

# IP layer load balance using fuzzy logic under IPv6 anycast mechanism

By Chi-Yuan Chang, Wei-Ming Chen, Han-Chieh Chao<sup>\*,†</sup>, T. G. Tsuei and Hong Bin Liu

*With the emergence of databases and Internet technology, the way data is acquired and accessed has changed with more convenient service. After years of evolution, the current IPv6 Internet Protocol (IP) has enabled the user to check any database worldwide by logging onto the Internet whenever you want and regardless of where you are. The load balance or efficiency issue arises when many users overload a server or database. This paper addresses this issue using the anycast characteristics. An active router design is developed to receive the server load information. This information is in turn used to determine the server's response to solve the load balance problem. Copyright © 2005 John Wiley & Sons, Ltd.*

## Introduction

The main purpose of load balancing is to continuously provide services even in cases of malfunction in any single server. With the load-balancing feature, the disordered host can be isolated, and the data traffic can be transferred to another effective host. The data-processing load is thus shared by all available servers instead of being concentrated in only a few servers. How does this work? First, let's assume that a user wants to connect and access a database

to retrieve data. Generally, the user must connect to the manager server first and the manager will tell the user which server to use to access the database. The manager server controls all portal servers and performs load balancing. These operations are performed in the application layer of the manager server.

We will utilize the anycast characteristics of IPv6 and design an active router using fuzzy logic to receive the load information. The information can be used to determine the server mechanism used to solve the load balance problem.

---

Chi-Yuan Chang is a PhD student in the Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan.

Wei-Ming Chen is an Assistant Professor in the Department of Information Management, National Dong Hwa University, Hualien, Taiwan.

Han-Chieh Chao is a Professor in the Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan.

T. G. Tsuei is an Associate Professor in the Department of Electrical Engineering, Ta Hwa Institute of Technology, Hsinchu, Taiwan.

Hong Bin Liu is a graduate student in the Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan.

\*Correspondence to: Han-Chieh Chao, Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan, ROC

†E-mail: hcc@mail.ndhu.edu.tw

This paper is organized as follows. The next section will focus on the anycast addresses. The third section will describe the fuzzy theory inference procedure. The fourth section will show the proposed architecture and simulation results. Finally, conclusions and future work are discussed.

## Anycast Address

Although IPv6<sup>1</sup> has allocated an address scheme for local anycast, it is not yet defined for global anycast. There are also two limitations for anycast in RFC3513:<sup>2</sup>

1. An anycast address cannot be used as the source address for an IPv6 packet.
2. An anycast address must not be assigned to an IPv6 host; that is, it may be assigned only to an IPv6 router.

The IPv6 unicast address can be deemed as one-to-one communication. The multicast address is one-to-many communication. The anycast address belongs to one of one-to-many communications and is described in RFC3513 as follows: An IPv6 anycast address is an address that is assigned to more than one interface (typically belonging to different nodes), with the property that a packet sent to an anycast address is routed to the 'nearest' interface having that address, according to the routing protocol distance measurement.<sup>2,3</sup>

The nearest distance in the routing protocol depends on which routing protocol is active. Each routing protocol method is different. The RFC concept has only the address scheme defined for local anycast and there is no standard routing protocol like RIPng (Routing Information Protocol v6) for anycast.

Today, the usage of IPv6 anycasting is still unclear. Ata *et al.*<sup>4</sup> have proposed the Internet draft and describe some applications which IPv6 anycasting intends to resolve. These application examples include 'establishing a service discovery', 'load distribution', 'improving system reliability', 'host auto-configuration', 'gate to overlay network' and 'local information service'.

