



M Ű E G Y E T E M 1 7 8 2

AGGREGÁLT ÁLLAPOTOKON ALAPULÓ ERŐFORRÁS-FOGLALÓ PROTOKOLLOK VIZSGÁLATA

PhD Tézisfüzet

írta:

Császár András

Tudományos témavezetők:

Dr. Szabó Róbert

Dr. Henk Tamás

Nagysebességű Hálózatok Laboratórium

Távközlési és Médiainformatikai Tanszék

Budapesti Műszaki és Gazdaságtudományi Egyetem

BEADVA A

PHD FOKOZAT

RÉSZLEGES KÖVETELMÉNYEKÉNT A
BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEMRE

BUDAPEST, 2007. NOVEMBER

1. Bevezetés

Az utóbbi években az IP hálózatokon átvitt valósidejű forgalmak volumene jelentősen nőtt [1]. Ez nemcsak a folyamatosan megjelenő nagysebességű videó alkalmazásoknak köszönhető, hanem a korábban már létező szolgáltatások IP hálózatok feletti átvitelének is, mint például a telefonálás Skype-pal. Mindemellett a hagyományos távközlési forgalmakat is, mint a vonalkapcsolt hangátvitel, az adatforgalommal közös IP transzport hálózaton viszik át. Egy jóminőségű hálózatban a QoS-t igénylő forgalmak számára biztosítani kell elegendő sáv szélességet, hogy teljesíteni lehessen átbocsátási és késleltetési elvárásaikat.

Disszertációmban az egyik sáv szélesség biztosító megoldással, az útvonalmenti erőforrás-foglaló protokollokkal foglalkozom, amelyben a folyam-engedélyezés csomópontról csomópontra történik. Az útvonalmenti erőforrás-foglalás egy alternatívája a jelenleg széleskörben alkalmazott túlméretezésnek [1, 2, 3, 4] vagy a központosított megoldásoknak, mint amilyenek a sáv szélesség ügynökök [5, 6]. Az útvonalmenti erőforrás-foglalás a központosított megoldásokkal ellentétben jól illik az IP hálózatok elosztott működési elvébe. A túlméretezéssel ellentétben bármilyen váratlanul megnövekedett forgalom esetén működik vagy olyan hálózatokban is, ahol a kapacitás drága, mint például a hozzáférési hálózatok. Sőt, a jó minőségű videó szolgáltatások terjedésével a sáv szélesség biztosításának kérdése gerinchálózatokban is felmerülhet.

A szabvány RSVP (Resource Reservation Protocol) [7] erőforrás-foglaló protokoll skálázódási problémái miatt az Internet Engineering Task Force (IETF) Next-Steps in Signaling (NSIS) munkacsoportja [8] a következő generációs útvonalmenti erőforrás-foglaló protokoll szabványosításával foglalkozik. A munkacsoport két változatot készít, egy *állapotgazdag* és egy *csökkentett állapotterű* szolgáltatásminőség (QoS) modellt. Ezek a kifejezések a belső csomópontok implementációs komplexitását tükrözik. Az állapotgazdag megoldás folyamankénti állapotokat tart nyilván minden belső csomópontban, amely az RSVP alapú integrált szolgáltatások architektúrába illeszkedik. Az újabb differenciált szolgáltatások (DiffServ) architektúrában [9], amely definiálja egy tartomány szélső és belső csomópontjainak fogalmát, csak a szélső csomópontok tárolhatnak folyamszintű információkat. A belső csomópontoknak a lehető legegyszerűbbnek kell lenniük, és így csak forgalmi aggregátumokkal dolgozhatnak. A csökkentett állapotterű megoldás ezért a belső csomópontokban forgalmi osztályokkal dolgozik a DiffServ erőforrás-menedzsment protokoll (RMD) segítségével [O1, C6].

Értekezésemben megmutatom, hogy egy nagysebességű hálózatban a csökkentett állapotterű megoldás általában, hálózati hibáktól mentes állapotban, előnyösebb az állapotgazdag működésnél. Ennek oka, hogy a csökkentett állapotterű protokoll egyszerűbb, kisebb számítási teljesítményt igényel a belső forgalomirányítóktól, és így jobban skálázódik nagy hálózatokhoz, miközben az elérhető teljesítmény javarészt hasonló.

A hibamentes eset mellett a foglaló protokollnak kezelnie kell a link vagy csomópont hibákból adódó kivételes helyzeteket is. Markopoulou és társai [10] megmutatták, hogy egy nagyobb IP hálózatban a linkhibákat mindennapos eseménynek lehet tekinteni. Kimutatták, hogy a Sprint gerinchálózatában átlagosan 30 percenként történnek linkhibák. Ezek okai különfélék lehetnek, mint egy optikai szál elvágása vagy más környezeti hatások, csomópont hardver vagy szoftver hibái vagy operátor hibák. Watson és társai [11]-ben megfigyelték, hogy az útvonalváltások gyakoriak egy élő hálózatban.

Hibák esetén az IP útválasztó protokolljai átirányítják a folyamatokat az eredeti útvonalokról alternatív útvonalakra. Habár az útválasztó protokollok szerves részét képezik a mai robusztus IP hálózatoknak, feltalálásuk abban az időben történt, amikor az IP hálózatoknak nem kellett különösebb minőségi igényeknek megfelelnie. Átirányításkor ugyanis a QoS megoldásoknak szembe kell nézniük azzal a lehetőséggel, hogy az új útvonalon nem biztosan áll elegendő szabad kapacitás rendelkezésre minden folyam minőségi kiszolgálásához. A korábbi beengedés ellenére tehát torlódás alakulhat ki. Iyer és társai [12]-ben megmutatták, hogy a túldimenzionált hálózatokban az Internet torlódásainak többségét átirányítások okozzák. Ez arra enged következtetni, hogy az átirányítások gyors és hatékony kezelése talán az egyik legfontosabb feladata egy sáv szélesség fogláló protokollnak. Éppen ezért az értekezésem jelentős része az RMD átirányítás kezelésével foglalkozik, amely a kutatásaim előtt nem rendelkezett a szükséges képességekkel.

2. Kutatási célok

Kutatásaim során kiértékeltem az aggregált állapotokon alapuló erőforrás-foglaló protokollokat, majd ezen eredményeket felhasználva javítottam az RMD képességein. Munkám egy jelentős célja volt az RMD szabványosításának támogatása. Habár a csökkentett állapotterű protokollok egyértelmű előnyökkel rendelkeznek állapotgazdag társaik felett, mint az egyszerűség és skálázhatóság, hátrányokra és megoldandó problémákra is fény derült.

A protokoll olyan aspektusait vizsgáltam, amelyek az eredeti tervezésben nem kaptak szerepet. Kíváncsi voltam, hogy az a tény, hogy a protokoll csak aggregált információkkal dolgozhat folyamszintű ismeretek helyett, hogyan korlátozza az alkalmazható folyamengedélyező algoritmusokat (1. Téziscsoport). Azt is tudni szerettem volna, hogy az RMD-nek vannak-e hátrányai a szabványos RSVP-hez képest (2. Téziscsoport).

Eredményeim azt mutatták, hogy normál körülmények között az RMD legalább olyan jó, mint állapotgazdag versenytársa, az RSVP, sőt néha még jobb is. Azt is megmutattam azonban, hogy milyen negatív következményekkel jár az átirányítás, amelynek gyors és precíz megoldásában hasznos szolgálatot tesznek a folyamszintű állapotok. Azt is megmutattam, hogy a helyzetet az RMD korábbi mechanizmusai nem tudják megfelelően megoldani. (3. Téziscsoport) A következő céloom ezért az volt, hogy jobb megoldást adjak az átirányítás kezelésére (4. Téziscsoport).

3. Eszköztár

Az aggregált állapotok folyamengedélyező formulákra gyakorolt limitációjának vizsgálatokor matematikai formalizmusra és analitikus vizsgálatra hagyatkoztam. Kutatásaim fő eszközei azonban protokollanalízis és csomagszintű szimulációk voltak az ns-2 [13] hálózati szimulátor programmal. Ezekhez a szimulációkhoz valószerű hang és videó forgalmi modelleket választottam, topológiaként pedig a legegyszerűbb delta hálózattól kezdve, amely a legkisebb átirányítást kezelni képes hálózat, speciális rádiós hozzáférési hálózatokon át összetettebb európai és amerikai gerinchálózatokat is vizsgáltam. A 2. Téziscsoport teljesítmény vizsgálatokor emellett architektúrális megfigyelésekre és következtetésekre is hagyatkoztam.

4. Új eredmények

4.1. Aggregát állapotokon alapuló folyam-engedélyezés

Egy csökkentett állapotterű protokollal a belső csomópontok csak aggregált információkat tárolhatnak, például forgalmi osztályonként. Ezért megvizsgáltam, hogy ez a kényszer hogyan csökkenti a lehetséges engedélyező formulák halmazát. Az aggregált állapotokkal implementálható formulák keresése azért fontos, mert ezek a csökkentett állapotú protokollokban, mint amilyen az RMD, közelítés nélkül használhatók.

1. TÉZISCSOPORT – Aggregát állapotokon alapuló folyam-engedélyezés

A csökkentett állapottérrel kompatibilis engedélyező formulák definiálása után megmutattam, hogy ezek a formulák az állapotgazdag folyamengedélyező formulák valódi részhalmazát alkotják, majd feltételeket adtam a kompatibilitásra. Arra jutottam, hogy bár nem minden, de azért több formula is használható aggregált, sőt akár még periodikusan frissített állapotokkal is.

Amikor egy belső csomópont egy folyamengedélyezési kérelmet kap, az engedélyező algoritmus egy formulát értékel ki. Ha a formula pozitív, a folyamat beengedi és a folyam leírót beolvasztja adatbázisába. Vizsgálódásaim során azon engedélyező formulákkal foglalkoztam, amelyek a következő információk alapján hozzák meg döntésüket:

- Az új folyam leírói (például csúcssebesség, átlagsebesség, aktivitási faktor)
- Az aktív folyamatokról tárolt információk, amelyeket korábbi engedélyezésükkor jeleztek
- Aggregátumokról mért adatok (például linkkihasználtság, sorhossz)
- Paraméterek (például kapacitás, engedélyezési küszöb, túlcsordulási valószínűség)

1.1. Definíció (Folyamengedélyező formula). Ha a rendszerben n darab előzőleg beengedett (aktív) folyam található, és döntenie kell egy új folyam engedélyezéséről, akkor az engedélyező formula definíciója:

$$F_n(\bar{\mathbf{P}}_n, \bar{p}_{új}, \bar{M}, \bar{C}) \leq 0, \text{ ahol}$$
$$\bar{\mathbf{P}}_n \triangleq \begin{pmatrix} p_1^1 & p_1^2 & \dots & p_1^k \\ \vdots & \vdots & \vdots & \vdots \\ p_n^1 & p_n^2 & \dots & p_n^k \end{pmatrix}, \quad \bar{p}_{új} \triangleq \{p_{új}^1, p_{új}^2, \dots, p_{új}^k\},$$
$$\bar{M} \triangleq \{M^1, M^2, \dots, M^m\} \quad \text{és} \quad \bar{C} \triangleq \{C^1, C^2, \dots, C^l\}.$$

$p_i^j \in \mathbb{R}$ jelenti az i -ik folyam j -ik ($j = 1..k$) leíróját, tehát egy folyamatot k darab leíró jellemez. A formula $m \in \mathbb{N}$ mért értéket ($M^i, i = 1..m$ és $l \in \mathbb{N}$ darab előre beállított paramétert ($C^i, i = 1..l$) tartalmaz.

