Hierarchical Group Protocol for Parallel Transmission of Multimedia Data

Yasutaka Nishimura[1], Naohiro Hayashibara[1], Tomoya Enokido[2], and Makoto Takizawa[1]
[1]*Tokyo Denki University, Japan*
{*yasu, haya, taki*}*@takilab.k.dendai.ac.jp*

[2]*Rissho University, Japan*
eno@ris.ac.jp

Abstract

A hierarchical group is composed of subgroups where each subgroup communicates with another subgroup through a gateway process. A gateway process implies performance bottleneck and a single point of failure since every message passes the gateway process. In order to increase the throughput and reliability of inter-subgroup communication, messages are in parallel transmitted in a striping way through multiple channels between the subgroups. We discuss a striping multi-channel inter-subgroup communication protocol (SMIP) for realizing high-performance multimedia communication among large number of peer processes. We evaluate SMIP in terms of stability of bandwidth.

1. Introduction

In various types of applications, multimedia messages are exchanged among application processes. Each application requires a system to support some quality of service (QoS) like bandwidth, delay time, and message loss ratio. It is critical to discuss how to support each of huge number and various types of application processes with enough QoS in change of network environments and requirements. In this paper, we discuss how to support flexible group communication service of multimedia data for applications.

Traditional communication protocols, TCP [15] and RTP [17] support processes with reliable one-to-one and one-to-many transmission of data, respectively. Recently, multiple connections are used to in parallel transmit data from a process to another process in *network striping* technologies like GridFTP [1], SplitStream [5], and PSockets [18] in order to increase the throughput.

In the group communication, processes not only send messages to but also receive messages from multiple processes and messages have to be causally ordered delivered [12]. Takizawa and Takamura [21] discuss how to support the causally ordered delivery of messages in a hierarchical group by using the vector clock whose size is the total number of processes. Here, a group is composed of subgroups where processes in different subgroups exchange messages via gateway processes. Taguchi *et al*. [20] discuss two-layered and multi-layered group protocols which adopt a type of vector clock whose size is the number of processes in a subgroup. In Totem [11], messages are ordered by using the token passing mechanism. The protocol cannot be adopted for a large-scale group due to delay time to pass a token. Kawanami *et al*. [8] discuss a hierarchical group where real-time clock is used to causally deliver messages to processes. The authors [13] discuss how to design a hierarchical group from large number of processes so that the delay time can be decreased.

In these hierarchical protocols, a gateway process in one subgroup exchanges messages with other subgroups. Each gateway process implies not only performance bottleneck but also single point of failure. In this paper, we discuss a *striping multi-channel inter-subgroup communication protocol* (*SMIP*) where a pair of subgroups are interconnected through multiple channels among multiple processes in the subgroups to realize parallel, striping communication [18] and to increase the reliability. That is, a pair of subgroups communicate with one another in a many-to-many type of communication. In addition, the number of connections among subgroups can be dynamically changed.

In section 2, we discuss a model of a hierarchical group. In section 3, we discuss SMIP. In section 4, we evaluate SMIP in terms of bandwidth compared with the traditional one-to-one communication.

2. Striping Hierarchical Group

2.1. Hierarchical group

A *group* of multiple peer processes are cooperating by exchanging messages in order to achieve some objectives. In one-to-one, (unicast) and one-to-many (multicast) communications, each message is *reliably* routed to one and more than one process, respectively, in tree routing protocols [15]. On the other hand, a process sends a message to multiple processes while receiving messages from multiple processes in group communications [3, 5, 12, 19]. Here, a message m_1 *causally precedes* another message m_2 ($m_1 \rightarrow m_2$) if and only if (iff) a sending event of m_1 *happens before* [9] a sending event of m_2 [3]. Here, every process is required to deliver the message m_1 before the other message m_2. Linear clock [9], vector clock [10], and physical clock with a GPS time server [8] are used to causally deliver messages in distributed systems.

