HÍVÁSENGEDÉLYEZÉS ÉS FORGALOMSZABÁLYOZÁS AZ ÁLTALÁNOS KISZOLGÁLÓ-MEGOSZTÁSI MODELLEN ALAPULÓ ÜTEMEZŐKBEN

Ph.D. tézisfüzet

Írta:
Szabó Róbert

Tudományos konzulensek:

Dr. Trón Tibor Dr. Bíró József
Budapesti Műszaki és Gazdaságtudományi Egyetem
Távközlési és Telematikai Tanszék

Beadva a
DOCTOR OF PHILOSOPHY
fokozat részleges követelményeként a
BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM-re
BUDAPEST, MAGYARORSZÁG
2000

1 Bevezetés

A modern, nagysebességű, különböző szolgáltatásokat nyújtó távközlő hálózatok egyik központi kérdését a szolgáltatásminőség /QoS - Quality of Service/ garantálása jelenti. A kutatók a QoS hálózatok égisze alatt már számos új módszert fejlesztettek ki és vizsgáltak meg. Ezek közül az egyik legfontosabb csoportot a különálló hálózati berendezések (kapcsolók /switch/, forgalomirányítók /router/) forgalomütemezési algoritmusainak családja alkotja [BZ94, BZ96, CSZ92, Gol90, Gol94, Zha95]. A disszertációban végig a berendézesek egyetlen, független ütemezéssel rendelkező részével (pl. egy ATM kapcsoló vagy egy forgalomirányító egyetlen kimeneti kapuja /portja/) foglalkozom.

A forgalomütemezési algoritmusok többnyire csomag/cella szinten működnek, és több *folyamból* /session/ származó forgalmat kell valamiképpen ütemezniük. Az algoritmusokat futtató kapcsoló csomópontokban ún. szolgáltatási alapelveket /service discipline/ alkalmaznak. Ezek célja a bejövő csomagok kiszolgálási sorrendjének meghatározása, és ezen keresztül a szolgáltatás minőségi paramétereinek garantálása (a kommunikáló végpontok közötti teljes útvonalon). Az szakirodalom számos ütemező algoritmust kínál. Az egyik legjelentősebb az ún. általános kiszolgáló-megosztási algoritmus /Generalized Processor Sharing (GPS)/ [PG92, PG93a, PG93b, PG94], amely az ún. egységes kiszolgáló-megosztási algoritmus /Uniform Processor Sharing/ általánosítása [Kle76].

A GPS a csomagokat egymás után feldolgozó változata /Packetized GPS (PGPS)/ gyakorlatilag megegyezik az ún. súlyozott igazságos sorbanállás /Weighted Fair Queuing (WFQ)/ algoritmussal [DKS90], bár egymástól függetlenül fejlesztették ki azokat. A GPS rendszer leírására Parekh és Gallager a következő jelöléseket vezette be [PG92]:

Egy N folyamot kiszolgáló **GPS kiszolgálót** N darab pozitív valós számmal jellemzünk, ezek $\phi_1, \phi_2, \ldots, \phi_N$. A kiszolgáló *rögzített* r *kiszolgálási sebességgel* működik, és ún. munkamegőrző tulajdonságú. A munkamegőrzés azt jelenti, hogy ha van elküldhető csomag, a kiszolgáló nem lehet tétlen állapotban, azaz el kell küldenie a csomago(ka)t. Jelölje $W_i(t_1, t_2)$ a $[t_1, t_2]$ intervallumban az i. folyamból kiszolgált forgalom mennyiségét. Ekkor a GPS kiszolgálót a következőképpen definiáljuk. Legyen

$$\frac{W_i(t_1, t_2)}{W_j(t_1, t_2)} \geq \frac{\phi_i}{\phi_j}, \quad j = 1, 2, \ldots, N, \tag{1.1}$$

minden olyan i folyamra, amelynek a $[t_1, t_2]$ intervallumban folyamatosan van hátraléka[1]. E definíció közvetlen következménye, hogy minden folyamhoz létezik egy garantált minimális kiszolgálási sebesség /*guaranteed service rate*/, amely a következő alakban írható fel: $g_i = \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} r$ [PG92].

A GPS kiszolgálóban a bejövő forgalmat a *leaky bucket* /lyukas vödör/ algoritmussal (lásd 1. ábra) lehet formázni. Az algoritmus viselkedését a maximális tokenszám (σ) és

[1] a folyamhoz tartozó várakozási sor nem üres

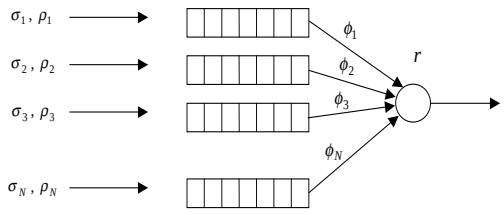

Figure 1: GPS kiszolgáló leaky bucket algoritmussal korlátozott bemenetekkel

a tokengenerálási sebesség (ρ) határozzák meg [Wor94, YSS92]. A leaky bucket algoritmus használatának egyik nagy előnye, hogy a csomagkésleltetés két összetevőre bontható: a vödörben elszenvedett késleltetésre és a hálózatban elszenvedett késleltetésre. Az első összetevő független a többi (aktív) folyamtól, míg a második összetevő független a bejövő forgalom nagyságától [PG92].

Ha végtelen kapacitású vonalakat feltételezünk, az i. aktív forráshoz tartozó vödörből a $[t_1, t_2]$ intervallumban érkező forgalom nagysága leírható egy $A_i(t_1, t_2)$ függvénnyel. Ha $A_i(t) = A_i(0, t) = \sigma_i + \rho_i t$, akkor definíció szerint az i. folyam *mohó* /greedy/ módon indul, azaz a nulla időpillanatban az adást a hozzátartozó legnagyobb löket /burst/ mérettel kezdi, majd a hozzátartozó maximális sebességgel (ρ_i) folytatja az adást. Ha valamennyi folyam mohó, akkor *mohó GPS rendszert* kapunk. Megállapításaimat erősítendő itt összefoglalom a [PG92] legfontosabb eredményét:

Tegyük fel, hogy $C_j > r$ minden j folyamra, ahol C_j a j. folyam vödre és a j. folyam várakozási sora közötti vonalkapacitás, r pedig a GPS kiszolgáló kapacitása (sebessége). Ez esetben a maximális késleltetés D_i^* és a maximális hátralék Q_i^* valamennyi i folyamra akkor adódik (nem feltétlenül egyidőben), ha minden folyam a nulla időpillanttól, a rendszer aktiv periódusának [2] *kezdetétől mohón viselkedik. Továbbá ha feltesszük, hogy valamennyi i folyamra $g_i \geq \rho_i$, akkor*

$$Q_i^* \leq \sigma_i \quad \textit{és} \quad D_i^* \leq \frac{\sigma_i}{g_i}. \tag{1.2}$$

A fenti eredmény jelentősége abban rejlik, hogy a mohó rendszer tekinthető a „legrosszabb esetnek”, minden más érkezési minta esetén jobb minőségi paramétereket kapunk. Így a minőségi paraméterek biztosításának feladata sokkal kezelhetőbbé válik, hiszen elegendő a mohó rendszert vizsgálnunk, nem kell általános érkezési mintákkal foglalkoznunk. Ebből kifolyólag a továbbiakban, az általánosság elvesztése nélkül, csupán a mohó GPS rendszerrel foglalkozom, amelyben tehát minden folyam a nulla időpillanattól, a rendszer aktív

[2] *olyan időintervallum, amelyben a kiszolgáló folyamatosan kiszolgálja a bejövő igényeket*

periódusának kezdetétől mohó viselkedést mutat. A fenti eredmény végtelen vonalkapacitás ($C_i = \infty$) mellett is érvényben marad, ezért vizsgálataimat a végtelen kapacitású esetre végeztem. A véges esetre való bővítés egyszerűen elvégezhető [LB96, LB98].