Egy folyam k darab leíróját rövid formában jelölheti egy k hosszú vektor:

$$\bar{p}_i \triangleq \{p_i^1, p_i^2, \dots, p_i^k\}$$

Ahhoz hogy egy folyamengedélyező formula alkalmazható legyen egy aggregált állapotokon alapuló erőforrás-foglaló protokollban, a folyamengedélyező formulának rendelkezni kell csökkentett állapotú implementációval. Az olyan algoritmusok, amelyek csak állapotgazdag módon implementálhatóak, nem skálázódnak nagysebességű hálózatokhoz, ahol egy belső csomópontnak folyamatosan több ezer vagy akár több százezer folyamat kellene nyilvántartania.

1.1. Észrevétel. Érdekes kérdés az, hogy csökkentett állapottérrel inkább memóriát vagy számítási teljesítményt spórolunk meg? A memóriabejegyzések száma sok esetben korrelál a számítási idővel, ami egy bejegyzés kereséséhez, felvételéhez vagy törléséhez szükséges. Ez azt jelenti, hogy több állapot tárolása általában nem csak több memóriát, de gyorsabb processzort igényel. Kivételt képeznek azonban a hardveres asszociatív memóriák, ahol a számítási idő konstans, a bejegyzések számától független. Persze ebben az esetben a memóriaméret növekedése sokkal hátrányosabb, mert ezek a tartalom címezhető memóriák igen drágák. Egy másik kivételt képez az az eset, amikor a memória olyan nagy, hogy minden lehetséges bejegyzésnek dedikált cellája van. Ekkor például keresés helyett elég címezni. Folyamengedélyezés esetén egy folyamat tipikusan egy ötös azonosít: a forrás és cél IP cím, a forrás és cél portszám és a protokoll azonosító. IPv4 esetén ha minden lehetséges értékre fel akarnánk készülni, akkor ez $2^{(2 \cdot 32 + 2 \cdot 16 + 8)}$ számú regisztret jelentene, ami elképzelhetetlen.

1.2. Definíció (Csökkentett állapot kompatibilitás). Egy folyamengedélyező formula csökkentett állapot kompatibilis, és $d \in \mathbb{N}$ darab változó nyilvántartását igényeli, ha bármely n aktív folyam esetén létezik $S_n : \mathbb{R}^{n \times k} \rightarrow \mathbb{R}^d$ transzformációs függvény úgy, hogy található egy

1. $G : \mathbb{R}^d \times \mathbb{R}^k \times \mathbb{R}^m \times \mathbb{R}^l \rightarrow \mathbb{R}$ transzformált engedélyező függvény, amelyre

$$G(S_n(\bar{\mathbf{P}}_n), \bar{p}_{új}, \bar{M}, \bar{C}) = F_n(\bar{\mathbf{P}}_n, \bar{p}_{új}, \bar{M}, \bar{C}), \text{ és egy}$$

2. $\Phi : \mathbb{R}^d \times \mathbb{R}^k \rightarrow \mathbb{R}^d$ érkezéskori állapottér frissítő függvény, amelyre

$$\Phi(S_n(\bar{\mathbf{P}}_n), \bar{p}_{új}) = S_{n+1}(\bar{\mathbf{P}}_{n+1}), \text{ és egy}$$

3. $\Psi : \mathbb{R}^d \times \mathbb{R}^k \rightarrow \mathbb{R}^d$ távozáskori állapottér frissítő függvény, amelyre

$$\Psi(S_{n+1}(\bar{\mathbf{P}}_{n+1}), \bar{p}_j) = S_n(\bar{p}_1, \dots, \bar{p}_{i \neq j}, \dots, \bar{p}_{n+1}) \quad \forall j = 1..(n+1).$$

Az $S_n(\cdot)$ függvények biztosítják, hogy tetszőleges számú folyamat leíró információ tömöríthető konstans számú változóra, tehát egy változó $(n \times k)$ méretű mátrix tömöríthető egy fix d elemű vektorral (állapottérrel).

Az 1.2. Definíció első feltétele azt jelenti, hogy létezik *egy közös* függvény, amely egy konstans méretű, aggregált információkat tároló vektor alapján pontosan ugyanazt az engedélyezési döntést hozza, mint az eredeti formula, azaz az n darab F_n függvény hozna egy változó méretű, folyamszintű információkat tároló mátrix alapján.

A 2-ik feltétel biztosítja, hogy ha egy folyamat beengedünk, akkor a leírót belefoglalhatjuk a meglévő konstans számú állapotleíróba. Míg állapotgazdag esetben a folyamleírót csak be-tennénk az adatbázisba, csökkentett állapotter esetén a folyamleírók mindig *ugyanazzal az egy* függvénnyel beleolvashatók az aggregált állapotokba.

A 3-ik feltétel biztosítja, hogy egy folyamat távozásakor mindig *ugyanazzal az egy* függvénnyel kiszámolható az az állapotter, amit akkor kapnánk, ha a folyamat nem is lett volna a rendszerben.

1.1. TÉZIS – Valódi részhalmaz [D]

Bebizonyítottam, hogy az 1.2. Definíció szerint aggregált állapotú engedélyező for-mulák valódi részhalmazát képezik az összes engedélyező formulának.

1.1. *Példa.* Az $n \cdot \max(p_1, \dots, p_n) \leq C$ formula nem csökkentett állapot kompatibilis:

Habár nem sikerült szükséges és elégséges feltételt találnom a csökkentett állapotter kompa-tibilitásra, találtam feltételeket, amelyek vagy elégségesek vagy szükségesek.

1.2. TÉZIS – Szükséges feltétel: érkezési sorrendtől való függetlenség [D]

Megmutattam, hogy az aggregált állapot kompatibilitás szükséges feltétele a folya-mok érkezési sorrendjétől való függetlenség.

Az előbbi feltétel „csak” szükséges, de nem elégséges. Az 1.1. Példában szereplő formula is független az érkezési sorrendtől. Ha az 1.2. Definíció 2 feltételére pillantunk, szembe-tűnő, mennyire emlékeztet a rekurzióra.

1.3. Definíció (Függvénysorozat). Munkámban a függvénysorozatot függvények egy sorozatá-nak definiáltam, amelyben az argumentumok száma a futó indextől függ, azaz $F_n(\bar{p}_1, \dots, \bar{p}_n)$.

1.4. Definíció (Rekurzív függvénysorozat). Egy $R_n(\bar{p}_1, \dots, \bar{p}_n)$ függvénysorozat rekurzív, ha bár-mely n és \bar{p}_n -re létezik egyetlen közös f függvény úgy, hogy

$$R_n(\bar{p}_1, \dots, \bar{p}_n) = f(R_{n-1}(\bar{p}_1, \dots, \bar{p}_{n-1}), \bar{p}_n).$$

1.3. TÉZIS – A kompatibilitás alapja a rekurzió [D]

Ha egy engedélyező formula felírható konstans ($d \in \mathbb{N}$) számú rekurzív függvényso-rozat ($R_n()$) függvényeként:

$$F_n(\bar{p}_1, \dots, \bar{p}_n, \bar{p}_{új}) = G(R_{n+1}^1(\bar{p}_1, \dots, \bar{p}_n, \bar{p}_{új}), \dots; R_{n+1}^d(\bar{p}_1, \dots, \bar{p}_n, \bar{p}_{új})),$$

akkor a formula implementálható konstans $O(1)$ számú változóval abban a speciális helyzetben, ha a folyamatok csak érkeznek, de nem távoznak, azaz az 1.2. Definíció (1) és (2) feltételei teljesülnek.

Egy folyamat érkezésekor a rekurzív függvénysorozat következő tagja kiszámítható az előző tagból és az új folyamat leíróiból. Csakhogy semmi nem garantálja, hogy a távozó folyamatok el-távolíthatók az aggregált állapotokból, mivel nem biztos, hogy létezik egy fordított függvény,

amely képes lenne az előző tagot számítani, mint ahogy ez az 1.2. Definíció 3. feltételéhez kel-
lene. Érdekes kérdés, hogy mikor található fordított függvény. A következő tézis, amely a [P9]
szabadalmi bejelentés alapját is képezi, elégséges feltételt ad a teljes csökkentett állapot kompa-
tibilitásra. A fordított függvény a szumma és produktum konstrukciókban található összeadás és
szorzás operátorok kommutatív, asszociatív és invertálható tulajdonsága miatt létezik.

A protokollok megbízhatóbbak, ha a foglalások nem ragadhatnak bent korlátlan ideig, emiatt
a legtöbb erőforrás-foglaló protokoll periodikusan frissíti a foglalásokat. A tárolt állapotokból ki
kell üríteni a nem frissített foglalásokat éppúgy, mintha az adott folyam explicit távozási jelzést
küldött volna. Ezt „puha állapotú” megközelítésnek is hívják. Mivel az RMD-ben nincsenek
folyamszintű állapotok, folyamonként egy-egy időzítő futtatása, mint RSVP-nél, itt nem meg-
oldható. Ehelyett az aggregált állapotok periodikusan újraépülnek. Eredetileg az RMD-t csak
az egyszerű, bitsebességek összegén alapuló engedélyezésre¹ tervezték, amihez Karagiannis és
társai javasoltak egy puha állapotú megoldást [14], amelyekben a folyamatok periodikus R időkö-
zönként frissítik a foglalásaikat. Az ötletet általánosítottam, hogy a következő tézis sablonjára
illeszkedő minden formulára lehetővé tegye a puha állapotok használatát:

1.4. TÉZIS – Szumma és produktum konstrukciók és puha állapotú megvalósításuk [P9,D]

*Megmutattam, hogy ha egy engedélyező formula az alábbi szumma és produktum
konstrukciókból álló sablonra illeszkedik, akkor csökkentett állapottér kompatibilis:*

Sablon

$$G\left(\left\{\left(\sum \mid \prod\right)_{i=1}^n g_1(\bar{p}_i), \dots, \left(\sum \mid \prod\right)_{i=1}^n g_d(\bar{p}_i)\right\}, \bar{p}_{ij}, \bar{M}, \bar{C}\right) \leq 0,$$

ahol $d \in \mathbb{N}$, $d < \infty$ és $g_j : \mathbb{R}^k \rightarrow \mathbb{R}$ $j = 1..d$.

