Performance Evaluation of Split Connection Methods for Session-based Group-oriented Communications

Hiroshi Emina, Hiroyuki Koga
Dept. of Information and Media Engineering
University of Kitakyushu, Japan
Email: {emihiro, koga}@net.is.env.kitakyu-u.ac.jp

Masayoshi Shimamura, Takeshi Ikenaga
Network Design Research Center
Kyushu Institute of Technology, Japan
Email: {shimamura, ike}@ndrc.kyutech.ac.jp

***Abstract*—Group-oriented communication services have become more attractive due to the diversifying demands of Internet users. In these services, the bandwidth available to the link in the network is inefficiently consumed by multiple TCP connections and these individual connections compete against each other. Thus, current group-oriented services cannot use network resources efficiently. To achieve efficient group-oriented communication, we previously proposed a split connection method, which introduced session agents to eliminate redundant connections with a packet caching function. Furthermore, we showed the effectiveness of the proposed method and raised the cache utilization issue of the session agents under high traffic loads. In this paper, we propose packet transmission control methods to improve the cache efficiency of the session agents, especially under high traffic loads.**

***Keywords*-group-oriented communication, session management, connection control, packet transmission control**

I. Introduction

New communication services have become more attractive because of the diversifying demands of Internet users [1]. A one-to-one communication style, based on the client-server model, has been widely used for web and mail services. However, in recent years, a group-oriented communication style, based on a peer-to-peer model, has realized content exchange services, such as file sharing and online game services, among multiple users. In such group-oriented communication services, users need to share session information with other group members to manage the session. This session information consists of member IP addresses and group identifiers. In addition, content shared or exchanged by users in a group must be reliably delivered to other group members.

The current Internet has a significant difficulty in providing these services efficiently, which is in the reliable content sharing among group members. For example, consider that a sender establishes individual TCP connections with correspondent receivers in a group using current Internet technologies such as the Application Layer Multicast (ALM). In this case, the sender establishes additional TCP connections as the number of receivers increase. Consequently, the bandwidth available to the link is inefficiently consumed by multiple TCP connections that deliver the same data and these individual connections compete with each other. Thus, group-oriented communication services on the current Internet use network resources inefficiently.

To achieve efficient group-oriented communication services, we previously proposed a split connection method, which introduced a session agent with a packet caching function on intelligent intermediate nodes to eliminate the redundant connections [2]. Although this method improves the efficiency of network resource usage, it has an issue of cache utilization of the session agent under high traffic loads. Therefore, in this paper, we propose packet transmission control methods to improve the session agent cache efficiency, especially under high traffic loads. Furthermore, we show the efficiency of the proposed method by means of simulations.

II. Related work

Many researchers have actively studied technologies to provide group-oriented communications. In this section, we describe IP multicast and ALM mechanisms as typical group-oriented communication technologies. Then, we introduce the split connection method from our previous work.

A. Multicast

Applications based on IP multicast technology provide one-to-many communication services, such as real-time audio and video streaming services over the Internet. Although IP multicast provides flexible and efficient communication, it commonly supports unreliable datagram services. Therefore, IP multicast does not meet the need for reliable group-oriented communication service.

TCP-based ALM technology is commonly used to provide reliable group-oriented communications. In ALM networks, a sender transmits data to other members on the basis of a one-to-one communication style, namely, each member duplicates and forwards data packets to other members. To achieve effective communication services, researchers have proposed a number of ALM architectures [3]–[7]. Although ALM provides flexible group-oriented communication, it results in an inefficient use of network resources. To describe the problem of current group-oriented communication, we

