



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Híradástechnikai Tanszék

Lencse Gábor:

Kommunikációs rendszerek hatékony szimulációjának
egyed kérdései

Ph.D. értekezés tézisei

Témavezető:

dr. Németh Gábor

Konzulens:

dr. Pongor György

Budapest, 2000.

Bevezetés

Kommunikációs rendszerek teljesítőképesség vizsgálatának – az analitikus módszer és a valóságos rendszeren való mérés mellett – nélkülözhetetlen eszköze a diszkrét idejű szimuláció. Ennek két fajtája az idővezérelt és az eseményvezérelt szimuláció, melyek közül az utóbbi az általánosabb, sokkal szélesebb körben alkalmazzák, én is ezt tettem vizsgálataim tárgyává.

Összetett rendszereknél a szimuláció erőforrásigénye igen nagy lehet, nem számítanak kirívónak a több napos futási idők sem. A párhuzamos szimuláció ismert szinkronizációs módszerei (konzervatív, optimista) nem hoztak áttörő eredményeket; a konzervatív módszer csak bizonyos (erős) feltételek teljesülése esetén hoz gyorsulást, az optimista pedig nem jól skálázható, ráadásult bonyolult is. Elméletileg létezik egy harmadik módszer is a *statisztikai szinkronizáció* (Statistical Synchronisation Method, a továbbiakban: SSM), amely eredetileg publikált formájában csak állandósult állapotban alkalmazható párhuzamos szimuláció szinkronizációs módszereként.

Célkitűzések

1. A kedvező tulajdonságai miatt ígéretes statisztikai szinkronizációs módszert fejleszttem tovább, hogy időben változó rendszerek párhuzamos szinkronizációs módszereként is alkalmazható legyen, a kidolgozott módszer tulajdonságait vizsgálom, a felmerülő új kérdésekre keresek választ.
2. Nagy eseményszám esetén a jövőbeli események halmaza (Future Event Set, a továbbiakban: FES) műveleteinek végrehajtási idő szükséglete kritikussá válhat, ezt vizsgálom megfelelő megoldást keresve.
3. A szimuláció lépésszámának csökkentése érdekében, a sok mintát igénylő statisztikagyűjtés helyett eleve statisztikákkal működő módszert dolgozok ki, amely közelítő eredményeket ad.

Alkalmazott vizsgálati módszerek

Az SSM továbbfejlesztéseként megalkottam az SSM-T módszert időben lassan változó állapotú rendszerek párhuzamos szimulációval történő vizsgálatára. Meghatároztam a módszer alkalmazhatósági kritériumait, és megvizsgáltam az elérhető gyorsulást. Az elérhető gyorsulásra vonatkozólag esettanulmány jelleggel egy valóságos rendszer pontos modelljén valóság-hű terheléssel kísérleteket végeztem.

Az SSM-T módszerhez szükséges statisztikagyűjtés erőforrásigényét az algoritmuselmélet módszereivel, pontosságát pedig szimulációval vizsgáltam.

A statisztikacsere vezérlésének kérdéseit általános esetben analitikusan (levelezéssel) vizsgáltam, speciális esetben pedig szimulációval.

A FES tárolására használható adatszerkezetek teljesítőképességét szimulációval hasonlítottam össze. A terhelés paramétereit széles tartományban ortogonálisan változtattam, a kapott nagy méretű adathalmaz elemzésével tettem javaslatot az optimális adatszerkezetek kiválasztására.

A 3. Célkitűzés megvalósításának érdekében új módszert dolgoztam ki, ami a szimulációnak és a numerikus, illetve analitikus módszereknek az ötvözete.

