Group Communication Protocol with Heterogeneous Clock

Satoshi Kawanami, Tomoya Enokido, and Makoto Takizawa
Tokyo Denki University, Japan
E-mail {kawa, eno, taki}@takilab.k.dendai.ac.jp

Abstract

Peer processes distributed in various types of networks are autonomically cooperating to achieve some objectives. We discuss a group communication protocol named HCG (heterogeneous clock group) protocol. Message are causally delivered by using synchronization mechanism like clocks in a group. Here, a group is composed of subgroups which are interconnected with the Internet and processes in each subgroup are interconnected with local and personal area network. Processes in subgroups use physical and linear clocks while processes in the Internet use vector clock. Messages are ordered by its own mechanism in each group. We discuss how to causally deliver messages by using local synchronization mechanisms of subgroups. We evaluate the HCG protocol in terms of number of messages ordered.

異種時計を用いたグループ通信プロトコル

河浪 悟士　榎戸 智也　滝沢 誠
東京電機大学理工学部情報システム工学科
E-mail {kawa, eno, taki}@takilab.k.dendai.ac.jp

現在の情報システムは、複数の計算機がPAN, LAN, WANなどの複数のネットワークにより相互接続された分散システムとなっている。各計算機上に分散された複数のプロセスはグループを構成し、グループ内のプロセスが互いにメッセージを交換し協調動作を行う。このようなグループ内のプロセス間の通信プロトコルでは、メッセージを因果順序に配送する必要がある。メッセージの因果順序配送を保証する手法として、ベクタ時刻、線形時刻という論理時間、または、各計算機の物理時間を用いた同期手法が提案されている。多数のプロセスから成る大規模なグループでは、ベクタ時刻の処理と通信の負荷が大きく、大規模グループ内のコンピュータ間では物理時計を同期させることが困難である。本論文では、大規模グループを構成するネットワーク範囲ごとに最適な論理・物理時間を適用し、メッセージの因果順序配送を行う方法を提案する。

1. Introduction

Distributed systems are composed of multiple processes interconnected with networks. Peer processes are cooperating to achieve some objectives by exchanging messages with each other in peer-to-peer (P2P) systems [11]. A *group* is a collection of cooperating peer processes. Messages have to be causally delivered to processes in a group [6,10]. There are discussions on group protocols [2, 14], where messages are causally delivered by using the vector clock [8]. In P2P applications, a large number of processes have to be cooperating. The paper [4] discusses a hierarchically structured group where processes are interconnected in a loop network and loops are also interconnected in a loop. Messages are transmitted in a token passing mechanism. In the paper [1], the *hierarchical daisy architecture* is discussed. A group is composed of *logical groups* which provide causal delivery of messages by a *causal server* in presence of processes faults. All causal servers are also members of causal servers group.

Processes are communicating with each other in various types of networks like personal area network (PAN) [13], local area network (LAN) [12], and wide-area network [12]. In a personal area network, processes in *last ten's meters* are interconnected with wireless communication channels [3] [7]. In order to synchronize processes in a group, types of clocks like logical clocks [6, 8] and physical clocks are used. Some mechanisms to synchronize physical clocks in multiple processes are discussed like radio and GPS clocks [15] with NTP (Network Time Protocol) [9]. Clock synchronization in a one-hop *ad hoc* network is also discussed [5]. Messages are stamped with physical time when transmitted. Messages are totally ordered in their time-stamps. Since the message length is $O(1)$, it is easy to design and implement algorithms for synchronizing processes. However, these algorithms are not applicable for a wide-area network due to long delay time among processes and every process cannot use a precise physical clock. In the linear clock, message length is $O(1)$ as well as physical clock. The vector clock [8] can be used to causally order only and all messages to be causally ordered but message length is $O(n)$ for total number n of processes in a group. The computation and communication overheads are too large to realize a large-scale group. In this paper, we discuss a structured group which is composed of subgroups, each of which takes usage of its own type of clock to synchronize local processes in order to realize scalable group communication. Thus, a *heterogeneous* group is composed of subgroups with different types of clocks. In this paper, we consider a system where a collection of processes are interconnected in a local network, i.e. local and personal area network, and the local networks are interconnected in a wide-area network. In a local network, processes are synchronized by using physical and linear clocks. In a wide-area network, processes are synchronized by using vector clocks. Even messages not to be causally ordered are ordered in physical and linear clocks. In addition, even if messages are locally causally ordered in a local subgroup, the messages may be globally causally concurrent in a group. In this paper, we reduce the number of messages to be unnecessarily ordered in a structured, heterogeneous group.