 ISBN:978-1-61208-113-7

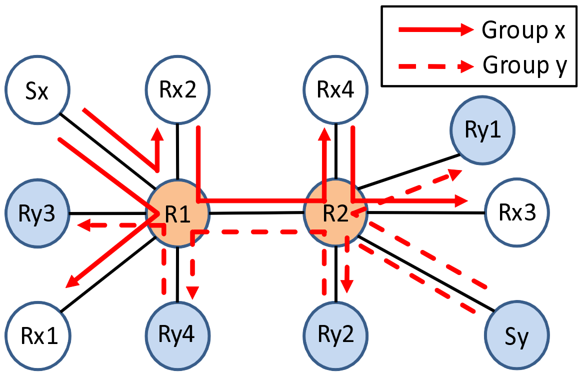


Figure 1. Group-oriented communications with TCP-based ALM

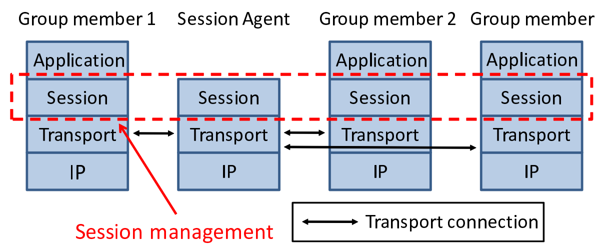


Figure 2. Layered structure of split connection method

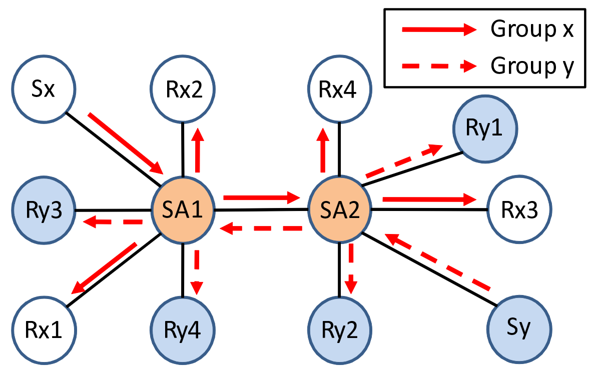


Figure 3. Group-oriented communications with a split connection method

show an example of group-oriented communications with TCP-based ALM in Figure 1. In this figure, two communication groups (x and y) exist, each consisting of one TCP sender and four TCP receivers. In group x, when sender Sx wants to transmit data to the four receivers, it needs to establish individual TCP connections with receivers Rx1 and Rx2. Rx2 then establishes TCP connection with Rx4, and Rx4 establishes another with Rx3 to forward the data. Similarly, in group y, each member also establishes multiple individual TCP connections with other members in the same group. In this situation, the bandwidth available to the link between Sx and R1, Sy and R2 are inefficiently consumed by multiple TCP connections that deliver the same data and these TCP connections compete with each other.

This inefficient group-oriented communication is caused by the management of the relationship between the session and transport connection in the current Internet architecture. To be more precise, the current Internet architecture does not have a session management layer and leaves session management to the application layer. Thus, each member must manage the session among the members on their application layer in group-oriented communications. Thus, each member establishes individual transport connections with other members on an end-to-end basis, and the number of connections increases proportionally to the number of receivers.

B. Split connection method

The current Internet architecture has a significant issue with achieving efficient group-oriented communications because of the lack of intelligent session management. To resolve this issue, in a previous work, we proposed a split connection method [2] based on a new network architecture concept [8]. In this method, we introduced an intelligent intermediate node, called a session agent. Figure 2 shows the layered structure of proposed architecture. In this figure, we assume that three group members and one session agent form a group. In this architecture, the session layer is inserted between the application and transport layers to efficiently manage a session among group members. The session layer provides several functions to manage join and leave requests from group members and to keep an end-to-end semantics among group members, which are provided by the application layer in the traditional architecture. This session management in the session layer enables this architecture to handle transport layer connections flexibly. More specifically, the session agent divides end-to-end TCP connections into multiple TCP connections partitioned by each session agent with packet caching, duplicating and forwarding functions. Consequently, the proposed method can improve network efficiency by aggregating redundant multiple TCP connections into one TCP connection in the same link.

