

# AS-aware Relay Selection Mechanism for Multiparty Streaming Service in Overlay Network

Cheng-Yun Ho, Jyun-Chen Huang, Li-Hsing Yen, and Chien-Chao Tseng

**Abstract**—It has been shown that relay nodes can improve the quality of streaming data delivery in an overlay network. Therefore, many researchers have proposed methods for relay node selection. However, existing methods have some issues, such as scalability, long service start-up time, and not on multiparty service. In this paper, we propose an AS-aware relay selection (ASRS) mechanism to enhance the efficiency of multiparty streaming service in overlay networks. We verified and evaluated ASRS using ns2 network simulator. The experimental results show that ASRS has a short relay selection time. It can also reduce the bandwidth need of multipoint control unit (MCU) effectively so as to supports more users.

**Index Terms**—AS-aware, relay selection, ASRS, multiparty streaming service, overlay.

## I. INTRODUCTION

Many real-time multimedia services, such as VoIP and video conferencing, use peer-to-peer (P2P) overlay networks as the underlying transmission infrastructure to convey streaming data. These services usually demand bounded delay and high bandwidth. However, current P2P overlay may not fulfil these requirements. Some researchers [1]-[5] have proposed the use of relay nodes to deliver streaming data for improved communication quality. Many efforts [6]-[10] have been made in relay selection. However, existing relay selection methods have some issues. Some of them do not scale well, because nodes must acquire the statuses of many other nodes in the whole overlay network. Others spend too much time on relay selection, causing long service start-up latency. In addition, most existing methods focus on two-party communications rather than multiparty service, and do not jointly consider delay time and bandwidth.

To address these issues, this paper proposes an AS-aware Relay Selection (ASRS) mechanism for multiparty streaming services in overlay networks. In this mechanism, users belonging to the same Autonomous System (AS) are grouped together. Because an inter-AS packet is first routed to the AS gateway of the destination host, a route to a host can be used as the route to another host in the same AS. As a consequence, we can speed up the process of finding new relay nodes because the information of relay nodes already in-service in same AS can be used as representative results of the probe to

other hosts of the AS. Then, we propose a relay selection algorithm that considers both delay time and bandwidth. The selection algorithm prefers relays that are already in service so as to reduce the bandwidth need of Multipoint Control Unit (MCU), and thus can support more users to join the session.

We used ns2 [11] network simulator to verify and evaluate ASRS. We compared ASRS with direct connection method. Experimental results show that ASRS spends little time on relay node selections and can effectively reduce the bandwidth need of MCU so as to supports more users. In summary, ASRS has the following features: (1) AS-based relay node classification can reduce probe messages; (2) relay node selection reduces the bandwidth need of MCU; (3) relay selection algorithm considers both delay time and bandwidth.

The rest of this paper is organized as follows. Section II introduces background and related works. Section III describes ASRS and Section IV shows our experimental results. The last section concludes this paper and discusses our future works.

## II. BACKGROUND AND RELATED WORKS

There are two basic ways to deliver a multiparty streaming service: decentralized multipoint and centralized multipoint [12]. In decentralized multipoint, every client directly transmits its streaming data to all other clients. Clients themselves integrate and synchronize all received streaming data. In this way, client needs adequate bandwidth to deliver streaming data to all others. The client's bandwidth capacity limits the total number of clients that can participate in a multiparty streaming service.

In centralized multipoint, a client called initiator starts up a multiparty streaming service. Initiator also acts as an MCU which collects, synchronizes, and integrates all clients' streaming data and distributes the rendered result to all clients. This approach demands little bandwidth from clients, but MCU needs sufficient bandwidth to deliver streaming service to every client. Therefore, MCU is likely to become a performance bottleneck.

Many researchers have pointed out that selecting a suitable peer to relay voice packets can have a shorter latency than a direct IP routing path between two parties [6]. However, since there can be a large number of potential relay nodes, relay-selection may cause a long delay and considerable protocol overhead. The most time-consuming task in relay-selection is to estimate the routing latency between two peers, a task called probing. AS-Aware Peer-relay (ASAP) [6] aims to shorten probing latency by grouping nodes with the same IP prefix into a cluster. Since nodes in the same cluster

Manuscript received February 5, 2015; revised July 29, 2015.

