híradástechnika

1945 VOLUME LXII. 2007

hírközlés ■ informatika

Többrétegű optikai hálózatok

WiFi alapú helymeghatározás

Biztonsági API analízis

2007/8

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

Tartalom

Előszó

szabo@hit.bme.hu

Jelen számunk válogatás az utóbbi időszakban a lap számára beküldött és részben bírált cikkekből. Amint az az alábbi rövid bemutatásokból látszik, a cikkek témái széles spektrumot ölelnek fel. Ebben a számunkban egy új törekvés első lépése is látszik: szeretnék rendszeresen bemutatni a hazai kutatási-fejlesztési projektek eredményeit, elsőként most az Aitia International Zrt. és a BME Távközlési és Médiainformatikai Tanszék munkatársai által kifejlesztett érdekes rendszert.

Nagysebességű TCP-protokollok együttműködésének modellezésével foglalkozik *Simon Boglárka, Sonkoly Balázs és Molnár Sándor* cikke. A hagyományos TCP torlódásvezérlésében jelentkező problémák miatt nagysebességű és nagy kiterjedésű hálózati környezethez a közelmúltban több új, nagysebességű TCP verziót fejlesztettek ki, mint például a HighSpeed TCP és a Scalable TCP. A cikkben szabályozástechnikai modellezés alapú eredményeket ismertetnek a szerzők.

A vezetéknélküli hálózati végpontok mozgása mellett bizonyos esetekben egy alhálózat is változtathatja a helyét, ennek tipikus példája a járműveken belüli, együtt mozgó hálózatrész (mozgó hálózat). Az IETF Network Mobility (hálózat-mobilitás) csoportja a Mobil IP-hez hasonlóan kezeli ezt a kérdést, amely azonban a mozgó hálózatból fakadóan összetettebb probléma, mint az önálló végpontok mobilitása. *Kanizsai Zoltán, Rózsás Balázs és Imre Sándor* cikkükben a mozgó hálózatok mobilitás-támogatásával kapcsolatos eredményeket tekintik át a Mobil IP-ből kiindulva.

Perényi Marcell, Soproni Péter és Cinkler Tibor dinamikusan változó multicast fákkal foglalkoznak kétrétegű optikai hálózatokban. A levél-csomópontok állandó váltakozásával a fa egyre távolabb kerül az optimális topológiától, ezért sok hálózati erőforrás és költség takarítható meg a fa rendszeres újrakonfigurálásával, az optimális topológia visszaállításával. Vizsgálják eredményességét több dinamikus útvonalválasztó algoritmus és az újrakonfigurálási intervallum hosszának függvényében is.

A web már régóta felvet adatvédelmi kérdéseket a látogatók számára is: bizonyos szolgáltatók megfigyelik, követik a felhasználók tevékenységeit, adatbázist építenek ízlésvilágukról. Az anonim böngészők megoldást kínálnak a felhasználóknak; elrejtik őket a figyelő szemek elől. *Gulyás Gábor és Schulcz Róbert* bemutatnak néhány követésre használt módszert, illetve egy új, az anonimizáló szolgáltatásokra vonatkozó konstrukciós paradigmát, majd egy ez alapján értelmezett osztályozási rendszert is az anonim böngészők besorolásához.

A vezetéknélküli számítógéphálózatok használata közben a felhasználó szabadon helyet változtathat, ehhez kapcsolódóan alakult ki a felhasználó pozíciójától függő szolgáltatások köre. Ehhez szükségessé válik egy helymeghatározó rendszer kialakítása, amely beltérben is használható és megfelelő pontossággal rendelkezik ahhoz, hogy az arra épülő alkalmazások igényeit kiszolgálja. A *Németh László Harri, Kis Zoltán Lajos és Szabó Róbert* által kifejlesztett WLANpos megoldás célja egy Wi-Fi hálózat és egy szabványos Wi-Fi eszköz segítségével a vevő, azaz a felhasználó helyének lehető legpontosabb meghatározása volt, amely jobb az eddigi megoldásoknál, amelyek általában drágák, nagy számítási igénnyel rendelkeznek, vagy csak korlátozott térben alkalmazhatók.

Tartalomszolgáltatási alkalmazásoknál kulcskérdés a tartalom védelme, amelynek egyik módszere a vízjelezés (watermarking). *Oláh István* cikkében összefoglalja a videó-vízjelezés sajátosságait és bemutat egy olyan videó vízjelezési eljárást, ami ellenáll a H.264/AVC tömörítésnek és a legáltalánosabb jelfeldolgozási módosításoknak.

Napjaink kommunikációs hálózatainak gyakran nincs kiépített fix infrastruktúrája (pl. ad-hoc hálózatok, ambient intelligencia hálózatok vagy szenzorhálózatok). Ezek a hálózatok nagymértékű önállósággal, autonómiával rendelkeznek, s gyakran akár önző módon is viselkedhetnek. Hogy megszüntessük, illetve mérsékeljük az önző viselkedést a hálózatban, egy elosztott keretrendszer válik szükségessé, amely ösztönzi a résztvevőket a kommunikációra és az együttműködésre. A *Németh László Harri és Szabó Róbert* által vizsgált megoldás a hálózati topológia figyelembevételében különbözik az eddigiektől és egy újszerű megoldást mutat be erre a problémára.

Az API szintű támadások komoly veszélyt jelentenek a hardver biztonsági modulokra nézve, ezért fontos követelmény az API-ban rejlő biztonsági lyukak felfedezése és foltozása. Az API analízis egyik ígéretes iránya a formális verifikációs módszerek alkalmazása. *Buttyán Levente és Ta Vinh Thong* cikkükben ezt az irányt követik, s egy processz-algebra alapú API verifikációs módszert javasolnak, mely különösen alkalmasnak látszik a biztonsági API-k működésének formális leírására, a biztonsági követelmények precíz definiálására és a megfogalmazott követelmények teljesítésének ellenőrzésére.

Végül *Tatai Péter, Varga Pál és Marosi Gyula* mutatnak be egy távközlő hálózatok üzemi állapotainak folyamatos figyelésére, monitorozására alkalmas rendszert. A rendszer lehetővé teszi, hogy a hálózat forgalmi- és hívásstatisztikái alapján segítséget nyújtson a hálózat üzemeltetőjének a hálózat skálázható, dinamikus növelésére a szolgáltatások egyre bővülő választéka mellett.

Szabó Csaba Attila
főszerkesztő

Nagysebességű TCP protokollok együttműködésének modellezése

SIMON BOGLÁRKA, SONKOLY BALÁZS, MOLNÁR SÁNDOR

BME Távközlési és Médiainformatikai Tanszék, HSNLab
simonbogi@gmail.com, molnar@tmit.bme.hu

Lektorált

Kulcsszavak: HSTCP, STCP, transzport protokoll modellezés, fairness

A hagyományos TCP (Transmission Control Protocol) torlódásvezérlésében jelentkező problémák miatt nagysebességű és nagykiterjedésű hálózati környezethez a közelmúltban több új, nagysebességű TCP verziót fejlesztettek ki. Ilyen többek között a HighSpeed TCP és a Scalable TCP. Bár több kutatás foglalkozott teljesítményanalízisükkel, számos nyitott kérdés maradt még a működésükkel kapcsolatban. Nagyon fontos kérdés az, hogy a különböző protokollok mennyire képesek igazságos együttműködésre (fairness). Ebben a cikkben szabályozástechnikai modellezés alapú eredményeket ismertetünk. A hálózatot egy visszacsatolt rendszerként értelmezve vizsgáltuk a nagysebességű protokollok együttműködését. Megadtuk a hálózat elemeinek – TCP források, szűk sávszélességű link és RED algoritmus – folyadékmodelljét. Az analitikusan nehezen kezelhető, bonyolult differenciálegyenlet-rendszerek numerikus approximációval történő megoldására terveztünk és implementáltunk egy MATLAB/Simulink környezetet és ebben vizsgáltuk a különböző folyamok egymásra hatását. Eredményeinket Ns-2 szimulációkkal validáltuk. Ennek eredményeképp pontosabb és részletesebb tudásbázist hoztunk létre a nagysebességű protokollok alapvető tulajdonságairól, előnyeiről és hátrányairól.

1. Bevezetés

A TCP napjainkban a leggyakrabban használt végpontok közötti szállítási protokoll az Interneten, melynek robbanásszerű terjedése és gyorsan növekvő kihasználtsága miatt a hálózaton már a kezdeti időkben szükségessé vált a torlódások és a csomagvesztések elkerülése. Az általunk hagyományos TCP-nek tekintett TCP Reno protokoll torlódásmegelőzési fázisában AIMD (Additive Increase Multiplicative Decrease) algoritmust használ. Ez az algoritmus eggyel növeli az ablakméretet (W) minden beérkező nyugta esetén és felezi azt, ha csomagvesztés történik. Ez a torlódásvezérlési algoritmus napjainkra nem nyújt hatékony működést nagysebességű, nagykiterjedésű hálózati környezetben, mert ablaknövelése túl lassú, míg az ablak méretének felezése torlódás esetén túl drasztikus megoldás. A hatékony működés érdekében a közelmúltban számos javaslat született, amelyek főként azt próbálják elérni, hogy a TCP torlódásvezérlési mechanizmusa rugalmasabb, dinamikusabb legyen és minél jobb hatásfokkal kihasználja a rendelkezésre álló kapacitást, sávszélességet. Ezek közül az egyik a csomagvesztés alapú algoritmusok fejlesztése, mint például a HighSpeed TCP (HSTCP) [1] vagy a Scalable TCP [2], egy másik a késleltetés alapú algoritmusok csoportja, ilyen a FAST TCP.

Az új protokollok elterjedéséhez alapvető fontosságú a más protokollokkal való igazságos együttműködés. Erősen vitatott kérdés, hogy a kezdeti késleltetés változtatása milyen hatással lehet a versenyző nagysebességű protokollok hosszútávú együttműködésére [6]. Munkánk során nagysebességű TCP protokollok együttműködését vizsgáltuk a torlódásszabályozási algoritmusok modellezése alapján.

Vizsgálatainkat aktív sorkezelési mechanizmust alkalmazó hálózatokra végeztük. Fontos hangsúlyozni, hogy a vizsgálatok során a RED (Random Early Detection) algoritmust használtuk, nem az úgynevezett Drop Tail-t, így eredményeink eltérnek a Drop Tail-nél tapasztaltaktól. Szabályozáselméleti terminológiát követve egy aktív sorkezelési mechanizmust alkalmazó TCP hálózat egyes komponensei egy visszacsatolt szabályozási kör egyes blokkjaival azonosíthatók [3-5].

Az egyes elemek működése, a használt algoritmusok és az egymásra hatások jól leírhatók analitikusan differenciálegyenlet-rendszerrel. A visszacsatolt rendszert leíró meglehetősen bonyolult differenciálegyenlet-rendszereket – melyek változó idejű késleltetéseket tartalmaznak bizonyos argumentumokban, illetve rekurzív összefüggéseket írnak le – implementáltuk egy általunk kialakított MATLAB/Simulink környezetben. Modelljeinket különböző hálózati elrendezések és beállítások mellett csomagszintű szimulációs vizsgálatokkal validáltuk (Ns-2) [7].

2. A HSTCP és Scalable TCP protokollok torlódásvezérlése

A HSTCP és a Scalable TCP újabb verziójú csomagvesztés alapú protokollok, melyek a rossz ablakdinamikán javítandó módosított AIMD-t használnak, melynek segítségével az ablakméret (W) növelése gyorsabb, míg torlódás esetén való csökkentése kevésbé drasztikus.

A Scalable TCP [2] módosított AIMD algoritmusa az MIMD (Multiplicative Increase Multiplicative Decrease) jóval dimamikusabb működést eredményez, így az átvitel hamarabb felgyorsul, a linkek kihasználtsága pedig rövid időn belül eléri a maximális értéket. Az algoritmus

aW-vel növeli az ablakméretet 1 helyett (ahol *a* egy paraméter és *W* a torlódási ablak mérete) egy-egy nyugtára és nem felezi azt csomagvesztés esetén, hanem csupán az 1/8-ával csökkenti. Ez a legagresszívabb, legrobusztusabb TCP protokoll. A Scalable TCP működését összevetve a hagyományos TCP-vel azt tapasztalhatjuk, hogy a Scalable TCP nagy sebességű hálózatokon jobb helyreállítási időket produkál és dinamikusabban képes kihasználni a hálózatok kapacitását. A protokollt úgy tervezték meg, hogy biztosítsa az erőforrások megosztását a linken, és mindezek mellett stabil és rugalmas legyen a hálózati körülményekkel szemben.

A HighSpeed TCP [1] szintén a TCP Reno torlódásvezérlési algoritmusára épül, csak jóval dinamikusabbra lett tervezve és ennek érdekében módosították az általa használt AIMD algoritmust. A protokoll egy bizonyos ablakméret *(low_W)* alatt a hagyományos TCP-nek megfelelően működik, tehát minden nyugtára eggyel növeli az ablakméretet és minden csomagdobásra felezi azt, *low_W* felett pedig két új paramétert (*a*-t és *b*-t) vezet be, ahol *a* az ablaknövelő, *b* az ablakcsökkentő paraméter. A hagyományos TCP-nél ez a két paraméter *a*=1 és *b*=0,5 volt. Az említett változók a HighSpeed TCP esetében több új paramétertől és a W aktuális értékétől függenek. Ennek köszönhetően a torlódási események gyakoriságának csökkenésével a HSTCP átviteli sebessége nagyobb mértékben növekszik, így adott átviteli sebességet több torlódási esemény mellett is el tud érni, szemben a hagyományos TCP-vel.

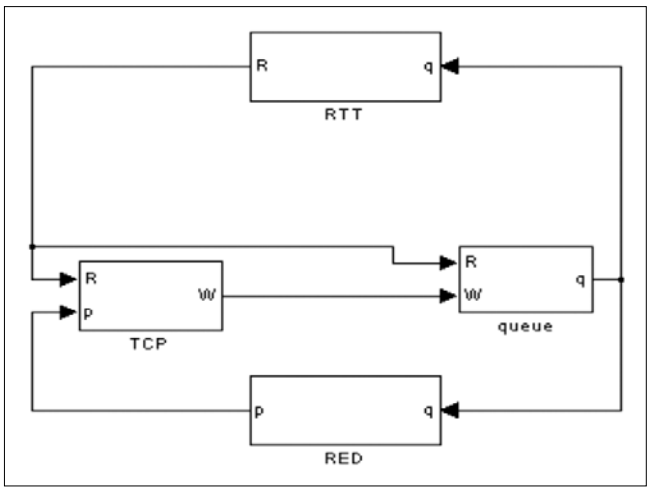
3. Vizsgálati környezet

Ebben a fejezetben bemutatjuk a folyamszintű vizsgálatokhoz kialakított Matlab/Simulink környezetet és ismertetjük a csomagszintű validáláshoz használt Ns-2 szimulációs környezetet.

a) MATLAB/Simulink környezet bemutatása

Az *1. ábra* mutatja be az általunk kialakított MATLAB/Simulink környezet felső szintjét.

1. ábra A MATLAB/Simulink környezet felső szintje



A TCP működését vizsgáló folyadékmodell folyamszintű ("flow-level") vizsgálatokat tesz lehetővé. Az egyes jellemzők, változók dinamikus viselkedését a várható értékükkel írjuk le. A Simulink modell csak a torlódás megelőzési fázist modellezi, nem tartalmazza a kezdeti (slow start) fázist, mert az a protokoll működésében a mi vizsgálataink szempontjából elhanyagolható, hiszen nagyon rövid időt tesz ki.

A Scalable TCP (1-2) és a HighSpeed TCP (3-4) működését leíró differenciálegyenetek a következők:

$$\frac{dW}{dt} = \frac{aW(t)}{R(t)} - \frac{bW(t)W(t-R(t))}{R(t-R(t))} p(t-R(t)) \quad (1)$$

$$\frac{dq}{dt} = \frac{W(t)}{R(t)} N(t) - C \quad (2)$$

$$\frac{dW}{dt} = \frac{a(W(t))}{R(t)} - \frac{b(W(t))W(t)W(t-R(t))}{R(t-R(t))} p(t-R(t)) \quad (3)$$

$$\frac{dq}{dt} = \frac{W(t)}{R(t)} N(t) - C \quad (4)$$

ahol:

W(t): az aktuális ablakméret
q(t): aktuális sorhossz
R(t): körülfordulási idő (RTT)
C: link kapacitás
a: Scalable TCP ablaknövelő paramétere
b: Scalable TCP ablakcsökkentő paramétere
a(*W(t)*): HSTCP *W(t)*-től függő ablaknövelő paramétere
b(*W(t)*): HSTCP *W(t)*-től függő ablakcsökkentő paramétere
N(t): TCP folyamok száma
p(t): csomagjelölési valószínűség

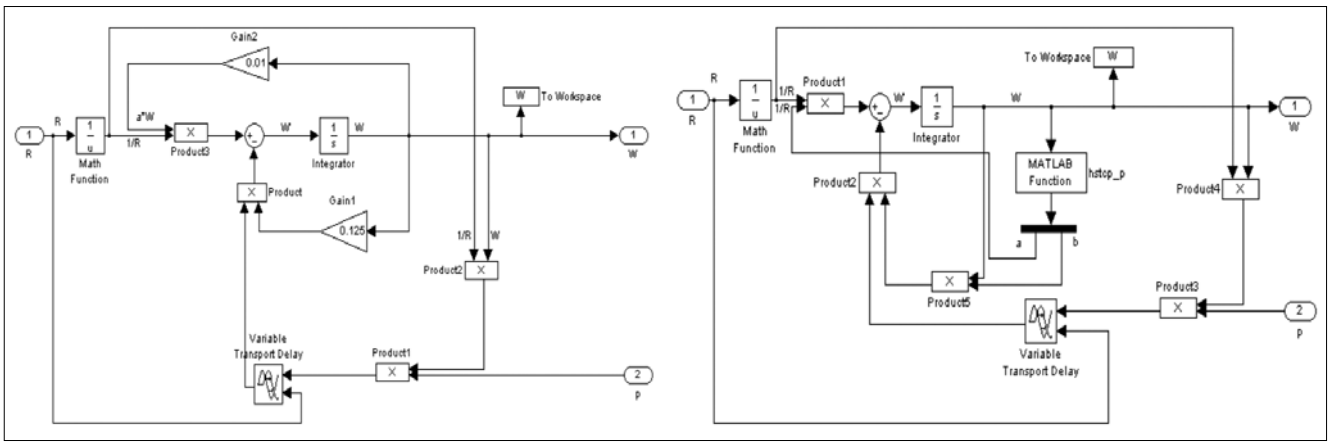
Az egyenletek közül az ablakméret szabályozását leíró egyenlet felépítése a következő. Az első rész valósítja meg az additív, illetve multiplikatív növekedési részt, mely növeli W-t minden RTT alatt a pozitív nyugták érkezési rátájával arányosan. A második (mínuszos) rész a multiplikatív csökkentési rész, mely a negatív nyugták (megjelölt csomagok) érkezési rátájával arányosan csökkenti a W-t.

A negatív nyugták érkezési rátája:

$$\frac{W(t-R(t))}{R(t-R(t))} p(t-R(t)).$$

A várakozási sor dinamikáját leíró egyenlet felépítése a következő. *W(t)/R(t)* mutatja a sorba érkező csomagok rátáját, mely N folyam esetén természetesen N-szeres lesz. A kiszolgálás C kapacitás szerint történik.

Mivel az egyenletek rekurzív összefüggéseket és változó késleltetéseket tartalmaznak, analitikusan nehezen kezelhetők, ezért implementáltuk őket a MATLAB/Simulink környezetben, ahol megoldhatók különböző kezdeti feltételek és késleltetések mellett.



2. ábra Scalable TCP és HSTCP modelljének ablakméret-szabályozó blokkja

A fenti differenciálegyenletek közül az ablakméret-szabályozást leíró MATLAB/Simulink modelleket a *2. ábra* mutatja.

b) Ns-2 környezet bemutatása

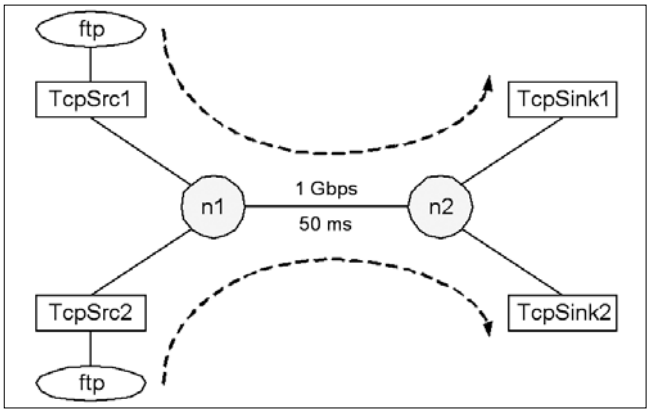
A Matlab/Simulink környezetben végzett szimulációk eredményeit az Ns-2 [7] csomagszintű szimulátorral validáltuk. Ez abban különbözik a folyamszintű vizsgálatoktól, hogy pontosabb eredményeket szolgáltat, de jóval lassabb lefutású. Ebben a környezetben különböző TCP folyamok együttműködése vizsgálható az egy linket és várakozási sort tartalmazó dumb-bell topológiával. A topológiát a *3. ábra* ismerteti.

A hálózat egy szűk sávszélességű linkből áll, amihez forrásokat és nyelőket kötünk. A hálózat forgalmát a vizsgált TCP protokollok által generált adatcsomagok adják. Természetesen a TCP kapcsolat kétirányú, de a nyelő-forrás irányban csak nyugták haladnak, ebben az értelemben tehát a forgalom egyirányúnak tekinthető.

4. Vizsgálati eredmények

A fairness vizsgálatoknál az összehasonlítandó protokollok ablakszabályozó blokkját egymás mellett alkalmazzuk és a forgalmakat összegezzük. Ekkor megfigyelhetjük a két protokoll ablakméretének változását és ezen keresztül az együttes viselkedésüket.

3. ábra Ns-2 dumb-bell környezet egyszerűsített rajza



A következőkben először megvizsgáljuk azt az esetet, amikor két egyforma protokoll verseng, majd pedig azt, amikor két nagysebességű TCP verzió kényszerül osztozni a hálózaton.

4.1. Intraprotokoll működés

A jelen esetben egy-egy Scalable TCP folyam versengését tanulmányoztuk 50 másodperces indulási késleltetés mellett. A *4. ábra* mutatja a MATLAB/Simulink és az Ns-2 által szolgáltatott eredményt.

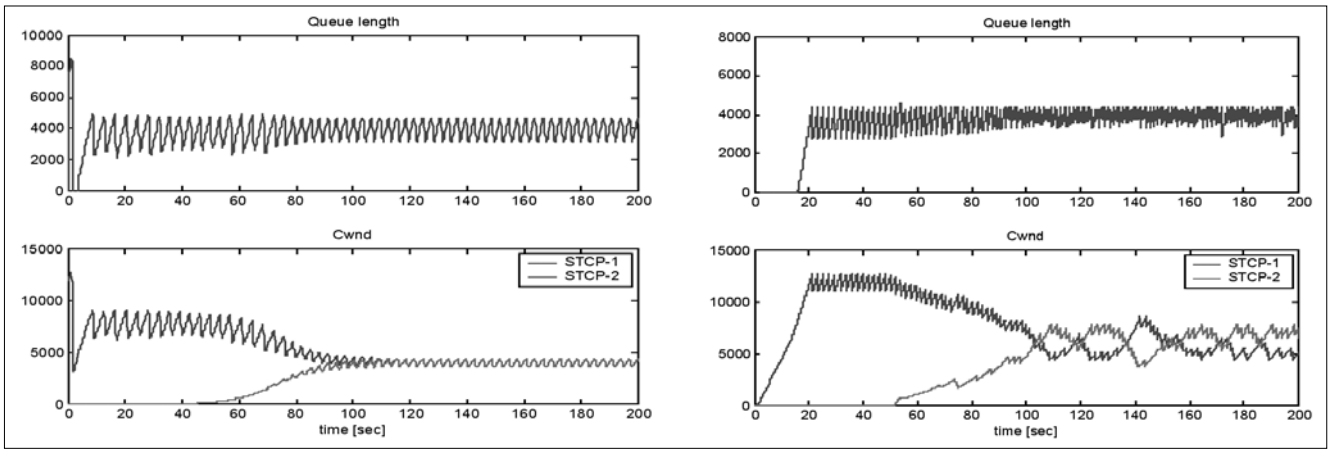
A modell és a szimuláció megfelelően működik, hiszen a különböző vizsgálati módszerekkel jellegre hasonló eredményt kaptunk. A különbségek a két program közötti eltérésekből adódnak. Az Ns-2-es szimulátor például figyelembe veszi a kezdeti (slow start) fázist, amit a Simulinkben nem modelleztünk, ebből ered a görbék elején tapasztalható eltérés. Ezenkívül míg az Ns-2 csomagszintű szimulációt végez, a Matlab/Simulinknél csupán várható értéket látunk. Az intraprotokoll működés mind a Scalable TCP, mind a HighSpeed TCP esetén hasonló eredményeket adott. Azt tapasztaltuk, hogy a protokollok mind egy, mind több folyam esetén késleltetéssel és anélkül is teljesen fair módon működtek együtt.

Az intraprotokoll együttműködések vizsgálatakor minden protokoll esetén azt tapasztaltuk, hogy mind késleltetéssel, mind anélkül, függetlenül a folyamszámtól fair módon osztoztak a hálózaton a protokollok. Ez abban az esetben igaz, amikor RED mellett vizsgáltuk az együttműködést. Drop Tail-lel az eredmények különbözhetnek.

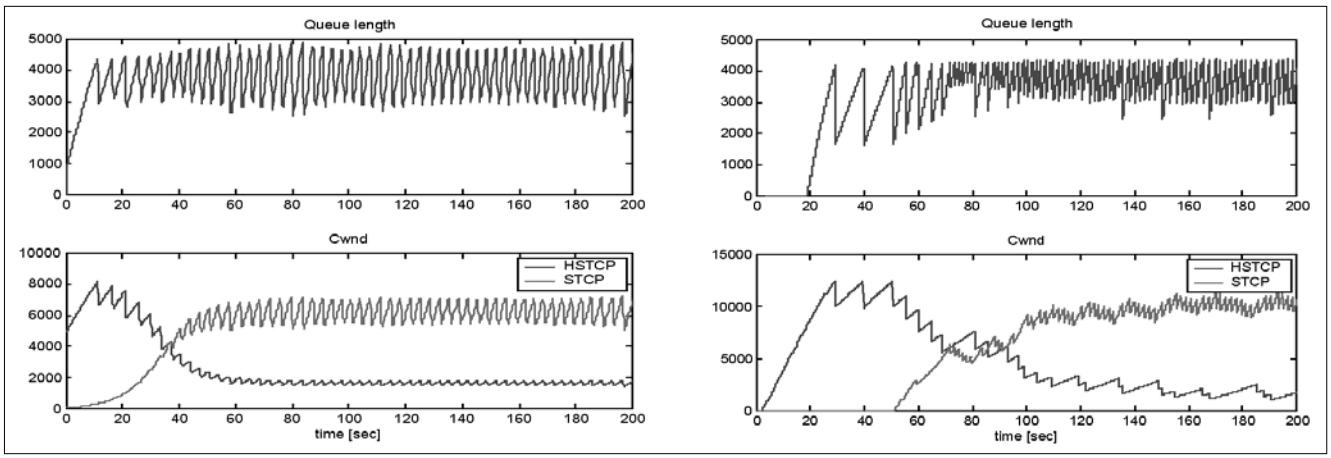
4.2. Interprotokoll működés

Ebben az esetben azt vizsgáltuk, hogy a két nagysebességű protokoll hogyan osztozik a hálózaton egyidejű illetve késleltetett indulás esetén. A *5. ábra* bal oldalán a MATLAB/Simulink által szolgáltatott eredmény, míg a jobboldali képen az Ns-2 szimuláció eredménye.

A Scalable TCP késleltetéstől függetlenül dinamikusabban működött és megszerezte csaknem a teljes sávszélességet még 50 másodperccel későbbi indulás esetén is. Jelen esetben is láthatóak kisebb különbségek a modell és a szimuláció eredményei között, de ezek okai a már korábban említett eltérések a két vizsgálati módszer között.



4. ábra 1-1 Scalable TCP folyam indítása késleltetéssel (modell és szimuláció)



5. ábra 1-1 HS és Scalable TCP folyam indítása 50 s késleltetéssel (modell és szimuláció)

4.3. Következtetések

A nagysebességű TCP protokollok együttműködésének vizsgálatánál is az elvártaknak megfeleltek a kapott eredmények. A Scalable TCP MIMD algoritmusa jóval dinamikusabb működést biztosít, jobban kihasználja a rendelkezésre álló sávszélességet. A dinamikusabb, agresszívebb protokoll minden esetben elnyomta a nála lassabbat. Jelen esetben az alkalmazott MIMD algoritmus dinamikusabb volt a HSTCP módosított AIMD algoritmusánál, így ez tudta nagyobb hatásfokkal kihasználni a rendelkezésre álló sávszélességet. Fontos kihangsúlyozni, hogy az eredményeket RED használata mellett kaptuk.

5. Összefoglalás

Cikkünkben nagysebességű TCP protokollok modelljeit vizsgáltuk olyan hálózaton, melynek torlódásvezérlése a RED algoritmust használja. A különböző nagysebességű verziók együttműködését ezen modelleken alapulva vizsgáltuk, majd eredményeinket csomagszintű szimulátorral validáltuk. Jövőbeli terveink között szerepel az eddigi vizsgálati környezet kibővítése további nagysebességű verziókkal, mint például az eddig még csak részben vizsgált késleltetés alapú protokollokkal, valamint a visszacsatolt rendszer stabilitásának vizsgálata.

Irodalom

[1] S. Floyd,
HighSpeed TCP for Large Congestion Window,
RFC 3649, December 2003.

[2] T. Kelly,
Scalable TCP: Improving Performance
in Highspeed Wide Area Networks,
ACM SIGCOMM Computer Communication Review,
33(2), April 2003.

[3] V. Misra, W. B. Gong, D. Towsley,
Fluid-based Analysis of a Network of AQM Routers
Supporting TCP Flows with an Application to RED,
in Proc. of ACM SIGCOMM, Stockholm, Sweden, 2000.

[4] C. V. Hollot, V. Misra, D. Towsley, W. B. Gong,
A Control Theoretic Analysis of RED,
in Proc. of IEEE INFOCOM, Anchorage, USA, 2001.

[5] B. Sonkoly, T. A. Trinh, S. Molnár,
Understanding HighSpeed TCP:
A Control-Theoretic Perspective, IASTED CCN 2005,
Marina del Rey, CA, USA, October 24-26, 2005.

[6] M.C. Weigle, P. Sharma, J. Freeman,
Performance of Competing High-Speed TCP Flows,
in Proc. of IFIP Networking, May 2006.

[7] Ns-2 Network Simulator,
http://www.isi.edu./nsnam/ns

Hálózat-mobilitás IP alapokon

KANIZSAI ZOLTÁN, RÓZSÁS BALÁZS, IMRE SÁNDOR

BME Híradástechnika Tanszék, Mobil Távközlési és Informatikai Laboratórium
{brozsas, kzoltan}@mcl.hu, imre@hit.bme.hu

Kulcsszavak: Mobil IP, hálózat-mobilitás (NEMO), egymásba ágyazott mobil hálózatok, terhelés-megosztás, szolgáltatásminőség

Napjainkban a vezetéknélküli és mobil technológiák terjedése töretlen. A felhasználók a hagyományos beszédátvitel mellett egyre inkább igénybe vesznek adatforgalmat generáló szolgáltatásokat is és az új, mobil környezetben is használni kívánják valamennyi megszokott kommunikációs csatornájukat. Jelenleg a leginkább elterjedt vegyes használatú telekommunikációs hálózat az IP alapú Internet, melyet emberek milliói használnak napi rendszerességgel, ennek megfelelően célszerű a mobilitás-támogatást is IP alapokon megvalósítani. Erre dolgozta ki az IETF a Mobil IP protokollt [2], amely alkalmas mobil eszközök mozgásának IP rétegbeli kezelésére. A vezetéknélküli hálózati végpontok mozgása mellett bizonyos esetekben egy alhálózat is változtathatja helyét, ennek tipikus példája a járműveken belüli, együtt mozgó hálózatrész (mozgó hálózat). Az alapötlet hasonló a Mobil IP-hez, azonban a mozgó hálózatból fakadóan ez összetettebb probléma, mint az önálló végpontok mobilitása. Cikkünkben a mozgó hálózatok támogatásával kapcsolatos eredményeket tekintjük át a Mobil IP-ből kiindulva.

1. Bevezetés

A jövőben a mobil hálózati eszközök elterjedésével és növekvő mértékű használatával egyre inkább fontossá válik, hogy az eszközök hálózaton belüli mozgása esetén ne legyen szükség kézi beavatkozásra a kapcsolat fenntartásához, illetve újraépítéséhez. A vezetékes hálózatok nagyrészt az Internet Protokollt (IP) használják a forgalom továbbításához és irányításához. Az előző, 4-es verzió [1] mellett folyamatosan terjed az IPv6-os változat [3], mi is elsősorban ezzel foglalkozunk cikkünkben. Másrészről a hagyományos mobilkommunikációban korábban a forgalom nagy részét kitevő beszédátvitel mellett egyre fontosabbá válnak a különböző adatszolgáltatások. Ezért kézenfekvő megoldás az IP rétegben elhelyezni a mobilitást támogató funkciókat, és integrálni a két hálózat (mobil-, illetve vezetékes IP hálózat) előnyeit. Erről a témáról egy részletesebb összefoglaló olvasható [4]-ben.

A hálózat-mobilitással egy külön munkacsoport foglalkozik az IETF-ben, – melyre az angol „NEtwork MObility” szavak összevonásából NEMO-ként is hivatkoznak – ahol a cél egy egész (al)hálózat mozgásának együttes menedzselése, folyamatos kapcsolatának biztosítása. Egy ilyen hálózatot mozgó hálózatnak nevezünk, és mint minden hálózatban, ebben is találhatók routerek (legalább egy), melyek a külvilág felé való kapcsolattartást biztosítják, illetve az IP forgalom irányítását, útvonalválasztását végzik. Ezeket mobil routereknek nevezzük (Mobile Router, MR). Egy mozgó hálózaton belül különféle (mobil és nem mobil) végpontok, illetve további mozgó hálózatok lehetnek. Ez utóbbi esetben egymásba ágyazott (nested) mozgó hálózatokról beszélünk.

A mozgó hálózatokra hozható példaként lehet említeni az egy ember által hordozott különféle eszközök összekapcsolásával létrehozott hálózatot (Personal Area Network, PAN), szenzorok hálózatát, vagy tömegközlekedés esetén az utasok által használt hálózati eszközök számára biztosított hálózatot. Ezek közös tulajdonsága, hogy a hálózatba kötött eszközök csakis együttesen változtatják meg helyüket a hálózati topológiában, hiszen – legalábbis egy ideig – a MR által meghatározott hálózathoz csatlakoznak. Ilyen esetekben a végpontok számára biztosított Mobil IP helyett célszerű valamilyen együttes megoldást kialakítani a mobilitás kezelésére, hiszen egy együttes váltás esetén az egyszerre jelentkező jelzéstöbbletnél hatékonyabb alternatívát kínál.

Bizonyos szempontból a cél ugyanaz, mint a Mobil IP-nél, azaz a kapcsolat megszakadása, illetve a felsőbb rétegek értesítése nélkül kell megoldani az IP rétegben azt a problémát, melyet a hálózat Internethez való csatlakozási pontjának megváltozása okoz. További érv a hálózat-mobilitás mellett, hogy olyan eszközök csatlakoztatását is lehetővé teszi, amelyek nincsenek semmiféle mobilitás-támogatásra felkészítve, mivel a protokoll tervezése során figyeltek a mozgó hálózathoz csatlakozó végpontok átlátszó működésének biztosítására.

A következőkben áttekintjük a hálózat-mobilitásról jelenleg Standards Track fázisban lévő IETF RFC [5] alapján a hálózat mozgásánál felmerülő problémák kezelésére javasolt megoldást, majd az egymásba ágyazott mozgó hálózatokról ejtünk pár szót. Végül az ezzel kapcsolatos nyitott problémákat mutatjuk be. A multihoming, illetve a szolgáltatások minőségbiztosítása után röviden kitérünk biztonsági kérdésekre is.

2. Hálózatok mobilitása

A mozgó hálózatok támogatására az IETF által kidolgozott javaslat [5] tulajdonképpen a Mobil IP protokoll viszszafele kompatibilis kiterjesztése. A cél tehát az, hogy

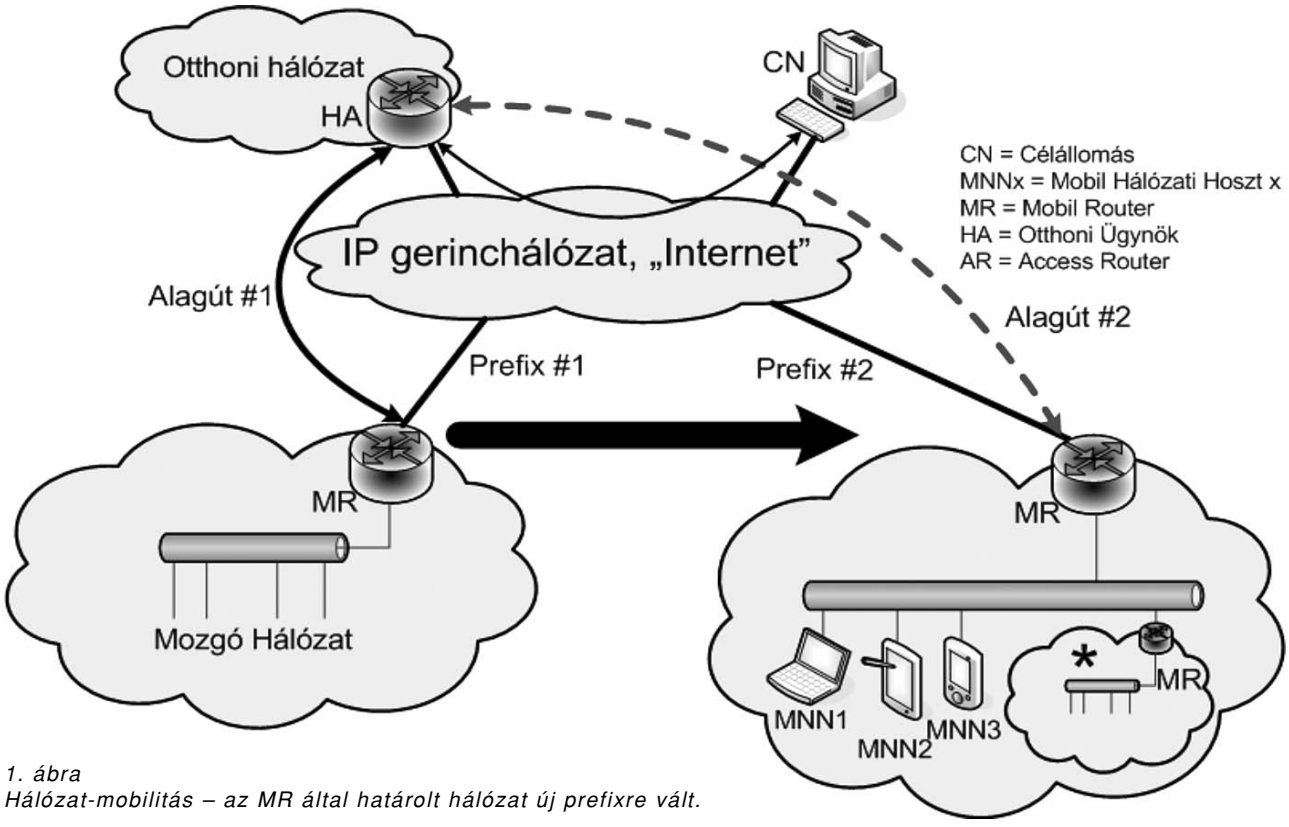
az IP alapon kommunikáló eszközök (mozgó hálózatbeli eszközök, Mobile Network Node, MNN) végig kapcsolatban maradhassanak a hálózattal (a felsőbb rétegek elől rejtett módon), függetlenül attól, hogy éppen hol vannak, illetve mozognak-e éppen.

Az IP-ben megvalósítandó mobilitásnál a problémát az IP hierarchikus címzési rendszere jelenti, a cím egyben interfészazonosító, illetve helyazonosító funkciót is (a cím-prefixeknek megfelelően) ellát. Ezt választja szét a mobil IP egy helyet azonosító ideiglenes címre (Care-of Address), illetve egy interfészazonosító otthoni címre (Home Address). Az otthoni címhez tartozó hálózat az otthoni hálózat (Home Network, HN), amelyben egy otthoni ügynöknek (Home Agent, HA) nevezett eszköz felelős a csomagoknak a mobil aktuális ideiglenes címére való eljuttatásáért, amennyiben a mobil távol (idegen hálózatban) van. A két cím összerendelését kötésnek (binding) nevezik. A mobil feladata, hogy a HA-et tájékoztassa az aktuális céméről, illetve periodikusan frissítési üzenetet küldjön róla. Hálózat-mobilitásnál annyi a különbség, hogy nem egy hálózati végpont az, ami változtatja a hálózatbeli helyét, hanem egy hálózat, azaz tulajdonképpen a hálózat cím-prefixe az, aminek a változását kezelni kell.

Mobil IPv6 esetén nem csak a HA felé küldhetők címfrissítések, hanem bármely kommunikáló fél számára, így azok az aktuális ideiglenes cím ismeretében már közvetlenül is küldhetnek csomagot a mobil felé. E megoldás neve az útvonal-optimalizálás (Route Optimization). Az úgynevezett háromszög probléma (melyre az útvonal-optimalizálás megoldást nyújt a Mobil IPv6 esetében) hálózat-mobilitás esetén ennél komplexebb, hiszen ott egy mozgó hálózaton belül lehet egy éppen ott tartózkodó mobil állomás is. A Mobil IP-ben a mobilnak arra is lehetősége van, hogy ne válaszoljon közvetlenül, hanem a HA-en keresztül. Ez a bidirectional tunneling (kétirányú alagút), amely a hálózat-mobilitás esetén az alapvető működési mód, de javaslatok ott is vannak az útvonal-optimalizálásra.

A NEMO-ban tehát a HA a MR otthoni címprefixére érkezett csomagokat küldi a MR aktuális ideiglenes címére egy kétirányú alagúton keresztül, amelyen a mozgó hálózat teljes forgalma keresztülhalad. Az alagút tulajdonképpen egy újabb, külső IP fejlécet jelent csomagonként.

A NEMO tervezési szempontjai a következők [6]:
- folyamatos kapcsolat biztosítása az Internet felé;
- átlátszóság teljesítmény és mobilitás tekintetében (minimális késleltetés, csomagvesztés);
- a mozgó hálózatbeli végpontok számára átlátszó működés;
- átlátszó működés a felsőbb (IP fölötti) rétegek számára;
- több mozgó hálózat egymásba ágyazásának lehetősége;
- lokális és globális mobilitás támogatása;
- skálázhatóság, nagyszámú mozgó hálózat esetén is működjön a megoldás;
- visszafele való kompatibilitás;
- biztonságos jelzés-üzenetek;
- helyinformáció elrejtésének lehetősége (location privacy);
- IPv4 és NAT (Network Address Translation – hálózati címfordítás) támogatása.



1. ábra
Hálózat-mobilitás – az MR által határolt hálózat új prefixre vált.

A megvalósítást a Mobil IP kiegészítésével oldották meg. A MR-ek a mozgó hálózat belépési pontjai, melyeken keresztül kapcsolat létesíthető a külvilág felé *(1. ábra)*. Legalább egy ilyennek lennie kell minden mozgó hálózatban, hogy lehetőség legyen az Internet felé, illetve onnan befelé forgalmat átvinni. A MR – ellentétben a nem mobil routerekkel – nem tájékoztatja a rajta keresztül az Internethez csatlakozó eszközöket az aktuális kapcsolódási pontján érvényes címtartományról. Ehelyett a kiszolgált eszközök mindvégig úgy látják, hogy a MR otthoni hálózatában vannak, eltekintve az esetleges nagyobb késleltetéstől, illetve megváltozott forgalmi jellemzőktől.

A protokoll tervezésénél célkitűzés volt az effajta teljesítménycsökkenés minimalizálása. Az otthoni hálózatban pedig a HA az, amely az MR hálózatát ismeri, és lehetővé teszi harmadik felek számára a kommunikációt a mozgó hálózatbeli csomópontokkal. Egy mozgó hálózathoz további mozgó hálózatok, valamint mobil csomópontok is csatlakozhatnak dinamikusan.

