynek során meg kell oldani az újraindítást is. Egy másik lehetséges mód egy alternatív indítóáramkör tervezése, amelynek nincsen statikus áramfelvétele, és a helyigénye is nagyon kicsi.

4.1. Az áramreferencia kikapcsolhatóságának megoldása

Ahogy arra már utaltunk, létezett egy korábbi kapcsolás [4], amely megoldani látszott a problémát, ám a működésével komoly gondok voltak:

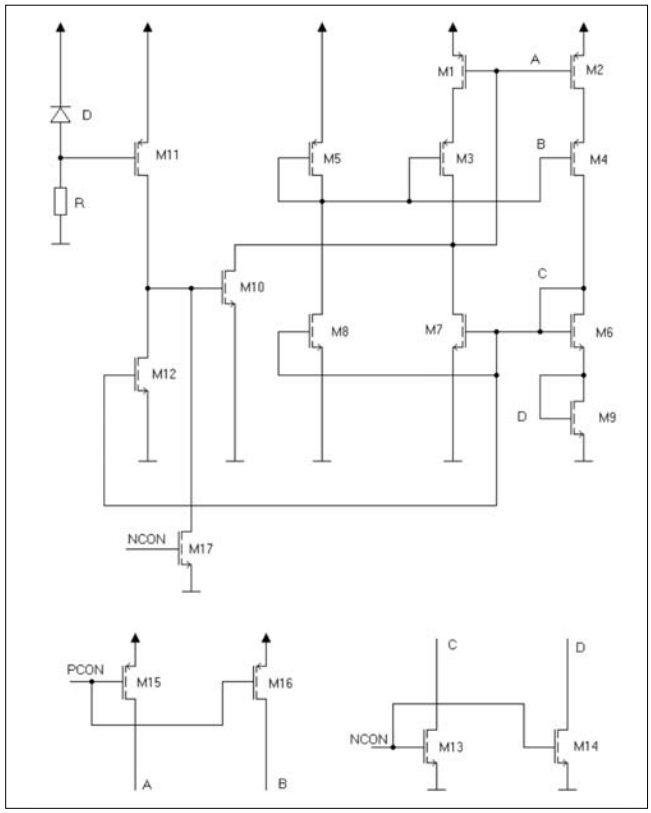
– az indító áramkör nagy hőmérsékleten bekapcsolt, és elrontotta a helyes működést;
– kikapcsoláskor az áramreferencia fogyasztása jelentős maradt;
– visszakapcsoláskor nem indult újra az áramkör – nem került vissza a munkapontba.

Célom a fenti hibák okának felderítése volt, és hogy megoldást találjak azok kiküszöbölésére.

Az első problémának az oka az volt, hogy magas hőmérsékleten az M12-es érzékelő tranzisztor nem nyitott ki eléggé, az azt vezérlő D pont potenciálja túl alacsony volt. A megoldást a vezérlő elektróda C pontra kötése jelentette.

A második hibát a kikapcsoló áramkör okozta. Az áramreferencia kikapcsolásakor az A és B pontokat tápfeszültségre, a C és D pontokat földpotenciálra kapcsoljuk. Ekkor azonban az M12 tranzisztor nem nyit ki, és így az M10-es kinyit. Ekkor az A pontot tápfeszültségre „húzó” tranzisztoron és a nyitott M10-esen keresztül kis ellenállású út nyílik a táp és a föld között, ami nagy áramot indít el. Erre a problémára az jelent megoldást, ha a beavatkozó (M10) tranzisztort is kikapcsoljuk. Ezzel jelentős csökkenést érhetünk el az áramfelvételben: a szimulációk alapján 6 V-os tápfeszültségnél, 27°C-on a teljes szenzor 72 µA áramot vesz fel bekapcsolt, és 3,6 µA áramot kikapcsolt állapotban.

A harmadik problémát az indító áramkör okozta. Indításkor az eredeti áramkör mindössze az A pont potenciálját állította be, ám ezzel nem nyitotta ki a B ponthoz gate-jükkel kapcsolódó M4-es, illetve M3-as tranzisztort, amelyek így megakadályozták az áram elindulását. Ennek kiküszöbölésére a B pontot is a földpotenciál közelébe kell húzni indításkor.



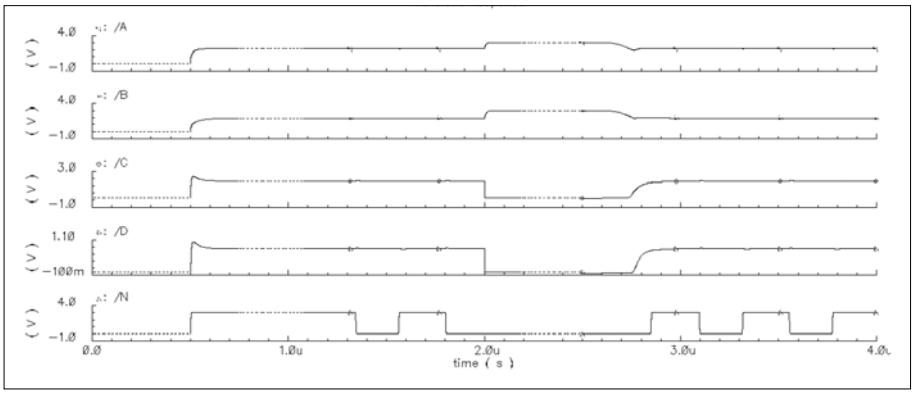
4. ábra A javított áramreferencia

A javított áramgenerátor a kikapcsoló tranzisztorokkal együtt a *4. ábrán* látható. A tervezés során tehát az M12-es tranzisztor vezérlő elektródáját a D helyett a C pontra kötöttem, kikapcsolhatóvá tettem az indító áramkört az M17-es tranzisztorral, és az indító áramkörben elhelyeztem az M18-as tranzisztort, mely a B pont potenciálját a földpotenciál közelébe húzza bekapcsoláskor.

A javított áramgenerátor kapcsolása kiegészült kikapcsoló tranzisztorokkal (M13-M15) is. Ezek segítségével az áramkört belső, logikai jelei (NCON, PCON) képesek az áramkört be- és kikapcsolni. Ezzel megvalósítható az, hogy, amikor a kiválasztunk egy adott szenzort, hogy annak jele megjelenjen a kimeneten, akkor a többi szenzor automatikusan kikapcsol.

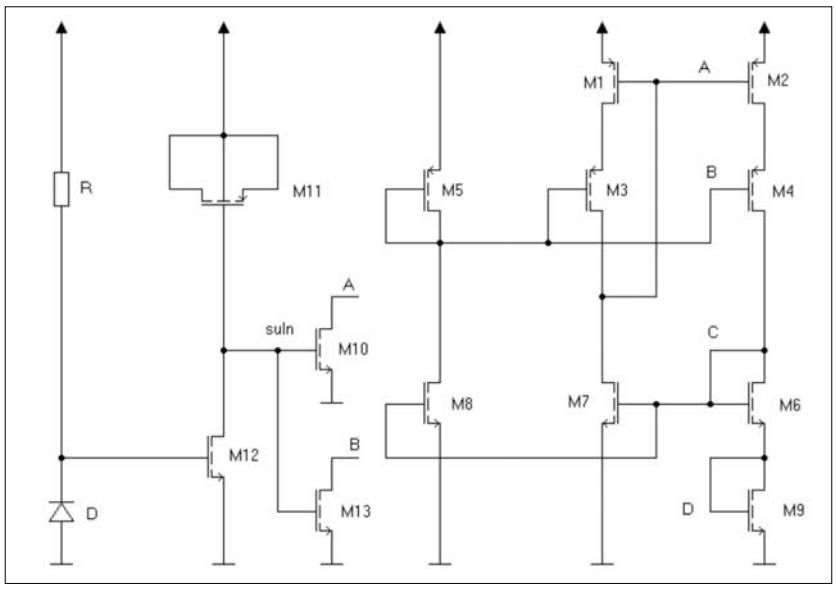
A tervezés a Cadence cég Opus tervezőrendszerében történt. Az *5. ábra* bemutatja a teljes szenzor egy

5. ábra Az áramreferencia szimulációja



szimulációs eredményét. A kirajzolt jelek felülről haladva sorban: az áramgenerátor referencia-pontjai (A, B, C, D), és a frekvencia-jellegű kimenet. Jól látható, hogy az áramkör bekapcsolásakor a referencia-pontok igen hamar felveszik a munkaponti értéküket, ám a kimeneti jel csak késve kezd oszcillálni. Ennek az az oka, hogy az áramreferenciának fel kell tölteni a kezdetben teljesen kisütött kondenzátort. A későbbiekben a töltés és kisütés csak két, egymástól nem távoli referencia-feszültség között történik.

Az ábra közepén látható egy kikapcsolás és újraindítás. Kikapcsoláskor az A és B referenciapontok a tápfeszültséghez, a C és D pontok a földpotenciálhoz közeli értéket vesznek fel, így a munkaponti áramok értéke zérus lesz. Megfigyelhető, hogy az újraindítás jóval gyorsabban történik, mint az első indítás, ennek oka, hogy ilyenkor a kondenzátor még őrzi a rávitt töltést és csak az elszivárgott mennyiséget kell pótolni.



7. ábra Áramreferencia dinamikus indító áramkörrel

4.2. Egy alternatív indító áramkör megtervezése

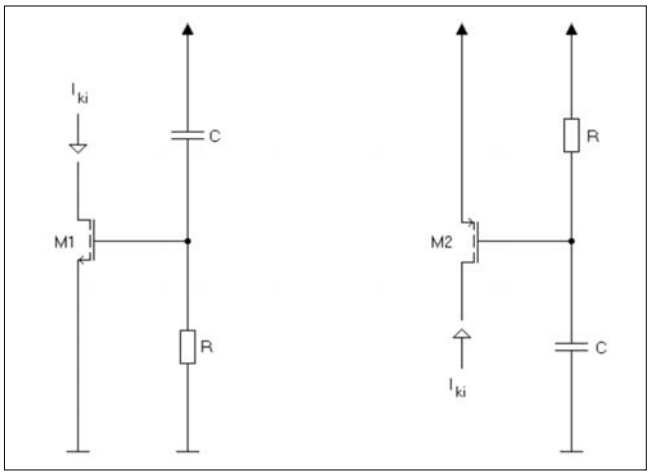
Mint láttuk, a statikus indítóáramkörök működési elvükből következően mindenképpen fogyasztanak olyankor is, amikor az általuk vezérelt áramkör már munkapontban van. Dinamikus indítással ez a fogyasztás megszűntethető *(6. ábra)*.

A dinamikus indítók egy tranzisztorból és egy RC tagból állnak. A tápfeszültség bekapcsolásakor a kondenzátor még nincs feltöltve, így az M1 n-MOS, illetve M2 p-MOS tranzisztorok kinyitnak. A kondenzátor az R ellenálláson keresztül feltöltődik, és ekkor a tranzisztorok bezárnak. A dinamikus indítóáramkörök előnye, hogy az elzáródás után zérus az áramfelvételük, hátrányuk, hogy amíg a tápfeszültség jelen van (a kondenzátor fel van töltve), addig nem tudnak újraindulni, így ha a kapcsolás kikerül a munkapontból (például ideiglenesen ki lett kapcsolva), nem tudják visszaállítani oda. Így dinamikus indítás esetén nem oldható meg, hogy az áramkör belső, logikai jelei által vezérelt módon kapcsoljuk be, illetve ki az áramreferenciát.

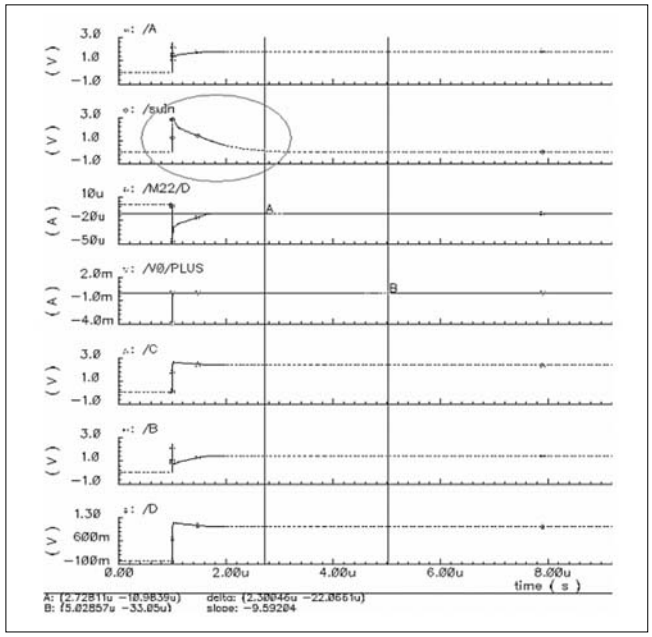
Megterveztem egy dinamikus indító áramkört a kapcsoláshoz *(7. ábra)*. Az RC-tagban az ellenállás (M12) és a kondenzátor (M11) is tranzisztorból épül fel. Ebben a kapcsolásban is vezérelni kell mind az A, mind a B pontot a biztos munkapontba állás érdekében.

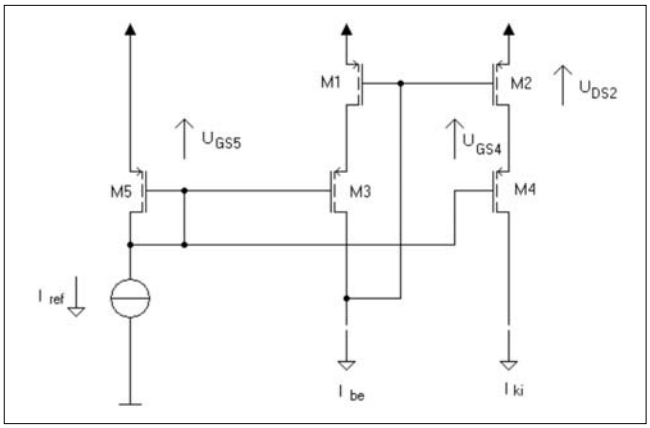
A kapcsolás helyes működését szimulációkkal igazoltam *(8. ábra)*. Az indításkor fellépő ugrásválasz bekarikázva látható. A kondenzátor kapacitását, ill. az ellenállás értékét úgy kellett beállítani, hogy az áramreferenciát a teljes vizsgált hőmérséklet-, és tápfeszültség tartományban elindítsa, de ne legyen feleslegesen nagy érték, ugyanis ez egyben nagy méretet is jelentene. A kapacitás ugyanis az adott tranzisztor gate-területétől, az ellenállás pedig a W/L arányától függ.

6. ábra Dinamikus indító áramkörök



8. ábra A dinamikus indítás szimulációja





9. ábra Javított áramtükör méretezése

5. Megjegyzés az áramkörméretezéshez

Az áramkör kizárólag növekményes MOS tranzisztorokat tartalmaz. Az áramgenerátor tranzisztorainak W/L arányait (és ezzel a töltőáramot), valamint a kondenzátor méretét a megcélzott frekvencia-tartomány függvényében kell beállítani. Az áramreferencia méretezés szempontjából érzékeny pontja az M1-M5 tranzisztorokból álló javított áramtükör. Itt a megfelelő W/L arányokkal lehet biztosítani a helyes működést.

