Tartalom

Címlap: A torlódás nem csak a távközlésben okoz problémákat

Hiány és sorbanállás

lajtha.gyorgy@ln.matav.hu

Már csak a legöregebbek emlékeznek a háborús évekre, amikor az alapvető élelmiszerekért sorba kellett állni. Volt, akinek jutott, mások megunva a várakozást eredmény nélkül távoztak. Később már csak déli gyümölcsből és néhány különleges csemegéből volt kisebb a kínálat, mint a kereslet. Sajnos a telefon iránti igényeket csak a rendszerváltás után sikerült teljesen kielégíteni.

Ekkor azonban felmerült a kérdés, szabad-e gyorsan, olcsón, azonnal valamennyi igénylőnek állomást adni? Ugyanakkor nem lehet megfelelő minőséget garantálni. Szabad-e a telefonálási igényeket tarifális módszerekkel korlátozni? Nem volt szükség egy fél évtizedre és ezek a problémák is megoldodtak. A tarifák kevésbé emelkedtek, mint az infláció és a világ minden része elérhető volt automatikus kapcsolással. A nagytávolságú összeköttetéseket fényvezetőkkel építették ki, így elegendő kapacitás állt rendelkezésre a gyors és jó minőségű kapcsoláshoz.

Mindezek ellenére a torlódás-elmélet és a torlódások számítása továbbra is a szakmai kutatások előterében maradt, bár sokszor leírtuk már, hogy a fényvezetők szinte korlátlan kapacitása valamennyi távközlési igény kielégítésére elegendő, és a korszerű, nagykapacitású irányító-kapcsoló eszközök sem korlátozzák az átviendő információk mennyiségét.

Az igények megjelenése kiszámíthatatlan és a hibák megjelenése váratlan lehet. Azok a rendszerek, melyek 1-2 százalék veszteségre tervezve általában tökéletesen kielégítik az igényeket, azok a körülmények szerencsétlen összejátszása esetén éppen a legfontosabb kapcsolatok kiépítésére lesznek alkalmatlanok. Egy-egy természeti katasztrófa, vagy váratlan politikai esemény oly mértékben megnövelheti a forgalmat, hogy a bőségesen rendelkezésre álló kapacitások sem tudják kielégíteni az igényeket. Hasonlóképpen váratlan torlódásokhoz vezethet egy összeköttetés megszakadása, vagy elektronikus eszköz kiesése.

A hálózatnak ilyen körülmények között is teljesíteni kell feladatát, sőt talán ezek azok a körülmények, amikor a felhasználók leginkább rászorulnak a távközlés segítségére. Ezt még nagyobb túlméretezéssel elérni gazdaságtalan lenne. Feladatunk tehát az, hogy valamilyen módon a torlódást elkerüljük, felhasználva a hálózatban másutt rendelkezésre álló szabad kapacitásokat. Megengedhető esetleg, hogy ilyenkor a hívások néhány másodpercig sorban álljanak, de bizonyos segélykérő, üzemirányító hívásoknak még az átlagosnál is gyorsabban célhoz kell érniük.

Ez indokolja a különböző forgalom-menedzselő eljárások fejlesztését. Ezek egyrészt hívásengedélyezési módszereket alkalmaznak, ahol prioritásokat adhatunk meg és így az adott irányban a legsürgősebb információk átvihetők. A prioritásos módszerek mellett érdemes végig gondolni a forgalomirányítás rugalmas megoldásait is, melynél a kisebb terhelésű szakaszokat is fel lehet használni, mint kerülő utat, a sorban álló, túlcsorduló forgalmak átvitelére. A többutas forgalomirányítás is jó lehetőségeket ad.

Ezen számunkban igyekszünk módszereket bemutatni arra nézve, hogy a bőségesen rendelkezésre álló átviteli lehetőségek a váratlanul fellépő, a tervezett értéket messzemenően meghaladó igények lebonyolítására hogyan válnak alkalmassá. Tehát összességében nincs hiány átviteli útban, vagy kapcsoló eszközben, csak a kijelölt irányok vannak túlterhelve, és ott kell a pillanatnyi látszólagos hiányokat sorbanállás nélkül, gyorsan megszüntetni. A cikkek ennek megfelelően a hívásengedélyezéssel, az útvonalválasztással, a forgalomeloszlás jellemzőinek felhasználásával foglalkoznak, bemutatva ezek hasznosítását az esetenként előforduló nagyobb forgalom kielégítésére.

A bevezető írás is kapcsolódik ehhez a témához, mert a hálózatok használhatóságának és megbízhatóságának fogalmait tisztázza, és elméleti hátterét adja meg.

Dr. Lajtha György

A rendszer-megbízhatóság műszaki tervezése

DR. BALOGH ALBERT

albert.balogh@axelero.hu

Egy rendszer megbízhatóságát számos egymással kölcsönhatásban lévő tényező határozza meg. Ezek közül a legfontosabbak a következők: alkotó elemek, alkatrészek, a rendszer szoftverje, az emberi hatások, az üzemeltetési profil, a rendszer üzemeltetésével kapcsolatos szolgáltatás minősége, a környezet. A Nemzetközi Elektrotechnikai Bizottság (IEC) 56. Műszaki Bizottsága (TC 56) javaslattervezetet dolgozott ki a rendszer megbízhatóságának műszaki tervezése során alkalmazandó irányelvekre. A jelen közlemény ezen útmutató [1] alapján foglalja össze a rendszer-megbízhatóságra ható tényezőket és azokat a módszereket, amelyekkel a kitűzött megbízhatósági célok elérhetők.

1. Bevezetés

Az irányelvek széleskörűen alkalmazhatók különböző rendszerekre, így például távközlési-, szállítási-, termelési rendszerekre. Az ismertetés kitér a szervezési és műszaki kérdésekre is, valamint arra is, hogy ezek hogyan kapcsolódnak a megbízhatósági célok elérésére alkalmazott módszerekhez és eszközökhöz. Különbséget kell tenni a szervezési és irányítási (menedzselési) tevékenységek között attól függően, hogy milyen hatást gyakorolnak a rendszer megbízhatóságára.

A rendszert az ISO 9000:2000 szabvány [2] a következőképpen határozza meg: „Egymással kapcsolatban vagy kölcsönhatásban lévő elemek összessége adott cél elérésére."

A rendszert hierarchikus felépítésűnek tekintik, így egy termék lehet rendszer vagy részrendszer (alkatrész, elem) attól függően, hogy milyen szinten vizsgáljuk a rendszer leírását, azaz a terméket részeire bontjuk (rendszer) vagy sem (elem). A rendszer tervező első lépése az, hogy meghatározza rendszer feladatait és azokat lebontsa részrendszerekre.

2. A rendszer-megbízhatóság műszaki tervezésével kapcsolatos fogalmak

2.1. Általános alapelvek

A tervezése arra irányul, hogy egy vagy több adott célt és rendeltetés szerinti feladatot teljesítsen a rendszer. A rendszer-megbízhatóság ugyan a rendszer belső eredetű jellemzője, mégis csak akkor értelmezhető, ha adott cél elérésére vonatkoztatjuk. A rendszer megbízhatósági elemzését azzal célszerű kezdeni, hogy egyértelműen meghatározzuk a különböző rendeltetés szerinti feladatokat és azok teljesítésének feltételeit.

Ezeket a feladatokat rendszerint a követelmények szabványában adják meg. Ezen túlmenően vannak magától értendő teljesítendő tulajdonságok is. A rendszer állapota akkor hibás, ha megkövetelt funkcióinak legalább egyikét nem tudja ellátni. A rendszer fejlesztési projekt egyik legfontosabb célja olyan rendszer kidolgozása, amelynek minősége és megbízhatósága megfelel a követelményeknek.

2.2. Az alkotó elemek közötti kölcsönhatások

A rendszer megbízhatósága nemcsak alkotó elemeinek megbízhatóságától függ, hanem az azok között fellépő kölcsönhatásoktól is. Például a tartalékolás megbízhatóságot növelő pozitív hatás, ugyanakkor több alkatész együttes, egymástól függő meghibásodása csökkenti a rendszer megbízhatóságát. Ha egy bonyolult rendszert vizsgálunk, akkor nem elegendő figyelembe venni az egyes alkatrészek megbízhatóságát, hanem azt is számításba kell venni, hogy azok kölcsönhatása hogyan befolyásolja a rendszer-működését. Például az emberi beavatkozás miatt kialakult tervezési hibák üzemeltetési hibához vezethetnek.

Annak elérésére, hogy megbízható rendszert fejlesszünk ki és értékelni tudjuk, hogy az megfelelő-e vagy sem, mennyiségi megbízhatósági célokat kell kitűznünk. Ezen célok teljesítését ellenőriznünk kell a megcélzott alkalmazási körülmények között. A megbízhatóság gyűjtőfogalom, amely több tulajdonságra vonatkozik, ezért önmagában nem számszerűsíthető. A mérőszámok azonban az egyes tulajdonságokra meghatározhatók, melyeknek szintje függ attól, hogy milyen követelményt támasztanak azokkal szemben. Például a veszélyes folyamatok ellenőrző rendszerének az átlagosnál nagyobb használhatósággal és hibamentességgel kell rendelkeznie.

A részrendszerek között úgy kell felosztani a megbízhatósági követelményeket, hogy azok megfeleljenek az egész rendszerre előírt megbízhatósági követelménynek, de legolcsóbban legyen realizálható. Fontos figyelembe venni, hogy a rendszer megbízhatósága változik az üzemeltetési környezettől függően. Ez hat azokra az igénybevételi feltételekre, amelyek között a rendszer és részegységei, alkatrészei működnek. Ebből adódik, hogy a környezet befolyásolja a különböző meghibásodások előfordulási valószínűségét is.

2.3. A szoftver fontossága

A korszerű rendszerekben döntő jelentőségű szerepe van a szoftvernek a tervezés és az üzemeltetés során. A szoftver ezért egyre lényegesebb megbízhatóság szempontjából is. A rendszer megbízhatóságát összes alkotó eleme közötti kölcsönhatás nagymértékben befolyásolja, ezért nem szabad a szoftver megbízhatóságát elkülönítve elemezni, vizsgálni és értékelni. Különösen távközlési berendezésekben meghatározóak a szoftverek.

Az IP alapú hálózatok működését meghatározó routerek, bridgek, switchek és szerverek szoftverjeinek megbízható működésére kiemelt figyelmet kell fordítani. Itt a hibák felderítése hosszadalmas, sok futtatást igénylő feladat. Ezek vizsgálata tudományos körültekintést igényel [5,6,7,8].

2.4. Emberi hatások

Az emberi beavatkozás hatásait két szempontból kell vizsgálni:

- Azok az emberi beavatkozások, amelyek nem az üzemeltetés során éreztetik hatásukat. Ilyen tevékenységet végeznek a tervező mérnökök és a menedzserek.
- Azok az emberi beavatkozások, amelyek közvetlenül befolyásolják a rendszer működését a rendszer üzemeltetése és karbantartása során.

Az emberi beavatkozások részletesebb csoportosítása a következő:

a) Beavatkozás a rendszer ember-gép kapcsolat során.
b) Beavatkozás a hálózaton keresztül (például egy másik rendszer ember-gép kapcsolatából kezdeményezték vagy a távközlési hálózat különböző pontjain végzett munkák hatása).
c) Beavatkozás, amely fizikailag a környezetből történik, eltérően az ember-gép kapcsolatból származó beavatkozásoktól.

Az ember-gép kapcsolatra a következő megbízhatósági követelmények vannak:

- Legyen védelem a rendszer illegális elérése és az illegális belépés ellen.
- Legyenek világosan érthető felhasználási utasítások.
- Legyen könnyen megvalósítható, magas szintű interaktív kapcsolat az ember és a gép között (könnyű kezelhetőség, üzembiztonság, a hibás bemeneti jelek megakadályozása, rövid idő alatti helyreállítás emberi hibák esetén.

A rendszer megbízhatósága szempontjából lényeges az emberi beavatkozás hatása, ugyanakkor alkatrész-megbízhatóság szempontjából kevésbé fontos. Az emberi tényező megbízhatóságát nem szabad az ember hibamentes tevékenységére korlátozni.

2.5. Üzemeltetési profil

Az üzemeltetési profilt a konfigurációk (kiépített kapcsolatok a rendszerben), a működési állapotok és a környezeti feltételek összessége határozza meg. Ezek a tényezők szükségesek a rendszer funkciónak ellátásához. A rendszer üzemeltetési profiljának elemzése a rendszerkövetelményekre és a kereskedelmi vonatkozásokra is kiterjed.

2.6. A szolgáltatás-minőség (QoS) és a rendszer-megbízhatóság kapcsolata

A szolgáltatás-minőség (Quality of Service, QoS) a rendszer általános teljesítmény mutatója. A QoS és a rendszer-megbízhatóság egymást átfedő fogalmak. Kapcsolatukat ismerni kell és ez alapvetően fontos a rendszer-megbízhatóság műszaki tervezése során. Az ITU-T Recommendation E. 800 [3] határozza meg ezt a kapcsolatot.

Ez látható a 6. pont *3. ábráján,* amely megadja a megbízhatósági és szolgáltatás-minőségi jellemzők négyszintű integrált tartományát, ahol a 3. szint jeleníti meg az átfedés területét.

2.7. A környezet hatása

A rendszer-megbízhatóságot mindig jelentősen befolyásolja a környezet. Ezért a rendszer műszaki előírásaiban meg kell határozni az üzemeltetés környezeti feltételeit, melyeket a kockázat és a megbízhatóság elemzése során figyelembe kell venni. A környezeti megfontolásokat érvényre kell juttatni az elavulás és a termék selejtezése szakaszában is.

Az elemzési módszerek összehangolása során a különböző elemzési módszereket, mint például a hibafa-elemzés (FTA), a meghibásodási mód, -hatás és -kritikusság elemzése (FMEA), a veszélyhelyzeteket és üzemeltethetőség vizsgálatokat (HAZOP) és a megbízhatóság előrejelzését, együtt kell áttekinteni.

Konfiguráció-menedzsment (rendszerek közötti kapcsolat kiépítésének irányítása) és menedzselése a megbízhatóság egyik legfontosabb kérdése. Példa erre a karbantartást ellátó szervezet létrehozása és rendszerhez való kapcsolása. Ezek a különböző konfigurációk által előállított kapcsolatok nagyon változatosak. Közvetlen kereskedelmi hatása van például annak, hogy az alkatrészek felcserélhetőségére és a rendszerek együttes működtetésére vonatkozóan egyre szigorúbb követelmények vannak. Ez különösen igaz hosszú élettartamú (például távközlési) rendszerekre, amelyeknek gyorsan avuló alkatrészei vannak, vagy amelyeknél technológiai változtatásokat végeznek. Ugyanakkor a hálózat különböző részeiért más üzemeltetők felelősek.

Teljesítőképesség (műszaki kapacitás) azt jelenti, hogy megfelelően méretezett erőforrások állnak rendelkezésre a rendszer által megkövetelt bármely szolgáltatási igény teljesítésére (például a karbantartó szervezet munkaerő és kapacitása). Ezek befolyásolják a szolgáltatás elérhetőségét és folyamatosságát. Ajánlatos a hibák súlyozása az általuk okozott rendszer-teljesítőképesség csökkenésének mértéke szerint.

A teljesítőképesség csökkenését az egész teljesítőképesség százalékában kell megadni. Itt kell figyelembe venni a forgalmi méretezést [9,10].

3. A megbízhatósági képességek (tulajdonságok) és azok mennyiségi jellemzői

A megbízhatósági fogalmakat az MSZ IEC 50(191) szabvány [4] határozza meg. A megbízhatósági képességekhez mérőszámokat rendelnek hozzá. A *megbízhatóságot meghatározó képességek* (használhatóság, hibamentesség, karbantarthatóság, karbantartás-ellátás, valamint a [4]-ben nem szereplő alkalmazhatóság, biztonság, adatbiztonság, hatékonyság) a felhasználónak azt az elvárását tükrözik, hogy káros események (meghibásodások, incidensek) milyen valószínűséggel fordulnak majd elő az üzemeltetés során.

A következőkben tárgyalandó fogalmak esetében az esemény lényeges voltát annak okai, módja, hatása és következménye szigorúsági fokozata határozza meg. Csoportosításuk a következő:

- Ha a meghibásodást nem tervezett (szándékos) hiba idézi elő és nem jelent veszélyt emberéletre és vagyonbiztonságra, valamint a környezetre, akkor „közönséges" meghibásodást jelent és a *hibamentesség* szempontjából lényeges.
- Ha a hiba a felhasználónál jelentkezik, amikor az igénybe akarja venni a szolgáltatást és ennek következtében a felhasználó képtelen a feladatok végrehajtására, akkor a meghibásodás az *alkalmazhatóság* szempontjából lényeges.
- Ha a hiba az élet- és vagyonbiztonságot veszélyeztető kritikus meghibásodást eredményez, akkor a *biztonság* szempontjából lényeges.
- Ha a hiba szándékos rendszer elleni támadást jelent és veszélyezteti az információbiztonságot, akkor a meghibásodás az *adatbiztonság* szempontjából lényeges.

A következőkben ismertetjük ezen meggondolások alapján a megbízhatósággal kapcsolatos fogalmakat a [4] szabvány alapján.

3.1. Megbízhatóság (Dependability)

Gyűjtőfogalom, amelyet a használhatóság (üzemkészség, rendelkezésre állás) és az azt befolyásoló tényezők, azaz a hibamentesség, a karbantarthatóság és a karbantartás-ellátás leírására használnak.

Ez a fogalom a termék időtől függő képességeit öleli fel, mennyiségi leírásra nem használható.

3.2. Hibamentesség (Reliability)

A terméknek az a képessége, hogy előírt funkcióit adott feltételek között, adott időszakaszban ellátja.

Mennyiségi jellemzői: *az R(t) hibamentes működési valószínűség (túlélési valószínűség), amely megadja annak valószínűségét, hogy a termék a t időpontot túléli, meghibásodási valószínűség, a $\lambda(t)$ meghibásodási ráta, meghibásodások közötti átlagos működési idő.*

A működési idő nem azonos általában a naptári idővel. A működési időt különböző mértékegységekben mérhetik (gépkocsi esetében a megtett km-ek számával, repülőgépek esetében a repülési órák számával, a távközlésben a sikeres kapcsolás felépítések számával. Szoftverek esetén a végrehajtási idővel, a programok előhívási számával, a végrehajtott utasítási ciklusszámmal lehet mérni a működés tartamát).

Az átlagos működési időt az első meghibásodásig (Mean Time To First Failure, MTTFF) főként nem javítható termékek esetében használják a meghibásodások közötti átlagos működési időt (Mean Time Between Failures, MTBF) javítható termékek esetében alkalmazzák. A működési idő exponenciális valószínűségi eloszlása esetén az MTBF (MTTFF) értéke a λ meghibásodási ráta reciprokával egyenlő, exponenciális eloszlás esetében λ állandó.

3.3. Karbantarthatóság (Maintainability)

A karbantarthatóságot a karbantartás adott (t) idő alatti elvégzésének M(t) valószínűségével és a javítási rátával jellemzik. A valószínűségi eloszlásból származtatják a többi jellemzőt a következők szerint:

Mennyiségi jellemzői: *átlagos javítási idő (Mean Repair Time, MRT), átlagos helyreállítási idő (Mean Time To Restoration, MTTR).*

A javítási idő vonatkozhat a javító karbantartás idejére, de szorítkozhat csak a javítás elvégzésének idejére. A javító karbantartási idő a felkészülési késedelem, a műszaki késedelem és a javítás tényleges elvégzése idejének összegével egyenlő.

A karbantartás fajtái a következők: megelőző-, javító-, alkalmazkodó- (új környezetben is tudja szolgáltatásait teljesíteni a rendszer), fejlesztő- (a rendszer változtatása, hogy növelje a rendszer szolgáltatás-minőségének megkövetelt szintjét) karbantartás. Szoftver esetében csak javító karbantartást végeznek.

A hardver karbantartása a személyzet és a karbantartáshoz szükséges anyagok mozgatását követeli meg. A szoftver karbantartása az információk mozgatását (továbbítását) igényli. A szoftver javított változatait a felhasználó otthonában (a helyszínen) le tudja tölteni.

3.4. Karbantartás-ellátás (Maintenance support)

A karbantartó szervezetnek az a képessége, hogy adott feltételek között – igény esetén – rendelkezésre bocsátja azokat az erőforrásokat és eszközöket, amelyek a termék karbantartásához szükségesek.

Mennyiségi jellemzői: *átlagos karbantartási munkaidő-ráfordítás, átlagos késedelmi idő (várakozás a javítás, karbantartás megkezdéséig).*

A karbantartást ellátó szervezetnek hardver esetében a következő feladatai vannak: tartalék-alkatrészek beszerzése, helyszínre szállítása és a karbantartó személyzet kiszállása (felkészülési késedelem), a rendszer javításra való felkészítése (műszaki késedelem), javítás elvégzése, helyreállítási vizsgálat (ellenőrzés). Szoftver esetében a feladatok a következők: a hiba bejelentés fogadása és a javító szakember kijelölése felkészülési késedelem), a vizsgálati hely kijelölése a diagnosztikai program lefuttatására (műszaki késedelem), a javított szoftver elkészítése, helyreállítási vizsgálat és a javított szoftver próbafuttatása, átvitele és ellenőrzése.

3.5. Használhatóság, rendelkezésre állás (Availability)

A terméknek az a képessége, hogy adott időpontban vagy időszakaszban, adott feltételek között ellátja funkcióit. Ez azt jelenti, hogy a terméket adott időpontban használatba tudjuk venni és ezt követően folyamatosan használni tudjuk. Ennek mennyiségi jellemzője a használhatóság valószínűsége, amelyet az A(t) használhatósági függvény ír le. Az A(t) függvény értékének 1-hez közelinek kell lennie. Ehhez egyrészt hosszú idejű hibamentes működés, másrészt rövid idejű karbantartás (megelőző vagy javító karbantartás) szükséges.

Ezért a használhatóság mennyiségi jellemzője közelítéssel az állandó stacionárius vagy aszimptotikus *használhatósági tényező,* amely azt fejezi ki, hogy a terméket az esetek hányad részében tudja a felhasználó működőképesen használni, ez képletben a következő:

$$\frac{MTBF}{MTBF + MTTR}$$

További megbízhatósággal kapcsolatos időfogalmak: *élettartam, illetve üzemidő (Life Time),* amely a működési idők összege a termék használatba vételétől a határállapotig. Határállapot az az állapot, amelyben a termék működését be kell szüntetni, mert vagy veszélyben van az élet- és vagyonbiztonság, vagy elavult a termék, gazdaságtalan a működtetés, teljes tönkremenés (totálkáros gépkocsi). A köznapi nyelvben, de a gyakorlatban is használják a *naptári idő* fogalmát, ekkor a termék használatba vételétől a határállapotig eltelt naptári időt veszik figyelembe, például a jótállási időt. Egy távközlési rendszer élettartamát is a naptári idő függvényében célszerű vizsgálni, ezért ezt a naptári időt szokták élettartamnak is nevezni.

3.6. A használhatóságot és az eszköz megítélését befolyásoló jellemzők

A következő további képességeket is figyelembe kell venni a rendszer-megbízhatóság műszaki tervezése során:

Alkalmazhatóság, más szóval kezelhetőség, használhatóság (Usability) – nem tévesztendő össze az üzemkészséggel. Az alkalmazhatóság az a képesség, hogy a felhasználó a szolgáltatást igénybe tudja venni.

Mennyiségi jellemzője: annak valószínűsége, hogy a felhasználó a megkövetelt funkciót sikeresen elő tudja hívni adott időszakaszon belül. Egy másik mérőszám az alkalmazhatóságra a következő: a feladat végrehajtásához szükséges idő. *Ezt lehet megtanulhatóságnak is nevezni (Learnability).*

Helyreállíthatóság (Recoverability, Restorability): a rendszernek az a képessége, hogy a meghibásodást követően helyreáll működőképessége és újra ellátja szolgáltatásait, függetlenül attól, hogy javító karbantartást végeztek-e vagy sem. Ezt feléleszthetőségnek is nevezik.

Mennyiségi meghatározása: annak valószínűsége, hogy a termék a meghibásodást követően ismételten ellátja megkövetelt szolgáltatásait adott feltételek között, adott időszakaszon belül, függetlenül attól, hogy javító karbantartást végeztek-e vagy sem. A helyreállíthatóság és az ezzel kapcsolatos javító karbantartás a hardver meghibásodásra, feléleszthetőség pedig a szoftver helyreállíthatóságára vonatkozhat.

Biztonság (Safety): a termék képessége az élet- és vagyonbiztonságot eredményező kritikus meghibásodások elhárítására. Mennyiségileg ezt nem jellemzik, mivel ennek nagy valószínűséggel kell teljesülnie.

Hatékonyság (Efficiency): annak mértéke, hogy a termék egy adott feladatot milyen hosszú idő alatt végez el.

Adatbiztonság (Security): az illegális behatolások elhárításának képessége.

Mennyiségi jellemzője: annak valószínűsége, hogy adott behatolást (támadást) sikeresen elhárítanak adott feltételek között, adott időn belül. Az [1] hivatkozás a behatolás sikerességének valószínűségét, azaz az adatbiztonság megsértésének a valószínűségét méri.

4. Hibaállapot, hiba és meghibásodás

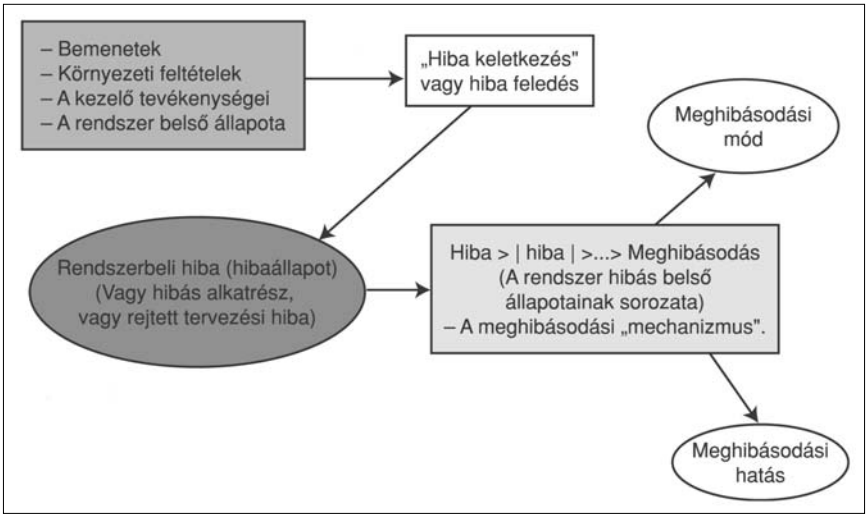
A következő oldalon, az *1. ábrán* látható az a kapcsolat, amely a hibaállapot, röviden: hiba (fault), a meghibásodási mechanizmus (failure mechanism) hatásaként keletkező meghibásodás (failure), annak módja (mode) és hatása (effect) között jön létre.

A rendszert meghibásodottnak tekintik, ha időlegesen vagy véglegesen nem látja el előírt funkcióit. A meghibásodás eseménye az okok sorozati láncának végpontján van.

Ez a folyamat a rejtett tervezési hiba feléledésével kezdődik (hiba-keletkezésnek nevezik), ezt követi a hibás belső állapot vagy azok sorozatának (ezt is hibának nevezzük, angol megfelelője: error) elterjedése a rendszerben. Ha hibás belső állapotot nem fedezik fel és nem javítják ki a rendszer tervezése során beépített *hibatűrő* tulajdonságok felhasználásával, akkor a hiba addig terjed, ameddig az rendszer-meghibásodást nem okoz.

A *tervezési hibák beépítésének hat szintje* van:

1. szint: a leírt követelmény nem felel meg a felhasználó „valós” követelményének (jelentős hibákat vihet be a rendszerbe).
2. szint: a rendszer általános tervezése nem megfelelő (hiányzik a hibatűrés). A rendszert lehet, hogy ellenőrizték és igazolták („ Jól építettük fel a rendszert?”), de lehet, hogy nem alkalmas módon hagyták jóvá („A megfelelő rendszert építettük fel?”).
3. szint: a tervezés nem felel meg a leírt követelményeknek.
4. szint: a felépített rendszer nem felel meg a részletes tervezésnek.
5. szint: kódolási hibák vannak a tervezésben.
6. szint: a nem gondos karbantartás új hibákat vezethet be.



1. ábra
A meghibásodási típusok elvi modellje

A *tervezési meghibásodások jellege* általában az alábbiak egyike:

a) Ember által előidézett meghibásodás:
kódolási hiba vagy karbantartási hiba.
b) Átmeneti (időleges) meghibásodás:
megszűnik, ha a keletkezés okát megszüntetik.
c) Rendszeres meghibásodás:
újra jelentkezik, ha az oka ismét előfordul.
d) Véletlen meghibásodás:
a keletkezés feltételei véletlenül fordulnak elő.
e) Rejtett hibaállapotok miatt fellépő meghibásodás:
a hiba feléledése miatt lép fel.
f) Ritka meghibásodás:
a keletkezés feltételei nem gyakran fordulnak elő.
g) Előre nem jelezhető meghibásodások:
ezek elhárítására nehéz védelmi eszközöket a rendszerbe tervezni és beépíteni.

A koncepcióbeli meghibásodásokhoz (a tervezési hiba miatti meghibásodásokhoz) vezető folyamatok és azok kapcsolatai a hiba-fogalmakkal a *2. ábrán* láthatók.

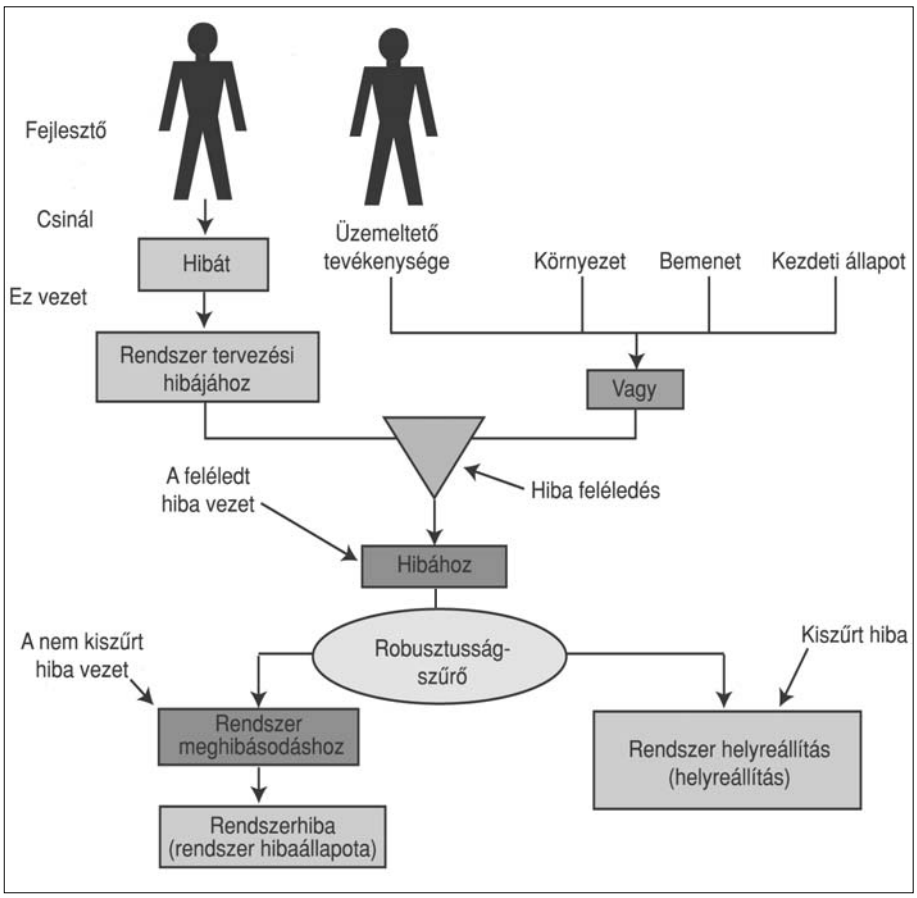
5. A megbízhatóság és a rendszer életciklus-szakaszai közötti kapcsolat

5.1. Általános áttekintés

A megbízhatóság-menedzsment (megbízhatóságirányítás) elősegíti kölcsönös kapcsolat létrehozását a termék életciklus-szakaszai és a termékhez kapcsolódó rendszer életciklus folyamatai között. A termék életciklus-szakaszait az ellátandó feladatokhoz (funkciókhoz) illesztik. A termék életciklus-szakaszai a következők: a termék koncepciójának kialakítása, tervezés és fejlesztés, gyártás, üzemeltetés, karbantartás és selejtezés. Ezekhez a szakaszokhoz kapcsolják és ezekbe a szakaszokba építik be a rendszer megbízhatósági programjának feladatait, amelyek felölelik többek között a beszerzést, a szállítást, a tervezést és szabályozást (ellenőrzést), az értékelést. *A rendszer megbízhatóságát a [4] szabványban megadott mennyiségi jellemzők alapján kell értékelni az előírt érték és az elért eredmény összehasonlításával.*

5.2. A megbízhatóság biztosítása

A bonyolult rendszerek meghibásodását vagy valamely alkotó részhibája, vagy rejtett tervezési hiba idézheti elő.



2. ábra
A koncepcióbeli (vagy a tervezési) meghibásodásokra vonatkozó „hiba (tévedés)", „hiba (hibaállapot)", „hiba" és „meghibásodás" fogalmak közötti kapcsolat

Ezért a rejtett tervezési hibák bekövetkezésének valószínűségét és feléledésüknek káros hatását a lehető legkisebbre kell csökkenteni. Ennek teljesüléséről biztosítékot kell nyújtani a leendő vevő számára, ezt a bizalomkeltési tevékenységet nevezik megbízhatóságbiztosításnak.

Három módszer van a rejtett tervezési hibák csökkentésére: a hiba elkerülése, a hiba eltávolítása és a hibatűrés (hibatűrő-képesség).

A tervezési hibákat elsődlegesen „jó tervezési gyakorlat"-ot követve hárítják el. Lényeges, hogy a vevő „tényleges" követelményeit a rendszer prototípusának alkalmazása és értékelése során érvényesítsék. A legsúlyosabb üzemeltetési meghibásodások a követelmények meghatározásában talált hiányosságoknak tulajdoníthatók. A jó tervezési gyakorlatot követve az egyes részegységek összehangolását az általános tervezési szinten kell elvégezni és a részletes tervezésben el kell kerülni az egymással kapcsolatban lévő részegységek felesleges kettőzését. Külön gondot kell fordítani a felhasználó-barát interfészek tervezésére. A szoftver változtatása után helyreállítási tesztet kell futtatni, hogy elkerüljük az új hibák beépítését, amikor megkíséreltük előzőleg a régi hibákat eltávolítani. A legjobb tervezési gyakorlat sem szavatolhatja a kiküszöbölését. A hibákat ezért már a fejlesztés során automatizált elemzési módszerekkel és a fejlesztés mindegyik szakaszában végzett ellenőrzésekkel el kell távolítani. Ezzel elkerülhető a hibák átvitele egyik szakaszból a következő szakaszba.

Az alapos ellenőrzést a felhasználói feltételeket jól közelítő (szimuláló) kísérleti üzemeltetés követi. A vizsgálatot részegységenként (modulonként), a nagyobb részrendszerek integrálása után kell elvégezni, végül pedig a teljes rendszert kell jóváhagyni a felhasználó követelményei szerint, még a szolgáltatás megkezdése előtt.

A rendszer üzemeltetésére olyan eljárásokat kell kidolgozni, amelyek lehetővé teszik a felhasználók számára a meghibásodás bejelentését úgy, hogy a szervizelő szervezet személyzete megállapíthassa a hiba okát és ennek alapján módosításokat végezzen a rendszerben azok eltávolítására.

A rendszer bemeneteinek és belső állapotainak száma igen nagy, ezért a hibák eltávolítása sem adhat tökéletes eredményt. Ezért hibatűrő rendszert kell tervezni, amelynél a rendszer egyik alkotó elemének hibája a többi alkotóelemet ne befolyásolja. A hiba így megszüntethető mielőtt a rendszerben elterjedne és a rendszer meghibásodna. Ez védelmi tervezést és programozást jelent, valamint tartalékegységek (modulok) beépítését teszi szükségessé azért, hogy a többi egység állapota ellenőrizhető legyen. *A hibatűrés beépítése biztosítja a rendszer hibahatás elleni védettségét és hibahatástól mentes működését.*

5.3. A rendszer-megbízhatóság vizsgálata az egyes életciklus-szakaszokban

A rendszer valamennyi életciklus-szakaszában meg kell vizsgálni a megbízhatóság, biztonság és adatbiztonság követelményeinek teljesülését.

A *rendszer tervezése és fejlesztése* során a részrendszerek megbízhatóságával érhető el, azok integrálása során, a megkövetelt rendszer-megbízhatóság. A biztonság kérdéseivel a közlemény alapjául szolgáló [1] dokumentum nem foglalkozik, az adatbiztonság kérdéseit csak érinti. A rendszer fejlesztését a megbízhatóság-növelési programmal összehangolva kell elvégezni. A tervezéskor gondot kell fordítani arra, hogy az alvállalkozói szerződések megbízhatósági követelményeket is tartalmazzanak.

A *rendszer gyártása* során figyelembe kell venni a jó vevő-szállító kapcsolat kialakítását, az üzemeltető és karbantartó személyzet képzését.

A *rendszer üzemeltetése és karbantartása* során a rendszer-megbízhatóságot befolyásoló tényezők a következők lehetnek: a rendszer üzemmódja, a rendszer együttes működtetése más rendszerekkel, karbantartás és karbantartás-ellátás, emberi hatások.

A *rendszer selejtezése* során a megbízhatóságra vonatkozó információkat gondosan meg kell őrizni, a selejtezést a környezet károsítása nélkül kell elvégezni.

6. Összefüggés a szolgáltatás minősége (QoS) és a rendszer megbízhatósága között

6.1. Általános fogalmi megfontolások

A rendszer üzemeltetésének célja az, hogy szolgáltatást nyújtson a felhasználó részére. A felhasználók akkor veszik igénybe a szolgáltatást, ha annak jellemzői (minősége) megfelelnek elvárásaiknak. A szolgáltatás-minőséget fontos tényezőnek kell tekinteni a tervezés során.

A rendszer megbízhatóságának és a szolgáltatás minőségének kapcsolatát [3] és [4] szabványok határozzák meg. A 2. pontban részletesen foglalkoztunk a hibamentesség, a karbantarthatóság és a karbantartás-ellátás fogalmainak, valamint azok együttes hatását leíró használhatóságnak meghatározásával. Az első három összetevő jelenti a QoS és a megbízhatósági jellemzők legalsó hierarchia-szintjét a *3. ábrán*. A használhatóságot az ábrán a 2. hierarchia- szint jeleníti meg. Ez pedig azt eredményezi, hogy a rendszernek késznek kell lennie a szolgáltatás teljesítésére. Ha nem üzemeltethető a rendszer, akkor nem tud szolgáltatást nyújtani. A rendszer szolgáltatás-minőségét a 3. szinten kell vizsgálni. Ez a rendszer használhatóságának szintje felett van. Minden szolgáltatás minőségét több fogalom együttesen írja le. Ezek közül két fontos QoS fogalmat kell kiemelni:

Szolgáltatás elérhetősége (Service accessibility)

A szolgáltatásnak az a képessége, hogy – adott tűréseken belül és adott feltételek között – megkapható, ha a felhasználó igényli.

Szolgáltatás folyamatossága (Service retainability)

Az egyszer már megkapott szolgáltatásnak az a képessége, hogy adott feltételek között, kívánt időtartamig folytatódik.

Ez a két fogalom a szolgáltatás használhatósági jellemzőjeként is értelmezhető, ha a szolgáltatást teljesítő rendszerek rendelkezésre állnak. Ezért ez a két jellemző nemcsak QoS fogalom, hanem használhatósági fogalom is a 3. szinten.

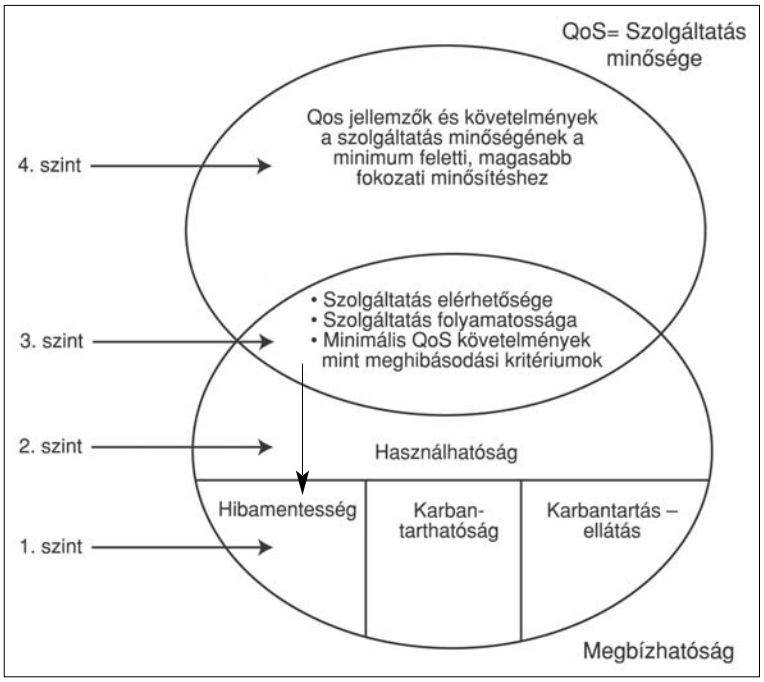
A QoS és a rendszer-megbízhatóság tervezésének további közös területei is vannak. A minimális QoS követelmények nem teljesítése meghibásodásnak tekintendő, ezért a műszaki tervezés során ezt meghibásodási kritériumként kell figyelembe venni. A QoS azonban többet jelent a megbízhatóságnál, mert tovább osztályozza az egyes rendszer-szolgáltatásukat minőségük szerint a minimális minőség-szint felett. Ennek azonban előfeltétele, hogy a QoS-hez mérhető, objektív minőségi paramétereket rendeljünk. Ezeket a *3. ábra* 3. és 4. szintjén szereplő jellemzőkre osszuk fel és az összegezési szabályokat előre rögzítsük. Ezáltal további különbséget tehetünk a másképpen azonosan „megbízható" rendszerek között. A megbízhatósági és QoS jellemzők egyesített modellje a *3.ábrán* látható.

6.2. A megbízhatóság műszaki tervezésének fontossága

A rendszer-megbízhatóság műszaki tervezése a minimális QoS szint figyelembe vételével kezdődik. A rendszer egyes előírt funkcióira megbízhatósági jellemzőket kell megállapítani.

Ezek lehetnek a következők: használhatóság, állási idő, MTTR. Ezt követően osztályozni kell a különböző hibatípusokat jelentős és jelentéktelen, teljes és részleges hibák. Ezt követően az általános követelményeket fel kell osztani minősítéses és mennyiségi követelményekre. A megbízhatósági vizsgálatoknak is arra kell irányulniuk, hogy igazolják a szolgáltatás használhatóságára vonatkozó követelmények teljesülését.

3. ábra A megbízhatóság és a szolgáltatás-minőség jellemzőinek négy szintű integrált tartománya



7. A rendszer életciklusának folyamatai

A megbízhatóság-irányítás a rendszer életciklus folyamatait használja fel a megfelelő megbízhatósági folyamatok irányítására. A folyamatok négy csoportra oszthatók fel:

1. Vállalati folyamatok, amelyek kiterjednek a vállalat-, a befektetés- és az életciklus-irányítási, valamint az erőforrás-gazdálkodási folyamatokra.
2. Szerződéses folyamatok, amelyek tartalmazzák a beszerzési- és a szállítási folyamatokat is.
3. A projekt irányításának folyamatai felölelik a tervezési-, értékelési-, szabályozási-, döntéshozási-, kockázatkezelési-, konfiguráció-kiépítési- és minőségirányítási folyamatokat.
4. A technikai folyamatok a következők: az érdekelt felek igényeinek meghatározását, a követelmények elemzését, a tervezést, integrálást, igazolást, telepítést, jóváhagyást, és selejtezést egyaránt tartalmazzák.

Irodalom

[1] IEC TC 56 (2004):
Guidance to Engineering of System Dependability

[2] ISO 9000:2000: Quality Management Systems:
Fundamentals and Vocabulary (2000)

[3] MSZ ISO 50(191):
Megbízhatóság és szolgáltatásminőség fogalmai (1993)

[4] ITU-T Recommedation E 800

[5] Methods for Testing and Specification (MTS);
Testing and Test Control Notation; Part 1:
TTCN-3 Core Language, ETSI ES 201 873-1.

[6] Gecse R.–Szabó J.–Csöndes T.:
Időzített véges automata alapú vizsgálati módszer.
Híradástechnika 2002/3., pp.19–24.

