Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 24, 2009

Wanming. Luo
X. Lee
Wei. Mao
CNNIC
Mei. Wang
Cisco Systems

OCTOBER 21, 2008

Load Balancing based on IPv6 Anycast and pseudo-Mobility
draft-luo-v6ops-6man-shim6-lbam-00.txt

Status of this Memo

Abstract

Load balancing is a key factor for both IPv4 to IPv6 transition mechnisms, e.g.NAT-PT or Tunnel broker, and Multihoming to improve their scalability and Robustness. In fact, that is a method, by which IP packet can be distributed across a pool of servers, instead of directing to a single server.Load balancing has been widely used by NAT, Web service and FTP service. However, current load balancing software and implementations have problems such as poor scalability, inability to balance session flow, long latency time and topological constraint on server pool.

This document describes a method using pseudo-anycast and pseudo-mobility based on Mobile IPv6 to implement load balancing in session level in IPv6 network, by which those problems above can be solved. Futhermore, this method only need little modification to Mobile IPv6 in the servers' and agent's side; as for the general users, it need not any modification.

Table of Contents

1. Introduction

The need for Load balancing arises when a single server is not able to cope with increasing demand for multiple sessions simultaneously. Clearly, load balancing across multiple servers would enhance responsiveness and scale well with session load. Both IPv4 to IPv6 transition mechnisms, e.g.NAT-PT,CGN(Carrier Grade NAT), NAT64 or Tunnel broker, and Multihoming require some kind of load balancing.

Load balancing is not new and goes back many years. A variety of techniques, including software, hardware and both of them, were applied to address it and each of them presents interesting advantages but also solves different problems than the LBAM, or pose drawbacks not present in the LBAM:

DNS Support for Load Balancing [RFC1794]is a simple mechanism to balance session flow. The problem with this approach is its long latency time, moreover, modifications to the widely deployed DNS protocol can introduce instability. This is not an acceptable option;

Load Balancing using IP Network Address Translation (LSNAT) [RFC2391] inherently has the problems with Network Address Translation (NAT) [RFC3022] , including topological constrain on the server pool, the long latency time caused by several translations, triangle routing, and breaking the end-to-end significance intelligence [RFC3439] (That is, some application, such as FTP, H.323 and SIP, can not pass through the LSNAT router), moreover, the LSNAT itself might be a bottle-neck if the bindings of session flow are too many;

Some layer 4 switch of Cisco and Extreme, can provide load balancing based on hardware, and the performance is much better than software, however, this approach is poor at scalability and has topological constrain.

LBAM is an alternative approach using pseudo-anycast and pseudo-Mobility in IPv6 without those problems above. LBAM has two kinds of entity: agent and the pool of servers, and all of those entities have a same anycast address. When a client attempts to access a server in the pool, the agent selects a server in the pool based on load balancing algorithm and redirects the request to that server by pseudo-Mobility. In this view, the agent is functionally similar to Home Agent in Mobile IPv6[RFC3775]. The client will directly access the selected server during the session. Subsequently, the client will continue to interact with this server until the binding expires or the client receives a binding update. LBAM poses no restriction on the organization and rearrangement of servers in the pool. Server may be added and deleted easily, no matter where it is.

The draft does not need any changes to Mobile IPv6 in CN, and a little modification on Mobile IPv6 in the Server and LBA. The draft is structured as follows. Section 2 provides terminology and concepts used. Section 3 overviews the method. Section 4 briefly details the how to implement LBAM based on mobility support in IPv6; Section 5, 6 and 7 describe operation of LBAM agents, correspondent nodes and servers. Section outlines security consideration. The Appendix describes several load balancing algorithms.

1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] .

2. Terminology

Load balance Load balance for the purpose of this document is defined as the spread of session load amongst a cluster of servers, which are functionally similar or the same. In other words, each of the servers in cluster can support a client session equally well with no discernible difference in functionality. Once a server is assigned to service a session, that session is bound to that server till termination. Sessions are not allowed to swap between servers in the midst of session.

