Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 29 (2006) 3013–3019

computer communications

www.elsevier.com/locate/comcom

Extension headers for IPv6 anycast

Ching-Yu Lin [a], Ing-Yi Chen [b], Sy-Yen Kuo [a,*]

[a] *Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan*

[b] *Department of Computer Science and Information Engineering, Nation Taipei University of Technology, Taipei, Taiwan*

Available online 10 January 2006

Abstract

Anycast is a new communication paradigm defined in IPv6. Different from unicast and multicast routing, routers on the internetwork deliver an anycast datagram to the nearest available node. By shifting the task of resolving destinations from source node to internetwork, anycasting is highly flexible and cost-effective on routing process and inherently load-balanced and robust on server selection. To achieve these objectives, not only "distance" but also other metrics, such as load balance, reliability, QoS, can and should be taken into account in anycast routing. The IPv6 basic header is designed in a simple and fixed-length format for the purpose of efficient forwarding. Extra data and options needed for packet processing are encoded into extension headers. Such a design makes possible the adding of extension headers for special purposes. In this paper, we define routing extension headers for IPv6 anycasting to enable various types of anycast routing mechanism. Scenarios are also provided to demonstrate how to apply them.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Anycasting; IPv6; Extension header; Routing header

1. Introduction

"Host Anycasting Service" for IP nedtwork, submitted by Partridge et al. [1], was released as IETF RFC-1546 on Nov. 1993. The original motivation for anycasting is to simplify the task of finding an appropriate server at IP layer. In [1], Partridge et al. described the behavior of anycasting as follows: "A host transmits a datagram to an anycast address and the internetwork is responsible for providing best effort delivery of the datagram to at least one, and preferably only one, of the servers that accept datagrams for the anycast address". In other words, the internetwork promises to deliver packets to the closest available host through the shortest path. By shifting the task of resolving destinations from source node to internetwork, anycasting is highly flexible as well as cost-effective on routing process. In addition, anycasting is inherently load-balanced and robust on server selection.

IP anycast service is first proposed on IPv4. As a new type of address, "anycast address" is explicitly defined in "Internet Protocol, Version 6 (IPv6) Specification" (RFC-1883) on December 1995. It is used to send a packet to any single node of a group. RFC-1883 was obsoleted by RFC-2460 on December 1998 [2], which is the current version of IPv6 specification and is at the "Draft Standard" level now. RFC-2526, "Reserved IPv6 Subnet Anycast Address", is another important RFC that defines a set of reserved anycast addresses within each subnet prefix, and lists the initial allocation of these reserved subnet anycast addresses [3].

It is recognized that anycast potentially has the ability of providing a number of important services on Internet such as efficiently accessing well-known services, mirrored databases, web sites, and message servers. Many proposals and research results on anycast service have been published since the 90'. The experimental or scoped IPv6 implementations are widely deployed today. Relatively, the realization of anycast service is still far away.

The usages of IPv6 anycasting are still evolving. The definition and behavior of anycasting are not clear enough

[*] Corresponding author. Tel.: +886 2 2366 3577; fax: +886 2 2368 8247. *E-mail address:* sykuo@cc.ee.ntu.edu.tw (S.-Y. Kuo).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.11.008

until now. To make anycasting be able to be implemented at current stage, RFC-3513, which defines the addressing architecture of IPv6, imposes two restrictions on anycast addresses for security reasons [4]:

- Using anycast address as the source address of an IPv6 packet is not allowed.
- Assigning anycast address to IPv6 host is not allowed either. An anycast address can only be assigned to an IPv6 router.

These restrictions substantially limit the usage of anycast address. As a matter of fact, only a few applications can use anycast address well without breaking the restrictions. One example is the Mobile IPv6 Home-Agent anycast address [5].

Many researchers are trying to solve these problems. The IETF Multicast & Anycast Group Membership (magma) Working Group in Internet area is working on the group management protocol which allows hosts to inform routers their member status within a group. Furthermore, another new IETF Practical Anycasting Working Group is also being established to work on these issues. As people expect to make anycast service extensively utilized, we believe that these restrictions will be removed in the near future.