C.-Y. Ho, J.-C. Huang, and C.-C. Tseng are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: cyho@cs.nctu.edu.tw, hicoy.hjc@gmail.com, cctseng@cs.nctu.edu.tw).

L.-H. Yen is with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan (e-mail: lhyen@nuk.edu.tw).

are relatively close to each other, the routing latency between two peers in different clusters can be estimated by that of any two peers in the corresponding clusters. ASAP also maintains an up-to-date annotated AS graph, based on which the latency of a routing path can be corrected to the number of AS hops of the path. With this information, two communicating peers can select a suitable relay to meet the VoIP delay constraint. However, the ASAP has some issues such as high relay node selection overhead and high start-up delay.

Liao *et al.* [7] proposed a new application-layer routing scheme that finds proper relay nodes to relay voice packets from source to destination. This scheme is based on Dynamic Promotion Landmark Mechanism (DPLM). In DPLM, a particular type of node called landmark keeps nearby relay node information. Nodes constantly communicate with a nearby landmark. When a node needs a relay, the landmark can offer a relay node candidate list, from which the node chooses the right relay node. Fig. 1 demonstrates a relay node selection process in DPLM. The simulation results show that performances of DPLM are better than those of Resilient Overlay Network (RON) [1] and Scalable One-hop Source Routing (SOSR) [2]. Compared with ASAP, DPLM is easy for implementation. However, DPLM has Triangle Inequality Violations (TIVs) issue [13] and only use latency information as the only metrics in relay selection.

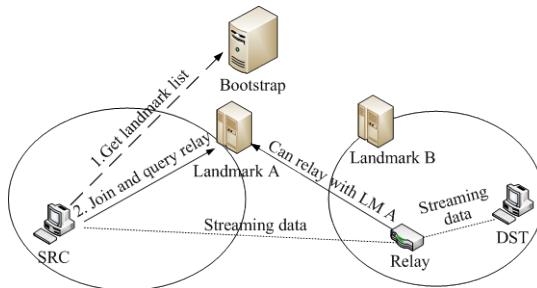


Fig. 1. Relay node selection process in DPLM.

### III. AS-AWARE RELAY SELECTION MECHANISM

This section presents the details of AS-aware Relay Selection (ASRS). In ASRS, each client is assigned a cluster to join. As the client joins the cluster, the client creates and maintains a relay node list. A client can also be a relay node. When a client joins a multiparty streaming session, the client exchanges relay node list with the initiator and selects a suitable routing path for this multiparty streaming session.

The characteristics of ASRS are as follows. 1) With ASRS, the clients are classified by their AS numbers and can reduce the probe overhead; 2) In relay selections, ASRS prefers relay nodes that already participate in relaying the requested streaming data. This strategy reduces the bandwidth need of MCUs; 3) Since ASRS creates relay node list when a node joins the cluster, ASRS can select relay node in a short time and decrease start-up delay.

#### A. ASRS Architecture and Entities

In this subsection, we introduce the ASRS architecture and entities in this architecture. Fig. 2 shows the ASRS architecture. Major entities in this architecture include bootstrap, signal server, cluster head, relay node, initiator,

and client. These entities are described below.

##### 1) Bootstrap

A bootstrap (BS) records the information of all clusters in ASRS. With this information, a client can select a best-suited cluster to join when it enters the overlay. A BS also records the topology and the cluster head of each AS.

##### 2) Signal server

The signal server (SS) records client's addresses and provides directory service. Moreover, the SS assists clients to exchange connection signals.

##### 3) Cluster head

Each cluster has a cluster head (CH), which is a relay node that has the largest bandwidth in the cluster. CH keeps information such as addresses and bandwidth of all relay nodes inside the cluster. CH discovers neighboring CHs by measuring the latency to every other CHs, and periodically updates the information. A new CH reports its address and AS number (ASN) to BS.