Az egyszerű kéttranzisztoros áramtükör kapcsolás továbbfejlesztésére azért van szükség *(9. ábra)*, mert annak kimeneti ellenállása nem elegendően nagy, így az általa megvalósított áramgenerátor árama nem lesz független a terheléstől.

A kapcsolásban nagyon fontos, hogy a tranzisztorok ne lépjenek ki a telítéses tartományból (szaturáció).

A huroktörvényből következően a munkaponti feszültségekre a következő egyenlet írható fel:

$$U_{GS5} = U_{GS4} + U_{DS2} \tag{1}$$

A gate-source feszültségek felírhatóak a nyitófeszültség (V_T) és a szaturációs feszültség segítségével:

$$U_{GS5} = U_{SAT5} + V_{T5} \tag{2}$$

$$U_{GS4} = U_{SAT4} + V_{T4} \tag{3}$$

Szaturációban a MOS tranzisztor drain-source feszültsége nem kisebb, mint a szaturációs feszültség. Határhelyzetben pont egyenlő vele. Mivel azt szeretnénk, hogy minden tranzisztor telítésben maradjon, ezért U_{DS2} helyére U_{SAT2}-t helyettesítünk, ami a határhelyzetet jelenti. Az M2 és az M4 tranzisztor azonos, így az ő szaturációs feszültségeik is azonosak ($U_{SAT4} = U_{SAT2}$), és a technológiából következően a nyitófeszültségek is jó közelítéssel azonosnak tekinthetőek minden tranzisztornál. Ezek figyelembevételével (1)-be helyettesítve:

$$U_{SAT5} + V_T = U_{SAT} + V_T + U_{SAT} \tag{4}$$

A MOS tranzisztor karakterisztika egyenletéből kifejezhető a szaturációs feszültség:

$$U_{SAT} = U_{GS} - V_T = \sqrt{\frac{I_D \cdot L}{K \cdot W}} \tag{5}$$

ahol $K = \mu C_0 / 2$, és a technológiából következően minden tranzisztorra azonosnak tekinthető. Az áramtükrök miatt az M5 tranzisztoron ugyanaz a drain-áram folyik át, mint ami M2-n és M4-en, így amikor (5)-öt (4)-be helyettesítjük, I_{D2}, I_{D4} és I_{D5}, helyére ugyanaz az áram írható:

$$\sqrt{\frac{I_D \cdot L_5}{K \cdot W_5}} = 2 \cdot \sqrt{\frac{I_D \cdot L_2}{K \cdot W_2}} \tag{6}$$

A jobb oldalon W_2 és L_2 került a képletbe, mivel M2 és M4 méretei megegyeznek. I_D-vel és K-val egyszerűsíthetünk, és a négyzetre emelés és átrendezés után:

$$\frac{W_5}{L_5} = \frac{1}{4} \cdot \frac{W_2}{L_2} \tag{7}$$

Azt kaptuk tehát, hogy ha munkapontba állított tranzisztoros áramtükröket *(9. ábra)* alkalmazunk, akkor a munkaponti feszültséget szolgáltató tranzisztor (M5) W/L aránya az áramtükör tranzisztorainak W/L arányának negyede kell legyen.

6. Összefoglalás

Jelen munka során a célom egy adott hőmérséklet-szenzor fogyasztásának a csökkentése volt. A problémára két megoldást adtam. Elsőként a már létező kapcsolást alakítottam át úgy, hogy kikapcsolható legyen, amikor nincs szükség az általa szolgáltatott jelre. Kikapcsolt állapotban a szenzor fogyasztása a huszadára csökken.

Alternatív megoldásként pedig megterveztem egy egészen kis áramfelvételű, dinamikus indító áramkört a szenzorhoz.

Irodalom

[1] Székely Vladimír,
CMOS compatible temperature sensors,
Journal on communications, Vol XLVII, May 1996,
pp.13–17.

[2] V. Székely, M. Rencz, S. Török,
Cs. Márta, L. Lipták-Fegó,
CMOS temperature sensors and built-in test circuitry
for thermal testing of IC's, Sensors and Actuators,
Special Issue, Vol. 471, No.1-2, Nov. 1998,
pp.10–18.

[3] Hainzmann János, Varga Sándor, Zoltai József,
Elektromos áramkörök,
Nemzeti Tankönyvkiadó, Budapest, 2000,
p.554.

[4] Harrer Margit,
Termikus tesztchip szenzorainak tervezése és mérése,
Diplomaterv, Budapest, 2003.

„Ambient” hálózatok

KOVÁCS BALÁZS, SIMON CSABA

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
kovacsb@tmit.bme.hu
simon@david.tmit.bme.hu

Kulcsszavak: „ambient” hálózati paradigma, abszorpciós és átjáró alapú kompozíciós modellek

Az „ambient” vagy más néven mindent körülölelő hálózatok a következő generációs számítástechnika paradigmája, mely a felhasználók számára lehetőséget ad a folyamatos számítástechnikai és hálózati együttműködésre. Az „ambient” paradigma túllép az egyes hálózati technológiákon, mert fókuszában ezen heterogén rendszertechnológiák háttérbe szorítása, láthatatlanná tétele és automatikus kezelése van. Mindezen célokat intelligens, önmenedzselő és skálázható módon próbája elérni, mely igen komoly kihívásként jelentkezik. Jelen cikkben bemutatjuk az „ambient” koncepciót, melynek központjában a dinamikus hálózati együttműködések (hálózatok kompozíciója) áll. Leírjuk a hálózatkompozíció működését, valamint röviden bemutatunk egy megközelítést, mely skálázható módon támogatja az automatikus és dinamikus hálózatkompozíciót.

1. Bevezető

A mobil hálózati technológiák napjainkban már beépültek mindennapi életünkbe. A már jól ismert második generációs (2G) és harmadik generációs (3G) technológiákon kívül előtérbe kerültek az ad hoc jelleggel létrehozott hálózatok előnyei, és ezáltal problémái is. A mindenütt jelenlevő („ubiquitous”), mindent átható („pervasive”), mindent körülölelő („ambient”) számítástechnikai elképzelések szerves részei az ad hoc önszerveződő hálózatok, melyek igény szerint alakulnak. A hálózatok különböző szinteken történő együttműködése, egybeolvadása, kompozíciója segíthet az imént említett elképzelések szerteágazó problémáinak megoldásában.

A hálózatkompozíció egy az Európai Unió által támogatott „Ambient Networks” projektben megjelent új architektúrális koncepció [1,2]. A hálózatkompozíció célja a statikus és dinamikus (akár önszervező) kompozíció, mely automatikus hálózat-konfigurációt és felügyeletet hajt végre. A technológiának támogatnia kell a különböző hálózati együttműködési politikákat (policy) és megkötéseket, hogy lehetővé tegye az információ védelmét és a szolgáltatások elérhetőségét egy adott csoporton belül. A jelenlegi hálózati technológiák statikus mivolta gátolja a dinamikus hálózatok automatikus létrehozását és az automatikus együttműködést.

A dinamikus hálózatkompozíció és ennek fordítottja – a dekompozíció – a hálózatok folyamatos újrakonfigurálását és a felajánlott szolgáltatások állandó felügyeletét jelenti. Ezeknek a változásoknak ráadásul nem szabad felhasználói beavatkozást igényelniük, hanem magas szintű felhasználói igények alapján kell végbe menniük.

Ebben a cikkben egy a kompozíciós problémák megoldására kialakított rendszert mutatunk be. A rendszerben a csomópontok önszervező módon hálózatok hálózatába szervezik magukat. A logikai hálózatok, melyek a fizikai topológia felett keletkeznek (az úgynevezett virtuális hálózatok) a vezérlési és menedzsment feladatok ellátására jönnek létre. A hálózatok határai a szomszédsági viszonyok és politikák figyelembevételével keletkeznek [3].

A következőkben egy rövid irodalom áttekintés keretében bemutatásra kerülnek a már létező csoportosító algoritmusok és virtuális hálózatok. Bemutatjuk az „ambient” paradigma követelményeit, továbbá az „Ambient Networks” projektben kialakított fogalmakat, komponenseket és relációikat. A cikk végén ismertetünk egy olyan új hálózati architektúrát, mely képes skálázhatóan kezelni heterogén hálózatok dinamikus együttműködését a hálózatkompozíción és az önszerveződésen keresztül.

2. Az irodalom áttekintése

A hálózatok önszerveződése és dinamikus kompozíciója kulcsfontosságú követelményei az „ambient” paradigmának. Míg a dinamikus hálózatkompozíció [4] az „Ambient Networks” projektben [1] nemrég kidolgozott koncepció, addig az önszerveződés a „peer-to-peer” és ad hoc hálózatok területén egy igen alaposan körüljárt terület. Számos elosztott algoritmust javasoltak, mely képes az ilyen jellegű hálózatokat skálázható architektúrába szervezni. A legtöbb javaslat két kategóriába sorolható: csoportosító algoritmusok, illetve virtuális hálózatok.

2.1. Csoportosító algoritmusok

Ezek az algoritmusok egysíkú ad hoc hálózatokban gyakran használt mechanizmusok skálázható hierarchikus struktúra kialakítására. A csoportosító algoritmusok (clustering) csoportokba szervezik a hálózati csomópontokat, úgy hogy mindegyik csomópont egy választott csoportfőnökhöz tartozik. A csoportok dinamikusan alakulnak a fizikai topológiának megfelelően. A csoport kialakítása során általában két dolgot vesznek figye-

lembe: a fizikai topológiát, illetve csomópontokhoz rendelt mérőszámot, mely a csoportfőnökségre vonatkozó jóságot adja meg. A legtöbb algoritmus a csoporttagoktól megköveteli a szomszédsági viszonyt [5], esetleg hogy legfeljebb „d" ugrásnyi távolságra [6] legyenek a csoportfőnöktől. Habár a csoportok fenntartása kommunikációs adminisztratív terhelést is jelent, kutatók bebizonyították, hogy a leggyakoribb körülmények között ez a többlet a csomópontok számának függvényében logaritmikus [7].

A csoportosítás hasznos útválasztásra, közegelérésre, címkiosztásra és más vezérlési síkon végrehajtandó feladatok során. Habár a kompozícióra képes hálózatok szervezésénél is hasznos lehet, mégis a következő problémákkal küszködik:

– A legtöbb csoportosító algoritmus nem enged meg kettőnél több hierarchia szintet, mely nagy hálózatokban skálázhatósági problémákhoz vezet.
– A csoportosítás nem veszi figyelembe a hálózatmenedzsment információkat, mint például a policy-ket, továbbá korlátozza a csoportok hatótávolságát (ugrás alapján) egy megadott számig.
– A jelenlegi csoportosító algoritmusok nem veszik figyelembe azt a tényt, hogy a csomópontoknak több hálózati interfésze is lehet, mely az „ambient" hálózatokban gyakran előfordulhat.

2.2. Virtuális hálózatok

A virtuális hálózatok a fizikai hálózati topológiára építve keletkeznek. Egy virtuális hálózatot a fizikai topológiából kiemelt néhány csomópont alkot, melyek egy absztrakt topológiát látnak a hálózatról. Ezek a hálózatok lehetnek statikusak, vagy folyamatosan változók, önszerveződők. A leginkább ismert önszerveződő virtuális hálózatok az elosztott hash tábla (Distributed Hash Table – DHT) alapon működő „peer-to-peer" hálózatok. Ilyen például a Chord, CAN, Pastry vagy a Tapestry [8-11].

A DHT-kat nagy elosztott hálózatokban hatékony adatlekérésre használják. Minden adat a hálózatban egy kulcs-érték párossal van leképezve. A hálózat minden csomópontja belép a DHT virtuális hálózatába, ezáltal felelős lesz a kulcstartomány egy adott területéért. A virtuális hálózatot használva az adatkérések csomópontról csomópontra szállnak egészen addig, míg el nem érkezünk a kulcsot tároló csomópontig, ahol az adatot is megtaláljuk. A DHT-kban a skálázhatóságot a hálózatmérettől (csomópontszám) függő logaritmikus átmérő megválasztásával biztosítják.

Habár a DHT-k önszervező virtuális hálózatokat valósítanak meg, mégis inkább a hatékony adatlekérésre optimalizálták azokat, és nem képesek skálázható hálózati architektúrát biztosítani a kompozícióra képes hálózatok számára. A leglényegesebb korlátai a DHT virtuális hálózatoknak a következők:

– Korábban elkülönülő DHT-k összefésülése igen körülményes, ugyanis az egész kulcstartomány újrafelosztását igényelné.
– A legtöbb DHT-nak egysíkú struktúrája van, vagy a hierarchiák száma korlátos.

3. Az „ambient" hálózati paradigma

Az „ambient" hálózati paradigma egy olyan számítástechnikai környezetet feltételez, melyben rengeteg számítástechnikai eszköz szerepel, működésükhöz elegendő kommunikációs erőforrással, melyek adminisztrációja az emberek elől kellő mértékben a háttérbe szorul. Egy ilyen technológia gyakorlati megvalósítása néhány kritikusan fontos követelmény teljesítését jelenti. A következőkben röviden összefoglalunk néhány ilyen követelményt.

A *mobilitás* nagyon fontos összetevője az „ambient" hálózatoknak, hiszen minden valós felhasználói környezetben előfordul. A vezeték nélküli technológiák fejlődésével és a vezeték nélküli kommunikációra képes eszközök elterjedésével egyértelmű, hogy a mobilitás kezelése kulcsfeltételt képez az „ambient" hálózatok megvalósítása során.

A *„láthatatlanság"* az „ambient" paradigma egy másik kulcsfeltétele. Ideális esetben a láthatatlanság a megvalósító technológia elrejtését jelenti a felhasználó elől. Gyakorlatilag azonban ezt csak megközelíteni lehet. A láthatatlanság biztosítását megkönnyíthetik az intelligens eszközök, melyek folyamatosan igazodnak a felhasználó igényeihez, akár minimális beavatkozással. Ez a probléma sokkal nehezebbé válik, ha a felhasználó egy dinamikusan változó környezetbe kerül.

Az *intelligens környezetek,* terek jelenléte szintén fontos összetevő. Lehetővé teszik az érzékelést és vezérlési interakciókat a számítástechnikai és az emberi környezet között. Például egy bizonyos alkalmazás különböző módon kell, hogy működjön a felhasználó helyétől függően. Az eltérő működési üzemmódokat a fizikai környezetbe épített szenzorok érzékelései alapján kezelhetjük.

A *skálázhatóság* szintén egy kritikus szempont, hiszen nagyszámú csomópontot és interakciót kell kezelnie a számítástechnikai környezetnek. Ez a tény azonban a számítástechnikai és hálózati erőforrások tekintetében (sávszélesség, memória és energia) komoly hatással lehet a felhasználóra is. A felhasználók közötti folyamatosan növekvő kommunikációs és számítási igény szintén nagy nyomást gyakorol az erőforrásokra, mely így a skálázhatóságot abszolút előtérbe helyezi az „ambient" rendszerek tervezése során.

A számítástechnikai és a kommunikációs technológiák széles választékának és különböző képességeinek köszönhetően igen *heterogén* környezet alakul ki. A helyzet tovább bonyolódik e technológiák helytől függő elterjedési szintjei miatt. Az ilyen esetekben a heterogenitást el kell tudni takarni a felhasználó elől.