In the following sections, we will analyze and design the routing extension headers for IPv6 anycasting and explain how to apply these headers. The ultimate goal is to clarify the definition and behavior of anycasting. Our design also enables various new types of anycast routing mechanisms. The rest of this paper is organized as follows. In Section 2, we survey and analyze the current researches on IP anycasting. Three observations of important factors for anycasting are given as well. In Section 3, we present our design of anycast routing extension headers. Section 4 depicts scenarios to demonstrate how to use these headers. Section 5 concludes the paper.

2. Survey and analysis

We briefed the definition of anycasting in RFCs and its current status in the last section. The use of IPv6 anycasting is still evolving, and the definition is unclear at present. In this section, we first categorize the current researches on anycast. Then, we will analyze the variations of anycast definition and behavior between RFCs and current researches. Based on the analysis, we will design anycast routing headers in the next section.

The researches on IPv6 anycasting can be divided into six categories in terms of topic. For each category, a lot of efforts have been done. Considering the relevance to our paper, we only explain some of them in more detail. The first category is survey and introduction. In these papers, some basic concepts on IPv6 anycast are discussed [6–8]. The second category focuses on anycast architecture [9–11]. The GIA system [9] provides a possible solution to global-scale anycasting. The i3 network [10] proposes a new infrastructure which is responsible for mapping the destination ID to the IP address of the final recipient. In [11] a protocol was given for anycast group membership management. Routing algorithm is the third category. Papers in this category deal with problems of server selection (or named as destination selection). The task of server selection is to select an appropriate server from all candidate destinations mapped to the anycast address according to the metric of routing algorithm. Possible metrics include response time, QoS, load balance, fault tolerance etc., and combinations of them. Once the destination is selected, router can send the datagram out using information in the routing table. [12–15] are some researches on routing algorithm.

In category four, application of anycast service is discussed [16,17]. How to apply anycast service is a very important topic. Facilities of anycast service should be better understood by people. Only by doing that, the full deployment of anycast-supported routers can become possible. Category five probes into the evaluations and analyses of anycasting [18]. As for category six, which pertains to mobile networks, a number of research results are presented on how anycast can be applied to a mobile environment. This category is out of the scope of this paper.

In the process of categorization, we observed three important factors for anycasting:

(i) The number of datagram copies transferred on the network or the number of destinations the datagram is sent to.
(ii) The policy of destination selection.
(iii) The replaceability of anycast address by the unicast address of an available destination.

The RFCs propose that an anycast datagram should be sent to “one” “nearest” available destination without address replacement. By assigning different values of these three factors, we can enable many interesting communication models. The details will be presented in the next two sections.

3. Design of anycast routing headers

All IP datagrams start with a basic header. Extension headers follow the basic header and contain options and information for datagram exception processing. Datagrams with the basic header only are handled by a router using its fast path; datagrams with additional processing requests are routed via a slow path. Headers are recommended to appear in a certain order [2]. Generally speaking, routers only parse hop-by-hop options and routing headers. Once a router reaches a header which does not belong to hop-by-hop options and routing headers, it does not need to look further at the datagram.

Fig. 1 is the format of IPv6 routing extension header. The routing header gives the source node the ability to con-

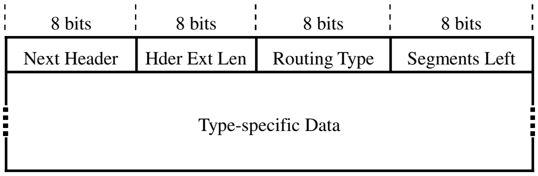

Fig. 1. Routing extension header of IPv6.

trol the routing path of datagrams. So far, three types of routing header have been proposed [19].

- Type 0: Source Routing
  In type 0 routing header, the source node lists those intermediate routers which must be visited from itself to the destination.
- Type 1: Nimrod
  Type 1 is proposed by Charles Lynn [19].
- Type 2
  A new type 2 routing header is defined in Mobile IPv6. It allows the packet to be routed directly from a correspondent node to the mobile node's care-of address.

3.1. Format of anycast routing header

Anycast addresses are allocated from the unicast address space and syntactically indistinguishable from unicast addresses. However, the routing algorithm applied to anycast datagrams should be different from that to unicast datagrams. To enable the ability of assigning routers to use different algorithms, we propose a new routing type variant for anycast routing and designate it as type 3.