## Fuzzy Theory<sup>5</sup>

This section will briefly introduce the inference technology structure composed of a fuzzy set,

fuzzy rule base and defuzzification. The procedure is shown as follows:

### —Step 1: Define the Fuzzy Set—

Choose proper out- and in-variables. The out- and in-variables can be acquired using practical and measurable data or selecting usable information with the knowledge and experience of experts. Some variables can be inferred from the original data. For example, a temperature difference in which the lowest temperature is subtracted from the highest temperature is inferred from known data. If it is useful for the whole inference, it can be thought of as in-variables. Let the selected out- and in-variables be expressed in fuzzy sets and define the membership function and linguistic items.

### —Step 2: Design the Fuzzy Rules—

Use of the if-then rule expresses the fuzzy set defined by linguistic variables in step 1. The set composed of several if-then statements is called the rule base. Usually, the if-then rule base can be inferred using the experience of the operator and the knowledge of experts.

### —Step 3: Use of Defuzzication—

Choose proper defuzzication and convert the fuzzy output value that is inferred with the if-then rule base into a practical crispy value.

## Proposed System Architecture

The proposed system architecture is shown in Figure 1. This study takes advantage of IPv6, using three addresses: unicast address, multicast address and anycast address, with active router.<sup>6</sup> A fuzzy logic algorithm is used in the active router to perform the load balance.<sup>7</sup> This fuzzy logic approach performs load balancing in the IP layer.

### —Active Router—

How is an active router different from the general router? As shown in figure 2, the active router is a node inside an active network.<sup>8</sup> Nor-

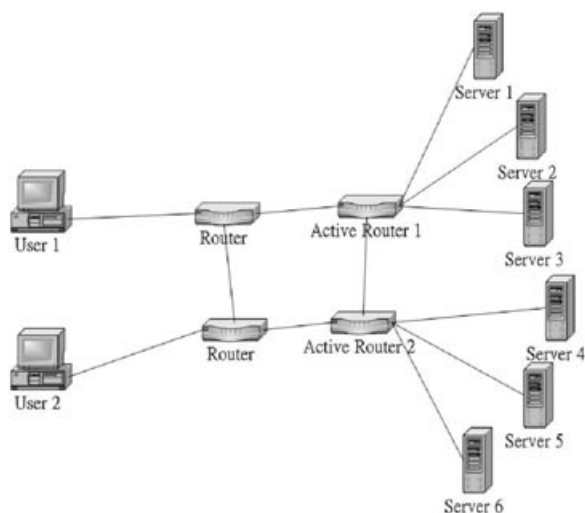
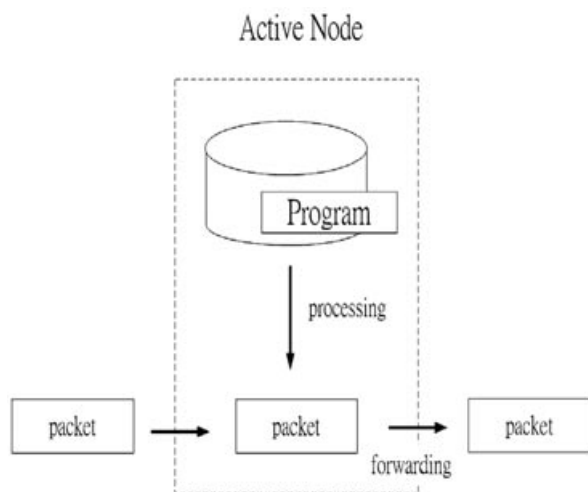


Figure 1. Proposed system architecture

Figure 2. Active node in active network<sup>8</sup>

mally, after a network packet enters a network node, the responsibility of the network node is to deliver packets that contain only data. It delivers the packet to the destination address according to the routing table. This model is also called the store-and-forward model. In an active network, a packet can contain a set of programs along with the data. The network node provides a programmable environment. It can load a program and process the packet data according to the processed result. This kind of packet routing is also called store-compute-and-forward.