A MR-nek van egy egyedi otthoni címe, amelyen távolléte alatt elérhető a HA-en keresztül. Ez a cím egy, a HA által hirdetett és aggregált prefixből, illetve egy egyedi azonosítóból tevődik össze (tehát az erre jövő csomagokat a HA kapja meg). A MR egy új hálózatba érkezve szerez egy ideiglenes (care-of) címet (ugyanúgy, mintha egy mobil végpont lenne a Mobil IP-ben), majd értesíti erről a HA-t egy Mobil IP kötés-frissítésben (Binding Update, BU).

Az eddigiekben semmi különbség nincs a mobil IP és a hálózat mobilitása között. Amennyiben a router MR-ként szeretne funkcionálni, azt egy jelzőbit (flag) beállításával jelezheti a HA-nek küldött üzenetében. Ebben az esetben a HA a MR-hez rendelt prefixe(ke)t képzi le a care-of címre, ellentétben a Mobil IP-vel, ahol teljes címek leképzése történik. A HA egy nyugtázó üzenetben közli a MR-rel a művelet befejeztét, vagy esetlegesen a hiba okát.

Siker esetén egy kétirányú alagút épül ki a MR care-of címe és a HA címe között. (Ez az alagút nem más, mint az IP csomagnak egy másik IP csomagba való ágyazása a belépési pontokon, illetve a belső csomag kivétele és továbbítása a kilépési pontokon [7]-nek megfelelően.) Látható, hogy egy ilyen alagút esetén nincs lehetőség egy mozgó hálózatban lévő mozgó eszköznek az útvonal optimalizációjára, hiszen nem is tud arról, hogy jelenleg a MR-nek mi a care-of címe. Az MNN-ek (Mobile Network Node) számára annyi látszik, hogy mindvégig az adott (mobil) hálózathoz csatlakozik, mivel a NEMO elrejti előle a hálózat helyének változásait. Az 1. ábrán látható prefixváltás esetén a MR és a HA között új – vastag szaggatott vonallal jelölt – alagút épül ki. A forgalom minden esetben az alagúton halad át az alap NEMO megoldásban, így egy kommunikáló fél mindig a HA felé küld, illetve onnan kap IP csomagokat. A mozgó hálózatban a szabadon végződő vonalvégek csatlakozó végpontokat jelképeznek. Ezek lehetnek mobil vagy vezetékes eszközök, illetve további mozgó hálózatok is, mint például a *-gal jelölt hálózat, amiről később még szót ejtünk.

A fentebb ismertetett algoritmusnak megfelelően a NEMO csak minimális változtatásokat eszközöl a mobil IP protokollban. A kötésfrissítés üzenetben csupán egy új flag-et vezet be annak jelzésére, hogy a MR routerként regisztrálja-e magát. Ugyanez a flag a nyugtában is megjelenik. Ezen kívül négy új hibakódot vezet be a nyugtában a flag-el, illetve a prefixekkel kapcsolatos hibák jelzésére. A prefixek közlésére bevezet egy új opciót is (mozgó hálózati prefix), amelyet a kötésfrissítési üzenet tartalmazhat.

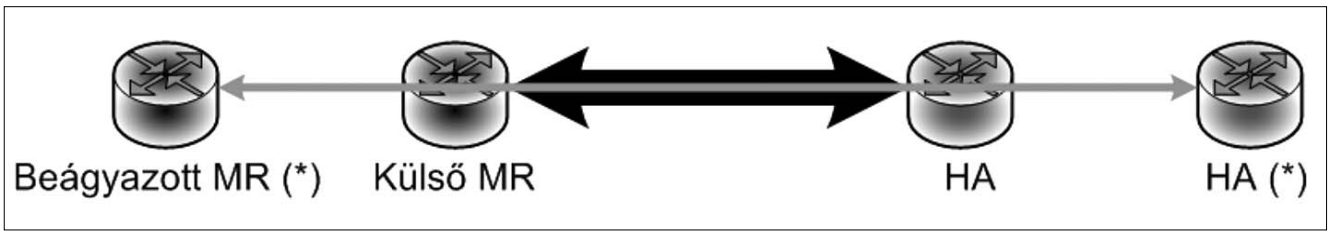
Visszatérve az otthoni hálózatába – csakúgy, mint az eredeti Mobil IP-ben – a MR-nek jeleznie kell ezt a HA felé, ám utána egy normál routerként viselkedhet. Biztonsági okokból a HA-ben is van prefix tábla azért, hogy adott prefixek csak adott MR által legyenek használhatók. A protokoll biztonsági okokból az összes jelzési üzenetre előírja az IPSec [8] használatát.

3. Egymásba ágyazott (nested) mozgó hálózatok

Mint már szó esett róla, egy mozgó hálózathoz egy másik mozgó hálózat is csatlakozhat. Szemléltetésként az 1. ábrán a prefixváltás után egy beágyazott mozgó hálózatot is feltüntettünk *-gal jelölve. Ez azt eredményezi, hogy a csatlakozott (belső) hálózat alagútja a külső alagúton keresztül jut el a saját HA-jéig. Azaz a *-gal jelölt router hálózatában feladott csomag először a külső MR-en keresztül annak HA-jéig jut el (külső alagút vége), majd onnan a *-os MR HA-jéhez kerül, ahol véget ér a beágyazott alagút is *(2. ábra)*.

Ez a dupla alagút értelemszerűen nagyobb hálózati terhelést, illetve kevésbé hatékony működést eredményez. A routing nem optimális úton történik, illetve több lesz a hálózati működéshez szükséges kommunikáció a többszörös alagút miatt. A terhelés fokozottabban jelentkezik a HA-nál, mivel ott több MR forgalma adódik

2. ábra Egymásba ágyazott alagutak



össze. Különösen romlik a hatékonyság, ha a csomag által érintett hálózati helyek távol esnek egymástól. Az egymásba ágyazott alagutak a csomagban a hasznos adat és a hálózati overhead arányát is rontják. Ez hasonló a Mobil IP-beli háromszög problémához, azonban itt nem csak három pontról, hanem többről is szó lehet az egymásba ágyazottság mértékétől függően. Ennek javítására több javaslat is született.

Például az útvonal-optimalizálás ötletét a NEMO-ra alkalmazva azt kapjuk, hogy a kötésfrissítési üzenetekben (BU) a közbeeső összes MR care-of címét el kell küldeni a kommunikáló felek (Correspondent Node, CN) számára [9]. Ez alapján a csomag már optimális útvonalon, az alagutak elkerülésével juthat el egy mozgó hálózatban található címzetthez. Ehhez természetesen a mozgó hálózatbeli csomópontnak címzett csomagnak is tartalmaznia kell ezt a listát, ami egy IPv6 Routing fejlécben adható meg.

Másik lehetőség egymásba ágyazott mozgó hálózatok esetén a probléma kiküszöbölésére, ha a mozgó hálózatban lévő csomópontokat (további MR-eket, illetve mobil végpontokat) felkészítjük a hálózat-mobilitás támogatására, így azok az aktuális ideiglenes prefixet használhatják saját kötéseik frissítésére [10]. Ehhez szükséges, hogy a MR saját hálózatán hirdesse az aktuális ideiglenes prefixet.

Az IP multicast címzésének kihasználásával a kötésfrissítési üzenetek mennyisége csökkenthető, amennyiben a HA-ek csatlakoznak egy multicast csoporthoz, és a mobil csomópont erre a multicast címre küldi el a frissítési üzenetet [11].

Egy kissé bonyolultabb megoldási javaslat található [12]-ben, aminek lényege, hogy a (beágyazott) MR-t tájékoztatni kell a hozzáférési pontja által használt HA címéről. Ezt az információt elküldi a saját HA-jének, ami aztán ezen a címen keresztül küldi el a nyugtát. Ennek hatására a hozzáférési router küld egy frissítést a másik HA-jének, csakúgy, mint egy Mobil IP-beli harmadik kommunikáló félnek tenné a háromszög probléma kiküszöbölésére, ami azután már optimális útvonalon küldheti a forgalmat.

További lehetőség az is, hogy az egymásba ágyazott MR-ek egymás között a HA-k kihagyásával is továbbíthatják a csomagokat, így nem lesz többszörös alagút [13].

4. Mozgó hálózatok hatékonyságának növelése

Mint láthattuk, az alap NEMO protokoll biztosítja a hálózat-mobilitás funkcionális működését, azonban bizonyos esetekben a teljesítőképessége javításra szorulhat. Erre nem egymásba ágyazott hálózatok esetén is vannak javaslatok. Több teszthálózat, illetve hozzájuk tartozó implementáció is megtalálható a témával foglalkozó publikációk között. A teljesítmény összehasonlításánál a mért jellemzők tipikusan a körülfordulási idő, és a csomagvesztés.

Egy NEMO tesztkörnyezetet alakítottak ki [14]-ben is, ahol részben a protokoll-implementációk funkcionalitását hasonlították össze a specifikációkkal, részben teljesítményelemzési méréseket végeztek. A mobilitást WLAN hozzáférési pontok biztosították, a NEMO szoftver implementációja a Nautilus6 [15] munkacsoport által fejlesztett NEPL megoldás volt.

Egy lehetséges javaslat a protokoll teljesítményének javítására a Make-before-break handover nevet viselő megoldás [16], amelynek feltétele az, hogy a MR egyszerre több helyre kapcsolódhasson. A szerzők két párhuzamos kapcsolatot javasolnak. Az egyiken a tényleges adatátvitel folyik, míg a másikon a MR figyelemmel kíséri az elérhető vezetéknélküli hálózatokat és ezek közül a legjobb paraméterekkel rendelkező kapcsolatra vált át valamilyen algoritmus alapján.

További lehetőség a mozgó hálózatok működésének javítására annak kihasználása, hogy a mozgó hálózatok útvonala sok esetben előre jelezhető. Míg egyedi mobil felhasználók esetében az ő mozgásuk egyedi lehet, tömegközlekedési eszközök esetében – ami a NEMO egyik tipikus felhasználási területe – mindenképpen feltételezhető, hogy azok (legalábbis az esetek nagy többségében) ugyanazon az útvonalon haladnak végig [17], mely lehetőséget biztosít a handoverek előrejelzésére és ennek megfelelően esetlegesen erőforrások előzetes foglalására is. A [17] emellett elsősorban egy felsőbb (TCP) rétegbeli modellt mutat be.

Egy további, NEMO implementációk összehasonlítását taglaló cikk található [18]-ban, ahol a következő teljesítményjellemzőket mérték: UDP és TCP forgalom maximális átviteli sebessége, körülfordulási idő (Round Trip Time), illetve cellaváltási késleltetés (handover latency). Az egyes megoldások hatékonyságának jellemzői mellett az együttműködésüket is tesztelték.

5. Multihoming használata mozgó hálózatokban

Amennyiben egy hoszt többféle prefix hirdetést kap (az-az több IP cím közül választhat), akkor multihoming-ról beszélünk. Ez a módszer jelentősen növelheti az egyes mozgó hálózatok teljesítményét. A multihoming akkor valósulhat meg, ha a hoszt vagy a MR egynél több interfésszel is csatlakozik az Internethez, illetve ha a mozgó hálózatban több MR működik egymással párhuzamosan – ami összhangban van azzal a korábban említett feltétellel, hogy egy mozgó hálózatban legalább egy mobil router található.

A multihoming használatának előnyei például:

• Router meghibásodás esetén egy másik MR átveheti a forgalmat a hibás útválasztótól (redundancia). Átlátszó esetben a hosztok külvilági kapcsolatai nem szakadnak meg, nem átlátszó módszer esetében az aktuális kapcsolatok megszakadnak, így azokat újra fel kell építeni.

• Terhelésmegosztás (load sharing) megvalósítása statikusan vagy dinamikusan több külvilág felé irányuló

alagút segítségével. A csomagok egymással párhuzamosan utazhatnak több alagúton keresztül is.

• Policy-routing alkalmazásánál a hosztok valamilyen típusú költségfüggvény alapján választják ki a legalkalmasabb útválasztót a csomagoknak.

Az előnyök mellett azonban jelentkezhet olyan hátrány is, hogy egy nem jól megválasztott útválasztási algoritmus esetén a rendszerben kritikus mennyiségűre nőhet az overhead-üzenetek száma. A mozgó hálózatokban lévő Mobil Routerek a bennük lévő interfészek és az ezekhez csatlakoztatható hálózatok száma alapján lehetnek:

– multi-prefixed tulajdonságúak: amennyiben a MR egress (az Internettel kapcsolatban álló) interfészén lévő linken több prefixet is hirdetnek;
– multi-linked tulajdonságúak: ha a MR-nek több egress interfésze van és ezek több linkhez kapcsolódnak;
– multi-interfaced tulajdonságúak: ha a MR-nek több egress interfésze van, de ezek egy linkhez kapcsolódnak.

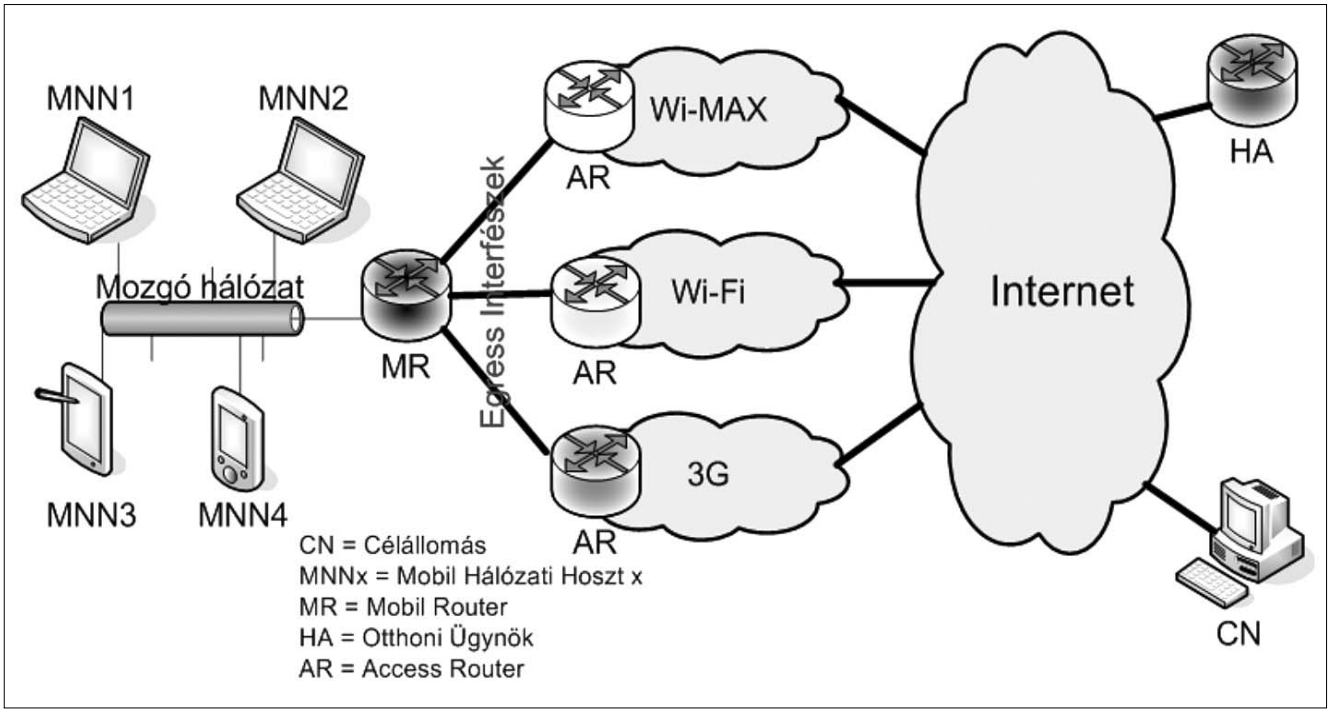
A mozgó hálózatokat bizonyos faktorok alapján a multihoming több, egymástól lényegesen eltérő alosztályába sorolhatjuk be. Ezek a faktorok rendre: a MR-ek száma, a mozgó hálózatokhoz hozzárendelt HA-k száma és mozgó hálózat hosztjainak hirdetett prefixek száma. Jelölése: (n,n,n). Ezen faktorok változtatásával nyolc jelentősebb alosztályt szoktak megkülönböztetni. A számokat vagy egynek veszik, vagy pedig n-nek, így jön ki a nyolc különböző kombináció [19].

Multihoming kialakítására számos módszer létezik, ilyen például [20] is, ahol olyan mozgó hálózatok kerülnek bemutatásra, amelyben mind a MR-ek, mind a HA-k megtöbbszörözve szerepelnek, ezzel hozva létre terhelésmegosztást a hálózaton és biztonságos, megbízható minőségű kapcsolatot a mobil eszközök számára. A cikkben részletesen bemutatnak egy azonosítási és regisztrálási mechanizmust, amelyet szomszédos MR-ek használhatnak egymás biztonságos azonosítására, valamint ezen regisztrált MR-ek felhasználásával javaslatot tesznek egy HA oldali terhelés megosztási algoritmusra. A módszer lényege, hogy a HA-k a periodikus BU üzenetek késleltetéseinek alapján alakítják ki a mozgó hálózat felé irányuló forgalom elosztását a különböző alagutakon. Ez a módszer nem igényli újabb jelzési üzenetek bevezetését a NEMO Basic support-ban tárgyaltakhoz képest, csupán a BU üzenet egyes opciós mezőit használja.

Egy merőben más megközelítést használnak a [21]-ben, amelyben a MR többféle egress interfészének hatékony kihasználásáról van szó. A MR egress interfészei természetesen akár több, különböző típusú vezetéknélküli hálózathoz is csatlakozhatnak egyidejűleg. Például előfordulhat, hogy az első egress interfész 3G mobilhálózathoz, a második Wi-Fi-hez, a harmadik pedig WiMAX-hoz kapcsolódik *(3. ábra)*. [21] szerzői különböző felhasználói profilokat dolgoztak ki, amelyekben figyelembe veszik az elérhető külső hálózati technológiákat és egy adott hoszt kommunikációs folyamatát (Real-time forgalom stb.) A rendszer a hoszt számára legmegfelelőbb profilt fogja alkalmazni és ezáltal próbálja a hoszt éppen aktuális igényeit a lehető legjobban kiszolgálni.

Több MR kezelésére kínál megoldást [22]. A megoldás neve Multiple Mobile Router Management (MMRM), amellyel a hálózatban található hosztok átlátszó módon kapcsolódhatnak az Internetre több mobil útválasztón keresztül. Az MMRM biztosítja, hogy MR-ek dinamikusan

3. ábra Load sharing és QoS megvalósítása több típusú egress interfésszel



csatlakozhassanak a mozgó hálózathoz, vagy szabadon elhagyhassák azt, miközben az egress interfészeiken rendelkezésre álló sávszélességet a mozgó hálózat hosztjai felhasználhatják kommunikációjuk során.

Kissé visszatekintve, tulajdonképpen az egymásba ágyazott (nested) mozgó hálózatok is, tágabb értelemben több MR-t használnak, azaz multihomingot alkalmaznak. Az ilyen hálózatok útválasztásának optimalizálására ad megoldási javaslatot a [23] cikk. Ebben a munkában kerül tárgyalásra egy olyan módszer, amellyel a hálózatokban lévő útválasztók hierarchikus rendszerbe helyezhetők, így minden hoszt kiválaszthatja a számára legkedvezőbb tulajdonságú útválasztót. A hierarchia létrehozása módosított Router Advertisement (RA) hirdetésekkel történik.

6. Szolgáltatások minőségbiztosítása mozgó hálózatokban

A szolgáltatásminőség biztosítása (Quality of Service, QoS) kiemelt szerepet játszik napjaink mobil kommunikációjában. A mozgó hálózatok gyakori helyváltoztatása miatt fontos, hogy az egyes hosztoknak nyújtott szolgáltatások minősége ne csökkenjen drasztikusan csatlakozási pont váltása (handover) esetén. Amikor ugyanis hálózatváltás történik, az új csatlakozási ponton lehetséges, hogy nem áll rendelkezésre annyi sávszélesség, amennyit a mozgó hálózat az előző csatlakozási ponton igényelt és használt. Ezekre a problémákra megoldást adhat a NEMO Reservation (NEMOR) protokoll [24].

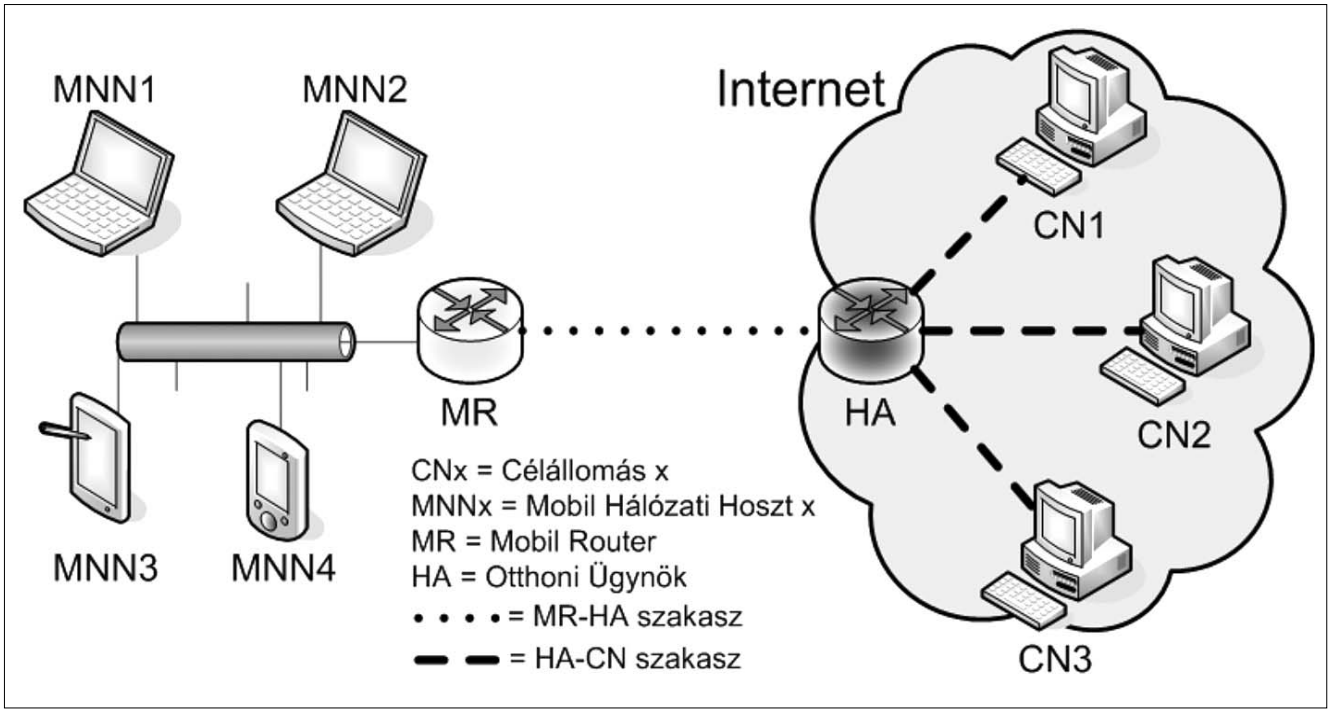
A NEMOR jelzési protokoll a két ismert QoS protokollt, az IntServ-et és a DiffServ-et használja fel. Az IntServ RSVP-t [25] használ, amely szerint az alkalmazásoknak előre fel kell építeniük csomagjaik útját, az útválasztókban erőforrásokat kell foglalniuk az adatforgalomnak. DiffServ esetében az adatforgalmak aggregálásáról beszélhetünk, tehát az egyes csomagokat forgalmi osztályokba sorolja és ezen forgalmi osztályoknak különböző prioritást ad. A csomagok a prioritásuk alapján kerülnek továbbításra.

Összetettebb rendszerekben (például mozgó hálózatokban) célszerű az IntServ és DiffServ együttes használata. A két protokoll kombinálására az IETF Next Step In Signaling (NSIS) csoportja egy általános jelzési protokollt hozott létre.

Az új protokoll két alappillére az NTLP (NSIS Network Layer Protocol) és az NSLP (NSIS Signaling Layer Protocol). Az NTLP (RSVP-t használva) felelős az egyes aggregált folyamok erőforrásainak lefoglalásáért, az NSLP pedig a különböző aggregált folyamok megkülönböztethetőségéért. A NEMOR-ban kétféle NSLP elem van definiálva: egy DiffServ-NSLP a HA felé vezető úton az erőforrások lefoglalásáért (MR-HA szakasz) és egy RSVP-NSLP a HA-tól a célállomásig tartó erőforrás foglalásért (HA-CN szakasz).

Így a NEMOR működése a fenti szakaszok alapján két fázisra bontható. Handover esetén az első szakaszban következik be változás, ekkor a MR-nek kell lefoglalnia az erőforrásokat a HA felé vezető úton. A másik eset, amikor egy MNN új célállomással veszi fel a kapcsolatot, ekkor a HA-nek kell lefoglalnia az erőforrásokat a célállomásig vezető útvonalon. A NEMOR előnye, hogy csak nagyon ritkán kell egy hoszttól egészen a célállomásig egyszerre erőforrást lefoglalni, hátránya azonban, hogy NEMO Basic Support szerinti működést feltételez, vagyis a mozgó hálózat minden forgalma a HA-n keresztül kell, hogy haladjon.

4. ábra A NEMOR protokoll fázisszakaszai



7. Biztonság mozgó hálózatokban

Már egyetlen, hálózatok között mozgó mobil eszköz is rengeteg biztonsági problémát vet fel, így könnyen belátható, hogy ezek a problémák mozgó hálózatok esetén (amelyekben akár több tucat mobil eszköz is működhet) hatványozottan jelentkeznek, azért mindenképpen foglalkozni kell a témával.

Védekezni kell a lehallgatások ellen (sok idegen hálózaton halad keresztül egy mozgó hálózat, könnyű lenne lehallgatni) titkosítással, az üzenet visszajátszásos támadás lehetősége ellen üzenet azonosítással, az üzenet tartalmának megváltoztatása ellen pedig hitelesítéssel. Hatékonyan védekezni egy AAA (Authentication, Authorization, Accounting; Hitelesítés, Engedélyezés, Számlázás) infrastruktúra telepítésével lehetséges, mely három alapvető elemből áll: egy AAA protokollból, egy hitelesítési eljárásból és egy hitelesítési protokollból *(5. ábra)*.

A hitelesítési protokollt a kliens hoszt és a mozgó hálózat határán helyet foglaló ügynök entitás között használják, az AAA protokollt pedig az ügynök és egy távoli hozzáférési hálózatban lévő AAA szerver között (itt tárolódnak a felhasználói profilok). A kliens és az AAA szerver között működik a hitelesítési eljárás. Egy biztonságos mozgó hálózatban a fentiek alkalmazásával a következőknek mindenképpen működnie kell:

– routerek azonosítása,
– mobil hosztok azonosítása,
– a mozgó hálózat szolgáltatásainak hitelesítése.

8. Összefoglalás

Napjainkban egyre nagyobb az igény arra, hogy a mobil eszközök már ne csak a hagyományos módon, „individuumként” legyenek képesek a különféle hozzáférési hálózatok között mozogni, hanem csoportosan, rendszerbe fogva is. Ennek legfontosabb technológiai oka az, hogy több együtt mozgó, egymással összefogott, megfelelően csoportba rendezett mobil eszköz sokkal hatékonyabban szolgálható ki (például kevesebb hálózati többletterhelést generál), mintha egymástól függetlenül, egyenként kellene megoldani kezelésüket. Ráadásul ilyen csoportok kialakítása nem csak elméleti lehetőség, hanem mára gyakorlati igény is, hiszen az egyszerű mindennapi életben is számos olyan helyzet létezik, ahol csoportos mozgás figyelhető meg (mozgó vonat vagy autóbusz utasai stb).

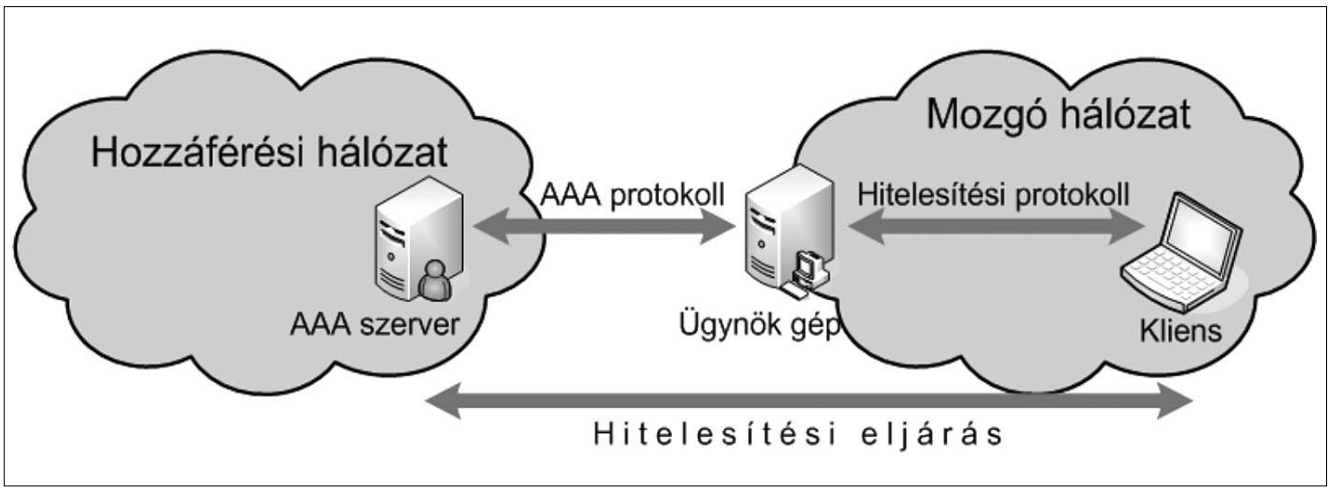
Cikkünkben azt a mozgó hálózatok kialakítását lehetővé tevő, NEMO Basic Support nevű protokollt mutattuk be és vizsgáltuk meg részletesen, több szempont alapján, mely a Mobil IP megoldásából kiindulva javasolt egy megoldást az együtt mozgó hálózati csomópontok mobilitásának kezelésére. A működés bemutatásán túl áttekintést adtunk a jelenlegi, mozgó hálózatokkal kapcsolatos szakirodalomról is olyan témák köré csoportosítva, mint az egymásba ágyazott mozgó hálózatok, a teljesítménybeli kérdések, a multihoming, valamint a szolgáltatásminőség biztosítása.

A kutatások szerteágazó mivoltából is látszik, hogy a területen igen élénk fejlődés tapasztalható. Munkánk során a szakirodalom mellett segítségünkre volt a BME Híradástechnikai tanszékén folyó IKRI projekt keretében felhalmozott tapasztalat, melynek rendelkezésünkre bocsátásáért külön köszönet illeti Bokor Lászlót, illetve a [14] munka szerzőit.

Irodalom

[1] J. Postel,
“Internet Protocol,”
RFC0791, September 1981.
[2] Johnson, D., Perkins, C., J. Arkko,
“Mobility Support in IPv6”,
IETF RFC 3775, June 2004.
[3] Deering, S. and R. Hinden, Editors,
“Internet Protocol, Version 6 (IPv6) Specification”,
IETF RFC 2460, December 1998.
[4] Huszák Árpád, Kiefer Tamás, Simon Vilmos,
Tilk Gergely László, Imre Sándor, Szabó Sándor:
Mobilitás kezelés az IP alapú hálózatokban.
Híradástechnika 2003/4, Vol. LVIII., pp.4–13.

5. ábra Az AAA infrastruktúra



[5] V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert: Network Mobility (NEMO) Basic Support Protocol, IETF RFC 3963, January 2005.
[6] Ernst, T., "Network Mobility Support Goals and Requirements", Work in Progress, October 2004.
[7] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.
[8] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
[9] P. Thubert and M. Molteni, "IPv6 Reverse Routing Header and its application to Mobile Networks", draft-thubert-nemo-reverse-routing-header-05, Work in Progress, June 2004.
[10] E. Perera, A. Seneviratne, V. Sivaraman, "OptiNets: An architecture to enable optimal routing for network mobility," In: Proceedings of the International Workshop on Wireless Ad-Hoc Networks, May 2004. pp.68–72.
[11] I. B. Hamida and L. Boukhatem, A Mobile-IPv6 extension with multicast for nested mobile networks, Vehicular Technology Conference, September 2004. pp.2974–2978.
[12] W. Ng and T. Tanaka, Securing Nested Tunnel Optimization with Access Router Option, draft-ng-nemo-access-router-option-01, July 2004.
[13] M. Watari, T. Ernst, R. Wakikawa, J. Murai, Routing Optimization for Nested Mobile Networks, IEICE Trans. Communications, Vol. E89-B, No.10, October 2006.
[14] Kis Tóth László, Kovács Gergely Kálmán, Kóder István, Bokor László, Semi-virtuális NEMO hálózat megvalósítása protokoll-tesztelési feladatok ellátására, IKRI projekt tervezési dokumentáció, 2006.
[15] Nautilus6 project, http://www.nautilus6.org
[16] H. Petander, E. Perera, K.C. Lan, A. Seneviratne, Measuring and Improving the Performance of Network Mobility Management in IPv6 Networks, IEEE Journal on Selected Areas in Communications, Vol. 24., No.9, September 2006. pp.1671–1681.
[17] A. Baig, L. Libman, M. Hassan, Performance Enhancement of On-Board Communication Networks Using Outage Prediction, IEEE Journal on Selected Areas in Communications, Vol. 24., No.9, September 2006. pp.1692–1701.
[18] R. Kuntz, K. Mitsuya, R. Wakikawa, Performance Evaluation of NEMO Basic Support Implementations.
[19] Thierry Ernst and Julien Charbon, "Multihoming with NEMO Basic Support," http://www.nautilus6.org/doc/paper/20040108-ICMU-NEMO-Multihoming-TErnst.pdf
[20] Seongho Cho, Jongkeun Na, Chongkwon Kim, "A Dynamic Load Sharing Mechanism in Multihomed Mobile Networks," IEEE 2005. pp.1459–1463.
[21] Lucian Suciu, Jean-Marie Bonnin, Karine Guillouard, Thierry Ernst, "Multiple Network Interfaces Management for Mobile Routers," http://www.nautilus6.org/doc/paper/20050627-ITST-NEMO-MultipleIfaces-LSuciu.pdf
[22] Manabu Tsukada, Thierry Ernst, Ryuji Wakikawa, Koshiro Mitsuya, "Dynamic Management of Multiple Mobile Routers," IEEE Malaysia International Conference on Communications and IEEE International Conference on Networks (MICC & ICON 2005), 16-18. November 2005.
[23] Nicolas Montavont, Thomas Noel, Thierry Ernst, "Multihoming in Nested Mobile Networking," In: Proceedings of the International Symposium on Applications and the Internet Workshops (SAINTW'04) 2004.
[24] Mazen Tlais and Houda Labiod, "Resource Reservation for NEMO Networks," Wireless Networks, Comm. and Mobile Computing, International Conference, 13-16. June 2005. pp.232–237.
[25] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O`Dell, A. Romanow, A. Weinrib, L. Zhang, "Resource ReSerVation Protocol (RSVP)," RFC2208, September 1997.

Multicast fák rendszeres újrakonfigurálása többrétegű optikai hálózatokban

PERÉNYI MARCELL, SOPRONI PÉTER, CINKLER TIBOR

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
{perenyim, soproni, cinkler}@tmit.bme.hu

Lektorált

Kulcsszavak: optikai hálózat, dinamikus multicast, újrakonfigurálás, ILP, heurisztika

Cikkünkben dinamikusan változó multicast fákkal foglalkozunk kétrétegű optikai hálózatokban. A levél csomópontok állandó váltakozásával a fa egyre távolabb kerül az optimális topológiától. Ezért sok hálózati erőforrás és költség takarítható meg a fa rendszeres újrakonfigurálásával, melynek során az optimális topológiát állítjuk vissza. Vizsgáljuk az újrakonfigurálás eredményességét több dinamikus útvonalválasztó algoritmus és az újrakonfigurálási intervallum hosszának függvényében.

1. Bevezetés

Az elmúlt években a multicast hálózati alkalmazásoknak is köszönhetően a gerinchálózatok forgalma folyamatosan növekedett. Számos nagyon fontos, szélessávú alkalmazás sorolható a pont-multipont kategóriába, mint például a digitális médiaszétosztás (például IP-tv, IP-rádió stb.), VoD (Video On Demand) médiafolyamok továbbítása, illetve távoktatási vagy virtuális magánhálózati szolgáltatások [1].

A rendkívül kedvező sávszélesség-megtakarítás ellenére jelenleg a legtöbb kereskedelmi Internet-szolgáltató – különböző gyakorlati problémák miatt – a végfelhasználók számára nem teszi lehetővé a multicast szolgáltatások használatát. Ennek következtében rengeteg sávszélességet pazarolunk el a multipont forgalmak alkalmazásszintű multicast (Application Layer Multicast, ALM) módszerrel történő kiszolgálásával, ami gyakorlatilag unicast forgalom-szétosztást jelent. Az egyik nemrégiben megjelent, növekvő népszerűségű, jelentős sávszélesség-igényű alkalmazás, mely a multicast szolgáltatás beindítására sarkallhatja a szolgáltatókat, a peer-to-peer alapú műsorterjesztés *(TV peer-casting)*. Ez az alkalmazás szükségtelenül nagy mértékű hálózati kapacitást emészt fel, hiszen ugyanaz a médiafolyam megy be és ki több ezer felhasználóhoz unicast átvitel segítségével.

Noha a végfelhasználók számára nem érzékelhető, a multicast szolgáltatás a gerinchálózatokban is fontos szerephez jut. Segítségével biztosítható televíziós csatornák hatékony és jól skálázható szétosztása, továbbítása a tartalom szerzőjétől (a terjesztőtől) a helyi szolgáltatók felé. A végfelhasználókkal közvetlen kapcsolatban álló helyi szolgáltató a műsorfolyam szórásán *(broadcast)* túl annak cache-elését is végezheti. A televíziós műsorszolgáltatás a szolgáltatók által nyújtott *triple-play* csomag fontos részét képezi.

Általánosságban elmondható, hogy a multicast szolgáltatás megvalósítása a hálózati hierarchia lehető legalsóbb rétegében a leggazdaságosabb. Ugyanakkor, ha az alsó rétegben alkalmazott technológia kapcsolatorientált (mint például az optikai hálózatok esetében), akkor a kiépíthető kapcsolatok száma korlátozó tényezővé válik. A hullámhosszkapcsolt optikai hálózatok esetében ez a korlátozás több részből tevődik össze: a hullámhosszak, a kapcsolókban lévő (optikai jelet) elágaztató eszközök, valamint ezek kimeneteinek korlátozott száma, illetve a jel teljesítményének változása az elágaztatás következtében (fizikai hatások). Mindezen korlátozásokat tekintve a hullámhossz-fák kiépítése, karbantartása és optimalizálása igen lényeges és nagy kihívást jelentő probléma az újgenerációs, multicast képességgel rendelkező optikai hálózatokban.

Cikkünkben dinamikus multicast fákkal foglalkozunk, melyekben a levél-csomópontok folyamatosan cserélődnek. Új levél-csomópontok (célcsomópontok) léphetnek be a fába az érkező tartalom elérése céljából, míg más tagok kijelentkezhetnek, hogy esetleg visszatérjenek majd egy későbbi időpontban. Ez egy olyan forgatókönyvnek felel meg, ahol az IP-multicast-tagság szabja meg az optikai fa kiépítését. A gyakorlatban egy fát több kisebb multicast kapcsolat, vagy néhány igen nagy sávszélességű multicast igény kiszolgálása érdekében építünk ki.

Egy tipikus alkalmazás lehet egy digitális médiaterjesztő, szétosztó rendszer, ahol a közönség időben változik. Új ügyfelek jelenhetnek meg, akik előfizetnek a tartalomra, más felhasználók – akiknek esetleg lejárt az előfizetésük – elhagyják a hálózatot. A felhasználók ebben az esetben nem feltétlenül egyéni felhasználókat jelentenek, hanem inkább helyi szolgáltatókat (például helyi kábeltévé szolgáltató), akik végfelhasználók egy csoportját testesítik meg.

Egy másik alkalmazási példa lehet egy virtuális magánhálózati szolgáltatás (VLAN), ahol a LAN üzenetszórást el kell juttatni minden végponthoz. Az előző példával ellentétben ez a szolgáltatás kevésbé érzékeny az átvitelben fellépő kisebb megszakadásokra, melyek az újrakonfigurálások alatt jelentkezhetnek. Kivétel ez alól, ha a VLAN forgalom VoIP forgalmat is tartalmaz, mert ez késleltetés-érzékeny.

A multicast fa tagjainak állandó cserélődése a fa „elromlását” eredményezi hálózati- és erőforrásköltségek szempontjából. Ez a folyamatos „leromlás” a fa bizonyos

időközönkénti rendszeres újrakonfigurálásával állítható meg. Az újrakonfigurálás, melynek során a fa optimális topológiája kerül visszaállításra, jelentős hálózati erőforrás- és költségmegtakarítást eredményezhet, mely a szolgáltató számára igen előnyös: a felszabadított erőforrások (ideértve a linkkapacitásokat is) hasznosíthatók más célokra.

Mindazonáltal van néhány hátulütője is az újrakonfigurálásnak:

- Az optimális topológia meghatározása igen számításigényes lehet, mivel a Steiner-fa kiszámítása NP-teljes probléma. Ugyanakkor jelentős időmegtakarítás érhető el gyors heurisztikus módszerek segítségével, melyek kompromisszumos megoldást jelentenek a sebesség és az optimalitás között.
- Az újrakonfigurálás rövid megszakadást okozhat az adatátvitelben, mely adatvesztést, kiesést eredményezhet egyes adatfolyamokban, vagy a csomagok sorrendjének megváltozását. Ez bizonyos alkalmazásokban elfogadhatatlan, ezért ki kell védeni valamilyen technikával.
- Az újrakonfigurálások többlet-jelzésforgalmat generálnak.

Az első probléma orvosolható azzal a feltétellel, hogy egyszerre csak egy fa topológiáját számítjuk ki. Ha több multicast fánk van, akkor pedig ezeket egymás után vezetjük el. Így a számítási idő elfogadható lesz (kb. 10-120 másodperc) több tucat csomópont esetén is. A számítási időt a fa megváltozásának várható idejével érdemes összevetni: a legtöbb alkalmazás (digitális műsorterjesztés, VLAN, de még VoD) esetén is a tartási idő nagyságrendekkel nagyobb, mint az újrakonfigurálási idő.

Ha a fákat együttesen akarjuk optimalizálni, akkor viszont a számítási idő robbanásszerűen megnő, mert megjelenik a kötegelés *(grooming)* lehetősége is.

1.1. Újrakonfigurálás alatt fennálló szolgáltatás-kiesés

Noha cikkünknek nem célja e probléma megoldása, javasolunk néhány módszert, hogy megmutassuk, a probléma kiküszöbölhető.

Egy lehetséges megoldás a megszakadásra érzékeny alkalmazások (pl. média streaming) számára az úgynevezett *soft switch-over (smooth reconfiguration)*. Az eljárás zökkenőmentes átállást biztosít a régi fáról az újra: az új multicast fa már ki van építve, amikor a régi lebontásra kerül. Van egy rövid időszak, amikor mindkét fa egyszerre létezik és képes adatot továbbítani. Annak érdekében, hogy az adatfolyam folyamatossága ne törjön meg a fa megváltozásakor, a csomagok (keretek) átvitele felfüggeszthető egy rövid időre a forrás-csomópontban, hogy biztosítsuk minden csomag „kiürülését" a régi fából. A másik megoldás, hogy az új fában érkező első néhány csomagot eltároljuk a kimeneti csomópont(ok)ban, amíg az átvitel végét jelentő jelzési csomag meg nem érkezik a régi fában.

Mindezekkel együtt a soft switch-over többlet-erőforrásokat igényel a hálózatban. A mi egyszerű modellünkben, amennyiben minden linken rendelkezésre áll egyetlen szabad hullámhossz, akkor egy darab multicast fa újrakonfigurálása lehetséges ILP optimalizálással. DWDM hálózatokban, ahol a hullámhosszak száma 30-nál is több, ez a plusz kapacitás elfogadható (különösen, ha összevetjük az optimalizálásból adódó jelentős kapacitás-nyereséggel). Ugyanakkor természetesen nem biztos, hogy ez a többletkapacitás minden pillanatban rendelkezésre áll.

