Streaming Cloud Service Concept by Peer-to-Peer Distributed Technology

Takeshi Tsuchiya
Faculty of Business Administration and Information
Tokyo University of Science, Suwa
Nagano, Japan
t-tsuchi@rs.suwa.tus.ac.jp

Hirokazu Yoshinaga, and Keiichi Koayanagi
Faculty of Science and Engineering
Waseda University
Kitakyushu, Japan
{yoshinaga@akane, keiichi.koyanagi}@waseda.jp

***Abstract*—In this paper, we discuss management manner for streaming services provided on complex network environment, and propose streaming management platform for realizing Streaming Cloud Services, which doesn't depend peer environments. These streaming service-based applications would be expected to diffuse more greatly as the next generational services, and have been used for security surveillance, vehicle-to-vehicle communication, and route guidance service. Therefore, it is required to achieve our proposal platform that overcome several difficulties of streaming systems, network environment, and functions for stable services continuously by distributed computing technology. We focus on the generating of the streaming session instead of each peers allocated environment, and the issues for continuous the service such unstable environment. The experimental results showing the streaming service with best effort using several hundred sessions are described in this paper.**

Keywords- Streaming Service; Peer-to-Peer; Cloud Computing.

I. INTRODUCTION

Demands of streaming services collaborated by distributed camera and devices allocating in ubiquitous services have been increased as for the applications such as security surveillance, vehicle-to-vehicle communication, and route guidance service [1]. Streaming data generated from cameras and devices are also used to improve service accuracy and robustness, and attachment devices are also utilized as for service accuracy such as a GPS (Global Positioning System) and acceleration sensor.

But the condition (status) of network changes dynamically on overlay networks and streaming data, although it is expected to use them whenever. And it is not ensured the quality and conditions of network to use specific streaming applications as the codec and control by the each application because the environment of peers would be complicated by location of connections. Therefore, we discuss the manner, which realizes streaming communications in these complicated environments, and propose the streaming management platform adapted several streaming and network environments using P2P (Peer-to-Peer) overlay networks [4]. The P2P overlay networks is to construct logical space based on peers, and allocate services by each peers on that space [5]. These are provided on the overlay network are divided to two types by their characteristics, synchronous and asynchronous type services. In the synchronous type services, the streaming service, which means the interaction service by sharing same time, such as the Skype [6], is very used for communications. Face tracking and recognition systems have been developed to use CV (Computer Vision) technologies with real time video streams. The asynchronous service is added based on the file synchronous one. However, the communication band for real time sharing should be required between distributed peers. In addition, the peer fluctuates in the overlay networks because of the move of a user from the network, and its network and system environments. Therefore, our goal is that a user does not need to use a specific application, and can use services except the knowledge of the network.

Java Spaces [7] has been proposed for the framework of distributed components. It manages connections and allocations; the unstable network environment such as large distribution of component is not provided in this framework. In Open Call Media Platform [8] proposed by HP Inc., it is possible to make sessions among nodes utilizing different streaming service. A static server is set as for codec transformation service in the border of closed streaming network. In this case, some problems should be solved to accept several codec and stream control methods, and scalability for the fixed a server caused by the concentration of traffic. This proposal model only solves managed and planed services like telephone networks.

In our considerations, streaming service can be used among distributed peers on overlay networks, and named it streaming cloud service. It enables to generate streaming sessions in best-effort network route on that time.

In the following part of this paper, we propose streaming platforms including adaptation manner on the route of session, one is the CODECs of streaming data and another is streaming control are adapted to peer environment. At the same time, our proposal provides priority control of streams by applying the session condition for unstable environment. As the result, it is available to make best-effort route for

ISBN: 978-1-61208-105-2

streaming services, and evaluate that functions and service scalability as for streaming platforms.

This paper is written as follows. The modeling of the proposed streaming platform in the Section II, and the proposal and implementation method are written in the Section III. The evaluations are described in the Section IV. Finally, the paper is concluded in the Section V.

II. FUNCTIONS OF STREAMING PLATFORM

In this section, we clarify the definitions and assumptions of service environments in this paper.

A. Service Environments

Streaming services in this paper are defined as follows: streaming services provide following functions either or both, generating the sessions and sending streaming data. Currently, the streaming services have own stream control methods and use of codec in the stream data described above. Therefore, controlling streams for starting communication among distributed objects, each system need to use same control methods and same data CODEC for service inspective each peer environments. Since peers provide most of the streaming services on the overlay networks, these services have the possibility to causes instability and loss of the services.