In section 2, we present a system model. In section 3, we discuss how to causally order messages in a hierarchical group. In section 4, we evaluate the HCG protocol in terms of the number of messages ordered compared with vector clock and linear clock protocols.

2 System Model

A *group* means a collection of peer processes $p_1, ..., p_n$ which are cooperating by exchanging messages in networks. Processes in a group have to be synchronized to do cooperation. For example, messages are causally delivered to destination processes in a group. A *large-scale* group means a group which is composed of a large number of processes, i.e. hundreds to thousands processes. A *heterogeneous* group is one where processes use different types of clocks like physical and linear clocks.

Each computer is equipped with a physical clock. However, every pair of physical clocks (RT) in different computers do not always show same time. Each computer has to synchronize its physical clock with the other computers in order to do the cooperation. Radio and GPS clocks [15] are now easily available for computers but can be only supported in time servers due to large cost. The accuracy of synchronization by radio and GPS clocks is about 100 [msec] and 1 [μsec], respectively. NTP (Network Time Protocol) [9] with time servers is used to synchronize physical clocks. Processes communicate with a time server to obtain the current time. It takes time to exchange messages with the time server. If processes in a subgroup communicate with a time server in the Internet, the accuracy is about several hundreds [msec]. If delay time is longer and variant, the process cannot obtain correct current time. In a wide-area network, the delay time is about one hundred times longer than a local network. Here, it is difficult to synchronize physical clocks of processes at higher accuracy.

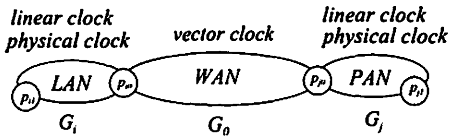

Figure 1. Structured group.

In *linear* clock (LT) [6], each process p_i manipulates a variable T. Each time p_i sends a message m, $T := T + 1$ and then m carries $m.T$ ($= T$). On receipt of a message m, $T := max(m.T, T)$. Here, a message m_1 causally precedes another message m_2 only if $m_1.T < m_2.T$.

In *vector clock* (VT) [8], each process p_i manipulates a vector $\langle T_1, ..., T_n \rangle$ for number n of processes, where each element T_j is initial 0 ($j = 1, ..., n$). Each time a process p_i sends a message m, the ith element T_i is incremented by one, i.e. $T_i := T_i + 1$ ($i = 1, ..., n$). Then, the message m carries the vector T of the sender process p_i as $m.T(= \langle m.T_1, ..., m.T_n \rangle)$. On receipt of a message m from a process p_k, a process p_i manipulates the vector T as $T_k := max(T_k, m.T_k)$ ($k = 1, ..., n$, $k \neq j$). Here, a message m_1 *causally precedes* another message m_2 ($m_1 \rightarrow m_2$) if and only if (iff) $m_1.T < m_2.T$. m_1 is *causally concurrent* with m_2 ($m_1 \parallel m_2$) neither $m_1.T \leq m_2.T$ nor $m_1.T \geq m_2.T$. Since a message length is $O(n)$ in the vector clock, it is not easy to use the vector clock for a large group.

