

Distributed Session Announcement Agents for Real-time Streaming Applications

Kazuhiro Mishima, Hitoshi Asaeda
Keio University, Graduate School of Media and Governance
Fujisawa, Kanagawa 252-8520, Japan
Email: {three, asaeda}@sfc.wide.ad.jp

Abstract—End-users who want to receive real-time streaming data on the Internet must first obtain the session information. A session information distribution system that distributes the available session information to all potential receivers must be scalable and work with low latency. We present the design and implementation of a session information distribution system using multiple session agents. This system distributes the session information to the Internet users effectively, and can limit information distribution to specified users by labeling. We evaluated the actual implementation on the PlanetLab and OpenDHT to measure the session information distribution delay and retrieval time. The experimental results have shown that the session agent can distribute session information to the entire Internet with good scalability and end-users can retrieve the information with low latency.

I. INTRODUCTION

Development of real-time media streaming systems has been giving users a great chance to receive audio and visual streaming content through the Internet. It enables users to conduct real-time applications, such as live events, conferences, and e-learning. Real-time media streaming improves the potential of Internet applications and will increase the future needs.

To access real-time media streaming contents, users must first obtain available or scheduled session information, including a content location, content type, and required application for playback. It is necessary to develop a mechanism that announces such session information to all potential receivers on the global Internet. From an operational perspective, a large number of media streaming contents will be available in the future Internet, and end-users' terminals, access methods, and access controls will be miscellaneous. For the future service-oriented Internet, therefore, the session announcement mechanism should be not only scalable but also adaptive to the communication environments.

Data-Oriented Network Architecture (DONA) [1] presented that the original Internet architecture was a purely transparent carrier of packets, but now the various network stakeholders use middleboxes to accelerate applications. The session announcement system for future media should take the factor into account. However, while DONA gives the good starting point for the corresponding discussion, the presented approach is still immature and far from completion in our opinion.

*This work was supported by National Institute of Information and Communications Technology (NICT), Japan.

The goal of our work is to propose a concrete architecture that can handle current and future needs, in particular when considering feasibility in terms of session information distribution, and can support a large number of real-time media session information in the Internet. We present design and implementation of a session information announcement system that consists of distributed agents to fulfill the demand. According to the system design, there is no contradiction to the dynamic nature of session information distribution scheme.

II. SESSION DISCOVERY OF MEDIA STREAMING

A. Session Information Description

Real-time media streaming sessions can be dynamically launched and terminated. When a data sender sends or plans to send media streaming session to the Internet, sender needs to announce the session information to the potential receivers, and potential receivers must obtain information to receive the content. Session information is mainly composed as “media locator” - a pair of source address and group address for multicast, or URL - indicating the source of media content, content type, and application. Proprietary information for each media streaming application can be indicated additionally. The session information is used for accessing the media streaming session by users who want to receive media streaming session.

The Session Description Protocol (SDP) [2] is the standard protocol that describes session information of media streaming session. SDP description includes “time” when media streaming continues, “address” from and to where media content is transmitted, “media type”, and other attributes that application uses. SDP has been widely used by Session Announcement Protocol (SAP) [3], Session Initiation Protocol (SIP) [4], and other media streaming applications (e.g. Real Player [5]).

Although SAP provides feasible information for any kind of media streaming application, there are various problems for session information announcement or discovery mechanisms (detailed in Section II-C). We need a mechanism that provides session information to potential receivers who can receive real-time media streaming sessions in the entire Internet.

B. Requirements

One of the requirements of a session information announcement or distribution mechanism is “scalability”. Even if a large number of streaming applications announce their

session information over the entire Internet simultaneously, the announcement mechanism should manage the information distribution in a scalable manner, and the potential receivers should be able to obtain session information smoothly.

“Latency” is also an important requirement for session announcement, since media streaming sessions may change frequently; if the announcement of session information to potential receivers requires amount of time, session itself may become obsolete or different from the expected one.

“Policy definition” is also an important factor. Stream senders often want to limit the target users of information distribution for their contents to keep their privacy. To not disclose the session information in the entire Internet, it is encouraged that the session announcement mechanism has the capability to limit announcement only to the users who can obtain secret or private session information. Policy definition is a necessary factor for adaptive communication in the future service-oriented Internet.

