Pseudo MIPv6 for Anycast and Fault Tolerant Services

Ting-Yun Chi
Department of Electrical Engineering,
National Dong Hwa University, Hualien,
louk@louk.dyndns.org,

Han-Chieh Chao
Department of Electronic Engineering,
National Ilan University, I-Lan, Taiwan
hcc@niu.edu.tw

ABSTRACT

In the paper, we would like to propose a pseudo anycast mechanism based on MIPv6 (RFC3775, RFC3667)[3][4]. By our solution, we can provide the function for anycast and layer 3 faul-tolerant by MIPv6-like mechanism. We will also compare some load balance solutions with our solution [1][2]. With our solution, the potential customers (clients) can utilize the powerful service functions without any patching work.

Keywords: MobileIPv6, anycast, load balance

1. Introduction

As we see in today's life, there is more and more service work on the network. The new services and ideas are proposed everyday. If the service is not popular and the idea doesn't work, the service will withdraw from the competition – network business. On the contrary, if the people love the service and they would like to pay for it. The company will get the other problem – how to make sure every customer can access the service without difficulty. Purchasing more bandwidth seems not like a good answer to the company. Load-balance and fail-tolerance sound much like a good option.

Anycast is a revolutionary IPv6 development that replaces the IPv4 load balance method. It takes the IPv4 load balance into the IP layer and provides a universal load balance standard. In other words, anycast is a visionary development on Ipv6. Anycast try to provide a simple mechanism for choose the best server. It's quite simple and easy to implement. Because IPv6 is a whole new protocol, so writing a new API for anycast purpose should not be a problem.

Currently, many Ipv6-related technologies have adopted the anycast mechanism for optimal search routing including the Dynamic Home Agent Address Discovery (DHAAD) and micro handover.[5]

Unfortunately, anycast still is a dream in the real world. Although it's quite simple, but no one has written the API for it yet. Most of the manufacturers only implement the unicast and

multicast for IPv6 and most of the Operation Systems don't provide the anycast API as well.

Only a few people use anycast in the experimental network. Some of the implementation need to patch the layer 3 or layer 4 and some of them even require to patch the application layer (layer 7). Keep in mind that the goal of anycast is to provide an "IP technology" solution to save the problem at the beginning.

IPv6 has quickly developed and correlated many operating systems to support IPv6. This development has extended to Mobile IP with its own RFC3775 and RFC3776 standard. Conversely, the pace for IPv6 anycast development was relatively slower.

The anycast protocol dictates that the Host creates a link to connect an anycast address in return for unicast site through the router. This is quite similar to the Mobile mechanism in which the CN starts a link to the home address and the HA returns a real connection to the CoA. Taking advantage of the available Mobile IP standard in the following three aspects realizes anycast development.

1. Mobile IP has already been standardized.
2. The amount of software in Mobile IP can be used directly on the nodes.
3. Mobile IP has taken CN support into consideration.

We want to provide a load balance and fault- tolerant service with the pseudo MIPv6 API.

2. Related Work

This session will introcude the main load-balance solutions over the world and the latest implementation for anycast and an anycast working group. A similar idea for use MIPv6 to

provide the anycast service will also presented.

There are many load balancing solutions. One is client oriented requiring each client to have special demand software for one special application [7]. Another method is DNS oriented [8]. It is simple and easy to deploy but less sensitive. The most famous method is NAT [9]. NAT is a popular IP layer solution, but it has a bottleneck problem.

[Table 1] Comparison between the famous load balance mechanisms

Name	DNS	LS-NAT MacNAT	App-client
Good	Easy to use	Easy to use	powerful
weakness	Refresh time.	bottle	specially designated
reconnection	no	yes	yes

.

2.1 DNS

Early work on distribution and assignment of incoming connections across a cluster of servers has relied on Round-Robin DNS (RR-DNS) to distribute incoming connections across a cluster of servers. This is done by providing a mapping from a single host name to multiple IP addresses. Due to DNS protocol intricacies (e.g. DNS caching and invalidation), RR-DNS was found to be of limited value for the purposes of load balancing and fault tolerance of scalable Web server clusters. The research described in quantifies these limitations.

