Design of Agent-based Middleware for Extranet Management

Susumu Konno†, Kenji Sugawara††, and Tetsuo Kinoshita†

†Tohoku University, Sendai, Miyagi, JAPAN
††Chiba Institute of Technology, Narashino, Chiba, JAPAN

Summary
We propose a novel, agent-based middleware for efficient management of an extranet. To keep an extranet in full working order, its network connections (e.g., routing, bandwidth assigning, etc.) have to be customized for the specific network services in a particular group. However, such customization work imposes a heavy workload on the extranet administrator, and this makes it very difficult to achieve efficient management of the extranet. To overcome this difficulty, we propose an agent-based middleware between the application layer and the transport layer of the extranet. This middleware controls network connections based on known specifications of the extranet, and thus it provides well-customized network connections for users and extranet applications. In this paper, we explain the design of the proposed agent-based middleware, describe its implementation, and report on experimental results.
Key words:
Network Management, Extranet, Middleware, Intelligent Agent, Multi Agent System

1. Introduction

Present-day global communication networks such as the Internet have been developed to provide efficient general network services. In recent years, by using such existing network infrastructures, commercial enterprises and government organizations have constructed their own extranets for specific network services [1][2][3][4][5].

An extranet is a private network used by a group of several cooperating organizations. It is constructed as a logical network within existing global communication networks, and the organizations' intranets are connected to the extranet thorough their gateways. Unlike conventional global communication networks, which regard general network services as important, the extranet attaches' importance to the specific network services required for predetermined projects in a particular group. Therefore, the efficiency of an extranet can be improved by customizing its network connections (e.g., routing, bandwidth assigning, etc.) for the specific network services in a particular group. To customize the network connections and keep the extranet in full working order, the extranet administrator is required to perform the following tasks:

- Plan customization of network connections
- Set parameters for intranet gateways
- Exchange information by co-operation with the intranet administrators
- Monitor network quality of service (QoS)
- Modify the customization in response to changes in network environment
- Replan a customization at the end of a current project and the beginning of a new project

However, such customization work imposes a heavy workload on the extranet administrator, and this makes it very difficult to achieve efficient management of the extranet. To provide general network services efficiently, several methods for controlling network connections have been proposed in the past. These methods are classified into two types; one is regarded as an extension of a certain network layer, and the other is implemented as "middleware".

IntServ [6] and DiffServ [7] proposed by Internet Engineering Task Force (IETF), and Scout Algorithm [8] proposed by Chen et al. are implemented as extensions of a certain network layer of conventional networks. Therefore, to use these methods, some substantial modifications to the existing network infrastructure are required. On the other hand, Flexible Network Layer (FNL) [9][10][11][12], ATM Virtual Path Management [13], and User Adaptive Agent [14] are implemented as "middleware". In these approaches, since the function for controlling network connections is introduced as a middleware layer placed between the application layer and the transport layer, the existing network infrastructure can be used effectively. However, these existing methods are designed to improve the efficiency in general network services, therefore it is difficult for these methods to customize an extranet efficiently.

To deal with the difficulty in customizing an extranet, we propose a new method for managing an extranet efficiently. In the proposed method, a novel, agent-based middleware, Agent-based Network Service Middleware (ANSM), is used between the application layer and the

Manuscript received June 5, 2007
Manuscript revised June 20, 2007

transport layer. This middleware controls network connections based on the specifications of an extranet (e.g., customization plan, network condition, etc.), and thus it provides well-customized network connections for users and applications of the extranet. In this paper, we explain the design of the proposed agent-based middleware, describe its implementation, and report on experimental results.

2. Agent-based Network Service Middleware

In this paper, we assume that an extranet is constructed as a logical network in the Internet, and several intranets are connected to the extranet thorough their gateways. An example of an extranet model is shown in Fig.1.

To provide well-customized network connections for users and applications of the extranet, the proposed method used an agent-based piece of middleware, called Agent-based Network Service Middleware (ANSM), between the application layer and the transport layer of the extranet (Fig.2).

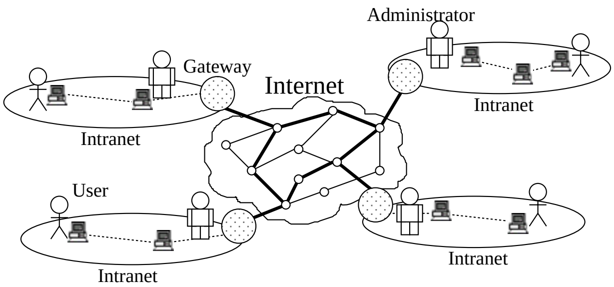


Fig.1 The Extranet: extranet is as a logical network in the Internet and several intranets.

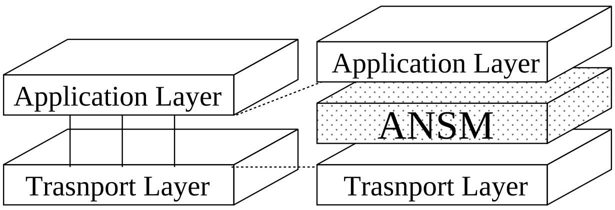


Fig.2 Conventional network application model vs proposed model with Agent-based Network Service Middleware (ANSM).