##### 4) Relay node

Relay node is configured to relay the requested multiparty streaming data.

##### 5) Client

A client is a user who uses a multiparty streaming service. The first client called initiator initiates a multiparty streaming session and all other clients join the session.

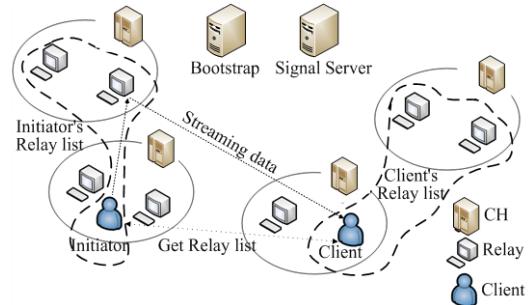


Fig. 2. ASRS architecture.

#### B. Client Joins a Cluster and Creates a Relay Node List

##### 1) Joins a cluster

When a client joins a cluster, the BS determines the client's CH based on the client's ASN. All clients with the same ASN are assigned to the same cluster. If the client is the first node in the cluster, the client becomes the CH. If the CH of the cluster already exists, the client reports its information (address and bandwidth) to the CH. The client will become the new CH of the cluster if the client has a larger bandwidth than the original CH. Fig. 3 shows the message flow when a client joins a cluster.

CH preselects its neighboring CHs to speedup on-line relay node selection. A new CH acquires from BS the addresses of all CHs that are within some distance (three hops or less) from it. These CHs constitutes a list of candidate neighboring CHs. The new CH then probes to obtain the latency to each CH in the list. Candidate CHs with latency shorter than a threshold form the neighbor list of the CH. If there is no such CH, the candidate CH with the smallest latency is the only element in the list. CH periodically measures the latency to each candidate neighboring CH to

update the neighbor list.

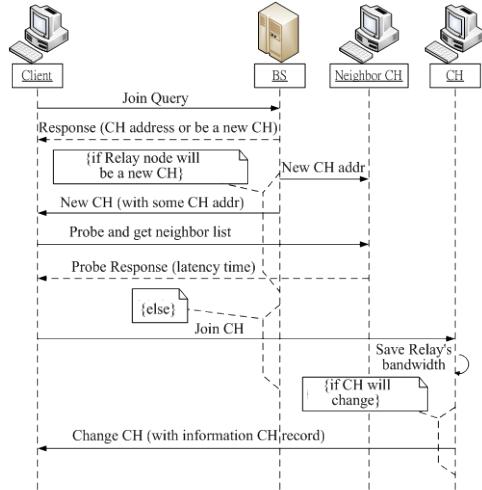


Fig. 3. Message flow when a client joins a cluster.

## 2) Creates a relay node list

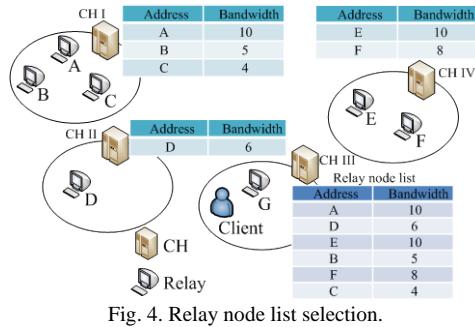


Fig. 4. Relay node list selection.

After a client joins the cluster, it needs to create a relay node list. The client first acquires the neighbor list of its CH. It then requests a relay node list from each CH in the neighbor list. After collecting a set of relay nodes from all neighboring CHs, the client probes each relay node to obtain its bandwidth and latency information. With this information, the client creates a relay node list by including relay nodes from each neighboring cluster. Relay nodes in this list must have latencies shorter than a threshold. For load balance concerns, the list includes relay nodes evenly from all clusters. In each cluster, the list includes relay nodes with high bandwidth first. Fig. 4 illustrates how a relay node list includes relay nodes from neighboring clusters.