2.2 NAT-tunnel based

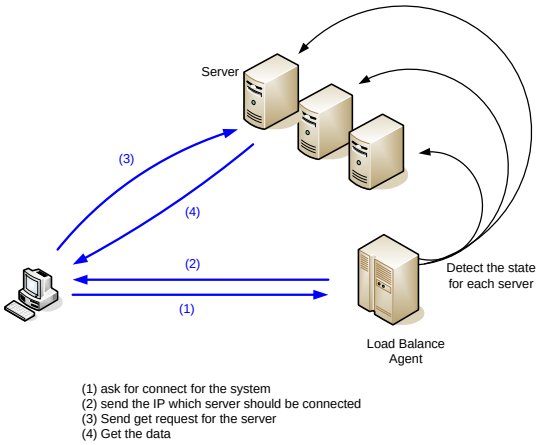
In the usual case (i.e., a non-clustered server), there is only one Web server serving the requests addressed to one hostname or Internet Protocol (IP) address. With a cluster-based server, several back-end Web servers cooperatively serve the requests addressed to the hostname or IP address corresponding to the: company's Web site. All of these servers provide the same content. The content is either replicated on each machine's local disk or shared on a network file system. Each request destined for that hostname or IP address will be distributed, based on load-sharing algorithms, to one back-end server within the cluster and served by that server. The distribution is realized by either a software module running on a common operating system or by a special-purpose hardware device plugged into the network. In either case, we refer to this entity as the 'dispatcher'. Busy sites such as Excite, Inc. depend heavily on clustering technologies to handle a large number of requests. There are two different kinds of cluster-based Web servers clustering technologies. The first is LSMAC, in which the dispatcher forwards packets by controlling Medium Access Control (MAC) addresses. The second is LSNAT, in which the dispatcher distributes packets by modifying IP addresses.

[Table 2] Compare with the LSMAC and LSNAT

Comparison of key feature of the LSMAC and LSNAT implementations		
Feature	LSMAC	LSNAT
OSI layer	L2	L3
Traffic Flow through dispatcher	Unidirectional	Bidirectional
Incoming Packet Modification	No	Des. IP address and checksum
Outgoing Packet Modification	Not applicable	Source. IP address and checksum
Routing table change in immediate router	Yes	No
Servers in different LANs	Requires interface on each LANs	Allowed

2.3 Application-client base

In the Figure1, you have three main components. The first one is Load Balance Agent, the second one is service server and the last one is client.

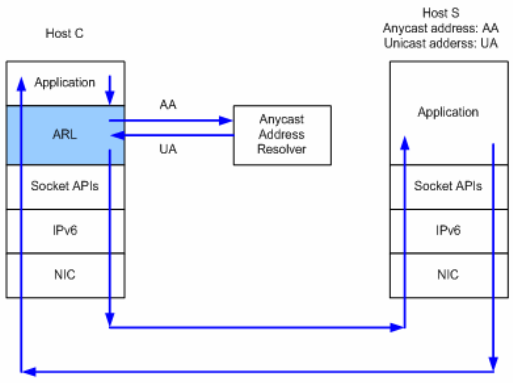


[Figure 1] The process for application based mechanism

The load balance Agent will get the CPU loading and other parameters. The agent will process the data to decide which server should be connected. When the client wants to use the service, the user must install the special application software first. The special client software will send signal to ask Load Balance Agent which server should be connected.
The process looks like "load balance DNS based". However you can understand from the figure 1, it's in the application layer.

2.4 Anycast

Several solutions have been used for anycast implementation. The first solution is the "Source Identification Option" which was proposed in an Internet Draft published in 1996 [10]. The second solution is "Anycast Address Mapper". Both solutions were implemented in the paper cited in [10]. Anycast Resolving Layer (ARL) [12] (Figure2) is a new IETF draft that uses a sub layer to resolve the anycast address issue.

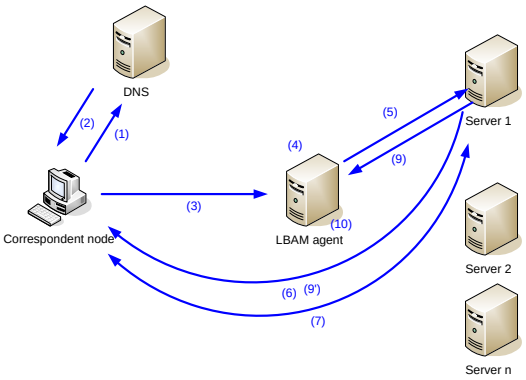


[Figure 2] Protocol Stack of AARP

We can go to Practical Anycasting.com [13] to get more information on recent anycast work. People discuss how to implement and deploy anycast service in the real world on this website. Users can post mail in the maillist to encourage interested parties to join an anycast work group involved in deploying anycast service. It was said the "Currently, IPv6 Anycast is used only in limited areas for limited purposes. It is a pity that IPv6 Anycast is not widely used. This situation should be changed". [14]

2.5 LBAM

The development of a whole new API for anycast is needed. Figure3 has a similar idea for using pseudo mobile IP for anycast, but this is not enough. We can do something more, late. The drawback of LBAM (Pseudo Anycast) is "it can't work with RFC3775."