C. Related Work

Sdr [6] is a well-known session directory system that has been intensively used in IP multicast environment. Session Announcement Protocol (SAP) [3] cooperates to distribute available multicast group information to the directory system. In a SAP announcement procedure, entire session information must be periodically transmitted and all active information must be continuously refreshed. If session information is no longer announced, its description eventually times out and is deleted from the available session list. Although SAP improves protocol robustness and keeps all the session directory instances synchronized, periodic data transmission of all active session descriptions increases additional transmission overhead and latency, and further reduces scalability especially when the number of sessions increases. In the practical situation, the SAP has major limitations on these requirements [7], [8]. To verify its condition, we calculated the session announcement interval based on the definition in RFC2974 [3]. While the announcement interval for each session is randomly calculated, it is always more than 200 seconds, and especially when the number of active sessions goes over 500, the announcement interval increases accordingly as shown in Fig. 1.

Session Announcement Server (SAS) [9] was proposed for another session announcement method in multicast network. SAS can announce session information by using a hierarchical architecture which enables the announcement for both intra- and inter-domain environment. Authors described that extension of the multicast scope of session announcement enables global-wide scale session announcement. In fact, the hierarchical architecture that SAS’s information distribution method uses resembles part of our proposed session information distribution architecture detailed in the later section. However, SAS does not have any function to reflect data sender’s policy even though it is necessary in the multicast scoping concept. Performance evaluation is also needed to be concerned because the SAS operation was not evaluated by an actual implementation.

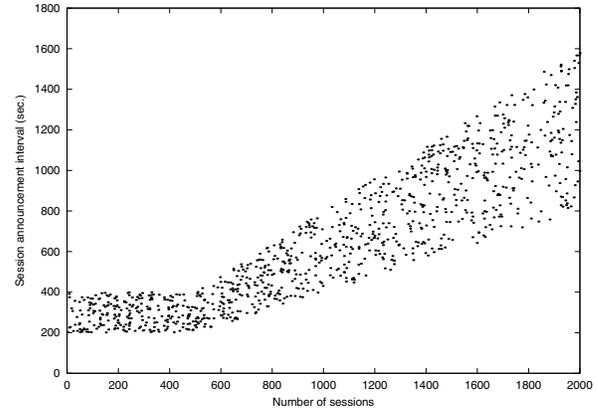


Fig. 1. Session announcement interval by SAP (bandwidth used for the announcement = 4 kbps, size of the SAP message = 300 bytes).

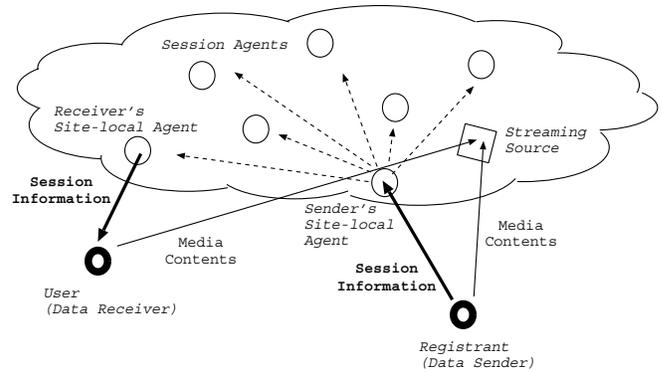


Fig. 2. System operation overview

As another approach, a robot-type web search engine [10] could be the candidate as it enables to find various information in the Internet by the crawling technique. Our concern is that although it is necessary to manage fully dynamic session information to a large number of users in the Internet for real-time media streaming, this system cannot manage sessions that are launched and stopped more frequently. In addition, since search engines rely on a single scope database, defining and applying sender’s policy or desired transmission users would be impossible whereas it contradicts the feasible requirements for session information management.

III. SESSION INFORMATION ANNOUNCEMENT SYSTEM

We propose a session information announcement system. Fig. 2 shows an overview of our session information announcement system.

This system consists of multiple agents (session agent, hereafter) globally distributed in the Internet. A data sender registers session information into his session agent located in sender’s site-local network. After completion of its registration, the session information is synchronized among agents by two possible systems; 1) Tree-based system and 2) DHT-based system using DHT (Distributed Hash Table) algorithm.