A GPS szolgáltatási alapelve olyan, teljes igazságosságot megvalósító folyadékmodell, amelyben a forgalom végtelenül kis részekre osztható, és minden folyam a kiszolgáló kapacitásán osztozva, párhuzamosan kerül kiszolgálásra. Bár a valóságban ilyen GPS rendszer nem hozható létre, számos olyan ütemező létezik, amely a csomagkiszolgálási sorrend meghatározásához a háttérben ezt a modellt emulálja. A folyadékmodell és a csomagszinten működő GPS változatok közötti lényeges összefüggések megállapításával sikerült elemezni a csomagokat egymás után feldolgozó GPS változatokat is [PG92]. Az esetek többségében elegendő a GPS modellt vizsgálni, hiszen az eredmények egyszerűen átültethetők a csomagszinten működő változatokra [Gol94, PG92].

2 A disszertáció célkitűzése

A disszertáció fő célja, hogy új alapra helyezze, hatékonyabbá tegye az általános kiszolgáló-megosztásti modellen alapuló / Generalized Processor Sharing (GPS)/ ütemezők jellemzőinek meghatározását. A korábbi determinisztikus megközelítések túlságosan óvatos leírásmódot választottak a GPS ütemezők jellemzésére, és korlátozták az elérhető kapacitást. Disszertációmban

- kidolgoztam egy új iteratív megoldást, amely képes GPS folyadék modellre épülő, tetszőlegesen súlyozható kiszolgálók jellemzésére;
- bemutattam egy olyan rendszert, amelyben a súlyokat nem a folyam sávszélesség-, hanem késleltetés-igényének megfelelően lehet beállítani, a két igény egymástól függetlenül kezelhető;
- A fenti megoldásra alapozva olyan algoritmust adtam, mellyel kiszámítható a GPS kiszolgálók legrosszabb esetben adódó késleltetésének felső korlátja. Az algoritmus a következő tulajdonságokkal rendelkezik:
  - sebesség-arányos súlyozás esetén a legrosszabb esetre vonatkozó késleltetésre szorosabb korlátot ad, mint a Parekh és társai által definiált, túlságosan óvatos korlát. [PG93b, PG92],
  - a lokálisan stabil folyamok esetében (ahol $g_i \geq \rho_i$) a késleltetéskorlát szorosabb, mint a g_i garantált sebesség alapján számított korlát,
  - a Parekh és társai által adott algoritmussal szemben tetszőleges súlyozással működik. [PG93b, PG92].

A fenti eredményeknek meglehetősen kevés hasznuk lenne, ha nem definiálnánk melléjük új hívásengedélyezési /call admission control (CAC)/ és súlyelosztási algoritmusokat. Ezért összevontam a hívásengedélyezés és a súlyelosztás feladatát, és a probléma megoldására számos új algoritmust dolgoztam ki. A disszertációban megmutatom, hogy a nem sebesség-arányos GPS rendszerben a súlyelosztás megoldása egyáltalán nem triviális feladat.

Disszertációmmal a szolgáltatásminőséget biztosító ütemezők területén próbáltam új eredményekkel gazdagítani a tudományt. Külön figyelmet szenteltem az idetartozó ideális folyadékmodellen alapuló GPS rendszer analitikus leírásának. Az analitikus megközelítéssel párhuzamosan teljesítmény értékelési célzattal a bemutatott algoritmusok numerikus értékelését is elvégeztem.

3 Kutatási módszer

A disszertációban igazságos ütemezők *determinisztikus folyadékmodelljeit* vizsgáltam. E rendszerek fontos jellemzőinek levezetéséhez matematikai analízist használtam. Ahol a feladat komplexitása nem tette lehetővé az eredmények zárt formában töténő leírását, ott numerikus kiértékelést (Mathematica) alkalmaztam.

4 Az új eredmények

4.1 GPS kiszolgálók tetszőleges súlyozása [W1, W2, C6, C7, C8, C9]

A GPS alapú ütemezők egy nagy alosztálya a sebességarányos /rate-proportional/ kiszolgálók (RPS), amelyekben a súlyokat az egyes folyamok sávszélesség-igényének megfelelően állítják be. [3] Ez viszont csatolást hoz létre a sávszélesség és a késleltetés között, ugyanis a késleltetés felső korlátja fordított arányban áll a hosszú távú sávszélességgel (ρ). Az ilyen súlyozás hátulütője, hogy egy folyam késleltetési korlátjának csökkentéséhez a hozzátartozó sávszélességet is meg kell növelni ahhoz, hogy a sebességarányos sávszélesség-foglalást fenntartsuk. Világos, hogy ez rugalmatlan erőforrás-foglaláshoz, a hálózati erőforrások pazarlásához vezethet.

A sebesség-arányos súlyozással ellentétben én olyan esetekkel foglalkoztam, amelyekben a ϕ_i súly a ρ_i paramétertől függetlenül kerül beállításra. A továbbiakban ezt a folyamok *tetszőleges súlyozásának* nevezem. Ilyen súlyozással egyes folyamok garantált minimális kiszolgálási sebessége kisebb lehet, mint a hozzájuk tartozó tokengenerálási sebesség ρ, de

[3] az itt használt RPS definíció nem egyezik meg [SV98] definíciójával.

a rendszer stabil marad, amíg a hosszantartó érkezési sebességek (ρ) összege szigorúan a GPS kiszolgáló sebessége alatt marad. [PG92, PG93b], [**C7**].

Mielőtt tovább mennénk, bevezetek egy általános jelölésrendszert, amely a későbbiek megértéséhez szükséges:

Mivel a GPS kiszolgáló *munkamegőrző* és a rendszer *mohó* (1.2), minden i folyam a nulla időpillanatban, a *rendszer aktív periódusának* kezdetekor σ_i nagyságú löketet küld, majd ρ_i sebességgel folytatja az adást. A továbbiakban kizárólag *globálisan stabil* rendszerekkel foglalkozom, amelyekre $\sum_{j=1}^{N} \rho_j < r$; így a GPS kiszolgáló előbb vagy utóbb valamennyi folyam várakozási sorát kiüríti, és utána csak a hosszantartó sebességekkel érkező forgalmat kell feldolgoznia. Felhívom a figyelmet, hogy folyadék-modellen alapuló rendszerről van szó, tehát a folyamok egyidejűleg kerülnek kiszolgálásra.

Az olyan folyamra, amelynek várakozási sora nem üres, azt mondjuk, *hátraléka* van. A $t > 0$ időpillanatban a sor nagyságát $Q_i(t)$ -vel jelöljük.

Legyen $\mathcal{L} = \{L(J)|\ J = 1, \ldots, N\}$ az indexek rendezett halmaza, ahol $L(I)$ azt az i sorszámot jelenti, amely sorszámú folyam várakozási sora I-edikként ürül ki. Továbbá legyen $I = L^{-1}(i)$, és t_I jelentse azt a pillanatot, amikor az i. folyam hátraléka nullává válik. Definíció szerint legyen $t_0 = 0$ a mohó folyamokat fogadó rendszer aktív periódusának kezdete. Továbbá legyen az i. folyam kiszolgálási sebessége a t időpillanatban $r_i(t)$.