4. Az „ambient" hálózati összetevők

Az „ambient" hálózatok tervezésének alapelve egy közös vezérlési tér létrehozása, mely általános vezérlési funkciókat biztosít változatos alkalmazások és hozzáférési technológiák számára. A projekt egy új hálózati

típust definiál, az „ambient” hálózatot. Ezek a hálózatok képesek dinamikus és automatikus módon együttműködési megállapodásokat kötni más „ambient” hálózatokkal.

Egy ilyen hálózat lényegi összetevője a közös vezérlési tér, melyet „Ambient Control Space”-nek (ACS) neveznek [12]; valamint az ACS kapcsolódását általánosan meghatározó interfészek *(1. ábra)*:

– az „Ambient Network Interface” (ANI), mely az együttműködő hálózatok ACS-ei közötti kommunikációt biztosítja.
– egy szolgáltatásokat támogató „Ambient Service Interface” (ASI) és
– a heterogén technológiák elrejtését szolgáló „Ambient Resource Interface” (ARI).

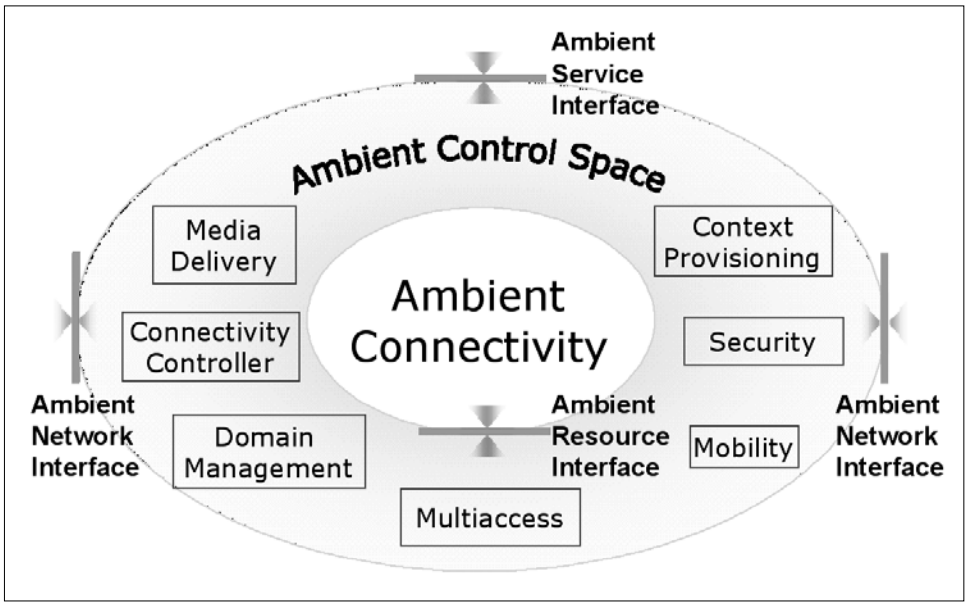
Az „ambient” hálózatok elképzelés nagyon sok mai és jövőbeli hálózati típusra alkalmazható, például személyi hálózatokra (PAN), vagy szenzor hálózatokra. Az „ambient” hálózati környezetben egy önálló eszköz, például egy felhasználói terminál egy hálózatot alkot, ennél fogva e hálózatok alap építőeleme inkább hálózatok mintsem csomópontok.

A hálózat-kompozíció koncepciója integráns része az „ambient” hálózatok fejlesztésének. Lehetővé teszi a hálózatok olyan együttműködését, mely túlmutat a jelenleg hálózati infrastruktúrák ismert együttműködési képességein.

A kompozíció az együttműködést olyan szintre emeli, mely nemcsak egyszerű címzési-, és útválasztási szintű együttműködést eredményez, hanem magasabb szintű funkciókat, mint a QoS és a mobilitás támogatás. A dinamikus és automatikus kompozíciók heterogén, akár más adminisztrációs vezérlés alatt lévő hálózatok között is létrejöhet.

Az 1. ábra által bemutatott kulcs „ambient” komponensek rövid leírását a következő szakaszok tartalmazzák.

1. ábra
„Ambient” vezérlési tér és az „ambient” interfészek [2]



4.1. Az „ambient” vezérlési tér

Az „ambient” vezérlési tér (ACS) célja, hogy felülkerekedjen a jelenlegi mobil hálózatok és az Internet problémáján, a közös vezérlési sík hiányán. Jelenleg vezérlési környezeteket hoztak létre olyan szolgáltatásokhoz, mint például a mobilitás menedzsment, QoS és biztonság. Az ACS vezérlési funkciók halmazából, avagy funkcionális területekből áll, mint például a kompozíció, a QoS, a mobilitás, a biztonság, illetve a környezet menedzsment. Mindegyik funkcionális terület egy speciális területtel foglalkozik. A kompozíciós terület felelős a kapcsolattartásért és a hálózatok komponálásával kapcsolatos vezérlési és szervezési tevékenységekért.

A különböző funkcionális entitások együttműködnek a komplex feladatok teljesítésének érdekében. Például a QoS és a mobilitás funkcionális területei együtt dolgoznak, hogy mobilitás függő QoS-t biztosítsanak [13].

Az *„ambient” hálózati interfész* (ANI) a hálózatok közötti kommunikációt lehetővé tevő felület. Egy ACS vezérlési funkciói számára biztosít általános mechanizmusokat egy másik ACS vezérlési funkcióival történő interakciókra. A közös felület eltakarja a különbségeket a hálózati technológiák között, és biztosítja az általánosság látszatát. Az „ambient” hálózatok vezérlési területei az ANI-n keresztül kommunikálnak, mely magában foglalja a jelzésátvitelt, információcserét, és más hálózatok felderítését. A felület leegyszerűsíti a „plug-and-play” egyeztetést és kompozíciót a hálózatok között.

Az *„ambient” szolgáltatási interfészt* (ASI) az alkalmazások számára teszi lehetővé, hogy elérjék az „ambient” hálózatok szolgáltatásait. Minden hálózatnak van egy ASI-ja az alkalmazások számára. Abban az esetben, ha több hálózat összeolvad eggyé, egy új ASI jön létre az új hálózat számára. Az általános ASI-nak köszönhetően egy alkalmazás számára a hálózatokban bekövetkező változás nem észrevehető.

Az *„ambient” erőforrás interfész* (ARI) a fizikai, hálózati erőforrások, interfészek egységes kezelését teszi lehetővé, mely nagymértékben hozzájárul a heterogenitás problémájának kezeléséhez.

Az *általános „ambient” hálózati jelzési protokollt* (Generic Ambient Network Signaling – GANS) a hálózatok az ANI-n keresztül történő információcserére használják. A hálózat kompozíció is az ANI-t használó GANS protokoll segítségével zajlik le. Mint jelzési protokoll, a GANS független az alatta lévő technológiától. A GANS

nem helyettesíti a jelenleg használt szabványos mobilitási és QoS protokollokat, ellenben körülfogja és az „ambient” hálózatok céljaira alkalmassá teszi őket. Ebből következik, hogy a GANS olyan információcserére alkalmas, melyet a ma létező protokollok nem támogatnak.

5. „Ambient” hálózatok kompozíciója

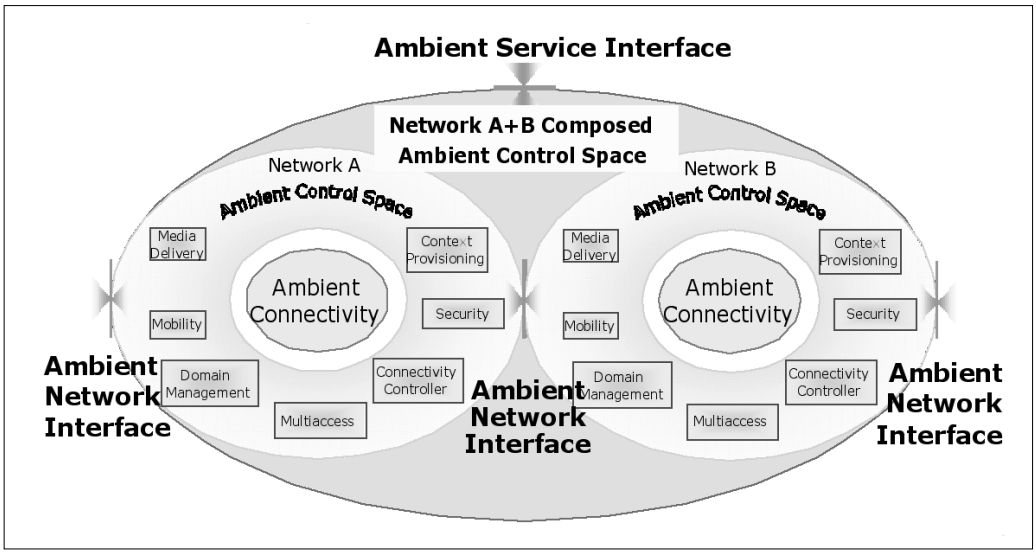
Különböző hálózatok dinamikusan és statikusan is összeolvadhatnak különböző célok érdekében. Például egy felhasználóhoz tartozó eszközök összeolvadhatnak annak érdekében, hogy személyi hálózatot formáljanak. Hasonlóan, különböző hozzáférési hálózatok összeolvadhatnak, hogy láthatatlan mobilitást és fejlett QoS-t támogassanak.

A hálózat kompozíció előre definiált szabályokat és lépéseket követ. Hogy létrejöjjön egy kompozíció, először hálózat, illetve szolgáltatás felderítést kell végezni. Ezután egy hitelesítési és meghatalmazási eljárást kell követni, hogy bizalmas kapcsolatot építhessen ki a két összeolvadni kívánó hálózat. Miután ez megtörtént, az „ambient” hálózatok egyezkednek és létrehoznak kompozíciós megállapodást a GANS-ot használván az ANI-n keresztül.

A kompozíciós megállapodás egy szerződés a kompozícióban résztvevő hálózatok között, mely tartalmaz minden, a kompozíciós időtartam alatt a hálózatok által követni kívánt szükséges és választható szabályt. Mind a kompozíciós megállapodáshoz szükséges egyezkedés és a megvalósítása teljesen automatikus, „plug-and-play”.

Az új összeolvadt „ambient” hálózatnak lesz egy közös ACS-e, közös ANI-val. A kompozíció teljesítése után a kialakult hálózat eltakarja a kapcsolódás részleteit a külvilágtól. Továbbá egy „ambient” hálózatnak jelenik meg a többi hálózat számára *(2. ábra)*.

Ezáltal egy felhasználói terminál, vagy egy felhasználó-központú hálózat (PAN) dinamikusan olvadhat össze más helyi és távoli hálózatokkal. Helyi hálózatok összeolvadhatnak, hogy nagyobb hálózatokat hozzanak létre. WLAN hálózat összeolvadhat egy cellás hálózattal, hogy új vezeték nélküli hálózatot alkossanak és hogy a felhasználók, illetve más hálózatok szemszögéből egy, homogén hálózatnak látszódjanak.



2. ábra
Két komponált „ambient” hálózat új interfészei: ACS, ANI és ASI [2]

6. „Ambient” hálózati architektúra

Az elosztott ACS architektúrában az önszerveződés és a dinamikus hálózati kompozíció alapja egy új hierarchikus virtuális hálózati modell. Ellenben a legtöbb csoportosító algoritmussal és „peer-to-peer” hálózattal, a javasolt hierarchikus virtuális hálózati struktúra korlátlan számú hierarchia szinttel rendelkezhet, mely skálázhatóvá teszi a rendszert. A virtuális hálózati struktúra szorosan kapcsolódik az alatta lévő fizikai hálózati topológiához. Mindemellett – a legtöbb csoportosító algoritmussal szemben – a virtuális hálózati topológia kialakításában nagy szerepet játszik az olyan hálózatmenedzsment információ, mint a hálózati policy [14].

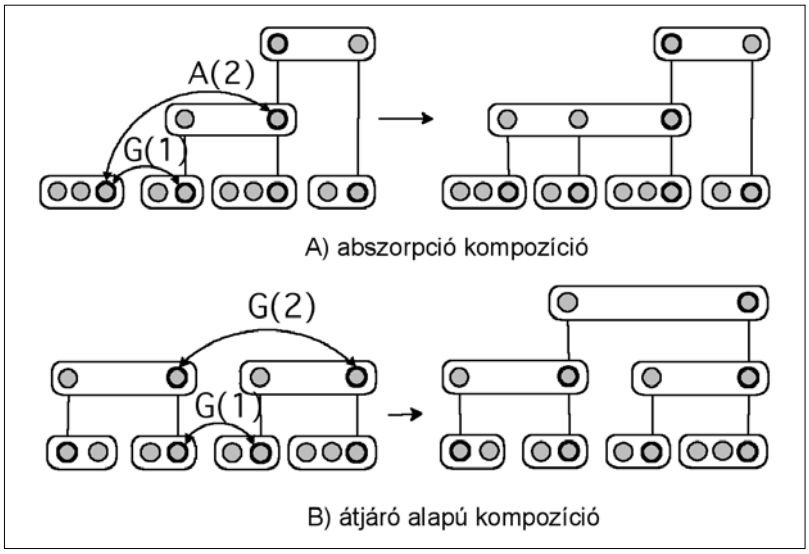
6.1. Hierarchikus ACS virtuális hálózatok

A hierarchikus modell alapvető építőelemei a virtuális hálózati csomópontok, a „peer”-ek, „super-peer”-ek és maguk a virtuális hálózatok. Egyik oldalról egy virtuális hálózat, egy „ambient” hálózatba tartozó „peer”-ek halmaza. Másrészről a virtuális hálózatok kiterjesztik az „ambient” hálózatokat virtuális kapcsolatokkal a résztvevő „peer”-ek között.

Mindegyik virtuális hálózat választ magának egy „super-peer”-t, hogy reprezentálja magát a külvilág felé. A „super-peer” mindössze a kompozíciós egyezkedésekért felelős, nincs semmilyen más kiváltsága a saját hálózatán belül. A „super-peer”-ek is formálhatnak virtuális hálózatokat magasabb hierarchia szinteken, ezáltal létrehozván egy hierarchikus virtuális hálózati struktúrát. A hálózati struktúra nem tartalmaz abszolút szinteket, azaz nem rendelhető a szintekhez egy indexszám. Ugyanakkor, a legalsó szintű virtuális hálózat minden csomópont számára definiált.

6.2. Hálózat kompozíciók

A hierarchikus virtuális hálózati gráf egyértelműen meghatározza a fizikai és logikai hálózati struktúrát, ennek következtében a hálózatok önszerveződése és a hálózat kompozíciók leírhatók eme gráf manipulálásaként.



3. ábra Alulról felfelé haladó kompozíció

A hálózati kompozíciók viselkedése és logikája a következő két alapelven működik:

1) az egybeolvadó vagy átjáró alapú kompozíció „peer-to-peer” egyezkedés alapján dől el,
2) a kompozíció a hálózati gráfban lentről felfelé halad.

Az első alapelv két, kompozíciós típust határoz meg: egybeolvadó (abszorpciós), vagy átjáró alapú kompozíciót.