[Figure 3] The process for LBAM

3. Pseudo MIPv6 for Anycast and Fault Tolerant Services

3.1 Trigger signal

Because we use the MIPv6 function in our solution, so we can ask the client reconnect to another server – IP layer Hot Swap. For the trigger signal, we try to use the natural signal in MIPv6 to re-initial the connection.[19]

MobileIPv6 don't provide reconnection signal for load balance naturally. However, we find two messages can be the trigger signal for our idea. The first one is icmpv6 destination unreachable message. The second one is sent to the binding message to pretend the MN return home. If you can't find it, try to search the error condition in RFC, I have said "MobileIPv6 don't provide reconnection signal for load balance

originally".

3.1.1 ICMPv6 Destination Unreachable-Address

If the binding update sent by the mobile node to a correspondent node is dropped from the network, the correspondent node continues to send packets to the mobile node's previous care-of address based on the contents of its current outdated binding cache entry. The packets are forwarded to the previous foreign link and the router on the previous foreign link attempts to deliver them. If the previous foreign link router still considers the mobile node reachable on the previous foreign link, packets are forwarded to the mobile node's link layer address. Because the mobile node is no longer attached to the previous foreign link, the packets are dropped.
The methods for correcting this error condition are as the following:
The mobile node, after not receiving a binding acknowledgment from the correspondent node, retransmits a binding update. The correspondent node receives the retransmitted binding update and its binding cache is updated with the mobile node's new care-of address.
The previous foreign link router uses neighbor unreachability detection to determine that the mobile node is no longer attached to the previous foreign link. For a point-to-point link such as a wireless connection, the unreachability of the mobile node is indicated immediately by the lack of a wireless signal from the mobile node. For a broadcast link such as an Ethernet segment, the entry in the previous foreign link router's neighbor cache goes through the REACHABLE, STALE, DELAY, and PROBE states. After the neighbor cache entry for the mobile node is removed, attempts to deliver to the mobile node's previous care-of address are unsuccessful and the previous *foreign link router will send an ICMPv6 Destination Unreachable-Address Unreachable message to the correspondent node. Upon receiving this message, the correspondent node will remove the entry for the mobile node from its binding cache and communication resumes as described in the "A New Correspondent Node Communicates with a Mobile Node"*.

3.1.2 Returning Home

When the mobile node attaches to its home link after being away from home, it must perform the following functions:

Send a binding update to the home agent to delete the binding for the mobile node.

Inform home link nodes that the correct link-layer address for the home address is now the mobile node's link-layer address.

Send binding updates to all correspondent nodes to delete the binding for the mobile node.

- Before sending a binding update to each correspondent node mobile node's binding update list to delete the binding for the mobile node, it performs a Return Routability procedure. Since the home address and the mobile's new address are the same, *it is sufficient to exchange only the HoTI and HoT messages. The CoTI and CoT messages are not sent when the mobile returns home.*
  Because the CoA can't work anymore, RR only exchange the HOT and HOTI. In the mechanism, the Binding manage key (Kbm) only related to the Home Keygen token.

(1) $Home\_key\_gentoken = First(64, MAC(K_{cn}, \text{hom}eaddress | nonce | 0))$

$K_{bm} = SHA1(Home\_key\_gentoken)$

- The mobile node sends a binding update to each correspondent node with the care-of address set to the mobile node's home address.

In, you can see the detail packet flow and format (figure4).

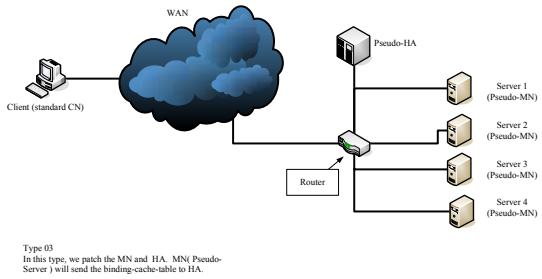
(20-2)Binding Update message format(MN->CN)

8	16	24	32
Ver=6	Traffic Class	Flow Label	
Payload Length		Next Header=135	Hop Limit
Source Address(Home Address of Mobile Node 128bit) [Home Address of Mobile Node#A]			
Destination Address(Correspondent Node Address 128bit) [Home Address of Mobile Node#B]			
Payload Proto=59	Header Len=1	MH Type=5	Reserved
Checksum		Sequence #	
A=1 H L K	Reserved	Lifetime=0	
Type=4	Length=4	Home Nonce Index	
Care-of Nonce Index		Type=5	Length
Authenticator(variable Len)			

[Figure 4] The format for BU when return home

3.2 Mechanism

The following picture (Figure5) is our topology for our scenario.

