Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 14, 2012

B. Carpenter
Univ. of Auckland
S. Jiang
Huawei Technologies Co., Ltd
W. Tarreau
Exceliance

June 12, 2012

Using the IPv6 Flow Label for Server Load Balancing
draft-carpenter-flow-label-balancing-01

Abstract

This document describes how the IPv6 flow label as currently specified can be used to enhance layer 3/4 load distribution and balancing for large server farms.

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 http://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 December 14, 2012.

Table of Contents

1. Introduction

The IPv6 flow label has been redefined [RFC6437] and is now a recommended IPv6 node requirement [RFC6434]. Its use for load sharing in multipath routing has been specified [RFC6438]. Another scenario in which the flow label could be used is in load distribution for large server farms. Load distribution is a slightly more general term than load balancing, but the latter is more commonly used. This document starts with brief introductions to the flow label and to load balancing techniques, and then describes how the flow label can be used to enhance layer 3/4 load balancers in particular.

The motivation for this approach is to improve the performance of most types of layer 3/4 load balancers, especially for traffic including multiple IPv6 extension headers and in particular for fragmented packets. Fragmented packets, often the result of customers reaching the load balancer via a VPN with a limited MTU, are a common performance problem.

2. Summary of Flow Label Specification

The IPv6 flow label is a 20 bit field included in every IPv6 header [RFC2460]. It is recommended to be supported in all IPv6 nodes by [RFC6434] and it is defined in [RFC6437]. According to this definition, the flow label should be set to a constant value for a given traffic flow (such as an HTTP connection).

Any device that has access to the IPv6 header has access to the flow label, and it is at a fixed position in every IPv6 packet. In contrast, transport layer information, such as the port numbers, is not always in a fixed position, since it follows any IPv6 extension headers that may be present. In fact, the logic of finding the transport header is always more complex for IPv6 than for IPv4, due to the absence of an Internet Header Length field in IPv6. Therefore, within the lifetime of a given transport layer connection, the flow label can be a more convenient "handle" than the port number for identifying that particular connection.

According to RFC 6437, source hosts should set the flow label, but if they do not (i.e. its value is zero), forwarding nodes (such as the first-hop router) may set it instead. In both cases, the flow label value must be constant for a given transport session, normally identified by the IPv6 and Transport header 5-tuple. By default, the flow label value should be calculated by a stateless algorithm. The resulting value should form part of a statistically uniform distribution.

A careful reading of RFC 6437 shows that for a given source accessing a well-known TCP port at a given destination, the flow label is in effect a substitute for the source port number, found at a fixed position in the layer 3 header.

The flow label is defined as an end-to-end component of the IPv6 header, but there are three qualifications to this:

1. Until the RFC 6437 standard is widely implemented as recommended by RFC 6434, the flow label will often be set to the default value of zero.
2. Because of the recommendation to use a stateless algorithm to calculate the label, there is a low (but non-zero) probability that two simultaneous flows from the same source to the same destination have the same flow label value despite having different transport protocol port numbers.
3. The flow label field is in an unprotected part of the IPv6 header, which means that intentional or unintentional changes to its value cannot be trivially detected by a receiver.

The first two points are addressed below in Section 4 and the third in Section 5.

3. Summary of Load Balancing Techniques

Load balancing for server farms is achieved by a variety of methods, often used in combination [Tarreau]. The flow label is not relevant to all of them, and the actual load balancing algorithm (the choice of which server to use for a new client session) is irrelevant to this discussion.

- The simplest method is simply using the DNS to return different server addresses for a single name such as www.example.com to different users. Typically this is done by rotating the order in which different addresses are listed by the relevant authoritative DNS server, assuming that the client will pick the first one. Routing may be configured such that the different addresses are handled by different ingress routers. The flow label can have no impact on this method and it is not discussed further.
- Another method, for HTTP servers, is to operate a layer 7 reverse proxy in front of the server farm. The reverse proxy will present a single IP address to the world, communicated to clients by a single AAAA record. For each new client session (an incoming TCP connection and HTTP request), it will pick a particular server and proxy the session to it. Hopefully the act of proxying will be cheap compared to the act of serving the required content. The proxy must retain TCP state and proxy state for the duration of

the session. This TCP state could, potentially, include the incoming flow label value.

