

Performance Studies of Synchronization in Multimedia Collaborative System Using Stochastic Petri Nets

Li Li,
Bell-Northern Research Ltd.,
Ottawa, Canada

N. D. Georganas & A. Karmouch
Dept. of Electrical Engineering
Univ. of Ottawa, Ottawa., Canada

Introduction

Multimedia collaborative applications, such as joint editing, allow group of users to interactively collaborate on a multimedia document. An example of this application may be the digitized film editing in a not so distant future. Multimedia collaborations carried out over telecommunication networks can save editor's travel time (or film tape's travel time) and their editing time. The distributed computing system can provide editors with variety of resource information, editing materials and tools through the on-line multimedia databases. Meanwhile, the collaborative session on a well controlled distributed computing system can facilitate the editors in discussing their different views to generate the final version of the document.

In multimedia collaborative system, synchronization control is one of the basic requirements [1]. The coordination and causal ordering of the editing events (e.g., paste a new picture by one of the editors) need to be managed. In joint editing, the multimedia document contains the multimedia objects shared by a group of editors over the network. Coordination is required among the distributed users for concurrent access of the shared objects. Meanwhile, the user's manipulation of the multimedia document in the shared window should be consistently "seen" by all the participants on their workstations, irrespective of their locations in the system and the random network conditions. If editor B, for example, makes some modification after having observed the changes made by another editor A, the editions supplied from A and B should be presented in the correct and same order to all the participants' on the network. Thus, causal ordering of the events is required in a multimedia collaborative system.

The event ordering problem in distributed computing systems, including the logic clocks, the partial and total ordering of events, have been well studied [3][4]. However, the multimedia cooperative system has introduced new requirements concerning the causal ordering in the distributed system. The concurrency degree needed in a cooperative system varies in different applications. For instance, in joint editing, mutual exclusive accessing of the shared document is not favored [5][6]. The application needs to allow the concurrent document manipulation from different participants (editors) [5]. At the same time, "conflicting", where two or more editors are changing the same portion of the document concurrently and thus generating different versions at the different participants' sites, should be prevented. For example, after every manipulation conducted by an editor in the system, the result ought to be delivered and displayed to every other participant before it can be changed again. Thus, the basic causal ordering models and algorithms developed in distributed computing have to be adapted and applied, to cope with the problem of coordination and causal ordering in the new multimedia collaborative applications.

The synchronization algorithms for concurrency control and events causal ordering for multimedia collaborative systems can be found in [1][7][8]. In the studies, it is found that the performance of the distributed multimedia collaborative system with synchronization control is vital for the behavior of the entire real-time system. However, the performance modeling and evaluation of synchronization behavior in distributed multimedia collaborative system remains an open problem. The difficulty stands as the cooperation of the collaborative events and their temporal characteristics are not easy to model. Meanwhile, lack of efficient analytical solutions for performance models that can represent the synchronization behavior also hampers the performance study of multimedia collaborative systems.

In this paper, we apply the Stochastic Petri Net (SPN) models to describe the temporal behavior of the distributed collaborative system and represent the synchronization requirements in the system. A new analytical solution for a class of SPN which has product form solution, is proposed and applied to the multimedia joint editing session. Performance measures for the joint editing system are obtained which may be used in the joint editing system design.

The rest of the paper is organized as follows. In section 2, the multimedia collaborative system and the synchronization control requirements are discussed. In section 3, we introduce the Stochastic Petri Nets and the new solution. Section 4 applies the algorithm and obtains the performance measures for a multimedia editing system. Finally, section 5 concludes the paper.