Vegyük észre, hogy

- a sablon konstans (d) számú szumma vagy produktum konstrukciót tartalmaz,
amelyeket $(\sum \mid \prod)$ jelöl;
- a szumma és produktum konstrukciókon belül minden tag vagy tényező egy
függvény lehet $(g_i(\cdot))$, de csak egyetlen folyam leíróinak függvénye;
- a külső (G) függvény nem függ közvetlenül a folyamleíróktól, csak a szumma
és produktum konstrukcióktól;
- a sablonra illeszkedés egy elégséges feltétel.

*Az 1.1. Algoritmussal megoldható a sablonra illeszkedő formulák puha állapotú
megvalósítására.*

¹lásd [D]/Sec. 3.2.1

1.1. Algoritmus. Tegyük fel, hogy a formula d szumma vagy produktum konstrukciót tartalmaz, melyek közül –az általánosság csorbítása nélkül– az első s darab szumma, a többi produktum, azaz

$$G\left(\left\{\sum_{i=1}^n g_1(\bar{p}_i), \dots, \sum_{i=1}^n g_s(\bar{p}_i), \prod_{i=1}^n g_{s+1}(\bar{p}_i), \dots, \prod_{i=1}^n g_d(\bar{p}_i)\right\}, \bar{p}_{új}, \bar{M}, \bar{C}\right) \leq 0.$$

Tegyük még fel, hogy $last_i$ ($i = 1..d$) jelöli a szumma vagy produktum konstrukciók aktuális értékét. Ezzel az engedélyezési feltétel egy új folyam érkezésekor az alábbi szerint alakul:

$$G(last_1, \dots, last_d; \bar{p}_{új}, \bar{M}, \bar{C}) \leq 0$$

Az R hosszú időablakokat minden forgalomirányítóban N darab időintervallumra, úgynevezett cellákra bontjuk. A rendszer külön-külön számolja és tárolja az aktív cellán kívül az utolsó N cellát is a C_i^j ($j = 1..N$, $i = 1..d$) számlálók segítségével. Az aktuális cell frissítéseit és foglalásait a $count_i$ ($i = 1..d$) változók gyűjtik.

Ha egy új foglalás kérelmet engedélyezünk, akkor

$$\begin{aligned} \text{for } i = 1..s : \{ last_i := last_i + g_i(\bar{p}_{új}); count_i := count_i + g_i(\bar{p}_{új}); \} \\ \text{for } i = (s+1)..d : \{ last_i := last_i \cdot g_i(\bar{p}_{új}); count_i := count_i \cdot g_i(\bar{p}_{új}); \} \end{aligned}$$

Ha egy foglalást frissítő üzenet érkezik:

$$\begin{aligned} \text{for } i = 1..s : count_i := count_i + g_i(\bar{p}_{új}) \\ \text{for } i = s+1..d : count_i := count_i \cdot g_i(\bar{p}_{új}) \end{aligned}$$

Minden R/N hosszú cella után az állapotokat újraszámoljuk a csúszóablak egy cellányi eltolásával:

$$\begin{aligned} \text{for } i = 1..d : \{ \\ \text{for } j = N..2 : C_i^j := C_i^{j-1} \\ C_i^1 := count_i \\ \text{if } i \leq s \{ last_i := \sum_{j=1}^N C_i^j ; count_i := 0 \} \\ \text{else } \{ last_i := \prod_{j=1}^N C_i^j ; count_i := 1 \} \\ \} \end{aligned}$$

1.1. Következmény (Formulák alkalmazhatósága).

Néhány, az irodalomban publikált engedélyező formuláról példaként megmutattam, lásd [D]/Sec. 3.2, hogy illeszkednek a fenti sablonra, például a Hoeffding-korlátot alkalmazó megoldás [15], a Heavy Traffic Approximation alapú engedélyezés [16] vagy a Tangent at Peak [17] engedélyező algoritmus. Így arra jutottam, hogy több formula alkalmazható csökkentett állapottérrel, sőt puha állapotú protokollban is megvalósíthatók.

4.2. Az RMD teljesítménye

Az előző alfejezetben megmutattam, hogy sok esetben ugyanaz a helyi engedélyező algoritmus használható RMD-ben aggregált állapotokkal, mint RSVP-ben folyamankénti állapotokkal.

Emellett megvizsgáltam az RMD és az RSVP protokollból adódó teljesítménykülönbségét is, ha azonos engedélyezést feltételezek. Ebben az esetben csak a protokoll mechanizmusai okozhatnak különbségeket olyan fontos teljesítmény-mutatókban, mint a jelzési idők, jelzési többletforgalom vagy a folyamatok blokkolási aránya. Az RMD és elődje, a Load Control [18] tervezésekor ezeket a szempontokat nem vizsgálták, ám az RMD szabványosításakor fontosak voltak annak kimutatásában, hogy az új RMD protokoll, miközben egyértelműen skálázhatóbb a korábbi RSVP-nél, nem rosszabb más paraméterekben sem.

2. TÉZISCSOPORT – Az RMD teljesítménye

Míg a kisebb memória és számításkapacitás igény, azaz a skálázhatóság nyilvánvaló előnyei a csökkentett állapotterű RMD-nek, megmutattam, hogy a jelzési ideje és a jelzési forgalma is legalább olyan jó, mint az állapotgazdag RSVP-nek. Megmutattam, hogy kis foglalási tranziens alatti eltérések ellenére is, ugyanazt az engedélyezési algoritmust használva, a folyamatok blokkolási aránya és az elérhető kihasználtság hasonló a két protokoll esetében. Ha viszont egyik protokoll sem alkalmazna explicit távozási jelzéseket, az RMD egyértelműen jobb volna.

Ezen szempontok figyelembe vételével arra a következtetésre jutottam, hogy hibamentes körülmények között az egyszerűbb RMD protokoll hasonló, vagy néhány esetben jobb teljesítményű, mint az RSVP.

Az RMD első verziója, a Load Control, nem rendelkezett explicit távozási jelzés üzenettel, csak a periodikusan frissített puha állapotokra hagyatkozott. Ha az explicit jelzéseket nem használnánk, vagy csak elvesznek, a folyamam tényleges távozásától a puha állapotok időtűllépéséig a fölös foglalatások kihasználatlan sáv szélességet eredményeznek. Szintén fölös foglalatást, és így kihasználatlan sáv szélességet eredményez, ha egy folyamot az útvonal egy részén beengedték a csomópontok, de később egy másik csomópont visszautasítja. Ekkor némi idő telik el, míg az útvonal első felén a fölös többletfoglalatás felszabadul.

A fölös többletfoglalatásokkal és a jelzési időkkel foglalkozik első tézisem:

2.1. TÉZIS – Fölös többletfoglalatások és jelzési idők [C8, O6, D]

Bebizonyítottam, hogy ha $E(H)$ jelöli a folyamatok várható élettartamát, R pedig a frissítési periódust, akkor explicit jelzés nélküli folyamamtávozás miatt a puha állapotok többletfoglalatása RMD-vel kevesebb mint RSVP-nél. Pontosabban, a fölös többletfoglalatás RMD-nél átlagosan $\frac{0.5(R+\frac{R}{N})}{E(H)}$, míg RSVP-nél $\frac{0.75R}{E(H)}$.

Megmutattam, hogy az RMD valamivel hosszabb ideig tartja fönna visszautasított folyamatok átmeneti fölös foglalatását, mint az RSVP. (RMD: 1 RTT; RSVP: átlagosan $\frac{1}{2}$ RTT, max. 1 RTT)

Megmutattam, hogy egyirányú adatáramlások esetén, ha figyelmen kívül hagyjuk a csomópontokban a jelzési üzenet kezelésével eltöltött időt, az RMD pontosan ugyanannyi idő alatt jelzi a küldőnek a foglalatás sikerét, mint az RSVP. Ha a csomópontokénti kezelési időt is figyelembe vesszük, az RMD gyorsabb. A fogadó oldalnak

az RMD mindig legalább egy körbefordulási idővel hamarabb jelez, mint az RSVP. Kétirányú áramlások esetén az RMD mindkét oldalnak gyorsabban jelzi a foglalás sikerét és sikertelenségét is.

2.1. *Példa.* Tegyük fel, hogy mindkét protokollnál az $R=30$ másodperces frissítési intervallumot és RMD-nél az $N = 10$ cella alapértelmezett beállításokat használjuk. Alkalmazzuk a tézis első állítását VoIP és más telefonhívások modellezésekor gyakran használt hívástartási időkre. Ha 90 másodperces átlagos hívástartási időt feltételezünk, explicit jelzések nélkül az RMD puha állapotainak átlagosan 18,3% fölös foglalása van, míg az RSVP-nek 25%. 120 másodperces tartási időt feltételezve az RMD többletfoglalása 13,75%-ra csökken, az RSVP-jé 18,8%-ra.

2.1. *Észrevétel.* Az RSVP frissítési és időzítési mechanizmusai módosíthatók lennének, hogy az RMD-vel hasonló fölös foglalási arányokat nyújtson. A protokollt leíró szabványok azonban ezt nem így specifikálják, mivel a hosszabb időtúllépési limitek jobb tűrést biztosítanak a frissítési csomagok véletlen veszteségek.

A visszautasított hívások fölös többletfoglalása esetleg kisebb hálózatkijáratlást eredményezhet. Ezért szimulációk segítségével megvizsgáltam a hálózati kijáratlást és a blokkolási arányt.

2.2. TÉZIS – Blokkolás és hálózatkijáratlást [C8, C3, C6, C11, J4]

Kimutattam, hogy az RMD képes ugyanazt a blokkolási arányt és ugyanazt a hálózati kijáratlást elérni mint az RSVP, miközben a folyamatoknak ugyanannyi adatsebességet biztosít.

Az erőforrás-foglaló protokoll több területen is egyfajta többletet igényel a hálózattól. A protokollüzenetek kezelése miatt számítási kapacitásbeli többletet igényel a csomópontokban, az állapotok nyilvántartása végett memóriakapacitás többletre van szüksége. A kereső algoritmusok futási ideje miatt a memóriában és a számítási kapacitásban mutatkozó többletigény összefügg². A memóriagigényeket az 1. táblázat mutatja, ahol f a folyamatok számát jelenti. Megállapítható, hogy az RMD-vel ellentétben az RSVP belső csomópontjai nem skálázódnak jól sok folyamatot kiszolgáló nagysebességű hálózatokban.