Type 3 routing header is used for a source node to assign the routing algorithm and parameters applied on the anycast datagram. The routing algorithm is specified in Routing Method field, and parameters are assigned in Method-specific data field. From Sections 3.3–3.5, we define three classes of anycast routing header for different routing methods. Other routing algorithms can define its own class of anycast routing header if needed.

Fig. 2 is the format of anycast routing header we define. The first three fields are same as defined in [2]. Fields are described as follows:

- Next Header:
  An 8-bit selector identifies the type of header immediately following the routing header, and uses the same values as the IPv6 Next Header field.
- Hder Ext Len:
  An 8-bit unsigned integer represents the length of the routing header in 8-octet units, not including the first 8 octets.
- Routing Type = 0x03:
  An 8-bit identifier indicates a particular routing header variant.

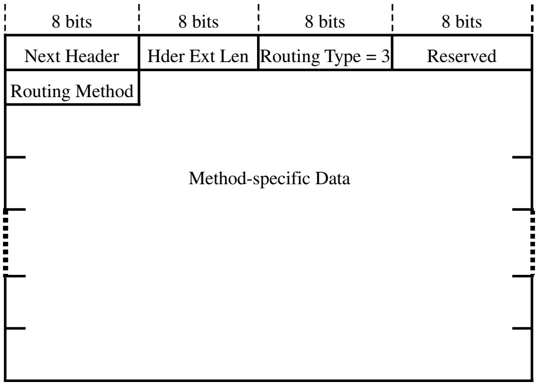

Fig. 2. Format of anycast routing header.

- Reserved
  An 8-bit reserved field defines this field as Segments Left field for type 0 routing header in RFC-2460. However, it may be meaningless for other routing types. We reserve this field for the compatibility reason.
- Routing Method:
  An 8-bit selector identifies the method of anycast routing used to route the datagram.
- Method-specific Data:
  A variable-length field, of format determined by the Routing Method, and of length such that the complete routing header is an integer multiple of 8 octets long.

3.2. Routing header for multiple destination routing method

Routing header for multiple destination routing method is used for source node to request routers to duplicate datagrams in routing process. The reasons for duplicating datagrams are twofold: Source node may want to find more than one server by one single anycast request, or it may want a datagram to be sent through a fault-tolerant and reliable routing method.

Fig. 3 is the format of routing header for multiple destination routing method. Fields in the header are described as follows:

- Routing Method = 0x01
- Diverse Level:
  An 8-bit unsigned integer represents the number of remaining diverse levels. If Diverse Level is 0xFF, the datagram will be attempted to be delivered to all nodes in the anycast group.

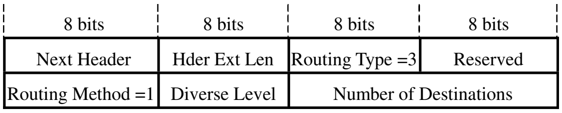

Fig. 3. Routing header for multiple destination routing method.

- Number of Destinations:An 16-bit unsigned integer represents the number of destination nodes that should be sent.

When a router receives a datagram with a multiple destination routing header appended, the following algorithm will be performed.

```
if Diverse Level = 0 {
    get Number of Destination, n0, from the received datagram according to the routing algorithm, determine the number of copies, i, and their Number of Destinations n1 ... ni, the summation of n1 to ni should equal to n0 duplicate the original datagram into i copies, set Diverse Level to 0, and Number of Destinations to n1 ... ni send out these i copies to different outgoing links
}
else if 0 < Diverse Level < 0xFF {
    duplicate the datagram into k copies specified in Number of Destinations in k duplicated datagrams, decrease Diverse Level by one and set Number of Destinations to k send out these k copies to different outgoing links
}
else if Diverse Level = 0xFF {
}   send out duplicated datagrams to all outgoing links with an available destination
```

3.3. Routing header for favorite-list routing method

In many situations, source node is not ignorant about the available destinations of an anycast address. It may have a favorite destination list for an anycast address obtained from some other ways or previous caches, and wants to provide the list to routers to improve the quality of destination selection. The favorite destination list can be either strictly limited or unstrictly limited. Also, it can be either ordered or unordered. There may be another list, a disfavorite destination list, and its behavior is similar to the favorite destination list.