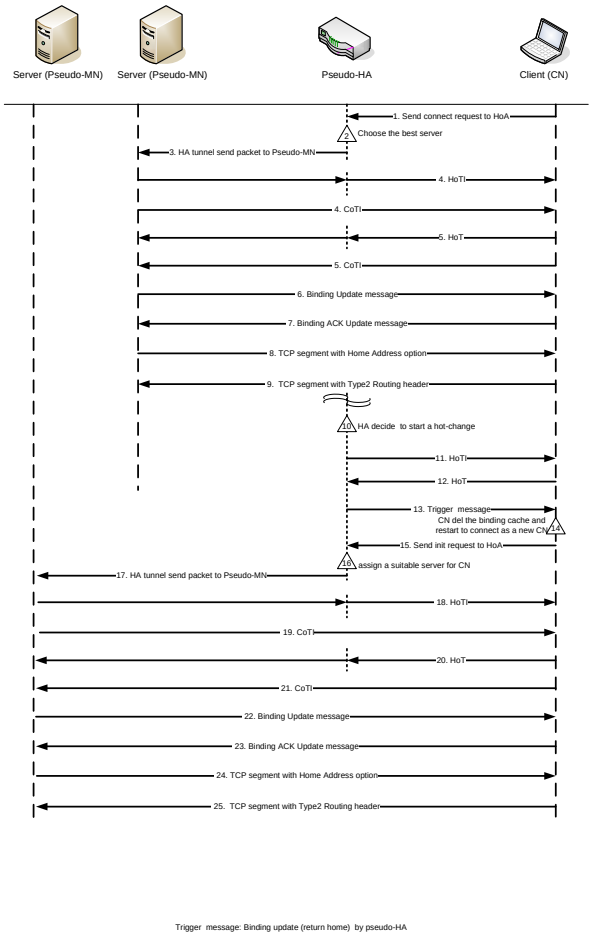


[Figure 5] The topology –patch the HA and MN for global deploy

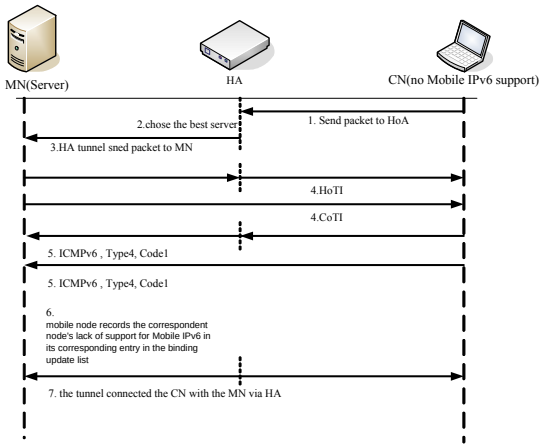
3.2.1 Connection mechanism

For the anycast function in our solution (Figure6), when a CN wants to connect with a server, the flowchart is nearly the same as that in the RFC3775 regulation except that a modified second step is involved. When the client begins to connect with HA, HA will choose the best MN (server) to execute the standard MobileIP linking process. After patching the HA will cheat on the CN and MN. The HA will remember many different Index tables while the CN do not need a patch to support our idea. (figure1). The pseudo-MN will periodically sent the home-token to pseudo-HA for IP-Hot-Swap, late.

RFC3775 also explains what should be done if the CN does not support MobileIPv6. If MobileIP cannot identify HOTI and COTI that means that the CN does not support MobileIP. The CN will send a response with ICMPv6 Bad Parameter-Unrecognized. Next the Header Type Encountered (ICMPv6 Type 4, Code 1) message. Upon receipt of the ICMPv6 message, the mobile node and HomeAgent will record all CNs that do not support MobileIPv6. The Home Agent will act as a relay to forward the data flow between the MN and CN. (figure7)



[Figure 6] The whole process for topology type3



[Figure 7]: CN (does not support Mobile IPv6) wants to connect server with TCP

At first, when servers (pseudo-MN) try to register with the Home Agent, all of the servers will share the same HoA. A server will not send special register but a binding update.