Start of session for TCP, UDP and others The first packet of every TCP session tries to establish a session and contains connection startup information. The first packet of a TCP session may be recognized by the presence of SYN bit and absence of ACK bit in the TCP flags. All TCP packets, with the exception of the first packet must have the ACK bit set. However, there is no deterministic way of recognizing the start of a UDP session or any other non-TCP session.

End of session for TCP, UDP and others The end of a TCP session is detected when FIN is acknowledged by both halves of the session or when either half receives RST bit in TCP flags field. Within a short period (say, a couple of seconds) after one of the session partners sets RST bit, the session can be safely assumed to have been terminated.

For all other types of session, there is no deterministic way of determining the end of session unless you know the application protocol. Many heuristic approaches are used to terminate sessions. Another way to handle session terminations is to timestamp sessions and keep them as long as possible and retire the longest idle session when it becomes necessary.

Pseudo-anycast An anycast address assigned to many nodes. However, only one server (say node A) takes part in the routing system, the others are "dumb" in routing system. That is a simple packet destined to the pseudo-anycast will reach the node A, rather than the others. To reach one (say node B) of the other nodes which have their own global uncast addresses, as well as the "dumb" pseudo-anycast address, an IPv6 Routing header [RFC4861] must be used in the packet (its pseudo-anycast address is the packet's destination, while its global unicast address is the intermediate).

In fact, a pseudo-anycast address is corresponding to a kind of service.

Load Balance Address A pseudo-anycast address assigned to both of LBAM and the servers in the pool. We will call it LBA address as following. From the perspective of LBAM Agent, LBA address is its global unicast address, while from the perspective of all the servers in the pool, the LBA address just is a "dumb" address that functions as home address, and they have their own global exclusive unicast address that functions as care-of address.

Server's unicast address In LBAM, this global unicast address is the care-of address for difference server, that is, all the servers in the pool have their own global unicast address (care-of address, CoA) besides the same anycast one (pseudo-anycast address) assigned to them. This address is functionally similar to care-of address in Mobile IPv6.

Server pool A set of servers that are functionally similar or the same. They have at least two kinds of address: the global exclusive unicast address and the LBA address.

Pseudo-mobility To implement LBAM, the LBAM Agent will tunnel a packets destined to the LBA address to the care-of address of one selected server, as if the very server changed its point, even through the very server is stationary in the same link (In the extreme case, all the servers stationary at the same link permanently). We call the false mobility as Pseudo-Mobility. When a server deployed, an entry for the LBAM Agent SHOULD be added in its Binding Update List, and then, Bind Update message SHOULD be sent out according to the Binding Update List. Or if the server is really away from it former point, the server will send out the Bind Update message according to the Binding Update List.

LBAM Agent A node intercepts packets destined to the LBA address, encapsulates them, and tunnels them to the mobile node's registered care-of address. Indeed, it is functionally similar to Home Agent in Mobile IPv6. LBAM Agent may be a general host, rather than must be a

router, as well there is only one LBAM Agent, rather than many Home Agent in Mobile IPv6.

LBAM Agent does nothing except balancing special load. The Agent may take charge in balancing several kinds of service, if it is assigned with several pseudo-anycast addresses.

Extended Binding Cache In Mobile IPv6, the Home Agent only has one binding entry for a home address in the Binding Cache. While in LBAM, the LBAM Agent may have as many binding entry as the number of the servers in the pool; each one entry corresponds to a server.

Load Balancing History Cache A cache to log all load balancing history for each server and the correspondent nodes the server is servicing

This document is based on Mobile Ipv6, and for other terms appeared in this document please refers to Mobile IPv6[RFC3775].

3. Overview

3.1. Overview of Mobility Support in IPv6

Each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a care-of address, which provides information about the mobile node's current location. IPv6 packets addressed to a mobile node's home address are encapsulated and tunneled transparently to its care-of address [RFC2473].