[7] Andriska Z.–Bátori G.–
Wu-Hen-Chang A.–Csopaki Gy.:
Automatikus tesztkiválasztás
formális specifikáció alapján

[8] Lócz B.–Zömbik L.:
Hálózati protokollok biztonsági tesztelése.
Híradástechnika 2004/3., pp.2–9.

[9] Konkoly Lászlóné–Fekete I.:
Nyereség optimalizálás
az üzleti kockázatelemzés és játékelmélet
együttes alkalmazásával.
Híradástechnika 2003/9., pp.4–11.

[10] Molnár S.–Szabó Z.–Kenesi Zs.:
Aktív tároló kezelő mechanizmusok
hatása a TCP adaptivitásra.
Híradástechnika 2003/4., pp.15–24.

[11] Ornicsay D.–Józsa B.:
Távközlési hálózatok
költséghatékony tervezése.
Híradástechnika 2003/4., pp.25–29.

[12] Kanász-Nagy L.:
Biztonság a távközlésben.
PKI Közl., 48. szám, 2004., pp.141–154.

Hívásengedélyezés megvalósítása integrált hang-adat hálózatokban

ÉGI NORBERT, DREILINGER TÍMEA

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

Reviewed

Kulcsszavak: sávszélesség-ügynök, minőségi szerződések (SLA), hálózatmenedzsment, teljesítményelemzés

Az Internet széleskörű elterjedése megnövelte az érdeklődést az Interneten keresztüli hangtovábbítás iránt. Mivel az Internetet nem valósidejű adatkommunikáció céljából fejlesztették ki, így számos technikai nehézséggel kell szembenézni és komoly akadályokat kell elhárítani, míg a telefonforgalmat megfelelő minőségben lehet rajta továbbítani. A sávszélesség-ügynök egy olyan eszköz, amely az erőforrások dinamikus menedzseléséhez szükséges feladatokat látja el. Kezeli a felhasználók és a szolgáltató között köttetett hosszútávú szerződéseket, hívásengedélyezést végez, valamint ellenőrzi az erőforrások lefoglalásához szükséges jelzéseket és jogosultságokat. E cikk egy általunk megvalósított hálózatmenedzsment eszközt mutat be, mely alkalmas az integrált hang-adat hálózatok erőforrásainak kezelésére és hívásengedélyezési funkcióval rendelkezik.

Az utóbbi néhány évben a világ számos pontján bevezették az Internet feletti hangtovábbítást *(Voice over IP, VoIP)*. Az Internetet minden eddiginél többen használják, így egyre inkább terjednek azon alkalmazások, amelyekkel az Interneten lehet telefonálni. Mindezeken felül a közeljövőben a harmadik generációs mobil távbeszélő rendszer, az *UMTS (Universal Mobile Telecommunication System)* is IP alapra helyezi a mobiltelefon kapcsolatokat, ezáltal a VoIP még inkább elterjedhet. Összességében tehát megállapítható, hogy az Internet alapú beszédátvitelnek jelentős szerepe lesz a távközlésben.

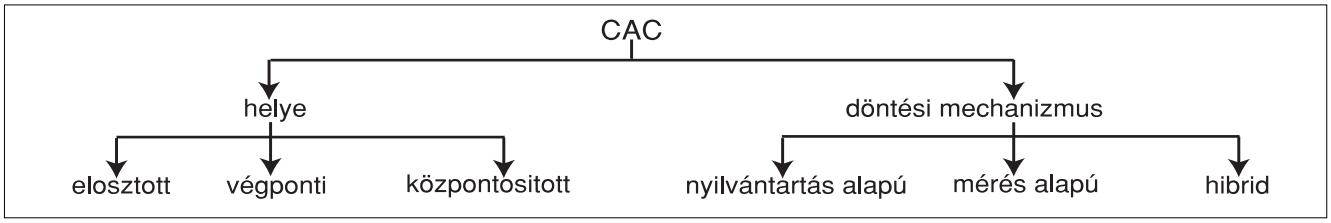
A szolgáltatók számára jelentős költségmegtakarítást jelenthet az adat és beszédátvitel egységesítése. Ez azonban nem könnyű feladat, hiszen ebben az esetben nagy figyelmet kell fordítani a beszédátvitel minőségére. Mivel a számítógép-hálózatokat eredetileg adatátvitelre tervezték, így ezek a hálózatok önmagukban nem megfelelőek párbeszédre vagy más interaktív kapcsolatra. Így, ha jó minőségű távbeszélő szolgáltatást kívánunk megvalósítani, néhány minőségi paramétert hálózati szinten is garantálni kell.

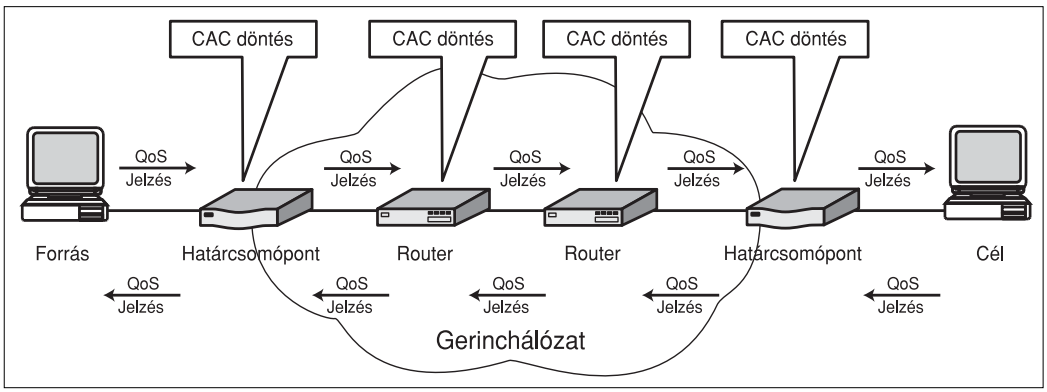
Minőségmenedzsment hívásengedélyezéssel

A VoIP forgalmak minőségi követelményeinek egyik lehetséges kielégítési módja *hívásengedélyezési (Call Admission Control, CAC)* mechanizmusok alkalmazása, amelyek korlátozzák a hálózatot egyidejűleg terhelő kapcsolatok számát. Ezt a feladatot úgy kell megoldani, hogy a szükséges minőségi paraméterek betartása mellett a hálózat kihasználtsága is a lehető legjobb legyen. A garantált *szolgáltatás-minőséget (Quality of Service, QoS)* biztosító Internet Protokoll (IP) alapú hálózatokra javasolt hívásengedélyezési algoritmusokat (CAC) két szempont alapján különböztetjük meg: a lehetséges döntési mechanizmusok és a különféle alkalmazott architektúra alapján *(1. ábra)*.

A lehetséges döntési mechanizmusok alapján a hívásengedélyezési módszerek nyilvántartás, mérés alapú és hibrid csoportra bonthatók. A *nyilvántartás alapú* módszerek olyan módon próbálják megbecsülni a szabad, illetve a felhasznált sávszélességet, hogy az egy forrás által kibocsátott adatmennyiséghez valamiféle forgalomleírót (például sávszélesség-jellemzőket, azaz egy változó sebességű forgalom esetén a csúcssebességet vagy az átlagsebességet) rendelnek, majd ezeket nyilvántartják és összegezik. *Csúcssebesség* használata esetén nem hasznosul a változó sebességű folyamok statisztikus multiplexálási nyeresége, így kisebb a hálózat kihasználtsága és nem garantálható a nulla csomagvesztési arány. Ezzel szemben *átlagsebesség* használata esetén a csomagvesztési arány szabályozható nehezen, ami különösen akkor jelenthet problémát, ha a csúcssebesség jóval nagyobb az átlagsebességnél. A nyilvántartás alapú hívásengedélyezési módszerek egyik fontos tulajdonsága, hogy a tárolt ál-

1. ábra A hívásengedélyezési algoritmusok csoportosítása





2. ábra Elosztott hívásengedélyezési architektúra

lapotok bizonyos idő után elévülnek, ezért ezen állapotok fenntartásához periodikus frissítésekre van szükség. Ezen frissítések pedig – hátrányos módon – a hálózati forgalomban többletterhelést jelentenek. A kizárólag nyilvántartás alapú hívásengedélyezési módszerek további hátránya, hogy a döntéshozatalhoz nem a tényleges hálózati foglaltságot, hanem annak egy elméleti felső korlátját veszik alapul.

A *mérés alapú* [1,2,3] módszereknél a döntéshez szükséges információkat a hálózati forgalom nagyságából próbálják kinyerni, ezáltal pontosabb képet kapnak a hálózat aktuális terheltségéről. A forgalmi mérések eredménye lehet a felhasznált sávszélesség, de vizsgálhatnak konkrét minőségi paramétereket is, például csomagvesztési arányt, vagy késleltetés-ingadozást. A méréseken alapuló módszerek hátránya, hogy lassan reagálnak a hálózatban bekövetkezett változásokra, a mérési eredmények csak bizonyos idő elteltével tükrözik a változásokat. Ezen a problémán a mérési idő csökkentésével lehet segíteni, azonban ekkor a hívásengedélyezés során néhány korábbi mérési eredményt is figyelembe kell venni, hogy a folyamok hosszú távú változásai is befolyásolhassák a döntést.

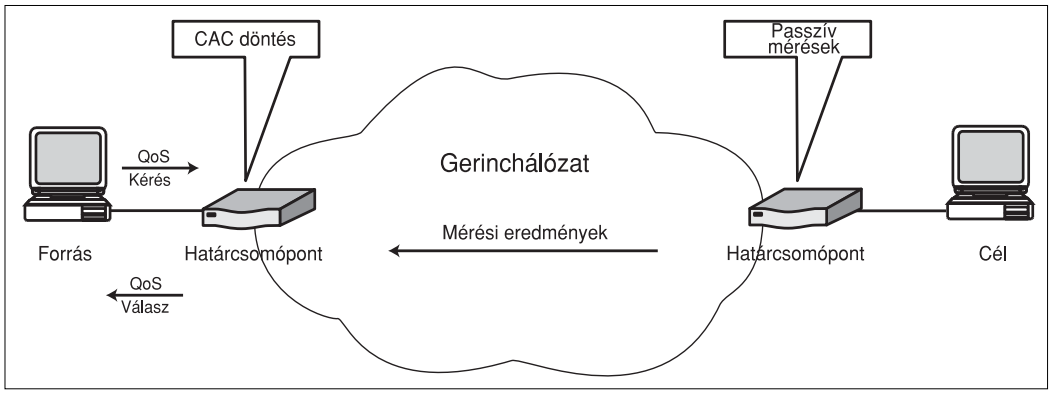
Végezetül léteznek *hibrid,* mérést és nyilvántartást egyaránt használó módszerek. Ezek a nyilvántartáson alapuló hívásengedélyezésnél már említett ekvivalens kapacitást számolnak, ugyanakkor nem élnek semmilyen feltételezéssel a bejövő forgalommal kapcsolatban. Ezért a döntés meghozatalához forgalmi mérésekre van szükség. Aszerint, hogy a döntés a hálózat melyik pontjában történik, a javasolt hívásengedélyezési protokollok és architektúrák három csoportba oszthatók: elosztott, végponti és központosított módszerekről beszélhetünk.

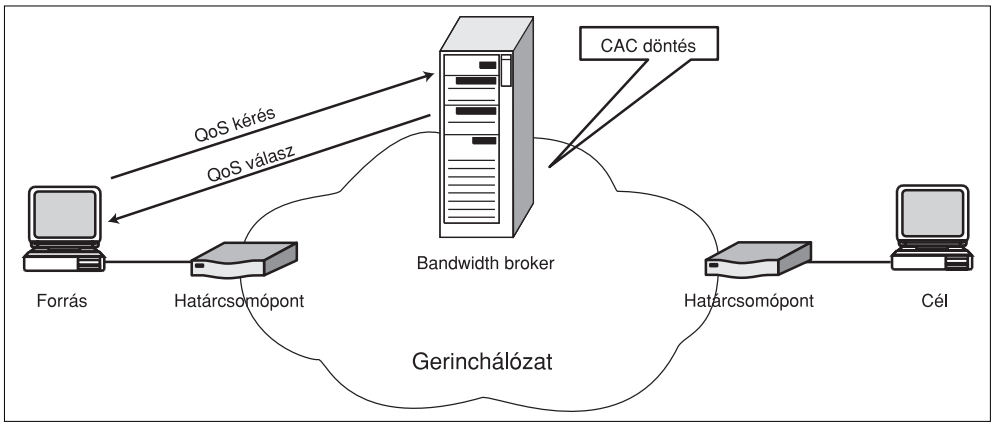
Elosztott architektúrák esetében [4] az erőforrást igénylő kérés végighalad a hálózatban bejárt lehetséges útvonal minden egyes csomópontján. Ezen csomópontok önállóan döntik el, hogy az új hívás beengedhető-e, vagy sem *(2. ábra)*. Az elosztott architektúrák hátránya, hogy az egymást követő, láncszerű hívásengedélyezési döntések felesleges erőforrás használatot eredményeznek, hiszen az egy adott kéréshez tartozó erőforrásokat akkor is lefoglalja, ha egy későbbi útvonalválasztó majd elutasítja azt.

Végponti architektúra esetén [5,6] a forgalmi források, vagy a hálózat határcsomópontjai vesznek részt a hívásengedélyezési mechanizmusban. Ha a hívásengedélyezést maga a forrás végzi, akkor a hálózat aktuális minőségi paramétereit aktív mérés segítségével, próbaforgalmat küldve lehet ellenőrizni. Ezzel szemben, ha a döntést a határcsomópontok hozzák, az aktív mérés mellett lehetőség van passzív mérésekre is *(3. ábra)*. A végponti hívásengedélyezési módszerek hátránya, hogy a források és a határcsomópont közt nincs valós idejű információcsere, így előfordulhat, hogy a beérkező hívások számára ugyanazt az erőforrást több forrás vagy határcsomópont is lefoglalja. Ezáltal megnő a minőségi követelmények sérülési valószínűsége. A sérülés esélye annál nagyobb, minél nagyobb a hívásbeérkezési intenzitás.

A *központosított architektúrák (4. ábra)* egy, a hálózati útválasztóktól független elemet, sávszélesség-ügynököt *(Bandwidth Brokert, BB)* [7] használnak a hívásengedélyezésre. A központi elrendezés előnye, hogy egyidejűleg rendelkezésre áll a döntéshez szükséges összes, a hálózat csomópontjával kapcsolatos információ. Továbbá nem léphet fel a végponti architektúránál vázolt túlfoglalás problémája, és ugyanígy nem következhet be az elosztott architektúránál gondot okozó részleges lefoglalás sem. A centralizált megoldás azonban megbízhatósági és teljesítőképességbeli kérdéseket vet fel.

3. ábra Tartomány szintű végponti hívásengedélyezési architektúra passzív mérésekkel





4. ábra Központosított hívásengedélyezési architektúra

A hívásengedélyezést végző hálózatmenedzsment eszköz

Az általunk megvalósított eszköz központosított architektúrájú és hibrid hívásengedélyezési döntésen alapszik, azaz a hívásengedélyezés során mérést és nyilvántartást is használ. Az eszköz a nyilvántartott adatokat adatbázisban tárolja. A mérést a hálózat szélein elhelyezkedő útválasztók végzik, és a mért adatokat a hívásengedélyezési eszköz adatbázisába töltik. Az eszköz a hívásengedélyezési döntéshez teljes mértékben az adatbázisban található adatokat használja fel. A nyilvántartott információk közt szerepelnek a hálózat elrendezését leíró adatok, a hálózatot alkotó csomópontok és összeköttetések paraméterei, a határcsomópontok közt lehetséges összes útvonal, valamint a hálózat útvonalain haladó aktuális forgalmak leíró jellemzői.

A mérés során az útválasztók a Cisco IOS NetFlow technológiát [8] felhasználva gyűjtik és mérik az útválasztókba vagy kapcsoló interfészekbe beérkező forgalmat. A NetFlow az egészen pontos és részletes forgalommérésre is alkalmas, és lehetővé teszi a hálózat számára az IP forgalom analízisét. Ezen megvalósításban minden útválasztó rögzített időközönként frissíti a mérési adatok tárolására szolgáló adatbázis-tábla bejegyzéseit. A NetFlow rendszer képes a folyamokat osztályok szerint elkülöníteni, így az IP fejléc szolgáltatás típusa *(Type of Service, TOS)* mezejének megfelelő beállításával külön tudjuk mérni a hálózatban haladó VoIP forgalmat.

A megvalósított rendszer a kapcsolatok felépítésére és bontására a *SIP (Session Initiation Protocol)* [9] jelzésrendszert használja. Ezért a hálózatban szükség van egy SIP Proxy szerverre, amely egy TCP kapcsolaton keresztül, XML üzenetekkel tájékoztatja a hívásengedélyező eszközt. Ennek során a hívásengedélyező eszköz a SIP Proxy szervertől kapja a hívásengedélyezési kérést, majd a döntés meghozatala után a választ a SIP Proxy szervernek továbbítja *(5. ábra)*. Ha a hívásengedélyezés eredményeképpen a folyam a hálózatba beengedhető, a hívásengedélyező eszköz lefoglalja a hálózatban a folyam által igényelt mennyiségű erőforrást. A hívás végét a SIP Proxy szerver jelzi a hívásengedélyező eszköznek. Utóbbi ekkor felszabadítja a korábban lefoglalt erőforrásokat.

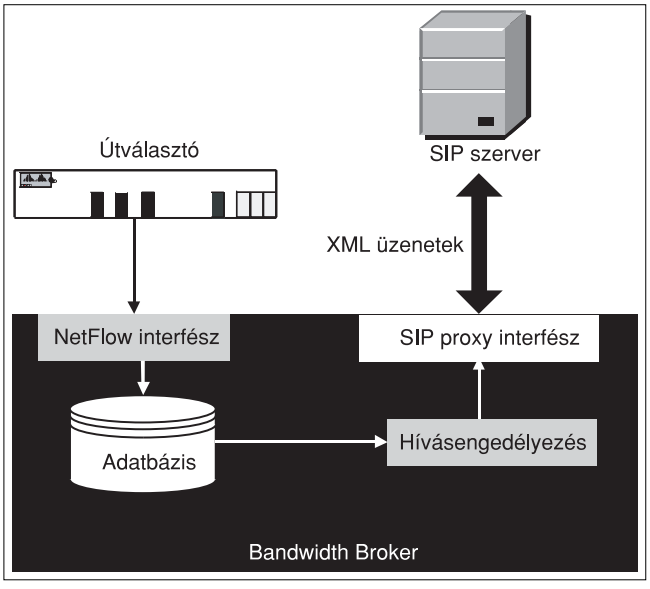
A megvalósított rendszerben a legelterjedtebben használt kódoló típusokat használjuk, úgymint a G.711-es PCM *(Pulse Code Modulation)* kódoló amerikai *μ*-law (56 kbit/s) és az európai *A*-law (64 kbit/s) szabványát. Ezen kívül a beszédkompressziót alkalmazó G.723.1 kódoló mindkét, 5,3 és 6,3 kbit/s sávszélesség-igényű változatát, továbbá a 8 kbit/s sávszélesség-igényű G.729 és a 13,2 kbit/s sávszélesség-igényű *GSM (Global System for Mobile Communication)* Full Rate kódoló típusokat vizsgáltuk.

A hívásengedélyezési algoritmus középpontjában álló döntés egy Hoeffding-korláton [10] alapuló eljárás. Hoeffding ezen korlátját már több helyütt Error! Reference source not found. ajánlották mérés alapú hívásengedélyezések megvalósításához. A becslő módszerek igen egyszerűek, mivel mindössze az aggregált forgalom középértékét és a független források számát használják fel. A farokeloszlás szempontjából ezek a (felső) korlátok azonban durvák, mivel a forgalom karakterisztikájával kapcsolatban kevés információ áll rendelkezésre. Ez a közelítés abban az esetben megfelelő, ha a nagy számú felhasználó által előállított forgalomról kevés információnk van és a forgalmat leíró jellemzőkből néhányat meg kell mérnünk.

A hívásengedélyezési döntéshez a Hoeffding-egyenlőtlenséget átalakítva az alábbi összefüggést használjuk:

$$P \leq C - M - \sqrt{\frac{1}{2}\gamma \sum_k p_k^2},$$

5. ábra A sávszélesség-ügynök felépítése



ahol P a belépni kívánó folyam (hívás) sávszélességigénye, C az adott link kapacitása, M a linken mért forgalom, γ egy konstans, amellyel a túlterhelés valószínűsége állítható be, míg p_k a linken lévő k-adik folyam sávszélessége. Az eljárás a felső korlát meghatározásához kis számítási komplexitást igényel, azonban a gyakorlatban sokszor túlságosan is konzervatív. Ennek ellenére a csúcssebességre foglalásnál hatékonyabb erőforrás-kihasználást tesz lehetővé anélkül, hogy az átlagsebességre történő foglalásnál fellépő esetleg nagy mértékű forgalomvesztés kialakulna.

A megvalósított hívásengedélyezési eszköz hatékonyságát 320 kbit/s sebességű linken végzett mérésekkel ellenőriztük. A teljesítményelemzés során két határcsomópont közt egy forgalomgenerátor programot felhasználva hoztunk létre forgalmat. Mértük, hogy az adott kódolóval kódolt hívásokból mennyit képes a két határcsomópont közti útvonal elvezetni anélkül, hogy a kívánt minőség sérülne. A vizsgálatok során különböző típusú kodekeket és túlterheltségi valószínűségeket használtunk. A kapott eredményeket az *1. táblázat* mutatja. Ebből az látható, hogy különböző túlterheltségi valószínűségek esetén a hálózatba beengedett hívások száma miként változik, valamint megfigyelhetjük azt is, hogy az alkalmazott eljárás hatékonyabb, mint a csúcssebességre való foglalás. Ennek oka, hogy a Hoeffding-korlát a hívások csomósodását használja ki a link kapacitásának minél jobb kihasználása érdekében. Így a döntési képlet egyszerre több, kisebb sávszélességet igénylő hívást enged be a hálózatba, mivel az ilyen hívások löketnagyságai jobban kiegyenlítik egymást, mint a kevesebb számú, de nagyobb sávszélesség igényű hívásoké.

Összefoglaló

A hívásengedélyező eszköz megvalósításával egy olyan rendszert sikerült létrehozni, amely módosítható valószínűséggel garantálja a felügyelt integrált hangadat hálózatban haladó hívások minőségét. A hívásengedélyezéshez mindössze a hálózat határain elhelyezkedő csomópontok közreműködésére van szükség, így a rendszer működése rugalmasabb és egyszerűbb, mintha minden csomópont részt venne a hívásengedélyezésben. Továbbá a nyilvántartást és mérést egyidejűleg használva garantálható a hívásengedélyezés helyessége.

Irodalom

[1] L. Westberg, Z. R. Turanyi, D. Partain, Load Control of Real-Time Traffic, draft-westberg-loadcntr-03.txt

[2] Viktória Elek, Gunnar Karlsson, Robert Rönngren, Admisson Control Based on End-to-End Measurements, IEEE INFOCOM 2000.

[3] G. Bianchi, A. Capone, C. Petrioli, Throughput Analysis of End-to-End Measurements-Based Admission Control in IP, IEEE INFOCOM 2000.

[4] Frank P. Kelly, Peter B. Key, Stan Zachary, Distributed admission control. (2000) IEEE Journal on Selected Areas in Communications

[5] F. Cao, H. Fang, M. Conlon, Performance analysis of measurement-based call admission control on voice gateways. In Proceedings of Internet Telephony Workshop 2001 (IPTEL' 2001), New York City, U.S.A., April 2001.

[6] G. Bianchi, F. Borgonovo, A. Capone, L. Fratta, C. Petrioli, Endpoint admission control with delay variation measurements for qos in ip networks. In Sigcomm, volume 32, April 2002.

[7] O. Pop, T. Máhr, T. Dreilinger, R. Szabó, Vendor-independent bandwidth broker architecture for diffserv networks. In IEEE ICT'2001, 2001.

[8] CISCO NetFlow Technology, www.cisco.com/warp/public/732/Tech/nmp/netflow/

[9] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R, Sparks, M. Handley, and E. Schooler, SIP: Session Initiation Protocol. RFC 3261, IETF, June 2002.

[10] W. Hoeffding, Probability Inequalities for Sums of Bounded Random Variables. J. Amer. Statist. Assoc., pp.13–30, 1963.

[11] Z. Heszberger, J. Zátonyi, J. Bíró, Efficient Chernoff-based Resource Assessment Techniques in Multi-Service Networks. TELECOM'01, 2001.

** Készült az IKTA-00092/2002 OM projekt támogatásával*

1. táblázat A teljesítményelemzés mérési eredményei

Kodek neve	Sávszélesség igény (kbit/s) hang/+IP+ETH	Hívások maximális száma linkenként	10%-os túlterheltségi valószínűség mellett a beengedett hívások darabszáma			1%-os túlterheltségi valószínűség mellett a beengedett hívások darabszáma			0,1%-os túlterheltségi valószínűség mellett a beengedett hívások darabszáma		
			Átlag	Min.	Max.	Átlag	Min.	Max.	Átlag	Min.	Max.
G.711	64 / 87.2	3	3.6	2	4	3.2	2	4	2.3	2	3
G.723	6.3 / 21.9	14	15.6	11	16	13.1	11	15	12.5	11	14
G.729	8 / 31.2	10	10.8	8	12	10	8	11	8.6	7	10
GSM	13 / 34.4	9	9.9	8	11	9	8	10	7.9	7	9

Újszerű erőforrásigény-becslő módszerek csomagkapcsolt hálózatokban

MARTINECZ MÁTYÁS, BÍRÓ JÓZSEF, HESZBERGER ZALÁN

martinecz@tmit.bme.hu

Reviewed

Kulcsszavak: ekvivalens kapacitás, QoS, hívásengedélyezés

Az átviteli minőségre vonatkozó garanciák hiánya a csomagkapcsolt hálózatok már-már klasszikusnak nevezhető problémája. Ilyen garanciák nélkül az új, értéknövelt szolgáltatások gyors elterjedése az Interneten annak ellenére sem lehetséges, hogy a megfelelő sebességet biztosító hozzáférési technológiák már elérhetők. Cikkünkben olyan könnyen megvalósítható erőforrásigény felmérő technikákat ismertetünk, amelyekkel megbecsülhető az aggregált hálózati forgalom számára a minőségjellemzőkre vállalt garanciák teljesítése mellett minimálisan szükséges sávszélesség. Ezen módszerek alapját képezhetik a csomagkapcsolt (hozzáférési) hálózatok terhelésszabályozását végző (például hívásengedélyező) algoritmusoknak.

1. Bevezetés

Napjainkban világszerte óriási ütemben növekszik a digitális előfizetői vonalat (DSL) használók száma. A gyors növekedés oka a DSL szolgáltatások kedvező árában és az általuk elérhető viszonylag nagy adatsebességben keresendő. Az alacsony árra magyarázatot ad, hogy a DSL hozzáféréshez szimmetrikus rézkábelek használhatók lesznek, azok mindezidáig használaton kívül eső 144 kHz feletti frekvenciatartományainak kiaknázásával. Mivel ezek a rézkábelek már elérik a telefonnal rendelkező felhasználókat, ezért bizonyos esetekben a digitális előfizetői vonalak telepítési költségei lehetnek a legkedvezőbbek.

A DSL-en keresztül nyújtott szolgáltatások között toronymagasan vezet az Internet hozzáférés [1]. A korszerű hozzáférési technológia elégséges átviteli sebességet biztosít a legtöbb, ma elérhető Internetes szolgáltatás számára. A korábban elérhető alacsony adatátviteli sebességek komoly visszahúzó erőt jelentettek az Internetes technológiák és alkalmazások fejlődésére. A digitális előfizetői vonalak megjelenésével a modern, szélessávú hálózati szolgáltatások evolúciója új lendületet kapott. Ezen új szolgáltatások elterjedését a hozzáférői hálózatok üzemeltetői is szorgalmazzák, hiszen ezek új előfizetőket vonzanak táborukba.

A TCP/IP alapú csomagkapcsolt hálózatok egyik nagy, már-már klasszikusnak nevezhető, problémája az átviteli minőségre vonatkozó (QoS) garanciák hiánya. Ilyen biztosítékok nélkül az értéknövelt szolgáltatások bevezetése és elterjedése nem lehetséges. Minőségi garanciákra valamint az őket lehetővé tevő hálózat- és forgalommenedzsment algoritmusokra leginkább ott van szükség, ahol az erőforrások is szűkösek: a helyi hurokban illetve a hozzáférési hálózatban.

Az adatátvitel minősége a hálózat aktuális terheltségének a függvénye, amit például a hálózati tartomány egyes linkjeinek túlcsordulási valószínűségével (a linkszaturációs valószínűséggel) vagy a csomagvesztési aránnyal jellemezhetünk. Előbbi mérték azt adja meg, hogy (tároló nélküli hálózati modellt feltételezve) egy adott linken átmenő aggregált forgalom pillanatnyi sávszélességigénye az idő hány százalékában haladja meg a link kapacitását. Ez a mérték sajnos nem mond sokat a ténylegesen elvesző információ mennyiségéről, ezért célszerűbb a csomagvesztési arányra előírást adni, ami a forgalom azon hányadát jelenti, amelyet a rendszer nem képes továbbítani linktúlcsordulás miatt.

Jelen cikkben újszerű módszereket mutatunk be, amelyekkel egy adott hálózati szegmens terheltsége előre becsülhető minimális számú paraméter segítségével. Ezt a becslést a hálózati tartományt felügyelő sávszélesség-ügynök használhatja fel forgalomszabályó döntések meghozatalához. A bemutatandó formulák a QoS előírás figyelembe vételével közvetlenül az aggregált forgalom minimális sávszélesség igényét becsülik, ellentétben azokkal az eljárásokkal, amelyek adott linkkapacitás mellett a garantált QoS mérték várható értékét határozzák meg. Az általunk alkalmazott közvetlen módszer előnye, hogy az igényelt sávszélesség aktuális értékét elegendő periodikusan elvégzett háttérszámításokkal frissen tartani, míg a QoS mérték várható értékét minden újonnan érkező folyam belépése előtt ellenőrizni kell, ami a hívásengedélyezési döntés meghozatalát lassítja.

Rövidítések

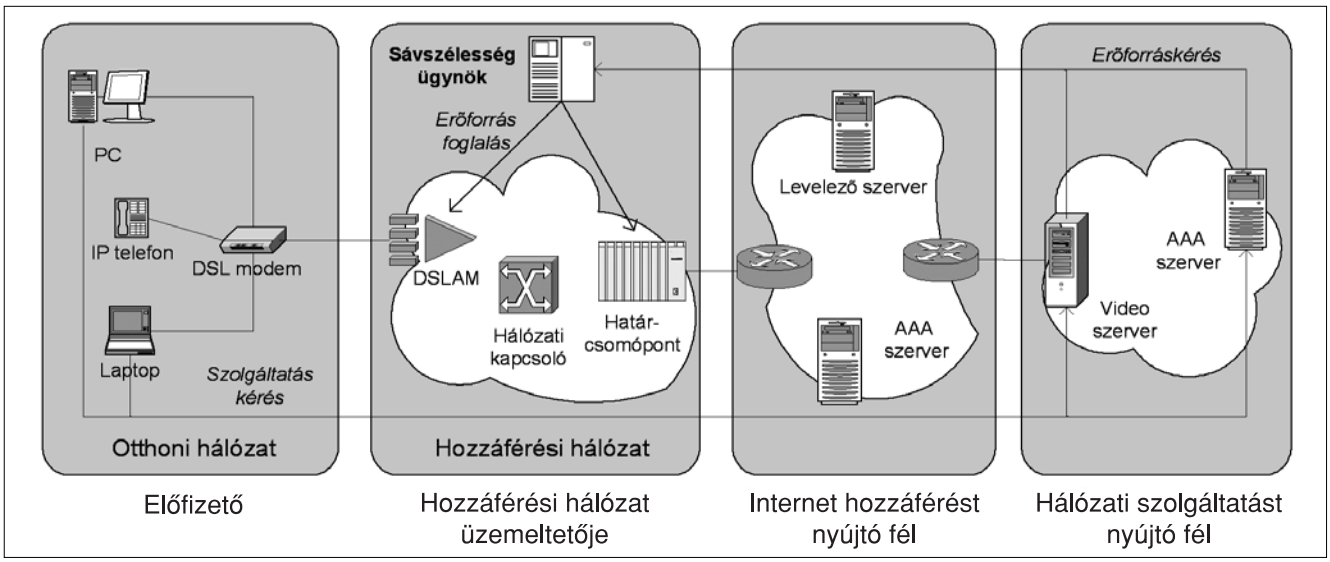
BFFM – Bufferless Fluid Flow Multiplexing *(tárolás mentes folyadék folyam aggregálás)*

DSL – Digital Subscriber Line *(digitális előfizetői vonal)*

LMGF – Logarithmic Moment Generating Function *(logaritmikus momentumgeneráló fv.)*

PLR – Packet Loss Ratio *(csomagvesztési arány)*

QoS – Quality of Sevice *(szolgáltatásminőség)*



1. ábra Minőségbiztosított Internetes szolgáltatás a hozzáférési hálózatban elhelyezett sávszélesség-ügynökkel

A következőkben először az általunk alkalmazott matematikai modellt ismertetjük, majd a harmadik részben olyan technikákat tárgyalunk, amelyekkel egy adott forgalom pillanatnyi sebességeloszlásának momentumgeneráló függvénye felülről becsülhető. A negyedik részben megmutatjuk, hogyan konvertálhatók a QoS mértékekre vonatkozó becslések sávszélesség jellegű mennyiséggé, végül numerikus példákon keresztül összehasonlítjuk az újonnan bemutatott és a régebbi eredmények hatékonyságát.

2. Minőségjellemzők és becslésük csomagkapcsolt hálózatokban

A bevezetőben felvázolt probléma megközelítésénél a népszerű tároló nélküli folyadék folyam aggregálási (BFFM) modellt alkalmaztuk. Mivel ebben a modellben nincs tároló, amely csökkentené a linkszaturációs valószínűséget, ezért ez a megközelítés alkalmas a számunkra fontos QoS mértékek konzervatív becslésére.

Az általunk alkalmazott BFFM modellben n darab folyadék folyamot aggregálunk egy C kapacitású linken. Jelölje az X_i valószínűségi változó az i-ik stacionárius folyam pillanatnyi adási sebességét. Tegyük fel, hogy minden folyam esetén megállapítható egy p_i csúcs adási sebesség, azaz $0 \le X_i \le p_i$. Továbbá jelölje az X valószínűségi változó az aggregált folyam adási sebességét:

$$X = \sum_{i=1}^{n} X_i$$

Ekkor a linkszaturációs valószínűséget az alábbi képlettel definiálhatjuk:

$$P_{sat} \overset{def}{=} P(X > C) \qquad (1)$$

Ez a valószínűség tehát arról ad információt, hogy (ergodikus rendszert feltételezve) az idő milyen hányadában haladja meg az aggregált forgalom pillanatnyi adási sebessége a link kapacitását. Ez a viszonylag egyszerűen számolható QoS mérték a hálózatüzemeltető szempontjából lehet hasznos, azonban a ténylegesen elvesző adatmennyiségről nem ad megbízható információt.

Vegyük észre, hogy azonos linkszaturációs valószínűség mellett az elveszett információmennyiség különböző lehet, ezért a felhasználók elégedettségét jobban jellemzi a csomagvesztési arány mértéke, melynek definíciója:

$$PLR \overset{def}{=} \frac{E\left[(X-C)^{+}\right]}{E[X]} \qquad (2)$$

ahol $E[.]$ a várható érték képzés operátora, továbbá $(X-C)^{+} = \max(X-C,0)$. Tehát a csomagvesztési arány a C linkkapacitást meghaladó, s így (tároló hiányában) csomagvesztést okozó pillanatnyi adási sebesség várható értékének, valamint a pillanatnyi adási sebesség átlagos értékének hányadosaként számolható.

A hívásengedélyezési döntés alapja a garantált QoS mérték és annak a rendelkezésre álló információkból becsült értéke közötti reláció:

$$P(X > C) \le e^{-\gamma} \quad \text{vagy} \quad \frac{E\left[(X-C)^{+}\right]}{E[X]} \le e^{-\gamma} \qquad (3)$$

A valóságban jobban alkalmazható, ha az adott link összkapacitás-értékét vetjük össze az aggregált forgalom számára az adott QoS korlát mellett minimálisan szükséges átviteli kapacitás értékével. Ezt az sávszélességet a szakirodalomban *ekvivalens kapacitásnak* nevezik, aminek definíciója linkszaturációs valószínűségre vagy csomagvesztési arányra vonatkozó korlát esetén az alábbi alakban írható fel:

$$C_{equ,sat} \overset{def}{=} \inf\left\{C: P_{sat} \le e^{-\gamma}\right\} \quad \text{vagy}$$

$$C_{equ,PLR} \overset{def}{=} \inf\left\{C: PLR \le e^{-\gamma}\right\} \qquad (4)$$

A linkszaturációs valószínűség vagy a csomagvesztési arány becslésére a jól ismert Csernov-korlát Baha-

dur-Rao-féle kiterjesztése alkalmazható. Ezzel a módszerrel a Csernov korláthoz képest pontosabb, de nem feltétlenül felső korlát jellegű becslést kaphatunk az előírt QoS mértékre [5,6]:

$$P(X > C) \approx \frac{1}{s^* \sqrt{2\pi\sigma^2(s^*)}} \exp(\Lambda_X(s^*) - s^* C) \quad \text{vagy} \quad (6)$$

$$PLR \approx \frac{1}{M(s^*)^2 \sqrt{2\pi\sigma^2(s^*)}} \exp(\Lambda_X(s^*) - s^* C) \quad (7)$$

ahol $\Lambda_X(s)$ az X valószínűségi változó logaritmikus momentumgeneráló függvénye (*LMGF*),

$$M \stackrel{def}{=} E[X], \ \sigma^2(s) = \frac{\partial^2}{\partial s^2} \Lambda_X(s), \ s^* = \arg\inf_s \{\Lambda_X(s) - sC\}$$

Látható, hogy a bemutatott becslések kiszámításához szükségünk van az aggregált forgalom pillanatnyi sebességeloszlásának logaritmikus momentumgeneráló függvényére. Ehhez azonban a vizsgált sztochasztikus folyamat minden momentumát ismernünk kellene, ami a gyakorlatban nem megoldható.

Erre problémára a következő szakaszban mutatunk megoldást, ahol három olyan momentumgeneráló függvény becslő módszert ismertetünk, amelyek paraméterigénye a források számára, csúcs adási sebességeire, valamint az aggregált forgalmi folyam átlagos adási sebességére korlátozódik.

3. Kevés paramétert igénylő momentumgeneráló függvény becslések

Az első módszer, amivel meghatározhatjuk az aggregált forgalmi folyam momentumgeneráló függvényének felső korlátját, Hoeffding 1963-ban publikált eredményeinek egyik következménye [2]. Legyenek X_i-k, *i = 1...n* független, korlátos valószínűségi változók, amelyekre

$$X = \sum_{i=1}^{n} X_i, \quad M \stackrel{def}{=} E[X], \quad 0 \le X_i \le p_i.$$

Ekkor *s>0* esetén,

$$G_X(s) \le \exp(sM) \exp\left(\frac{s^2 \sum_{i=1}^{n} p_i^2}{8}\right) \quad (8)$$

ahol $G_X(s)$ az X valószínűségi változó momentumgeneráló függvénye.

Hoeffding eredményeit felhasználva jutottak el Heszberger és mások [3] az alábbi – korlátos értékkészletű valószínűségi változók ($0 \le X_i \le p_i$) összegének

$\left(X = \sum_{i=1}^{n} X_i\right)$ momentumgeneráló függvényére vonatkozó – felső korláthoz:

$$G_X(s) \le \left(\frac{M + \sum_{k=1}^{n} \frac{p_k}{e^{sp_k} - 1}}{n}\right)^n \prod_{i=1}^{n} \left(\frac{e^{sp_i} - 1}{p_i}\right) \quad (9)$$

Az eddig bemutatott két korlát tehát Hoeffding eredményeit használja fel. Alkalmazhatunk azonban egy másik, újszerű megközelítést a momentumgeneráló függvény felső korlátjának becsléséhez. A harmadik korlát megkonstruálásánál a valószínűségi változók bizonyos típusú sztochasztikus sorbarendezését használjuk fel. Legyen adott két valószínűségi változó X és Y, melyek eloszlásfüggvénye F_X és F_Y. Ekkor azt mondjuk, hogy növekvő konvex sorbarendezés [4] szerint X kisebb, mint Y (azaz $X <_{icx} Y$), ha

$$\int_{-\infty}^{\infty} \phi(x) dF_X(x) \le \int_{-\infty}^{\infty} \phi(x) dF_Y(x)$$

teljesül minden olyan növekvő, konvex $\phi(x)$ függvényre, amelyre az integrál létezik.

A definícióból következik, hogy ha $X <_{icx} Y$, akkor *s>0* esetén $G_X(s) \le G_Y(s)$ fennáll. Ez könnyen ellenőrizhető, ha $\phi(x)$ helyére e^{sx}-et helyettesítünk.

A következő lemmát felhasználva egy új momentumgeneráló függvény felső korlát becslést kapunk [4]. Jelöljenek az $X_1^{onoff},...,X_n^{onoff}$ valószínűségi változók n független heterogén on-off forrást, amelyek csúcs adási sebességei rendre $p_1,...,p_n$, átlagos adási sebességeik pedig rendre $m_1,...,m_n$. Jelöljön $Y_1^{onoff},...,Y_{n_Y}^{onoff}$ n független homogén on-off forrást, amelyek csúcs adási sebességei azonosan

$p = \max(p_i, i=1,...,n)$, $n_Y = \text{int}\left\{\sum_{i=1}^{n} p_i / p\right\}$

(azaz a kapcsos zárójelek által határolt kifejezés egészrésze), átlagos adási sebességei azonosan

$m = \sum_{i=1}^{n} m_i / n_Y$. Ekkor

$$X_{onoff} <_{icx} Y_{onoff}, \qquad X_{onoff} \stackrel{def}{=} \sum_{i=1}^{n} X_i^{onoff}, \qquad Y_{onoff} \stackrel{def}{=} \sum_{i=1}^{n_Y} Y_i^{onoff}$$

A növekvő konvex sorbarendezés definíciójának következményét és a lemmát felhasználva az aggregált folyam momentumgeneráló függvényének felső korlátja a következő módon írható fel [8]. Legyenek az X_i-k, *i = 1...n* független, korlátos valószínűségi változók, amelyekre

$$X = \sum_{i=1}^{n} X_i, \ M \stackrel{def}{=} E[X], \quad 0 \le X_i \le p_i.$$

Ekkor *s>0* esetén,

$$G_X(s) \le \left(1 - \frac{M\left(1 + e^{sp}\right)}{n_Y p}\right)^{n_Y} \quad (10)$$

A továbbiakban a (8), (9) és (10) momentumgeneráló függvény korlátokra rendre a $\tilde{G}_{X,hoe}(s)$, $\tilde{G}_{X,ih}(s)$ és $\tilde{G}_{X,so}(s)$ jelölésekkel fogunk hivatkozni, míg az ezekhez tartozó LMGF-ekre a $\tilde{\Lambda}_{X,hoe}(s)$, $\tilde{\Lambda}_{X,ih}(s)$ és $\tilde{\Lambda}_{X,so}(s)$ jelöléseket használjuk majd.

4. Közvetlen módszerek az ekvivalens kapacitás meghatározására

A előző szakaszban bemutatott momentumgeneráló függvény közelítéseket alkalmazva a (6) és (7) formulákban, az adott aggregált forgalmi folyam szükséges paramétereinek és a linkkapacitás ismeretében becslést adhatunk az előírt QoS mérték (linkszaturációs va-

lószínűség vagy csomagvesztési arány) várható értékére. A becsült értéket összevetve az előírással (lásd (3)) hívásengedélyezési döntést hozhatunk.

Vegyük észre, hogy a (6) és (7) eredeti Bahadur-Rao formulákban nemcsak a logaritmikus momentumgeneráló függvény, de annak második deriváltja is szerepel.

A vizsgálatok azt mutatják, hogy mivel a momentumgeneráló függvényt sem ismerjük egzaktul (annak csak egy felső korlátját ismerjük), azért annak második deriváltjára végképp pontatlan becslésünk lesz, ami elrontja a formulák alkalmazhatóságát. Valahogy tehát ki kell küszöbölnünk az egyenletekből a második deriváltat. Ezt megtehetjük, ha alkalmazzuk Montgomery és Veciana eredményeit [7]:

$$P_{sat} \approx \exp(-I - \frac{1}{2}\log 4\pi I), \qquad (11)$$

$$PLR \approx \exp(-I - \frac{1}{2}\log 4\pi I - \log s^* M), \text{ ahol} \qquad (12)$$

$$I = -\inf_s \{\Lambda_X(s) - sC\}, \quad s^* = \arg\inf_s \{\Lambda_X(s) - sC\}.$$

Már korábban utaltunk rá, hogy a gyakorlatban a garantált QoS mérték várható értéke helyett jobban alkalmazható megoldás az aggregált forgalom ekvivalens kapacitását kiszámolni. Ebben az esetben ugyanis nem kell minden egyes újonnan érkező folyam belépésekor kiszámolni a várható QoS mértéket, hanem folyamatosan frissen tartva az ekvivalens kapacitás értékét az új folyamot beengedhetjük, ha a szabad kapacitás nagyobb, mint mondjuk a belépni kívánó folyam csúcs adási sebessége.