B. Functions

Some functions are needed in the network between different peers, as follows.

1) Detection of Services

Peers in overlay networks by normal users, and the time pass must change its number of services, status, and performance dynamically. The information of these peers is notified to other peers in the process of joining and separation to the overlay network.

2) Retrieval of Services

The retrieval service would be important functions, and used in overlay networks by query indicating network ID from detected service, and service type of peers. However, the reliability of overlay network is not so high caused by each peer without servers. Therefore, all information shared among peer is updated periodically, and normal peers without fixed peer provide all services.

3) Allocation of Components in Streaming Session

The components and allocation of route on streaming session is decided by the function on platform when the peer generates streaming communications. The “trans-code” (Trans-code means transferring function of CODECs such as from Mpeg 2 to DV) service is selected suitably from overlay network to transfer data codec, and it is allocated on session between peers.

Relay peer that relays all communications on overlay networks is allocated for through NAT, and firewalls on the session between peers because of user’s environments. A CODEC is transferred to another through trans-code service, and it transfers the data continuously while the session existing. Functions for controlling the streams are independent from CODEC transferring, and defined “basic control interface” such as “request” and “stop” common methods. Therefore, it happens asynchronously in streaming session. The orders of controlling method make status changing. Relay peer also provide to adapt environment for transport protocol.

C. Stable Session Provision

Created session has issues of buffer overflow and the debasement of its quality. The size of buffer and the quality are adjusted to recover one.

III. PLATFORM FOR IMPLEMENTATIONS

In this section, we propose streaming platform that include efficient functions demanded previous sections. This proposal platform is allocated as for middleware between user and each streaming application shown in Figure 1. The role and provided functions of each layer shown in Figure 1 is as follows.

A. Pipe Management Layer

This layer manages all communications and connections to other platforms implemented other peers. In these communications, there are not only but collaborative data, all communication data included management data of overlay network, query for retrieval service, administration information of shared information and so on utilize this layer.

This layer only manages all communications, therefore, peer and service information notified to overlay network are controlled in upper layers such as retrieval service, service selection and so on. The update of peer information is distributed and managed periodically, and statistic information is also performed derived from peer by the demands. In the retrieval service, the query for retrievals and its result are sent and received in the layer by the demands. In the streaming services, its data are treated between platforms as shown in Figure 2. Each session is identified as an added pipe with 16 bit ID, and is managed by a table based on the ID.

1) Session Environment Support

The transport protocols (UDP, TCP, Http over TCP) [9] which each streaming service adapt in peer environment are chosen and use to create streaming session for best effort, while user generate the session by streams. The transport protocols are corresponding to each peer environment based on received peer and service information, and transport protocols are decided by demand from the result of analysis in upper layer.

2) Stable Streaming Service

The sending and receiving data in each session is kept continuously while the session exists. It needs to handle plural streams depending on the service condition at the same time. Some peers are assumed to provide trans-code services that are able to transform CODECs, and to generate new sessions. Each session is controlled to keep service quality, and it is available to stably treat plural sessions. The peer separation can be notified as the problem occurs because RTP and RTCP [10] are utilized between streaming platforms.

