Scalable Multilateral Communication Technique for Large-Scale Information Systems

Khaled Ragab, Naohiro Kaji, Koichi Moriyama and Kinji Mori

Tokyo Institute of Technology
2-12-1 Ookayama, Meguro, Tokyo 152-8552, Japan
Tel: +81-3-5734-2664, Fax: +81-3-5734-2510,
Email: {ragab@mori., nkaji@mori, moriyama@mori., mori@}cs.titech.ac.jp

Abstract

Autonomous Community Information System (ACIS) is a proposition made to contend with the extreme dynamism in the large-scale information system. ACIS is a decentralized bilateral-hierarchy architecture that forms a community of individual end-users (*community members*) having the same interests and demands in somewhere, at specified time. It allows the community members to mutually cooperate and share information without loading up any single node excessively. In this paper, an *autonomous decentralized community communication technique* is proposed to assure a flexible, scalable and a multilateral communication among the community members. The main ideas behind this communication technique are: content-code communication (community service-based) for flexibility and multilateral benefits communication for scalable and productive cooperation among members. All members communicate productively for the satisfaction of all the community members. The scalability of the system's response time regardless of the number of the community members has been shown by simulation. Thus, the autonomous decentralized community communication technique reveals great results of the response time with continuous increasing in the total number of members.

1. Introduction

The Internet's phenomenal impact, the subsequent growth and the evolving in social and economic environments promote more sever and complex requirements for the information service systems. Current Internet information services are provided for anyone, anywhere, anytime. These systems are constructed from the service providers (SP)' point of view. SPs provide information regardless of the end-users' demands and situations. There is no discernment between differences in place and time; end-users in any situation receive the same contents. In addition, end-users know in advance what content will satisfy their demands and then access the SP to obtain it. In a rapidly changing environment, the large-scale information systems are confronted to some challenges. First, the number of worldwide Internet and mobile users are predicted to exceed 1 billion by the end of 2005 [1]. Those users have rapidly, and dynamically changing demands and interests. Second, about 300 terabytes of information every year the world publishes on-line [2]. Constantly, new information services are added, others are modified, removed or in fault, making it more and more intractable to maintain a coherent image of the information environment. Therefore, customizing the service to the end-users is increasingly difficult, whereas end-users require well-customized, timely, continual, reliable, and available information services [3]. In addition, under the evolving situations they have heterogeneous and dynamically changing requirement levels of timeliness [4]. Timeliness is an essential component in modern high-assurance systems [5].

As the end-users demands are dynamically changing, anywhere/somewhere at specified time there are significant numbers of users sharing the same interests and demands. Consequently, a rapid and dramatic surge in the volume of requests arriving at a server often results in the server being overwhelmed and response times shooting up. Current information systems do not sustain such situation. For example, on the web the ubiquitous access of browsers and rapid spread of news about an event, lead to a flash crowd when a huge number of users simultaneously access a popular web site. Flash crowds are typically triggered by events of great interest, whether planned ones such as sport events (e.g. FIFA 1998 world cup event [6]) or unplanned ones such as an earthquake, etc. However the trigger need not necessarily be an event of widespread global interest. Depending on the capacity of a server, even a humble flash crowd can overwhelm the server. Obviously, current Internet information systems are not scalable and have failed to fulfill the stringent Internet users' requirements in such situations [7].

The complexity and dynamism of the large-scale information systems promote an imperative need for high-assurance in these systems. These systems can be seen as instances of the *Complex Adaptive Systems* alike *social community* [8]. Cooperation is the key of the evolution and continuity of the social communities [9]. Inspired from both the spirits of cooperation in the social communities and the Autonomous Decentralized System (ADS) concept [10] [11], the concept of an *Autonomous Community Information System* (ACIS) is proposed to meet the rapidly changing users' requirements. It customizes the service for the specific end-users (*community members*) have interests in that service, in somewhere/anywhere, at specified time. ACIS is completely decentralized in the sense that each member of the community performs the same set of tasks. ACIS allows community members to communicate directly with one another and share information without relying on any specified servers. Community members mutually cooperate to assure the high quality and well-customized information service provision and utilization for all members.

