A New Deployment Scheme of Expert Knowledge on Ubiquitous Service Provision for Non-expert Users

Shintaro Imai[†‡], Atsushi Takeda[§], Takuo Suganuma[‡¶] and Norio Shiratori[§¶]

[†]Research Fellow of the Japan Society for the Promotion of Science
[‡]Research Institute of Electrical Communication, Tohoku University
[§]Department of Computer Science, Tohoku Bunka Gakuen University
[¶]Graduate School of Information Sciences, Tohoku University

Abstract To provide high quality communication services to non-expert users including elderly people and children in ubiquitous information environments, it is necessary to provide expert knowledge about network and computers effectively to reduce the burden of the users. We have been working on the systematic support of circulation of the expert knowledge to resolve such kind of issue. In this paper, we propose a new Knowledge Circulation Framework (KCF) for ubiquitous service provisioning. This framework aims at realizing user-oriented and resource-aware services by deploying expert knowledge acquired in the digital space. We realize KCF by using the concept of knowledge-based multiagent system. Moreover, we applied KCF to deploy QoS control knowledge for ubiquitous videoconference system and performed several experiments. The experimental results show that the proposed system can provide better and more stable QoS, by improving the adaptability to the various types of ubiquitous computing environments, without any increase of user's burden.

1 Introduction

In ubiquitous information environment, service provisioning ability of systems is significantly limited due to the lack of computational power of small devices/sensors, and limitations in availability and stability of wireless networks. In this environment, it is difficult for non-expert users like elderly people and computer beginners, to control the ubiquitous services strategically according to node status and network condition in an appropriate manner. For these users, effective provisioning of many kinds of the expert knowledge concerning diverse services and intricately changing environment is a challenging research topic.

To resolve such kind of the issue, it is essential to effectively acquire, maintain, place and reuse the expert knowledge like operational heuristics through the network. To do this, we have proposed a Knowledge Circulation Framework (KCF) [1], a framework that enables deployment of the operational knowledge for effective service provisioning. In [1], we achieved an adequate QoS control by KCF in the situation where computational resources are greatly degraded. In this paper, we propose a design and implementation of KCF, targeting the ubiquitous information environments, where computational and network resources are extremely unstable and limited. To overcome the hurdle in service provisioning in such kind of environment, selection of the knowledge acquired in the operation on a similar environment is a core issue. In the proposed design, we develop KCF with a concept of knowledge-based multiagent system, and introduce a similarity metric to compare characteristics of different ubiquitous environments.

We applied KCF to the ubiquitous videoconference system (VCS), and evaluated it through experiments. We verified the proposed system from the users' viewpoints of QoS. As a result, frame-rate of the video increased from 6.2 fps to 10.7 fps, on average. On the other hand, variance of the fps decreased from 43.8 to 32.1. From the experimental results, we confirmed that the proposed system can provide better and more stable QoS, by improving the adaptability to the various environments.

2 Knowledge Circulation Framework

2.1 Related works

There are several previous works dedicated to adaptive service control in application level on best effort networks and systems [2, 3]. In addition, more advanced QoS control mechanisms specialized to the unstable network environments including wireless networks are considered in recent years. In these works, the service control mechanisms, in range from simple algorithms to more intelligent functions, are basically hard-corded in the application. Since the service control mechanisms are tightly coupled with the service elements, they can not be separated off and reused, in spite of their advanced control ability.

2.2 Knowledge-based service control

One of the possible solutions to realize more adaptive service control is to acquire and reuse operational knowledge of human operators/designers in the system to control the quality of the services. To improve upon the previous works described in the section 2.1, we have proposed Flexible Multimedia Communication Service (FMCS) with knowledge-based QoS control scheme [4]. In this scheme, QoS control