Ha a (4) formulákban alkalmazzuk a linkszaturációs valószínűségre (11) és a csomagvesztési arányra (12) vonatkozó egyszerűsített Bahadur-Rao-féle közelítéseket, indirekt módszert kapunk az ekvivalens kapacitás meghatározására. Ebben az esetben azonban kettős szélsőérték keresést (*C*-ben és *s*-ben is infimumot kell keresni) kell végrehajtanunk, ami nagyon megnöveli a módszer számításigényét.

Ennek a problémának a megoldására olyan direkt formulákat dolgoztunk ki, amelyek segítségével egy lépésben kiszámolható a keresett ekvivalens kapacitás. Ezek az alábbi formában írhatók:

$$\widetilde{C}^{B-R}_{equ,sat} \overset{def}{=} \inf_{s>0}\left\{\frac{\widetilde{\Lambda}_X(s)}{s} + \frac{\gamma}{s} - \frac{\gamma \log 4\pi\gamma}{s(1+2\gamma)}\right\} \qquad (13) \text{ és } (14)$$

$$\widetilde{C}^{B-R}_{equ,WLR} \overset{def}{=} \inf_{s>0}\left\{\frac{\widetilde{\Lambda}_X(s) + \gamma - 1 + \log M + \frac{2\gamma}{1+2\gamma}\log\frac{1+2\gamma}{4M\sqrt{\pi}\gamma^{\frac{3}{2}}}}{-\frac{1}{M} + s}\right\}$$

ahol $\tilde{\Lambda}_X(s)$ bármilyen megfelelő becslése $\Lambda_X(s)$-nek.

A most ismertetett eredmények részletesebb tárgyalása megtalálható [8]-ban.

5. Numerikus vizsgálatok

Ebben a szakaszban a bemutatott momentumgeneráló függvény közelítések és az azokat alkalmazó ekvivalens kapacitás formulák összehasonlító elemzését végezzük el numerikus példák segítségével. Ehhez egyszerű, két osztályú, on-off forrásokat tartalmazó forgalmi összeállítást definiálunk. Az egyes osztályokba tartozó források számát n_1-gyel és n_2-vel jelöltük. Az azonos osztályba tartozó források csúcs és átlagos adási sebessége azonos, ezeket rendre m_i és p_i, i∈{1,2} jelöli. A numerikus példákban alkalmazott forgalmi helyzetek jellemző paramétereit az *1. táblázatban* is összefoglaltuk. Az első forgalomkészlet (K1) tekinthető tömörítetlen hang és tömörített videófolyamok aggregátumának, míg a második (K2) forgalomkészlet tekinthető tömörítetlen és tömörített hang közös folyamjának. A két összeállítás közötti különbség a csúcs és az átlagos adási sebességek eltéréséből adódik (lásd a táblázat utolsó oszlopa).

A *2.-5. ábrákon* a linkszaturációs valószínűség és a csomagvesztési arány egzakt értékeinek és felső korlátjainak 10 alapú logaritmusát ábrázoltuk a *C* átviteli kapacitás függvényében. Mivel a bemutatott korlátok az

$$M < C < P \ (\mathrm{P} \overset{def}{=} \sum_{i=1}^{n} p_i \ )$$

intervallumban adnak értelmes eredményt, ezért az (*M,C*) intervallum egy részét ábrázoltuk úgy, hogy a linkszaturációs valószínűség és a csomagvesztési arány egzakt értékei ne legyenek kisebbek, mint 10^{-8}. Az egzakt értékeket folytonos vonallal, míg a korlátokat pont-vonallal ($\tilde{\Lambda}_{X,hoe}(s)$), szakasz-pont vonallal ($\tilde{\Lambda}_{X,ih}(s)$) és szakasz-pont-pont vonallal ($\tilde{\Lambda}_{X,so}(s)$) ábrázoltuk.

Az ábrákon látható, hogy általában a $\tilde{\Lambda}_{X,hoe}(s)$ korlátot használó közelítések a legpontatlanabbak, míg a másik két becslés pontossága elfogadhatóbb. A görbék közötti vízszintes és függőleges távolságok növekvő γ (szigorodó QoS előírás) esetén általában nőnek. A $\tilde{\Lambda}_{X,ih}(s)$ és $\tilde{\Lambda}_{X,so}(s)$ korlátokon alapuló becslések különbsége esetenként elenyésző, számításigényük azonban nagyban eltér: a sztochasztikus sorbarendezésen alapuló formula jóval könnyebben alkalmazható. A $\tilde{\Lambda}_{X,hoe}(s)$-t használó ekvivalens kapacitás becslés használata csak abban az esetben javasolható, ha a számítási sebesség döntő fontosságú.

A sávszélességigény-becslő eljárások teljesítőképességét is megvizsgáltuk. A numerikus analízist a következő módszerrel végeztük. Először egy adott *C* ér-

1. táblázat Az alkalmazott forgalmi összeállítások

	n_1	m_1 [kbit/s]	p_1 [kbit/s]	n_2	m_2 [kbit/s]	p_2 [kbit/s]	P/M
K1	100	51	64	10	200	500	2,24
K2	100	51	64	1000	4,8	5,8	1,34

ték mellett kiszámoltuk az egzakt linkszaturációs valószínűséget és a csomagvesztési arányt. Ezután a $P_{sat}=e^{-\gamma}$ és PLR=$e^{-\gamma}$ képletekből meghatároztuk a megfelelő γ értékeket, amelyeket végül az előzőekben bemutatott $\tilde{\Lambda}_{X,hoe}(s)$, $\tilde{\Lambda}_{X,ih}(s)$ és $\tilde{\Lambda}_{X,so}(s)$ korlátokkal együtt a (13), (14) képletekbe helyettesítettünk. Az így kapott ekvivalens kapacitás közelítések és annak egzakt értéke (C) közötti relációt vizsgáltuk.

A *6. és 7. ábrán* a $(\tilde{C}^{B\text{-}R}_{equ,sat}-C)/C$ relatív hibát ábrázoltuk a K1 és K2 forgalmi terhelés esetén. Folytonos vonallal a $\tilde{\Lambda}_{X,hoe}(s)$, pontvonallal a $\tilde{\Lambda}_{X,ih}(s)$, szakaszpontvonallal pedig $\tilde{\Lambda}_{X,so}(s)$ LMGF korlát felhasználásával kapott ekvivalens kapacitás becslés relatív hibáit ábrázoltuk. Vegyük észre, hogy a $\tilde{\Lambda}_{X,hoe}(s)$ alapú becslés súlyosan alulbecsli, míg a kettes számú forgalomkészlet esetén a $\tilde{\Lambda}_{X,so}(s)$ alapú becslés részben alulbecsli az egzakt ekvivalens kapacitás értéket.

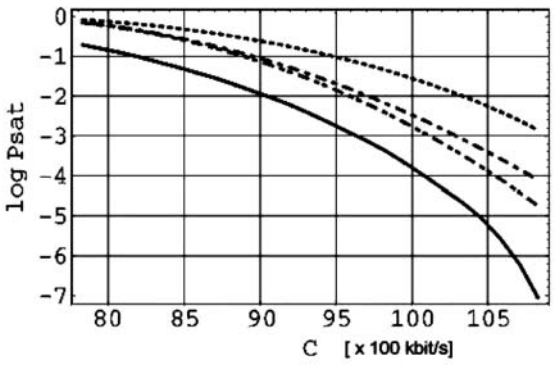
A *8. és 9. ábrán* a $(\tilde{C}^{B\text{-}R}_{equ,WLR}-C)/C$ relatív hibát ábrázoltuk a két forgalomkészlet esetén. Folytonos vonallal a $\tilde{\Lambda}_{X,hoe}(s)$ korlát, pontvonallal a $\tilde{\Lambda}_{X,ih}(s)$ korlát, szakaszpontvonallal pedig a $\tilde{\Lambda}_{X,so}(s)$ korlát behelyettesítésével számolt ekvivalens kapacitás becslések relatív hibáit ábrázoltuk.

A $\tilde{\Lambda}_{X,hoe}(s)$ korlátot használó sávszélességigény-becslő módszerek pontossága a legrosszabb (gyakran megengedhetetlenül pontatlanok) majdnem minden esetben, míg általában a sztochasztikus sorbarendezésen alapuló módszer relatív hibája a legkisebb. Megfigyelhető, hogy a relatív hibák közötti különbségek kisebb abszolút értékű egzakt sávszélességigény (tehát kisebb γ érték) esetén nagyobbak. Az is látható, hogy, a vizsgált formulák nagy valószínűséggel felső becslést adnak az egzakt sávszélességigényre, ha azokba a $\tilde{\Lambda}_{X,ih}(s)$ korlátot vagy a $\tilde{\Lambda}_{X,so}(s)$ korlátot helyettesítjük. A közelítések relatív hibáinak abszolút értéke γ növekedésével (azaz a QoS előírás szigorodásával) csökken.

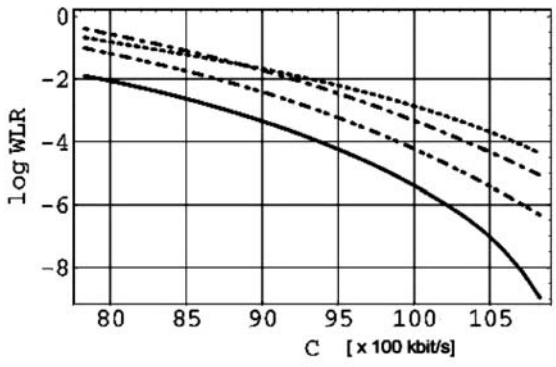
Összefoglalás

A cikkben újszerű sávszélességigény-becslő eljárásokat mutattunk be, amelyekkel a linkszaturációs valószínűségre vagy a csomagvesztési arányra vonatkozó garancia vállalása mellett a minimálisan szükséges átviteli kapacitás számolható. A bemutatott módszerek nagy előnye, hogy működésükhöz csak a forgalmi folyamok számát, adási sebességeik felső korlátját valamint az aggregált folyam átlagos adási sebességét kell ismernünk.

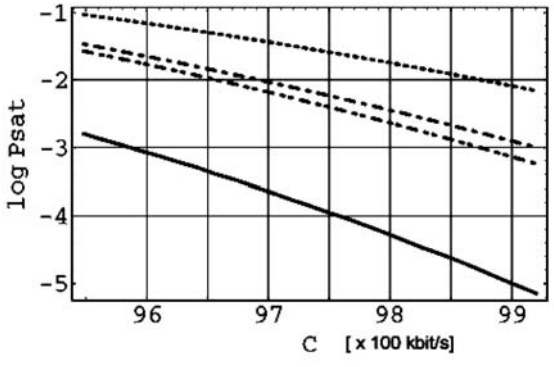
Az ismertetett ekvivalens kapacitás formulák kiszámolásához szükség van a vizsgált folyam statisztikai jellemzőinek pontos ismeretére, azonban erre vonatkozóan a rendelkezésre álló három paraméter alapján csak becslés adható. A cikk harmadik szakaszában ezért bemutattunk két ismert és egy újdonságnak számító momentumgeneráló függvény felső korlát becslő módszert. Habár ezen formulák paraméterigénye azonos, az ötödik szakaszban tárgyalt numerikus példákon keresztül megmutattuk, hogy alkalmazhatóságuk és pontosságuk eltér.



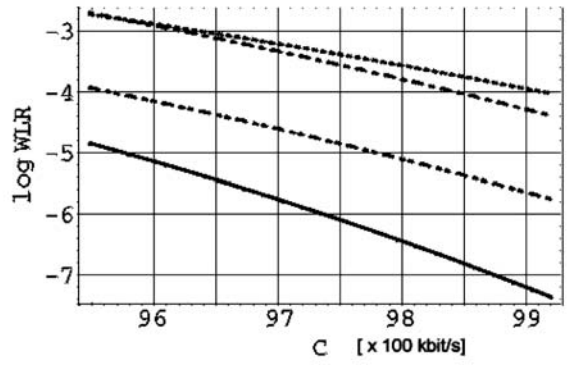
2. ábra Linkszaturációs valószínűség korlátok, K1



3. ábra Csomagvesztési arány korlátok, K1



4. ábra Linkszaturációs valószínűség korlátok, K2



5. ábra Csomagvesztési arány korlátok, K2

Numerikus vizsgálatainkat a sávszélességigény-becslő eljárásokra is elvégeztük. Az eredmények alapján megállapítottuk, hogy az esetek többségében az újonnan bemutatott, sztochasztikus sorbarendezésen alapuló momentumgeneráló függvény korláttal alkalmazott ekvivalens kapacitás becslések adják a legpontosabb eredményt.

Abban az esetben viszont, ha a számításigény és a döntési sebesség fontosabb a sávszélesség minél hatékonyabb kihasználásánál, egyszerűsége miatt továbbra is a jól ismert Hoeffding-féle (8) felső korlát alkalmazását javasoljuk.

Irodalom

[1] C. Bouchat, S. van den Bosch, T. Pollet, „QoS in DSL Access”, IEEE Communications Magazine, Vol. 41, no.9, Nov. 2003, pp.108–114.

[2] W. Hoeffding, „Probability Inequalities for Sums of Bounded Random Variables”, Journal of the American Statistical Association, 58:13–30, March 1963.

[3] Z. Heszberger, J. Zátonyi, J. Bíró, „Efficient Chernoff-based Resource Assessment Techniques in Multi-service Networks”, Telecommunication Systems, 20(1):59–80, 2002.

[4] G. Mao, D. Habibi, „Loss Performance Analysis for Heterogeneous On-Off Sources with Application to Connection Admission Control”, IEEE/ACM Transactions on Networking, 10(1):125–138, 2002.

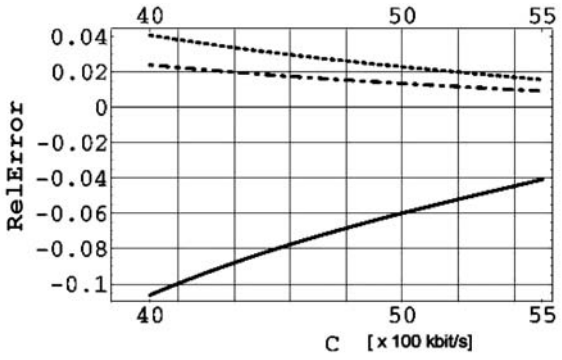
[5] R. R. Bahadur, R. Rao, „On Deviations of the Sample Mean”, Ann. Math. Statis., 31(27):1015–1027, 1960.

[6] J. Y. Hui, „Resource Allocation for Broadband Networks”, IEEE Journal on Selected Areas in Communications, 6(9):1598–1608, Dec. 1988.

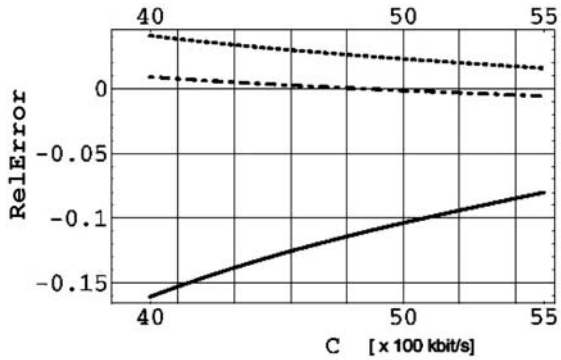
[7] M. Montgomery, G. de Veciana, „On the Relevance of Time Scales in Performance Oriented Traffic Characterizations”, Proc. of the Conference on Computer Communications, Vol. 2, pp.513–520, San Francisco, USA, March 1996.

[8] J. Bíró, Z. Heszberger, F. Németh, M. Martinecz, „Bandwidth Requirement Estimators for Quality of Service Packet Networks”, Proceedings of the International Network Optimization Conference, pp.95–100, Evry, Paris, Oct. 2003.

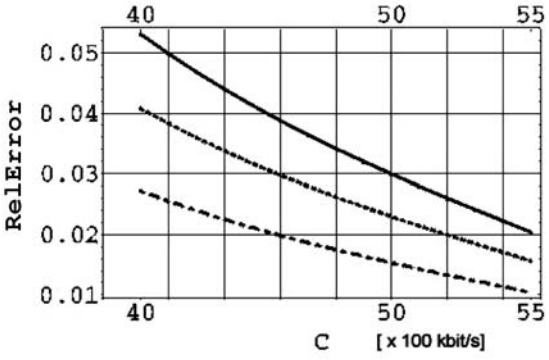
** A szerzőt az ETIK (www.etik.hu) támogatta*



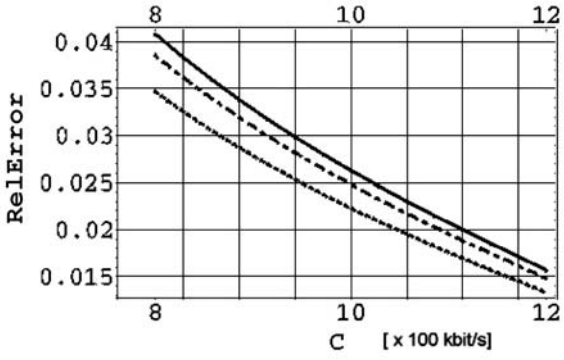
6. ábra $\tilde{C}^{B\text{-}R}_{equ,sat}$ becslések relatív hibája, K1



7. ábra $\tilde{C}^{B\text{-}R}_{equ,sat}$ becslések relatív hibája, K2



8. ábra $\tilde{C}^{B\text{-}R}_{equ,WLR}$ becslések relatív hibája, K1



9. ábra $\tilde{C}^{B\text{-}R}_{equ,WLR}$ becslések relatív hibája, K2

Forgalommenedzsment többszörös kapcsolatú tartományoknál

TAKÁCS ATTILA, CSÁSZÁR ANDRÁS, SZABÓ RÓBERT, HENK TAMÁS

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
{takacs, Andras.Csaszar, Robert.Szabo, henk}@tmit.bme.hu

Kulcsszavak: útválasztás, forgalommenedzsment, terhelésmegosztás

Az utóbbi időben egyre többen kutatják az IP hálózatok hatékony forgalommenedzsmentjének módjait. A korábbi kutatások jó része csak egyetlen autonóm hálózat belső forgalmának optimalizálásra irányult, és csak néhányan vizsgálták a tartományok közötti forgalommenedzsment kérdéseit. Ennek az oka az, hogy míg egy önálló tartományon belül teljes információ és teljes hatáskör áll a forgalommenedzsment rendelkezésére, addig a tartomány határain kívül igen korlátozottak mind az információgyűjtési, mind az útvonalakat befolyásoló lehetőségek. Cikkünkben felvázoljuk a tartományok közötti forgalommenedzsment nehézségeit, és bemutatjuk a BGP által nyújtott lehetőségeket. Ezen felül adunk egy módszert, mellyel a már kifinomult tartományon belüli forgalommenedzsment módszerek kiterjesztésével a szomszédos hálózatok közötti forgalmat szabályozhatjuk.

Bevezető

Az utóbbi időben egyre többen kutatják az IP hálózatok hatékony forgalommenedzsmentjének *(Traffic Engineering, TE)* módjait. A forgalommenedzsment célja tágabb értelemben a felhasználói folyamok minőségi igényeinek támogatása úgy, hogy közben hatékony, gazdaságos és megbízható módon aknázzuk ki a hálózat lehetőségeit. Szűkebb értelemben a forgalommenedzsment sokszor csak a hatékony és megbízható hálózatkihasználtságra utal.

A korábbi TE kutatások jó része csak egyetlen autonóm hálózat belső forgalmának optimalizálásra irányult [1–7]. Ennek eredményeképpen mára számos jól használható javaslat született tartományokon belüli *(intra-domain)* TE megvalósítására, jóllehet gyakorlati alkalmazásuk még nem széleskörű. Ezek a módszerek többnyire megpróbálják csökkenteni a torlódás kialakulásának valószínűségét. Az útválasztókat statikus vagy dinamikus módon képessé teszik a tipikusan használt legrövidebb utak mellett más, alternatív utak használatára is. Ezt úgy érik el, hogy mintegy „elkenik" a forgalmat a hálózatban, így a terhelést megosztják több útvonal között *(load balancing, load sharing)*.

Csak néhányan vizsgálták a tartományok közötti *(inter-domain)* forgalommenedzsment kérdéseit. Ennek az oka az, hogy míg egy önálló tartományon belül teljes információ és teljes hatáskör áll a forgalommenedzsment rendelkezésére, addig a tartomány határain kívül igen korlátozottak mind az információgyűjtési, mind az útvonalakat befolyásoló lehetőségek. Ez nem is csoda, hiszen az Internet hierarchikus szervezésű, azaz jól elkülöníthető az autonóm hálózati tartományokon belüli útválasztástól a tartományok között útválasztás. A hierarchikus szerveződést egyrészt a jobb skálázhatóság indokolja, hiszen így egy útválasztónak nem kell ismernie a világ minden egyes csomópontjához vezető utat, elég ha csak az adott hálózat felé vezető utat ismeri. Hálózatból márpedig sokkal kevesebb van, mint csomópontból. A kétszintű szerveződés másik fontos indokát üzleti és politikai okok adják, mint például a versenyhelyzetből adódó információ és topológia elrejtés igénye, vagy, hogy az egyes operátorok maguk akarják meghatározni, hogy mely célcímek felé vállalnak csomagtovábbítást, mely szomszédokon keresztül, vagy épp mely szomszédoktól hol fogadnak el csomagokat.

Ezeket az üzleti/politikai igényeket tökéletesen kielégíti a mindmáig egyetlen gyakorlatban alkalmazott tartományok közti útvonalválasztó protokoll – a *Border Gateway Protocol (BGP)* [16]. Ennek a kifejlesztésekor még fel sem merültek a forgalommenedzsment igényei, ehelyett technikailag a stabilitás, robosztusság és a skálázhatóság voltak a főbb szempontok.

Mára azonban nyilvánvalóvá vált, hogy egy önálló hálózat működésének jelentős költségét jelenti a tartományból kimenő forgalom [8]. Ezért a hálózatközi forgalommenedzsment jól szolgálhatja az egyes tartományok üzemeltetőit, hogy optimalizálják kimenő forgalmukat, és így csökkentsék költségeiket.

További fontos kérdés az egyes tartományok összekapcsolása (melyik tartományt melyikkel kössük össze, hányszor, mely csomópontokon keresztül), amelyek helyes megválasztása jelentősen csökkentheti és gyorsíthatja a tartományok közötti forgalmat. Persze nem csupán a gazdaságosság az a kérdés, amelyre a tartományok közötti forgalommenedzsment választ adhat, hanem az egyre fontosabb szolgáltatás minőségi (QoS) elvárások is [9]. Ahogy egy tartományon belül, úgy tartományok között is szükség van a megfelelő késleltetés előírások és adatsebesség igények teljesítésére. Ebben segíthet a megfelelő útvonalak megtalálása és lefoglalása, melynek eredménye a minőségi szolgáltatás.

A cikkben felvázoljuk a tartományok közötti forgalommenedzsment nehézségeit, és bemutatjuk a BGP

által nyújtott lehetőségeket. Ezen felül adunk egy módszert, mellyel a már kifinomult tartományon belüli forgalommenedzsment módszerek kiterjesztésével a szomszédos hálózatok közötti forgalmat szabályozhatjuk.

Mielőtt rátérnénk a BGP és TE kérdéseire, bemutatjuk a tartományok összekapcsolásának szempontjait.

Tartományok csoportosítása

Az Interneten, mint IP hálózatok hálózatán, alapjában véve két tartomány típust különíthetünk el, a vég- és a tranzithálózatokat *(stub/transit domain)*.

Beszélhetünk tehát:

• *Egyszerű véghálózatról,* mely csak egy összeköttetéssel kapcsolódik a „külvilághoz", és átmenő forgalmat nem szállít. Tipikusan ilyen egy kisebb szervezet üzemi- illetve magánhálózata, például egy cég telephelyen létesített hálózat, egy irodai hálózat stb;

• *Többszörös kapcsolatú véghálózatról,* mely több összeköttetéssel is rendelkezik az Internet felé, de átmenő forgalmat nem bonyolít. Ez is tipikusan üzemi- illetve magán hálózat, de olyan, melynél az elérhetőség és a rendelkezésre állás kulcsfontosságú, ezért több kapcsolatot tart fent a külvilággal;

• *Tranzit hálózatról,* amely egy olyan szolgáltatói tartomány, ami átmenő forgalmat szállít. Ezek a hálózatok azok, melyek összekötik a kisebb magánhálózatokat és tulajdonképpen az Internet magját képezik.

Egy másik felosztás a tartományok gazdasági kapcsolatrendszere alapján lehetséges. Kétféle kapcsolattípust különítünk el. A *szolgáltató-ügyfél* viszonyt, ahol az ügyfél fizet a szolgáltatónak, és cserébe a szolgáltató biztosítja a külvilághoz való kapcsolatot, szerződésben rögzített feltételekkel. Az előző felosztásban ismertetett véghálózat–tranzit hálózathoz kapcsolat egy példa szolgáltató–ügyfél kapcsolatra. A *közvetlen (peering)* kapcsolat esetén két tartomány egyenrangúként kapcsolódik össze, megosztva ennek költségét. Az egymás közötti forgalmat ezen az ún. első választású útvonalon keresztül bonyolítják, megkerülve a magasabb hálózati szinteket, így költséget takarítanak meg.

A tartományokat ezen kívül még hierarchikus elhelyezkedésük alapján is csoportosíthatjuk:

- *Tier-1:* globális méretű hálózatok, melyek az Internet gerincét alkotják,
- *Tier-2:* nemzeti méretű hálózatok és
- *Tier-3:* regionális hálózatok, a helyi hozzáférést biztosítják, tipikusan ezekhez kapcsolódnak a magán hálózatok

A különböző szintekbe sorolt tartományok mind kapacitásban mind számosságban jelentős eltérést mutatnak. Míg *Tier-1*-es hálózat csupán néhány van, addig a *Tier-3* és ez alatti hálózatok rendkívül nagy számosságúak. Ha még arra is gondolunk, hogy a tartományok mindegyikében ott kell legyen az összes másik tartomány elérhetőségi információja, akkor nem meglepő, hogy a skálázhatóság egy igen fontos szempont a tartományok közötti információ cserében.

Tartományok közti útválasztás BGP-vel

Mint korábban említettük, az Interneten élesen elkülönül a tartományon belüli és a tartományok között útválasztás. Az utóbbi célra a gyakorlatban egyetlen egy megoldást alkalmaz minden hálózati szolgáltató, a BGP-t. Egy tartomány a BGP segítségével tudja közölni a szomszédaival, hogy rajta keresztül mely címhalmazok, azaz *prefixek,* érhetőek el. A BGP távolságvektor alapú útvonalválasztást használ, ami annyit tesz, hogy egy címhalmaz elérhetőségét hirdető üzenetben a szükséges útvonal az útba eső tranzit tartományok listájaként kerül feltüntetésre. Idáig nem is tűnik bonyolultnak a dolog, azonban a BGP leglényegesebb része, amely jelentős befolyással van az útvonalválasztásra, egy szövevényes szabályrendszer *(policy),* és ez az, ami a BGP rugalmasságát, de sajnos a komplikált kezelhetőséget is adja. Ezen szabályokat minden AS lokálisan határozza meg saját maga számára, és ezzel szabályozza a saját tartományok közötti útvonal választását, így gyakorolva – közvetett – hatást a többi tartományra.

Két szabályrendszert különítünk el. Az úgynevezett import szabályok mondják meg, hogy az egyes kívülről érkező útvonalak közül a tartomány mely útvonalakat használja. Az export szabályok pedig meghatározzák, hogy az autonóm hálózat mely útvonalakat hirdeti tovább szomszédai számára. Ezek a szabályok lehetőséget adnak arra, hogy meghatározzuk, hogy milyen útvonalhirdetéseket fogadunk el illetve adunk tovább. Ilyen módon egy üzemeltető megteheti például, hogy hiába érhető el tőle egy versenytársa, ő ezt az elérhetőséget nem hirdeti más versenytársainak, csak saját ügyfeleinek. Hiszen neki nem érdeke a konkurencia forgalmának szállítása. A szabályok az üzleti érdekek mellett azt a célt is szolgálják, hogy ha egy útválasztó egy célcím-halmaz elérhetőségéről több különböző hirdetést is kapott, akkor az útválasztó képes legyen kiválasztani az egyetlen legjobbat.

A BGP azon tulajdonsága, hogy csak egyetlen utat jelöl ki használatra, illetve továbbhirdetésre a lehető legnagyobb stabilitást szolgálja. Ugyanakkor ezen tulajdonságának is köszönheti, hogy alig alkalmas a forgalommenedzsment támogatására. Ha egy BGP útválasztó több különböző hirdetést kap ugyanazon célcímhalmaz felé, akkor a hirdetésekben megtalálható attribútumok segítségével dönti el a preferencia sorrendet, s a végén csak a legmagasabb preferenciájú útvonallal foglalkozik.

Itt most csak két fő attribútumra térünk ki, melyeket a későbbiekben használni fogunk. A `local_preference` lehetőséget ad a tartomány adminisztrátora számára, hogy meghatározza, hogy melyik kimenő útvonalválasztót használják a belső forgalomirányítók a tartományok közötti forgalom bonyolítására. Minden határcsomóponthoz rendelhető egy ilyen preferencia, és a beérkező hirdetéseket úgy szűrjük meg, hogy azt az útvonalválasztót használjuk a lehetségesek közül, amelynek a legnagyobb a preferenciája. A `multi_`

`exit_discriminator` (MED) hasonlóan használható, csak itt mi kérhetjük a szomszéd tartományt, hogy az ebben az attribútumban beállított értékek alapján válassza ki melyik bemeneti csomópontunkon át továbbítsa felénk a forgalmat. Ehhez persze szükséges, hogy több kapcsolatunk legyen a szomszédunkkal, hiszen különben nincs értelme a MED használatának. Másrészt meg kell egyezni a szomszéd hálózat operátorával, hogy valóban vegye figyelembe az általunk kért preferenciákat. A BGP TE lehetőségei iránt érdeklődő olvasó figyelmét felhívjuk a következő kitűnő összefoglalást adó cikkekre: [13,14,15,17].

Mint láttuk, a szabályok meghatározásában nagy szerepe van az üzleti szempontoknak. Ez az „üzleti szemlélet" tovább szűkíti a TE lehetőségeket, hiszen jól tükrözi, hogy itt nem lehet az egyes tartományokat vagy linkeket egyenrangúnak tekinteni, mint azt lehetett a tartományon belüli TE esetében. Így az sem csoda, hogy igen szűkösek az információ szerzési lehetőségek is, mivel egyik szolgáltató sem akarja felfedni saját belső hálózatának még a topológiáját sem, nemhogy a kihasználtság információit. Ilyen megszorítások mellet a legésszerűbb TE lépés az lehet, ha kis lépést teszünk, és nem az egész Internet forgalmát akarjuk egyszerre kezelni.

A legjobb kiindulás a tartományunkból kimenő forgalom megfelelő menedzselése. Ezzel lehetőségünk van a kimenő forgalmunk és a tartományon belüli forgalommenedzsment összehangolására. Az üzemeltető saját hatáskörében tudja optimalizálni – akár költségesség szempontjából is – a tartomány kimenő forgalmát figyelembe véve a kimenő linkek telítettségét, és a tartomány belső terhelés kihasználtság eloszlását. A kimenő forgalom menedzseléséhez minden szükséges információ rendelkezésre áll, hiszen a tartományunk kimenő linkjeinek telítettségét az üzemeltető maga mérheti hasonlóan a belső linkek kihasználtság méréséhez, így a hálózati adminisztrátor képes a tartományon belül működő terhelés megosztó megoldásának kiterjesztésére egy lépéssel a hálózat határain túl.

A továbblépés előtt, bemutatunk néhány terhelésmegosztó módszert.

Forgalomelosztás tartományon belül

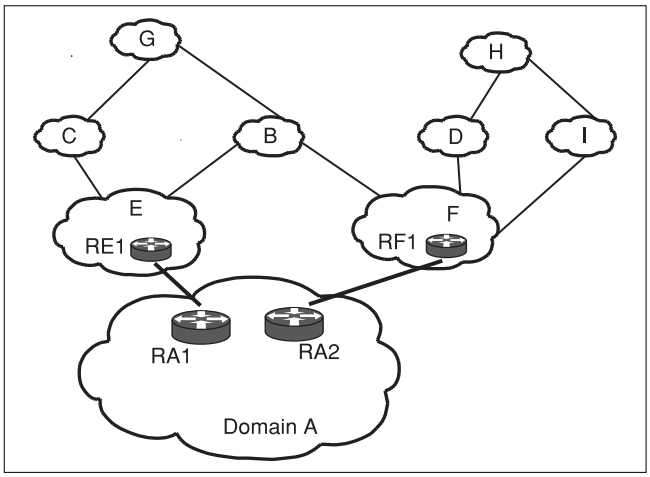
A hálózaton belüli útválasztó protokollok (OSPF vagy IS-IS) a legrövidebb utakat keresik meg a célcsomópontig, és ezek közül is csak egyet használnak. Az utak hosszát a felhasznált élek költségeinek összege adja. A terhelésmegosztás érdekében az irodalomban javasolt eljárások megpróbálnak több útvonalat is kihasználni a forgalom terelésekor. Statikus megoldásról akkor beszélünk, ha a terhelésmegosztás nem veszi figyelembe a hálózat változó állapotát és a kihasználtsági viszonyokat. A dinamikus megoldások figyelik a hálózat állapotát, és a torlódott utakról elterelik a forgalom egy részét a több szabad erőforrással rendelkező útvonalak felé.

A legtöbb útválasztóban megtalálható *ECMP (Equal Cost Multi-Path)* [11] egy egyszerű, statikus terhelés elosztó megoldás, mely továbbra is csak legrövidebb utakkal dolgozik, azonban ha több egyforma legkisebb költségű út is létezik egy forrás/cél csomópont pár között, akkor ezekre egyenlő arányban tereli a forgalmat. Annak érdekében, hogy az egy folyamhoz tartozó csomagok azonos útvonalon haladjanak, az útválasztók az egy folyamra nézve közös csomagmezőkből (forrás/cél IP cím, forrás/cél port, protokollazonosító) hash eljárással egy adott intervallumba eső véletlenszerű számot képeznek. Az intervallum egyforma méretű részintervallumokból áll, melyek mindegyikét az útválasztó egy-egy alternatív kimeneti kapcsolathoz rendeli. Attól függően dől el egy adott csomaghoz használt kimeneti kapcsolat, hogy a hash érték mely részintervallumba esik.

Egy dinamikus megoldás az *OMP (Optimised Multi-Path)* [10]. Egy forrás/cél csomópont pár között több alternatív útvonalat képes felhasználni, mint az ECMP, mert egy útválasztó minden olyan szomszédot alternatívának tekint, amely közelebb van a célponthoz, mint saját maga. Ezáltal ugyan nem feltétlenül a legrövidebb utakat használja, mégis elkerüli, hogy kör alakuljon ki az útvonalban, hisz egy csomag nem juthat vissza egy korábban már érintett útválasztóhoz. Az alternatív utak közötti terhelésmegosztás hasonlóan megy, mint az ECMP-nél, hash eljárás és részintervallumok segítségével. Az OMP azonban a részintervallumokat nem egyforma hosszúra választja, hanem a hozzárendelt útvonalak kihasználtsághoz igazítja: minél telítettebb egy út, annál kisebb részintervallum tartozik hozzá. Működéséhez szükséges a hálózatban a kihasználtsági információk periodikus terjesztése. Az OMP egyik problémája, hogy ha változik a hálózat állapota, megváltoznak a részintervallumok hosszai, így lehet, hogy egy hosszabb ideig bent lévő folyam, melynek hash értéke változatlan, egyszer egyik útvonalhoz tartozik, másszor egy másik intervallumba esik. Ez azt jelenti, hogy egy folyam útvonala változhat, azaz a csomagok érkezési sorrendje különbözhet az indulásitól, és az átviteli késleltetés is ingadozhat. Ezek csökkentik a TCP teljesítményét is, egyúttal a valós idejű forgalmak minőségét is rontják.

Az általunk javasolt *Core State Limited Load Sharing (CSLLS)* [12] eljárás csökkenti az útvonalak megváltózásának a valószínűségét azáltal, hogy fix ideig garantálja az útvonal választási döntések stabilitását. Ezt úgy éri el, hogy a kihasználtsági viszonyokhoz való alkalmazkodáskor csak az újonnan induló folyamok útvonalát befolyásolja, a korábban beengedett folyamokét nem. Legegyszerűbb esetben az új hívásokra vonatkozó részintervallumok súlyai arányosak a hozzájuk tartozó útvonalak szabad kapacitásaival.

Szimulációs vizsgálatainknál a CSLLS tartományon belüli forgalomelosztó megoldást terjesztettük ki hálózatközi működésre.



1. ábra Tartományok és kapcsolatuk

Prefix	Kimenet	Útvonal
B	RA1-RE1	A-E-B
B	RA2-RF1	A-F-B
C	RA1-RE1	A-E-C
D	RA2-RF1	A-F-D
E	RA1-RE1	A-E
F	RA2-RF1	A-F
G	RA1-RE1	A-E-B-G
G	RA2-RF1	A-F-B-G
H	RA2-RF1	A-F-D-H
I	RA2-RF1	A-F-I

1. táblázat Az „A” tartomány ismeretei

A tartományon belüli forgalom-menedzsment kiterjesztése

A hálózatközi útválasztás megértéséhez tekintsük az alábbi példát *(1. ábra)* az „A” többszörös kapcsolatú véghálózat szemszögéből. A BGP protokoll segítségével az „A” tartomány tudja, hogy elérheti a „B”-től „I”-ig terjedő valamennyi címhalmazt *(prefix*eket). Ismeri azt is, hogy mely kimeneti éleken (RA1-RE1 vagy RA2-RF1) kell küldenie a csomagokat az egyes címek felé, sőt az útba eső autonóm hálózatok azonosítóit is ismeri. Az „A” tartomány hálózatközi útvonalainak ismereteit az *1. táblázat* mutatja. Vegyük figyelembe, hogy a BGP működéséből adódóan egy tranzit hálózat egy prefixről több útvonalon is kaphat hirdetést, azonban ő maga ebből csak az egyiket hirdetheti tovább. Jelen példában az „E” tartomány megkaphatja a „G” címhalmaz útvonalát az „C” hálózaton keresztül és a „B” hálózaton keresztül is, azonban az „A”-nak ebből csak az egyiket adhatja tovább. Amint azt a táblázat mutatja, ebben a példában az „E” tranzit hálózat a „G” prefix felé csak a „B” hálózaton keresztülvezető útvonalat hirdette tovább „A”-nak (a hírdetések hátterét a „Tartományok közti útválasztás BGP-vel” fejezetben tárgyaltuk).

Amint az a táblázatból látható, az „A” tartomány a „B” és a „G” címhalmazok felé is rendelkezik alternatív útvonal ismeretekkel a BGP használatával is, de ezt a mai útválasztók nem képesek tudatosan kihasználni. A BGP protokoll lehetőséget ad arra, hogy az önálló tartomány súlyozza kimeneteit a `local_preference` metrika segítségével. Lényegében ennek segítségével dönti el a tartomány, hogy mely útvonalát használja illetve hirdesse tovább, ha több is van. Egy egyszerű útválasztási megoldás, ha a tartományon belüli útválasztók, ha egy célcímhez több kimenet tartozik, akkor azt választják, amely a `local_preference` metrika szerint a preferáltabb.

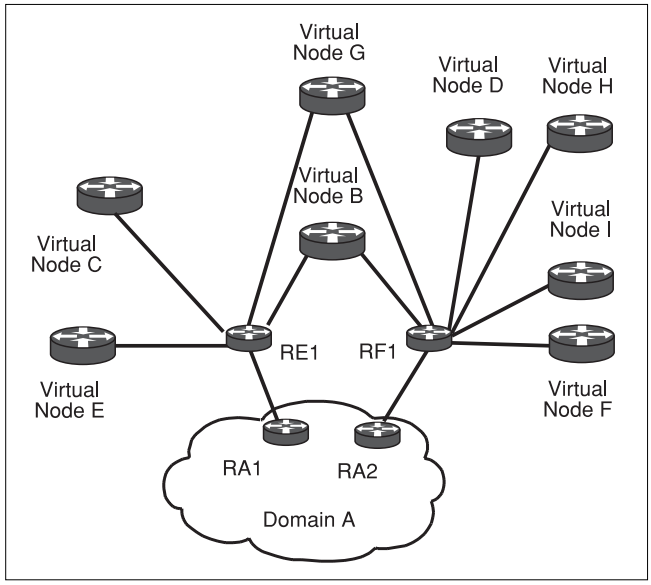
Jelenleg azonban egy tartományon belül az útválasztás nem az előző módon, hanem tipikusan a találóan „forró krumpli útválasztásnak” *(hot potato routing)* nevezett módon történik. Ennek célja, hogy a csomagoktól a lehető legrövidebb úton szabaduljon meg a hálózat (mint egy forró krumplitól), azaz a hálózaton belüli útválasztó a legközelebbi alkalmas kimeneti csomópont felé továbbítja a csomagot. Amint látható, ez a módszer ugyan több kimenetet használ, azaz megosztja köztük a terhelést, de ez nem tudatos, nehezen befolyásolható és nem veszi figyelembe a kimeneti kapcsolatok telítettségét.

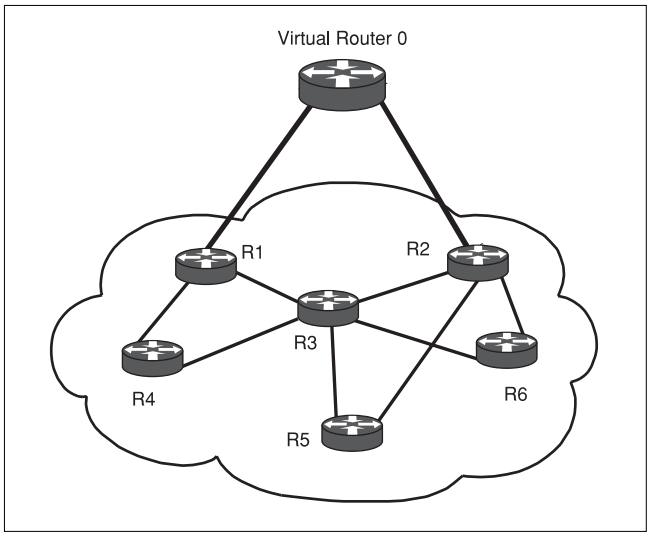
A javaslatunk szerint az „A” tartomány a hálózaton belül működő dinamikus terhelésmegosztó architektúráját egyszerűen, de tudatosan kiterjesztheti hálózatközi térbe is.

Ehhez először vázoljuk grafikusan a táblázat első két oszlopának információit *(2. ábra)*, azaz hogy mely kimeneti kapcsolatokon mely célcím halmazok érhetők el. Ha ezeket a célcím halmazokat mint virtuális útválasztókat tekintjük, akkor az így kapott virtuális hálózaton már használhatjuk a tartományon belül korábban használt terhelésmegosztó megoldásunkat, mellyel például nagyobb átbocsátóképességet nyerhetünk a “B” vagy “G” célcímek felé.

Az elérhető prefixek ilyetén leképzésével és felhasználásával tehát a már meglévő IP hálózatokban is képesek vagyunk terhelésmegosztást megvalósítani. Az elérhető nyereségről a korábbi útválasztó eljárásokhoz képest a következő fejezetben írunk.

2. ábra
Tartományok leképzése az „A” tartomány szempontjából





3. ábra A vizsgált tartomány leképzett képe

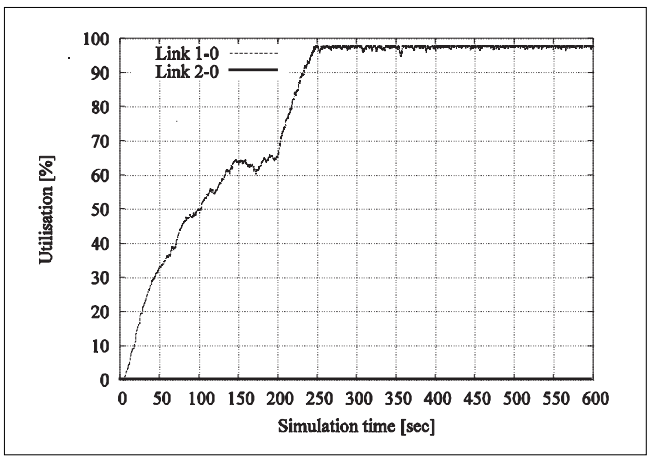
Vegyük persze figyelembe, hogy az autonóm hálózaton kívüli ismereteink szűkösek. A közvetlen a hálózatból kimenő kapcsolatok telítettségén kívül például biztos, hogy semmilyen információval nem rendelkezünk a virtuális útválasztók felé menő kapcsolatok kihasználtságáról, vagy aktuális szabad kapacitásáról. Ebben a cikkben azonban csak az a célunk, hogy a tartományunkból kimenő forgalmat optimalizáljuk.