Figure 3 depicts an example of group-oriented communications with the proposed method under conditions similar to those in Figure 1. In group x, all end-to-end TCP connections are split into multiple TCP connections. As a result, the sender Sx can aggregate two TCP connections into one connection and transmit a packet to session agent SA1 through the connection. After that, SA1 duplicates the packet into three flows, to the next session agent SA2, and group members Rx1 and Rx2. Similarly, SA2 duplicates and forwards the received packets to Rx3 and Rx4. For group y, the splitting of end-to-end connections can be performed similar to group x. As a consequence, the proposed method reduces redundant multiple TCP connections and improves network efficiency.

 ISBN:978-1-61208-113-7

III. Packet transmission control method

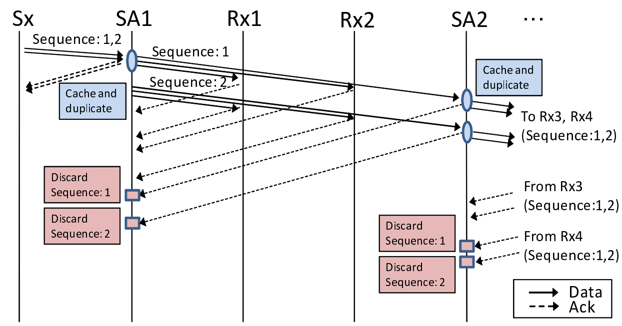
The split connection method improves network efficiency because of the aggregation of redundant TCP connections into one connection. However, this method has an issue in the packet cache utilization of session agents, especially for high traffic loads. Therefore, a sender should take into account the size of the packet cache of session agents when sending data packets. Therefore, to improve packet cache efficiency, we propose two packet transmission control methods: node-to-node and end-to-end.

A. Node-to-Node control

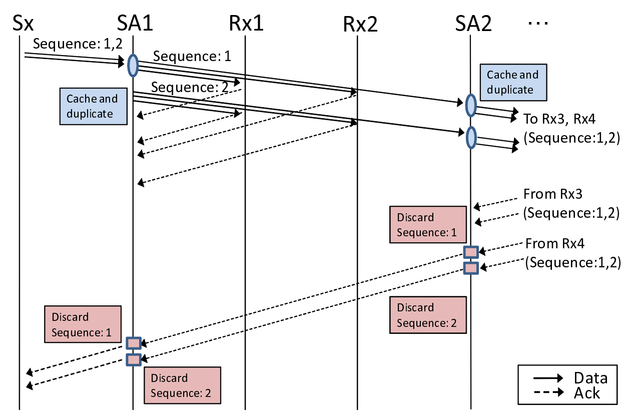
In node-to-node packet transmission control, each TCP connection partitioned by session agents controls its transmission rate independently, namely, a receiver (including session agents) requests the next data packet to a sender immediately after it receives data packets, regardless of the transmission state to other group member. Therefore, each sender (including group members and session agents) can optimize the available link bandwidth. Figure 4(a) illustrates the operation of session agents when a member in group x communicates with other members in Figure 3. When session agent SA1 receives successive packets of sequence number 1 and 2 from sender Sx, it caches the received packets, and then immediately returns acknowledgment (ACK) packets to the sender. After that, SA1 duplicates the received packets to transmit them to receivers Rx1, Rx2, and the session agent SA2. It keeps the cached packets for future retransmission until all packets sent are acknowledged. When it receives ACK packets from all receivers and the session agent, it discards the cached packet. Note that session agents perform flow control to avoid the overflow of an output buffer. It means that the output packets transmitted from session agents are not lost when forwarding the cached packets.

B. End-to-End control

In the end-to-end packet transmission control method, session agents collaborate to control the transmission rate of each partitioned TCP connection on an end-to-end basis. This method controls transmission rates with regard to the bottleneck link bandwidth between group members. Figure 4(b) illustrates the operation of session agents in this method. Similar to Figure 4(a), this method also provides packet caching, duplicating, and forwarding functions. The difference from the node-to-node control method is that the session agent returns ACK packets to the sender after receiving it from all the receivers and session agents. Consequently, session agents discard cached packets before they receive new data packets, and this suppresses the packet cache size of session agents. Note that the output packets transmitted from session agents are not lost, as described in section III-A.