	Szélső cs.p.	Belső cs.p.
RMD	$O(f)$	$O(1)$
RSVP	$O(f)$	$O(f)$

1. táblázat. Memória igény

A jelzési protokollok egy másik fajta többletet is igényelnek a hálózattól: a jelzési üzenetek továbbítása érdekében extra sávszélesség szükséges a linkeken. A folyamatonkénti jelzési üzenetek, illetve a periodikus frissítések miatt a jelzési többletforgalmat meg kellett vizsgálni.

2.3. TÉZIS – Jelzési többletforgalom [C8, C6, D]

Beláttam, hogy egyirányú foglalásoknál, ha a linkek nincsenek túlterhelve, az RMD többlet jelzési forgalma elhanyagolhatóan, de alatta van az RSVP-nek. Ha a hálózat terhelése növekszik visszautasított folyamatokat eredményezve, a különbség növekszik, és az RSVP észrevehetően nagyobb többletforgalmat generál. Kétirányú foglalásoknál az RMD mindig kevesebb jelzési forgalmat generál, mint az RSVP.

²Lásd az 1.1. Észrevételt az 5. oldalon

2.1. Következmény (Az RMD és az RSVP teljesítménye).

Habár az RMD rendes, hálózati hibáktól mentes körülmények között sokkal egyszerűbb és skálázhatóbb, ugyanolyan vagy jobb teljesítményű, mint az RSVP.

4.3. Erőforrás-foglalás hálózati hibák esetén

Az IP hálózatok szerves részét képezik a dinamikus útválasztó protokollok, amelyek a forgalmat új útvonalakra irányítják hibák után. Manapság egyre gyorsabb átirányítási mechanizmusok jellemnek meg, csökkentve az átirányítási időt, hogy az IP hálózatok minél megbízhatóbbak legyenek. Egy minőségi hálózat azonban nem csak az átirányítási idő fontos, hanem a QoS igények kielégítésének képessége. Megvizsgáltam, hogy az RMD és az RSVP hogy viselkedik és hogyan kellene viselkednie a folyamatok átroutolása esetén.

Az RMD szabványosítás alatt álló verziója, és legtöbbször az RSVP implementációk is egy egyszerű folyamengedélyező eljárást implementálnak, amely az adatsebességek összegén alapszik. Ha p jelöli egy folyam csúcsebességét, akkor ez a módszer akkor enged be egy folyamatot, ha

$$\sum_{i=1}^n p_i + p_{új} \leq A ,$$

ahol A az adott link engedélyezési korlátja.

2.1. Definíció (Torlódás, túlterhelés). Ha egy link pillanatnyi QoS forgalmának volumenét V jelöli, akkor a link torlódott, ha $V > A$.

Az erőforrás-foglaló protokoll torlódásmentes átvitelt ígért a folyamatoknak, amikor beengedte őket. Ezt az ígéret lehet, hogy kénytelen megszegni, mivel a ma elterjedten használt útválasztó protokollok, mint az OSPF, RIP, IS-IS vagy a BGP tőle függetlenül működnek, és átterelhetik a folyamatokat olyan útvonalra, ahol nincs mindegyiknek elég kapacitás. Azaz, az erőforrás-foglalás ígérete ellenére a tőle független átirányítás *azonnali torlódást* okozhat. Ez a torlódás mind az átirányított folyamatok, mind a már korábban is ott haladó folyamatok minőségét elronthatja.

Formálisan, ha $V_{után}$ jelöli egy link átirányítás utáni QoS forgalmát, akkor a *túlterhelés* mértéke átirányítás után $V_{túlterh} = V_{után} - A$.

Egy átirányítás lehet, hogy nem csak a már aktív folyamatok minőségét rontja, hanem az újonnan érkező folyamatokat is degradálja, ha a foglaltsági állapotok nem tükrözik azonnal a megnövekedett forgalmat, és az átirányított folyamatokat az eredetileg ott haladó folyamatokkal közös precedencia szinten továbbítjuk, akkor az engedélyező eljárás *az újonnan érkező hívásokat, hibásan, az engedélyezési szint felett is beengedheti*. Végző soron, az engedélyezés lehet, hogy nem éri el elsődleges célját, a torlódás elkerülését.

3. TÉZISCSOPORT – Erőforrás-foglalás hálózati hibák esetén

Megfogalmaztam azokat a tulajdonságokat, amelyeket az átirányítást megoldó eljárásoknak teljesítenie kell a minőségromlás minimalizálása és a hálózatüzemeltető felügyeletének biztosítása érdekében. Megmutattam, hogy bár a folyamszintű állapotok hasznosak az átirányítás kezelésében, egyik létező megoldás sem teljesíti az összes tulajdonságot.

3.1. TÉZIS – Az átirányítás hatékony kezelésének tulajdonságai [J4, D]

A következő listában megfogalmaztam azokat a tulajdonságokat, amelyeket az átirányítást megoldó eljárásoknak teljesítenie kell a minőségromlás minimalizálása és a hálózatiüzemeltető felügyeletének biztosítása érdekében:

1. Tulajdonság: A torlódás megszüntetése érdekében $V_{\text{túlterh}}$ volumennek megfelelő mennyiségű folyamat meg kell szüntetni. A „megszüntetni” a megkülönböztetett átvitel megszűnésére utal, és a következőket jelentheti:

- Nem engedjük be a hálózatba a megszüntetendő folyamat egyetlen további csomagját sem. Ez a megközelítés megszünteti a forrás és a vevő közti kapcsolatot.
- A megszüntetendő folyamat csomagjait egy alacsonyabb prioritású forgalmi osztályban továbbítjuk. A folyamat megmarad, de a korábban egyeztetett sáv szélességet már nem biztosítjuk neki.

2. Tulajdonság: A megsértett megállapodások számának minimalizálása érdekében *csak annyi folyamat szüntessünk meg, amennyi feltétlenül szükséges a torlódás feloldásához, és jelezzük a szélső csomópontoknak a folyamatok megszűnését.*

Néhány folyamat azért kell megszüntetni, hogy a maradéknak biztosítsuk a szükséges sáv szélességet. Ha a protokoll nem áldozna fel néhány folyamatot, mindegyik folyamat érintené a torlódás. Ha ez a torlódás túl hosszú, lehet hogy már nem elviselhető. Az Amerikai Nemzeti Szabványügyi Hivatal (ANSI) tanulmánya [19] szerint például a hangszolgáltatásnál 200 ms alatti kiesési idők nem rontják jelentősen a szolgáltatást, 200 ms és 2 másodperc közti kiesési idők már észrevehetőek, de csak minimális hatásuk van a szolgáltatásra. 2 másodperc kiesés esetén viszont az emberek elkezdik letenni telefonjaikat. Hasonlóképp, a multimédia lejátszó szoftverek is megszakítják az átvitelt, ha rossz minőségű átvitelt tapasztalnak.

3. Tulajdonság: Ahhoz, hogy a teljes kiesési idő 2 másodperc alatt maradjon, *a fölös folyamatokat a lehető leggyorsabban meg kell szüntetni, lehetőleg az átirányítást követő 1 másodpercen belül.*

4. Tulajdonság: A helyes engedélyezési döntések érdekében *az engedélyező eljárás tudjon az átirányított folyamatokról.*

5. Tulajdonság: Ahhoz, hogy képesek legyünk megőrizni az átirányított folyamatokat akár a normál engedélyezési limit felett is, *lehetővé kell tenni egy második, torlódást definiáló limitet $A_{\text{torl}} > A$, ami meghatározza azt a szintet, amely alatt a korábban beengedett folyamatokat nem szüntetjük meg.*

Formálisan, a 2.1. Definíció helyett akkor kezeljük a torlódást, ha $V_{\text{után}} > A_{\text{torl}}$, ahol $A_{\text{torl}} > A$. Ekkor a kezelendő torlódás mértéke $V_{\text{túlterh}} = V_{\text{után}} - A_{\text{torl}}$ értékre csökken.

3.1. *Példa.* Tegyük fel, hogy az operátor azt szeretné elérni, hogy a drága telefon forgalma, amelynek a link kapacitás 40%-át allokálta, átirányítás esetén mégis képes legyen akár 60%-ot is kihasználni, még akkor is, ha ez más forgalmi osztályokat hátrányosan érint. Ha C jelöli a link kapacitást, akkor az operátornak $A = 0,4C$ és $A_{\text{torl}} = 0,6C$ paraméterekkel kellene dolgoznia.

6. Tulajdonság: A hálózatüzemeltető megfelelő felügyelete érdekében konfigurálható szabályokkal lehessen kiválasztani, hogy mely folyamatok szűnjenek meg átirányítás után.

3.2. *Példa.* Lehet, hogy a sürgősségi telefonhívásokat szeretnénk preferálni más hangforgalmak helyett, vagy gazdasági megfontolások miatt inkább a kis- vagy nagysebességű folyamatokat szüntetjük meg inkább.

Az 1. Téziscsoportban megmutattam, hogy a folyamszintű állapotok elméletileg korlátozzák az alkalmazható engedélyező formulák halmazát. Ebben a részben megmutatom, hogy a belső csomópontok folyamszintű állapotai kihatással vannak az átirányítás utáni reakcióra is.

Az állapotgazdag protokollok, mint az RSVP, gyakran kötődnek a folyamatok külön-külön kezelő ütemezőkhöz. Az átirányított folyamatok csomagjait azonban csak az alapértelmezett „mindent megtesz” sorba teszi, hiszen az új útvonalon az átirányított folyamatok nem küldtek jelzést, így nem állították be az ütemezőket sem. Sajnos azonban sem a belépési csomópontot, sem a küldőt nem értesíti erről a minőségi változásról.

DiffServ hálózatokban, ahol az RMD-t is használnák, folyamszintű ütemezők nem képzelhetők el a belső csomópontokban, csak a belépési szélső csomópontban. A belső csomópontok a csomag fejlécben hordozott DiffServ kódpont alapján osztályok megkülönböztetésére alkalmasak. Ilyen módon az átirányított folyamatok a korábbi forgalmi osztályukba kerülnek az új útvonalon is, anélkül, hogy ott sáv szélességet foglaltak volna, és ezáltal az eredetileg is ott haladó folyamatokat is zavarhatják.