TÉZIS 1 ([W1, W2, C6, C7]): *A teszőlegesen súlyozott GPS kiszolgálók leírásához, kezeléséhez új, iteratív algoritmust dolgoztam ki.*

TÉZIS 1.1 ([C6, C7, C9]): *Megmutattam, hogy a hátralékkal rendelkező folyamok kiszolgálási sebessége a GPS kiszolgálóban egy szakaszonként lineáris függvény, melyet úgy lehet meghatározni, hogy a kiszolgáló hátralékos folyamokra redukált sebességének súlyarányos hányadát vesszük.*

Tétel 1.1.1 (Diszkrét sebességfüggvény [C6, C7, C9]). *A rendszer aktív periódusában lévő*
$[t_{k-1}, t_k) intervallumban,\ k = 1, \ldots, N$ *a* j*. folyam kiszolgálási sebessége*

$$
\begin{aligned}
r_j(t) = r_1^j = g_j = \frac{\phi_j}{\sum_{l=1}^{N} \phi_l} \qquad & j \in \{1, \ldots, N\},\ k = 1 \\
r_j(t) = r_k^j = \frac{(1 - \sum_{l=1}^{k-1} \rho_{L(l)})\phi_j}{\sum_{i=1}^{N} \phi_i - \sum_{l=1}^{k-1} \phi_{L(l)}}\, r \qquad & j \in \{1, \ldots, N\} \setminus \{\bigcup_{m=1}^{k-1} L(m)\} \\
& k = 2, \ldots, N. \qquad (4.1)
\end{aligned}
$$

Megjegyzés: Parekh és Gallager [PG93b] publikációjukban hasonló képletet használtak, de formálisan nem bizonyították be. Esetemben viszont az algoritmus felépítéséhez szükség volt a fenti tétel formális bizonyítására.

A most megmutatandó következmény szerint a fenti tételben szereplő függvény monoton növekvő:

Következmény 1.1.2 ([C6, C7, C9]). *r_k^j monoton növekvő függvénye k-nak minden olyan hátralékkal rendelkező j folyamra, amelyre $\phi_j > 0$.*

TÉZIS 1.2 ([C6, C7, C9]): *A következő két lemmában és az azokat követő tételben megmutattam, hogy a mohó GPS rendszer adott folyamainál* hol *és* mikor *áll elő a maximális hátralék.*

Felhívom a figyelmet az én munkám valamint Parekh és Gallager [PG93b] publikációjában bemutatott eredmény közötti különbségekre. Míg ők a hátralék maximális nagyságának meghatározására egy elméleti megközelítést adtak, addig én itt azokat az időpillanatokat határoztam meg, amelyekben egy adott folyam a súlyától függő legnagyobb sorhosszt tapasztalja.

Első lemmám azt állítja, hogy a rendszer aktív periódusának kezdetén legalább egy olyan folyam van, melynek várakozási sora elkezd kiürülni. Második lemmám az előző kibővítése, amely az eredményeket a folytonos időbe helyezi át.

Lemma 1.2.1 ([C6, C9]). *Ha egy rendszer aktív periódusában minden folyam mohó módon indul, van legalább egy olyan folyam, amelynek a kezdeti kiszolgálási sebessége nagyobb az érkezési sebességénél.*

Lemma 1.2.2 ([C6, C9]). *Ha egy rendszer aktív periódusában minden folyam mohó, minden t időpillanatban van legalább egy olyan folyam, amelynek kiszolgálási sebessége nagyobb az érkezési sebességénél.*

Továbbá megmutattam, hogy

Tétel 1.2.3 (Maximális hátralék [C6, C7, C9]). *Minden i folyamnak vagy a nulla időpillanatban van, amikor maximális hátraléka egyenlő σ_i-vel, vagy abban az időpillanatban lesz maximális a hátraléka, amelyben kiszolgálási sebessége növekedni (változni) kezd.*

A fenti tétel fontos következménye, hogy a maximális hátralék meghatározásához elegendő akkor vizsgálni a hátralékot, amikor az egyes folyamok várakozási sora kiürül, mert ekkor változik a többi folyam kiszolgálási sebessége.

Mivel a sorok kiürülési időpontjai nagyon fontos szerepet játszottak,

TÉZIS 1.3 ([C6, C7, C9]): *ezek kiszámítására iteratív algoritmust dolgoztam ki.*

Tétel 1.3.1 ([C6, C7, C9]). *Egy tetszőlegesen súlyozott GPS rendszerben a sorkiürülési (hátraléktörlési) időpontok a k-adik lépésben (t_k) a következő iteratív képlet segítségével számíthatók:*

$$t_k = \min\{t_{i,k} \mid t_{i,k} > 0;\ i \in \{1, \ldots, N\} \setminus \bigcup_{j=1}^{k-1} L(j)\} \tag{4.2}$$

ahol

$$t_{i,k} = \frac{S_{i,k}}{r_k^i - \rho_i} \tag{4.3}$$

és

$$S_{i,k} = \sigma_i + \sum_{j=1}^{k-1} r_j^i\ (t_{j-1} - t_j) + r_k^i\ t_{k-1}. \tag{4.4}$$

Megjegyzés: a $t_{i,k}$ időpontok értelmezhetők a k-adik lépés lehetséges befejezési időpontjainak, amelyek közül a legkisebb pozitív értéket választjuk.

A 2.ábrán az algoritmus folyamatábrája látható.

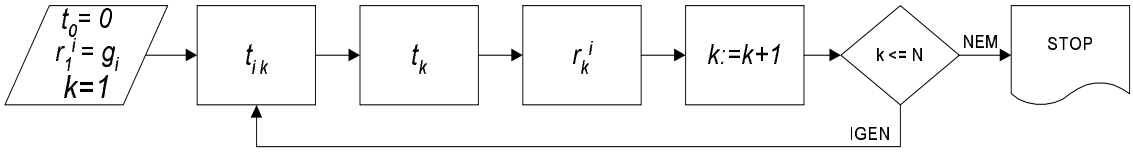

Figure 2: Az iteratív algoritmus folyamatárája

Vegyük észre a rekurziót is:

$$S_{i,1} = \sigma_i \tag{4.5}$$

$$S_{i,k+1} = S_{i,k} + (r_{k+1}^i - r_k^i)\ t_k, \tag{4.6}$$

és azt, hogy

$$L(k) = \arg\min_i\{t_{i,k} \mid t_{i,k} > 0;\ i \in \{1, \ldots, N\} \setminus \bigcup_{j=1}^{k-1} L(j)\}. \tag{4.7}$$

4.2 A tetszőlegesen súlyozott GPS kiszolgálók késleltetési korlátai

TÉZIS 2 ([C6, C7, C9]): *Bebizonyítottam, hogy a Parekh és társai által adott, általános, széles körűen használt késleltetési korlát gyenge, ezért a késleltetés kiszámítására a lokálisan stabil rendszerekre ($g_i \geq \rho_i$) érvényes új képletet vezettem le, amely erősebb korlátot ad. Az irodalomban megtalálható képletekkel ellentétben az új formula tetszőlegesen súlyozott rendszerekre is működik.*

TÉZIS 2.1 ([C6, C7, C9]): *Bebizonyítottam, hogy (i) azon folyamok esetében, amelyek kezdeti lökete nagyobb sebességű kiszolgálást kap, mint a folyam érkezési sebessége, a legnagyobb késleltetés megfelel a kezdeti löketük utolsóként kiszolgált darabja által elszenvedett késleltetésnek; míg (ii) azok a folyamok, amelyek nem felelnek meg a fenti kritériumnak, abban a pillanatban tapasztalják a legnagyobb késleltetést, amikor a kiszolgálási sebességük* először *haladja meg vagy éri el az érkezési sebességüket.*