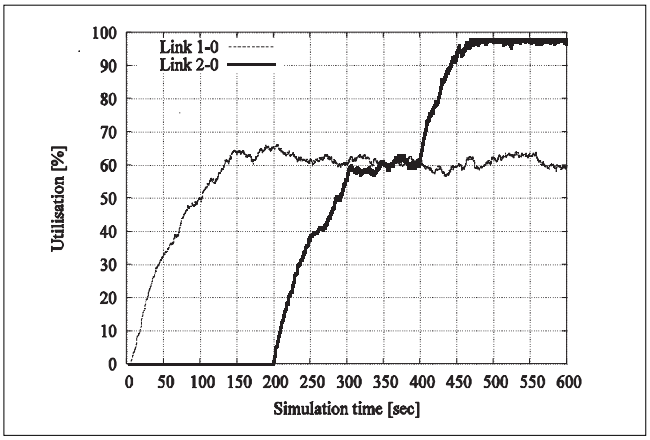
Numerikus eredmények

A következőkben néhány szemléletes eredményt mutatunk be. A *3. ábra* mutatja az általunk vizsgált többszörös kapcsolatú végtartomány által a külvilágról leképzett virtuális képet. Az egyszerűség kedvéért bármely címtartomány elérhető mindkét határoló csomóponton keresztül, így a hálózatunkon kívüli teljes világ egy virtuális csomópontnak tekinthető.

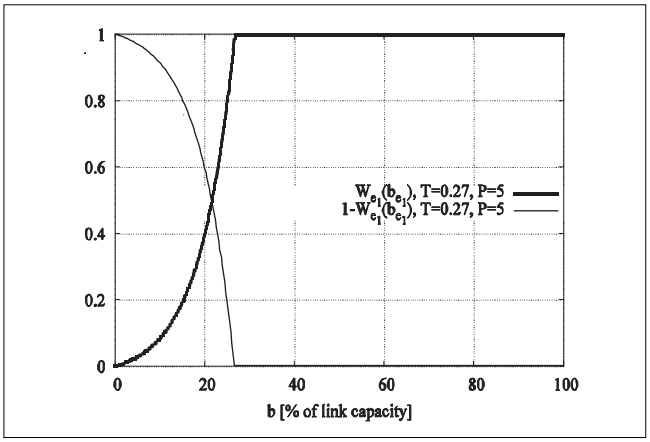
A vizsgálatokat a *Network Simulator* csomagszintű hálózati szimulációs szoftverrel végeztük. Hogy jól megfigyelhetőek legyenek az algoritmusok sajátosságai, lépésenként növekvő forgalmat generáltunk. Ezt úgy értük el, hogy először csak a 4-es csomópontról érkezett kimenő forgalom, majd az 5-ös csomópont is bekapcsolódott az adatküldésbe, és végül a 4-es, 5-ös és 6-os csomópont mind generált forgalmat.

4. ábra Egyszerű BGP megoldás





5. ábra BGP megoldás IGP metrika alapján

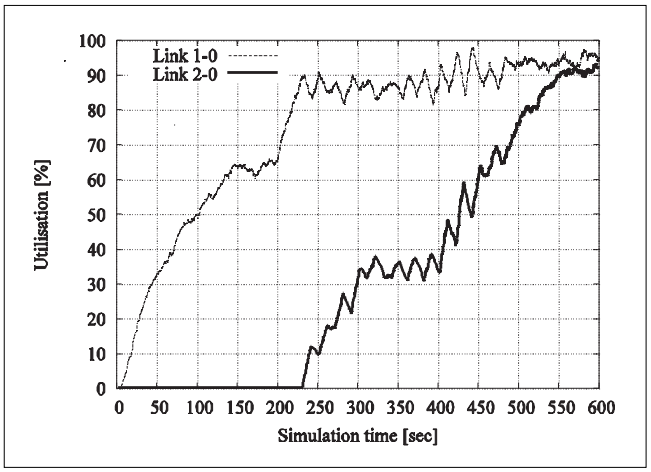


6. ábra Egy lehetséges súlyfüggvény a local_preference figyelembevételéhez

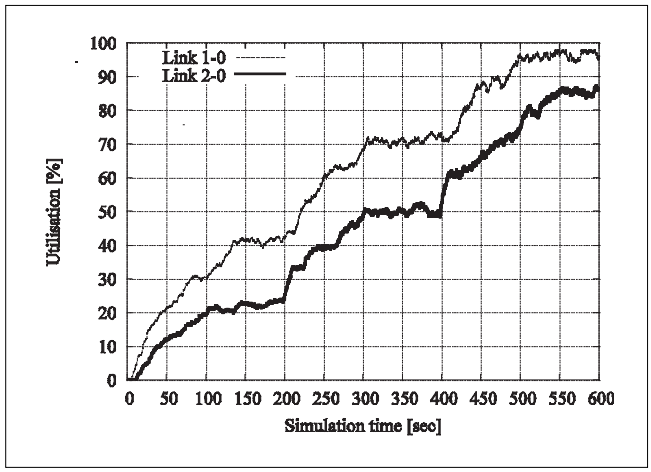
Nézzünk meg két alap BGP esetet. Először is, ha az egyik link `local_preference` attribútuma nagyobbra van állítva, ezzel például tükrözve, hogy ez az elsődleges útvonal és a másik csak egy tartalék kapcsolat. Ekkor minden kimenő forgalom ezt a linket fogja igénybe venni, és annak ellenére, hogy ezen torlódás alakul ki, a másik link kihasználatlan marad. Ezt mutatja a *4. ábra*.

Természetesen, ha nem csak védelmi útvonalként kívánjuk használni a másik linket, akkor lehetőségünk van ezt a BGP-vel is megvalósítani. Erre egy módszer, ha a belső útvonalválasztó protokoll *(Interior Gateway Protocol – IGP)* által látott legrövidebb útvonalat használjuk a kimenő forgalom mielőbbi kijuttatására a hálózatunkból. Ez esetben azonban az elérhető terhelés kiegyenlítés a topológiai sajátosságoktól függ. Ezt jól mutatja a *5. ábra*, ahol látszik, hogy amint a forgalom érkezése topológiai szempontból előnytelen, akkor a terhelés kiegyenlítés rögtön felborul.

Kínálkozik azonban egy jó alternatíva, hogy egyesítsük a terhelés kiegyenlítés és a gazdaságossági szempontok előnyeit. Lazíthatjuk a local_preference



7. ábra Terhelés kiegyenlítés a local_preference attribútum alapján



8. ábra Terhelés kiegyenlítés a MED attribútum segítségével

attribútum kötöttségeit, ha egy súlyfüggvényt alkalmazunk, amely egy bizonyos terheltségig határozottan preferálja az elsődleges linket, de annak túlzott telítődése esetén a védelmi utat is elkezdi terhelni, hogy elkerülje a linkek telítődését. Egy lehetséges függvényt mutat a *6. ábra*. Az ennek segítségével kapott link terheléseke a *7. ábra* mutatja.

Egy másik lehetőség a `MED` attribútum alkalmazásával tesz lehetővé terhelés kiegyenlítést. Ennek segítségével a szomszédos tartomány belső terhelésének jobb eloszlására nyílik lehetőség, azáltal, hogy az általa kívánt arányban osztjuk meg a bemenő forgalmát, mely a mi kimenő forgalmunk. Egy egyszerű módszer lehet, hogy az egyes tartományok közötti linkekre terjesztett `MED` értékeket súlynak tekintve, a súlyok valamilyen függvényeként terheljük meg az egyes linkeket. Egy egyszerű függvény alkalmazásának eredménye a *8. ábra,* ahol az 1–0 linket nagyobb súlyúnak állítottunk be a `MED` segítségével.

Összegzés

A tartományok közötti TE egy igen fontos kérdés lehet a hálózatok közötti forgalmak optimalizálásában. Sajnos az igen erős gazdasági érdekeltségek illetve érdekellentétek jelentősen megnehezítik, sőt akár lehetetlenné is teszik, egy globális méretekben gondolkodó TE megoldás alkalmazását.

Egy frappáns megoldás lehet az egyes tartományok üzemeltetői számára, hogy a korlátozott lehetőségeket számukra a lehető leghatékonyabb módon kihasználják. Mi ehhez adtunk egy javaslatot cikkünkben. Nevezetesen, a tartományon belül alkalmazható – esetleg már alkalmazott – TE megoldást összehangoljuk a tartomány kimenő forgalmának hatékony irányításával. Így a tartomány belső terheltség elosztását – mely teljes hatáskörünkben van – úgy irányíthatjuk, hogy az a tartomány számára legköltségesebb adatáramlást – a tartomány kimenő forgalmát – optimalizálja.

Irodalom

[1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, X. Xiao, "Overview and principles of internet traffic engineering" RFC 3272, IETF, 2002. május

[2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, "Requirements for traffic engineering over mpls," RFC 2702, IETF, 1999. szeptember

[3] Daniel O. Awduche, "Mpls and traffic engineering in IP networks," IEEE Communications Magazine, Vol. 37, no.12, pp.42–47, 1999. december

[4] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, "Netscope: Traffic engineering for IP networks," IEEE Network Magazine, Vol. 12, no.2, pp.11–19, 2000. március

[5] Ashwin Sridharan, Roch Guérin, Christophe Diot, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," Tech. Rep., University of Pennsylvania, 2003. (egy rövidebb verzió megjelent az InfoCom'2003 konferencia kiadványban)

[6] Yufei Wang, Zheng Wang, Leah Zhang, „Internet traffic engineering without full mesh overlaying", InfoCom'2001, Anchorage, Alaska, 2001. április

[7] Bernard Fortz, Mikkel Thorup, "Internet traffic engineering by optimizing OSPF weights", IEEE InfoCom'2000, pp.519–528, Tel-Aviv, Israel, 2000. április

[8] Tracie E. Monk, "Inter-domain Traffic Engineering: Principles and Case Examples," INET2002, 2002. június

[9] Raymond Zhang, J.P. Vasseur, "Mpls inter-as traffic engineering requirements," Internet Draft, 2004. január <draft-ietf-tewg-interas-mpls-te-req-04.txt>

[10] C. Villamizar,
"OSPF optimized multipath (OSPF-OMP),"
A 44-ik IETF kiadványban, 1999. március,
draft-ietf-ospf-omp-02.
Minneapolis, MN, USA: IETF, 1999. március
[11] C. Hopps,
„Analysis of an equal-cost multi-path algorithm,"
IETF, RFC2992, 2000. november
[12] A. Takács, A. Császár, R. Szabó, T. Cinkler,
"Thrifty traffic engineering through CSLLS,"
18th International Teletraffic Congress ITC18,
ser. LNCS, Teletraffic Science and Engineering,
Vol. 5., Germany: Elsevier, pp.61–70, 2003. szept.
[13] Nick Feamster, Jay Borkenhagen, Jennifer Rexford,
„Guidelines for interdomain traffic engineering,"
ACM SigComm Computer Communication Review,
2003. ősz
[14] Tracie E. Monk,
"Inter-domain Traffic Engineering:
Principles and Case Examples," INET 2002. június
[15] Bruno Quoitin, Louis Swinnen,
Oliver Bonaventure, Steve Uhlig,
„Interdomain traffic engineering with BGP,"
IEEE Communications Magazine,
pp.122–128, 2003. május
[16] Y. Rekhter, T. Li, és mások,
„A border gateway protocol 4 (BGP-4)",
IETF, RFC1654, 1994. július
[17] Marco Gaertler, Maurizio Patrignani,
„Dynamic Analysis of the Autonomous System Graph",
Proceedings of the 2nd International Workshop
on Inter-Domain Performance and Simulation
(IPS2004), Budapest, 2004. március

Hírek

A **Sun Misrosystems,** mint a nyílt forráskódot támogató közösség tagja, a Linux World konferencia és Expo-n elhatározta, hogy az AMD Opteron(tm) processzorokra épülő rendszerek továbbfejlesztett családjával biztosítja a 64 bites vállalati rendszerek, a nagytömegű szerverek és munkaállomások alkalmazását. Így a Sun Java Desktop Systems egyértelmű előnyei világszerte tovább növelik a LINUX népszerűségét. A cég a konferencia keretében bepillantást engedett a Solaris 10 OS egyik új, Project Janus kódnevű technológiájába, amely lehetővé teszi a bináris Linux-alkalmazások változatlan formában történő futtatását Solaris OS alatt.

Az **Oracle** Application Server 10g egy olyan integrált, szabványokon alapuló szoftverplatform, amely az adott vállalat méretétől függetlenül lehetővé teszi a változó üzleti követelményekhez történő hatékony alkalmazkodást. Az Oracle Application Server 10g egy csomagba integrálja a J2EE és a számítóhálózatok támogatását, a beépített nagyvállalati portálszoftvert, a gyorsítóárazást, az operatív üzleti adatelemzést, és üzleti folyamatok integrációját, a vezeték nélküli szolgáltatásokat, valamint a webszolgáltatásokat. Az Oracle Application Server Portal 10g nemrégiben elnyerte a Network Computing rangos szerkesztői díját (Editor's Choice). A „Well-Connected" díjat, amelyet a vállalati környezetben megfigyelt teljesítmény és a különböző teszteredmények alapján ítélnek oda, és az adott év kiemelkedő technológiai termékei és szolgáltatásai kapják. A nyerteseket a Network Computing magazin „Real-World" laboratóriumában tesztelt és értékelt termékek közül szakértő szerkesztők választják ki. A Network Computing az egyes portáltermékeket általános felépítésük, megvalósításuk, biztonsági jellemzőik, áruk és bővíthetőségük alapján értékelte ki, és figyelembe vette, hogy a termékek szabvány alapú alkalmazásokkal és szolgáltatásokkal integrálhatók-e.

A **rEVOLUTION** augusztusban adta el tízezredik Iroda Sorozat szoftverét. Az uniós csatlakozás óta érezhetően megnőtt a kisvállalkozói szoftverek iránti kereslet. Ez egyrészt az Európai Uniós csatlakozást követő új törvényi előírásokkal indokolható – amelyek többek között az egyéni vállalkozókat, illetve a devizaszámlát kiállító cégeket érintik leginkább –, másrészt az elvárt, uniókonform cégmegjelenéssel, valamint a piaci verseny erősödése miatt szükséges, naprakész nyilvántartások vezetésével.
Az európai uniós csatlakozás ugyanis a számlakiállítás módjában is számos változást ír elő. A csatlakozást követően kötelezővé vált számos új adat feltüntetése (például közösségi adószám feltüntetése közösségi termékértékesítés és szolgáltatásnyújtás esetén), átalakult az előlegszámla, más adatok szerepeltetése pedig (például vállalkozói igazolvány számának feltüntetése) a megváltozott adóbefizetési kötelezettségek miatt a vállalkozások egy részénél ajánlott, hiszen elmulasztása plusz adóelőleg-, illetve áfabefizetést von maga után. A kisvállalkozói számlázó szoftverek piacának egyik szereplője, a rEVOLUTION Software a nyári hónapokban is eladást-növekedést jelzett e szoftverek értékesítése terén.

Skálázható útválasztás mobil környezetben

BICZÓK GERGELY, ÉGI NORBERT, FODOR PÉTER, KOVÁCS BALÁZS, VIDA ROLLAND

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
{biczok, egi, fodorp, kovacsb, vida}@tmit.bme.hu

Reviewed

Kulcsszavak: hierarchikus irányítás, többszolgáltatós kapcsolat, hálózati struktúrák, proaktív tervezés

*A jövő hálózatait az eszközök számának robbanásszerű növekedése, és ezeknek jelentős mobilitása jellemzi majd. Mivel a jelenlegi IP alapú megoldások előreláthatóan nem fognak megfelelni az új hálózati struktúráknak, olyan alternatív címzési és útválasztási megoldások válnak szükségessé, melyek képesek lesznek ezen nagyméretű és mobilitású dinamikus rendszerek hatékony kezelésére. E cikk átfogó képet szeretne nyújtani azokról a már létező algoritmusokról, javaslatokról, melyek hasznos kiindulópontot jelenthetnek egy hasonló követelményrendszerre épülő jövőbeli hálózati architektúra kiépítéséhez.**

1. Bevezetés

A napjainkban elterjedőben lévő elképzelés a jövő hálózatairól, hogy annak minden és mindenki között egy helytől és időtől független távközlési csatornát kell fenntartania. Érzékelhető, hogy egy ilyen jellegű elképzelés megvalósulásához számos különböző eszköz és technológia összehangolt együttműködésére van szükség. A hálózati trendek alapján az egyik legszembetűnőbb változás, mely a jövőbeli hálózati architektúrát jellemzi majd, az a résztvevő fix és mobil eszközök arányának eltolódása.

Ma már léteznek technológiák, melyek képesek a számítógépes hálózatok perifériájában megjelenő mobilitást megfelelően kezelni, azonban az imént említett arányok radikális változásához már nem képesek kielégítő és hatékony mobilitási és címzési támogatást nyújtani.

Másfelől, ha a mobilitás nem csak a perifériára lesz jellemző, az útválasztási mechanizmusok újragondolása válik szükségessé. Gondoljunk csak a vezetékes hálózatokban használt IGP (*Interior Gateway Protocol*) és EGP (*Exterior Gateway Protocol*) protokollokra, melyek a különböző összeköttetés-meghibásodásokra is nagyon lassan reagálnak, hosszú konvergenciaidővel. Egy dinamikus hálózatban, melyben a mobil csomópontok más és más, esetleg szintén mobil csomópontokon keresztül érik el a hálózatot, nehezen elképzelhető eme útválasztó protokollok helyes működése.

A másik alapvető tulajdonság, mely meghatározza majd a jövő hálózatait a kommunikálni kívánó csomópontok számának növekedése. Elsősorban itt nem a hagyományos fix telepítésű számítógépekre gondolunk, hanem mobiltelefonokra, laptopokra, PDA-kra, személyazonosító kártyákra, szenzorokra stb. Ez a változás leginkább a skálázhatóság szempontjából kérdőjelezi meg a jelenlegi távközlési megoldásokat.

Ezen tendenciák alapján tehát olyan új, skálázható útválasztási és címzési mechanizmusok kidolgozása válik szükségessé, melyek képesek lesznek folyamatos és konzisztens kapcsolattartást biztosítani különféle technológiákat használó eszközökből álló hálózat számára. Cikkünkben szeretnénk eme területeket megcélzó, a hagyományos IP-alapú címzési és útválasztási architektúra alternatíváját kínáló algoritmusokról egy tömör összefoglalót adni.

2. Egysíkú (flat) útválasztó eljárások

A különböző ad-hoc útválasztási protokollok több szempontból is érdekes kiindulópontot nyújtanak. Bár általában kisméretű hálózatokra voltak tervezve, a skálázhatóság viszonylagos hiányának ellenére hatékonyak a dinamikus felépítésű, nagy mobilitással rendelkező hálózatok kezelésénél. Egysíkú algoritmusok esetén minden útválasztó ismeri az összes lehetséges célcsomópontot. Ez nagy hálózatoknál természetesen nagy útválasztó táblákat eredményez. Az egysíkú eljárások alapvetően két csoportba sorolhatóak: *proaktív* és *reaktív* (*kérés alapú*) algoritmusok. A proaktív algoritmusok folyamatosan gyűjtik a hálózati topológiára és az útválasztásra vonatkozó információkat, ezáltal többletforgalmat generálnak a hálózaton, míg a kérés alapú eljárások csak akkor keresnek utat, ha a hálózati forgalom azt indokolja.

A legegyszerűbb proaktív eljárás a *Fisheye State Routing* (*FSR*) [1], amely *összeköttetés-állapot* (*Link State, LS*) algoritmust használ. Az *LS* algoritmusokban az útválasztók topológia-információkat terjesztenek, és minden eszköz ismeri a hálózat teljes topológiáját. Az *FSR* útválasztók a közeli szomszédokról pontosabb, a távolabbiakról kevésbé pontos információkat ismernek. Az *Optimized Link State Routing (OLSR)* [2] megoldás

** A szerkesztő megjegyzése:*
A Nemzetközi Távközlési Únió (ITU) is keresi az egységes, minden szolgáltatást elérő címzést, melyet ENUM betűszóval jellemez.

ezzel szemben csökkenti a szórt (broadcast) üzenetek számát, közvetítő pontokat kijelölve a hálózatban. Minden útválasztó ezeknek küldi el LS frissítő üzeneteit, és utána ezek a csomópontok cserélgetik egymás között az információkat. Ez a megoldás csak sűrű hálózatokon hatékony, egyébként hagyományos *LS*-ként működik.

A *Topology Dissemination Based on Reverse Path Forwarding (TBRPF)* [3] egy feszítő fát készít a topológiából, és a frissítő üzenetek elárasztás helyett ezen a fán, az üzenet forrásával ellentétes irányba terjednek. Az üzenetek vonatkozhatnak teljes vagy részleges topológia információkra. A részleges esetben a csomópontok a forrás fa egy részébe csak a változásokat küldik tovább. Ez a megoldás is leginkább sűrű topológia esetén hatékony.

Ezzel szemben a kérés alapú (reaktív) útválasztó eljárások csak akkor keresnek útvonalat, ha van átviendő adat. Az *Ad-Hoc On Demand Distance Vector Routing (AODV)* [4] algoritmusban például a célcsomópont felfedezése a hálózat elárasztásával történik. Amint megvan a cél, a visszairányú, illetve a további forgalom már egyértelmű, mivel a köztes csomópontok megtanulják a továbbküldési irányokat. Az AODV-hez hasonló távolság alapú (*Distance Vector, DV*) eljárások nem ismerik a hálózati topológiát; az útválasztók az egyes csomópontokhoz tartozó távolságaikat és a következő csomópont címét hirdetik.

A kérés alapú megoldás egy másik vállfaja forrás útválasztást alkalmaz [5], melyben a küldő a csomag fejlécébe teszi a köztes csomópontok címeit. Ez a megoldás okozza a legkisebb többletforgalmat és a küldő több útvonalat is használhat. Egy hálózati hiba esetén azonban egy új útvonal választása késleltetést okozhat. A kérés alapú algoritmusok nagyméretű hálózatokban is használhatók, bár nagyobb többletforgalmat termelnek, de igazi hátrányuk, hogy a mobilitást csak késleltetve képesek kezelni a folyamatos útvonalkeresés miatt.

3. Hierarchikus útválasztás

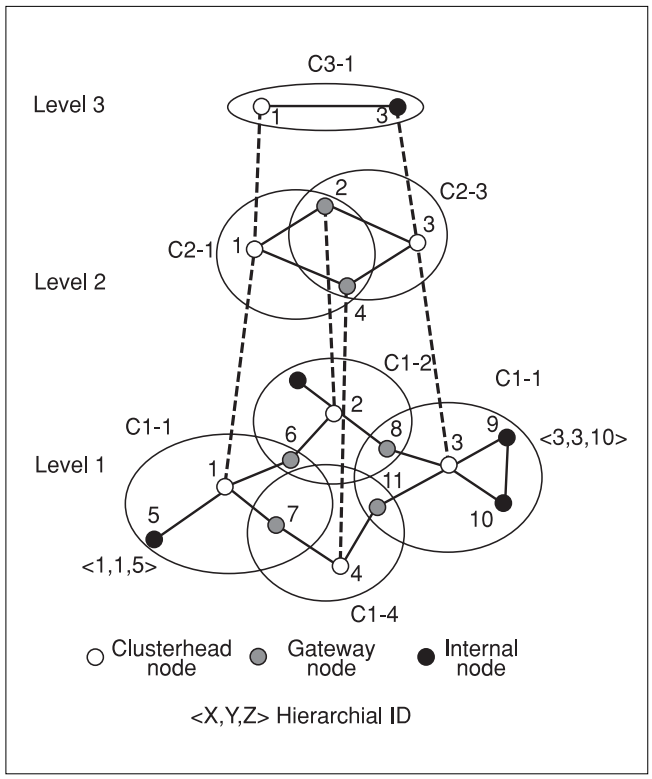
Hierarchikus szervezési struktúrákat már a távbeszélő hálózatok építésénél is alkalmaztak [6], az úgynevezett *multipoláris* rendszerek formájában. A multipoláris rendszerek több csillaghálózatból állnak, melyek csúcspontjai szövevényes hálózatot alkotnak. Mindemellett, a távbeszélő hálózatok forgalomirányítási technikáit, a topológián kívül, a használt harántnyalábok típusa is meghatározza. A multipoláris és csillag alaphálózatot haránt kötőélek egészíthetik ki, melyek lerövidíthetik az összeköttetések hosszát. Egy harántnyaláb lehet túlcsordulásos, azaz a nyaláb csatornáinak foglaltsága esetén a hívásokat megkísérelnjük más nyalábon felépíteni, avagy lehet veszteséges, azaz a nyaláb foglaltsága esetén a forgalom elvész.

A fentiek alapján a szövevényes hálózatokban tulajdonképpen nincs szükség forgalomirányításra (hisz minden csúcspont között közvetlen kötőél létezik), míg a veszteséges harántnyalábok esetén direkt-utas rendszert, a túlcsordulásos harántnyaláb esetén pedig alternatív (több utas) forgalomirányítási rendszert lehet alkalmazni. Az alternatív forgalomirányítási technikák a szükséges csatornaszám kiszámításához figyelembe veszik a hálózati forgalom átlag és szórásértékeit, a felkínált forgalom méretét, az érkezési intenzitást, valamint a szabad alternatív utak által felajánlott kapacitást is [7].

A telefonhálózatok építésében használt hierarchikus szervezési megoldásokat fellelhetjük napjaink csomagkapcsolt útválasztási struktúráiban is. Egy többszáz csomópontból álló hálózat esetén az egysíkú megoldások már nem jelentenek skálázható alternatívát. Célszerű tehát egy ilyen hálózatot hierarchikusan felépíteni, a csomópontokat csoportokba sorolni, és a csomópontokhoz különböző funkciókat rendelni. Így elég, ha a csomópontoknak csak a hálózat egy részéről van információjuk. Minden csoportnak van egy vezetője, mely a többi csoporttal tartja a kapcsolatot átjárókon (több csoporthoz csatlakozó csomópontok) keresztül.

A legegyszerűbb ilyen eljárás, a *Clusterhead-Gateway Switch Routing (CGSR)* [8], mely *DV* algoritmust használ. Minden csomópont tárol egy útválasztási táblát (csoportonként egy bejegyzés) és egy csoport táblát (ki melyik csoporthoz tartozik). Ha valaki üzenetet akar küldeni, a csoport táblában megnézi, hogy melyik csoporthoz tartozik a célcsomópont, majd az útválasztási tábla alapján továbbküldi a csomagot. *Hierarchical State Routing (HSR)* [9] esetén a csoportvezetőket is csoportokba szervezik, többszintű hierarchiát kialakítva *(1. ábra)*.

1. ábra Hierarchical State Routing



A csoportvezetőt minden hierarchia szinten több lehetséges szempont szerint (pl. QoS paraméterek) jelölik ki. Minden csomópontnak van egy címe, ez alapján történik az útválasztás. A cím a csomópont egyedi azonosítójából, valamint az egyes hierarchia szinten levő csoportvezetők azonosítóiból képezik.

Csoporton belül és azon kívül lehet különböző útválasztási algoritmusokat is használni, mint teszi azt a *Zone Routing Protocol* (*ZRP*) [10]. Ez a hibrid eljárás zónán (csoporton) belül proaktív (*Intrazone Routing*), míg zónán kívül kérés alapú (*Interzone Routing*) útválasztást használ.

4. Földrajzi információkon alapuló útválasztási eljárások

Egy csomópont fizikai pozíciójának ismerete megkönnyíti a csomag célcsomópont felé való irányítását. Ehhez természetesen szükséges, hogy az egyes csomópontok képesek legyenek helyzetük meghatározására. *Geographical Addressing and Routing* (*GeoCast*) [11] esetén a forrás először egy helyi központi csomóponthoz (GeoNode) továbbítja a csomagot. Ha a cél nincs annak környezetében, akkor a csomag egy magasabb hierarchia szintű csomóponthoz (GeoRouter) kerül, mely a többi GeoNode környezetében próbálja megkeresni a célcsomópontot. A GeoRouterek hierarchikus szervezése által az útválasztási táblák mérete alaposan lecsökken.

A csomópont utolsó ismert koordinátái alapján egy részleges elárasztást alkalmaz a kérés alapú *Location-Aided Routing* (*LAR*) [12] eljárás. A cél pozíciója és mozgása, valamint forrás helyzete meghatároz egy területet, amelyen belül a csomópontok elárasztással továbbítják a csomagot *(2. ábra)*. Egy másik megoldás az, hogy egy csomópont akkor árasztja tovább a csomagot, ha önmaga közelebb van a célhoz, mint ahonnan azt kapta.

Egy proaktív megoldás a *Distance Routing Effect Algorithm for Mobility (DREAM)* [13], melyben a csomópontok a linkek helyzetét tartják nyilván. A többletforgalom azáltal csökken, hogy a periodikus frissítési üzenetekben a csomópont a helyzetét két szempont szerint terjeszti: minél távolabbi egy csomópont, annál ritkább a frissítés, valamint a gyorsan mozgó csomópontok gyakrabban küldenek frissítéseket. A csomagok részleges árasztása miatt a többszörös kézbesítés redundanciát okoz, így a rendszer a mobilitásra kevésbé érzékeny.

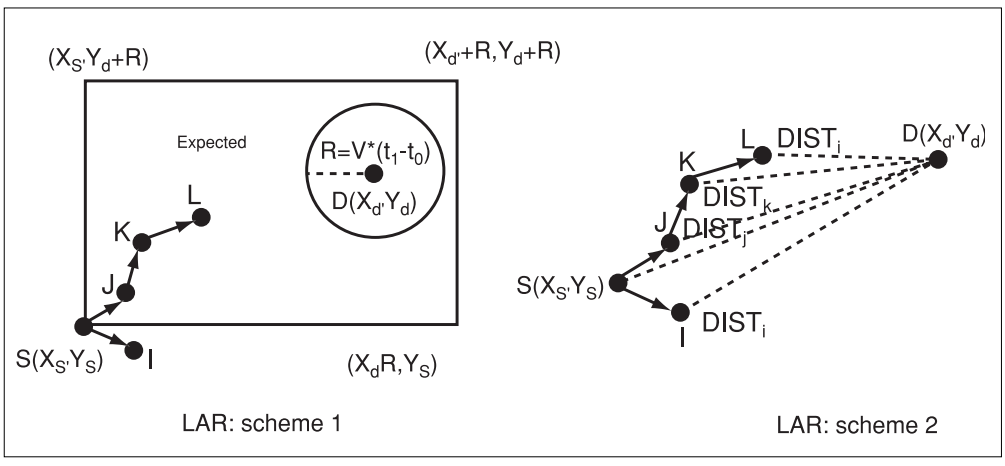
Míg a *DREAM* és *LAR* algoritmusok helymeghatározásra elárasztást használnak, addig a *Greedy Perimeter Stateless Routing* (*GPSR*) [14] eljárás forrás útválasztást alkalmaz. A csomópont szomszédainak pozíciói periodikusan terjesztődnek minden csomópontban, így a továbbítás mindig a célhoz földrajzilag legközelebbi csomópont felé történik, melyet a szomszédossági gráfból számolunk ki. Mivel a szomszédossági útválasztási táblák kicsik, ezért skálázhatósági problémák itt nincsenek.

A *Terminode* [15] eljárás proaktív *DV*-t használ helyi útválasztásra. Az útválasztó táblákban a helyi csomópontok azonosítói és pozíciójuk van tárolva. Egy távoli cél felé a hálózatban található fix pontok mentén továbbítja a csomagokat. Amennyiben fix ponton alapuló utat nem talál a forrás, akkor a csomag a célhoz földrajzilag legközelebbi csomópont felé továbbítódik. Ezen fix pontok, útvonalak mentén történő továbbítás egy jól skálázható megoldást biztosít, mivel a forrás és a cél közti hálózat nagyságától szinte független.

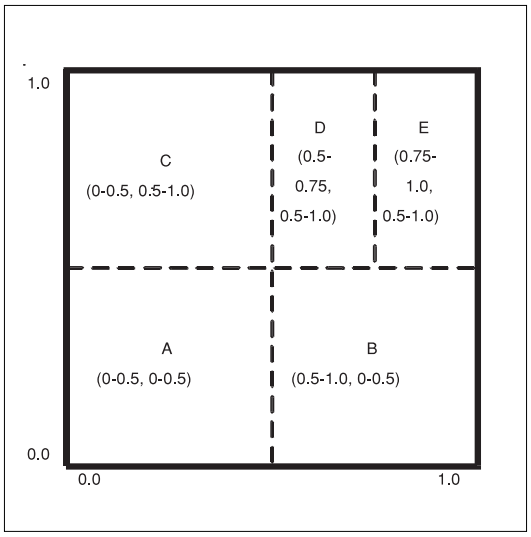
5. Elosztott hash tábla alapú megoldások

Az *elosztott hash tábla* (*Distributed Hash Table, DHT*) alapú módszerek leginkább az új generációs peer-to-peer rendszerekben terjedtek el, de alkalmazhatóságuk nem korlátozott. A DHT megoldások hasznos kiindulópontnak bizonyultak jónéhány nagyméretű rendszer kifejlesztése során. Számos kutatási eredmény ajánlja használatukat, mint egy lehetséges réteget, amelyre akár több millió csomópontból álló hálózati megoldásokat építhetünk, legyen az egy elosztott fájlrendszer, alkalmazás szintű multicast megoldás, eseményközlő vagy chat szolgáltatás stb.

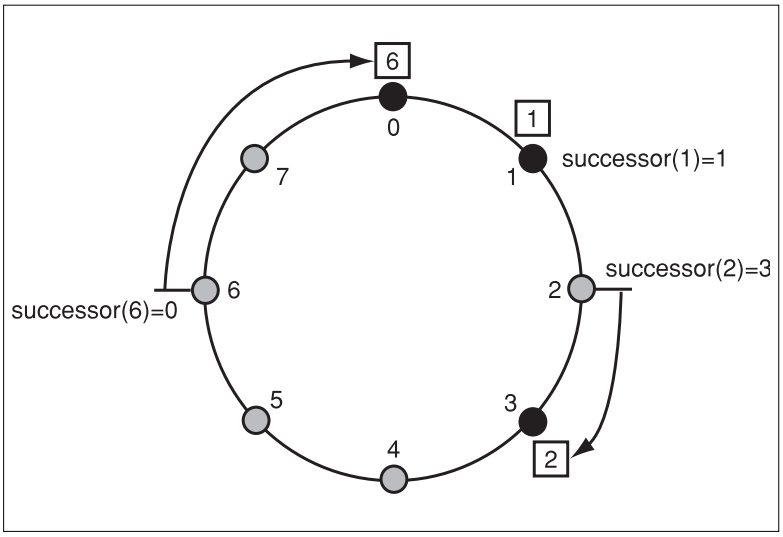
2. ábra Location-Aided Routing



Az alábbiakban bemutatunk néhány olyan ismertebb címzési és útválasztó megoldást, mely elosztott hash tábla használatára épül. Ha egy elosztott fájlrendszer esetét elemezzük, a cél az, hogy megtaláljuk az utat ahhoz a résztvevőhöz, amely a keresett fájlt tárolja. A fájlokhoz kulcsok vannak rendelve, melyeket például a fájl nevén végrehajtott hash függ-



3. ábra Content Addressable Network



4. ábra Keresés a Chord rendszerben

vénnyel állítunk elő. A hálózat minden csomópontja felelős a kulcsok egy bizonyos tartományának tárolásáért. Ezen rendszerek alapvető művelete a kulcs leképzés (*lookup(key)*), amely megadja azon csomópont azonosítóját (például IP címét), amely tárolja a kulcs által meghatározott objektumot. A csomópontok egy hálózati fedőréteget (*overlay*) alkotnak, melyben a szomszédossági viszonyok különböznek a fizikai rétegbeli szomszédosságtól.

A DHT alapú rendszerek magja a címzési struktúrájuk és az ehhez tartozó útválasztó algoritmus. A *tartalom szerint címezhető hálózat* (*Content Addressable Network, CAN*) [16] egy *d* dimenziós Descartes-féle koordináta rendszert használ a címtér ábrázolására, mely koordináta rendszer minden időpillanatban dinamikusan felosztott a hálózat összes csomópontja közt. Minden csomóponthoz egy-egy tartomány tartozik *(3. ábra)*.

Ebben a virtuális koordináta rendszerben van tárolva minden (*K, V*) kulcs-érték páros, az alábbi módon: a *K* kulcsot leképezve egy egyenletes hash függvénnyel, megkapunk egy *P* pontot a koordináta rendszerben. A kulcs-érték párost az a csomópont fogja tárolni, mely jelenleg felelős a *P* pontot magába foglaló tartományért.

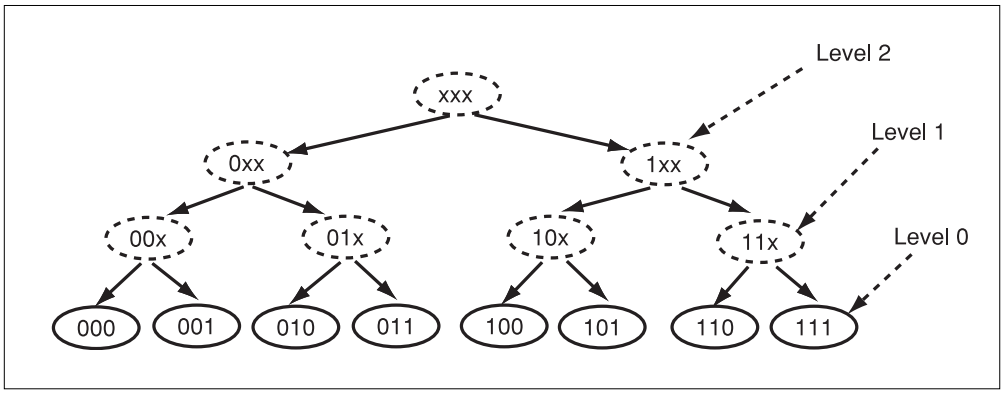
Egy CAN csomópont útválasztó táblája a virtuális koordináta rendszerben szomszédos csomópontok IP címét és a koordináta rendszerbeli címét tartalmazza. Útválasztás közben mindig a cél koordinátához legközelebb fekvő tartomány csomópontjához lépünk tovább.

A *Chord* [17] rendszer az érték-azonosító leképzés során szintén egy hash függvényt használ, amelynek kimenete egy *m* bit hosszúságú bitsorozat. A hash függvény kimenete nagy valószínűséggel egyenletes eloszlású, és a használt címtér egy modulo 2^m egyenletes méretű részre osztott kör. Egy *K* kulcs a körön azon csomóponthoz van rendelve, melynek azonosítója azonos, vagy legelsőként követi a *K* kulcs azonosítóját. Ezt a csomópontot a *K* kulcs *successor* csomópontjának nevezik *(4. ábra)*. A címtér minden csomópontjában mindössze a successor címét kell tárolni, így egy lekérdezés során a successor-okon körbe haladva, a kört bejárva megtalálhatjuk a keresett értéket.

A *Pastry* [18] rendszerben minden csomóponthoz egy 128 bites azonosító van rendelve, mely megadja egy szintén kör alakú címtérben a csomópont pozícióját (0 és 2^{128}-1 között). Itt is feltételezhető, hogy a generált azonosítók egyenletes eloszlásúak. Az útválasztáshoz egy csomópont nyilvántartja a címtérben szomszédos csomópontok címét, a valós hálózati szomszédok címtérbeli címét és egy útválasztó táblát. Az útválasztás ezen adatok felhasználásával történik, az előbbi megoldásoknál dinamikusabban és költséghatékonyabban.

A *PeerNet* [19] hálózati megoldásban a címtér egy bináris fa, ahol minden csomópont címe egy-egy levél a fában *(5. ábra)*. A csomagtovábbítás során a küldő csomópontnak mindössze a célcsomópont azonosítóját kell ismernie, mely egy cím leképzéssel tudható meg. A csomópontok útválasztó táblája azon csomó-

5. ábra PeerNet hierarchia



pontok címét tartalmazza, amelyek az egyes helyiértékek által azonosított – az érintett csomópont szempontjából a fa ellenkező oldalán lévő – részfák csúcsai. Mivel a címleképzés és a címkiosztás mechanizmusa ebben az esetben is összefügg, garantált, hogy bármely a hálózatban jelenlévő csomópont elérhető bármely más csomópontból.

Végül megemlítjük röviden a *Tribe* [20] protokollt, mely egy vezetéknélküli önszerveződő hálózatokban használható közvetett útválasztó stratégiának felel meg. A Tribe egy egyenrangú elemekből álló elosztott rendszer, ahol a csomópontok egy globálisan egyedi, valamint egy elhelyezkedéstől függő ideiglenes azonosítóval rendelkeznek. A csomópontok által alkotott infrastruktúra leírja az adott csomópont környezete szerinti relatív elhelyezkedését. Az útválasztás egyedi módon történik, a home-agent koncepciót követve, amely teljesen független bármely hálózati szintű útválasztó protokoll által biztosított globális összeköttetéstől. A csomópontok pozíciójából meghatározható azok relatív elhelyezkedése a hálózatban, így nincs szükség semmilyen fix csomópontokkal rendelkező infrastruktúrára, földrajzi helymeghatározásra, vagy távolságmérésre.

6. Valószínűségen alapuló módszerek

Nagyméretű mobil hálózatoknál fontos, hogy a rendszer hamar észlelje az esetleges topológia változásokat, és képes legyen dinamikusan adaptálni útválasztását ezekhez. Léteznek különböző, valószínűségen alapuló útválasztó megoldások, melyek lehetővé teszik mindezt. A *Hangya* (Ant) útválasztó algoritmus [21] alapötletét a természettől kölcsönzi: azt utánozza, ahogy a hangyakolóniák megtanulják az élelmiszer-lelőhelyekhez vezető legrövidebb utakat.

Kétféle változat létezik. Az egyik a *szabályos hangya algoritmus* (*Regular Ant Algorithm, RAA*), amely egy legrövidebb utat keres, és csak szimmetrikus linkköltségű hálózatokban használható. A másik az *egyenletes hangya algoritmus* (*Uniform Ant Algorithm, UAA*), mely több alternatív utat kezel és aszimmetrikus linkköltségek esetén is működik.

Minden csomópont (*hd*) periodikusan generál egy rövid üzenetet (hangyát), melyet elküld egy véletlenszerűen választott másik csomópontnak (*hs*). A (*hd, hs, c*) hangyában *c* a két csomópont közti út költsége (kezdetben $c = 0$). A *hs* felé vezető úton található összes útválasztó növeli a *c* mező értékét annak a linknek a költségével, amelyen az üzenetet kapta. A hangyák kis méretűek (<10 bájt), és ellenirányban (a céltól a forrásig) térképezik fel a hálózatot. Minden *r* útválasztó minden *x* célcímhez egy $(x, (p_1, y_1),...,(p_n, y_n))$ bejegyzést tárol, ahol y_i az *r* útválasztó egy szomszédja, és p_i annak a valószínűsége, hogy a célcím felé vezető út következő állomása y_i. Az útválasztó táblák tehát valószínűségi alapon működnek, és frissítésük a megerősítéses tanulás módszerével történik.

RAA esetén a p_i értékek különbözőek, de összegük 1. Az UAA megoldásnál minden $p_i = 1/n$, és a hangyák nem tartalmazzák a célcím (*hs*) mezőt, így szükség van egy TTL (time-to-live) mezőre, mely garantálja az üzenetek véges érvényességét. RAA esetén egy idő után beáll egy állandósult állapot, amikor a továbbítási valószínűségek 0-hoz vagy 1-hez konvergálnak (a követendő linkek kapnak 1-hez tartó valószínűséget). Az UAA eljárás viszont alkalmazkodni tud a hálózatban bekövetkezett változásokhoz, így a két módszert vegyítve egy flexibilis és ugyanakkor hatékony útválasztási algoritmust kapunk.

Valószínűségen alapuló útválasztást akkor is lehet alkalmazni, ha nem csak egy adott célállomás, hanem a hálózat egy tartományában lévő összes állomás felé szeretnénk továbbítani egy adott csomagot. A hagyományos elárasztás alapú megoldások hátránya a generált forgalom mérete, hiszen a legtöbb eljárásban egy állomás többször is megkapja ugyanazt a csomagot más és más szomszédjától. Ezzel szemben a *pletyka alapú útválasztás* (*Gossip Based Routing*) [22] valószínűségen alapuló elárasztást használ, ezzel csökkentve a routing forgalmat.