3.2.2 The rules for choosing a server

In order not to modify RFC3775 to achieve the Anycast function, we will not ask servers to send an application message to the Home Agent (ex: CPU load, Memory usage…etc). We use the RTT to determine the linking quality, priority and weight for each server. Using RTT we can also identify if the server is alive or dead.

The two proposed methods are：

(1) HomeAgent will periodically send a ICMPv6 request to each server. (Send to ff02::1 or using the Mobile Node-list). This is simple and useful.

(2) The HA will send a short life-time ICMPv6 Home Prefix Advertisement message regularly. The MN will send an ICMPv6 Home Prefix Solicitation message back after receiving.

We will obtain the same effect regardless which method is used. However, the linking quality may be incorrect under the instant message approach, so we put the result into a smoothed RTT equation [15] ：

$$SRTT = \alpha * SRTT + (1-\alpha) * RTT$$

RTT represents the time the reaction information was received after setting. α to 0.9. The linking quality smoothness number is thereby calculated

Citing reference [16], we can find parameters to determine the network state. The request is sent in interval H and is calculated using

$$(3)\ H = DefaultInterval \times (1+\delta)$$

Where the Default Interval is a constant and δ is a random value uniformly distributed between -0.5 and 0.5 to represent the fluctuation in the computer or network load.

The equation for calculating the priority and determining which server should be connected is acquired from [20]. The SRTT for each server is acquired first. The priority is then calculated using

$$ServerPrioiry[i] = \frac{\frac{1}{SRTT[i]}}{\sum \frac{1}{SRTT[i]}} \quad (4)$$

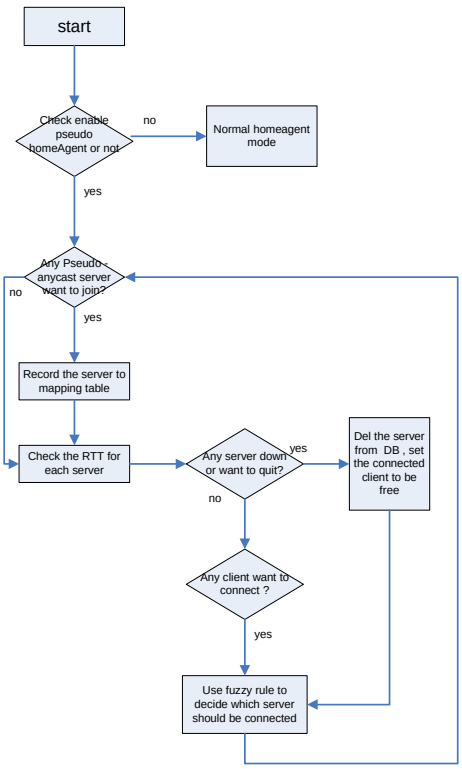
Each time the Home Agent receives an anycast connect request form the CN, the HomeAgent will randomly generate a number (form 0~1). For example, if we have four servers and the server priority is 0.3, 0.2, 0.35 and 0.15, the server priority list with will be.

Server1 (0~0.3)
Server2 (0.3~0.5)
Server3 (0.5~0.85)
Server4 (0.85~1.0)

If the randomly generated number is 0.7, the Home Agent will connect with Server3.The following is our pseudo-code. And you can see the flowchart in Figure8.

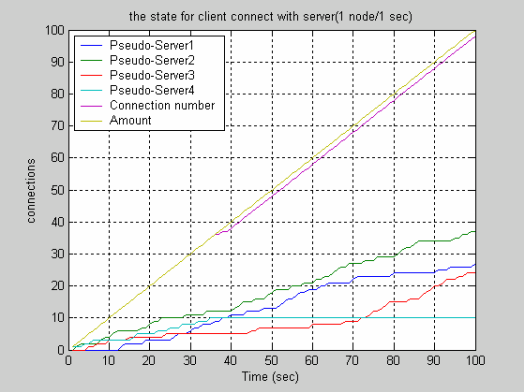
```
#for HomeAgent
Get binding message
{
If (address is anycast address)
{
Add DB (anycast address)(server list)
/*Pay attention, the server sends the normal binding message, but the HomeAddress is anycast address.*/
}
Otherwise
{
Run for normal MobileIP
}
}
```

```
Get connect message form CN
{
If (the dest is anycast address)
{
X=rand (1);
Connect
server=which_serverDB(anycast,x)
}
Otherwise
{
Run for normal MobileIP
}
}

Thread
{
SRTT DB (anycast)
For (i=0, DBend, i++)
{
ServerPriority(anycast)

#Find the server is down (serverdown parameter)
If the server priority <0.1
Send BU to CN to change server
}
}
```



[Figure 8]: Decide which server should be connected

In the following picture (Figure9), you can see the effect for the detect mechanism. When one of the server crashes down, the last clients will connect with the other servers.



