Network Working Group
Internet-Draft
Intended status: Informational
Expires: 21 April 2026

N. Matsuhira
Neptela

18 October 2025

Multi-Stage Transparent Server Load Balancing
draft-matsuhira-mslb-19

Abstract

This document specifies Multi-Stage Transparent Server Load Balancing (MSLB) specification. MSLB makes server load balancing over Layer3 network without packet header change at client and server. MSLB makes server load balancing with any protocol and protocol with encryption such as IPsec ESP, SSL/TLS.

Requirements Language

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 RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 21 April 2026.

Copyright Notice

Table of Contents

1. Introduction

This document specifies Multi-Stage Transparent Server Load Balancing (MSLB) specification.

MSLB provides server load balancing function over Layer3 network without packet header change at client and server. MSLB works with any protocol and protocol with payload encryption such as IPsec ESP, SSL/TLS.

2. Traditional load balancing method

There are several load-balancing techniques, such as round-robin DNS, IP Anycasting [RFC1546] and destination address translation. Figure 1 shows a load-balancing system with a typical server load balancer with destination address translation technique.

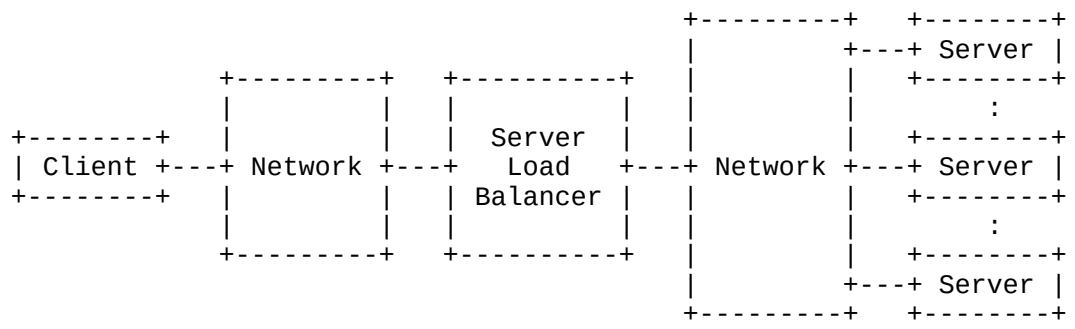

Figure 1

It is well-known that Network address translators break the internet transparency [RFC2775] and have an application dependency [RFC2993] characteristic.

Some server load balancers use application data, so with IPsec ESP, SSL/TLS, these mechanisms may not work well.

3. Architecture of MSLB

Load balancing is the technique that distributes packets to multiple servers. For packet distribution, the destination address translation technique is useful, however, this technique itself breaks internet transparency.

After distribution, if writing back to the original destination address may be possible, it is possible to recover transparency. This is the basic idea and architecture of MSLB. Figure 2 shows the architecture of MSLB.

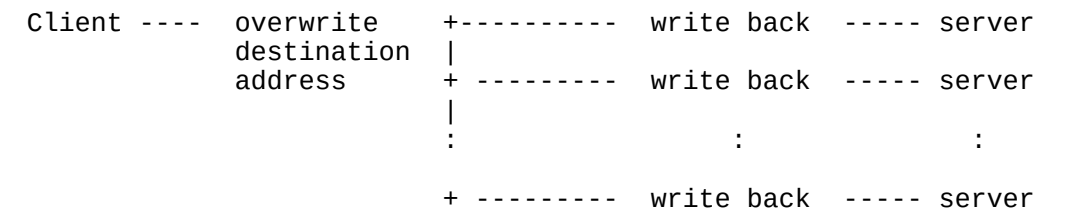

Figure 2

This method processes only the destination address of the IP header. This method can be applied to both IPv4 and IPv6.

4. configuration

4.1. basic configuration

Figure 3 shows a basic server load balancing system with MSLB. This case two-stage configuration with one MSLB-F and one-stage many MSLB-Bs.