The contribution of this paper is the proposition of the scalable multilateral communication technique for large-scale information systems. The remainder of this paper is organized as follow. Section 2 clarifies the autonomous community information system concept and exhibits the system architecture. Section 3 exposes our proposed communication technique. Section 4 presents evaluation and simulation results showing improvement. The last section draws conclusions.

2. Autonomous Community Information System: Concept and Architecture

2.1 Concept

The main concern of the information systems has been in the past to efficiently retrieve relevant data for a particular request from immense repositories [12]. In such systems, two actors are coexisted: *Service Providers* and *End-users*. Service Providers offer the information content in the system. End-users consume the information services. The current information systems have been designed on the basis of the centralized model. Users individually send their requests to the service providers. Because of the end-users' demands are rapidly changing, in specified time the majority of them may have the same demands and then load up the service provider. For instance, the flash crowds caused by the September 11, 2001 terrorists attacked in the U.S. overwhelmed major news sites such as MSNBC and CNN, pushing site availability down close to 0% and response times to over 45 seconds [7]. Consequently, MSNBC quickly switched to serving static HTML and the percentage of error status codes dropped to 6.7%. Therefore, the information systems based on the centralized model are not scalable and failed to satisfy the Internet users' requirements of timeliness.

Currently, 90% of Internet resources are invisible and untapped [13]. *Peer-Peer information sharing systems* have turned to take into accounting the data and processing power that resides at the end-users. These systems are characterized by *unilateral benefits* because peers coordinate together for the satisfaction of only one of them, which requests the information. Thus, the average satisfaction rate for M peers in the systems is approximately 1/M and converges to zero as M increases. Peers share efforts for identifying the location of the required information. Then information downloads are done directly between two peers [14]. These systems have two lacks. First, the number of the identical requests is increased by the growth of the number of peers those send the same request. As a result a constant increase in traffic per peer is too high. In addition, these peer-peer systems do not specify how many connections a peer may initiate, accept, or simultaneously maintain. Consequently some peers may have high load than others. Unfairness among users pushes them to give up from such systems. As a result the availability of the system is gradually decreased and the system becomes distasteful for its users. Obviously, these systems have failed to satisfy the Internet users' requirements (e.g. timeliness) too.

Thus the next generation of the large-scale information systems must be characterized not only by a decentralized management but also a cooperative management, similar to the social community. The *community* as a social phenomenon deals with establishing and working with meaningful connections among people. In addition, it can be seen as an instance of the complex adaptive systems. In these very rapidly changing environments, the information system has to cope with unsteady network conditions and extreme dynamism in the end-user's demands and interests. The system has consequently to assure the online-expansion, maintenance and fault tolerance of the information systems. Inspired from both Autonomous Decentralized System (ADS) concept [10] [11], and the spirit of cooperation in the social communities, we propose the concept of *Autonomous Community Information System* (ACIS). We define *Autonomous Community* as a place where a coherent group of autonomous members having individual objectives, common interests and demands at specified time and somewhere/anywhere. They mutually cooperate to enhance the objectives for all of them timely and reliably. ACIS has the following properties:

- *Autonomy.* Actors of the system are members. They autonomously manage themselves and autonomously coordinate with the others.
- *Self-adapting*. The system must be adaptable to meet the constantly and dynamically changes of the members' interests and demands.
- *Synergy.* Mutual cooperation improves the objectives of all the individual members and increases the availability of the system they form. As more and more members participate in the community, the system is constantly scalable. Moreover, the well-customized information volume and quality is constantly increased.