Tétel 2.1.1 (Késleltetési korlát [C6, C7, C9]).
(i) Ha az (4.1) részben definiált $r_i(t)$ kielégíti a

$$r_i(\tau_i) \geq \rho_i \tag{4.8}$$

egyenlőtlenséget, akkor

$$D_i = \tau_i \tag{4.9}$$

ahol τ_i definíciója a következő:

$$W_i(0, \tau_i) = \int_0^{\tau_i} r_i(t)\,\mathrm{d}t = \sigma_i. \tag{4.10}$$

(ii) Ha egy $i = L(I)$ folyamra (4.8) nem áll fenn, akkor a folyam hátralékot felhalmozó szakasszal indul, melyre létezik olyan $j \in \{1, \ldots, I-1\}$, hogy ha

$$\forall t < t_j : \; r_i(t) < \rho_i \tag{4.11}$$

és

$$\forall t \geq t_j : \; r_i(t) \geq \rho_i \tag{4.12}$$

akkor

$$D_i = t_j - \frac{W_i(0, t_j) - \sigma_i}{\rho_i} \tag{4.13}$$

ahol $W_i(0,t)$ az a forgalommennyiség, amelyet a kiszolgáló az i folyam forgalmából a t időpillanatig kiszolgált.

Megjegyzés: (ii) olyan szcenárióknak felel meg, amelyekben van lokálisan instabil folyam. Másrészt ha a lokális stabilitás biztosított, akkor a (4.10) egyenlettel kiszámított késleltetési korlát legalább olyan erős, mint a Parekh és társai által adott (1.2) korlát. Formálisabban:

TÉZIS 2.2 ([C6, C7, C9]): *Bebizonyítottam, hogy a* lokálisan stabil rendszerek *esetében 2.1.1 tétel által a késleltetésre adott korlát legalább olyan jó, mint a Parekh és társai által adott korlát.*

Tétel 2.2.1 (Erősebb késleltetési korlát [C6, C7, C9]). *Minden lokálisan stabil folyamra ($g_i \geq \rho_i$), ahol Parekh és társai eredménye érvényes, a következő kapcsolat mutatható meg*

$$D_i \leq D_i^*, \tag{4.14}$$

ahol D_i a 2.1.1. tételben definiált, míg D_i^* (1.2) szerinti.

A lokálisan stabil rendszerekre vonatkozó 2. tézis a 2.2.1. tétel következménye, míg tetszőleges súlyozásra való kibővítését a 2.1.1. tétel adja. Valamennyi tételt analitikusan bizonyítottam.

A 3. ábra egy négy folyamos szcenáriót mutat, amelyen a legrosszabb esetbeli késleltetést láthatjuk a két számítási módszernek megfelelően. A vörös (vékony fekete) vonal a garantált sebességű megközelítéssel számított késleltetést mutatja az i. folyamra. A kék (vastag fekete) vonal szakaszonként lineáris, és azt mutatja, hogyan változik az i. folyam kiszolgálási görbéje amikor a hátralékok törlődnek, azaz hogyan növekszik kiszolgálási sebesség. D_i az én módszeremmel számított maximális késleltetést mutatja, t_i i. folyam hátralék törlési ideje, míg D_i^* a garantált kiszolgálási sebességű megközelítés alapján számított késleltetés. A szakaszonként lineáris görbe minden töréspontjánál egy folyam várakozási sora kiürül, így a továbbiakban kevesebb kiszolgáló-kapacitást igényel, mint korábban. Felhívom a figyelmet, hogy mohó rendszerről van szó.

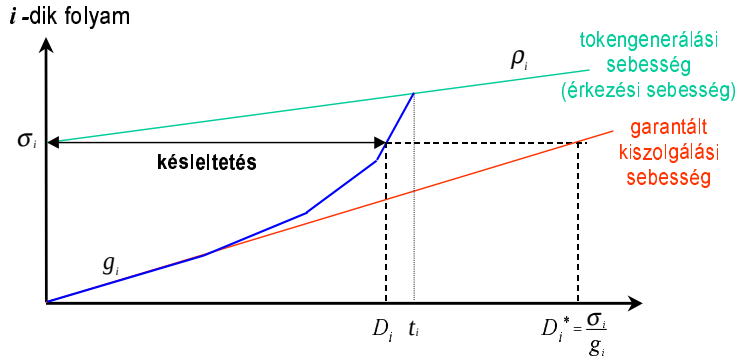

Figure 3: Egy 4 folyamos GPS rendszer kiszolgálási és érkezési függvényei

A 4. ábrán a lehetséges súlyozási szcenáriókat láthatjuk. Ha a súlyozási teret normalizáljuk, akkor az érkezési sebességgel arányos (vagy röviden sebesség-arányos) kiszolgálók halmaza (RPS) egyetlen ponttá zsugorodik a lehetséges paraméterek terében.

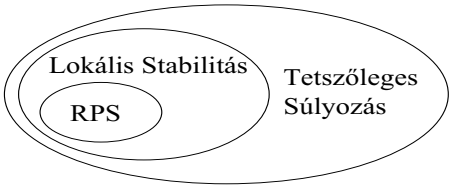

Figure 4: Folyamok súlyozásának lehetőségei

4.3 A súlyozás megváltoztatásának hatásai a késleltetési korlátokra [C8, C6]

Az én hívásengedélyezési algoritmusaim az előzőleg felvázolt tetszőleges (nem sebességarányos) súlyozást használó megközelítésre épülnek, és a bevezetett, erősebb késleltetési korlátokat veszik figyelembe. Itt olyan GPS kiszolgálót vettem alapul, amely adott számú, leaky bucket algoritmussal korlátozott folyamot szolgál ki egyszerre. A sebességarányos súlyozás felrúgásával a ϕ_i paraméterek meghatározása már nem egyértelmű, így új súlyelosztási módszert kellett bevezetnem, amely kielégíti mind a késleltetés-, mind a sávszélesség-igényeket. A sávszélességre vonatkozó hívásengedélyezés meglehetősen egyszerű, a jól ismert stabilitási kritériumra épül ($\sum_{i=1}^{N} \rho_i < r$). A súlyok késleltetési követelményeknek megfelelő beállítása viszont, mint később látni fogjuk, egyáltalán nem nyilvánvaló.

TÉZIS 3 ([C8, C6]): *Matematikailag levezettem, hogy egy tetszőlegesen súlyozott GPS rendszerben egyetlen folyam súlyának megváltoztatása milyen hatással van az általa tapasztalt legnagyobb késleltetésre.*

A megváltoztatott rendszerparamétereket a továbbiakban jelölje egy **'** jel.

Első lemmám azt mutatja, hogy ha egy folyam súlyát növeljük, a folyamhoz tartozó kiszolgálási sebesség növekszik, a többi, hátralékkal rendelkező folyam kiszolgálása pedig lelassul, amíg az i. folyam hátraléka el nem fogy.