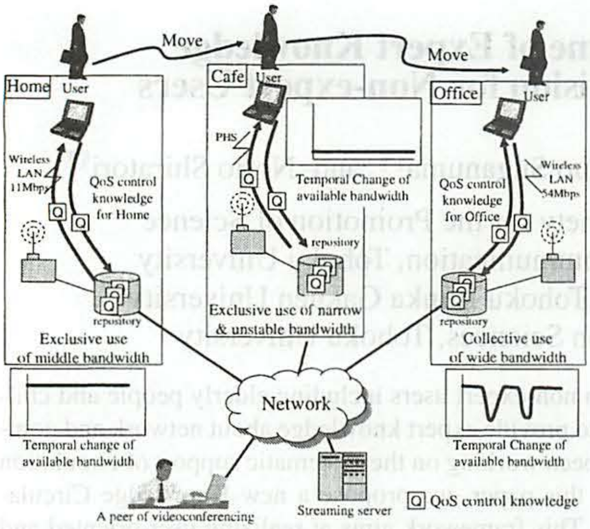


Fig. 1: QoS control knowledge circulated in ubiquitous information environments

knowledge is described in rule-type representation and is separated from the service elements. In addition, knowledge repository is newly introduced in this scheme. Using FMCS, designers of QoS control knowledge describe the knowledge and register it in the repository. Then the knowledge is instantiated and reused in run-time to control the QoS. Compare to the previous works, the range of QoS control ability was improved by FMCS.

2.3 Concept of KCF

Considering the problems described above, in this paper, we propose and develop Knowledge Circulation Framework (KCF) which enables dynamic circulation of knowledge in ubiquitous information environments. Fig.1 shows the basic idea of the proposed framework.

In KCF, the QoS control knowledge is circulated in the network via repository. Using this framework, the knowledge, created during operation, such as sequence of changes on parameters against the typical fluctuation of network resources, can be stored and reused in other similar situations. There are mainly two distinguished ideas in KCF as follows:

(1) Knowledge acquisition through operation: In related works described in previous sections, useful operational history and heuristics on service control, created during operation time, are basically put away. Using KCF, such heuristics are actively acquired and tailored for reuse.

(2) Reuse of acquired knowledge: In related works, service control mechanisms are statically embedded in application. While in FMCS, service control knowledge can be easily replaced, thus adaptability of the system is refined. However, there is no way to circulate the knowledge automatically. Using KCF, the acquired heuristics are effectively reused in many situations.

```
(rule warning_service_down
  (record :record1 ?record1 :record2 ?record2 :record3 ?record3
          :record4 ?record4 :record5 ?record5)
  (threshold :threshold ?thre)
    (> ?record1 ?thre)
    (> ?record2 ?thre)
    (> ?record3 ?thre)
    (> ?record4 ?thre)
    (> ?record5 ?thre)
  (count :count ?cou) = ?cha
    (> ?cou 10)
  (name :vcm ?vcm)
  -->
  (send :performative warning :to ?vcm
        :content (request_action :action service_down))
  (modify ?cha:count 0)
)
```

Fig. 2: Example of knowledge representation in an agent

Employing this framework, service control ability would be greatly improved without any hard maintenances of knowledge repository against the drastic change of user requirement and environment, such as movement of the user among different environments.

3 Agent-based Design of KCF

3.1 Overview of KCF

KCF consists of three phases of circulation, i.e., knowledge acquisition, management, and placement phases.

Knowledge acquisition: In this phase, system acquires service control knowledge dynamically when some distinctive phenomenon occurs during service provisioning.

Knowledge management: In this phase, the system performs accumulation and arrangement of the acquired knowledge based on the characteristics and the reusability of it.

Knowledge placement: In this phase, the system decides which and where service control knowledge should be places to provide service effectively, when the service is required under some conditions.

3.2 Design of agent architecture for KCF

In order to circulate knowledge efficiently, KCF employs mobile agent technology to carry the knowledge. The agents play a role of knowledge carrier on the network. Moreover, we introduce agent repository as a warehouse and dispatcher of the knowledge.

We applied an agent architecture in [5] in KCF. The agent holds the knowledge in form of "Fact" and "Rule". Fig.2 shows an example of the rule type knowledge which a resource-monitoring agent has.

