

**A MARKOV CHAIN ANALYSIS  
OF A COMPUTER SUPPORTED INFORMATION NETWORK SYSTEM  
—ONE APPROACH TO A MODEL OF BUILDING CONSENSUS  
IN ELECTRONIC BRAINSTORMING—**

Shun-ichi Kigawa  
*Kantou-Gakuen University*

(Received December 21, 1988; Revised January 8, 1990)

*Abstract* The study of Group Decision Support Systems (GDSS) has matured considerably over the past five or six years. It is suggested that the chief value of GDSS is that they help improve communication. This paper reports on a Markov and a combinatorial model of a GDSS for idea generation, particularly we focus our issues on building consensus in Electronic Brainstorming (EBS) session of GDSS. Our model can explain one aspect of dynamics of an EBS and an idea generation process. Group members of GDSS are located in separated places. Each group member holds a set of different information at the beginning of the task. These group members communicate among themselves by a computer supported information network system. Three communication types are categorized according to the transmission mode, transaction pattern, number of transaction pairs and the transaction direction.

We compare three communication types A, B and C based on the performance measured by the wait time for consensus and the cost measured by the complexity of protocols and the number of message buffers in each node of the network. When we select the communication type in EBS, we must consider the trade-off among three communication types. Type A is the lowest one of the all based on the performance measured by the wait time but is the best one of the all based on the cost measurement. Type B is the highest one of all based on the performance measured by the wait time but is the worst one of the all based on the cost measurement. Type C is the middle of the all. For designers who are interested in developing their own GDSS facilities, these results are useful.

## 1. Introduction

Previously Schumpeter, J.A. has said that innovation is created by the innovative connection of production factors. The innovator invents the new product through making connection between market information and technological information. The more different the information, the greater the innovation. Nowadays we are attentive to the interaction between different social groups. To be truly innovative, an organization must foster the flow of new ideas. Brainstorming and other idea generation techniques were popularized during the 1950s to assist organizations in the innovation process. Advances in technology, especially in computer networks communication, microcomputer information procession power, and graphic information display, have created an environment where the opportunity for anonymous or registered, yet interactive, brainstorming using the computer is technically possible. The expansion of computer networks and microcomputers in organizations has resulted in a significant increase in computer-mediated communication [1, 2, 3]. This paper presents mathematical model of a group decision support system for idea generation. The models allow parallel processing of ideas.

The ultimate goal of GDSS is to improve decision quality and reduce meeting time. DeSanctis and Gallupe [4] defined GDSS as integrated computer-based systems which facilitate

solution of semi- or unstructured problems by a group who has joint responsibility for making the decision. Automated support for group decision making involves numerous disciplines including computer science, management information systems (MIS), organizational behavior, and speech/communication. The impact that we seek from the use of GDSS is primarily qualitative and not quantitative as almost all uses of Information Technology in the past –including DSS– have been. Many researchers have made experiments on the effects of GDSS rather than theoretical studies. One key finding from previous research studies is that the use of GDSS to support groups improves the quality of meeting outcome [5]. The improvement in decision quality results from one member thinking of a new and useful idea as a result of awareness of the contribution of other members. Typical four automated tool of GDSS to assist decision maker with the idea generation and analysis process are: (1) Electronic Brainstorming (group brainstorming version); (2) Electronic Brainstorming (individual brainstorming version); (3) Idea Structuring and Analysis; and (4) a callable voting program for interactive idea prioritization [6]. EBS is an idea and knowledge elicitation tool that enables participants to anonymously share ideas and comments.

In this paper, we suppose that EBS elicits many comments at a time and concurrently from each participants and we mainly focus our issues on EBS (group brainstorming version) and presents models for idea generation by human parallel processing and building consensus in EBS as an computer supported information network system. Hereafter a computer supported information network system will be termed simply as network.

In this paper three communication types  $A, B$  and  $C$  are discussed. We compute as performance of the network the mean and the standard deviation of the time taken to build consensus. We also analyze the cost of the network by considering the complexity of protocols and the number of message buffers in each node of the network. In the next section, we describe the model for networks in detail. We also give some considerations about cost of the network. Markov chain analysis and combinatorial analysis for the model is then given in Section 3. In Section 4 some numerical results are given to show the comparison of efficiency of three communication types. In the last section, we consider the cost–performance consideration of the network and give some sociological implications to EBS. In addition we suggest directions for extending the model.

## 2. Description of Computer Supported Information Network Systems

Information has two different aspects, one is its static aspect and others is its dynamic aspect. The static aspect of information is seen in its form as numerical data and manuals. The dynamic aspect of information is seen in the process where ideas are created through interaction in the network. We assume that an idea is made of messages and the term message precisely corresponds to the old combinatorial term, complexion. Each group member in the network has the set of messages labeled by  $1, 2, \dots, n$ . The subset of  $\{1, 2, \dots, n\}$  is also considered as a message. Issues relative to group dynamics are particularly central to GDSS design and use. Building GDSS requires a deep understanding of group process. We will consider primarily the dynamic aspect of information and the dynamics of an EBS process. In EBS, each group member works on the same task at the same time as all of the other group members.

The issue to be dealt with has been recognized by all group members before opening session. Group members can enter a comment or an idea to the question on their individual computer screen and send it out onto the network to be received by another group member. Each group member of the network makes choices of brainstorming partner using subjective selection criteria, therefore his selection fundamentally fluctuates. The ideas may either be tagged with the names of their creators, or the process may be entirely anonymous. After

receiving new ideas, a group member reads these ideas and contributes another new ideas. The idea is then sent out as before. The ideas generated during brainstorming circulate throughout the group. All participants interact randomly and continue the process in a relatively simultaneous and parallel fashion (See Figure 1.1). The final goal of EBS process is to build consensus. In this paper we assume that group members of GDSS are located in separated places. But in past research issues of factor demonstrated to promote effective and efficient GDSS utilization was addressed based on the findings of the one room studies, and little is known about differences that may result as a function of geographic dispersement of group members. In this paper, we have focused our attention on the group members located geographically apart. The objective of the network is for all group members to obtain perfect information, that is to say building consensus. By perfect information, we mean the set  $\{1, 2, \dots, n\}$ .

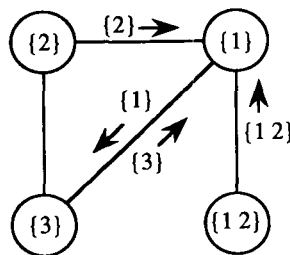


Figure 1.1 Communication of the ideas

### 2.1 Network Systems Model

In this paper, we define the network node as having the functions grouped into two categories. First, the source/destination functions pertain to providing an interface to the group member. These functions also include the management of end-to-end information transmission between a source node and a destination node. The second class of functions, store-and-forward functions, provides the services for information transmission through a network node. A given node receives a message, stores it, and then forwards it to the next node on the route. In this paper, if there is no confusion in a context, a network node is exchangeable for a group member, that is to say, a group member can use these functions.