Az alapötlet egyszerű: egy csomópont bizonyos valószínűséggel szór tovább egy kapott üzenetet. Négy pletyka algoritmus ismeretes, ezek egymás továbbfejlesztett változatai. A $GOSSIP_1(p)$ az alapalgoritmus, mely *p* valószínűséggel elárasztást alkalmaz, (1-*p*)-vel pedig eldobja a csomagot. Ezzel szemben a $GOSSIP_1(p,k)$ a forrástól számított *k* linken keresztül $GOSSIP_1(1)$-ként viselkedik, megelőzve az üzenet korai kihalását. Kihalás akkor történhet, ha egy csomópontnak nagyon kevés szomszédja van. Ezért hasznos a $GOSSIP_2(p_1,k,p_2,n)$ változat: ha egy csomópont fokszáma kisebb mint *n*, akkor a p_2 ($>p_1$) valószínűség lép érvénybe. A $GOSSIP_3$(p,k,m) szerint, ha egy csomópont eredetileg nem szórt tovább egy kapott üzenetet, de adott időn belül nem kapta meg ugyanazt az üzenetet legalább *m* másik csomóponttól, akkor azonnal tovább szórja azt. A negyedik megoldás zónák bevezetésével ér el további javulást.

7. Összefoglalás

Ez a cikk rövid áttekintést nyújt a ma létező, alternatív címzési és útválasztási javaslatokról. Az itt bemutatott eljárások önmagukban nem képesek megoldani napjaink útválasztási kérdéseit, azonban olyan ötleteket tartalmaznak, melyek felhasználhatók egy új architektúra kidolgozásához. A hierarchikus megoldások a csomópontok struktúrába rendezésével, a földrajzi adatokra épülő protokollok a számítógépek tényleges tartózkodási helyének figyelembevételével, az elosztott hash táblás algoritmusok új címzési és továbbítási ötleteikkel, míg a valószínűségi alapokon nyugvó mechanizmusok hasznos redundancia-képző tulajdonságaikkal járulhatnak hozzá egy valóban skálázható és hatékony útválasztási megoldás megszületéséhez.

Irodalom

[1] T.-W. Chen, „Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks”, Proc. of ICC 2000, New Orleans, LA, June 2000.

[2] T. Clausen, P. Jacquet, „Optimized Link State Routing Protocol”, RFC 3626, Oct. 2003.

[3] R. G. Ogier, F. L. Templin, and M. G. Lewis, „Topology Dissemination based on Reverse-Path Forwarding (TBRPF)”, RFC 3684, Feb. 2004.

[4] C. E. Perkins and E. M. Royer, „Ad-Hoc On-Demand Distance Vector Routing”, Proc. of IEEE WMCSA ‘99, New Orleans, LA, Feb. 1999, pp.90–100.

[5] D. B. Johnson and D. A. Maltz, „Dynamic Source Routing in Ad Hoc Wireless Networks”, Mobile Computing, T. Imielinski and H. Korth, Eds., Ch. 5, Kluwer, 1996, pp.153–181.

[6] Dely Z., Ecsedi Gné., Huszty G., Madarász E., Oprics Gy., Plank Gy., Sallai Gy., „Távközlő hálózatok forgalmi tervezése”, PKI, Budapest, 1980.

[7] Wilkinson, R.I., „Theories for Toll Traffic Engineering in the USA”, Bell System Technical Journal, Vol. 35, no.2, pp.421–514., March 1956.

[8] C.-C. Chiang and M. Gerla, „Routing and Multicast in Multihop, Mobile Wireless Networks”, Proc. of IEEE ICUPC ‘97, San Diego, CA, Oct. 1997.

[9] G. Pei, M. Gerla, X. Hong and C.-C. Chiang, „A Wireless Hierarchical Routing Protocol with Group Mobility”, Proc. of IEEE WCNC ‘99, New Orleans, LA, Sept. 1999.

[10] Z. J. Haas and M. R. Pearlman, „The Performance of Query Control Schemes for the Zone Routing Protocol”, ACM/IEEE Trans. Net., Vol. 9, no.4, Aug. 2001, pp.427.

[11] J. C. Navas and T. Imielinski, „Geographic Addressing and Routing”, Proc. of 3rd ACM/IEEE Intn’l. Conf. Mobile Comp. Net., Budapest, Sept. 26-30, 1997.

[12] Y.-B. Ko and N. H. Vaidya, „Location-aided Routing (LAR) in Mobile Ad Hoc Networks”, ACM/IEEE Int’l. Conf. Mobile Comp. Net., 1998, pp.66–75.

[13] S. Basagni, I. Chlamtac, V.R. Syrotiuk and B.A. Woodward, „A Distance Routing Effect Algorithm for Mobility (DREAM)”, Proc of. ACM/IEEE Int’l. Conf. Mobile Comp. Net., 1998, pp.76–84.

[14] B. Karp and H. T. Kung, „GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”, Proc. of Mobicom 2000, Boston, MA, USA, 2000, pp.243–254.

[15] L. Blazevic, S. Giordano and J.-Y. Le Boudec, „Self Organized Terminode Routing”, Technical Report, DSC/2001/024, Swiss Federal Insitute of Techology, Lausanne.

[16] S. Ratnasamy et al., „A Scalable Content-Addressable Network”, Proc. of ACM SIGCOMM, San Diego, CA, Aug. 2001, pp.161–172.

[17] I. Stoica et al., „Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”, Proc. of ACM SIGCOMM, San Diego, CA, Aug. 2001.

[18] P. Druschel, A. Rowstorn, „Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems”, Proc. of Middleware 2001, 18th IFIP/ACM International Conference on Distributed Systems Platforms, Nov. 2001.

[19] J. Eriksson, M. Faloutsos, S. Krishnamurthy, „PeerNet: Pushing Peer-to-Peer Down the Stack”, Proc of IPTPS’03, 2nd International Workshop on Peer-to-Peer Systems, Berkeley, CA, USA, Feb. 2003.

[20] A. Viana, M. Dias de Amorim, S. Fdida, and J.F. de Rezende, „Indirect Routing Using Distributed Location Information”, in Proc of PerCom’03, IEEE International Conference on Pervasive Computing and Communications, Fort Worth, TX, USA, March 2003.

[21] D. Subramanian, P. Druschel, J. Chen, „Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks”, Proc. of IJCAI-97, Nagoya, Japan, Aug. 1997.

[22] Z.J. Haas, J.Y. Halpern, and L. Li, „Gossip-based Ad Hoc Routing”, Proc. of INFOCOM’02, New York, USA, June 2002, pp.1707-1716.

[23] Gódor Balázs, „Technológia-független, univerzális azonosítók”, PKI Közlemények, 2004/48. szám, pp.127–140.

Távközlési hálózatok tervezése a forgalomeloszlás változásainak figyelembevételével

TAMÁSI LEVENTE, JÓZSA BALÁZS GÁBOR, ORINCSAY DÁNIEL

{Levente.Tamasi, Balazs.Jozsa, Daniel.Orincsay}@eth.ericsson.se

Kulcsszavak: időfüggő forgalom, forgalom-átterhelés, kihasználtságnövelés, útvonalválasztás

Jelen tanulmány egy új algoritmust javasol távközlési hálózatok költséghatékony tervezésére. Az alkalmazott hálózati modell kétféle hálózati elemet használ: útvonalválasztókat és átviteli utakat. A gyakorlatban ezek kapacitásai diszkrét értékek, ezért az egyes hálózati eszközökhöz lépcsős költségfüggvények rendelhetők. A forgalom periodikus (napi, heti) változásai is figyelembe vehetők a hálózattervezés során, ami olcsóbb költségű hálózatot eredményezhet.

Írásunkban egy új algoritmust javaslunk, amely egyszerre oldja meg a hálózattervezés és az útvonalválasztás problémáját. A bemutatott módszer kiindulásként egy hatékony, globális útvonal-optimalizáláson alapuló heurisztikus algoritmust alkalmaz, azonban kiterjeszti azt több forgalmi időszak kezelésére. Az új algoritmus teljesítményét szimulációval vizsgáljuk két referenciaalgoritmus segítségével, valós és véletlenszerűen generált problémapéldányokon.

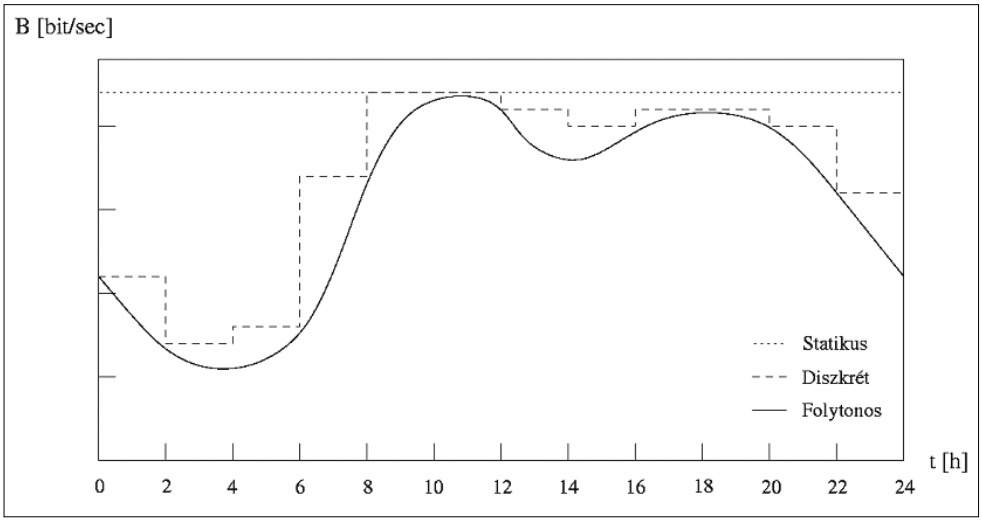
1. Bevezetés

Napjainkban a sávszélesség iránti igény rohamosan növekszik, mivel egyrészről új, nagy sávszélesség-igényű alkalmazások jelennek meg, másrészről a felhasználók száma is gyorsan emelkedik. Emellett megfigyelhető az újabb és újabb technológiák elterjedése is. A fentiek miatt gyakran felmerülő probléma a kommunikációs hálózatok tervezése és méretezése. Mivel a gerinchálózatok kiépítése jelentős költségű beruházás, indokolt egy olyan algoritmus megtervezése, amely ezt a feladatot úgy oldja meg, hogy a keletkező hálózat költsége minél kisebb legyen. Ennek a célnak az eléréséhez fontos a hálózat jó kihasználtságának biztosítása, amihez célszerű figyelembe venni a forgalomeloszlás napi változásait. A jó kihasználtság a szolgáltatók számára fontos szempont, mivel így nagyobb profitra tehetnek szert. A napi forgalomeloszlás-változásokat figyelembe vevő megközelítés előnye, hogy a hálózattervezés során kihasználható, hogy az egyes útvonalválasztó-párok közötti maximális kapacitásigény fellépése különböző időszakokra eshet, így alacsonyabb eszközkapacitások is elegendőek lehetnek a hálózatban a legforgalmasabb időszakra tervezéshez képest.

Az *1. ábrán* látható a folytonos forgalmi függvény statikus és diszkrét modellezése. A statikus modellezés során az útvonalválasztó-párok között a nap folyamán fellépő maximális kapacitásigényre tervezünk hálózatot, míg diszkrét modellezés esetén a napot több – akár különböző hosszúságú – időszakra osztjuk, és az egyes időszakokra külön-külön adjuk meg a forgalmi igényeket.

Jelen tanulmányban ez utóbbi megközelítést alkalmazzuk a napi forgalomeloszlás-változások modellezésére. Ehhez szükséges, hogy a forgalmi igényekhez rendelt útvonalrendszer a nap folyamán dinamikusan átkonfigurálható legyen. Ezt a lehetőséget több technológia, így például az egyre jobban terjedő *MPLS (Multi-Protocol Label Switching,* többprotokollos címkekapcsolás) biztosítja. A napi forgalomváltozások által indokolt újrakonfigurálások témájával az ITU–T E.360.6 [1] ajánlása is foglalkozik, és a *kapacitásmenedzsment* fogalma alá sorolja.

1. ábra A folytonos forgalmi függvény statikus és diszkrét modellezése



Az ajánlás több módszert is javasol a periodikus forgalomeloszlás-változásokat figyelembe vevő hálózattervezésre. A *DEFO (Discrete Event Flow Optimization)* modellek a forgalmi igényeket diszkrét hívási eseményekké alakítják, a *TLFO (Traffic Load Flow Optimization)* modellek az igények leírásához a lineáris programozás eszköztárát veszik igénybe, míg a *VTFO (Virtual Trunking Flow Optimization)* modellek az igényeket előre definiált kapacitásegységekké konvertálják. Mindhárom módszer iteratív módon oldja meg a problémát: egy inicializálási fázist követően felváltva ismétlődik az útvonaltervezés és a kapacitástervezés.

Áramkörkapcsolt hálózatokra Medhi [2] és Ouveysi [3] ad a napi forgalomeloszlás-változásokat kihasználó hálózattervezési módszert. Medhi [2]-ben bemutatott módszere az egyszeres meghibásodások ellen is védelmet nyújt. Kétszintű modellt alkalmaz, melynek lényege, hogy együtt kezeli a rendezőket *(cross connect)* és fizikai összeköttetéseket tartalmazó fizikai hálózat, valamint a kapcsolókból és logikai összeköttetésekből álló logikai hálózatot. A tervezés célja adott szolgáltatásminőség biztosítása a logikai hálózatban hibamentes működés, valamint egy fizikai összeköttetés meghibásodása esetén. A probléma megoldásához heurisztikus algoritmusokat javasol, melyek a probléma hibaállapotok szerinti dekompozícióján alapulnak.

Ouveysi [3]-ban javasolt algoritmusa teljesen összekötött *(full mesh)* áramkörkapcsolt hálózatokra hasonló ötleten alapul. Modelljében minden kapcsolópár közötti forgalom vagy a közvetlen összeköttetésen, vagy egy harmadik kapcsoló közbeiktatásával zajlik. Megadja a probléma egy lineáris programozási megfogalmazását, és egy forgalmi időszakok szerinti dekompozíción alapuló heurisztikus algoritmust javasol.

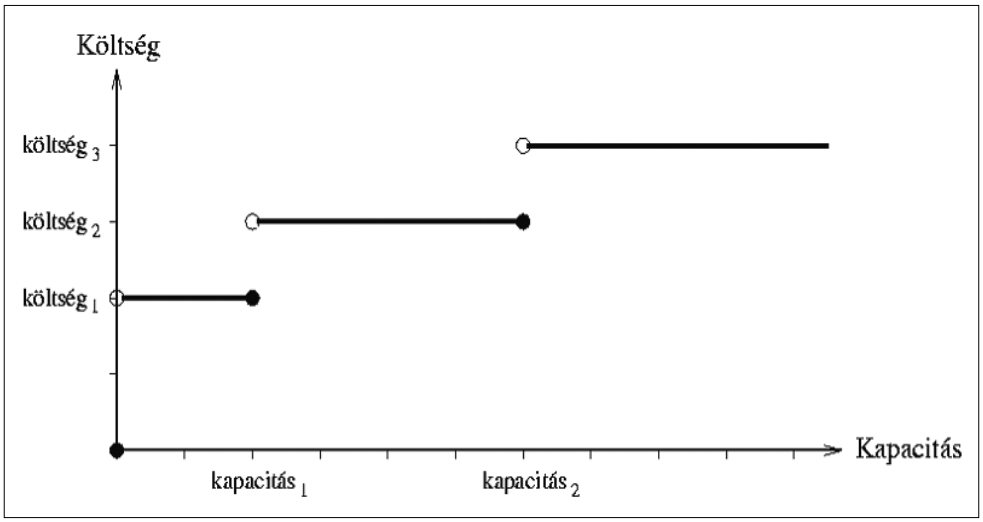
ATM (Asynchronous Transfer Mode) hálózatokra Bauschert [4], valamint Medhi [5] [6] [7] ad módszereket. Bauschert [4]-ben javasolt algoritmusának célja adott szolgáltatásminőség garantálása és a virtuális útvonalak *(virtual path, VP)* rendszerének optimális kialakítása azzal a feltételezéssel, hogy egy virtuális áramkör *(virtual circuit, VC)* több virtuális útvonalat is igénybe vehet. Módszere a feladatot három részproblémára – virtuális áramkörök útvonaltervezése, virtuális útvonalak méretezése és virtuális útvonalak tervezése – bontja, melyeket iteratívan old meg, két részfeladatot mindig állandónak tekintve. A forgalomeloszlás periodikus változásait a virtuális áramkörök útvonaltervezése során veszi figyelembe.

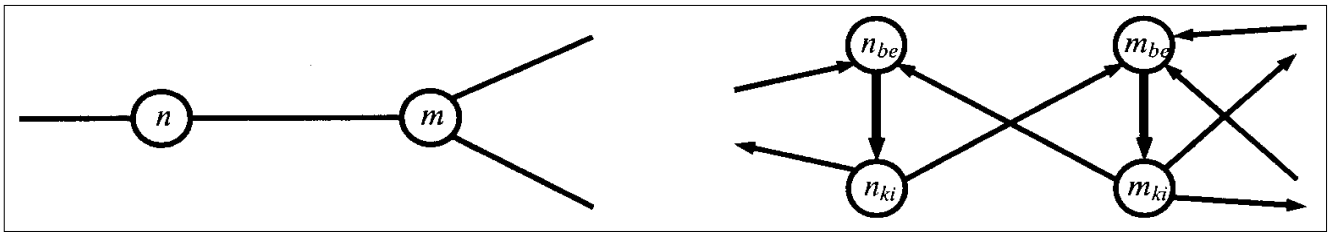
Medhi [5,6] cikkeiben bemutatott módszerei ATM hálózatokra eltérő modellt használnak: minden útvonalválasztó-pár esetén minden forgalmi osztályhoz külön-külön virtuális útvonalat rendelnek. Medhi megmutatja, hogy az egyes virtuális útvonalakon belül statisztikus, a virtuális útvonalak között pedig determinisztikus nyalábolást használva a probléma két részfeladatra bomlik: egy sávszélesség-becslési és egy kombinált útvonal- és kapacitástervezési problémára. A feladat megoldására több algoritmust is javasol: egy genetikus algoritmust, egy Lagrange-relaxáción alapuló módszert és az előző kettő ötvözetét.

Medhi és Lu [7]-ben javasolt módszere a [2]-ben bemutatotthoz hasonló modellt alkalmaz ATM hálózatokra azzal a különbséggel, hogy mind a virtuális útvonalak, mind a virtuális összeköttetések esetén dinamikusan változó útvonalakat tételeznek fel. Módszerük a probléma kevert egész értékű lineáris programozási megfogalmazásán alapul, melyet a két hálózati szint szerinti dekompozícióval oldanak meg.

Látható tehát, hogy a periodikus forgalomeloszlásváltozások figyelembevételével történő hálózattervezés problémájával többen foglalkoztak az irodalomban. Azonban a fent bemutatott módszerek közös jellemzője, hogy költségfüggvény-modelljük nem közelíti megfelelően a valóságot. A gyakorlatban az eszközök moduláris felépítésűek, fix – általában szabványos – kapacitású egységekből állnak, ezért költségfüggvényük lépcsős, és csökkenő határköltséget mutatnak *(2. ábra)*. Emellett a [2,5,6,7] cikkekben javasolt módszerek fontos tulajdonsága, hogy előre számított útvonalak közül választanak az útvonal-optimalizálás során, amely jelentősen korlátozza az algoritmus lehetőségeit. A fentiek indokolják egy hatékony, a forgalomeloszlás periodikus változásait is figyelembe vevő, lépcsős költségfüggvény modellt alkalmazó hálózattervező algoritmus kifejlesztését. A probléma NP-nehéz, vagyis – a tudomány mai állása szerint – nem adható rá olyan algoritmus, amely polinom időben garantálja az optimális megoldás megtalálását. Ezért indokolt a heurisztikus megközelítés alkalmazása. Az általunk javasolt algoritmus a [8]-ban bemutatott heurisztikus algoritmusból indul ki, és azt terjeszti ki több forgalmi időszak kezelésének képességére.

2. ábra Példa lépcsős költségfüggvényre





3. ábra Csúcs helyettesítése virtuális éllel

2. A tervezési probléma

Az alábbiakban bemutatjuk a tervezési probléma formalizálásához alkalmazott hálózati, forgalmi és költségmodellt. A hálózatot egy irányított gráf segítségével reprezentáljuk, a csúcsok az útvonalválasztókat, az élek a fizikai összeköttetéseket jelképezik. A csúcsokon és éleken kapacitáskorlátokat definiálunk az egyes eszközöknek megfelelően. Mivel a gráfalgoritmusok általában élek kapacitáskorlátját tudják könnyen kezelni, a csúcsokat virtuális élekkel helyettesítjük. Ezek szerint egy n csúcs helyét két új csomópontot veszi át: az eredeti csúcsba bejövő forgalom az nbe csomópontba érkezik, a kimenő forgalom pedig az nki csomópontból indul. Az útvonalválasztó kapacitáskorlátját az (n_{be},n_{ki}) virtuális élen vesszük figyelembe *(3. ábra)*.

Az útvonalválasztó-párok közötti forgalmat diszkrét módon modellezzük: a nap rögzített számú időszakra oszlik, és minden egyes időszakra külön-külön adottak a forgalmi igények. Az időszakok hossza eltérő is lehet, ezért minden időszakhoz egy súlyt is rendelünk. Az alkalmazott forgalmi modell az úgynevezett pipe-modell: az egyes forgalmi igényeket a forrás és nyelő útvonalválasztó, valamint az igényelt sávszélesség definiálja. Bár a továbbiakban egyetlen oszthatatlan forgalmi igényt feltételezünk minden útvonalválasztó-pár között, a modell képes egy útvonalválasztó-pár között több igényt is kezelni. Így lehetőség van több forgalmi osztály alkalmazására is; elkülöníthető például a hang- és az adatforgalom.

A különböző időszakokban fellépő forgalmi igények elvezetése során az igényekhez egy-egy kapacitásfoglalt útvonalat rendelünk, ami azt jelenti, hogy az elvezetéshez elegendő kapacitást kell lefoglalni az igényhez rendelt útvonalon elhelyezkedő útvonalválasztókban és fizikai összeköttetéseken. Természetesen egy adott hálózati eszközön történt kapacitásfoglalások összege nem haladhatja meg az adott eszköz teljes kapacitását.

Az egyes hálózati eszközök ár/kapacitás viszonyainak modellezésére jelen tanulmányban lépcsős költségfüggvényeket alkalmazunk, amelyek minden hálózati eszközre (útvonalválasztók, illetve fizikai összeköttetések) egyediek lehetnek. Ez lehetővé teszi speciális költségmódosító faktorok, például már kiépített, ingyen rendelkezésre álló hálózati eszközök figyelembe vételét is.

A fentiek tükrében a tervezési feladat a következő módon definiálható. Az algoritmus bemenetei:

- az útvonalválasztók elhelyezkedése,
- az útvonalválasztók közötti lehetséges fizikai összeköttetések,
- az egyes hálózati eszközök (útvonalválasztók, illetve fizikai összeköttetések) egyedi költségfüggvényei,
- az időszakokhoz rendelt súlyok,
- minden egyes időszakra a forgalmi igények halmaza.

A tervezési feladat egy megoldása, vagyis az algoritmus kimenete az alábbiakat tartalmazza:

- a hálózati topológiát (a lehetséges fizikai összeköttetések közül nem feltétlenül szerepel az összes a megoldásban),
- az egyes hálózati eszközök kapacitását,
- minden egyes időszakra a forgalmi igényekhez rendelt útvonalakat.

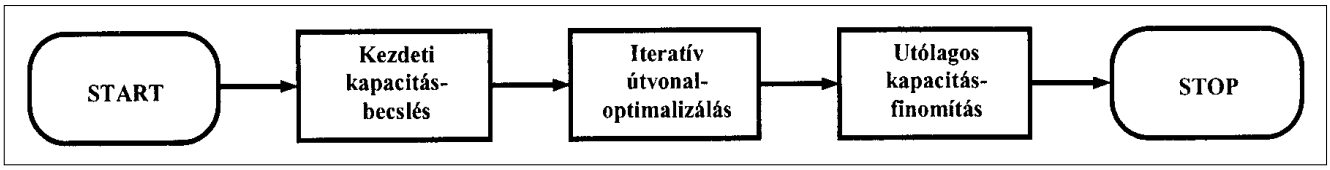
3. A javasolt hálózattervező algoritmus

Az alábbiakban először vázoljuk a javasolt algoritmus felépítését, majd részletesen is bemutatjuk az egyes fázisok működését. Az algoritmus a *Multi-Hour Core Network Designer (CND^{MH})* nevet kapta.

3.1. Az algoritmus felépítése

A forgalomeloszlás napi változásait figyelembe vevő hálózattervezési probléma megoldására kifejlesztett algoritmus a [8]-ban ismertetett heurisztikus algoritmusból (a továbbiakban ezt értjük *eredeti algoritmus* alatt) indul ki, amely hatékony megoldást nyújt az egy időszakkal, vagyis egy igényhalmazzal dolgozó hálózattervezési feladatra. Az új algoritmus három fázisból áll *(4. ábra)*, melyek az eredeti algoritmus fázisainak kiterjesztései több forgalmi időszak kezelésére:

4. ábra Az algoritmus folyamatábrája



- *Kezdeti kapacitásbecslés (KKB):* feladata az összes időszak forgalmi igényeinek függvényében megbecsülni a szükséges eszközkapacitásokat, hogy megfelelő kiindulási alapot biztosítson a következő fázis számára. Az eredményül kapott hálózatban tipikusan nem vezethető el az összes forgalmi igény, vagyis ez a fázis általában még nem ad megoldást a problémára.
- *Iteratív útvonal-optimalizálás (IUO):* célja a forgalmi igények elvezetése egy globális útvonal-optimalizáló algoritmus segítségével oly módon, hogy a fázis végére minden időszakban az összes forgalmi igény kielégíthetővé váljon, ugyanakkor a hálózat összköltsége minél kisebb legyen. Ez a fázis már teljes megoldást ad az adott problémára, de az eredmény költsége a harmadik fázisban még finomítható.
- *Utólagos kapacitás-finomítás (UKF):* megkísérli a hálózat összköltségének mérséklését azáltal, hogy megpróbálja csökkenteni azon hálózati eszközök méretét, melyekről viszonylag kicsi sávszélességet más eszközökre terelve kisebb eszközméret is elegendő lenne.

3.2. Kezdeti kapacitásbecslés

A KKB fázis kezdetben minden időszakra egy két lépésből álló iterációt végez: az algoritmus először megkeveri az adott időszak forgalmi igényeit, majd a kapott sorrendben elvezeti azokat a hálózatban Dijkstra algoritmusának felhasználásával. Az elvezetés során alkalmazott súlyfüggvény előnyben részesíti az alacsony fajlagos költségű eszközöket. A kezdeti kapacitásbecslő algoritmus ezután minden eszköz kapacitását az iterációk során az adott eszközön előforduló legkisebb értékre állítja minden időszak minden elvezetését figyelembe véve. Emögött az a meggondolás áll, hogy amennyiben az összes időszak összes elvezetése során szükség volt egy bizonyos kapacitásra az adott eszközön, akkor valószínűleg optimális esetben is legalább ekkora eszközméret szükséges.

3.3. Iteratív útvonal-optimalizálás

Az algoritmus második (IUO) fázisa szintén egy két lépésből álló iteráció: először sorra minden időszakra megpróbálja elvezetni a forgalmi igényeket az alkalmazott globális útvonal-optimalizáló segítségével figyelembe véve a fennálló kapacitáskorlátokat. Ha ez a lépés sikeres, a fázis véget ér, ellenkező esetben pedig egy – a hálózatba be nem fért igények által meghatározott – eszköz kapacitását megnöveli, és újra megpróbálkozik az elvezetéssel. A kiválasztáshoz használt metrikának minden időszak forgalmi igényeit számításba kell vennie. Ezért a következő metrika alkalmazása mellett döntöttünk: az algoritmus minden időszakban elvezeti a hálózatba be nem fért igényeket egy kapacitáskorlátok nélküli gráfban, és az elvezetések során eszközönként összegzi, hogy a hálózatba be nem fért igények az adott eszköz kapacitáskorlátját mennyivel sértették volna, ha a korlátok figyelembe vételével történt volna az elvezetés. A második lépésben ezután azt az eszközt választja, amelyre a fenti összegzett kapacitáskorlát-sértés a legnagyobb. Az összegzés során az algoritmus az egyes időszakok súlyát is figyelembe veszi. A módszer fontos előnye, hogy a fázis által alkalmazott globális útvonal-optimalizáló modulárisan lecserélhető bármely hasonló célú algoritmusra. Jelen tanulmányban a [9]-ben javasolt módszert alkalmazzuk.

3.4. Utólagos kapacitásfinomítás

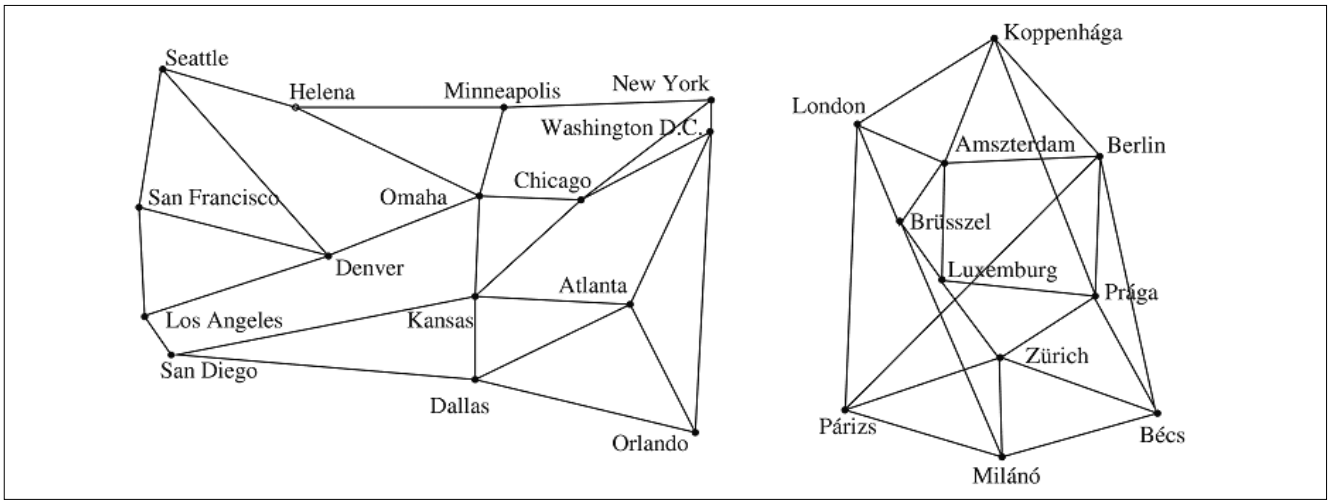
Az algoritmus harmadik fázisának célja a hálózat összköltségének finomítása oly módon, hogy megkísérli azon hálózati eszközök méretének csökkentését, amelyek gyengén kihasználtak olyan értelemben, hogy viszonylag kis forgalmat más eszközökre terelve egy kisebb méretű eszköz is elegendő lenne. Ehhez a jelenséghez kapcsolódóan bevezetjük a *relatív lépcsőkihasználtság* mértékét, amely megadja, hogy az adott eszközt használó forgalmi igények összsávszélessége és az eggyel kisebb kapacitáslépcsőhöz tartozó kapacitásérték különbsége hogyan aránylik az aktuális kapacitáslépcső hosszához. Az alkalmazott algoritmus először sorbarendezi az eszközöket a relatív lépcsőkihasználtságok forgalmi időszakok felett képzett súlyozott átlaga szerint. Ezután végiglépked az eszközökön, és az aktuális eszközt egy lépcsővel kisebb kapacitásúra állítja, majd újrafuttatja a teljes második fázist (IUO) az új kapacitáskorlátokkal. Az IUO adott esetben egyes eszközök kapacitását megnövelheti, ezért újrafuttatása után a hálózat költsége nem feltétlenül csökken. Amennyiben költségcsökkenés következik be, újraindul a harmadik fázis (UKF), ellenkező esetben az algoritmus visszatér a csökkentés előtti megoldáshoz, és a következő eszközre lép. A fázis akkor ér véget, ha egyik eszköz kapacitását sem sikerült csökkenteni. A módszer UKF fázisa során alkalmazott algoritmus a futás során kivonja a további vizsgálatok alól azokat az eszközöket, melyeket egymás után többször nem sikerült kisebb kapacitásúra cserélni.

4. A teljesítményvizsgálat eredményei

A következőkben bemutatjuk a javasolt CND^{MH} algoritmus teljesítményvizsgálata során a tesztproblémák előállításához alkalmazott módszereket és a két referenciaalgoritmust. Ezt követően ismertetjük a vizsgálatok legfontosabb eredményeit.

4.1. Tesztproblémák generálása

A teljesítményvizsgálat során az algoritmust különböző – valós és véletlenszerűen generált – problémapéldányokra futtattuk. A következőkben ismertetjük a problémapéldányok előállításának részleteit, azaz a hálózati topológiák, a forgalmi igényhalmazok és a költségfüggvények generálását. A problémapéldányok előállítása a [8]-ban ismertetetthez hasonló, de a forgalom generálásakor egy bonyolultabb módszer alkalmazására volt szükség a periodikus forgalomeloszlás-változások modellezéséhez.



5. ábra A teljesítményvizsgálathoz használt két valós topológia

A véletlenszerű hálózati topológiák előállítását a [9]-beli módszerrel végeztük, amely a valós hálózatok jellemzőit jól közelíti. Jelen tanulmányban 15, 25 és 35 útvonalválasztót tartalmazó topológiákra mutatunk be eredményeket. Az útvonalválasztók kiindulási átlagos fokszáma minden esetben 5 (a kezdetben adott fizikai összeköttetések közül nem feltétlenül szerepel az összes a végső hálózatban). A véletlenszerű topológiák mellett az *5. ábrán* látható két valós hálózatra (a továbbiakban USA és Európa) is vizsgáljuk az algoritmus teljesítményét.

A forgalmi igények meghatározása során a [8]-beli módszerből indultunk ki, azonban a nap folyamán változó forgalom előállításához több módosításra volt szükség. A módosított forgalomgeneráló algoritmus lépései:

• A hálózati topológia felosztása különböző területekre, úgynevezett *régiókra* úgy, hogy minden régióba megközelítőleg azonos darabszámú csomópont kerüljön. A régiókra való felosztás során az algoritmus a gráf középpontjához legközelebbi útvonalválasztókat besorolja egy központi régióba, a többi régióba pedig az azonos szögtartományba eső útvonalválasztók kerülnek *(6. ábra)*.

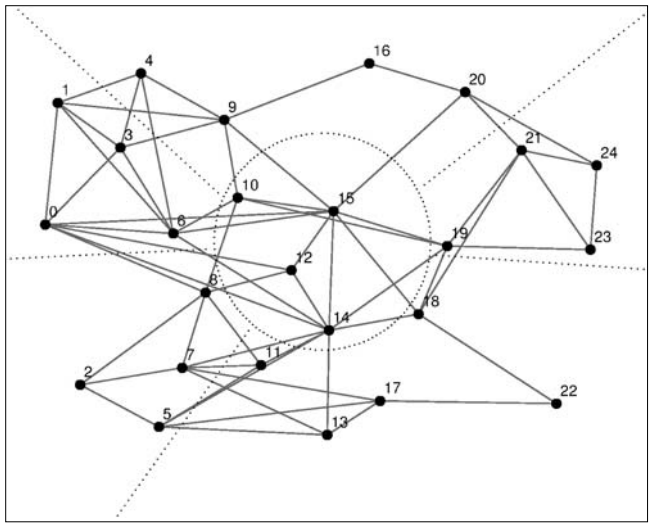
• A régiók párba állítása minden időszakra. Ezzel a megoldással modellezzük azt a jelenséget, hogy a nap folyamán a hálózat más-más területei között lép fel nagyobb forgalom. A párba állítás úgy történik, hogy két régió egymásnak ne lehessen párja egynél több időszakban. A vizsgálatok során a régiók és az időszakok száma is 6 volt, az időszakokhoz pedig egységnyi súlyt rendeltünk.

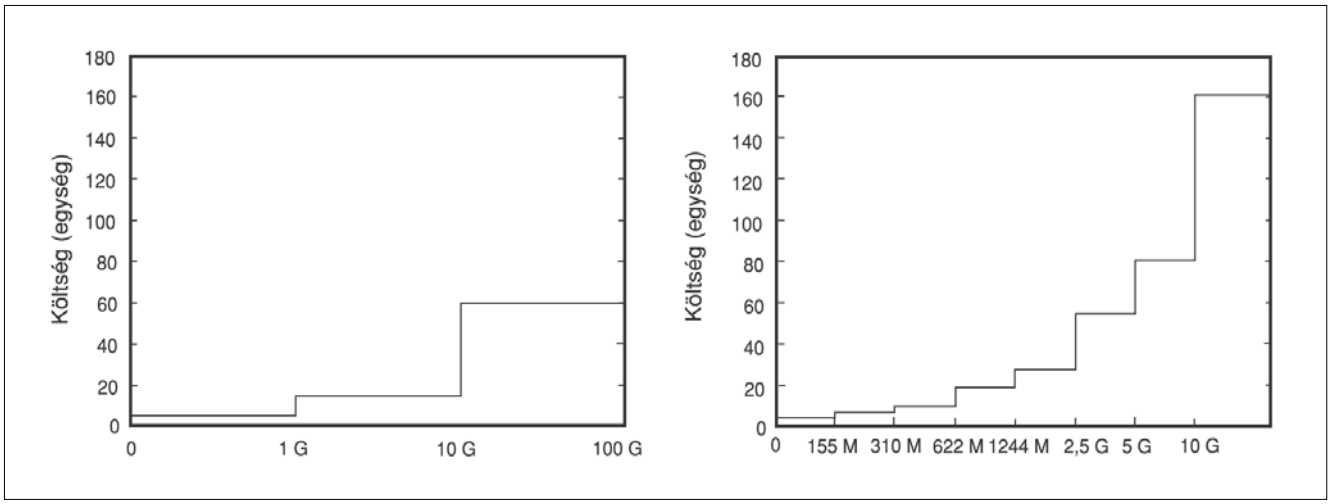
• A forgalmi igények meghatározása minden időszakra egy Δ paraméter, az úgynevezett *régiós torzítási faktor* figyelembevételével. Egy adott időszakban a régióhoz rendelt pár kitüntetett szerepet játszik: a régióból induló forgalom egy része kijelölten a régió párjába irányul. Ennek a forgalomnak az arányát a Δ régiós torzítási faktor adja meg. A régióból induló forgalom fennmaradó része egyenletesen oszlik meg az összes régió között, beleértve magát a régiót és az adott időszakbeli párját is. Így a Δ=0%-os régiós torzítás jelenti a teljesen egyenletes forgalomeloszlást, Δ=100% esetén pedig a régióból induló teljes forgalom a régió párjába jut. A vizsgálatok során öt régiós torzítási faktor mellett elemeztük az algoritmusok teljesítményét: 0%, 25%, 50%, 75%, 100%.

• Az egyes időszakok forgalmi igényeinek skálázása a *forgalomvolumen* paraméternek megfelelően, amely az útvonalválasztó-párok közötti teljes forgalom nagyságát írja le. Az igények skálázása úgy történik, hogy az egyes időszakok igényhalmazának elemeit Dijkstra algoritmusával elvezetve a hálózatban egy fizikai összeköttetésre átlagosan ezen paraméter által megadott nagyságú forgalom jusson.

A költségfüggvények generálásához az alábbi módszert használtuk. Az útvonalválasztóknál három, a fizikai összeköttetéseknél kilenc kapacitásszintet definiáltunk. A fizikai összeköttetések esetén a kapacitásszintek az *STM (Synchronous Transfer Mode)* szabvány kapacitásszintjeinek felelnek meg, illetve azoknak az eseteknek, amikor párhuzamosan két azonos kapacitású eszközt telepítenek.

6. ábra A hálózati topológia régiókra osztása

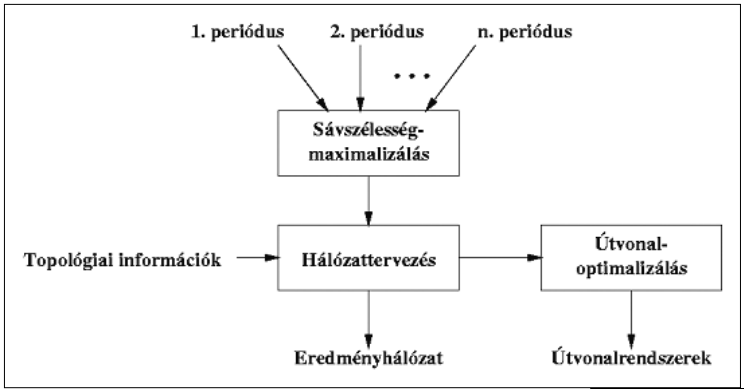




7. ábra A vizsgálatok során alkalmazott költségfüggvények

A költségfüggvény azon a feltételezésen alapul, hogy két azonos kapacitású eszköz költsége kisebb, mint az eggyel nagyobb kapacitásúé, azonban három azonos méretű eszköz telepítésénél jobban megéri eggyel nagyobb eszközt telepíteni. Emellett a 0 kapacitásszint is megengedett, ami azt jelenti, hogy az adott két útvonalválasztó között a tervezett hálózatban nincs fizikai összeköttetés, vagyis a tervezőalgoritmus az adott fizikai összeköttetést elhagyta a kiindulási halmazból. Az útvonalválasztók és fizikai összeköttetések kiindulásként használt kapacitásértékeit és a hozzájuk tartozó költségeket a *7. ábra* mutatja. A kiindulási értékeket minden hálózati eszköznél véletlenszerűen torzítja az algoritmus, és így egyedi költségfüggvényeket hoz létre.

8. ábra A sávszélesség-maximalizáló (SM) referenciaalgoritmus folyamatábrája



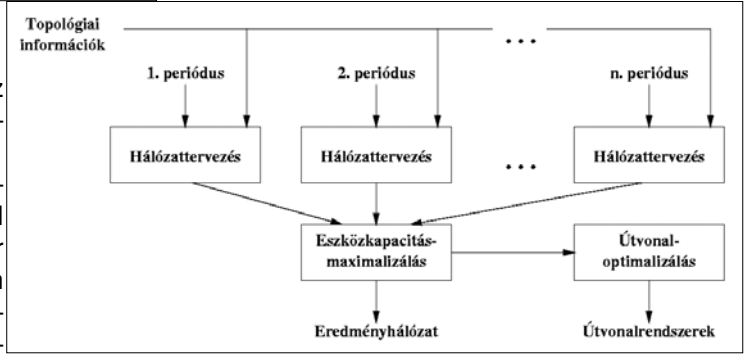
4.2. Referenciaalgoritmusok

Az algoritmus teljesítményének vizsgálatához az eredeti algoritmusból kiindulva két referenciamódszert implementáltunk.

A *sávszélesség-maximalizáló (SM)* referenciaalgoritmus a statikus forgalommodellezésnek felel meg. Az algoritmus minden útvonalválasztó-pár esetén meghatározza a közöttük a nap folyamán fellépő maximális kapacitásigényt, és erre a maximalizált igényhalmazra futtatja le az eredeti algoritmust az egyes hálózati eszközök kapacitásának kiszámításához *(8. ábra)*. Ezután az IUO fázis által használt globális útvonal-optimalizáló algoritmus segítségével minden időszak forgalmi igényeit elvezeti a hálózatban, hogy meghatározza az egyes időszakokhoz tartozó útvonalrendszert. Az SM referenciaalgoritmus előnye, hogy az algoritmus által az adott útvonalválasztópár közötti maximális forgalomra kiszámított útvonal statikus útvonalként is alkalmazható az adott útvonalválasztó-pár között különböző időszakokban fellépő forgalmi igényekre.

Az *eszközkapacitás-maximalizáló (EKM)* referenciaalgoritmus minden egyes forgalmi igényhalmazra külön-külön tervez egy hálózatot a kiindulási algoritmus segítségével, majd minden hálózati eszköz kapacitását az egyes időszakokra adódó hálózatokban az adott eszközön előforduló legnagyobb értékre állítja *(9. ábra)*. Ezután az SM referenciaalgoritmushoz hasonlóan határozza meg a végleges útvonalrendszereket, vagyis a végső kapacitáskorlátok mellett az alkalmazott globális útvonal-optimalizáló algoritmus segítségével elvezeti az egyes időszakok forgalmi igényeit. Természetesen az egyes időszakokra végzett hálózattervezés során előálló útvonalrendszerek is alkalmazhatók, de a fenti módszer hatékonyabb elvezetést tesz lehetővé.