1.2. Kapcsolódó publikációk

Eddig viszonylag sok cikk jelent meg optikai hálózatokban kialakítandó dinamikus multicast fák optimalizálásának témakörében. Mivel az igények optimális elvezetése technikailag gyakran nem megoldható vagy időigényes, ezért sok heurisztikus megközelítés született. Ezek teljesítményét általában ILP-vel számolt optimális megoldáshoz is hozzámérték.

Statikus multicast igények elvezetését vizsgálták gyűrű és háló topológiájú, hullámhossz-kapcsolt optikai hálózatokban többek között a [2,3]-ban. A [3] szerzői a kötegelés problémájára írtak fel egy analitikus modellt, nemlineáris programozási feladat formájában. Az eredményt heurisztikákkal hasonlították össze. További heurisztikus optimalizáló algoritmusokat mutat be a [4-6]. A [7] szerzői ILP formulák segítségével oldották meg a hullámhossz-hozzárendelés és -elvezetés problémáját. Megmutatták, hogy már viszonylag kevés hullámhosszosztó és konverter használatával is hatékonyan kezelhető a multicast probléma. Mustafa és szerzőtársai [8] szintén ILP, illetve heurisztika segítségével minimalizálták az elektromos rétegbeli eszközök és hullámhosszok számát.

Napjainkban a dinamikusan változó multicast fák problémájával többet foglalkoznak. A dinamikus problémánál a cél általában a blokkolási ráta minimalizálása, nem pedig az összes igény elvezetése (különböző kényszerek betartásával), mint a statikus probléma esetében. Ez általánosságban még a statikus esetnél is erőforrás- és számításigényesebb. Azt tapasztaltuk ugyanakkor, hogy az útvonalválasztás néhány részproblémája (egyetlen fa vagy több fa külön-külön való optimalizálása) megoldható – elfogadható időn belül – ILP [9] segítségével optimális módon. Így lehetővé válik a dinamikus elvezetési algoritmusok eredményeinek összehasonlítása az optimális megoldáséval, valamint a nyereség meghatározása.

Dinamikus fák elvezetésére és karbantartására *(routing and provisioning)* – kötegelésre képes optikai hálózatban – számos algoritmust mutat be [10-12]. A [13]-ban dinamikus igények elvezetését vizsgálták forgalomtervezéssel *(traffic engineering)* unicast esetben, kötegelésre képes WDM hálózatokban. A [14] szerzői egy dinamikus hullámhossz-hozzárendelési algoritmust mutatnak be multicast esetben. A cél a blokkolási valószínűség minimalizálása azáltal, hogy minden lépésben maximalizálják a maradék szabad sávszélességet a linkeken. Chowdhary és szerzőtársai hasonló problémát vizsgálnak [15]-ben, on-line multicast fák karbantartására dolgoztak ki algoritmust. A cél, hogy növeljék a hasz-

nált eszközök kihasználtságát és minimalizálják a később érkező igények blokkolási valószínűségét. A [16] szerzői bevezettek egy hullámhosszfa *(light-tree)* alapú védelmi sémát az egyszeres hibák kivédésére. Kidolgoztak egy ILP felírást a javasolt módszerhez szükséges minimális többlet-sávszélesség meghatározására.

Tudomásunk szerint még nem született olyan publikáció, mely dinamikus multicast fák rendszeres újrakonfigurálásának hatásait vizsgálná és elemezné a dinamikus útvonalválasztó algoritmusok eltérését az optimális topológiától összehasonlítva a dinamikusan változó költséget az optimummal.

2. A probléma megfogalmazása

Kétrétegű hálózatot tételeztünk fel: a felső, elektronikus réteg időosztásra, míg az alsó, optikai réteg hullámhosszkapcsolásra képes. Az elektronikus réteg forgalomkötegelést is végezhet, tehát több, alacsony sávszélességű igény fogható össze *(multiplexálás)* egyetlen hullámhossz csatornába. A két réteg a *peer* (társ, vertikálisan összekapcsolt) *modell* [17] szerint működik együtt. Ennek megfelelően az elvezetés során a vezérlő sík számára mind az elektronikus, mind az optikai réteg állapotinformációi rendelkezésre állnak. Ez lehetővé teszi, hogy mindkét rétegre kiterjedő optimális megoldást találjunk.

A hálózati topológiát, az összeköttetéseket alkotó szálak számát és a forgalmi igények leírását előre ismertnek tekintjük. Emellett a használható hullámhosszak számát, azok kapacitását, vagy a forgalomelvezetés elemeinek költségeit (pl. térkapcsolás, O/E konverzió) is rögzítettnek vesszük.

Multicast igényeket tételezünk fel a hálózatban, amelyek dinamikusan változnak. A korábban leírtaknak megfelelően az igények egyetlen igen nagy sávszélességű IP multicast kapcsolatnak vagy több kisebb kapcsolat összefogásának felelnek meg, amelyek a levél-csomópontok többségében osztoznak. Nem foglalkozunk a cikkben azzal a problémával, hogyan érdemes különböző kapcsolatokat összeszervezni egy multicast igénnyé.

Hasonló megkötés érvényes a multicast fák (hullámhossz fák) együttes optimalizálásával, illetve azok összevonásával kapcsolatban: az egyszerűség kedvéért a cikkben az egyes hullámhossz-fák külön-külön kerülnek optimalizálásra. A szimulációk során mindig egy darab fát optimalizálunk, melynek gyökér csomópontja állandó, a levél csomópontok pedig folyamatosan változnak, de az eredmények több futtatás (tehát több eltérő gyökerű fa) összeátlagolását mutatják. Természetesen az együttes optimalizálás nagyobb költségnyereséget tenne lehetővé, de jelentősen nagyobb számítási igény árán.

A multicast fa olyan úgynevezett „részigények" összessége, amelyek képesek erőforrásokon osztozni a hálózatban, azaz sávszélességük nem adódik össze. Minden cél-csomóponthoz egy-egy részigény rendelhető (minden részigény forrása a multicast fa egyetlen gyökér csomópontja). A fa célcsomópontjainak halmaza dinamikusan változik: új csomópontok léphetnek be a fába, míg mások elhagyhatják azt. Az egyes új részigények útvonalát valós időben kell kiszámítani, míg a távozó csomópontokhoz kapcsolódó utakat a lehető legnagyobb odafigyeléssel kell lebontani, hogy a többi igényt a lehető legkisebb mértékben befolyásoljuk.

Minden célcsomópontra a kapcsolat tartási ideje és kapcsolatok közti idő exponenciális eloszlású. A forgalom intenzitása az eloszlások várható értékének (λ paraméter) megfelelő megválasztásával szabályozható. A cél, hogy a forráscsomópontból minden időpillanatban, minden aktuális célcsomópontba eljusson az információ.

3. Hálózati modell

A kétrétegű optikai hálózatot hullámhossz-gráffal modelleztük, mely lehetővé teszi az igények elvezetését a két réteg együttes kezelésével, különböző csomóponttípusok és hálózati topológiák mellett. A hullámhosszgráf a fizikai hálózatból származtatható a topológia és a különböző eszközök tulajdonságainak figyelembevételével. A modell egy egyszerű változata [18]-ban került publikálásra.

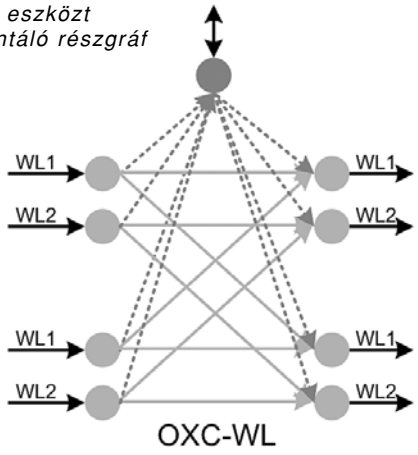
A RWA (Útvonal-választási és hullámhossz-hozzárendelési) probléma ILP alapú, forgalomkötegelést tartalmazó megfogalmazása unicast igényekre a [19]-ben olvasható.

Számos eltérő típusú hálózati kapcsolót különböztethetünk meg: Optical Add-and-Drop Multiplexers *(OADM)*, Optical Cross-Connect (*OXC*: optikai maggal), Opto-Electrical Cross-Connect (*OEXC*: elektronikus mag), melyek teljes vagy részleges, illetve tiszta optikai vagy elektronikus hullámhossz konverziót támogathatnak. Az eszközök egy része kötegelésre is alkalmas lehet. Ezek a képességek a hullámhossz-gráfokban egyszerűen figyelembe vehetők. A hálózatot kapcsolóeszközök és összeköttetések (optikai szálak) alkotják. Az optikai szálak végei fizikai eszközök interfészeihez kapcsolódnak. Ez utóbbi határozza meg az adott fényszálban használható hullámhosszak számát. Minden csomópont interfészekből és egy belső kapcsológépből áll. Minden hullámhossz és kapcsológép reprezentálásra kerül a hullámhossz gráfban.

Egy fényszál pontosan annyi logikai éllel kerül megjelenítésre, ahány hullámhosszon terjedhet benne a jel. Egy eszköz logikai ábrázolása az eszköz fizikai képességeitől függ. A gráf minden éle rendelkezik használati költséggel és kapacitással. Az élek kapacitása általában a hullámhossz-kapacitással egyenlő, ami a használt vivőtől függ (általában 2.5 vagy 10 Gbit/s – a szimulációk során mi az előbbit tételeztük fel). Egy él használatának költségét annak funkciója határozza meg (O/E konverzió, térkapcsolás, hullámhossz él). Az élköltségek a szimuláció paraméterei, azonban igyekeztünk a valóságnak megfelelően beállítani azokat: egy hullámhossz használatát tételeztük fel a legdrágábbnak, feltettük továbbá, hogy az O/E vagy E/O konverzió drágább, mint a térkapcsolás.

Minden hálózati kapcsolót egy részgráf reprezentál, mellyel az eszköz összes interfészét és az eszköz belső kapcsoló képességét is modellezzük. A hullámhosszgráf – kiegészítve a későbbiekben bemutatásra kerülő ILP megfogalmazásokkal – lehetőséget biztosít a különbözőbb képességű fizikai eszközök leképezésére, még akkor is, ha azok egy adott hálózatban egyszerre vannak jelen. A modell könnyen kiterjeszthető, fejleszthető. Az egyes eszközök képességeinek változása könnyen követhető új részgráf típusok bevezetésével.

1. ábra
OXC-WL eszközt reprezentáló részgráf



Egy sokoldalú eszköz részgráfját mutatja az *1. ábra.* Az eszköz egyszerre rendelkezik egy OXC és egy OADM képességével: lehetőség van igények indítására, végződtetésére, illetve hullámhossz-konverzióra és kötegelésre. Hullámhossz-elágaztatásra csak az elektromos rétegen keresztül van mód. Az elektromos réteget egyetlen (a legfelső) csomópont reprezentálja. A többi csomópont interfészt reprezentál. Az ábrán látható eszközök két bejövő és két kimenő interfésszel rendelkeznek, melyek mind két hullámhosszt támogatnak. A szimulációk során ezt a csomópont típust használtuk.

4. Útvonalválasztó algoritmusok

Több útvonalválasztó algoritmust is alkalmaztunk az igények elvezetésére. A cél ezek költségeinek és teljesítményének összehasonlítása volt. Az ILP alapú optimális útvonalválasztást tekintjük referenciának. A Dijsktra-algoritmuson alapuló „legrövidebb utak láncolata” pedig egy igen egyszerű mohó módszer. Ezek mellett két feszítőfa-módszeren alapuló heurisztikát is kipróbáltunk. A módszerek közötti különbséget szemlélteti a *2. ábra*.

4.1. ILP útvonalválasztás és felírása

ILP segítségével meghatározható a hálózatban lévő igények optimális elvezetésének költsége. Ezért minden összehasonlítás alapjául szolgál. Természetesen az optimális költség nem azt jelenti, hogy az így kapott konfigurációban lévő erőforrások (pl. használt hullámhosszak, O/E, E/O átalakítók stb.) száma mind minimális.

Ugyanakkor az ILP megoldása kiszámítás nagyon sok időbe telhet. Szerencsére egyetlen multicast fa optimális elvezetésének meghatározása elfogadható idejű még meglehetősen nagy hálózatokban is. A megoldási idő szimulációink szerint 3 és 180 másodperc között változik a – 28 csomópontból és 41 linkből álló – COST 266 [20] hálózatban egy 2.8 GHz-es Pentium D számítógépen. Ha több fát egyszerre vezetünk el, a költségmegtakarítás még ennél is jelentősebb lehet, de a megoldás meghatározásához szükséges idő elfogadhatatlanul megnő. Ezért az egyetlen lehetőség, hogy a multicast fákat egymástól függetlenül egyesével vezessük el.

Az ILP egyik komoly hátránya, hogy az egymást követő megoldások nagymértékben eltérőek, így az igények útvonalának (beleértve az út során érintett kapcsolókat is) újrakonfigurálása elkerülhetetlen.

A következő ILP megfogalmazás több multicast fa együttes, optimális elvezetését teszi lehetővé a hálózatban:

$z_{ij}^{or} \in \{0, 1\}$: az r multicast fa o részigénye használja-e az (i, j) élt vagy sem.

$x_{ij}^{r} \in \{0, 1\}$: az r (unicast vagy multicast igény) használja-e (i, j) élet.

$y_{ij} \in \{0, 1\}$: az (i, j) élt használja-e bármelyik igény.

$$\sum_{\forall j \in V_i^+} z_{ji}^{or} - \sum_{\forall k \in V_i^-} z_{ik}^{or} = \begin{cases} -1 & \text{if } i = s^r \\ 0 & \text{if } i \notin \{s^r, t^{or}\} \\ +1 & \text{if } i = t^{or} \end{cases} \quad (1)$$

minden $i \in V$ (logikai) csomópontra,
r igényre és o részigényre

V_i^+ jelenti azon csomópontok halmazát, amelyek i-ből elérhetők kimenő élen. V_i^- azon csomópontokat reprezentálja, melyekből i elérhető irányított élen át. *A, V, V_E, O, R* jelentése sorrendben a következő: élek, csomópontok, elektromos csomópontok, részigények, végül igények halmaza. Az r igény forrását s^r, míg nyelőjét t^{or} jelöli, ahol o a részigény azonosítója.

$$z_{ij}^{or} \le x_{ij}^{r}, \ \forall (i, j) \in A, \ \forall o \in O, \ \forall r \in R \quad (2)$$

$$x_{ij}^{r} \le \sum_{\forall o \in O} z_{ij}^{or}, \ \forall (i, j) \in A, \ \forall r \in R \quad (3)$$

$$\sum_{\forall j \in V_i^+} x_{ji}^{r} = \sum_{\forall k \in V_i^-} x_{ik}^{r} \le 1, \ \forall i \notin V_E, \ \forall r \in R \quad (4)$$

$$\sum_{\forall j \in V_i^+} x_{ji}^{r} \le \begin{cases} 0 & \text{if } i = s^r \\ 1 & \text{if } i \ne s^r \end{cases}, \ \forall i \in V_E, \ \forall r \in R \quad (5)$$

$$\sum_{\forall r \in R} x_{ij}^{r} \cdot b^r \le B_{ij}, \ \forall (i, j) \in A \quad (6)$$

$$x_{ij}^{r} \le y_{ij}, \ \forall (i, j) \in A, \ \forall r \in R \quad (7)$$

$$y_{ij} \le \sum_{\forall r \in R} x_{ij}^{r}, \ \forall (i, j) \in A \quad (8)$$

$$\sum_{\forall j \in V_i^+} y_{ji} = \sum_{\forall k \in V_i^-} y_{ik} \le 1, \ \forall i \notin V_E \quad (9)$$

Változók:

$$z_{ij}^{ro} \in \{0, 1\},\ \forall (i, j) \in A,\ \forall o \in O,\ \forall r \in R \quad (10)$$

$$x_{ij}^{r} \in \{0, 1\},\ \forall (i, j) \in A,\ \forall r \in R \quad (11)$$

$$y_{ij} \in \{0, 1\},\ \forall (i, j) \in A \quad (12)$$

Célfüggvény:

$$\text{Minimalizálandó} \sum_{\forall (i, j) \in A} c_{ij} y_{ij} \quad (13)$$

(1) a folyammegmaradás törvényét fejezi ki a részigényekre. (2) szerint egy multicast fa használ egy adott *(i, j)* élet, ha bármelyik részigénye áthalad rajta. (3) az előző fordítottja: egy *(i, j)* élet csak akkor használ egy fa, ha legalább egy részigénye áthalad rajta. Ez biztosítja, hogy fölöslegesen ne foglaljunk le kapacitást. (4) biztosítja, hogy igény ne tűnhessen el, illetve ne ágazhasson el olyan csomópontban, ahol ez nem engedélyezett. (6) szerint az adott *(i, j)* élen áthaladó igények sávszélességeinek összege nem haladhatja meg az él (hullámhossz) kapacitását. (7) biztosítja, hogy egy élen csak akkor haladhasson át egy igény, ha az használatra le van foglalva. (8) ismét a fölösleges lefoglalást akadályozza meg: csak akkor kell lefoglalni egy élet, ha azon legalább egy igény áthalad. (8) elhagyható, mivel ezt a célfüggvény implicit módon tartalmazza. (9) nagyon hasonlít (4)-re, csak eggyel magasabb absztrakciós szinten. (9) elhagyható (mert redundáns kényszer), azonban gyorsíthatja a megoldást.

A (13) célfüggvény kifejezi, hogy a lefoglalt élek összköltségének minimumát keressük. Célunk tehát egy minimális költségű elvezetés megtalálása.

4.2. Legrövidebb utak láncolata (Dijsktra-algoritmus)

A legrövidebb utak láncolatán alapuló algoritmus gyors és egyszerű. Anélkül használható új levélcsomópontok fába való becsatlakoztatására, hogy ez hatással lenne a már bent lévő részigények útvonalára. Ugyanakkor a módszer költségpazarló.

Az algoritmus a következőképpen működik: minden levélcsomóponthoz egy „részigényt” rendelünk, a részigények útvonalai egymás után kerülnek kiszámításra a levélcsomópontok és a forrás között. Az algoritmus közvetlenül a logikai hálózatot tekinti. A részigény forrás- és cél-csomópontja egyaránt a hullámhossz gráf egy-egy elektronikus csomópontja. A fa által éppen használt élek költsége nulla, tehát ezeket egy új részigény ingyen használhatja.

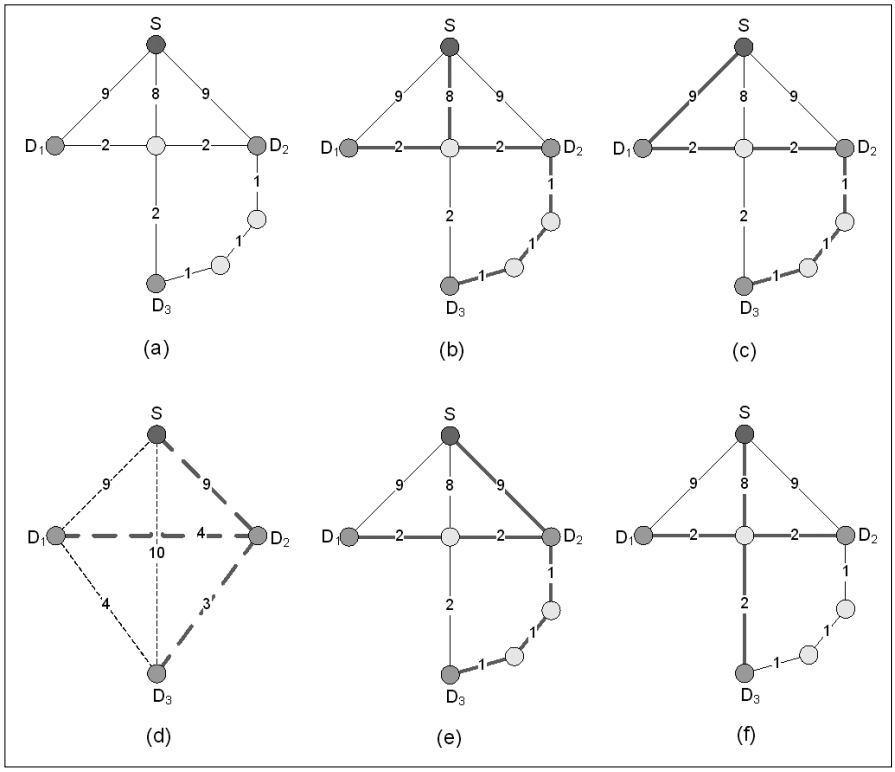
A fát elhagyó igények útvonalai törlődnek. Minden olyan él, melyet már nem használ a multicast fa (tehát egyetlen részigény sem használja már), felszabadításra kerül. A módszer sohasem változtatja meg a már elvezetett részigények útvonalát, ami végeredményben gyakran hosszabb útvonalakhoz, nem optimális megoldásokhoz vezet.

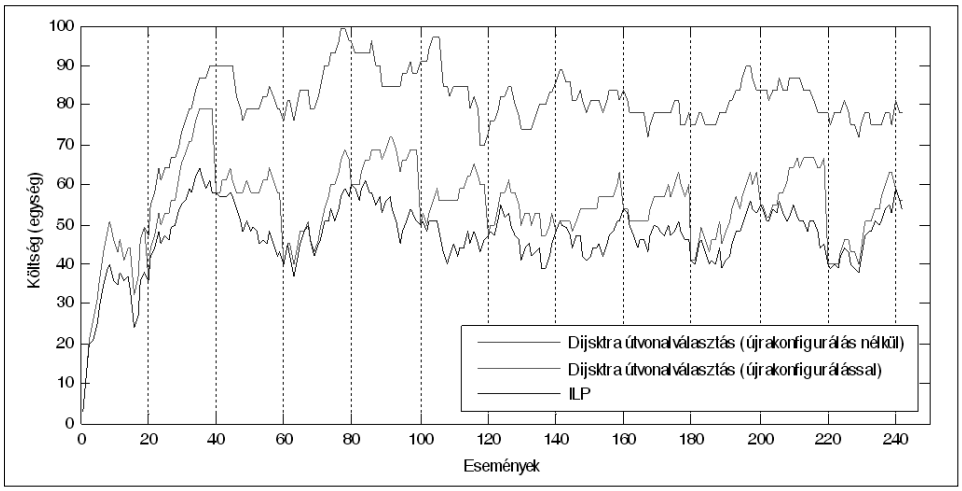
4.3. Legrövidebb út heurisztika (Minimal Path Heuristic, MPH)

Az MPH algoritmus az eredeti gráfot egy virtuális gráffá transzformálja, majd ezen alkalmazza Prim módszerét [21]. Így próbálja meghatározni a legjobb megoldását. A virtuális gráf egy teljes hálózat, ahol a forrást és minden célt egy-egy pont reprezentál. A virtuális hálózat minden éle a legolcsóbb utat reprezentálja a valós gráfban az él kiinduló és végpontja között. Az élek költsége megegyezik a reprezentált út összköltségével. Ennek megfelelően a módszer alkalmazásához meg kell határozni a forrás és a nyelők, illetve a célok egymástól vett távolságát. Ez utóbbiakat két irányban is.

A Prim-algoritmust a virtuális hálózaton kell alkalmazni. Miután a minimális költségű feszítőfa meghatározásra került, annak egyes éleit vissza kell vezetni az eredeti hálózatba, azaz a reprezentált utat le kell foglalni. Amennyiben új csomópont kerül hozzáadásra a hálózathoz, az új fa számítása során a már használt élek költségét nullára kell állítani. Ez garantálja, hogy a már elvezett részigények útvonala sose változzon. A részletek [22]-ben olvashatók.

2. ábra a) az eredeti topológia a forráscsomóponttal és három levélcsomóponttal, b) feszítőfa útvonalválasztás, c) legrövidebb utak láncolata, d) MPH virtuális topológia és útvonalválasztás, e) MPH útvonalválasztás, f) ILP optimális útvonalválasztás





3. ábra
Igények elvezetésének költsége a bekövetkezett események számának függvényében, Dijkstra algoritmusával, újrakonfigurálással és anélkül, valamint az ILP optimális útvonalválasztáshoz hasonlítva

4.4. Feszítőfa útvonalválasztás (Tree routing)

Ez az algoritmus nagymértékben hasonlít az MPH-hoz. Az eltérés annyi, hogy a Prim-algoritmust ezúttal közvetlenül a hullámhossz gráfon alkalmazzuk, nem a virtuális gráfban. A kiszámított feszítőfának a multicast részfa által nem használt élei eltávolításra kerülnek. A fa frissítése és az élköltségek módosítása hasonlóan történik az előzőekhez.

Mind az MPH, mind a feszítőfa alapú útvonalválasztásnál problémát jelent, hogy elvezetéskor az újonnan felvett részigények olyan (nem elektromos rétegbeli) csomópontokban is szétágazhatnak, ahol ez nem engedélyezett. Az ilyen eseteket egy utómunkálati fázissal meg kell szüntetni. A probléma könnyen megoldható az elágazásnak az elektromos rétegbe való áthelyezésével.

5. Eredmények

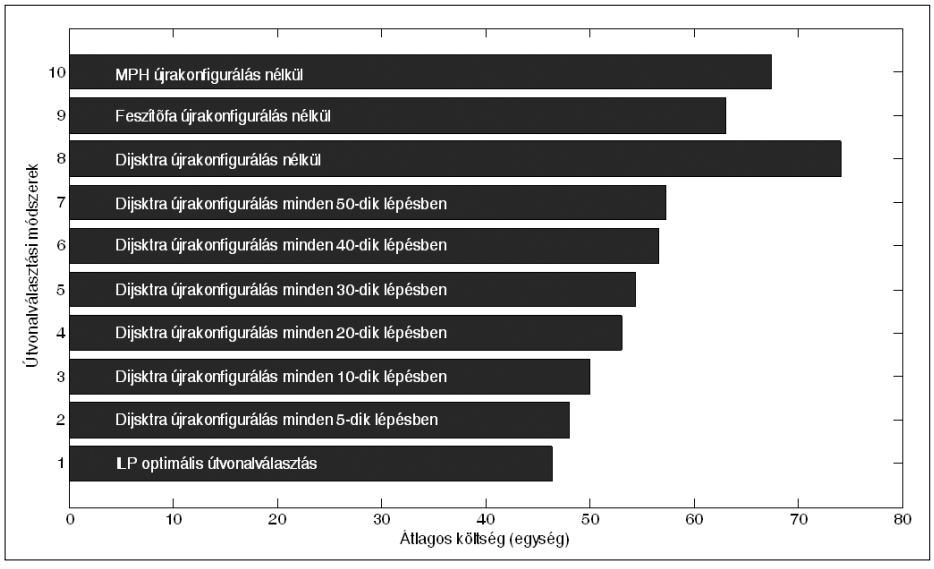
A szimulációkat a COST266 Európai referencia hálózatban [20] végeztük, minden csomópont csak elektronikus osztóképséggel rendelkezett. Minden futtatás során ugyanaz a dinamikus igényhalmaz került behelyettesítésre.

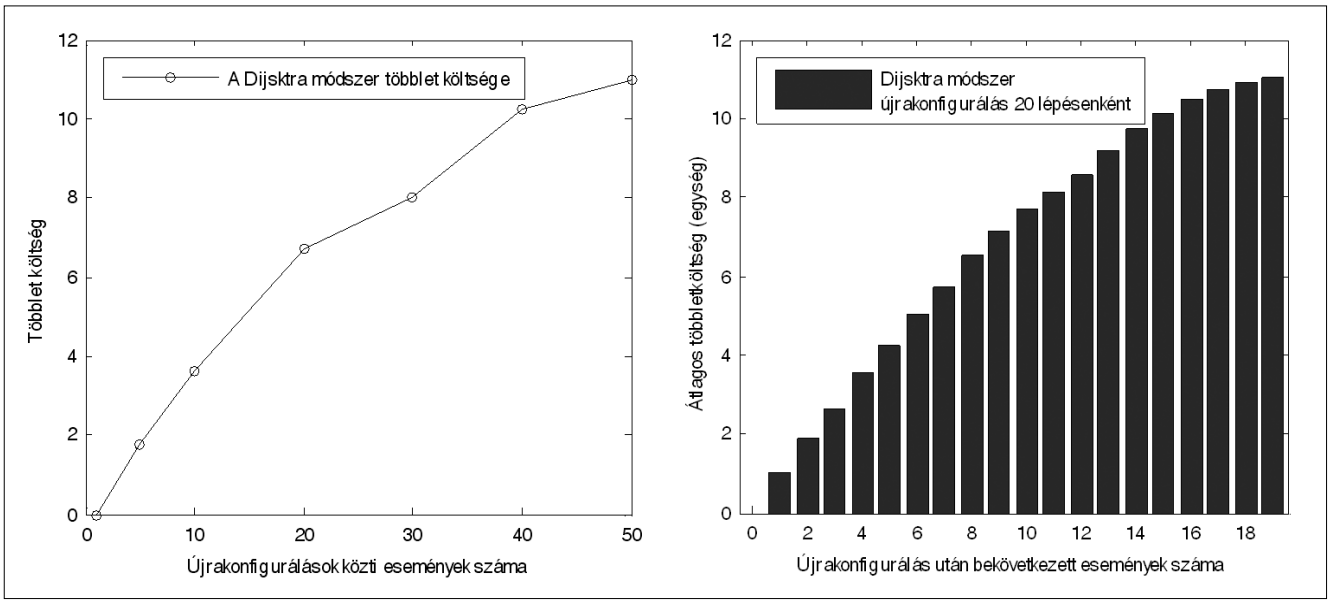
A *3. ábra* a teljes elvezetés költségét szemlélteti a bekövetkezett események függvényben. Esemény alatt a célcsomópontok halmazának megváltozását értjük. Az alsó görbe a mindenkori optimális elvezetés költségét mutatja. A felső a Dijkstra-alapú elvezetést jelöli újrakonfigurálás nélkül. A középső 30 eseményenkénti újrakonfigurálás mellett mutatja a költségek alakulását.

A szimuláció során Dijkstra algoritmusa átlagosan több mint 60%-kal felülmúlta költségben az optimális megoldást. Az újrakonfigurálásos görbe általában gyorsan távolodik az optimális megoldástól. A divergálás egészen addig folytatódik, amíg nem következik be a következő újrakonfigurálás. Bár az újrakonfigurálás egyértelműen kedvező (lásd 3. ábra), jósága mégis a hálózat topológiájától függ, a használt dinamikus elvezető algoritmustól és az újrakonfigurálás gyakoriságától.

Ezért megvizsgáltuk, hogy a különböző algoritmusok (részletek a 4. szakaszban) költségei hogyan viszonyulnak egymáshoz és a Dijkstra-algoritmushoz különböző újrakonfigurálási időket (periódusidő) feltételezve. Az eredményeket a *4. ábra* mutatja. Egyértelmű, hogy a különböző újrakonfigurálás nélküli algoritmusok, a jelen szimuláció szerint, az optimálistól igen távol esnek. Átlagosan 35-60% körüli mértékben múlják felül az optimális megoldást. Ugyanakkor jelentős megtakarítás érhető el ismétlődő újrakonfigurálással. A várakozásoknak megfelelően a kisebb újrakonfigurálási idő kedvezőbb átlagos költséget jelent. Természetesen az újrakonfigurálás számításigényes és más hátrányokkal is rendelkezhet (1.1. szakasz). Ezeket a hátrányokat nem számítottuk bele a költségekbe.

4. ábra
Az egyes algoritmusok átlagos útvonal-választási költsége, illetve a legrövidebb út módszerének (Dijsktra) költsége különböző újrakonfigurálási értékek mellett





5. ábra
Átlagos plusz költség az újrakonfigurálási gyakoriság (bal), illetve az újrakonfigurálás óta eltelt idő (jobb) függvényében

Szintén megvizsgáltuk, hogy az újrakonfigurálási periódus hossza hogyan befolyásolja az átlagos többletköltséget (veszteséget).

Az *5. ábra* baloldali görbéje szemlélteti az átlagos plusz költséget az újrakonfigurálási gyakoriságának függvényében. Az ábra egy folyamatosan csökkenő meredekségű (telítődő, parabola-jellegű) görbét mutat. Tehát nagyobb nyereséghez gyakoribb újrakonfigurálás szükséges. Ritka újrakonfigurálás hozzáadott költségei között nincs jelentős eltérés.

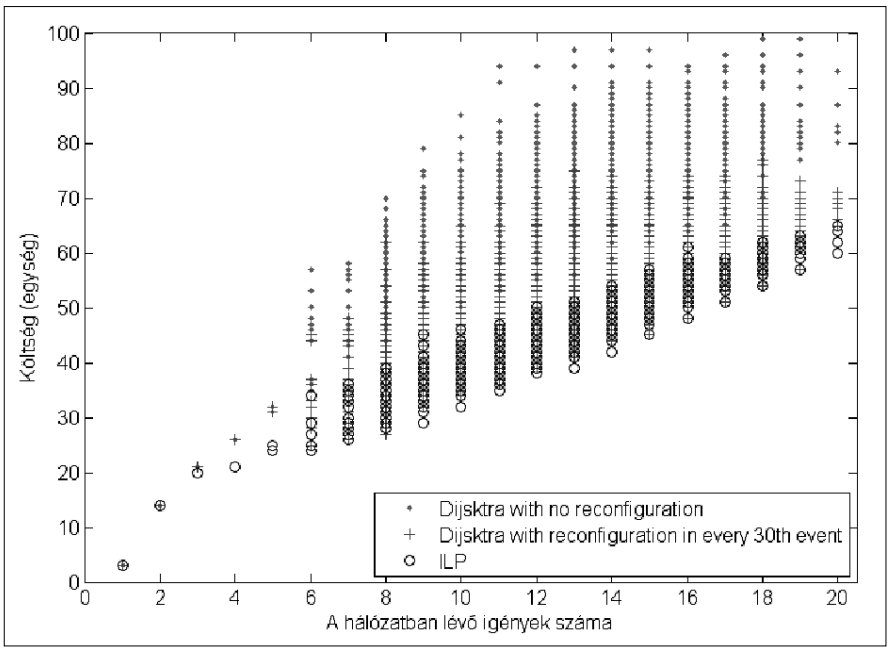
Az *5. ábra* jobb oldala azt szemlélteti, hogy az újrakonfigurálás óta eltelt eseményszám növekedésével milyen gyorsan távolodik a megoldás az optimálistól. Az előzőhöz hasonlóan ez is ellaposodó görbe, csökkenő meredekséggel. Ez arra utal, hogy a heurisztikus folytatás az első pár esemény során már nagymértékben távolodik az optimális megoldástól, majd a romlás üteme lassul.

A *6. ábra* az elvezetés költségét mutatja az egyes algoritmusok esetében a cél-csomópontok számának függvényében. Minden pont a szimuláció során adott pillanatbeli megoldást reprezentál. Ahogyan az várható, az elvezetés költsége nő az igények számának növekedésével. Érdekes megfigyelés, hogy az újrakonfigurálás melletti legrövidebb utak módszerének jellemző pontjai általában az optimális megoldáshoz tartozó és az újrakonfigurálás nélküli pontok jellegzetes tartományai között helyezkednek el.

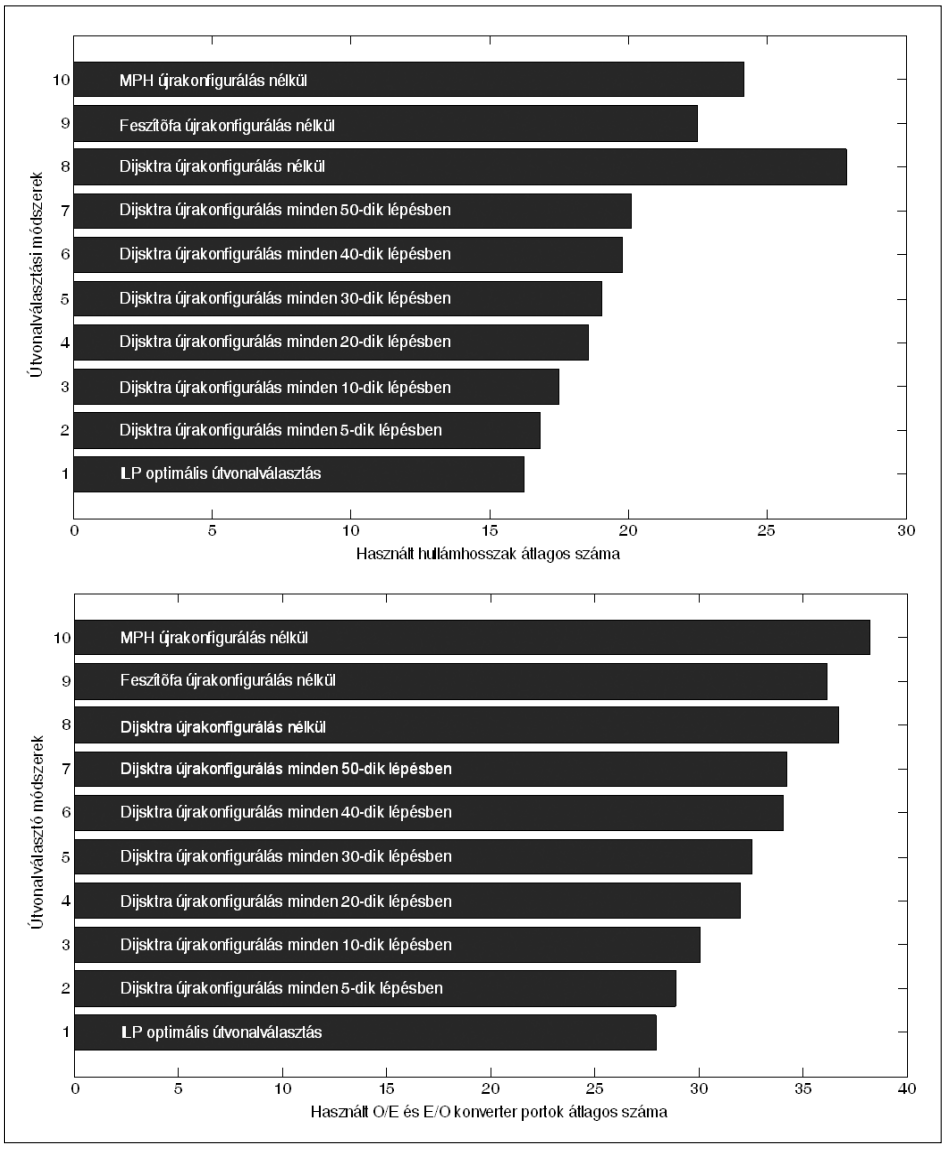
A költségek mellett más, az elvezetés során lényeges hálózati elemek kihasználtságát is vizsgáltuk (például használt O/E, E/O portok, hullámhosszak száma). A korábbiakhoz hasonló eredményekre jutottunk.

A *7. ábra* a szükséges hullámhosszak, illetve O/E és E/O konverziós egységek számát mutatja a különböző algoritmusokra.

Lényeges megfigyelés az, hogy a Dijkstra-módszer kimagaslóan nagy hullámhossz igényű, miközben O/E és E/O port használata kisebb, mint az MPH heurisztikáé. Mindkét erőforrás használtsága csökkenthető az újrakonfigurálási gyakoriságának növelésével.



6. ábra
Az elvezetés költsége
a célok számának függvényében



7. ábra
A szükséges hullámhosszak átlagos száma (felső), illetve a szükséges O/E és E/O konverziós egységek száma (alsó) a különböző algoritmusokkal és újrakonfigurálási periódus hosszokkal

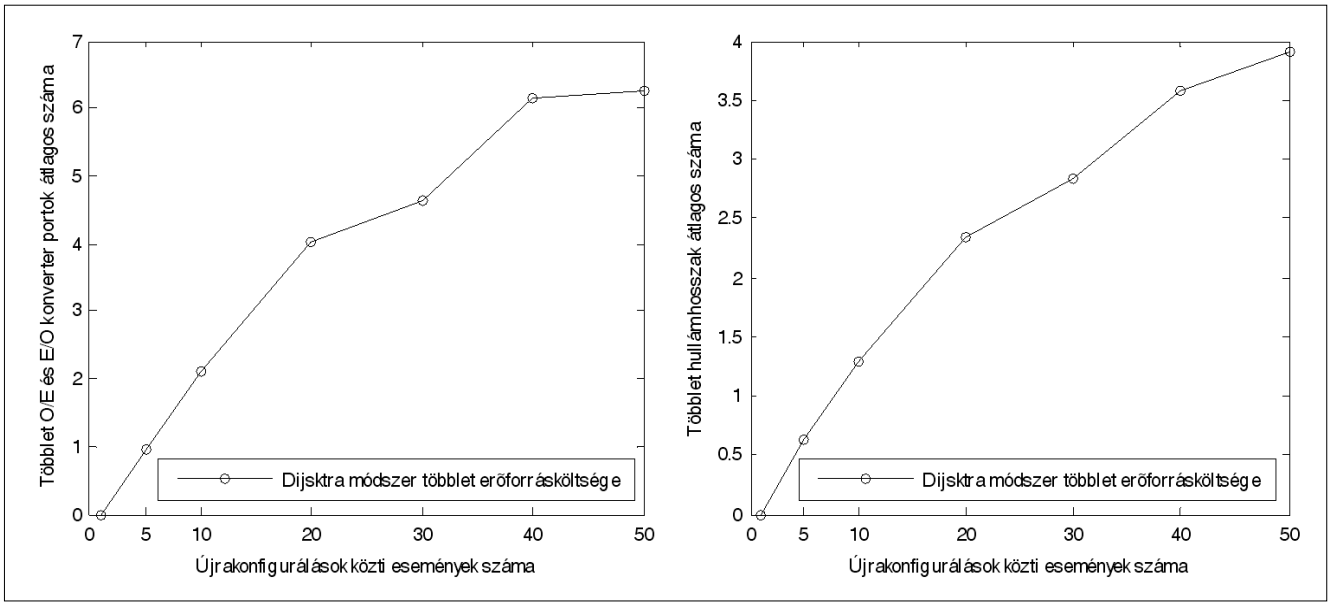
A *8. ábra* azt mutatja, hogy hogyan változik a többlet konverziós portok és a hullámhosszak száma az újrakonfigurálások gyakoriságának csökkentésével. A szimulációs eredmények a korábbiakhoz hasonló, telítődő függvényt adnak.

A *9. ábra* alapján a konverziós egységek száma közel lineáris, míg a használt hullámhosszak száma négyzetgyök-jelleggel nő a szaturációs pontig.

6. Összefoglalás

Cikkünkben megmutattuk, hogy a dinamikusan változó multicast fák esetén az újrakonfigurálás előnyös a hálózati szolgáltató számára. A költségveszteség (beleértve a hálózati erőforrásokat, mint például a hullámhosszak vagy a konverziós

8. ábra
Többlet O/E és E/O konverterek (bal), illetve hullámhosszak átlagos száma az újrakonfigurálási periódus hosszának függvényében



portok száma) csökkenthető az optimális elvezetéshez való visszatéréssel. Mivel a fa tulajdonságai az újrakonfigurálás után gyorsan romlanak, ezért azt gyakran meg kell ismételni.

A cikkben megpróbáltuk megbecsülni a várható költségmegtakarítást, illetve a periodikus újrakonfigurálás várható hatásait. Az eredmények arra utalnak, hogy az újrakonfigurálás költséghatékony megoldás lehet, ha az egyes események közötti átlagos idő elégséges ahhoz, hogy a hullámhossz- és egyéb erőforrás-megtakarítások ellensúlyozzák az újrakonfigurálási művelet által okozott technikai problémákat (például a fennakadás nélküli átállás a régi fáról az újra). Ezeket azonban meg kell megoldani ahhoz, hogy az újrakonfigurálás jól használhatóvá váljon.

Irodalom

[1] B. Quinn and K. Almeroth, "IP multicast applications: Challenges and solutions", IETF RFC 3170, September 2001.

[2] Madhyastha et al., "Grooming of multicast sessions in WDM ring networks", Optical Networking and Comm. (OptiComm 2003), November 2003.

[3] G. V. Chowdhary and C. S. R. Murthy, "Grooming of Multicast Sessions in WDM Mesh Networks", Workshop on Traffic Grooming, 2004.

[4] X. Zhang et al., "Constrained Multicast Routing in WDM Networks with Sparse Light Splitting", Journal of Lightwave Technology, Vol. 18., No.12, p.1917., December 2000.

[5] X. H. Jia et al., "Optimization of Wavelength Assignment for QoS Multicast in WDM Networks", IEEE Transactions on Communications, Vol. 49., No.2, February 2001.

[6] Fatih Köksal and Cem Ersoy, "Multicasting for all-optical multifiber networks", Journal of Optical Networking, Vol. 6., No.2, pp.219–238., January 2007.