Két hálózat akkor kapcsolódik össze *abszorpciós modell* szerint, ha kölcsönösen elfogadható policy-jük van és meg tudnak egyezni egy közös vezérlési tér (ACS) felállításában. Két virtuális hálózat abszorpciós kompozíciója egy új virtuális hálózatként fog megjelenni, melyet egy „super-peer” képvisel. Ez a „super-peer” lehet a korábbi „super-peer”-ek egyike, vagy az egyesített hálózatban újonnan választott „super-peer”.

Abban az esetben, ha a két hálózat nem tud közös vezérlési teret építeni (címütközés, nem elfogadható politika vagy egyéb hiba miatt) az átjáró alapú kompozíciót választják. Az *átjáró alapú kompozíció* során a két virtuális hálózat megtartja a saját ACS-ét, de egy újabb virtuális hálózati hierarchia szint keletkezik, melynek tagjai a két hálózat „super-peer”-jei. Az ehhez a szinthez rendelt ACS felelős a két hálózat közötti együttműködés biztosításáért és szabályozásáért.

A hierarchia szintek száma a virtuális hálózati struktúrában az átjáró alapú kompozíciók létrejöttével növekszik. A folyamat működése a második alapelven történik. Mikor két, előzőleg különálló hálózat találkozik, a legalsó szintű virtuális hálózatok a szomszéd-felderítési eljárás alapján érzékelik egymást. Miután felismerték, hogy különböző a legfelső szintű virtuális hálózatuk (az abszolút ősük), a legalsó szintek kompozíciót kezdeményeznek. Amennyiben meg tudnak egyezni abszorpcióban, végrehajtják azt. Más esetben a két „super-peer” továbbítja a kompozíciós kérést a következő, felsőbb szintre egészen addig, míg valamely szinten mindkét fél igent mond, avagy elérik a legfelső szintet. Az előbbi esetben, az egyik hálózat, teljes egészében beleolvad (abszorpcióval) a másik hálózat megfelelő szintjébe *(3/a. ábra)*. Az utóbbi esetben, ha a legfelső szinteket elérték, a két hálózat az átjáró alapú kompozíciót választja, és a két legfelső szintű „super-peer” létrehoz egy új, magasabb szintű virtuális hálózati szintet *(3/b. ábra)*.

A kompozíció típusát a két egyezkedő hálózat „super-peer”-jei döntik el. Abban az esetben, ha a két találkozó hálózat „super-peer”-jei nincsenek egymással kommunikációs kapcsolatban, a kompozíció elvégzéséhez szükséges üzenetek a legalsó szintű hálózatok közvetítő csomópontjain haladnak keresztül. Habár egy „super-peer” több szinten kezdeményezett hálózat kompozíciókban vehet részt, egy hálózat egyszerre csak egy kompozíciót kezelhet. Míg az éppen futó kompozíció be nem fejeződik, a többi kompozíciós kezdeményezés sorban áll.

A virtuális hálózatok dekompozíciója, szétválása szintén fontos feladata az architektúrának, a részleteit azonban e cikk keretében nem tárgyaljuk. Azt azonban érdemes megjegyezni, hogy a szétválás nem a kompozíció inverze, hanem egy teljesen különböző folyamat.

7. Összefoglalás

A cikkben bemutattuk az „Ambient Networks” IST projektet, céljait, elképzeléseit az „ambient” hálózati paradigma megvalósítására. Az olvasó megismerkedhetett a dinamikus hálózat kompozíció elvével és problémáival. Bemutattuk a hálózat kompozícióhoz kapcsolódó korábbi munkákat, a csoportosító algoritmusokat és a virtuális hálózatokat. Az olvasó megismerhette az „Ambient Networks” projektben használt hálózati komponenseket, interfészeket és funkcionális egységeket.

A továbbiakban bemutattunk egy saját fejlesztésű rendszert, amely képes skálázható módon kezelni a policy-ket is figyelembe vevő dinamikus, önszervező hálózat kompozíciókat. Ez a rendszer hierarchikus hálózati struktúrát épít fel két különböző kompozíciós modell segítségével, az abszorpciós és az átjáró alapú modellel.

Irodalom

[1] N. Niebert, H. Flinck, R. Hancock, H. Karl, C. Prehofer, “Ambient Networks – Research for Communication Networks Beyond 3G”, 13th IST Mobile & Wireless Communications Summit, Lyon, France, June 2004.

[2] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, U. Horn, C. Prehofer, H. Karl, “Ambient Networks: An Architecture for Communication Networks Beyond 3G”, IEEE Wireless Communication, pp.1536–1284, April 2004.

[3] R. Szabó, P. Kersch, B. Kovács, Cs. Simon, M. Erdei, A. Wagner, "Dynamic Network Composition for Ambient Networks: a Management View", Eurescom Summit, 2005.
[4] C. Kappler, P. Mendes, C. Prehofer, P. Poeyhoenen, D. Zhou, "A Framework for Self-Organized Network Composition", 1st International Workshop on Autonomic Communication, 2004.
[5] S. Basagni, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks", VTC 1999.
[6] Alan D. Amis, Ravi Prakash, Thai H.P. Vuong, Dung T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks", INFOCOM 2000.
[7] John Sucec, Ivan Marsic, "Clustering Overhead for Hierarchical Routing in Mobile Ad hoc Networks", INFOCOM 2002.
[8] Stoica, R. Morris, D. Karger, M. Kaashoek, H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", ACM SIGCOMM, 2001.
[9] S. Ratnasamy, M. Handley, R. Karp, S. Shenker, "A Scalable Content Addressable Network", SIGCOMM 2001.
[10] A. Rowstron, P. Druschel: Pastry, "Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems", IFIP/ACM International Conference on Distributed Systems Platforms 2001
[11] B. Y. Zhao, J. Kubiatowicz, A. D. Joseph, "Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing", Tech. Rep. UCB/CSD-01-1141, Comp. Science Division, University of California, Berkeley, April 2001.
[12] Marcus Brunner et al., "Sytem Management Research Challenges in Ambient Networks: a Synthesis study", MATA2004, Florianopolis, Brazil, October 2004.
[13] Cornelia Kappler et al, Ambient Network Deliverable 3-1 "Scenarios, Requirements and Concepts", July 2004.
[14] Róbert Szabó, Péter Kersch, Balázs Kovács, Csaba Simon, Márk Erdei, Ambrus Wagner, "Dynamic Network Composition for Ambient Networks: a Management View", Eurescom Summit 2005, Heidelberg, Germany, April 2005.

Köszönetnyilvánítás

Ez a dokumentum az Európai Bizottság által részben a 6. keretprogramban támogatott „Ambient Networks" projekt melléktermé. A dokumentumban nincs semmiféle garancia, hogy valóban alkalmas-e egy adott problématerület megoldására. A nézetek és következtetések a szerzők véleményét tükrözi és nem értelmezhetőek az „Ambient Networks" projekt, illetve az Európai Bizottság" általános hivatalos elképzeléseként, még explicit megjelenés esetén sem.

Hírek

A Debreceni Egyetemen átadásra került a biológia és szerves-kémia tudományterületeket, valamint a könyvtári és informatikai kiszolgáló tevékenységeket befogadó Élettudományi Épület és Könyvtár.

A mintegy 10 milliárd forintos beruházással, közel két év alatt létrehozott épület-komplexum az integrált intézmény egyben legnagyobb beruházása. A megvalósítás során az egyetem kiemelt figyelmet fordított a legmodernebb informatikai infrastruktúra kialakítására. Az épületekben a hang- adat és videoforgalom egységes, konvergens hálózaton zajlik. A Cisco Catalyst eszközökre épülő redundáns aktív gerinchálózati rendszer minden végpont számára egyenként legkevesebb 100 Megabit/másodperc kapcsolt adatátviteli sebességet biztosít, ami nemcsak a hagyományos adatok továbbítását, hanem az interaktív multimédiás alkalmazások kényelmes működését is lehetővé teszi, ezáltal az animációs oktatási elektronikus anyagok on-line használata is biztosított.

A két épületben az 1350, illetve 570 darab strukturált végpontot 87 kilométernyi kábel köti össze, amely Debrecenből számítva légvonalban három szomszédos országhatárt is elérne. A dolgozói, illetve oktatói szobákban lévő számítógépes hálózati rendszer a virtuális helyi hálózati (VLAN) technológiát alkalmazza, így az új épületekben lehetőség van bármely számítógép logikai átcsoportosítására, vagyis a számítógépek fizikai elmozdítása nélkül olyan belső elektronikus alkalmazások használhatók ezeken, amelyek adatvédelmi és adatátviteli teljesítmény szempontjából az adott konkrét igényeknek legjobban megfelelnek. Az új épület kábelezés szempontjából már elő van készítve a legmodernebb WiFi technológia kiépítésére is.

Az integrált informatikai rendszer kiemelkedő szolgáltatása a Cisco IP telefon rendszer, amely 340 darab folyadékkristályos kijelzővel rendelkező intelligens készülék segítségével biztosítja az intézményen belüli, illetve kívülre irányuló telefonálás lehetőségét. A telefonrendszer szervesen kapcsolódik az intézmény városi méretű hagyományos belső telefonhálózatához is. Az IP telefonok mindegyike eléri az egyetemi elektronikus telefonkönyv adatbázisát, ami lényegesen kényelmesebbé teszi a hívószámok keresését és tárcsázását.

Adatmenedzsment Ambient Control Space-ekben

KIS ZOLTÁN LAJOS

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

Kulcsszavak: adatmenedzsment, hash táblák, minta alapú menedzsment

A cikk bemutatja, hogy az Ambient Control Space adatmenedzsment mechanizmusa megvalósítható egy olyan keretrendszer segítségével, mely implementálja az elosztott hash táblákat – az indexelési, illetve adattárolási, archiválási feladatokhoz, valamint a minta alapú menedzsment üzenetküldési modelljét az adatok begyűjtéséhez illetve terjesztéséhez.

1. Bevezető

A mára széles körben elterjedő vezeték nélküli számítógép-hálózati technológiák új kutatási területeket hoztak a telekommunikációba. Az egyik ilyen terület a mobil, ad-hoc hálózatok dinamikus együttműködésének, önszerveződésének kutatásával foglalkozik.

Az EU által támogatott Ambient Networks projekt [1] is ennek a területnek vizsgálatára jött létre. Célja egy egységes, a felhasznált hálózati technológiáktól független keretrendszer létrehozása, mely lehetővé teszi a hálózati elemek önszerveződését. Az Ambient Network-ökben (körülölelő hálózatok) a hálózati elemek emberi beavatkozás nélkül képesek hálózatokat létrehozni és azokban kapcsolódni egymással. Továbbá ezek a hálózatok is képesek újabb, magasabb szintű hálózatokba szerveződni, így tetszőlegesen nagy sugarú és mélységű hálózatok jöhetnek létre.

Az Ambient Network-öket felépítő elemek egy közös vezérlő téren, az Ambient Control Space-en (körülölelő vezérlő tér) osztoznak. Ez egy egységes keretrendszert nyújt a hálózati elemeken futó vezérlési és menedzsment feladatokat ellátó funkcionális egységei számára *(1. ábra)*.

Egyrészt egy általános kommunikációs felületet biztosít, melyben a funkcionális egységek helyüktől és szintjüktől függetlenül képesek egymással kommunikálni, másrészt pedig ezeknek a funkcionális egységeknek egy általános adatmenedzsment szolgáltatást nyújt.

Jelen cikk ez utóbbi megvalósításának kérdéseit vizsgálja. Az adatmenedzsment szolgáltatás feladata a funkcionális egységek igényei szerint adatok szétterjesztése a hálózatban, adatok begyűjtése, aggregációja, illetve az adatok biztonságos eltárolása.

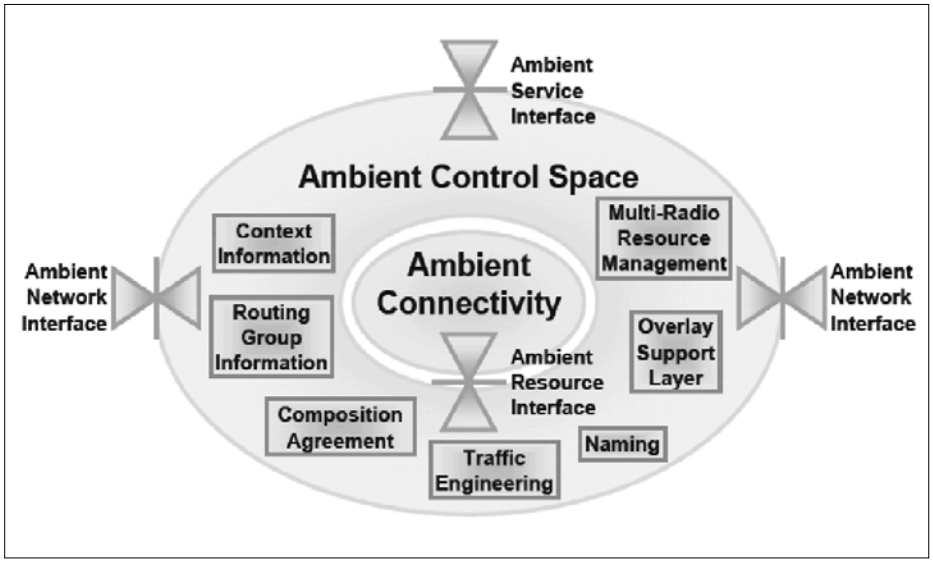
A következő fejezet áttekinti, hogy az Ambient Control Space-ekben milyen adatmenedzsment igények léphetnek fel, illetve ezeket milyen technikákkal lehet kielégíteni. A 3. és 4. fejezetek ezen a technikákat részletezik, majd a 5. fejezet megmutatja, hogyan szervezhetőek ezek egy közös keretrendszerbe. A cikk végül összefoglalással zárul.

2. Adat-hozzáférési modellek

Az adat-hozzáférési modell egy olyan – adatok köré épülő – leírás, amely megadja az adat felhasználásával kapcsolatos információkat. Ilyen információ lehet például az, hogy mely hálózati elemek hozhatják létre vagy férhetnek hozzá az adatot, vagy például, hogy szükséges-e az adat biztonságos tárolása. A fenti jellemzők lehetnek a priori ismertek, de akár statisztikai meghatározásuk is lehetséges.

Ha az általános problémát tekintjük, akkor természetesen végtelen sok jellemző létezhet. Azonban az Ambient Control Space esetében csak a vezérlési és

1. ábra Ambient Control Space



menedzsment adatok kezelésével kell foglalkoznunk, így elégséges ezen adattípusok lehetséges jellemzőinek az áttekintése.

2.1. A felhasználási modellek kategorizálása

Az Ambient Control Space szempontjából fontos hozzáférési modelleket négyféle kategória szerint csoportosíthatjuk: a modell szerinti forrás és cél szerint, illetve az általa érintett adatok maradandósága, valamint a modell igénybevételének módja szerint.

Forrás – Egy hozzáférési modellben azokat a hálózati elemeket tekintjük forrásnak, amelyek a modellben érintett adat egészét, vagy annak egy részét képesek előállítani, vagy frissíteni.