Here we use the symbols in Flament, C. [7]. Let  $X = \{A_1, A_2, \dots, A_n\}$  be the set of  $n$  members in the network. Let  $\Gamma$  be a subset of  $X \times X$ . The elements of  $\Gamma$ , which is a pairs of group members, are called “links of the network”. Each directed arc in  $\Gamma$  represents one direction of link flow.

$$(A_i, A_j) \in \Gamma \text{ iff } A_i \text{ can address a communication to } A_j.$$

Given a link such as  $(A_i, A_j)$ ,  $A_i$  is called “the initial end of the link” and  $A_j$  is its “terminal end”. A link such as  $(A_i, A_i)$  is called a “loop”. A group member  $A_i$  is the sender of information and  $A_j$  is the receiver of that information if  $(A_i, A_j)$  is a link. The association between senders and receivers for the purpose of communication is called a connection. The group members communicate with each other and this communication is a message which is sent by a source node via a connection and received by the connected destination node. A directed path of a network is a sequence of links

$$\gamma(A_i, A_j) = (u_1, u_2, \dots, u_L) \text{ with } u_i \in \Gamma, i = 1, \dots, L,$$

such that the terminal point of each link  $u_i (i = 1, 2, \dots, L - 1)$  coincides with the initial point of the following link  $u_{i+1}$ . These directed paths through the network are also called

network routes or information routes. A message starting from its origination node follows the network route to reach its destination node. A directed path such that the terminal point of the link  $u_L$  coincides with the initial point of the link  $u_1$  is called a “circuit”.

Nodes, links and buffers in a network can be represented as a directed graph on which each has some number of buffers in it. We assume that each message occupies exactly one buffer. A buffer is either free (available) for a message or allocated to one. It is assumed that a message in a node is transmitted to the next node only after a buffer has been allocated for it in the next node, and we assume that the number of buffers in a node is enough for messages transmitted and a queue of messages does not occur in a node. Network activity of group member is defined by the allocation and request for resources. A unit of activity has one buffer allocated to it in a node, and can request one additional buffer in an adjacent node. Thus, we can define a network  $N$  in terms of the nodes, links and buffers.

$$N = (X, \Gamma, B),$$

where  $X$  is the set of nodes,  $\Gamma$  is the set of links of the network and  $B$  is defined such that  $B(x_i)$  or, simply,  $b_i$  is the number of buffers in node  $x_i$ . Suppose  $A_i$  wants to send message  $\{i\}$  to  $A_j$ ; if  $(A_i, A_j) \in \Gamma$ , there is no problem, and message  $\{i\}$  from  $A_i$  to  $A_j$  is said to be direct; if  $(A_i, A_j) \notin \Gamma$ , and if there exists in  $N$  a path

$$\gamma(A_i, A_j) = (u_1, u_2, \dots, u_L) \text{ with } u_i \in \Gamma, i = 1, \dots, L,$$

the link  $u_i (i = 1, \dots, L)$  can relay message  $\{i\}$  from  $A_i$  to  $A_j$ , and the message  $\{i\}$  from  $A_i$  to  $A_j$  is said to be indirect. Finally, if no directed path  $(A_i, A_j)$  exists in  $N$ , a message  $\{i\}$  from  $A_i$  to  $A_j$  is said to be impossible. Let  $Y = \{1, 2, \dots, n\}$  be the set of  $n$  messages.  $A_i$  holds  $\{i\} (i = 1, 2, \dots, n)$  at the beginning of the task of the network. It is assumed that each group member of the network makes choices of brainstorming partner using subjective selection criteria, therefore his selection can fundamentally fluctuate. So  $A_i$  can transmit his message to  $A_j$  with probability  $p_{ij}$ . We can now define a network state. A network state is defined by a vector such that  $i$ -th element of the vector is a message held by  $A_i$ . The initial state of the network is designated by  $(1, 2, \dots, n)$  as a vector. Next, if  $A_i$  transmits  $\{i\}$  to  $A_j$ ,  $A_j$  obtains message  $\{ij\}$  from pooling messages  $\{i\}$  and  $\{j\}$ . The subsequent state of the network becomes  $(1, \dots, i, \dots, j - 1, ij, j + 1, \dots, n)$ .

## 2.2 Connection and Transaction Patterns

A connection is a logical link between all the possible correspondents in a communication, each connection providing a potential communication path from senders to receivers. The possible connection patterns are one-to-one ( $1 - 1$ ), one-to-many ( $1 - n$ ), many-to-one ( $m - 1$ ) and many-to-many ( $m - n$ ) connections. This paper considers only one-to-one ( $1 - 1$ ) and many-to-one ( $m - 1$ ). Therefore we do not treat a broadcast medium. In fact all participants will be engaged on a communication transaction randomly and continue the process in a relatively simultaneous and parallel fashion. The connection therefore indicates only the potential correspondents. If a single sender in a source node can address only a single destination node, and if a receiver in a destination node can address only a single-source node, then the connection is termed one-to-one ( $1 - 1$ ). If the corresponding receiver can address multiple sources, then the connection is many-to-one ( $m - 1$ ).

The communication transaction determines the actual correspondents involved in communicating on a connection. The transaction pattern defines the number of sources and receivers involved in a single use of the communication primitives. The transaction patterns treated in this paper are one-to-one ( $1 - 1$ ) and many-to-one ( $m - 1$ ). In one-to-one transaction, one sender and one receiver are involved in a particular information transaction. This

is the simplest and most common and is used with all connection patterns and all primitives. In many-to-one transaction, many senders and only one receiver are involved in a particular information transaction. This is very unusual but can be used to collect information from multiple sources at exactly the same time and very difficult to implement without global time. The use of  $m - 1$  can be simulated by the use of multiple  $1 - 1$  transactions, but the protocols can become fairly complex.

### 2.3 Transmission Modes

There are a number of characteristics of the transmission mode whether parallel or serial, full or half duplex, digital or analogue signaling, but the we consider only full or half duplex mode. A communication circuit can be used in one of the following modes (See Figure 2.1):

- Simplex transmission: Information is transferred in one direction only and never in the other direction.
- Half-duplex transmission: Information is transferred in either direction but not in both directions simultaneously.
- Full-duplex transmission: Information is transferred simultaneously in both directions. Sometimes this mode of transmission is also referred as two-way simultaneous transmission. The full-duplex transmission offers maximum function and performance for a transmission link. In order to obtain a full-duplex operation, both the modems and the protocols controlling the transmission must provide full-duplex operation. The full-duplex operation also requires special consideration for buffer management. For example, simultaneous read and write operation also require simultaneous release and allocation of buffers in the buffer pool.

The terms simplex, half-duplex and full-duplex can also be used to describe an intrinsic property of the circuit.

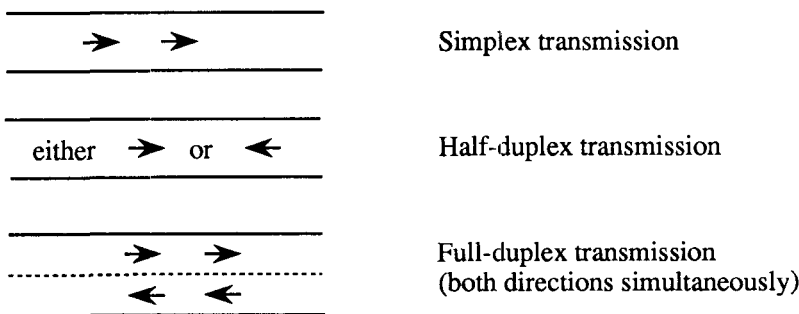


Figure 2.1 Simplex-duplex transmission

In this paper we consider three communication types. (See Table 2.1).