[7] D. Yang and W. Liao, "Design of light-tree based logical topologies for multicast streams in wavelength routed optical networks," In Proc. of the IEEE Information Communications (INFOCOM), San Francisco, CA, April 2003.

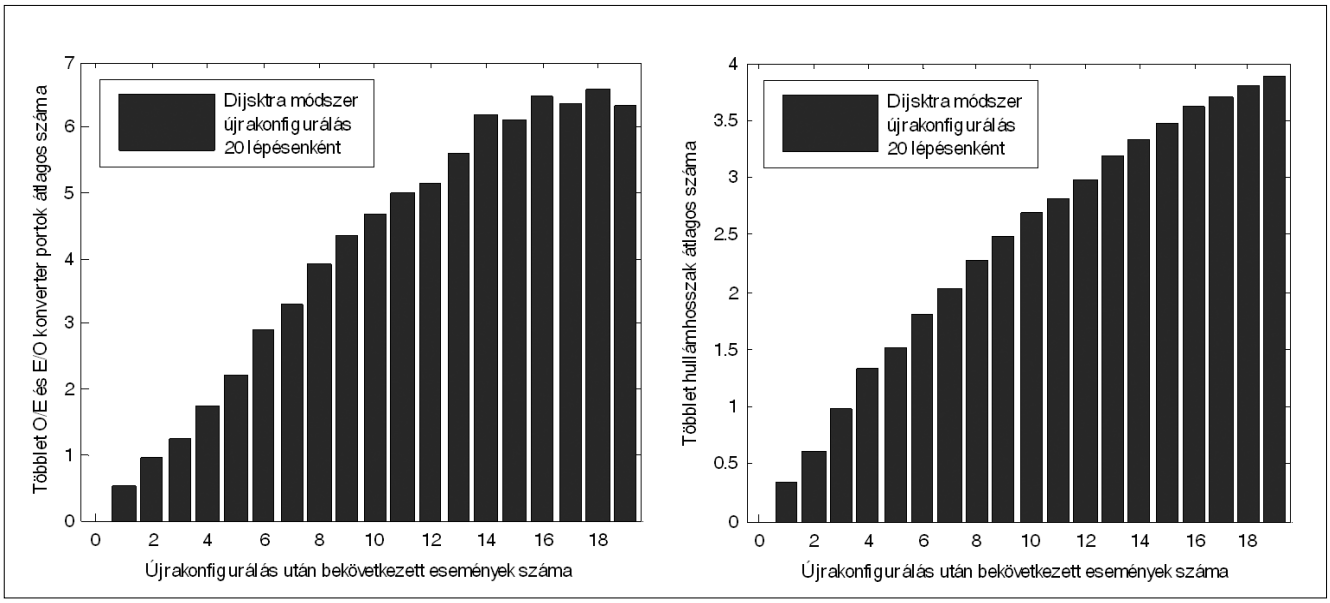
[8] R. Mustafa, A.E. Kamal, "Design and provisioning of WDM networks with multicast traffic grooming", IEEE Journal on Selected Areas in Communications, Vol. 24., No.4, 2006.

[9] Alexander Schrijver, "Theory of Linear and Integer Programming", John Wiley and Sons, 1998.

[10] X. Huang et al., "Multicast Traffic Grooming in Wavelength-Routed WDM Mesh Networks Using Dynamically Changing Light-Trees", Journal of Lightwave Technology, Vol. 23., No.10, October 2005.

[11] Ahmed E. Kamat et al., "Algorithms for multicast traffic grooming in WDM mesh networks", IEEE Communications Magazine, Vol. 44., No.11, November 2006.

9. ábra
Többlet O/E, E/O konverterek (bal), illetve hullámhosszak átlagos száma az újrakonfigurálás óta eltelt események számának függvényében



[12] Ahmad Khalil et al.,
"Dynamic provisioning of low-speed unicast/multicast traffic demands in mesh-based WDM optical networks", Journal of Lightwave Technology, Vol. 24., No.2, February 2006.
[13] Keyao Zhu et al.,
"Traffic Engineering in Multi-granularity Heterogeneous Optical WDM Mesh Networks Through Dynamic Traffic Grooming", IEEE NETWORK, Vol. 17., No.2, pp.8–15., March/April 2003.
[14] Jianping Wang, Biao Chen,
"Dynamic Wavelength Assignment for Multicast in All-Optical WDM Networks to Maximize the Network Capacity", IEEE Journal On Selected Areas in Communication", Vol. 21., No.8, October 2003.
[15] G. Chowdhary, C. S. R. Murthy,
"Dynamic multicast traffic engineering in WDM groomed mesh networks", Workshop on Traffic Grooming, 2004.
[16] C. Bowortummarat et al.,
Light-tree based protection strategies for multicast traffic in transport WDM mesh networks with multifiber systems, IEEE International Conference on Communications, Vol. 3., June 2004.
[17] E. Dotaro, M. Vigoureux, D. Papadimitriou,
"Multi-Region Networks: Generalized Multi-Protocol Label Switching (GMPLS) as Enabler for Vertical Integration", Alcatel Technology White Paper, February 2005.
[18] T. Cinkler et al.,
"Configuration and Reconfiguration of WDM networks", European Conference on Networks and Optical Communications (NOC'98), Manchester, UK, 1998.
[19] T. Cinkler,
"ILP formulation of Grooming over Wavelength Routing with Protection", 5th Conf. on Optical Network Design and Modeling, ONDM 2001, Wien, February 2001.
[20] A. Betker et al.,
"Reference transport network scenarios", Technical report, BMBF-Project MultiTeraNet, 2003. www.pt-it.pt-dlr.de/_media/MTN_Referenz_Netze.pdf
[21] Thomas H. Cormen et al.,
"Introduction to Algorithms", Section 23.2: "The algorithms of Kruskal and Prim", Second Edition, MIT Press and McGraw-Hill, 2001. pp.567–574.
[22] M. Ali, J. S. Deogun,
"Cost-effective implementation of multicasting in wavelength-routed networks", Journal of Lightwave Technology, Vol. 18., No.12, 2000.

Hírek

A Novell bejelentette a SUSE Linux Enterprise Real Time operációs rendszer új fejlesztéseit és legújabb partnermegállapodásait a Novell alacsonykésleltetésű Linux-megoldásainak kiterjesztéséhez. Az asztali gépektől az adatközpontokig terjedő igényeket kiszolgáló SUSE Linux Enterprise platformra épülő Real Time tartalmazza azokat a kernelfejlesztéseket, csomagokat, eszközöket és alkalmazásokat, amelyek alapvető elemei egy nagyteljesítményű, determinisztikus és alacsony késleltetésű operációs rendszernek. A valósidejű technológia segítségével az ügyfelek szegmentálhatják a processzoridőt, a hálózati sávszélességet és az egyéb hardvererőforrásokat a magas prioritású, kulcsfontosságú munkaterhelésekhez. Ez biztosítja, hogy az alacsonyabb prioritású munkafolyamatok vagy rendszerfeladatok rendszerhívásai nem szakítják meg ezeket a munkaterheléseket, és kiszámítható teljesítményt nyújtanak az időkritikus környezetekben.

Közép-Kelet-Európa 23 – köztük két magyar – fiatal mérnöke kezdte meg egyéves gyakornoki munkáját a Ciscónál július végén. A szakemberek a Sales Associate Program keretében csatlakoznak a világ 144 országából származó kollégáikhoz és az év során elsajátítják a legújabb hálózati és kommunikációs technológiákat, valamint finomítják készségeiket a munkahelyi kommunikáció és együttműködés terén. A cég 3 éve indította el programját Közép-Kelet-Európában azzal a céllal, hogy gyakornoki munkát biztosítson a frissdiplomás műszaki szakemberek számára. Ebben az évben 1400 pályázóból választották ki azokat, akik lehetőséget kaptak arra, hogy a Cisco rendszermérnökeként dolgozzanak. Az amszterdami központú program kurzusai Cisco-minősítésekre, valamint műszaki és kereskedelmi készségek fejlesztésére koncentrálnak és általános bevezetést adnak a Cisco vállalati kultúrájába.

Az APC, az energiaellátási és hűtési megoldások piacvezető szállítója bemutatta új Capacity Manager és Change Manager szoftveralkalmazásait. Ezeknek az adatközpont-kezelő eszközöknek a megjelenésével az APC az első és egyetlen olyan vállalat, amely a három létfontosságú elemet; az adatközpontok tervezését, üzemeltetését és kezelését integrálja a fizikai réteg teljesítményének és hatékonyságának optimalizálása érdekében. Az adatközpontok vezetői gyorsan, biztosan tervezhetik meg és végezhetik el a rack-alapú eszközök beépítését, hiszen a szoftverek segítségével képesek az optimális helymeghatározásra és automatikusan létrehozhatják a munka elvégzéséhez szükséges megrendelőket.

Újgenerációs anonim böngészők

GULYÁS GÁBOR, SCHULCZ RÓBERT

Budapesti Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék
schulcz@hit.bme.hu

Kulcsszavak: web, böngészők, anonimitás, paradigma

A web már régóta felvet adatvédelmi kérdéseket a látogatók számára is: bizonyos szolgáltatók megfigyelik a felhasználók tevékenységeit, követik őket, adatbázist építenek ízlésvilágukról. Az anonim böngészők megoldást kínálnak a felhasználóknak, elrejtik őket a figyelő szemek elől. A cikkben bemutatunk néhány követésre használt módszert, illetve egy új, az anonimizáló szolgáltatásokra vonatkozó konstrukciós paradigmát, majd egy ez alapján értelmezett osztályozási rendszert is az anonim böngészők besorolásához.

1. Bevezető

A web böngészése során a látogatott weboldalak információszolgáltatás mellett adatokat is gyűjtenek; hálózati jelenlétünk, a használt böngésző, a böngészés folyamata olyan információkat árulnak el rólunk, amely tevékenységünket megfigyelhetővé, követhetővé teszi. Később az összegyűjtött információkat tartalmazó profil adatbázis direktmarketing célokra használható (célzott hirdetések, SPAM), kereskedni lehet vele, vagy akár dinamikus, személyre szabott árlista generálására használható webes boltokban. Ez az adatgyűjtés sokszor nem egyetlen weboldalra, vagy egy on-line tartalomszolgáltató hálózatára vonatkozik, hanem a szolgáltatók más partnerekkel együtt dolgozhatnak, így szélesebb skálájú profilt készítve a felhasználóról.

Bizonyos szituációkban valamilyen szolgáltató (vagy például egy cenzúrázó szerv) ki szeretné deríteni egy felhasználó személyét, az aktivitásáról készít naplót, vagy – a leggyakoribb esetben – blokkolná bizonyos tartalmak elérését.

Valamennyi felsorolt esetben megoldást kínálnak az anonim böngészők, garantálva a szolgáltatásban az anonimitást, és lehetőséget adva a cenzúra megkerülésére. Az anonim felhasználó tevékenységeit a tartalmak szűrésével összeköthetetlenné, titkosítással pedig a külső szemlélők számára megfigyelhetetlenné teszik. Továbbá biztosítják, hogy a felhasználónak az anonimitásból eredően nem fedhető fel valós személye, mivel tevékenységei sem illethetők pszeudonim azonosítóval.

A cikk bemutatja a külső-belső világ konstrukciós paradigmát, amelynek minden anonimizáló rendszernél szükséges feltétel és azt is körüljárjuk, milyen feltételeknek kell teljesülnie a paradigma megvalósításához a webes világban. Egy új, továbbfejlesztett osztályozási szempontrendszert is ismertetünk, amely a paradigma teljesítésének vizsgálatában játszik szerepet, így arra épül és felöleli [1]-ben olvasható attribútumokat is. Továbbá vizsgálunk olyan követési módszereket és technikákat, amelyek kiegészítő képet adnak [1]-hez képest.

2. Külső-belső világ konstrukciós paradigma

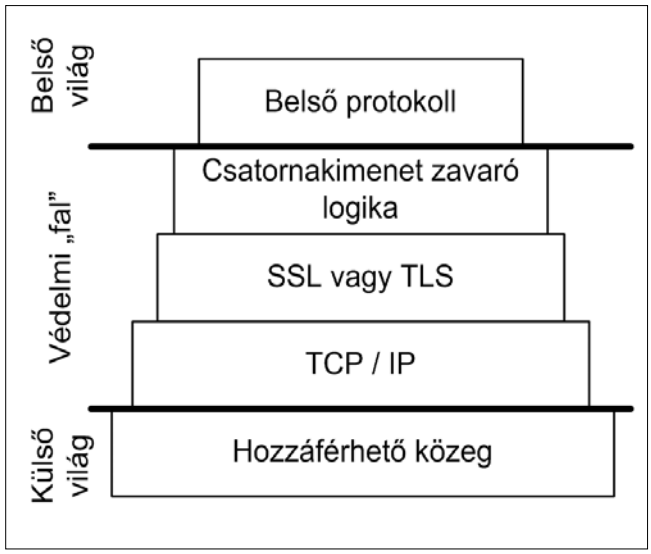
2.1. A paradigma általában: anonimitást nyújtó rendszerek

A külső-belső világ paradigma szerint egy szolgáltatás akkor nyújthat anonimitási lehetőséget a belső felhasználók számára, ha a belső protokoll működése külvilágtól teljesen szeparált (a belső műveletek forgalomanalízistől, megtekintéstől és módosítástól védettek), valamint egy szolgáltatásbéli felhasználó nem képes kompromittálni más belső felhasználó anonimitását, kivéve, ha az adott művelet a belső működést leíró protokoll szerint szabályos (például, ha operátor tehet ilyet).

Szükséges az alábbi két feltétel teljesülése:

1. A külső világ a megfelelő anonimizáló protokoll által le van választva a belső világ működéséről, tehát egy olyan fél, amely a szolgáltatásnak nem résztvevője, képtelen megfigyelni a belső működést, valamint befolyásolni azt.

1. ábra Egy lehetséges protokollverem a külső világ szeparációjához



2. A belső protokollban a tervezése során biztosítani kell az anonimitás lehetőségét a belső világ felhasználói között (nem csak opció, kötelező is lehet).

Ennek egy lehetséges megvalósítása látható a *1. ábrán*. A védelmi fal funkciójában egy anonimizáló protokoll kell, hogy legyen, amely vagy külön hálózatra épül, vagy a kommunikációs protokollba van beépítve, mint az ábrán.

2.2. Szükséges kritériumok az anonimitás teljesüléséhez

Az anonimitás teljesüléséhez további kritériumoknak kell teljesülnie, amelyet a külső-belső világ paradigmában megfogalmazott szeparációs modellnek meg kell valósítania (1-2. kritérium), és a belső világ protokollnak is ki kell elégítenie (4. kritérium). A kritériumokat [11] a négy fő magánszféravédő és adatvédelmi kritérium alapján fogalmaztuk meg a webes világ értelmezésében.

A kritériumok értelmezésénél a *2. ábrán* látható viszony áll fenn: a felhasználó kéréseket küld B webes kiszolgálónak, ezt próbálja megfigyelni C külső megfigyelő. A kritériumok nem vonatkoznak a D anonimizáló szolgáltatásra, mivel az a működéséből fakadóan ismeri a forgalmazott tartalmat és a kommunikáló feleket is, így feltesszük, hogy az megbízható.

Az A felhasználó és a D anonimizáló hálózat között a kapcsolat titkosított, így a tartalomból a kommunikáció célja nem deríthető ki. Az anonimizáló szolgáltatás egyik célja annak a megvalósítása is, hogy az itt megfigyelhető üzeneteket a D-B szakaszon megjelenő üzenetekkel ne lehessen párba állítani.

1. *Összeköthetetlenség*:
   sem C külső megfigyelő, sem a B kiszolgáló nem képes eldönteni A felhasználó két üzenetéről, hogy azokat ugyanaz az A felhasználó küldte-e, sem pedig, hogy a két üzenetnek egyezik a feladója.
2. *Megfigyelhetetlenség*:
   a kommunikáció tartalmát csak A és B ismerhetik.
3. *Pszeudonim azonosítóval rendelkező felhasználó:*
   a felhasználó tevékenységei összeköthetőek, és egy fedőnévhez, vagy egyéb azonosítóhoz kapcsolhatóak, de ez alapján a valódi személyazonosság nem deríthető fel.
4. *Anonim azonosítóval rendelkező felhasználó:*
   a felhasználó cselekedetei viszony szinten sem köthetőek össze, így üzenetenként független pszeudonim azonosítóval rendelkezik (1 és 3. kritérium).

A kritériumok növekvő sorrendben erősödnek, és az anonimitás eléréséhez valamennyinek teljesülnie kell.

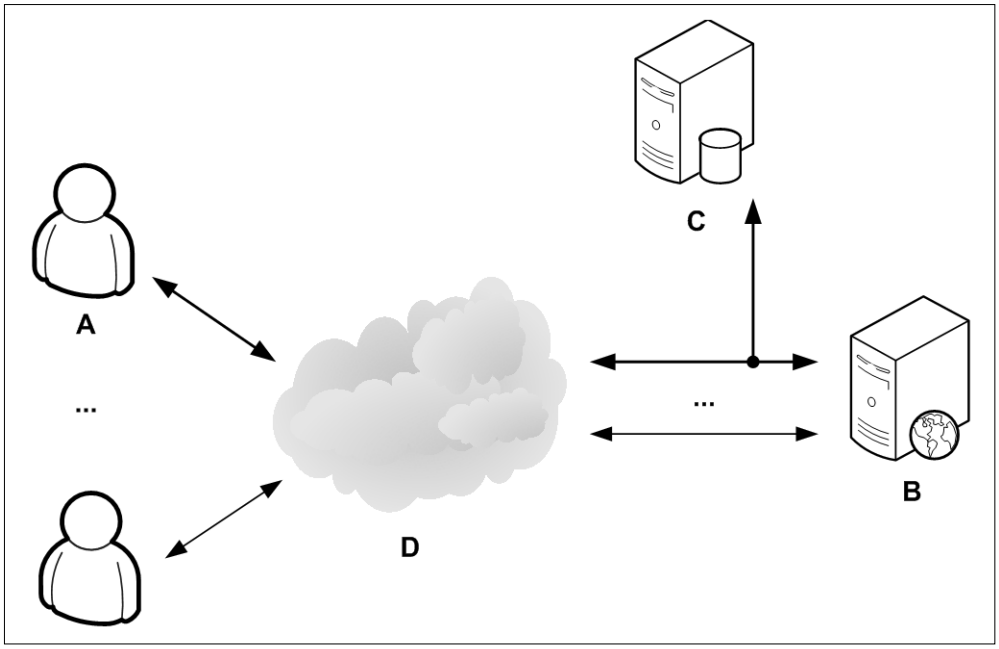
2.3. A paradigma és az anonim böngészők

A vizsgált szolgáltatások a külső világ szeparációját anonimizáló hálózattal oldják meg, amelyek általában TOR [9] hálózatot jelent, mint például a TorPark [10] szolgáltatása esetén. Fontos kiemelni, hogy a szolgáltatás célja nem a belső világ valamennyi szereplőjének az anonimizálása, hanem csak a felhasználóké, a többi résztvevő (például hirdetők) felé.

Anonim böngészők esetén a belső protokoll jól definiált és szabványosított, így ebben az esetben csak tartalomszűrésre van lehetőség. A tartalomszűrés igen sokrétű lehet, mert – amint az osztályozási szempontrendszerből és a következőkben bemutatott módszerekből is látható – számos elem és trükk használható a felhasználó kompromittálására, a szűrés megkerülésére is.

3. Szabotázs az anonimitás ellen – követési lehetőségek

Ezen módszerek közös jellemzője, hogy a felhasználót az első találkozás alkalmával megjelölik valamilyen módon, s ezt a – lehetőleg pszeudonim – azonosítót felhasználva a későbbiekben a felhasználó profilját adatbázisban tárolják.



2. ábra
Hálózati elrendezés a kritérium vizsgálatnál

3.1. Yahoo Web Beacons

A Yahoo Web Beacons egy jó példa arra, hogy megfigyelni nem csak titokban lehet, hanem nyilvános üzletet is lehet csinálni belőle. A Yahoo adatvédelmi szabályzatában található leírás szerint [3] az azonosítókat a saját oldalaikon a felhasználók azonosítására (például számlálóknál), a felület személyre szabására alkalmazzák, amellett névtelen információgyűjtésre a partnereiknél.

A szolgáltatás nyomkövetése OPT-OUT, de a [3] oldalon található linken keresztül lemondható. Ez csak a használt böngészőre érvényes, ráadásul az azonosító süti (cookie) megmarad, így a követés a sütik törlése után folytatódik.

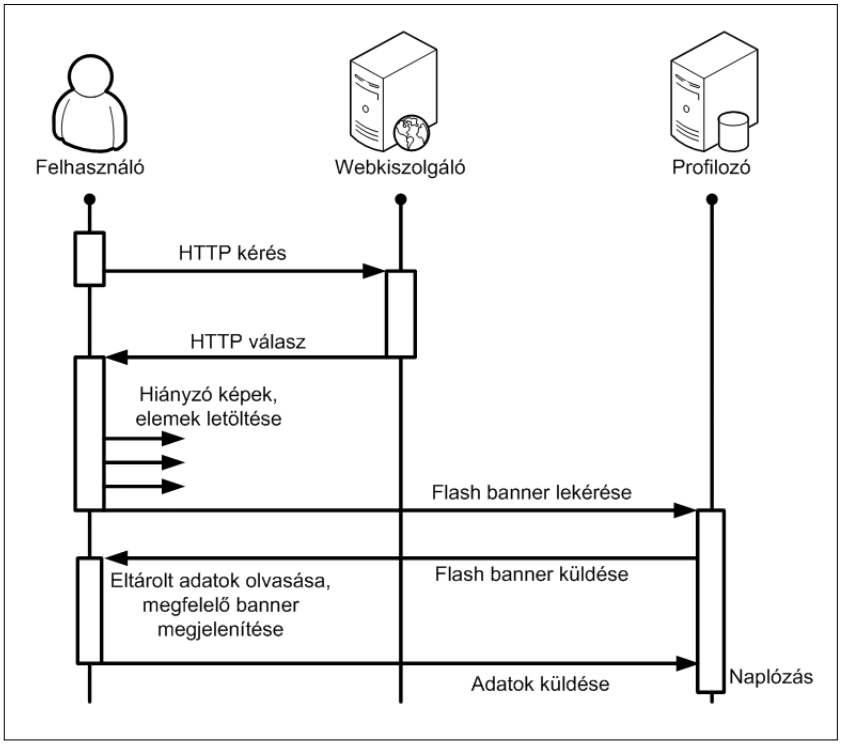
A web beacons a web poloskákhoz [2] hasonló megoldást alkalmaz. A weboldalakon olyan fájlokra mutató linkeket helyez el, amelyek letöltése után a böngészőt egy süti jelöli majd meg. A Yahoo rendszere ennek a sütinek a segítségével követi a felhasználót, és végzi későbbi tevékenységeit.

3.2. Flash PIE

A Flash PIE [4] (Persistent Identification Element, perzisztens azonosító elem) hasonló a web bugokhoz. Működésükben is hasonlítanak és az azonosító tárolása sütikhez hasonló elemek segítségével, az SO-kkal (Shared Object, megosztott objektumok) történik [7].

Az SO-k esetében több, összejátszó weboldalnak könnyebb dolga van, mint a sütiknél, ugyanis esetükben az elérhetőség csak opcionális paraméter. Ennek köszönhetően az oldalak kollaborációja, az egymás között megosztott információk cseréje rendkívül egyszerűvé válik, ugyanígy a felhasználók azonosítása (és a követése) is.

3. ábra
A flash PIE működése



Mivel az SO-k nem a böngésző sütijei között tárolódnak, így könnyen rejtve maradnak a felhasználó szeme elől. Az is előfordulhat, hogy a felhasználó által használt takarító szoftverek nem foglalkoznak ezekkel az objektumokkal, egyszerűen figyelmen kívül hagyják őket.

Egyszerű és általános védekezési megoldás lehet a SO-k tiltása, azonban ez bizonyos szolgáltatások használhatatlanná válásához vezethet, csak úgy, mint ahogy weboldalaknál a sütik mellőzése. A Flash-objektumok bináris formátumúak, így a tartalom szerinti szűrésük nem lehetséges anonim böngészők számára.

3.3. Követés gyorsítótárba mentett JavaScripttel

A JavaScript gyorsítótárazott fájljai alapján történő követés [5] hasonló a web poloskák módszeréhez: a weblaphoz egy JavaScript állományt csatolnak, amelyben egy változóban eltárolnak egy azonosító értéket. Ez az érték elérhető lesz a későbbi látogatások alkalmával, amíg a gyorsítótár nem frissül.

Ezt lehet használni az azonosítót tartalmazó süti pótlására, amelyből minden betöltéskor, ha a süti nincs jelen, újra létrejön és fordítva is: ha a süti létezik, a JavaScript állományba a kiszolgáló a süti értékét írja. Ezt a módszert szemlélteti a [8].

4. Taxonómia a külső-belső világ paradigma alapján

A taxonómia két fő csoportját a külső világ szeparációja, valamint a szűrési mechanizmusok alkotják. Ezen kívül egyéb kritériumokat is érdemes vizsgálni, mert jelentősen befolyásolják a böngésző értékét.

Az osztályozási szempontrendszer célja az anonim böngészők funkcionalitási vizsgálatának lehetővé tétele, a különböző attribútumokba sorolás segítségével. Az alábbiakban az osztályozási szempontrendszer látható.

Külső világ szeparációs megoldásai
- HTTPS
- Alkalmazott anonimizáló protokollok

Belső világ védelme
- Tartalomszűrés helye
  - Kliensoldalon
  - Szerveroldalon
- Szűrt tartalmi elemek, információk
  - Kiszolgálótól érkező
  - JavaScript
  - Java
  - Flash
  - Klienstől távozó
  - Böngésző, operációs rendszer információk
  - URL referer
- „Rosszindulatú” elemek szűrése
  - Reklámok

  - – Felugró ablakok
- Sütikezelési szintek
  - – Szerveroldali tárolás
  - – Bizonyos sütik szűrése
  - – Blokkolás (mind)
- Egyéb helyi nyomok törlése
  - – Gyorsítótár
  - – Előzmények
- HTTPS átjátszása (és szűrése)

Hordozhatóság

- Alternatív csatlakozási pont
  - – Klienseken keresztül
  - – Dedikált átjátszók
- Szolgáltatás típusa
  - – Telepített proxy alkalmazás
  - – Webes proxy
  - – Hagyományos proxy (csak be kell állítani)
  - – Hordozható alkalmazás (pl. USB meghajtón)
- Operációsrendszer-, böngészőfüggetlenség

Egyéb kritériumok

- Saját reklámok a szolgáltatásban
- Kezelőfelület alkalmassága (szolgáltatás elhagyása figyelmetlenségből, vagy túl zavaró, sok helyet foglal)
- Forgalomkorlátozás
- Sebességkorlátozás
- Naplózási feltételek

5. Összefoglalás

Webes privátszférát érintő kérdésekkel egyre többen ismerkednek meg, és egyre többen használnak anonim böngészőket. A felhasználók és kutatók számára is fontos, hogy ezeket a szolgáltatásokat objektíven, az elvárt tudásuk alapján értékelni tudják. Ehhez nyújt segítséget a cikkben bemutatott osztályozási szempontrendszer, valamint a kritériumok, amely a szolgáltatás architektúrájának ellenőrzésében nyújtanak segítséget.

Jelenleg kevés szolgáltatás van, amely teljes megoldást kíván nyújtani a felsorolt problémákra, de nagyobb problémát jelent, hogy a jó szolgáltatások fizetősek. Például a TorPark-ot e cikk írásakor fizetőssé alakítják át, miközben az ingyenes, mindenki számára elérhető változat egyszerűen használhatatlan a rendelkezésre bocsátott túl kicsi sávszélesség miatt.

A szerzők remélik, hogy a jövőben a privátszférához való jog nem csak egy szlogen lesz, hanem tény, és nem azon fog múlni a jog teljesülése, hogy valakinek van-e rá pénze, vagy sem.

Irodalom

[1] Gulyás G.:
Anonim-e az anonim böngésző?
Technológiák és szolgáltatások elemzése.
Alma Mater sorozat.
BME GTK ITM, Budapest, 2006. március.

[2] Hullám G.:
A web bug technológia – barát vagy ellenség?
Székely Iván–Szabó Máté Dániel (szerk.):
Szabad adatok, védett adatok.
Alma Mater sorozat.
BME GTK ITM, Budapest, 2005. március.

[3] Yahoo Web Beacons
http://info.yahoo.com/privacy/us/yahoo/webbeacons/details.html

[4] Flash PIE
http://www.mistered.us/tips/flash/settings.shtml

[5] JavaScript alapú követés gyorsítótárral
http://www.mukund.org/blog/101/

[6] Electronic Privacy Information Center
http://www.epic.org

[7] Flash: SO-k (megosztott objektumok)
http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_16194

[8] JavaScript gyorsítótárazást használó követési módszer
http://www.mukund.org/files/archive/2006/09/14/tracking-using-cache.html

[9] TOR anonimizáló hálózat
http://tor.eff.org/

[10] TorPark
http://www.torrify.com/

[11] Gulyás G.:
Az anonimitás és a privacy kérdései a csevegő szolgáltatásokban.
Tanulmányok az információ- és tudásfolyamatokról 11.
BME GTK ITM, Budapest, 2007. május.

WLANpos: Wi-Fi alapú beltéri helymeghatározó rendszer

NÉMETH LÁSZLÓ HARRI, KIS ZOLTÁN LAJOS, SZABÓ RÓBERT

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
{nemethl, kiszl, robert.szabo}@tmit.bme.hu

Lektorált

Kulcsszavak: vezetéknélküli hálózatok, Wi-Fi, helyfüggő szolgáltatások, helymeghatározás, pozícionáló algoritmus

A vezetéknélküli számítógép-hálózatok elterjedése a vezetéknélküli technológia folyamatos fejlesztésének köszönhetően ma már hétköznapinak számít. Képesek felvenni a versenyt a vezetékes szolgáltatásokkal mind kapacitás, mind megbízhatóság tekintetében, miközben a felhasználó szabadon helyet változtathat. Ehhez kapcsolódóan alakult ki a felhasználó pozíciójától függő szolgáltatások köre. Szükségessé válik egy helymeghatározó rendszer kialakítása, amely beltérben is használható és megfelelő pontossággal rendelkezik ahhoz, hogy a rá épülő alkalmazások igényeit kiszolgálja. Célunk egy Wi-Fi hálózat és egy szabványos Wi-Fi eszköz segítségével a felhasználó helyének lehető legpontosabb meghatározása volt. A problémára született megoldások általában drágák, nagy számítási igénnyel rendelkeznek, vagy csak korlátozott térben alkalmazhatók. A BME Távközlési és Médiainformatikai Tanszékén kifejlesztett WLANpos egy teljes értékű alkalmazás, amely megfelelő megoldást igyekszik kínálni; képes térképek megjelenítésére, rajzolására és persze a pozícionálás eredményének megjelenítésére is.

1. Bevezetés

A helyfüggő szolgáltatások és a mindent körülvevő számítástechnikai elképzelések méginkább szükségessé teszik egy helymeghatározó rendszer kidolgozását, mely beltéren is megfelelő pontossággal rendelkezik. A GPS vagy a mobiltelefon hálózatok egyelőre nem képesek megfelelni ennek az elvárásnak, az egyre nagyobb lefedettséggel bíró WLAN segítségével történő pozícionálás viszont igen. A WLAN szabványok egy fajtája a Wi-Fi-nek nevezett szabvány.

A Budapesti Műszaki és Gazdaságtudományi Egyetem Távközlési és Médiainformatikai Tanszékén végzett fejlesztés célja egy Wi-Fi hálózat és egy szabványos Wi-Fi eszköz segítségével a vevő, azaz a felhasználó helyének lehető legpontosabb meghatározása. Már születtek megoldások erre a problémára, de általában drágák és csak korlátozott térben alkalmazhatók. A TMIT-en fejlesztett WLANpos próbál az igényeknek minél inkább megfelelő megoldást kínálni.

A megfelelő pozícionáló megoldás megtalálásához több helymeghatározási módszert is ki kellett próbálni, megvalósítani majd értékelni. A WLANpos alkalmazás egy teljes értékű alkalmazás, a pozícionáláson kívül képes térképek megjelenítésére, térképrajzolására, mérési adatbázis létrehozására.

2. Helyfüggő szolgáltatások

A mobil életmód kapcsán különösen fontos jelentőséggel bír aktuális tartózkodási helyünk az információszerzés és továbbítás szempontjából. Este az ország másik végében az úton haladva fogytán a benzinünk, de nem tudjuk, hogy hol találjuk a közelben a megszokott benzinkút-hálózatunkat.

A kérdés megválaszolásához a megoldást a helyfüggő szolgáltatások nyújtják, melyeket a következő főbb csoportokra oszthatunk fel [1]:

- *helyfüggő információk:* időjárás-előrejelzési adatok, közlekedési, forgalmi információk, szolgáltatási információk a közelben található üzletekről, bankokról, éttermekről, szálláshelyekről;
- *helyfüggő számlázás:* hívás-kezdeményezés és -fogadás helye alapján különböző díjzónák;
- *segélynyújtó szolgáltatások:* aktív és a passzív segélyhívások pontos helyéről információ eljuttatása a hívást fogadó szolgálat számára;
- *követés:* gyermekek, idős személyek követése biztonsági szempontból, állatok vagy járművek követése, például lopás esetén;
- *helyfüggő marketing és kereskedelem:* áruházak, kereskedők a vonzáskörzetükbe érkező mobil eszközök kijelzőjén hirdetéseket, reklámokat jelenítenek meg;
- *játékok, szabadidős alkalmazások:* helyfüggő stratégiai játékok;
- *barátkeresés:* olyan ismerősök keresése, akik a közelben tartózkodnak – Japánban már létező, ismert szolgáltatás.

3. A WLAN-okról

A WLAN-ok éppen olyan hálózatok, mint a LAN-ok (Local Area Network), csak az eszközök közötti átviteli közeg nem vezetékes. Általában ezek a hálózatok egy nagyobb LAN részét alkotják.

Ha létezik központi elem, legalább egy elérési pont (AP, Access Point), melyhez a csomópontok kapcsolódnak, akkor infrastruktúra-alapú (Infrastructure-based) hálózatról beszélünk. Az AP-khoz kapcsolódnak a mobil

kliensek (MC, Mobil Client) tipikusan egy laptop vagy egy PDA (Personal Digital Assistant) WLAN kártyával. Ahhoz, hogy a WLAN hálózat használható legyen, AP-kat kell elhelyezni az épület különböző pontjain. Az AP-k többnyire Ethernet kábelekkel vannak összekapcsolva egymással, a helyi LAN hálózattal, egyéb hálózati eszközökkel és az Internettel.

4. Helymeghatározás WLAN-nal

Számos különböző megközelítése van a mobil eszközök helymeghatározásának. Három nagy csoportba sorolhatjuk a megoldásokat: távolságvizsgálat, hely- (helyszín-) analízis és háromszögelés.

A távolságvizsgálat módszere úgy működik, hogy mérjük a jel erősségét az adó és a vevő helyén. A kettő között lévő különbségből meghatározható a távolság, amit a hullám a levegőben megtett. Hasonló elven működő, WLAN-nal történő helymeghatározás esetén így megkaphatjuk az MC AP-hoz viszonyított relatív helyét.

A hely-analízis során egy külső pontból mérjük, figyeljük az eszköz pozícióját. A Microsoft által fejlesztett RADAR [2] rendszer méri az eszköz jelerősségét egy adott pontból és ez alapján helyezi el az MC-t egy épülethez viszonyított koordinátarendszerben.

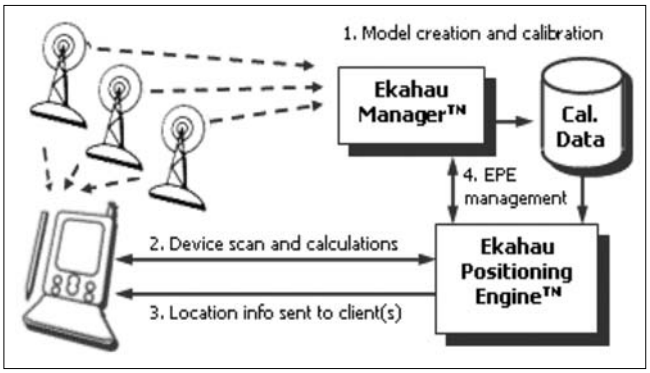
A háromszögelési módszerek két részre oszthatók. Az iránymérésen alapuló technikák az eszköz adott pontoktól való iránya alapján számolnak, míg a távolságmérésen alapulóak a tárgy távolságát határozzák meg rögzített pontoktól, ez alapján számítanak. Az irányméréshez irányított antennákra van szükség, szabványos Wi-Fi eszközök használata esetén ezért ez a módszer nem jöhet szóba. A távolságmérésen alapuló megoldások több meghatározott helyen lévő adó és a vevő közti távolságok mérésével számolják az eszköz helyét. A helymeghatározás a jel erősségének, fázisának vagy késleltetésének mérésével történhet. Rádióhullámok terjedési sebessége ismeretében, ha pontosan le tudjuk mérni az időt, amíg a jelek az adótól a vevőbe elértek (Time Of Arrival, TOA) akkor megkaphatunk az adó körül egy *r* sugarú kört (vagy gömböt) és ezen a körön helyezkedik el az eszköz. Több ilyen kör (vagy adott esetben gömb) metszéspontja jelöli ki a pozíciót.

5. Létező megoldások Wi-Fi alapú pozícionálásra

5.1. Ekahau

Következőkben néhány kereskedelmi, WLAN hálózatot használó helymeghatározó rendszer működését tekintjük át. Az első a finn Ekahau cég által fejlesztett rendszer, amely több WLAN szabvánnyal (pl. 802.11, HIPERLAN) is képes együttműködni. A rendszert arra tervezték, hogy a GPS szolgáltatásait nyújtsa épületeken belül is. A rendszer a pozícionáláshoz a szabványos Wi-Fi hálózattal együttműködni képes, opcionális T301-es TAG-eket is használhat. Ezek egyszerű eszközök, amelyeket a pozícionálandó tárgyra, eszközre, személyre lehet rögzíteni. A rendszer három alkomponensre osztható: Ekahau Kliens, Ekahau Pozícionáló, Ekahau Manager, melyek kapcsolatát az *1. ábrán* láthatjuk..

1. ábra Az Ekahau architektúrája



5.2. LOCUS

A Worcester-i Politechnikumban készült LOCUS [3] szintén a mért jelerősség alapján végzi a pozícionálást. Itt létezik egy kalibrációs fázis. A programhoz tartozó grafikus felületen a kalibrációt elindítva megadhatjuk az épületet és a szintet, ahol méréseket szeretnénk végezni. A megadott pontokban mind a négy irányban el kell végeznünk a mérést, majd a program átlagolja ezeket és a pont koordinátáival együtt eltárolja azt egy adatbázisban.

A helymeghatározó algoritmus két fázist különböztet meg. Az első fázisban a rendszer azt a térképet (szintet) határozza meg, ahol a felhasználó tartózkodhat. A második fázisban a térkép meghatározása után a felhasználó pontos helyét határozza meg.

Az első fázis alatt a mért adathalmazból kiválasztunk három AP-t. Az adatbázisból kiszűrjük azon pontok halmazát, melyekhez tartozó jelerősség értékek között mindhárom AP-hoz tartozó érték szerepel. Az így kapott adatbázisból a mért értékektől a megadott határértéknél kisebb mértékben eltérő pontokat választja ki. Ezután az algoritmust addig ismételjük, míg csak egy pont marad. Így megkapjuk, melyik térképen lehetünk. A második fázisban a kiválasztott térképen felvett pontok között keresünk ugyanezzel a rekurzív módszerrel.

5.3. Locadio

A Microsoft fejlesztése a Locadio nevű rendszer [4]. Ez a mi céljainkhoz hasonlóan csak a meglévő 802.11 infrastruktúrát használja fel a helymeghatározáshoz. A rendszer a pozíción kívül képes megállapítani, hogy a felhasználó éppen egyhelyben áll, vagy mozog. Hogy mozog-e a felhasználó, azt egy kétállapotú Markov-modell alapján döntik el, a pozíciót pedig egy másik Markov-modell segítségével határozzák meg, ami az AP-k jelerősségeinek a helyváltoztatással történő változásait használja ki.

A Locadio csak a kliens oldalon használ erőforrásokat. Nincsenek központi szerverek, sem az AP-kon futó

alkalmazás, ezáltal növelve a rendszer biztonságát. Az általunk fejlesztett rendszer is ezen alapelvet alkalmaztuk.

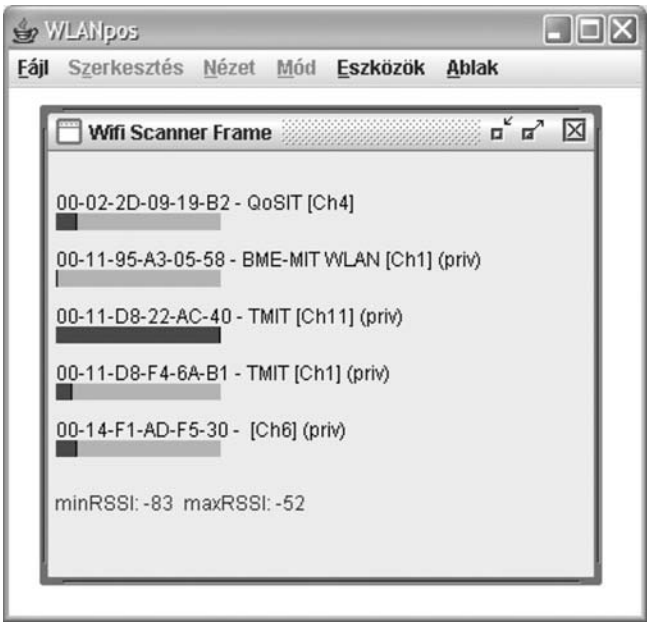
A Microsoft kutatói arra a megállapításra jutottak, hogy ha mozog a Wi-Fi vevő, az AP-k mérhető jelerőssége nagyobb mértékben ingadozik, mintha egyhelyben állna. Ennek alapján pozícionáláskor egy algoritmus végzi a döntést a felhasználó mozgásállapotáról: ha a jel nagyobb amplitúdóval, de alacsonyabb frekvenciával változott, akkor az azt jelenti, hogy a vevő áll, míg ha kisebb amplitúdóval, de gyorsan ingadozott, akkor mozog.

A felhasználó pozíciójának és mozgásának modellezéséhez rejtett Markov-modellt használtak. Ez egy olyan Markov-lánc, amelynél az aktuális állapotot nem ismerjük, csak az átmenetvalószínűségeket és bizonyos paramétereket, amelyekből egy algoritmus segítségével a Markov-lánc állapotát megbecsülhetjük. A Locadio rendszer empirikus módon működik, a pozícionáláshoz egy előzetes adatgyűjtés szükséges.

A pozícionálás működéséhez szükség van egy gráfra, ami az épület topológiájához igazodik. A gráf csomópontjai azok a pontok, ahol méréseket végeztünk, az élek pedig az egyes helyiségek közötti átmeneteket jelentik. Ez a gráf a Markov-lánc állapotgráfja is egyben. Az élek elhelyezkedése a gráfban nyilvánvalóan attól is függ, hogy milyen sűrűn vesszük fel a pontokat a térképen, ez viszont a pozícionálás pontosságát is befolyásolja.

A Markov-lánc állapotátmenet-valószínűségei a mintavételi frekvenciától (vagyis a Wi-Fi jelerősségek olvasásának gyakoriságától), a felhasználó mozgásától, a mérési pontok fizikai távolságától és az épület topológiájától függenek. (Nyilvánvaló például, hogy kicsi a valószínűsége annak, hogy rövid idő alatt a felhasználó olyan pontok között mozog, amelyek fizikailag, vagy az épület topológiája miatt távol vannak egymástól, egymáshoz közeli pontok között viszont nagyobb a mozgás valószínűsége.)

2. ábra A WifiScanner programrész



6. A WLANpos

A munka során egy pozícionálásra alkalmas, bárki által könnyen használatba vehető szoftver fejlesztése volt a célunk, jelen cikk célja azonban elsősorban magának a pozícionáló motornak és algoritmusnak az ismertetése. Mindazonáltal fontos említést tenni a szoftver egyéb funkcionalitásairól is, amelyek lehetővé tették a pozícionáló algoritmus implementálását, pontosságának vizsgálatát. A WLANpos szoftver a következő főbb funkciókkal rendelkezik:

– térképrajzolás,
– felmérés,
– pozícionálás,
– hibamérés,
– Wi-Fi AP-k jeleinek figyelése *(2. ábra)*.