Eszerint az adat forrása lehet egyetlen vagy több hálózati elem. Az előbbi esetben természetesen az elem kiléte időben változhat, ám ennek transzparens kezeléséhez szükséges egy indexelő mechanizmus bevezetése. Az utóbbi esetben a hálózat tagjai különkülön csak egy-egy részét képesek előállítani az adatnak. Ez esetben az adat-részletek összegyűjtésének, aggregációjának megoldása is szükséges.

Cél – Egy modell céljai azok a hálózati elemek, amelyeknek szüksége lehet a modell adatainak aktuális tartamára, így képesnek kell lenniük arra, hogy igény szerint az adat aktuális értékét kinyerjék. Egy modell célja lehet egyetlen hálózati elem, például egy megválasztott hálózat-felügyelő elem, melyet szintén egy indexelő mechanizmus segítségével kezelhetünk transzparensen. Itt is előfordulhat azonban, hogy több tag is érdekelt az adatokban aktuális értékében; ez esetben gondoskodni kell a változások igény szerinti megfelelő szétterjesztéséről.

Maradandóság – Egy adatelem maradandó, ha az akkor is érdekes lehet a modell céljai számára, ha a forrás elemek elhagyták a hálózatot. Ez esetben nem elégséges az adatok forrásokban való tárolása, hanem szükség van azok redundáns eltárolása, hogy bármilyen hálózati hiba esetén azok továbbra is rendelkezésre álljanak.

Igénybevétel – Az ACS által menedzselt adatok értéke folyamatosan változik, ezért az egyes modellekben folyamatos frissítési illetve lekérdezési műveletek lesznek. Ezek gyakoriságának ismerete (vagy becslése, mérése) elengedhetetlen, hiszen arányuk alapvetően befolyásolja az alkalmazott menedzsment technika kiválasztását. Például egy olyan modellben, melyben a frissítések száma lényegesen kisebb, mint a lekérdezéseké, a választott megoldás a frissítés után az új érték azonnali szétterjesztése, míg ellenkező esetben a hagyományos lekérdezéses módszer alkalmazása lehet kifizetődő.

Ezek alapján például az Ambient Network policy-ját [8] a következő modell írhatja le:

Forrás: a hálózat minden egyes eleme
Cél: az aktuálisan megválasztott képviselő elem
Maradandóság: ideiglenes (bármilyen hálózati változás esetén megváltozik az értéke)
Igénybevétel: közel egyforma számú frissítés és lekérdezés (mindkét esemény a hálózat kompozíciójának változásából fakad)

2.2. Adatmenedzsment módszerek

A kategóriák áttekintése után összefoglalható, hogy milyen adatmenedzsment feladatok megoldására van szükség, illetve, hogy ezek milyen technikákkal valósíthatóak meg az Ambient Control Space-ben.

Először is szükséges egy indexelési technika alkalmazása, mely segítségével az egy-forrású, illetve -célú modellekben az ideiglenesen megválasztott elemek transzparens módon megtalálhatóak. Ennek a feladatnak a megvalósítására a napjainkban igen népszerű elosztott hash táblák (Distributed Hash Table) [2,3] a legalkalmasabbak.

Több forrású modellek esetén megoldandó az adatrészletek összegyűjtése, azok változásainak követése, illetve az így összegyűjtött adat-elemek aggregálása, majd célba juttatása. Erre kínál megoldást a minta alapú menedzsment (Pattern Based Management) [4], mely a hálózatban rejlő elosztott számítási kapacitás segítségével skálázható módon párhuzamosítja az adatok gyűjtését és aggregációját,

A több célú modellekben az igénybevételi aránymutatótól függően szükség lehet többesadás illetve üzenetszórás jellegű üzenetszórási megoldásokra. Ezekre szintén jó megoldást adnak a minta alapú menedzsmentben alkalmazott technológiák.

Végezetül szükség van a maradandó adatok redundáns tárolására, illetve az így eltárolt adatokhoz való transzparens hozzáférés biztosítására. Erre szintén az elosztott hash táblák, ígérnek megoldást.

Látható, hogy két módszer, az elosztott hash táblák illetve a minta alapú menedzsment alkalmazásával minden támasztott igény kielégíthető. A következő két fejezet ezeket a technológiákat mutatja be, az azt követő pedig ezek lehetséges keretrendszerbe foglalását.

3. Elosztott hash táblák

Az elosztott hash táblák nagyméretű hálózatok transzparens adattárolásának megoldására jöttek létre. Lényegében egy leképzést valósítanak meg az adatok azonosítói és a hálózati elemek között, így minden egyes hálózati elem tudhatja, hogy egy bizonyos kulcsú adat melyik hálózati elemen van eltárolva. Segítségükkel a rájuk épülő alkalmazások egy egységes programozói interfészen érhetik el a kívánt adatokat, függetlenül attól, hogy azt melyik hálózati elem tárolja aktuálisan. A programozói interfész lényegében megegyezik a hagyományos hash táblák által biztosítottal, vagyis egy egyedi azonosítóval rendelkező adat értékét tudja írni, illetve olvasni segítségével az alkalmazás.

3.1. Jelenlegi megoldások

Az elosztott hash táblák kutatásának elsődleges területe az adatok visszakeresésének idejének illetve forgalmának miminalizálásával foglalkozik. Éppen ezért, bár számos implementáció létezik, azok főleg az útválasztási algoritmusukban illetve az adatok hálózati elemekhez rendelésében különböznek.

Ezek a hálózat menedzsment szempontból fontos kérdésekre sajnos nem adnak megoldást. A hálózati elemek kiesésével csak az útválasztás robosztusságának biztosítása miatt foglalkoznak, de az elemek kiesésével elvesző adatok minimalizálásával nem törődnek. Ezen kívül nem támogatják a hash táblák összeolvadását, esetleg csak egy központosított megoldás erejéig.

Látható, hogy az elosztott hash táblák, mint elv jó megoldást nyújtanak, ám szükség van a jelenlegi megoldások módosításokra, hogy az megfeleljen minden Ambient Control Space számára fontos követelménynek.

3.2. ACS hash-tábla

A fentebb említett problémák kiküszöbölésére az elosztott hash táblát a funkciók különválasztása miatt két rétegre – egy vezérlő és egy adattároló – osztottuk fel *(2. ábra)*. A vezérlő réteg felelős a hálózati elemek egységben tartásáért, illetve az elosztott hash tábla vezérléséért, monitorozásáért. A vezérlő réteg feladata még, hogy a begyűjtött adatok alapján a hálózat bizonyos elemeit adattároló funkcióra is kijelölje, amelyek így a másik – adattároló – rétegbe is csatlakoznak. A külvilág csak a vezérlő réteggel van kapcsolatban, így annak további feladata a hagyományos elosztott hash tábla interfész biztosítása. A kívülről érkező parancsokat azután ez továbbítja az adattároló réteg megfelelő elemei felé, illetve az onnan érkező válaszokat a külvilág felé. Ezáltal a külvilág számára teljesen rejtett marad az adattároló réteg. Ez lehetőséget ad arra, hogy egy vezérlő réteg több adattároló réteget is fenntartson, amivel egyszerűbbé válik a hálózatok összeolvadásának megoldása. Az adattároló réteg pedig csak az adatok tárolásával foglalkozik, így könnyebbé vált annak redundánssá tétele.

3.3. A vezérlő réteg

A vezérlő réteg feladata – a külvilággal való kapcsolattartáson kívül – a hálózat integritásának megőrzése. A hálózat egészét átlátja így, lehetősége van különböző statisztikák gyűjtésére (mint például a hálózati elemek száma, tárolt kulcsok száma). Ezek alapján a képes annak megítélésére, hogy a hálózat mely elemeinek kell részt vennie az adattároló rétegben.

Hálózatok összeolvadásakor az elosztott hash táblának minél hamarabb konzisztens képet kell mutatnia a funkcionális elemek számára. Az ACS hash tábla ezt úgy oldja meg, hogy a vezérlőrétegek képesek gyorsan összeolvadni, majd az új vezérlőréteg mindkét adattároló réteget képes párhuzamosan fenntartani *(3. ábra)*, illetve azok összeolvadását levezényelni. Ez alatt az átmeneti időszak alatt a beérkező kéréseket mindkét adattároló réteg felé továbbítja.

3.4. Adattároló réteg

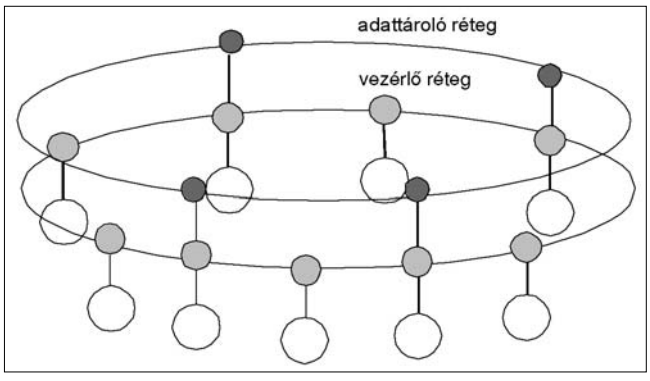
Az adattároló réteg feladata az általános elosztott hash tábla megoldásokkal szemben csak az adattárolás illetve indexelés, hiszen a hálózat fenntartás feladatának egy részét átveszi a vezérlő réteg. Az adattároló réteg feladata továbbá, hogy a vezérlő rétegtől érkező adatokat redundánsan tárolja el, vagyis önállóan hozzon létre belőle megfelelő számú másolatot. Hasonlóan, elemek kiesése esetén ennek a rétegnek a feladata a kiesett másolatok helyett újak létrehozása.

4. Minta alapú menedzsment

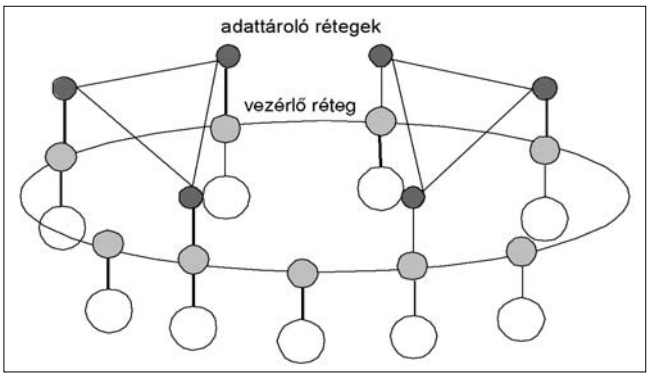
A mobil ügynök architektúra [5] jelentős változást hozott a hálózat menedzsment számára, hiszen segítségével az addig központosítottan végrehajtott feladatok elosztottá tétele jelentősen leegyszerűsödött. Később ennek az általános architektúrának számos specializált változata jött létre (mint például az aktív hálózatok [6], programozható hálózatok [7]), melyek mind a mobil ügynök elv más-más aspektusára helyezik a hangsúlyt.

A minta alapú menedzsment (Pattern Based Management) is egy hasonló specializált változat, mely a hálózati elemek által nyújtott számítási kapacitás párhuzamos felhasználásában rejlő lehetőségeket igyekszik kihasználni. A koncepció alapeleme a navigációs minta, ami lényegében azt írja le, hogy az adott menedzsment funkció milyen módon terjedjen szét a hálózatban, illetve hogyan térjen vissza az összegyűjtött infor-

2. ábra ACS hash tábla



3. ábra ACS hash tábla – összeolvadás után



mációval. A konkrét menedzsment feladat így ráültethető az annak elvégzéséhez legmegfelelőbb navigációs mintára, aminek segítségével az végrehajtódik a hálózatban.

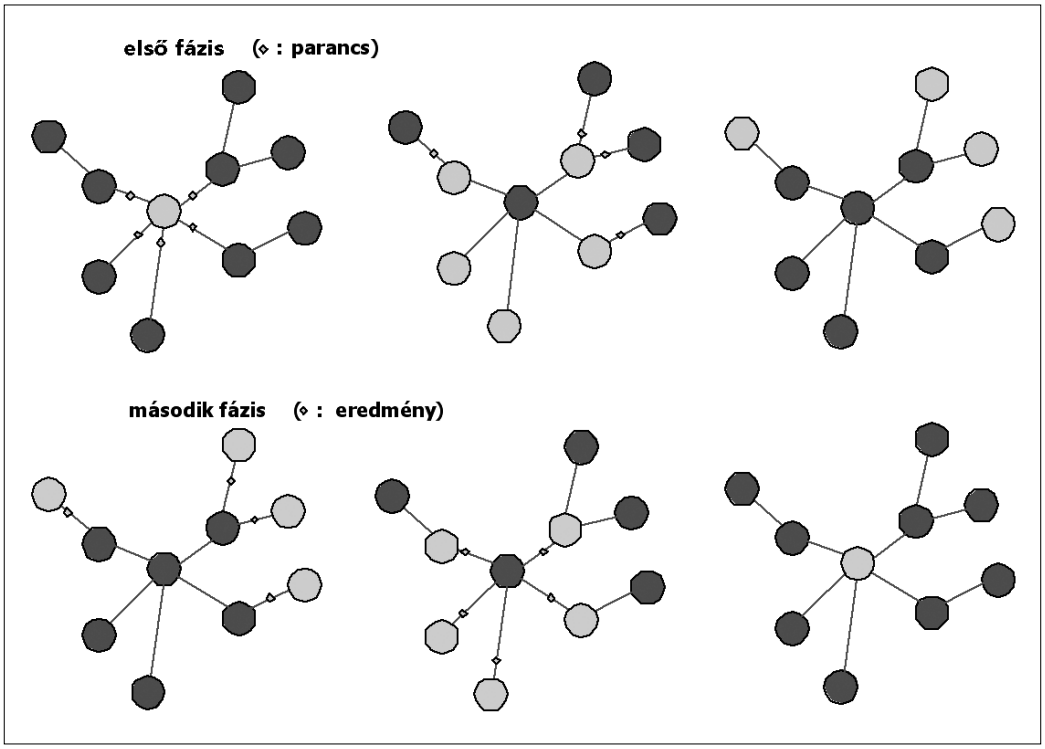
A számítási feladatok általában állandóak, így az üzenetek csak az aktuális állapotokat tartalmazzák, programkódot nem. Továbbá az alapvető minták csak egy ugrásos kommunikációt használnak, így nincs szükségük útválasztó algoritmusok futtatására. Ez utóbbi két tulajdonság miatt a minta alapú menedzsmenttel lényegesen gyorsabb futás érhető el a többi megoldáshoz képest.

4.1. Minta alapú üzenetszórás

A minta alapú menedzsment során felfedezett egyik leghasznosabb navigációs minta a visszhang (echo) minta [4], amely igen sokrétű felhasználhatóságot ígér. Az alap visszhang minta a PIF (Propagation with Information Feedback – terjesztés információ visszacsatolással) algoritmuson alapszik. Működése két fázisból áll *(4. ábra)*. Az első fázisban a menedzsment funkciót végrehajtó elem elárasztja a hálózat egy részét (vagy akár egészet) egy adott menedzsment paranccsal. Az elárasztás, mivel csak szomszédok közötti kommunikációt használ gyors, és sávszélesség-hatékony is.