In a *flat* group, every pair of peer processes directly ex-

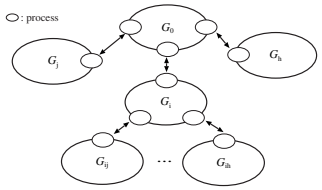

Figure 1. Hierarchical group.

change messages with one another. Most group protocols [12, 19] are discussed for flat groups and adopt the vector clock to causally order messages. Due to computation and communication overheads $O(n)$ to $O(n^2)$ for the total number n of processes in a flat group with the vector clock, it is difficult to support a large number n of processes with group communication service like causally ordered delivery of messages. In addition, it is difficult for each process to maintain the membership in a scalable group. In order to realize a scalable group, a *hierarchical group* is discussed. Processes in a group G are partitioned into multiple subgroups. There is one *root* subgroup G_0 which is connected with subgroups $G_1, \ldots, G_k$ ($k \geq 1$). Then, each subgroup G_i is furthermore connected with subgroups $G_{i1} \ldots G_{ik_i} (k_i \geq 0)(i = 1, \ldots, k)$ as shown in Figure 1. Here, G_i is a parent subgroup of a child subgroup G_{ij}. In a hierarchical group [20], every pair of parent and child subgroups G_i and G_{ij} communicate with one another through one channel between a pair of gateway processes g_i and g_{ij} as shown. Hence, the gateways and communication channel between the gateways imply performance bottleneck and a single point of failure.

2.2. Inter-subgroup communication

In order to increase the performance and reliability of inter-subgroup communications, we newly discuss a Striping Multichannel Inter-subgroup communication Protocol (*SMIP*) where every pair of parent and child subgroups communicate through multiple channels as shown in Figure 2. For example, a process p_{i2} in a parent group G_i communicates with a pair of processes p_{ij1} and p_{ij2} in a child subgroup G_{ij}, and another process p_{i3} in G_i communicates with processes p_{ij1} and p_{ij4} in G_{ij} as shown in Figure 2. Each of the processes p_{i2}, p_{i3}, p_{ij1}, p_{ij2}, and p_{ij4} plays a role of gateway between the subgroups G_i and G_{ij}. Thus, a gateway in the parent subgroup G_i is interconnected with gateways p_{ij1} and p_{ij2} in the child subgroup G_{ij} through multiple channels. A gateway in G_{ij} has also multiple channels with gateways in G_i. Thus, a pair of parent and child subgroups are interconnected with many-to-many communication channels among gateways. Thus, a subgroup G_{ij} communicates with one parent subgroup G_i and child subgroups $G_{ij1} \ldots G_{ijk_{ij}} (k_{ij} \geq 0)$. A gateway p_{ij} in G_{ij} communicates with processes in other subgroups. Gateway processes communicating with G_i and G_{ijh} are referred to as *upward* and *downward* gateways, respectively, in a subgroup G_{ij}. In Figure 2, p_{ij1} and p_{ij2} are upward gateway processes in G_{ij}, and processes Normal processes are ones which are not gateways. Gateway processes also have functions for transmitting messages in a same way as normal processes. In a *root* subgroup, there are normal processes and only downward gateway processes. A *leaf* subgroup includes normal processes and only upward gateways. If all the leaf subgroups are at the same layer of the hierarchy, the hierarchical group is *height-balanced*.

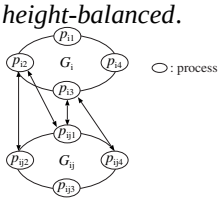

Figure 2. Striping inter-subgroup communication.

Since the communication and computation overheads are $O(n^2)$ for number n of processes in a group, the maximum size of each subgroup is bounded due to the limited computation power of each process. The number s_i of processes in a subgroup G_i is smaller than some constant value S ($s_i \leq S$). The smaller size of each subgroup is, the more number of subgroups are connected, i.e. the height or breadth is increased. If the number k_i of child subgroups of a subgroup G_i is increased, the overhead for inter-group communication is increased. Hence, $s_i \geq s$ where a constant s shows the minimum number of processes in G_i. A pair of the constants S and s are decided based on the computation power of each process and the required delay time. Processes leave and join a subgroup G_i, e.g. due to the fault and recovery from the fault, respectively. In addition, quality of service (QoS) supported by a process or network is changed. Processes in a subgroup may move to another subgroup to satisfy the performance and QoS requirements. If $s_i > S$, the subgroup G_i is split. On the other hand, if $s_i < s$, G_i is merged into a sibling subgroup G_j, or processes in G_i and its sibling subgroup G_j are redistributed in G_i and G_j. Thus, $S \geq s_i \geq s$. A hierarchical group is dynamically kept height-balanced given the constraints of the size of each subgroup G_i as discussed in B-tree [2]. The authors discuss how to construct and maintain a hierarchical group from a large number of peer processes [13].