A rule is represented in the following form:

```
(rule Rule-name
   Condition-part (If-part)
    --> Action-part (Then-part))
```

When a conditional part is TRUE, then actions described in the action part are performed. Knowledge shown in Fig.2 is to inform a warning, when the agent

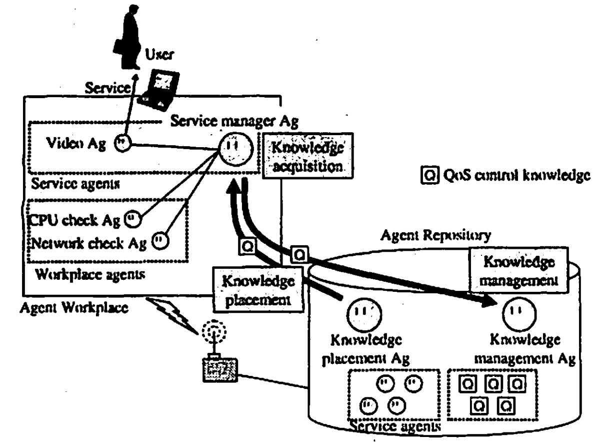


Fig. 3: Agent organization design for KCF

detects degradation of computer resource.

3.3 Agent Organization Design of KCF

Agent organization design of KCF is shown in Fig.3.

KCF is constructed on multiagent framework DASH [6]. DASH is a framework that constructed by the agent repository and the agent workplace. The agent repository is a server to store the agents, and the agent workplace is an execution environment of agent on each workstation (WS). The agent workplace offers basic functions like message delivery and name resolution etc. These functions are necessary so that the agent group may operate as a multiagent system.

Here we give details of agents in KCF.

(1)Service agents:
These agents manage and control the services. In case of multimedia communication service, services are video, audio and white-board etc. They are stored in agent repository, and when the service is required, they move, are placed at an agent workplace, and offer service.

(2)Workplace agents:
These agents reside in agent workplace permanently and monitor computer resource status, network resource status and user requirements etc.

(3)Knowledge management agent:
This agent arranges the knowledge which is brought back to agent repository by the Service agents and the Workplace agents. It resides in agent repository permanently. It classifies and accumulates the knowledge based on its characteristics and usability, etc. Moreover, this agent gives the knowledge to the Service agents group for the services based on knowledge placement agent's requirement.

(4)Knowledge placement agent:
This agent performs knowledge placement in agent repository. When it receives the service requirement, it divides the requirement, selects the suitable agent from the Service agents in agent repository, and requests the extraction of knowledge for knowledge management agent.

Knowledge	:=	<S, Condition, CK, C>
S	:=	<UNS_{QoS}, STA_{QoS}, UNS_{res}>
UNS_{QoS}	:=	Degree of user's QoS requirement unsatisfaction
STA_{QoS}	:=	Degree of QoS is stability
UNS_{res}	:=	Degree of user's resource requirement unsatisfaction
Condition	:=	<Hardware, R_{ave}, R_{SD}, UR>
Hardware	:=	<Type, P>
Type	:=	Type of hardware
P	:=	Performance of hardware
R_{ave}	:=	Average of resource usage
R_{SD}	:=	Standard deviation of resource usage
UR	:=	User Requirement
CK	:=	Control Knowledge used for QoS adjustment
C	:=	Count of the knowledge used

Fig. 4: Structure of accumulated knowledge

3.4 Agent Behavior Design of KCF

In this section we illustrate behavior of agents according to the phases described in section 3.1. Here, we circulate the following three types of knowledge: knowledge about the environment, history of usage of QoS control knowledge, and heuristics on effects of the applied QoS control knowledge.

(1) Knowledge acquisition phase:
(1-a) Knowledge acquisition during service provisioning: Workplace agents acquire four kinds of information: they are, type and status of the resource, average of resource usage rate, standard deviation of the resource usage rate, and user requirement.
(1-b) Feed back to agent repository: When the service provisioning ends, Service manager agent, that is one of the Service agents, derives the user requirement achievement level S from the above four kinds of acquired information by the following expressions:

$$S = UNS_{QoS} + STA_{QoS} + UNS_{res}$$

Here, each element is described as follows:

UNS_{QoS}: Degree of user's QoS requirement unsatisfaction

STA_{QoS}: Degree of QoS is stability

UNS_{res}: Degree of user's resource requirement unsatisfaction

As a result, above three kinds of knowledge is acquired for circulation. Moreover, Service manager agent sends the knowledge acquired by Service/Workplace agents to Knowledge management agent.