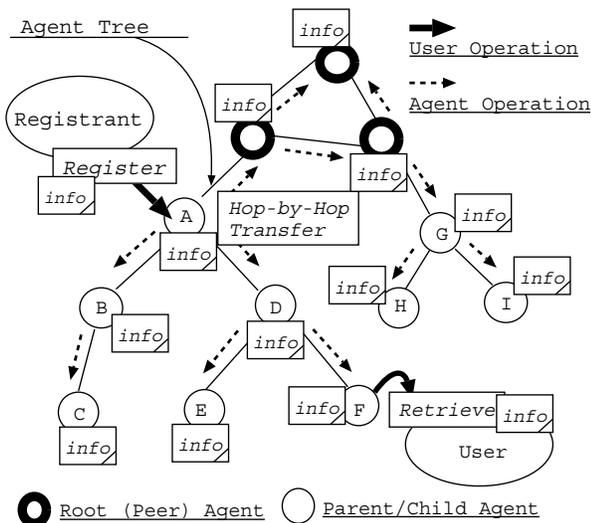


Fig. 3. Tree-based System operation

A data receiver retrieves session information from his session agent located in receiver's site-local network, and finally can receive the media streaming.

A. Tree-based System

1) *Tree Configuration*: Each site administrator prepares his session agent and cooperates with other session agents located in other sites to create "agent tree".

Basically, each interconnection on an agent tree indicates a "parent-and-child" relation. Each agent statically connects to one parent agent (either a root agent or another agent that is closer to a root agent) and either several child agents or no child at all (as a leaf). A root agent is logically a single node that is a root of an agent tree. Peering configuration with multiple root agents enables to attach multiple trees that are located on wide networks as shown in Fig. 3.

2) *Information Distribution in Agent Tree*: In the tree-based system, once a data sender registers the session information into his session agent, the agent transmits information to the agent tree reliably. After receiving session information, each agent in that group stores the information in own database and transmits to the neighbor agents (i.e. parent and/or child agents). The session information is transmitted by a hop-by-hop manner, and finally synchronized with all agents in the group. To announce information to the entire Internet, hop-by-hop transfer will make on all agent node one by one. The procedures of session information modification and cancellation are same as that of the session information registration.

When an agent in the group does not respond upon data transfer in matter of failure, session information distribution is stopped on the agent, and the information is not transmitted to every agents. In that case, to recover from the failure, a neighbor agent of the failure node records the failure information and retries transmission periodically.

B. DHT-based System

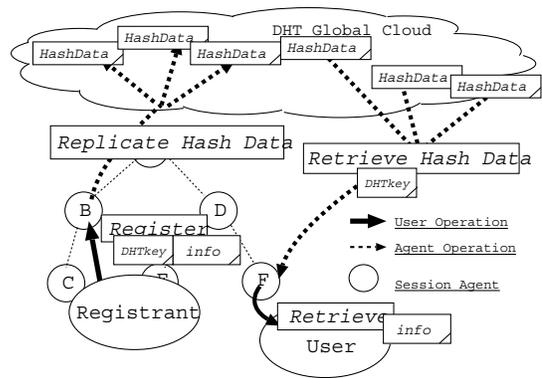


Fig. 4. DHT-based System operation

1) *Global Cloud Configuration*: The agent system in a DHT-based system uses "DHT global cloud" (Fig. 4) to distribute session information in the Internet. DHT global cloud is a logical data storage that maintains global-wide session information. A session agent can register session information with the global cloud and retrieve session information from the global cloud. Session information is replicated by the DHT algorithm into the global cloud robustly. An agent can become either a DHT node or a non-DHT node. When an agent is a non-DHT node, it must access one of DHT nodes (as its bootstrap node) to connect to the DHT global cloud. In this paper, we assume that the bootstrap node address is notified by the site administrators prior to its use.

2) *Information Distribution in Global Cloud*: In the DHT-based system, a session agent with which a data sender registers the session information makes three steps to distribute the session information; 1) the agent stores the information to the local database, 2) creates a hash data from the session information, and 3) replicates the hash data (as for DHT value) with session search keywords, like a media locator (as for DHT key) into a DHT global cloud. Fig.5 is shown an example of DHT key and value.