2.1 Agent-based Middleware

In a conventional extranet, the administrator first obtains the requirement from a user or application, then the administrator has to adjust several parameters in the transport layer, and finally the user or application receives the requested service. Consequently, providing the requested service to the user or application takes a long time and a heavy workload is imposed on the extranet administrator. Therefore, it would be very useful if these tasks could be carried out automatically.

Therefore, in this paper, we describe how ANSM can be used as middleware between the application layer and the transport layer as shown in Fig.3. ANSM is also designed as an agent-based system. Agents possessing knowledge of the administrator management are installed to ANSM. They obtain the requirement from a user or application, and cooperate with each other to control the transport layer automatically. Using ANSM, the requirements are sent, not to the administrator, but to the agents. As a result, the user or application can receive the requested service quickly.

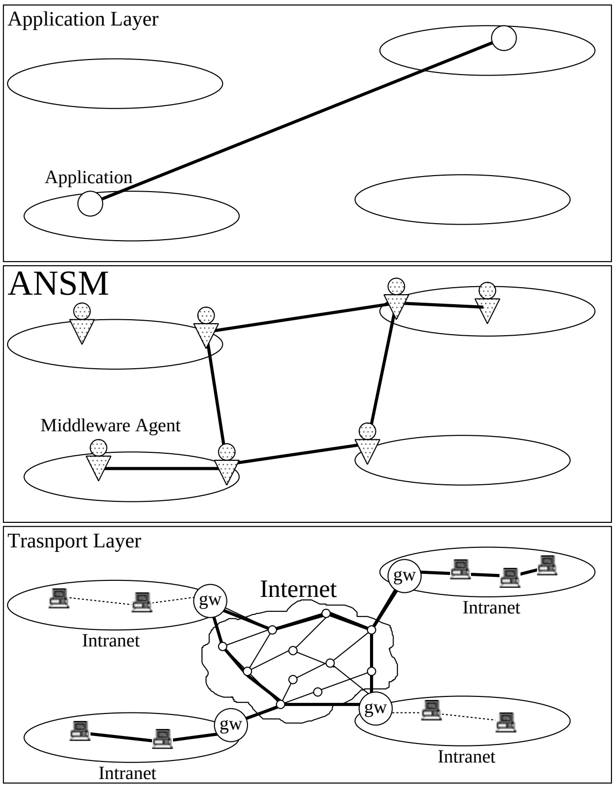


Fig.3 Connection model of Extranet in each Layer

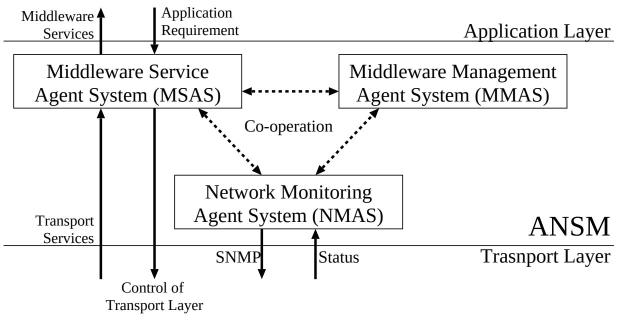


Fig.4 Structure of ANSM: ANSM consists of Middleware Service Agent System, Middleware Management Agent System and Network Monitoring Agent System.

Three types of agent system for realizing functions for extranet management are included in ANSM, as shown in Fig.4.

1. Middleware Service Agent System (MSAS)
2. Middleware Management Agent System (MMAS)
3. Network Management Agent System (NMAS)

2.2 Application Layer

The application layer is a layer in which many distributed applications exist in an extranet. Usually, the application layer provides a service to a user through the use of transport layer services [15]. However, services from the transport layer based on IPv4 are very fixed, thus services provided from the application layer to a user are also static services.

In this paper, ANSM provides the application layer with ANSM services, which are transport layer services controlled and customized according to the requirements of an application.

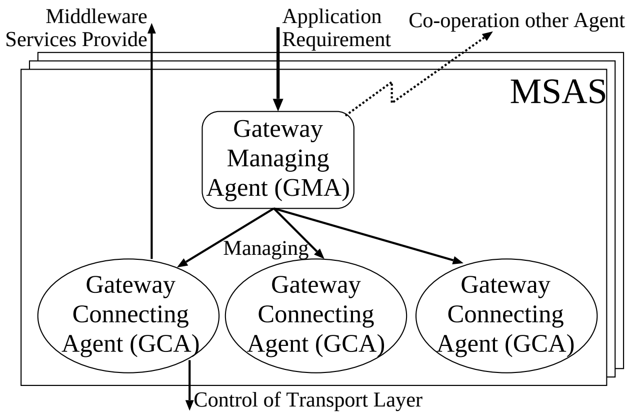
2.3 Transport Layer

The transport layer is a layer in which the connection process between each intranet logically exists in an extranet. Usually, the transport layer uses a simple protocol in order to ensure connection connectivity and network stability [15]. Therefore, it is difficult for the transport layer to respond to the diverse requirements of users and applications in the network.