Fig. 4 is the format of routing header for favorite-list routing method. Fields in the header are described as follows:

- Routing Method = 0x02
- F flag:
  A 1-bit flag indicates either a favorite destination list ($F = 1$), or a disfavorite destination list ($F = 0$).
- S flag:
  A 1-bti flag indicates the destination list is either strictly limited ($S = 1$), or unstrictly limited ($S = l0$). If the flag is set, a router can only send the datagram to an address in the favorite destination list. Or it can not send the datagram to any address in the disfavorite destination list.

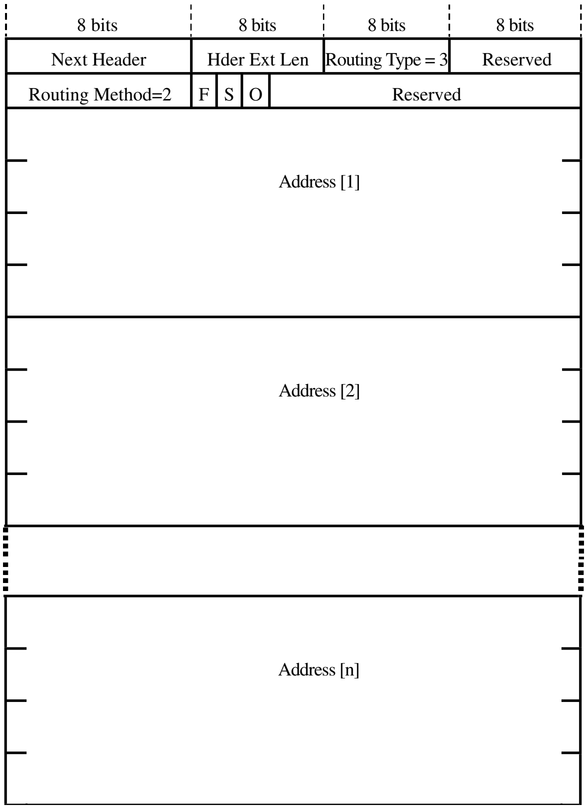

Fig. 4. Routing header for favorite-list routing method.

- O flag:
  A 1-bit flag indicates the list is either ordered ($O = 1$), or unordered ($O = 0$). If the flag is set, it means that an address in a favorite destination list is preferred to any others behind it.
- Address [1..n]:
  128-bit addresses comprise the favorite/disfavorite destination list. Addresses in the list should be the IPv6 unicast addresses of destination nodes that the anycast address refers to.

When a router receives a datagram with a favorite-list routing header appended, the following algorithm will be performed.

```
let k be the number of destinations to be selected. k = 1 by default, or determined by multiple destination routing header if it exists.
select case {
case: F = 1, S = 0, O = 0
    randomly select k destinations from the favorite destination list if they are reachable.
    if the number of reachable destinations in the list is less than k, the insufficient parts are determined by routing algorithm
case: F = 1, S = 0, O = 1
    select the first k reachable destinations from the list.
```

```
        if the number of reachable destinations in the list is less
        than k, the insufficient parts are determined by routing
        algorithm.
    case: $F = 1$, $S = 1$, $O = 0$
        randomly select k reachable destinations from the list.
        no other destinations are selected even the number of
        reachable destinations in the list is less than k
    case: $F = 1$, $S = 1$, $O = 1$
        select the first k reachable destinations from the list.
        no other destinations are selected even the number of
        reachable destinations in the list is less than k
    case: $F = 0$, $S = 0$, $O = 0$
        select k reachable destinations not on the disfavorite
        list.
        if the number of reachable destinations not on the list is
        less than k, randomly select the insufficient parts from
        the list.
    case: $F = 0$, $S = 0$, $O = 1$
        select k reachable destinations not on the disfavorite
        list.
        if the number of reachable destinations not on the list is
        less than k, select the insufficient parts from the last of list.
    case: $F = 0$, $S = 1$, O do not care
        select k reachable destinations not on the disfavorite
        list.
        no other destinations are selected even the number of
        reachable destinations not on the list is less than k
    }
```

3.4. Routing header for decision-deferred routing method

If the anycast address specified in the destination field of a datagram can be replaced by a unicast address of a selected destination during its routing process, the datagram can be routed through a fast path as normal unicast datagrams at its latter remaining itinerary. This improves the routing performance of an anycast datagram. On the other hand, an anycast address which is not replaced by a unicast address in a datagram can keep the flexibility on selecting an appropriate destination. This benefits the destination selection task. A source node may want to express whether an address replacement is needed. The decision-deferred routing header is designed for this purpose.