 ISBN: 978-1-61208-105-2

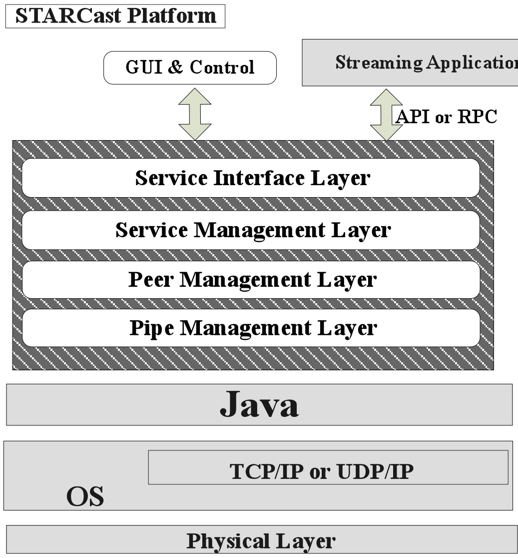

Figure 1. Platform Stack

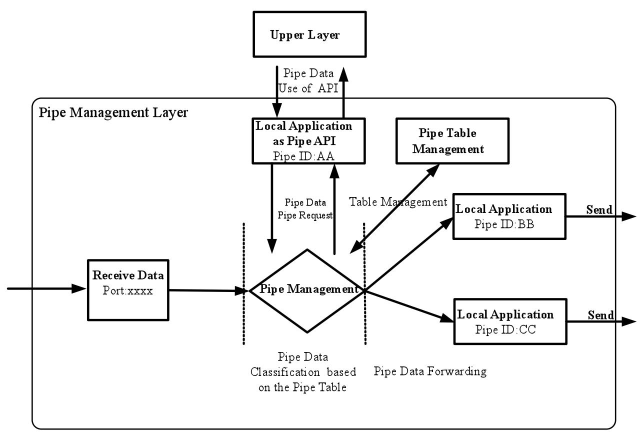

Figure 2. Pipe Processing

B. Peer Management Layer

This layer manages taken peer information, service information, and management information for generating session.

1) Distributed Information Management

Peer and service information notified and shared to overlay to network are written as following elements. A 128-bit ID (IDentification) is used for identification of each peer, and the ID and unique name are for user one. Peer and service information are identified with this ID and name in the upper layer of platform.

2) Service Retrieval

Our proposal platform provides function of retrieval service for peer and service information notified to overlay networks. Writing detail, the information for codec of streaming data and stream control protocol in each streaming service, peer ID and so on is used as query for retrievals by the platforms. In this function, the retrieval manner is used our proposal distributed information retrieval manner [5], which realizes almost same availability comparing current information retrieval services like google among distributed peers without any servers. Therefore, all users can retrieve information same as current normal services on overlay networks.

3) Adaptation for Streaming Control Environment

Transition of service status can be controlled by each stream control method such as session generation and disappearing, start and stop of stream sending and receiving. It is required to control stream that does not depend on the environments of session environment. Therefore, this layer adapts and absorbs various streaming control methods. We define common stream control method based on the conceptualistic transition of service status because specific stream control method would not unify any services in the near feature. The control command and a list of its order are shared in the control method. As for example, this control method corresponds to RTSP [11], SIP [12], and DVTS [13] in present implementation in TABLE 1.

Some system except a stream control function exists. The system can be control from outside, and some scripts are prepared to solve this problem.

TABLE 1. Mapping of Common Interfaces

Common Interface	RTSP	SIP	DVTS
Init	SETUP	BYE	etup.sh
Play-Snd	PLAY	INVITE	dvsnd
Play-Rcv	Rcv	INVITE	dvrcv
Stop	TEARDOWN	BYE	stop.sh

4) Collaboration with Applications

A streaming application is connected to the platform, and it can be controlled with the stream control method written in TABLE 1. Common Interface is defined Common Stream Control Method as for utilizing streaming services seamlessly, and implemented by Web Service Description Language (WSDL) shown in Figure 3. The defined method by mapping enables to change status for services.

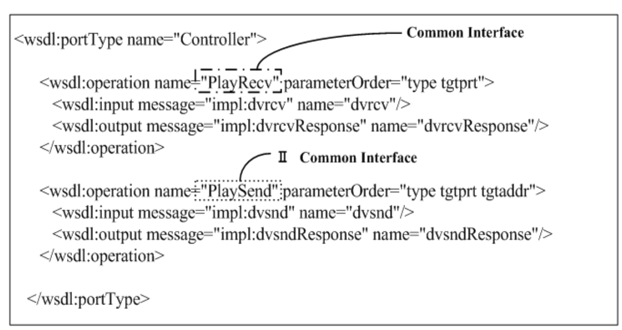

Figure 3. Implementation by WSDL

C. Service Management Layer

The services allocated in session are chosen from all peers who have same functions by peer selection manner, and the construction of the session is determined and generated by chosen members.

1) Generating the Session

In the case that peer environments are different for generation of session, codec transferring should be performed in the session, and allocates trans-code service for transferring stream data. The service for transferring is determined by the total load and codec demand on that service. We could assume that there are many services existing on overlay network, and a peer that creates session can provide codec transferring. The service is located on the session between constructor peers, and starts the transferring. If relay of peers is required in the network environment, peers who compose sessions are tandem located, and they have some effects to the service processing, quality of streaming service of peer.