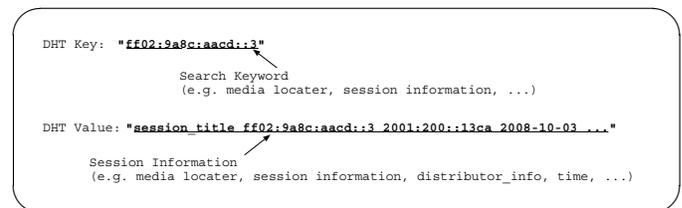


Fig. 5. DHT key and value in Global Cloud

To retrieve session information in the DHT-based system, user accesses his session agent with specifying session search keywords (as for DHT key). And then the agent accesses the DHT global cloud and gives the corresponding session information. Fig. 4 shows the procedures of session information distribution in the DHT-based system; 1) registrant registers session information on Agent.B, 2) Agent.B replicates the information to the global cloud, 3) Agent.F retrieves the

session information from the global cloud.

C. Policy-aware Information Distribution

In order to fulfill the “policy definition” requirement, we propose a method to limit session announcement only to legitimate users who have the corresponding pre-shared key (“Secret Share”). The method of the “Policy-aware Information Distribution” uses a secret share which has preliminarily shared among the session registrant and legitimate data receivers. In this paper, we assume that the corresponding secret share is preliminarily and securely given to the legitimate users with an external authentication system, and the session information is encrypted with an SSL (Secure Socket Layer).

In this section, we explain the use case with the DHT-based system. Upon session information registration to the DHT global cloud, if a registrant wants to limit the session announcement to legitimate users, he registers the session information with its secret share to avoid disclosing the information to non-legitimate receivers.

With the DHT-based system, a session search keyword with secret share and session information is created for secure session information as shown in Fig. 6. Therefore, only users who preliminarily obtain secret share can specify the search keyword for secret session information and retrieve secure session information.

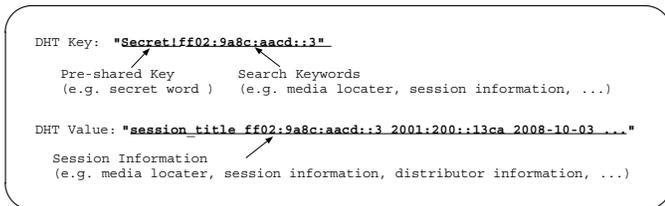


Fig. 6. Pre-shared key and Session information

Fig. 7 shows the procedures of policy-aware session information distribution; 1) registrant registers session information on *Agent.B* with a secret share, 2) *Agent.B* replicates the information to the global cloud, 3) since a user who accesses *Agent.F* has the secret share, *Agent.F* can transmit the session information from the global cloud to him. However, another user who accesses *Agent.G* does not know the secret share, *Agent.G* cannot transmit the session information to him.

IV. IMPLEMENTATION AND EVALUATION

We implemented both tree-based and DHT-based systems. To synchronize session information among session agents, the agent that the session information was initially registered extracts the session information stored in its local database in an XML format and transmits the information to other agents by SOAP messages over HTTP/SSL. We used gSOAP [11] for the development. As the component of DHT global cloud, Bamboo-DHT [12] based on the Pastry algorithm [13] is used in our experimental environments. And XMLRPC-C which is

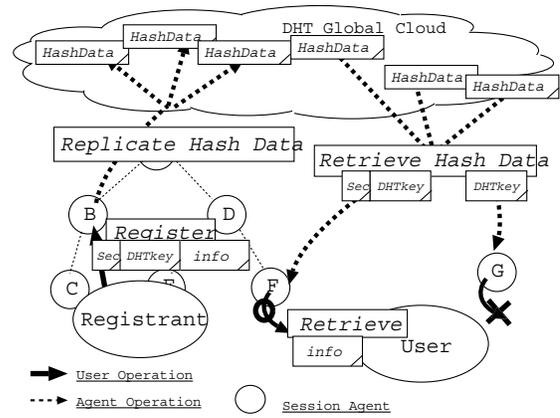


Fig. 7. Policy-aware Information Distribution in a DHT-based system

a lightweight RPC library is used to cooperate with Bamboo-DHT servers. Local database is implemented with MySQL database manager.

We evaluate the system by measuring the time of synchronizing session information to distributed agents. We set up a PlanetLab testbed [14] and analyzed the behavior of the system. PlanetLab is useful for the evaluation, because: 1) performance measurement can be carried out on a number of geographically distributed nodes (i.e. agents) over the real Internet, and 2) the actual implementation and its behavior can be justified during the experiment. Toward large scale deployment, both tree-based and DHT-based system were evaluated, and the following performances were measured; 1) session information registration, and 2) session information retrieval. The results are shown in Table. I.