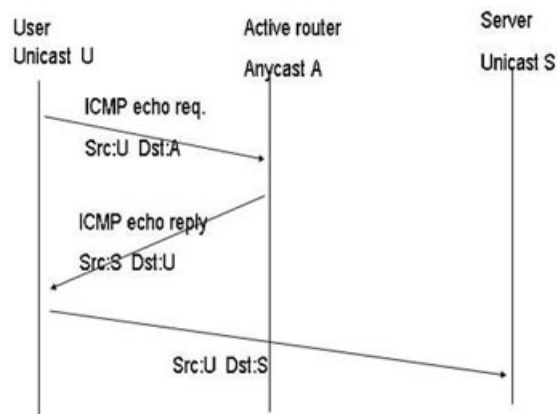


Figure 3. Anycast address mapper

There are two types of implementations proposed for active network technology: programmable switches and capsules. In programmable switches, users inject their custom process into the required routers and the customized process deals with each IP datagram going through the active routers. In capsules, every message contains a program that could be executed at an active router. Here we adopt programmable switches.

As shown in Figure 1, it should be noted that only the nearest router for the server acts as the active router and the others do not.

### —Anycast Address Mapper<sup>9</sup>—

Since an anycast address is what the user uses to find an active router to connect to a server, the mapping mechanism is required to make a mapping from an anycast address to its corresponding unicast address. We could use the modified anycast address mapper mechanism (as shown in Figure 3) to establish the connection between the user and server.

The process of the anycast address mapper is shown below:

1. User sends ICMP<sup>10</sup> ECHO request packet to an anycast address A.
2. An active router receives the ICMP ECHO request with the anycast address A.
3. The active router chooses an appropriate server with unicast address S according to the load information and fuzzy logic algorithm.

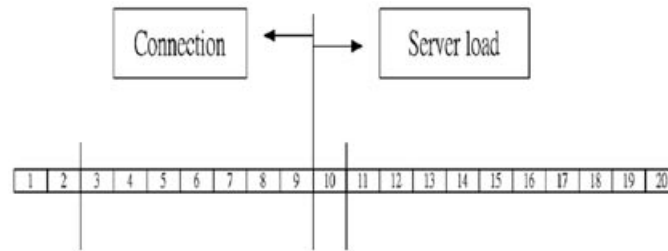


Figure 4. Definition of the flow label field in the IPv6 header used for the load information

4. The active router sends ICMP ECHO reply with the unicast address S as a source address to the user client.
5. After the user receives the ICMP ECHO reply packet, it changes the destination address from A to S.
6. The user then establishes a connection to the server with the unicast address S.

The advantage of this method is that only the application needs to be modified instead of the TCP/IP protocol. Clearly the protocol modification will be more complicated than the application.

---

***The advantage of this method is that only the application needs to be modified instead of the TCP/IP protocol.***

---

The IPv6 anycast mechanism is only used to find the nearest active routers in our proposed framework. Although the implementation of anycast is important, it is outside the scope of this article.

### —Load Information Retrieved by Active Router—

In our architecture, the load balance ability is achieved by the use of fuzzy logic in the active router. Detailed information will be described later under 'Fuzzy Logic Algorithm'. The fuzzy logic algorithm determines load balance using the load information. The load information is the connection count and data traffic for each server.

At first, the server must record its own connection and data traffic. If a server sends information to the active router frequently, it will result in a

large load for the whole network. As shown in Figure 4, the load information of a server can be represented by the flow label field of the IPv6 header. When a packet goes through the active router, it can retrieve the load information from the flow label field and use the fuzzy logic algorithm to determine the load status of each server.

The 20-bit flow label field is used to store connection and data traffic. The first two bits store the single digit, tens digit, hundreds digit and thousands digit, and the next seven bits (from the third to the ninth) are used to represent values 0 to 100. These nine bits are used to store the connection number. The 10th bit stores the unit of the server load (0 stands for KB and 1 stands for MB), and bits 11 to 20 are used to store the value of the server load.