7. A pozícionálás működése a WLANpos-ban

A pozícionáló algoritmus kifejlesztése során fokozatosan egyre bonyolultabb megoldások kidolgozásában gondolkodtunk, azt remélve, hogy egy összetettebb algoritmus nagyobb pontosságot eredményez majd. A következőben két különböző algoritmus ismertetése következik. A rendszer megalkotása során a legelső általunk kifejlesztett algoritmus egy könnyen érthető és implementálható, NNSS-en (Nearest Neighbour in Signal Space – legközelebbi szomszéd a jelerősség-térben) alapuló megoldás volt, amely a mérési eredményeink szerint kis pontosságot biztosított, de a fejlesztés első lépésében megfelelőnek bizonyult. A második, részletesen ismertetett algoritmus egy jelerősségi valószínűség-eloszlást használó megoldás, amely már pontosabbnak bizonyult, de egyúttal a számítási igényei is nagyobbak. A két megoldás pozícionálási pontosságának vizsgálatával elért eredményeket a következő szakasz tartalmazza.

Az első algoritmusunk lényege, hogy az AP-k jelerősségeit úgy képzeljük el, mint egy vektor komponenseit, vagy másképpen fogalmazva; pont koordinátákat egy térben. Ha rendelkezésre áll *n* darab AP, akkor egy adott pozícióban mért jelerősség-adatok kijelölnek egy pontot egy *n* dimenziós térben. Az adatfelvétel úgy történik, hogy a programmal a felhasználó bizonyos pontokon fix számú mérést végez. A mérés eredményeként létrejött naplófájlban a mért jelerősség-értékek mediánja szerepel minden egyes AP-ra, mérési pontonként külön-külön. (Használhatnánk átlagértékeket is, a medián azért előnyösebb, mert az AP pillanatnyi kiesését, illetve a jelerősség-értékek rövid időtartamú ugrálását könnyebben kiküszöböli.) Pozícionáláskor sem az éppen az adott pillanatban mért értékeket használjuk, hanem az elmúlt néhány, például tíz mérés eredményének a mediánját számoljuk ki minden egyes AP-re.

Pozícionáláskor a mért és a naplófájlban eltárolt jelerősségvektorokat hasonlítjuk össze egymással és amelyik eltárolt vektorhoz a legközelebb vannak az éppen mért értékek, arra a pontra pozícionálunk. A távolság megállapításakor euklideszi távolságot számítunk.

A megvalósítás során azonban egyéb problémák is jelentkeznek. A különböző típusú Wi-Fi kártyák a jelerősség-értékeket nem azonos skálázás szerint adják vissza. Emiatt egy adott kártyával készült mérési logfile nem használható egy olyan kártyával történő pozíconáláskor, amely teljesen más értékeket ad vissza. A problémára egy lehetséges megoldás, hogy különböző kártyákhoz külön logfile-okat készítünk, és pozícionáláskor az adott típusú kártyának megfelelő logfile-t használjuk. Egy másik lehetséges megoldás, hogy már adatfelvételkor százalékos értékekre alakítjuk a jelerősségeket. Ez utóbbit használtuk az első pozícionáló algoritmusnál.

Algoritmusunk valójában nem is a tárolt és a mért értékek közti különbség alapján számolja az euklideszi távolságot. Veszi az *n*-edik és az (*n*+1)-edik AP mért értékének hányadosát, majd ebből kivonja ugyanezen két AP-hoz tartozó tárolt érték hányadosát.

A módszerrel kivédhetjük azokat a hibákat, mikor minden AP jele ugyanolyan mértékben csökken valamilyen külső hatásra. A módszer azonban jelentősen befolyásolja az egyes AP-k jelének súlyát az euklideszi távolság kiszámolásánál. Ezért a következő módszert használja az algoritmus, amit korrekciós módszernek neveztünk:

$$D_n = \left(\frac{M_n}{M_{n+1}} - \frac{T_n}{T_{n+1}} \right) \cdot \frac{T_{n+1}}{T_n}$$

$$D = \sqrt{\sum_{n=0}^{m} D_n^2}$$

ahol:

- D_n a távolság n-edik összetevője,
- D az euklideszi távolság,
- M_n az n-edik AP mért jelerőssége,
- T_n az n-edik AP adatbázisban tárolt jelerőssége az adott pontban,
- m azon AP-k száma, melyeknek jelerőssége nagyobb egy definiált értéknél, de minimum 3.

A második pozícionáló algoritmusunk működéséhez meg kellett változtatnunk az adatfelvételi módszert. Ennél az algoritmusnál az adatfelvételi fázisban eltároljuk az adott pont koordinátáit, a ponton mérhető AP-k MAC (Medium Access Control) címeit és az AP-k mérhető jelerősség-értékeiből készített statisztikát. Ez a statisztika gyakorlatilag egy hisztogram, ami azt adja meg, hogy az adott ponton történt mérés során az adott AP a mérések hány százalékában milyen jelerősséggel látszott. Egy ilyen mérés eredménye látható a *3. ábrán*. A program tehát valójában kiszámolja annak a valószínűségnek a becslését, hogy az adott AP adott RSSI-vel látszik. A mérés eredményéről készült logfile elején külön is eltároljuk, hogy a mérés során milyen AP-k látszottak.

Mivel az emberi test a 2,4 GHz környékén lévő frekvenciájú rádióhullámok nagy részét elnyeli, ezért az adatrögzítés során az eszköz iránya befolyásolja a mérési eredményeket, tehát ugyanazon a helyen mérhetünk különböző RSSI értékeket attól függően, hogy melyik égtáj felé nézünk. Mivel az adatrögzítés során az eszközt mindig egy ember tartja a kezében, elkerülhetetlen, hogy ne befolyásolja az eredményeket ez a jelenség, ezért a WLANpos négy különböző irányból gyűjti az AP-k RSSI értékeit, így csökkentve az emberi test hatásait.

Amikor az új WLANpos-ban a pozícionálásra kattintunk, egy összetett algoritmus kezdődik. Ez elsőként lekérdezi a WLAN interfésztől az aktuális RSSI értékeket, majd tíz ilyen lekérdezés után kiszámolja minden mért AP-ra az RSSI értékek mediánját. Ezek után egy olyan algoritmus végrehajtása kezdődik, ami egy megadott logfile-ból és az aktuális pozícióban mért értékekből (medián) kiszámolja a logfile összes pontjára annak a valószínűségét, hogy abban a pontban tartózkodunk.

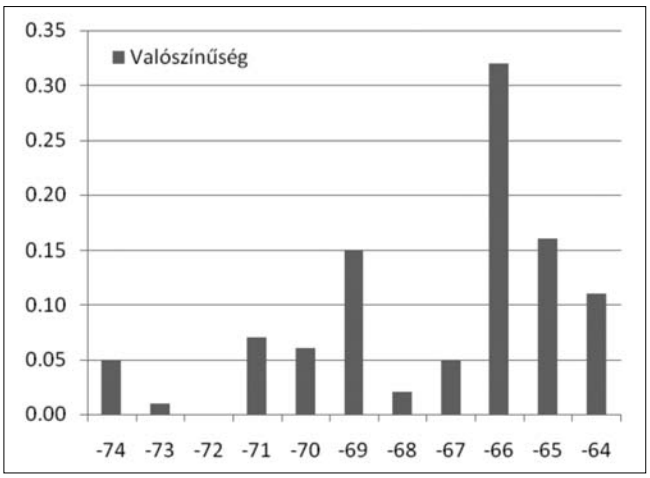
Első lépésként beolvassa a naplófájl tartalmát, aminek az elején felsorolt AP-k egy listába kerülnek. Minden listaelemhez egy RSSI érték tartozik (ez kezdetben egy „NOSIGNAL” érték). Ha egy AP-t mérni tudtunk, beírjuk a listába a mért értéket. Így kaptunk egy listát, melyben a térképünkhöz tartozó logfile összes AP-ja benne van, kivéve azokat, amelyeket csak pozícionáláskor látunk, de adatfelvételkor nem.

Nincs más hátra, mint a listánkban szereplő értékek és a hisztogrammok összehasonlításával kiszámolni a pontok valószínűségét *(4. ábra)*. Jelölje a pozícionáláshoz használt térkép pontjait $X_1, X_2,...X_m$. Az előző bekezdésekben bemutatott AP-RSSI listát jelölje S, ahol $S_1, S_2,...S_m$ az egyes AP-khoz tartozó RSSI értékek. Legyen *I* annak az igaz/hamis értékekből álló listának a neve, ami minden AP esetén megmondja, hogy az mérhető volt-e pozícionáláskor. $P(AP_j \mid X_i)$ annak valószínűsége, hogy a *j.* AP mérhető az X_i pontban. $P(S_j \mid AP_j, X_i)$ pedig annak a valószínűsége, hogy a *j.* AP-t pontosan S_j térerősséggel mértük az X_i pontban.

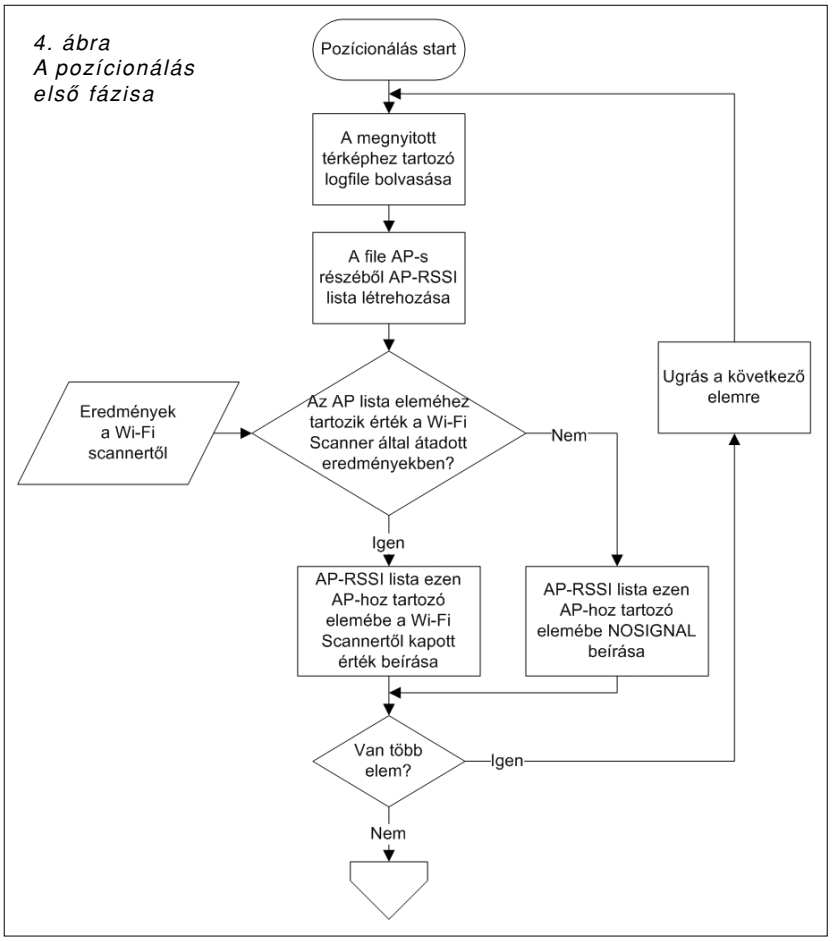
Legyen kezdetben $P(I,S \mid X_i)=1$ minden pont esetén. Ez annak a valószínűsége, hogy az adott *S* illetve *I* listát az X_i pontban rögzítettük. Menjünk sorban végig az *S* illetve *I* listában található AP-kon a következő egyenlet alapján, így kaphatjuk meg a ponthoz tartozó igazi valószínűség értékét.

$$P(I,S \mid X_i) = \prod_{j=1}^{n} \begin{cases} P(AP_j \mid X_i) * P(S_j \mid AP_j, X_i) & \text{Ha } I_j = \text{igaz} \\ 1 - P(AP_j \mid X_i) & \text{Ha } I_j = \text{hamis} \end{cases}$$

3. ábra Jelerősség-értékek hisztogramja



4. ábra
A pozícionálás első fázisa



Bayes-tétele kimondja, hogy:

$$P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B)},$$

vagy más alakban:

$$P(A_i \mid B) = \frac{P(B \mid A_i) * P(A_i)}{\sum_j P(B \mid A_j) * P(A_j)}.$$

Ezért a mi esetünkben:

$$P(X_i \mid I,S) = \frac{P(I,S \mid X_i) * P(X_i)}{\sum_{i=1}^{n} [P(I,S \mid X_i) * P(X_i)]}.$$

A program feltételezi, hogy kezdetben minden pontban ugyanakkora eséllyel tartózkodhatunk ($P(X_i) = 1/n, i = 1,...n$).

Mivel $P(X_i)$ mindig ugyanakkora, ezért egyszerűsödik a számolás és mivel a nevező minden pontra ugyanakkora, így a $P(X_i \mid I,S)$ valószínűségek nagyság szerinti sorrendjét csak a $P(I,S \mid X_i)$ valószínűségek nagyság szerinti sorrendje határozza meg.

Tehát az a pont, amihez a legnagyobb $P(I,S \mid X_i)$ feltételes valószínűség tartozik, ahhoz tartozik a legnagyobb $P(I,S \mid X_i)$ is. Ez az X_i pont lesz a pozícionálásunk eredménye.

Tehát ha mérni tudtuk az AP-t a pozícionálás pillanatában, ellenben sosem láttuk adatrögzítés során, akkor $P(I,S \mid X_i) = 0$. Amennyiben az AP-t legalább egyszer láttuk adatfelvételkor, akkor a $P(I,S \mid X_i)$-t adó szorzat *j*-edik eleme egyenlő annak a valószínűségével, hogy láttuk az AP-t a logolásnál, szorozva azzal a feltételes valószínűséggel, hogy pontosan S_j térerősséggel láttuk az AP-t a logolásnál. Amennyiben nem volt mérhető az AP a helymeghatározáskor, akkor a szorzat *j*-edik eleme annak a valószínűsége, hogy nem láttuk az AP-t a logolásnál.

Ha minden AP-n végigértünk az AP-RSSI listában, folytathatjuk a számításokat a következő ponttal. A folyamat ezen részét az *5. ábra* mutatja be. A kapott eredményeket egy újabb listába mentjük el, ami minden pont adatait és a pontokhoz tartozó valószínűséget tartalmazza.

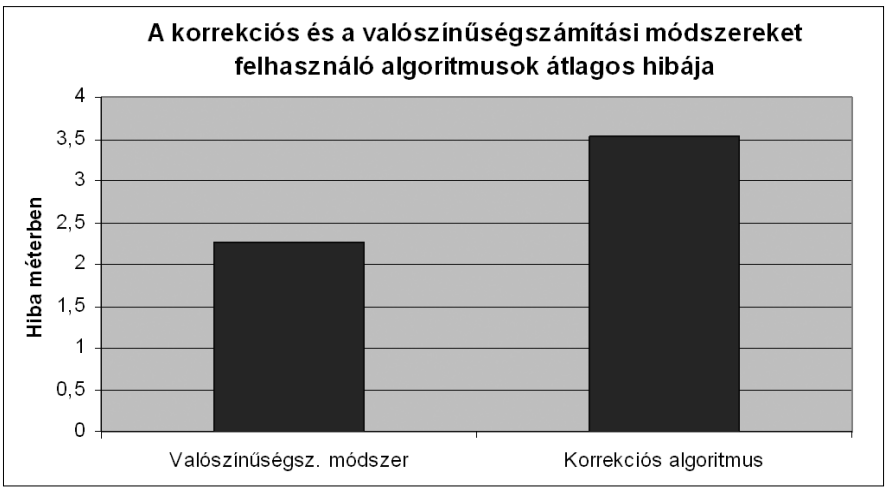
A $P(I,S \mid X_i)$ értékek tehát megmondják, mi annak a valószínűsége, hogy az adott *S* illetve *I* listát az X_i pontban rögzítettük. Mi viszont azt szeretnénk megtudni, hogy mi a valószínűsége annak, hogy X_i pontban vagyunk, ha *S*-t (és *I*-t) mértük ($P(X_i \mid I,S)$). Ahol a legnagyobb ez a valószínűség, legvalószínűbben ott tartózkodunk a pozícionálás pillanatában.

8. Eredmények

A programot mindkét eljárással húsz pozícionálást végezve teszteltük. Az átlagos hiba mértékének összehasonlítása a két algoritmus használatával a *6. ábrán* látható.

A grafikonból kitűnik, hogy a pozícionálást valószínűségértékek alapján végző algoritmus a korrekciós algoritmusnál pontosabb. A korrekciós 3,53 métert, míg a valószínűségekkel dolgozó módszer 2,27 métert hibázott átlagosan.

6. ábra
Pozícionáló algoritmusok pontosságának összehasonlítása

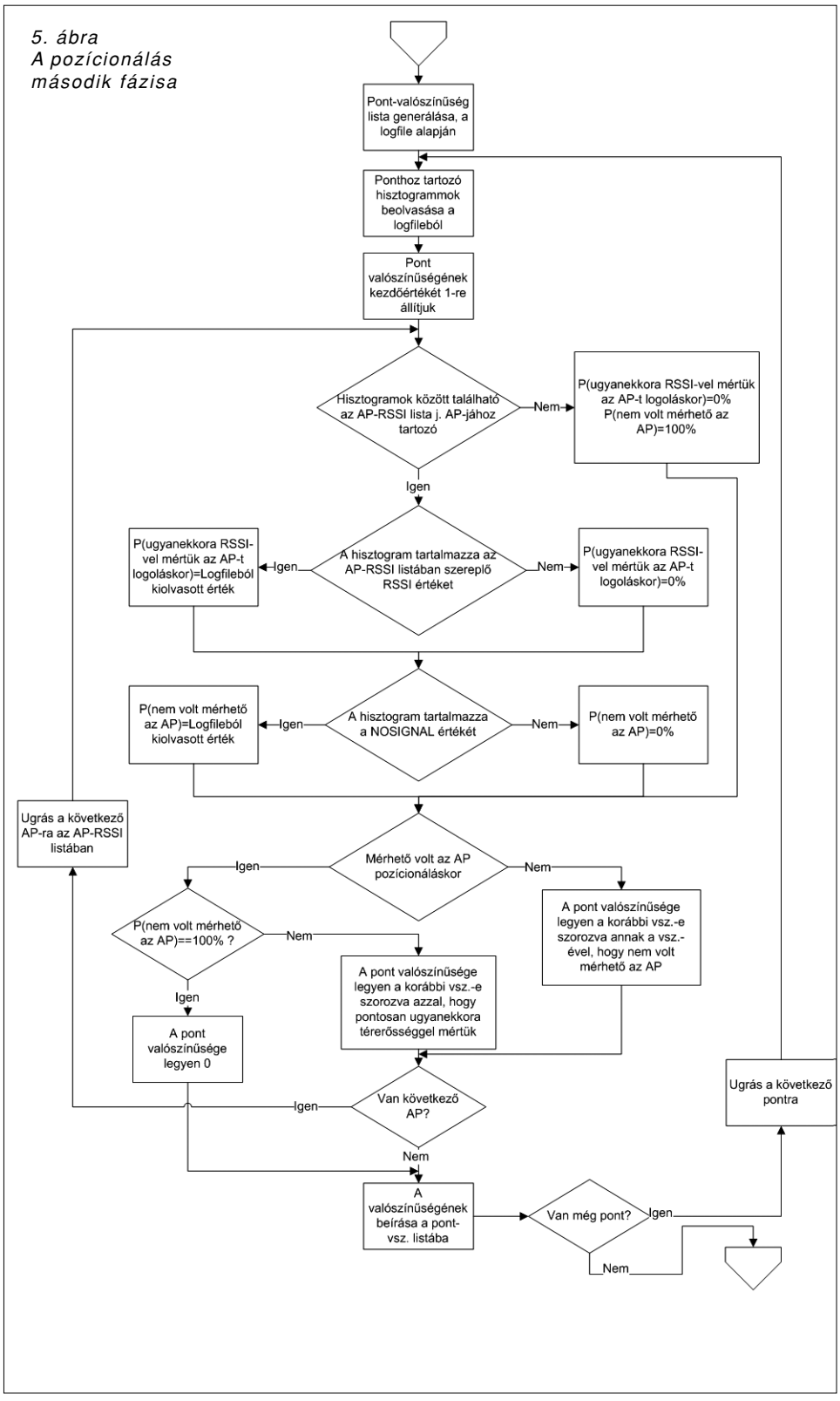


Ugyan a különbség nem túl jelentős, de egy ilyen, viszonylag kis mértékű hibánál már ez is jó eredménynek számít.

A tesztelések alatt már tapasztalhattuk, hogy a kidolgozott rendszer működőképes. A pozícionálási hibát a munka során sikerült leszorítani 2-4 méter közöttire az alkalmazott adatrögzítési és pozícionálási módszer módosításával.

A WLANpos még közel sincs kész. A pontosság növelhető, de szem előtt kell tartani, hogy a kapott rendszer ne legyen túl nagy erőforrás-igényű, hiszen főleg olyan eszközökön használják majd, ahol a merevlemez vagy a memória, illetve a processzor kapacitása nagyon korlátozott lehet. Összességében egy ígéretes szoftver-prototípust sikerült létrehoznunk, így mindenképpen érdemes energiát fektetni a későbbi fejlesztésbe.

5. ábra
A pozícionálás második fázisa



Irodalom

[1] Geox Térinformatikai Kft, „Helyfüggő szolgáltatások”, http://www.geox.hu/showPage.asp?CLICKMENU=3_5_1.

[2] Paramvir Bahl, Venkata N. Padmanabhan, Microsoft Research, “RADAR: An In-Building RF-based User Location and Tracking System”, 1999.

[3] Arvinder Singh, “LOCUS: Wireless LAN Location Sensing”, Major Qualifying Project Rep. Worcester Polytechnic Inst., 19 January 2004.

[4] John Krumm, Eric Horovitz, Microsoft Research, “Locadio: Inferring Motion and Location from Wi-Fi Signal Strengths”, In Proc. of the 1st Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services, Boston, 2004.

H.264 kódolt videófolyamok vízjelezése

OLÁH ISTVÁN

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

Lektorált

Kulcsszavak: videó vízjelezés, H.264 vízjelezés, NCG

Cikkünkben összefoglaljuk a videó vízjelezés sajátosságait, majd bemutatunk egy olyan videó vízjelezési eljárást, ami ellenáll a H.264/AVC tömörítésnek, és a legáltalánosabb jelfeldolgozási módosításoknak.

1. Bevezetés

A digitális vízjelezés az adatrejtés (szteganográfia) egyik változata, segítségével információt rejthetünk valamilyen digitális hordozómédiába, úgy, hogy az egy laikus szemlélő számára észrevétlen marad. Legtöbbször a vízjel elhelyezése egy titkos kulcs segítségével történik, hogy csak az tudja detektálni a vízjelet, akinek birtokában van a megfelelő kulcs.

A legáltalánosabb esetben a vízjelezés első lépése a vízjel létrehozása, majd ezt a vízjelet valamilyen algoritmus segítségével elhelyezzük a hordozó médiában, és így létrejön a vízjelezett tartalom. A detektálás során feltesszük, hogy a hordozómédia időközben valamilyen módosításon esett át. Ilyen módosítás lehet a tömörítés, zajszűrés, vagy bármilyen más változtatás a tartalomban. A detektálás során a torzított vízjelezett tartalomban próbáljuk meg észlelni a vízjelet.

2. A videó vízjelezés sajátosságai

A mozgóképek vízjelezése nagyon hasonló az állóképek vízjelezéséhez, azonban van néhány speciális kérdés, amit meg kell említeni.

Az első ilyen különbség, hogy számos olyan szándékos módosítással kell számolni, amikor a tartalom terjesztője módosítja a tartalmat, hogy az megfeleljen a továbbítási csatorna sajátosságainak. Ilyen módosítás lehet például a bitráta megváltoztatása. Ezért egy jó vízjelnek jól kell tűrnie ezeket a változtatásokat.

Másik sajátossága a mozgókép-vízjelezésnek, hogy a detektálás során nem csak képkockán belüli szinkronizáció szükséges (hogy azon a helyen keresse a detektáló algoritmus a vízjelet, ahová azt elhelyezték), hanem időbeli is. Ilyenkor a vízjelet a meghatározott képkockán keresi a detektor. Időbeli szinkron elvesztését idézheti például elő, ha kihagynak néhány képkockát.

Harmadik sajátosság, hogy az egymás utáni képkockák csak egy kicsit különböznek egymástól, ami bizonyos alkalmazásoknál támadási lehetőséget kínálhat azoknak, akik el akarják tüntetni a vízjelet. Gondoljunk arra, hogy ha sok hasonló képkockát helyettesítünk az átlagukkal, akkor a néző számára a tartalom csak egy kicsit változik, viszont az átlagolás miatt esetleg eltűnik a vízjel.

Videó vízjelezésénél szükség lehet arra, hogy az algoritmus képes legyen másodpercenként 25-30 képkockát is vízjelezni. Ezért fontos az, hogy a vízjelezés amennyire csak lehet, egyszerű legyen.

Végül szempont lehet az is, hogy a beültetett vízjel ne változtassa meg a videófolyam bitsebességét és így ne következzen be esetleges torlódás a folyam hálózati továbbítása során.

3. Videóknál használt vízjelek fontos tulajdonságai

A vízjelező rendszereket sok tulajdonsággal lehet jellemezni. Ilyen tulajdonságok lehetnek a beültetés hatékonysága, a képromlás foka, a szállított adat mennyisége, a detektálás módja, a hibás detektálások aránya, a vízjel robusztussága, a biztonság tulajdonságai, a kulcsok milyensége, az ellenállás foka különböző támadások esetén, skálázhatóság és még számos egyéb tulajdonság is. Az, hogy e tulajdonságok közül melyiket tekintjük fontosnak és melyiket elhanyagolhatónak, nagyban függ attól, hogy mire akarjuk használni a vízjelet.

Az egyik legfontosabb tulajdonság, hogy mennyire rontja el a beültetett vízjel a hordozó képet. Ideális esetben a vízjel láthatatlan marad az emberi szem számára.

Egy másik fontos tulajdonság a robusztusság. Annál robusztusabb a vízjel, minél jobban ellenáll a tartalommódosításokkal szemben, legyenek azok egyszerű átkódolások, vagy rosszindulatú támadások, amik arra irányulnak, hogy eltávolítsák a vízjelet. Ezért célszerű lehet olyan részletgazdag helyekre elhelyezni vízjelet, melyek módosítása már észrevehető romlást eredményez a tartalomban.

Harmadik fontos tulajdonsága a videó vízjelezési algoritmusoknak a vak detektálás, ami azt jelenti, hogy a vízjelet az eredeti, vízjelezetlen tartalom nélkül is lehet detektálni. Ez a tulajdonság azért fontos, mert a tartalom nagyméretű és nehéz lehet eljuttatni a detektálás helyére.

A kapacitás azt jelenti, hogy mennyi adatot tud a vízjel szállítani egy képkockában. Ez lehet akár csak egy bitnyi adat, vagy akár egy egész szövegrész vagy kép is, a vízjel alkalmazási területétől függően. A vízjel kapacitása összefüggésben van annak láthatóságával és robusztusságával, ugyanis minél több információt tartalmaz a vízjel, annál jobban láthatóak a változások, vagy annál kevésbé lesz robusztus.

4. Videó vízjelezés alkalmazásai

A következőkben a digitális videó vízjelezés néhány alkalmazási módját tekintjük át.

Az első felhasználási mód a *másolásvezérlés*, amin azt értjük, hogy a médiában elhelyezett vízjel-bitek jelzik a lejátszó készülék számára, hogy az adott tartalom lejátszható vagy másolható-e.

A második elterjedt felhasználási mód a tévécsatornákon sugárzott *reklámok automatizált figyelése*. Ilyenkor a reklám tartalmaz egy vízjelet, amelynek detektálásával megállapítható, hogy mikor és hányszor sugározta a csatorna az adott reklámot.

A *nyomon követés (fingerprinting)* során egy adott tartalom (például egy film) minden példányát olyan vízjellel látják el, ami azonosítja azt a személyt, aki a filmet megvette, letöltötte. Ha a tartalom egy másolata felbukkan valamilyen illegális terjesztési hálózaton, akkor a vízjel segítségével azonosítható annak forrása.

A vízjelek használhatóak *hitelesítési feladatok* ellátására is. Ilyenkor egy olyan „törékeny” vízjelet helyeznek el a tartalomban, ami bizonyos fokú módosítások után nem detektálható többé. A tolerálható módosítás foka legtöbb esetben jól beállítható. Ezáltal megállapítható, hogy módosította-e valaki a vízjelezett tartalmat.

A *szerzői jogvédelmi alkalmazás* lényege, hogy a videóban elhelyezett vízjel olyan információt hordoz, ami azonosítja a tartalom tulajdonosát. Vitás esetekben így egyértelműen megállapítható, hogy ki a tartalom valódi tulajdonosa.

Az utolsó felhasználási mód a *tartalomhoz kapcsolódó információk szállítása*. Ilyenek lehetnek a szerzők neve, címek, feliratok és egyéb mellékinformációk.

5. A javasolt vízjelező rendszer

Az általunk tervezett vízjelező módszer célja, hogy H.264 kódolóval [1] tömörített videókat vízjelezhessünk. A vízjelezés és a tömörítés egymással ellentmondásban állnak, mert a vízjelezés során olyan apró változások keletkeznek a képen, amelyeket az emberi szem már nem vesz észre. A tömörítés célja pedig éppen az, hogy ezeket az észrevehetetlen változásokat eltüntesse és ezáltal a videó méretét csökkentse. A javasolt vízjelezési módszer felhasználási területe lehet valamilyen információszállítási feladat.

A vízjelező algoritmusok általában a tartalom kevésbé fontos részében rejtik el a vízjelet, így mérsékelve a minőség romlását. A jobb szubjektív minőség azonban a robusztusság csökkenését okozza: a különböző veszteséges tömörítési eljárások a kevésbé fontosnak tartott részeket durvábban kvantálják, így az ott elrejtett információ is nagyobb mértékben sérül.

Ezért, ahogyan az [2]-ben is láthatjuk, a vízjelet olyan helyre kell rejteni, ahol az emberi szem számára láthatatlan és mégis ellenáll a különböző kodekek általi veszteséges tömörítésnek. Ezek a helyek például a képkockán levő objektumok határai: ezeket a tömörítő algoritmusok finomabban kvantálják és kis mértékű változtatásuk nem lesz észrevehető az emberi szem számára.

A vízjelezés során megkeressük tehát azokat a blokkokat, amik alkalmasak vízjel rejtésére. Ezt a blokkok Normált Gravitációs Középpontjának (NCG) kiszámításával döntjük el. Az NCG értékek megadják, hogy melyek azok a blokk, amelyekben élek (éles átmenetek) találhatóak. Ha egy blokkhoz tartozó NCG érték egy előre meghatározott határ felett van, a blokkot vízjelezésre alkalmasnak választjuk. Bővebben a [2]-ben olvashatunk az NCG koordinátákról és azok tulajdonságairól.

Az algoritmus futása során minden egyes képkockában, a képkocka összes blokkjának kiszámítjuk az NCG értékét. Amely blokkok értéke egy előre meghatározott átlag felett van (tesztjeinkben ezt az értéket 420-nak választottuk – ennél nagyobb NCG értékű blokkokat már elegendően finoman kvantál a H.264 kódoló), alkalmasak vízjel beágyazására.

Az első képkockánál egy titkos kulcs alapján döntjük el, hogy a lehetséges blokkok közül melyek lesznek azok, amelyekben ténylegesen elhelyezzük a vízjelet. A további képkockákon megvizsgáljuk, hogy az előző képkockán használt blokkok megfelelnek-e az aktuális képkockán is. Ha igen, akkor ugyanazokat használjuk. Ez azért lehetséges, mert az egymást követő képkockák hasonlóak lehetnek egymáshoz, így az élek is ugyanott találhatóak. Így elkerülhetjük, hogy a képernyőn gyorsan ugráló mintablokkok látsszanak, ezáltal a vízjel kevésbé lesz zavaró.

Ha nem felelnek meg az előzőleg használt blokkok, akkor a bemenetként kapott kulcs segítségével a vízjelezésre alkalmas blokkok közül kiválasztjuk azokat, amikbe a tényleges rejtés fog történni.

Az egyes blokkok vízjelezésére a Dittmann-algoritmus [3] egy módosított változatát fogjuk használni. Dittmann-algoritmusa eredetileg MPEG videófájlok vízjelezésére készült, azonban alkalmas bármilyen kodekkel tömörített videófolyam vízjelezésére is. Az adatrejtést a képtartományban végzi, úgy, hogy közvetlenül a pixelek fényességértékeit módosítja. A rejtendő adatot 8x8 pixel méretű blokkokba ágyazza be, minden blokkba 1-1 bitet. Az általunk használt algoritmus 16x16-os blokkokat használ, a robosztusság növelése miatt.

A vízjel beágyazása mintablokkok segítségével történik. Ezek a mintablokkok 16x16 pixel méretűek, képzésük a következőképpen történik: egy 16x16-os blokkot feltöltünk véletlenszerűn -1 és 1 értékekkel. A blokkból eltávolítjuk a magas frekvenciákat, így egybefüggő 1 és -1 területek alakulnak ki a blokkon belül. Ezáltal a

blokk könnyebben azonosítható lesz. Az eredeti Dittmann-algoritmushoz képest változás, hogy csak olyan mintablokkok felelnek meg, ahol a sorokban és oszlopokban nagyjából egyenlő a -1 és 1 értékek száma. Erre a Dittmann- és az NCG-algoritmus ellentmondása miatt van szükség: az első azon alapul, hogy egy blokkon belül a változás nem jelentős, míg a másik eljárás az NCG értékek segítségével olyan blokkokat választ ki, amelyeken belül jelentős változás van a pixelek fényességértékében. Fontos tehát, hogy a mintablokk soraiban és oszlopaiban is nagyjából egyenlő legyen a -1 és 1 értékek száma.

Az így elkészült mintablokkokból többet is készítünk. A létrehozott mintablokkokat két csoportba osztjuk, a titkos kulcsnak megfelelően. Az első csoportban kerülnek azok a mintablokkok, amik a tényleges információt fogják hordozni, míg a második csoportba azok a mintablokkok kerülnek, amelyek célja a megtévesztés. Az alkalmazás függvényében egy képkockába mindkét csoportból választunk mintablokkot. Ha valaki szándékosan el akarja távolítani a vízjelet, akkor nem fogja tudni megkülönböztetni a ténylegesen információt hordozó és a megtévesztő blokkokat, ezért mindkettőt el kell távolítsa, ami már jelentős minőségromlást okozhat a videóban.

Az NCG koordináták alapján kiválasztott blokkokhoz az algoritmus a titkos kulcs alapján kiválaszt mintablokkokat mindkét csoportból. Beágyazáskor az algoritmus a mintablokkokat felskálázza a vízjelezés erősségével, majd az így kapott blokkokat az eredeti képkocka blokkjaihoz hozzáadja, vagy kivonja, attól függően, hogy a rejteni kívánt információ 1 vagy 0. Az információt a pixel fényesség értékébe rejtjük.

A tesztjeink során minden képkockába 4 bitnyi adatot helyeztünk el, négy 16x16-os blokkot felhasználva. Azért, hogy az esetlegesen kimaradt vagy kitörölt képkockák ne okozzanak gondot a detekciónál, a 4 bitnyi információt több egymást követő képkockában is elrejtettük, ezzel tovább növelve az algoritmus robusztusságát. A tesztjeink során 9 egymást követő képkockát használtunk. Így az algoritmus ellenállóvá tehetjük a képkocka elhagyások ellen, vagy az átlagolásos támadások ellen.

A detektálás folyamata

Detektáláskor a mintablokkokat keressük meg a vízjelezett képkockákon. A titkos kulcs alapján a detektor ugyanazokat a mintablokkokat állítja elő, mint a vízjel elhelyezés során. Minden képkockánál kiszámítjuk a blokkok NCG értékeit, majd ezek alapján kiválasztjuk az alkalmas blokkokat. Mivel itt már a módosított videót vizsgáljuk, itt nagyobb tűrést állítunk be az NCG értékek vizsgálatánál, mint a beültetés során.

A titkos kulcsból meghatározzuk, mely négy mintablokkot keressük az adott képkockában, majd az összes alkalmas blokkot megvizsgáljuk, tartalmazza-e a mintablokkok valamelyikét. Azt, hogy egy adott blokk milyen bitet tartalmaz (egyáltalán tartalmaz-e adatot) a vizsgált blokk és a hozzá tartozó mintablokk közti korreláció határozza meg: pozitív korreláció esetén a detektált bit 1, negatív korreláció esetén 0, korrelálatlanság esetén a vízjelezett blokk vagy nem tartalmaz adatot, vagy a használt mintablokk nem megfelelő.

Mivel több, egymást követő képkockát is használunk ugyanannak az információnak elrejtésére, ezért a ténylegesen kapott bitekről egyszerű többségi szavazással döntünk.

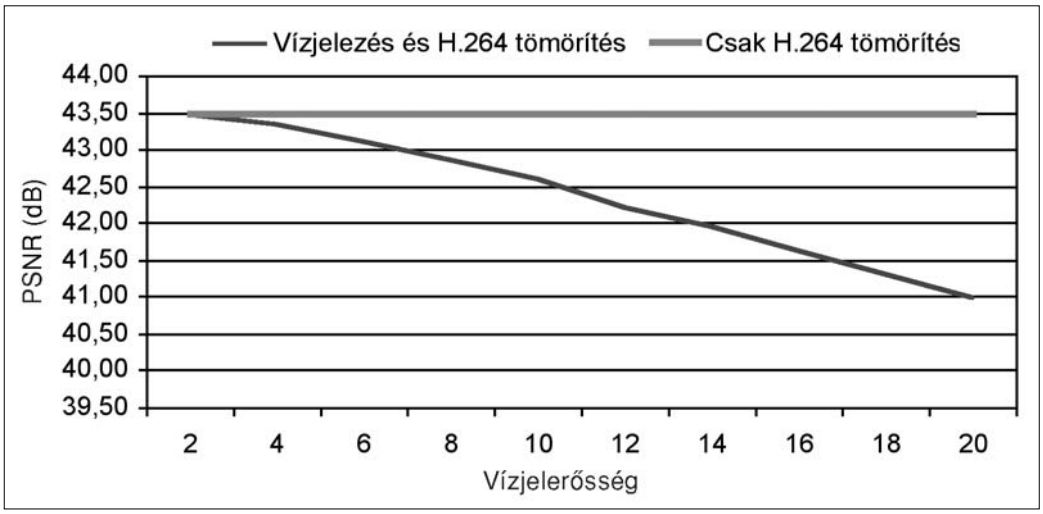
Az algoritmus a detektáláshoz nem használja az eredeti videó állományt, így vak vízjelezést valósít meg.

6. Teszteredmények

Az algoritmus működését 3 különböző videó fájlon teszteltük, melyek felbontása 720x576 pixel volt és egyenként 376 képkockából álltak. A vízjelezés erősségét 2 és 20 között állítottunk.

Az *1. ábra* bemutatja, hogy hogyan változik a videó minősége a vízjel erősségének a függvényében. Más szubjektív mérések azt az eredményt adták, hogy a 20-as erősség felé közeledve, a vízjelezett blokkok esetében már erős kockásodás figyelhető meg.

A *2. ábra* mutatja be, hogy a vízjel hogyan áll ellen a különböző módosításoknak. A tartalmakat vízjeleztük, majd a végrehajtottuk a módosítást, végül megpróbáltuk kinyerni a vízjelet a módosított tartalomból.



1. ábra
Jel-zaj viszony a vízjel erősségének függvényében

Az ábrákon látható, hogy H.264 szerinti tömörítés (1,5 Mbit/s értékkel), zajszűrés és zaj hozzáadása esetén a vízjel a beültetés erősségének a növelésével egyre jobban detektálható.

Az ábrán az is látható, hogy átméretezés, vágás és forgatás esetén a beültetés erősségének a növelése a detektort döntésképtelen állapotba hozza. Ennek az az oka, hogy a detektor elveszítette a szinkront a képkockákban található vízjellel és máshol kereste a vízjelet.

7. Összefoglalás

A bemutatott módszerrel elhelyezett vízjelek túlélik a H.264 tömörítést és azokat a támadásokat, melyek zaj hozzáadásán, vagy zajszűrésen alapulnak. A módszer fő hátránya, hogy nem ellenálló a szinkronizáció megzavarásán alapuló támadásokkal szemben.

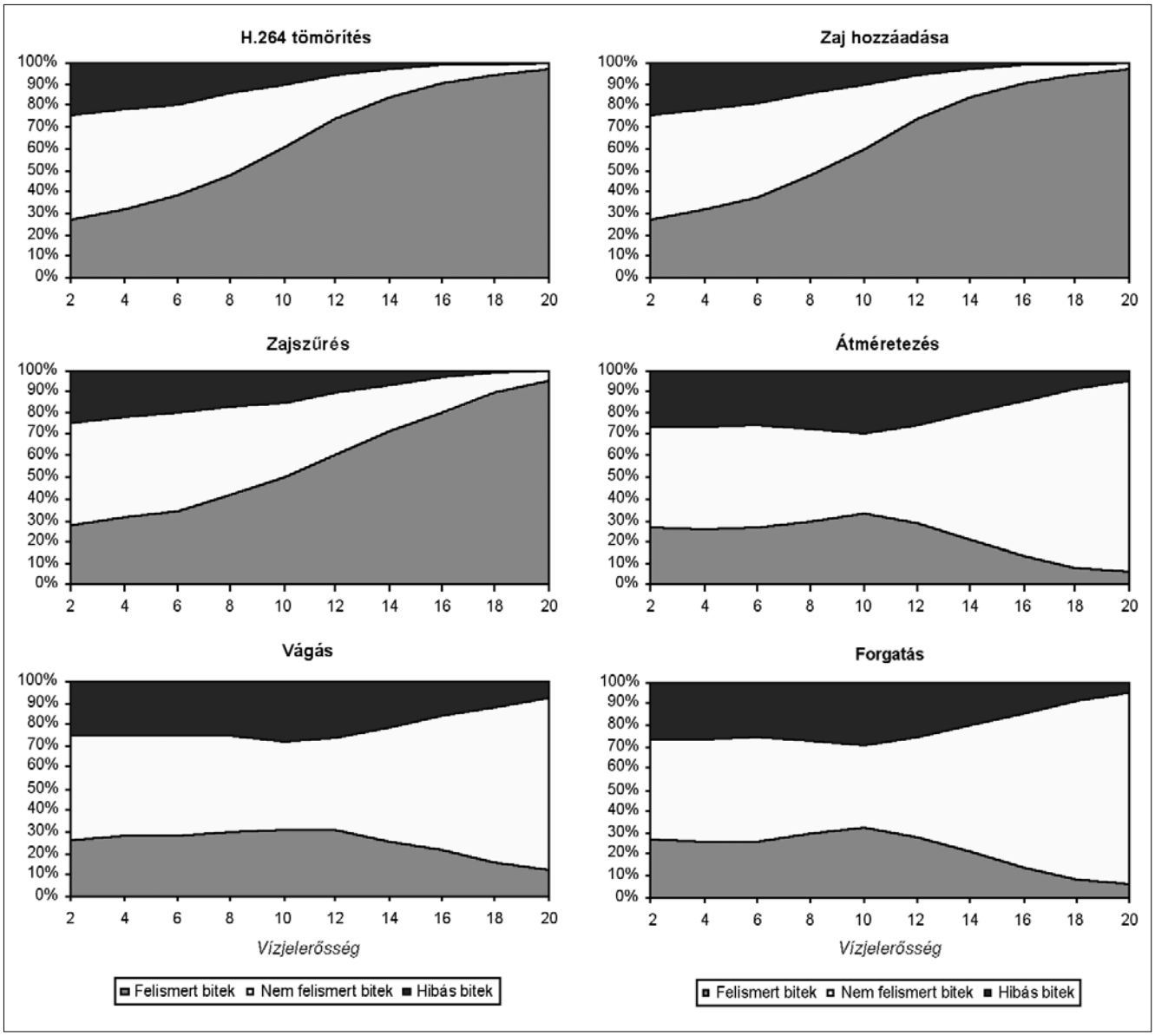
Irodalom

[1] ISO/IEC 14496-10 and ITU-T Rec. H.264, "Advanced video coding", 2003.

[2] D. Pröfrock, M. Schlauweg, E. Müller, "A New Uncompressed-Domain Video Watermarking Approach Robust to H.264/AVC Compression", Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA), Austria, 2006.

[3] Jana Dittmann, Mark Stabenau, Ralf Steinmetz, "Robust MPEG Video Watermarking Technologies", Proceedings of the ACM Multimedia, 1998. pp.71–80.

