Internet Engineering Task Force (IETF)
Request for Comments: 7098
Category: Informational
ISSN: 2070-1721

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

January 2014

Using the IPv6 Flow Label for Load Balancing in Server Farms

Abstract

This document describes how the currently specified IPv6 flow label can be used to enhance layer 3/4 (L3/4) load distribution and balancing for large server farms.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7098.

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. In the context of a server farm, both terms refer to mechanisms that distribute the workload of a server farm among different servers in order to optimize performance. Server load balancing commonly applies to HTTP traffic, but most of the techniques described would apply to other upper-layer applications as well. This document starts with brief introductions to the flow label and to server load-balancing techniques, and then describes how the flow label can be used to enhance load balancers operating on IP packets and TCP sessions, commonly known as layer 3/4 load balancers.

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 [RFC6437] is a 20-bit field included in every IPv6 header [RFC2460]. It is recommended to be supported in all IPv6 nodes by [RFC6434]. There is additional background material in [RFC6436] and [RFC6294]. According to its definition, the flow label should be set to a constant value for a given traffic flow (such as an HTTP connection), and that value will belong to a uniform statistical distribution, making it potentially valuable for load-balancing purposes.

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. Additionally, if packets are fragmented, the flow label will be present in all fragments, but the transport header will only be in one packet. 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; however, 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, regardless of which node sets it.

It is recognized that at the time of writing, very few traffic flows include a non-zero flow label value. The mechanism described below is one that can be added to existing load-balancing mechanisms, so that it will become effective as more and more flows contain a non-zero label. Even if the flow label is chosen from an imperfectly uniform distribution, it will nevertheless increase the information entropy of the IPv6 header as a whole. This allows for progressive introduction of load balancing based on the flow label.

If the recommendations in Section 3 of RFC 6437 are followed for traffic from a given source accessing a well-known TCP port at a given destination, the flow label can act as a substitute for the port numbers as far as a load balancer is concerned, and it can be found at a fixed position in the layer 3 header even if extension headers are present.

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 IPv6 flow label specification in RFC 6437 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 easily detected by a receiver.

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

3. Summary of Server Farm Load-Balancing Techniques

Load balancing for server farms is achieved by a variety of methods, often used in combination [Tarreau]. This section gives a general overview of common methods, although the flow label is not relevant to all of them. The actual load-balancing algorithm (the choice of which server to use for a new client session) is irrelevant to this discussion. We give examples for HTTP, but analogous techniques may be used for other application protocols.

- The simplest method is using the DNS to return different server addresses for a single name such as www.example.com to different users. This is typically done by rotating the order in which different addresses within the server site are listed by the relevant authoritative DNS server, on the assumption that the client will pick the first one. Routing may be configured such that the different addresses are handled by different ingress routers. Several variants of this load-balancing mechanism exist, such as expecting some clients to use all the advertised addresses when multiple connections are involved, or directing the traffic to multiple sites, also known as global load balancing. None of these mechanisms are in the scope of this document, and the proposal in this document does not affect their usability nor aim to replace them, so they will not be 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. The act of proxying should be more efficient and less resource-intensive than 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.

- A component of some load-balancing systems is an SSL reverse proxy farm. The individual SSL proxies handle all cryptographic aspects and exchange unencrypted 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 specialized 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.

- Finally the "front end" of many load-balancing systems is a layer 3/4 load balancer. While it can be a dedicated device, it is also a standard function of some network switches or routers (e.g. using Equal-Cost Multipath Routing (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. Depending on the specific scenario, the balancer will assign new sessions among the actual application servers, across an SSL proxy farm, or among a set of layer 7 proxies. In all cases, the layer 3/4 load balancer has to classify incoming packets very quickly and choose the target server or proxy so as to ensure persistence. 'Persistence' is defined as the guarantee that a given client session will run to completion on a single server. The layer 3/4 load balancer therefore needs to inspect each incoming packet to classify it. There are two common types of layer 3/4 load balancers, the totally stateless ones which only act on single 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 that 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 are not 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. In this case, return packets from the server to the client are sent back without passing through the balancer, a technique known as direct server return, but we are not concerned here with the return packets.

+ All servers are configured with the same IP address, treated locally as an anycast address by layer 3 ECMP routing.