In ACIS, each community member acts both as an information sender and a receiver, and furthermore each message from a participant is meaningful to all the other community members and at the same time every member is typically interested in data from all other senders in the community. ACIS brings new importance to the end-users by making them active actors in providing and utilizing the information. The cooperation among community members

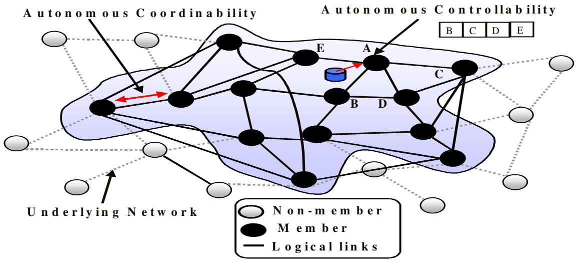

Fig. 1 Autonomous decentralized community system architecture.

assured the well-customized information service's provision and utilization among them. The communication among the community members is conducted on multilateral basis, as shown later in section 3. Community members cooperate not only for the satisfaction of one of them but also for all of them. Thus, the average satisfaction rate of M members is approximately one.

ACIS is a promising concept for information services operating at the edge of the network. It realizes the large-scale information system that successfully able to carry out, and enhance community members' objectives (e.g. timely information sharing) in a very dynamic environment. It guarantees the constructive cooperation and fairness among the community members with a very high degree of autonomy of them. We have developed a system architecture, called *Autonomous Decentralized Community System* (ADCS), that fosters the concept of the autonomous community information system.

2.2 Architecture

The autonomous decentralized community network is a self-organized logical topology. It is a set of nodes with considering the bilateral-hierarchy, the symmetric connectivity and the existence of loops. Community nodes are networked on a bilateral hierarchy basis. The bilateral logical contact between two community nodes will occur on the basis, the users of those nodes have same interests and demands, at specified time in somewhere. It is likely that in bilateral contacts, community members are get to know each other and share information. Each node keeps track of its immediate neighbors in a table contains their addresses. Each node knows its neighbor's nodes and shares this knowledge with other nodes for forming a loosely connected mass of nodes. For example, Figure 1 shows that each community node knows only four members. The bold lines represent the logical bilateral-link among the community nodes. Each node judges autonomously to join/leave the community network by creating/destroying its logical links with its neighbor's members based on its user's preferences.

2.2.1 Community Network Construction

Any node can join and leave the community at any time and via any node already in the community network. If no scheme is imposed on the way nodes join and leave the community network, then the network is likely to grow to become exponential network. This uncontrolled evolution may lead to some hotspots in the community network. For example, peer-peer systems do not specify how many connections a peer may initiate, accept, or simultaneously maintain. Consequently some peers may have high load than others. In that respect, we have proposed an autonomous decentralized community construction technique for making the potential hotspots very unlikely [15]. Community network construction polices the nodes joining and leaving the community network. Nodes are made to connect to randomly selected nodes already in the community network on joining with reserving the same degree for all the nodes and the short network diameter. Thus, the hotspots are very doubtfully and fairness is achieved in the community network by distributing the network traffic evenly among the community nodes during the communication.

2.2.2 Node Autonomy

Each node recognizes autonomously member from a non-member and cooperatively forwards the community information to only its neighbor's members. Community node does not forward the community information/request out of the community. Moreover, each node "*think globally and act locally*" by taking a decision autonomously based on its local information to store the relevant received information. The decision was taken not only according to the node situation (e.g. limited resources) and the importance of the offered information but also according to the other members' requirements. Each community node keeps a short memory of the recently routed messages in order to avoid the congestion in the community network. Each node autonomously coordinates (cooperates) with the others for locating, and/or providing the information in the community. If any member leaves, fails, joins the community, the other community members still can coordinate their individual objectives among themselves.

ADCS architecture has no central server whatsoever, as you can see in figure 1. It is a fully decentralized model, where each participated node has equal responsibilities, and does not rely on any central authority to organize the network. Thus, it does not load up any single node excessively and enables the development of the large-scale information systems.