Fig. 5 is the format of routing header for decision-deferred routing method. Fields in the header are described as follows:

- Routing Method = 0x03
- TTR:

A 2-bit selector stands for Time-To-Replace.

0: an anycast address specified in the destination field is replaceable by the unicast address of a selected destination, and the replacement time is determined by the routing algorithm.
1: replaces the anycast address as soon as possible.
2: replaces the anycast address as late as possible.
3: decision-deferred routing; replaces the anycast address if no more information can be obtained.

8 bits	8 bits	8 bits	8 bits
Next Header	Hder Ext Len	Routing Type =3	Reserved
Routing Method=3	TTR	Reserved	

Fig. 5. Routing header for decision-deferred routing method.

When a router receives a datagram with a decision-deferred routing header appended, the following algorithm will be performed.

```
if TTR = 0 {
    replace the anycast address in the destination field with
    the unicast address of a selected destination at any time
    the routing algorithm requests to
}
else if TTR = 1 {
    replace the anycast address once any available destination is found
}
else if TTR = 2 {
    don't replace the anycast address unless it is needed
}
else if TTR = 3 {
    replace the anycast address only if no more information
    can be obtained
}
```

3.5. Extensibility of anycast routing header

Anycast routing headers are the type 3 routing extension headers of IPv6; therefore, they have the same ordering relations as type 0 routing header defined in RFC-2460. As for the ordering of various classes of anycast routing headers, it can be arranged at will.

The number of various classes of anycast routing headers in a datagram is not limited. It can be from 0 to several. The same class header may appear more than once too. For example, a source node may specify a favorite destination list and disfavorite destination list in the same datagram. However, the combination of different classes of headers should still remain reasonable and achievable.

Anycast routing header is extensible. In the above sections, we have defined three classes of anycast routing headers for different anycast routing methods. More classes of headers can be defined in the future if needed. As many as 256 classes of headers can be defined in our design.

4. Scenarios

In this section, we will give each class of anycast routing header at least one scenario to demonstrate how the header can be applied.

4.1. Scenarios for the multiple destination routing header

4.1.1. Scenario 1

Considering a distributed computing system on Internet, servers providing computing power are joined together with an anycast address. A client node wants to allocate a computing power which needs to be provided by a number of servers collectively. The client node can send the request using an anycast datagram with a multiple destination routing header. And the number of servers needed, say k, can be assigned in the Number of Destinations field. Then the request datagram will be duplicated and delivered to the nearest k servers. Furthermore, the multiple destination routing header can combine with routing headers related to server selection policy, e.g., the favorite-list routing header, to select the k servers using different criteria.

4.1.2. Scenario 2

In some cases, the client node may want to locate a specific server in an anycast group but has no idea which and where the server is. The multiple destination routing header with the Diverse Level greater than 0 is designed for this purpose. The datagram will be duplicated and delivered to different destinations. The values of the Diverse Level and the Number of Destinations fields determine the depth and width of the spread of the datagram. With the help of the Hop Limit field in the basic header, datagrams are prevented from being forwarded infinitely.

4.1.3. Scenario 3

In most of the distributed systems, a server discovery process or a state synchronization process will be needed at the initial time or periodically. The multiple destination routing header with Diverse Level equal to 0xFF can be used to notify the routers to deliver the datagram to all the available servers. In fact, it is the multicast scheme for an anycast group.

4.2. Scenarios for the favorite-list routing header

4.2.1. Scenario 1

The current implementation of a P2P application usually needs to connect to one of a number of directory servers in the beginning. Anycast can be utilized to connect to the nearest server. Theoretically, every directory server can provide the same service. However, distance, server loading, capability, information of these directory servers could be different. By way of the favorite-list routing header, users can assign their favorite servers in Address [1..n] according to their own priorities.