### —Message Exchange between Active Routers—

As illustrated in Figure 1, we can see that the load information of servers 1 to 3 can be retrieved by active router 1. Active router 2 can retrieve the load information of servers 4, 5 and 6. At the same time, active router 2 would not know the load information of servers 1, 2 or 3. Similarly, neither would active router 1 know the load information for servers 4, 5 or server 6. The load information should be exchanged within all servers to achieve a good load balance.

During every interval of time, the active router actively delivers the load information collected from each server to other active routers.

### —Fuzzy Logic Algorithm—

The reason why we choose a fuzzy logic algorithm to perform load balance is that fuzzy logic

has the characteristic of inheriting experience. Assume that there is a course selection system that originally operated in the application layer and we would like to place the load balance function in the IP layer because of the IPv6 anycast address advantage. The experience in the application layer, such as how many students entered the system to select a given course at the same time and the equipment capacity required for that traffic load, is all inherited by the fuzzy logic algorithm. This experience helps data to be easily transferred from the application layer to the IP layer.

---

***The reason why we choose a fuzzy logic algorithm to perform load balance is that fuzzy logic has the characteristic of inheriting experience.***

---

There is another advantage for using the fuzzy logic algorithm. After the active router retrieves the load information, it can determine the server load using the current load information. If the active router did not receive any load information from the server, it would be decided that this server is non-existent. Therefore, if any of these servers failed, no load information would be generated. By the use of the fuzzy logic algorithm, the active router can immediately know that this server does not exist, so that the load balance function will not be affected.

The fuzzy logic algorithm is composed of the following three steps.

**Step 1**—We employed the triangular membership function and the fuzzy set to define the input data as illustrated below:

1. Three connection ranks: low (L), medium (M), high (H) (Figure 5).
2. Three server load ranks: low (L), medium (M), high (H) (Figure 6).
3. The fuzzy set for output data uses the server status and is shown below. There are five different grades applicable to the server: very low (VL), medium low (ML), medium (M), medium high (MH), very high (VH) (Figure 7).

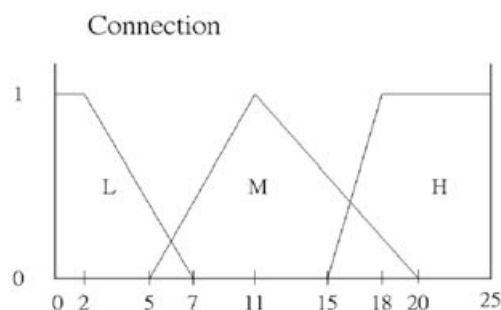


Figure 5. Input membership graph for connection

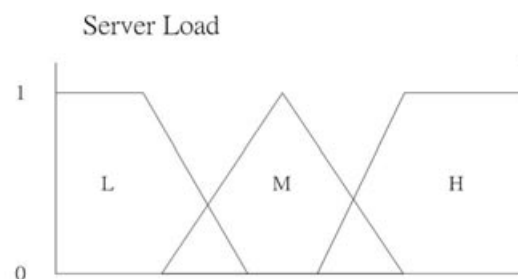


Figure 6. Input membership graph for server load

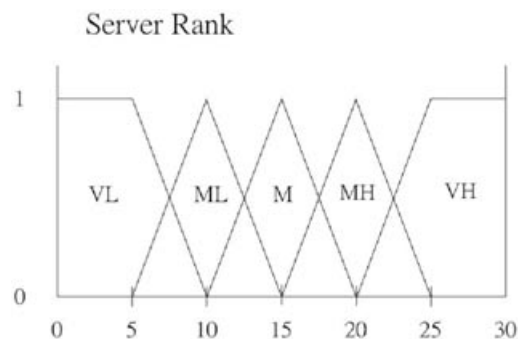


Figure 7. Output membership graph for server rank

**Step 2. Fuzzy rule of if-then**—The fuzzy inference rules are defined and shown as in Figure 8.

**Step 3. Defuzzication**—The centroid method for defuzzification is adopted because of its simplicity. The algorithm cannot be too complicated because it should begin to perform right after the active router retrieves the load information. The relationship is shown in the equation below:



$$y = \frac{\sum_{i=1}^n W_i B_i}{\sum_{i=1}^n W_i} \quad (1)$$

where  $y$  stands for the output of the fuzzy logic algorithm,  $W$  stands for the  $i$ th applicable grade of the fuzzy inference rule and  $B$  stands for the  $i$ th centroid point of the output membership function.