(2) Knowledge management phase:
Knowledge management agent classifies and accumulates knowledge is fed back by Service manager agent. This phase includes the following operations:
(2-a) Knowledge association: Knowledge management agent associates the knowledge about user requirement achievement level S and the knowledge about environment with the used QoS control knowledge. Fig.4 represents the structure of the accumulated knowledge.
(2-b) Knowledge integration: Knowledge management agent compares the knowledge that is fed back

with existing knowledge about *condition* and CK. When $Hardware$ and CK match, and R_{ave}, R_{SD} and UR are similar in some measure, Knowledge management agent integrates these knowledge.

(3) Knowledge placement phase:

(3-a) Selection of Service agents: When a service is required, the Knowledge placement agent determines Service agents to be placed to the Agent Workplace using contract-net protocol [7].

(3-b) Knowledge hand-over: When Service agents to be placed to Agent Workplace are determined, the Knowledge management agent compares the knowledge about environment that has been accumulated, with the information about environment where the new service will start, in a similar way to (2-b) in Knowledge management phase. Knowledge management agent selects the knowledge that has the highest S and that meets the following conditions: (i) $Hardware$ matches, (ii) R_{ave}, R_{SD} and UR are similar in some measure. Knowledge management agent gives the CK to the Service agents.

(3-c) Service initiation: Knowledge is provided to the Service agents and are placed at an Agent Workplace, start to offer the service, and the process returns to (1-a) of Knowledge acquisition phase.

4 Experiments

4.1 Implementation

We implemented KCF based on the design described in section 3.4 using DASH framework [6], a multiagent framework. We developed seven agents: Knowledge management agent, Knowledge placement agent, VideoConference Manager (VCM) agent, Video agent, Netcheck agent, NetcheckVCS agent and UserVCS agent. Each agent has a computational base process (BP) to realize the function described in section 3.4. As for the video transmission/receiving software, we employed Java Media Framework (JMF), and embedded it as a BP in the Video agent.

4.2 Experiment environment and conditions

The configuration of experiment is as shown in Fig.5. This configuration assumes ubiquitous information environment. We installed the agents based on KCF on WS-{A, B} whose specifications are shown in the figure. User-{A, B} receive services on WS-{A, B}. WS-A is connected with 100 Mbps Ethernet. WS-B is connected with 11 Mbps wireless LAN. In this experiment, we assume the general environment that is configured by average spec computers and networks with resource limitations, such as home or outside location. We also assume that videoconferencing is performed with other applications at the same time in such kind of environment. Under these conditions, VCS has the limitation concerning the resource usage rate for the user to use other applications. WS-{C, D} add load to a wireless LAN by the data communication based on the traffic pattern that is actually observed in real networks. By this intended load, resource availability limitation becomes not achieved, and then the agents control QoS to improve this undesirable situation. In these experiments, VCS had acquired 16 kinds of knowledge by preliminary videoconference operation based on KCF. And we configured user requirement is 20 fps. Here, we discuss the effectiveness of KCF by observing temporal changes of QoS and resource availability of VCS, comparing the VCS with KCF to the VCS with existing scheme (FMCS [4]).