When a mobile node receives a packet tunneled to it from its home agent, the mobile node uses that as an indication that the original sending correspondent node has no Binding Cache entry for the mobile node, since the correspondent node would otherwise have sent the packet directly to the mobile node using a Routing header, or when away from an area, a mobile node will send "Bind Update" message to those very nodes according to its "Binding Update List". After receiving the Bind Update message, those correspondent nodes will change its "Binding Cache" in response to the message. From then on, the node uses an IPv6 Routing header [RFC4861] (instead of IPv6 encapsulation) to route the packet to the mobile node by way of the care-of address indicated in this binding cache for the mobile node.

Mobile IPv6 defines one additional IPv6 destination option. When a mobile node sends a packet while away from home, it will generally set the Source Address in the packet's IPv6 header to one of its

current care-of addresses, and will also include a "Home Address" destination option in the packet, giving the mobile node's home address. By using the care-of address as the Source Address in the IPv6 header, with the mobile node's home address instead in the Home Address option, the packet will be able to safely pass through any router implementing ingress filtering [RFC2460] .

3.2. Overview of LBAM

Every server in the server pool has a unicast address called pseudo care-of-address, named A1, ..., Ai, ..., An. The LBAM agent and these servers have the same address called LBA address. These servers have the MN's function and the LBAM Agent has the Home Agent's function according to the Mobile IPv6 [RFC3775].

Firstly, when the CN sends initial IP packet to the LBA address, the LBAM agent receives this packet and tunnels this packet to a server (saying Server-i) by some load algorithm using IPv6 encapsulation [RFC2473].

Then, when the Server-i receives this tunneled packet, it will send Binding Update message to the CN. After receiving Binding Update message, CN will send the following packets to the server-i directly.

Before sending any packet, the sending node SHOULD examine its Binding Cache for an entry for the destination address to which the packet is being sent. If the sending node has a Binding Cache entry for this address, the sending node SHOULD use Type 2 Routing header to route the packet to this server (the destination node) by way of the care-of address in the binding recorded in that Binding Cache entry.

If a new server will join this server pool, it sends a Binding Update message to LBAM Agent for "Home" registration, to tell the LBAM its unicast address.

Figure 1 gives a scenario of this method.

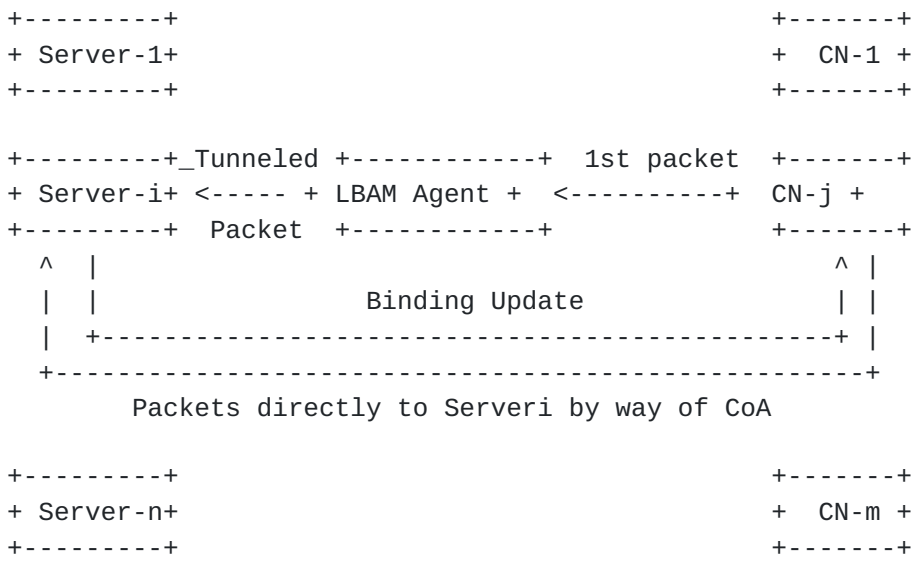


Figure 1 A scenario of the LBAM

4. Modification of MIPv6 in Servers and LBAM Agent

Both of the LBAM Agent and the servers MUST modify their Mobile IPv6 little as follows. Note that except those modifications followed, the other parts of LBAM Agent and servers are same as Mobile IPv6. So, it is to easy to implement LBAM based on Mobile IPv6.