- Type A: At a given time, only one group member of the network can transmit his message to other one group member, and the message is permitted to flow only in one direction (half-duplex,  $1 - 1$  and one pair).
- Type B: Many pairs in the network may concurrently and simultaneously transmit their messages to each other, and the messages may concurrently and simultaneously flow in both directions (full-duplex,  $m - 1$  and  $n$  pairs).
- Type C: At a given time, between only two members in the network the messages may simultaneously flow in both directions (full-duplex,  $1 - 1$  and one pair).

Table 2.1. Three communication types

Type	Transmission mode	Transaction pattern	Number of transaction pairs	Transaction direction
A	Half-duplex	1 – 1	1	Unidirectional
B	Full-duplex	$m - 1$	$n$	Bidirectional
C	Full-duplex	1 – 1	1	Bidirectional

## 2.4 Interconnection Structure [8]

Network topology defines the interconnection structure of nodes and links. In the following we describe the typical four types of interconnection from the following points.

- expansion cost—the incremental cost of adding another member;
- reliability—dependency on a single component for network operation;
- software complexity—the complexity of protocols required;
- performance—in terms of throughput or delay, efficiency of the problem solving and member satisfaction.

[Complete interconnection]

Each member is connected by a dedicated point-to-point link to every other member (See Figure 2.2).

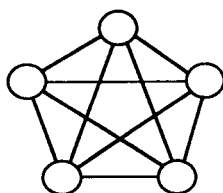


Figure 2.2 Fully connected

The multiple links can operate in parallel resulting in high throughput and the delay is low as there are no intermediate nodes. The communication software can be simple as no routing is needed and links are end-to-end. A fully connected network can give very high reliability if routing is included to take advantage of the alternative paths available after failure of a link. This then requires store-and-forward transmission via an intermediate node. The main disadvantage is that the topology is very expensive:  $n$  nodes require  $(n - 1)n/2$  links and each node must have  $n - 1$  interfaces for all the links. Fully connected networks are not very common because of the high cost. They are sometimes used for small local networks or in military applications where redundancy is of prime consideration. They are models of the group communication which consist of members who have equal status and equal effect.

[Mesh (partial interconnection)]

A mesh has point-to-point links between some nodes i.e. all nodes are not directly connected (See Figure 2.3). Store-and-forward transmission is required between some pairs of nodes. If each node is connected to at least two others, alternative paths are easily provided in case of failure of a links or node. Mesh networks are commonly used for wide area networks as it is

possible to provide high reliability at comparatively low cost. Network delay depends on the number of intermediate nodes or switching nodes which can be fairly high for a large network. The communication system must incorporate routing strategies and generally provides both point-to-point and end-to-end error and flow control, which increases its complexity.

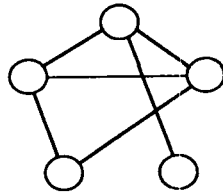


Figure 2.3 Mesh

[Star]

All stations are connected via a single link to a central core node (See Figure 2.4) which results in a low expansion cost, simple table lookup routing in the core node and a maximum delay of only one intermediate node. The main disadvantage is poor reliability, as a failure of a link isolates a node. Failure of the central core stops all communication so redundancy is sometimes provided at the core. Throughput is limited by that of the central core, which may be a bottleneck. The network is usually not homogeneous in that the central core is different from the remote node, i.e. all nodes are not interchangeable. The functions and the power of the network is concentrated at the central core, so the remote node can not contribute to the network and the central core control the remote node.

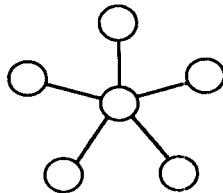


Figure 2.4 Star

[Tree or hierarchical network]

This is really an extension of the star topology and so has very similar characteristics (See Figure 2.5). It is often used for terminal networks, where the top level is a central computer, the intermediate level consists of remote multiplexors and the lowest level is terminals. It is also used for process control as it reflects the hierarchical organization of the control system, but a single failure can isolate part of the network. As to the communication network, this network is the model of a bureaucratic organization.

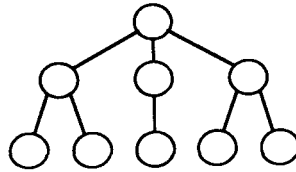


Figure 2.5 Tree

The network categories defined by Shaw, M.E. [9] are the centralized network and the distributed network. According to this definition the complete interconnection and the partial interconnection are categorized in the distributed network and the star and the tree are categorized in the centralized network. He concluded that for the relatively less complex problems or questions (e.g. identification of symbol, character or collar) the centralized network is more effective or efficient than the distributed network but for issues associated with complex problems or questions (e.g. group deliberation) the distributed network is more effective or efficient than the centralized network. In member satisfaction with group process, the distributed network is better than the centralized network in both issues. Our model can explain the above three types of network topology by controlling probability  $p_{ij}$ . Our model can not treat a serial bus and a ring topology. In this paper we do not consider deeply the network topology and its effect on cost and performance of the network.

## 2.5 Network congestion and flow control [10]

Networks consist of finite resources, such as the message buffers that reside in network nodes. An unrestricted flow of messages into the network may lead to congestion. When the network becomes congested, it transmits few, if any, messages and congestion rapidly spreads. If no remedy is applied, network congestion can ultimately lead to deadlock situation in which no message flow takes place due to lack of buffers. There is high possibility of deadlock in type, *B, C*, but low in type *A*. Congestion control aims to prevent the network reaching saturation by limiting the amount of traffic entering the network. Congestion control can be defined as the set of mechanisms used to prevent congestion or deadlock and to manage shared network resources in order to optimize performance, it is really a global mechanism. Most congestion-control mechanisms are based on reservation of resources (e.g. buffers) or discarding of messages. Flow control is used to regulate the flow of information between a pair of communicating entities, to prevent one from sending more information than the partner can handle. Flow and congestion control are closely related, although flow-control mechanisms can not prevent congestion. Deadlock can occur in store-and-forward networks when nodes are unable to exchange message because of lack of buffers. When two neighboring nodes connected by a direct link are deadlocked, this deadlock is sometimes called a direct store-and-forward deadlock. In this case, all the buffers in the first node contain message bound for the second node and all the buffers in the second node contain messages bound for the first. Neither node can receive a message due to lack of buffer space and no buffers can be freed until a message has been successfully transmitted. The situation can only be resolved by one node discarding his message. This can be prevented by not allowing all buffers to be assigned to a single output queue and reserving buffers for input lines. Indirect or circular store-and-forward deadlock involves a 'loop' of nodes each filled with message destined for the next node in the loop. This type of traffic pattern is unlikely to occur in a store-and-forward network but it is theoretically possible. It can be prevented by reserving two buffers in each node for use as overflow if deadlock is detected. Thus at least one



overflow message at a time can be delivered. Deadlock caused by lack of buffer space is called buffer deadlock. Consider the network shown in Figure 2.6 which indicates Type B as an example. Let us assume that nodes  $A_1, A_2$  and  $A_3$  have one message buffer available to store new message. Furthermore, assume that each message in nodes  $A_1, A_2$  and  $A_3$  required assignment of a buffer in nodes  $A_1, A_2$  and  $A_3$  as denoted in Figure 2.6 before the message can be transmitted. If message in nodes  $A_1, A_2$  and  $A_3$  are also waiting for a buffer in nodes  $A_1, A_2$  and  $A_3$  respectively, then nodes  $A_1, A_2$  and  $A_3$  are in store-and-forward buffer deadlock. If each node have three buffers, then there is no buffer deadlock. In Type C, two buffers is enough for each node to prevent buffer deadlock. In Type A, one buffer is enough for each node. Cost measured by the number of buffers in nodes is lowest in Type A, middle in Type C and highest in Type B.