4.2.2. Scenario 2

Most anycast routing algorithms have the tendency to route datagrams with the same anycast address to the same destination. Sometimes, the destination selected by a router could be unreachable because of network link or hardware failures. Anycasting will select another available one automatically after the network detects the failure. During this period, the datagram cannot be delivered to any other available destinations. Furthermore, a destination selected by a router could contain fabricated data. In both cases, the source node can explicitly list their disfavorite servers in Address[1..n] of a favorite-list routing header and set F flag to 0. S flag can be set to 1 to indicate the strictness as well. By doing that, a basic defense against counterfeit servers is provided.

4.3. Scenario for the decision-deferred routing header

4.3.1. Scenario

In [20], Yamamoto et al. proposed an active anycast routing method. The router selects a destination based on some probability and previously measured metrics. Once the destination is selected, the destination address of the datagram is replaced with the unicast address of the target node.

In this case, a decision-deferred routing header can be used. TTR can be set to 1 to inform router to replace the anycast address with the corresponding unicast address as soon as possible.

5. Conclusions

Contributions of our paper to IPv6 anycast are in three aspects:

First, we have clarified the definition and behavior of anycasting. Anycast service is expected to be extensively implemented. Many researchers and organizations have been devoted to the development of the service with much effort. However, due to the obscuration of definition and restrictions, the use of anycast service is limited and far from realized. We presented three variation factors of anycasting base on observations. Then we defined the type 3 routing extension header for IPv6 anycasting. By defining the headers, the definition and behavior of anycasting were refined. With respect to routing algorithm, anycasting should be different from unicasting and multicasting. The proposed anycast routing headers enable anycast datagrams to indicate their desired routing algorithms.

Second, a number of new possible anycast routing models were explored. We have proposed three classes of routing extension headers in this paper. The new routing models were explored by assigning various values to fields in the headers. Anycast routing headers we have defined substantially increase the capability of anycasting and make it more complete.

Third, we have provided a good architecture to incorporate most of the current anycast routing algorithms and made them possible to be implemented in a standard way. The extensibility of anycast routing headers is well-designed. More classes of headers can be defined for different anycast routing models in the future.

There is still much work to be done before IPv6 anycasting can be realized. It is hoped that this paper can bring us one step toward the realization.

Acknowledgement

This research was supported by the National Science Council, Taiwan, under Grant NSC 92-2213-E-002-011.

References

[1] C. Partridge, T. Mendez, W. Milliken, Host Anycasting Service, IETF RFC 1546, November 1993.

[2] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, IETF RFC 2460, December 1998.

[3] D. Johnson, S. Deering, Reserved IPv6 Subnet Anycast Addresses, IETF RFC 2526, March 1999.

[4] R. Hinden, S. Deering, Internet Protocol, Version 6 (IPv6) Specification, IETF RFC 3513, April 2003.

[5] D. Johnson, C. Perkins, J. Arkko, Mobility Support in IPv6, IETF RFC 3775, June 2004.

[6] S. Weber, L. Cheng, A survey of anycast in IPv6 networks, IEEE Commun. Mag. 42 (1) (2004) 127–132.

[7] C. Metz, IP anycast point-to-(any) point communication, IEEE Internet Comput 6 (2) (2002) 94–98.

[8] S. Ata, H. Kitamura, and M. Murata, Applicability Statement of IPv6 Anycasting, draft-ata-ipv6-anycast-app-00.txt, October 2004.

[9] D. Katabi and J. Wroclawski, A Framework for Scalable Global IP-Anycast (GIA), Proceedings of ACM SIGCOMM, August 2000.

[10] I. Stoica, D. Adkins, S. Zhung, S. Shenker, S. Surana, Internet indirection infrastructure, IEEE/ACM Trans. Netwoking 12 (2) (2004) 205–218.

[11] V. Ponnusamy, E.K. Karuppiah, R. Abdullah, Anycast group membership management protocol, The 9th Asia-Pacific Conference on Communications (APCC 2003), vol. 3, pp. 1052–1056, September 2003.

[12] D. Xuan, W. Jia, W. Zhao, H. Zhu, A routing protocol for anycast messages, IEEE Trans. Parallel Distributed Syst. 11 (6) (2000) 571–588.

[13] C.Y. Lin, J.H. Lo, S.Y. Kuo, Load-balanced anycast routing, Proceedings of Tenth International Conference on Parallel and Distributed Systems (ICPADS 2004), July 2004.