A második fázis akkor kezdődik, amikor a minta elért olyan elemeket, amelyek már nem tudják azt tovább terjeszteni (mert minden szomszédjuk már megkapta a mintát). Az első fázis tulajdonképpen egy feszítőfát húz ki az aktuális fizikai topológiára, és a második fázis ennek mentén gyűjti be a válaszokat. Az egyes csomópontok megvárják az összes függő választ, és azt csak egyben, aggregálva küldik tovább.

4. ábra Visszhang minta



Ez a minta skálázható megoldást kínál mind az információk szétterjesztésére, mind pedig a pull-jellegű menedzsment feladatok skálázható végrehajtására. Ez a tulajdonsága teszi ideálissá az Ambient Control Space-ben való alkalmazásra, hiszen megoldhatóvá teszi az adatok szétterjesztését, illetve az igény szerinti, egész hálózatra kiterjedő menedzsment feladatok végrehajtását.

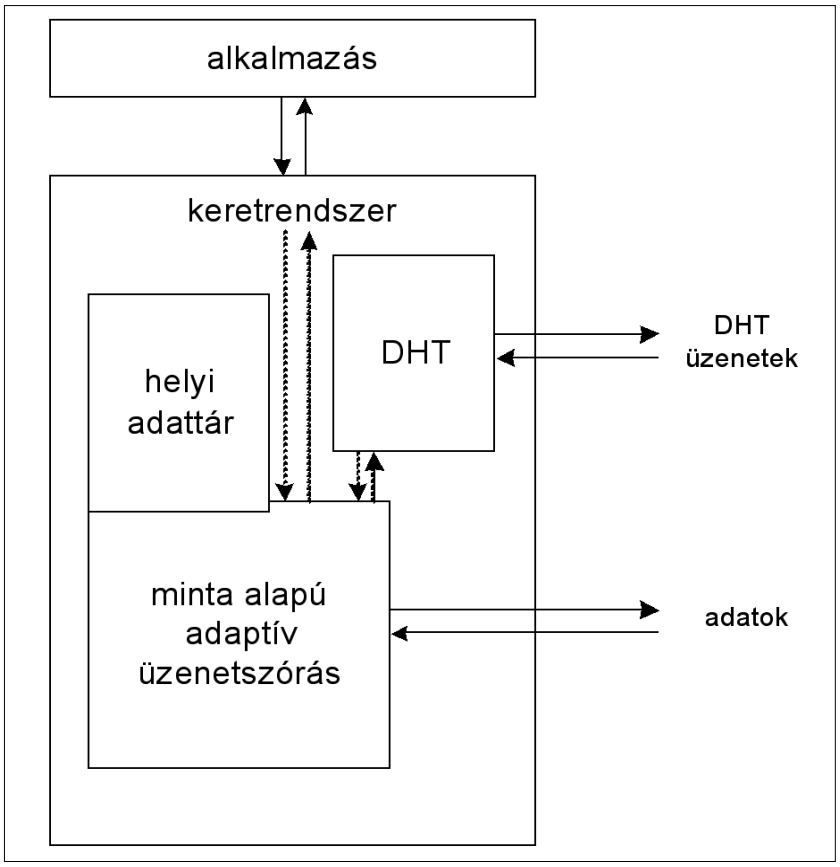
4.2. Minta alapú adaptív üzenetszórás

Az adatok frissülésének és lekérdezésének arányától függően a modell optimális kiszolgálásához másmás technikát kell alkalmaznunk. Az egyik szélsőséges esetben – amikor az adat igen ritkán frissül, ám arra igen sűrűn szüksége van az elemeknek, a legcélravezetőbb, ha frissüléskor üzenetszórással tájékoztatjuk az elemeket. Ellenkező esetben – amikor az információ igen sűrűn frissül, ám csak ritkán van rá igény, a legcélravezetőbb a hagyományos lekérdezés.

A két szélsőség közti átmenet kiszolgálásához megoldást nyújt a két módszer kombinálása, melyben a forrástól bizonyos (fizikai szintű) környezetéig többesadás jelleggel jutna el az információ, az ezen a környezeten kívüliek pedig a hozzájuk legközelebb eső tagtól kérdezik.

A rendszer elindításakor az információ automatikusan senkinek sem továbbítódik, mindenki kérés-válasz jelleggel igényli azt. Ebből a forrás annyit érzékel, hogy melyik fizikai szomszédján át mennyi lekérdezés érkezik hozzá. Amennyiben úgy ítéli meg, hogy valamelyik szomszédtól nagyobb arányú igény érkezik, mint az adat frissítése, akkor ettől kezdve azt a szomszédot automatikusan értesíti amikor az adat frissül. Így egyrészt csökkentik az egymás közti forgalmat, másrészt az eddig erről érkező lekérdezésekre mostantól az értesített elem is tud válaszolni, így mindenki egy ugrással rövidebb távolságból kapja meg a választ. Hasonlóan, az így értesített elemek is továbbíthatják a frissített értékeket a szomszédaiknak, amennyiben úgy ítélik, hogy szükséges.

Ezáltal megoldható az adatok optimális és skálázható frissítése, hiszen a hálózat nagyságától függetlenül a „belső” elemek csak a szomszédjaiknak továbbítják a változásokat, és csak a hálózat szélein működnek lekérdezések.



5. ábra ACS keretrendszer

4.3. Minta alapú változás-követés

Egy másik minta, mely az echo egyfajta általánosítása, a GAP (Generalized Aggregation Pattren – általános aggregáló minta). Ez – az echo-val ellentétben – a push típusú megközelítést alkalmazza. Segítségével lehetőség nyílik a hálózati elemek adat változásának folyamatos, elosztott megfigyelésére.

Az alap GAP működése hasonló az echo mintáéhoz, mindössze annyiban tér el, hogy miután az első fázis lezajlott, a második fázist minden egyes hálózati elem elindíthatja, ha olyan változást észlel az adataiban, vagy kap a szomszédjától, melyet jelentenie kell.

A GAP az echo mintához hasonlóan kiváló megoldást nyújt az Ambient Control Space-ben a több forrású adatok skálázható frissítésére.

5. ACS adatmenedzsment keretrendszer

Szükség van még a fenti két ismertetett technika egységes keretrendszerbe foglalására, amely egy általános interfészt biztosít az ACS-ben működő funkcionális elemek számára *(5. ábra)*. Az interfész két parancsa természetesen az adat értékének frissítése, illetve lekérdezése.

Frissítés esetén a keretrendszer első lépésben az elosztott hash táblába regisztrálja az adott hálózati elemet, mint forrást, illetve az adatot eltárolja a keretrendszer az elemen futó részében, illetve ha a hálózati elem kérte, az elosztott hash táblában is eltárolja az értéket. Amennyiben több elem is beregisztrálta magát forrásként, a keretrendszer kijelöl egy ideiglenes aggregátor elemet, amely kiépít a forrásokhoz egy-egy útvonalat, majd a GAP minta segítségével gyűjti tőlük az adatokat.

Lekérdezés esetén a keretrendszer kikeresi az elosztott hash táblából a forrást, majd lekérdezi tőle az aktuális értéket. Az adatok küldése minden esetben az adaptív üzenetszóró rendszeren keresztül történik.

Az így kialakult keretrendszer skálázhatóságát biztosítja, hogy az azt alkotó technikák is skálázhatóak.

Irodalom

[1] N. Niebert, H. Flinck, R. Hancock, H. Karl, C. Prehofer, "Ambient Networks – Research for Communication Networks Beyond 3G," 13th IST Mobile & Wireless Communications Summit, Lyon, France, June 2004.

[2] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan., "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," ACS SIGCOMM, San Diego, USA, August 2001.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shekner, "A Scalable Content-Addressable Network," ACS SIGCOMM, San Diego, USA, August 2001.

[4] K.-S. Lim, R. Stadler, "Developing Pattern-Based Management Programs," MMNS, Chicago, USA, October 2001.

[5] A. Silva and J. Delgado, "The agent pattern for mobile agent systems," 3rd European Conference on Pattern Languages for Programming end Computing, 1998.

[6] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, Gary J. Minden, "A Survey of Active Network Research," IEEE Communications Magazine, Vol. 35, No.1, pp.80–86., January 1997.

[7] A. T. Campbell, H. G. De Meer, M. E. Kounavis, K. Miki, J. B. Vicente, D. Villela, "A Survey of Programmable Networks," ACM SIGCOMM Comp. Commun. Rev., April 1999.

[8] Erdei M., Wagner A., „Policy keretrendszer – adatmodell és algoritmusok – dinamikus hálózatkompozíciók automatizált tárgyalási folyamatához", Híradástechnika, 2005/7.

Policy keretrendszer dinamikus hálózatkompozíciók automatizált tárgyalási folyamatához

ERDEI MÁRK, WAGNER AMBRUS

Budapesti Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék
{merdei, awagner}@hit.bme.hu

Kulcsszavak: policy keretrendszer, adatmodell, policy-kezelő algoritmusok

Az utóbbi években a vezetéknélküli hálózati technológiák széleskörű elterjedésének köszönhetően intenzív kutatás indult az Ambient Network (AN) témakörében. Az AN-ek célja az, hogy a felhasználó számára átlátszó módon biztosítson kapcsolatot többféle hálózati technológiát felhasználva. Ezen cél elérése érdekében az AN-ek automatikusan konfigurálják az egyes hálózati elemeket, így egy AN a tartalmazott elemek management domainjeként szolgál. A vezetéknélküli hálózatok dinamizmusa szükségessé teszi az AN-ek folyamatos karbantartását elemek hozzáadása és eltávolítása, illetve AN-ek kompozíciója és dekompozíciója révén. Ideális esetben ez a karbantartási tevékenység a háttérben, emberi beavatkozás nélkül, önszervező módon történik. Az AN-ekkel kapcsolatos kutatás az Európai Közösség „Wireless World Initiative – Ambient Networks" projektje keretében zajlik [1,2]. A cikk az AN-ek kompozícióját irányító, úgynevezett policy keretrendszert mutatja be.

1. Bevezető

A széles körben használt hálózati technológiák sokfélesége rugalmas konfigurációs és karbantartási megoldások létrehozását teszi szükségessé. Egy természetes megközelítés az alacsonyszintű konfiguráció származtatása a felhasználó és a hálózat adminisztrátor igényeit és szándékait leíró magasszintű definíciókból.

Ezen szándékok kifejezésének egyik lehetséges módja a policy-k használata. Az irodalomban több különböző felhasználást, értelmezést és megvalósítást találunk. Policy-kat használnak például a DiffServ erőforrásgazdálkozásban [3], a hierarchikus IP menedzsmentben a hálózatmenedzsment viselkedés futásidejű kiterjesztéséhez [4], szolgáltatás elemek magasszintű kontextus információ alapján történő adaptációjához [5], és emberi beavatkozás nélküli hozzáférésvezérléshez [6].

A policy-k szabványos tárolása és kezelése iránti igényre reagálva az IETF kidolgozott egy információs modellt [7,8], és meghatározta a policy menedzsment terminológiáját [9]. Az ezen cikkben bemutatott policy keretrendszer nagy vonalakban szintén az IETF terminológiát követi. Mivel a legtöbb létező megoldás specifikus alkalmazásokhoz készült, dinamikus hálózatkompizícióval kapcsolatos kutatásunkhoz egy új policy keretrendszert dolgoztunk ki, melynek kidolgozásakor fontos szempont volt a széleskörű alkalmazhatóság.

Cikkünkben bemutatjuk a policy adatmodell felépítését és szemantikáját, a policy-k és a kontextusok kapcsolatát, végül a kapcsolódó algoritmusokat tárgyaljuk.

2. A policy-k szerepe a dinamikus hálózatkompozícióban

Az AN koncepcióban a peerek (felhasználók) hierarchikus overlay (csoport) struktúrákat alkotnak, melyek menedzsment tartományoknak felelnek meg. Egy csoportot hasonló tulajdonságú és igényű peerek alkotnak. Magasabb szintről szemlélve egy csoport egyetlen olyan peernek tekinthető, amely egyesíti magában a csoport tagjainak jellemzőit. A csoportok így hierarchikus struktúrákba rendezhetők.

A hierarchia legalsó szintjén lévő peerek csoportokba, majd felsőbb szinteken csoportok csoportjaiba szerveződnek.

Ha két csoport megközelíti egymást, kompozíciót hajthatnak végre, ha tulajdonságaik és szándékaik (melyeket policy-k fejeznek ki) elegendően hasonlóak.

Minden peernek van policy-ja, melyek egy csoport policy-ba egyesíthetők. A csoport-policy a csoportban lévő policy-k esszenciáját hivatott kifejezni. Az alábbiakban definiált policy keretrendszer a policy adatmodellt, a kapcsolódó algoritmusokat, illetve a kontextusok és policy-k közötti kapcsolatot definiálja.

Minden peer rendelkezik egy policy adatbázissal, amely egy, vagy több policy-t tartalmaz. Minden pillanatban ezen policy-k egyike az aktív policy. Az aktív policy kiválasztása a peer kontextusa alapján történik. A peer egy policy halmazzal rendelkezik, az alkalmazható policy-k halmazát több tényező határozza meg:

– hardver és szoftver környezet,
– hálózati környezet,
– hely információ stb.

A peer az aktív policy-t az alkalmazható policy-k halmazából választja ki.

3. Adatmodell

Az adatmodell némiképp hasonló az irodalomban leírtakhoz [10,11]. A policy-t három elem határozza meg *(1. ábra)*:

– profil, állítások, szabályok.

3.1. Profil

A profil olyan tulajdonságok halmaza, amelyeket a peer fontosnak tekint a többi peerrel történő tárgyalás során. Minden tulajdonságot egy-egy kulcs-érték pár ír le. Sem a kulcsokra, sem az értékekre nincs semmiféle típus-, vagy értékbeli megkötés, tetszőleges szövegek. Előre definiált kulcsok létezhetnek, de definiálhatnak kulcsokat a peerek is.

Példák:
– „memory-capacity” = „10M”
– „company-meeting-participant” = „yes”
– „nationality” = „Hungarian”

3.2. Állítások

Az állítások az adott peer és a tárgyalópartner profilja közötti relációkat fogalmaznak meg. Egy állítás három részből áll:
– a tárgyalópartner profiljából származó kulcs,
– relációs operátor,
– az adott peer profiljából származó kulcs, vagy konstans.