4.1. Modifying Binding Cache in LBAM Agent

LBAM Agent needs some modifications including Binding Cache Agent and how to process Binding Update message received in LBAM Agent. Every IPv6 node with mobility support has a Binding Cache, which is used to send original packets by correspondent nodes (CN)or to tunnel intercepted packet destined to the mobile node. In Mobile IPv6, each Binding Cache only contain one entry for a mobile node, no matter the node is a correspondent node or Home Agent. In LBAM Agent, we redefine and name it as Extended Binding Cache, in which each entry conceptually contains the following fields:

LBA address which acts as the home address of a server in the pool. If the destination address of the packet matches one or more LBA address in the Binding Cache entry, one entry selected by load balancing algorithm, SHOULD be used in routing that packet.

The global unicast addresses of the servers in the pool, which act as care-of address in Mobile IPv6. If the destination address of a packet intercepted by the agent matches the LBA address of the LBAM

Agent, the packet SHOULD be tunneled to the unicast address of an entry selected from several entries by load balancing algorithm. Combination of both fields above is used as the key for searching the Binding Cache for the destination address of a packet being sent.

- The maximum value of the Sequence Number field received in previous Binding Updates for this mobile node home address. The Sequence Number field is 8 bits long, and all comparisons between Sequence Number values MUST be performed modulo 2**8. For example, using an implementation in the C programming language, a Sequence Number value A is greater than another Sequence Number value B if ((char)((a) - (b)) > 0), if the "int" data type is a 8-bit signed integer.

- The Binding Security Association (BSA) to be used when authenticating Binding Updates that are received for this Binding Cache entry.

- The Binding Security Association (BSA) to be used when calculating authentication data for inclusion in Binding Acknowledgements in response to Binding Updates that are received for this Binding Cache entry.

An entry for a server in the LBAM Agent's Extend Binding Cache SHOULD NOT be deleted by the agent, until the agent receives a Binding Update message from the server in which the lifetime is zero, or the agent finds out the server has died or can not offers service.

We can find out that there is little difference with Binding Cache in Mobile IPv6, except the four fields at foreside; the rest fields are same as their corresponding ones in Binding Cache of MIPv6.

4.2. Adding Load Balancing History Cache in LBAM Agent

The LBAM Agent SHOULD add Load Balancing History Cache to log all the load balancing history for each server and the correspondent nodes that the server is servicing. Each entry of Load Balancing History Cache has those fields as follows:

- LBA address;

- The global unicast address of a server in the pool. Combination of both fields above is used as the key to index;