2) Selection of Peer and Service

Selection of service, which chooses services provided, peers by their condition of status as best effort, and these chosen peers compose streaming session. The authors have proposed the adaptive coordinator election (ACE) platform [14] to select peer as best effort on overlay network.

First, the candidate service for session are chosen by their conditions, transferring data codec and transport protocol, and taken by retrieval service written in section B. Where, the algorithm is used in the Java Bench [15] for derivation of peer processing. After this process, the selections of services are executed and chosen the service that composes the session by our proposed algorism based on derivation result. After all, each user could create a streaming session by best efforts. The result of selection is listed, and shared this information as for back up of streaming session. However, status of listed peer would be change by time pass, and execute selection again. Therefore, the list of peer that the order is from creation can be updated to reconstruct session for best effort. The period of update makes load big in the network and the platform, and that is ten minutes in our system.

D. Service Management Interface Layer

This layer provides interface for GUI and control to a user, and the streaming application has the function of API and proxy in the platform.

1) API and Proxy Function

TABLE 2 shows an example of API (Application Programmable Interface) for streaming service developer and upper streaming services as functions of streaming platform. Each API can execute each function mentioned above. In the following, APIs are sample that adapt to basic functions for streaming sessions and managements. The proposed platform is allocated and affected as local proximal node for existing streaming services and applications, not only for news streaming service. Therefore, it effectively uses the existing resources. Therefore, it makes to use existing resources effectively.

TABLE 2. Examples of API

Module	Method (Object type)	Explanation
BSC	notify (Network ID)	Notifying join and separation
BSC	set, get (Peer ID)	peer information management
SRV	retrieval (Query)	retrieval the query
ARV	selection (Peer Array)	peer selection
CTL	sndCtrl (Cmd)	control request
CTL	getStatus (Peer Array)	session peer information

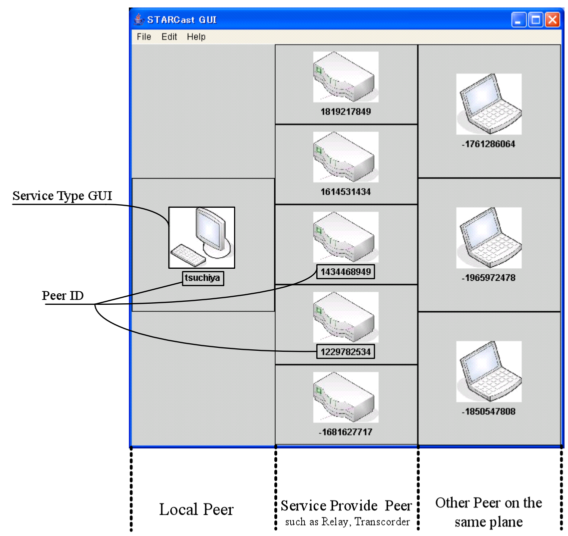

Figure 4. GUI for peers

Figure 4 shows GUI provided for users. Each user can detect peers and provided services on the overlay network. User can control, and generate streaming session.

E. Transcode Service

Trans-code service is one of functions provided by a proposal platform and a peer. These are allocated on the streaming session by the demands on peer environment that means data codec of stream and connecting network environment. It performs as a process, and the codec of streams would be transferred adaptively, while transferring streaming data on session.

In the current implementation, each peer allocates this function as a service. When the peer generates the streaming session, most suitable trans-code service are chosen adaptively from all same services provided overlay network by above mentioned manner.

 ISBN: 978-1-61208-105-2

IV. Evaluation

In this section, our proposal platforms are evaluated adaptation functions for each environment by the simulation and implementations.

A. Service Scalability

In this section, a service on overlay network by simulation is evaluated, and adaptability is also done with an implemented platform. The environment of simulation is Pentium 4 2.8 [GHz], RAM 2,048M [Bytes], Linux 2.4.28, and Java J2SE 1.5, and constructs streaming environments assumed distribution of several services on the same overlay network. Each peer in overlay networks composed overlay networks, and generates streaming session among these peers under their environment.