## C. Join a Multiparty Streaming Session

When a client joins a multiparty streaming session, it needs to find an appropriate transport path to the initiator. The client first requests the initiator address from BS. Then, the client sends its relay node list to the initiator. The initiator also sends its relay node list and in-use relay node list of the multiparty streaming service to the client. After the client receives the list, it first probes in-use relay node list, and picks a qualified relay node, if any, with sufficient bandwidth and latency shorter than a threshold. If the client cannot find any qualified in-use relay node, it probes and selects a qualified relay node from the relay node list of the initiator. If the client still cannot find any qualified relay node, it selects a relay node from its relay node list. If the client fails to find a

qualified relay node either, the client attempts direct connection instead. The flowchart for the transport path selection policy is shown in Fig. 5.

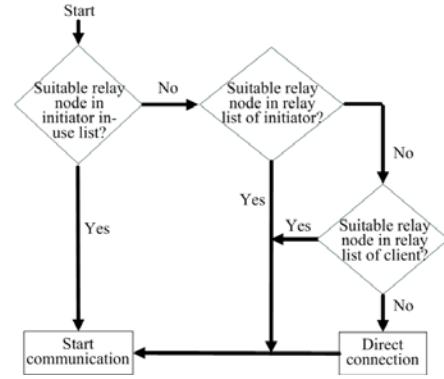


Fig. 5. Methodology to transport path selection.

## IV. SIMULATION AND RESULTS

To observe the performance of ASRS for multiparty streaming services in overlay networks, we implemented and simulated ASRS using ns2 [11]. Besides its popularity, we chose ns2 for the following reasons. 1) Naicken *et al.* [14] investigated many existing P2P simulators, such as PlanetLab [15] and OverSim [16], and pointed out that only few simulators were designed for P2P streaming applications since most of them focused on providing a convenient user interface (UI) for either distributed hash table (DHT) or gossip-based searching algorithm; 2) Many networking modules including P2P streaming applications have been designed for ns2.

### A. Experiment Environments and Parameter Settings

We obtained the datasets of Internet topology from [17]-[19] and analyzed these datasets to realize the number of ASs and the connection between two ASs in the Internet. Besides, the average link delay time between two nodes in different ASs, called the average inter-AS latency, and the average link delay time between two nodes in the same AS, called the average intra-AS latency, were obtained from [20]. In our experiments, there are 500 ASs and 1500 relay nodes. Each AS has at least one relay node. We set the average inter-AS latency to 20 ms and the average intra-AS latency to 30 ms. The demanded bandwidth for streaming data is 384 Kbps. Each client's bandwidth is 10 Mbps. More parameter settings are listed in Table I.

TABLE I: SIMULATION PARAMETERS IN ASRS

Name	Parameter	Name	Parameter
Number of ASs	500	Number of relay nodes	1500
Average inter-AS latency	20 ms	Average intra-AS latency	30 ms
Streaming data bit-rate	384 Kbps	Client bandwidth	10 Mbps
Number of experiments	1000	Simulation time	5 minutes
Number of clients	1-10		

### B. Experiment Results and Findings

For ASRS and direct connection (DIR), Fig. 6 shows the average delay time per session. The average delay time

generally increases with the number of clients participated in the session. However, ASRS can support more clients than DIR with the constraint that the average delay time is less than 200 ms. When delay time exceeds 200 ms, the quality of streaming service experienced by clients is considered unacceptable. That is why we take 200 ms the threshold for multiparty streaming service.

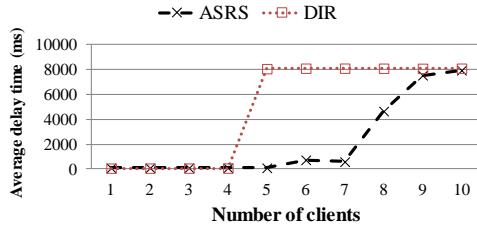


Fig. 6. Average delay time per session versus the number of clients per session.