- o A component of some load balancing systems is an SSL reverse proxy farm. The individual SSL proxies handle all cryptographic aspects and exchange raw HTTP with the actual servers. Thus, from the load balancing point of view, this really looks just like a server farm, except that it's specialised for HTTPS. Each proxy will retain SSL and TCP and maybe HTTP state for the duration of the session, and the TCP state could potentially include the flow label.
- o Finally the "front end" of many load balancing systems is a layer 3/4 load balancer. While it can sometimes be a dedicated hardware, it also happens to be a standard function of some network switches or routers (eg: using ECMP, [RFC2991]). In this case, it is the layer 3/4 load balancer whose IP address is published as the primary AAAA record for the service. All client sessions will pass through this device. According to the precise scenario, it will spread new sessions across the actual application servers, across an SSL proxy farm, or across a set of layer 7 proxies. In all cases, the layer 3/4 load balancer has to recognize incoming packets as belonging to new or existing client sessions, and choose the target server or proxy so as to ensure persistence. 'Persistence' is defined as guaranteeing that a given session will run to completion on a single server. The layer 3/4 load balancer therefore needs to inspect each incoming packet to identify the session. There are two common types of layer 3/4 load balancers, the totally stateless ones which only act on packets, generally involving a per-packet hashing of easy-to-find information such as the source address and/or port into a server number, and the stateful ones which take the routing decision on the very first packets of a session and maintain the same direction for all packets belonging to the same session. Clearly, both types of layer 3/4 balancers could inspect and make use of the flow label value.

Our focus is on how the balancer identifies a particular flow. For clarity, note that two aspects of layer 3/4 load balancers could not be affected by use of the flow label to identify sessions:

1. Balancers use various techniques to redirect traffic to a specific target server.

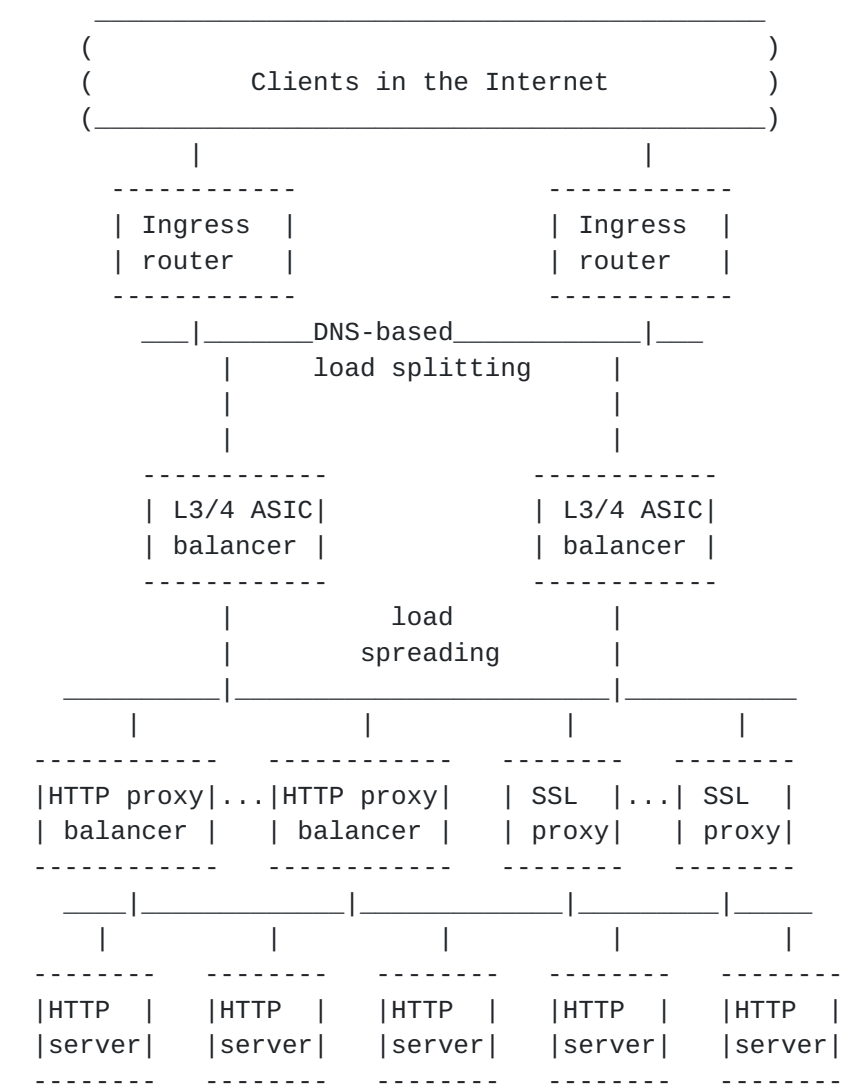
   - All servers are configured with the same IP address, they are all on the same LAN, and the load balancer sends directly to their individual MAC addresses.
   - Each server has its own IP address, and the balancer uses an IP-in-IP tunnel to reach it.