3. Autonomous Decentralized Community Communication Technique

3.1 Conventional Communication

The conventional communication, typically through Web browsers, has been built on the one-to-one communication protocol. In one-to-one, data travels between two users, e.g., e-mail, e-talk. This protocol gobbles up the network bandwidth and makes the real time services unresponsive. Caching most popular web pages on the proxy server reduces the network bandwidth consumption and the access latency for the users. While, the web caches techniques have some disadvantages. First, a single proxy server is a single point of failure. Second, the limited number of users per the proxy manifests bottleneck affects. Third, Data does not updated automatically. Finally, cache miss cause increase in the latency (i.e. extra proxy processing). While, in the conventional one-to-many group's communication the message travels primarily from a server to multiple users, e.g., web download and software distribution. For very large groups (thousands of members) or very dynamic multicast groups (frequent joins and leaves), having a single group controller might not scale well. Currently, there is no design for the application-level multicast protocol that scales to thousands of members. For example, *Overcast* [16] builds the mesh per a group containing all the group members, and then constructs a spanning tree for each source to multicast information. The mesh creation algorithm assumes that all group members know one another and therefore, does not scale for large groups. *Bayeux* [17] builds a multicast tree per a group. Each request to join a group is routed to a node acting as the root. This root keeps a list of the all group members. All group management traffic must go through that root. It generates more traffic for handling a very dynamic group membership.

3.2 Community Content-code and Multilateral Communication Technique

Conventional communication techniques use the destination address (e.g. unicast address, multicast address) to send the data. In very changing environment likes ADCS (i.e. end-users are frequently joined and left), these conventional communication techniques are not applicable. Thus, the autonomous decentralized community communication technique has broached [18] [20], to assure a productive cooperation, a flexible and a timely communication among members. The main ideas behind our proposed communication technique are: content-code communication (community service-based) for flexibility and multilateral communication for timely and productive cooperation among members. The first main idea behind the autonomous decentralized community communication technique is the separation of the logical community service identifier from the physical node address. In this communication technique, the sender does not specify the destination address but only sends the content/request with its interest *content Code* (CC) to its neighbor's nodes. CC is assigned on a type of the community service basis and enables a service to act as a logical node appropriate for the community service. Figure 2 shows the community communication message format. CC is uniquely defined with respect to the common interest of the community members (e.g. politic, news, etc.). The information content is further specified by its *Characterized Code* (CH). The CH is the hash of the message content. It is uniquely specified with respect to the message content (e.g. data or request).

CC	CH_i	Data/Request

Fig. 2 Community communication message format.

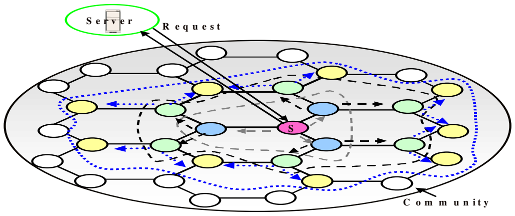

Fig. 3 Publish based protocol.

The second main idea behind the autonomous decentralized community communication technique is *multilateral benefits* communication for timely and productive cooperation. The multilateral communication likely occurs between the community members that are already networked on a bilateral basis. All members communicate productively for the satisfaction of all the community members, as follow.

The autonomous decentralized community communication technique performs the communication among the community members, and we called it "*1→N*". A brief scenario for the 1→N community communication is as follows. The community node asynchronously sends a message to N neighbor's nodes. Then, those N nodes forward the same message to another N nodes in the next layer, except the node that delivered the incoming message and so on gradually until all the community nodes received it. This technique handles, as the model knew like *viral propagation*. The autonomy of the 1→N communication can be seen as follow. Each community node recognizes autonomously member from non-member and judges autonomously to forward community messages to only N community neighbor's nodes. The 1→N communication technique does not rely one any central controller. Each community node has its own local information and communicates only with specified number (N) of the neighbor's nodes. There is no global information (e.g. multicast group address).