A *hierarchical* group G is composed of subgroups G_0, $G_1, ..., G_k$. Each subgroup G_i adopts some type of clock *clock*(G_i) $\in$ $\{$*RT*(*real time*), *LT*(*linear clock*), *VT*(*vector clock*)$\}$ ($i = 1, ..., k$). A pair of different subgroups G_i and G_j may use different types of clocks. For example, real time supported by physical clock is used to causally order messages in a local network like PAN and LAN since the delay time is shorter, i.e. one millisecond [msec]. A logical clock like vector clock is used in a wide-area network where delay time is about one hundred [msec], one hundred times longer than local network. Subgroups with physical, linear, and vector clocks are referred to as RT, LT, and VT subgroups, respectively.

Suppose there are a pair of subgroups G_i and G_j in each of which processes are interconnected in a local network. Gateway processes p_{i0} and p_{j0} of the subgroups G_i and G_j are interconnected with a wide-area network and a group G_0 composed of gateways is referred to as *global* [Figure 1]. Here, in the local subgroups G_i and G_j, RT and LT are used. Each message m is assigned with time-stamps $m.RT$ and $m.LT$ showing "local real time" and "local linear time" when the message m is sent in a subgroup. Processes in the subgroups deliver messages in time-stamp order. On the other hand, the vector clock is

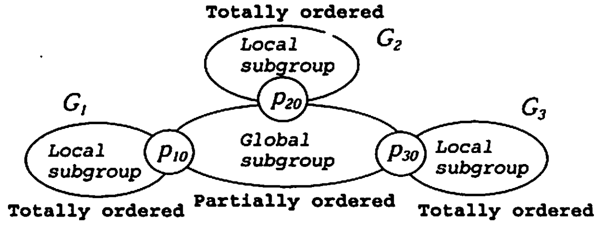

Figure 2. Ordering of messages.

used to exchange messages among gateway processes in the global subgroup. Here, each element VT_i in the vector clock $\langle VT_1, ..., VT_k \rangle$ shows logical time of a local subgroup G_i ($i = 1, ..., k$). When a gateway of G_i sends a message m in a global subgroup G_0, VT_i is incremented by one and then the message m carries the vector clock VT.

3 Scalable Causality in Heterogeneous Group

3.1 Structured group

Suppose a group G is composed of a *global VT* subgroup G_0 and multiple *local LT/RT* subgroups G_1, ..., G_k ($k \geq 2$). Each local subgroup G_i is composed of a *gateway* process p_{i0} and normal processes $p_{i1}, ..., p_{il_i}$ ($l_i \geq 1$) ($i = 1, ..., k$). A gateway process p_{i0} communicates with gateway processes of other subgroups. A global subgroup G_0 is composed of gateway processes. We make following assumptions:

1. Processes in each local subgroup G_i are interconnected with a local network like LAN and PAN ($i = 1, ..., k$).
2. Gateway processes in a global subgroup G_0 are interconnected with a wide-area network (WAN).

In each local subgroup G_i, every process uses a pair of clocks, physical (*RT*) and linear (*LT*) clocks to synchronize message communication since delay time between every pair of processes is so short that variance among physical clocks in different processes can be neglected. Messages transmitted in a global subgroup are referred to as *global* while messages in local subgroups are *local*. *Global* messages are causally ordered in the vector clock while local messages are totally ordered in each local subgroup [Figure 2].

Suppose a pair of messages m_1 and m_2 are transmitted in a local subgroup G_i. Here, m_1 *locally causally precedes* m_2 in G_i ($m_1 \rightarrow_i m_2$) iff a sending event of m_1 *happens before* [6] m_2 in G_i. Next, suppose a pair of messages m_1 and m_2 are transmitted in different local subgroups G_i and G_j, respectively. m_1 *globally causally precedes* m_2 ($m_1 \rightarrow m_2$) iff a sending event of m_1 happens before a sending event of m_2 in a group G.