2. Multimedia Collaboration and Synchronization Control

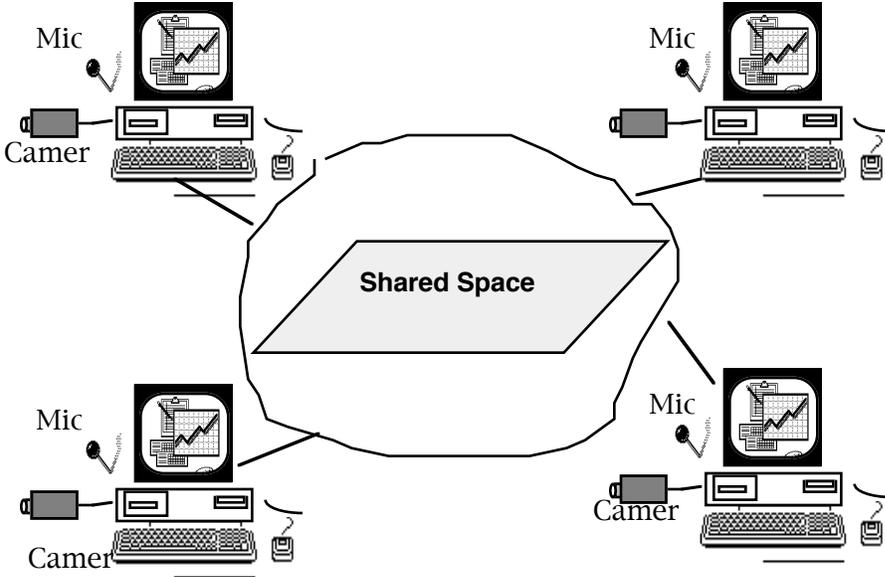


Fig. 2.1 A Multimedia Joint Editing Session

In multimedia collaborations, users are working in a shared multimedia space over the communication network. In the application of joint editing, this shared space is usually occupied by a multimedia document. Participants are editing and modifying the portions of the document,

where the data in various media, e.g., video, graphics, text, audio, etc., are used. When one of the editors makes a modification on a portion of the document, the new version of this portion shall be sent and displayed to all the other participants. Thus multimedia messages are transferred over the network. Eventually, when the editing session is finished, a new multimedia document is obtained. Here, we are discussing the editing session where only a unique version of the document should be generated from the editing, although any participant can keep his/her different "private" versions. As discussed in [1], the synchronization model for this system should handle the concurrency control for accessing the shared space, and at the same time order the multimedia messages at all the participants' sites for display.

Given a group with a shared multimedia space SP, assume that users at the nodes are viewing and/or editing the multimedia document in SP. When the user is viewing the document, the data in the SP are presented at each user's node. To edit the shared space, the editor commits the new edition of a corresponding portion, say pt_i to SP by sending it to all the participants' workstation. This new version of pt_i should be either available to (e.g., "seen" by) all the sites or none of them. Meanwhile, when two editors are attempting to access the same portion in the SP, i.e., pt_i sent from editor 1 and pt_j from editor 2 are not disjoint, the two versions pt_i and pt_j should be presented to all the sites according to the same and correct order.

A synchronization algorithm was developed to handle the joint-editing in a shared multimedia space described above [1][8]. For every part of the document in SP, a mutual exclusive access (editing) is maintained by the algorithm. This is done by the total ordering of the request messages for the portions which are not disjoint. The editor has to request a portion of the document for making the change. If the request is granted, the portion is locked for this editor. After making the change, the editor will commit the new version of the reserved portion to the document, i.e., to the shared space. That is, the new portion was sent to and displayed at each editor's workstation. Meanwhile, the algorithm will "lock" the new portion at each editors site for some time before a new change can be made. This ensures that every editor sees the newest version before making a new change to the document. The synchronization also allows the concurrent editing on the shared multimedia document if the editing portions are disjoint, and are not conflicting with any other "locked" portion in the document. The details of the algorithm are not discussed in this paper, interested readers are referred to [1][8].

In the joint editing described above, the size of the editing portion can play a key role in the synchronization control. The size of a request portion is usually chosen by the editor. A minimum as well as a maximum size can be set by the system. If the minimum size of a portion is big enough so that any two portions will have overlap, a mutual exclusive access is managed. That is, concurrent editing on different part of the document will not happen.

In the implementation of the synchronization control algorithm presented in [1][8], the performance issues become critical. The relevant performance measures for the synchronization control algorithm are required to justify the design of the multimedia collaboration system. However, multimedia synchronization problem raises a challenge to the techniques for performance modeling and evaluation of the distributed computing as well as communication systems. It is known that the available analytical modeling methods have encountered various

difficulties in modeling and evaluating the synchronization aspects in communications [8]. The performance evaluation of the multimedia collaboration, e.g., the joint-editing session still remains an open problem. The difficulty arises from the fact that the cooperation of the editors and their temporal characteristics are not easy to model. It can therefore be envisioned that the development of methods for the modeling of multimedia synchronization will not only take a fruitful direction, but also contribute to the general performance modeling techniques. In the next section, we will first describe an analytical performance modeling tool, a class of Stochastic Petri Nets which has product form solutions. Then, we will apply this class of Petri Net Models to evaluate the multimedia joint editing session.