[Figure 9]: Choose the server by detect mechanism (one server stop to provide service)

3.2.3 For the fault tolerant

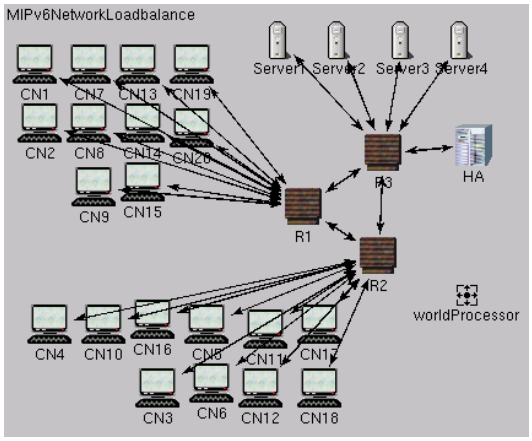
If you ask me "Your idea is special?" before this session, I will say "no, the idea is quite normal and you can find someone have the better idea or mechanism". However, it the first time the people can redirect the traffic in the IP layer by our idea. Most of the load balance ideas request the Application layer solution or NAT-tunnel mapping. Almost all of them can't avoid the bottle problem

When the pseudo-HA detect one of the server crash, it will send the trigger signal to each clients to reconnect with the other server.

4. Simulation and Results

4.1 Environment

We used OMNet++ to simulate the environments for our idea. In our configuration, we have 200 CNs and four servers. The server provides the UDP service and each CN ask 320kbps for UDP service. The figure10 shows the topology



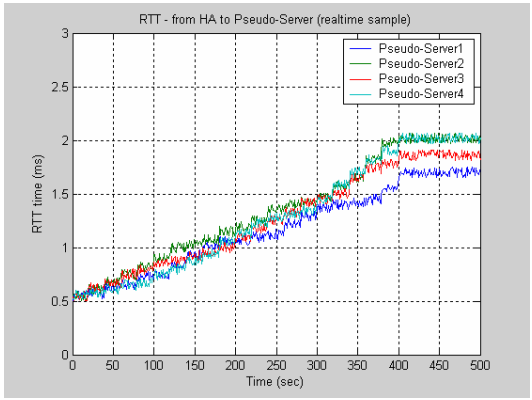
[Figure 10]: Simulation Topology

The simulation was run for 400(20*20) seconds, The Default Interval was set at 20sec and timeout is set to 20sec. CN will begin to send the connect requests at 20sec one by one. (Join interval 20sec)

4.2 Result

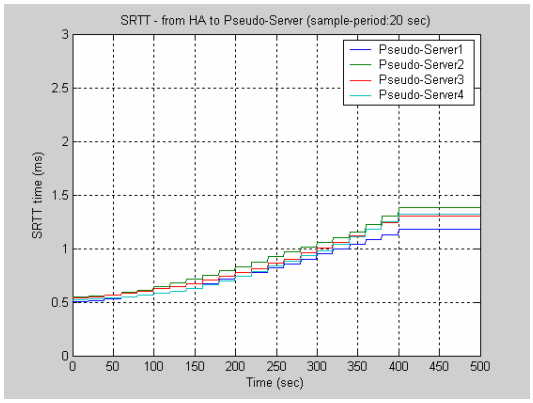
4.2.1 Load balance

Figure 11 shows the RTT from HA to each server change at every second. This will be used to set the load balance parameter later.



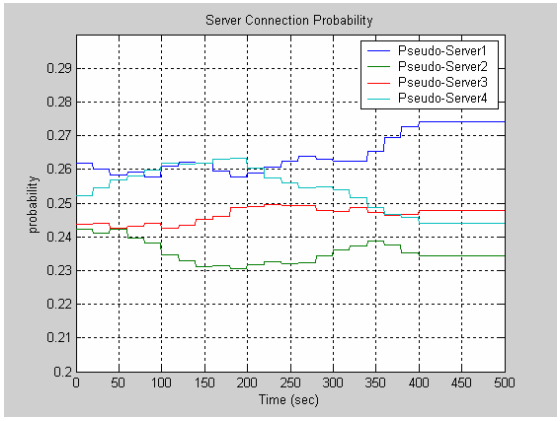
[Figure 11]: RTT from pseudo-HA to pseudo-Server

From Figures 11, you can see that RTT cannot be used directly as a parameter to choose the connect server because the RTT variation is so huge. Figures 12 demonstrate the variation result after using the SRTT process.