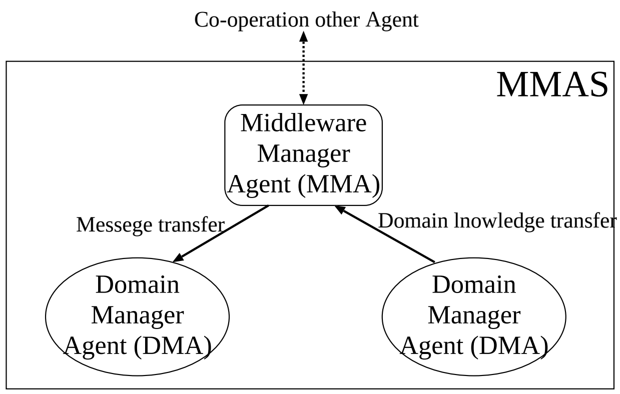
For example, an ATM network can obtain bandwidth requirement from users. However, the ATM network uses proprietary protocols and routers, thus it can be used only for a limited network. Of course, an Ethernet cannot gain possession of bandwidth.

Much research has been done to solve this problem. However, most of this research has aimed at keeping the stability of a network, thus an approach which controls a network according to the requirements of users or applications, as is the subject of this paper, has not been taken.

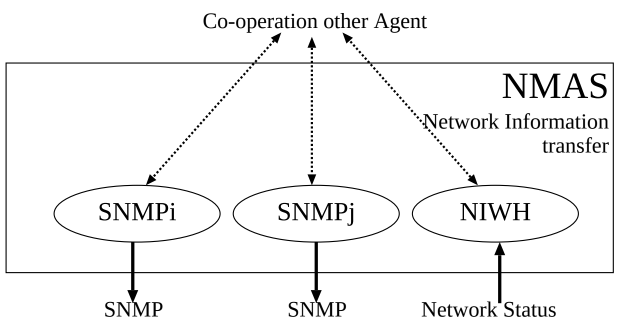
In this paper, we discuss how ANSM is controlled and customized to the transport layer by the requirements of the user or application, and how ANSM provides the application layer with ANSM services.



(a) Structure of Middleware Service Agent System



(b) Structure of Middleware Management Agent System



(c) Structure of Network Monitoring Agent System

Fig. 5 Structure of MSAS(a) , MMAS(b) and NMAS(c).

3. Agent Oriented Design of ANSM

ANSM consists of three agent systems: MSAS, MMAS, and NMAS (Fig.4). ANSM controls the transport layer with cooperation between these agent systems, and provides service to the application layer. These agent systems possess various types of knowledge about management of the extranet. Knowledge in ANSM is divided into two types; one is knowledge utilized in agent communication, and the other is knowledge of network management. The latter includes specific knowledge about network configuration, methods of gateway setting, methods of network management, and procedures for creation of connections among applications. This knowledge changes with different intranets.

3.1 Model of Middleware Service Agent System (MSAS)

This system exists on each gateway. MSAS obtains the requirements of an application from the application layer, and provides sufficient service to the application layer with control of the transport layer.

MSAS includes two types of agent: the gateway managing agent (GMA), and the gateway connecting agent (GCA) (Fig.5(a)). The former agent works as a gateway manager while the latter makes the actual connection between gateways. One GMA exists for each gateway. A GMA cooperates with another GMA according to application requirements, and they search one or more paths in the transport layer. If the search is successful, each GMA produces a GCA on its gateway. Finally, two GCAs make a connection in the transport layer as a result of cooperation. This connection in the ANSM layer is called the “ANSM path” hereafter.

Agents in MSAS have a management policy and gateway control procedures and knowledge as follows:

1. Connection property in gateway: This includes application requirement, available bandwidth in the transport layer, user ID, etc. Some GMAs cooperate with each other according to this knowledge when ANSM paths are searched.
2. Knowledge of control of the transport layer: Knowledge in making a transport layer connection
3. Procedure for providing ANSM path to the application layer: Procedural knowledge in providing ANSM path to the application layer
4. Procedure for searching for ANSM path: Procedural knowledge in searching ANSM path
5. Procedure for searching for multiple ANSM path: Procedural knowledge in searching multiple ANSM paths
6. Procedure of reconstruction of connection: Procedural knowledge in reconstruction of all ANSM paths
7. Procedure of release of ANSM path: Knowledge in releasing ANSM path

Details of agent behavior are described in the appendix.

3.2 Model of Middleware Management Agent System (MMAS)

MMAS provides knowledge to MSAS, and obtains some knowledge from NMAS mentioned below. Knowledge in MMAS includes knowledge regarding the overall extranet as well as that for a specific intranet.

MMAS includes two types of agent, namely middleware manager agent (MMA), and domain manager agent (DMA) (Fig.5(b)). The former analyzes a query from MSAS or NMAS, obtains knowledge from DMA, and return message to query. The latter keeps specific knowledge of the extranet.

Knowledge included in MMAS is as follows:

1. Information on the structure of the extranet
2. Information of current connections in overall extranet: This includes application requirements, bandwidth used, and intranet ID of each connection. This information is used when properties of the overall extranet are needed such as in procedure of reconstruction.
3. Knowledge of reconstruction management This includes priority level of the intranet connection and the application connection.
4. Knowledge on releasing a connection

3.3 Model of Network Monitoring Agent System (NMAS)

This system observes the current network QoS parameters, and processes them as knowledge. Knowledge in NMAS is provided to MMAS. NMAS uses agents for observation of the network, such as SNMP and NIWH (Network Information Warehouse)[16] (Fig.5(c)).

3.4 Actual Procedures in ANSM

We describe the actual experimental procedures as follows.