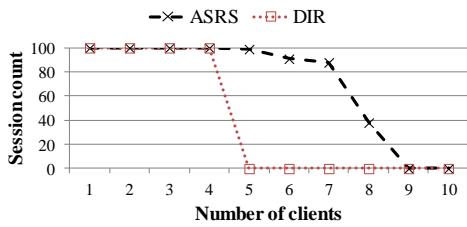


Fig. 7. Two mechanisms with variable clients.

We executed 100 multiparty streaming sessions with varied number of clients per session and observed the number of sessions with longest delay time less than 200 ms in each mechanism. Fig. 7 shows the result, which indicates that DIR can support only four clients in a multiparty streaming session. When the number of clients increases to five, no sessions had a longest delay time less than 200 ms. In contrast, about 90 sessions had a longest delay time less than 200 ms when ASRS was used to support seven clients per session. When the number of clients per session became eight, there were still 40 multiparty streaming sessions working well in ASRS.

Fig. 8 shows the average delay time of ASRS compared with those of SHORTEST and DIR when the number of clients was set to four. The SHORTEST mechanism probes all relay nodes and selects a relay node that has the shortest delay time. Although ASRS had a longer average delay time than SHORTEST and DIR in this setting, ASRS in general can support more clients as we have shown in Fig. 6 and Fig. 7.

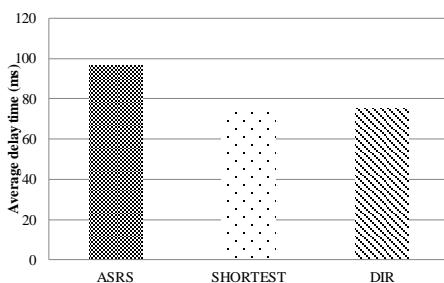


Fig. 8. Delay time among three different mechanisms.

Fig. 9 shows the number of packets forwarded by MCU. We observed that when using DIR, this amount increases

rapidly with the number of clients. The MCU forwards fewer packets in ASRS than in DIR. This is because ASRS adopts the idea of relay node to reduce the bandwidth demand of MCU.

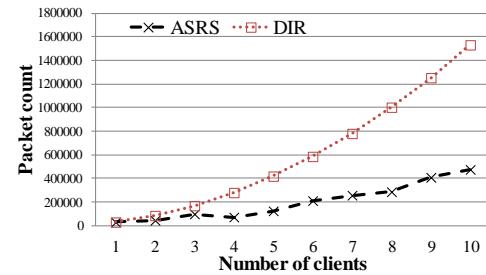


Fig. 9. Number of packets uploaded by the initiator.

Fig. 10 shows average streaming data delay time for various sizes of relay node list. This experiment used five clients in a multiparty streaming session. We varied the sizes of relay node list and observed the differences in average streaming data delay time. We found that the average streaming data delay time was longer than 180 ms when the relay node list size was smaller than five. As the relay node list size was larger than seven, the average streaming data delay time was shorter than 150 ms. We obtained the shortest average delay time when the size of relay node list was seven.

Fig. 11 shows the time spent in relay node selection and list formation. The relay node selection time is the time from a client joining a cluster to the client finishing a relay node selection. The relay node list formation time is the time from a client joining a cluster to the client finishing the relay node list creation. We observed that ASRS has a short relay node selection time, which would not affect service start-up time significantly.

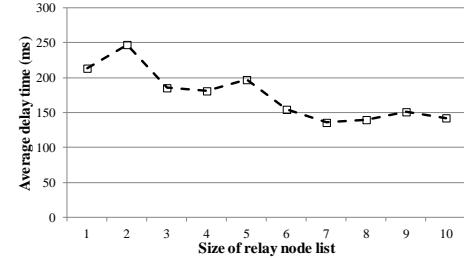


Fig. 10. Delay time for the various sizes of relay node list.

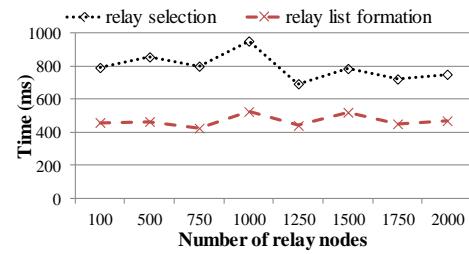


Fig. 11. Relay node selection and relay node list formation time in ASRS.