[Figure 12]: SRTT from pseudo-HA to pseudo-Server (sample)

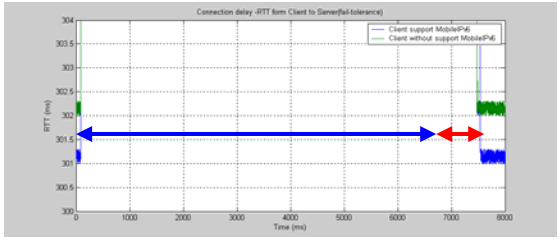
The following picture (figure 13) shows the Server connection probability. You can see the probability is quite equal, so that's meaning our mechanism work.



[Figure 13]: Server connection probability

Figure14 shows the time for fault tolerant. In our idea, the reconnect time is close to the time that detects the server crash down. The main delay case by the server delay time.

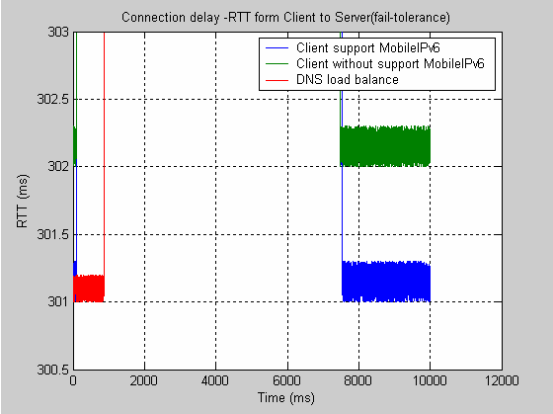
4.2.2 Fault Tolerant



[Figure 14]: RTT form CN pseudo-Server

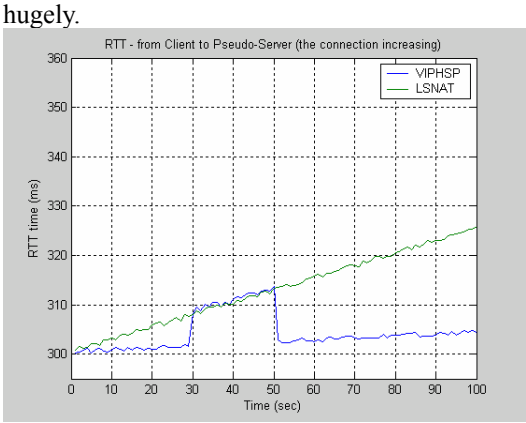
$$(5)\quad \begin{aligned} delay &= rand(sample - period) \\ &+ (time\_to\_detect\_the\_fail) \\ &+ (handover\_time) \end{aligned}$$

In this section, we will compare the proposed mechanism with the other load balance solutions. The first one is DNS load balance. It's an easy and simple solution, however it don't support the reconnection. In Figure15, you will notice the DNS can't recover the layer3 connection to get the service



[Figure 15]: RTT form CN pseudo-Server

Now, let's compare with the VIPHSP and LSNAT. In Figure16, we can see LSNAT have the bottle problem. If it takes a long time to measure, the service delay time will increase hugely.



[Figure 16]: Compare RTT with VIPHSP and LSNAT

5. Future Work

There are still numerous interesting issues for pseudo-anycast.

- Global deployment.
- Security issue.
- Find the suitable service for anycast.

With our solution, we can reduce the traffic on HA using the RO (Routing optima). However, we still cannot determine which server is the nearest for the CN. This causes a serious problem for global service. Maybe GIA [17] can give us some ideas to solve this problem.

Security is always the big issue for nowadays network. We should do something more to protect the pseudo-HA from hack or attack.

Finding suitable service is the most important issue. Anycast does not support handovers to other servers during the connection phase. With our mechanism, all of the application will support load balance and fault tolerant. We should take sometime to discuss the issue.

6. Conclusion

Ipv6 developed quickly and correlates many operating systems to support IPv6. The pace of anycast development for IPv6 was relatively slower. Our proposal can utilize the MobileIPv6 and its API to implement a load balance solution. Because of the simple approach used in measuring the linking quality, we can reduce the system resource requirement to choose which server is better. We have also made a workable IP-layer load balance standard with MobilIPv6.