3.4.1 Procedure for providing ANSM path to application layer

This procedure is used in providing the ANSM path to user and application. Agents in ANSM cooperate with each other according to requirements. This procedure is used in providing ANSM path to user and application. Agents in ANSM cooperate with each other according to requirement.

[step1] A gateway managing agent (GCAi) obtains the requirement from the application layer. GCA1 cooperates with other GCAs, and the bandwidth needed to satisfy requirement is determined.

[step2] An ANSM path that satisfies the bandwidth decided in setup1 is searched for (Procedure for searching for ANSM path).

[step3] If an ANSM path is found, GMA1 orders a gateway connecting agent (GCA)to make a connection.

[step4] If step3 fails, a procedure for searching for multiple ANSM paths, and a procedure of reconstruction of connection are carried out.

[step5] If all procedures fail, a failure message is send to the application layer.

End of procedure

3.4.2 Procedure for searching for ANSM path

This procedure is used to search for an ANSM path that satisfies an application requirement.

[step1] The requirement of ANSM path from the gateway managing agent in the source gateway (GMAs) to that in the destination gateway (GMAd) is obtained by GMAs. GMAs orders neighboring GMAs to analyze the ANSM path to GMAd. ANSM paths from source to destination are selected from information regarding the structure of the extranet.

[step2] When the neighboring gateway managing agent (GMAn) receives the message from GMAs, GMAn requests information on the connection between GMAn and GMAd from NMAS. If n = d search finishes, an end message is sent to GMAs. If n 6= j, GCAn sends a message to the next gateway.

[step3] When GMAs receives an end of search message, GMAs evaluates the located ANSM path.

[step4] If the evaluation is successful, GMAs orders GCAs to make connections.

End of procedure

3.4.3 Procedure for searching for multiple ANSM paths

This procedure is used to search for a number of ANSM paths that satisfy the application requirement. If the procedure for searching for an ANSM path fails, this procedure is carried out. If this procedure fails, the procedure of reconstruction of connections is carried out.

[step1] If requirement (rq) cannot be satisfied with one ANSM path, GMAs reduces the requirement. The reduced requirement (rq') is processed using the procedure for searching for an ANSM path mentioned above.

[step2] If an ANSM path, which can satisfy rq', is found, GMAs acquires this path. If rq ¡rq0 < 0, the procedure finishes. On the other hand, if rq ¡rq0 > 0, GMAs begins to search for another path that satisfies rq ¡ rq0. This procedure is repeated until rq ¡ rq0 < 0. If no path is found, the procedure returns to step1.

[step3] When a number of ANSM paths satisfying the requirement are found, GCAs are generated to make all connections.

End of procedure

3.4.4 Procedure of reconstruction of connections

This procedure reconstructs current connections. When either a single ANSM path or multiple paths cannot satisfy the requirement, ANSM tries to make a connection by reconstructing the current connection.

[step1] When a requirement of reconstruction is sent from GMAs to MMA in MMAS, MMA analyzes the possibility of reconstruction of the connection using information of connection properties in the extranet, and reconstruction management knowledge.

[step2] If reconstruction can be achieved, GMAs sends a message of reconstruction to the intended GMAi.

[step3] When GMAi receives the message from GMAs, GMAi orders gateway connection agents to reconstruct the connection.

[step4] When reconstruction is finished, MMA sends an end of procedure message to GMAs. GMAs generate a number of GCAs and realize the new connections.

End of procedure

3.4.5 Procedure of release of connections

This procedure is carried out when an application is finished.

[step1] GMA sends a message to MMA that indicates the end of the connection.

[step2] MMA checks the connection using information of connection properties in the extranet, and releases management knowledge.

[step3] MMA evaluates the existence of the GMAw that is waiting for reconstruction. If GMAw exists, MMA permits the GMAw to process the restructuring.

End of procedure

4. Experiment

An experimental setup was constructed to evaluate the connection service of ANSM for applications among intranets. ANSM were used in an ADIPS framework (Agent-based Distributed Information Processing System framework)[17], which enabled development of an agent-oriented distributed system. ADIPS was developed with Java, and agents in ADIPS framework consisted of three modules as follows:

1. Agent program module: This module was designed to realize actual processing of agent knowledge. It consisted of an agent program and its interpreter. The agent program was described using ADIPS/R programming language. ADIPS/R is a language based on a production model [18]. In this module, condition rules regarding messages and events were described in the conditional part, while associated procedure rules were written in the action part.

2. Communication module: This module enabled communication among agents. In this module, control of transmission/reception and packeting/de-packeting of messages was carried out [19].

3. Java programming module: This module supplied an interface of actions for agents to a computer platform. Actions to the platform were realized with Java only, or with native functions of a platform wrapped by Java. In this paper, control and observation functions to the transport layer were implemented with both methods.