Új kutatási eredmények

SSM-T téziscsoport

1. tézis: az SSM-T módszer és tulajdonságai

1.1. altézis: az SSM-T módszer

Megalkottam a statisztikai szinkronizációs módszer idővezérelt változatát (SSM-T), ahol a *laza időszinkronizáció* biztosítja, hogy az egyes szegmensek virtuális idői a t_i szinkronizációs időpontokban találkozzanak – ekkor történik meg a szegmensek interfészei között a statisztikacsere. Az **A** és **B** szegmens közötti laza időszinkronizáció formális definíciója a következő:

Legyenek $t_1, t_2, \dots, t_i, \dots, t_n$ szinkronizációs időpontok. Jelölje t_A és t_B az **A** illetve **B** szegmens lokális virtuális idejét. Az **A** és **B** szegmens között laza időszinkronizáció van, ha:

$$((t_A < t_i) \Rightarrow (t_B \leq t_i)) \wedge ((t_A > t_i) \Rightarrow (t_B \geq t_i)), \quad i = 1, 2, \dots, n.$$

A két szegmens közötti laza időszinkronizáció azt jelenti, hogy egyik szegmens sem hagyhat el egy szinkronizációs időpontot, míg a másik el nem érte azt.

Mivel a lokális kauzalitási feltétel biztosított, az általam kidolgozott SSM-T módszer alkalmas párhuzamos diszkrét idejű szimuláció szinkronizációs módszerének.

1.2. altézis: az SSM-T módszer alkalmazhatósági kritériumai

Jelölje az üzenetfolyam statisztikai jellemzőit az X valószínűségi vektorváltozó, a közelítését pedig X^* . A közelítés hibája is egy valószínűségi vektorváltozó: $h_X = X^* - X$. A szimuláció általunk megfigyelt kimenetét jelölje O . Az üzenetfolyamok statisztikai jellemzőinek közelítésének h_X hibája a kimenetben $h_O = O^* - O$ hibát okoz. Legyen H_O felső korlát a kimeneti hiba normájára az SSM-T által okozott tranziens alatt, azaz $\|h_O\| < H_O$. Jelölje N_{TR} a tranziens alatt gyűjtött megfigyelések számát N_{QST} pedig a kvázistacionárius állapot időtartama alatt gyűjtött szimulációs kimeneti értékek számát.

Az SSM-T módszer alkalmazható, és az alkalmazásával végzett párhuzamos szimuláció kimenetében a módszer által okozott hibáról bizonyítható, hogy $\|\bar{h}_O\| < \varepsilon$, ha az alábbi feltételek mindegyike fennáll:

- (a) A szimulált rendszer szegmensekre osztható úgy, hogy a szegmensek közötti forgalomban csak az üzenetfolyam statisztikai jellemzői számítanak, az egyes üzenetek maguk a rendszer működése és az általunk megfigyelt kimeneti jellemzők szempontjából nem fontosak.
- (b) $h_O \leq f(h_X, X^0)$, - azaz h_O felülről becsülhető egy olyan értékkel, ami csak h_X -től függ, X értékétől nem - és $\forall \varepsilon > 0 \exists \delta: \|h_X\| < \delta \Rightarrow \|h_O\| < \varepsilon$.
- (c) A modell paraméterei változhatnak a szimuláció alatt, de a szegmensek közötti üzenetfolyamok statisztikai jellemzőinek változásai *kellően ritkák*, azaz: $N_{QST} \geq \frac{2N_{TR}H_O}{\varepsilon} - N_{TR}$.

- (d) Mivel egy üzenetfolyam statisztikai jellemzőinek változása csak akkor jut el más szegmensekhez, amikor a kimeneti interfész elküldi az általa a megváltozott statisztikai jellemzők szerint gyűjtött statisztika csomagot, a negyedik kritérium az, hogy ez a késleltetés csak a késleltetés alatt, vagy még legfeljebb a késleltetés idejével arányos ideig okoz hibát a szimuláció kimenetében.

Megmutattam, hogy az alkalmazhatósági kritériumok teljesülése esetén az egyprocesszoros szimuláció eredményeit tetszőleges pontossággal közelíteni tudja az SSM-T módszer. (A bizonyítás az értekezés 2.2.3. pontjában található.)