- Each server has its own IP address, and the balancer performs NAPT (network address and port translation) to deliver the client's packets to that address.

The choice between these methods is not affected by use of the flow label.

2. A layer 3/4 balancer must correctly handle Path MTU Discovery by forwarding relevant ICMPv6 packets in both directions. This too is not affected by use of the flow label.

The following diagram, inspired by [Tarreau], shows a maximum layout.



From the previous paragraphs, we can identify several points in this diagram where the flow label might be relevant:

1. Layer 3/4 load balancers.
2. SSL proxies.
3. HTTP proxies.

However, usage by the proxies seems unlikely to be cost-effective, so in this document we focus only on layer 3/4 balancers.

4. Applying the Flow Label to L3/L4 Load Balancing

The suggested model for using the flow label in a load balancing mechanism is as follows:

- We are only concerned with IPv6 traffic in which the flow label value has been set at or near the source according to [RFC6437]. If the flow label of an incoming packet is zero, load balancers will continue to use the transport header in the traditional way. As the use of the flow label becomes more prevalent according to RFC 6434, load balancers, and therefore users, will reap a growing performance benefit.
- If the flow label of an incoming packet is non-zero, layer 3/4 load balancers can use the 2-tuple {source address, flow label} as the session key for whatever load distribution algorithm they support. If any IPv6 extension headers, including fragment headers, are present, this will be significantly quicker than searching for the transport port numbers later in the packet. Moreover, the transport layer information such as the source port is not repeated in fragments, which generally prevents stateless load balancers from supporting fragmented traffic since they generally cannot reassemble fragments.

  Note that balancers usually do not need to consider the destination address as it is always the same, i.e., the server address.

  A stateless layer 3/4 load balancer would simply apply a hash algorithm to the 2-tuple {source address, flow label} on all packets, in order to select the same target server consistently for a given flow.

  A stateful layer 3/4 load balancer would apply its usual load distribution algorithm to the first packet of a session, and store the {2-tuple, server} association in a table so that subsequent packets belonging to the same session are forwarded to the same server. Thus, for all subsequent packets of the session, it can ignore all IPv6 extension headers, which should lead to a performance benefit. Whether this benefit is valuable will depend on engineering details of the specific load balancer.

Layer 3/4 balancers that redirect the incoming packets by NAPT are not expected to obtain any saving of time by using the flow label, because they must in any case follow the extension header chain in order to locate and modify the port number and transport checksum. The same would apply to balancers that perform TCP state tracking for any reason.

- Note that correct handling of ICMPv6 for Path MTU Discovery requires the layer 3/4 balancer to keep state for the client source address, independently of either the port numbers or the flow label.
- SSL and HTTP proxies, if present, should forward the flow label value towards the server. This has no performance benefit, but is consistent with the general RFC 6437 model for the flow label.

It should be noted that the performance benefit, if any, depends entirely on engineering trade-offs in the design of the L3/L4 balancer. An extra test is needed (is the label non-zero?), but all logic for handling extension headers can be omitted except for the first packet of a new flow. Since the only state to be stored is the 2-tuple and the server identifier, storage requirements will be reduced. Additionally, the method will work for fragmented traffic and for flows where the transport information is missing (unknown transport protocol) or obfuscated (e.g., IPsec). Traffic reaching the load balancer via a VPN is particularly prone to the fragmentation issue, due to MTU size issues. For some load balancer designs, these are very significant advantages.