A. Data Synchronization and Retrieval using the Tree-based System

To evaluate the performance of session information distribution using the tree-based system, we created a session agent tree having 51 nodes (6 root agents and 45 neighbor agents) over the PlanetLab testbed. Fig. 8 roughly shows the geographical topology of the agent tree we set up. Each root agent connects to other two root agents and creates the ring topology. Each root agent has own child agents, and we set up 4 child agents per branch as maximum. As the experimental scenarios, session information registrations and retrievals were initiated from one root agent in Tokyo, Japan to other session agents in USA. We measured session information synchronization delay on this topology for 1000 times.

Fig. 9 shows the synchronization delay of session information distribution from a local session agent on the topology. In our measurement, the average delay was 5.18 seconds, while the minimum delay was 2.40 seconds, and the standard deviation was 3.9.

Fig. 10 shows the delay of session information retrieval from a local session agent in our site local network. In our measurement, the average delay was 0.047 seconds, while minimum delay was 0.045 seconds, and the standard devi-

TABLE I
PERFORMANCE OF INFORMATION REGISTRATION AND RETRIEVAL USING BOTH SYSTEMS

	Registration				Retrieval			
	MessageSize	Min.	Avg.	Std.	MessageSize	Min.	Avg.	Std.
Tree – basedSystem	800bytes	2.39s	5.18s	3.9	800bytes	0.045s	0.047s	0.001
DHT – basedSystem	400bytes	0.55s	0.59s	0.079	800bytes	0.23s	1.13s	0.088

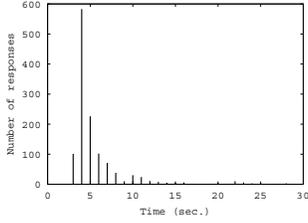


Fig. 9. Registration (Tree-based)

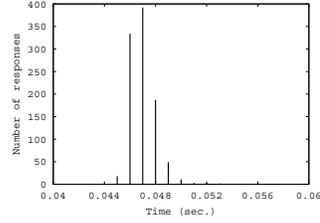


Fig. 10. Retrieval (Tree-based)

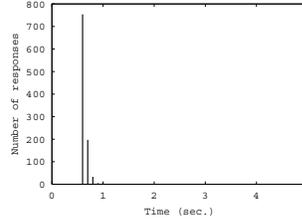


Fig. 11. Registration (DHT-based)

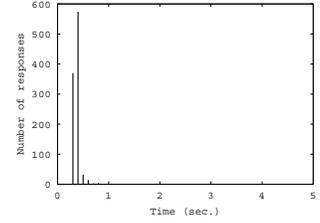


Fig. 12. Retrieval (DHT-based)

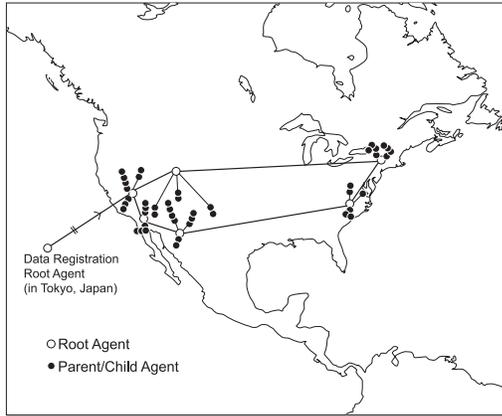


Fig. 8. Topology for the Tree-based System evaluation

ation was 0.001. Since the available session information is simply retrieved from the local agent's database, the time for information retrieval is very short.

B. Data Registration and Synchronization using the DHT-based System

To evaluate the performance of session information distribution using the DHT-based system, we created a session agents' topology over the PlanetLab testbed. Session information registrations and retrievals are initiated from the agent in Tokyo, Japan. We measured the session information synchronization delay in the DHT global cloud for 1000 times. Two types of tests were performed; with and without secret share. Since the results of delay time are almost same, we show the result measured with secret share in this paper.

Fig. 11 shows the performance of session information registration into a DHT global cloud. In our measurement, the average delay was 0.59 seconds, while the minimum delay was 0.55 seconds, and the standard deviation was 0.078. Since a hop-by-hop transmission among all agent nodes is not needed in DHT-based system, the synchronization delay is shorter than

the one using the tree-based system.

Fig. 12 shows the performance of session information retrieval from a DHT global cloud. In our measurement, the average delay was 1.13 seconds, while the minimum delay was 0.23 seconds, and the standard deviation was 0.088. Since the information may be retrieved from the remote DHT node, the time for information retrieval using the DHT-based system is occasionally longer than the one using the tree-based system.