Mindkét esetben a hálózati engedélyezés sikertelennek tekinthető, mivel nem éri el célját, a folyamat torlódás-mentes átvitelét. Az RSVP-nél ez az átirányított folyamatokat érinti (mivel alacsony prioritással továbbítódnak az új úton), míg RMD-nél az átirányított folyamatok mellett az eredetileg ott haladó folyamatokat is.

A kedvezőtlen helyzet RMD-vel és RSVP-vel is helyreáll a puha állapotok frissítésének köszönhetően. A periodikus frissítő üzenetek ugyanis beállítják a nyilvántartott foglaltságokat a helyes értékre, és felhasználhatók arra, hogy megszüntessék a többletforgalmat. A következő tézisben azonban megmutatom, hogy a normál frissítési mechanizmus nem kielégítő.

A normális frissítés mellett az állapotgazdag protokollok rendelkeznek egy másik, sokkal gyorsabb mechanizmussal is, amelyet kifejezetten a folyamszintű állapotok tesznek lehetővé. Az RSVP-nél ezt helyi javításnak (local repair) hívják. A forgalmat átirányító router képes az átirányított folyamatokra azonnali frissítést küldeni, így nem kell kivárni a soron lévő normál frissítést. Azonban még a helyi javítás sem tökéletes:

3.2. TÉZIS – Puha állapotok frissítésén alapuló reakciók [J4]

Megmutattam, hogy a periodikus puha állapot frissítéseken alapuló mechanizmusok nem teljesítik az összes előbbi feltételt. RSVP esetén a 2., 5. és 6. tulajdonságok nem

teljesülnek. RMD esetén a 4., 5. és 6. nem teljesül. Sőt, a gyakorlatban használható 10-30 másodperces frissítési intervallummal egyik protokoll sem teljesíti még a 3. Tulajdonságot sem.

Az RSVP helyi javítás mechanizmusa már megfelelően gyors, teljesíti az 1–4. tulajdonságot, azonban az 5. és 6. tulajdonságokat még ez sem.

Habár az alap protokollok az 5. Tulajdonságot nem teljesítik, javasoltam egy egyszerű megoldást a már beengedett folyamat megőrzésére akár a normál beengedési küszöb fölött is.

3.3. TÉZIS – Az aktív folyamat megőrzése [J4]

A 3.1. Algoritmusból megoldást adtam az 5. Tulajdonság teljesítésére a frissítési mechanizmus segítségével mindkét protokollra. Átírányítás után az aktív folyamat átmenetileg nagyobb sávszélességet is elfoglalhatnak, de egyéb körülmények között a normál engedélyezési küszöbnél több folyamatot nem engedünk be.

3.1. Algoritmus.

- 1) Biztosítsuk, hogy az új folyamat foglalásai és a korábban beengedett folyamat frissítései megkülönböztethetők legyenek.³
- 2) Az eredeti A engedélyezési küszöb mellé állítsunk be a csomópontokba egy második küszöböt A_{torl} küszöböt.
- 3) Egy p rátájú frissítési üzenetre a normál engedélyezés helyett vizsgáljuk meg az alábbi feltételt:

$$\text{nyilvántartott foglalás} + p > A_{\text{torl}}$$

Ha igaz, jelöljük be a csomagban, hogy a folyamat (újra)engedélyezése sikertelen volt.

- 4) Ha a szélső csomópontok bejelölt frissítési üzenetet kapnak, megszüntetik a hozzájuk tartozó folyamatokat.

4.4. Az átírányítás kezelése aggregált állapotú protokollokkal

Az előző fejezet megmutattam, hogy az RMD beépített mechanizmusai nem elegendőek az átírányítás kezelésére, miközben az RSVP-nek a folyamszintű állapotoknak köszönhetően van egy gyors, majdnem tökéletes megoldása. Ezért megoldást kerestem és találtam az azonnali torlódás megoldására. Azonban ez nem az egyetlen probléma. A nem aktuális nyilvántartott foglalásokból adódó helytelen engedélyezés problémáját is meg akartam oldani.

³Az RMD a specifikáció szerint is megkülönbözteti a két üzenettípust. Az RSVP-hez vagy egy külön üzenettípust vagy a Resv üzenetbe egy egybites információs mezőt javasolok.

4. TÉZISCSOPORT – Az átirányítás kezelése

A túlterhelés problémájának megoldására javasoltam egy algoritmust, amely gyorsan és precízen megszünteti a kívánt mennyiségű folyamatot, megvédi a korábban beengedett folyamatokat akár a normál engedélyezési küszöb felett is, és megszüntetendő folyamatok kiválasztására szabályok használatát is lehetővé teszi.

Azt is megmutattam, hogy az átirányítás utáni tranziens alatt a normál engedélyezés felülbírálásával visszautasíthatók a másként hibásan beengedett folyamatok. Erre tiszta mérésen vagy a frissítési üzenetek alapján alapuló technikákat javasoltam.

A megoldásokat kombinálva, a 3.1. Tézis összes tulajdonsága teljesül.

Egy RMD-s belső csomópont, mivel nem tárol folyamat szintű információt, nem képes folyamatok megszüntetését kezdeményezni. Mivel a szélső csomópontok már ismerik a folyamatokat, ők meg tudnák szüntetni annyit, amennyi kell. Hogy mennyi kell, azt viszont a torlódással sújtott belső csomópont tudja, így ezt az információt kell eljuttatni a szélső csomópontokhoz.

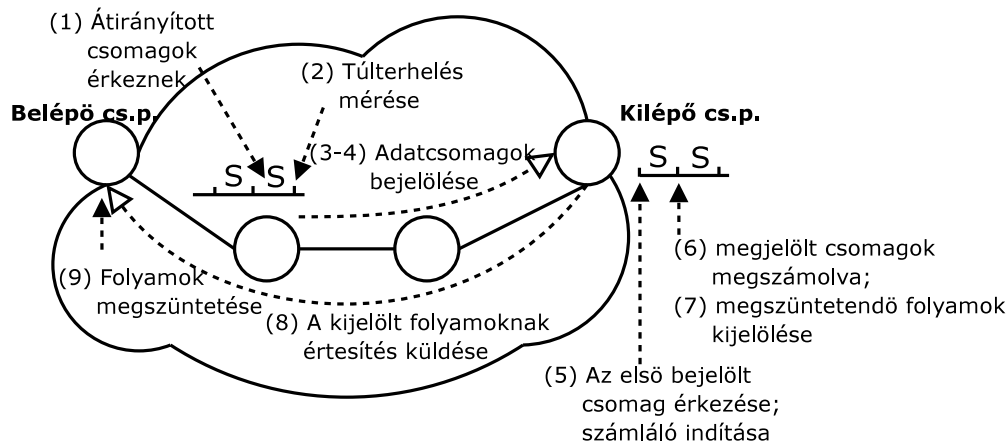
4.1. TÉZIS – Azonnali torlódás kezelése [O1, J4, C10, C8, C2, C3]

Bővítettem az RMD képességét, hogy a torlódás mértékét sávon belül jelezze a kilépő szélső csomópontnak, és javasoltam egy algoritmust (4.1. Alg.), amellyel a küszöb feletti folyamat meg lehet szüntetni az 1. és 5. Tulajdonságnak megfelelően. Sőt, a folyamatok kijelölésekor szabályok alkalmazhatók, azaz a 6. Tulajdonság is teljesül.

4.1. Algoritmus (Torlódás azonnali kezelése – lásd az 1. ábrát).

- 1) Átirányított csomagok érkeznek egy belső csomópontra.
- 2) A belső csomópont érzékeli és megbecsüli a túlterhelés mértékét ($\hat{V}_{\text{túlterh}}$) S hosszú mérési periódussal, amely konstans az egész hálózat.
- 3) Ha egy mérési periódus végén $\hat{V}_{\text{túlterh}} > 0$, akkor számítsuk ki a periódusba már be nem férő adatmennyiséget: $B_{\text{túlterh}} = \hat{V}_{\text{túlterh}} \cdot S$.
- 4) A következő mérési periódus alatt a belső csomópont megjelöl néhány adatcsomagot annak érdekében, hogy *i)* azonosítsa a torlódott linken áthaladó csomagokat, és hogy *ii)* tudassa a kilépsi szélső csomóponttal torlódás becsült mértékét.

Egy csomagnak 3 jelölési állapota lehetséges: *jelöletlen*, *méret-info* és *érintett*. A *méret-info* jelölt csomagok segítségével a belső csomópont belekódolja az adatcsomagokba a torlódás mértékét és ezáltal „sávon belül” jelzi ezt a szélső csomópontnak. A belső csomópont annyi induló csomagot jelöl meg *méret-info*val, amennyinek az összege kiad B_{overload} byte-ot. Minden más csomagot, amíg fennál a torlódás, *érintettnek* jelöl. Ha a szélső csomópont kap egy *méret-info* vagy egy *érintett* csomagot, tudni fogja, hogy a hozzá tartozó folyamat torlódáson haladt át, így egy lehetséges megszüntetendő folyamat.



1. ábra. Az RMD torlódáskezelő mechanizmusának alapkonceptiója

- 5) Az első jelölt csomag érkezésekor, minden $e \in E$ kilépési csomópont S hosszú ideig megszámlolja az általa kapott *méret-info* csomagok össz méretét, B_e -t, és közben nyilvántartja a potenciális megszüntethető folyamatok halmazát.
- 6) A számolási periódus végén az e kilépési csomópont a számlálóból kiszámítja a túlterhelést okozó sávszélességet: B_e/S .
- 7) Ha $B_e/S > 0$, akkor az e kilépési csomópont kiválaszt egy folyamhalmazt, amelyek együttesen ekkora forgalmat generálnak.
- 8) A szélső csomópont minden megszüntetendő folyamatot jelez a megfelelő belépési csomópontnak.
- 9) A belépési csomópont megszünteti a kívánt folyamatokat.

4.1. *Észrevétel.* A csomagok megjelölése a 4) lépésben többféle képpen is történhet. Felhasználható például az IP fejléc két kihasználatlan bite (3 állapot kódolásához legalább 2 bit kell), esetleg az ECN bitek [20]. Lehetséges az is, hogy minden forgalmi osztálynak a normál kódpontja mellett lefoglalunk két másik kódpontot is, és a kódpont megváltoztatásával jelöljük meg a csomagot. A gyakorlatban viszont nehéz szabad IP biteket találni, de sok kódpont allokálása is nehéz. Az algoritmusom azonban működik 3 helyett 2 (*jelöletlen* és *méret-info*) állapottal is. Ekkor már csak 1 szabad bit vagy 1 extra kódpontra van szükség, de a szélső csomópontok még így is pontos képet kapnak a torlódásról. A hátrány ilyenkor abban van, hogy a kilépési csomópont nem ismervén az összes érintett folyamat, csak egy kisebb halmazból tud megszüntetendő folyamatot választani.