In order to avoid the congestion that may be happening if some of the community nodes synchronously send identical messages, each node keeps a short memory of the recently routed messages and judges autonomously to forward only one copy of the received messages to the other neighbor's nodes. Moreover, each node autonomously takes a decision to keep or delete the short memory of the received message based on the frequency of receiving that message.

3.3 Community Communication Protocols

The autonomous decentralized community communication technique has the two communication protocols: *publish based* and *request /reply-all based*.

- *Publish based protocol*. When one of the community members has new information, she/he publishes it to all the community members using "1→N". The publish-based protocol offers an effective solution to the flash crowd problem as shown in figure 3. The solution scenario is as follows. As soon as one of the community members S downloaded an interested content for the community from

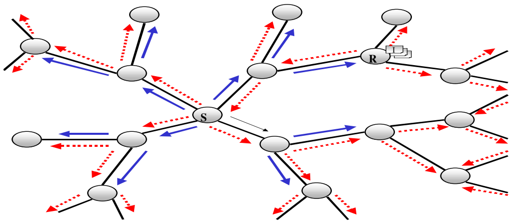

Fig. 4 Messages flow in request/replay-all based protocol.

the server, she/he publishes it to all the community members, thereby relieving the server of this task and alleviating a load on the server. Thus, the load is distributed among the community nodes and increased slightly even as the number of nodes increases dramatically. In addition, it represents a scalable solution for large-scale information dissemination systems.

- *Request/reply-all based protocol.* When a community member wants to locate information, she/he emits a request message. Then the others community members cooperate to locate the requested information. When any community node receives the requested message, it processes the request. If no results are found at that node, the node will forward the request to its neighbor's nodes with using "1→N". Otherwise, if any results are found at that node then the node will produce results, such as pointers to the information or the whole content based on the size of the information. Then that node will send a reply message not only to the node, which requested the information but also to all the community members. Figure 4 shows the message flow when the community node S sends a request (solid arrows) to its neighbor's and node R replies (dotted arrows) to all the community members by the required information I. The *reply to all* protocol affluent the other community members to emit the same request. Consequently, all the community members enrich their experiences and/or get to know new services without requesting, in which individually they cannot get to know. Thus the multilateral benefits characteristic of the community can be satisfied. In addition, it decreases the traffic per node by avoiding multiple requests for the same content.

The originality of our proposed communication technique does not come only from the *content-based* communication but also from the *reply-all* that satisfies the *multilateral benefits.* In 1→N community communication all members cooperate for the satisfaction of all the community members contrary to the peer-peer (P2P) communication techniques. In P2P, peers cooperate for the satisfaction of only one, which request the information (unilateral benefits). The comparisons between the community information system and the conventional information systems: client/server and peer-peer are tabulated in table 1. From this table we conclude that the community communication is: service-based, cooperative, relationship and multilateral communication. As more and more users join the community, the average satisfaction rate is increases and the availability gradually increases.

Table 1: Comparison

			Conventional (Client/server)	Peer-Peer	Community
Technological	Membership Management		Centralized	Centralized (e.g.Overcast, Bayeux)	Decentralized Loosely control
	Communication	Model	Address-based	Address-based	Service-based
		Request	One-one	One-many	Cooperative (1→N)
		Reply	One-one	One-one	Cooperative '1→N)
		Feature	Unilateral	Unilateral	Multilateral
Char.^{s}	Users		Passive	Active	Active
	Load		Server congestion	Peers congestion	No congestion (Fairness)

Moreover, the system is scalable of the response time with a huge number of members. Thus, it guarantees a timely communication among the community members.