(a) Node-to-Node packet transmission control method



(b) End-to-End packet transmission control method

Figure 4. Operation of session agent

IV. Simulation environment

We evaluate the performance of the proposed methods through computer simulation in contrast with the unicast-based group communication method (traditional method). We used the network simulator ns-2.31, after adding the functions of the proposed methods.

A. Simulation model

Figure 5 shows the network topology used in this simulation. In this model, all end nodes which be discretely located in four networks *a, b, c* and *d* form a group and communicate with each other. Nodes m*a1*, m*bx*, m*cx*, and m*dx* (x = 1–5) represent group members. *a, b, c*, and *d* represent network identifiers and x represents the member identifier on each network. In one-to-many communication, node m*a1* sends data to all receivers through session agents SA1 and SA2, while nodes m*a1*, m*b1*, m*c1*, and m*d1* send data to members whose member identifier is *1* in the many-to-many communication. Note that all session agents operate as normal routers when the traditional method is employed.

Table I summarizes the simulation parameters. The propagation delay time of the link between nodes SA1 and SA2 varies from 10 to 100 ms and that of other links is set to 10 ms. The bandwidth of all links is set to 100 Mb/s. In addition, the number of members on each network varies

 ISBN:978-1-61208-113-7

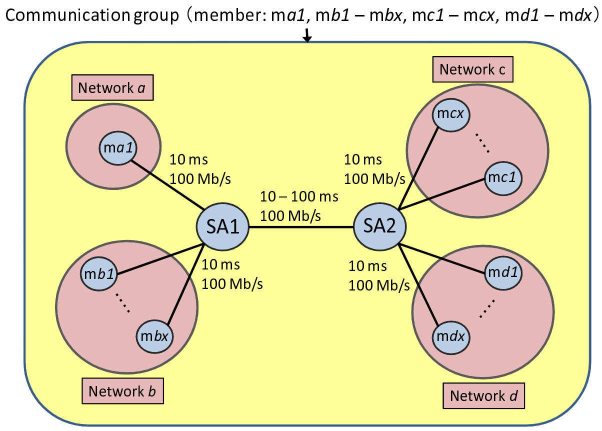


Figure 5. Simulation model

Table I
SIMULATION PARAMETERS

The number of group members on each network	1–5
Output buffer size of router	200 packet
Buffer size of SA for packet caching	∞
Data size	5 MB
Transport protocol	TCP SACK
Packet size	1500 B
Simulation time	100 s

from 1 to 5. To investigate the required cache size of session agents, the buffer size of the session agent for packet caching is set to infinity. The output buffer size of each node is set to 200 packets. All senders employ TCP SACK for data transmission, and the TCP packet size is set to 1500 Bytes.

After the simulation starts, the sender repeatedly transmits data with a size of 5 MB, based on the following traffic generation model, to vary the traffic load. In our simulation, the traffic load is defined as $t_r/(t_r + t_i)$, where t_r [s] and t_i [s] represent the required time for data transmission and the idle time, respectively. Each send time of the sender is $(t_r+t_i)(x+i)$ [s], where x and i represent a uniform random variable ranging from -0.5 to 0.5 and an incremental variable, respectively. We conduct simulation experiments for 100 seconds each and show the performance of each method by the mean value of five simulation runs.

B. Evaluation indices

To evaluate the effectiveness of the proposed methods, we focus on bandwidth consumption, reception throughput, effective throughput, and the cache ratio of session agents as evaluation indices. In group-oriented communications, the amount of data that all group members are correctly received is important. Thus, the reception throughput is defined as the minimum value of the reception throughput of each member. To evaluate the transmission efficiency, the effective throughput is defined as T_r/D, where T_r and D represent the reception throughput and the transmission rate of the sender, according to each traffic load, respectively. To investigate cache efficiency, the cache ratio is defined as C_{max}/D_{max}, where C_{max} and D_{max} represent the maximum cache data size and transmitted data size of all senders, respectively.

V. SIMULATION RESULT

In this section, we discuss the effectiveness of the proposed methods compared with the traditional method.