By our idea, we can see it really can work in the real world. Of course, you can write a whole new protocol stack for anycast or IP layer reconnection. But we know that, patch whole world's computer is impossible. In our idea, we provide a common solution to save the problem. We don't write the whole new protocol by our self and try to use the exist protocol.

In our idea, the server (or HomeAgent) can send the handover message to the client. In this method we can ask clients to change their connection to other servers to balance the traffic between servers. In another example, the Home Agent may force the CN (if it does not support MIPv6) to redirect the traffic to the nearest server for the Home Agent (If the CN does not support MibileIPv6, the Home Agent will use a tunnel to forward the data flow between the CN and MN [server]. The Home Agent should then tunnel to the nearest server to save the backbone bandwidth.)

7. Acknowledgement

This work is a partial result of project no NSC 93-2219-E-259-001- conducted by National Dong Hwa University under the sponsorship of the National Science Council, Taiwan, ROC

References

[1]. Luling, R.; Monien, B.; Ramme, F., “Load balancing in large networks: a comparative study”, Parallel and Distributed Processing, 1991. Proceedings of the Third IEEE Symposium on, 2-5 Dec. 1991.

[2]. Yanjun Feng; Chuck Song; Wanming Luo; Runguo Ye, **“**Load balancing based on pseudo-anycast and pseudo-mobility in IPv6**”,**Communication Technology Proceedings, 2003. ICCT 2003. International Conference on , Volume: 1 , 9-11 April 2003.

[3]. D. Johnson, C. Perkins and J. Arkko,” Mobility Support in IPv6 “, RFC3775, June 2004.

[4]. F. Dupont, V. Devarapalli and J. Arkko,” Mobility Support in IPv6 “, RFC3776, June 2004.

[5]. Microsoft Corporation,**”** Understanding Mobile IPv6”, Published: April 2004 Updated: June 2004

[6]. Dudas, I.; Bokor, L.; Bilek, G.; Imre, S.; Szabo, S.; Jeney, G.; “Examining anycast address supported mobility management using mobile IPv6 testbed ” Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean , Volume: 2 , 12-15 May 2004 Pages:555 - 558 Vol.2

[7]. Cherkasova, L.; “FLEX: load balancing and management strategy for scalable Web hosting service” Computers and Communications, 2000. Proceedings. ISCC 2000. Fifth IEEE Symposium on , 3-6 July 2000 Pages:8 – 13

[8]. Cardellini, V.; Colajanni, M.; Yu, P.S.” Redirection algorithms for load sharing in distributed Web-server systems”;Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on , 31 May-4 June 1999 Pages:528 – 535

[9]. Yanjun Feng; Runguo Ye; Chuck Song; Jian Ma; Yu Wu; “Load balance and fault tolerance in NAT-PT” Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on , Volume: 3 , 15-18 Dec. 2003 Pages:1957 - 1961 vol.3

[10]. J. Bound and P. Roque, "IPv6 Anycasting Service: Minimum requirements for end nodes," draft-bound-anycast-00.txt (EXPIRED), August 1996.

[11]. Masafumi OE ;Suguru YAMAGUCHI Nara Institute of Science and Technology Japan , ”Implementation and Evaluation of IPv6 Anycast”

[12]. S. Ata ,Osaka City University; H. Kitamura, NEC Corporation; M. Murata, Osaka University ; ”draft-ata-ipv6-anycast-resolving-02.txt” 27-Oct-2004.

[13]. http://anycast.anarg.jp/ is a web site for IPv6 Anycast discussion

[14]. http://www1.ietf.org/mail-archive/web/ipv6/current/msg03801.html

[15]. Defense Advanced Research Projects Agency Information Processing Techniques Office. Transmission Control Protocol Darpa Internet Program Protocol Specification. IETF RFC793, September 1981..

[16]. Kashihara, S.; Nishiyama, T.; Iida, K.; Koga, H.; Kadobayashi, Y.; Yamaguchi, S.;” Path selection using active measurement in multi-homed wireless networks” Applications and the Internet, 2004. Proceedings. 2004 International Symposium on , 2004 Pages:273 – 276

[17]. Yokota, H.; Kimura, S.; Ebihara , Y.; “A proposal of DNS-based adaptive load balancing method for mirror server systems and its implementation” Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on , Volume: 2 , 2004 Pages:208 – 213

[18]. Ting-Yun Chi ,Han-Chieh Chao and T. G. Tsuei , Improved Mobile IP based Pseudo-anycast , ICACT 2005.

[19]. Microsoft Corporation,” Understanding Mobile IPv6”, Published: April 2004 Updated: June 2004

[20]. http://citeseer.ist.psu.edu/392863.htm