In the unlikely event of two simultaneous flows from the same source address having the same flow label value, the two flows would end up assigned to the same server, where they would be distinguished as normal by their port numbers. Since this would be a statistically rare event, it would not damage the overall load balancing effect. Moreover, it is very likely that there will be many more flow label values than servers at most sites (1 million possible label values), so it is already expected that multiple flow label values will end up on the same server for a given IP address. In the case where many thousands of clients are hidden behind the same large-scale NAT with a single IP address, the assumption of low probability of conflicts might become incorrect unless flow label values are random enough to avoid following similar sequences for all clients. This is not expected to be a factor for IPv6 anyway, since there is no valid reason to implement NAT [RFC4864]. The statistical assumption is valid for sites that implement network prefix translation [RFC6296], since this technique provides a different address for each client.

5. Security Considerations

Security aspects of the flow label are discussed in [RFC6437]. As noted there, a malicious source or man-in-the-middle could disturb load balancing by manipulating flow labels. This risk already exists today where the source address and port are used as hashing key in layer 3/4 load balancers, as well as where a persistence cookie is used in HTTP to designate a server. It even exists on layer 3 components which only rely on the source address to select a destination, making them more DDoS-prone. Nevertheless, all these methods are currently used because the benefits for load balancing and persistence hugely outweigh the risks.

Specifically, [RFC6437] states that "stateless classifiers should not use the flow label alone to control load distribution, and stateful classifiers should include explicit methods to detect and ignore suspect flow label values." The former point is answered by also using the source address. The latter point is more complex. If the risk is considered serious, the site ingress router or the layer 3/4 balancer should verify incoming flows with non-zero flow label values. If a flow from a given source address and port number does not have a constant flow label value, it is suspect and should be dropped. This would deal with both intentional and accidental changes to the flow label.

RFC 6437 notes in its Security Considerations that if the covert channel risk is considered significant, a firewall might rewrite non-zero flow labels. As long as this is done as described in RFC 6437, it will not invalidate the mechanisms described above.

The flow label may be of use in protecting against distributed denial of service (DDOS) attacks against servers. As noted in RFC 6437, a source should generate flow label values that are hard to predict, most likely by including a secret nonce in the hash used to generate each label. The attacker does not know the nonce and therefore has no way to invent flow labels which will all target the same server, even with knowledge of both the hash algorithm and the load balancing algorithm. Still, it is important to understand that it is always trivial to force a load balancer to stick to the same server during an attack, so the security of the whole solution must not rely on the unpredicatability of the flow label values alone, but should include defensive measures like most load balancers already have against abnormal use of source address or session cookies.

New flows are assigned to a server according to any of the usual algorithms available on the load balancer (e.g., least connections, round robin, etc.). The association between the flow label value and the server is stored in a table (often called stick table) so that

future connections using the same flow label can be sent to the same server. This method is more robust against a loss of server and also makes it harder for an attacker to target a specific server, because the association between a flow label value and a server is not known externally.

6. IANA Considerations

This document requests no action by IANA.

7. Acknowledgements

Valuable comments and contributions were made by Fred Baker, Lorenzo Colitti, Joel Jaeggli, Gurudeep Kamat, Julius Volz, and others.

This document was produced using the xml2rfc tool [RFC2629].

8. Change log [RFC Editor: Please remove]

draft-carpenter-flow-label-balancing-01: update following comments, 2012-06-12.

draft-carpenter-flow-label-balancing-00: restructured after IETF83, 2012-05-08.

draft-carpenter-v6ops-label-balance-02: clarified after WG discussions, 2012-03-06.

draft-carpenter-v6ops-label-balance-01: updated with community comments, additional author, 2012-01-17.

draft-carpenter-v6ops-label-balance-00: original version, 2011-10-13.

9. References

9.1. Normative References

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

[RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, December 2011.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme,

"IPv6 Flow Label Specification", RFC 6437, November 2011.

9.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, November 2000.

[RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, May 2007.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, November 2011.

[Tarreau] Tarreau, W., "Making applications scalable with load balancing", 2006, <http://1wt.eu/articles/2006_lb/>.

Authors' Addresses

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: jiangsheng@huawei.com

Willy Tarreau
Exceliance
R&D Produits reseau
3 rue du petit Robinson
78350 Jouy-en-Josas
France

Email: w@1wt.eu