Példák:
– „software-version” >= „software-version”
– „company-meeting-participant” = „yes”

A tárgyalás kimenetelére gyakorolt hatás		Az állítás értéke	
		igaz	hamis
Minősítő	MUST	nincs	kudarc
	SHOULD	nincs	nincs
	DON’T CARE	nincs	nincs
	SHOULD NOT	nincs	nincs
	MUST NOT	kudarc	nincs

1. táblázat Minősítők

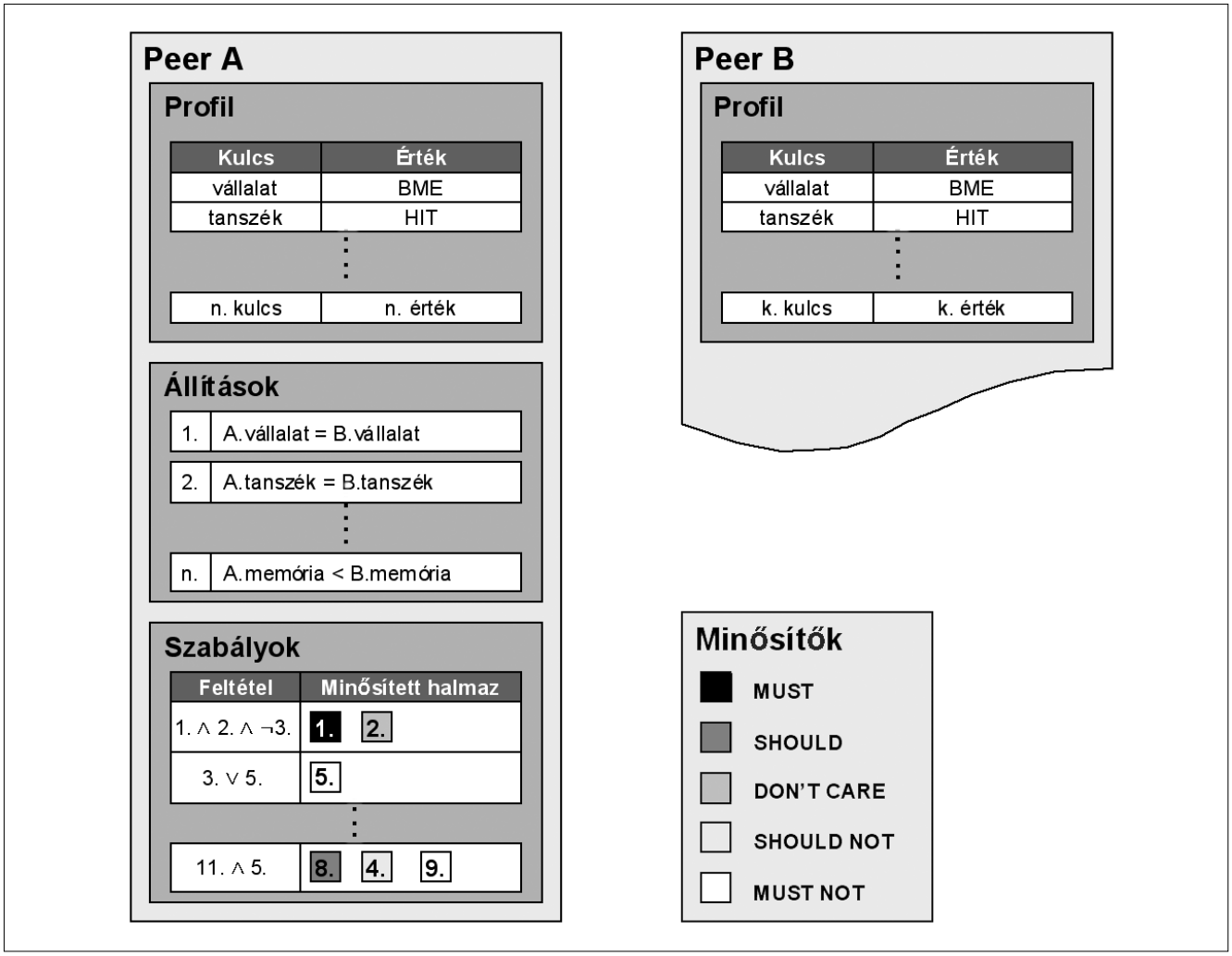
3.3. Szabályok

A szabályok a tárgyalási algoritmus paraméterei és két részből állnak:
– feltétel,
– minősített halmaz.

A feltétel egy logikai kifejezés. Ez a kifejezés állításokra való hivatkozásokat és logikai operátorokat tartalmaz. Behelyettesítve a tárgyalópartner profil információját, a feltétel kiértékelése igaz vagy hamis eredmény ad.

A feltétel határozza meg, hogy a szabályt alkalmazzuk-e a tárgyalás során, vagy nem.

1. ábra A policy adatmodell



A minősített halmaz állításokra vonatkozó minősített hivatkozások halmaza. A minősített halmaz minden hivatkozott állítást az alábbi minősítőkkel lát el:

– MUST (ragaszkodom hozzá),
– SHOULD (előnyben részesítem),
– DON'T CARE (érdektelen),
– SHOULD NOT (hátrányban részesítem),
– MUST NOT (nem fogadom el).

A minősítők a kiértékelt állítás sikeres tárgyaláshoz szükséges logikai értékét határozza meg. Az *1. táblázat (előző oldalon)* definiálja a minősítők jelentését.

A SHOULD és SHOULD NOT minősítők egyenértékűek a DON'T CARE minősítővel a tárgyalás során, a policy karbantartó algoritmusok azonban másként kezelik őket. Annak érdekében, hogy a policy konzisztens legyen önmagával, a szabályok közötti ellentmondásokat fel kell deríteni új szabály létrehozásakor.

Két szabály ellentmond, ha létezik az állítások olyan kiértékelése, hogy mindkét szabály feltétele igaz, és van olyan állítás, amelyet a két szabály ellentmondóan minősít. A meglévő szabályoknak ellentmondó szabályt nem lehet hozzáadni a policy-hoz.

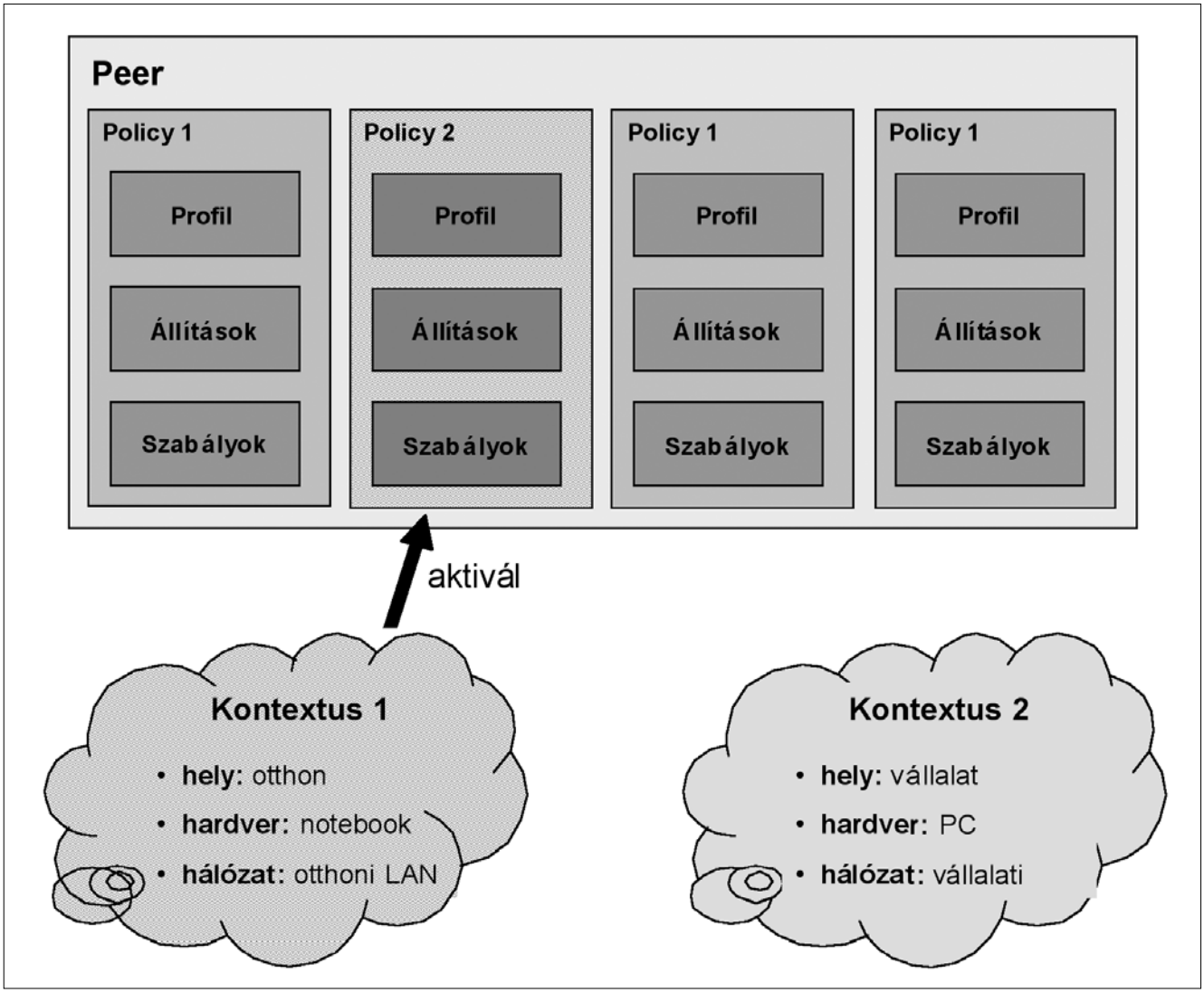
4. Összefüggés a policy-k és a kontextusok között

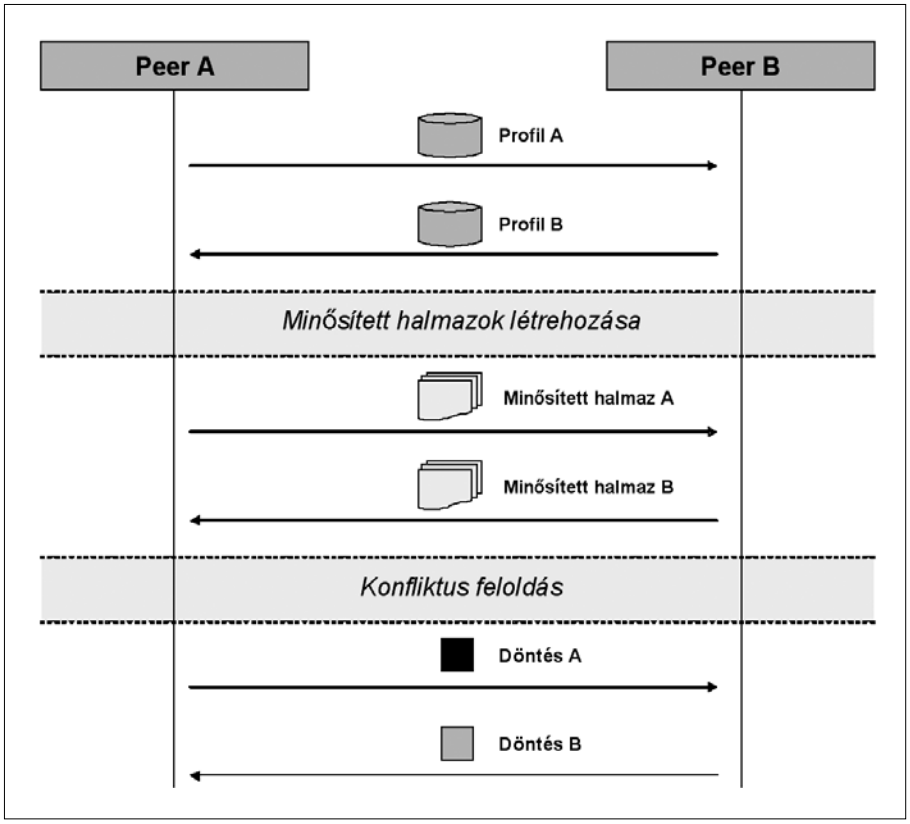
Egy peernek több policy-ja is lehet. A rendelkezésre álló policy-k közül egyik az aktív policy. Minden pillanatban pontosan egy aktív policy-nak kell lennie. A megfelelő policy kiválasztásában a felhasználót a kontextus fogalma segíti.

A kontextus a felhasználó környezetének különböző szempontjairól tartalmaz információt. Néhány a lehetséges szempontok közül:

- **Hely információ.** Információ a felhasználó tartózkodási helyéről. Lehetőségek: otthon, iroda, közlekedés stb. Ha hozzáférhető, a vezeték nélküli hálózatok vagy egy GPS vevő által adott információ tovább finomíthatja a kontextust.
- **Hardver/szoftver konfiguráció.** Információ az aktuálisan használt számítógépes platformról. Lehetséges értékek: asztali gép, notebook, PDA, mobiltelefon stb.
- **Hálózati kapcsolatokra vonatkozó információk.** Otthoni vagy irodai hálózat, vezetékes, vagy vezeték nélküli kapcsolat, rendelkezésre álló sávszélesség, ár stb.

2. ábra Kontextusok és policy-k





3. ábra Tárgyalási algoritmus

A fenti információk alapján a használható policy-k köre szűkíthető. A felhasználó a szűkített körből választja ki az aktív policy-t (illetve a kiválasztás automatikusan megtörténik, ha a szűkített kör már csak egy elemből áll). Ahhoz, hogy meghatározhassuk, egy adott kontextusban aktiválható-e egy policy, a policynak definiálnia kell a minimális, kontextussal szembeni elvárásait.

Az alábbi szituáció egy példa kontextusok használatára *(2. ábra)*. Egy dolgozó otthon és a munkahelyén is a notebookját használja. Hazaérkezik, bekapcsolja a notebookot, a megváltozott kontextus pedig két policyra szűkíti a használható policy-k körét:

– személyes, amely elérhetővé teszi számára a családi levelezést, és a családdal és barátokkal történő chat-elést, illetve
– otthoni iroda, amely összeköti őt a vállalati hálózattal a rendelkezésre álló szélessávú kapcsolaton keresztül.

A felhasználónak ekkor egy párbeszédablakban kell választania a fenti két lehetőség közül.

5. Policy-kezelő algoritmusok

5.1. Tárgyalási algoritmus

A tárgyalási algoritmus feladata, hogy a két kommunikáló peer policy-ját összehasonlítsa, és eldöntse, hogy azok kölcsönösen elfogadhatóak-e egymás számára. A peerek nem specifikus tárgyalási algoritmust használnak, hanem mindketten ugyanazt a szabványos algoritmust különféleképpen felparaméterezve. A rendszer felépítésétől függően, az alábbinál komplexebb tárgyalási algoritmusokra is szükség lehet [12].

Az alapvető algoritmus lépései az alábbiak *(3. ábra)*:

1) Minden policy állítás minősítésének semlegesre (DON'T CARE-be) állítása.
2) A peerek kicserélik egymás között a profiljukat.
3) Minden szabály feltétele kiértékelésre kerül. Ha a feltétel teljesül, a szabályban szereplő állítások a megadott minősítőkkel bekerülnek az összevont minősített halmazba.
4) A peerek összehasonlítják egymás minősített halmazait.
5) Ha a minősített halmazok között ellentmondás van, a peerek ezt megpróbálhatják feloldani a konfliktus-feloldó algoritmus segítségével.
6) A peerek a tárgyalást sikeresnek tekintik, ha a folyamat végén nincs ellentmondás a minősített halmazok között.

5.2. Csoport policy

Egy csoport policy-ja a csoportban lévő entitiások policy-jainak aggregátuma. Az egyes peerek személyes policy-jaiból a csoport policy kezelő algoritmus képezi a csoport policy-ját. A csoport policy-ja a csoportot alkotó felhasználói halmaz változásának és az egyes felhasználók policy-jai változásának megfelelően folyamatosan alkalmazkodik a csoport pillanatnyi állapotához, a csoport pillanatnyi képét tükrözi.