Lemma 3.1 ([C8, C6]). *Vegyünk egy mohó GPS rendszert, amelyben a várakozási sorok kiürülésének sorrendje $\{L(1), \ldots, L(N)\}$. Most legyen $\phi'_i = \phi_i + \delta$ úgy, hogy a várakozási sorok kiürülési sorrendje ne változzon meg, tehát $\{L(1), \ldots, L(N)\} = \{L'(1), \ldots, L'(N)\}$ legyen. Ekkor*

$$\exists I : i = L(I) = L'(I)$$

melyre

$$\begin{aligned} r'^{j}_{k} &< r^{j}_{k} \quad \forall j \in \mathcal{B}(k) \setminus \{i\},\ k \in \{1, \ldots, I\} \\ r'^{i}_{m} &> r^{i}_{m} \quad \forall m \in \{1, \ldots, I\}, \end{aligned} \tag{4.16}$$

ahol $\mathcal{B}(k)$ a k-adik lépésben hátralékkal rendelkező folyamok halmaza.

Mivel a 3.1 lemma a kiszolgálási sebességek kapcsolatát csupán a várakozási sorok ürítésének pillanatában írja le, szükség volt a következő lemmára, amely a 3.1 lemma állítását folytonos időbe helyezi át.

Lemma 3.2 ([C8, C6]). *Ha a 3.1 lemma feltételeit megtartjuk, továbbá feltesszük, hogy* $t'_k > t_k \; \forall k \in \{1, \ldots, I-1\}$ *akkor*

$$\begin{aligned} r'_j(t) &< r_j(t) \quad \forall j \in \mathcal{B}(t) \setminus \{i\}, \; t \in [0, t'_I] \\ r'_i(t) &> r_i(t) \qquad\qquad \forall t \in [0, t'_I] \end{aligned} \tag{4.17}$$

ahol $r_j(t)$ *a* j*. folyam* t *időpontbeli kiszolgálási sebessége a (4.1) definíciónak megfelelően.*

Vegyük észre, hogy ez a lemma többet állít, mint a megelőző, hiszen a várakozási sorok ürítési időpontjai között is megadja a kiszolgálási sebességek kapcsolatát.

Ezután lépésről lépésre gyengítettem a 3.2 lemma feltételeit. Először bebizonyítottam, hogyan tolódnak el a befejezési idők, ha egy folyam súlyát megváltoztatjuk. Következő lépésként enyhítettem a változatlan befejezési sorrendre vonatkozó megkötésen.

Lemma 3.3 ([C8, C6]). *Ha egy GPS rendszerben a* ϕ_i *súly* $\phi'_i = \phi_i + \delta$ *-ra változik, ahol* $\delta > 0$ *és a várakozási sorok ürítésének sorrendje változatlan, azaz* $\{L(1), \ldots, L(N)\} = \{L'(1), \ldots, L'(N)\}$*, akkor*

$$\exists I : i = L(I) = L'(I)$$

melyre

$$\begin{aligned} t'_k &\geq t_k \quad k \in \{1, \ldots, I-1\} \\ t'_I &\leq t_I \quad \text{, ahol } i = L(I). \end{aligned} \tag{4.19}$$

Lemma 3.4. *Még ha a GPS rendszerben a várakozási sorok ürítésének sorrendje az átsúlyozás miatt meg is változik, a 3.1 - 3.3 lemmák állításai továbbra is igazak maradnak.*

A fenti lemmákból a kiszolgálási függvényre vonatkozó, másik fontos lemmát is levezettem:

Lemma 3.5 ([C8, C6]). *Az* i*. folyam kiszolgálási függvénye (*$W_i(t, \underline{\phi}, \underline{\sigma}, \underline{\rho}) = W_i(0, t)$ *az adott* $\{\underline{\phi}, \underline{\sigma}, \underline{\rho}\}$ *bemenő paraméter vektorokkal) a folyam súlyának monoton* növekvő *függvénye. Más szavakkal, tetszőleges* t *időpillanatban*

$$\frac{\Delta W_i(t, \underline{\phi}, \underline{\sigma}, \underline{\rho})}{\Delta \phi_i} \geq 0. \tag{4.20}$$

A fenti lemmából következik, hogy egy folyam súlyának növelésével csökkenthető a folyam várakozási sorának kiürítési ideje, tehát

Következmény 3.6 ([C8, C6]). *A 3.5 lemmából következik, hogy* $t'_i \leq t_i$*, ha* $\phi'_i = \phi_i + \delta$*, ahol* $\delta > 0$*.*

A most következő, fő tételem a tézisfüzet eredményeit összegzi, és a legrosszabb esetben előálló késleltetési korlát és a folyam súlyának kapcsolatát adja meg. A tételt a korábbi lemmákkal bizonyítottam be.

Tétel 3.7 ([C8, C6]). *Az* i*-dik folyam legrosszabb esetben előálló késleltetése a folyam súlyának monoton csökkenő függvénye. Más szavakkal*

$$\frac{\Delta D_i(\underline{\phi}, \underline{\sigma}, \underline{\rho})}{\Delta \phi_i} \leq 0. \tag{4.21}$$

A fentiekkel azt akartam elérni, hogy kézben lehessen tartani a folyamok legrosszabb esetben előálló késleltetését. A 3.7 tételből most már tudjuk, hogy egy folyam legrosszabb esetben előálló késleltetésének csökkentéséhez növelni kell a folyam súlyát. Sajnos, ha egynél több súlyt módosítunk, az nagyon bonyolulttá teszi a probléma kezelését. Ennek ellenére a hívásengedéyezési (CAC) algoritmusoknál is ezt a feltételt használtam.

4.4 Tetszőleges súlyozású GPS kiszolgálók hívásengedélyezési algoritmusai[C8, C6]

Ideális esetben a hívásengedélyezés /call admission control (CAC)/ feladata abban merül ki, hogy megmondja, a rendszer képes-e kielégíteni az újonnan érkező folyam igényeit. A valóságban azonban nem elegendő tudni, hogy van megoldás, meg is kell találni egy megvalósítható elosztást. Ezért a továbbiakban a hívásengedélyezést összekapcsolom egy megvalósítható megoldáshalmaz $\mathcal{F}$ megtalálásával, ha ilyen létezik, bármilyen adott $(\underline{\sigma}, \underline{\rho}, \underline{D})$ kezdővektorra úgy, hogy $\underline{D} \geq \underline{d} = \text{GPS}(\underline{\sigma}, \underline{\rho}, \underline{\phi})$, ahol $\underline{\phi} \in \mathcal{F}$; $\underline{D}, \underline{d}(\underline{\phi})$ a késleltetési igény és a GPS aktuális késleltetési vektora ebben a sorrendben, $(\underline{\sigma}, \underline{\rho})$ pedig a folyamokhoz tartozó leaky bucket leírói.

TÉZIS 4 ([C8, C6]): *A GPS rendszer hívásengedélyezésének megoldására új megoldást dolgoztam ki, és más tudományos területekről származó hívásengedélyezési algoritmusokkal vetettem össze. A teljesítményük méréséhez egy új metrikát vezettem be, a* késleltetés-kezelés korrektsége */delay-fairness/ néven.*

Algoritmus 4.1 *(Véletlen keresés [**C8**, **C6**]).* A megoldást véletlenszerű alapon keressük korlátozott öröklődéssel és gyakori mutációkkal. Az algoritmus a genetikus algoritmusok leszármazottjának tekinthető.