### —Simulation—

The fuzzy logic algorithm was simulated using Borland Delphi 7.0 software.

Connection/Server load	L	M	H
L	VH	MH	M
M	MH	M	ML
H	M	ML	VL

Figure 8. Fuzzy inference rules

There are circumstances under which the load balance cannot be expressed, such as cases when several users are not connected to request services, or the data traffic is too low. Also, fuzzy logic has no function if too many users are requesting service at the same time or the data traffic is too busy, i.e. every server is fully loaded. Therefore, we express load balance by generating a connection every 0.3s. Each connection generates 800 units of data traffic, and every server can process 50 units per 0.1s.

We use a random method to generate connections and data traffic in response to the 'inheriting experience' characteristic of fuzzy logic. This data is used to define the fuzzy membership function.

As shown in Figure 9, we can see that the connection load balance capacity is very poor with the random method. For example, the connection of server 3 at 39s is 0, and the connection of server 4 is 14 units. The connection of server 5 is 2 units, and the connection of server 1 is 15 units.

As shown in Figure 10, we can see that the data traffic load balance capacity of the random method is also very poor. For example, the connection of

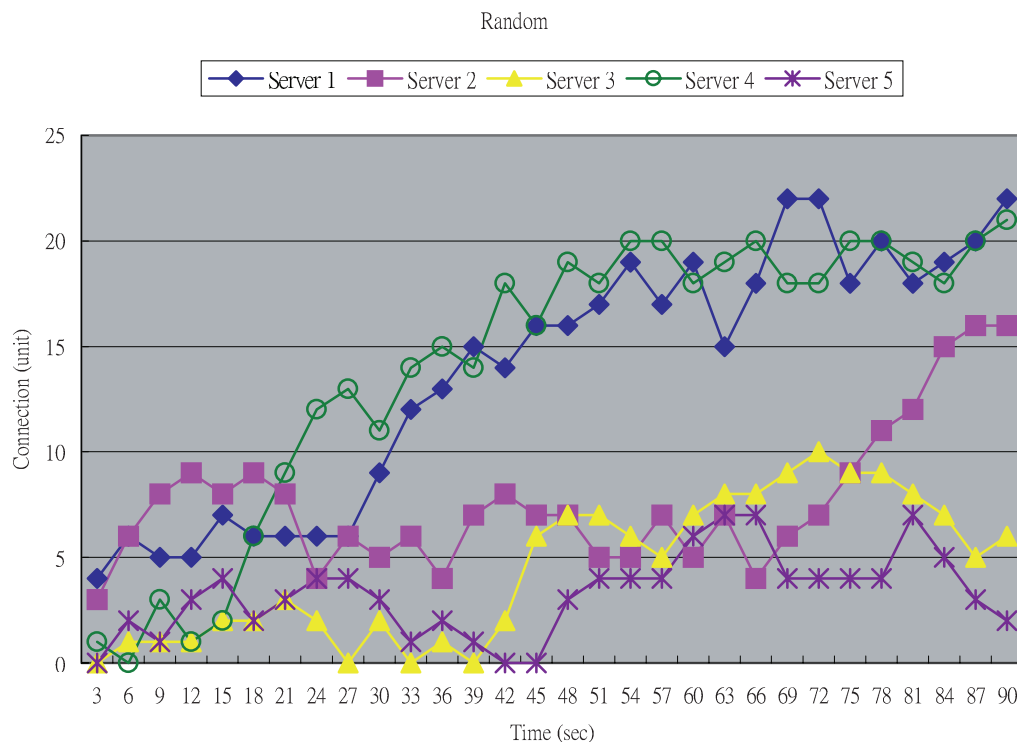