4. Evaluation

To evaluate the performance of our proposed technique, we consider the community network topology as random-regular graph that has been constructed by our proposed construction technique [15]. Thus, the number of the community nodes is

$$M \leq 1 + \sum_{i=0}^{L-1} k(k-1)^{i} \qquad (1)$$

Where k is the connectivity of each node and L is the diameter of the network (number of layers). Then the number of layers can be calculated as follows.

$$L \approx \left\lceil \frac{\log \left[\frac{(M-1)*(K-2)}{K} + 1 \right]}{\log (K-1)} \right\rceil \qquad (2)$$

Thus, the transmission time T tọ send a message from one member to all the other members is bounded by O(Nlog$_{N}$(M)), where N=k-1. Consequently we can drive the optimal 1->N communication as follows.

$$\frac{dT}{dN} = \frac{d}{dN}\left(N * \log_{N}(M)\right) \qquad (3)$$

$$= \log(M)\left(\frac{\log(N) - 1/\ln(10)}{\left(\log(N)\right)^{2}}\right)$$

From equation 3, we conclude that $dT/dN = 0, d^{2}T/dN > 0 \Rightarrow N \approx 3$ and T is concave up. For any number of nodes M, the (1→3) community communication technique is the optimal. Similarly, figure 5 shows that ever-increasing the number of nodes the optimal communication is 1→3.

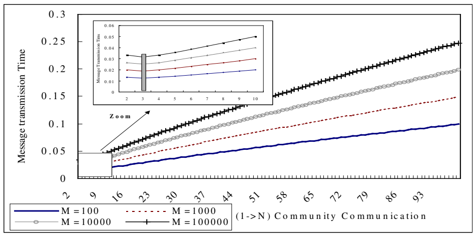

Fig. 5 Optimal 1→N community communication.

4.1 Simulation and Results

We simulated our communication technique on a network spending 4-array connectivity for each community node. Our experiment is conducted over 100,000 community

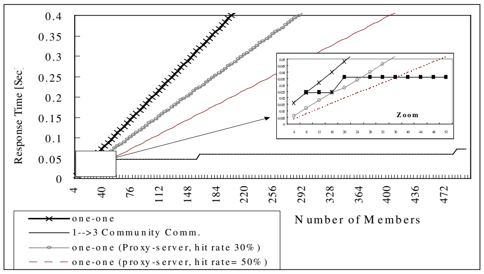

Fig. 6 Simulation result: scalable communication.

members, using 1→3 communication technique and is constituted of average communication cost between each node τ_{cc} = 0.001 seconds. Take τ_m = 0.001 seconds is the average time each nodes are needed for monitoring the recent received messages to avoid the congestion. Thus the transmission time τ to send a message from any node to all the other community members is bounded by $L*N*(\tau_{cc}+\tau_m)$. For one-one communications, we assuming that all community members are from different organizations in the world so the web caching techniques having slightly effect in the response time. It has been proven that a chasing proxy has an upper bound of 30-50% in its hit rate [19]. We tried our experiment for one-one communications with caching proxy. We concentrate in this experiment on the comparison between the conventional one-to-one communication techniques without and with caching proxy (hit rate of 30%, 50%) and (1→N) community communication technique. Figure 6 depicts the effectiveness of our communication technique in compared with the conventional ones. The 1→N communication technique is able to send a message to all the community members within an average of less than about 6 times in compare with the one-one communications. Furthermore it approves that the community communication technique is scalable of the response time with the number of the members. For a very small number of members, our proposed community communication technique is not effective but it reveals great results with continuous increasing in the total number of members (see figure 6 zoom part).

5. Conclusion

Efficiently making the rapidly changing users demands meet the offer is of main importance in the large-scale and very dynamic information systems. While the current information technologies do not sustain the rapid and dramatic surge in the volume of requests arriving at a server. To ride out these challenges, this paper has presented the autonomous community information system ACIS concept. ACIS is a large-scale information system that meets the rapidly changing users' requirements for services with coping the extreme dynamism of the operating environment. To sustain the proposed concept, the bilateral-hierarchy system architecture that has called *Autonomous Decentralized Community System* (ADCS) has been developed. Finally, this paper has presented the autonomous decentralized community communication technique for achieving a productive cooperation, flexible and a timely communication among the community members. This communication technique is not only content-code communication (service-based) but also multilateral communication. The simulation results have depicted and approved that the community communication technique is scalable of the response time with the number of the members.