3. Performance Modeling for Multimedia Synchronization

3.1 Performance Modeling Methods

Two of the best known analytical methods used for computer communications performance evaluation are Queueing Networks (QNs) and Petri Nets (PNs). QNs have the incontestable advantage of combining expressive power and solution efficiency. However, QNs have serious limitations in modeling concurrency in distributed systems; they are unable to represent in a direct way the synchronization in communications. *Ad hoc* extensions and approximation solution methods have been proposed to overcome some of these shortcomings, but they are not general or straightforward enough.

Petri Nets (PNs) with timing properties are the second types of models used for performance analysis of concurrent systems. The representation of synchronization in PN models is fundamental to the notations. Classical PNs consist of a set of places, a number of tokens that reside at places, and a set of transitions that represent events causing tokens to move from one place to another. For performance modeling of computer systems, the classical PNs haven been extended by introducing timing properties. For instance, "Stochastic" PNs (SPNs) are one of these extensions. SPN models have been applied to the analysis of multiprocessor systems, communication networks, etc. Refer to [11] for more references.

A major difficulty in modeling a system by Stochastic Petri Net (SPN) is the explosion of the Markov Chain state space when the system is getting complicated. The modeling power of the SPNs is actually paid with the complexity in solving the models. In fact, a "simple looking" SPN model can generate a very big state space. As a result, the numerical solution of even middle-sized SPN models can often be intractable. The inefficiency in solving the SPN models has always been an open problem, especially in the performance modeling of communication systems [10]. It is known that the distributed multimedia computing systems such as the joint-editing involve complicated temporal cooperation among the system elements. Therefore, when modeling the multimedia synchronization system using SPN models, a big state space is expected and efficient solution algorithms have to be investigated.

In searching for efficient ways to solve SPNs, the class of SPNs which has product form solutions has been investigated. In this paper, we apply a class of SPN models, the Locally Balanced Stochastic Petri Nets (LBSPN) models which have product form solutions [10]. We

have developed a computational algorithm for the LBSPN, which is based on the convolution of the subnets in a LBSPN model. Then, using the computational algorithm, LBSPN can be employed to model the multimedia collaborative application of joint editing.

3.2 Locally Balanced Stochastic Petri Nets And Product Form Solutions

Locally Balanced Stochastic Petri Nets (LBSPN) were first introduced by Lazar and Robertazzi [12]. M. Li and Georganas have further extended the boundary of LBSPNs, proven that the LBSPNs falling into the new boundary also have product form solutions [10][13]. A systematic way for identifying the LBSPN that fall into the new boundary has also been designed [10]. A computational algorithm to evaluate the normalization constant G in LBSPN's product form solution was later developed by Li Li and Georganas [8] which is based on the convolution of the subnets in LBSPN.

Subnet is a basic unit to constitute a LBSPN [10][8]. Given a SPN N_1 , assume Ψ is the set of all the T-invariants of N_1 . For every $\psi_i \in \Psi$, $\|\psi_i\|$ is its invariant support [11], i.e., the set of transitions corresponding to nonzero entries in the T-invariant $\psi_i \geq 0$. If $\exists \psi_i, \psi_j \in \Psi$, and $\psi_i \cap \psi_j \neq \phi$, then we say that ψ_i and ψ_j are in one group. One group of transitions and their input/output places constitute a *subnet*. Thus, the subnets have no common transitions, although they may have common places. These subnets constitute a LBSPN. A test procedure to check if the SPN is a LBSPN was developed in [10]. A LBSPN which consists of N subnets, was proven to have product form solutions [10] as:

$$P(\mu_i) = \frac{1}{G} \prod_{j=1}^N P_j(\rho(\mu_i^j)) \quad \text{where } G \text{ is a normalization constant:}$$

$$G = \sum_{\mu_i} \prod_{j=1}^N P_j(\rho(\mu_i^j)) \quad (1)$$

$P_j(\rho(\mu_i^j))$ is the probability distribution obtained from the Basic Markov Chain of subnet j . In [8], it was proven that the calculation of the normalization constant G can be conducted by the convolution of the subnets. That is, one subnet is calculated first. The second subnet is added into the net, which now is a LBSPN with two subnets, to obtain the G' by convolution. Continuously, all the N subnets can be added into the system to obtain the G of the final LBSPN model. A software program was developed for this computational algorithm which can help to solve middle-sized or even large LBSPNs. The derivation and formulas are not addressed here for the sake of brevity. Readers are referred to [8] for details.