A. One-to-many communications

Figure 6 shows the bandwidth consumption of a link between nodes m*a1* and SA1 when the number of members in each network varies from 1 to 5. In this figure, "Unicast," "End-End," and "Node-Node" represent the results of the traditional method, the end-to-end, and the node-to-node packet transmission control methods, respectively. The bandwidth consumption of the proposed methods is lower than that of the traditional method because the traditional method requires multiple connections in the link in proportion to the number of members, whereas both proposed methods can aggregate these connections into one connection. Therefore, both proposed methods consume only the required bandwidth for data transmission and improve network efficiency.

Next, Figure 7 shows the reception throughput when the propagation delay time of a link between nodes SA1 and SA2 is 10 and 100 ms. The reception throughput of the proposed methods is higher than that of the traditional method. In the traditional method, a member whose round trip time (RTT) is longer than that of other members attains lower throughput performance due to the competition among multiple connections. Especially, when the traffic load is high, the short RTT member occupies most of an available bandwidth. Therefore, the reception throughput which means a minimum throughput among members reaches zero as the traffic load increases. On the other hand, both proposed methods achieve a high reception throughput in proportion to the traffic load because of the avoidance of competing with multiple connections. Moreover, both proposed methods decrease the impact of the propagation delay time due to the efficient utilization of the link bandwidth. Therefore, both proposed methods improve throughput performance.

Finally, Figure 8 shows the effective throughput. In the traditional method, the effective throughput decreases in proportion to the traffic load due to degradation of the reception throughput, while the proposed methods maintain a high effective throughput regardless of the traffic load. These results show that, the proposed methods improve network efficiency and communication performance.

B. Many-to-many communications

In this section, we discuss the results of the case where m*a1*, m*b1*, m*c1*, and m*d1* are senders. Figure 9 shows the average bandwidth consumption of four links between

 ISBN:978-1-61208-113-7

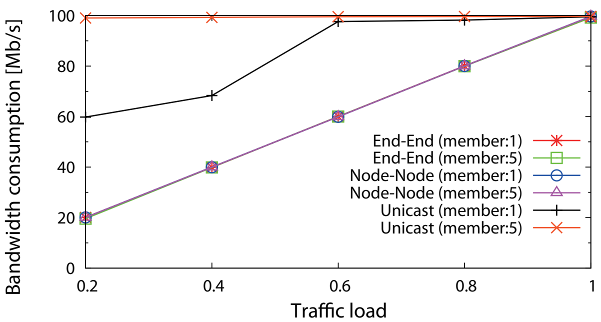


Figure 6. Bandwidth consumption of a link between ma*1* and SA1

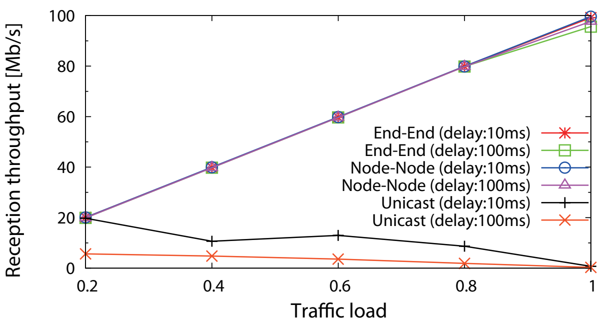


Figure 7. Reception throughput

each sender and the SA. When the traffic load is low, the bandwidth consumption of both proposed methods is lower than that of the traditional method because each sender in the traditional method must transmit the same data to three members, whereas both proposed methods eliminate redundant data transmissions. On the other hand, when the traffic load is high, the bandwidth consumption of the node-to-node packet transmission control method is higher than that of the other methods. In the node-to-node packet transmission control method, each sender can optimize the available link bandwidth, so the bandwidth consumption increases in proportion to the traffic load. In the end-to-end packet transmission control method, competition among multiple flows occurs, so that the throughput of each flow from all senders is degraded, which decreases the bandwidth consumption.