3.2 Scalable causality

Suppose a process p_{is} in a local subgroup G_i sends a local message m with a pair of LT and RT time-stamps $m.LT$ and $m.RT$. It is noted that a pair of local messages m_1 and m_2 in a local subgroup G_i may be causally concurrent even if $m_1.LT \neq m_2.LT$ or $m_1.RT \neq m_2.RT$ as presented in the preceding subsection. On receipt of a message m, the gateway process p_{i0} in G_i forwards m to other gateway processes in the global subgroup G_0. Here, the message m is assigned with the vector clock which is shown by $m.VT$. Next, a gateway process p_{j0} in a local subgroup G_j receives the message m. The gateway process p_{j0} forwards the message m to local processes in G_j. Here, the message m is time-stamped with $m.LT$ showing local time when p_{j0} sends m in the local subgroup G_j. *Local time* means linear time (*LT*) and real time (*RT*). Finally, a process p_{jt} in a local subgroup G_j receives a message m_1 which is sent by a process p_{is} in the local subgroup G_i. Delay time between local subgroups G_i and G_j is so long that physical clocks in different local subgroups cannot be synchronized at high accuracy. Since RT time-stamp of message sent from another local subgroup is not reliable, messages are ordered by LT time-stamps in a local subgroup G_j.

Suppose a gateway process p_{j0} sends a message m_1 and a process p_{ju} sends a message m_2 in a subgroup G_j. Let $m_1.T$ show local time when the gateway process p_{j0} sends a local message m_1 in the local subgroup G_j, i.e. RT or LT. Here, m_2 *locally causally precedes* m_1 ($m_2 \rightarrow_j m_1$) in G_j since $m_2.T < m_1.T$. However, the source process p_{is} does not send the message m_1 after receiving the message m_2. Hence, m_1 and m_2 are causally concurrent in a group G ($m_1 \| m_2$) even if $m_2 \rightarrow m_1$ in the subgroup G_j. A gateway process p_{j0} receives a message m from another gateway process p_{i0}.

Let $c_{ik}(t)$ be a physical clock of a local process like p_{ik} at UTC time t. Assume every pair of clocks c_{is} and c_{ju} show the *same* time, i.e $|c_{is}(t) - c_{ju}(t)| \leq \tau$ for each UTC time t where τ is the maximum allowable difference. If each process knows maximum allowable difference τ, local processes can deliver messages by using their time-stamps.

It is straightforward for the following theorem to hold from the definitions:

[Theorem] Let m_1 and m_2 be messages in a local subgroup G_i. m_1 globally causally precedes m_2 ($m_1 \rightarrow m_2$) only if $m_1.T < m_2.T$. □

As pointed out here, a pair of local messages m_1 and m_2 are totally ordered by using the physical clock and linear clock even if m_1 and m_2 are causally concurrent in a local subgroup G_i. If a pair of messages

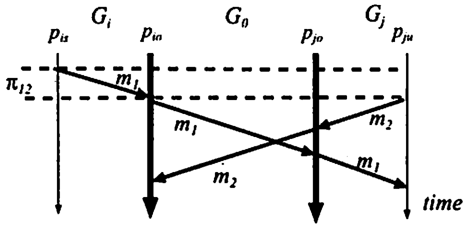

Figure 3. Causal precedency.

m_1 and m_2 are sent out to other local subgroups in a global subgroup G_0 of gateway processes, a pair of global messages m_1 and m_2 are ordered as well, i.e. $m_1.VT < m_2.VT$ if $m_1.LT < m_2.LT$. A vector clock VT is in a form $\langle VT_1, ..., VT_k \rangle$ where each element VT_i shows logical time of a gateway process p_{i0} of a local subgroup G_i ($i = 1, ..., k$). Each time a gateway process p_{i0} sends a global message, the ith element VT_i in the vector VT is incremented by one independently of which local process sends the message in the local subgroup G_i. Hence, $m_1.VT < m_2.VT$ if and only if (iff) $m_1.LT < m_2.LT$.

Each gateway process p_{i0} in a local subgroup G_i has a local message log LML_i where global and local messages which p_{i0} has received and sent are stored. The vector clock $VT = \langle VT_1, ..., VT_k \rangle$ is manipulated in a gateway process p_{i0} of a local subgroup G_i ($i = 1, ..., k$) as follows:

- On receipt of a local message m, $m.VT_i := m'.VT_i + 1$ where m' is a message whose time-stamp VT_i is the largest in the local message log LML_i.