Figure 9. Load distribution of connection by random method

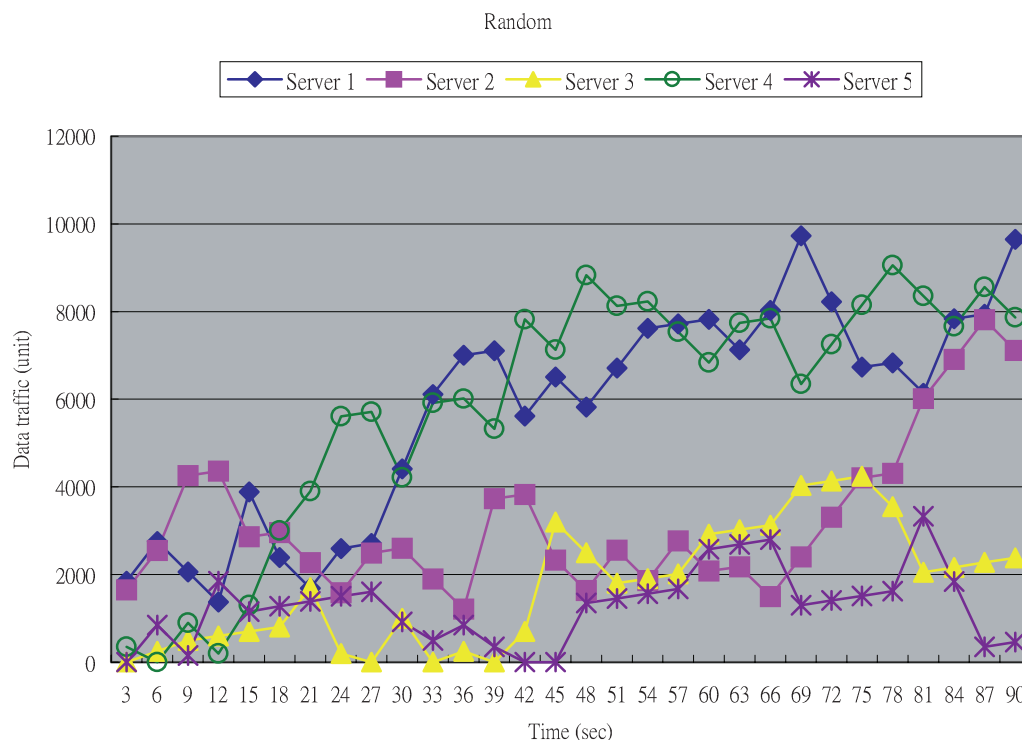


Figure 10. Data traffic load distribution using the random method

server 3 at 27s is 0, but the connection of server 4 is 5706 units. The connection of server 5 at 87s is 348 units, but the connection of server 4 is 8565 units.

Next, we use the same data traffic to generate load balancing using the fuzzy logic algorithm. Under the same conditions, we also generate a connection per 0.3s. Every connection can generate 800 units of data traffic, and each server can generate 50 units of data traffic per 0.1s.

From Figure 11, we can see that the connection generated by the fuzzy logic algorithm can be equally distributed at each server. If any server connection has a load greater than the others, the fuzzy logic algorithm will adjust it, such as server 2 at 12s. Similarly, if any connection of a certain server is less than the others, the fuzzy logic algorithm will also adjust it, such as server 3 at 81s.

As shown in Figure 12, we can see that the data traffic generated by the fuzzy logic algorithm can also be equally distributed at each server. No data traffic is concentrated at a certain server. If the data traffic for a certain server is greater than that of the

others, the fuzzy logic algorithm will adjust it, such as server 2 at 12s.