The "Network Link" in Figure 5 means link of the Internet connection, and the ratio of network link in each peer changes dynamically by the traffic under Poisson distribution (λ=1). In this time, the loss of data as the throughput of transit ranges in $0 \leq$ Link ≤ 30 [%]. The trans-cord services which transfer the CODEC of streaming data, are set in the point A or B, and execute the service for the sessions of peers.

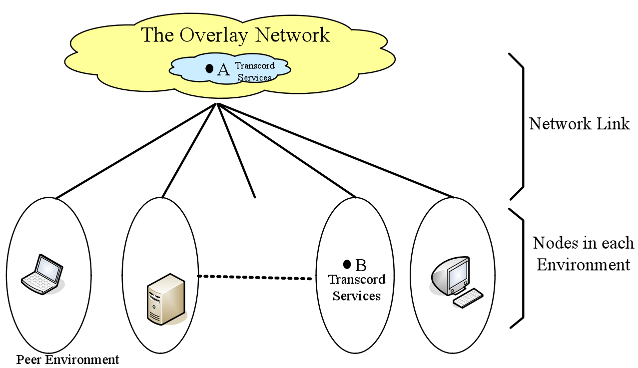

Figure 5. Environment of simulations

The result of simulation is shown in Figure 6. The x axis shows the number of streaming sessions among peers on the overlay network, and the y axis shows increment ratio of session time comparison to normal one. At this time, the trans-code services are disappearing randomly. Each lines in Figure 6 show the difference of number of services for trans-code and the selection manner. These lines are affected by other streaming sessions, and affect each other. "The line xx%" in Figure 6 mean the ratio of trans-code service for all streaming sessions. And, "Static Server" means a trans-code service provided by a static server on the point A. "Normal-50%" means the generated streaming sessions without above mentioned peer selection, and the peers who need the service use nearest service.

In the range of x-axis between 0 and 50, these lines are looks similar. But the "Static Server" line would be occurring increment rapidly caused by concentration of streaming traffic to server. But, in the case of changing the number of peers dynamically such as P2P overlay networks; there is problem in the viewpoint of the scalability of service. This system needs to provide load-balancing manner of this service.

The effect of peer selection manner enables to compare "Proposal-50%" with "Normal-50%" in the same number of sessions, and these lines have almost same characteristic in the range of less 100 sessions. In the more large range of x-axis, the increment of "Normal-50%" value grows large rapidly. This increment of line have caused by the selection of specific services for trans-coding with bias in the streaming session. Therefore, the concentration of traffic to specific services makes the line of increment ratio larger. Comparison of lines using our proposal ("Proposal-xx%"), these lines are ranges increment inversely proportional to the number of service allocation such as 50%, 25%, and 20%. However, all the transmission time is almost fixed by increasing the session.

Therefore, the streaming session for best effort is created from trans-code service distributed ion overlay networks in the proposed platform.