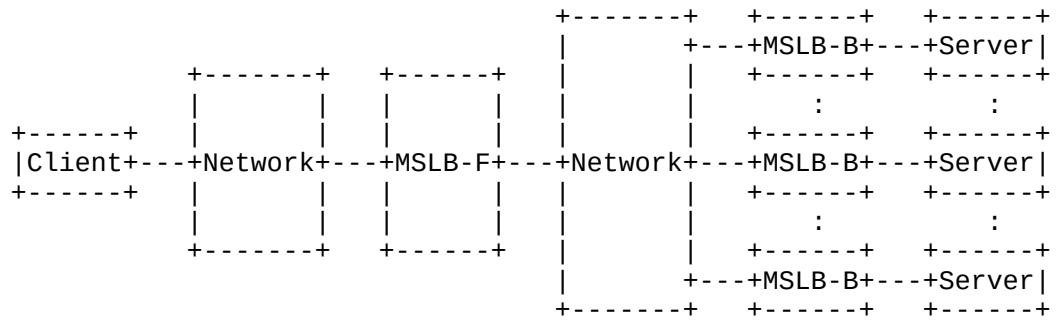

Figure 3

MSLB-F is a front function of MSLB and translates the destination address to one of the addresses of MSLB-B. BSLB-B is the backend function of MSLB and translates the destination address to the original server address, i.e. address of MSLB-F. The IP address of MSLB-F and all servers are the same value.

MSLB-F may multi-stage configuration. Figure 4 shows a three-stage configuration with two-stage MSLB-F and one-stage many MSLB-Bs.

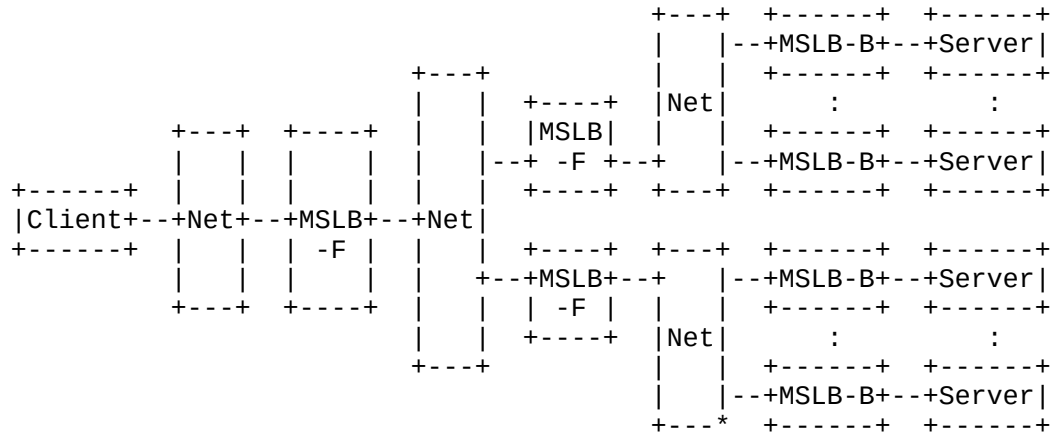

Figure 4

4.2. one arm configuration

Figure 5 shows one arm configuration of the server load balancing system with MSLB.

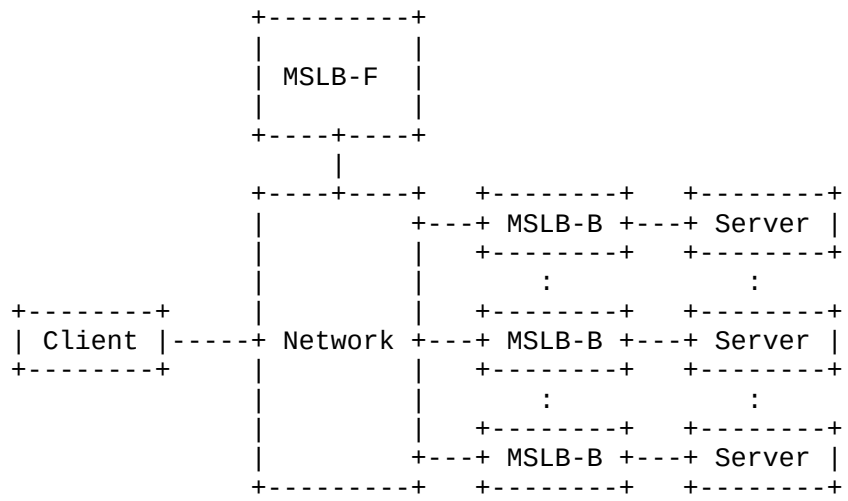

Figure 5

MSLB-F is a front function of MSLB and translates the destination address to one of the addresses of MSLB-B. BSLB-B is a backend function of MSLB and translates the destination address to the original server address, i.e. address of MSLB-F. The IP address of MSLB-F and all servers are the same value.