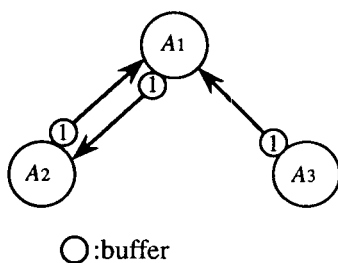


Figure 2.6 Buffer deadlock in case of Type B

### 3. Analysis of performance of the network

In the real world, the number  $n$  of group members and messages in such a network as described above is in general quite large. If the number of group members is quite large, it is difficult, or impossible, to get an exact solution. Therefore we deal with the case where there are 2 and 3 group members and messages. In this section, Markov chain method and combinatorial method are applied to 3 types of network. But the methods of obtaining solutions are general and any extension to cases that have more than 3 members and messages is straightforward.

#### 3.1 Analysis of the case in which the network has 2 members and the communication type is A

Let  $X = \{A_1, A_2\}$  be the set of 2 members in the network. Let  $Y = \{1, 2\}$  be the set of 2 messages.  $A_i$  holds message  $\{i\}(i = 1, 2)$  at the beginning of the task  $T$ . At every discrete time, either  $A_1$  or  $A_2$  can transmit his message to the other but they can not transmit simultaneously. The sender is selected with probability  $r_1$  for  $A_1$  and  $r_2$  for  $A_2, (r_1 + r_2 = 1)$ . After the selection of the sender,  $A_i$  can transmit his message to  $A_j$  with probability  $p_{ij}, (i, j = 1, 2)$ . The above description is shown in Figure 3.1.

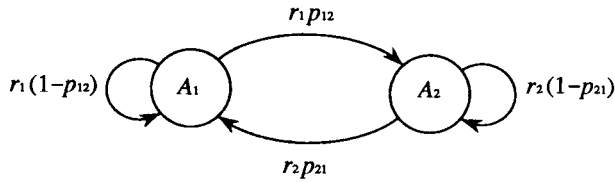


Figure 3.1.

In Figure 3.1,  $r_i(1 - p_{ij}) = r_i p_{ii}$  ( $i \neq j; i, j = 1, 2$ ) is the probability with which  $A_i$  does not transmit his message. The states of the network are  $s_1 = (1, 2), s_2 = (1, 12), s_3 = (12, 2)$ , and  $s_4 = (12, 12), s_1$  is the initial state,  $s_4$  is the persistent and the absorbing state. The transition diagram between the state space  $S = \{s_1, s_2, s_3, s_4\}$  is given in Figure 3.2.

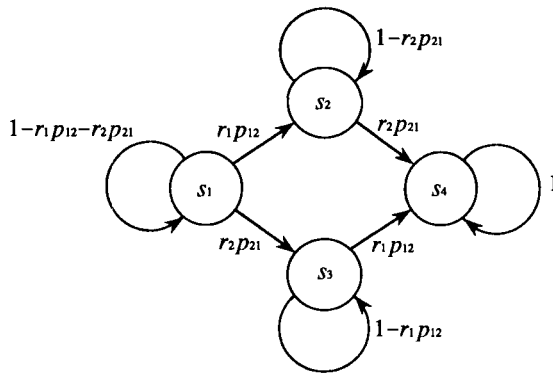


Figure 3.2

Let the corresponding transition probability matrix be given by

$$P = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 & s_4 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{bmatrix} 1 - r_1 p_{12} - r_2 p_{21} & r_1 p_{12} & r_2 p_{21} & 0 \\ 0 & 1 - r_2 p_{21} & 0 & r_2 p_{21} \\ 0 & 0 & 1 - r_1 p_{12} & r_1 p_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

This chain has three transient states,  $\{s_1, s_2, s_3\}$ , and one absorbing state  $s_4$ . By the fundamental matrix method [11], the way to calculate the fundamental matrix  $N$  is

$$N = (I - Q)^{-1} = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{bmatrix} 1/(r_1 p_{12} + r_2 p_{21}) & r_1 p_{12}/\{r_2 p_{21}(r_1 p_{12} + r_2 p_{21})\} & r_2 p_{21}/\{r_1 p_{12}(r_1 p_{12} + r_2 p_{21})\} \\ 0 & 1/(r_2 p_{21}) & 0 \\ 0 & 0 & 1/(r_1 p_{12}) \end{bmatrix} \end{matrix},$$

where  $Q$  is the substochastic matrix corresponding to transitions among the transient states of a finite Markov chain. Therefore  $Q$  is the  $3 \times 3$  matrix obtained from  $P$  by eliminating its 4th row and 4th column.  $I$  is the identity matrix.

The mean absorption time from state  $s_1$  is the sum of the entries in the first row so

$$\mu_{s_1} = \frac{r_1 r_2 p_{12} p_{21} + r_1^2 p_{12}^2 + r_2^2 p_{21}^2}{r_1 r_2 p_{12} p_{21} (r_1 p_{12} + r_2 p_{21})}.$$

Similarly,  $\mu_{s_2} = 1/(r_2 p_{21}), \mu_{s_3} = 1/(r_1 p_{12})$ .

If  $\mathbf{1}'$  denotes a column vector of ones then  $N\mathbf{1}'$  is a vector,  $\boldsymbol{\mu}'$ , in which the  $i$ -th entry is the mean absorption time from the  $i$ -th transient state. In our case we get

$$\boldsymbol{\mu}' = N\mathbf{1}' = (\mu_{s_1}, \mu_{s_2}, \mu_{s_3})'.$$

In addition to finding the mean absorption time to the absorbing state, the fundamental matrix  $N$  can be used to find the second moment of the absorption time. In particular the vector of second moment is given by  $\boldsymbol{\mu}^{(2)'} = N(2\boldsymbol{\mu}' - \mathbf{1}')$  where  $\boldsymbol{\mu}$  is the vector of mean absorption times. Here we have

$$\boldsymbol{\mu}^{(2)'} = (\mu_{s_1}^{(2)}, \mu_{s_2}^{(2)}, \mu_{s_3}^{(2)})',$$

where

$$\mu_{s_1}^{(2)} = \frac{2(r_1 p_{12})(r_2 p_{21})(r_1 p_{12} + r_2 p_{21})^2 + \{(r_1 p_{12})^3(2 - r_2 p_{21}) + (r_2 p_{21})^3(2 - r_1 p_{12}) - (r_1 p_{12})^2(r_2 p_{21})^2\} \cdot (r_1 p_{12} + r_2 p_{21}) - 2(r_1 p_{12})^2(r_2 p_{21})^2}{(r_1 p_{12})^2(r_2 p_{21})^2(r_1 p_{12} + r_2 p_{21})^2}$$

The vector of standard deviations of the absorption times is given by

$$\mathbf{SD}' = (\sqrt{\mu_{s_1}^{(2)} - \mu_{s_1}^2}, \sqrt{\mu_{s_2}^{(2)} - \mu_{s_2}^2}, \sqrt{\mu_{s_3}^{(2)} - \mu_{s_3}^2})'. \tag{3.1}$$

### 3.2 Analysis of the case in which the network has 2 members and the communication type is B

Now let us suppose,  $X = \{A_1, A_2\}, Y = \{1, 2\}$  and  $A_i$  holds message  $\{i\} (i = 1, 2)$  at the beginning of the task  $T$ . At every discrete time  $A_1$  and  $A_2$  can communicate each other simultaneously. It is shown in Figure 3.3.