Fig. 12 compares the average delay time between applying and not applying the in-use relay node first selection policy. The former had some negligible effects on delay time, but the delay time does not increase linearly as the number of clients' increases.

Fig. 13 shows the average number of clients served by a relay node. We found that a high proportion of relay nodes were reused by ASRS. Consequently, the average number of

clients served by a relay node increases as the number of clients increases.

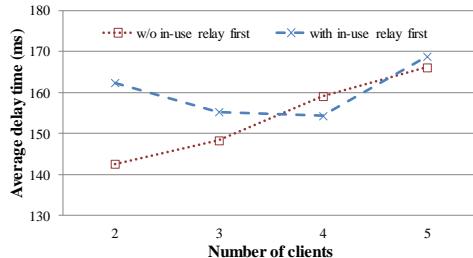


Fig. 12. With or without in-use relay node first.

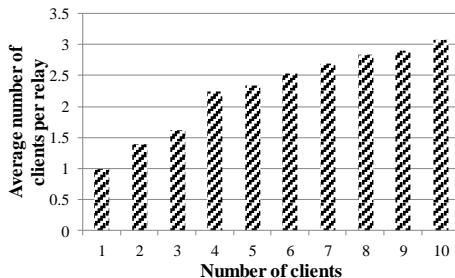


Fig. 13. Average number of clients served by a relay node.

## V. CONCLUSIONS AND FUTURE WORKS

In this paper, an effective relay node selection mechanism, ASRS, is proposed to enhance the performance in delivering multiparty streaming services. Since ASRS groups relay nodes by their AS numbers, ASRS could save considerable probe overheads. Experimental results show that the ASRS supports more clients and sent fewer packets than the counterparts. The results also show that the time spent by ASRS in relay node selection is not too significant. The in-use relay node first selection strategy is also confirmed effective. However, ASRS does not consider the case that a client is located behind an NAT or a firewall. Integration with NAT traversal scheme is an issue for our future study.

## REFERENCES

- [1] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *ACM SIGOPS Operating Systems Review*, vol. 35, issue 5, pp. 131-145, December 2001.
- [2] K.-P. Gummadi, H.-V. Madhyastha, S.-D. Gribble, H.-M. Levy, and D. Wetherall, "Improving the reliability of internet paths with one-hop source routing," in *Proc. the 6th conference on Symposium on Operating Systems Design & Implementation*, December 2004, vol. 6, pp. 13-13.
- [3] T. Fei, S. Tao, L. Gao, and R. Guerin, "How to select a good alternate path in large peer-to-peer systems?" *IEEE INFOCOM*, pp. 1-13, April 2006.
- [4] T. Fei, S. Tao, L. Gao, R. Guerin, and Z.-l. Zhang, "Light-weight overlay path selection in a peer-to-peer environment," *IEEE INFOCOM*, pp. 1-6, April 2006.
- [5] H. Yao, H. Jin, S. Zhou, L. Yu, and X. Liao, "A novel application-layer relay strategy for low delay P2P SIP VoIP system," in *Proc. the 4th International Conference on Ubiquitous Information Technologies & Applications (ICUT)*, Fukuoka, December 2009, pp. 1-6.
- [6] S. Ren, L. Guo, and X. Zhang, "ASAP: An AS-Aware Peer-Relay Protocol for High Quality VoIP," in *Proc. the 26th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2006.
- [7] X. Liao, H. Jin, and L. Zhu, "DPLM: A novel application-layer routing scheme for low delay VoIP," in *Proc. the 4th Asian Conference on Internet Engineering (AINTEC)*, November 2008, pp. 75-83.