[14] F. Hao, E.W. Zegura, M.H. Ammar, QoS routing for anycast communications: motivation and an architecture for diffserv networks, IEEE Commun. Mag. 40 (6) (2002).

[15] Zhang Li, Weijia Jia, Yan Wei, Li Xiao-Ming, An efficient anycast routing protocol based on multi-metrics, Proceedings of 7th International Symposium on Parallel Architectures, Algorithms and Networks, May 2004.

[16] G. Agarwal, R. Shah, J. Walrand, Content distribution architecture using network layer anycast, Proceedings of the Second IEEE Workshop on Internet Applications (WIAPP 2001), July 2001.

[17] D. Katabi, The use of ip-anycast for building efficient multicast trees, Global Telecommunications Conference (GLOBECOM '99), vol. 3, 1999.

[18] K.M. Hanna, N. Natarajan, B.N. Levine, Evaluation of a novel two-step server selection metric, Ninth International Conference on Network Protocols, November 2001.

[19] J. Reynolds, Assigned numbers: RFC 1700 is replaced by an on-line database, IETF RFC 3232, January 2002.

[20] M. Yamamoto, H. Miura, K. Nishimura, H. Ikeda, A network supported server load balancing method: active anycast, IEICE Transactions on Communications, Special Issue on New Development on QoS Technologies of Information Networks, vol. E84-B, No. 6, June 2001, pp. 1561–1568.

Ching-Yu Lin received the MS in Computer Science from Nation Chung-Hsing University at 1998, and Ph.D. degrees in Electrical Engineering from National Taiwan University. His research interests include computer networks, mobile computing, and reliable computing, etc. His recently publications more focus on the high-availability network.

Ing-Yi Chen received a BS in Physics from National Central University, Taiwan, in 1984, and MS and Ph.D. degrees in Electrical and Computer Engineering from the University of Arizona, USA, in 1989 and 1992, respectively. Dr. Chen is currently an associate professor in Computer Science and Information Engineering Department, National Taipei University of Technology, Taipei, Taiwan. Prior to joining NTUT, he served as chief technology officer for Chinatimes from 2000 to 2001 with responsibility for the corporation's strategic technology planning and external technical affiliations with universities. Previously, he was a faculty member in the Department of Electronic Engineering at Chung Yuan Christian University, Taiwan between 1992 and 2000. His research interests include various topics in Computer Communications and web services. A member of the Phi Tau Phi scholastic honor society, Dr. Chen was a recipient of the IEEE/ACM Design Automation Scholarship Award in 1991, and the Distinguished Teaching Award at CYCU in 1996.

Sy-Yen Kuo is Dean of the College of Electrical Engineering and Computer Science, National Taiwan Ocean University, Keelung, Taiwan. He is also a Professor at the Department of Electrical Engineering, National Taiwan University where he is currently on leave and was the Chairman at the same department from 2001 to 2004. He received the BS (1979) in Electrical Engineering from National Taiwan University, the MS (1982) in Electrical and Computer Engineering from the University of California at Santa Barbara, and the Ph.D. (1987) in Computer Science from the University of Illinois at Urbana-Champaign. He spent his sabbatical years as a Visiting Professor at the Computer Science and Engineering Department, the Chinese University of Hong Kong from 2004 to 2005 and as a visiting researcher at AT&T Labs-Research, New Jersey from 1999 to 2000, respectively. He was the Chairman of the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan from 1995 to 1998, a faculty member in the Department of Electrical and Computer Engineering at the University of Arizona from 1988 to 1991, and an engineer at Fairchild Semiconductor and Silvar-Lisco, both in California, from 1982 to 1984. In 1989, he also worked as a summer faculty fellow at Jet Propulsion Laboratory of California Institute of Technology. His current research interests include dependable systems and networks, software reliability engineering, mobile computing, and reliable sensor networks.

Professor Kuo is an IEEE Fellow. He has published more than 240 papers in journals and conferences. He received the distinguished research award between 1997 and 2005 consecutively from the National Science Council in Taiwan and is now a Research Fellow there. He was also a recipient of the Best Paper Award in the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/ACM Design Automation Conference (DAC), the National Science Foundation's Research Initiation Award in 1989, and the IEEE/ACM Design Automation Scholarship in 1990 and 1991.