In this paper, we assume each process broadcasts a message to all the processes in a group G. Suppose a process p_{ijs} originally transmits a message m in a subgroup G_{ij}. The messages are forwarded to processes in other subgroups of the group G as follows :

1. The process first sends a message m to every process in the subgroup G_{ij}.
2. An upward gateway forwards the message m up to downward gateways in the parent subgroup G_i.
3. Downward gateways forward the message m down to upward gateways in child subgroups $G_{ij1}, \ldots G_{ijk_{ij}}$.

In each subgroup, a process delivers messages to all the processes by using its own synchronization mechanism like vector clock [10].

3. Striping Inter-subgroup Communication

Suppose a source gateway in a subgroup G_i would like to send messages to destination gateways in another subgroup G_j.

In this paper, we take the following inter-subgroup transmission protocol from a source subgroup G_i to another destination subgroup G_j :

1. One process, say p_{is}, is taken as a source gateway in a subgroup G_i. Here, if G_i is a parent subgroup of G_j, p_{is} is a downward gateway process in G_i.
2. On receipt of a message in G_j, the source gateway process p_{is} forwards the message to some process, say p_{jt_1} in G_j. Here, p_{jt_1} is a destination process of G_j.
3. The process p_{is} sends messages to the destination gateway p_{jt_i} in the subgroup G_j on recept of messages in G_i.
4. If the channel between a pair of the gateways p_{is} and p_{jt_1} might not support enough QoS due to congestion and fault, the source gateway p_{is} takes another process p_{jt_2} as a gateway in the subgroup G_j.
5. Thus, the source gateway p_{is} in G_i sends different messages to the destination gateways p_{jt_1} and p_{jt_2} in G_j. The source gateway p_{is} distributes messages to a pair of gateways p_{jt_1} and p_{jt_2} so that both the channels with the gateways p_{jt_1} and p_{jt_2} satisfy the QoS requirement.
6. Thus, the larger bandwidth is required, the more number of destination gateways are taken in G_j. Here, let $p_{jt_1} \ldots p_{jt_h}$ be destination gateways in G_j. The source gateway p_{is} sends messages to the destination gateways $p_{jt_1} \ldots p_{jt_h}$ in G_i.

Messages are transmitted in a channel between a pair of gateway processes by the congestion control algorithm used in TCP [7]. If a pair of subgroups are interconnected in a single channel, the subgroups cannot be communicated due to the congestion and fault of the channel. In our striping multi-channel protocol, a pair of subgroups G_i and G_j are interconnected with multiple channels. Even if some channel is faulty or doe not support QoS requirement due to congestions, the subgroups can communicate with one another with enough QoS through the other operational channels. The message traffic can be distributed to multiple channels and the other channels compensate the QoS degradation of one channel even if QoS of the channel is degraded.

Messages are transmitted in each channel between a pair of source and destination gateway processes through the congestion control algorithm, *additive increase and multiplicative decrease* (*AIMD*) algorithm used in TCP [7]. In the TCP congestion control algorithm [7], two parameters, *congestion window size* (*cwnd*) and *receiver window size* (*rwnd*) are used for each channel. In our protocol, an additional parameter *requirement window* (*qwnd*) showing the size of data in the buffer is used for a set of the channels. The window size (*wnd*) of each channel is decided as follows :

$$wnd = min(cwnd, rwnd, qwnd)$$

The source gateway p_{is} in a subgroup G_i sends messages to a destination gateway p_{jt_l} in another subgroup G_j through a channel. Then, the window size is calculated. The requirement window size (*qwnd*) is decided as follows:

$$qwnd = qwnd - wnd$$

4. Evaluation

We evaluate the striping multi-channel inter-subgroup communication protocol (SMIP) for inter-subgroup communication in terms of the stability of bandwidth compared with the traditional one-channel transmission protocol. In the traditional one-to-one multimedia communication approach, protocols like RSVP [16] at a lower layer than the transport layer are used to support QoS required by applications. In our striping multi-channel approach, QoS is supported on the end-to-end basis with QoS control at network layer. In the simulator, the bandwidth of the network channel is bounded to be 30Mbps by the evaluation tool although the channel support larger bandwidth 30Mbps means the transmission speed of the digital video (DV) data.

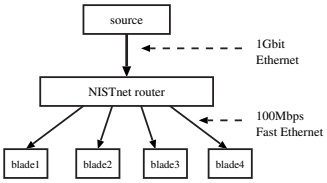

Figure 3. Data transfer arrangement for source striping communication.

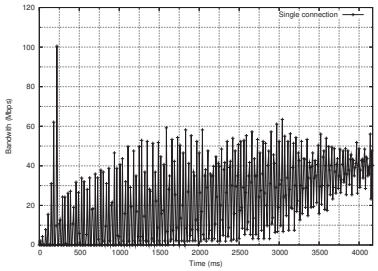

Figure 4. Bandwidth adaptation on traditional one-channel model.

Figure 3 shows the evaluation environment of SMIP. A source gateway process is realized in a computer Dell Precision 530 with dual Intel Pentium Xeon 1.8Ghz and 1.5B memory on Linux 2.6.10. Four destination gateway processes are realized in an HP Proliant BL10e blade server with Intel PentiumM 1Ghz and 512MB memory on Linux 2.4.26. These gateways are interconnected through a computer HP Proliant DL145 with dual AMD Opeteron 2.2Ghz and 2GB memory on Linux 2.4.21 named NISTnet router where NISTnet [4] is installed. The delay time between source and destination gateways is emulated to be 40 milliseconds by using the NISTnet.

In the evaluation, the source gateway process sends multimedia messages like DV data with 30Mbps. The NewReno algorithm [6] of TCP is used for transmitting messages in each channel. The data transmission procedure of TCP is emulated over UDP/IP [14]. Figure 4 shows how the bandwidth is changed for time in the traditional one-channel transmission. The bandwidth supported is largely changed. Figure 5 shows the bandwidth in our striping multi-channel transmission. Compared with the one-channel transmission, the striping multi-channel transmission supports more stable bandwidth, i.e. 30Mbps. The DV data is required to be transmitted with bandwidth 30Mbps. In addition, the bandwidth of 30Mbps

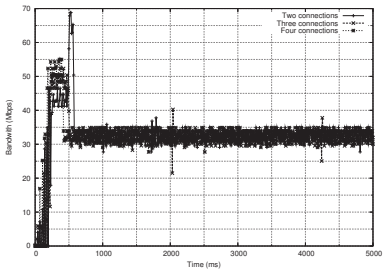

Figure 5. Bandwidth adaptation on striping multi-channel model.

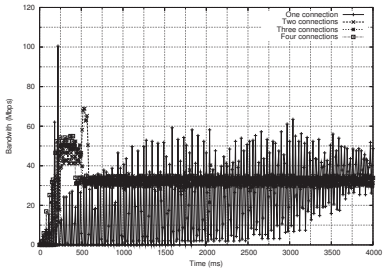

Figure 6. Bandwidth adaptation.

can be continually supported. However, the bandwidth supported by the traditional one-channel way is not so stable that the DV data cannot be transmitted. Figure 6 shows both the one-channel and the striping multi-channel ways to show how stable the striping multi-channel way is. Even if QoS is degraded in a channel, messages which cannot be transmitted in the channel can be transmitted through the other channels in the striping multi-channel approach.