A csoport policy azokban a helyzetekben kerül felhasználásra, amikor egy szituáció során a csoportot egyetlen kommunkáló félként kell reprezentálni (például amikor egy újabb csomópont szeretne a csoporthoz csatlakozni, akkor a tárgyalási algoritmus az új csomópont és a csoport policy-ja felett fog lefutni, hasonlóképpen két AN kompozíciójakor is a tárgyalási algoritmus a két csoport policy-t fogja használni).

5.2.1. A csoport policy működése

A csoport policy ugyanazokból a főbb elemekből épül fel, mint egy felhasználói policy (profil, állítások, szabályok). Az adatszerkezet ezen felül még ki van egészítve a profil tulajdonságainak, az állításoknak és a szabályoknak a kardinalitásával.

A kardinalitás azt írja le, hogy a csoporton belül hány felhasználó rendelkezik ugyanazokkal a profiltulajdonságokkal, állításokkal illetve szabályokkal. Ez kifejezi az adott leíró fontosságát is a csoporton belül.

Tehát a csoport policy kismértékben eltér az egyszerű felhasználói policy-tól. Ennek következtében vagy a tárgyalási algoritmust kell módosítani úgy, hogy kezelni tudja a kardinalitásokat is (a felhasználói policyban ebben az esetben minden elem kardinalitása egy), vagy a csoport policy-t le kell képezni egy egyszerű felhasználói policy-ra mielőtt a tárgyalásra sor kerülne. Ebben az esetben a csoport policy-ban előforduló kis kardinalitású elemeket el kell távolítani (ezek ugyanis nem a csoport jelentős részének álláspontját képviselik), valamint az egymásnak ellentmondó elemek között fel kell oldani a konfliktust (például egyszerű többségi döntés elvén).

A második esetben nem elegendő a leképzést egyszer elvégezni, hiszen ebben az esetben elvesztenénk a számossági információkat, melyekre szükség lehet, ha csomópontok lépnek ki a csoportból, vagy csatlakoznak a csoporthoz (ebben az esetben ugyanis megváltozhatnak az erőviszonyok amit a csoport policy-nak is tükröznie kell), hanem minden tárgyalás elindulása előtt, vagy minden csoportbeli változás után újra el kell végezni a leképzést.

5.2.2. Csoport policy kezelő algoritmus

A csoport policy kezelő algoritmus feladatai az alábbiak:

– az újonnan csatlakozó felhasználó policy-jának a csoport policy-ba való beolvasztása,
– a csoport policy állapotának frissítése, ha egy felhasználó távozik a csoportból, és
– a csoport policy folyamatos közelítése a csoportot alkotó felhasználók közösen megfogalmazható álláspontjához.

5.2.3. Beolvasztási algoritmus

A beolvasztási algoritmus az egyes felhasználói policy-kat olvasztja be a csoport policy-ba. Az algoritmus lépései az alábbiak:

1. A felhasználói profil elemeit hozzáadja a csoport profilhoz. Az újonnan létrejövő elemek kardinalitása egy, a már létező elemek kardinalitását eggyel növeli.
2. A csoport profilból eltávolítja az egymással ütköző profil elemeket. E célból kiszámítja az egymással ütköző profil elemek kardinalitásának arányát (0 és 1 közötti érték). A kapott értéktől függően kétféleképpen kerülhet sor a profil elem eliminálására:
   a) Ha az arány kisebb mint egy megadott küszöbérték, akkor a kisebb kardinalitású profil elemet törli a profilból (a kevésbé fontosat),
   b) Egyébként mindkettőt törli (egymásnak ellentmondóak, és mindkettő a csoport jelentős hányadát reprezentálja, tehát egyiket sem lehet felhasználni a csoport leírására).
3. Az állítások fontossági súlyát két tényező határozza meg: egyrészről a saját kardinalitásuk, valamint azoknak a profil elemeknek a kardinalitása, melyekre hivatkoznak. A konfliktusban lévő állításokat az algoritmus eltávolítja a csoport profilból a profil elemeknél használt algoritmushoz hasonló eljárással.
4. A szabályok esetében szintén a fontosság szerinti kiemelésre, valamint letisztításra van szükség, hogy az eredményül kapott szabályhalmaz minél jobban tükrözze a csoport közös álláspontját. A szabályok letisztítása az alábbi lépésekben történik:
   a) Az egyes szabályok fontossága a bennük szereplő állítások fontosságából származik.
   b) A szabályok feltételrészéből a kis fontosságú illetve esetlegesen a korábbi lépések során már törölt állításokat eltávolítja az algoritmus.
   c) Hasonlóképpen a kis fontosságú illetve korábban törölt állításokat a szabály minősített halmazából is törli.
   d) A szabályok között fennálló ellentmondások feloldása a minősített halmazukban szereplő minősítők alapján történik. Két szabály potenciális konfliktusban van, ha a feltételrészüknek létezik olyan kiértékelése mely esetén mindkettő szabály életbe lép. Valódi konfliktus akkor lép fel, ha a két szabály minősített halmazában ugyanaz az állítás szerepel eltérő minősítéssel. Ekkor az ellentmondó minősítésekből eredő eltérést fel kell oldani. A feloldás a *2. táblázat* alapján történik.
   e) Azok a szabályok, amelyek a folyamat végére üresek lettek, törlésre kerülnek.

5.2.4. A csoportot elhagyó felhasználók kezelése

Ha egy vagy több felhasználó elhagyja a csoportot, akkor ezt a csoport policy-nak tükröznie kell. Mivel a felhasználói policy-k nem vonhatók ki a csoport policy-ból, ezért ebben az esetben a csoport policy-t újra fel kell építeni a bent maradt felhasználók policy-jaiból.

2. táblázat Policy beolvasztási szabályok (= többféle feloldás lehetséges)*

Feloldás	MUST NOT	SHOULD NOT	DON'T CARE	SHOULD	MUST
MUST	szabály eldobása	*	*	*	N/A
SHOULD	*	*	SHOULD	N/A	
DON'T CARE	MUST NOT	SHOULD NOT	N/A		
SHOULD NOT	*	N/A			
MUST NOT	N/A				

6. Összefoglalás

A fentiekben bemutattunk egy policy keretrendszert, melyet dinamikus hálózatkompozícióval kapcsolatos kutatásainkhoz dolgoztunk ki. A keretrendszer egy adatmodellből, és a hozzá kapcsolódó algoritmusokból áll.

Ez a policy keretrendszer lehetővé teszi, hogy közvetlen felhasználói beavatkozás nélkül – a magasabb szintű felhasználói szándékok alapján –, automatikusan menjenek végbe a hálózat-kompozíciós döntések. Továbbá, a csoport policy bevezetésével lehetővé válik ezen döntési folyamatok automatikus kiterjesztése magasabb hierachia szintekre is.

Az ismertetett policy keretrendszer egy az Ambient Networks projektben [1,2] megvalósított Ambient Networks hálózatmenedzsment [14] rendszer prototípus részeként került megvalósításra [15].

A kutatási célok között szerepel az adatmodell kiterjesztése és altalánosítása (például hierarchikus profilok kezelésével). A profiltulajdonságokra adattípusok és típusspecifikus operátorok definiálása (ez jelentősen növelni fogja a tárgyalási és beolvasztási algoritmusok mozgásterét és az adatok kezelését is kifinomultabbá fogja tenni) [13].

Ezen felül a dinamikus karbantartási feladatok (például távozó felhasználók esetén a csoport policy frissítése) további vizsgálata és optimalizálása szükséges teljesítőképesség, érvényesség és koherencia szempontjából.

Irodalom

[1] N. Niebert, H. Flinck, R. Hancock, H. Karl, C. Prehofer, "Ambient Networks – Research for Communication Networks Beyond 3G", 13th IST Mobile & Wireless Communications Summit, Lyon, France, June 2004.

[2] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, U. Horn, C. Prehofer, H. Karl, "Ambient Networks: An Architecture for Communication Networks Beyond 3G", IEEE Wireless Comm., pp.1536–1584., April 2004.

[3] P. Flegkas, P. Trimintzios, G. Pavlou, A. Liotta, "Design and Implementation of a Policy-Based Resource Management Architecture", Proceedings of IEEE/IFIP Integrated Management Symposium (IM'2003), Colorado Springs, USA, pp.215–229., Kluwer, March 2003.

[4] P. Flegkas, P. Trimintzios, G. Pavlou, I. Andrikopoulos, C.F. Cavalcanti, "On Policy-based Extensible Hierarchical Network Management in QoS-enabled IP Networks", Proceedings of the IEEE Workshop on Policies for Distributed Systems and Networks (Policy 2001), Bristol, UK, M. Sloman, J. Lobo, E. Lupu, eds. pp.230–246., Springer, January 2001.

[5] J. Keeney, V. Cahill, "Chisel: A Policy-Driven, Context-Aware, Dynamic Adaptation Framework", Proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks, Lake Como, Italy, pp.3–15., June 2003.

[6] V. G. Bharadwaj, J. S. Baras, "Towards Automated Negotiation of Access Control Policies", Presented at the IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy, June 2003.

[7] B. Moore, E. Ellesson, J. Strassner, A. Westerinen, "Policy Core Information Model – Ver.1 Specification", IETF RFC 3060

[8] B. Moore ed., "Policy Core Information Model (PCIM) Extensions", IETF RFC 3460

[9] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn. S. Herzog, A. Huynh, M. Carlson, J. Perry, S. Waldbusser, "Terminology for Policy-Based Management", IETF RFC 3198

[10] G. Patz, M. Condell, R. Krishnan, L. Sanchez, "Multidimensional Security Policy Management for Dynamic Coalitions", DARPA Information Survivability Conference and Exposition 2001 (DISCEX II), Anaheim, CA, USA, June 2001.

[11] A. Di Ferdinando, P. McKee, A. Amoroso, "A Policy Based Approach for Automated Topology Management of Peer To Peer Networks and a Prototype Implementation", IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy, June 2003.

[12] K. C. Feeney, D. Lewis, V. P. Wade, "Policy Based Management for Internet Communities", IEEE 5th International Workshop on Policies for Distributed Systems and Networks (POLICY 2004), Yorktown Heights, New York, June 2004.

[13] A. Sahai, S. Singhal, V. Machiraju, R. Joshi, "Automated Generation of Resource Configurations through Policies", IEEE 5th International Workshop on Policies for Distributed Systems and Networks (POLICY 2004), Yorktown Heights, New York, June 2004.

[14] M. Brunner, et al, "Ambient networks management challenges and approaches", In Proceedings of the First International Workshop on Mobility Aware Technologies and Applications (MATA2004), Florianopolis, Brazil, 2004.

[15] R. Szabó, P. Kersch, B. Kovács, Cs. Simon, M. Erdei, A. Wagner, "Dynamic network composition for ambient networks: A management view", In Eurescom Summit 2005: Ubiquitous Services and Applications Exploiting the Potential, Heidelberg, 2005.

MULTIMEDIA COMMUNICATIONS

Local adaptation of multimedia services employing cross-layer communications

Key words: wireless networks, inter-layer communications

Over wireless networks, real-time service delivery is a highly non-trivial task as the varying channel conditions make it difficult to meet the strict delay constraints. To achieve high user-perceived quality, adaptive applications and adaptation-supportive network entities play a key role. Recently, cross-layer adaptation is getting acceptance as a method that efficiently increases the performance of wireless communications. We present a cross-layer signaling framework that can provide the necessary information for service adaptation.

Real-time speaker normalization method and its application in the SpeechMaster speech impediment therapy

Key words: speech recognition, speech improvement

It is well known that applications of various speaker-normalization methods can significantly improve the accuracy of speech recognizers. These kinds of methods imply a set of procedures aiming at vocal tract length normalization (VTLN). With the support of the Hungarian Ministry of Education as part of an IKTA project a computer aided software package called SpeechMaster was developed for speech impediment therapy and the teaching of reading. The system is based on both real-time phoneme recognition and visual feedback. Therefore, in SpeechMaster it is not possible to use VTLN methods that require the analysis of the speech signal lasting for several seconds – at least in their original form.

New protocol to implement wireless MIDI connections in a Bluetooth system

Key words: wireless MIDI, Bluetooth

A new protocol is presented that supports the implementation of wireless MIDI connections in a Bluetooth system. Taking into account the practical applications, the protocol allows for virtually arbitrary connection topology. The design of a general Bluetooth-based MIDI system is presented together with the data transmission methods, delay is calculated and the limits of applicability is investigated, proposing possible extensions of the system functions.

NETWORK PROTOCOLS

Approximate analysis of a two-class WFQ service

Key words: WFQ service, 2D Markov-chain, expected value and deviation of delay

The class based weigthed fair queueing is a popular way to share a common resource in computer and telecommunication systems. To find an efficient analytical solution has been an open question for a long time. Different solutions were proposed, but all have their limits in usability. In this paper we present a simple approach that provides a fast approximation for the mean and the variation of the queue length and the waiting time.

Secure Communications

Key words: IT security, secure processing, box structure

The technical background of reliable, confidential processing has become a key issue, because ensuring intellectual property rights is questionable on digital transfer mediums without a reliable platform and toolkit. Securing user rights is an important issue, too. The mathematical background does exist, and with the advent of the age of quantum processors practical answer can be given to the theoretical questions. In our paper we would like to summarize the theoretical questions of secure processing starting from a high level, and provide guidelines to the elaboration of such specific systems, that, when used, will not harm either party's interest.

Design of low-consumption sensors

Key words: integrated temperature sensors, low-consumption circuits, current reference circuit

The power consumption of circuits can be reduced by making possible the switch-off of unused units. The consumption is considerably lower if only those circuit elements get power which are just executing an operation or storing value. In certain cases the switch-off and re-start of analogue circuits is not a trivial task. This article depicts a solution for the switch-off and re-start of a temperature dependent current reference. In addition, two solutions for helping the minimization of power consumption are introduced.

AMBIENT NETWORKS

Ambient Networks – an overview

Key words: "ambient" network paradigm, absorption and gateway-based composite models

Nowadays the spread of wireless technologies and mobile devices shape networking visions of ambient and persistent mobile networks that mobile users are able to have resort to. Since ambient networks are heterogeneous, distributed and dynamically forming networks, new and sophisticated network management architecture is needed to cope with the special network preferences. In our paper we reveal a peer-to-peer management architecture, which creates management domains by dynamically organizing network components into hierarchical management overlay networks.

Data management in Ambient Control Spaces

Key words: data management, hash tables, sample-based management

Ambient Networking is a novel network architecture for networks beyond 3G. Ambient Control Space is a common control space distributed throughout the composed Ambient Networks. The paper examines the requirements the Ambient Control Space should meet, and by the outcome the paper proposes an architectural framework based on currently available data management techniques.

Policy subsystem for automated negotiation of dynamic network compostitions

Key words: policy framework, data model, policy-management algorithms

The widespread adoption of wireless networking technologies during recent years has fueled intense research in the field of Ambient Networks (AN). ANs provide network access for users transparently across a variety of technologies. In a dynamically changing environment ANs continuously perform background self-management tasks including the composition and decomposition of ANs ideally with no human intervention. Policies are used in a number of applications where automated decision making is required. This paper presents the policy subsystem governing AN composition in an AN prototype developed as part of the EU research project Wireless World Initiative – Ambient Networks.

Summaries • of the papers published in this issue