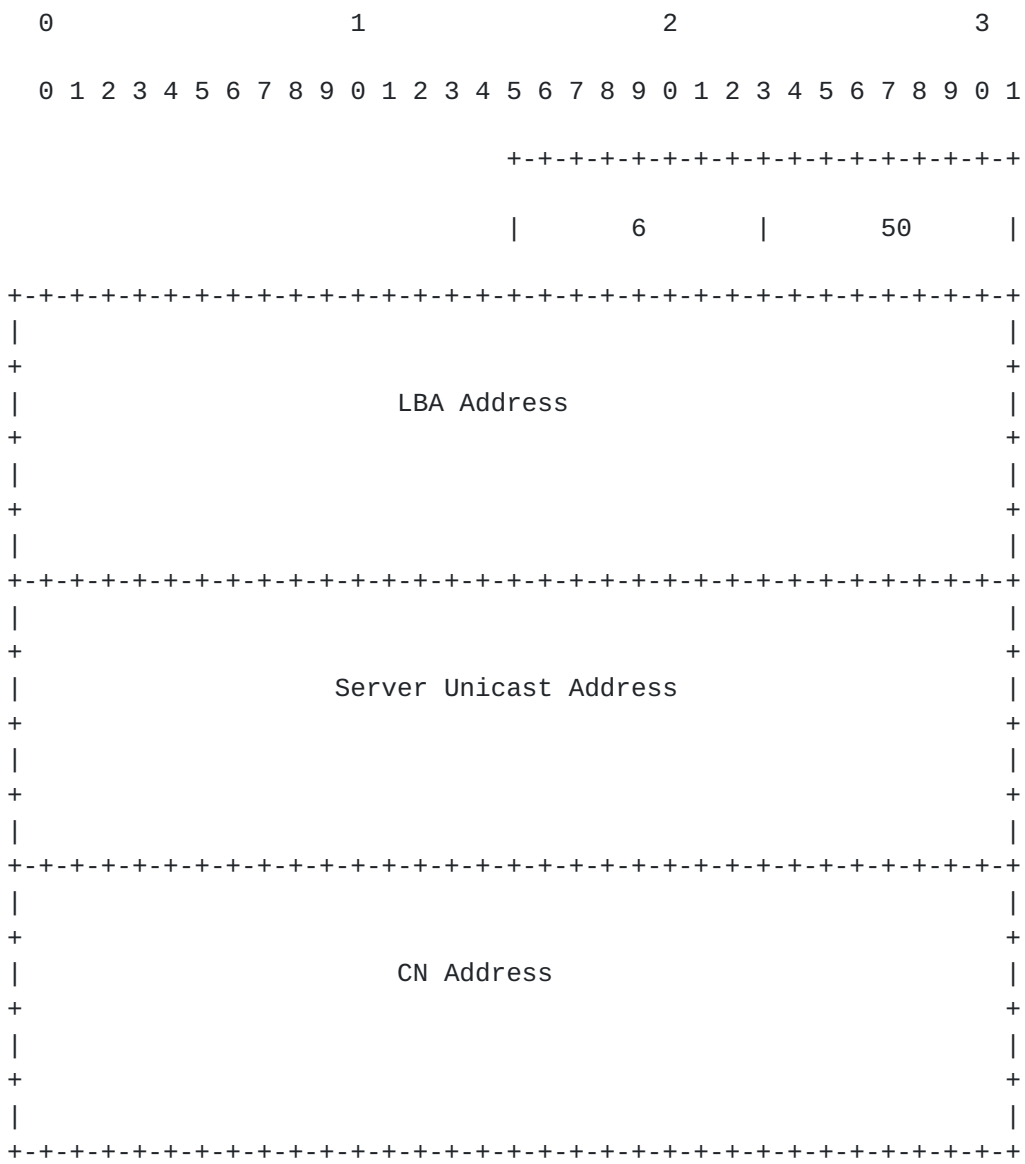
- Correspondent node list, which includes all the correspondent nodes directed to the very server;

Firstly, the Load Balancing History Cache MAY be used in load balancing algorithm. Secondly, if the LBAM Agent finds out a sever

has died, it can send out Binding Update message to all the Correspondent Nodes binded to the server immediately, by which those Correspondent Nodes can get out the "black-hole " (the dead server) quickly, rather than have to wait until their very binding to the server time out.

4.3. Adding No Active Session Notice parameter

No Active Session Notice parameter (alignment requirement: 8n+6)



The No Active Session Notice Header parameter is valid only in the Binding Update message sent from servers to LBAM Agent.

The No Active Session Notice parameter contains 3 addresses (tri-tuple): LBA address, Server Unicast address and CN address which are used by LBAM Agent to index and delete the corresponding entries in Loading Balancing History Cache and Extended Binding Cache.

5. Operation of LBAM Agent

5.1. LBAM Agent's process to tunnel the start of session

To redirect the start of session, the LBAM Agent MUST attempt to intercept packets on the LBA address, and MUST tunnel each intercepted packet to a server selected by load balancing algorithm using IPv6 encapsulation [RFC2473].

In order to intercept such packets on the home link, when a node begins serving as the LBAM Agent for a service (or say for a pool of servers), the LBAM Agent MUST receive all "pure" start packet of a session destined to the LBA address (Simply, a pure packet is a packet without Routing header). It is very easy to implement the process above, because it is the LBAM Agent who takes part in the routing system in the name of LBA address; so all the "pure" packet will reach the LBAM Agent.