4.2. *Észrevétel.* A jelölési eljárás gyakorlatilag jelzési üzenetet ültet a felhasználók csomagjaira, ezért érdemes biztosítani, hogy ezen csomagok ne vesszenek el. Ha például egy *méret-info* csomag elveszik, kevesebb folyamat dobunk el az adott körben, ami lassítja a torlódás feloldását.

4.3. *Észrevétel.* Algoritmusom alkalmazásakor a 3.3. Tézis továbbra is érvényes: a frissítési üzenetekre egy magasabb korlát lehet érvényes, mint új hívásokra. Emellett az is könnyen biztosítható, hogy a belső csomópontban csak a magasabb szint felett jelezzen torlódást. Így az 5. Tulajdonság teljesül.

4.4. *Észrevétel.* A 6. Tulajdonság teljesül, mert a 7) lépésben a folyamatok többféle elv szerint is választhatjuk, például: véletlen módszerrel; adatsebesség alapján (több kisebb sebességű folyamat szüntetünk meg vagy kevesebb nagysebességűt); jogi kötelezettségek alapján (például a sürgősségi és katonai célú folyamatok megtartása); egyéb megfontolások (például gazdasági megfontolások alapján azonos forgalmi osztályon belül is lehetnek „jobban fizető” folyamatok).

4.2. TÉZIS – Az algoritmus válaszideje [C2, C3, J4]

Megmutattam, hogy a 4.1. Algoritmusom, ha bejelölt csomagok nem vesznek el, a torlódást az első átirányított csomag megjelenésétől számított $3S + RTT_{max}$ időn belül feloldja, ahol RTT_{max} jelöli a hálózat maximális körbefordulási idejét. Így, ha S -et megfelelően kicsire választjuk, teljesül a 3.1. Tézisben megfogalmazott 3. Tulajdonság.

Ismeretes, hogy a torlódáskezelés egy szabályozástechnikai probléma [21]. A 4.1. Algoritmusom is egy késleltetett visszacsatoláson alapul: a torlódás mérését és a bejelölést a belső csomópont, a folyamatok kijelölését a kilépési csomópont, de a folyamatok megszüntetését a belépési csomópont végzi. Az első csomag megjelenésétől számítva a kör ideje az út idejéből a belső csomóponttól a kilépési csomópontig, a kilépési csomópont S számolási intervallumából, és a kilépési-belépési csomópontok közti útból áll. Továbbá, a túlterhelés csökkenéséig még egy kis időnek el kell telnie, a belépési és a belső csomópont közti út idejéből. Mindezt figyelembe véve, a belső csomópont jelzése nem változtat a túlterhelésen $S + RTT$ ideig.

A klasszikus szabályozástechnikában könnyű a késleltetett visszacsatolás miatt túlreagálást demonstrálni. A 4.1. Algoritmusban is megjelenik ez a túlszabályozás, ha nem teszünk további lépéseket, hiszen jóllehet a szükséges mennyiségű folyamat megszüntetése folyamatban van, a belső csomópont mégis folyamatosan jelzi a túlterhelést a kilépési pontnak. Ez azt eredményezi, hogy több folyam szűnik meg, mint kellene, amit a linkkihasználtsági ábrákon egy a célszintnél alacsonyabb aláesés formájában figyelhetünk meg.

4.3. TÉZIS – Túlreagálás elkerülése [C2]

A visszacsatolási késleltetésből adódó túllövés megoldására javasoltam egy módszert, amely a belső csomópontokban alkalmazott pár cellából álló memóriákon alapszik, amelyek segítségével meg lehet becsülni a torlódás még nem jelzett részét. Megmutattam, hogy egy $(1 + \lceil \frac{RTT}{S} \rceil)$ darab cellából álló memóriával a túllövés eliminálható.

4.2. Algoritmus (Túllövés mentes torlódáskezelés).

A 4.1. Algoritmust kiegészítve tartjuk nyilván a belső csomópontokban a korábban jelzett forgalmat néhány mérési periódusra visszamenőleg. Egy mérési periódus végén, mielőtt a mért

értéket elkezdene bejelölni a csomagokon, a mért értéket csökkentjük a memóriacellákban nyilvántartott korábbi mennyiségekkel, mivel ezek kezelése már folyamatban van. Az új jelzett értéket csúszóablakos módszerrel a legrégebbi cella helyére töltjük.

A fenti módszerekkel az azonnali torlódás megoldható, de a hibás folyamengedélyezésre még mindig gondolni kell.

4.4. TÉZIS – Felülbírálat tiszta mérés alapú engedélyezéssel és mozgóátlaggal [J4, C3, P6]

Az RMD tárolt állapotokon alapuló engedélyező mechanizmusát kiegészítettem egy tisztán mérés alapú engedélyezéssel is, amely az átirányítások utáni tranziensekben felülbíráhatja a normál engedélyezés döntéseit. A kiegészítés bekapcsolásához egy laposabb és egy gyorsabb mozgóátlag alapú mérések kombinációján alapuló mechanizmust javasoltam (4.3. Algoritmus), amely gyorsan bekapcsol hirtelen nagy változásokra, de nem kapcsol be a forgalom normális fluktuációjára.

A hálózat normális működése során a foglaltság alapú engedélyezést azért használják, mert akkor is garantálja egy ügyfél által lefoglalt erőforrásokat, ha éppen nem használja a hálózatot.

4.5. *Észrevétel.* Egy hagyományos csúszóablakos vagy exponenciális mozgóátlag segítségével a mért sávszélességek ingadozása laposítható. A paraméter beállításoktól függ, hogy vagy gyorsan követik a változásokat (ekkor kevésbé lapos, kis változásra is reagál) vagy erősebben kiátlagolják a mintákat, de ekkor lassan reagálnak a nagy változásokra is.

4.3. Algoritmus ([P6]).

1) Használjunk egy exponenciális mozgóátlagot w paraméterrel, amely kellően lapos ahhoz, hogy a forgalom normál fluktuációját kilapítsa.

2) Minden mérési periódus végén, amikor egy új s minta értéke rendelkezésre áll, ellenőrizzük, hogy a minta jelentősen különbözik-e az utoljára számított átlagtól. Ha igen, az új érték kapjon nagyobb súlyt ($w_a > w$, $w_a \approx 1$), hogy átlag gyorsan adaptálódjon az új szinthez:

Ha $|s - \text{átlag}_{i-1}|$ jelentős Akkor

$$\text{átlag}_i = \text{átlag}_{i-1} \cdot (1.0 - w_a) + m_i \cdot w_a$$

Egyébként

$$\text{átlag}_i = \text{átlag}_{i-1} \cdot (1.0 - w) + m_i \cdot w$$

HaVége

A „jelentős” azt jelenti, hogy az új érték és az előző átlag közti eltérés abszolút értéke nagyobb, mint egy fix vagy relatív küszöb.

Adtam alternatív megoldásokat is, amelyek képesek a hibás engedélyezést megoldani kizárólag az RMD jelzések alapján. Ez akkor jöhet jól, ha az operátor nem tud vagy nem akar egy párhuzamos mérés alapú engedélyezést futtatni.

4.5. TÉZIS – Felülbírálás a frissítések alapján [J4, C3]

Javasoltam egy módszert az átirányított folyamat érzékelésére a frissítő üzenetek számának jelenlegi és az egy frissítési ablakkal korábbi értékének összehasonlításával. Javasoltam egy i) mohó megoldást, amely az érzékelés után nem enged be egyetlen új folyamat sem egy teljes frissítési intervallumig; ii) és javasoltam egy algoritmust (4.4. Alg.) is, amely megbecsüli az átirányított forgalom nagyságát a frissítési üzenetekből, és ez alapján javítja az engedélyezési döntést.

A mohó algoritmus egy teljes frissítési intervallumig minden új kérést visszautasít, hiszen legrosszabb esetben lehetséges, hogy egy folyamat csak ekkor frissíti foglalását az új útvonalon, így csak ekkor lesznek a foglaltsági állapotok tökéletesek.

4.4. Algoritmus (Becslés frissítésekből [J4, C3]).

Az RMD-ben a hálózat széli belépési csomópontok a folyamat foglalásait periodikus frissítik (alapbeállításban $R = 30$ másodpercenként). A belső csomópontok időzítői viszont rendre R/N időközönként jeleznek, ahol alapesetben $N = 10$, ami a teljes R frissítési ablakot N darab kisebb időintervallumra, úgynevezett cellára osztja. Ha a belső csomópontban egy R hosszú időszakot tekintünk, minden aktív folyamat foglalása vagy frissítése egy cellában található. Minden cellának megfelel egy számláló, C^i , $i = 1$ to N , amely a cellában foglalt vagy frissített sávcsomagtartást tartja nyilván. Minden cella végén az aktuális számlálót bemásolja C^0 -ba, és az ablak egy cellát lép előre úgy, hogy C^N tartalmazza a legrégibbi cellát.

Ehhez az eredeti mechanizmushoz képest azt javasoltam, hogy az aktív cellában a frissítések és az új foglalások összegét külön számláló tartsa nyilván, jelölje az előbbi $refcount$. Ha ϵ egy tolerancia szintet jelöl a hálózati jitter miatt fellépő késő vagy korai frissítésekben, akkor

$$\text{ha } (C^N + \epsilon < refcount) \Rightarrow \text{átirányítás történt.}$$

Azaz, ha egy cella végén a frissített sávcsomagtartás magasabb mint az egy ablakkal korábbi cella értéke, akkor átirányítás történt, és a különbség (jelölje d_i) megfelel az átirányított folyamat azon részhalmazának aki épp az adott cellában frissít, azaz $d_i = refcount - C^N$. Kumuláljuk az ilyen módon észrevett többletforgalmat az i -edik cella után az A változóban. Minden cella végén:

$$A_i \leftarrow A_{i-1} + d_i$$

Az i -edik cella végén a maradék $(N - i)$ cellába tartozó folyamatok nem frissítettek még. Ha az időben egyenletes eloszlásúnak tekintjük a frissítések érkezését, akkor ennek ez a maradék $\frac{N-i}{i} \cdot A_i$ formulával becsülhető, így az átirányított folyamat teljes sávcsomagtartása $A_i + \frac{N-i}{i} A_i$ képlettel becsülhető, és a becslés az idő előrehaladtával celláról cellára egyre pontosabb.

4.1. Következmény.

A javasolt megoldásokat alkalmazva minden elvárásom teljesül: az azonnali torlódás gyorsan és precízen feloldható (1-3. tulajdonságok); a hibás folyamat-engedélyezést megakadályozható (4. Tulajdonság); szükség esetén a már beengedett folyamatok megőrizhetők (5. Tulajdonság); a szélső csomópontok pedig szabály alapján választhatják ki a megszüntetendő folyamatokat (6. Tulajdonság).