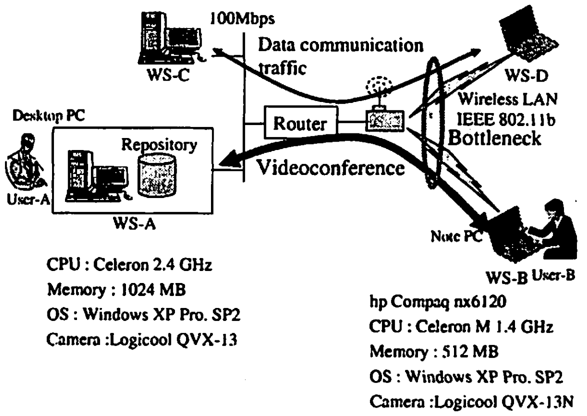


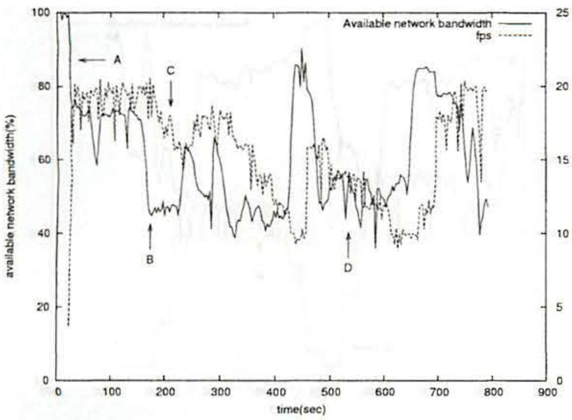
Fig. 5: Experimental environment

4.3 Experimental results

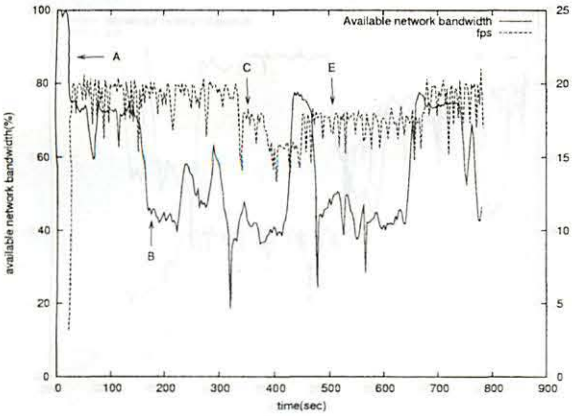
First, we set user's resource requirement to 50%. This means that the user requests to leave 50% of the total available bandwidth. We added two network load patterns to VCS with KCF and with existing scheme. Results can be found in Fig.6 and Fig.7

After the experiments started, we initiated videoconference and added the network load at point A based on the load pattern actually observed. Then, the available network bandwidth decreased and network resource limitation was not achieved (point B). Here, VCS corresponded by adjustment of the QoS (point C). VCS with the existing scheme, in Fig.6(a) and Fig.7(a), frequently adjusted QoS, directly according to the change of the network resource (point D). As a result, QoS became unstable. On the other hand, VCS with KCF in Fig.6(b) and Fig.7(b), controlled QoS that did not adjust frequently reacting to the change of network resource situation by using the knowledge that is accumulated beforehand (point E). As a result, KCF was able to avoid the frequent QoS fluctuation, and provided the service with stable quality.

Next, we increased the user's resource requirement to 75%. This means that the user requests to leave 75% of the total available bandwidth, thus this is more difficult requirement than the case of 50%. We gave the same two network load patterns as the previous experiments. The results are shown in Fig.8 and Fig.9.

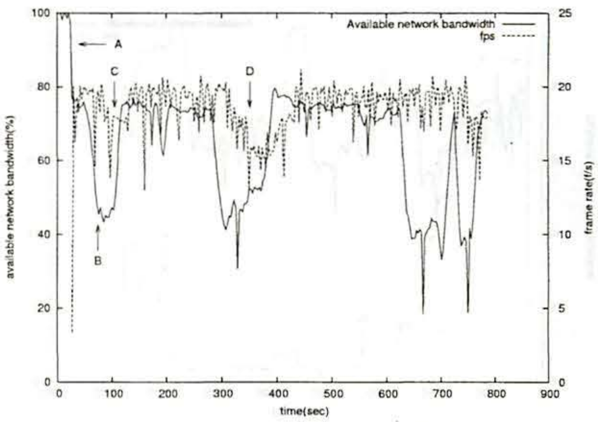


(a) VCS with existing scheme (FMCS [4])

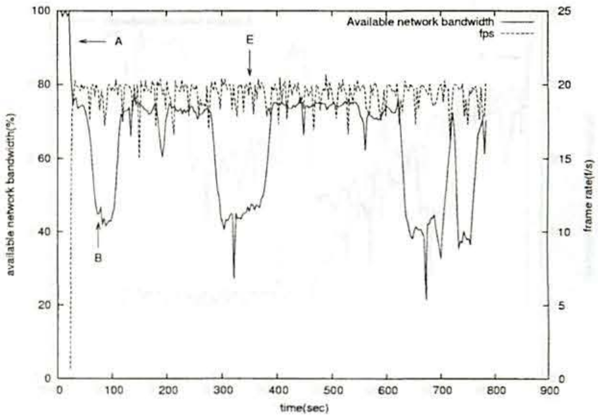


(b) VCS with proposed scheme (KCF)