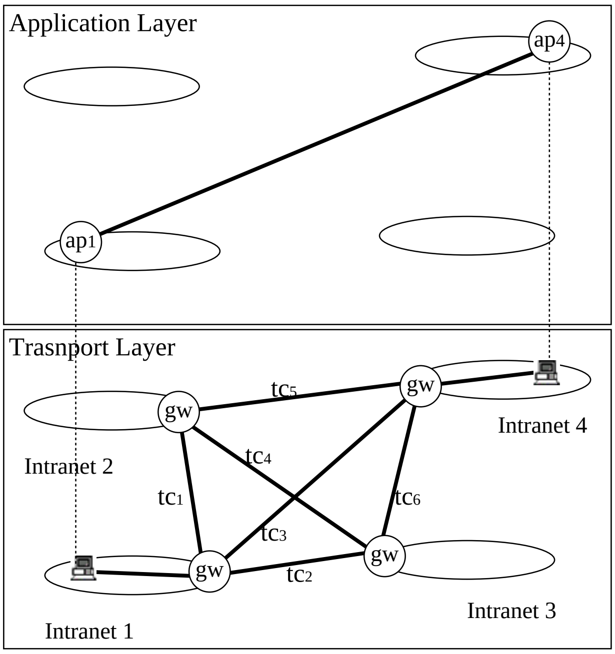


Fig.6 Structure of transport layer and application layer of experiment extranet

4.1 Experimental Setup

The structure of the extranet for evaluation of ANSM is shown in Fig.6. Four intranets (intraneti, i=1,2,3,4) were connected to each other. Therefore six logical connections in the transport layer existed, namely:

- tc1: connection between intranet1 and intranet2
- tc2: connection between intranet1 and intranet3
- tc3: connection between intranet1 and intranet4
- tc4: connection between intranet2 and intranet3
- tc5: connection between intranet2 and intranet4
- tc6: connection between intranet3 and intranet4

Application bandwidth was treated as a QoS factor, and the maximum bandwidth in each connection was 10Mbps. In the tested condition, applications in intranet1 (ap1) accessed a multimedia database in intranet4 (ap4).

MSASs, in which one GMA and a number of GCAs were included, were installed in each gateway. MMAS and NMSA were prepared in intranet1. Experiments were carried out in six situations, namely:

S1) Available bandwidth of all TCP/IP connections was 10Mbps, and requirement from ap1 was 10Mbps.

S2) Available bandwidth of all TCP/IP connections was 5Mbps, and requirement from ap1 was 10Mbps.

S3) Available bandwidth was 10Mbps in tc1, tc4, and tc6, and 5Mbps in tc2, tc3, and tc5, requirement from ap1 was 10Mbps.

S4) Available bandwidth of all TCP/IP connections was 10Mbps, and requirement from ap1 was 30 Mbps.

S5) Available bandwidth used by ap1 (10Mbps) in S1 changed to 5Mbps.

S6) Requirement of ap1 in S1 (10Mbps) changed to 30Mbps.

S1 - S4 were set for evaluation of realization of connection required from application, and S5 - S6 were for evaluation of dynamic adaptation of network conditions and requirements (Table. 1).

Table.1 Set up condition in each situation

Situation	tc_1	tc_2	tc_3	tc_4	tc_5	tc_6	user requirement
S1	10	10	10	10	10	10	10
S2	5	5	5	5	5	5	10
S3	10	5	5	10	5	10	10
S4	10	10	10	10	10	10	30
S5	$10 \to 5$	$10 \to 5$	$10 \to 5$	$10 \to 5$	$10 \to 5$	$10 \to 5$	10
S6	10	10	10	10	10	10	$10 \to 30$

[Mbps]

4.2 Experimental Knowledge

Two types of knowledge were installed into ANSM. One was a simple knowledge (K–A), in which only one fixed connection between intranet1 to intranet4 was used for service. This knowledge was similar to conventional static routing.

On the other hand in another knowledge type (K–B), the empirical knowledge of one professional administrator was modeled. The following tasks had been needed for him to satisfy user requirements (Table.II).

(K1) Attempt to satisfy requirement with single path

(K2) Attempt to satisfy requirement with multiple paths

(K3) Attempt to satisfy requirement with reconstruction of all connections in extranet

(K4) Reconstruction of connections when available bandwidth of connection has been changed

(K5) Reconstruction of connections when user requirements have been changed.

In his procedure, K1, K2, and K3 were tried in a sequential order to satisfy requirement when initial connections were determined. In contrast, when properties of connections were changed, he used K4 or K5 in his procedure. K1 - K5 were installed into ANSM as knowledge regarding the procedures mentioned in previous sections.

Table.2 Installed knowledge type K-A and K-B

Knowledge	K1	K2	K3	K4	K5
K–A	○	×	×	×	×
K–B	○	○	○	○	○

5. Results and Discussion

We discuss the experimental results from the following points of view:

- Realization of user requirements.
- Cooperation among GMA agents.
- Robustness to changes of requirement and bandwidth.

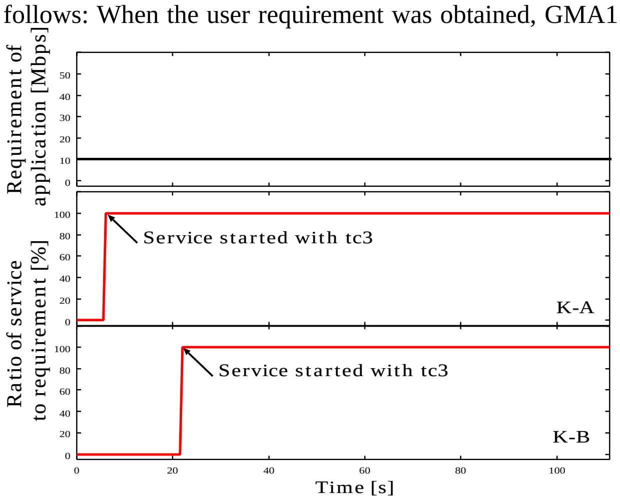
(S1) In this case, user requirement (10Mbps) could be satisfied with all connections (10Mbps). As shown in Fig.7(a), both K–A and K–B were able to satisfy user requirements. However, because agents needed to cooperate with each other in K–B, the time required for K–B to establish a connection was longer than for K–A.