5. Összegzés

Disszertációmban megmutattam, hogy az aggregált állapotok elméletileg korlátozzák az alkalmazható folyam-engedélyező formulák halmazát. Azt is megmutattam azonban, hogy több, az irodalomban publikált formula is alkalmazható csökkentett állapotterű protokollokkal, és feltételeket is adtam a csökkentett állapotterrel való kompatibilitásra.

Egyik fontos eredményem, hogy az egyszerűbb RMD protokoll a blokkolási arány, a jelzési idők és a jelzési többletforgalom szempontjából legalább olyan jól teljesít, mint az RSVP.

Azonosítottam az aggregált állapotú protokollok hátrányát hálózati átirányítások esetén, és megoldásokat adtam ezen problémák megoldására. Javasoltam egy módszert a többletforgalom visszaszabályozására és megoldásokat javasoltam az átirányítás utáni tranziens alatt a helyes folyam-engedélyezés biztosítására is.

A módszereimmel a csökkentett állapotterű protokollok, mint az RMD, képesek kezelni az átirányítás negatív hatásait, gyorsan helyre tudják állítani a lehető legtöbb folyam minőségét és helyes folyam-engedélyezést végeznek.

6. Eredményeim alkalmazhatósága

Munkámat felhasználták az IETF NSIS munkacsoportjában szabványosítás alatt álló RMD QoS modellben [O1]. Az RMD-vel kapcsolatos ötleteim és megoldásai 7 szabadalmi bejelentés alapját képezik, szimulátoromat pedig különböző munkák kapcsán műegyetemi kollégáim mellett a Ciprusi Egyetem [22], a holland Twente Egyetem [23] és az Ericsson Research is felhasználta és továbbfejlesztette.

Habár eredményeim az RMD-re fókuszálnak, egy részük ennél általánosabb érvényű. Az 1. Téziscsoport belső eredményeim a csökkentett állapotterrel kompatibilis folyam-engedélyező formulákról bármilyen környezetre érvényes, ahol nem engedünk meg folyam szintű állapotokat. A 4. Téziscsoportban leírt megoldásaim a 4.5. Tézis kivételével csak annyit tételeznek fel, hogy *a)* a belső csomópontok nem képesek egyedi folyamatok azonosítására és *b)* a szélső csomópontok rendelkeznek folyamszintű ismeretekkel, így ezek a módszerek bármilyen ezeket a tulajdonságokat teljesítű, azaz DiffServ, protokollal működnek.

Az RMD/NSIS protokollban a torlódáskezelés csak egy, bár fontos, részprobléma. Egy nemrég megalakult másik IETF munkacsoport, a Torlódás (előre)jelzés (PCN – Pre-Congestion Notification), azonban kifejezetten ezzel a problémakörrel foglalkozik, és így itt is alkalmazhatók eredményeim. A PCN problémát leíró dokumentuma [24] szabadfordításban így fogalmaz:

„...Míg a folyam-engedélyezés gondoskodik a QoS normális működési körülmények közti fenntartásáról, egy további mechanizmus szükséges bizonyos folyamatok QoS-ének megszüntetésére erős torlódások esetén (amelyet például link vagy csomóponthiba okozta útvonal változások eredményeznek).”

A munkacsoport saját magáról adott leírása ekként fogalmaz⁴:

„A PCN munkacsoport az aktív nem elasztikus folyamatok QoS védelmére dolgoz ki megoldásokat egy torlódás sújtotta torlódás előtt álló DiffServ hálózatban. Ezeket a módszereket a tartományok szélén alkalmazzák a tartomány belsejéből származó aggregált, a túlterhelést leíró információk alapján.”

A munkacsoport tehát az általam is leírt torlódással foglalkozik egy aggregált állapotú belső csomópontokkal rendelkező DiffServ hálózatban.

„A munkacsoport fő célja szabványok kidolgozása, a belső csomópontok csomagjelölési módszereire, a túlterhelés kódolására és szállítására. . . , a túlterhelés információ kinyerésére a kilépési csomópontban, illetve az információ visszajuttatására a belépési csomópontban.”

E szerint a munkacsoport az alapműködését tekintve az általam javasolt 4.1. Algoritmushoz hasonló megoldáson gondolkodik, amely megoldást ad a túlterhelés mértékének kódolására és kilépési csomópontba szállítására, a szélső csomópontban történő visszafejtésére, és a belépési csomópontba küldésére a megfelelő válaszlépés végrehajtására.

Kutatási eredményeim tehát ebben a munkacsoportban is hasznosnak bizonyulhatnak.

Köszönetnyilvánítás

Számos embernek tartozok hálával, amiért munkámban valamilyen módon segített. Kiemelném Takács Attilát, akinek köszönet a sokévi közös munkáért, és Szabó Róbertet, iránymutatásáért és a segítségéért a folyamat végén is. Számos további kollégám külön köszönetet érdemel, mivel értekezésemhez kötődő munkámban segített: köszönet Németh Feliciánnak az első téziscsoport-hoz nyújtott segítségéért és ellenőrzéséért; Antal Csabának, Harmatos Jánosnak és Heszberger Zalánnak a téziszövegem többszöri átnézéséért; Báder Attilának konstruktív javaslataiért és az IETF fejlemények továbbadásáért; és Barta „Pöpi” Péternek a disszertációm kiindulópontját jelentő folyóiratcikk [J4] rigorózus véleményezéséért és konstruktív javaslataiért.

Köszönet Mihály Attilának, Papp Oktáviánnak és Rétvári Gábornak szakmai és tudományos hozzáállásom egyéni szemléletükből adódó formálásáért.

A legnagyobb köszönettel azonban Katinak, szüleimnek, testvéremnek és nagyszüleimnek tartozom, akik képzeletet meghaladó türelemmel támogattak és ösztönöztek. Nélkülük soha semm jutottam volna idáig.

Kutató munkámat a TrafficLab (Ericsson Research), illetve az egyetem Távközlési és Média-Informatikai Tanszékének Nagysebességű Hálózatok Laboratóriuma támogatta.

⁴Forrás: <http://www.ietf.org/html.charters/pcn-charter.html>

Hivatkozások

- [1] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, „Packet-level traffic measurement from the Sprint IP backbone,” *IEEE Network Magazine*, Nov 2003.
- [2] Lee W. McKnight and Joseph P. Bailey, Eds., *Internet Economics*, The MIT Press, 1998.
- [3] Clarence Filstils and John Evans, „Engineering a multiservice IP backbone to support tight SLAs,” *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 40 Special issue: Towards a new internet architecture, pp. 131–148, Sept. 2002.
- [4] Michael Menth, Rüdiger Martin, and Joachim Charzinski, „Capacity overprovisioning for networks with resilience requirements,” in *Proc. of SigComm’06*, Pisa, Italy, Sept. 2006.
- [5] „Internet 2 QBone Bandwidth Broker Advisory Council,” homepage: <http://qos.internet2.edu/qbone/QBBAC.shtml>.
- [6] K. Nichols, V. Jacobson, and L. Zhang, „A Two-bit Differentiated Services Architecture for the Internet,” RFC 2638 (Informational), July 1999.
- [7] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, „Resource reservation protocol (RSVP) – version 1 functional specification,” RFC 2205, IETF, Sept. 1997.
- [8] „Next Steps in Signaling (nsis),” IETF Working Group web page, <http://www.ietf.org/html.charters/nsis-charter.html>.
- [9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, „An architecture for differentiated services,” RFC 2475, IETF, Network WG, Dec. 1998.
- [10] Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, and Christophe Diot, „Characterization of failures in an IP backbone,” in *Proceedings of IEEE InfoCom*, Hong Kong, Mar. 2004, IEEE.
- [11] D. Watson, C. Labovitz, and F. Jahanian, „Experiences with monitoring OSPF on a regional service provider network,” in *Proc. International Conference on Distributed Computing Systems*, 2003, pp. 204–213.
- [12] Sundar Iyer, Supratik Bhattacharyya, Nina Taft, and Christophe Diot, „An approach to alleviate link overload as observed on an IP backbone,” in *Proceedings of IEEE InfoCom*, San Francisco, CA, USA, Mar. 2003, IEEE.
- [13] Kevin Fall and Kannan Varadhan, *ns Manual*, UC Berkeley.
- [14] Goergios Karagiannis, Simon Oosthoek, and Martin Jacobson, „Maintenance of sliding window aggregated state using combination of soft state and resource release principles,” International Patent Application, WO 02/076035 A1, PCT/EP02/01856, Sept. 2002.

- [15] Sally Floyd, „Comments on measurement-based admissions control for controlled-load services,” Tech. Rep., Lawrence Berkeley National Laboratory, July 1996.
- [16] Harry G. Perros and Khaled M. Elsayed, „Call admission control schemes: A review,” *IEEE Communications Magazine*, vol. 34, no. 11, pp. 82–91, Nov. 1996.
- [17] Lee Breslau, Sugih Jamin, and Scott Shenker, „Comments on the performance of measurement-based admission control algorithms,” in *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, Tel-Aviv, Israel, Mar. 2000.
- [18] Zoltán R. Turányi and Lars Westberg, „Load control: Congestion notifications for real-time traffic,” in *9th IFIP Working Conference on Performance Modelling and Evaluation of ATM and IP Networks*, Budapest, Hungary, July 2001.
- [19] „Technical report on enhanced network survivability performance,” Tech. Rep. T1.TR.68, ANSI, Feb. 2001.
- [20] S. Floyd, „Specifying alternate semantics for the explicit congestion notification (ECN) field,” RFC 4774, IETF, Network WG, Nov. 2006.
- [21] Andreas Pitsillides, Petros Ioannou, Marios Lestas, and Loukas Rossides, „Adaptive non-linear congestion controller for a differentiated-services framework,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 94–107, Feb. 2005.
- [22] Costas Djouvas, „Extending diffserv architecture: Integration of idcc and rmd framework,” M.S. thesis, University Of Cyprus, Department Of Computer Science, May 2003.
- [23] Gerjan Stokkink, „Performance evaluation of severe congestion handling solutions for multilevel service in rmd domains,” in *Proceedings of the 4th Twente Student Conference on Information Technology (TSC on IT)*, Twente, The Netherlands, Jan. 2006, pp. 89–98.
- [24] K. Chan, A. Charny, and P. Eardley, „Pre-congestion notification problem statement,” Internet Draft draft-chan-pcn-problem-statement-01, IETF, Oct. 2006, Work in progress!