[Theorem] Let m_1 and m_2 be messages sent in a local subgroup G_i. m_1 causally precedes m_2 ($m_1 \rightarrow m_2$) only if $m_1.VT_i < m_2.VT_i$. □

3.3 Causality with physical clock

In Figures 4 and 5, a gateway process p_{j0} receives a message m_1. After sending a message m_1 to a process in a local subgroup G_j, the gateway process p_{j0} receives a message m_2. Here, $m_2.VT_j := m_1.VT_j + 1$. Suppose a gateway process p_{j0} receives a global message m_1 and forwards m_1 to processes in a local subgroup G_j. As presented before, each local message m in a subgroup G_i carries RT time-stamp (real time) $m.RT$ and LT time-stamp (linear time) $m.LT$. The gateway process p_{j0} assigns the message m with its physical time $m.RT$ and linear time $m.LT$. Then, the gateway p_{j0} forwards the message m_1 to local processes in the subgroup G_j. Then, suppose the gateway process p_{j0} receives a local message m_2 from a local process p_{js}. Let δ_j be the minimum delay time in a subgroup G_j. Let Ts be time when a message m_1 is sent and Tr be time when a message m_2 is received. If $Tr - Ts > 2\delta_j$, the process p_{js} might send m_2 after receiving m_1, i.e. m_1 causally precedes m_2 as shown in Figure 4. Hence, $m_2.VT_j > m_1.VT_j$. On the other hands, if $Tr - Ts \leq 2\delta_j$, it is sure the process p_{js} sends m_2 before receiving m_1. That is, m_1 and m_2 are causally concurrent as shown in Figure 5. Here, $m_2.VT_j = m_1.VT_j$.

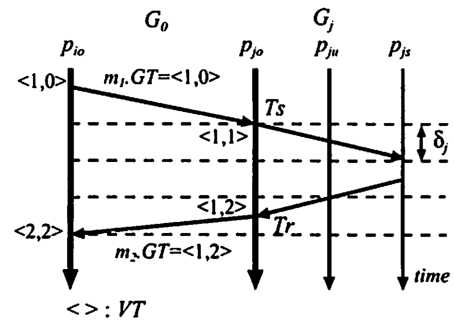

Figure 4. Causality with physical clock.

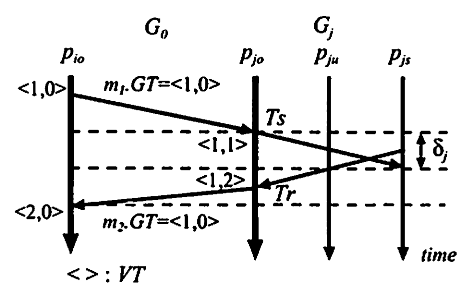

Figure 5. Causality with physical clock.

3.4 Local causality

Suppose that a pair of processes p_{j1} and p_{j2} send a message m_1 at local time $c_{j1}(t_1)$ and a message m_2 at $c_{j2}(t_2)$ to a process p_{j3}, respectively. Here, $m_1.RT = c_{j1}(t_1)$ and $m_2.RT = c_{j2}(t_2)$. We assume $|c_{jk}(t) - c_{jl}(t)| \leq \tau$ for every pair of processes p_{jk} and p_{jl} and UTC time t, i.e. every physical clock is synchronized with the other physical clocks in a local subgroup. The common destination process p_{j3} of messages m_1 and m_2 perceives that anyone of m_1 and m_2 is never sent after the other one is received, i.e. neither $m_1 \rightarrow m_2$ nor $m_2 \rightarrow m_1$ if $|m_1.RT - m_2.RT| \leq 2\tau$. Otherwise, the process p_{j3} perceives that one of m_1 might causally precede the other message.

[Theorem] Let m_1 and m_2 be messages sent in a local subgroup G_j. m_1 causally precedes m_2 ($m_1 \rightarrow m_2$) if

1. $m_2.RT > m_1.RT$ and $m_2.LT > m_1.LT$ if $|m_2.RT - m_1.RT| > 2\tau$.
2. $m_2.LT > m_1.LT$ if $|m_2.RT - m_1.RT| \leq 2\tau$.