Next section, we apply this subnet convolution algorithm for LBSPN to conduct the performance modeling and evaluation for the multimedia collaborative system of joint editing.

4. Performance Modeling of Multimedia Collaborations

4.1 LBSPN models for Multimedia Joint Editing

As we have discussed in section 2, in multimedia joint editing, editors form a collaborative group. For a collaborative group with multiple editors, we establish its LBSPN model.

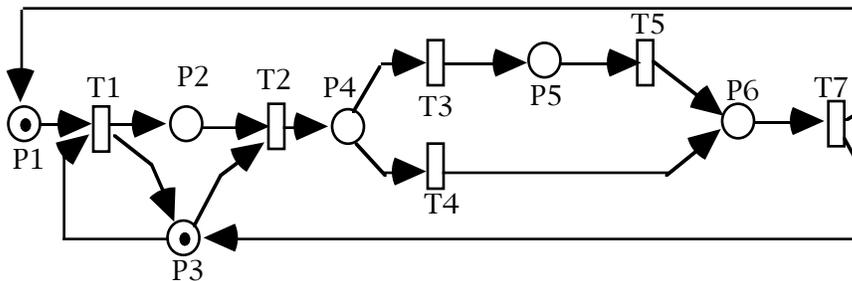
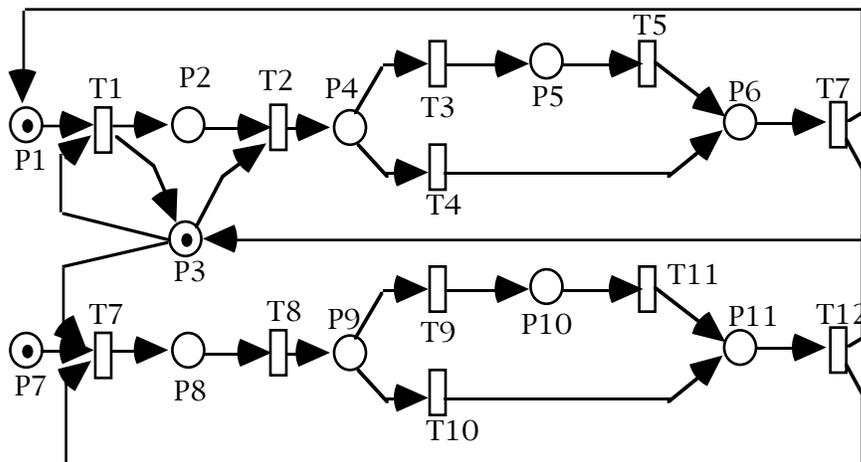


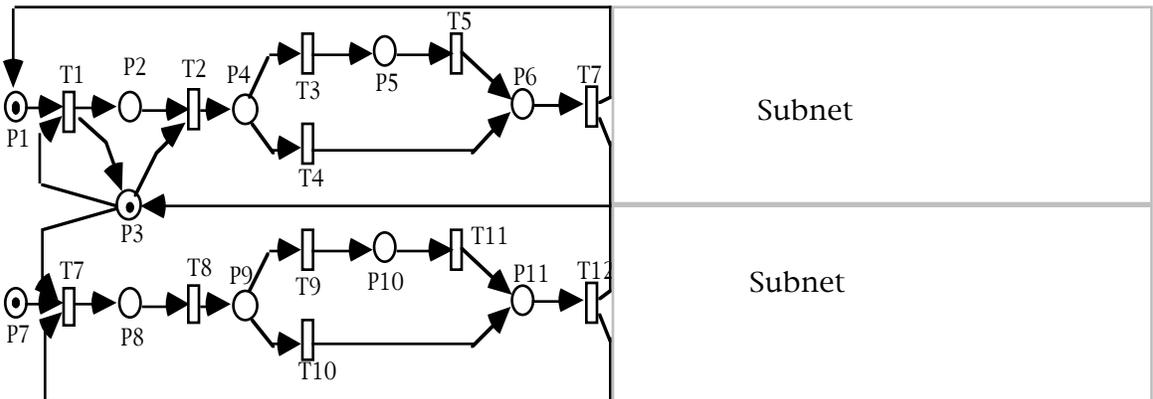
Fig. 4.1 SPN model of one editor in a collaborative group