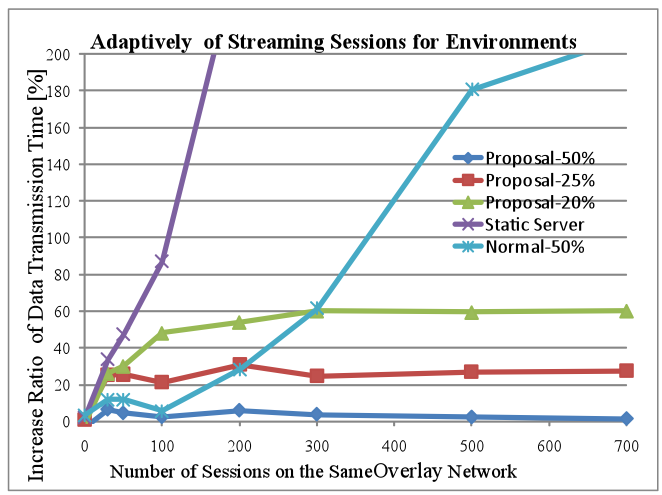

Figure 6. The number of session and the increase ratio of data transmission time

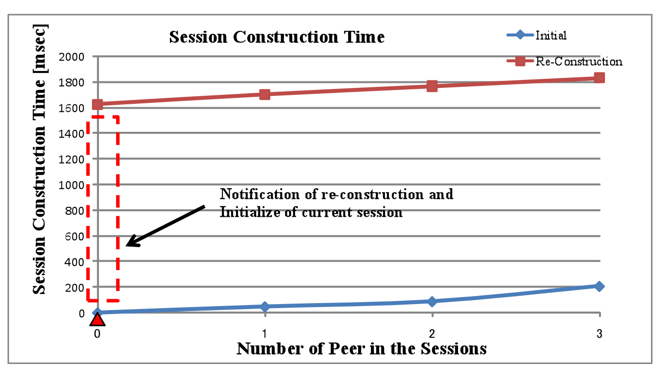

Figure 7. Construction and Reconstruction of sessions

 ISBN: 978-1-61208-105-2

Figure 7 indicates the time sequence to generate new streaming session between peers, and re-construction time from session broken. x and y are the hop number and the construction time in this Figure. *Initial* indicates time to notify session requirement to all peer for construction, and to create session based on the response from peers. *Re-Construction* is the time to stop a session in the time 0. The peer in the session detects the changing and disappearing the peer and trans-code service that reconstructs a session. The peer notifies the other peer constructing the session. In each receiving peer, a creating session is initialized, and the session is reconstructed. The time is 1,600 m[sec] from notifying session stop to reconstruction beginning in this graph.

V. Conclusion

We proposed streaming management platform for streaming cloud service on overlay network. The streaming service can be provided on unstable overlay networks without considering environments of each peer, network environment, and services. The stable service is shown using proposal platforms including some functions to avoid the delay and data loss between platforms.

However, scalability of service should be improved, and need to consider the manner for continuous service in an unstable session. In this time, the session should be re-generated to others when a user does not notice it. An idea is to use seamless service, and the changing is available that it does not need to stop the session for future works.

Acknowledgment

This work is partly supported by Grant-in-Aid for Research Activity Start-up (21800063).

References

[1] H. Sawano and M. Okada, "Next-generational Route Guidance Method by Using Multimodal Information Integration", Proc. of Information and Computer Elements, pp.273-278, Sept.2007

[2] A. Barton-Sweeney, D. Lymberopoulos and A. Sawides, "Sensor Localization and Camera Calibration in Distributed Camera Sensor Networks",Proc. of the 3rd Int'l Conf. on Broadband Communications, Networks and Systems, 2006, pp.1-10, Oct.2006

[3] R. Tron, R. Vidal and A. Terzis, "Sensor Localization and Camera Calibration in Distributed Camera Sensor Networks", Proc. of ACM/IEEE Distributed Smart Cameras, 2008, pp.1-10, Sept.2008

[4] E. Lua, J.Crowcroft and M.Pias, "A Survey and Comparison of Peer-to-peer Overlay Network Schemes", The Electronic Magazine of Original Peer-Reviewed Survey Articles, vol.7,pp.72-93, Mar.2006

[5] T. Tsuchiya, M. Lihan, H. Yoshinaga, and K. Koyanagi. "Distributed Information Retrieval Service for Ubiquitous Services" Proc. of Availability, Reliability and Security, pp. 842-850, Barcelona, March 2008

[6] S. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol", technical reports 2004, Columbia University

[7] Sun Microsystems Inc. JavaSpaces Service Specification, version 1.1, October 2000, http://www.sun.com/jini/specs/js11.pdf, [Retrieved; August 24, 2010]

[8] Y. Makino, Y. TAN, "Integrated Visual Communication Network Architecture", Proc. Of Towards Peta-Bit Ultra-Networks, Ishikawa Japan, pp. 156-161, September, 2003

[9] T. Tsuchiya, H. Yoshinaga,and K. Koyanagi "STARCast: Streaming Collaboration Architecture on Heterogeneous Environment Everywhere", Proc. Of ACM Multi Media, pp. 57-62, October 2004

[10] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", Request for Comments: 1889, January 1996

[11] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol ", RFC 2326, April 1998

[12] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: session initiation protocol", RFC 3261, IETF, June 2002.

[13] A. Ogawa, "DVTS:http://www.sfc.wide.ad.jp/DVTS/" [Retrieved; August 24, 2010]

[14] H. Yoshinaga, T. Tsuchiya, and K. Koyanagi, "Coordinator Election Using the Object Model in P2P Networks", Springer Berlin / Heidelberg, Volume 3601/2005, pp.161-172,

[15] Java Benchmark, http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/bench /Bench.html, [Retrieved; August 24, 2010]

[16] Fine Packet Builder,http://www.fineqt.com/, [Retrieved; August 24, 2010]