TÉZIS 4.1 ([C8, C6]): *A gradiens módszer az adott problémára való adaptációja során egy új teljesítmény-mutató metrikát vezettem be* (késleltetés-kezelés korrektsége)*, amely tetszőleges GPS kiszolgáló relatív késleltetés-kezelését jellemzi. Olyan célfüggvényt vezettem le, amely a súlyozás beállításával optimalizálja a fenti metrikát.*

Algoritmus 4.2 *(Gradiens módszer [**C8**, **C6**]).* Az algoritmus egy N dimenziós vektor-skalár függvényt használ, amely a minimalizáció alapjául szolgál (O). A késleltetési korlát túllépését minden folyam esetében egy büntető függvény szabályozza. A 3.7 tétel egyértelműen megadja, hogyan kell a késleltetés csökkentése érdekében a folyamok súlyozását beállítani és ezzel egy megfelelő folyamhalmazt megkapni.

A gradiens módszer két célfüggvényt használ, a továbbiakban az elsőt *laposnak* nevezzük, mert csak a késleltetési korlát átlépését bünteti, a folyam követelményeinek túlteljesítését nem veszi figyelembe:

$$O_1(\underline{\sigma}, \underline{\rho}, \underline{D}, \underline{\phi}) = \sum_{i=1}^{N} I_{(d_i - D_i)}(d_i - D_i)^2, \tag{4.22}$$

ahol $I_{(x)}$ egy indikátorfüggvény, melyet a következők szerint definiálunk:

$$\begin{aligned} I_{(x)} &= 0 \quad x < 0 \\ I_{(x)} &= 1 \quad x \geq 0. \end{aligned} \tag{4.23}$$

A célfüggvénynek van néhány hátrányos tuljadonsága, például igazságtalan előnyhöz juttatja azokat a folyamokat, amelyek késleltetése jóval alatta marad az igényeiknek. Emellett a célfüggvény laposssága miatt az algoritmus sokára szabadítja fel a célfüggvény lapos részén elhelyezkedő folyamok által feleslegesen lefoglalt erőforrásokat, mert a paraméterek megváltozása nem hoz azonnal eredményt. Mivel a paraméterteret minden lépésben normalizáljuk, a beférő régión kívüli folyamok fognak megoldást adni. A fenti hiányosságot kiküszöbölödendő olyan célfüggvényt szerkesztettem, amely gyorsabb konvergenciát biztosít és *korrekt késleltetésű* megoldást talál, ha ilyen létezik.

A GPS rendszerekre a következő teljesítmény-mutató metrikát vezettem be:

Definíció 4.3 (késleltetés-kezelés korrektsége). Egy GPS rendszer késleltetéskezelésének korrektsége (Γ) egy adott $\{\underline{\sigma}, \underline{\rho}, \underline{D}, \underline{\phi}\}$ vektorhalmaz esetén egyenlő a késleltetések szórásnégyzetének összegével, vagy formálisabban

$$\Gamma(\underline{\sigma}, \underline{\rho}, \underline{D}, \underline{\phi}) = \frac{1}{N} \sum_{i=1}^{N} (\frac{d_i}{D_i} - \frac{1}{N} \sum_{j=1}^{N} \frac{d_j}{D_j})^2. \tag{4.24}$$

A késleltetési korrektséget figyelembe veendő, az alábbi célfüggvényt szerkesztettem:

$$O_2(\underline{\sigma}, \underline{\rho}, \underline{D}, \underline{\phi}) = \Gamma(\underline{\sigma}, \underline{\rho}, \underline{D}, \underline{\phi}) + O_1. \tag{4.25}$$

Megjegyzés: a konvergencia sebesség növelése érdekében különböző súlyozási faktorok használhatók, de ennek részletes tanulmányozása kívül esik e disszertáció keretein.

Továbbá egyszerűen megmutatható, hogy O_2 optimalizálása a $\underline{\phi}$ térben olyan megoldáshoz vezet (ha létezik ilyen), amelyben a legrosszabb esetbeli késleltetések arányos javulása egyforma, azaz

$$\frac{d_i^*}{D_i} = \frac{d_j^*}{D_j} \quad \forall i,j \in \{1,\ldots,N\}. \tag{4.26}$$

Ezen felül megmutattam, hogy ϕ^* nem csak késleltetési szempontból nyújt korrekt megoldást, de azt is megmutatja, hogy mennyivel lehet még a folyamok késleltetését csökkenteni a kiszolgálhatóság fenntartása mellett. Azt mondhatjuk, hogy a $\{\underline{\sigma}, \underline{\rho}, \underline{\mathbf{d}}^*, \underline{\phi}\}$ együttessel definiált GPS rendszer eléri *hatékonysága határát*, ha ugyanis valamelyik folyam késleltetési igényei megnőnek, akkor már nem lehet valamennyi folyam igényét kielégíteni.

TÉZIS 4.2 (Inkrementális hívásengedélyezés [C8, C6]): *A következő jellemzőkkel rendelkező hívásengedélyezési algoritmust dolgoztam ki: Feltételezzük, hogy a hívásengedélyezés minden alkalommal lefut, amikor egy új folyam érkezik a GPS rendszerbe a kiszolgált $N-1$ folyam mellé. A hívásengedélyezést a következők szerint kell elvégezni:*

- *Adjunk a rendszerbe egy $\sigma_d = \infty$ paraméterrel rendelkező* látszólagos *folyamot, amely minden szabad erőforrást lefoglal;*
- *a súlyokat úgy állítsuk be, hogy minden folyam a megengedett maximális késleltetését tapasztalja, amikor a látszólagos folyamot is figyelembe vesszük.*

Most kapja meg az új folyam a látszólagos folyam súlyát.

- *Ha a látszólagos folyam súlyával az új folyam késleltetési igényei kielégíthetőek, akkor a hívásengedélyező függvény rögtön kiadhatja a megfelelő súlyt tartalmazó befogadó nyugtát a folyamnak.*
- *Ha a látszólagos folyam súlya nem elegendő a folyam számára, akkor újra kell optimalizálni a rendszerben lévő folyamok súlyait. Mivel az új folyamnak kevesebb erőforrásra van szüksége, mint a látszólagos folyamnak, a többi folyam lemondhat a neki allokált súly egy részéről az új folyam javára. Ha a kiszolgáló az újraoptimalizálás után sem képes a folyam igényeit kielégíteni, akkor a hívásengedélyezés elutasítja a hívást.*

Pozitív nyugta esetén a rendszert a 3.7 tételnek megfelelően úgy kell újraoptimalizálni, hogy a látszólagos folyam most az $(N+1)$-edik helyre kerüljön. Ez a lépés a rendszer szabad idejében is elvégezhető, így a hívásengedélyezés eldöntése a legtöbb esetben egyetlen lépésben elvégezhető. Kivételt csak az jelent, ha a rendszer a hatékonysága korlátjánál vagy ahhoz közel áll. Az algoritmus folyamatábrája az 5. ábrán látható.