1.3. altézis: az SSM-T alkalmazása esetén elérhető gyorsulás

Két szegmensből álló szimulációnál a gyorsulás:

$$s = \frac{\tau_A + \tau_B - \tau_{IF}}{\max(\tau_A, \tau_B) + \tau_C}$$

ahol a szegmenseket végrehajtó processzorok azonosak, τ_A és τ_B az **A** és **B** szegmens T hosszú virtuális idejéhez tartozó végrehajtási idő (tartalmazza a saját ki- és bemeneti interfészeinek végrehajtási idejét is), τ_{IF} az ki- és bemeneti interfészek végrehajtási ideje, τ_C pedig a kommunikációval töltött idő.

Ez az érték közelíthet a 2-höz, ha $\tau_A \approx \tau_B$, $\tau_C \ll \tau_A$ és $\tau_{IF} \ll \tau_A$, azaz a processzorok közti terhelés jól kiegyensúlyozott, a kommunikáció, valamint a statisztikagyűjtés és üzenetfolyam regenerálás által okozott többletmunka kicsi.

Egy valós életből vett hálózat nagy pontosságú szimulációjának a példáján megmutattam, hogy ezek a feltételek nagyon is elérhetőek: **két összekapcsolt FDDI gyűrű párhuzamos szimulációjánál a szimulációt SSM-T alkalmazásával két processzoron futtatva, 1.91 illetve 1.86 szoros gyorsulást értem el az egy processzoros szimulációhoz képest.** A kimenet hibája az 1.91 szoros gyorsulást adó esetben az egyik gyűrű esetén 0.63% a másik gyűrű esetén pedig 3.56% volt az egyprocesszoros szimulációhoz képest.

2. tézis: az SSM-T-nél alkalmazható statisztikagyűjtési eljárások tulajdonságai

Az SSM-T számára különböző statisztikagyűjtési eljárások erőforrásigényét és pontosságát (konvergenciasebességét) vizsgáltam a szimuláció során előforduló fontosabb eloszlások esetén. Az L_1 hibakritériumot választottam, mert ez valóban **események valószínűsét** fejezi ki. (Az indoklás az értekezés 2.2.3. pontjában található.)

2.1. altézis: statisztikagyűjtés abszolút folytonos eloszlások esetén

Exponenciális és gamma eloszlás esetén vizsgáltam meg a következő statisztikagyűjtési eljárásokat:

- Barron estimate
- egyenlő osztásközű hisztogram (*equidistant histogram*)
- egyenlő cellavalószínűségű hisztogram (*equiprobable bin histogram*)
- kvázi egyenlő cellavalószínűségű hisztogram (*semi equiprobable bin histogram*)

A Barron estimate esetén többféle közelítő függvényt is alkalmaztam, a különböző hisztogramoknál pedig meghatároztam az optimális cellaméretet. Az egyes módszerek L_1 hibáját a minta elemszámának függvényében vizsgáltam.

Megállapítottam, hogy abszolút folytonos eloszlásokra a Barron estimate adja a legkisebb L_1 hibát, de az eloszlásról a-priori információra van szüksége, ami az általános esetben nem áll rendelkezésünkre.

Számítás útján megmutattam, hogy exponenciális eloszlás esetén az egyenlő valószínűségű cellájú hisztogram kisebb L_1 hibát ad, mint az egyenlő osztásközű. Az optimálisnál jóval kisebb számú cella esetén ezt mutatta a gyakorlat is, **elegendő számú cella esetén azonban az egyenlő osztásközű hisztogram bizonyult jobbnak, ezért abszolút folytonos eloszlásokra ezt a módszert javasolom**, ennek ráadásul erőforrás igénye is kedvezőbb. (A szimulációs eredmények az értekezés 2.4.4.1.3. pontjában a 9. táblázatban található.)

2.2. altézis: statisztikagyűjtés diszkrét eloszlások esetén

Diszkrét eloszlásoknál elfogadható eredményeket kaptam az L_1 hibára a relatív gyakoriság alkalmazásával: egy adott empirikus eloszlásra (FDDI há-

lózaton mért érkezési időköz kvantálva) $N=2000$ megfigyelés esetén az L_1 hiba átlag értéke 0.1160, (0.0113 szórás mellett). További szimulációs eredmények az értekezés 2.4.4.3. és 2.4.4.4. pontjában találhatóak.