2. ábra A vízjel robusztussága különböző módosítások esetén



Ösztönző keretrendszer
önkéntes, autonóm együttműködéshez elosztott hálózatokban

NÉMETH LÁSZLÓ HARRI, SZABÓ RÓBERT

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
{nemethl, robert.szabo}@tmit.bme.hu

Lektorált

Kulcsszavak: ambiens hálózatok, önkéntes együttműködés, játékelmélet, peer-to-peer hálózatok, ígéretek elmélete

Napjaink kommunikációs hálózatai dinamikussá válnak, ami azt jelenti, hogy ezeknek a hálózatoknak nincsen kiépített fix infrastruktúrája (például WLAN-on keresztül megosztott hozzáférési hálózatok, ad-hoc hálózatok, ambient intelligencia [1,2] hálózatok vagy szenzorhálózatok) vagy az infrastruktúra-alapú hálózatok konfigurációja folyamatosan változik. Ezek a hálózatok nagymértékű önállósággal, autonómiával rendelkeznek és gyakran akár önző módon is viselkedhetnek. Az autonómia azt jelenti, hogy az ilyen hálózatoknak nincsen semmilyen központi adminisztrációs vagy menedzsment-alapelvük, amely meghatározná működésüket. A hálózat önző viselkedését sokféle módon szabályozhatjuk – ez egy széleskörben vizsgált terület a kutatók körében, különösen a peer-to-peer fájlmegosztó hálózatok népszerűvé válása óta. Az általunk vizsgált megoldás a hálózati topológia figyelembevételében különbözik az eddigiektől. Mivel minden csomópont csak a saját szomszédjaival képes közvetlenül kommunikálni, egy újfajta megoldást kell találni a csomópontok együttműködésre való motiválására. Jelen cikk egy újszerű megoldást mutat be erre a problémára.

1. Bevezetés, motiváció

Hogy megszüntessük, illetve mérsékeljük az önző viselkedést a hálózatban, egy elosztott keretrendszer válik szükségessé, amely ösztönzi a résztvevőket a kommunikációra és az együttműködésre. Egy ilyen környezetben az autonóm hálózatok önkéntes együttműködésére elosztott architektúra válik szükségessé, amely vezérli az együttműködéseket [3]. Nem feltételezünk semmilyen központi bizalmat vagy infrastruktúrát.

Az ígéretek elmélete egy gráfelméleti keretrendszer, amely egyszerűbbé teszi az összetett kapcsolatok, relációk megértését olyan hálózati környezetben, ahol sokféle korlátozásnak kell megfelelni [3,4]. Az alapötlet szerint teljesen autonóm csomópontok ígéreteken keresztül lépnek kapcsolatba egymással. Az együttműködő csomópontok csoportokba szerveződnek. Minden egyes ígéret egy korlátozást jelent az ígérő csomópont viselkedésére nézve.

Nagyméretű elosztott hálózatokban a hálózat komponensei megosztják a szolgáltatásaikat és hálózatmenedzsment-funkcióikat egymással. De a csomópontok nem járnak jól azzal, ha az összes szolgáltatásukat megosztják mások számára.

Minden hálózati csomópont igényel szolgáltatásokat más csomópontoktól. Ha egy csomópont csak szolgáltatásokat kér, de maga nem szolgálja ki más csomópontok kéréseit, akkor ez azt jelenti, hogy ez a csomópont önző módon viselkedik. Annak érdekében, hogy ezt a viselkedést megszüntessük a hálózatban és hogy arra ösztönözzük a csomópontokat, hogy együttműködjenek, különféle technikákat használhatunk. Az alapelve ezeknek a megoldásoknak az, hogy a nagylelkű csomópontokat megjutalmazzuk, az önző csomópontokat pedig megbüntetjük.

Ha egy csomópont jutalmat kap, az azt jelenti, hogy nagyobb valószínűséggel fogják kiszolgálni az ő kéréseit más csomópontok. Ha pedig egy csomópont büntetést kap, akkor kisebb valószínűséggel lesz kiszolgálva. A felvázolt rendszer modellezéséhez egyfajta játékelméleti megközelítés a legalkalmasabb. Az erre a modellre illeszkedő játék az általános fogolydilemma. A rendszer működéséhez a csomópontoknak információkat kell tárolniuk más csomópontok viselkedéséről annak érdekében, hogy egy szolgáltatáskéréskor döntést tudjanak hozni, hogy kiszolgálják-e az adott kérést, vagy sem.

Ennek az információnak, élettörténetnek a tárolása alapvetően két különböző módon történhet: közös területen (shared history), vagy egyénileg (private history) [5]. A kétféle tárolási módnak különféle hátrányai vannak, közös területen történő tárolás esetén előfordulhat, hogy egy csomópont hamis ajánlásokat küld egy másik csomópontra vonatkozóan, vagyis hazudik egy másik csomópontról és ez tönkreteheti a kooperációt. A közös területen történő tároláshoz egy elosztott adattárolási módszer is szükséges, például elosztott hash táblákkal. Az egyénileg tárolt history a nagyszámú csomópont esetén teljesíthetetlen memóriakövetelményeket támasztana a csomópontokkal szemben, ezért ebből a szempontból ez a megoldás csak korlátozott mértékben használható.

Az erőforrások megosztásának játékelméleti modellel történő leírása széleskörben vizsgált, kutatott terület, különösen a P2P fájlmegosztó hálózatok népszerűvé válása óta. Sok különféle megközelítést dolgoztak ki, hogy ösztönözzék a hálózat résztvevőit a saját erőforrásaik megosztására. Ezekben a hírnév alapú ösztönző rendszerekben a csomópontoknak egy hasznossági értéke van, amelyet a csomópont működése során növelni, maximalizálni akar. Ennek a hasznossági értéknek a számítása az alapján történik, hogy a csomópont milyen mér-

tékben osztotta meg erőforrásait és milyen mértékben használt ki más csomópontokat. Az egyik legátfogóbb kutatás ezen a téren Ion Stoica-nak és csapatának köszönhetjük [5], de számos más értékes publikáció is született a témában. Ezek a kutatások sokmindenben különböznek, például abban a tekintetben, hogy milyen játékelméleti modellel elemzik a rendszert. Ion Stoica és csapata két résztvevős, aszimmetrikus modellt használ, míg Philippe Golle sokrésztvevős játékkal végzi az elemzést [6].

Ezekben a megoldásokban a rendszer P2P alapelveken működik, vagyis bárki bárkivel kapcsolatba léphet, szolgáltatást kérhet és szolgáltatást nyújthat. Jelen cikkben vázolt megoldás abban különbözik ettől, hogy topológia épül ki a hálózatban. Az ambiens hálózatokban a csomópontoknak csak egy korlátozott lefedettségi területe van és ez alapján csak a szomszédaikkal tudnak közvetlenül kommunikálni. Ennek az a következménye, hogy útvonalirányításra van szükség a hálózatban és egy szolgáltatáskérés több csomóponton is áthalad.

Ebből az következik, hogy egy szolgáltatás kérésekor három fajta csomópontot lehet megkülönböztetni, amely ebben a folyamatban részt vesz: egy kezdeményező csomópontot, amely szolgáltatást kér, egy célcsomópontot, amelytől a szolgáltatást kérték és opcionálisan valahány közbenső, továbbító csomópontot, amelyek továbbítják a kérést és a választ. Természetesen olyan eset is előfordulhat, hogy valamely csomópont a közvetlen szomszédjától kér szolgáltatást, ebben az esetben a közbenső csomópontok kimaradnak.

2. Elméleti megfontolások, újdonságok

A játékelmélet a matematikának egy olyan ága, amely azzal a kérdéssel foglalkozik, hogy mi az ésszerű viselkedés egy olyan helyzetben, amikor a résztvevő döntéseinek eredményét, hatását más résztvevők döntései is befolyásolják.

Egy játék leírásához alapvetően három dolog megadása szükséges: a játékosok, a stratégiák és a kifizetések. A játékosok a játék résztvevői, akik a kifizetésüket maximalizálni szeretnék. Stratégia alatt a játékosok viselkedését értjük, vagyis azt, hogy a játék során milyen döntéseket hozhatnak. Kifizetés alatt a játékos hasznossági függvényét értjük, azt az értéket, amelyet a játékos, mint hasznot elkönyvelhet a játék végén. Ez az érték függ attól, hogy a játékos milyen stratégiát választ, illetve attól is, hogy más játékosok milyen stratégiával játszanak. Mivel a játékos racionális, azt szeretné, hogy ez a hasznosság-érték minél nagyobb legyen. Ehhez azonban figyelembe kell vennie más játékosok döntéseit, illetve döntési lehetőségeit is, valamint saját kifizetéseit ezek függvényében.

A játékoknak sok fajtája, csoportosítása létezik, például normál formájú vagy extenzív formájú játékok, szimmetrikus vagy aszimmetrikus, zérus összegű, vagy nem zérus összegű játékok. A normál formájú játékok legegyszerűbb megadási módja a kifizetési mátrix.

A rendszer működésének megértéséhez először néhány ejtsünk szót a fogolydilemmáról. Ennek a játéknak sokfajta változata létezik. Az alapötlet az, hogy két, egymástól elkülönített, bűnténnyel gyanúsított fogoly van bezárva külön börtöncellába. Mindkettejüknek ugyanazok a lehetőségei: ha egyikőjük vallomást tesz a másik ellen, akkor szabadon engedik, míg a másik fogoly 10 évet kap. Ha egyikőjük sem tesz vallomást, 6 hónapot kapnak, ha mindketten vallomást tesznek, 6 évet kapnak. A foglyok nem kommunikálhatnak egymással, nem tudnak együttműködni, ez egy nem kooperatív játék. Itt tehát a büntetésként kapott időt lehet úgy felfogni, mint egyfajta negatív hasznosságot és ezt szeretnénk minimalizálni. Az itt felvázolt játék kifizetési mátrixát az *1. táblázat* mutatja (egy cellában az első szám az 1. játékos kifizetése (hasznossága) míg a második szám a 2. játékosé).

A játék általánosítása abban különbözik az eredeti játéktól, hogy a kifizetések értékeire különféle korlátozásokat és szabályokat definiáltak. Ennek alapján sokféle különböző fogolydilemma játékot lehet felírni, amelyek teljesítik ezeket a szabályokat. Ennek a részleteivel nem foglalkozunk.

Az általunk definiált együttműködési rendszerben a csomópontok viselkedését egy – a fentihez hasonló – fogolydilemma játék modellezi. A játékban azonban a felek nem ugyanolyanok, mivel itt a hálózatban kérésekről és kérések kiszolgálásáról, illetve továbbításáról van szó. Ez azt jelenti, hogy egy interakció során mindig lesz egy kliens csomópont és egy szerver csomópont. A kliens kezdeményezi a kérést a szerver felé. Ezt a megközelítést aszimmetrikus fogolydilemmának is hívják. A szimuláció során használt kifizetési mátrixot a *2. táblázat* tartalmazza. Ez a mátrix megegyezik a már előbb említett publikációban alkalmazottal.

1. táblázat
A klasszikus fogolydilemma játék

		2. játékos	
		Hallgat	**Vall**
1. játékos	**Hallgat**	-0.5 / -0.5	-10 / 0
	Vall	0 / -10	-6 / -6

2. táblázat
A csomópontok által játszott játék kifizetési mátrixa

		Szerver játékos	
		Szolgáltat	**Figyelmen kívül hagyja**
Kliens játékos	**Szolgáltatást kér**	7 / -1	0 / 0
	Nem kér szolgáltatást	0 / 0	0 / 0

A táblázatban lévő számok az egyes játékosok hasznosságát, kifizetését jelentik. Ezt a játékot sokszor játszák egymással a hálózat szereplői, s az így kapott pontszámokat gyűjtik. Ez azt jelenti, hogy ha egy csomópont egy szolgáltatást kér egy másik csomóponttól, akkor két eset történhet: vagy kiszolgálja a csomópont a kérő csomópontot, ez esetben a szerver csomópont -1 pontot kap, a kliens pedig 7 pontot, vagy nem szolgálja ki, ebben az esetben mindketten 0 pontot kapnak.

A játékosoknak háromfajta stratégiája lehet: mindig együttműködő, sosem együttműködő és viszonzó. Az első stratégia azt jelenti, hogy minden hozzá érkező kérést feltétel nélkül kiszolgál a csomópont. A második stratégia ennek pont az ellentéte; sosem szolgálja ki a csomópont a kérést. A harmadik stratégiában játszanak szerepet a csomópont által tárolt információk a szolgáltatást kérő csomópont viselkedéséről. Ennek a stratégiának a használatakor a csomópont ezek alapján az információk alapján dönti el, hogy kiszolgálja-e a kérő csomópontot vagy sem.

A folyamat során, játékról játékra gyűjtik (vagy vesztik el) a csomópontok az összpontszámukat. Minden csomópont statisztikát készít arról, hogy melyik stratégia használata volt számára a legjövedelmezőbb. Ha egy csomópont úgy látja, hogy egy másik stratégia jövedelmezőbb számára, mint amit jelenleg használ, akkor stratégiát vált. Ebben az esetben a csomópont azonosítója is megváltozik, vagyis a csomópontról mások által tárolt információk is érvényüket vesztik. (Kivétel ez alól az áruló csomópont, amely akkor is megtartja az azonosítóját, ha stratégiát váltott. Erről később lesz szó.)

Egy csomópont nem csak akkor növelheti a hasznosságát, ha szolgáltatást nyújt, hanem akkor is, ha szolgáltatást továbbít. A szolgáltatás-továbbítás ugyanolyan súlyú szolgáltatásnak számít, mintha azt közvetlenül a csomópont nyújtanál. A szolgáltatást kérő csomópont számára gyakorlatilag átlátszó, hogy ki nyújtja a szolgáltatást. Így valósul meg a szolgáltatások továbbítása, amely egy útvonalirányítási mechanizmuson keresztül működik. A csomópontok ismerik azokat az útvonalakat, amelyeken keresztül eljuthatnak más csomópontokhoz, ezáltal tudják, hogy ha egy adott csomóponttól kérnek szolgáltatást, akkor melyik szomszédjukhoz kell ezt a kérést először továbbítaniuk.

Felmerül a kérdés: miért szolgálná ki egy szerver csomópont egy kliens kérését, ha ez neki negatív pontszámot jelent? A válasz erre a már ismertetett tárolt élettörténet működésében rejlik. Ha egy csomópont nem szolgál ki más csomópontokat, akkor előbb-utóbb az ő kéréseit sem fogják kiszolgálni, amiből az következik, hogy nem tud pontszámot gyűjteni, hosszú távon tehát ez a működésmód nem fogja megérni neki. Ez persze attól is függ, hogy ez a kiszolgáló csomópont milyen más csomópontokkal kerül kapcsolatba, mert majd látni fogjuk, hogy bizonyos esetben előfordulhat, hogy egy csomópont a nem együttműködő stratégiát részesíti előnyben a többi stratégiával szemben.

Egy újabb fajta csomópont is be lett vezetve, az áruló csomópont, amely működése különbözik az eddig tárgyalt csomópontokétól. Ez a csomópont úgy működik, hogy miután stratégiát váltott, az azonosítója megmarad és érvényben maradnak a róla tárolt információk a többi csomópontban. Egy ilyen csomópont elvben például megtehetné azt, hogy a szimuláció első felében együttműködik mindenkivel, majd a második felében senkinek a kérését nem szolgálja ki, mert az elején gyűjtött sok pont miatt az ő kérését úgyis nagy valószínűséggel ki fogják szolgálni más, viszonzó stratégiát folytató csomópontok. A rendszer működését ilyen csomópontok jelenlétével is vizsgáltuk.

A rendszer működése során a csomópontok információkat tárolnak arról is, hogy milyen más csomópontokkal kerültek eddig kapcsolatba. A csomópontok emlékeznek rá, hogy mely kliensek kértek tőlük szolgáltatást. Ezt az emlékezetét felhasználják, amikor ők kerülnek kliens szerepkörbe és azoktól a csomópontoktól nagyobb valószínűséggel kérnek szolgáltatást, akik tőlük is kértek már szolgáltatást. Ebből következően egy csomópont viszonozni tudja, ha őt kiszolgálták, azáltal, hogy ő is kiszolgálja a társát. Emiatt az alapelv miatt a hálózat viselkedése konvergál egy viszonylag stabil állapot felé a szimuláció során, s bár a szimuláció utolsó szakaszaiban is történnek még stratégiaváltások, de drasztikus átpártolás már nem történik, stabilizálódik a rendszer.

3. Szimuláció, eredmények

A kidolgozott rendszer vizsgálata szimulációval történt. A szimuláció körökre oszlott, minden egyes körben minden csomópont szolgáltatást kért valamely más csomóponttól, vagyis a fentebb ismertetett játékot játszák. Ez a játék végighalad a teljes kiszolgálási útvonalon, vagyis azon az útvonalon, amelyen a szolgáltatás továbbítása történik a kliens és a szerver csomópont között. Egy szimuláció 1000 körből áll. A rendszer működésének vizsgálata különféle esetekre történt. A más csomópontokról tárolt élettörténet tárolását módját rövid és hosszútávon is megvizsgáltuk. Ha csak rövid ideig tároljuk ezeket az információkat, az azt jelenti, hogy egy csomópont hamar „tisztára tudja mosni” magát, megbocsátó a rendszer, de ennek később kárát is láthatják más csomópontok. Hosszútávú élettörténet tárolásához viszont nagyobb memória szükséges és hatékony keresést is implementálni kellene benne, hogy megfelelően működjön. Ezt a két esetet megosztott és egyedileg tárolt élettörténet esetén is megvizsgáltuk.

A szimuláció során egy 100 csomópontból álló hálózat működését vizsgáltuk. A csomópontok elhelyezkedése véletlenszerű volt, így a kialakult topológia is véletlenszerű. Azt vizsgáltuk, hogy az egyes csomópontoknak mely stratégiát éri meg leginkább használni. Ez sok mindentől függhet, például a csomópont elhelyezkedésétől a hálózatban (sok csomópont veszi-e körül, vagy kevés) illetve attól is, hogy az azt körülvevő csomópontok milyen stratégiával játszanak. A szimuláció kezdetén az egyes stratégiák egyenlő arányban, véletlen-

szerűen oszlottak meg a csomópontok között, tehát a csomópontok 1/3 része kezdetben együttműködő volt, 1/3 része nem együttműködő, 1/3 része pedig a viszonzó stratégiát játszotta.

Általánosságban elmondható, hogy a legtöbb esetben az együttműködő és a viszonzó stratégia volt a legjövedelmezőbb. Bizonyos esetekben azonban egyes hálózatrészekben jobban elterjedt a nem együttműködő viselkedés. Különbözőképpen viselkedett a rendszer, ha megengedtük az áruló csomópontok jelenlétét is, ezek arányát 25%-ra állítottuk be.

A szimuláció során a hálózat egy statikus állapot felé közelített. Ez azt jelenti, hogy a csomópontok nagy részének már nem volt érdeke, hogy stratégiát változtasson, a stratégiaváltások gyakorisága a teljes hálózatra nézve csökkent. A grafikonokon csak az látszik, hogy egy adott stratégiát hány csomópont használ az adott szimulációs körben, de az nem, hogy melyek ezek a csomópontok, így az nem derül ki, hogy nagyjából ugyanazok a csomópontok használták-e a stratégiát, vagy más csomópontok. Ennek szemléltetésére minden egyes szimulációs körben készítettünk egy térképet a hálózatról, amely különböző színnel jelöli a különböző stratégiákat.

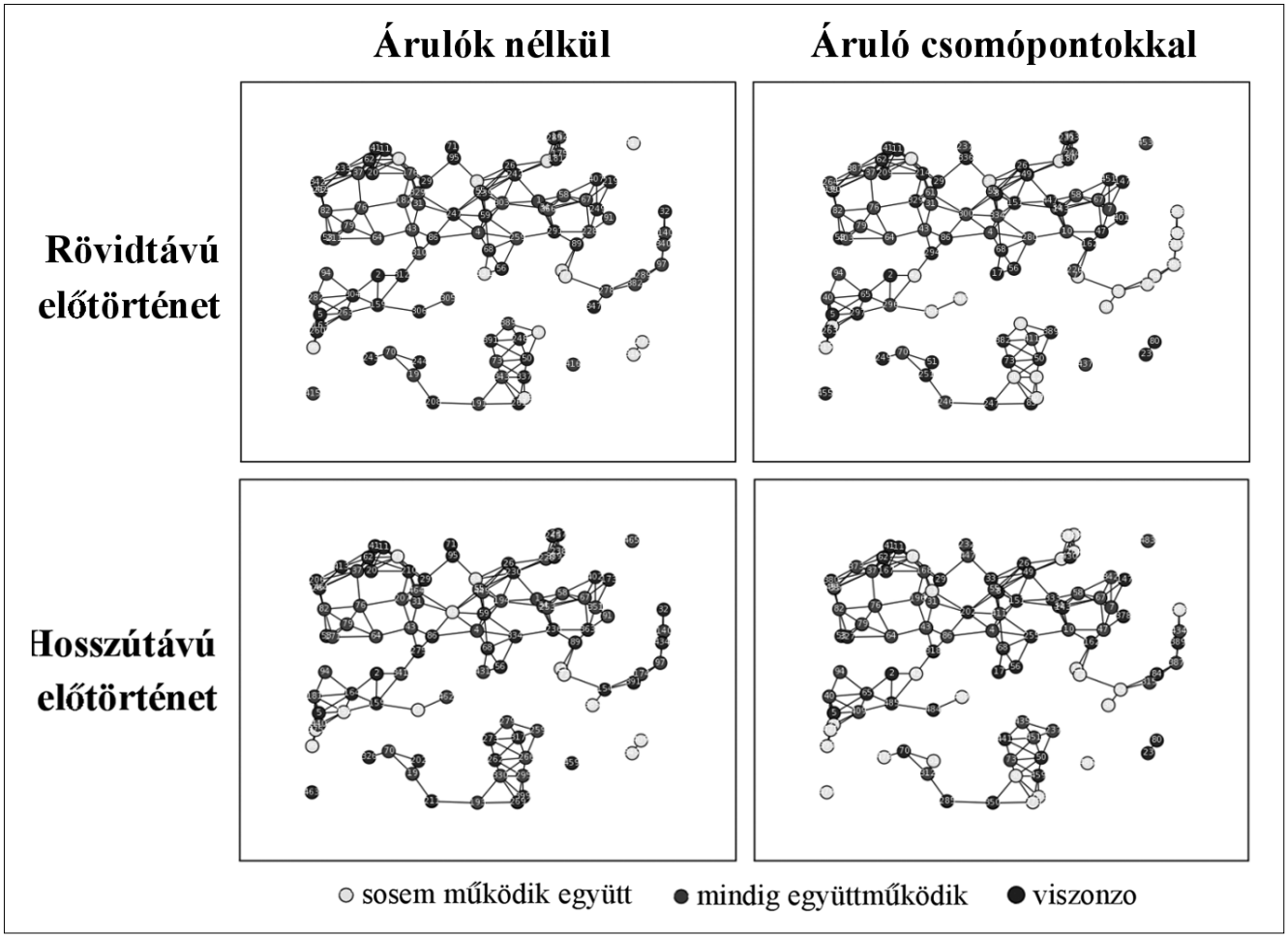
Ezeket a térképeket megvizsgálva azt állapítottuk meg, hogy csak a szimuláció elején történik tömeges stratégiaváltás, a későbbi szakaszokban már lényegi változás nem megy végbe. Ennek a folyamatnak a vizsgálatával lehetőség nyílt arra is, hogy felderítsük a stratégiák eloszlásának a topológián belüli helytől való függését.

Az *1. ábra* a szimuláció végére kialakult térképet szemlélteti különféle szimulációs forgatókönyvek esetén. Jól látható, hogy áruló csomópontok jelenlétekor több a nem együttműködő csomópont, mintha csak normál működésű csomópontok lettek volna a hálózatban. Érdemes megfigyelni, hogy a rövidtávú előtörténet használatakor és áruló csomópontok jelenlétekor az ábra jobb oldalán levő „nyúlványban” minden csomópont ellenszenvesen viselkedett.

Ez a hatás tehát az egész hálózatrészben elterjedt, ez a viselkedés pedig az áruló csomópontok jelenlétekor tapasztalható. Azokban a hálózatrészekben, ahol viszonylag sűrűn helyezkednek el a csomópontok, nagyjából ugyanolyan viselkedést láthatunk minden esetben, vannak azonban területek, amelyek az áruló csomópontok jelenléte miatt kevésbé együttműködővé válnak.

A *2. ábrán* az egyes stratégiát követő csomópontok eloszlása látható a szimuláció során. Látható, hogy az áruló csomópontok jelenléte esetén jobban ingadozik az eloszlás, gyakrabban változtatnak stratégiát a csomó-

1. ábra A csomópontok stratégiák szerinti eloszlása a topológia gráfban



pontok. Ez a hatás a hosszútávú és rövidtávú előtörténeteket alkalmazó megoldások összehasonlításakor is látható. Az előző ábrával összhangban látható, hogy a szimuláció végén az egyes szimulációs esetekben a csomópontok mekkora hányada követte az egyik vagy másik stratégiát. Az áruló csomópontok esetében jól látható a különbség, a szimuláció végére ténylegesen több csomópont választotta azt, hogy sosem kooperál más csomópontokkal.

4. Összefoglalás

Összefoglalásként elmondható, hogy a kidolgozott keretrendszer képes ösztönözni a csomópontokat az önkéntes együttműködésére. Bizonyos esetekben ez az együttműködés magasfokú és kevés a nem együttműködő csomópont, más esetekben a hálózat egyes részei nem együttműködő csoportokat alkotnak.

Többféle módon lehet folytatni a rendszer vizsgálatát, például bele lehet vinni, hogy a csomópontok ne csak egyhelyben álljanak, hanem helyet is változtathassanak. Ebben az esetben természetesen gondoskodni kell az útvonalirányítás megfelelő működéséről is, hogy a gyorsan változó hálózatban a csomópontok továbbra is megtalálják egymást. Ebben az irányban is történtek vizsgálatok, azonban a hálózat topológiájának kismértékű, de rendszeres változtatása is erősen éreztette a hatását a csomópontok által választott stratégiák eloszlásában. Ez azt jelenti, hogy a hasonló grafikonok ábrázolásából nem lehet sok mindenre következtetni. Ennek az esetnek a vizsgálata is még további kutatás tárgyát képezi.

Irodalom

[1] N. Niebert, H. Flinck, R. Hancock, H. Karl, C. Prehofer, Ambient Networks – Research for Communication Networks Beyond 3G, 2004.

[2] Kovács Balázs, Simon Csaba, "Ambient" hálózatok, 2005.

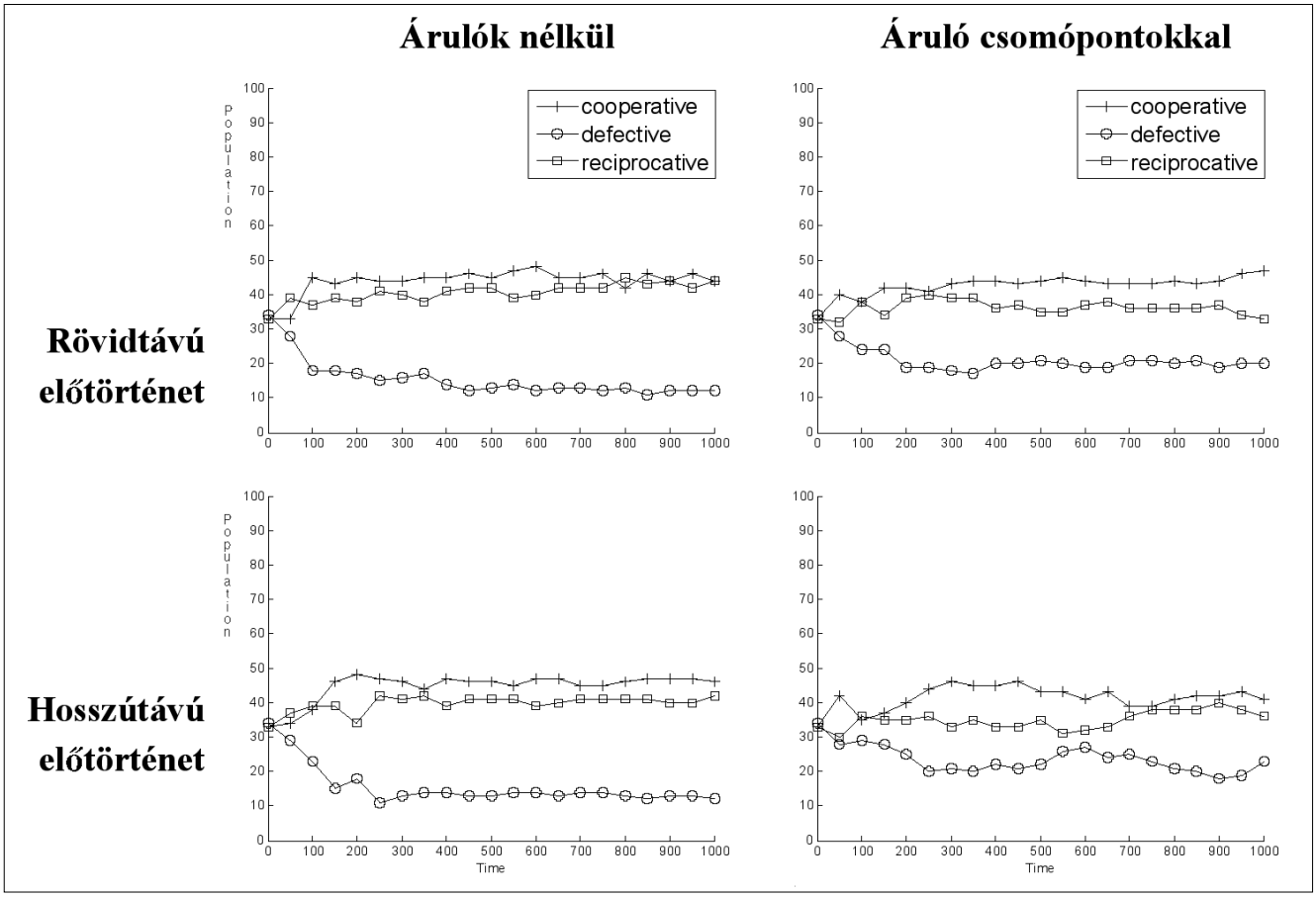
[3] Mark Burgess, An Approach to Understanding Policy Based on Autonomy and Voluntary Co-operation, Lecture Notes on Computer Science, 2005.

[4] Mark Burgess and Siri Fagernes, Pervasive Computer Management: A Model of Network Policy with Local Autonomy, IEEE Transactions on Networking, 1999.

[5] Michalel Feldman, Kevin Lai, Ion Stoica, John Chuang, Robust Incentive Techniques for Peer-to-Peer Networks, ACM Conference on Electronic Commerce, June 2004.

[6] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, Incentives for sharing in peer-to-peer networks, 3rd ACM conference on Electronic Commerce, Tampa, Florida, USA, 2001.

2. ábra Az egyes stratégiákat használó csomópontok számának alakulása a szimuláció során



Biztonsági API analízis a spi-kalkulussal

BUTTYÁN LEVENTE, TA VINH THONG

BME Híradástechnikai Tanszék, CrySyS Adatbiztonsági Laboratórium
{buttyan, thong}@crysys.hu

Lektorált

Kulcsszavak: hardver biztonsági modul, formális módszerek, processz-algebra, biztonság, titkosság

Az API-szintű támadások komoly veszélyt jelentenek a hardver biztonsági modulokra nézve, ezért fontos követelmény az API-ban rejlő biztonsági lyukak felfedezése és foltozása. Az API analízis egyik ígéretes iránya a formális verifikációs módszerek alkalmazása. Cikkünkben ezt az irányt követjük, s egy processz-algebra alapú API verifikációs módszert javaslunk, mely különösen alkalmasnak látszik a biztonsági API-k működésének formális leírására, a biztonsági követelmények precíz definiálására, és a megfogalmazott követelmények teljesítésének ellenőrzésére. Munkánk motiválása céljából ismertetünk néhány konkrét API-szintű támadást is egy a gyakorlatban elterjedten használt hardver biztonsági modul ellen.

1. Bevezetés

Számos alkalmazásban használnak hardver biztonsági modulokat (Hardware Security Module, HSM). HSM alatt olyan hardver eszközt értünk (a rajta futó firmvare és szoftver komponensekkel együtt), mely bontás-ellenálló (tamper resistant) tulajdonságokkal rendelkezik, s ezáltal alkalmas kriptográfiai kulcsok biztonságos tárolására, valamint különböző biztonság-kritikus kriptográfiai algoritmusok (például digitális aláírás generálás, PIN-kód generálás) végrehajtására.

A hardver biztonsági modulok polgári célú alkalmazása a bankszférában kezdődött az 1960-as években. Az ebben az időszakban történt bankkártya hamisítások arra ösztönözték az IBM-et (mint a kor banki számítástechnikai rendszereinek legfőbb szállítóját), hogy kifejlesszen egy olyan rendszert, amely lehetővé teszi a felhasználók PIN-kódjának előállítását a bankkártyán tárolt számlaszámból egy PIN-kód származtatási kulcs (PIN derivation key) segítségével. Ennek kapcsán szükségessé vált a PIN-kód származtatási kulcsok megfelelő védelme, mind külső támadók mind pedig a bank belső alkalmazottai ellen. Ez a követelmény vezetett az IBM 3848, első generációs HSM kifejlesztéséhez, melyet később széleskörben alkalmaztak a banki ATM hálózatokban. Mára a HSM-ek alkalmazási köre kiszélesedett, s a banki alkalmazásokon túl, elterjedten használják őket például a nyilvános kulcs infrastruktúrákban (Public Key Infrastructure, vagy PKI), a tömegközlekedési elektronikus díjbeszedési rendszerekben és általában az elektronikus kereskedelem területén.

A HSM-ek elleni klasszikus támadási módszerek a fizikai támadások [2]. Ezek lehetnek a hardver modul fizikai megbontásával, esetleg roncsolásával járó intruzív támadások, vagy a HSM működési környezetének, például időzítéseinek, áramfelvételének megfigyeléséből és manipulálásából származó támadások. A fizikai támadások hatékonyak, ám sokszor költséges berendezéseket igényelnek.

A fizikai támadások mellett a közelmúltban megjelentek a jóval kisebb költséggel járó szoftver alapú támadások, melyek a HSM alkalmazás programozói interfészében (Application Programming Interface, API) rejlő gyengeségeket, hibákat aknázzák ki. Számos elterjedten használt (s különben erős fizikai védelmet biztosító) HSM ellen találtak API-szintű támadást [3-7,10-11]. Nyilvánvaló, hogy kívánatos lenne az API-ban rejlő biztonsági lyukak felfedezése és foltozása, ideálisan még az adott HSM széleskörű telepítése előtt. Ugyanakkor a gyakorlatban használt API-k több száz függvényt tartalmazó komplex rendszerek, ami megnehezíti az analízisüket.

Az API analízis egyik ígéretes iránya a szoftverfejlesztés területén használt formális verifikációs módszerek alkalmazása [8-9,11-12,14-15]. Cikkünkben ezt az irányt követjük, s egy processz-algebra alapú API verifikációs módszert javaslunk, mely különösen alkalmasnak látszik a biztonsági API-k működésének formális leírására és a biztonsági követelmények precíz definiálására. Konkrétan az itt bemutatott módszer a spi-kalkulusra épül [1], melyet eredetileg kulcscsere protokollok analízisére fejlesztettek ki. Ismereteink szerint mi használtuk először a spi-kalkulust biztonsági API-k analízisére.

A továbbiakban először egy konkrét HSM (a Visa Biztonsági Modul) elleni, API-szintű támadásokat mutatunk be illusztratív céllal. Hasonló támadások léteznek más HSM-ek ellen is. Ezek a támadások motiválják a 4. szakaszban bemutatásra kerülő API analízis módszert, melynek alapját a 3. szakaszban ismertetésre kerülő spi-kalkulus képezi.

2. A Visa Biztonsági Modul támadása az API-n keresztül

A Visa Biztonsági Modul (Visa Security Module, VSM) kifejlesztésével a Visa célja az volt, hogy meggyőzze a hozzá tartozó tagbankokat, hogy csatlakoztassák ATM-

jeiket a Visa hálózatához és ezáltal lehetővé váljon, hogy bármely tagbank ügyfele pénzt vehessen fel egy olyan ATM-ből, amely egy másik tagbankhoz tartozik. Ennek érdekében a Visa-nak biztosítania kellett, hogy bármely tagbank más tagbank gondatlanságából származó esetleges vesztesége a lehető legkisebb legyen. Ez többek között azt is jelenti, hogy az egyes tagbankok ügyfeleinek PIN-kódját, más tagbankok belső alkalmazottai nem tudhatják meg. Azaz a PIN-kódokat nem lehet egyszerűen a bankok mainframe-jein futó szoftverben kezelni. Ezért a PIN-kódok kezelése a fizikai védelmet biztosító VSM-ekben történik.

Mivel a HSM-ek belső tárkapacitása korlátos, ezért általában csak a legfontosabb kulcsokat (az úgynevezett mesterkulcsokat) tárolják a HSM-ben. Minden más kulcsot a típusuknak megfelelő mesterkulccsal kódolják és külső tárban tárolják. A megszokott tárolási mód a hierarchikus struktúra, amelynek előnye, hogy hatékony és áttekinthető. Hátránya azonban, hogy ha egy felsőszintű kulcs kompromittálódik, akkor minden, a hierarchában alatta elhelyezkedő kulcs is kompromittálódik.

A VSM kulcshierarchiája az *1. ábrán* látható. A VSM kilenc kulcstípust támogat, ezeket az ábrán a téglalapok jelképezik. A kulcshierarchia legfelső szintjén helyezkednek el a mesterkulcsok, melyek a VSM-en belül tárolódnak. Minden más kulcsot ezekkel a mesterkulcsokkal kódolva külső tárban tárolnak. Látható, hogy a belül tárolt mesterkulcsokból öt darab van.

A *ZCMK* az a mesterkulcs, amivel az összes *ZCK* típusú kulcsot kódolják. A *ZCK* típus a zónavezérlő kulcsokat (Zone Control Key) jelöli. Ezek a kulcsok a különböző bankhálózatok között vannak megosztva és a bankhálózatok közötti kulcscserében játszanak szerepet. A *WMK* az a mesterkulcs, amivel az összes *WK* típusú kulcsot kódolják, ahol a *WK* munkakulcsokat (Working Key) jelöl. A *WK* típusú kulcsok funkciója az, hogy a beütött PIN-kódot védjék, miközben az eljut a banki hálózaton keresztül ahhoz a bankhoz, ahol ellenőrizni tudják. A *TCMK* mesterkulccsal kódolják a *TCK* típusú kulcsokat, ahol a *TCK* típus a terminál kommuninkációs kulcsokat (Terminal Communication Key) tartalmazza. A terminál kommunkációs kulcsok funkcióihoz tartozik a VSM-ek között cserélendő üzenetek integritásvédő kódjának kiszámítása. Az *MK* mesterkulcs a *TMK* terminál-mesterkulcsok és a *P* PIN-kód származtatási kulcsok kódolásáért felelős. A *TMK* kulcsot később még tágyaljuk. Mivel a *TMK* kulcsokat más kulcsok kódolására használják (például a zónakulcsok kódolására), a *P* kulcs pedig a PIN-kód kiszámításában játszik szerepet, ezért helyezkedik el két hierarchia szinten is mindkét típus. Mivel nem lényegesek jelen cikk szempontjából, ezért az *LPMK* és *LPK* kulcsokat nem tárgyaljuk. Az *X{}* típusba olyan kulcsok vagy adatok tartoznak, amelyeket az *X* típusú kulccsal kódoltak.

Egy új ATM üzembehelyezésekor a banknak el kell juttatnia az új ATM-nek a működéséhez szükséges kulcsokat. Ehhez először a bank egy új terminál-mesterkulcsot *(TMK)* oszt meg az ATM-mel, majd minden más kulcsot ezzel a *TMK* kulccsal kódolva juttat el az ATM-hez.

A *TMK* kulcs létrehozása a következő módon történik. A hoszt meghívja a VSM API-jának *GenerateKeyShares* nevű[1] függvényét:

$$Host \rightarrow VSM : "GenerateKeyShares"$$

Erre a VSM generál egy TMK_i részkulcsot, majd egyrészt kinyomtatja a generált részulcsot a megfelelő biztonságos printeren:

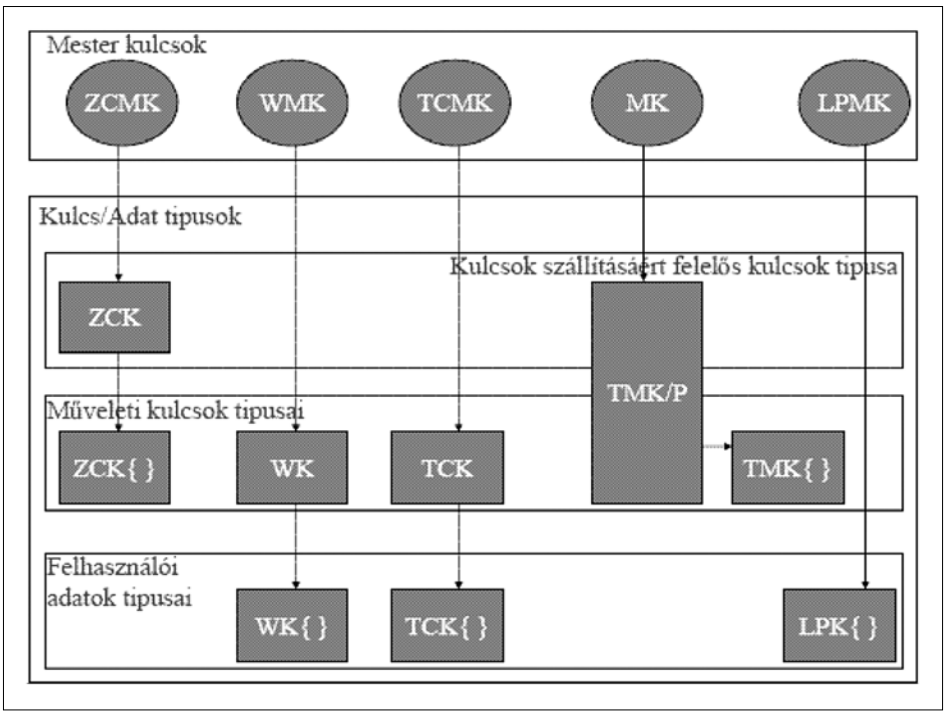
$$VSM \rightarrow SecurePrinter : TMK_i$$

másrészt visszaadja a részkulcsot egy a VSM belsejében tárolt *MK* mesterkulccsal kódolva a hosztnak:

$$VSM \rightarrow Host : \{TMK_i\}_{MK}$$

A hoszt annyiszor hajtja végre a fenti hívást, amennyi részkulcsot szeretne generálni. A továbbiakban feltesszük, hogy a szükséges részkulcsok száma kettő. A biztonságos printereken kinyomtatott részkulcsokat a

1. ábra A Visa Biztonsági Modul (VSM) kulcshierarchiája



1 *A cikkben használt függvénynevek nem mindenhol egyeznek a VSM specifikációban szereplő függvénynevekkel.*

meghatalmazott személyek eljuttatják az új ATM-hez. Miután megkapta mindegyik részkulcsot, az ATM előállítja a *TMK* terminál-mesterkulcsot a részkulcsok XOR-olásával: $TMK=TMK_1 \oplus TMK_2$. A banknál ugyanez a *TMK* kulcs áll elő a VSM API *CombineKeyShares* függvényének meghívása után:

$$Host \rightarrow VSM : "CombineKeyShares", \{TMK_1\}_{MK}, \{TMK_2\}_{MK}$$
$$VSM \rightarrow Host : \{TMK_1 \oplus TMK_2\}_{MK} = \{TMK\}_{MK}$$

A fenti terminál-mesterkulcs generálási eljárás egy támadási lehetőséget rejt magában. Nevezetesen, a hoszt (illetve az azt kezelő alkalmazott) meghívhatja a *CombineKeyShares* függvényt két azonos paraméterrel (például a kódolt részkulcsok egyikével):

$$Host \rightarrow VSM : "CombineKeyShares", \{TMK_1\}_{MK}, \{TMK_1\}_{MK}$$
$$VSM \rightarrow Host : \{TMK_1 \oplus TMK_1\}_{MK} = \{0\}_{MK}$$

Az így létrehozott *TMK* a csupa nulla bitből álló kulcs. A VSM ezt a kulcsot használja többek között a *P* PIN-kód származtatási kulcs kódolására, mielőtt az átküldésre kerül az ATM-nek. A VSM által előállított $\{P\}_0$ kódolt kulcsot azonban a támadó is könnyen dekódolni tudja a csupa nulla kulccsal. A *P* kulcs segítségével ezek után tetszőleges számlaszámhoz tartozó PIN kódot elő tud állítani a támadó.