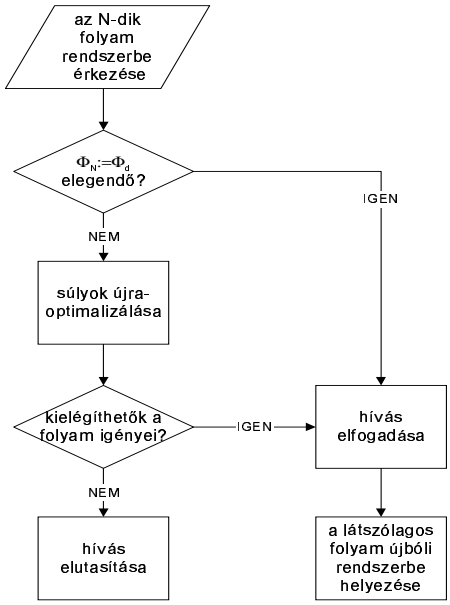

Figure 5: Az inkrementális hívásengedélyezési algoritmus folyamatábrája

TÉZIS 4.3 ([C8, C6]): *Megvizsgáltam a GPS rendszerben a késleltetés-számítás komplexitását, a gradiens módszer használata, illetve inkrementális hívásengedélyezés esetében. Azt állapítottam meg, hogy valamennyi algoritmus polinomiális időben elvégezhető.*

Az új hívásengedélyezési algoritmusok számítási komplexitása a következőképpen összegezhető: [**C8**, **C6**]: egyetlen GPS késleltetés-számítás egy adott $\{\underline{\sigma}, \underline{\rho}, \underline{\phi}\}$ együttesre vonatkozóan $o(N^2)$ komplexitással rendelketik; a gradiens módszer további $o(N)$ komplexitással járul a GPS számításhoz, amely összeségében $o(N^3)$ numerikus komplexitást eredményez. Mivel az inkrementális hívásengedélyezés képes 1 lépésben dönteni, a GPS kalkuláció csupán $o(N^2)$ komplexitású is lehet, ha a látszólagos folyamnak kiosztott súly kielégíti az új folyam igényeit. Más esetekben ez a módszer is $o(N^3)$ komplexitású.

Bár az algoritmusok számításigénye nagyon hasonló, más céllal lettek kitalálva. Az inkrementális hívásengedélyezés a látszólagos folyamon keresztül képes egy minőségi garanciákkal nem rendelkező /best effort/ szolgáltatási osztály kezelésére, míg a gradiens módszer korrekt késleltetéskezelést biztosít a GPS alapú ütemezőkben.

4.4.1 Záró megjegyzések

Röviden összefoglalva, az új hívásengedélyezési algoritmusok számításigénye egyértelműen nagyobb a hagyományos, garantált sebességet biztosító hívásengedélyezési algoritmusok számításigényénél. Az új algoritmusokkal viszont:

- a hálózati erőforrások kihasználtsága jobb,
- a sávszélesség és a késleltetés szétválasztható[**C7**],
- a súlyozás rugalmasabb
- az inkrementális hívásengedélyezés esetében best-effort forgalom is kezelhető [**C8**].

Mivel az új hívásengedélyezési algoritmusok számításigénye megnőtt, felmerül a kérdés, hogy alkalmazhatók-e a gyakorlatban az azonnali döntéshozási feladat megoldására. A gyakorlati felhasználhatóság érdekében

- olyan megoldásokra van szükség, amelyek azonnali elfogadást vagy visszautást képesek megvalósítani – a rendszer szabadidejében történő folyam-optimalizációval –

  vagy

- az algoritmusok számításigényét olyan alacsonyan kell tartani, hogy azok a hardverek/szoftverek mai fejlettségi szintjén használhatók legyenek.

Hívásengedélyezési algoritmusommal az első megközelítést követem, és a kiszolgáló szabad idejét használom a rendszer tartózkodó folyamok súlyainak újraoptimalizálására. Másrészt mivel a berendezések számítási teljesítménye folyamatosan növekszik, a gradiens módszerhez hasonló, nagyobb bonyolultságú algoritmusok is egyre inkább használhatóvá válnak/váltak.

Vegyük észre, hogy ezekben az esetekben nem csak a folyamok elfogadásáról vagy visszautasításáról döntünk, hanem mind a sávszélesség-, mind a késleltetés-igényeket figyelembe vevő súlyelosztás is megtörténik.

5 Az új eredmények alkalmazási lehetőségei

A disszertációban az ideális folyadékmodellen alapuló igazságos ütemezőket vizsgáltam. Ezek a modellek a valós rendszerekben közvetlenül nem valósíthatók meg, de az elért

eredmények gyakorlatba ültetése egyszerűen elvégezhető a GPS rendszer csomagokkal dolgozó változatának mintájára.

Az igazságos ütemezők tervezésének ma elfogadott módszere a Parekh és társai által adott determinisztikus modellen alapul. Amint rámutattam, ez a modell túlságosan óvatos, ebből kifolyólag korlátozza az elérhető kapacitást. Az általam javasolt késleltetési korlátok bevezetésével ugyan nagyobb lesz a számítási komplexitás, de javul a hálózat kihasználtsága, csökken az erőforrások pazarlása.

Egy másik problémakör, amellyel foglalkoztam, a GPS alapú ütemezőkben a sávszélesség és az időzítés közötti csatolás *csökkentését* célozza.

Ez a jövőbeli, kis sávszélességigényű, valós idejű alkalmazásoknál lesz fontos (pl. ATM/IP fölötti beszédátvitel). A hagyományos súlyelosztási módszerrel ugyanis kis késleltetés csak úgy garantálható, ha túl sok erőforrást foglalunk le a folyam számára. Azt javasoltam, hogy a folyamok súlyát úgy állítsuk be, hogy a késleltetési igényeieket is figyelembe vesszük. Ennek megvalósítására algoritmusokat dolgoztam ki.

Meg kell jegyeznem, hogy az *inkrementális hívásengedélyezés* algoritmusa a mai, minőségi garanciákat nem biztosító /best-effort/ hálózatokban is használható, hiszen a *látszólagos* folyamon keresztül a prioritást élvező folyamok QoS garanciáinak megtartása mellett a best-effort forgalmat is ki lehet szolgálni.

Az általam bemutatott modell az őséhez, a GPS modellhez hasonlóan elsősorban elvi eredménynek számít, bár nagy számításigénye ellenére a GPS rendszer csomagokkal dolgozó változatát a gyakorlatban is használják. Az általam javasolt, a sávszélesség és a késleltetés külön kezelését lehetővé tevő rendszer, valamint a hívásengedélyezési algoritmusok gyakorlati megvalósítása folyamatban van.

Köszönetnyilvánítás

Sok köszönettel tartozom az ütemezéssel foglalkozó csoportban velem dolgozóknak, név szerint Bíró Józsefnek, Barta Péternek és Németh Feliciánnak.

Az itt leírt kutatást a stockholmi székhelyű Ericsson Applied Research - Switch Lab, valamint a Budapesti Műszaki és Gazdaságtudományi Egyetem Távközlési és Telematikai Tanszékének Nagysebességű Hálózatok Laboratóriuma támogatta. Külön szeretném megköszönni Carl-Gunnar Perntz, Boda Miklós és Henk Tamás támogatását.

References

[BZ94] J.C.R. Bennett and H. Zhang. WF^2Q: Worst-case fair weighted fair queueing. In *Proceedings of IEEE INFOCOM'94*, pages 120–128, San Francisco, March 1994.

[BZ96] J.C.R. Bennett and H. Zhang. Hierarchical packet fair queueing algorithms. In *Proceedings ACM SIGCOMM'96*, pages 143–156, August 1996.

[CSZ92] D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: Architetcture and mechanism. In *Proceedings of ACM SIGCOMM'92*, pages 14–26, Baltimore, Maryland, August 1992.

[DKS90] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. *Internetworking: Research and Experience*, 1(1):3–26, January 1990.

[Gol90] S. Golestani. A stop-and-go queueing framework for congestion management. In *Proceedings of ACM SIGCOMM'90*, pages 8–18, Philadelphia, PA, September 1990.

[Gol94] S. Golestani. A self-clocked fair queueing scheme for broadband applications. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 636–646, Toronto, CA, June 1994.

[Kle76] L. Kleinrock. *Queueing Systems, Volume 2: Computer Applications*. Wiley, 1976.