Otherwise, m_1 and m_2 are causally concurrent. □

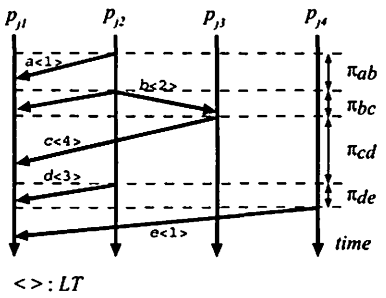

Figure 6. Order in local subgroup.

Let us consider Figure 6 where messages a, b, c, d, and e are exchanged among four processes p_{j1}, ..., p_{j4} in a local subgroup G_j. Here, $a.RT$ shows time when a process p_{j2} sends a message a. Let π_{xy} be $|x.RT - y.RT|$ for every pair of messages x and y. If $2\tau < \pi_{ab}$, a process can the deliver the messages a and b using RT time-stamp and LT time-stamp. If $2\tau \geq \pi_{ab}$, a process can order the messages a and b by using only LT time-stamp. In Figure 6, if $\pi_{bc} \leq 2\tau$ and $\pi_{de} \leq 2\tau$, a pair of the messages b and c are ordered by $b.LT$ and $c.LT$. Since $b.LT < c.LT$, the message b causally precedes the message c ($b \rightarrow c$). Here, $c.LT > d.LT$. Hence, messages c and d are causally concurrent, although $\pi_{cd} > 2\tau$ and $c.RT < d.RT$.

4 Evaluation

We evaluate the HCG (heterogeneous clock group) protocol. We measure how many messages are causally ordered with each other in a local subgroup by three protocols, HCG, linear clock (LT), and vector clock (VT) protocols. If a message m_1 causally precedes another message m_2 ($m_1 \rightarrow m_2$) or $m_2 \rightarrow m_1$, the messages m_1 and m_2 are referred to as *causally ordered*. Suppose a gateway process p_{j0} receives a message m_2 after receiving a message m_1 as shown in Figure 7. Here, we assume that each process sends a message every λ time units to destination processes which are randomly selected. We assume every physical clock in a local subgroup is perfectly synchronized ($\tau = 0$).

In the evaluation, a local subgroup G_j includes one gateway process p_{j0} and three local processes p_{js}, p_{jt}, and p_{ju} [Figure 7]. The HCG, LT, and VT protocols are implemented in these four processes. We measure the number of messages ordered in the local subgroup G_j by the HCG, VT, and LT protocols. Figure 8 shows the ratio[%] of the number of messages ordered to the total number of messages in the HCG, LT, and VT protocols for inter-transmission time λ. Figure 8 shows 15% to 20% of messages ordered are reduced in the HCG protocol than the linear time (LT) protocol. On the other hand, about two times more number of messages are ordered in the HCG protocol than the VT protocol. Messages ordered in the VT protocol are ones to be causally ordered. In the HCG protocol, about 35% of messages are unnecessarily ordered. However, message length is $O(n)$ for number n of processes in the VT protocol while $O(1)$ for the HCG protocol. The VT protocol cannot be adopted for large-scale groups due to the complete $O(n)$.

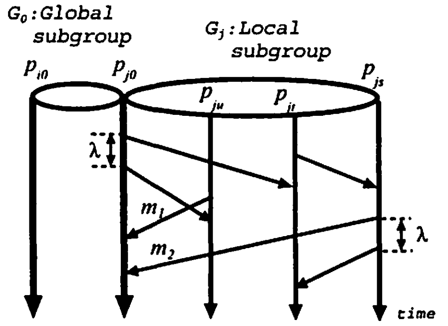

Figure 7. Evaluation model.