Fig. 6: QoS control results of VCS against a network traffic pattern (pattern-1) in case that available bandwidth requirement is 50%



(a) VCS with existing scheme (FMCS [4])



(b) VCS with proposed scheme (KCF)

Fig. 7: QoS control results of VCS against a network traffic pattern (pattern-2) in case that available bandwidth requirement is 50%

After the videoconference started, we added the network load (point F). Then, the available network bandwidth decreased and user's resource requirement was not fulfilled. Here, VCS without KCF corresponded by decreasing QoS to the minimum as shown in Fig.8(a) and Fig.9(a) (point G). After that, when VCS observed that the load of the network became light, it recovered the QoS (point H). On the other hand, VCS with KCF in Fig.8(b) and Fig.9(b), did not reduce QoS to the minimum. In addition, VCS controlled QoS did not adjust frequently, reacting to the quick change of network resource situation (point I). Actually, when we added a network traffic pattern-1, the frame-rate of the video increased from 6.2 fps to 10.7 fps, on average. On the other hand, variance of the fps decreased from 43.8 to 32.1. This behavior is realized by using the knowledge that is accumulated beforehand. Finally, KCF was able to avoid the frequent QoS fluctuation, and provided service with stable quality.

4.4 Evaluation

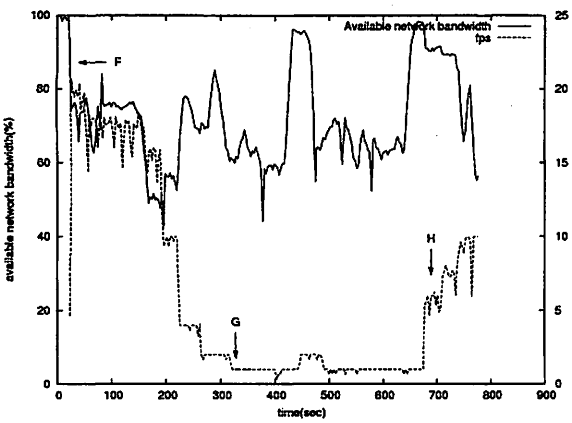
From the experimental results of section 4.3, we compare operation of VCS with existing scheme and VCS with KCF.

When user's resource requirement was 50%, as shown in Fig.6 and Fig.7, (b) with KCF was able to improve the QoS and its stability compare to (a) without KCF. This is because VCS with KCF was able to control the QoS effectively, because it has knowledge about the environment. This knowledge was selected by the highest user requirement achievement level S in prior operation. In this case, the QoS control knowledge that suppresses the reactive control is selected, considering the quickly changing status of the available bandwidth on the target environment. This reduced the excessive control of the QoS, and led the improvement of the stability of QoS. From this experiment, we found that the adaptability of VCS to ubiquitous information environments would be improved by applying the KCF.

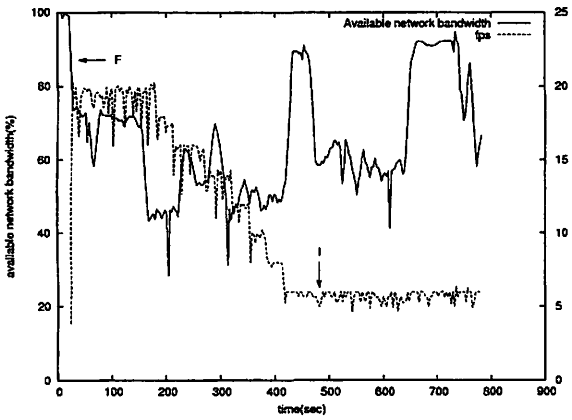
When the user's resource requirement was set to 75%, as shown in Fig.8 and Fig.9, the effect of KCF is larger than the case of user's resource requirement was 50%. Therefore, the improvement of adaptability is more effective in the severe environment.