In this configuration, MSLB-F connects to the network with a single link, that is one arm configuration. In this case, the return packet, i.e.packet from server to client does not pass through the MSLB-F.

5. mode

MSLB has two modes, one is address translation mode, and the other is encapsulation mode.

5.1. address translation mode

This mode uses an address translation technique.

Figure Figure 6 shows packet processing with address translation mode.

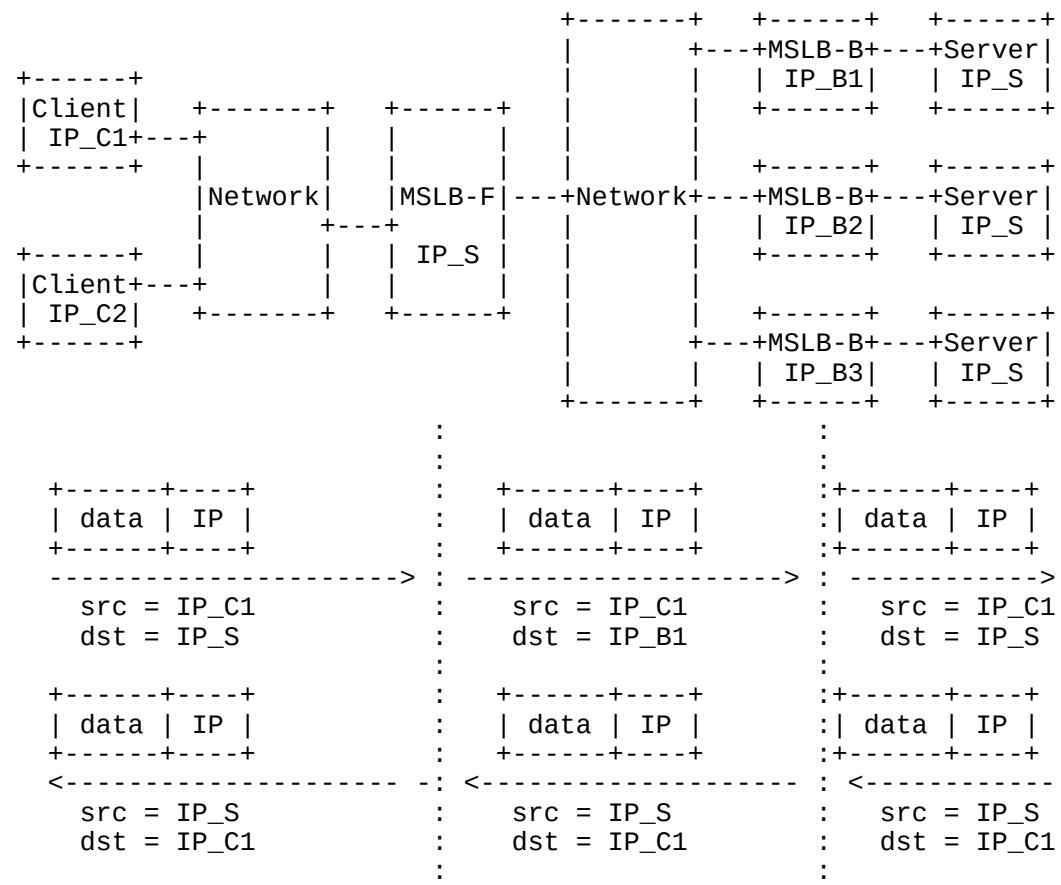

Figure 6

In this figure, to the client, the IP address is allocated IP_C1, IP_C2, and the server IP address is IP_S. In this case, IP_S is also allocated to all servers and MSLB-F. And to the MSLB-B, IP_B1, IP_B2, IP_B3 is allocated. This allocation is shown in the upper part of Figure 6.