6. Reference

[1] L. G. Roberts, "Beyond Moore's law: Internet growth trend ", IEEE Computer, Vol. 33, no. 1, p. 117, 2000.

[2] Peter Lyman and Hal R. Varian, "How Much Information", Journal of Electronic Publishing, Vol. 6, Issue 2, Dec. 2000.

[3] David B. Ingham, S. K. Shrivastava, F. Panzieri, "Constructing Dependable Web Services," IEEE Internet Computing, Vol. 4, No. 1,pp. 25-33, 2000.

[4] K. Mori, "Applications in Rapidly Changing Environments," IEEE Computers, Vol. 31, No. 4, PP. 42-44, 1998.

[5] K. Mori, "Towards integrated methods for high assurance systems," IEEE Computer, vol. 31, No. 4, PP. 32-34, 1998

[6] Martin Arlitt, Tai Jin, "Workload Characterization of the 1998 World Cup Web Site," Hewlett Packard Co. 1999.

[7] J. Jung, B. Kirshanmurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites", The 11th Int. World Wide Web Conference, WWW2002, Hawaii, USA, May, 2002.

[8] C. G. Langton, "Complex Adaptive Systems," MIT Press, 1995.

[9] John Stewart, "Evolution's Arrow: The direction of evolution and the future of the humanity", Chapman press, Australia, 2000.

[10] K. Mori, "Autonomous Decentralized Systems: Concept, Data Field Architecture and Future Trends," Proc. of the first Int. Sym. On ADS, (ISADS'93), IEEE, Kawasaki, Japan, pp. 28-34, 1993.

[11] K. Mori, H. Ihara, et al., "Autonomous Decentralized Software Structure and its Application," Proc. IEEE FJCC'86, pp.1056-1063. November 1986.

[12] V.N. Gudivada, V.V. Raghavan, W.I. Grosky, and R. Kasanagottu, "Information retrieval in the World Wide Web", IEEE Internet Computing Magazine, Vol. 1, no. 5, pp. 58-68, Sept. Oct. 1997.

[13] R. Dornfest, "Dark Matter, Sheep and the Cluster: Resolving Metaphor Collision in P2P," The O'Reilly Peer-to-Peer and Web Services Conference Washington, D.C., November 5-8, 2001

[14] M. Ripeanu, "Peer-Peer Architecture Case Study: Gnutella Network," In Proc. of Int. Conf. on P2P computing, 2001.

[15] K. Ragab, T. Ono, N. Kaji, K. Mori, " Autonomous Decentralized Community Concept and Architecture for a Complex Adaptive Information System," Proc. IEEE FTDCS, Puerto Rico, May 2003.

[16] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'toole. "Overcast: Reliable Multicasting with as n Overlay Network," In Proc. Of the Fourth Symposium on Operating System Design and Implementation (OSDI), pages 197-212, Oct. 2000.

[17] S. Q. Zhuang, B. Y. Zhao, A. D. Hoseph, R. H. Katz and J. D. Kubiatowicz, "Bayeux: An Architecture for Scalable and Fault tolerant Wide-area Data Dissemination," In Proc. Of the 11th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), June 2001.

[18] K. Ragab, T. Ono, N. Kaji, K. Mori, "Community Communication Technique for Achieving Timeliness in Autonomous Decentralized Community Systems," Proc. IEEE IWADS, Beijing, China, pp 56-60, Nov., 2002.

[19] Stephen Williams, and Edward A. Fox. "*Caching proxies: Limitations and potentials,*" In Proceedings of the Fourth International World Wide Web Conference, December 1995.

[20] K. Ragab, T. Ono, N. Kaji, K. Mori, "An Efficient Communication Technology for Autonomous Decentralized Community Information System," Supplement Proc. IEEE ISADS, Pisa, Italy, April 2003.