5 Conclusion

In this paper, we proposed a Knowledge Circulation Framework (KCF) that enables deployment of

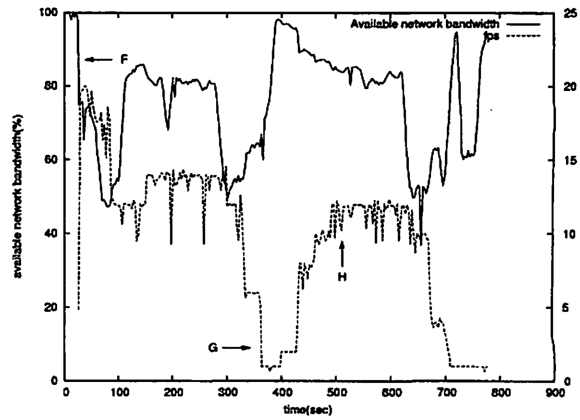


(a) VCS with existing scheme (FMCS [4])

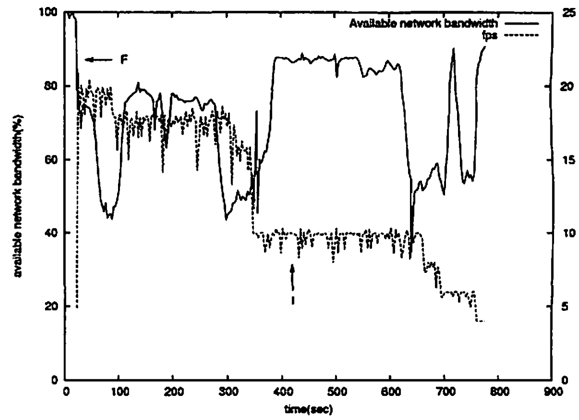


(b) VCS with proposed scheme (KCF)

Fig. 8: QoS control results of VCS against a network traffic pattern (pattern-1) in case that available bandwidth requirement is 75%



(a) VCS with existing scheme (FMCS [4])



(b) VCS with proposed scheme (KCF)

Fig. 9: QoS control results of VCS against a network traffic pattern (pattern-2) in case that available bandwidth requirement is 75%

knowledge on the ubiquitous information environments. This framework supports circulation of the knowledge in the network, by a sequence of acquisition, management, and placement of the knowledge to reuse it effectively. We proposed a design of KCF with knowledge-based multiagent system, and performed experiments by applying the KCF to VCS. From the results of experiments, we concluded that the adaptability of VCS to ubiquitous information environments has improved by applying the proposed framework. In our future work, we will evaluate the KCF by conducting experiment in various actual situations.

Acknowledgement

This work has been supported by the Grant-in-Aid for JSPS Fellows No.18-5190 of Japan Society for the Promotion of Science (JSPS).

References

[1] S. Imai, T. Suganuma and N. Shiratori, "Knowledge Circulation Framework for Flexible Multimedia Communication Services," *IEICE Trans. Inf. & Syst.*, Vol. E88-D, No. 9, pp. 2059-2066, Sep. 2005.

[2] T. Turletti and C. Huitema, "Videoconferencing on the Internet," *IEEE/ACM Trans. on Networking*, vol.4, pp.340-351, Mar. 1996.

[3] J. Bolliger, and T. Gross, "A framework-based approach to the development of network-aware applications," *IEEE Trans. Software Eng.*, vol.24, pp. 376-390, May. 1998.

[4] T. Suganuma, S. Imai, T. Kinoshita, K. Sugawara and N. Shiratori, "A Flexible Videoconference System Based on Multiagent Framework," *IEEE Trans. on Systems, Man, Cybernetics, partA*, vol. 33, no.5, pp. 633-641, Sep. 2003.

[5] S. Fujita, H. Hara, K. Sugawara, T. Kinoshita and N. Shiratori, "Agent-based design model of adaptive distributed systems," *Applied Intelligence*, vol.9, pp. 57-70, Jul/Aug, 1998.

[6] DASH -Distributed Agent System based on Hybrid Architecture- Webpage, http://www.agent-town.com/dash/index.html

[7] R. G. Smith, "The Contract net protocol: High-level communication and control in a *distributed* Problem solver," *IEEE Trans. Comput.*, vol. 29, no. 12, pp. 1104-1113, 1980.