Fig.4.1 illustrates a SPN model which represents an editor working on a portion of the document. The token in place P1 represents the editor and the token in P3 stands for the document portion. The editor may view a certain portion and crank ideas to modify it. This is represented by the transition T1 in Fig.4.1. Then the editor needs to reserve the portion, i.e., to "lock" it (see section 2) through transition T2. After the editing (transition T3), the new version of the portion will be committed (sent through T5) into the shared space P3, so that all the other participants can view it. That is, the portion is available to all the participants in the collaborative group. On the other hand, collision may be incurred by an editor who wants to reserve the portion, while the portion is reserved by another editor in the group, at almost the same time. In this case, the editor is assumed to wait in P2 until the new version of this portion is committed from the one who has reserved it. Then the editor tries again to reserve the portion. In any case, after reserving the portion, the editor can cancel the reservation without making any modifications. This abortion of the editing is represented by transition T4 in the SPN model.

Take the SPN model in Fig.4.1 as a subnet, the LBSPN models can be constructed, describing multiple editors working on a multimedia document which consists of multiple portions. In Fig.4.2 (a), two editors are shown collaborating on the same document portion. In Fig.4.2 (b), the document comprises two portions so that the two editors can work on the document concurrently by reserving different portions. The models in Fig.4.2 are both LBSPN models. Similarly, LBSPN models can be constructed by connecting the subnet to represent the collaborating group which includes more editors and document portions. When more editors and document portions are added in the model, the state space is dramatically increased. Thus, product form solution method is employed to solve the model.



(a)



(b)

Fig. 4.2 LBSPN model containing two editors

4.2 Product Form Solutions and Performance Results

Class\Transitions	T1	T2	T3	T4	T5	T6
Class 1	0.0083	1.0	0.0083	0.0056	0.0042	1
Class 2	0.0056	1.0	0.0056	0.0042	0.0033	1

Fig. 4.3 Assignment of the Firing Rates

Different transition rates can be selected for the transitions in the subnet of Fig.4.1 to represent the occurring frequency of the editing events. In our example, two different sets of rates are assigned to the subnet, shown in Fig.4.3 as the working speed of class 1 and class 2. For each

class of the transition rates, the subnet is solved. Thus the subnet which has only 5 steady states is solved twice, resulting in the different steady state distributions shown in Fig. 4.4.

With Class 1 Rates	With Class 2 Rates
$P\{1(1), 3(1)\} = 0.357942$	$P\{1(1), 3(1)\} = 0.391800$
$P\{2(1), 3(1)\} = 0.002971$	$P\{2(1), 3(1)\} = 0.002194$
$P\{4(1)\} = 0.213735$	$P\{4(1)\} = 0.223885$
$P\{5(1)\} = 0.422381$	$P\{5(1)\} = 0.379927$
$P\{6(1)\} = 0.002971$	$P\{6(1)\} = 0.002194$

Fig.4.4 Steady State Distribution of The Subnet with Different Firing R

First, we study the case where only class 1 editors are working on a single document portion. When the number of the subnets is increased, Fig.4.5 gives the normalization constant G computed through the convolution of the subnet. From G , the probabilities of some relevant states are calculated, given the LBSPN models which included different number of the editors. Then the throughput of every editor as well as that of the entire group is obtained, as presented in Fig.4.6.