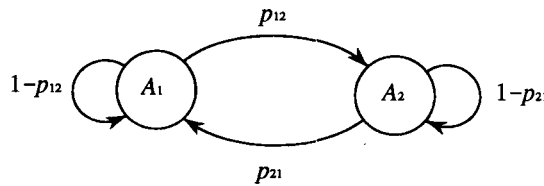


Figure 3.3

The states of the network are the same as section 3.1, but the transition diagram between states is given in Figure 3.4

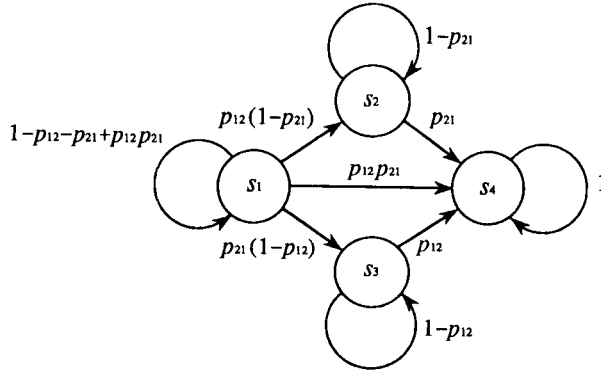


Figure 3.4

The corresponding transition probability matrix is given by

$$P = \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \end{array} \begin{bmatrix} 1 - p_{12} - p_{21} + p_{12}p_{21} & p_{12}(1 - p_{21}) & p_{21}(1 - p_{12}) & p_{12}p_{21} \\ 0 & 1 - p_{21} & 0 & 0 \\ 0 & 0 & 1 - p_{12} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \end{array}$$

By the same method as in section 3.1, the mean absorption times from transition states,  $s_1, s_2$  and  $s_3$  respectively are calculated from the fundamental matrix  $N$ ,

$$N = (I - Q)^{-1} = \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \left[ \begin{array}{c} \begin{array}{cc} 1/(p_{12} + p_{21} - p_{12}p_{21}) & \{p_{12}(1 - p_{21})\}/\{p_{21}(p_{12} + p_{21} - p_{12}p_{21})\} \\ 0 & 1/p_{21} \end{array} \\ \begin{array}{c} s_3 \\ \{p_{21}(1 - p_{12})\}/\{p_{12}(p_{12} + p_{21} - p_{12}p_{21})\} \\ 0 \\ 1/p_{12} \end{array} \end{array} \right]$$

A vector  $\mu'$ , in which the  $i$ -th entry is the mean absorption time from the  $i$ -th transient state, is

$$\mu' = N\mathbf{1}' = \left( \frac{p_{12}^2 + p_{21}^2 + p_{12}p_{21}(1 - p_{12} - p_{21})}{p_{12}p_{21}(p_{12} + p_{21} - p_{12}p_{21})}, \frac{1}{p_{21}}, \frac{1}{p_{12}} \right)'$$

The vector of second moments  $\mu^{(2)}$  is given by

$$\mu^{(2)'} = (\mu_{s_1}^{(2)}, \mu_{s_2}^{(2)}, \mu_{s_3}^{(2)})'$$

where

$$\mu_{s_1}^{(2)} = \frac{p_{12}p_{21}\{2p_{12}^2 + 2p_{21}^2 + p_{12}p_{21}(2 - 3p_{12} - 3p_{21} + p_{12}p_{21})\} + p_{12}^3(p_{12} + p_{21} - p_{12}p_{21})}{p_{12}^2p_{21}^2(p_{12} + p_{21} - p_{12}p_{21})^2} \cdot \frac{(2 - p_{21})(1 - p_{21}) + p_{21}^3(p_{12} + p_{21} - p_{12}p_{21})(2 - p_{12})(1 - p_{12})}{p_{12}^2p_{21}^2(p_{12} + p_{21} - p_{12}p_{21})^2},$$

$$\mu_{s_2}^{(2)} = (2 - p_{21})/p_{21}^2, \mu_{s_3}^{(2)} = (2 - p_{12})/p_{12}^2.$$

The vector of standard deviations of the absorption times is the same vector given by (3.1).

**3.3 Analysis of the case in which the network has 3 members and the communication type is A**

In the following three cases, the method of obtaining the means and the standard deviations of the absorption times is different from the previous cases (2 members cases). Let  $X = \{A_1, A_2, A_3\}$  be the set of 3 members in the network. Let  $Y = \{1, 2, 3\}$  be the set of 3 messages.  $A_i$  holds message  $\{i\} (i = 1, 2, 3)$  at the beginning of the task  $T$ . At every discrete time, only one of 3 members  $A_1, A_2$  and  $A_3$  is selected as the sender with probabilities  $r_1, r_2$  and  $r_3$  respectively. After the selection of the sender, the selected sender  $A_i$  can transmit his message to  $A_j$  with probability  $p_{ij} (i, j = 1, 2, 3)$ . This fact is shown in Figure 3.5.

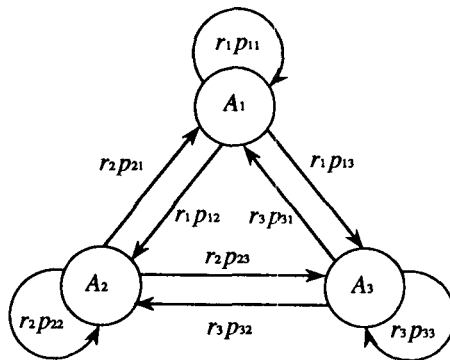


Figure 3.5

In Figure 3.5  $r_i p_{ii} = r_i (1 - \sum_{j \neq i} p_{ij}) (i \neq j; i, j = 1, 2, 3)$  are the probabilities that  $A_i$  does not transmit his message. Next we consider the states of the network. The messages held by  $A_1$  as a result of the pooling of the messages are  $\{1\}, \{12\}, \{13\}$  or  $\{123\}$ . For  $A_2$ , these are  $\{2\}, \{12\}, \{23\}$  or  $\{123\}$ . For  $A_3$ , these are  $\{3\}, \{13\}, \{23\}$  or  $\{123\}$ . By the rule of the product of combinatorial analysis, the states of the network are  $s_1 = (1, 2, 3), s_2 = (1, 2, 13), \dots, s_{64} = (123, 123, 123)$  and the number of states is  $4^3 = 64$ . The initial state is  $s_1 = (1, 2, 3)$  and the absorbing state is  $s_{64} = (123, 123, 123)$  and the transient states are  $\{s_1, s_2, \dots, s_{63}\}$ .