After intercepting the start packet of a session, the LBAM Agent MUST browse the Load Balancing History Cache, to see whether there is an entry according to the combination key of the correspondent's source address and the LBA address. If yes, that is the correspondent node ever requested a server and the binding between it and the server has been timeout, so the LBAM Agent Must delete the previous entry and selected a new server using load balancing algorithm, and then create an entry in the Load Balancing History Cache for the correspondent node; if not, the LBAM Agent just creates an entry in the Load Balancing History Cache for the correspondent node.

In order to forward each intercepted packet to a selected server, the LBAM Agent MUST tunnel the packet to the selected server using IPv6 encapsulation [RFC2473], the tunnel entry point node is the LBAM Agent, and the tunnel exit point node is the selected server's global unicast address extracted from Extended Binding Cache. When the LBAM Agent encapsulates an intercepted packet for forwarding to the server, the agent sets the Source Address in the prepended tunnel IP header to the LBAM agent's own IP address (the LBA address), and sets the Destination Address in the tunnel IP header to the server's uncast address. When received by the server (using its unicast

address), normal processing of the tunnel header in [RFC2473] will result in decapsulation and processing of the original packet by the server.

5.2. LBAM Agent receiving Binding Update message

From the perspective of other nodes except the LBAM Agent, the process for received Binding Update message is same as in Mobile IPv6. Only LBAM Agent processes received Binding message differently against Mobile IPv6, as follows (see [RFC3775]; for reference):

Before accepting a Binding Update option received in any packet, the receiving node MUST validate the Binding Update according to the following tests:

- The packet meets the specific authentication requirements for Binding Updates;

- The packet MUST contain a Home Address option;

- The Option Length field in the Binding Update option is greater than or equal to the length specified in of [RFC3775];

- The Sequence Number field in the Binding Update option is greater than the Sequence Number received in the previous Binding Update for this home address, if any. As noted in [RFC3775], this Sequence Number comparison MUST be performed modulo 2**8.

If the server sends a sequence number which is not greater than the sequence number from the last successful Binding Update, then the LBAM Agent MUST send back a Binding Acknowledgement with status code 141, and the last accepted sequence number in the Sequence Number field of the Binding Acknowledgement.

Any Binding Update which fails to satisfy all of these tests for any other reason (than insufficiency of the Sequence Number) MUST be silently ignored, and the packet carrying the Binding Update MUST be discarded.

If the Binding Update is valid according to the tests above, then the Binding Update is processed further as follows:

- If there is not the entry for the care-of address extracted from the message, and the Lifetime is not zero, then create a entry, in which LBA address is extracted from the Home option in the received Binding Update message, and the global care-of address is equal to the source address of the received Binding Update message (the server's global unicast address);

- If there is the entry for the care-of address and LBA address extracted from the message, and the Lifetime specified in the Binding Update is not zero, then ignore this Binding Update message;

- If the Lifetime specified in the Binding Update is zero or the specified Care-of Address included in the Alternate Care-of address sub-option matches the LBA address in the Home option, then this is a request to delete entry. In other word, this server sent this message would not offer service now.

- If LBAM Agent receives a Binding Update message carrying No Active Session Notice parameters, it will delete the two matched entries in Load Balancing History Cache and Extended Binding Cache according to the tri-tuple retrieved from the Binding Update message.

5.3. Detecting dead host and announcing to CNs

As sessions are assigned to servers, it is important to detect the live-ness of the servers. Otherwise, sessions could simply be black-holed into a dead server. Many heuristic approaches are adopted. Meanwhile, LBAM Agent sending pings periodically would be one way to determine the live-ness. When the LBAM Agent detects a dead server, it should send Binding Update message according to its Load Balancing History Cache to relative correspondent nodes to reset the binding with the dead server. Meanwhile, the agent should delete the entry for the server in its Extended Binding Cache.

5.4. Selecting the best server for a start session

Certainly, the LBAM SHOULD select a best server for a start session. The algorithms are described in appendix. The Load Balancing History Cache and the Extended Binding Cache MAY be a reference for selecting.

6. Operation of correspondent nodes

First of all, the node should support Mobile IPv6, then load balancing and LBAM is transparent to it.