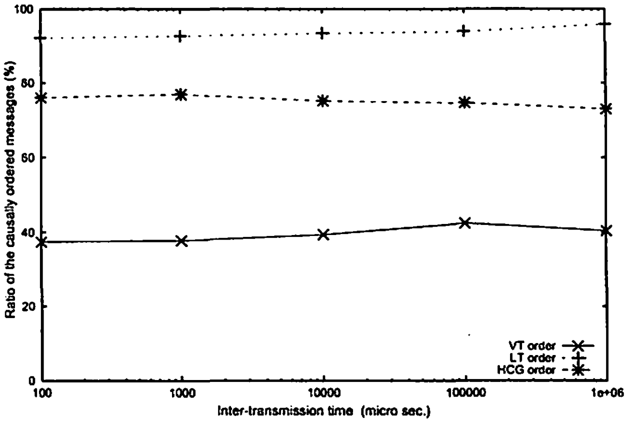

Figure 8. Ratio of causally ordered messages.

5 Concluding Remarks

In distributed applications, a large number of peer processes distributed in various types of networks are cooperating. In this paper, we proposed a hierarchically structured group where local subgroups are interconnected with the Internet and local processes are interconnected in local and personal area networks. can Each subgroup takes usage of its own clock synchronization mechanism. We discussed how to causally order messages exchanged among subgroups with different clocks, i.e. vector, and linear, and physical clock. If messages are ordered according to the local ordering mechanism in each subgroup, some message m_1 precedes another message m_2 in one subgroup even if m_1 and m_2 are causally concurrent in a global group. In this paper, we discussed how to prevent from unnecessary ordering of messages. We showed the number of messages unnecessarily ordered can be decreased in our protocol named HCG protocol than the traditional linear time and real time protocols.

References

[1] R. Baldoni, R. Beraldi, R. Friedman, and R. van Renesse. The Hierarchical Daisy Architecture for Dausal Delivery. *Distrib. Syst. Engng.*, 6(2):71–81, 1999.

[2] K. P. Birman and R. V. Renesse. *Reliable Distributed Computing with the Isis Toolkit.* IEEE Computer Society Press, 1993.

[3] Bluetooth SIG, Inc. *Bluetooth V1.1 Core Specifications*, 2001.

[4] X. Chen, L. E. Moser, and P. M. Melliar-Smith. Reservation-Based Totally Ordered Multicasting. In *Proc. of IEEE ICDCS-16*, pages 511–519, 1996.

[5] L. Huang, T. Lai, and D. Zhou. On the Scalability of IEEE 802.11 Ad Hoc Networks. In *Proc. of The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2002)*, pages 172–181, 2002.

[6] L. Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. *Comm. ACM*, 21(7):558–565, 1978.

[7] LAN MAN Standards Committee of the IEEE Computer Society. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher speed Physical Layer (PHY) Extension in the 2.4 GHz Band*, Sept. 2001.

[8] F. Mattern. Virtual Time and Global States of Distributed Systems. *Parallel and Distributed Algorithms*, pages 215–226, 1989.

[9] D. L. Mills. Network Time Protocol. RFC 1350, 1992.

[10] L. E. Moser, Y. Amir, P. M. Melliar-Smith, and D. A. Agarwal. Extended Virtual Synchrony. In *Proc. of IEEE ICDCS-14*, pages 56–65, 1994.

[11] A. Oram. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies.* O'Reilly & Associates, 2001.

[12] T. Peatfield. *Network Systems Tutorial for IEEE Std 802.3: Repeater Functions and System Design Topology Considerations for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Local Area Networks (LANs).* Inst of Elect & Electronic, 1995.

[13] R. Prasad. Basic Concept of Personal Area Networks. *WWRF, Kick off Meeting*, 2001.

[14] T. Tachikawa, H. Higaki, and M. Takizawa. Group Communication Protocol for Realtime Applications. In *Proc. of IEEE ICDCS-18*, pages 40–47, 1998.

[15] P. Verissimo, L. Rodrigues, and A. Casimiro. Cesiumspray: a Precise and Accurate Global Time Service for Large-scale Systems. *Real-Time Systems*, 12(3):243–294, 1997.