	1 editor	2 editors	3 editors	4 editors	5 editors	6 editors	7 editors	8 editors
G	1	0.591568	0.296750	0.137146	0.060341	0.025691	0.010685	0.004366

Fig.4.5 Normalization Constant G

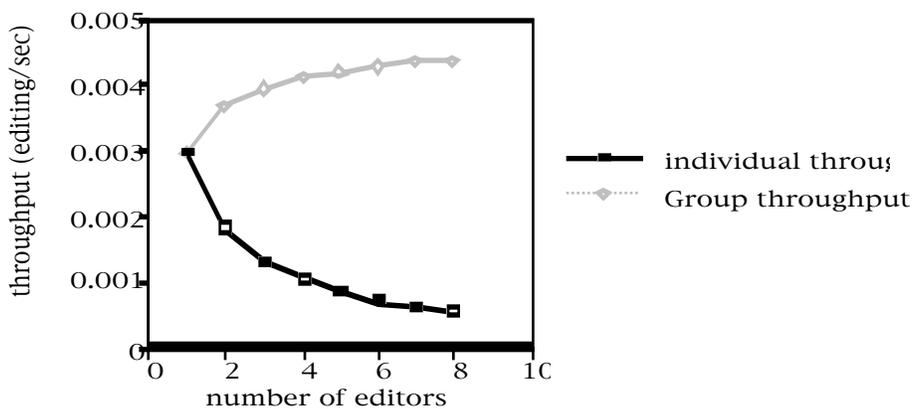


Fig. 4.6. Throughput of Case 1

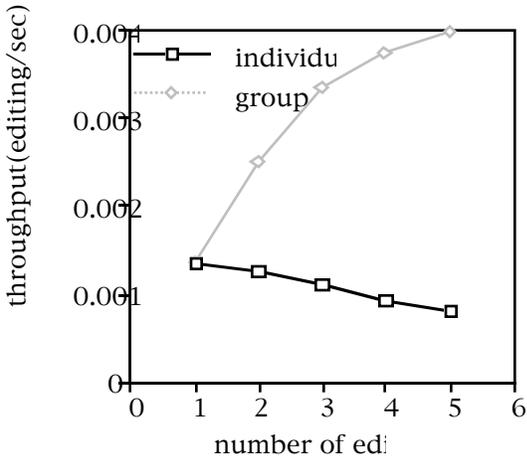
From the throughput given in Fig.4.6, we can see the system is saturated when more than 4 editors are involved. This indicates the number of editors that can be accommodated by the joint

editing system when only one of them can make the editing at one time. In order to improve the editing efficiency and motivate the participants in the editing, concurrent editing of the document should be provided.

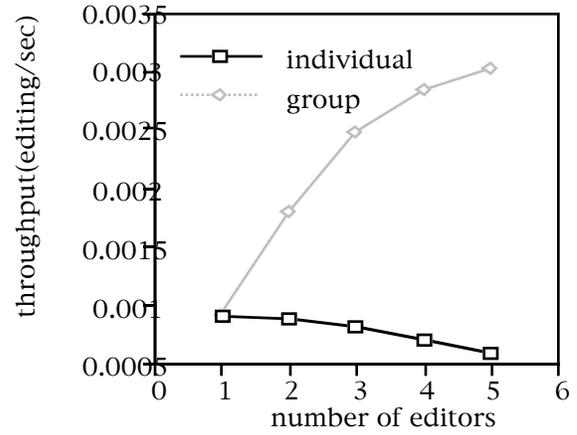
Next, the collaborating group with 3 concurrently accessible portions of the shared document is studied. Similarly, the number of the editors is increased. In this case, we assume that the editor works on the first portion with the rates of class 2 and the other two portions with the rates of class 1. This can be caused by either the different contents of the portion or the different media used in the portion. For instance, modifications made on a video clip takes different time from those made on a text piece in the portion. In Fig.4.7, the normalization constant G for the LBSPN models which include different number of the editors are given. Fig.4.8 presents the throughput of every editor and of the entire system.