3. tézis: az SSM-T statisztikacsere vezérlő algoritmusának vizsgálata

Előrejelzés és szinkronizációs időpont törlés segítségével működő statisztikacsere vezérlő algoritmust dolgoztam ki a statisztikacsere virtuális időpontjának meghatározására. Ennek lényege, hogy az i . statisztikagyűjtési intervallumban az R_i üzenetgyakoriságot a korábbi R_j $j < i$ alapján becsülve tűzi ki a t_i szinkronizációs időpontot, majd szükség esetén (ha t_i -ben a statisztikai minta még nem tartalmaz kellő számú megfigyelést) új időpont megadásával a kitűzött szinkronizációs időpontot törli. (Részletes leírás az értekezés 2.5.1.3. pontjában van.)

3.1. altézis: büntetőfüggvények és analitikus kezelésük

Büntetőfüggvényeket vezettem be, amelyekkel mérni tudom a statisztikacsere virtuális időpontjának az optimálistól való eltérésének különböző hatásait. Ezek felhasználásával analitikusan meghatároztam a teljes büntetés várható értékét.

A büntetőfüggvények a mintában lévő megfigyelések n számának függvényében azt fejezik ki, hogy mennyire tartjuk rossznak az optimálistól való következő fajta eltéréseket:

1. túl sok a törölt szinkronizációs időpont – $P_{SP}(n)$: *speed-up penalty*
2. a statisztikák túl kevés megfigyelésen alapulnak – $P_{L1}(n)$: L_1 *penalty*
3. a tranziens túl hosszú (túl sok a megfigyelés) – $P_{TR}(n)$: *transient penalty*

Büntetőfüggvényként a levezetésben az alábbiakat használtam:

$$P_{SP}(n) = c_1 / n, \quad n > 0$$

$$P_{L1}(n) = \begin{cases} c_2, & 0 < n < N \\ 0, & n \geq N \end{cases}$$

$$P_{TR}(n) = c_3 n^2$$

Az üzenetekre exponenciális érkezési időközt feltételezve, vagy azt közelítés-ként használva a teljes büntetés várható értékére a következőt kaptam:

$$E\{P(K, \dots, T_k, \dots)\} = c_1 \sum_{k=1}^K \frac{\lambda T_k + 3}{(\lambda T_k)^2} + c_2 \sum_{n'=1}^{N-1} \frac{(\lambda T)^{n'}}{n'!} e^{-\lambda T} + c_3 ((\lambda T)^2 + \lambda T),$$

$$\text{ahol } T = \sum_{k=1}^K T_k,$$

K a kitűzött szinkronizációs időpontok száma, ebből az első $(K-1)$ darab törölt, T_k ($k = 1..K$) a kitűzött szinkronizációs időpontok közti intervallumokat, N a kívánt megfigyelés számot, n' pedig a minta megfigyelés számát jelöli. A c_1 , c_2 és c_3 konstans megválasztása egy kompromisszumot jelent az eredmények pontosságának garantált szintje és a párhuzamos szimuláció gyorsulásának mértéke között.

3.2. altézis: szimulációs vizsgálat speciális esetre

Szimulációs vizsgálatokat végeztem arra az esetre nézve, amikor mindig megköveteljük, hogy a statisztikáink legalább N számú megfigyelésen alapuljanak. A vizsgálataimhoz két féle módszert használtam a szinkronizációs időpontok megválasztására, és meghatároztam a módszerek paramétereinek optimális tartományát. A vizsgálatok legfontosabb következtetése az, hogy **az adott feltételek mellett az SSM-T kellő robusztusságot mutat, azaz jól tolerálja paramétereinek az optimálistól való eltérését.**

A jövőbeli események halmazával kapcsolatos tézis

4. tézis: A FES implementációjára alkalmazható adatszerkezetek vizsgálata

Hét adatszerkezetet (egyirányban láncolt lista, bináris fa, AVL-fa, B-fa, 2-3-fa, kupac, skip-list) vizsgáltam meg szimuláció segítségével, hogy melyiket célszerű diszkrét idejű szimulátorokban a jövőbeli események halmazának (FES) a tárolására alkalmazni. A problémát az események számának függvényében aszimptotikusan vizsgáltam, számos paraméter (a FES állapota, az események száma a FES-ben, az érkezési időköz eloszlása, a törlések aránya, a processzor típusa) összes lehetséges kombinációja mellett.