The lower part of Figure 6 shows packet transfered between client and server. From the client to the server, only the destination address is translated, MSLB-F translates from IP_S to IP_B1, and MSLB-B translate from IP_B1 to IP_S. Then the destination address of the packet sent to the client and the destination address of the packet that receives the server are the same. That means transparency remains.

Return packet, i.e., from the server to the client is not translated, it just forwarded.

On the Internet, the client IP address and server IP address must Global IP address, however, the IP address of MSLB-B may private IP address.

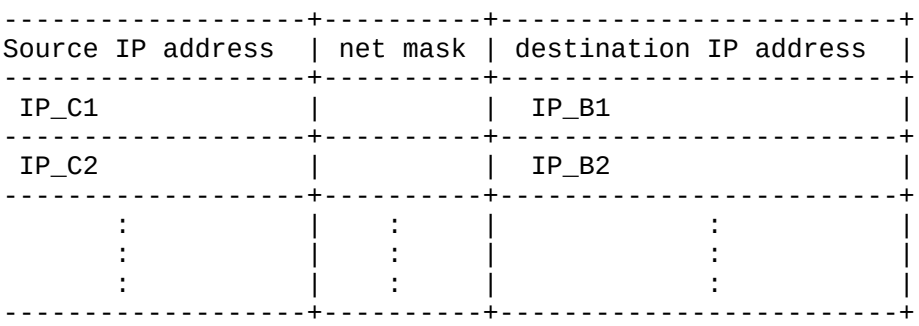

Figure 7

Figure 7 shows the MSLB table. MSLB has this table and translates the destination address using this table value. MSLB-F checks the source IP address and translates the destination address with this table.

Using IPv4-IPv6 translation may be possible, i.e., IPv4 packet translated to IPv6, then translate to IPv4 or IPv6 packet translate to IPv4, then translate IPv6 may possibleFigure 8 shows possible combination of IPv4 and IPv6. These IPv4-IPv6 translation cases will be defined in the future.

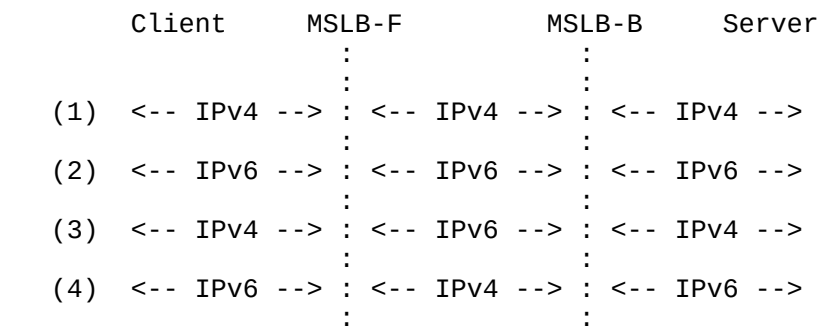

Figure 8

5.2. encapsulation mode

This mode uses an encapsulation technique.

Figure Figure 9 shows packet processing with encapsulation mode.