Next, Figure 10 shows the average reception throughput of four receivers. The node-to-node packet transmission control method and the traditional method degrade the reception throughput in proportion to the traffic load. This is because the traditional method decreases the transmission rate of all members due to the competition among multiple connections. In the node-to-node packet transmission control method, the transmission rate of each sender increases in proportion to the traffic load, leading to an enormous increase in the number of ACK packets sent from session agents to each member. As a consequence, the reception throughput decreases drastically due to the obstacle of

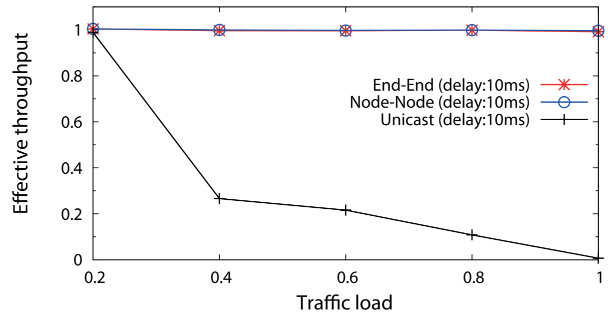


Figure 8. Effective throughput

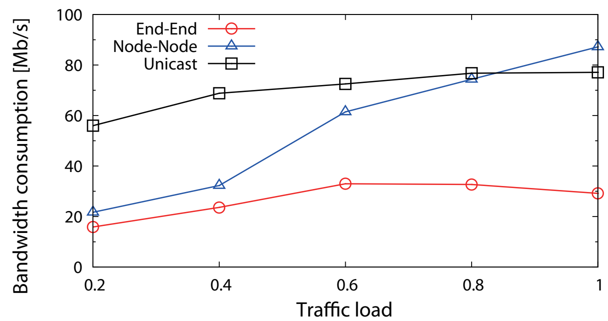


Figure 9. Average bandwidth consumption of links between senders and SAs

data transmission that is caused by the explosion in ACK packet transmission from session agents. On the other hand, in the end-to-end packet transmission control method, the transmission rate of each sender is lower than that of the node-to-node packet transmission control method due to the competition among multiple connections. As a result, each session agent can send more data packets to each receiver compared with the node-to-node packet transmission control method—it avoids the high level of ACK packet forwarding. Therefore, the end-to-end packet transmission control method provides high performance throughput.

Finally, Figure 11 shows the average effective throughput of the four senders. When the traffic load is high, the effective throughput of the end-to-end packet transmission control method is higher than that of other methods. From the above results, we can see that the end-to-end packet transmission control method improves network resource usage and throughput performance, even under high traffic loads.

C. *Cache ratio of session agent*

In this section, we investigate the required cache size of session agents. Figure 12 shows the cache ratio of SA1 when the sender is ma*1* only and ma*1*, mb*1*, mc*1*, and md*1* are senders. When the sender is ma*1* only, the cache ratio is less than 10%. This is because the difference between the received and transmitting rate of session agents is small. On the other hand, when four group members are senders,