5. Concluding Remarks

We discussed the striping multi-channel inter-subgroup communication protocol (*SMIP*) where subgroups are hierarchically interconnected through gateway processes. In order to improve the reliability and throughput of the inter-subgroup communication, a pair of parent and child subgroups are interconnected through multiple communication channels between multiple gateway processes in the subgroup. Gateways in different subgroups exchange messages through multiple channels in the striping transmission. In the evaluation, we showed that SMIP can support the higher stability of the bandwidth compared with the traditional one-channel protocol.

References

[1] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke. Data Management and Transfer in High-performance Computational Grid Environments. *Parallel Computing Journal*, 28(5):749–771, 2002.

[2] R. Bayer and E. M. McCreight. Organization and Maintenance of Large Ordered Indices. *Acta Informatica*, 1:173–189, 1972.

[3] K. Birman. Lightweight Causal and Atomic Group Multicast. *ACM Trans. on Computer Systems*, pages 272–290, 1991.

[4] M. Carson and D. Santay. NIST Net: a Linux-based Network Emulation Tool. *Computer Communication Review*, 33(3):111–126, 2003.

[5] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream: High-bandwidth Multicast in a Cooperative Environment. In *Proc. of the 19th ACM Symp. on Operating Systems Principles (SOSP2003)*, pages 298–313, 2003.

[6] S. Floyd, ICSI, T. Henderson, Boeing, A. Gurtov, and TeliaSonera. The NewReno Modification to TCP's Fast Recovery Algorithm. *Request for Comments 3782*, 2004.

[7] V. Jacobson. Congestion Avoidance and Control. In *Proc. of the ACM Symp. on Communications Architectures and Protocols (SIGCOMM '88)*, pages 314–329, 1988.

[8] S. Kawanami, T. Enokido, and M. Takizawa. Heterogeneous Groups to Causally Ordered Delivery. In *Proc. of the IEEE 6th International Workshop on Multimedia Network Systems and Applications (MNSA 2004)*, pages 70–75, 2004.

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

[10] F. Mattern. Virtual Time and Global States of Distributed Systems. *Proc. of the International Workshop on Parallel and Distributed Algorithms*, pages 215–226, 1989.

[11] L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia, and C. A. Lingley-Papadopoulos. Totem: A Fault-Tolerant Multicast Group Communication System. *Communications of the ACM*, 39(4):54–63, 1996.

[12] A. Nakamura and M. Takizawa. Causally Ordering Broadcast Protocol. In *Proc. of the IEEE 14th International Conf. on Distributed Computing Systems (ICDCS 94)*, pages 48–55, 1994.

[13] Y. Nishimura, T. Enokido, and M. Takizawa. Design of a Hierarchical Group to Realize a Scalable Group. In *Proc. of the IEEE 19th International Conf. on Advanced Information Networking and Applications (AINA 2005)*, volume 1, pages 9–14, 2005.

[14] J. Postel. User Datagram Protocol. *Request for Comments 768*, 1980.

[15] J. Postel. Transmission Control Protocol. *Request for Comments 0793*, 1992.

[16] E. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. *Request for Comments 2205*, 1997.

[17] H. Schulzrinne, R. Casner, S. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-time Applications. *Request for Comments 1889*, 1996.

[18] H. Sivakumar, S. Bailey, and R. L. Grossman. PSockets: The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks. In *Proc. of the 2000 ACM/IEEE Conf. on Supercomputing*, page http://www.sc2000.org/proceedings/techpapr/papers/pap.pap240.pdf, 2000.

[19] T. Tachikawa, H. Higaki, and M. Takizawa. Group Communication Protocol for Realtime Applications. In *Proc. of the IEEE 18th International Conf. on Distributed Computing Systems (ICDCS 98)*, pages 40–47, 1998.

[20] K. Taguchi, T. Enokido, and M. Takizawa. Causal Ordered Delivery for a Hierarchical Group. In *Proc. of the IEEE 10th International Conf. on Parallel and Distributed Systems (ICPADS 2004)*, pages 485–492, 2004.

[21] M. Takizawa, M. Takamura, and A. Nakamura. Group communication protocol for large group. In *Proc. of the IEEE Conf. on Local Computer Networks (LCN '93)*, pages 310–319, 1993.