Before sending any packet, the sending node SHOULD examine its Binding Cache for an entry for the destination address to which the packet is being sent. If the sending node has a Binding Cache entry for this address, the sending node SHOULD use a Routing header to route the packet to this server (the destination node) by way of the care-of address (the server's global unicast address) in the binding recorded in that Binding Cache entry.

Consequently, as long as there is a valid entry in the CN's Binding Cache for a LBA address, all the sessions originated from the same CN will be directed to the same server indicated by the valid entry. This is a constraint of LBAM.

In addition, a correspondent node with a Binding Cache entry for a server may refresh this binding, for example if the binding's lifetime is near expiration, by sending a Binding Request to the mobile node. Normally, a correspondent node will only refresh a Binding Cache entry in this way if it is actively communicating with the server and has indications, such as an open TCP connection to the server, that it will continue this communication in the future [RFC3775]. After receiving a responding Binding Update message from the server, the correspondent node will refresh this binding.

7. Operation of servers

First of all, when being deployed, all servers MUST send out Binding Update message to the LBAM Agent, so that LBAM Agent can register the binding in its Extended Binding Cache. This process is same as [RFC3775].

When receiving a tunneled packet by LBAM Agent, the server uses that as an indication that the original sending correspondent node has no Binding Cache entry for it, since the correspondent node would otherwise have directly sent the packet to the server using a Routing header. So, the server will send out a Binding Update to the correspondent node; this process is same as [RFC3775].

When receiving a packet with Routing Header, a server will process the Routing header as follows:

- The server swaps the Destination Address in the packet's IPv6 header and the Address specified in the Routing header. This results in the packet's IP Destination Address being set to the server's LBA address.

- The server then resubmits the packet to its IPv6 module for further processing, "looping back" the packet inside the server. Since the server recognizes its own LBA address as one of its current IP addresses, the packet is processed further within the server.

When a server sends a packet, it will generally set the Source Address in the packet's IPv6 header to its current care-of address, the global unicast address, and will also include a "Home Address" destination option in the packet, giving the server's LBA address. By using the care-of address as the Source Address in the IPv6

header, with the server's home address (LBA address) instead in the Home Address option, the packet will be able to safely pass through any router implementing ingress filtering [RFC2460].

Before shut down, the server SHOULD send out Binding Update message according to its Binding Update List by setting the Lifetime zero or specifying Care-of Address included in the Alternate Care-of address sub-option matches the LBA address in the Home option. Moreover, when a server receives a packet containing a Binding Request, it SHOULD return to the sender a packet containing a Binding Update. The Lifetime field in this Binding Update SHOULD be set to a new lifetime, refreshing current lifetime remaining from a previous Binding Update sent to this node. When sending this Binding Update, the server MUST update its Binding Update List in the same way as for any other Binding Update sent by the server. Please see [RFC3775] for reference.

Note: the Default Lifetime in the Binding Update message MAY be defined according to different service the server provides. For example, if the server provides multimedia service using UDP, the Lifetime should not be too long; if the server provides FTP or WWW services, the Lifetime may be much long. In some senses, a short Lifetime means that perhaps the server would receive Binding Request message frequently, while a long Lifetime means the Binding will keep for more time even through a CN does not need the server no longer. So, it is a tradeoff to define the Lifetime in Binding Update message.

At last, when a server detects no active session with some CN, it will send a Binding Update message carrying the No Active Session Notice parameters to LBAM Agent and it will delete the relative entries in its own Binding Cache and Binding Update List at the same time; meanwhile, the server sends a Binding Update message with LifeTime = 0, to tells the CN to delete corresponding entry of the sever in its Binding Cache.

Note: As said in [RFC3775], if the sender knows that the Binding Cache entry is still in active use and the remain lifetime is near expiration, it MAY send a Binding Request option to the mobile. When the mobile node receives a packet from some sender containing a Binding Request option, it returns a Binding Update option to that sender, giving its current binding and a new lifetime. So, it implies that CN MAY or MAY not send a Binding Request option to the server when the binding is near expiration. Supposed that the CN sends Binding Request when the binding is near expiration; a server can definitely know that the binding is invalid when its remaining Lifetime in Binding Update List is equal 0; otherwise, the server has to depend on other methods to detect whether there is active session

with a CN when the remaining Lifetime is near expiration. How to detect the "no active session" is out of the scope of this document.