 ISBN:978-1-61208-113-7

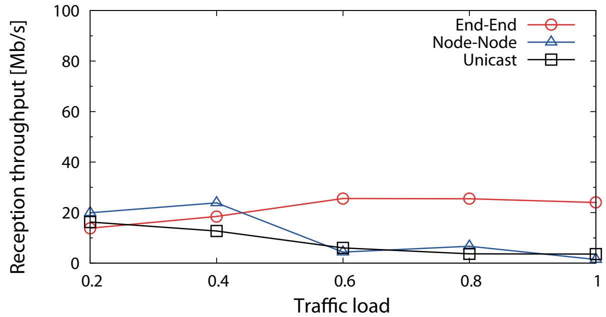


Figure 10. Average reception throughput

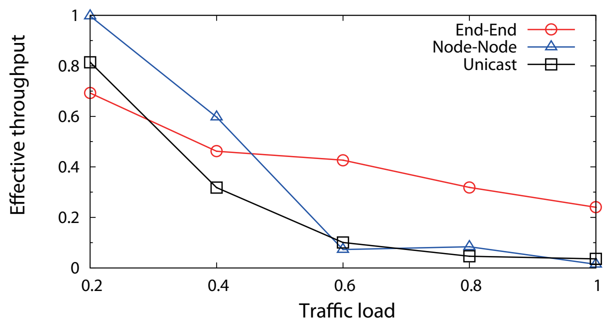


Figure 11. Average effective throughput

the cache ratio is higher compared with the case when ma*1* is the only sender. When four group members are senders, the difference between the received and transmitting rate of the session agents becomes large due to the competition among multiple flows. Moreover, the cache ratio reaches about 80% in the node-to-node packet transmission control method. In this method, the session agent requests new data packets to the sender as soon as it receives data packets, so that it receives a large number of data packets before discarding cached packets. On the other hand, in the end-to-end packet transmission control method, the session agent requests new data packets to the sender after receiving ACK packets from all receivers. Therefore, this method suppresses the packet cache ratio. These simulation results show that the end-to-end packet transmission control method improves cache efficiency.

VI. Concluding Remarks

We focus on the efficient group-oriented communication based on the split connection methods, which introduced a session agent with a packet caching function. In this paper, we proposed packet transmission control methods to improve cache efficiency of session agents, especially under high traffic loads. From our performance evaluations, the proposed methods can improve throughput performance and cache efficiency. In particular, the end-to-end packet transmission control method achieves excellent performance even under high traffic loads. In future work, we will discuss

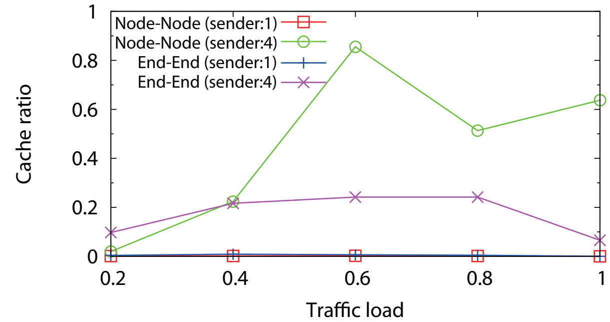


Figure 12. Cache ratio of SA1

an algorithm to find the optimal location of session agents.

Acknowledgment

This work was supported in part by the National Institute of Information and Communications Technology, Japan and the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (S) (No. 18100001).

References

[1] Y. Takahashi, K. Sugiyama, H. Ohsaki, T. Yagi, J. Murayama, and M. Imase, “Group-oriented communication: Concept and network architecture,” *Proc. First International Workshop on Security of Computer Communications and Networks (SoCCaN2008)*, pp. 649–655, August 2008.

[2] H. Emina, H. Koga, M. Shimamura, and T. Ikenaga, “Split connection scheme for session-based group communication,” *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim2009)*, August 2009.

[3] S. Y. Shi and J. S. Turner, “Multicast routing and bandwidth dimensioning in overlay networks,” *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1444–1455, October 2002.

[4] M. Kobayashi, H. Nakayama, N. Ansari, and N. Kato, “Robust and efficient stream delivery for application layer multicasting in heterogeneous networks,” *IEEE Transactions on Multimedia*, vol. 11, no. 1, pp. 166–176, January 2009.

[5] W. Wang, D. A. Helder, S. Jamin, and L. Zhang, “Overlay optimizations for end-host multicast,” *Proc. ACM International Workshop on Networked Group Communication(NGC2002)*, October 2002.

[6] L. Lao, J. H. Cui, M. Gerla, and S. Chen, “A scalable overlay multicast architecture for large-scale applications,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 449–459, April 2007.

[7] K. To and J. Y. Lee, “Parallel overlays for high data-rate multicast data transfer,” *Elsevier Computer Networks*, vol. 51, no. 1, pp. 31–42, January 2007.

[8] N. Ryoki, H. Koga, K. Kawahara, and Y. Oie, “Proposal of new layer of indirection between application and transport layers for flexible communications in IP networks,” *Proc. IEEE/IPSJ International Symposium on Applications and the Internet (SAINT2008)*, pp. 289–292, July/August 2008.

 ISBN:978-1-61208-113-7