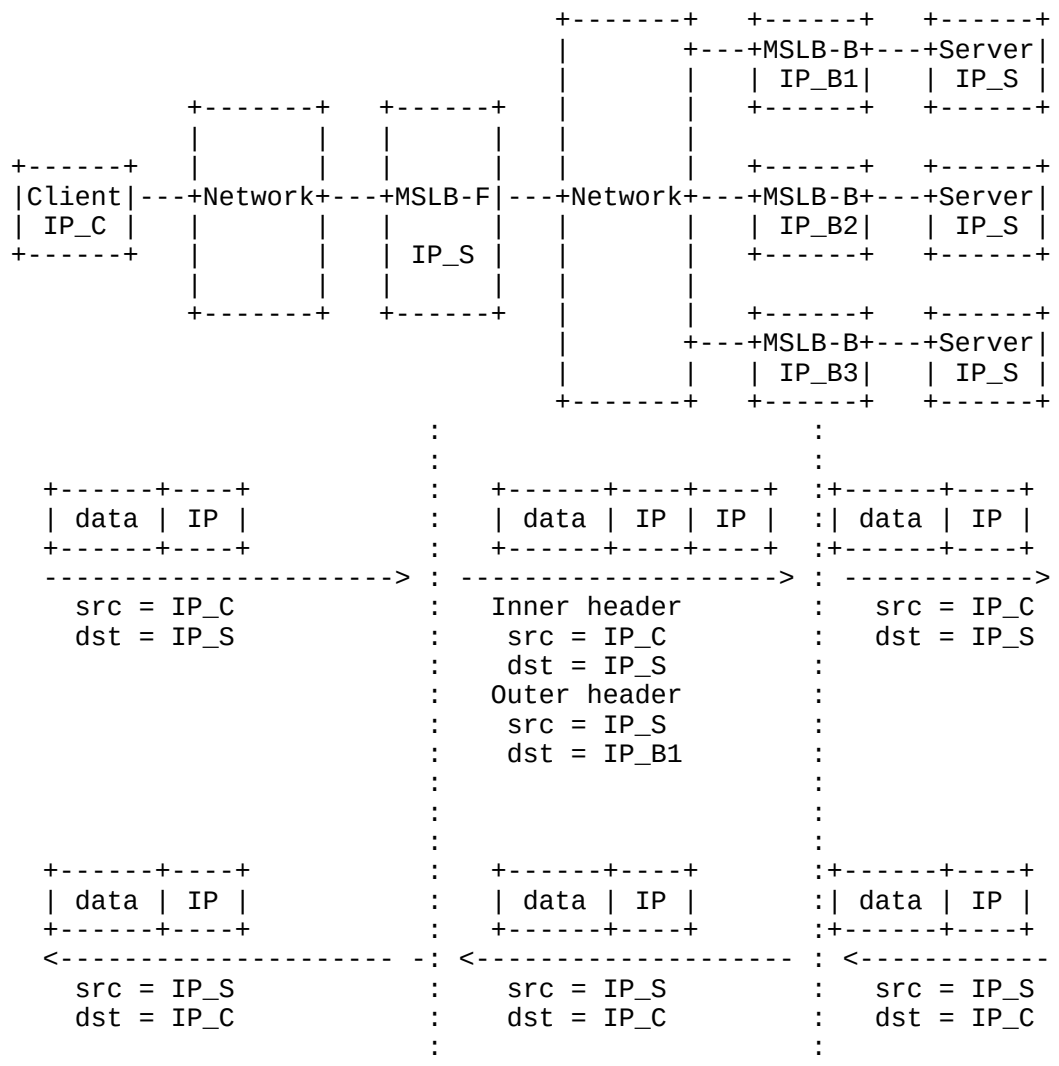
+-------+ +------+ +------+
| +---+MSLB-B+---+Server|
| | IP_B1| | IP_S |
+-------+ +------+ +------+ | | +------+ +------+
| | | | | | | |
+------+ | | | | | | +------+ +------+
|Client|---+Network+---+MSLB-F|---+Network+---+MSLB-B+---+Server|
| IP_C | | | | | | | | IP_B2| | IP_S |
+------+ | | | IP_S | | | | +------+ +------+
| | | | | | |
+-------+ +------+ | | +------+ +------+
| +---+MSLB-B+---+Server|
| | IP_B3| | IP_S |
+-------+ +------+ +------+
: : :
: : :
+------+----+ : +------+----+----+ :+------+----+
| data | IP | : | data | IP | IP | :| data | IP |
+------+----+ : +------+----+----+ :+------+----+
----------------------> : --------------------> : ------------>
src = IP_C : Inner header : src = IP_C
dst = IP_S : src = IP_C : dst = IP_S
: dst = IP_S :
: Outer header :
: src = IP_S :
: dst = IP_B1 :
: : :
: : :
: : :
+------+----+ : +------+----+ :+------+----+
| data | IP | : | data | IP | :| data | IP |
+------+----+ : +------+----+ :+------+----+
<---------------------- -: <-------------------- : <------------
src = IP_S : src = IP_S : src = IP_S
dst = IP_C : dst = IP_C : dst = IP_C
: : :

Figure 9

In this figure, to the client, the IP address is allocated IP_C1, IP_C2, and the server IP address is IP_S. In this case, IP_S is also allocated to all servers and MSLB-F. And to the MSLB-B, IP_B1, IP_B2, IP_B3 is allocated. This allocation is shown in the upper part of Figure 6.