9. ábra Az eszközkapacitás-maximalizáló (EKM) referenciaalgoritmus folyamatábrája



4.3. Numerikus eredmények

Ez a fejezet a szimulációs vizsgálatok eredményeit mutatja be. A véletlenszerű hálózati topológiák esetén minden hálózatméretre 5 topológiára és topológiánként 3 forgalmi szituációra, a valós topológiák esetén pedig 15 forgalmi szituációra futtattuk az algoritmusokat. A forgalomvolumen paramétert 200 és 5500 Mbit/s között változtattuk a vizsgálatok során. A legfontosabb mérték a teljesítményvizsgálat során a hálózat összköltsége volt, de emellett vizsgáltuk a futási időket is.

A hálózat összköltsége

Az *1. táblázat* mutatja az egyes algoritmusok által tervezett hálózatok átlagos összköltségének arányait (az adott problémaosztályban a legjobb megoldást tekintve 100%-nak) az összes vizsgált forgalomvolumenre átlagolva a 15, 25 és 35 útvonalválasztót tartalmazó véletlenszerű topológiákra, valamint a vizsgált két valós hálózatra. A véletlenszerű topológiák esetén három dolgot fontos megjegyezni: először is a javasolt algoritmus az esetek túlnyomó többségében a legalacsonyabb összköltségű hálózatot eredményezte. Kivételes esetek Δ=0% régiós torzítási faktor mellett fordultak elő, azonban teljesen egyenletes forgalomeloszlás esetén nem is várható az eredeti algoritmusnál jobb teljesítmény. Másodszor, a régiós torzítás növelésével egy adott hálózatméret mellett a javasolt algoritmus előnye növekszik a referenciaalgoritmusokkal szemben, és a Δ=100% esetben elérheti a 28%-ot az EKM-mel szemben, illetve az 54%-ot az SM-mel szemben. Harmadszor, a Δ=100% esetet leszámítva a hálózat méretének növelésével a javasolt algoritmus előnye enyhén csökkenő tendenciát mutat.

A valós hálózatok esetén is hasonló eredmények figyelhetők meg. A vizsgálatok azt mutatják, hogy a javasolt algoritmus az USA topológia esetén 8-34%-kal alacsonyabb költséget eredményez, mint az SM-é, és 18-24%-kal alacsonyabbat, mint az EKM-é. Az Európa topológia esetén az algoritmus alkalmazásával elérhető nyereség 7-30% az SM-hez képest, illetve 15-28% az EKM-hez képest.

Futási idők

A javasolt algoritmus és a referenciaalgoritmusok hozzávetőleges futási idejét a *2. táblázat* tartalmazza. A szimulációk egy 450 MHz-es Ultra II processzorral és 1 GB RAM-mal rendelkező Sun Ultra Enterprise 420R számítógépen futottak. A táblázat a javasolt algoritmusra és a referenciákra az adott hálózatméret és Δ régiós torzítás melletti minimális és maximális futási időt tartalmazza percekre kerekítve.

Látható, hogy 15 útvonalválasztót tartalmazó hálózati topológia esetén az algoritmusok futási ideje nagyon alacsony, maximum pár perc. 25 útvonalválasztó esetén a javasolt algoritmus és az EKM futási ideje a 2-20 perc, az SM-é pedig a 0-8 perc nagyságrendbe esik. 35 útvonalválasztó esetén az órás nagyságrendbe esik a javasolt algoritmus futási ideje, ami elfogadható a hálózattervezési probléma jellegét és a gyakorlatban előforduló hálózatméreteket tekintve.

Hálózat	Algoritmus	Δ=0%	Δ=25%	Δ=50%	Δ=75%	Δ=100%
15 csomópont	CND^{MH}	100,00%	100,00%	100,00%	100,00%	100,00%
	SM	100,17%	107,32%	123,51%	145,96%	153,75%
	EKM	105,75%	108,59%	114,42%	125,36%	127,90%
25 csomópont	CND^{MH}	100,70%	100,00%	100,00%	100,00%	100,00%
	SM	100,00%	103,97%	118,83%	139,26%	141,90%
	EKM	106,48%	106,89%	108,34%	113,99%	117,67%
35 csomópont	CND^{MH}	104,70%	100,00%	100,00%	100,00%	100,00%
	SM	100,00%	102,39%	116,50%	134,20%	149,62%
	EKM	107,99%	107,09%	107,56%	111,44%	121,88%
USA	CND^{MH}	100,00%	100,00%	100,00%	100,00%	100,00%
	SM	108,08%	109,14%	115,12%	126,66%	134,96%
	EKM	121,11%	121,98%	123,12%	123,69%	117,57%
Európa	CND^{MH}	100,00%	100,00%	100,00%	100,00%	100,00%
	SM	111,42%	106,86%	120,05%	129,85%	128,74%
	EKM	126,74%	122,54%	127,83%	128,40%	115,19%

1. táblázat
A hálózat relatív összköltsége a 15, 25 és 35 útvonalválasztós véletlenszerű, valamint a vizsgált két valós topológia esetén

Hálózatméret	Algoritmus	Δ=0%	Δ=25%	Δ=50%	Δ=75%	Δ=100%
15	CND^{MH}	0–2	0–3	0–3	0–2	0
	SM	0	0–1	0–1	0–1	0
	EKM	0–2	0–2	0–2	0–2	0
25	CND^{MH}	2–18	2–13	3–19	4–20	0–2
	SM	0–4	0–8	0–7	0–7	0–7
	EKM	3–24	3–26	4–23	5–16	0–1
35	CND^{MH}	12–45	17–40	19–46	18–61	2–5
	SM	4–14	4–17	6–48	5–29	5–17
	EKM	19–49	20–189	21–60	21–66	1–3

2. táblázat
Az algoritmusok futási ideje percben 15, 25 és 35 útvonalválasztós topológiák esetén (min-max)

25 és 35 útvonalválasztós topológiák esetén megfigyelhető, hogy a javasolt algoritmus és az EKM referenciaalgoritmus futási ideje az SM futási idejének többszöröse Δ=0%, 25%, 50%, 75% esetén.

Ennek magyarázata, hogy az SM egyetlen maximalizált igényhalmazzal dolgozik, vagyis futási ideje az eredeti algoritmus futási idejének nagyságrendjében van. Az EKM referenciaalgoritmus minden időszak forgalmi igényhalmazára külön futtatja a kiindulási algoritmust, a javasolt algoritmus pedig szintén több igényhalmazzal számol egyszerre, ami indokolja a többszörös futási időt. Kivételt képez a Δ=100% régiós torzítási faktor esete: ekkor mind a javasolt algoritmus, mind az EKM referenciaalgoritmus jóval rövidebb idő alatt szolgáltat megoldást, mint az SM.

Ennek az az oka, hogy a Δ=100% érték miatt a javasolt algoritmusnak és az EKM-nek időszakonként jóval kevesebb forgalmi igénnyel kell számolnia (mivel minden útvonalválasztóból csak a pár-régióbeli útvonalválasztókba irányul forgalmi igény), mint a maximalizálás miatt ilyenkor is egy „teljes" (vagyis minden útvonalválasztó-pár közötti igényt tartalmazó) igényhalmazzal dolgozó SM-nek.

5. Összegzés és konklúzió

Jelen tanulmány a kommunikációs hálózatok napi forgalomváltozásokat is figyelembe vevő tervezését tárgyalta. Az egyes hálózati eszközök költség/kapacitás viszonyait lépcsős költségfüggvényekkel írtuk le, emiatt a vizsgált probléma az NP-nehéz kategóriába tartozik. A forgalomeloszlás napi változásait diszkrét módon modelleztük: a nap ekkor rögzített számú időszakra oszlik, és minden időszakra külön-külön adott a forgalmi igények halmaza.

A javasolt algoritmus az összes forgalmi igényhalmazt egyszerre figyelembe véve oldja meg a hálózattervezési problémát. Referenciának két módszert implementáltunk: az SM minden útvonalválasztó-pár közötti napi maximális forgalomra tervez, az EKM pedig külön tervez hálózatot az egyes időszakokra, majd végül az adódó kapacitások maximumát képzi minden eszközre.

A javasolt algoritmus teljesítményvizsgálatát szimuláció segítségével végeztük különböző problémapéldányokra. Az alkalmazott hálózati topológiák között valós és véletlenszerű topológiák is megtalálhatók voltak. Bemutattuk, hogy a javasolt algoritmus segítségével jelentős költségcsökkenés érhető el mind az SM, mind az EKM referenciaalgoritmushoz képest. A javasolt algoritmus futási ideje még a legnagyobb vizsgált hálózatméret esetén is elfogadható volt.

A tárgyalt problémával kapcsolatban további vizsgálatok lehetségesek. A javasolt algoritmus esetében különböző fejlesztések képzelhetők el, ilyen például a védelmi útvonalak alkalmazásának lehetősége. Emellett a forgalmi igényhalmazok előállításakor más régiógenerálási és régiópárosítási módszerek hatását is érdemes megvizsgálni. A teljesítményvizsgálathoz újabb referenciamódszereket is célszerű lenne implementálni: ehhez adott esetben a probléma definíciók illesztése is szükséges lehet, például ha az adott módszer nem lépcsős költségfüggvényeket alkalmaz.

Irodalom

[1] ITU–T Recommendation E.360.6, QoS Routing and Related Traffic Engineering Methods – Capacity Management Methods, 2002.

[2] Medhi, D., A Unified Approach to Network Survivability for Teletraffic Networks: Models, Algorithms and Analysis, IEEE Transactions on Communications, Vol. 42, pp.534–548, 1994.

[3] Ouveysi, I., Network Design for Multi-Hour Traffic Profile, Proceedings of Australian Telecommunication Networks and Applications Conference (ATNAC), 1995.

[4] Bauschert, T., Multihour Design of Multi-Hop Virtual Path based Wide-Area ATM Networks, Proceedings of the 15th International Teletraffic Congress (ITC–15), Washington, D.C., USA, 1997.

[5] Medhi, D., Multi-Hour, Multi-Traffic Class Network Design for Virtual Path-based Dynamically Reconfigurable Wide-Area ATM Networks, Proceedings of the IEEE INFOCOM '95, Boston, MA, USA, pp.900–907, 1995.

[6] Medhi, D., Some Approaches to Solving a Multi-Hour Broadband Network Capacity Design Problem with Single-Path Routing, Telecommunication Systems, Vol. 13, pp.269–291, 2000.

[7] Medhi, D., Lu, C. T., Dimensioning and Computational Results for Wide-Area Broadband Networks with Two-level Dynamic Routing, IEICE Transactions on Communications, Vol. E80-B, No.2, pp.273–281, 1997.

[8] Orincsay, D., Józsa, B. G., Távközlési hálózatok költséghatékony tervezése, Híradástechnika, Vol. 58, No.4, pp. 39–45, 2003.

[9] Józsa, B. G., Király, Z., Magyar, G., Szentesi, Á., An Efficient Algorithm for Global Path Optimization in MPLS Networks, Optimization and Engineering, Vol. 2, No.3, pp.321–347, 2001.

SYN-áradatok automatikus szűrése a RESPIRE algoritmussal

GYIMESI JUDIT, KORN ANDRÁS, FEHÉR GÁBOR

BME Távközlési és Médianiformatikai Tanszék, HSNLab

gj309@hszk.bme.hu, gume@eik.bme.hu, korn@chardonnay.math.bme.hu

Reviewed

Kulcsszavak: támadás, számlálás, elárasztás, SYN cookie-k, szűrés

Számos ismert webszervert bénítottak meg rosszindulatú felhasználók hosszabb-rövidebb időre egy SYN-elárasztás (SYN flood) néven ismert támadás segítségével. Ezeknek a támadásoknak a kivédésére több olyan módszert javasoltak, amely bevált és elterjedt. Cikkünkben azonban egy olyan újszerű megoldást ismertetünk, amely lehetőséget biztosít a SYN-áradatok automatikus felismerésére és szűrésére anélkül, hogy számottevő többletterhelést okozna. Hatékonyságát mind szimulációval, mind numerikus analízissel alátámasztjuk.

Bevezető

A TCP SYN-elárasztás a TCP-kapcsolatfelépítés (handshake) egy sajátosságát használja ki. A kapcsolatot kezdeményező kliens először olyan csomagot küld a szervernek, amelyen a SYN bit be van állítva, egy kezdeti szekvenciaszámmal. A szerver erre egy SYNACK csomaggal válaszol, a saját szekvenciaszámával. Végül a kapcsolat létrejön, amint a kliens visszaküld egy olyan csomagot, amelyben csak az ACK bit van beállítva, és a szerver kezdeti szekvenciaszámát nyugtázza.

Ahhoz azonban, hogy a szerver el tudja dönteni, hogy egy beérkező ACK üzenet egy kapcsolat-felépítés utolsó üzenete-e, meg kell jegyeznie, melyik kliensnek milyen kezdeti szekvenciaszámú SYNACK csomagot küldött. Ezeket az adatokat a TCP kapcsolatpuffer (TCP backlog-queue) nevű adatstruktúrában tárolja, amelynek a mérete véges (gyakran portonként csak pár tucat félig nyitott kapcsolat tárolására elegendő).

A SYN-áradat során a támadó rengeteg SYN csomagot küld, gyakran hamis IP-címről, ám egyik kapcsolat felépítését sem fejezi be ACK üzenettel, így a szerver puffere megtelik, nem képes új kapcsolatokat fogadni. A pufferben tárolt, úgynevezett félig nyitott kapcsolatok egy idő után (timeout), ha addig nem érkezik rájuk ACK, törlődnek. Ám ha a támadó csomagok gyorsan jönnek, akkor elárasztják a tárolót, és az áldozat nem lesz képes a legtöbb legitim kliens kérésének feldolgozására.

Létező megoldások

Egyes gyártók, például a Cisco, forgalmaznak a SYN-elárasztás ellen védelmet nyújtó routereket. Általában ugyanazt a módszert alkalmazzák, mint az OpenBSD: a TCP-kapcsolatfelépítést ezek maguk végzik el, és a védett szervernek csak akkor küldik el a SYN csomagot, ha ők maguk már megkapták a végső ACK-ot. Ahhoz, hogy a kapcsolat működjön, a továbbiakban minden áthaladó csomagon módosítani kell a TCP-szekvenciaszámot (hiszen a router bizonyára más első szekvenciaszámot választott, amikor a SYNACK-ot küldte, mint a szerver). Emellett ezek a routerek hamarabb eldobják a félig nyitott kapcsolatokat, mint a szerverek, így valóban kevésbé érzékenyek a SYN-elárasztásra. Azt azonban látnunk kell, hogy a voltaképpeni problémát nem oldják meg, csupán megnövelik a támadás költségét. Továbbra is szükség van erőforrások (memória) allokálására minden félig nyitott kapcsolathoz, ezek az erőforrások pedig végesek. A támadó nem a szerveren, hanem a routeren foglalja le őket, de a hatás szempontjából ennek nincs jelentősége.

Egy másik javasolt védelem alapja a SYN csomagok véletlenszerű eldobása (RED jelleggel) [RAD]. Hasonlóan a félig nyitott kapcsolatok élettartamának csökkentéséhez, ez a megoldás sem nyújt valódi védelmet, csupán a támadás költségét növeli meg.

Az elárasztás globális kezelésére léteznek nagyon általános, ennélfogva bonyolult, nehézsúlyú megoldások is [ACC].

A SYN-elárasztás elleni védekezés egy igen eredeti, és széles körben elterjedt módja a SYN cookie-k használata [SCS]. A módszer lényege az, hogy a szerver a saját SYNACK csomagjában beállított szekvenciaszámba belekódolja azokat az információkat, amelyeket különben a helyi pufferben kellene tárolni; így nincs szükség memória-allokációra, csak néhány számításra.

A bejövő ACK csomag által nyugtázott szekvenciaszám segítségével a kapcsolat helyi adatstruktúrája felépíthető anélkül, hogy a SYN és az ACK beérkezése között bármit is tárolnunk kellene. Annak érdekében, hogy ez ne járjon azzal a veszéllyel, hogy az algoritmus ismeretében egy támadó helyesen megválasztott nyugtaszámmal kapcsolatot tudjon hamisítani, a beállított kezdő szekvenciaszámnak kriptográfiai értelemben erősnek kell lennie; így a támadónak csak nagy erőfeszítés (sok próbálkozás) árán sikerülhet érvényes nyugtaszámot generálni.

A SYN-cookie-k hátrányai

Azonban a SYN cookie-k használata hátrányokkal is jár. Először is, az ilyen módon létrehozott kapcsolatok nem használhatnak nagy ablakméretet, és a maximális szegmensméret sem választható meg szabadon. Másodszor, az erős szekvenciaszám előállítása számításigényes; Bernstein pl. a Rijndael kódoló használatát javasolja minden egyes SYNACK csomag szekvenciaszámának kiszámítására. Harmadszor, és talán ez a legnagyobb probléma, a SYN cookie-kat használó szerver a SYN-áradatra SYNACK-áradattal válaszol – ha a SYN csomagok feladója hamis cím, akkor a hamis címre, vagyis egy mit sem sejtő, ártatlan harmadik félnek (bounce attack).

Így, annak ellenére, hogy a SYN cookie-k még támadás esetén is garantálják a szolgáltatás elérhetőségét, továbbra is van értelme a támadók csomagjait szűrni (vagyis megakadályozni, hogy eljussanak a szerverhez).

Az itt bemutatott RESPIRE algoritmus jó kiegészítése a SYN cookie-knak is; ezek szavatolják a szolgáltatás zavartalanságát, a RESPIRE mechanizmus pedig felderíti a SYN-áradat forrásait, és kiszűri őket. *(RESPIRE: Resource Efficient SYN-flood Protection for Internet Routers and End-systems* – Erőforráshatékony SYN-áradat elleni védelem az Internet hosztjai és routerei számára).

Mivel azonban a RESPIRE reakcióideje igen rövid, a SYN cookie-kra elegendően nagy kapcsolatpuffer megléte esetén voltaképpen nincs szükség.

Az irodalomban több módszert is találunk a folyamatban levő SYN-elárasztás felismerésére; az újabb algoritmusok egyikét az Olvasó a [DSF]-ben találhatja meg. Ennek a módszernek az a hátránya, hogy a támadás elleni védekezéshez csak akkor nyújt hathatós segítséget, ha a támadó közelében lehet elhelyezni azt az eszközt, amely megvalósítja; ez lényegében azt jelenti, hogy minden Internet-szolgáltatónál be kellene vezetni. Amíg erre nem kerül sor, vagy ha az eszközt az áldozat közelében helyezzük el, az algoritmus csak felismerni képes a támadást, azt azonban nem tudja megállapítani, melyik állomás küldi az áradatot.

A RESPIRE rendszer elve

Az itt javasolt módszer nem igényli további adatgyűjtő eszközök elhelyezését. Azokat az adatokat használjuk fel a támadás felismerésére, amelyeket az áldozatnak amúgy is gyűjtenie kell ahhoz, hogy TCP szolgáltatást legyen képes nyújtani.

Az alábbi adatok mind rendelkezésre állnak (vagy könnyen előállíthatók), és alkalmasak a támadás felismerésére, vagy legalábbis valószínűsítésére:

– a másodpercenként beérkező SYN csomagok száma meghalad egy küszöbértéket;
– valamely TCP port kapcsolatpuffere (backlog queue) megtelik, SYN cookie-kat kell küldeni a további kapcsolatok fogadásához;
– a félig nyitott kapcsolatok száma meghalad egy küszöbértéket;
– aránytalanul több SYNACK csomag hagyja el a rendszert, mint ahány kapcsolat-felépítést véglegesítő ACK csomag érkezik (a továbbiakban ACK üzeneten mindig ilyen csomagot értünk).

A RESPIRE itt leírt változata az utóbbi heurisztikát alkalmazza, azonban minimális módosításokkal akár az összes módszer kombinációja is használható.

Fogalmak, rövidítések

A.B.C.D/E
Ez a jelölés egy olyan IP-alhálózatot jelöl, ahol a rendelkezésre álló 32 bitből az első *E* darab a hálózatot azonosítja, a maradék 32-*E* darab pedig az állomásokat a hálózaton belül.
A Budapesti Műszaki és Gazdaságtudományi Egyetem címtartománya például a 152.66.0.0/16. *E*-t szokás maszkméretnek hívni, mivel az alhálózati maszkban található 1-es bitek számát adja meg.

ACK
A TCP-fejléc egyik jelzőbitje; azt jelzi, hogy a csomag nyugtázza valahány korábbi adategység vételét.

cookie
Szó szerint *„süti”*; az informatikai biztonságtechnika területén valamilyen kriptográfiai módszerrel előállított adat, amit általában hitelesítésre használnak.

DoS
A *„Denial of Service”* (szolgáltatásmegtagadás) rövidítése. A támadások azon csoportja, amely egy szolgáltatás szabotálását, megbénítását tűzi ki célul.

SYN
A TCP-fejléc egyik jelzőbitje; azokat a csomagokat, amelyeknek a fejlécében ez a bit egyes értékű, szokás SYN-csomagoknak nevezni. A TCP-kapcsolat felépítése egy SYN-csomag küldésével kezdődik.

SYNACK
SYNACK-csomagnak szokás hívni a TCP-kapcsolatfelépítés második csomagját, amelynek fejlécében mint a SYN, mind az ACK bit „1” értékű.

port
Kétbájtos végpont-azonosító, amivel a TCP (és mellesleg az UDP is) kiegészíti az IP-címet; így egy IP-címen több TCP-vel kommunikáló folyamat is létezhet, amelyeket a portszám különböztet meg egymástól.

RED
Random Early Drop – Olyan torlódásvezérlési mechanizmus, amely úgy kerüli el a torlódást, hogy még a torlódás kialakulása előtt valamilyen szempontok alapján kiválasztott csomagokat egy általában a torlódásveszély mértékétől függő valószínűséggel eldob.

spoofing
Címhamisítás.

szekvenciaszám
Minden TCP-vel átvitt adategységnek van egy szekvenciaszáma; ez lényegében az átvitt byte sorszáma plusz egy a kapcsolat elején kiválasztott véletlen eltolás. A véletlen eltolás megnehezíti a csomagok hamisítását.

Megjegyezzük, hogy lehetséges lenne a bejövő SYN és bejövő ACK üzenetek számának arányát is vizsgálni. A SYNACK üzenetekben küldött szekvenciaszám ismeretére mindenképpen szükségünk van a számolandó ACK-üzenetek azonosításához, a SYNACK üzenet alapján pedig rekonstruálható a hozzá tartozó SYN, így a SYN-ek számolása redundánsnak tűnhet. Hozzá kell azonban tennünk, hogy ahhoz, hogy a SYNACK-ok számolására alapozhassuk a védelmet, az áldozat képes kell, hogy legyen a bejövő SYN-ek elegendően nagy részére SYNACK-kal válaszolni. Ezt a SYN-cookie-k garantálják, ha azonban nem használunk SYN-cookie-kat, akkor a kapcsolatpuffer méretét kell úgy megválasztanunk, hogy a RESPIRE számára elegendő SYNACK üzenet termelődjön a puffer megtelése előtt. Ha ez sem lehetséges, akkor számolhatjuk a SYN-üzeneteket a SYNACK-ok helyett, ám a SYNACK-okkal ekkor is foglalkoznunk kell a szekvenciaszámok miatt.

Összefoglalva: a SYNACK-üzenetek helyett akkor célszerű a SYN-üzeneteket számolni, ha a védendő szerver nincs felkészítve SYN-cookie-k használatára, és a kapcsolatpuffer méretét sem tudjuk elegendően nagyra beállítani.

Támadáskor egy lehetséges védekezés, ha nem válaszolunk azokra a SYN csomagokra, amelyeket a támadó küld; ezt a legegyszerűbben úgy biztosíthatjuk, hogy tűzfalunkban kiszűrjük őket. Tehát a legfontosabb dolgunk izolálni a támadás forrásait. (Ennél többet csak akkor tehetünk, ha *pushback*kel [PSB] vagy más, hasonló mechanizmussal a támadó csomagjai által használt útvonalon a támadó felé „toljuk” a szűrést.) A támadó kiszűrése az Internet hőskorában gyakorlatilag lehetetlen lett volna, mivel a csomagok forráscímei szabadon hamisíthatóak voltak. Mostanra azonban a legtöbb hálózatból nem engednek ki olyan csomagot, amelynek az állítólagos feladója nem része a hálózatnak. Emiatt a támadók általában csak saját C osztályú hálózatukon belüli címeket tudnak használni. Globális szolgáltatást nyújtó szerverek esetében ennek az egész tartománynak a szűrése is csak elenyészően kevés legitim klienst érinthet, hiszen nagyon kedvező az arány az összes létező és a támadó által használt hálózatok száma között. Megjegyezzük, hogy a fenti feltételezés alapfeltétele a RESPIRE működésének; ha a támadó tetszőleges forráscímet képes lenne hamisítani, a RESPIRE még ronthatna is a helyzeten, mivel legitim klienseket is kiszűrhet a támadás vélt forrásának kitiltása során. Ennek a veszélye számottevő mértékben csökkenthető, ha a RESPIRE-t csomaghamisítás-érzékelővel (spoof detector), pl. a [HCF]-ben leírt algoritmussal kombináljuk.

A SYN-támadások anatómiája

Napjainkban az ilyen támadások során a támadó általában több tucat olyan számítógépről, „zombiról” küldi a SYN-áradatot, amelyekre korábban betört. Ezeken a gépeken elosztott támadások megvalósítására kifejlesztett programokat helyez el, amelyeket egyszerű utasításokkal képes távvezérelni. Hogy az áradat szűrését megnehezítsék, a zombik általában hamisított forráscímekről küldik a csomagokat; a fent leírtak miatt azonban mindegyik hamisított forráscím ugyanahhoz az IP-alhálózathoz tartozik, mint a zombi tényleges címe.

Megjegyezzük, hogy amennyiben a bejövő SYN-áradat sávszélessége elegendően nagy ahhoz, hogy az áldozat vonalát telítse, már nincs értelme SYN-támadásról beszélni; a támadás olyan általános kapcsolat-telítő támadás, amely történetesen TCP SYN csomagokat használ. Nem célunk ezzel az esettel foglalkozni, noha a *pushback*kel kombinálva a bemutatott RESPIRE algoritmus az ilyen támadások ellen is hatásos.

Ki a támadó?

Korábban már utaltunk arra, hogy SYN-áradat esetén a kimenő SYNACK csomagok és a bejövő, kapcsolatfelépítést véglegesítő ACK csomagok aránya sokkal nagyobb lesz egynél. Mivel a legtöbb válasz nélkül maradó SYNACK csomagot éppen a támadó SYN-csomagjaira adott válaszként küldjük el, a támadót úgy találhatjuk meg, ha megkeressük az(oka)t a hálózato(ka)t, amely(ek)nél nagy az egy érvényes bejövő ACK csomagra eső kimenő SYNACK csomagok száma.

Erre egy lehetséges naiv módszer az lenne, ha egy nagy (2^{24}, azaz 16,7 millió sort tartalmazó) táblázatban számolnánk, hogy hány SYNACK csomagot küldünk az egyes C osztályú hálózatokba, ill. hogy hány ACKot küldenek ezekből nekünk. Jól látható, hogy ez a megoldás nem lenne hatékony: a legtöbb számláló a nullán állna, és a pozitív értékeket mutató számlálópárok többsége is „normális” (1 körüli) arányt tükrözne. A támadó azonosításához mégis az összes sort meg kellene vizsgálnunk (*egy* támadó megtalálásához átlagosan mintegy nyolc és fél millió sort).

A RESPIRE működése

A RESPIRE alapötletét szolgáltató MULTOPS[MPS] ezt a problémát úgy oldja meg, hogy a számlálókat dinamikusan bővíthető hierarchikus adatstruktúrában, egy 256-odrendű fában tárolja, kihasználva az IP-címek hierarchikus jellegét.

A RESPIRE-ben a fa gyökere két, kezdetben nulla értékű számlálót, és 256 darab kezdetben NULL mutatót tartalmaz.Az egyik számláló (a neve `Synack_Out`), a rendszert elhagyó SYNACK üzeneteket számolja, a másik – `Ack_In` nevű – pedig a beérkező hiteles ACK csomagokat (azokat, amelyek egy TCP-kapcsolat felépítését véglegesítik).

Miután legalább `Synack_Min` darab SYNACK csomagot küldtünk ki, minden további `Synack_Period`

darab csomag után inkrementáljuk a megfelelő számlálókat és megvizsgáljuk a tárolt érékek arányait. Mint azt később látni fogjuk, a fastruktúra miatt ez aránylag olcsó művelet; így azt javasoljuk, hogy `Synack_Period` értéke legyen 1. Olyan szervereken, amelyeken különösen nagy forgalomra számítunk, a paraméter értéke növelhető a többletterhelés csökkentése érdekében; ez azonban rontja az algoritmus pontosságát és reakcióidejét. A determinisztikus mintavételezés helyett természetesen alkalmazhatunk valamilyen sztochasztikus módszert is, vagy változtathatjuk a paraméter értékét dinamikusan (pl. a forgalomtól függően), ez azonban a lényeget nem érinti.

Amint a fastruktúra gyökérelemében `Synack_Out` és `Ack_In` aránya meghaladja a választott, 1-nél nagyobb R_{max} paraméter értékét (1.5 körül javasoljuk megválasztani; az alacsonyabb értékekhez jobb reakcióidő tartozik, ám növelik a tévedés valószínűségét), feltételezzük, hogy SYN-elárasztás áldozatai vagyunk, és megkezdjük a kiküldött SYN ACK csomagok célcímeinek figyelését a támadók felismerése érdekében. Ezeknek megfelelően bővítésjük a fát.

Minden további `Synack_Period` darab kimenő SYNACK vagy bejövő ACK csomag esetén megjegyezzük a távoli IP-címet: legyen ez A.B.C.D. Amennyiben a gyökér A-adik mutatója NULL, létrehozunk egy új csúcsot, és befűzzük a gyökér alá (`root`→`A`). A továbbiakban minden, az A.0.0.0/8 hálózattal összefüggő SYN/ACK forgalmat két helyen számolunk: a gyökérben és az imént létrehozott új csúcsban.

Ha `root`→`A` már létezik, megvizsgáljuk, igaz-e, hogy

$$\text{root}\rightarrow A\rightarrow\texttt{Synack\_Out} \geq \texttt{Synack\_Min}_1 \quad \text{és}$$

$$\frac{\text{root}\rightarrow A\rightarrow\texttt{Synack\_Out}}{\text{root}\rightarrow A\rightarrow\texttt{Ack\_In}} > R_{max}$$

Amennyiben mindez teljesül, valószínűsíthető, hogy az A.0.0.0/8 hálózat rejti a(z egyik) támadót.

A fát a fenti algoritmussal tovább bővítjük, amíg az `A`→`B`→`C` levél létre nem jön.

A `Synack_Min` paraméter értéke különbözhet a fa egyes szintjein. Ha a lefelé haladva csökkentjük a paraméter értékét, a támadások felismerése gyorsul, a pontosság azonban romlik; ezt ellensúlyozandó növelhetjük, például R_{max} értékét. Ezeket a finomhangolási lehetőségeket majd egy későbbi cikkben vizsgáljuk meg.

Amennyiben az `A`→`B`→`C` levél létezik, már legalább $\texttt{Synack\_Min}_C$ SYNACK csomagot gyűjtött, és számlálóinak aránya meghaladja R_{max}-ot, feltételezzük, hogy az A.B.C.0/24 egy támadó ellenőrzése alatt áll, és forgalmát a továbbiakban kiszűrjük. Néhány lehetőség a szűrés megvalósítására, a teljesség igénye nélkül:

– Az operációs rendszer TCP-megvalósításának bővítése a szűrés képességével.
– Az operációs rendszer beépített csomagszűrőjének használata (ha van ilyen).
– A *pushback* [PSB] vagy más hasonló mechanizmus segítségével szűrés kérése egy a támadóhoz közelebbi routertől.

Célszerű ezeket a szűréseket egy idő (`Block_Timeout`) után megszüntetni. A SYN-támadások általában nem tartanak 15 percnél tovább, így ezt az értéket javasoljuk. A támadás megszűntének érzékeléséhez lehetne ugyan néhány drága heurisztikával próbálkozni, mint például az újraküldött SYN-ek felismerése, ennél azonban sokkal gazdaságosabb a szűrés átmeneti megszüntetésével kipróbálni, véget ért-e a támadás. Amennyiben a támadás folytatódik, természetesen újra felismerjük és újabb 15 percre kiszűrjük (itt is lehetséges lenne valamilyen adaptív viselkedés bevezetése).

Ha találtunk és kiszűrtünk egy támadó alhálózatot, a hozzá tartozó levelet törölhetjük a fából, mivel már nem fogunk innen SYN csomagot kapni, és levonjuk számlálóinak értékét a fában fölötte levő csúcsok számlálóiból. Ha a gyökérben még ezután sem áll helyre az arány, további támadók is létezhetnek. További optimalizációs lehetőség az imént eltávolított csúcsok újbóli létrehozása a szűrés feloldásakor, hogy gyorsabban tudjunk reagálni, ha a támadás még nem ért volna véget.

`Prune_Interval` másodpercenként (2-nél alacsonyabb érték nem javasolt) megvizsgáljuk, van-e „gyanús” csúcs a fában (tehát olyan, ahol a számlálók hányadosa nagyobb R_{max}-nál, de ahol `Synack_Min`-t még nem értük el). Az összes nemgyanús csúcsot töröljük, a gyanúsaknak pedig nullázzuk a számlálóit. Ezzel memóriát és későbbi feldolgozási időt takarítunk meg. Megjegyezzük, hogy a törlendő csúcsok kiválasztására összetettebb algoritmust is használhatunk, amely például figyelembe veszi, hogy az adott csúcsban hogyan változott a számlálók aránya az előző `Prune_Interval` alatt; a RESPIRE alapötletének megértéséhez azonban elegendő az itt bemutatott naiv módszer. Egy összetettebb algoritmus a „lassú áradatok” lejjebb ismertetett problémájának megoldásában segíthet.

Azáltal, hogy a gyanús csúcsoknak csak nullázzuk a számlálóit, de a csúcsokat nem szüntetjük meg, az adott alhálózatból érkező támadót a következő `Prune_Interval` alatt hamarabb megtaláljuk, mivel nem kell megvárnunk, amíg a szülő-csúcsokban összegyűlik `Synack_Min` csomag; az alacsonyabb szintű csúcs eleve létezik. Az összetettebb algoritmusra vonatkozó fenti megjegyzések itt is megállják a helyüket.

A számlálók nullázására azért van szükség, mert csak a ténylegesen folyamatban levő támadásokra akarunk reagálni. Sajnos azonban így a támadó „lassú áradatok” („slow flood”) segítségével elkerülheti az észlelést. Ha rendkívül sok különböző C-osztályú hálózatból küld másodpercenként kevesebb, mint $\texttt{Synack\_Min}_C/\texttt{Prune\_Interval}$ csomagot, akkor a fa gyökerében ugyan látjuk, hogy támadás alatt vagyunk, de a támadó folyamok egyenként nem okoznak akkora forgalmat, hogy a fa alsóbb szintjein is elérjék `Synack_Min`-t a számlálók a nullázás előtt; együttes hatásukra mégis betelik a kapcsolatpuffer.

Ebben az esetben egy lehetséges reakció a következő: addig változtatjuk iteratívan a paraméterek értékét, amíg legalább B szintű címtartományig beazono-

sítjuk a támadás forrását. Ez célszerűen a `Synack_Min` és/vagy a `Prune_Interval` csökkentését jelentené. Természetesen így valószínűbb, hogy tévesen értékelünk valamit támadásnak, ráadásul lényegesen nagyobb címtartomány válik gyanússá, mint egy C osztályú hálózat esetén, tehát nem foganatosíthatunk drasztikus ellenintézkedést.

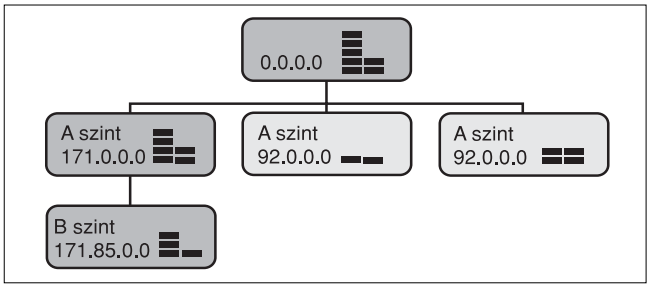
Ehelyett egy módosított RED algoritmust javaslunk. A Random Early Drop lényege, hogy amint a puffer telítettsége meghalad egy küszöbértéket, elkezd eldobni tetszőlegesen kiválasztott félig nyitott kapcsolatokat, a sor telítettségétől függően egyre többet. Ennek megvan az a hátránya, hogy a legitim kapcsolatkezdeményezéseket is érinti; kis változtatással azonban lényegesen csökkenthetjük ennek az esélyét. Legyenek nem kizárandóak, de gyanúsak azok az IP-címtartományok, amelyeket a fa alapján annak találtunk. Amint a puffer a megadott mértékig megtelik, csak ezek közül szelektáljunk a RED algoritmussal, így nagy valószínűséggel nem dobunk el legitim kapcsolatokat. Ennek a módszernek az a hátránya, hogy csak a kapcsolatpuffer megtelése ellen véd: a SYN-áradat által kiváltott SYNACK-áradatok keletkezését nem akadályozza meg.

Az *1. ábrán* egy háromszintű RESPIRE-fát láthatunk. Az egyes csúcsokban levő oszlopok a SYNACK és ACK csomagok relatív számát mutatják (nem pontos számértékeket). A 92.0.0.0/8-as és a 96.0.0.0/8-as csomópont közel ugyanannyi ACK csomagot küldött, amennyi SYNACK-ot kapott, tehát valószínűsíthetően legitim. A bal oldalon látható sötétebb hátterű csúcsok viszont nagyon is gyanúsak. Vegyük észre, hogy a gyökér is gyanús – ebből következtethetünk a támadás tényére.

A RESPIRE memóriaigénye

Mivel dinamikus adatstruktúrákkal dolgozunk, el kell kerülnünk, hogy maga a RESPIRE rendszer DoS-támadás eszköze lehessen: korlátoznunk kell a memóriahasználatát. Egy csúcs memóriaigénye 32 bites architektúrán (2+256)×32 bit (a két számláló plusz a 256 mutató). Ez összesen 1032 byte. Ha nem korlátoznánk a létrehozható csúcsok számát, összesen legfeljebb 16777216+65536+256 darab csúcsunk lehetne (ez a fenntartott címtartományok miatt nem túl pontos felső becslés); így a teljes RESPIRE adatstruktúra mérete elérhetné a 16,25 gigabájtot, aminek a kezelése már nehézkes.

1. ábra Példa egy RESPIRE-fára



A létrehozandó csomópontok száma a gyanús (támadó) alhálózatok számától függ. Nem valószínű, hogy egyetlen támadó kétszáznál több C-osztályú hálózatot ellenőrizne. A legkedvezőtlenebb eset az, ha ez a kétszáz C-hálózat mind különböző A-hálózatban található, ekkor ugyanis mindegyik áradatforráshoz három csomópontot kell létrehoznunk, összesen tehát hatszáz darabot (mintegy 600 kilobyte). Általában elegendőnek tűnik ötszázra korlátozni a létrehozható csúcsok számát; értelemszerűen, ha rendkívül elosztott támadásokra számítunk, ez a korlát növelhető.

Ha elértük a korlátot, de új csúcsot kellene létrehoznunk, megkeressük a gyökér legkevésbé gyanús gyermekét, és töröljük a hozzá tartozó részfával együtt. Ha csak egyetlen A-csúcsunk van, folytassuk a keresést az A-szinten; az egyetlen A-csúcsnak bizonyosan egynél több gyermeke lesz, mivel különben nem érhettük volna el az ötszázas korlátot. Idővel feltehetően izolálunk egy támadót, és csökken majd a csúcsok száma.

Analízis – a RESPIRE reakcióideje

Először általános közelítést adunk az algoritmus reakcióidejére, majd pedig felső határértéket. Látni fogjuk, hogy a RESPIRE még a legrosszabb esetben is gyorsan reagál (így a SYN cookie-k használata nem szükséges), ráadásul az észlelés ideje szempontjából a kis intenzitású áradatok jelentik a legrosszabb esetet: minél nagyobb az áradat intenzitása, annál gyorsabban ki tudjuk szűrni.

Jelen analízis arra az esetre vonatkozik, amikor a támadók – ha több van –, azonos C alhálózatban vannak. A számítások általánosíthatók összetettebb támadásra is, kivéve a „slow SYN flood” esetét, amit már tárgyaltunk.

Feltesszük, hogy a rosszindulatú SYN csomagok egyenletesen érkeznek, Ψ csomag/sec intenzitással. Ha több támadó van, akkor hatásukat összegződve tartalmazza ez az érték. A legitim kliensek SYN forgalma Poisson-folyamattal közelíthető, mivel a támadáson kívül az időegységenként érkező csomagok száma független egymástól. Az egyszerűség kedvéért feltesszük, hogy a SYNACK válaszcsomagot a szerver a SYN kézhezvételével egy időben kiküldi, így a kimenő SYNACK csomagok is Poisson-eloszlást mutatnak. Ugyanez érvényes a beérkező ACK csomagokra is, mivel bár az egy szakaszban bejövő ACK-ok nem az abban a szakaszban kimenő SYNACK-okra adott válaszok, de a darabszámuk várható értéke megegyezik, hiszen Poisson-folyamatnál csak a vizsgált időintervallum hossza, és nem a kezdőideje számít. Az *RTT (Round Trip Time)* figyelembevétele ugyan bonyolíthatná a helyzetet, de ha nincs észrevehető börsztösödés, akkor nem okoz nagy hibát a közelítés.

A támadás kezdetét attól az időponttól számoljuk, amikor a szerverhez ér az első rosszindulatú SYN csomag. Ez az adat csak annyiban számít, hogy mennyi idővel van egy `Prune_Interval` kezdete után.

Ez az érték legyen Δt_t. A továbbiakban Δt idő múlva már $\Psi \cdot \Delta t$ támadó SYN csomag érkezett.

Hogy eközben mennyi legitim kapcsolatkezdeményezés történt, azt a következő módon határozhatjuk meg: Poisson-eloszlásnál az egy időintervallumban beérkező csomagok számának várható értéke az intervallum hosszával arányos, $\lambda_l \cdot \Delta t$ (ahol az *l* index a felhasználók legitim voltára utal). Ugyanennyi SYNACK-ot küld ki a szerver, és közel ennyi ACK-nak is kell visszaérkeznie.

A RESPIRE mechanizmusa szerint két feltételnek kell teljesülnie, hogy felismerje a támadás tényét:

$$\frac{\lambda_l \cdot \Delta t + \psi_t \cdot \Delta t}{\lambda_l \cdot \Delta t} \geq R_{\max}$$

és

$$\lambda \cdot \Delta t + \psi_t \cdot \Delta t \geq synack\_min_{root}$$

Látszik, hogy jelen feltételezések mellett az első egyenlőtlenségben az arány nem függ az időtől, támadás esetén ennek mindig teljesülnie kell. Tehát ha egy támadás detektálható, akkor felismerjük, amint a minimális SYNACK értéket elérjük az adott `Prune_Interval`-ban (Δt_p).

A detektálhatóság feltétele pedig a megfelelő arány, amelyből a támadás intenzitására a következő adódik:

$$\Psi_t \geq (R_{\max} - 1) \cdot \lambda_l$$

Minden szintre hasonló gondolatmenet követhető, mint a gyökér esetén, azzal a különbséggel, hogy az egyes szintekre a legitim forgalomnak csak valamekkora hányada jut el. Ha feltételezzük, hogy minden alhálózatból azonos mennyiségű csomag jön, az egy A alhálózatból érkezők csak körülbelül az 1/256 hányadát teszik ki az egész beérkező forgalomnak, B-nél ennek négyzetét, C-nél köbét.

Ez természetesen nem lesz igaz, mivel a kliensek IP-címeinek eloszlása nem egyenletes. Az A szinten még közelítőleg sem az, mivel a teljes címtér jelentős része speciális célokra van fenntartva. Sajnos azonban még a B szinten sem tételezhetünk fel egyenletes eloszlást: Magyarországon például a 195-tel kezdődő IP-címek második oktetje nagy valószínűséggel 228, stb.

Bevezethetnénk a $\phi(x)$ paramétert úgy, hogy $\phi(x)$ a bejövő összes legitim SYN csomagnak az x alhálózatra eső részét jelenti.

Közelítésnek azonban elfogadhatjuk azt a megoldást, hogy A szinten valamilyen 'a' paraméterrel számolunk, a B és C szinten pedig egyenletesnek vesszük az eloszlást.

Ha szintenként másképp választottuk meg `Synack_Min`-t, akkor ezt is figyelembe kell venni.

Könnyen belátható, hogy hosszabb idő szükséges, ha a csomagszámlálás közben átlépünk egy `Prune_Interval`-határt, és törlődnek a számláló-értékek. Ekkor csak az adott szint számlálása kezdődik újból, hiszen magukat a csomópontokat nem töröljük.