## Conclusion and Future Work

In this paper, the anycast address method was introduced and the fuzzy logic inference procedure was described. We proposed a framework using the active router based on the anycast address of IPv6 to capture the load information. The fuzzy logic algorithm was used to perform load balancing within the active router. The active router could collect the load information of service servers from the flow label of each IPv6 packet passing through it. The trade-off of this load balance framework is that the user application and the service server application should be modified and the active router should be designed. Based on the proposed framework, no additional broker server was needed for the load balance job. The IPv6 anycast mechanism would choose the suitable active router and the selected active router

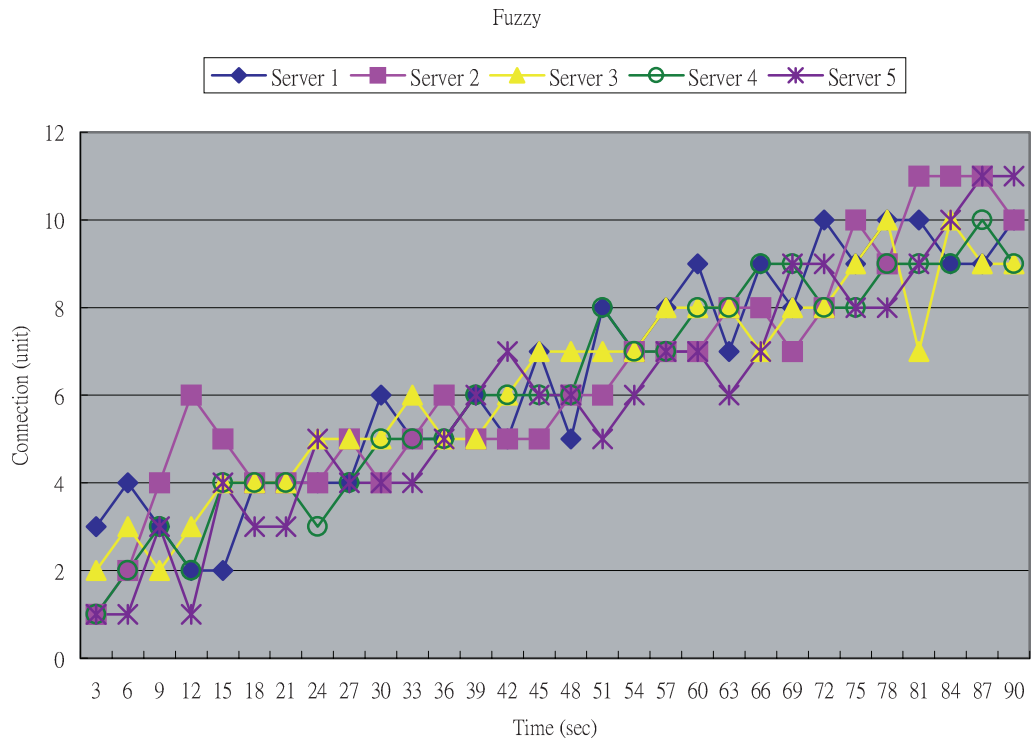


Figure 11. Fuzzy connection load distribution

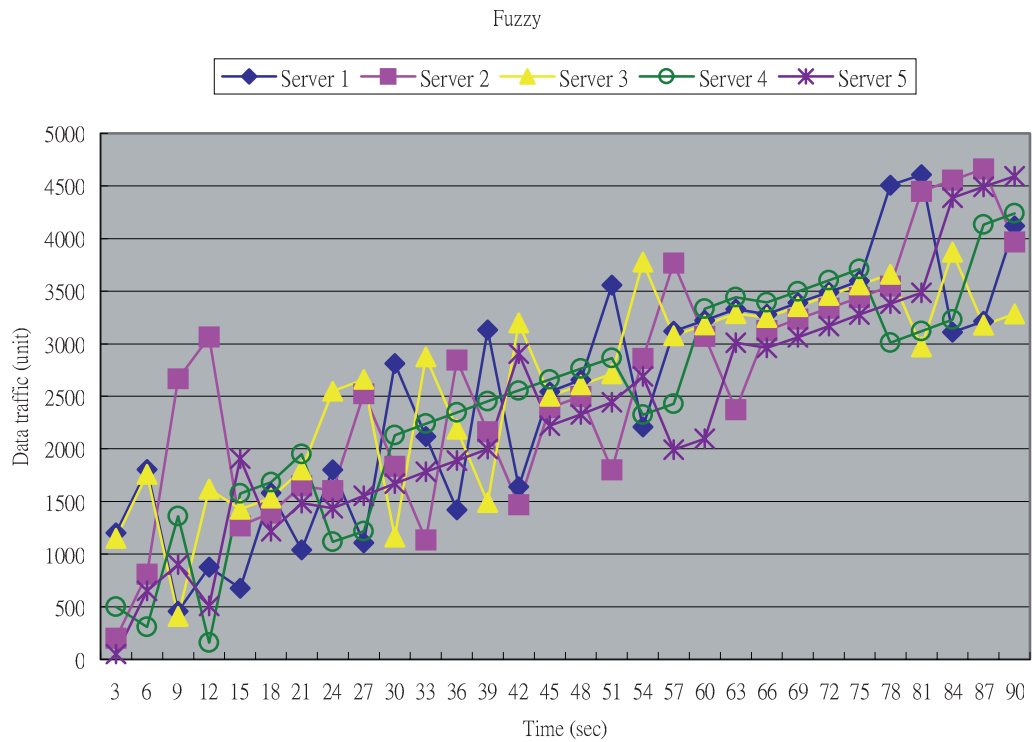


Figure 12. Load distribution of data traffic by fuzzy



would determine the appropriate service server; that is, the proposed approach functions like the traditional broker server in the application layer. Moreover, load balancing was accomplished in the IP layer. The simulation result verifies that the fuzzy logic algorithm can perform load balancing well.

As for future work, this could be addressed in the following topics:

1. Join the mobile devices for data access.
2. Enhance the security and combine with an IP layer firewall.

## Acknowledgement

This work is partially supported by the National Science Council of Taiwan, ROC, under grant number 92-2219-E-259-002.