Although all paths from intranet1 to intranet4, such as tc1–tc5, tc2–tc6, tc1–tc4–tc6, etc, satisfied user requirements, only tc3 was selected as a result in K–B. The reason is as follows: When the user requirement was obtained, GMA1 in intranet1 cooperated with GMA2, GMA3, and GMA4. GMA4 had a direct path tc3, therefore GMA4 could answer to GMA1 rapidly. However GMA2 and GMA3 needed to search for an available path to intranet4 by cooperation with other GMAs. Therefore, the answer of GMA2 and GMA3 were later than that of GMA4.

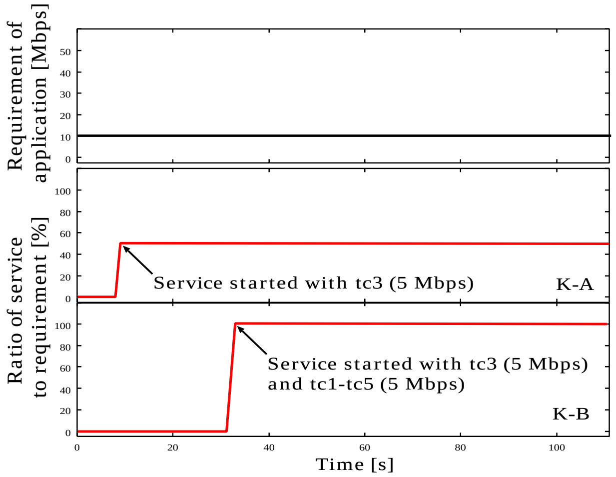
(S2) In this case, available bandwidths (5Mbps) were less than the requirement (10Mbps). Therefore, K–A could not satisfy the user requirement. On the other hand in K–B, the procedure of multiple paths was carried out in ANSM, and the user requirement was satisfied as shown in Fig.7(b).

Selected combination paths in K–B were (tc3 (5Mbps) and tc1–tc5 (5Mbps)) or (tc3 (5Mbps) and tc2–tc6 (5Mbps)). As mentioned in (S1), tc3 was determined first, and 5Mbps was retained. The residual 5Mbps was obtained by tc1–tc5 or tc2–tc6. Because of network topological symmetry, response time from GMA2 to GMA1 and that from GMA3 to GMA1 was almost the same. As a result, The residual 5Mbps was different depending on which one of the two paths was chosen.

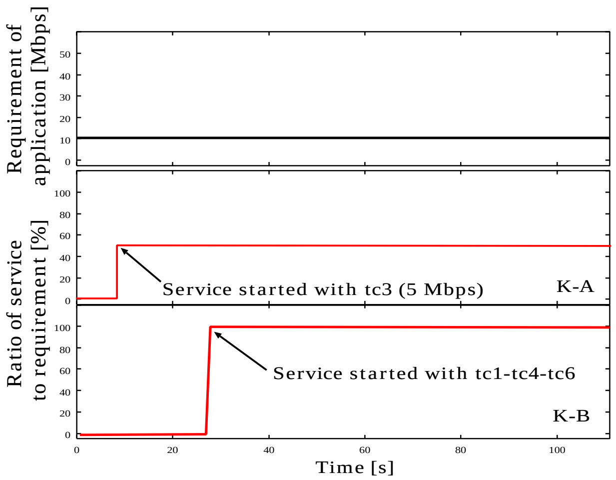
(S3) Available bandwidth of tc3 was less than that of the requirement. The result of K–A could not satisfy the



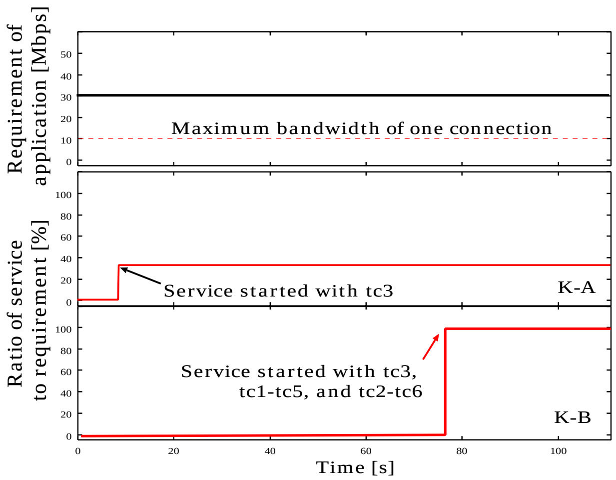
(a) Situation1: Available bandwidth of all TCP/IP connections were 10Mbps, and requirement from ap1 was 10Mbps.



(b) Situation2: Available bandwidth of all TCP/IP connections were 5Mbps, and requirement from ap1 was 10Mbps.