Thus the state space of the network is  $S = \{s_1, s_2, \dots, s_{64}\}$ .

[Algorithm to find the transition matrix]

Table 3.1 shows the patterns of sender and receiver in the network and their probabilities. The pattern of sender  $A_i$  and receiver  $A_j$  is denoted by  $[ij]$  in the Table 3.1.

Table 3.1

Pattern	Prob.	Pattern	Prob.	Pattern	Prob.
[11]	$r_1p_{11}$	[21]	$r_2p_{21}$	[31]	$r_3p_{31}$
[12]	$r_1p_{12}$	[22]	$r_2p_{22}$	[32]	$r_3p_{32}$
[13]	$r_1p_{13}$	[23]	$r_2p_{23}$	[33]	$r_3p_{33}$

When we are in the initial state (1,2,3) of the network, Table 3.2 shows the subsequent states of the network from the initial state through all patterns of Table 3.1.

Table 3.2

Pattern	State	Pattern	State	Pattern	State
[11]	(1,2,3)	[21]	(12,2,3)	[31]	(13,2,3)
[12]	(1,12,3)	[22]	(1,2,3)	[32]	(1,23,3)
[13]	(1,2,13)	[23]	(1,2,23)	[33]	(1,2,3)

From Tables 3.1 and 3.2, we get the possible subsequent states of the network and their transition probabilities from the initial state by adding the probabilities in Table 3.1 having the same subsequent states in Table 3.2. These are given by Table 3.3. This algorithm in a sense is the Path Integral of Quantum Mechanics.

Table 3.3

State	Prob.
(1,2,3)	$r_1p_{11} + r_2p_{22} + r_3p_{33}$
(1,12,3)	$r_1p_{12}$
(1,2,13)	$r_1p_{13}$
(12,2,3)	$r_2p_{21}$
(1,2,23)	$r_2p_{23}$
(13,2,3)	$r_3p_{31}$
(1,23,3)	$r_3p_{32}$

The subsequent states and their transition probabilities from the other states are given by the same algorithm. Therefore we obtain the corresponding transition probability matrix  $P$ . The entries in the first row of matrix  $P$  are the probabilities in Table 3.3. The mean absorption times from transient states  $\{s_1, s_2, \dots, s_{63}\}$  are calculated from the fundamental matrix  $N = (I - Q)^{-1}$ . A vector  $\boldsymbol{\mu}'$ , in which the  $i$ -th entry is the mean absorption time from the  $i$ -th transient state, is  $\boldsymbol{\mu}' = N\mathbf{1}' = (\mu_{s_1}, \mu_{s_2}, \dots, \mu_{s_{63}})'$ . The vector of second moments is given by

$$\boldsymbol{\mu}^{(2)'} = N(2\boldsymbol{\mu}' - \mathbf{1}') = (\mu_{s_1}^{(2)}, \mu_{s_2}^{(2)}, \dots, \mu_{s_{63}}^{(2)})'$$

The vector of standard deviation **S.D** of the absorption times is given by

$$\mathbf{SD}' = (\sqrt{\mu_{s_1}^{(2)} - \mu_{s_1}^2}, \sqrt{\mu_{s_2}^{(2)} - \mu_{s_2}^2}, \dots, \sqrt{\mu_{s_{63}}^{(2)} - \mu_{s_{63}}^2})'$$

These quantities can be calculated symbolically but the transition probability matrix  $P$  is too large and tedious to write down symbolically and we are particularly interested in the first element of  $\mu'$  and  $SD'$  so we will not write them explicitly here but in Section 4 these quantities are calculated numerically.

### 3.4 Analysis of the case in which the network has 3 members and the communication type is B

In this subsection, let us suppose there are 3 members and 3 different messages and  $A_i$  holds message  $\{i\}$  ( $i = 1, 2, 3$ ) at the beginning of the task  $T$  the same as Section 3.3. But at every discrete time,  $A_1, A_2$  and  $A_3$  can communicate with each other in simultaneous and parallel fashion. Between any two members messages may simultaneously flow in both directions. This is shown in Figure 3.6.

In Figure 3.6,  $p_{ii} = 1 - \sum_{j \neq i} p_{ij}$  ( $i \neq j; i, j = 1, 2, 3$ ) are the probabilities that  $A_i$  does not transmit his message. The states of the network are the same as Section 3.3, therefore the state space is  $S = \{s_1, s_2, \dots, s_{64}\}$ . The initial state is  $s_1$  and the absorbing state is  $s_{64}$ .

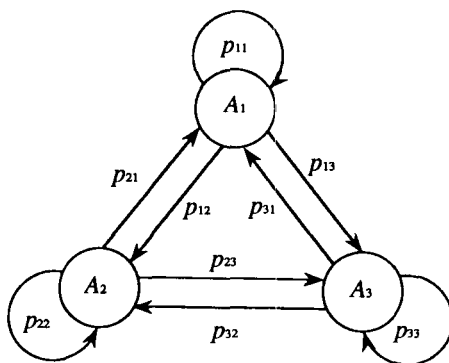


Figure 3.6

[Algorithm to find the transition matrix]

The same algorithm is used as in Section 3.3. We continue to use the framework of Section 3.3. Table 3.4 shows the patterns of sender and receiver in the network and their probabilities. For example the pattern that  $A_1$  transmits to  $A_2, A_2$  transmits to  $A_1$  and  $A_3$  transmits to  $A_1$  simultaneously is denoted by  $[12, 21, 31]$  in Table 3.4. The probability of this event is  $p_{12}p_{21}p_{31}$ . The number of patterns is  $3^3 = 27$ . For example, when we are in the initial state  $(1, 2, 3)$ , the subsequent state of the network through the pattern  $[12, 21, 31]$  is  $(123, 12, 3)$ .

Table 3.4

Pattern	Prob.	Pattern	Prob.	Pattern	Prob.
[11,21,31]	$p_{11}p_{21}p_{31}$ ①	[12,21,31]	$p_{12}p_{21}p_{31}$ ⑩	[13,21,31]	$p_{13}p_{21}p_{31}$ ⑲
[11,21,32]	$p_{11}p_{21}p_{32}$ ②	[12,21,32]	$p_{12}p_{21}p_{32}$ ⑪	[13,21,32]	$p_{13}p_{21}p_{32}$ ⑳
[11,21,33]	$p_{11}p_{21}p_{33}$ ③	[12,21,33]	$p_{12}p_{21}p_{33}$ ⑫	[13,21,33]	$p_{13}p_{21}p_{33}$ ㉑
[11,22,31]	$p_{11}p_{22}p_{31}$ ④	[12,22,31]	$p_{12}p_{22}p_{31}$ ⑬	[13,22,31]	$p_{13}p_{22}p_{31}$ ㉒
[11,22,32]	$p_{11}p_{22}p_{32}$ ⑤	[12,22,32]	$p_{12}p_{22}p_{32}$ ⑭	[13,22,32]	$p_{13}p_{22}p_{32}$ ㉓
[11,22,33]	$p_{11}p_{22}p_{33}$ ⑥	[12,22,33]	$p_{12}p_{22}p_{33}$ ⑮	[13,22,33]	$p_{13}p_{22}p_{33}$ ㉔
[11,23,31]	$p_{11}p_{23}p_{31}$ ⑦	[12,23,31]	$p_{12}p_{23}p_{31}$ ⑯	[13,23,31]	$p_{13}p_{23}p_{31}$ ㉕
[11,23,32]	$p_{11}p_{23}p_{32}$ ⑧	[12,23,32]	$p_{12}p_{23}p_{32}$ ⑰	[13,23,32]	$p_{13}p_{23}p_{32}$ ㉖
[11,23,33]	$p_{11}p_{23}p_{33}$ ⑨	[12,23,33]	$p_{12}p_{23}p_{33}$ ⑱	[13,23,33]	$p_{13}p_{23}p_{33}$ ㉗

When we are in the state (1, 23, 3), Table 3.5 shows the subsequent states of the network through all patterns of Table 3.4.