- [8] S.-J. Lee, S. Banerjee, P. Sharma, P. Yalagandula, and S. Basu, "Bandwidth-aware routing in overlay networks," *IEEE INFOCOM*, April 2008.
- [9] T. Ma, C. Zhang, L. Li, and Y. Ji, "AS-aware relay selection using dht service discovery in P2P overlays," in *Proc. International Conference on Communications and Mobile Computing (CMC)*, Yunnan, January 2009, pp. 206-210.
- [10] H. Feng, Y. Xiao, and L.-J. Cimini, "Spectral efficiency of centralized and decentralized cooperative networks with relay selection," in *Proc. IEEE Military Communications Conference (MILCOM)*, San Diego, CA, November 2013, pp. 7-12.
- [11] The Network Simulator-ns-2. [Online]. Available: <http://www.isi.edu/nsnam/ns>
- [12] W. Jia, F.-P. Tso, and L. Zhang, "Efficient 3G324M protocol implementation for low bit rate multipoint video conferencing," *Journal of Networks*, vol. 1, no. 5, pp. 1-8, October 2006.
- [13] H. Zheng, E.-K. Lua, M. Pias, and T.-G. Griffin, "Internet routing policies and round-trip-times," *Passive and Active Network Measurement*, vol. 3431, pp. 236-250, 2005.
- [14] S. Naicken, A. Basu, B. Livingston, S. Rodhetbhai, and I. Wakeman. (April 2012). *Towards Yet Another Peer-to-Peer Simulator*. [Online]. Available: <http://sro.sussex.ac.uk/37851>
- [15] PlanetLab. [Online]. Available: <http://www.planet-lab.org>
- [16] I. Baumgart, B. Heep, C. Hubsch, and A. Brocco, "OverArch: a common architecture for structured and unstructured overlay networks," in *Proc. IEEE Conference Computer Communications Workshops*, Orlando, FL, USA, March 2012, pp. 19-24.
- [17] Route Views Project. [Online]. Available: <http://www.routeviews.org>
- [18] Y. Shavitt and E. Shir, "DIMES: let the internet measure itself," *ACM SIGCOMM Computer Communication Review*, vol. 35, issue 5, pp. 71-74, October 2005.
- [19] Internet Topology Collection. [Online]. Available: <http://irl.cs.ucla.edu/topology>
- [20] Internet Assigned Numbers Authority: Autonomous System Number Allocations. [Online]. Available: <http://www.iana.org/assignments/as-numbers/as-numbers.xml>



**Cheng-Yun Ho** received his B.S. degree in computer science from National Cheng Chi University, Taiwan, in 2007; the M.S. degree in networking engineering from National Chiao Tung University, Taiwan, in 2008 (enrolled into Ph.D. program directly). He is currently pursuing the Ph.D. degree in computer science and engineering at National Chiao Tung University, Taiwan. His research interests include peer-to-peer (P2P) network, information-centric networking (ICN), computer networks (including wireless and mobile networks), and network protocols.



**Jyun-Chen Huang** received his B.S. degree in computer science and M.S. degree in computer science and engineering from National Chiao Tung University, Taiwan, in 2008 and 2010 respectively. His research interests include peer-to-peer (P2P) network, wireless internet, and mobile computing.



**Li-Hsing Yen** received his PhD degree in computer science from National Chiao Tung University, Taiwan (1997). He had been with Chung Hua University, Taiwan from 1998 to 2006. He was an associate professor (2006 to 2010) and has been elevated to full professor (2010~) at the Department of Computer Science and Information Engineering, National University of Kaohsiung, Taiwan. His current research interests include mobile computing, wireless networking, and distributed algorithms. Dr. Yen has won IEEE WCNC 2013 Best Paper Award. He has served on the editor boards of Springer's Wireless Networks. Dr. Yen is a member of IEEE.



**Chien-Chao Tseng** received his B.S. degree in industrial engineering from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981; M.S. and Ph.D. degrees in computer science from the Southern Methodist University, Dallas, Texas, USA, in 1986 and 1989, respectively. He is currently a professor in the Department of Computer Science at National Chiao-Tung University, Hsin-Chu, Taiwan. His research interests include wireless internet, heterogeneous networks, and software defined networks.