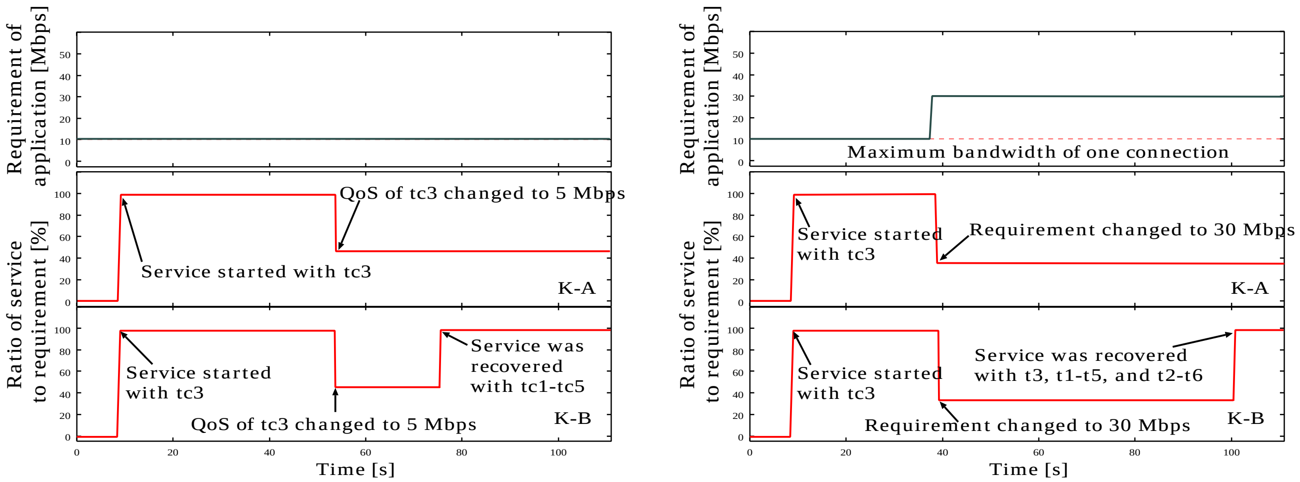
(c) Situation3: Available bandwidth were 10Mbps in tc1, tc4, and tc6, and 5Mbps in tc2, tc3, and tc5, requirement from ap1 was 10Mbps.



(d) Situation4: Available bandwidth of all TCP/IP connections were 10Mbps, and requirement from ap1 was 30 Mbps.

Fig. 7. Ratio of service to requirement of Situation1(a), Situation2(b), Situation3(c) and Situation4(d).

requirement as shown in Fig.7(c). In this case, only one combination path tc1–tc4–tc6 could satisfy the requirement. The result of K–B selected this bypassed path adequately, and the requirement was satisfied.



(a) Situation5: Available bandwidth used by ap1 (10Mbps) in S1 changed to 5Mbps.

(b) Situation6: Requirement of ap1 in S1 (10Mbps) changed to 30Mbps.

Fig. 8. Ratio of service to requirement of Situation5(a) which is changed network bandwidth, and Situation6(b) which is changed user requirement.

(S4) The required bandwidth (30Mbps) was 3 times larger than that of maximum bandwidth (10Mbps) in this case. K–A was totally unable to satisfy the requirement. Three paths were selected each time in K–B, namely: tc3 (10Mbps), tc1–tc5 (10Mbps), and tc2–tc6 (10Mbps). A total 30Mbps was obtained and the requirement was satisfied as a result (shown in Fig.7(d)).

In this case, tc1–tc4–tc6 and tc2–tc4–tc5 also had 10 Mbps bandwidth, but they were not selected at all. Because cooperation among agents in these paths was more complicated that in other paths, the response time of those two paths was longer than that of other combination paths.

(S5) In this case, network properties changed dynamically. When the available bandwidth of tc3 changed from 10Mbps to 5Mbps, sufficient service could not be provided with both K–A and K-B, as shown in Fig.8(a). However, the procedure of reconstruction of network connection was carried out in K–B, and a new connection was selected, namely: tc1–tc5 or tc2–tc6. Because of the long time required for cooperation mentioned above, tc1–tc4–tc6 and tc2–tc4–tc5 were not selected. As a result, the available bandwidth of the application was recovered in K–B.

(S6) In this case, a change of requirement exceeding maximum bandwidth occurred. K–A was totally unable to satisfy the requirement. The procedure of reconstruction of the network connection was carried out in K–B, and sufficient service was recovered. Three paths were selected, namely: tc3, tc1–tc5, tc2–tc6 (shown in Fig.8(b)). Because cooperation in reconstruction was more complicated than with (S5), the time to recover service was longer than was the case with (S5).

From the results mentioned above, we can see that if professional empirical knowledge is modeled and described properly, ANSM can select adequate paths of application automatically. Furthermore ANSM can adapt connections dynamically according to changes in network properties and requirements. The workload of the administrator in dealing with conventional manual settings can be carried out automatically by ANSM using high order knowledge such as K–B.

5. Conclusion

To achieve efficient management of an extranet, we have proposed a novel agent-based middleware. The proposed middleware controls network connections based on knowledge about the extranet. By using the proposed middleware, well-customized network connections are provided for extranet users and applications without an increase in the administrator's workload.

We also implemented the proposed middleware and performed several simulations. These results confirmed that the proposed middleware controls the routing and bandwidth assignments in an extranet at the requirements of users and applications, and this shows the effectiveness of the proposed method. We plan to look at several potential improvements to our approach:

- Mapping to network QoS from user requirements
- Reducing middleware overheads

References

[1] Kirk Hoffman, Pat Pallansch, and Steve Shafer, "Estabelishing a medical knowledge extranet to enhance medical communications," *Drug Information Journal*, vol. 34, pp. 1063–1068, 2000.