C. Tree-based System v.s. DHT-based System

The following characteristics were observed in the experimental results;

Latency: Since the tree-based system requires data synchronization with all agents in a group, it takes longer time to complete session information registration. On the other hand, since in the tree-based system each agent stores session information in own local database, users can retrieve session information from their local agent faster.

Scalability: The DHT-based system gives higher scalability for session information registration, because it does not transmit complete data set to every agent. The transmitted message size is half of the one of the tree-based system.

Management: The tree-based system is statically configured by each site administrator. This static configuration gives operational cost higher, especially when the number of distributed agents increases. Since all agents in the DHT-based are connected and managed by DHT algorithm dynamically, the operational cost for its configuration is lower.

V. CONCLUSION AND FUTURE WORK

We proposed a new session announcement system using distributed session agents. We explained requirements of a session information announcement architecture, and proposed two systems to meet the requirements; 1) Tree-based System: using tree structure and hop-by-hop transfer, and 2) DHT-based System: using DHT global cloud based on DHT algorithm. We then illustrated the design of the distributed session agent, and provided the actual implementation.

We evaluated the behavior of the implemented session agents in the PlanetLab overlay network, and measured session information distribution delay. The experimental results have shown that our two types of session agent system can distribute and retrieve information to the entire Internet in feasible delay. According to the results, the DHT-based system is advantageous to announce session information to a large number of users in the Internet.

Our future work includes promotion of session agent architecture to the Internet communities and further evaluation with a large number of agents. While we have mainly focused the system architecture in this paper, designing not only simple but also functional user interface is important for real use of the proposed mechanism. Regarding an agent and a DHT bootstrap node address acquisition, while adopting an anycast mechanism would be the candidate, we will consider more effective way.

The service-oriented session announcement system should be adaptive to end-users' communication environments; it will provide individual media streaming information that is highly appropriate to *each* end-user. Our current session announcement system shows the first step for the service-oriented architecture by "policy-aware information distribution" that gives the access control with using a secret share. We will investigate other requirements towards the purely service-oriented systems and complete the system design and implementation in the future.

REFERENCES

- [1] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, 2007, pp. 181–192.
- [2] M. Handley and V. Jacobson, "SDP: Session Description Protocol," IETF, RFC 2327, Apr. 1998.
- [3] M. Handley, C. Perkins, and E. Whelan, "Session Announcement Protocol," IETF, RFC 2974, Oct. 2000.
- [4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," IETF, RFC 3261, Jun. 2002.
- [5] A. Lippman, K. Aizawa, R. L. Stevenson, and Y. Zhang, "Video coding for multiple target audiences," in *SPIE VCIP '99: Proceedings of Visual Communications and Image Processing*, vol. 3653, no. 1. SPIE, 1999, pp. 780–782.
- [6] L. Clark and M. A. Sasse, "Conceptual Design Reconsidered: The Case of the Internet Session Directory Tool," in *HCI '97: Proceedings of HCI on People and Computers XII*. Springer-Verlag, 1997, pp. 67–84.
- [7] H. Asaeda, W. Pokavanich, and S. Yamamoto, "Channel Reflector: An Interdomain Channel Directory System," *IEICE Transactions on Communications*, vol. E89-B, no. 10, pp. 2860–2867, October 2006.
- [8] H. Asaeda, K. Mishima, and V. Roca, "Requirement for IP multicast Session Announcement in the Internet," IETF, Internet Draft, Jul. 2008, draft-asaeda-mboned-session-announcement-req-00.
- [9] P. Namburi and K. Sarac, "Multicast session announcements on top of SSM," in *IEEE ICC '04: Proceedings of International Conference on Communications*, vol. 3. IEEE, 2004, pp. 1446–1450.
- [10] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.
- [11] R. van Engelen and K. Gallivan, "The gSOAP toolkit for web services and peer-to-peer computing networks," in *2nd IEEE International Symposium on Cluster Computing and the Grid*, 2002, p. 128.
- [12] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," in *Proceedings of USENIX Technical Conference*, 2004, pp. 127–140.
- [13] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," *Lecture Notes in Computer Science*, vol. 2218, pp. 329–350, 2001.
- [14] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: an overlay testbed for broad-coverage services," *SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 3–12, 2003.