Table 3.5

Pattern	State	Pattern	State	Pattern	State
[11,21,31]	(123,23,3)	[12,21,31]	(123,123,3)	[13,21,31]	(123,23,13)
[11,21,32]	(123,23,3)	[12,21,32]	(123,123,3)	[13,21,32]	(123,23,13)
[11,21,33]	(123,23,3)	[12,21,33]	(123,123,3)	[13,21,33]	(123,23,13)
[11,22,31]	(13,23,3)	[12,22,31]	(13,123,3)	[13,22,31]	(13,23,13)
[11,22,32]	(1,23,3)	[12,22,32]	(1,123,3)	[13,22,32]	(1,23,13)
[11,22,33]	(1,23,3)	[12,22,33]	(1,123,3)	[13,22,33]	(1,23,13)
[11,23,31]	(13,23,23)	[12,23,31]	(13,123,23)	[13,23,31]	(13,23,123)
[11,23,32]	(1,23,23)	[12,23,32]	(1,123,23)	[13,23,32]	(1,23,123)
[11,23,33]	(1,23,23)	[12,23,33]	(1,123,23)	[13,23,33]	(1,23,123)

From Tables 3.5 and 3.4, we get the possible subsequent states of the network and their transition probabilities from the transient state (1, 23, 3) by adding the probabilities in Table 3.4 having the same subsequent states in Table 3.5. These are given by Table 3.6.



Table 3.6

State	Prob.
(123,23,3)	①+②+③
(13,23,3)	④
(1,23,3)	⑤+⑥
(13,23,23)	⑦
(1,23,23)	⑧+⑨
(123,123,3)	⑩+⑪+⑫
(13,123,3)	⑬
(1,123,3)	⑭+⑮
(13,123,23)	⑯
(1,123,23)	⑰+⑱
(123,23,13)	⑲+⑳+㉑
(13,23,13)	㉒
(1,23,13)	㉓+㉔
(13,23,123)	㉕
(1,23,123)	㉖+㉗

The subsequent states and their transition probabilities from the other states are given by the same algorithm. The corresponding transition probability matrix  $P$ , the vector of the mean absorption times and the vector of the standard deviations of the absorption times are constructed from the same algorithm as in Section 3.3.

### 3.5 Analysis of the case in which the network has 3 members and the communication type is C

In this subsection, let us assume there are 3 members and 3 different messages and  $A_i$  holds message  $\{i\}(i = 1, 2, 3)$  at the beginning of the task  $T$  just as in Section 3.3. But at every discrete time, only two of the three members  $A_1, A_2$  and  $A_3$  can communicate simultaneously with each other. This is shown in Figure 3.7.

At first, one of three patterns (a), (b) and (c) is selected with probabilities  $r'_1, r'_2$  and  $r'_3$  respectively ( $r'_1 + r'_2 + r'_3 = 1$ ). After the selection,  $A_i$  can transmit his message to  $A_j$  with probability  $p_{ij}(i, j = 1, 2, 3)$  and  $p_{ii} = 1 - p_{ij}(i \neq j; i, j = 1, 2, 3)$  are the probabilities that  $A_i$  does not transmit his message to  $A_j$ .

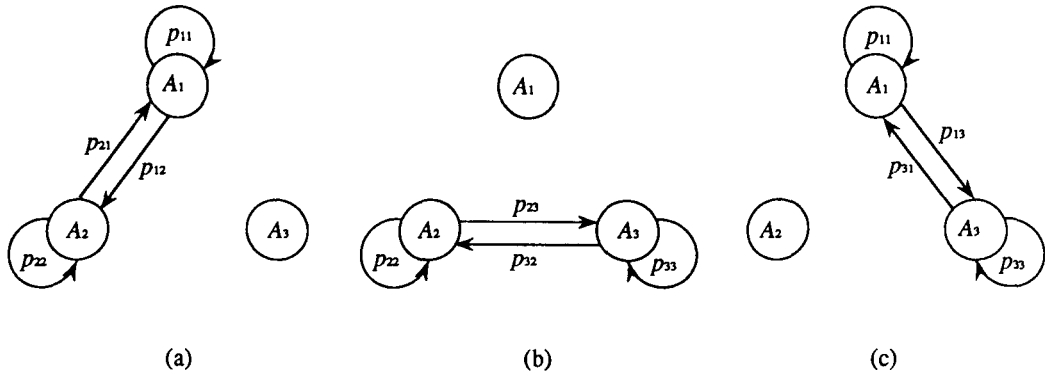


Figure 3.7

The states of the network are the same as Section 3.3, therefore the state space is  $S = \{s_1, s_2, \dots, s_{64}\}$ .

[Algorithm to find the transition matrix]

Table 3.7 shows the patterns of sender and receiver in the network and their probabilities.

Table 3.7

Pattern	Prob.	Pattern	Prob.	Pattern	Prob.
[11,21]	$r_1 p_{11} p_{21}$ ①	[22,32]	$r_2 p_{22} p_{32}$ ⑤	[11,31]	$r_3 p_{11} p_{31}$ ⑨
[11,22]	$r_1 p_{11} p_{22}$ ②	[22,33]	$r_2 p_{22} p_{33}$ ⑥	[11,33]	$r_3 p_{11} p_{33}$ ⑩
[12,21]	$r_1 p_{12} p_{21}$ ③	[23,32]	$r_2 p_{23} p_{32}$ ⑦	[13,31]	$r_3 p_{13} p_{31}$ ⑪
[12,22]	$r_1 p_{12} p_{22}$ ④	[23,33]	$r_2 p_{23} p_{33}$ ⑧	[13,33]	$r_3 p_{13} p_{33}$ ⑫

When we find ourselves in the initial state described by (1, 2, 3), Table 3.8 shows subsequent states of the network through all patterns of Table 3.7.

Table 3.8

Pattern	State	Pattern	State	Pattern	State
[11,21]	(12,2,3)	[22,32]	(1,23,3)	[11,31]	(13,2,3)
[11,22]	(1,2,3)	[22,33]	(1,2,3)	[11,33]	(1,2,3)
[12,21]	(12,12,3)	[23,32]	(1,23,23)	[13,31]	(13,2,13)
[12,22]	(1,12,3)	[23,33]	(1,2,23)	[13,33]	(1,2,13)

From Tables 3.7 and 3.8, we get the possible subsequent states of the network and their transition probabilities from the initial state by adding the probabilities in Table 3.7 having the same subsequent states in the Table 3.8. These are given by Table 3.9.