## References

1. Hinden R, Deering S. *Internet Protocol Version 6 (IPv6) Specification*. RFC 2460, December 1998.
2. Hinden R, Deering S. *Internet Protocol Version 6 (IPv6) Addressing Architecture*. RFC 3513, April 2003.
3. Metz C. IP anycast: point-to-any point communication. *IEEE Internet Computing* 2002; 6(2): 94–98.
4. Ata S, Kitamura H, Murata M. Application of IPv6 anycasting. draft-ata-ipv6-anycast-app-00, December 2004.
5. Kaufmann A, Gupta MM. *Introduction to Fuzzy Arithmetic*. VNR: New York, 1991.
6. Hashim H, Manan JA. Active anycast method for server load balancing. In *Student Conference on Research and Development*, 16–17 July 2002; 105–108.
7. Cheung LS. A fuzzy approach to load balancing in a distributed object computing network. In *Proceedings of Cluster Computing and the Grid*, Brisbane, Australia, 15–18 May, 2001; 694–699.
8. Yamamoto M, Miura H, Nishimura K, Ikeda H. A network-supported server load balancing method: active anycast. *IEICE Transactions on Communications* 2001; E84-B(6): 1561–1568.
9. Masafumi O, Suguru Y. Implementation and evaluation of IPv6 anycast. In *Proceedings of INET 2000*, Yokohama, Japan, 18–21 July 2000 (available online: <http://www.isoc.org/inet2000/cdproceedings/>).
10. Conta A, Deering S. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*. RFC2463, December 1998. ■

**If you wish to order reprints for this or any other articles in the *International Journal of Network Management*, please see the Special Reprint Instructions inside the front cover.**