Egy másik támadási lehetőség abból adódik, hogy a VSM API-ja tartalmaz egy *EncryptCommsKey* függvényt, amely egy *TCK* típusú kulcsot vár paraméterként és válaszként az ehhez a kulcstípushoz tartozó *TCMK* mesterkulccsal kódolva adja vissza a kulcsot:

$$Host \rightarrow VSM : "EncryptCommsKey", TCK$$
$$VSM \rightarrow Host : \{TCK\}_{TCMK}$$

Említettük, hogy minden szükséges kulcsot el kell juttatni az új ATM-hez, s ez alól a *TCK* kulcs sem kivétel. A szállítás a *TMK* kulccsal kódolva történik, így szükség van egy $\{TCK\}_{TMK}$. kulcstokenre. Ennek létrehozását a *TranslateCommsKeytoTMK* függvény biztosítja:

$$Host \rightarrow VSM : "TranslateCommsKeytoTMK", \{TCK\}_{TCMK}, \{TMK\}_{MK}$$
$$VSM \rightarrow Host : \{TCK\}_{TMK}$$

A támadás azt használja ki, hogy a VSM azonos *MK* kulcs alatt tárolja a *TMK* és a *P* kulcsokat. A támadás menete a következő. A bank rosszindulatú alkalmazottja kiadja az *EncryptCommsKey* parancsot, de paraméterként nem egy *TCK* kulcsot ad meg, hanem egy *PAN* felhasználói számlaszámot:

$$Host \rightarrow VSM : "EncryptCommsKey", PAN$$
$$VSM \rightarrow Host : \{PAN\}_{TCMK}$$

Ezután, a támadó meghívja a *TranslateCommsKeytoTMK* függvényt az előző lépésben kapott $\{PAN\}_{TCMK}$ értékkel és egy korábban megszerzett $\{P\}_{MK}$ rejtjelezett PIN-kód származtatási kulccsal mint paraméterrel[2]:

$$Host \rightarrow VSM : "TranslateCommsKeytoTMK", \{PAN\}_{TCMK}, \{P\}_{MK}$$
$$VSM \rightarrow Host : \{PAN\}_P = PIN$$

A visszakapott érték a PIN-kód származtatási kulccsal rejtjelezett számlaszám, azaz pontosan a számlatulajdonos PIN-kódja.

3. A spi-kalkulus áttekintése

Ebben a részben rövid áttekintést adunk a spi-kalkulusról [1], ami a π-kalkulus [13] kiterjesztése különböző kriptográfiai primitívekkel. A π-kalkulushoz hasonlóan, a spi-kalkulus is egy egyszerű programozási nyelvnek tekinthető. Ennélfogva a spi-kalkulus kiválóan alkalmas a biztonsági API-k működésének modellezésére.

3.1. A spi-kalkulus nyelvtana

A spi-kalkulusban a kommunikációs csatornákat nevekkel jelöljük. Végtelen névhalmazt feltételezünk, ezenkívül bevezetjük a változók végtelen halmazát is, amelyeknek az értékadásnál lesz majd szerepük. A változókat az *x, y,* és *z* betűkkel jelöljük, a neveket pedig többek között *m, n,* és *c* betűkkel. Két alapvető nyelvi elemet különböztetünk meg: *term*-eket (üzenetek, csatorna azonosítók, kulcsok stb.), melyek adatot reprezentálnak és *processz*-eket, melyek a viselkedést írják le. A termek lehetnek atomiak, mint a konstansok és változók, vagy összetett termek.

A termeket a következő nyelvtan szerint definiáljuk:

$L, M, N ::=$	*termek*
n	*név*
(M, N)	*pár*
0	*nulla*
$suc(M)$	*következő*
x	*változó*
$\{M_1, M_2, ..., M_k\}_N$	*szimmetrikus kulcsú titkosítás*

Tehát egy term lehet egy név, egy term pár, nulla, egy adott term utáni term, vagy egy változó. Külön kiemeljük továbbá az $\{M_1,M_2,...,M_k\}_N$ formájú termeket, melyek szimmetrikus kulcsú titkosítással előállított kritpogramokat reprezentálnak, ahol *N* jelöli a kulcsot, az $M_1,M_2,...,M_k$ termek pedig a nyílt üzenet mezőit.

A processzeket a következő nyelvtan szerint definiáljuk:

$P, R, Q ::=$	*processzek*
$\overline{M}\langle N_1, N_2, ..., N_k \rangle.P$	*küldés* $(k \geq 0)$
$M(x_1, x_2, ..., x_k).P$	*vétel* $(k \geq 0)$
$P \mid Q$	*(párh.) kompozició*
$(vn)P$	*megkötés*
$!P$	*replikáció*
$[M \text{ is } N]P$	*összehasonlítás*
0	*null processz*
$let\ (x, y) = M\ in\ P$	*pár szétválasztás*

2 *Ez lehetséges, mivel* TMK *és* P *ugyanúgy az* MK *kulccsal vannak kódolva és a VSM csak azt nézi, hogy az adott kulcstoken sikeresen kódolható-e az* MK *kulccsal.*

$case\ M\ of\ 0 : P\ suc(x) : Q$	*egész szám eset*
$case\ L\ of\ \{x_1, x_2, \ldots, x_k\}_N\ in\ P$	*szimmetrikus kulcsú dekódolás* $(k \geq 0)$

Az egyes konstrukciók jelentése a következő:

• **Küldés**

Az *M* term itt egy csatornát reprezentál. Ez a processz kész a $N_1,N_2,...,N_k$ termeket elküldeni az *M* csatornán keresztül. Ha egy üzenetváltás (lásd később) létrejön, akkor $N_1,N_2,...,N_k$ elküldésre kerül az *M* csatornán keresztül és a *P* processz fut tovább.

• **Vétel**

Ez a processz az előző párja. Egy üzenetváltás során a *küldés* processz elküldi az $N_1,N_2,...,N_k$ termeket mint üzeneteket az *M* csatornán, a *vétel* processz pedig ugyanezen a csatornán veszi ezeket a termeket és a $P[N_1/x_1,N_2/x_2,...,N_k/x_k]$ processz fut tovább, ahol *N/x* az értékadást jelöli. Azaz vétel során a vett termekkel mint értékekkel helyettesítjük a megfelelő változókat a *P* processzben.

• **Kompozició** (P|Q)

Ez a konstrukció a *P* és *Q* processzek párhuzamos futását jelöli. *P* és *Q* kommunikálhat egymással egy közös megosztott csatornán keresztül, vagy *P* és *Q* egymástól függetlenül kommunikálhat a környezettel.

• **Megkötés** (*vn*)*P*

A *P* processz létrehoz egy új *n* lokális nevet. A *P* processzen kívül más processzben – hacsak nem kapta meg explicite valamilyen kommunikáció során – ez a név nem jelenhet meg. E konstrukció segítségével modellezhetjük egy új titkos kulcs létrehozását.

• **Replikáció** (!*P*)

Ez a konstrukció a *P* processz végtelen sok példányának párhuzamos kompozicióját jelöli.

• **Összehasonlítás** ([*M is N*]*P*):

Ez a processz úgy viselkedik, hogy amennyiben *M*=*N* akkor a *P* processz fut, különben a futás elakad.

• **Null processz** (0)

Ez konstrukció a semmittevést vagy elakadást jelöli.

• **Pár szétválasztás** (*let* (x,y)=*M in P*)

Ez a processz a termek nyelvtanában a definiált párképzésnek ellentettje. Ha *M*=(*N,L*), akkor a *P*[*N/x*][*L/y*] processz fut tovább, egyébként a futás elakad.

• **Egész szám eset** (*case M of* 0:*P suc* (x):*Q*)

Ez a processz úgy viselkedik mint *P,* ha *M* értéke 0, vagy úgy, mint *Q*[*N/x*], ha *M*=*suc(N)*, máskülönben elakad.

• **Szimmetrikus kulcsú dekódolás**

A *case L of* $\{x_1,x_2,...,x_k\}_N$ *in P* processz megpróbálja dekódolni az *L* termet az *N* kulccsal. Ha *L* egy $\{M_1,M_2,...,M_k\}_N$ formájú term, akkor a $P[M_1/x_1,M_2/x_2,...,M_k/x_k]$ processz fut tovább. Különben a processz elakad.

A fent leírt, kriptográfiai elemeket használó nyelvi konstrukciók a következő alapfeltevésekre épülnek:

– Egy rejtjelezett üzenet csak a rejtjelezés kulcsának megfelelő dekódoló kulccsal fejthető meg.
– A rejtjelező kulcs nem következtethető ki a vele rejtjelezett üzenetből.
– A rejtjelezett üzenet elég redundanciát hordoz ahhoz, hogy a dekódoló algoritmus egyértelműen el tudja dönteni, hogy a dekódolás sikeres volt, vagy nem.
– A támadó nem képes kitalálni és/vagy létrehozni bármilyen titkosnak minősített protokoll adatot.

3.2. A titkosság modellezése a spi-kalkulusban

A spi-kalkulusban a támadó egy tetszőleges *R* processz, melyről csak annyit tételezünk fel, hogy kezdetben nincsenek nála titkos adatok. A támadó processz párhuzamosan fut a rendszert modellező processzel, és azzal interakcióba léphet (kommunikálhat) a publikus csatornákat használva. Ezen interakció során szerzett információkból próbálja a támadó kinyerni a titkokat a rendszerből.

A *titkosság* mint biztonsági tulajdonság alapja a spi-kalkulusban a processzek *megkülönböztethetetlensége*. Azaz a protokoll titokban tart egy *M* adatot, ha tetszőleges *M'* adat esetén, a támadó *R* processz nem tud különbséget tenni a *P*(*M*) és a *P*(*M'*) processzek között, ahol *P*(*x*) a protokollt reprezentáló (paraméterezhető) processz.

A megkülönböztethetetlenség formális definiciója a *tesztelési ekvivalencia* fogalmára épül. Ennek megértéséhez be kell vezetnünk néhány további fogalmat:

• **Szabad és kötött változók**

A *P* processzben az *x* változó *kötött változó*, ha *P* tartalmaz egy *m*(*x*) vétel részprocesszt (tetszőleges *m*-re). A *P* processzben az *x* változó *szabad változó*, ha *P* nem tartalmaz *m*(*x*) vétel részprocesszt. Egy *P* processz szabad változóinak halmazát *fv*(*P*)-vel jelöljük.

• **Zárt processz**

Akkor mondjuk egy processzre, hogy zárt, ha nincs szabad változó benne. A spi-kalkulusban minden támadó processzről felteszük, hogy zárt.

• **Üzenetváltás**

Egy üzenetváltás akkor jön létre, amikor egy $\overline{m}\langle M\rangle.P$ küldés processz és egy $m(x).Q$ vétel processz párhuzamos kompozicióban áll egymással. Ekkor a küldés processzes elküldi az *M* termet az *m* csatornán, ezt veszi a vétel processzt és *P*|*Q*[*M/x*] fut tovább.

Formálisan:

$$\overline{m}\langle M\rangle.P \mid m(x).Q \rightarrow P \mid Q[M/x]$$

• **Barb kimutatás**

A *barb kimutatás* intuitív jelentése, hogy egy processz használja-e az adott csatornát üzenet küldésre vagy fogadásra. A barb kimutatást a ↓ jelöli. A barb kimutatás teljességgel független a kiadott vagy kapott üzenettől. A barb kimutatásra a következő axiómák érvényesek:

– **Barb In**: Ha egy processz azonnal használja az *m* csatornát adat fogadásra, akkor az az *m* barb-ot kimutatja, azaz $m(x).P{\downarrow}m$.
– **Barb Out**: Ha egy processz azonnal használja az *m* csatornát adat küldésre, akkor az az $\overline{m}$ barb-ot kimutatja, azaz $\overline{m}\langle M\rangle.P{\downarrow}\overline{m}$

• **Konvergencia**

A konvergencia intuitíven azt jelenti, hogy a processz nem feltétlenül azonnal használja az adott csatornát, hanem csak az üzenetváltásainak sorozata során valamikor használja azt. Ennek jelölése $\Downarrow$ és a kapcsolódó axiómák a következők:

– Ha egy processz a β barb-ot kimutatja, akkor konvergál a β-hoz.
– Ha egy P processz át tud alakulni egy olyan Q processzbe, ami a β barb-ot kimutatja, akkor P konvergál a β barb-hoz.

Most, hogy a szükséges fogalmakat bevezettük, megadjuk a *tesztelési ekvivalencia* formális definícióját:

Definíció (tesztelési ekvivalencia)

Egy teszt egy (R,β) pár, ahol R egy tetszőleges zárt proccessz és β egy barb (m vagy $\overline{m}$). P és Q között fennáll a teszlelési ekvivalencia, azaz $P \approx Q$, akkor és csak akkor, ha $P \subseteq Q$ és $Q \subseteq P$ egyszerre fennállnak, ahol $P \subseteq Q$ akkor és csak akkor áll fenn, ha minden (R,β) teszt esetén $(P|R)\Downarrow\beta$-ból következik $(Q|R)\Downarrow\beta$.

Intuitíven, $P \approx Q$ azt jelenti tehát, hogy a P és Q processzek megkülönböztethetetlenek egy külső R megfigyelő számára. Azaz, P és Q belső struktúrája lehet különböző, de ezt a P és Q-val párhuzamos kompozícióban levő harmadik zárt R processz nem tudja detektálni, a P-vel és Q-val folytatott üzenetváltások során.

4. Egy egyszerű API modellezése spi-kalkulussal

Bár a spi-kalkulust elsősorban kulcscsere-protokollok modellezésére dolgozták ki, jól alkalmazható HSM-mel történő API-n keresztüli interakciók modellezésére is. Ez annak köszönhető, hogy egy API függvényhívás nagyon hasonló egy két lépéses protokollhoz, melyben a hoszt kiad egy kérést, és a HSM visszaad egy választ. A teljes API-t a lehetséges függvényhívásokat reprezentáló processzek párhuzamos kompozíciójával modellezhetjük. Erre mutatunk példát ebben a szakaszban.

Először egy egyszerű biztonsági API-t definiálunk. Feltesszük, hogy a HSM tartalmaz egy *MK* mesterkulcsot. Megkülönböztetünk két kulcstípust, a K_i adatkódoló-kulcsot és a KEK_j kulcskódoló-kulcsot, amikhez hozzá rendeljük a *DataKey* és *KEKKey* típus indikátorokat. A K_i adatkódoló-kulcsot tartalmazó kulcstokenek *DataKey* típus indikátort kapnak, míg a KEK_j kulcskódoló-kulcsot tartalmazó tokeneket *KEKKey* indikátorokkal látjuk el. Bevezetünk egy *TData* típusindikátort is, amit a felhasználói adatot tartalmazó rejtjeles szövegek típusának jelzésére használunk. Feltesszük még, hogy a modul nem tárolja az adatkódoló-kulcsokat és a kulcskódoló-kulcsokat, helyette kiadja magából kulcstokenként ezeket $\{DataKey,K_i\}_{MK}$ és $\{KEKKey,KEK_j\}_{MK}$ formában.

Példa API-nk négy függvényt tartalmaz:

• **Adat kódolás**

Ez a függvény argumentumként valamilyen *Data* felhasználói adatot és egy $\{DataKey,K_i\}_{MK}$ kulcstokent vár. Dekódolja a $\{DataKey,K_i\}_{MK}$-t az *MK* mesterkulccsal, ellenőrzi a kulcs típusát és ha az *DataKey*, akkor K_i-vel kódolja a *Data* adatot. Ezután visszaadja a $\{TData, Data\}_{K_i}$ rejtjelezett adatot.

• **Adat dekódolás**

Ez a függvény argumentumként egy $\{TData, Data\}_{K_i}$ kódolt adatot és egy $\{DataKey,K_i\}_{MK}$ kulcstokent vár. Dekódolja a $\{DataKey,K_i\}_{MK}$-t az *MK* mesterkulccsal, majd ellenőrzi a kulcs típusát, és ha az *DataKey* akkor K_i-vel dekódolja a $\{TData, Data\}_{K_i}$-t. Végül ellenőrzi, hogy a típus *TData*-e, s ha igen, akkor (és csak akkor) visszaadja a *Data* adatot.

• **Adatkulcs exportálása**

Ez a függvény két kulcstokent kap inputként, $\{DataKey,K_i\}_{MK}$-t és $\{KEKKey,KEK_j\}_{MK}$-t. Dekódolja mindkét kulcstokent *MK*-val, ellenőrzi, hogy a kulcsok típusa a várt *DataKey* és *KEKKey* típus-e, s ha igen, K_i-t kódolja KEK_j-vel, majd visszaadja a $\{DataKey,K_i\}_{KEK_j}$ kulcstokent. Ez kerül majd átküldésre egy másik modulnak, amely importálhatja a K_i kulcsot.

• **Adatkulcs importálása**

Ez a függvény két kulcstokent kap inputként, $\{DataKey,K_i\}_{KEK_j}$-t és $\{KEKKey,KEK_j\}_{MK}$-t. Dekódolja a $\{KEKKey,KEK_j\}_{MK}$ kulcstokent *MK*-val és ellenőrzi a kulcs típusát. Majd a $\{DataKey,K_i\}_{KEK_j}$-t az eredményként kapott KEK_j-vel dekódolja és ellenőrzi a kapott kulcs típusát. Végül az eredményként kapott K_i-t kódolja a mesterkulccsal. Ezután visszaadja az így kapott $\{DataKey, K_i\}_{MK}$-t.

A fent definiált egyszerű API-t a következőképpen modellezhetjük a spi-kalkulus segítségével. Jelölje $MODULE^{ENC}$, $MODULE^{DEC}$, $MODULE^{EXP}$, $MODULE^{IMP}$ rendre az adat-kódoló, adat-dekódoló, adatkulcs-export és adatkulcs-import processzeket. Minden processz kommunikációs csatornákon keresztül kapja az adatokat, ebben az esetben az argumentumokat. A fogadási kommunikációs csatornákat rendre a c_{enc}, c_{dec}, c_{exp}, c_{imp} nevek jelölik, ezeken keresztül kapják a processzek az adatot. Továbbá definiálunk egy $\overline{c_{user}}$ csatornát, melyen keresztül a processzek a környezetnek (hosztnak) küldhetnek adatokat. A processzek formális leírása a következő:

1. $MODULE^{ENC}(MK)$

$$c_{enc}(x_{data}, x_{token0}).case\ x_{token0}\ of\ \left\{x_{typeK}, x_{K_i}\right\}_{MK}\ in\ \left[x_{typeK}\ is\ DataKey\right]$$
$$\overline{c_{user}} < \left\{TData, x_{Data}\right\}_{x_{K_i}} >$$

2. $MODULE^{DEC}(MK)$

$$c_{dec}(x_{token1}, x_{token2}).case\ x_{token2}\ of\ \left\{x_{typeK}, x_{K_i}\right\}_{MK},\ x_{token1}$$
$$of\ \left\{x_{typeData}, x_{Data}\right\}_{x_{K_i}}\ in\ \left[x_{typeK}\ is\ DataKey\right]\ \left[x_{typeData}\ is\ TData\right]$$
$$\overline{c_{user}} < x_{Data} >$$

3. $MODULE^{EXP}(MK)$

$$c_{\exp}(x_{token3}, x_{token4}).case\ x_{token3}\ of\ \left\{x_{typeK}, x_{K_i}\right\}_{MK},\ x_{token4}$$
$$of\ \left\{x_{typeKEK}, x_{KEK}\right\}_{MK}\ in\ \left[x_{typeK}\ is\ DataKey\right]\ \left[x_{typeKEK}\ is\ KEKKey\right]$$
$$\overline{c_{user}} < \left\{DataKey, x_{K_i}\right\}_{x_{KEK}} >$$

4. $MODULE^{IMP}(MK)$

$c_{imp}(x_{token5}, x_{token6}).case\ x_{token6}\ of\ \{x_{typeKEK}, x_{KEK}\}_{MK}\ ,\ x_{token5}$

$of\ \{x_{typeK}, x_{K_i}\}_{x_{KEK}}\ in\ [x_{typeK}\ is\ DataKey]\ [x_{typeKEK}\ is\ KEKKey]$

$\overline{c_{user}} < \{DataKey, x_{K_i}\}_{MK} >$

A teljes API-t a fenti processzek replikációinak párhuzamos kompozíciójaként reprezentáljuk, néhány kezdeti kulcstoken kiadásával. Ezek a kulcstokenek azért kerülnek kiadásra, mert ezek a HSM-en kívül tárolódnak, s ezért bárki (beleértve a támadót) hozzájuk férhet.

$Sys_{API}(K_i, KEK_j)$

$$(\nu MK)\left(\begin{array}{l}\overline{c_{user}} < \{DataKey, K_i\}_{MK}, \{KEKKey, KEK_j\}_{MK} > . \\ \left(!MODULE^{ENC}(MK) \mid !MODULE^{DEC}(MK) \mid !MODULE^{EXP}(MK) \mid !MODULE^{IMP}(MK)\right)\end{array}\right)$$

Formálisan is bizonyítható (amit helyhiány miatt most mellőzünk), hogy ez az API semilyen körülmények között nem fogja kiadni a környezete számára a kulcsokat. Az intuitív magyarázat az, hogy az egyetlen függvény, amely nyíltszöveget ad vissza, az *adat dekódolás* függvény, ám a típusindikátorok miatt az *adat dekódolás* függvény csak akkor adja vissza a nyíltszöveget, ha annak típusa *TData*, vagyis nem kulcs.

A formális bizonyítás során a következő tesztelési ekvivalenciák fennállását kell bizonyítani: $Sys_{API}(K_i, KEK_j) \approx Sys_{API}(K_i', KEK_j)$ és $Sys_{API}(K_i, KEK_j) \approx Sys_{API}(K_i, KEK_j')$ minden K_i, K_i', KEK_j, KEK_j' esetén. Ennek bizonyítása indukció segítségével történik. Felteszük, hogy kezdetben az *R* támadó processz nem rendelkezik semmilyen kulccsal, azaz a rendszer biztonságos állapotban van. Majd bebizonyítjuk, hogy ha a rendszer biztonságos állapotban van, akkor az *R* és a rendszer közötti bármely üzenetváltást követően is biztonságos marad. Ez tehát azt jelenti, hogy a támadó nem tud egyetlen kulcsot sem megszerezni a rendszerből az API-n keresztül.

5. Következtetés

Az API szintű támadások komoly veszélyt jelentenek a hardver biztonsági modulokra nézve. Cikkünkben egy formális módszert javasoltunk az API biztonsági analízisére, mely lehetővé teszi annak bizonyítását, hogy egy külső támadó nem képes titkos kulcsot kinyerni a modulból az API-n keresztül. A sikeres bizonyítás az API biztonságosságát jelenti, míg a sikertelen bizonyítás általában valamilyen API hibára hívja fel a figyelmet. A javasolt módszer alapját a spi-kalkulus képezi, melyet eredetileg kulcscsere-protokollok analízisére terveztek. Egy egyszerű API-n keresztül bemutattuk a spi-kalkulus alkalmazhatóságát. Tapasztalataink azt mutatták, hogy a spi-kalkulus jól alkalmazható API ellenőrzési célokra.

Irodalom

[1] M. Abadi and A. Gordon,
Calculus for cryptographic protocols: the Spi calculus.
Technical Report SRC RR 149, Digital Equipment Co.,
Systems Research Center, January 1998.

[2] R. Anderson, M. Bond, J. Clulow, S. Skorobogatov,
Cryptographic processors – a survey.
Technical Report UCAM-CL-TR-641,
University of Cambridge, Computer Laboratory,
August 2005.

[3] M. Bond,
Attacks on cryptoprocessor transaction sets.
In Proceedings of the CHES 2001 Workshop,
Springer LNCS 2162, 2001.

[4] M. Bond,
Understanding security APIs.
PhD thesis,
University of Cambridge, 2004.

[5] M. Bond and R. Anderson,
API level attacks on
embedded systems.
IEEE Computer Magazine,
October 2001.

[6] M. Bond and J. Clulow,
Encrypted? Randomised? Compromised?
(When cryptographically secured data is not secure).
In Proceedings of the Workshop on Cryptographic
Algortihms and their Uses, 2004.

[7] M. Bond and P. Zielinski,
Decimalisation table attacks for PIN cracking.
Technical Report UCAM-CL-TR-560,
University of Cambridge, Computer Laboratory,
January 2003.

[8] E. M. Clarke, A. Biere, R. Raimi, Y. Zhu,
Bounded model checking using satisfiability solving.
Formal Methods in System Design, 19 July 2001.

[9] J. Clulow,
The design and analysis of cryptographic APIs.
MSc thesis, University of Natal, South Africa, 2003.

[10] J. Clulow,
On the security of PKCS#11.
In Proceedings of the CHES 2003 Workshop.
Springer LNCS 2779, 2003.

[11] V. Ganapathy, S. A. Seshia, S. Jha,
T. W. Reps, R. E. Bryant,
Automatic discovery of API-level vulnerabilities.
In Proceedings of the ACM/IEEE Conference
on Software Engineering (ICSE), 2005.

[12] A. H. Lin,
Automated analysis of security APIs. MSc Thesis,
Massachusetts Institute of Technology, May 2005.

[13] R. Milner, J. Parrow, D. Walker,
A calculus of mobile processes, Parts I-II.
Information and Computation, September 1992.

[14] M. Moskewicz, C. Madigan, Y. Zhao,
L. Zhang, S. Malik,
Engineering an efficient SAT solver.
In Proc. of the 38th Design Automation Conference,
June 2001.

[15] P. Youn,
The analysis of cryptographic APIs using
the theorem prover otter. MSc Thesis,
Massachusetts Institute of Technology, May 2004.

Távközlő hálózati folyamatok monitorozása

TATAI PÉTER
AITIA International Zrt.
VARGA PÁL, MAROSI GYULA
BME Távközlési és Médiainformatikai Tanszék, TSPLab
{varga, marosi}@tmit.bme.hu

Kulcsszavak: passzív hálózat, GSM, GPRS, távmonitorozás, forgalmi statisztikák, híváskövetés, No.7-es jelzések dekódolása

A dinamikusan bővülő hálózatok monitorozásához jól skálázható elosztott adatgyűjtésre és tárolásra van szükség, ugyanakkor az adatok korrelálása csak központi feldolgozással valósítható meg. A hálózatmonitorozás során többszáz gigabájtnyi információ gyűlik össze napok alatt. Rendkívül fontos tehát az adatok feldolgozása és tömör prezentálása a felhasználók számára. A távmonitorozás és az üzenetek dekódolása (olvasható értelmezése) csak az első lépés. Összetett forgalmi és sikerességi statisztikák, hívásrekordok készítése, valamint kiválasztott hívások követése is alapvető fontosságú a hibakeresés és fenntartás céljából. Jelen cikk a hálózatok passzív vizsgálatát ismerteti és bemutatja, hogyan lehet feloldani az ellentmondást a nagyvolumenű adatgyűjtés valamint a valós idejű protokoll analízis és híváskövetés között.

1. Bevezetés

A távközlő hálózatok fenntartása és a bővítések tervezése során nélkülözhetetlenek a hálózat működésére, a forgalom mennyiségére és vonalak állapotára vonatkozó részletes, rendszeresen gyűjtött adatok. Ezek alapján lehet a hálózati hibákat és szűk keresztmetszeteket kiküszöbölni, valamint az előfizetői panaszok okát és az esetleges csalásokat felderíteni. A forgalom mennyiségének és növekedésének részletes analízise pedig támpontot ad a vonalak és a hálózati csomópontok számának és kapacitásának tervezéséhez. Figyelembe véve a távközlési igények rohamos növekedését, ezek a feladatok különösen fontosak az egyre nagyobb forgalom kezeléséhez, valamint az új szolgáltatások bevezetéséhez.

A hálózati forgalom mérése igen nagy mennyiségű adat összegyűjtését, feldolgozását és tárolását igényli. Tipikusan több millió telefonhívás vagy adat tranzakció adatait kell feldolgozni naponta vagy akár forgalmas óránként, a hálózat méretétől és a szolgáltatás elterjedtségétől függően. Ugyanakkor a hibák, előfizetői panaszok vagy csalások felderítése gyakran közel valós idejű feldolgozást igényel, amellyel akár a vizsgált kapcsolat bontása előtt meg lehet találni egyedi hívásokat is. Ezek a szélsőséges követelmények csak jól skálázható, nagyteljesítményű mérőrendszerrel teljesíthetőek, amelynek feldolgozási-tárolási kapacitása megfelelően szét van osztva, és így tudja követni a távközlő hálózati forgalom fokozatos és gyors növekedését, valamint az alkalmazott protokollkészlet állandó bővülését.

Egy ilyen mérőrendszer, amelyet a következőkben monitorozó rendszernek nevezünk, célszerűen független a hálózati berendezésektől, mert azok nem eléggé flexibilisek a szolgáltatói igények teljesítéséhez és nincs elegendő többlet számítástechnikai kapacitásuk részletesebb és időkritikus vizsgálatokra, valamint az igényelt újabb meg újabb szolgáltatásokra.

A jelenlegi távközlő hálózatok működését többségében közös csatornás (CCS, Common Channel Signaling) 7-es jelzésrendszerű (SS7, Signaling System No.7) jelzésüzenetek (MSU, Message Signal Unit) vezérlik, és ezek figyelésével, monitorozásával részletes információt kaphatunk a hálózat állapotáról, továbbá a tendenciák figyelésével nagyobb problémák is megelőzhetők, és a fejlesztési irányok is kijelölhetők.

Emiatt a szolgáltatók többsége használ részleges vagy teljes hálózatot lefedő jelzés monitorozó rendszert, amely zavarás nélkül, passzívan csatlakozik a vonalakhoz és gyűjti, majd központilag feldolgozza a jelzésüzeneteket [1,2].

A hálózati berendezéseket összekötő vonalak jelentős része 2048 kbit/s-os (E1) PCM trönk, amelyek egy vagy több 64 kbit/s-os időrésében, úgynevezett jelzéslinkeken haladnak a 7-es jelzésüzenetek HDLC (High Level Data Link Control) keretekben. A rézvezetékes E1 vonalakat az utóbbi években részben felváltották a fénykábeles, főként 155,52 Mbit/s-os SDH összeköttetések, illetve rohamosan terjednek világszerte az IP alapú jelzés-összeköttetések, tipikusan Sigtran protokollon. Amerikában E1 vonalak helyett a 1544 kbit/s sebességű T1 vonalak és ezeken belül 56 kbit/s-os jelzéslinkek használatosak. A monitorozásnál tehát többféle fizikai csatlakozásra is fel kell készülni.

Noha a jelzésüzenetek egy-egy vonalon egyedi műszerekkel, protokoll-analizátorokkal is megfigyelhetőek, azonban egyetlen hívás vagy adat tranzakció üzenetei is rendszerint számos vonalról gyűjthetők csak össze, sőt ezek tipikusan földrajzilag is eltérő helyeken érhetők el. Emiatt az adatgyűjtés lehetősége kiterjedt távközlő hálózat esetén protokoll analizátorokkal nagyon korlátozott, a gyakorlatban egyedi műszerek helyett jelzéseket monitorozó, gyűjtő és feldolgozó rendszerre van szükség, amelynél a központi feldolgozás biztosítja az eltérő helyekről gyűjtött üzenetek korrelálását és összerendelését.

Az adatgyűjtést és feldolgozást jelentősen nehezíti, ha a jelzésüzenetek a monitorozott vonalakon már titkosított formában kerülnek továbbításra, mint például a GPRS rendszereknél [3]. Ebben az esetben már az üzenetek típusa is titkosítva van, ezért a szolgáltató számára is nehézséget jelent a forgalmi statisztikák létrehozása. A megoldást a titkosító kulcsokat szállító vonalak monitorozása és a kulcsok gyűjtése jelenti, ami után a jelzésüzenetek és a kulcsok összerendelésével „kititkosíthatók" a fenntartáshoz szükséges adatok. Mindez természetesen nem érinti a felhasználói információkat, amelyek továbbra is titkosan továbbítódnak, de ezek ismeretére nincs is szükség az üzemeltetéshez.

A monitorozó rendszerben összegyűjtött jelzésinformáció – például ki kit hívott és mikor – szintén „érzékeny" adat, ezért a hozzáférést itt is szigorú szabályok rögzítik. Többszintű jogosultság kezelésre van szükség, ahol pontosan szabályozható, hogy ki milyen információhoz férhet hozzá és annak a megadása is lényeges, hogy milyen célból használja éppen valaki a monitorozó rendszert. Ilymódon biztosítható, hogy csak illetékes szakember férhessen a rendszerhez és csak a tényleges fenntartási, üzemeltetési, hibakeresési célokra használják az adatokat.

2. Rendszer áttekintés

Az *1. ábra* egy jelzés-monitorozó rendszer főbb elemeit és azok kapcsolatát jeleníti meg. A monitor egységek a jelzéseket szállító trönk vonalakhoz csatlakoznak. A monitorozás nem zavarhatja a vonali átvitelt, ezért rézvezetékek esetén nagyimpedanciás leválasztással, fénykábelek esetén pedig jelosztóval (splitter) szokás csatlakozni, de egyes berendezéseken külön monitorozó kivezetések is találhatók erre a célra.

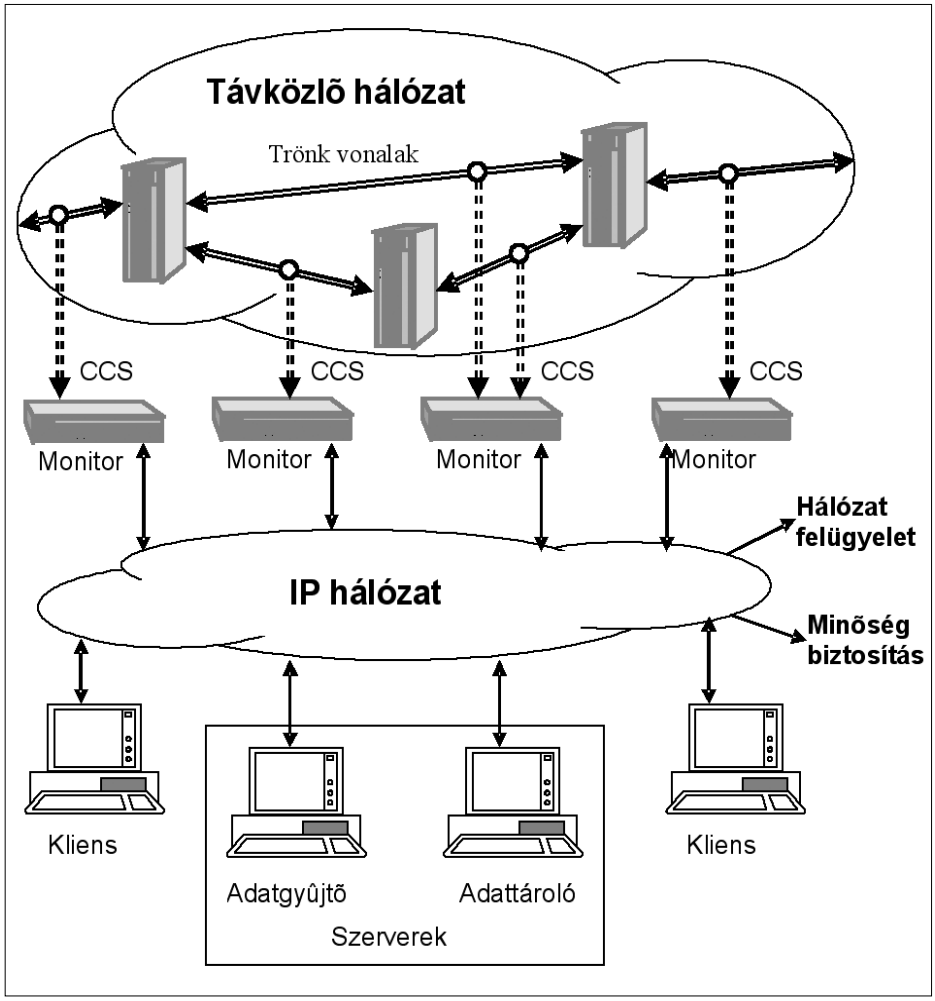
A trönkvonalak jeleiből csak a jelzéseket szállító linkeket kell leágaztatni. Természetesen mindkét irányban figyelni kell a jelzéseket, ezért a monitor egységek trönk vonalanként két bemenetet igényelnek.

A monitor egységek, valamint az adatok gyűjtését, feldolgozását és tárolását végző szerver gépek az IP hálózaton keresztül vannak kapcsolatban és elosztott feldolgozó rendszerként működnek. Az összes jelzésüzenet tárolását, előfeldolgozását, szűrését és esetleges „kititkosítását", valamint a statisztikák gyűjtését a monitor gépek végzik. Itt kerül minden üzenetre egy 1 ms felbontású időpecsét, amely például egy pontos órajel (ilyen a bejövő vonali jelekből kinyert időzítés) és a hálózatban szinkronizált gépek órájának segítségével állítható elő. Nagyobb pontossági igény esetén GPS alapú időpecsétek is előállíthatók.

Az adatok lekérdezésének, sorbarendezésének, korrelálásának, a hívás- és egyéb rekordok összeállításának feladata a szerver gépekre hárul, mert ezt csak egy központi helyen, az összegyűjtött üzeneteken lehet elvégezni. Néhány jelzéslinktől sokezer linkig, gyakorlatilag korlátlanul bővíthető a rendszer egyszerűen az elemek számának növelésével, vagyis újabb monitorok beállításával, valamint a feladatmegosztásban működtetett szerver gépek számának növelésével. Ily módon a rendszer igen hatékonyan skálázható.

Ugyanakkor a monitorokban történő előfeldolgozás és szűrés jelentősen csökkenti az IP hálózat terhelését. Tovább csökkenti a hálózat terhelését a bináris üzenetek helyi tárolása, ami azt is lehetővé teszi, hogy a helyi merevlemez kapacitásának korlátjáig – akár több hétig visszamenőlegesen – éppúgy használható az összes funkció, mintha valós időben éppen az aktuális jelzési folyamatokat vizsgálnánk.

1. ábra
Jelzés monitorozó rendszer elemei és csatlakozása a távközlő hálózathoz



Szerver célra egy vagy több gép is alkalmazható a forgalomtól és a feladatok mennyiségétől függően. A feldolgozott adatok történelmi (historikus) adatbázisba kerülnek, ahonnan később is lehívhatók. Figyelembe véve, hogy nagyobb hálózatok esetén óránként akár több millió adatbázis bejegyzés is készülhet, a folyamatos betöltés közben történő visszaolvasás és egyéb műveletek, például az adatok részletesebb analízise, kritikus sebesség-problémákat is felvet, amelyek megoldása egyáltalán nem triviális. A feladat az ismert adatbázisok (pl. Oracle) használatával is megoldható, de a célra optimalizált adatbázissal nagyságrenddel nagyobb keresési és visszaolvasási sebesség érhető el.

A monitorozó rendszer mind a távközlő hálózat, mind a saját működése során jelentkező hibákról gyűjt adatokat, amelyeket a hálózat felügyeleti központjába (NOC/NMC, Network Operating/Maintenance Center) továbbít. Így a személyzet azonnal értesül a hibákról és megteheti a szükséges hibaelhárító intézkedéseket. Ez természetesen csak kiegészíti a hálózat hibafelügyeleti rendszerét, de hozzájárulhat a hibák gyorsabb érzékeléséhez és az okok feltárásához.

A rendszer monitor egységei földrajzilag általában szétszórtan helyezkednek el az ország területén, a jelzéseket kezelő központok (STP, Signaling Transfer Point) közelében. Ezért fontos, hogy fenntartásuk, vagyis a működésük ellenőrzése, konfigurálásuk és új verziók letöltése egy központi telephelyről legyen megoldható. Az esetleges hardver hibáktól eltekintve minden szoftver funkció és konfiguráció beállítás távolról is elvégezhető, ami egy elosztott rendszernél alapvető követelmény.

A felhasználók kliens gépekről, vagyis akár saját asztali vagy hordozható gépükről bárhonnan hozzáférhetnek a gyűjtött és feldolgozott adatokhoz, ahonnan engedélyezett IP kapcsolat van a monitorozó hálózat elemeihez. A használatot kliens alkalmazói programok könnyítik meg, amelyek az egyes funkciókra vannak optimalizálva. A hozzáférések engedélyezésének vezérlése a szerveren tárolt jogosultsági adatokkal történik és minden hozzáférés és művelet jegyzőkönyvezésre kerül a biztonság érdekében. Ezenkívül a rendszer valamennyi elemének működése is jegyzőkönyvezhető normál, részletes vagy hibakeresési szinten. Utóbbi esetben azonban olyan nagy mennyiségű adat keletkezik, amiért ezt tartósan nem célszerű alkalmazni.

Egyes fontosabb statisztikai adatok a monitorozó rendszer saját adatbázisa mellett vagy helyett a szolgáltató saját minőségbiztosító rendszerébe (PMS, Performance Management System) is küldhetők további feldolgozás és archiválás céljából.

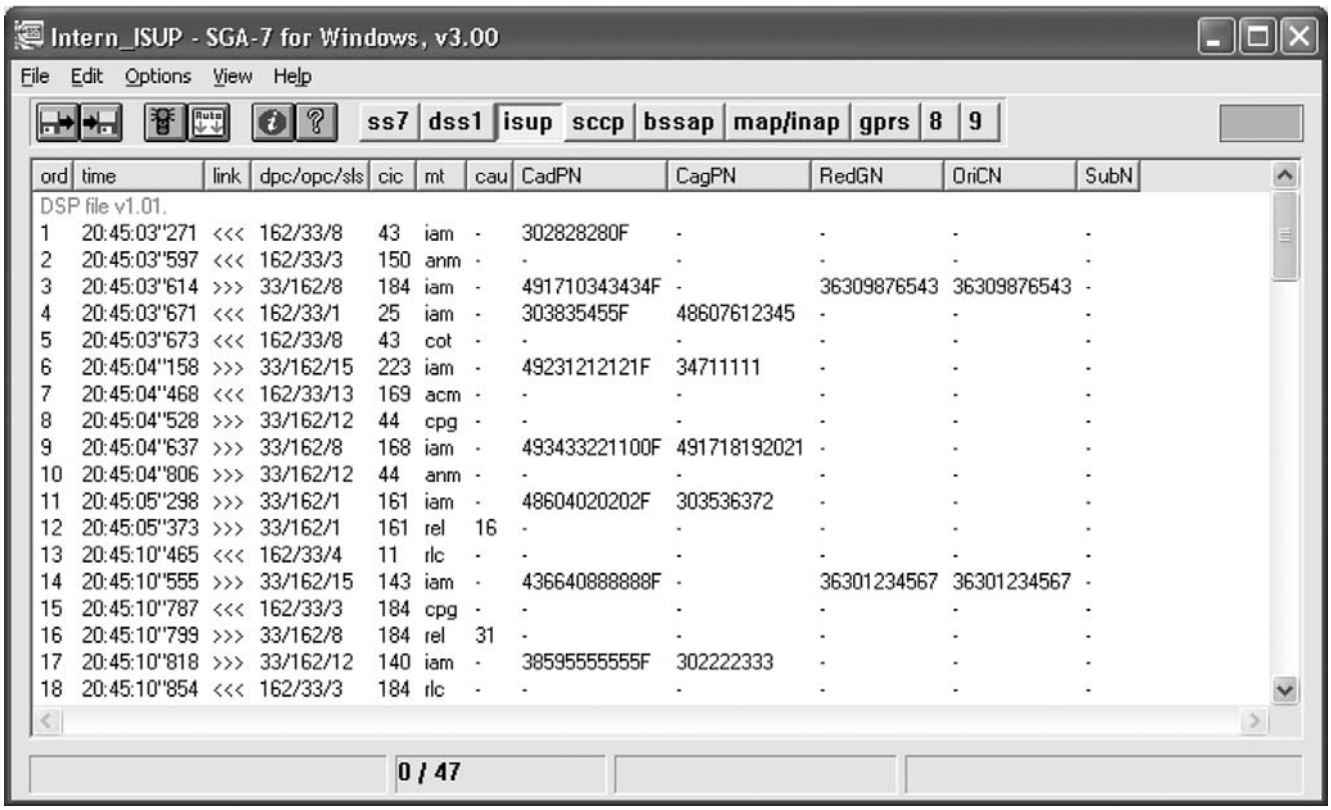
3. Főbb szolgáltatások

A jelzés-monitorozó rendszer igen sokoldalú eszköze lehet a fenntartó személyzetnek és a hálózattervezőknek. A legfontosabb általános funkciók az alábbiak:

Távmonitorozás

Ez az alkalmazás hasonló a kezelő személyzet által végzett protokoll analizátoros vizsgálathoz, de anélkül, hogy a helyszínre kellene utazni. Ezenfelül egy moni-

2. ábra Üzenet dekóder főablak



torozó hálózat számos többletszolgáltatást is tud nyújtani. Egyszerre számos linkre lehet csatlakozni és megfelelő üzenettárolási lehetőségek esetén, időben visszamenőlegesen is lehetőség van részletes vizsgálatokra éppúgy, mintha „élő” forgalom lenne. Lényeges megjegyezni, hogy különálló protokoll-analizátorokkal részben sem lenne pótolható a monitorozás, mert egyetlen hívás üzenetei is számos linken haladhatnak, amelyek csak elosztott monitorozással és központi feldolgozással értékelhetők ki.