Table 3.9

State	Prob.
(12,2,3)	①
(1,2,3)	②+⑥+⑩
(12,12,3)	③
(1,12,3)	④
(1,23,3)	⑤
(1,23,23)	⑦
(1,2,23)	⑧
(13,2,3)	⑨
(13,2,13)	⑪
(1,2,13)	⑫

The subsequent states and their transition probabilities from the other states are given by the same algorithm. The corresponding transition probability matrix  $P$ , the vector of the mean absorption times and the vector of the standard deviations of the absorption times are constructed from the same algorithm as in Section 3.3.

**4. Comparison of performance of 3 communication types**

In this Section we compare performance of 3 communication types, A, B and C based on the performance measured by the time taken for all members to get perfect information {1, 2, 3} or to build consensus.

**4.1 Comparison of the cases where the network has 2 members**

Here we compare two communication types: Types A and B.

Table 4.1

Type		$p_{12} = p_{21} = 0.5$	$p_{12} = p_{21} = 0.6$	$p_{12} = 0.4, p_{21} = 0.6$
A	Mean	6.00(2.25)	5.00(2.34)	6.33(2.22)
	S.D	3.74(2.29)	2.98(2.40)	4.22(2.22)
B	Mean	2.67(1.00)	2.14(1.00)	2.85(1.00)
	S.D	1.67(1.00)	1.24(1.00)	1.90(1.00)

In Type A of Table 4.1, we let  $r_1 = r_2 = 1/2$ . Here the mean absorption times are given in terms of a suitable time unit. Table 4.1 suggests that in mean and S.D Type A takes approximately 2.3 times as long as Type B regardless of  $p_{ij}$ . Therefore Type B is a more efficient communication type than Type A in mean and S.D of absorption time.

**4.2 Comparison of the cases where the network has 3 members**

Here we compare 3 communication types: Types A, B and C.

Table 4.2

Case Type		1	2	3
A (*1)	Mean	13.31(3.1)	9.86(3.2)	44.38(3.0)
	S.D	5.85(3.7)	4.03(4.0)	21.99(3.1)
B	Mean	4.34(1.0)	3.13(1.0)	14.74(1.0)
	S.D	1.60(1.0)	1.00(1.0)	7.00(1.0)
C (*2)	Mean	6.26(1.4)	4.50(1.4)	21.85(1.5)
	S.D	2.53(1.6)	1.55(1.6)	10.66(1.5)

case 1:  $p_{ij} = 1/3 (i, j = 1, 2, 3)$     case 1':  $p_{ii} = 1/3 (i = 1, 2, 3)$  others=2/3  
 case 2:  $p_{ii} = 0.10 (i = 1, 2, 3)$     case 2':  $p_{ii} = 0.10 (i = 1, 2, 3)$  others=0.90  
           others=0.45  
 case 3:  $p_{ii} = 0.80 (i = 1, 2, 3)$     case 3':  $p_{ii} = 0.80 (i = 1, 2, 3)$  others=0.20  
           others=0.10

(\*1) For Type A, we let  $r_1 = r_2 = r_3 = 1/3$ .

(\*2) For Type C, the mean and the S.D are calculated for the case 1', 2' and 3' and we let  $r'_1 = r'_2 = r'_3 = 1/3$ .

Table 4.2 suggests that in terms of mean time taken, Type A takes approximately 3 times as long as Type B and Type C takes approximately 1.4~1.5 times as long as Type B. In terms of S.D Type A approximately 3~4 times as great as Type B and Type C is approximately 1.5~1.6 times as great as Type B. Therefore in this case Type B is the most efficient of the three communication types in terms of means and S.D of times taken for all member to build consensus.

## 5. Conclusion

Groups who participate in electronic brainstorming through the network generate information, share information among members and use information to reach consensus or decision. The key issues of electronic brainstorming are how to provide effective network system and how to provide network users with maximum performance at minimum cost. In Type A, the performance measured by delay is the lowest one of the all, but the transaction pattern is 1-1 and possible transaction direction is unidirectional so there is little possibility of network congestion and flow and congestion control are not necessary and the protocols can become fairly simple and the number of information buffers in each node can become small. In Type B, a wait time for consensus is the shortest one of all, but the transaction pattern is  $m - 1$  and possible transaction direction is bidirectional so flow and congestion control are necessary to prevent deadlock in the network and protocols become complex and the pool of buffers become large. In Type C, the cost performance is between Type A and B.

This paper presents research on the design and evaluation of three types of computer supported group decision system for support of idea generation and building consensus. Numerous problems must be solved to achieve high levels of network performance under varying information loads. In the near future, if the problems of network congestion could be solved technically and economically by software and hardware development, Type B would become dominant system in EBS by reason of its efficiency of idea generation by interactions

of members and its rapid convergence of ideas. One of the assumptions of our model is that group members locate apart each other and the network has no central computer so it is impossible to use public screen and central file servers for handling the knowledge base and databases as usual GDSS. Being able to show the screens of the individual workstations on a public screen enables the group to follow the activities of issue identification, issue consolidation, rank ordering, voting, etc. Central file servers facilitate coordination and management of input from individual decision makers and serve as “organizational memory” from session to session. If we assume these facilities, the model of the network has to be greatly changed.

### References

- [1] Hill, S. R., and Turoff, M.: *The Network Nation: Human Communication via Computer*. Reading, Ma. Addison-Wesley, 1978.
- [2] Siegel, J., et al.: Group processes in computer-mediated communication. *Organizational Behavior and Human Decision Processes* (1986), 37.
- [3] Tapscott, D.: Investigating the electronic office. *Datamation*, (1982), 130-138.
- [4] DeSanctis, G., and Gallupe, B.: Group Decision Support Systems: A New Frontier, *Data Base*, Winter, (1985), 3-10.
- [5] Vogel, D., Nunamaker, J., Applegate, L. and Konsynski, B.: Group Decision Support Systems: Determinants of Success, *Proceedings of the 7th International Conference on Decision Support Systems*, June (1987), 8-11.
- [6] Nunamaker, J., Applegate, L. and Konsynski, G.: Facilitating Group Creativity: Experience with a Group Decision Support System. *J. Management Information Systems*, (1987).
- [7] Flament, C.: *Applications of Graph Theory to Group Structure*. Prentice-Hall, 1963, 48-50.
- [8] Soloman, M and Kramer, J.: *Distributed Systems and Computer Network*. Prentice-Hall, 1987, 126-128.
- [9] Shaw, M. E.: Communication networks. in L. Beakowitz (Ed.) *Advances in experimental social psychology*. Vol. 1, New York; Academic Press, 1964, 111-147.
- [10] Ahuja, V.: *Design and Analysis of Computer Communication Network*. McGraw-Hill, 1985, 127-128.
- [11] Isaacson, D. L., and Madsen, R. W.: *Markov Chains : Theory and Applications*. John Wiley & Sons, 1976, 95-97.

Shun-ichi Kigawa:  
Dept. of Economics  
Kantou Gakuen University  
Fujiagu, Gunma 200  
Japan.