Legyen I{*A*} az *A* esemény indikátora, mely 1 értékű, ha az esemény bekövetkezik, egyébként 0 – így jelenítjük meg azt, hogy az adott szint vizsgálatának elkezdése más intervallumba esik-e, mint a vége. Szintenként csak egy határátlépés lehetséges, ugyanis ha a csomópont gyanúságának eldöntéséhez több, mint egy intervallumnyi időre volna szükség, akkor a vizsgálat soha nem fejeződne be.

A szintenként szükséges időre a második felismerési feltétel alapján a következő képlet adható, ha határátlépés sehol nem történik:

$$\Delta t'_{root} \geq \frac{synack\_min_{root}}{\lambda_l + \psi_t}$$

$$\Delta t'_A \geq \frac{synack\_min_A}{\frac{1}{a} \cdot \lambda_l + \psi_t}$$

$$\Delta t'_B \geq \frac{synack\_min_B}{\frac{1}{a \cdot 256} \cdot \lambda_l + \psi_t}$$

$$\Delta t'_C \geq \frac{synack\_min_C}{\frac{1}{a \cdot 256^2} \cdot \lambda_l + \psi_t}$$

Ha az intervallumhatár-átlépéseket is figyelembe vesszük, akkor a fenti közelítésekkel élve az alábbi módon fejezhető ki az algoritmus reakcióideje a fa egyes szintjein *(lásd alul)*:

$$\Delta t_{root} = I\left\{\left[\frac{\Delta t_t}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t'_{root}}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t\right) + \Delta t'_{root}$$

$$\Delta t_A = I\left\{\left[\frac{\Delta t_t + \Delta t_{root}}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t_{root} + \Delta t'_A}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t + \Delta t_{root}}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t - \Delta t_{root}\right) + \Delta t'_A$$

$$\Delta t_B = I\left\{\left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A + \Delta t'_B}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t - \Delta t_{root} - \Delta t_A\right) + \Delta t'_B$$

$$\Delta t_C = I\left\{\left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A + \Delta t_B}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A + \Delta t'_C}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A + \Delta t_B}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t - \Delta t_{root} - \Delta t_A - \Delta t_B\right) + \Delta t'_C$$

Ha az előző szint vizsgálatának vége és az aktuális szint vizsgálatának vége külön intervallumba esik, akkor a szükséges idő a teljes, ehhez a szinthez szükséges, intervallumátlépés nélküli idő, plusz az előző szint vizsgálatának végétől az intervallumhatárig eltelő idő.

A teljes reakcióidő a fa egyes szintjein eltöltött idő összege:

$$\Delta T = \Delta t_{root} + \Delta t_A + \Delta t_B + \Delta t_C$$

Felső becslésként nézzük a reakcióidő szempontjából legrosszabb esetet, amely – mivel a reakcióidő detektálható támadás esetén csak a `Synack_Min` elérésétől függ – akkor következik be, amikor minden legális SYN csomag más A alhálózatból jön, mint a támadó csomagok. Ebben az esetben nem kell semmiféle feltételezéssel élnünk a forgalom eloszlásával kapcsolatban. Az értékek a következők szerint változnak:

$$\Delta t'_{szint} = \frac{synack\_min_{szint}}{\psi_t}$$

Az egyes szinteken töltött idő felülről becsülhető az átlépés nélküli eset idejének kétszeresével, hiszen ha a vizsgálat nem fejeződik be az intervallum-határig, akkor az indikátorfüggvény utáni szorzó kisebb, mint $\Delta t'_{szint}$, ellenkező esetben a vizsgálat befejeződhetett volna abban az intervallumban, amelyben kezdődött.

$$\Delta t_{szint} \leq 2 \cdot \Delta t'_{szint}$$

Tehát az algoritmus teljes reakcióideje nem haladhatja meg a legtöbb időt igénylő szint átlépés nélküli idejének négyszeresét:

$$\Delta T \leq 8 \cdot \max_{szint} \left\{ \frac{synack\_min_{szint}}{\psi_t} \right\}$$

A formulából jól látszik, hogy minél nagyobb intenzitású a támadás – vagyis minél nagyobb kárt okoz –, annál gyorsabban reagál a RESPIRE. Mint már utaltunk rá, ezt a felső becslést csak a különösen kis intenzitású támadások tudják megközelíteni. Ψ egyre nagyobb értékeinél egyre kisebb a valószínűsége, hogy kettőnél több intervallum kellene a támadó hálózat azonosításához; ugyanis a legkártékonyabb, nagyon gyors támadások felismerési ideje 0-hoz tart. A második intervallum csak akkor szükséges, ha a támadás egy intervallum végéhez közel kezdődik.

Szimuláció

Az algoritmus teljesítőképességét szimulációval is vizsgáltuk [RSP]. A kép teljessé tétele érdekében röviden e helyütt is összefoglaljuk a szimuláció eredményeit.

Egy nagy forgalmú SMTP-szervert szimuláltunk, átlagosan 62,8 folyamatban levő legitim kapcsolattal és 12,6 új kapcsolatkéréssel másodpercenként. A legitim kliensek IP-címe teljesen véletlenszerű volt. A rendszerben elhelyeztünk nyolc támadót, akik véletlen időpontban véletlen intenzitású SYN-áradattal támadták meg a szervert; a rendelkezésükre álló alhálózat mérete szintén véletlenszerű volt, /24-es alhálózati maszktól /16-osig (vagyis 256-65536 különböző IP-címről tudták küldeni csomagjaikat). Az alhálózatukon belül véletlenszerűen választottak forráscímet minden egyes támadó SYN csomaguknak, egy támadáson belül is váltogatva azt.

Nem feltételezünk annyi intelligenciát a támadótól, hogy azokból a címtartományokból nem küld, amit az áldozat algoritmusa már kiszűrt; ehhez figyelnie kellene az áldozat által elküldött SYN ACK üzenetek célcímeit. Még ha meg is oldaná, a RESPIRE reakcióját csak gyorsítaná, mivel az egész támadó intenzitás legitim címről érkezne, hamarabb elérnénk a `Synack_Min` -t.

A szimuláció néhány számszerű eredményét az *1. táblázat* foglalja össze. Láthatjuk, hogy az ötös és a hatos támadás kétszer is szerepel; ennek az a magyarázata, hogy a tűzfalszabály maximális élettartamát 900 másodpercre (15 perc) állítottuk be, ezek a támadások pedig ennél hosszabb ideig tartottak. A szabály elévülése után ismét fel kellett őket ismerni és kiszűrni. Az „elfogadott csomagok” oszlopban található értékek azt adják meg, hány – az adott támadótól származó – hamis SYN csomag érte el a szervert a támadás kiszűrésének pillanatáig; a többi oszlop jelentése remélhetőleg egyértelmű.

1. táblázat Szimulációs eredmények

Támadás sorszáma	Első csomag (s)	Alhálózat mérete	Csomagráta (csomag/s)	Elfogadott csomagok	Reakcióidő (s)
1	80,780	32768	41274	22191	0,629
2	239,046	512	91938	505	0,011
3	764,754	4096	58563	1920	0,045
4	890,803	512	82013	505	0,011
5	1229,573	16384	39586	6766	0,244
6	2039,08	8192	40932	3535	0,101
5*	2129,699	16384	39586	6767	0,165
6*	2939,157	8192	40932	3535	0,113
7	4060,253	32768	88895	13231	0,193
8	4729,277	512	31267	505	0,021

A reakcióidő tekintetében megállapíthatjuk, hogy az azonos mértékben elosztott, de nagyobb intenzitású támadások kiszűréséhez szükséges idő általában kisebb volt; a közel azonos csomagrátájú, ám elosztottabb támadás kiszűréséhez szükséges idő pedig általában nagyobb.

Elvégeztünk néhány próbaszámítást is, hogy összevessük az analízis eredményeit a szimulációéival:

Vegyük a 3. támadót. Ő 4096 címből álló tartományt ellenőriz; ez 16 darab szomszédos C osztályú hálózatot jelent. Így egyszerre 16 gyanús C szintű csúcsunk lesz a fában, mind azonos B csúcs alatt. Egyenletes címkiválasztást feltételezve a támadótól, minden C csúcsba a teljes intenzitás 1/16 része jut, vagyis 3660,19 csomag/s. A B szintig tehát alkalmazhatjuk az egy támadó alosztályt feltételező analízis eredményét:

$$\Delta T_{root} + \Delta T_A + \Delta T_B \le 6 \cdot \max_{szint=root,A,B} \left\{ \frac{synack\_min_{szint}}{\psi_t} \right\} = \quad ,01$$

A C csúcsok felismerésénél pedig egyenként 1830,09 csomag/s támadó intenzitással számolva:

$$\Delta T_C \le 2 \cdot \frac{synack\_min_C}{\psi_C} = 0,055$$

Az összes időre tehát az analízis alapján a fenti két eredmény összegét, 0,065s-ot kaptunk, ami jó felső becslése a mért értéknek. Nagyságrendileg tehát megegyezik a két módszer által mutatott eredmény.

Hasonlóképpen az 1. támadás által kiváltott reakció idejének felső becslésére 0,635s-ot, a 2.-éra 0,011s-ot, a 4.-ére 0,012s-ot, az 5.-ére 0,338s-ot, a 6.-éra 0,17s-ot, a 7.-ére 0,293s-ot, a 8.-éra pedig 0,032s-ot kapunk.

Nem kell figyelembe vennünk a szimulációs eredmények numerikus vizsgálatánál, ha több támadó gép is közel egy időben kezd el támadni. Ez meggyorsítja ugyan a gyökér gyanússá válását, ám az alsóbb szintekre nincs befolyással, mivel minden támadó más-más A osztályú hálózatban tartózkodik; a felső becslésnél pedig a legidőigényesebb szint idejét számítjuk minden szintre, ez pedig biztosan nem a gyökér. Így az a tény, hogy a gyökér hamarabb válik gyanússá, nem befolyásolja a számítást.

Természetesen előfordulhat, hogy nem azonos a választott hamis IP címek eloszlása, és esetleg egy C alhálózatot hamarabb felismerünk, mint egy másikat. Ha vannak olyan alhálózatok, amikben lényegesen kisebb az intenzitás, akkor az összidő növekedhet.

Vegyük észre azonban, hogy ez lényegét tekintve az az eset, amikor a lassú, ennélfogva veszélytelenebb támadásokat később ismerjük föl, hiszen a tartomány többi részét már tiltottuk, így az áldozatot ténylegesen elérő támadó forgalom kisebb intenzitású lesz.

Értékelés

A fent ismertetett RESPIRE algoritmus nem az egyetlen, amely a SYN-áradatok elleni védelmet szolgálja, ám kétségkívül az egyik legkisebb járulékos számításigényű módszer, ami megbízhatóan, gyorsan szűri ki a támadást; ezenkívül részben védelmet nyújt az ellen is, ha a szervert használják kisebb sávszélességű áldozatok kapcsolat-telítésére („bounce attack"). A SYN-támadások elleni védelem többi eszközének, elsősorban a SYN cookie-k kiegészítéseként is hasznos. Memóriaigénye is csak támadás esetén nő meg néhány kilobájtnyinál nagyobbra.

A fejlesztés korai fázisában levő linuxos referenciaimplementáció elkészülte után életszerű körülmények között is kipróbálható lesz az algoritmus.

Irodalom

[ACC] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, Scott Shenker, "Controlling High Bandwidth Aggregates in the Network", Computer Communications Review 32:3, July 2002, pp.62–73.

[HCF] Cheng Jin, Haining Wang, Kang G. Shin, "Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic", Proceedings of the 10th ACM conference on Computer and communication security, 2003, pp.30–41.

[SCS] Daniel J. Bernstein, "SYN cookies", http://cr.yp.to/syncookies.html, 1997.

[DSF] Haining Wang, Danlu Zhang, Kang G. Shin, "Detecting SYN Flooding Attacks", Proceedings of IEEE InfoCom, 2002

[PSB] John Ioannidis, Steven M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks", Network and Distributed System Security Symposium, February 2002.

[RAD] Livio Ricciulli, Patrick Lincoln, Pankaj Kakkar, "TCP SYN Flooding Defense", Comm. Net. and Dist. Systems Modeling and Simulation Conf. (CNDS' 99), 1999.

[MPS] Thomer M. Gil, Massimiliano Poletto, "MULTOPS: a data-structure for bandwidth attack detection", Proceedings of the 10th Usenix Security Symposium, August 2001.

[RSP] Gábor Fehér, András Korn, "RESPIRE – a Novel Approach to Automatically Blocking SYN Flooding Attacks", Proceedings of EUNICE 2004, pp.181–187.

Adatátviteli teljesítményvizsgálat szimmetrikus DSL berendezéseken

SZÉKELY SÁNDOR, KISS SZABOLCS MÁTÉ

Ericsson Magyarország,

sandor.szekely@siemens.com, szabolcs.mate.kiss@ericsson.com

Kulcsszavak: DSL, SHDSL, szélessáv, mérőhálózat

A DSL technológia az utóbbi időben széles körben terjed a kis- és közepes vállalkozások (SME, Small-to-Medium sized Enterprise), mint felhasználók körében, illetve az otthoni vagy kisebb irodákban (SOHO, Small Office/Home Office) is. Az xDSL technológia a hang és adatkapcsolatot réz érpáron valósítja meg. Cikkünkben miután ismertetjük a szimmetrikus DSL (SDSL, Symmetrical DSL) technológiát, valamint ennek jelenlegi világpiaci helyzetét, áttekintjük az SDSL technológián alapuló alkalmazások átviteli teljesítményéről, továbbá különböző gyártók termékeinek összehasonlításához is segítséget adjunk.

A cikk öt különböző gyártmányú CPE-páros (Customer Premises Equipment) mérési eredményét mutatja be mind az ATM mind az IP alapú gerinchálózathoz csatlakozó SDSL vonalakon. IP szinten vizsgáljuk a 2,3 Mb/s-os SDSL vonalakon elért adatátviteli sebességet és csomagkésleltetést különböző csomagméret esetén, valamint csomagvesztést is mérünk két különböző vonali kódolási módban (2B1Q és TCPAM16), ami az SDSL technológiában jelenleg használatos. A vizsgált eszközök mindegyike DSLAM-en (DSL Access Multiplexer) keresztül csatlakozik a gerinchálózathoz. Egyik CPE-páros esetében a fejléc az L2TP (Layer 2 Tunnelling Protocol), ezt a biztonságos IP (IPSecurity) és a titkosító kódolás miatt szintén elemeztük. Megmértük a CPE-k FTP (File Transfer Protocol) fájl forgalmi teljesítményét is. Végül pedig, minden egyes vizsgált eszköz vonatkozásában (vonalszimulátorok alkalmazásával) a vonali átviteli sebességtől függően a maximális előfizetői hurok hosszát is megmértük.

1. Bevezető

1.1. A szimmetrikus DSL technológiáról általában

A cikk célja - alapul véve a SHDSL (Symmetric High bit rate Digital Subscriber Line) szabványt [ITU01] - az új generációs szimmetrikus DSL szolgáltatásokat kínáló CPE-k teljesítményanalízise. Az általános DSLAM (DSL Access Multiplexer), amire kezdetben az ADSL épült (Asymmetrical DSL), az utóbbi időben már használják más szabványos DSL típusoknál is, úgy, mint SHDSL, VoDSL (Voice over DSL), és VDSL (Very high bit rate DSL) [Hum97].

A SHDSL általában a DSL technológiák fejlődése szempontjából [Bal98] az SDSL-t követi. Az egy réz érpáron működő ITU-T G. SHDSL ajánlás [ITU01] kedvezőbb sávszélesség-kihasználtsága, nyílt kezelhetősége és több szolgáltatási lehetősége miatt az eddigi technológia helyébe lépett. A nagy sávszélességű SHDSL folyamatosan leváltja majd a szimmetrikus kapcsolattípusokat egészen a T1/E1- és a régebbi generációs SDSL technológiákig, mivel akár 3600 m távolságig is képes 2,3 Mb/s-os adat- és jó minőségű hangátvitelt produkálni egyetlen réz érpáron [Bre02].

A régebbi HDSL/SDSL technológiák spektrum-kompatibilitási problémáinak leküzdése érdekében a piaci szabályozók jellemzően visszautasítják a helyi hurokban mindazokat a DSL-technológiákat, amelyek a spektrum lehetőségeit nem hatékonyan használják ki [TSAG01].

A spektrum kompatíbilitási problémák vezették a létező SDSL megoldásokat az SHDSL felé, amely a távközlési elérési hálózatokban a szimmetrikus adattovábbítás egy módja. A SHDSL átviteli egységeket tetszőleges szabványú kétirányú működésre tervezték, kéteres sodrott réz-érpáron (Magyarországon négyes sodrású réz erű kábeleket használunk).

A SHDSL kapcsolatok szimmetrikus előfizetői adatátvitelre képesek a 192-2304 kb/s-os tartományban. Ez lehetővé teszi a szolgáltatóknak, hogy a központ távolságától függően optimalizálják a kiosztott sávszélességet. Az új generációs DSLAM a TC-PAM16 (Trellis Coded Pulse Amplitude Modulation) vonali kódolást használja, amely módot ad a sávszélesség és az előfizetői vonalszakasz hossza közötti kompromisszumra. A kiterjesztett elérésű alkalmazások a négyeres változatban használhatók, jelregenerátorok szintén specifikálhatók mind a kéteres, mind pedig a négyeres működéshez.

Ezek a megoldások a hozzájuk jól illő alkalmazásokhoz a piacon elérhetők. A továbbfejlesztett változatok (például nagyobb sávszélesség, új virtuális nyalábok, Virtual Path) a szolgáltatásintegrációt lehetővé tevő maximális rugalmasságot és növekvő bevételt eredményeznek a szolgáltatóknak. Nagyobb sávszélesség kiosztása egy egyszerű parancsváltással oldható meg anélkül, hogy a központban vagy az előfizető oldalán mindaddig bármilyen eszköz cseréjére szükség lenne, amíg a maximális kiosztható sávszélességet nem érte el.

1.2. A szimmetrikus DSL jelenlegi világpiaci helyzetéről

A DSL piacon számos versenyző van, ezért a CPE-k hardver- és szoftver-verzióinak folyamatos aktualizálása kiemelt jelentőségű. Könnyen előfordulhat, hogy hat hónap leforgása alatt a piacon jelenlévő egyes átviteli eszközök (CPE-k) gyártása leáll, aminek következtében a tervezett valamint a kivitelezett szolgáltatások és protokollok összehasonlító teljesítménymérése és részletes elemzése különösen nagy hangsúlyt kap.

Európában a globális és a regionális szolgáltatók a 2000. év elején kezdték kínálni a hang- és adatforgalmi szolgáltatásaikat szimmetrikus DSL vonalakon, ám mérsékelt növekedés volt csak tapasztalható. 2002 májusában Európában egy szolgáltató csoport az ötezret, egy másik szolgáltató Németországban több mint negyven nagyvárosbeli jelenléttel a hatezret is meghaladó üzleti SDSL előfizetővel rendelkezett. Ez a szám az elmúlt két évben megtöbbszöröződött, de pontos adatokat nem sikerült megtudnunk. Általában igaz a nagy szolgáltatókra, hogy az utóbbi időben sem csupán ADSL-t kínálnak ma a magánfelhasználóknak, hanem SDSL-t is. A teljes képet kiegészítik a regionális szolgáltatók, amelyek némelyike szintén kínál SDSL-t. Mindeközben az erős verseny eredményeként néhány szolgáltató (pl. Riodata és KPNqwest) 2002-ben csődbe ment, előfizetőik nagy hányada más szolgáltatók táborát gazdagítja. Az első kísérleti SDSL hálózatokat az Egyesült Államok szolgáltatóinál és Dél-Koreában 2002 év közepén helyezték üzembe. Ehhez hozzátartozik, hogy az Egyesült Államokban hagyományosan a legtöbb szélessávú előfizető kábel-modemen keresztül csatlakozik a világhálóra, a DSL csak a második helyen áll. A statisztikák szerint a legdinamikusabb fejlődést 2003-ban az ázsiai régió produkálta (Japán, Dél-Korea, Kína). Magyarországon nem jellemző az SHDSL térhódítása, bár apró jelek mutatnak arra, hogy az SHDSL lassan terjed. A Matáv, a Siemens, az Emitel pedig az Ericsson készülékeivel [ERI02] ajánlja mind az ADSL-t, mind pedig az SHDSL-t, de az SHDSL előfizetések száma egyelőre még elhanyagolható az ADSL összeköttetések száma mellett.

A második fejezetben vázoljuk az alkalmazott követelményeket és a mérési módszereket. A harmadik fejezet bemutatja a mérési környezetet. A negyedik fejezet bevezeti az olvasót a protokoll mélységeinek, illetve a hozzátartozó fejléc kiszámításának részleteibe különböző átviteli módokban, egy kis elméleti hátteret szolgáltatva ez által a mérési eredmények értelmezéséhez. Az ötödik fejezet a numerikus eredményeket ismerteti és értelmezi, a hatodik fejezetben, pedig a következtetéseinket vonjuk le.

2. Mérési eljárások és mérőszámok

A CPE-ket úgy választottuk ki, hogy:
- össze tudjuk hasonlítani t0bb gyártó termékeit,
- egynél több termékkel rendelkezzünk mindegyik gyártótól, és
- az SDSL vonalakat meg tudjuk vizsgálni mindkét vonali kódolási eljárás szerint (2B1Q és PAM16).

Nem volt célunk rangsorolni a berendezéseket, de a jellegzetességekről és a teljesítményekről áttekintést nyújtunk. Az eredmények a következők: IP szintű csomagátviteli teljesítmény, egyirányú csomagkésleltetés (késleltetés), csomagvesztés, FTP letöltési sebesség és előfizetői távolság.

Két mérési eljárást különböztettünk meg: egy- és kétirányú mérés fél duplex és duplex módban. Ha csomagok meghibásodtak vagy elvesztek az átvitel során, azokat a beállítások szerint nem küldte újra a forgalomgenerátor (az IP-ben nincs garantált megbízhatósági szint).

Két vonali szimulátor, egy Acterna LS 10.03 és egy Sparnex LSX2020 valósították meg az SDSL/SHDSL előfizetői hurkok távolságparaméterét. A teljesítménymérésnél a kezdeti értékeket a következők szerint állítottuk be: előfizetői hurok hossza = 2,6 km, fehér zaj szintje = 0 dB, kábeltípus = R 0,4 mm PE, SHDSL sebesség = 2312 kb/s, forgalom típus = UBR (Unspecified Bit Rate). Néhány tesztünket alacsonyabb SHDSL sebességen is megismételtük (pl. 1552 kb/s, 1168 kb/s és 768 kb/s), de ezen mérések nem szolgáltak új információval.

Kombinálva az átviteli teljesítmény-, a késleltetés- és a csomagvesztés méréseket a vizsgálati idő CPE-páronként összesen 12-24 órát tett ki. Az átviteli teljesítmény mérése esetén 3 db. 10 másodperces próbát végeztünk minden lépésben a minimális (100 kb/s) és maximális (2320 kb/s) bitsebesség között, 10 kb/s-os lépésekkel. A még hibátlan átvitelt nyújtó maximális sebesség meghatározásához (egy adott csomagméretre) az SHDSL vonalon bináris keresési algoritmust alkalmaztunk. A hibátlan átvitel (100%) alatt azt a maximális IP szintű átviteli sebességet értjük, ahol még nem jelenik meg csomagvesztés.

A szolgáltatók a gyakorlatban igazodnak a megvalósítható értékekhez, így pl. a MATÁV az ADSL és az SHDSL elérés termékopciójában [IPC04] a következő paramétereket kínálja:
- Átlagos csomagvesztési arány: < 5 %,
- Maximális csomagkésleltetés: < 500 ms (a felhasználó végberendezése és a központi telephely CE routere között, egy irányban mérve).

3. Mérőhálózat

A teljesítményt öt CPE-páron hasonlítottuk össze *(1. táblázat első öt oszlopa)*. Eddig nem volt lehetőségünk az SDSL piac másik három vezető gyártójának eszközeit tesztelnünk (Cisco, Alcatel és Netopia), ezért az információink ezen eszközökről a termékleírásokra hagyatkoznak, amelyek a fent említett gyártók hivatalos weboldalain találhatók *(1. táblázat utolsó 3 oszlopa)*, a teljesség kedvéért itt mégis felsoroltuk őket.

Az első lépésben biztosítani kell a kompatibilitást a CPE-k és a DSLAM között (mivel rendszerint különböző gyártók chipkészleteivel vannak felszerelve).

Termékek → / Szolgáltatások ↓	Siemens Attane i210 (IAD)	Siemens Attane i470 (IAD)	Efficient Networks SS5851	Efficient Networks SS5950	Xavi 3102r	Cisco 828	Alcatel SpeedT 610s	Netopia 4553
FW/SW ver.	3.0	2.2	4.0.2	5.3.10	1.56	12.2	3.4	5.3.4
(Mar'02)	(Jan'01)	(Maj'00)	(Okt'01)	(Jul'01)	(Maj'02)	(Maj'02)	(Maj'02)	
Bridge	Igen	Igen	Igen	Igen	Igen	Igen	Igen	Igen
Router	Igen	Nem	Igen	Igen	Igen	Igen	Igen	Igen
PPP	Igen	Nem	Igen	Igen	Igen	Igen	Igen	Igen
LAN10/100b-T	1 port	1port	Hub 4x	Switch 8x	Hub 4x	Hub 4x	Switch 4x	1 port
L2TP	Nem	Nem	Igen	Igen	Nem	Nem	Nem	Nem
IPSec	Nem	Nem	Igen	Igen	Nem	Igen	Igen	Igen
ISDN hang	4x AAL2	(1...8)x AAL1	Nem	Nem	Nem	Nem	Nem	Nem
Vonali kódolás	PAM16	2B1Q/PAM16	2B1Q	PAM16	PAM16	PAM16	PAM16	PAM16

1. táblázat A CPE-k funkcionális áttekintése

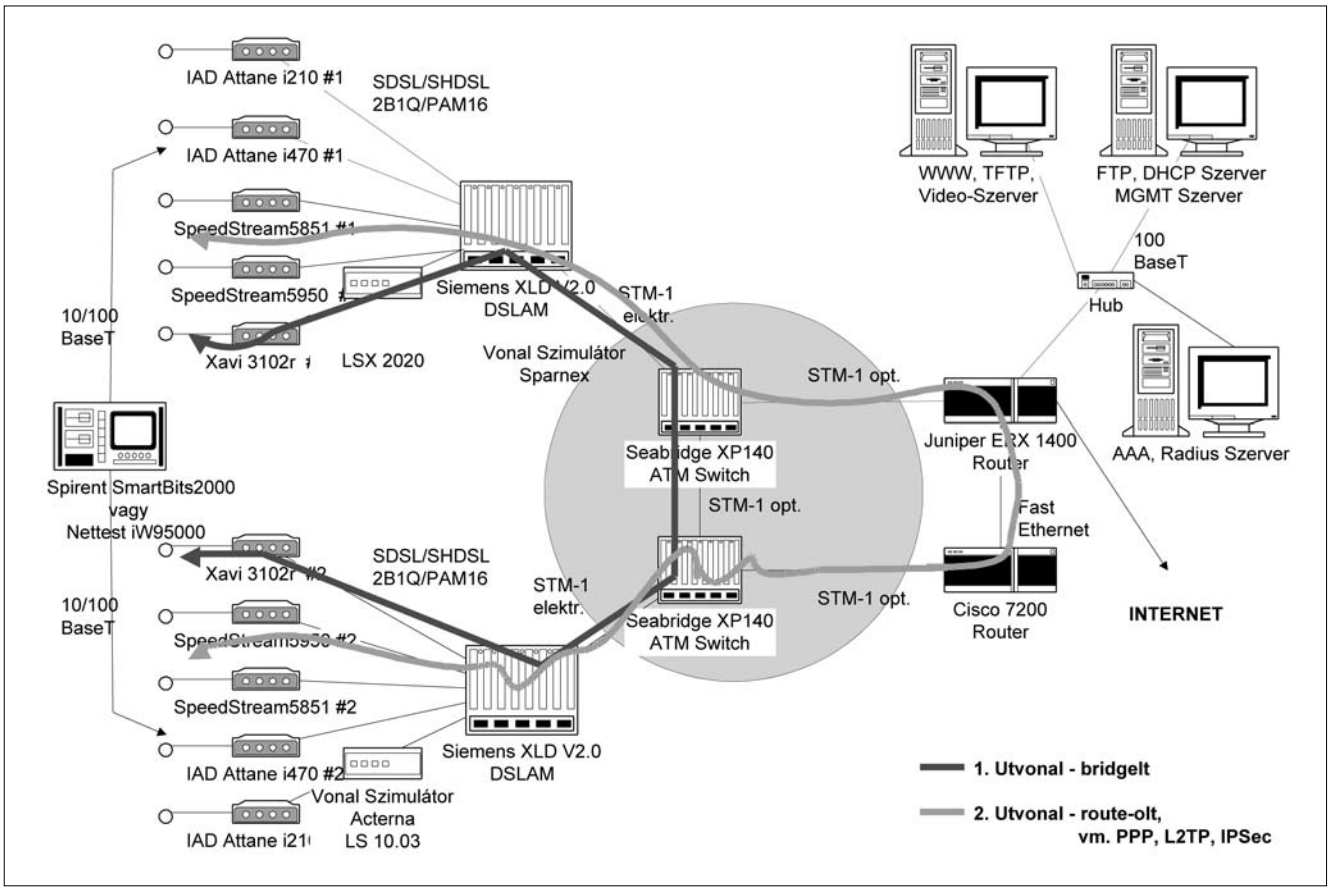
A tesztelést a következő átviteli eszközökkel végeztük: Siemens Attane i470 és Attane i210, Efficient Networks SpeedStream SS5851 és SS5950, és Xavi 3102r.

A táblázatban bemutatott eszközök további szolgáltatásokat is nyújtanak (NAT/NAPT, tűzfal, DHCP szolgáltatás, modemes dial backup, biztonságos menedzsment stb.). Mivel ezek vizsgálata nem volt kitűzött célunk, tárgyalásuk nem szerepel a jelen cikkben. Az 5. fejezetben leírt vizsgálatokhoz a következő hálózati eszközöket használtuk *(1. ábra)*.

A 1. táblázatban bemutatott átviteli eszközök közül kettő támogatja a VoDSL szolgáltatást is (amely üzleti célú hang- és adatszolgáltatást valósít meg DSL vonalon): a Siemens Attane i210 és az Attane i470. Az előfizetői oldalon ezek az integrált elérési eszközök (IAD, Integrated Access Devices) 2,3 Mb/s SDSL vonalon csatlakoznak a DSLAM-hez. Az átviteli protokoll az ATM, a DSLAM pedig összefogja a teljes bejövő hang- és adatforgalmat.

Az aggregát ATM forgalom az ATM hálózaton keresztül jut el az IP gerinchálózatra egy optikai vagy elektromos STM-1 (155 Mb/s) interfészen keresztül. Alternatívaként lehetséges E3 (34 Mb/s) vagy nxE1 (nx2 Mb/s) interfészen is. A VoDSL technológia vizsgálata túlmutat cikkünk keretein, de részben megtalálhatók a [Hab02]-ben.

1. ábra A teszthálózat áttekintése



4. A tesztelés folyamán használt protokoll felépítés (Protocol Stack)

A *2/a. ábra* különböző Protocol Stack-eket hasonlít össze, amelyeknek az eredményei az 1. táblázatban megemlített összeállításokra vonatkoznak. A kapcsolatarchitektúrában megszokott, hogy alul az xDSL alapú ATM foglal helyet. Az IP adatillesztése érdekében az AAL 5 (ATM Adaptation Layer 5) vagy aal5snap-pal vagy pedig aal5mux-szal működik.

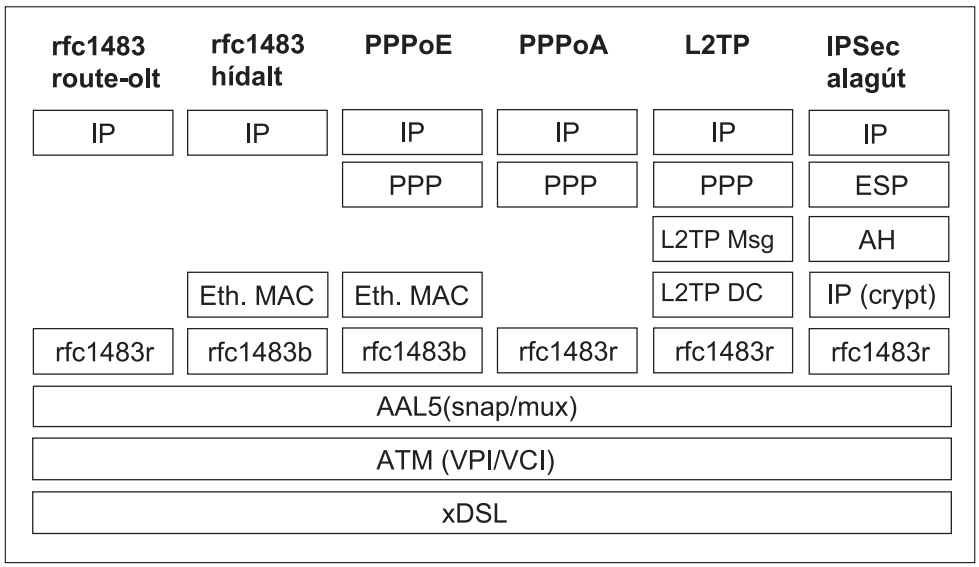
A 2/a. ábra bal oldalán a CPE eszköz szimpla xDSL routerként működik PPP nélkül. Ennek következtében csak egy többlet fejléc kerül be az AAL5 és az IP fejléc közé, nevezetesen az RFC1483. A következőn két fejléc van az AAL 5 és az IP között: bridge-elt RFC 1483, majd az Ethernet fejléc. Nem jeleztük a táblázatban az Ethernet csomagot lezáró mezőt, de természetesen ez is része az Ethernet keretnek. Következik a PPPoE (PPP over Ethernet) összeállítás, ahol pont-pont kapcsolat valósul meg Etherneten. A PPP over ATM (PPPoA) esetben az eszköz routerként funkcionál, így kihagyja az Ethernet keretet az adatfolyamból. Éppen a PPP hordozza az IP keretet, ebben különbözik a bal szélső stack-től.

Folytatva, az L2TP (Layer 2 Tunneling Protocol) elrendezésben az eszköz LAC-ként (L2TP Access Concentrator) funkcionál, pl. a PPP összeköttetés után egy szélessávú távoli elérési szerver (BRAS) következik egy layer 2 csatornán keresztül. Így a PPP fejlécet megelőzi még az L2TP adatcsatorna (DC, Data Channel) fejléce és az L2TP üzenetek (Mgs, Messages) is. Végül az IPSec (IP Security) hierarchiában az egész IP csomag kódolt, ahol az eredeti IP fejléc láthatatlan. Ennek helyére egy új IP fejléc kerül a titkosított csatorna végpontjai szerinti forrás és cél IP-címekkel. Amióta az eredeti IP fejléc rejtett, a titkosító kódolás miatt (ESP, Encapsulating Security Payload) a régi hálózati címek a későbbiekben sem láthatók. Ez egy fontos biztonsági funkciója ennek az átviteli módnak.

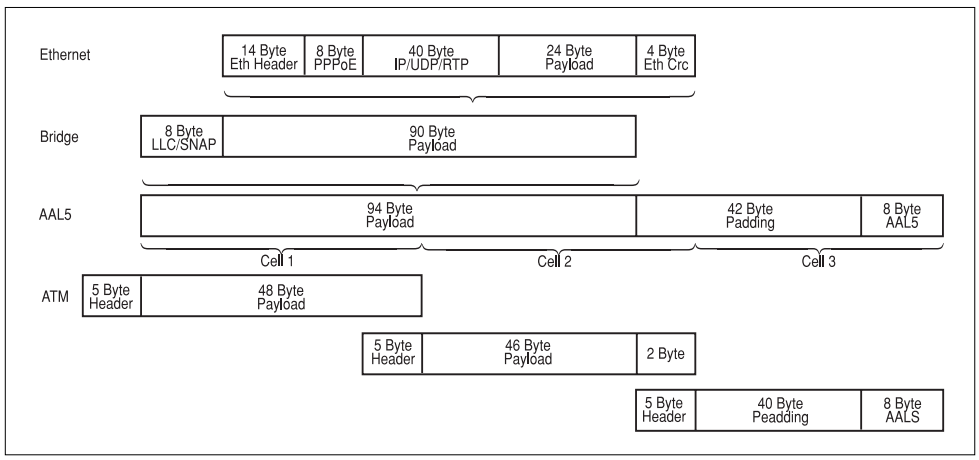
Ezzel szemben áll az IP-mód, ahol az átviteli forma IPSec (nincs a 2/a. ábrán), a titkosító kódolás csak az IP csomag tartalmát érinti, a fejlécet nem. A régi IP fejlécet megőrzi, azonban a protokoll mező már nem jelzi, például a TCP=6/UDP=17 stb. paramétereket, de az AH=51/ESP=50-et igen. Kizárólag hitelesítésre tulajdonképpen az AH (Authentication Header) egyedül is alkalmas lenne, illetve az ESP kizárólag a titkosító kódolásra, de szokás szerint az AH-t vagy az AH/ESP-t alkalmazzuk. A biztonsági paramétereket (SPI, security parameters index) lényegében az AH és az ESP alkotják. Mindez egy olyan táblázatra mutat, amely az elengedhetetlen paramétereket (algoritmus, titkosító kulcsok) tartalmazza. Ezen táblázat értékeit vagy a két végrendszerben, vagy pedig automatikusan egy kulcscsere algoritmussal (key exchange protocol, pl. IKE) határozzák meg, amelyről részleteket a [Bor00]-ban találunk.

A *2/b. ábra* szemléletesen mutatja, hogy egy 24 bájtos rövid csomag (pl. egy RTP csomag) 90 bájtra duzzad az Ethernet keretben, majd további 42 bájt kiegészítés szükséges az AAL5-ös keretben, illetve 15 bájt ATM fejléc is csatlakozik hozzá az ATM rétegben. Ez a „látványos” rossz kihasználtság csak a nagyon rövid csomagméretekre jellemző.

2/a. ábra A CPE-kben használt különböző protokoll struktúrák



2/b. ábra Egy példa a fejléc kialakítására



5. Mérési eredmények

Ebben a fejezetben az SDSL/SHDSL átviteli eszközök teljesítményét hasonlítottuk össze, PPP és L2TP valamint IPSec módokban. Mindig két meghatározott eszköz között végponttól végpontig mérjük az IP-szintű átvitelt, késleltetést és csomagvesztést. Az angol terminológia kétféle kifejezést használ az IP szintű késleltetésre: delay, latency; mi a két fogalmat egyenértékűen késleltetésnek értelmezzük. Amennyiben szükséges, rámutatunk a teljesítménybeli különbségekre a fél duplex (HD, Half Duplex) és a duplex (FD, Full Duplex) módok között. A teszteket minden módban FTP letöltési teljesítménnyel (lásd. 5.5 teszteset) tettük teljessé. Az eredményeket hat csoportba osztottuk, az utolsó esetben a maximális elérhető bitsebességre állítottuk be.

5.1. Vizsgálat:
Bridge-elt üzemmód: HD/FD, kétirányú forgalom

(1. útvonal az 1. ábrán és 2. oszlop a 2/a. ábrán)

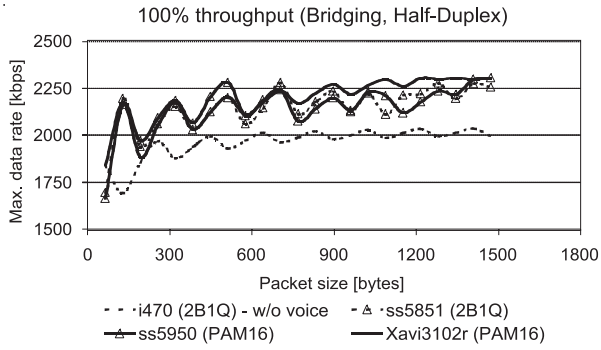
Bridge-elt duplex módban, kétirányú forgalom mellett az adatforgalom mind az öt átviteli eszközön megközelíti a lehetséges maximális fizikai sebességet *(3. ábra)*.

Minden sikeresen átvitt csomagra (a hozzá tartozó csomagméret függvényében) az ábra egy pontja jelzi az elért adatsebességet. Kis csomagoknál az átvitel mértéke (throughput) kissé alacsonyabb (pl. a 64 byte-os csomagoknál 1,7-1,8 Mb/s), de a csomagméretek növekedésével ez az érték konvergál a 2,3 Mb/s-hoz, miközben hullámzást mutat az AAL 5 keret kitöltése (padding) miatt. Amint elérte a szabványos Ethernet keret hosszát (1522 byte-os tartalom), az átvitel nullára esett vissza.

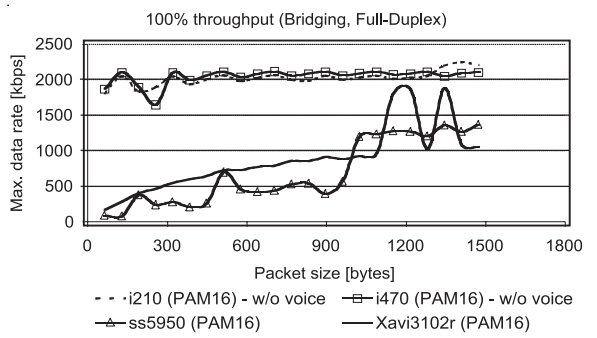
Mivel a 2B1Q vonali kódolás kevésbé hatékony, mint a 16 szintű trellis kódolt TC-PAM, nyilvánvaló, hogy kedvezőbb az átviteli jelleggörbe, ha az eszközök PAM 16 kódolást használnak *(3/a. ábra)*. Mindemellett a különbség így sem haladja meg a vonalsebesség 10%-át, és az eredmények mindkét irányban hasonlóak (upstream és downstream). Nem tapasztaltunk jelentős eltérést különböző szoftverek miatt sem, és az sem jelentett különbséget, ha 10 vagy 100 Mb/s-os kártyával csatlakoztunk a hálózati (LAN) interfészhez. A többi SHDSL átviteli eszközt megmértük mind duplex, mind fél duplex módban, de az áttekinthetőség érdekében csak négyet ábrázoltunk a 3. ábrán.

A *3/b. ábra* szerint, a 100%-os átviteli görbe bridge-elt duplex módban jelentős különbségeket mutat a négy vizsgált átviteli eszköz között. Mivel az Attane i210 és az Attane i470 csak egy-egy Ethernet port csatlakozóval rendelkezik a LAN (Local Area Network) interfészen, bridge-elt módban sorba tudják rendezni a duplex szolgáltatást. Átviteli karakterisztikájuk megközelítően olyan, mint a fél duplex módban (lásd. 3/a. és 3/b. ábrák). Másrészről a Xavi3102r hálózati interfésze rendelkezik egy 4 portos hub-bal és az SS5950-hez tartozik egy 8 portos Ethernet switch. Ez duplex módban csomagütközést okozhat.

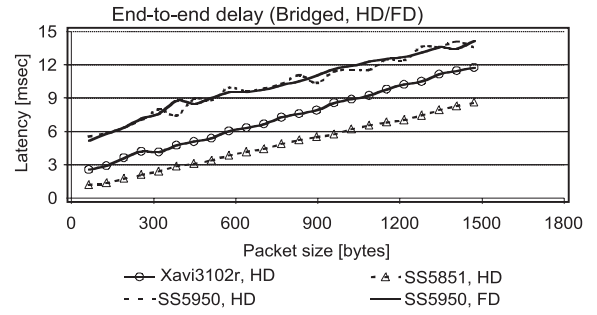
A maximum átviteli sebesség 0%-os csomagvesztés mellett kevesebb, mint 500 kb/s (a két utóbbi eszközön), amikor a csomagméretek nem érik el az 500 byte-ot. Mindemellett ha megengedünk egy kis mértékű csomagvesztést, jobb átviteli karakterisztikát érhetünk el. 10-20% csomagvesztés mellett, kis csomagokra (64-től 192 byte-ig) 2 Mb/s sebesség is elérhető *(3/d. ábra)*. Hosszabb csomagoknál általában jobb átviteli sebes-



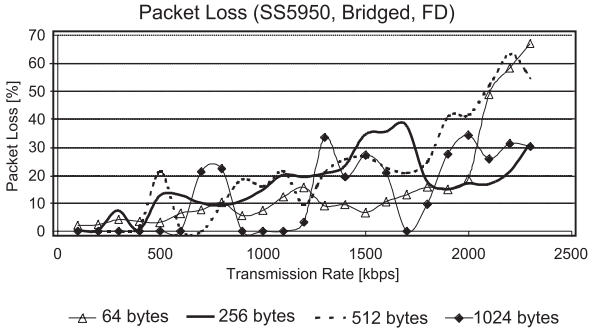
3/a. ábra Fél duplex átvitel



3/b. ábra Duplex átvitel



3/c. ábra Késleltetés



3/d. ábra Csomagvesztés bridgelt módban

ség is megvalósulhat, de átlagosan az átviteli sebesség nem lesz több, mint a fizikai sebesség 50%-a. A nagyobb átviteli sebesség érdekében számolnunk kell a növekvő csomagvesztéssel, pl. 1500 kb/s-nál ez az arány 30%, 2000 kb/s-nál a 40%-ot is meghaladja.