[LB96] Jean-Yves Le Boudec. Network calculus made easy. Technical Report EPFL/DI 96/218, Networking and Communication Lab, EPFL, Lausanne, Switzerland, December 1996.

[LB98] Jean-Yves Le Boudec. Selected lecture notes. Technical report, ICA Ecole Polytechnique Federale de Lausanne, October 1998.

[PG92] Abhay K. Parekh and Robert G. Gallager. A generalized processor sharing approach to flow control in integrated services networks — the single node case. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 2, pages 915–924 (7A.3), Florence, Italy, May 1992. IEEE.

[PG93a] A.K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 521–530, San Francisco, CA, March 1993.

[PG93b] A.K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[PG94] A.K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[SV98] D. Stiliadis and A. Varma. Rate-proportional servers: A design methodology for fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 6(2):164–174, April 1998.

[Wor94] T. Worster. Modelling deterministic queues: The leaky bucket as an arrival process. In *International Teletraffic Congress, ITC-14*, volume 1a, pages 581–590. ITC, June 1994. Antibes, France.

[YSS92] N. Yamanaka, Y. Sato, and K. Sato. Performance limitation of leaky bucket algorithms for usage parameter control and bandwidth allocation methods. *IEICE Transactions on Communications*, E75-B(2):82–86, 1992.

[Zha95] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. In *Proceedings of the IEEE*, 83(10), October 1995.

Az új eredmények publikációi

[J] **FOLYÓIRAT CIKKEK**

[J1] I. Cselényi, **Róbert Szabó**, I. Szabó, A. Latour-Henner, C. Gisglrd, and N. Bjőrkman. Experimental platform for telecommunication resource management. *Computer Communications*, 21:1624–1640, 1998. Angol nyelven jelent meg.

[J2] G. Fodor, T. Henk, T. Marosits, **Róbert Szabó** and L. Westberg. Simulative analysis of optimal routing and link allocation strategies in B-ISDN networks. *Periodica Polytechnica*, 42(3):275–197, 1998. Angol nyelven jelent meg.

[J3] P. Barta, Cs. Lukovszki, Ádám Marquentatn, G. Fodor, and **Róbert Szabó**. Optimized resource utilization on call and cell level in atm networks. *Magyar Távközlés: Selected Papers from the Hungarian Telcommunication Periodicals*, 2–6. oldal, 1999. Angol nyelven jelent meg.

[J4] I. Cselényi and **Róbert Szabó**. Service specific information based resource allocation for multimedia applications. *Design Issues of Gigabit Networking of INFORMATICA JOURNAL*, 23(3):317–325, 1999. ISSN 0350-5596. Angol nyelven jelent meg.

[J5] Hoványi D., Kováts G., Daróczy S., és **Szabó Róbert**. Voip lehetőségei. *Magyar Távközlés*, 11:10–14, 1999. nov. Magyar nyelven jelent meg.

[J6] András C., Takács A., és Szabó Róbert. VoIP szolgálatok minőségbiztosítása *Magyar Távközlés*, 5:14–17, 2000. máj. Magyar nyelven jelent meg.

[J7] Cs. Lukovszki, **Róbert Szabó**, and T. Henk. Performance evaluation of a hybrid atm switch architecture by parallel discrete event simulation. *INFORMATICA JOURNAL*, 2000. A 2000-ben megjelenő második számban, angol nyelven jelenik meg.

[C] **KONFERENCIÁK**

[C1] G. Fodor, T. Henk, T. Marosits, and **Róbert Szabó**. On the call and cell level resource allocation in atm networks. Megjelent a *Proceedings of European Simulation Symposium 1995* kiadványban, 475–484. oldal, Erlangen-Nuremberg, Németország, 1995. okt. SCS, The Society for Computer Simulations. Angol nyelven.

[C2] G. Fodor, T. Marosits, and **Róbert Szabó**. Comparison of conservative parallel simulation techniques for multistage interconnection networks. Megjelent a *Proceedings of European Simulation Multiconference 1996* kiadványban, 513–517. oldal, Budapest, 1996. jún. SCS, The Society for Computer Simulation International. Angol nyelven.

[**C3**] **Róbert Szabó** and Cs. Lukovszki. Parallel cell scale simulation of a hybrid atm switch architecture. Megjelent a *Proceedings of SPECTS '98 - 1998 Symposium on Performance Evaluation of Computer and Telecommunication Systems* kiadványban, 53–57. oldal, Reno, Nevada, 1998. júl. SCS. Angol nyelven.

[**C4**] I. Cselényi and **Róbert Szabó**. Performance evaluation of an intelligent resource allocation scheme for multimedia applications. Megjelent a *Proceedings of SPECTS '98 - 1998 Symposium on Performance Evaluation of Computer and Telecommunication Systems* kiadványban, 302–306. oldal, Reno, Nevada, 1998. júl. SCS, The Society for Computer Simulation International, SCS. Angol nyelven.

[**C5**] Cs. Lukovszki, **Róbert Szabó**, and T. Henk. Parallel simulation model of a novel hybrid min architecture. Megjelent a *Proceedings of European Simulation Symposium, ESS'98* kiadványban, 725–729. oldal, 1998. okt. Angol nyelven.

[**C6**] **Róbert Szabó**, P. Barta, F. Németh, and J. Bíró. A novel approach to admission and flow control in generalized processor sharing (GPS) schedulers. *4th International Conference on Applied Informatics*, 1999. aug., Eger-Noszvaj. Angol nyelven.

[**C7**] **Róbert Szabó**, P. Barta, F. Németh, J. Bíró, and C.-G. Perntz. Non rate proportional weighting of generalized processor sharing schedulers. Megjelent a *Proceedings of GLOBECOM'99* kiadványban, 2. kötet, 1334–1339. oldal, Rio de Janerio, Brazília, 1999. dec. Angol nyelven.

[**C8**] **Róbert Szabó**, P. Barta, F. Németh, J. Bíró, and C.-G. Perntz. Call admission control in generalized processor sharing (GPS) schedulers using non-rate proportional weighting of sessions. Megjelent a *Proceedings of INFOCOM 2000* kiadványban, 3. kötet, 1243–1252. oldal, Tel-Aviv, Izrael, 2000. márc. Angol nyelven.

[**C9**] **Róbert Szabó**, P. Barta, F. Németh, and J. Bíró. Worst-case deterministic delay bounds for arbitrary weighted generalized processor sharing schedulers. Megjelent a *Proceedings of Networking 2000* kiadványban, 727–739. oldal, Párizs, Franciaország, 2000. máj. Angol nyelven.

[C10] P. Barta, F. Németh, **Róbert Szabó**, and J. Bíró. Network qos provisioning through per node traffic characterization. Globecomm2000 konferenciára benyújtva.

[W] **SZAKMAI WORKSHOPOK ÉS EGYEBEK**

[**W1**] J. Bíró, T. Henk, **Róbert Szabó**, P. Barta, and F. Németh. Issues on packet service disciplines for guaranteed quality of service networks. *COST 257 MC Meeting*, 1999. jan. Faro, Portugália. Angol nyelven.

[**W2]** J. Bíró, T. Henk, **Róbert Szabó**, and P. Barta. Non rate proportional generalized processor sharing schedulers. *IFIP WG 6.3 Workshop*, 1999. aug. Rethymnon, Kréta, Görögország. Angol nyelven.

[**W3]** G. Rétvári and **Róbert Szabó**. Qos-based routing and ip multicasting: A framework. *Fifth EUNICE Open European Summer School*, 1999. szept. Barcelona, Spanyolország. Angol nyelven.