[2] Jim Barksdale, "The next step: Extranets," Netscape Communications Corporation: Netscape Columns: The Main Thing, 12 1996.

[3] Nahum Goldmann, "Extranet: The 'third wave' in the internet electronic commerce," Journal of Internet Banking and Commerce, 1 1997.

[4] HyoungJun Kim, KyoungCheol Koo, GiPyeung Song, and KiShik Park, "Design and implementation of an extranet based web system dedicated to the on-line standards development process," in *International* Conference on Computer Communications and Networks, 1998, pp. 624–629.

[5] Frederick J. Riggins and Hyeun-Suk Rhee, "Developing the learning network using extranets," in *Thirty-First* Annual Hawaii International Conference on System Sciences, 1998, pp. 339–349.

[6] J.Wroclawski, "The use of rsvp with ietf integrated services," IETF Request for Comments 2210, 1997.

[7] S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang, and W.Weiss, "An architecture for differentiated service," IETF Request for Comments 2475, 1998.

[8] Johnny Chen, Peter Druschel, and Devika Subramanian, "A simple, practical distributed multi-path routing algorithm," Rice Computer Science Technical Report TR98-320, 1998.

[9] Norio Shiratori, Kenji Sugawara, Tetsuo Kinoshita, and Goutam Chakraborty, "Flexible networks: Basic concepts and architecture," *IEICE Trans. Commun.*, vol. E77-B, no. 11, pp. 1287–1294, 1994.

[10] Norio Shiratori, Takuo Suganuma, Shigeki Sugiura, Goutam Chakraborty, Kenji Sugawara, Tetsuo Kinoshita, and E.S.Lee., "Framework of a flexible computer communication network," *Computer Communications*, vol. 19, pp. 1268–1275, 1996.

[11] Takuo Suganuma, Tetsuo Kinoshita, and Norio Shiratori, "Flexible network layer in dynamic networking architecture," in International Workshop on Flexible Network and Cooperative Distributed Agents (FNCDA2000), 2000, pp. 473–478.

[12] Susumu Konno, Gen Kitagata, Takuo Suganuma, Tetsuo Kinoshita, Kenji Sugawara, and Norio Shiratori, "Dynamic networking: Architecture and prototype systems," in *International Conference on Computational* Intelligence and Multimedia Applications (ICCIMA2001), 2001, pp. 93–97.

[13] P. Vil, J. Marzo, R. Fabregat, and D. Harle, "A multi-agent approach to dynamic virtual path management in atm networks," *IMPACT'99, Seattle (USA)*, 1999.

[14] Minoru Nakazawa, Takashi Suda, Noruyiki Abe, and Shimmi Hattori, "Autonomous network qos control by the user adaptive agent and its evaluation," *Electronics and Communication in Japan, Part.1*, vol. 86, no. 4, 2000.

[15] Jonathan B. Postel, "Internet official protocol standards," IETF Request for Comments 1780, 1982.

[16] Takeo Saito, Glenn Mansfield, Tetsuo Kinoshita, and Norio Shiratori, "Privacy consideration in network information warehouse," in *Internet Conference 1999*, 1999, p. 143.

[17] Shigeru Fujita, Hideki Hara, Kenji Sugawara, Tetsuo Kinoshita, and Norio Shiratori, "Agent-based design model of adaptive distributed systems," *Applied Intelligence*, vol. 9, no. 1, pp. 57 – 70, July/August 1998.

[18] D. C. Brown and B. Chandrasekaran, Design Problem Solving: Knowledge Structures and Control Strategies, Morgan Kaufmann Publishers, 1989.

[19] Tim Finin, Jay Waber, Gio Wiederhold, Michael Genesereth, Richard Fritzson, James McGuire, Stuart Shapiro, and Chris Beck, DRAFT Specification of the KQML Agent-Communication Language, 1993.

Susumu Konno is a senior assistant professor of Graduate School of Information Sciences, Tohoku University, Japan. He received his Ph.D degree in information engineering form Chiba Institute of Technology in 2001. His research interests include flexible networks, agent-based computing, knowledge engineering and symbiotic computing. He is a member of IEEE, AAAI, IEICE and IPSJ.

Kenji Sugawara is a professor of Department of Information and Network Science, Chiba Institute of Technology, Japan. He received his Dr.Eng. degree from Tohoku University in 1980. He received the Merit Prize from the Japanese Society for Engineering Education, the IEICE Yamashita Commemoration Prize , the IPSJ Paper Prize, and the IEICE Achievement Award in 1992, 1994, 1996 and 2000, respectively. He is a member of the IEEE, IEICE, IPSJ and JSAI.

Tetsuo Kinoshita is a professor of Information Synergy Center and Graduate School of Information Science, Tohoku University, Japan. He received his Dr.Eng. degree in information engineering from Tohoku University in 1993. His research interests include agent engineering, knowledge engineering, knowledge-based systems and knowledge-based systems. He received the IPSJ Research Award, the IPSJ Best Paper Award and the IEICE Achievement Award in 1989, 1997 and 2001, respectively. Dr. Kinoshita is a member of IEEE, ACM, AAAI, IEICE, IPSJ, JSAI, and Society for Cognitive Science of Japan.