A full-duplex bridge-elt módban fellépő ütközés okozza a csomagvesztés 10-40%-os arányát. Szeretnénk hangsúlyozni, hogy ez a viselkedés nem az Efficient Network SS5950 vagy a Xavi 3102r eszközök hibája, hanem nagy forgalmi terhelés mellett normális jellemzője minden hub-nak vagy switch-nek, amikor az IP fölött már nincs megbízható protokoll. Megbízható protokollok (úgy, mint FTP, vagy TCP/IP) elérik a közel maximális sebességet (hozzávetőlegesen 1900 kb/s), ahogy azt az 5.5.-ös eset mutatja.

A végpontok között (egy irányban) mért csomagkésleltetés 2-5 msec-tól (64 byte-os csomagok küldése) 10-14 msec-ig (1500 byte-os csomagméret) lineárisan növekszik. A csomagok méretével arányos késleltetés a nagyobb csomagok összerakásához szükséges idő növekedése miatt lép fel *(3/c. ábra)*. Különböző bitsebességek (0,1 Mb/s a maximális átvitelig), de állandó méretű csomagok esetén csomagvesztés nélkül a végpontok közötti késleltetésben nem tapasztalható jelentős eltérés. A 3/c. ábrán bemutatott értékek az átlagos késleltetést mutatják a hozzá tartozó csomagméret szerint 0,1 és 1,7 Mb/s között. Nagyobb bitsebességeknél 1,7 és 2,3 Mb/s között a rövid csomagok 0,1-10%-os arányban vesznek el. Az Ethernet interfész jellemzői miatt (MTU méret = 1500 byte) az 1522 byte-nál (payload) nagyobb csomagok mindegyike elvész (pl. 1536 byte). Méréseinkkel összhangban a legjobb teljesítményt (pl. a legkisebb késleltetést) az SS5851 éri el (1,5 msec 64 byte-os csomagoknál), amelyet a Xavi3102r (2,5 msec) követ, míg az SS5950 a 64 byte-os csomagokat 5 msec alatt dolgozza föl és továbbítja.

Az SS5950 esetén az egy irányban mért átlagos csomagkésleltetés duplex módban hasonlóan viselkedik, mint fél duplex módban, azaz a késleltetés szinte lineárisan növekszik 5 msec-ről (64 byte) 14 msec-re (1500 byte). Természetesen FD módban csak azokat az értékeket vesszük figyelembe, ahol 100%-os az átviteli sebesség. Az SS5851 és a Xavi3102r hasonló késleltetési eredményeket produkál úgy full-duplex, mint fél duplex módban, és ezért ezt nem tüntettük fel a 3/c. ábrán.

5.2. Vizsgálat:
Route-olt üzemmód: HD/FD, kétirányú forgalom

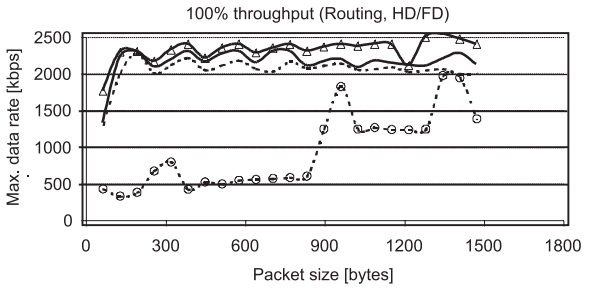
(1. ábra 2. útvonal és 1. oszlop a 2/a. ábrán)

A *4/a. ábra* mutatja az átviteli eredményeket, mind fél duplex, mind pedig duplex módban két CPE-re, nevezetesen az SS5950-re és a Xavi3102r-re. A görbék hullámzását az AAL5 keret kitöltő byte-jai (padding) okozzák. Az átviteli sebesség több mint 2 Mb/s, kivéve a 64 byte hosszú csomagokat az SS5950 eszköz esetében.

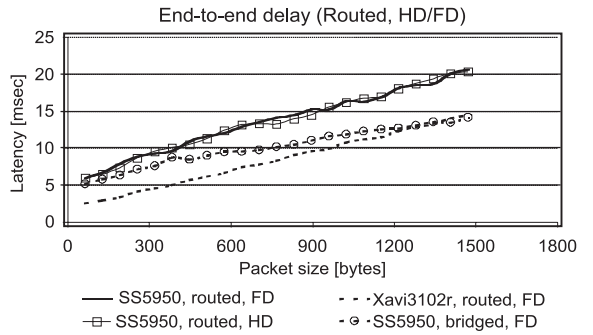
Érdekességként megemlíthetjük, hogy a Xavi3102r fél duplex módban 1300 byte-os csomagméretnél nagyobb sebességet ér el, mint a beállított fizikai bitsebesség (pld. 2,52 Mb/s (!), míg a jelzett fizikai sebesség 2,32 Mb/s). A 4.a ábrán látható Xavi3102r átvitele route-olt full-duplex módban hasonló karakterisztikát mutat, mint bridge-elt full-duplex módban *(3/b. ábra)*, így jóval elvárásaink alatt marad.

Elemezve az egyirányú késleltetést a Xavi3102r route-olt módjában, HD/FD esetekben (2,5-től 14,5 msec-ig növekedve), az eredmények 0,54 msec-ot lépnek 64 byte-onként (lásd. 4.b ábra). Felhívjuk a figyelmet a különbségekre a bridge-elt módhoz képest (ahol 2,5-től 12 msec-ig növekszik). Route-olva - összehasonlítva a bridge-elt móddal - a hosszabb csomagok átviteli ideje (20%-kal) hosszabb.

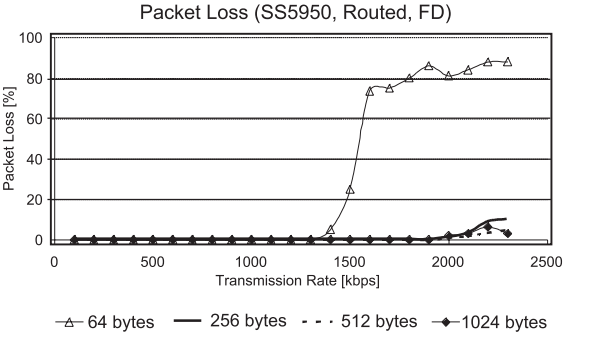
Hasonlóan érvényes ez a feltétel SS5950 esetén is; a bridge-elt (5msec; 14msec) intervallum route-olva eltolódik a (6msec; 20msec) időintervallumra, miközben a csomagméretek 64 byte-ról 1500 byte-ra növekednek. Továbbá nincs jelentős különbség az átlagos késleltetésben, ha a 10 Mb/s-os hálózati interfészt (LAN)



4/a. ábra Átviteli sebesség



4/b. ábra Egyirányú késleltetés



4/c. ábra Csomagvesztés routeolt módban

felcseréljük 100 Mb/s-osra (kicsit kisebb késleltetés érhető el). Hasonlóan az átviteli sebességben sincs számottevő eltérés az Ethernet interfész 10-ről 100 Mb/s-osra cserélésekor. A *4/c. ábra* a csomagvesztési jelleget mutatja a SS5950 eszköz full-duplex route-olt módjában. 64 byte-os csomagok esetén 1500 kb/s bitsebességtől kezdődően 80% fölötti csomagvesztés figyelhető meg, ugyanakkor a 256 byte méretű csomagok veszteség nélkül átvihetők egészen 1900 kb/s bitsebességig. Ennél hosszabb csomagok legfeljebb 10%-os csomagveszteséget szenvednek 2000 kb/s sebesség fölött.

5.3. Pont-pont összeköttetés: PPP ATM felett, HD/FD

(2. útvonal az 1. ábrán és 4. oszlop a 2/a. ábrán)

A vizsgált átviteli eszközök egyike sem alkalmas az Ethernetes pont-pont összeköttetésre (PPP over Ethernet) az 1. táblázatban leírt szoftververzióikkal, de mindegyiken megvalósítható az ATM-es pont-pont összeköttetés (PPPoA). A PPPoE kapcsolat lérehozása érdekében egy szoftveres alkalmazásnak jelen kell lennie az előfizetői PC-n, és az SHDSL eszköznek bridge-elt módban kell futnia. Mivel műszereink sem illeszkednek a PPPoE elrendezéshez, PPPoE módban nem tudtunk teljesítménytesztet végrehajtani. Más esetben (PPPoA) az eszközök routerként viselkedve eliminálják az Ethernet kereteket az adatfolyamból.

Amint az ATM virtuális kapcsolat létrejön az átviteli eszköz WAN (Wide Area Network) interfésze és a gerinchálózati router között, a pont-pont összeköttetést (lásd. 1. ábra, 2. útvonal) egy autentikáló eljárás követi a modem és az AAA (Authentication, Authorization and Accounting) szerver között; esetünkben egy PAP (Password Authentication Protocol) típusú autentikáció zajlott le. A modem ezután IP címet kap a Radius szervertől, amelyet követően elkezdődhet a mérés.

Az *5/a. ábra* mutatja, hogy a PPP fél duplex módban a hibamentes átviteli sebesség nagyságrendileg azonos a roulte-olt (PPP nélküli) konfigurációban elért eredményekkel. Ellenben – PPP full-duplex módban - a PPP összeköttetés átviteli képessége drámaian visszaesik 1300 kb/s sebesség alá, ráadásul a 64 byte-os csomagokra nézve az átviteli görbe (melyet 0% csomagvesztés mellett értelmezünk) nullát mutat, amelynek oka egy 3%-os csomagvesztési arány *(5/c. ábra)*. Általában véve sokkal magasabb csomagvesztés figyelhető meg PPP FD esetben, mint route-olt FD esetben (összehasonlítható a *4/c. és az 5/c. ábrák)*. Másrészről, ha a késleltetési eredményeket szemléljük nagyban hasonlítanak a route-olt mód késleltetéseihez *(5/b. ábra)*.

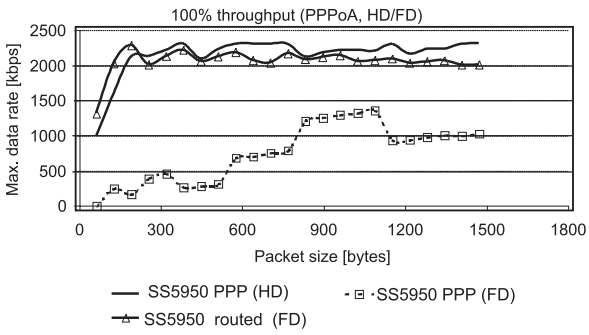
5.4. Vizsgálat: L2TP + IPSecurity: HD/FD, kétirányú forgalom

(1. ábra 2. útvonala és az 5., illetve a 6. oszlop a 2/a. ábrán)

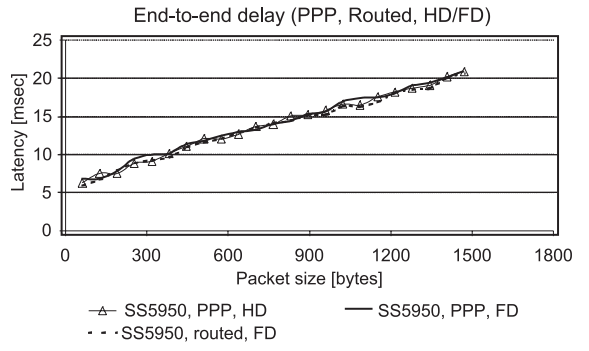
Csak egyetlen olyan átviteli eszközt találtunk, amely mind az L2TP és az IPSec protokollt támogatta, ez pedig az Efficient Networks SS5950 eszköze (lásd. 1. táblázat). Amint a két egyenrangú SS5950-es router Internet-kapcsolata létrejön, felépül egy ú.n. alagút, amely lehetővé teszi a biztonságos internetes kapcsolatot, esetünkben az ERX1400 és a Cisco 7200 routereken keresztül, lásd. 1. ábra). A *6. ábra* a teljesítménymérések eredményeit mutatja titkosító kódolással és anélkül. A tesztek megindítása előtt a pontos eredmények érdekében szükséges volt előbb megnyitni az alagutat (ping csomagok küldésével a két eszköz között).

Az általunk mért átviteli görbének L2TP esetben két része van: egy lineárisan növekvő szakasz 400 kb/s-tól (64 byte hosszú csomagméret esetén) 2300 kb/s sebességig (itt 900 byte hosszúak a csomagok), valamint egy vízszintes szakasz a továbbiakban 1522 byte-os csomagméretig. L2TP módusú titkosító kódolás esetén az átvitelnek csupán csekély mértékű teljesítménycsökkenését észleltük *(6/a. ábra)*.

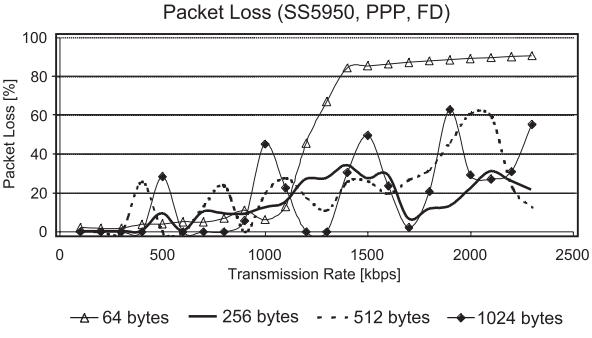
Viszonyítási alapként a route-olt átviteli mód sebességgörbéjét is ábrázoltuk a 6/a. ábrán. Azonnal kiderül,



5/a. ábra Átviteli sebesség



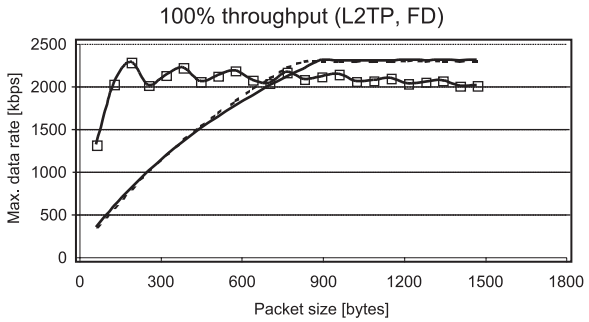
5/b. ábra Késleltetés



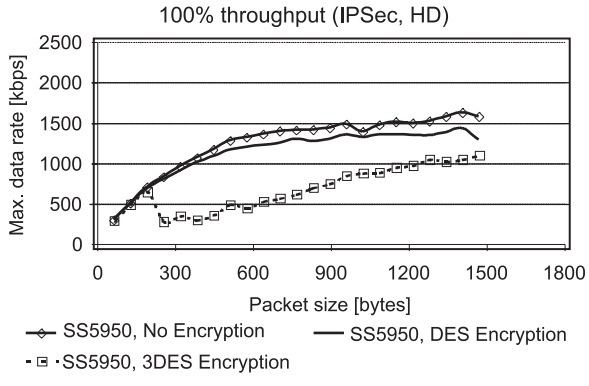
5/c. ábra Csomagvesztés PPP módban

hogy ezzel az eredménnyel nem lehetünk elégedettek, hiszen az L2TP beállítás által megnövekedett fejléc nem annyira számottevő, hogy ezt az eltérést indokolná. És miért éppen lineáris az első szakasz? 900 bytenál hosszabb csomagokra viszont miért jobb az átvitel és mitől kisebb a késleltetés L2TP módban, mint routeolt módban *(6/c. ábra)*?

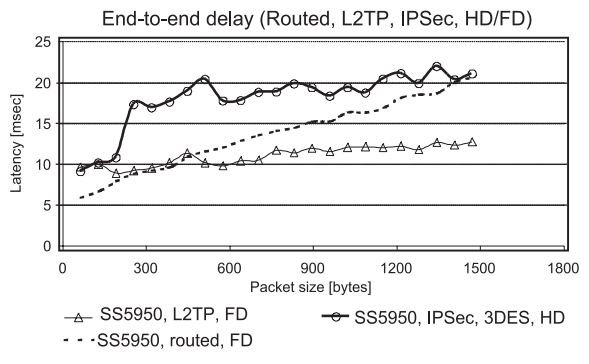
A késleltetés csökkenését elsősorban arra vezethetjük vissza, hogy a csomagok nem igényelnek routeolást a gerinchálózatban, mivel az L2TP csatorna már létrejött a két eszköz között. De az átviteli görbe alakjára csak egyetlen magyarázat létezik, a hozzátartozó hardver illetve a L2TP-t kiszolgáló szoftver nem eléggé gyors a CPE jelen verziójában. Erre azonban egyértelmű választ csak akkor kapunk, ha majd a jövőben sikerül egy újabb verziót is letesztelni.



6/a. ábra Átviteli sebesség



6/b. ábra Átviteli sebesség



6/c. ábra Késleltetés L2TP és IPSec módokban

A *6/c. ábra* szerint az átlagos késleltetés L2TP FD módban lineárisan növekszik 9 msec-ról (64 byte hosszú csomagok) 13 msec-ra (1500 byte hosszú csomagok). Összehasonlítva a full-duplex route-olt módot az itteni L2TP móddal, rövid csomagokra nagyobb, hosszabb csomagokra pedig kisebb a késleltetés. Nem tapasztalunk az átlagos késleltetésben szignifikáns különbséget, ha a 10 Mb/s-os hálózati interfészt (LAN) felcseréljük 100 Mb/s-osra.

Az IPSec a biztonságos internetes alkalmazások elegáns megoldásaként ismert, bővebben [Bor00] és [Kar01]. Mindemellett, ha egy pillantást vetünk az adatátviteli teljesítmények eredményeire *(6/b. ábra)*, láthatjuk, hogy az eredmények itt sem túl bíztatóak (hasonlóan az L2TP megoldáshoz), holott a fejlécnövekedés ennél a megoldásnál sem indokolja az eredményt. Az általunk elért legnagyobb adatátvitel csupán 1500 kb/s, ami a 2320 kb/s-os fizikai átviteli sávszélességnek mindössze 66%-a. A csomagok DES (Data Encryption Standard) vagy tripla DES titkosító kódolásával az átvitel még ennél is rosszabb. Például 3DES kódolás esetén a legnagyobb átviteli sebesség 1500 byte hosszú csomagokra 1100 kb/s. Tetszőleges csomagméretre az egyirányú késleltetés IPSec módban közel kétszer olyan nagy, mint L2TP módban (hozzávetőlegesen 20ms). A tényleges válasz itt is a hardver illetve a hozzátartozó szoftverben keresendő.

5.5. Vizsgálat: Hosszú FTP letöltések

Teljesítményvizsgálatunk következő állomásaként egy megbízható adatátviteli kapcsolatot hoztunk létre egy CPE-páros között, azaz FTP (File Transfer Protocol) fájl-forgalommal egészítettük ki a nem megbízható átvitelű IP-forgalomgenerátorral megvalósított adatátviteli eredményeinket. A forgalomgenerátorok helyett nevezetesen két FTP szervert csatlakoztattunk az SS5950 modemek LAN interfészeire, és mindkét irányban fájlküldést kezdeményeztünk. Három különböző méretű állomány letöltését hajtottuk végre: 11, 34, illetve 83 Mbyte. Az eredményül kapott átlagos letöltési bitsebességeket a *2. táblázat* tartalmazza.

Az FTP letöltések időtartamai a bridge-elt, route-olt, és a PPP módokban ígéretesnek bizonyultak, míg az L2TP és az IPSec módokban inkább meglepetésszerű eredményeket kaptunk. Az eredmények nem annyira meglepőek, ha az 5.4.-es fejezetet elolvassuk. Kézenfekvő magyarázat lehetne a nagyon rövid nyugták jelenléte a letöltés során, ebben az esetben a *6/a. és 6/b. ábrák* szerinti alacsony átviteli jelleggel függ össze. Feltehetőleg a DSP processzor teljesítményének megnövelésével ez a probléma kiküszöbölhető.

5.6. Vizsgálat: Előfizetői hurok hossza és a bitsebesség

Annak eldöntése végett, hogy megállapítsuk, hogy az előfizetői hurkok szabványosak-e vagy sem, mind az öt eszközzel teszteltük az előfizetői hurok sávszélességét. Az eredményeket a *3. táblázatban* foglaltuk össze.

Modem	Bridge-elt HD [kb/s]	Bridge-elt FD [kb/s]	Route-olt [kb/s]	PPPoA [kb/s]	L2TP [kb/s]	IPSec [kb/s]
SS5950	1918	1828	1896	1920	900	800

2. táblázat FTP letöltések átlagos sebességei

Bitsebesség [kb/s]	ETSI szabvány előírás	Efficient SS5851 (SDSL) Fix	Attane i470 (SDSL) Fix	Attane i210 (SHDSL) Adaptív	Attane i470 (SHDSL) Adaptív	Efficient SS5950 (SHDSL) Adaptív	Xavi 3102r (SHDSL) Adaptív
2304	2.23	3.0	3.2	3.4	3.2	2.8	3.3
2048	2.45	3.3	3.3	3.6	3.4	3.0	3.55
1536	3.12	3.7	3.6	4.0	3.75	3.4	3.95
1168	--	3.75	3.7	4.45	4.15	3.7	4.4
768	4.33	3.8	3.8	5.1	nem szink.	4.2	5.0
584	--	3.8	4.0	5.4	nem szink.	4.7	5.4
384	5.70	4.0	4.1	5.9	nem szink.	5.4	5.9

3. táblázat
Fehérzajmentes előfizetői hurok teszt (értékek: határtávolság [km])

Beállításaink a következők voltak: teszthurok az ETSI ETR 152 (SDSL-hez) és ITU-T Q.991.2 (SHDSL-hez), R 0.4mm PE (Polyetilén) kábel, illesztés nélkül (1 hurok), DSLAM/CPE jelteljesítmény=13,5 dBm (szabványos) vagy változó 8,5-14,5 dBm 2304 kb/s bitsebességnél nagyobb esetben, amikor a hurok 1,1 km-nél rövidebb, a megkívánt tartalék = 3 dB a DSLAM-ben, 0dB a CPE-ben, fehér zaj nélkül, fix vagy adaptív CPE módban. További információk lásd a [Dac01]-ben.

A következőket figyeltük meg: 768 kb/s sebesség alatt az Attane i470 és az SS5950 nem elégíti ki az ETSI szabvány követelményeit. Ezen a sebességen az Attane i470 nem is szinkronizál. Feltételezéseink szerint az eszközt legalább 660 kb/s konstans bitsebességre tervezték, hogy az adatforgalom mellett még kiszolgálja az AAL1 alapú 8 hangcsatornát. Mindezen túl az eszközök egyike sem volt képes szinkronizálni adaptív módban 192 kb/s-on 5 km-nél hosszabb előfizetői huroknál.

Példaként vettük az Attane i470 eszközt, amely felajánlja úgy a 2B1Q, mint a PAM16 vonali kódolást egy szimpla váltással a konfigurációs menüben, majd az elért bitsebességek tekintetében összehasonlítottuk a korábbi SDSL-t az SHDSL-lel. Az eredményt a *3. táblázat* mutatja, miszerint 35-45% távolságbeli növekedés tapasztalható egy adott bitsebesség mellett (pl. 1168 kb/s). Mindemellett a legnagyobb vonali sebességre (2320 kb/s) a hurok hossza mindkét esetben 3,2 km-nek adódott. Általában az SHDSL sokkal hatékonyabb és jobb spektrum kompatibilitást mutat, mint azon rendszerek, amelyek a régebbi szimmetrikus technológiát - beleértve HDSL-t és SDSL-t - 2B1Q vonali kódolással valósítják meg.

Következtetés

A cikk egy olyan vizsgálatsorozatot mutat be, amellyel elemezhető az adatforgalom szimmetrikus DSL hálózatokban, miáltal lehetőségünk van egy általános összehasonlításra különböző gyártók berendezései között. Ily módon számos különbséget mutattunk ki a teljesítménymérések során. Részletesen vizsgáltunk bridge-elt, route-olt, PPP, L2TP és IPSec kapcsolatokat, különös tekintettel a hibamentes átvitelt biztosító maximális sebességre, csomagkésleltetésre és csomagvesztésre. A biztonságos adatátvitelhez (L2TP, IPSec) kapcsolódó vizsgálatunk során kifejtettük, hogy szükség van még ezen megvalósítások feljavítására gyorsabb DSP processzorok alkalmazásával. A szerzők reményüket fejezik ki, hogy jelen cikkükkel hasznos információt tudnak nyújtani az SDSL technológiát bevezetni óhajtó szolgáltatóknak.

Irodalom

[Bal98] Balogh Tamás, A dgitális előfizetői vonalak evolúciója, Magyar Távközlés, 1998/4, pp.23–28.

[Bre02] S. Bregni, R. Melen, Local Loop Unbundling in the Italian Network, IEEE Communications Magazine, Vol.40, no.10, October 2002, pp.86–93.

[Bor00] M.S. Borella, Methods and Protocols for Secure Key Negotiation Using IKE, IEEE Network, Vol.14, July/Aug. 2000, pp.18–27

[Dac01] D. Daecke, Overview of SHDSL System Performance, ETSBC 2001, pp.2–6.

[ERI02] Az Emitel az Ericssont kérte fel ADSL szélessávú hozzáférési berendezések telepítésére www.ericsson.com/hu/press/eripress_2002_10-25_511.shtml

[Hab02] A. Habib, H. Saiedian, Channelized Voice over Digital Subscriber Line, IEEE Communications Magazine, Vol.40, no.10, October 2002, pp.94–100.

[Hum97] M. Humphrey, J. Freeman, How xDSL Supports Broadband Services to the Home, IEEE Network, Vol.11, January/February 1997, pp.14–23.

[IPC04] A Magyar Távközlési Rt. általános szerződési feltételei IP Complex Plusz szolgáltatásra, 2004.04.01., www.matav.hu/magyar/mtav/aszf_mod/ipcomplexplusz_torzs.pdf

[ITU01] ITU-T, Single-pair High-bit-rate DSL, G.991.2, Geneva, February 2001

[Kar01] A. Kara, Secure Remote Access from Office to Home, Communications Magazine, October 2001, pp.68–72.

[TSAG01] Telecommunication Standards Advisory Group, Report on Testing of Broadband Interface in Local Access Link – Short-Loop Condition, TSAC Paper No.24, September 2001, pp.1–12.

A média-konvergencia előrehaladása és hatása a médiaszabályozásra

A nyár derekán két tudományos kutatóműhely, az MTA Jogtudományi Intézete Infokommunikációs Jogi Centrumának és a győri Széchenyi István Egyetem Jog- és Gazdaságtudományi Karának közös szervezésében az infokommunikáció, és azon belül a média-szabályozás egyik legaktuálisabb kérdéséről, a konvergencia jelenségének szabályozásáról rendeztek konferenciát júliusban Győrött.

A Magyar Tudományos Akadémia Jogtudományi Intézete 2003-ban alapította az Infokommunikációs Jogi Centrumot (a továbbiakban: IJC), amelynek elsődleges feladata, hogy tudományos irányultságú, független szakmai műhelyként az infokommunikációs jogok területén hiteles elemzéseket, tanulmányokat, szakértői anyagokat készítve járuljon hozzá az információs társadalom fejlődéséhez.

A Széchenyi István Egyetem Jog- és Gazdaságtudományi karának Állam- és Jogtudományi Intézete szintén kiemelt feladatának tartja az információs társadalom és a jogrendszer viszonyának a vizsgálatát, amelyet többek közt egy, az infokommunikációs jogokat előtérbe helyező nyolc szemeszteres képzési szakirány és egy médiajogi kutatóműhely kialakításával is elősegíti.

A konferencia célja az volt, hogy a témakör tudományos igényű bemutatása mellett elősegítse a tudományos-szakmai párbeszéd kialakulását a jogalkotó szervek, a tudományos elemzők és a jogalkalmazók között. Emellett lehetőséget kívánt adni arra is, hogy a konferencia résztvevői a jogterület legnevesebb szakértőitől naprakész információkat kapjanak az ezen a téren tapasztalható jogfejlődés helyzetéről. E szempontok érvényesítése érdekében a konferencia szervezői úgy állították össze az előadók névsorát, hogy azok az Informatikai és Hírközlési Minisztérium, a Nemzeti Kulturális Örökség Minisztériuma, az Országos Rádió- és Televízió Testület, a Nemzeti Hírközlési Hatóság, a tudományos kutatóintézetek és a piaci szereplők oldaláról is azonosítsák a médiakonvergenciával kapcsolatban felmerülő problémákat, és egyúttal együttesen keressenek azokra válaszokat. A konferencia mindemellett a két említett kutatási intézmény, valamint a Budapesti Műszaki és Gazdaságtudományi Egyetem (BME) együttműködésében a jövőben megvalósuló átfogó szabályozói konvergencia kutatási program problémafeltáró rendezvényének is tekinthető, melyben a kutatók azokat a témaköröket azonosították, amelyekben a későbbiekben kiterjedt vizsgálatokra lesz szükség.

A konferenciát *Kovács György*, az ORTT elnöke nyitotta meg. A konferencia délelőtti szekciója a konvergenciának a hálózatokra gyakorolt hatását vizsgálta.

Sallai Gyula, a BME Távközlési és Médiainformatikai Tanszékének vezetője a digitalizáció hatásainak elemzéséből kiindulva adott bevezetést a témához. Előadásában az infokommunikációs konvergencia vizsgálati modelljét és a konvergencia tizenkét típusát ismertette. Hangsúlyozta a trendtanulmányok fontosságát, majd kifejtette, hogy az infokommunikáció elterjedtségének és a konvergencia hatásainak mérésére használt makromutatók közül jelenleg az ITU digitális hozzáférés indexe (DAI), azon belül a szélessávú Internet-penetráció a legalkalmasabb.

Koppányi Szabolcs, az IJC vezetője az „A szabályozói konvergencia helyzete és lehetséges irányai ma Magyarországon" című előadásában a szabályozói konvergencia Európai Uniós helyzetét mutatta be. Kiemelte azokat a problémaköröket, amelyek Magyarország esetében is felmerülnek a témával kapcsolatban. Utalt a kétharmados fogságban szenvedő médiatörvény körüli helyzetre, a szabályozói szervezetben felmerülő problémákra és választ keresett arra is, hogy milyen határok között lehet szükséges a szektor-specifikus szabályozás integrációja. Előadása végén a konvergens szabályozás általa feltárt előnyeit és hátrányait mutatta be.

Révész T. Mihály, Kovács György és Lamm Vanda


Nyakas Levente, Sallai Gyula és Gyenge Anikó

Rozgonyi Krisztina, a Nemzeti Hírközlési Hatóság Tanácsának tagja a konvergencia egy gyakorlati példájáról, a digitális televíziózásról tartott előadást. Általánosságban a konvergencia lehetséges dimenzióiból kiindulva vizsgálta a digitális televízió szerepét a konvergencia-folyamatban. Részletesen elemezte a területen szóba jöhető állami szerepvállalás eszközeit (a policy-making-től egészen a tulajdonosi szerepvállalásig), majd a szerepvállalás lehetséges forgatókönyveit mutatta be (a minimál-programtól a full extra-programig).

Tóth András, az IJC kutatója a média-konvergencia és a kábeltelevíziózás kapcsolatát elemezte előadásában. A hazai kábeltelevíziózásban problémát jelent a 40-50%-os penetráció, a fragmentált, és rövidtávon nem támadható piaci struktúra, és a kizsákmányoló típusú visszaélések elterjedtsége. A konvergencia-modellek átalakításával megválaszolható, hogy szükséges-e ezen a területen az önálló szabályozás, vagy elegendő a távközlésre (elektronikus hírközlésre) vonatkozó új szabályozási csomag (a továbbiakban: NRF) alkalmazása. A kutató szerint a konvergencia által generált verseny középtávon is megoldhatja a piac problémáit, egy önálló szabályozás azonban veszélyeztetheti a konvergenciát és az NRF automatikusan a korszerű szabályozáson kívül marad. Ebből következik tehát, hogy nincs szükség a területen önálló szabályozásra.

Kovács András, a Széchenyi Egyetem kutatója előadásában a média-konvergencia és médiatulajdonlás kapcsolatát vizsgálta. Az alapvető gazdasági-technikai háttérből kiindulva határozta meg a médiakoncentrációkra vonatkozó szabályozás elsődleges célját, amit a médiaverseny megvalósításában lát. Ezután a különböző szabályozási formákat vetette össze az előadó, így a társ- és önszabályozás, az általános versenyjogi illetve szektor-specifikus szabályozás előnyeit és hátrányait ismertette, végül a szabályozó hatóságok elhelyezésének kérdéskörében a tagállami szintű szervezetek felállítása mellett érvelt.

Révész T. Mihálynak, a győri Széchenyi Egyetem Állam- és Jogtudományi Intézete tanszékvezetőjének előadása a jelenleg hatályban lévő médiatörvény összeegyeztethetőségét vizsgálta a digitalizációval, majd az össze nem egyeztethetőség kérdéskörét járta körül. Szerinte egy új, vagy legalábbis a régit koncepcionálisan átalakítani szándékozó szabályozásnak kellene következnie ahhoz, hogy az megfeleljen a digitális kihívásoknak.

Nyakas Levente, a Széchenyi Egyetem kutatója a nemzetközi audiovizuális szabályozás legutóbbi időszakából ragadott ki néhány fontosabb eseményt és dokumentumot. Ezeken keresztül próbált meg képet adni arról, hogy Európában az utóbbi másfél évben milyen elmozdulások, irányok mutathatók ki. A klasszikus média-szabályozási tárgykörökben ismertette a Bizottság 2003-as Közleménye alapján az audiovizuális szabályozás technikáját érintő változásokat.

Gyenge Anikó, az IJC kutatója a média-konvergencia és a szerzői jogi szabályozás viszonyát vizsgálva bemutatta, hogy az Európai Közösség szerzői jogi jogalkotása milyen mértékben veszi figyelembe a digitalizációt, és ismertette a magyar szabályozásnak az e téren az utóbbi időben bekövetkezett változásait. Az ismeretlen felhasználási módokra vonatkozó szerződési feltételek és a digitális jogkezelési rendszerek elemzése kapcsán pedig arra a következtetésre jutott, hogy a szerzői jog ugyan néha lassíthatja a konvergencia terjedését, de ez a „fék” elengedhetetlenül szükséges a hozzáféréssel kapcsolatban érvényesítendő érdekek kiegyensúlyozásához.

A konferenciát egy kerekasztal-beszélgetés zárta le, amelynek résztvevői, *Rozgonyi Krisztina* (NHH), *Cseh Gabriella* ügyvéd, *Zachar Balázs* (NKÖM), *Sere Péter* (IHM) és *Hazay István*, az ORTT képviseletében olyan kérdésekre kerestek választ, mint a műsorszórás szerepe a konvergencia-folyamatokban, a jövő média-szabályozóhatósága, egy átfogó audiovizuális koncepció megalkotásának lehetőségei, valamint az ön- és társszabályozás lehetőségei.

A konferencián elhangzott előadások a közeljövőben önálló tanulmánykötetben olvashatók lesznek, az előadásokhoz készült bemutatók pedig jelenleg is hozzáférhetők a www.ijc.hu weboldalon.

Gyenge Anikó

Hírek

A **Határőrség** vezetése elhatározta egy, a schengeni követelményeknek megfelelő határellenőrzési és az illegális migráció ellenes integrált, nyílt és átjárható információs rendszer megvalósítását, amely biztosítja az események folyamatos ellenőrzését, a gyors reagálást, az operatív beavatkozást és a járőrök hatékonyabb működését. A feladat bonyolultsága nyilvánvalóvá tette, hogy szükséges egy szakértői rendszer kialakítása, amely képes egységbe foglalni és átláthatóvá tenni a különböző forrásokból származó adatokat. A rendszer megvalósítása lehetőséget teremt a határőrök elhelyezkedésének jelzésére és egységes térképi megjelenítésére, egyben dokumentált kommunikációt biztosít a központban és a terepen szolgálatot teljesítő határőrök között.
A rendszert az első fázisában az Orosházi Határőr Igazgatóságon és annak két kirendeltségén telepítették. A rendszert a 2003 decemberében megkötött szerződés alapján az **Ericsson Magyarország Kft.** fejlesztette ki, helyezte üzembe és integrálta a Határőrség meglévő rendszereibe. A követelmények és a jogszabályok figyelembevételével a legmodernebb, térinformatikai alapokon nyugvó műveletirányítási rendszer született. A GPS alapú megoldásokat és a kétirányú kapcsolattartást az irányítóközponttal különböző rádiós távközlési technológiák integrálása tette lehetővé. Az irányítóközpont munkáját digitális térképek, integrált vezénylési mechanizmusok, továbbá eszköz- és erőforrásgazdálkodási rendszer segíti.

Az IDC piackutató cég nemrégiben nyilvánosságra hozott adatai szerint a **Hewlett-Packard** 2004. második negyedévében is stabilan őrizte vezető helyét az iparági szabvány (vagyis az x86-os processzorarchitektúrára épülő) szerverek hazai piacán.
A pozíció megtartása mellett a cég számos további pozitív fejleményről is beszámolhatott, hiszen az elmúlt év hasonló időszakával összehasonlítva piaci részesedése 47,5 százalékról 52 százalékra növekedett, míg forgalmának volumenét több mint 40 százalékkal sikerült növelnie. Mindez azt jelenti, hogy a hazánkban április és június között eladott, mindösszesen 2250 iparági szabvány szerverből, kiterjedt partnerhálózata segítségével 1170-et a HP értékesített. Hasonló folyamatok zajlottak Magyarország tágabb környezetében, az EMEA (Europe, Middle East and Africa) régióban is: a HP-nak itt is egy eleve magas (40 százalékot meghaladó) bázisról sikerült még tovább növelnie piaci részesedését, amely így elérte a 42,2 százalékot. Ez konkrét számokra lefordítva több mint 178 000 eladott szervert jelent, mindössze 3 hónap leforgása alatt. Az elmúlt év hasonló időszakával összehasonlítva a HP eladásai több mint 40 000 darabbal (27,6 százalékkal) nőttek, vagyis a cég a piac egészénél gyorsabban növekedve, konkurensei ellenében szerzett újabb részesedést. Külön érdekesség, hogy a HP nemcsak az összesítésben, hanem a régió országait külön-külön vizsgálva is mindenütt az élen végzett, sőt, az IDC által vizsgált országok jelentős részében piaci részesedése immár meghaladja három legnagyobb vetélytársának (Dell, IBM, FSC) egyesített eredményét is.

A **Tele Atlas,** a digitális földrajzi adatbázisok egyik vezető fejlesztője stratégiai szövetséget kötött az **Oracle**-lel, amely lehetővé teszi, hogy a Tele Atlas az Oracle-szoftverekre alapozza belső tevékenységét a terepadatok gyűjtésétől a végleges termékekig.
Az egész világra kiterjedő együttműködés lehetővé teszi, hogy a Tele Atlas térképadatai az Oracle Database 10g térinformatikai adatformátumában álljanak majd rendelkezésre. A megállapodás nagy hangsúlyt fektet a térképadatok online elérhetőségére és arra, hogy azokat az Oracle szoftvereivel és szolgáltatásaival közvetlenül használni lehessen, különösen az ügyfélkapcsolat-kezelés (CRM), az e-üzlet, az integrált vállalatirányítási alkalmazások (ERP) és a kapcsolódó szolgáltatások terén. Ezek a tipikus üzleti alkalmazások egyre inkább igénylik a helyfüggő funkciókat, például a helyszíni szervizelés és a szállítói láncok kezelése esetében.

TECHNOLOGICAL DESIGN OF SYSTEM RELIABILITY
Technical Committee 65 (TC 65) of the International Electrotechnical Committee (IEC) developed guidelines for the technological design of system reliability. Based on these guidelines this paper reviews factors influencing system reliability and the methods through which reliability targets can be achieved.

TRAFFIC MANAGEMENT

IMPLEMENTING CALL ADMISSION CONTROL IN INTEGRATED VOICE AND DATA NETWORKS
Keywords: bandwidth agent, network management, service level agreement (SLA), performance analysis
The wide range penetration of the Internet has increased the interest in voice transmission over the Internet. Since the Net was not developed for real-time data communication, several technical problems occur which have to be solved for high quality transmission of voice. Bandwidth agent is a device for the dynamic management of resources. It manages long-term agreement between users and service providers, performs call admission control as well as verifies signals and rights to the reservation of resources. The paper introduces a network management device implementation that can be used for managing resources of integrated voice and data communication networks and has also a call admission control function.

NEW METHODS FOR THE ESTIMATION OF RESOURCE REQUIREMENT OF PACKET-SWITCHED NETWORKS
Keywords: equivalent capacity, QoS, call admission
The lack of guarantees for the quality of transmission is a classical problem of packet-switched networks. Without such guarantees new, value-added services cannot proliferate on the Internet, even if the access technologies offer enough speed. This article presents resource requirement estimation schema which are easy to implement and are able to determine the minimum bandwidth required for meeting the guaranteed quality parameters. These methods can form the basis for load control algorithms (e.g. call admission) in packet-switched (access) networks.

TRAFFIC MANAGEMENT IN MULTILINK REGIONS
Keywords: routing, traffic management, load sharing
Traffic management solutions of IP networks is becoming a hot topic. Former studies were focussed on the optimization of the internal traffic of one single autonomous network, inter-regional traffic management issues were hardly addressed. This can be explained by the fact that traffic management has full information and competency within the boundaries of the region but its capabilities are rather limited in collecting information or influencing routing. In this article the difficulties of inter-regional traffic management are outlined and solutions offered by BGP are introduced. In addition a method is given for the extension of sophisticated intra-regional traffic management to regulate traffic between adjacent networks.

ROUTING

SCALABLE ROUTING IN MOBILE ENVIRONMENT
Keywords: hierarchical routing, network structures, multiservice connection, proactive planning
Future networks will be characterized by the explosion in number of devices, and most of them will be mobile. Since current IP based solutions probably will not support new network structures, new alternative addressing and routing techniques will be required which will be able to handle large and highly mobile dynamic systems. This article gives an overview of certain existing algorithms and proposals which can be the basis of a future network structure with similar requirements.

DESIGN OF TELECOMMUNICATIONS NETWORKS WITH TAKING INTO ACCOUNT THE CHANGES IN TRAFFIC DISTRIBUTION
Keywords: time-variable traffic, traffic re-loading, increase of exploitage, routing
This study proposes a new algorithm for the cost-effective planning of telecommunications networks. The applied network model uses two types of networking elements: routers and transmission routes. In practice, their capacities are discrete values therefore stepped cost functions can be attributed to each networking element. Network planning can take into account periodic (daily, weekly) variations of the traffic which could lead to lower-cost networks.

SOLUTION OF TRANSMISSION PROBLEMS

AUTOMATIC FILTERING OF SYN-FLOODS WITH THE RESPIRE ALGORITHM
Keywords: attack, counting, flooding, SYN-cookies
Several well-known servers were blocked by malevolent users with the use of the so-called SYN-flood. There are some proven and widely used methods to defend servers against such attacks. Our article proposes a new solution which allows for the automatic recognition and filtering of SYM-floods without causing considerable extra burden. Its efficiency is demonstrated with simulation and numeric analysis as well.

PERFORMANCE STUDY OF DATA COMMUNICATIONS IN SYMMETRIC DSL EQUIPMENT
Keywords: SHDSL, broadband, measuring network
In recent times DSL technology has been getting more and more popular in the SME and the SOHO sector. The xDSL technology implements voice and data connection on copper. The paper discusses the symmetrical DSL (SDSL) technology and its current position on the world market, then transmission performance of SDSL-based application is analyzed and finally products of different vendors are compared.

Summaries • of the papers published in this issue

Contents

Cover: Congestion is a problem not only in road traffic

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

Hirdetési árak
1/1 (205x290 mm) 4C 120.000 Ft + áfa
Borító 3 (205x290mm) 4 C 180.000 Ft + áfa
Borító 4 (205x290mm) 4 C 240.000 Ft + áfa

Cikkek eljuttathatók az alábbi címre is
BME Szélessávú Hírközlő Rendszerek
Budapest XI., Goldmann Gy. tér 3.
Tel.: 463-1559, Fax: 463-3289,
e-mail: zombory@mht.bme.hu

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

2004-es előfizetési díjak
Hazai közületi előfizetők részére:
1 évre bruttó 31.200 Ft
Hazai egyéni előfizetők részére:
1 évre bruttó 7.000 Ft

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

www.hte.hu
Felelős kiadó: MÁTÉ MÁRIA
Lapmenedzser: Dankó András

HU ISSN 0018-2028

Layout: MATT DTP Bt. • Printed by: Regiszter Kft.