The lower part of Figure 6 shows packet transfered between client and server. From the client to the server, MSLB-F encapsulates the original IP packet and sends it to MSLB-B. MSLB-B decapsulates the outer IP header and forwards it to the server. The inner IP packet does not change, which means, transparency is remained.

With encapsulation mode, packet size is increased, so fragmentation is needed if the encapsulated packet size exceeds MTU or Path MTU. MSLB-F MUST support tunnel MTU discovery [RFC1853]. Fragmentation and Path MTU discovery [RFC1191] issue will be described in the future.

Return packet, i.e., from the server to the client is not encapsulated, just forwarded.

On the Internet, the client IP address and the server IP address must Global IP address, however, the IP address of MSLB-B may private IP address.

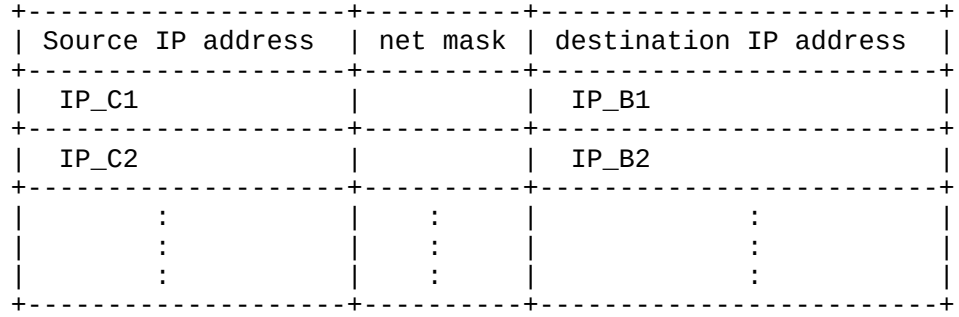

Figure 10

Figure 10 shows the MSLB table. MSLB has this table and encapsulates and generates an outer header with the destination address using this table value. MSLB-F checks the source IP address and generates the destination address of the outer header with this table.

Using IPv4 over IPv6 encapsulation or IPv6 over IPv4 encapsulation may be possible, i.e., IPv4 packet encapsulated to IPv6, then decapsulate to IPv4 or IPv6 packet encapsulated to IPv4, then de-encapsulated IPv6 may be possible. Figure 11 shows the possible combination of IPv4 and IPv6. These IPv4-IPv6 encapsulation cases will be defined in the future.

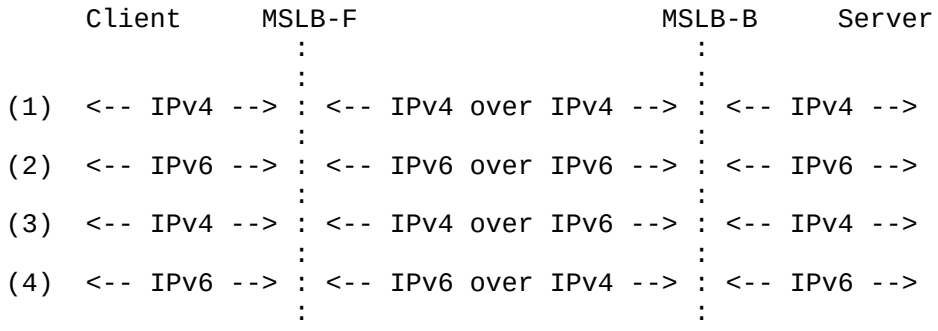

Figure 11

6. Ingress filtering environment

[RFC2827] describes ingress filtering for defending against DoS attacks that employ IP source address spoofing.