	1 editor	2 editors	3 editors	4 editors	5 editors
G	0.3082096	0.072898	0.012665	0.001607	0.000167

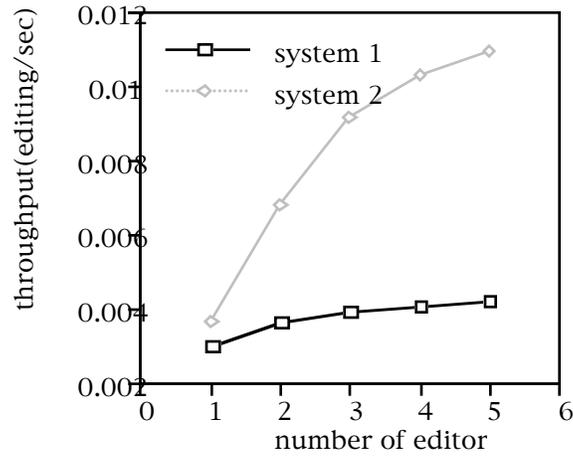
Fig.4.7 Normalization Constant G



(a) Throughput of class 1 edi



(b) Throughput of class 2 edi



system 1: Only one po
system 2: Three portions available cc

(c) Comparison of the thro

Fig. 4.8 Throughput of Case 2

From Fig.4.8, it can be observed that the system throughput has been increased. Meanwhile, the "block situation", represented by the decrease rate of the throughput of individual editors is alleviated. The comparison of the overall throughput of the two cases is depicted in Fig. 4.8 (c).

In this section, we employed a new performance model, the Locally Balanced Stochastic Petri Nets (LBSPN) to evaluate the multimedia collaborative application of joint editing. We first have developed a computational algorithm for the LBSPN models so that the product form solution of

LBSPN can be obtained efficiently. The algorithm is based on the convolution of the subnets in the LBSPN model.

5. Conclusions

In this paper, we applied the Locally Balanced Stochastic Petri Nets (LBSPN) to model the synchronization behavior in the distributed multimedia collaborative system. The LBSPN has product form solutions and can thus be solved for relatively large systems. In our studies, we have found that the LBSPN are useful in modeling the shared multimedia collaborative applications. We took the joint editing session as an example and obtained the performance results for the system design.

References

- [1] L. Li, L. Lamont, A. Karmouch and N. D. Georganas, "A Distributed Synchronization Control Scheme in a Group-oriented Multimedia Conferencing System", *Proc. II International Conference on Broadband Islands*, Greece, June 1993
- [2] L. Li, A. Karmouch and N. D. Georganas, "Performance Modelling of Distributed Integration in Real-Time Multimedia Scenarios", *Proc. ICCM (International Council for Computer Communications) Multimedia'93*, Banff, Canada, April, 1993
- [3] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System", *Communications of the ACM*, Vol.21, No.7, July 1978, pp.558-565
- [4] A. Goscinski, "Distributed Operating Systems: The Logical Design", Addison-Wesley Publishing Company, 1991
- [5] L. Lamont and N. D. Georganas, "Protocols for Real-Time Multimedia Conferencing", *Canadian Conference on Electrical & Computer Engineering*, Toronto, Sept. 1992
- [6] C. A. Ellis, S. J. Gibbs, and G. I. Rein, "Groupware --- Some issues and Experiences", *Communications of The ACM*, Vol.34, No.1, January 1991, pp38-58
- [7] R. Yavatkar, "MCP: A Protocol for Coordination and Temporal Synchronization in Collaborative Applications", *Proc. of the ICDCS'92*, pp.606-613
- [8] L Li, "Synchronization Studies in Multimedia Communications", Ph.D dissertation: Electrical Engineering Dept., University of Ottawa, Ottawa, Canada, Aug. 1993
- [9] M. K. Molloy, "Performance Analysis Using Stochastic Petri Nets", *IEEE Transactions on Computers*, Vol 31, No.9, Sept. 1992
- [10] M. Li, "Applying Decomposition and Aggregation Theory in The Analysis of Stochastic Petri Nets and Queueing Networks", Ph.D dissertation, Electrical Engineering Dept., University of Ottawa, Ottawa, Canada, Dec.1991
- [11] T. Murata, "Petri Nets: Properties, Analysis and Applications", *Proceedings of the IEEE*, vol.77, No.4m April, 1989

[12] A. A. Lazar and T. G. Robertazzi, "Markovian Petri Net Protocols with Product Solution", *Performance Evaluation*, Vol. 12, No.1, Jan. 1991

[13], M. Li and N. D. Georganas, "Exact Parametric Analysis of Stochastic Petri Nets", *Trans. on Computers*, Vol. 41, No.9, Sept. 1992, pp.1176-1180