Távmonitorozásnál a hálózati terhelés csökkentése érdekében a monitorok célszerűen közvetlenül a felhasználók kliens gépének küldik a jelzésüzeneteket, a szerver csak a jogosultság ellenőrzést végzi. Ezenkívül fontos az adatmennyiség csökkentése szűréssel, azaz csak azok az üzenetek kerüljenek továbbításra, amelyek kiválasztott paraméter-értékekhez tartoznak, ilyenek a Service Information Octet (SIO), üzenet típus, pontkód (OPC/DPC), áramkör azonosító kód (CIC). A vizsgálatokat megkönnyíti, ha a hívásrekordok alapján automatikusan is indítható az adott hívások távmonitorozása. Egyébként nagy forgalom esetén nehéz lehet egy-egy konkrét üzenet megtalálása, ha az időpont nem ismert.

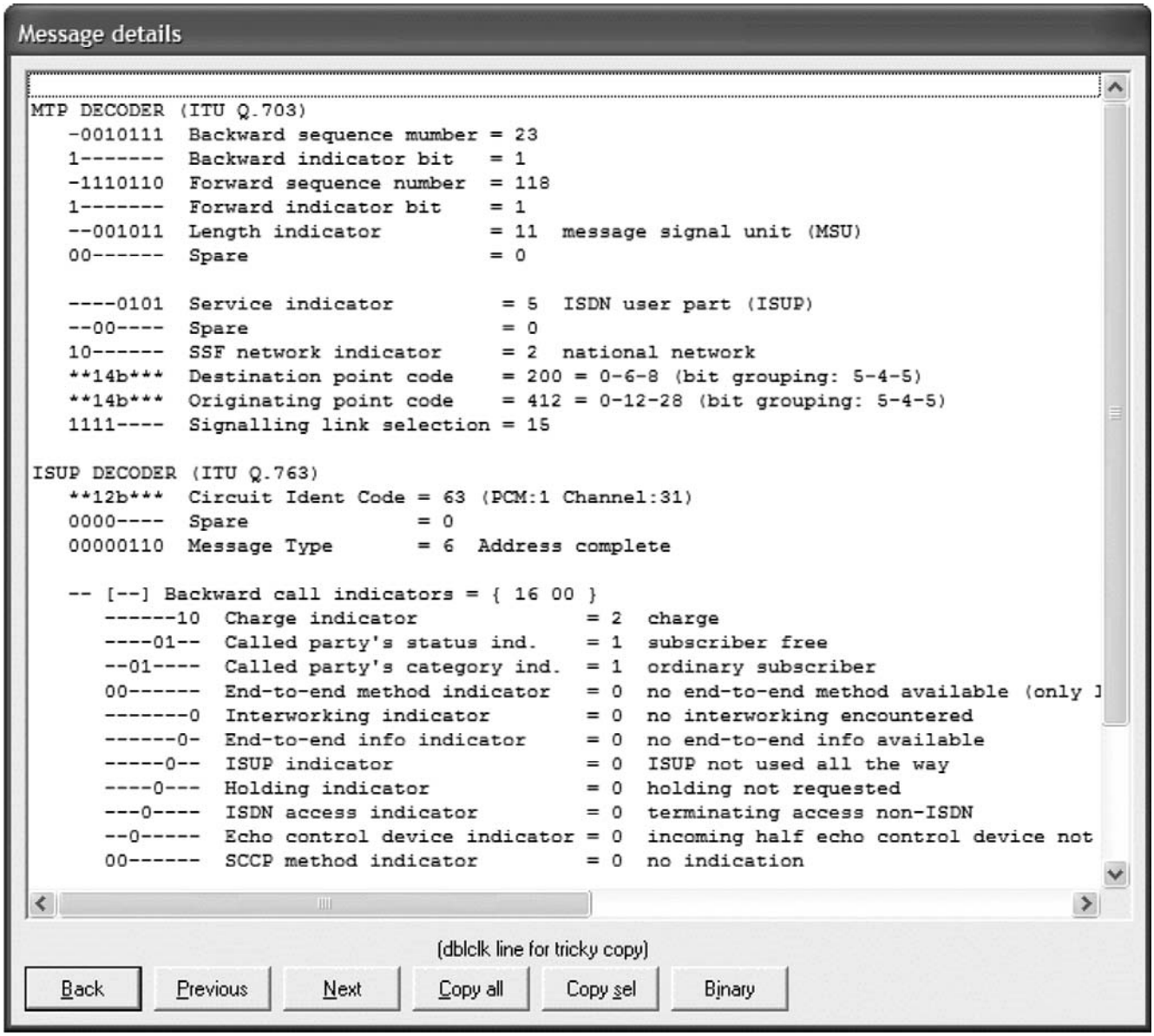
Dekódolás

A kliens gépen a távmonitorozással kapott üzenetek kívánságra dekódolhatóak és számos kijelzési/keresési kényelmi funkció segíti a kezelő személyek számára a manuális vizsgálatokat. A *2. és 3. ábrákon* egy tipikus dekódolási ablak, valamint a kiválasztott üzenet teljes dekódolásának részlete látható.

Statisztika készítés

A fizikai szint hibáitól az összetett üzenet- és hívásstatisztikákig számos adatot lehet rendszeresen és automatikusan gyűjteni, amelyeknek a vizsgálata működési problémákra és tendenciákra deríthet fényt, és jelzi a szolgáltatások színvonalát, a hívások és tranzakciók si-

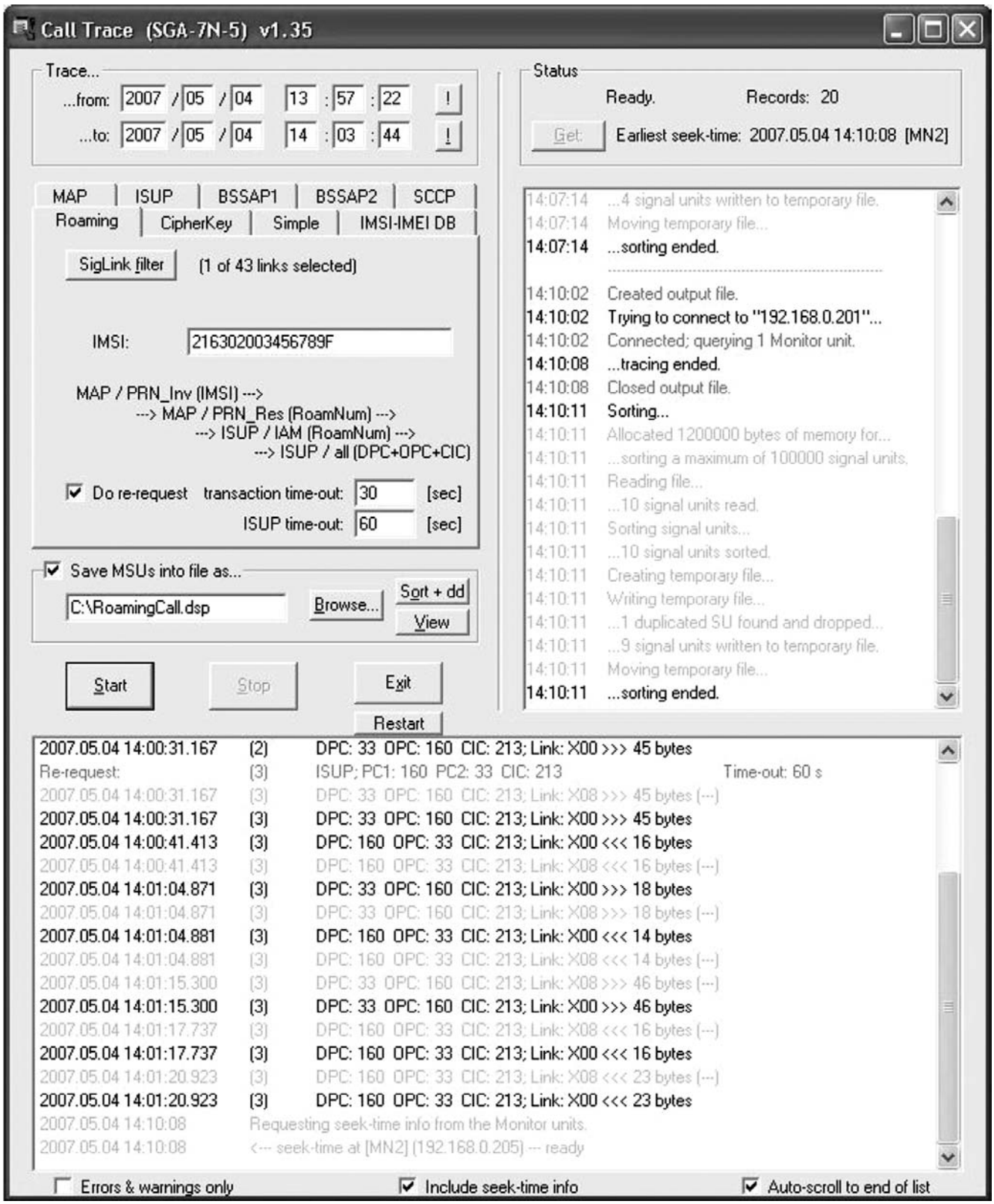
3. ábra Teljes üzenet dekódolás részlet



kerességét. Az ajánlások több, mint 150 különböző esemény és üzenetfajta számlálását javasolják [4,5]. A szerver periódikusan (tipikusan 5-15 percenként) lekérdezi és adatbázisba rendezi a statisztikai adatokat, amelyek fontos információt hordoznak a forgalom eloszlásáról, a hálózati szűk keresztmetszetekről és időben figyelmeztethetnek kritikus tendenciákra, mielőtt azok súlyosabb problémákat okoznának.

A statisztikakészítés nemcsak jelentős memóriaigényt támaszt, de a számítási kapacitás szempontjából sem elhanyagolható, hiszen az üzeneteket részben dekódolni kell, legalább az üzenettípus meghatározásához. Emel-

4. ábra Híváskövetés főablak



lett minden statisztikai számlálóhoz küszöbszinteket kell definiálni, amelyek adott időn belüli túllépése figyelmeztető eseményt kelt, hogy már a részletes elemzés előtt, szinte azonnal észlelhetők legyenek a kedvezőtlen tendenciák. A küszöbök egyedileg állíthatók helyileg és távolról is.

Hívásrekord készítés

Az egyes hívásokhoz, rövid üzenetekhez (SMS) vagy egyéb tranzakciókhoz tartozó jelzésüzenetek gyűjtésével rekordok készíthetők (CDR, Call Data Record vagy xDR: egyéb Data Record). Ezek adatbázisba rendezve visszamenőleg is alkalmasak hibakeresésre, statisztika készítésre vagy egyéb vizsgálatokra, például a korábban említett távmonitorozás vagy a híváskövetés automatikus indítására. A monitorok által gyűjtött, lényeges adatok összerendezésével a szerver készíti a rekordokat, amelyeket adatbázisba tölt. Egy-egy CDR az adott híváshoz tartozó minden lényeges adatot tartalmaz, beleértve az időpecséttel ellátott hívásüzeneteket, bontási okot (normál, foglalt, nem válaszol stb.), hívó és hívott számokat, pontkódokat, áramkörazonosítót, a használt linkek azonosítóját. Ezen adatok alapján számos hasznos alkalmazás is létrehozható, például hibakeresés, számlaellenőrzés, csalások detektálása, a sikeres és sikertelen hívások statisztikája.

Híváskövetés

Az egyik leghatékonyabb módja a hibakeresésnek és a folyamatok követésének az egy-egy híváshoz tartozó jelzésüzenetek összegyűjtése. Ebben az esetben az üzenetek tárolása lehetővé teszi azt is, hogy mind valós időben, mind a tárolási időtartamig visszamenőlegesen azonos módon lehessen hívásokat analizálni. A híváskövetés a felhasználói kliens gépekről indítható a hívó vagy hívott szám (MSISDN), az IMSI (International Mobile Station Identity), TMSI (Temporary Mobile Station Identity), vagy IMEI (International Mobile Equipment Identity) azonosítók alapján *(4. ábra)*.

Ezeket az elsődleges azonosítókat minden olyan monitor egység megkapja, amelynek linkjein a kívánt üzenetek előfordulhatnak és keresi a kezdő üzenetet, melyben ez az elsődleges paraméter előfordulhat. A működést nehezíti, hogy a későbbi üzenetek általában már nem tartalmazzák ezt a paramétert, ezért a kezdő üzenetből további, másodlagos paramétert kell kivonni, majd a szerver segítségével ezt is szét kell osztani a megfelelő monitorok között.

A hívás további üzeneteinek megtalálását már az olyan másodlagos paraméterek teszik lehetővé, mint a pontkódok és áramkör azonosító (OPC/DPC/CIC) az ISUP protokollnál, a helyi referenciák (Local References: SLR/DLR) a kapcsolat orientált SCCP protokollnál, mint a BSSAP, vagy a tranzakció azonosítók (OTID/DTID) a TCAP alapú protokollnál, a MAP és INAP a kapcsolat nélküli SCCP üzenet átvitelnél, de BSSAP esetén cella azonosítóra, roaming számokra vagy akár adott bájtokra is lehet keresni. Ehhez az összetett működéshez elengedhetetlen minden egyes üzenet dekódolása egész magas szintig, amíg az említett azonosítók megtalálhatók. Végül a híváskövetést kezdeményező felhasználó megláthatja a kiválasztott elsődleges paraméterhez tartozó teljes üzenet folyamot anélkül, hogy a fenti bonyolult működési háttérrel kellene foglalkoznia. A tesztelés érdekében különösen fontos, hogy ez a funkció gyakorlatilag valós időben működjön, és így folyamatban lévő hívások is követhetők legyenek, azonban órákkal vagy napokkal korábbi hívások is ugyanígy vizsgálhatóak.

Esemény kijelzés és riasztás

Átviteli vagy berendezés hibák, illetve a beállított statisztikai küszöbök túllépései eseményt keltenek és ezek esetén riasztás küldhető a fenntartó központnak. Így a hálózati berendezésektől függetlenül is észlelhetők a hibák, sőt gyakran a veszélyes tendenciák is. Az események tárolásán és utólagos értékelésén túlmenően a felhasználói állomáson is kijelezhetők a figyelmeztetések (sárga) és a kritikus hibaesemények (piros) sorokkal.

4. Kliens programok

A felhasználói gépen futó főbb kliens programokat az *5. ábra* mutatja. A felhasználó időpont, linkazonosító stb. adatok alapján kérheti le az adatokat az adatbázisból.

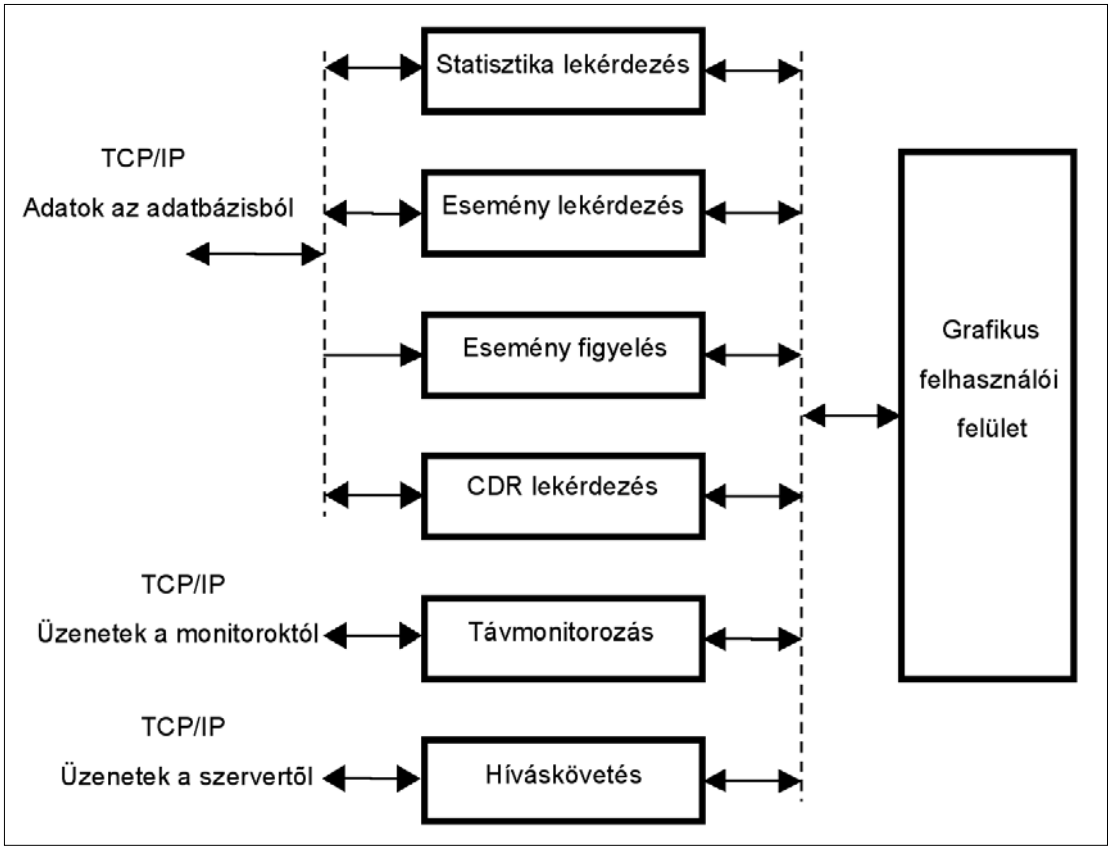
A távmonitorozó kliens alkalmazás közvetlenül a monitoroktól kaphatja a szűrt üzeneteket, így ebben az üzemmódban a szervernek csak a jogosultság ellenőrzés a feladata, miként ezt az összes többi kliens hozzáférést igénylő alkalmazásnál is biztosítani kell.

A híváskövetésben a szervernek már aktívabb szerepe van; ez osztja szét a monitorok felé az elsődleges és másodlagos paramétereket, majd a követés során összegyűjtött üzeneteket továbbítja a kliens géphez, ahonnan a követést indították.

A lekérdezések és az eredménykijelzések egyszerűen kezelhető, MS-Windows alapú grafikai felhasználói felületen történnek. A protokoll dekódolás eredménye, a hívásrekordok, a riasztási, statisztikai és híváskövetési információk egyszerű angol nyelvű kijelzése kevés gyakorlattal, könnyen értelmezhető.

5. Összefoglalás

A telefon- és adathálózatokat egy önálló, közös csatornás jelzéshálózat vezérli, amelynek megfelelő működése alapvető fontosságú a szolgáltatások folyamatos jó minősége szempontjából. Ennek a jelzéshálózatnak a monitorozásával a hálózat állapotára, a forgalom eloszlására, az egyes hívásokra, az esetleges hibákra és csalásokra nyerhetők adatok, amelyek a fenntartásához és a hálózat továbbfejlesztésének tervezéséhez egyaránt fontosak. Lényeges megjegyezni, hogy különálló protokoll-analizátorokkal részben sem pótolható a monitorozás, mert egyetlen hívás üzenetei is számos linken haladhatnak, amelyek csak elosztott monitorozással és központi feldolgozással értékelhetők ki.



5. ábra
Főbb
kliens programok

A jelen cikk áttekinti a monitorozás főbb szempontjait és feladatait, valamint egy monitorozó rendszer kialakítását. A leírás a megoldásokat egy konkrét rendszer példáján keresztül mutatja be [6], de a főbb funkciók és követelmények más rendszerekre is vonatkoznak (pl. [1,7]).

A monitorozó rendszerek kialakításánál alapvető szempont a skálázhatóság, amely lehetővé teszi a hálózatok dinamikus növekedésének és a szaporodó szolgáltatásoknak a kezelését. Ugyanezen okból a monitorozó rendszert a hálózati berendezésektől függetlenül, azok többlet terhelése nélkül célszerű megvalósítani.

A teljesség igénye nélkül ismertettük a monitorozás által nyújtott főbb szolgáltatásokat: a távmonitorozást, hívás- és üzenetstatisztika-, valamint hívásrekordok készítését, híváskövetést, eseménykijelzést és riasztást. Miközben a statisztika és a hívásrekordok készítése igen nagy mennyiségű adat feldolgozását követeli meg, a híváskövetés és a távmonitorozás gyakorlatilag valós idejű működést igényel. Ezeknek a szélsőséges követelményeknek a teljesítése is megoldható a rendszer elemeinek és a feladatoknak a megfelelő elosztásával.

A monitorozó rendszereknek napi 24 órában folyamatosan kell működnie, miközben állandóan bővül a monitorozott hálózat mind kapacitásban, mind szolgáltatásban. Az elmúlt 5-10 évben a szolgáltatóknál tipikus volt a több, mint egy nagyságrenddel bővülő hálózati forgalom, amelyet kezelni kellett és a bővülés jelenleg is folytatódik, hasonlóképpen az új szolgáltatások fejlesztéséhez. Ezt a folyamatot hatékonyan segítik a jelzéshálózat-monitorozó rendszerek.

Irodalom

[1] G. Cooper,
"SS7 signaling monitoring systems,"
In Comm. Network Test & Measurement Handbook,
Chapter 33, C. F. Coombs and C. A. Coombs,
Eds. McGraw Hill, pp.761–786., 1998.

2. P. Tatai, Gy. Marosi, L. Osváth,
"A Flexible Approach to Mobile Telephone Traffic Mass Measurements and Analysis",
IEEE Instrum. and Measurement Technology Conf.,
Budapest, Hungary, 21-23 May 2001.
3. P. Varga, P. Tatai,
"Advanced Methods on GPRS Network Analysis,
10th Eunice Summer School & IFIP WG 6.3 Workshop,
Tampere, Finland, 14-16 June 2004.
4. ETSI GSM 12.04,
"Performance management and measurements for a GSM PLMN," 1993.
5. ITU-T Recommendation Q.752,
"Monitoring and measurements for signalling system No. 7 networks," 1994.
6. http://www.aitia.ai/telecom/products/ss7_signaling_network
7. http://www.gl.com/netsurveyor.html

Összefoglalás a hangtechnika és az akusztikai tudományos élet fórumairól

WERSÉNYI GYÖRGY

Széchenyi István Egyetem, Győr
wersenyi@sze.hu

A szakmai élet legfontosabb elemei a tudományos folyóiratok és konferenciák. Ez a cikk röviden bemutatja az akusztika és hangtechnika legfontosabb külföldi és magyar szaklapjait, konferenciáit, különös tekintettel az elterjedőben lévő on-line hozzáférésű médiumokra.

E cikk kivételesen mellőzi a tudományos témát és mélységet. Célja, hogy a hazánkban élő, dolgozó szakemberek megismerhessék a hangtechnikával és az akusztikával közvetlenül foglalkozó rendezvények, konferenciák lebonyolítását, a szakmailag is elismert folyóiratokat (különösen a terjedőben lévő on-line elérhetőségekre). Az áttekintés a legújabb, 2007-es állapotokat tükrözi.

Folyóiratok

Kezdjük a legfontosabbal, a tudományos folyóiratokkal. Többféle besorolás is létezik ezek osztályozására, ahol a fontosabb paraméterek közé tartozik a megjelenés helye (belföldi vagy nemzetközi), a megjelenés nyelve (magyar vagy idegen nyelvű), a lektorálás megléte (az ún. *peer review*), illetve az elismertsége.

A komoly folyóiratok nemzetköziek, a világ bármely országában hozzáférhetők, angol nyelvűek és kivétel nélkül lektoráltak. Tulajdonképpen ezt nevezzük tudományos folyóiratnak. A mérnöki tudományokban viszonylag ritka, az akusztika és hangtechnikai életben pedig különösen ritka az úgynevezett *impact factor*-os újságok megléte. Ez a mérőszám hivatott elvben megmondani, hogy egy adott újságban megjelenő cikknek mekkora lesz a „hatása”. Ezt elég nehéz objektíven mérni és mivel sok újság nem is adja meg, felesleges ennek különösebb jelentőséget tulajdonítani. Hasonlóan, a *citation index* is egy nehezen követhető és ebben a formában alkalmatlan mérőszám.

A lektorálás azonban nagyon fontos, ennek hiányát csak néhány konferenciánál fogadhatjuk el. Be kell látnunk, hogy mára a komolyabb konferenciák is csak lektorálás után fogadják el az előadásokat, mégpedig egyre gyakrabban a teljes cikk alapján (nem tekintjük lektorálásnak az absztrakt alapján történő döntést). A lektorálás nagy hátránya azonban a hosszas átfutás: előfordul, hogy egy évnél is több telik el a beküldés és a megjelenés között! Sajnos, néhány folyóirat azt gondolja, minél hosszabbra engedi bírálói idejét, annál nívósabb lesz az újság... Azonban ezeknek a lapoknak konkurenciái is vannak: a rohamosan terjedő on-line megjelenés. Ezek a lapok ugyanolyan szakmai színvonal mellett általában lényegesen gyorsabb (néhány hét vagy hónapos) átfutást ígérnek és mindenki számára elérhető PDF formátumú letöltést. Ezzel garantálják a cikk aktualitását is, hiszen egy éves nagyságrendben mérhető átfutás során sokat veszíthet egy cikk az újszerűségéből. Eljött az idő, amikor nem söpörhetjük szőnyeg alá ezeket a médiumokat!

Jegyezzük még meg, hogy a folyóiratok többsége (és a konferenciák is a regisztrációs díj formájában) pénzt szednek a megjelenésért. Ez részben érthető, hiszen vannak költségek, ugyanakkor az ár általában igencsak borsos, elérheti a több száz dollárt is. Hasonlóan, a konferenciák részvételi díja is a 400-500 Eurós nagyságrendben mozog!

Ezek után lássuk, mely szaklapok állnak rendelkezésünkre, hogy akusztikai, zajvédelmi, általános hang, illetve méréstechnikai tudományunkat megjelentessük.

Az **Applied Acoustics** az egyik legrégebbi folyóirat az akusztikában [1,2]. 1968 óta jelenik meg és komoly szakmai hírnévre tett szert azóta. A legismertebb lap az amerikai akusztikai társaság „nagy sárga könyve”, mely havonta jelenik meg. Ez a mindenki által csak JASA-nak hívott **Journal of the Acoustical Society of America** [3]. Elsősorban az amerikai kontinens szerzői képviseltetik magukat benne, de mindenképpen érdemes európai kutatónak is „elsütni” egy-egy cikket a hasábjain.

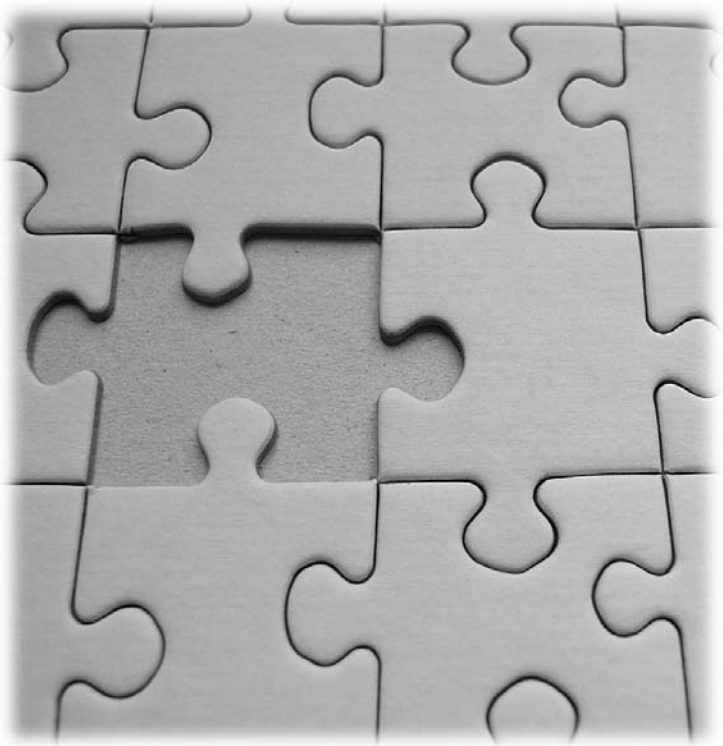
Hasonlóan amerikai központú az egész világ szakembereit magába foglaló Audio Engineering Society (AES). E szervezetnek van magyar tagozata is, együttműködve az OPAKFI-val. Éves tagdíj ellenében kaphatjuk a papír alapú vagy az on-line letölthető havi szaklapot, a **Journal of the Audio Engineering Society**-t [4,5]. Az ázsiai szakmai élet elsősorban tokiói központú japán tagozatában képviselteti magát. A röviden csak „japán AES” újságnak nevezett lap szintén havi és japán mellett angolul is közöl cikket. Létezik on-line és papír alapú verziója is (különböző ISSN szám alatt), jelenleg a legismertebb a **Journal of Acoustic Science and Technology** (AST) [6].

Európa sem marad el, a European Acoustic Association (EAA) lapja a nagynevű és híres **Acta Acustica united with Acustica** [7]. Ez korábban két különböző lap volt, de ma már egyben jelenik meg. Itt érdemes megjegyezni, hogy az Akusztikai Szemle (lásd később) 2001-2005 közötti számai megjelentek a Nuntius Acusticus CD-n, amit ehhez az újsághoz mellékeltek még 2005-ben.

Az IEEE is rendelkezik olyan *transaction*-nel, amely szakmánkhoz közel áll: az **IEEE Transactions on Signal Processing** és az **IEEE Transactions on Speech and Audio Processing** [8]. A beszédkutatás egyik legfontosabb lapja a **Speech Communication** [9] és néha megjelenik egy-egy cikk a **Physics Today**-ben is [10].

Ahogy korábban volt róla szó, a folyóiratok egy része csak on-line jelenik meg (a fenti lapok többsége rendelkezik on-line elérhető és letöltő szolgálattal, de papíron is megjelennek). Feltehetőleg a nagymúltú lapok soha nem fognak leszokni a papír alapú megjelenésről, de ez nem tartja vissza az on-line folyóiratokat a szerzők „elszipkázásától". Sajnálatosan még mindig sok az előítélet ezekkel a lapokkal szemben, elsősorban arra hivatkozva, hogy a papír nem vész el, de a honlapok megváltozhatnak, törlődhetnek. Ez a szemléletmód azonban lassan kezd megváltozni, így a lektorált, megbízható webes folyóiratok semmivel sem érnek kevesebbet, mint papíros társaik. Természetesen ez esetenként magával vonja a hivatkozások szokványos formájának elvetését is; néhány webes újságnál legfeljebb évfolyamok vannak, de számok, illetve oldalszámok nincsenek. Ilyenkor a hivatkozásban a cikk hosszúságát kell megadni oldalszámokkal [11]. Íme néhány on-line folyóirat, melyet érdemes virtuálisan lapozgatni:

Electronic Journal Technical Acoustics (EJTA) [12],
Scientific Journals International (SJI) [13],
On-line Journal of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering (ICST) [14].



Az ehhez hasonló lektorált lapokban történő megjelenést nem szabad csekélyebb értékűnek tekinteni csak azért, mert papíron nem adják ki. A könnyebb hozzáférhetőség (ezek az oldalak általában ingyen engedik letölteni a PDF formátumú cikkeket) és sokszor az olcsóbb megjelenési költség, valamint a gyorsabb átfutási idő egyenesen előnyösebbé teszi őket a papír formátumú lapokhoz képest.

A magyar szakirodalom kimondottan akusztikai vonatkozású lapja az **Akusztikai Szemle**. Több éves múltra tekint vissza, remek publikációs lehetőség a szükséges magyar nyelvű cikk megjelenítésére, de német és angol nyelven is érkezhet hozzájuk dolgozat. A megjelenés meglehetősen rapszodikus, évente általában 3-4 szám jelenik meg. A szélesebb mérnöki réteget megcélzó, hagyományos szakmai folyóiratunk a **Híradástechnika**, melyben akusztika és hangtechnika is helyet kap a távközlési és híradástechnikai témák között.

Részletesebb információk és kevésbé ismertebb lapok után kutatva érdemes ellátogatni az internetre [15].

Konferenciák

A szakmai megmérettetés másik módja a konferenciák látogatása. Itt 15-20 perces időtartományban szóban, illetve poszterrel jelenhetünk meg. A cikkek általában lektoráltak, de korlátozott tartományúak, tipikusan 4-6 oldalasak lehetnek.

A legnevesebb konferencia a **Forum Acusticum**, mely 3 évente kerül megrendezésre. 2005-ben Budapesten volt, 2008-ban Párizsban lesz. Szintén háromévenkénti az **International Congress on Acoustics** (ICA), amelyből már a tizenkilencedik lesz idén Madridban. Az ázsiai és óceániai kutatók seregszemléje is hároméves periodicitású, ezt legközelebb 2009-ben Pekingben rendezik **Western Pacific Acoustics Conference** (WESPAC) elnevezéssel.

A konferenciák többsége éves rendezésű. Ilyen az **InterNoise**, mely inkább zajvédelemmel és méréstechnikával foglalkozik. 1997-ben Budapesten tartottuk és jövőre Shanghai-ban már a 37.-et rendezik. A 29. sorszámnál tart az audiológiai konferenciája, az **International Congress on Audiology**. Ez is ICA rövidítést használ (2008 Hong Kong). Elsősorban a beszédkutatás konferenciája az **InterSpeech**, mely jövőre Brisbane-ben lesz. Van a beszédkutatásnak és az akusztikának közös konferenciája is: ez az **International Conference on Acoustics, Speech and Signal Processing** (ICASSP), legközelebb 2008-ban Las Vegasban. A már említett AES szervezet évente kétszer rendez úgynevezett *konvenciót*, ebből egy európai, az őszi pedig általában amerikai, leggyakrabban a new york-i központban. Hatalmas rendezvény, a sorrendben 124., jövő év tavaszán Amszterdamban lesz. A részvétel ezen jó „beugró" lehet egy cikkhez a folyóiratban! Az amerikai és a japán szervezetek is tartanak évente találkozót, a **JASA** és a **Japan AES meeting**-re várják az előadókat a világ minden tájáról.

A német nyelvterület uralkodó rendezvénye a **DAGA** (Tagung der Deutschen Arbeitsgemeinschaft für Akustik) melyet évente rendez meg a német akusztikai társaság

és a következő 2008-ban Drezdában lesz. Angol nyelvű előadással is lehet jelentkezni és aki németországi kapcsolatokkal rendelkezik vagy netán szüksége lenne rá, ne hagyja ki.

Még két eseményt szeretnék megemlíteni, melyek kisebb konferenciák, de rétegérdeklődésre számot tarthatnak: **International Conference on Auditory Display** (ICAD), illetve az **International Conference on Computer Graphics and Interactive Techniques** (SIGGRAPH). Előbbi 2008-ban Párizsban a Forum Acusticum szatellit rendezvénye, utóbbi pedig 2008 nyarán Los Angelesben lesz.

Tekintettel arra, hogy ezek a konferenciák mindig más helyszínen kerülnek megrendezésre, érdemes az interneten egyesével rákeresni, melyik mikor, hol lesz. Ne feledjük, hogy a határidők már egy évvel előbb kiírásra kerülnek! A fentiek többségénél ősszel indulnak a „call for papers” akciók és általában 2007 végén kell leadni az *abstract*-okat.

Szokjunk hozzá ahhoz is, hogy a konferenciakiadványok, az úgynevezett *proceedings*-ek mára szinte kizárólag CD-n jelennek meg, melyeken gyakran nem oldalszámozott PDF fájlok találhatók, ezért itt is módosul a megszokott hivatkozási forma. Ez a formátum továbbá lehetőséget ad arra, hogy a konferencia-cikk hosszát ne korlátozzák indokolatlanul. Ne lepődjünk meg, ha a megszokott néhány oldalas cikk helyett akár húsz oldalt is meghaladó irományt kapunk a PDF fájlban, ráadásul ott a lehetőség a CD-n hanganyagok elhelyezésére is, ami a papír formátumnál nem lehetséges.

Irodalom

[1] http://www.elsevier.com/wps/find/journaldescription.cws_home/405890/description#description
[2] http://www.sciencedirect.com/science/journal/0003682X
[3] http://asa.aip.org/jasa.html
[4] http://www.aes.org/
[5] http://www.opakfi.mtesz.hu/
[6] http://www.jstage.jst.go.jp/browse/ast/-char/en
[7] http://www.eaa-fenestra.org/
[8] http://www.ieee.org/web/publications/journmag/index.html
[9] http://www.elsevier.com/wps/find/journaldescription.cws_home/505597/description#description
[10] http://www.physicstoday.org/
[11] Wersényi, Gy.: Localization in a HRTF-based Minimum-Audible-Angle Listening Test for GUIB Applications. Electronic Journal of “Technical Acoustics” (EJTA), 2007/1. (16 oldal), http://www.ejta.org
[12] http://ejta.org/
[13] http://www.scientificjournals.org/
[14] http://www.icst.org
[15] http://www.acoustics.org/journals.html

Modelling the Inter-operation of high speed TCP protocols

Keywords: HSTCP, Scalable TCP, fairness inter-operation of transport protocols

Recently, new TCP protocols have been proposed to achieve better network utilization due to the poor performance of the AIMD based TCP Reno in high speed wide-area networks. Two promising suggestions are the HighSpeed TCP and the Scalable TCP. We have analysed both the inter- and intraprotocol fairness behavior of these versions by control-theoretic approach condisering the network as a feedback network and describing the interaction of the blocks via differential-equation systems. A MATLAB/Simulink environment has also been designed and implemented to solve the analytically not tractable differential equations by numerical approximations. The models have been validated by Ns-2 simulations. The results of our analysis help us to get a deeper understanding of the operation behavior of these new transport protocols.

IP-based network mobility

Keywords: Mobile IP, network mobility (NEMO), nested mobile networks, load sharing, QoS

Nowadays the mobile users become more and more dependent on data besides the traditional voice transmission. Regardless of using wired or wireless access, they would like to use all services. The IP-based Internet was designed for data transmission and has become the most ubiquitous wired internetwork, used by millions of people every day. According to these trends the next generation networks are designed as a combination of these two types of networks (mobile and IP-based). The IETF Mobile IP protocol handles mobility in the IP layer globally, but it is not well-adopted to some special scenarios, for example to mobility of hosts moving together. A typical example of such a mobile network is a network of IP-enabled devices in a vehicle. This problem is investigated by the Network Mobility Workgroup of IETF. In our paper we survey the results in the area of mobile network support starting from Mobile IP.

Repeated reconfiguration of multicast trees in multi-layer optical networks

Keywords: optical network, dynamic multicast, reconfiguration, ILP, heuristics

The paper deals with dynamically changing multicast trees in two-layer optical networks. When leaves permanently change, the tree differs more and more from the optimal topology. Therefore a repeated reconfiguration of the tree, when the otpimal topology is reconstructed, can save network resources and costs. The paper investigates the efficiency of the reconfiguration for several dynamic routing algorithms and as a function of the length of the reconfiguration interval.

Summaries • August 2007

New generation anonymous browsers

Keywords: WWW, browsers, anonymous browsers, paradigm

The World Wide Web presents data protection issues for its visitors, too: some service providers can observe the activity of users, can track them and can build databases from their customs. Anonymous browsers offer a solution for the users, they can hide them from potential observers. This paper presents some methods used for tracking, a new paradigm regarding anonimyzing services, and a new classification system for anonymous browsers.

WLANpos: Wi-Fi based indoor positioning system

Keywords: wireless networks, Wi-Fi, location-based services, positioning, algorithm

Wireless networks are more and more common these days thanks to the constant development in wireless technology. They can even compete with wired networking services in terms of throughput and reliability. While using the wireless network, the user can move freely, he or she can use the network in any location. This is the reason why services based on the position of the user have evolved. The need for a user positioning system has also emerged, which can be used indoors, and provides the accuracy needed for the services based upon it. Our goal was to infer the location of the user as accurate as possible, given a Wi-Fi network and a standard Wi-Fi receiver. There are already some existing solutions for this problem, but most of them are either expensive, need high computing capacity or are only usable in a restricted area. WLANpos was developed at the Department of Telecommunication and Media Informatics to provide the solution that meets the expectations best. It is a complete application capable of viewing and drawing maps and of course pointing out the results on the maps.

Watermarking of H.264 coded video streams

Keywords: video watermarking, H.264, NCG

In this paper we summarize the flavors of video watermarking. Then we present a new video watermarking method, which is robust to H.264/AVC compression and the most common signal processing modifications.

Incentives framework for voluntary autonomous cooperation in distributed networks

Keywords: ambient networks, voluntary cooperation, game theory, peer-to-peer networks, distributed networks, promise theory

Today's communication networks become dynamic which means that such networks do not have infrastructure or the configuration of the infrastructure-based networks constantly changes. These networks have high degree of autonomy, and they often behave in a selfish way. Autonomy means that such networks do not have any central administration or management principle that would determine the functioning of the network. To eliminate selfish behaviour from the network, a distributed framework has to be defined, that incites network nodes to communicate and cooperate. This can be done in various ways, this field has been widely explored by the research community, especially since peer-to-peer file sharing networks became popular. Our solution differs from these approaches in the existence of network topology. As each node is able to directly communicate only with its neighbours, a new way has to be found to motivate nodes to cooperate with each other. In this paper we describe a novel framework to solve this problem.

Security API analysis with the spi-calculus

Keywords: hardware security module, formal methods, process algebra, security, confidentiality

API level attacks represent a serious risk for Hardware Security Modules, therefore, it is important to discover and patch security vulnerabilities in APIs. A promising approach in this direction is to use formal verification methods. In this paper, we follow this approach and propose an API verification method based on a process algebra that seems to be extremely well-suited for the modelling of the operation of security APIs, for the definition of the relevant security requirements, as well as for the verification of whether those requirements are satisfied or not. In order to motivate our work, we also describe some specific API attacks against a security module that is widely used in practice.

Monitoring signaling processes in telecommunication networks

Keywords: non-intrusive network monitoring, GSM, GPRS, remote monitoring, traffic statistics, call tracing, SS7 protocol decoding

The dynamic expansion of networks necessitates highly scalable distributed data collection and storage and, at the same time, the correlation of the data can only be realized by central information processing. In a typical network the amount of collected information amounts to hundreds of Gigabytes within days. Therefore the processing and the concise presentation of data for the users is extremely important. This article describes non-intrusive monitoring of networks and presents a way to meet the contradictionary requirements of high volume data collection and real-time protocol analysis and call tracing.

Overview of journals and conferences related to acoustic and audio engineering

Keywords: conference, acoustics, journal

This paper introduces the most important Hungarian and international journals (paper and on-line) and conferences related to acoustics, speech communication, signal processing and audio technologies.

Summaries • of the papers published in this issue

Contents

Szerkesztőség
HTE Budapest V., Kossuth L. tér 6-8.
Tel.: 353-1027, Fax: 353-0451, e-mail: info@hte.hu

Hirdetési árak
Belív 1/1 (205x290 mm) FF, 120.000 Ft + áfa
Borító II-III (205x290mm) 4C, 180.000 Ft + áfa
Borító IV (205x290mm) 4C, 240.000 Ft + áfa

Cikkek eljuttathatók az alábbi címre is
Szabó A. Csaba, BME Híradástechnikai Tanszék
Tel.: 463-3261, Fax: 463-3263
e-mail: szabo@hit.bme.hu

Előfizetés
HTE Budapest V., Kossuth L. tér 6-8.
Tel.: 353-1027, Fax: 353-0451
e-mail: info@hte.hu

2007-es előfizetési díjak
Közületi előfizetők részére: bruttó 32.130 Ft/év
Hazai egyéni előfizetők részére: bruttó 7.140 Ft/év
HTE egyéni tagok részére: bruttó 3.570 Ft/év

Subscription rates for foreign subscribers:
12 issues 150 USD,
single copies 15 USD

www.hte.hu
Felelős kiadó: NAGY PÉTER
Lapmenedzser: DANKÓ ANDRÁS

HU ISSN 0018-2028
Layout: MATT DTP Bt. • Printed by: Regiszter Kft.