Természetesen mind a szimulációt végrehajtó processzor architektúrája, mind a szimulációban egymást követő események időeloszlása jelentősen befolyásolhatja, hogy melyik adatszerkezetnek a legjobbak a végrehajtási idő jellemzői. Ezeket a vizsgálatokat 1993-ban kezdtem TDK munka keretében, és doktoranduszként 1995-ben fejeztem be. Az akkori processzorok még nem mind rendelkeztek hardver lebegőpontos művelet támogatással. Az eredményeim azért hasznosak ma is, mert processzoronként különböző lehet a lebegőpontos műveletek végzésének relatív sebessége az egyéb műveletekhez képest.

Speciális esetek kivételével az egymást követő események időeloszlását nem ismerjük.

A műveletek során használt kulcsösszehasonlítások számára a kiegyensúlyozott fák és a kupac adták a legkisebb értékeket. (A szimulációs eredmények az értekezés 3.3.2.1. pontjában találhatóak.) **Ezért azokban az esetekben, ahol az adatszerkezet műveleteinek processzoridejét a kulcsösszehasonlítások dominálják (ahol nincsen hardver lebegőpontos művelet támogatás), a kiegyensúlyozott fák és a kupac jelentik a legjobb választást.** A közönséges (kiegyensúlyozatlan) bináris fa jó eredményeket mutatott exponenciális eloszlás esetén, de más eloszlásokra eredményei rosszabbak a kiegyensúlyozott fák és a kupac eredményeinél.

Erős hardver lebegőpontos támogatás esetén a skip-list és a kupac jobb időjellemzőket adott, mint a kiegyensúlyozott fák. (A szimulációs eredmények az értekezés 3.3.2.3. pontjában találhatóak.) **Ebben az esetben tehát a skip-list vagy a kupac alkalmazását javaslom, ezek algoritmusai egyszerűbbek is, mint a kiegyensúlyozott fákéi.**

Fontosnak tartom megjegyezni a következőket: A skip-list, amely erős hardver lebegőpontos művelet támogatás esetén a legjobb időjellemzőket adta, egy 1990-ben publikált adatszerkezet és nem szerepel az általam ismert FES implementációkkal foglalkozó cikkekben. Ismereteim szerint a processzorok hardver lebegőpontos támogatásának a többi művelethez képesti sebessége vizsgálataim óta tovább nőtt, ami még inkább kiemeli a skip-list jelentőségét.

5. tézis: a Traffic Flow Analysis módszer

Kommunikációs hálózatok teljesítőképesség vizsgálatára alkalmazható új módszert dolgoztam ki, amely a szimulációnak és numerikus módszereknek a kombinációja. A *traffic-flow analysis* (TFA) a hálózat részletes szimulációjához képest lényegesen kisebb számításigénnyel – így sokkal gyorsabban – közelítő eredményeket ad. Működése során a forgalmat megfelelően választott méretű egységekben a hálózat tényleges routolási algoritmusának felhasználásával osztja szét a rendszerben. A hálózat véges kapacitásainak a forgalom időeloszlására gyakorolt hatását a forgalom térbeli elosztása után veszi figyelembe.

Létrehoztam és értekezésemben be is mutattam egy konkrét TFA implementációt, amelynek segítségével egy egyszerű hálózat példáján összehasonlítottam a TFA eredményeit a számítással kapott értékekkel. **Az egyszerű példahálózat esetén a TFA eredményei jól közelítik az analitikus módszer eredményeit.**

5.1. altézis: a TFA módszer alkalmazhatósága

A TFA alkalmazható, ha a hálózat modellje kielégíti az alábbi feltételeket:

- a hálózat csomópontokból és azokat összekötő vonalakból épül fel
- a hálózat forgalmát a csomópontokhoz kapcsolódó alkalmazásmodellek generálják
- a forgalom mindig adott csomóponttól adott csomópont felé irányul (azaz a TFA nem kezel üzenetszórást vagy többesküldést)
- A forgalom mértéke és időeloszlása matematikai modellel leírható. A továbbiakban ezt nevezzük **forgalommodellnek**.
- A forgalommodell az összeadás műveletére nézve zárt, azaz az összeadás eredménye ugyan olyan típusú lesz, mint az összeadandók.
- A csomópontok és vonalak véges kapacitásának a forgalom időeloszlására való hatása kiszámítható. (Nem követelmény a zártság.)
- Forrás- és cél csomópont páronként az azonos típusú alkalmazások azonos cél csomópont felé irányuló forgalma együttesen, de akár tetszőleges méretű egységekben is kezelhető. (aggregált forgalommodell)

- Az egyes forrás- és cél csomópont párok forgalma tetszőleges sorrendben kezelhető

5.2. altézis: a kapacitáskorlátozás megvalósítása throughput-eloszlás, mint forgalommodell esetén

Throughput-eloszlás, mint forgalommodell esetén kidolgoztam a kapacitáskorlátozás (véges kapacitás figyelembevétele) módját.

A módosult throughput-eloszlás és késleltetés eloszlás számítása az értekezés 4.3.3. pontjában található.

Az eredmények alkalmazása

A jövőbeli események halmazával kapcsolatos eredményeimet felhasználták az OMNeT++ diszkrét idejű szimulátor implementációjakor: a FES az eredeti láncolt lista helyett kupac lett.

Az általam kidolgozott TFA módszert az Elassy Consulting Kft. alkalmazza az Iminet hálózati szakértői rendszerében.

Az SSM-T példaként megjelenik az OMNeT++ rendszerben, az Iminetben pedig az alkalmazására irányuló fejlesztés folyamatban van.

Az SSM-T módszer az ausztráliai Monash Universityvel való együttműködés keretében a közeljövőben szuperszámítógépen is kipróbálásra kerül.

A további kutatás témái

Az általam kidolgozott TFA módszernél megvizsgálandók a térbeli szétosztás és a kapacitáskorlátozás pontossági és konvergencia kérdései. Ugyancsak a TFA esetén hasznos lehet további forgalommodellek kidolgozása is.

A téziseket alátámasztó publikációim

Lencse, G. "Investigation of Event Set Algorithms" Proceedings of the 1995 European Simulation Multiconference (ESM 95) Prague, Czech Republic, June 5-7 1995. SCS Europe, pp. 821-825.

Lencse, G. "Performance of Future Event Set Implementations" *Journal on Communications*, Vol. XLVII, December 1996. (pp. 33.)

Lencse, G. "Efficient Simulation of Large Systems - Transient Behaviour and Accuracy" *Proceedings of the 1997 European Simulation Symposium (ESS'97)* Passau, Germany, Oct. 19-23 1997. SCS Europe, 660-665.

Lencse, G. "Efficient Parallel Simulation with the Statistical Synchronization Method" *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation (CNDS'98)* part of the Western Multiconference, San Diego, CA. Jan. 11-14 1998. SCS International, 3-8.

Lencse, G. "Statistics Collection for the Statistical Synchronisation Method" *Proceedings of the 1998 European Simulation Symposium (ESS'98)* Nottingham, UK. Oct. 26-28 1998. SCS Europe, 46-51.

Lencse, G. "Applicability Criteria of the Statistical Synchronization Method" *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation (CNDS'99)* part of the Western Multiconference, San Francisco, CA. Jan. 17-20 1999. SCS International, pp. 159-164.

Lencse, G. "Design Criterion for the Statistics Exchange Control Algorithm used in the Statistical Synchronization Method" *Proceedings of the Advanced Simulation Technologies Conference (ASTC 1999)* part of the 32nd Annual Simulation Symposium, April 11-15 1999, San Diego, CA, USA (pp. 138-144).

Lencse, G. "Traffic-Flow Analysis for Fast Performance Estimation of Communication Systems" - *elfogadva a Journal of Computing and Information Technology-nál*