Depending on the location of the MSLB-F and MSLB-B, packets from the server to the client may be discarded by ingress filtering. In such a case, encapsulating the packet from server to client might resolve. Figure 12 shows such a solution.

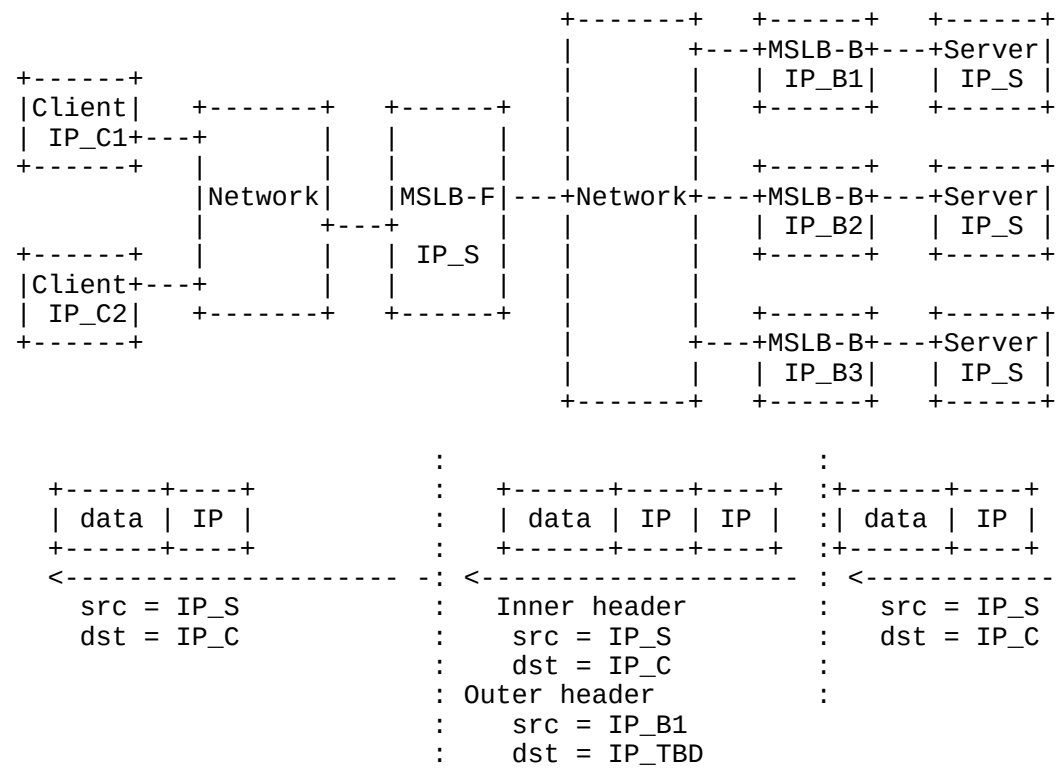

Figure 12

7. Characteristic

MSLB has the following characteristics.

* Layer 3 Load balancer
* Support NAT unfriendly applications such as FTP
* Work with any application layer protocol (maybe)
* Work with encryption (IPsec ESP, SSL/TLS)
* Work over Layer 3 network
* May enforce policy with static configuration

8. IANA Considerations

This document does not request IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

Security consideration is not discussed in this memo.

10. Acknowledgements

11. Normative References

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[RFC1546] Partridge, C., Mendez, T., and W. Milliken, "Host Anycasting Service", RFC 1546, DOI 10.17487/RFC1546, November 1993, <https://www.rfc-editor.org/info/rfc1546>.

[RFC1853] Simpson, W., "IP in IP Tunneling", RFC 1853, DOI 10.17487/RFC1853, October 1995, <https://www.rfc-editor.org/info/rfc1853>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2775] Carpenter, B., "Internet Transparency", RFC 2775, DOI 10.17487/RFC2775, February 2000, <https://www.rfc-editor.org/info/rfc2775>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, DOI 10.17487/RFC2993, November 2000, <https://www.rfc-editor.org/info/rfc2993>.

Author’s Address

Naoki Matsuhira
Neptela
Japan
Email: matsuhira.ietf@gmail.com