Publikációim

Folyóirat cikkek

- [J1] András Császár, Gábor Enyedi, Markus Hidell, Gábor Rétvári, and Peter Sjödin, „Converging the evolution of router architectures and IP networks,” *IEEE Network Special Issue on Advances in Network Systems Architecture*, July-August 2007.
- [J2] András Zahemszky, András Császár, Gábor Tóth, and Attila Takács, „Átjáró szerver választás a GMPLS PCE architektúrában,” *Híradástechnika*, Apr. 2007, in Hungarian.

- [J3] Attila Takács, András Császár, Róbert Szabó, and Tamás Henk, „Generic multipath routing concept for dynamic traffic engineering,” *IEEE Communications Letters*, vol. 10, no. 2, pp. 126–128, Feb. 2006.
- [J4] András Császár, Attila Takács, Róbert Szabó, and Tamás Henk, „Resilient reduced-state resource reservation,” *Journal of Communications and Networks*, vol. 7, no. 4, pp. 509–524, Dec. 2005.
- [J5] Attila Takács, András Császár, Róbert Szabó, and Tamás Henk, „Forgalommenedzsment többszörös kapcsolatú tartományoknál,” *Híradástechnika*, vol. LIX, no. 2004/9, pp. 19–25, Sept. 2004, in Hungarian.
- [J6] András Császár, Csaba Lukovszki, and Róbert Szabó, „CBQ alkalmazása differenciált szolgáltatásokhoz 3-ik generációs mobil rendszerekben,” *Híradástechnika*, vol. 11, pp. 27–34, Dec. 2001, in Hungarian.
- [J7] András Császár, Attila Takács, and Róbert Szabó, „VoIP szolgálatok minőségbiztosítása,” *Magyar Távközlés*, vol. 5, pp. 14–17, May 2000, in Hungarian.

Konferencia cikkek

- [C1] András Zahemszky, András Császár, Gábor Tóth, Attila Takács, and Tibor Cinkler, „Dual purpose gateway selection in the GMPLS PCE architecture,” in *Proceedings of Transcom2007*, Zilina, Slovak Republic, June 2007.
- [C2] András Császár, Attila Takács, and Attila Báder, „A practical method for the efficient resolution of congestion in an on-path reduced-state signalling environment,” in *Proceedings of the Thirteenth International Workshop on Quality of Service (IWQoS 2005)*, Passau, Germany, June 2005, number 3552 in Lecture Notes in Computer Science, pp. 282–293.
- [C3] András Császár, Attila Takács, Róbert Szabó, and Tamás Henk, „State correction after re-routing with reduced state resource reservation protocols,” in *Proceedings of IEEE Global Telecommunications Conference (Globecom2004)*, Dallas, TX, USA, Dec. 2004, IEEE.
- [C4] Attila Takács, András Császár, József Bíró, Róbert Szabó, and Tamás Henk, „Path integrity aware traffic engineering,” in *Proceedings of IEEE Global Telecommunications Conference (Globecom2004)*, Dallas, TX, USA, Dec. 2004, IEEE.
- [C5] Róbert Szabó, Attila Takács, and András Császár, „Optimised multi homing – an approach for inter-domain traffic engineering,” in *Proceedings of the 2nd International Workshop on Inter-Domain Performance and Simulation (IPS2004)*, Budapest, Hungary, Mar. 2004, pp. 48–57.

- [C6] Georgios Karagiannis, Attila Báder, Gergely Pongrácz, András Császár, Attila Takács, Róbert Szabó, and Lars Westberg, „RMD – a lightweight application of NSIS,” in *Proceedings of the 11th International Telecommunications Network Strategy and Planning Symposium (Networks2004)*, Vienna, Austria, June 2004, pp. 211–216.
- [C7] Attila Takács, András Császár, Róbert Szabó, and Tamás Henk, „Examination of free capacity based load sharing,” in *Proceedings of the 2nd IASTED International Conference on Communications, Internet, & Information Technology (CIIT2003)*, Scottsdale, AZ, USA, Nov. 2003.
- [C8] András Császár and Attila Takács, „Comparative performance analysis of RSVP and RMD,” in *Proceedings of the Fourth COST 263 International Workshop on Quality of Future Internet Services QoFIS2003*, Stockholm, Sweden, Oct. 2003, vol. 2811 of *Lecture Notes in Computer Science*, pp. 41–51, Springer Verlag.
- [C9] Attila Takács, András Császár, Róbert Szabó, and Tibor Cinkler, „Thrifty traffic engineering through CSLLS,” in *Proceedings of the 18th International Teletraffic Congress ITC18*, Berlin, Germany, Sept. 2003, vol. 5 of *Teletraffic Science and Engineering*, pp. 61–70, Elsevier.
- [C10] András Császár, Attila Takács, Róbert Szabó, Vlora Rexhepi, and Georgios Karagiannis, „Severe congestion handling with resource management in diffserv on demand,” in *Proceedings of Networking 2002 – The Second Intl. IFIP-TC6 Networking Conference*, Pisa, Italy, May 2002, vol. 2345 of *Lecture Notes in Computer Science*, pp. 443–454, Springer Verlag.
- [C11] Lars Westberg, András Császár, Georgios Karagiannis, Ádám Marquetant, David Partain, Octavian Pop, Vlora Rexhepi, Róbert Szabó, and Attila Takács, „Resource management in diffserv (RMD): A functionality and performance behavior overview,” in *Proceedings of PfHSN'2002 – Seventh International Workshop on Protocols For High-Speed Networks*, Berlin, Germany, Apr. 2002, vol. 2334 of *Lecture Notes in Computer Science*, pp. 17–34, Springer Verlag.
- [C12] András Császár, Csaba Lukovszki, and Róbert Szabó, „A differentiated services approach using CBQ for 3G communication,” in *Proceedings of the Polish-Czech-Hungarian Workshop on Circuit Theory, Signal Processing and Telecommunication Networks*, Budapest, Hungary, Sept. 2001, pp. 274–286.
- [C13] András Császár, Attila Takács, Csaba Lukovszki, and Róbert Szabó, „Simulation study over IP based GSM backbone,” in *Proceedings of the IEEE International Conference on Telecommunications - IEEE ICT2001*, Bucharest, Romania, June 2001.

Szabadalmi bejelentések

- [P1] András Császár, Attila Mihály, and Oktávián Papp, „Enhanced fast re-route for bandwidth-efficient link protection,” International Patent Application PCT/EP2007/057515, July 2007.

- [P2] Gábor Enyedi and András Császár, „Loop-free fast interface based rerouting,” International Patent Application PCT/EP2007/057322, July 2007.
- [P3] Attila Báder and András Császár, „Priority flow handling in stateless IP network domains,” International Patent Application PCT/EP2007/055178, May 2007.
- [P4] Alpár Jüttner, András Császár, and Attila Mihály, „Concept and method for reducing the forwarding tables in network routers,” International Patent Application PCT/EP2007/051220, Feb. 2007.
- [P5] Ferenc Pintér, Attila Báder, András Császár, and Attila Takács, „Explicit congestion control method for stateless domains,” International Patent Application PCT/EP2006/068317, Nov. 2006.
- [P6] András Császár and Attila Báder, „A method for fast traffic measurement and monitoring,” International Patent Application PCT/IB2006/003405, Nov. 2006.
- [P7] Attila Báder, András Császár, and Attila Takács, „Stateless congestion control,” International Patent Application PCT/EP2006/010820, nov 2006.
- [P8] András Császár, Attila Báder, and Attila Takács, „A method for handling inter-domain re-routing between stateless quality-of-service domains,” International Patent Application, PCT/IB2006/050990, Mar. 2006.
- [P9] András Császár, Attila Takács, and Attila Báder, „A method and system for aggregated resource reservation in internet protocol networks using generalised sliding window algorithm,” International Patent Application, Aug. 2005, PCT/EP2005/009094, WO-2007022789 A1.
- [P10] András Császár, Attila Takács, Róbert Szabó, Attila Báder, and Lars Westberg, „Algorithms for fast handling server congestion in an IP network using differentiated services,” International Patent Application, Nov. 2004, PCT/SE2004/001657, WO 2006/052174 A1.

Más publikációk

- [O1] Attila Báder, Lars Westberg, Georgios Karagiannis, Cornelia Kappler, Tom Phelan, Attila Takács, and András Császár, „RMD-QOSM - the Resource Management in Diffserv QoS model,” Internet Draft draft-ietf-nsis-rmd-12, IETF, Nov. 2007, Work in progress!
- [O2] András Császár, Csaba Antal, Attila Mihály, and Árpád Szlávik, „Carrier-grade resilience in multi-service IP networks,” in *Proceedings of the High Speed Networking 2005 Spring Workshop*, may 2005, pp. 26–28.
- [O3] András Császár, „Az analitikus hierarchikus eljárás alkalmazása befektetési döntéstámogató rendszerben,” M.S. thesis, Budapest University of Technology and Economics, Dept. of Information and Knowledge Management, Dec. 2004, in Hungarian.

- [O4] Attila Takács, András Császár, Róbert Szabó, and Tamás Henk, „Re-routing in IP networks from the aspect of reduced state resource reservation,” in *Proceedings of the High Speed Networking 2004 Spring Workshop*, May 2004, pp. 102–105.
- [O5] Attila Takács, András Császár, József Bíró, Róbert Szabó, and Tamás Henk, „Path integrity aware traffic engineering,” in *Proceedings of the High Speed Networking 2004 Spring Workshop*, May 2004, pp. 30–33.
- [O6] András Császár, Attila Takács, Attila Báder, Róbert Szabó, and Csaba Antal, „RMD: Proof of concept,” Internal Technical Report ETH/RT-2004:0011, Ericsson, Jan. 2004.
- [O7] András Császár, Attila Takács, and Róbert Szabó, „Severe congestion handling with resource management in diffserv on demand,” in *Proceedings of the High Speed Networking 2002 Spring Workshop*, May 2002, pp. 122–129.
- [O8] András Császár, „Differentiated services for voice communication,” M.S. thesis, Budapest University of Technology and Economics, May 2001.
- [O9] András Császár and Attila Takács, „Voice traffic control study over IP based GSM backbone,” Tech. Rep., Budapest University of Technology and Economics, Nov. 2000, Essay for Scientific Student Conference.
- [O10] András Császár and Attila Takács, „VoIP szimuláció network simulator-ral,” Tech. Rep., Budapest University of Technology and Economics, Nov. 1999, Essay for Scientific Student Conference, in Hungarian.

Disszertáció

- [D] András Császár, *Reduced-State Resource Reservation*, Ph.D. thesis, Budapest University of Technology and Economics, Department of Telecommunication and Media Informatics, 2007.