8. Security Considerations

This document has no direct impact on Internet infrastructure security. The security of this document is dependent on the security mechanisms of Mobile IPv6.

9. Normative References

[RFC1794] Brisco, T., "DNS Support for Load Balancing", RFC 1794, April 1995.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2391] Srisuresh, P. and D. Gan, "Load Sharing using IP Network Address Translation (LSNAT)", RFC 2391, August 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC3439] Bush, R. and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", RFC 3439, December 2002.

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

Appendix A. Appendix A: Load balancing algorithms

Many algorithms are available to select a host from a pool of servers to service a new session. The load distribution is based primarily on (a) resource availability and system load on the server, and (b) cost of accessing the network on which a server resides and load on the network interface used to access the server [RFC2391].

A.1 First criterion: resource availability on the server

Ideally speaking, the selection process would have precise knowledge of real-time resource availability and system load for each host in server pool, so that the selection of host with maximum unutilized capacity would be the obvious choice. However, this is not so easy to achieve.

We consider here two kinds of heuristic approaches to monitor session load on server pool members. The first kind is where the load balance selector tracks system load on individual servers in non-intrusive way. The second kind is where the individual members actively participate in communicating with the load balance selector, notifying the selector of their load capacity.

Listed below are the most common selection algorithms adapted in the non-intrusive category.

1. Round-Robin algorithm

This is the simplest scheme, where a server is selected simply on a round robin basis, without regard to load on the host.

2. Least Load first algorithm

This is an improvement over round-robin approach, in that, the server with least number of sessions bound to it is selected to service a new session. This approach is not without its caveats. Each session is assumed to be as resource consuming as any other session, independent of the type of service the session represents and all servers in server pool are assumed to be equally resourceful.

3. Ping to find the most responsive host.

Till now, capacity of a server is determined exclusively using heuristic approaches by the LBAM Agent. In reality, it is impossible to predict system capacity from remote, without interaction with those servers. A prudent approach would be to periodically ping those servers and measure the response time to determine how busy the servers really are. Use the response time in conjunction with the

heuristics to select the server most appropriate for the new session.

A.2 Second criterion: cost of accessing server

When server nodes are distributed geographically across different areas and cost to access them vary widely, the load balance selector could use that information in selecting a server to service a new session. In order to do this, the load balance selector would need to consult the routing tables maintained by routing protocols such as RIP and OSPF, to find the cost of accessing a server from the correspondent node to a server of the pool.

To meet this criterion, the selector has to master the enough routing information to calculate the access cost. In fact, when the pool of servers distributes widely, and the correspondent node is far away from the LBAM Agent, the criterion is hard to implement.

Simply, the LBAM can choose a server with the longest prefix match to the correspondent node.

A.3 Weighted load balancing algorithm

In fact, the selection criteria would be based on the two ones above. The product of the two facts would be evaluated to select the server with least weight for each new session. Say, cost of accessing server S1 is twice as much as that of server S2. In that case, S1 will be assigned twice as much load as that of S2 during the distribution process. When a server is not accessible due to network failure, the cost of access is set to infinity and hence no further load can be assigned to that server.

As for how to assign the weight for each criterion, it is upon to what you want to get and what you afford.

Authors' Addresses

Wanming Luo
CNNIC
No.4 South 4th Street, Zhongguancun
Beijing

Phone: +86 10 58813213
Email: luowanming@cnnic.cn

Xiaodong LEE
CNNIC
No.4 South 4th Street, Zhongguancun
Beijing

Phone: +86 10 58813020
Email: lee@cnnic.cn

Wei Mao
CNNIC
No.4 South 4th Street, Zhongguancun
Beijing

Phone: +86 10 58812230
Email: mao@cnnic.cn

Mei Wang
Cisco Systems
170 West Tasman Drive
San Jose, CA,95134-1706 USA

Phone: 408-853-5842
Email: mei@cisco.com

Full Copyright Statement

Intellectual Property