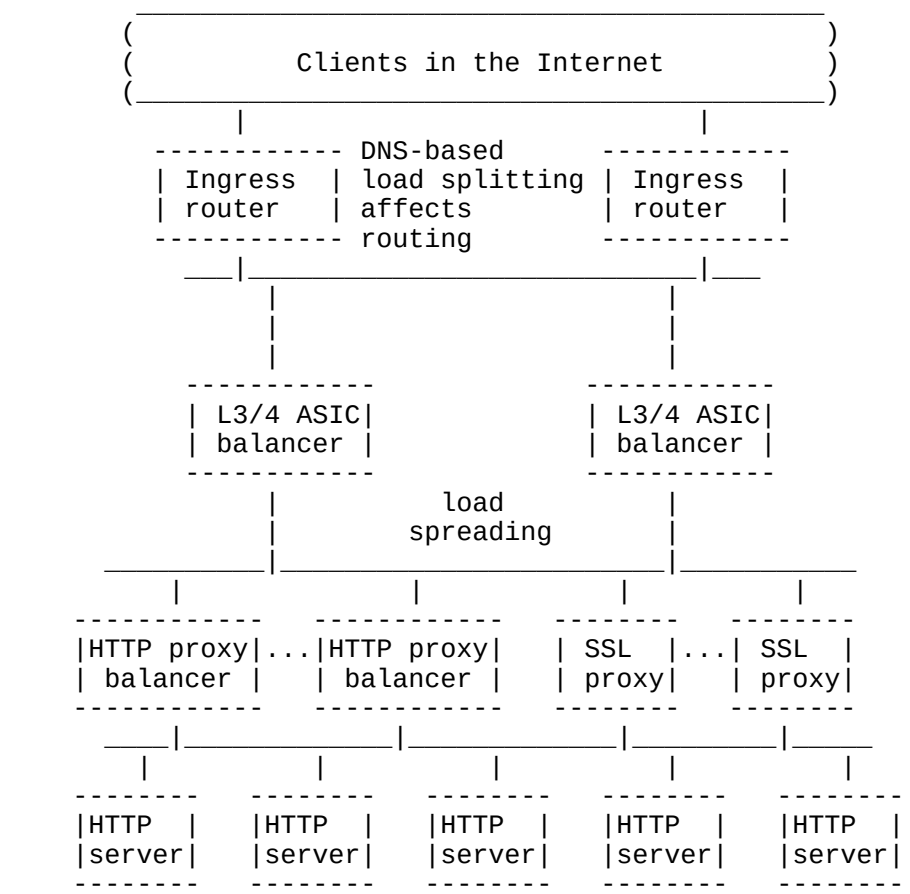
+ Each server has its own IP address, and the balancer uses an IP-in-IP tunnel to reach it.

+ 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 directly affected by use of the flow label. It should be noted that there may be difficulty correlating an ICMPv6 "Packet too big" response with the session it refers to, but that is out of the scope of the present document.

The following diagram, inspired by [Tarreau], shows a layout with various methods in use together. (Below, "ASIC" stands for "Application-Specific Integrated Circuit".)



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 affect performance, because they must in any case process the application-layer header, so in this document we focus only on layer 3/4 balancers.

4. Applying the Flow Label to Layer 3/4 Load Balancing

The suggested model for using the flow label to enhance an layer 3/4 load-balancing mechanism is as follows:

- We are only concerned with IPv6 traffic in which the flow label value has been set 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. Alternatively, they might use the 3-tuple {dest address, source address, flow label}, especially if the server farm supports multiple server IP addresses, but using the 3-tuple 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.

  A stateless layer 3/4 load balancer would simply apply a hash algorithm to the 2-tuple or 3-tuple on all packets in order to select the same target server consistently for a given flow. Needless to say, the hash algorithm has to be well chosen for its purpose, but this problem is common to several forms of stateless load balancing. The discussion in [RFC6438] applies.

  A stateful layer 3/4 load balancer would apply its usual load distribution algorithm to the first packet of a session, and store the {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.

  Note that such a balancer will not identify new transport sessions from the same source that use the same flow label; they will be delivered to the same server. This is like the behavior of existing hash-based layer 4 balancers that always send similarly hashed packets to the same destination. However, a global state table in a flow label balancer cannot be shared between multiple

services if these services rely on transport-layer information, since the goal of using the flow label is to avoid looking up that information.

A related issue is that the balancer will not detect FIN/ACK sequences at the end of sessions. Therefore, it will rely on inactivity timers to delete session state. However, all existing balancers must maintain such timers to deal with hung sessions, and the practical impact on memory utilization is unlikely to be significant.

- 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 have no choice but to 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 usually has no performance benefit, but it is consistent with the general model for the flow label described in RFC 6437.

It should be noted that the performance benefit, if any, depends entirely on engineering trade-offs in the design of the layer 3/4 balancer. An extra test is needed to check if the label is non-zero, but if there is a non-zero label, all logic for handling extension headers can be skipped except for the first packet of a new flow. Since the identifying state to be stored is only the 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. There are approximately one million possible flow label values, and if the rules for flow label generation [RFC6437] are followed, this would be a statistically rare

event, and would not damage the overall load-balancing effect. Moreover, with a million possible label values, it is very likely that there will be many more flow label values than servers at most sites, so it is already expected that multiple flow label values will end up on the same server for a given client IP address.

In the case that many thousands of clients are hidden behind the same large-scale NAPT with a single shared 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 need to implement large-scale NAPT with address sharing [RFC4864]. The probability of conflicts is low 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 a 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 that 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. The flow label does not significantly alter this situation.

Specifically, the IPv6 flow label specification [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 use a suitable heuristic to 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.

A malicious source or man-in-the-middle could generate a flow in which the flow label is constant but the transport port numbers in some packets are invalid. Such packets, if load-balanced only on the basis of the flow label, could reach the target server and create a single-source DoS attack on its TCP engine.

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 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 that 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 unpredictability of the flow label values alone, but should include defensive measures like most load balancers already have against abnormal use of source addresses 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 2-tuple {source address, flow label} and the server is stored in a table (often called stick table) so that future traffic from the same source 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.

In the case that a stateless hash function is used to assign client packets to specific servers, it may be advisable to use a cryptographic hash function of some kind, to ensure that an attacker cannot predict the behavior of the load balancer.

6. Acknowledgements

Valuable comments and contributions were made by Fred Baker, Olivier Bonaventure, Ben Campbell, Lorenzo Colitti, Linda Dunbar, Donald Eastlake, Joel Jaeggli, Gurudeep Kamat, Warren Kumari, Julia Renouard, Julius Volz, and others.

7. References

7.1. 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.

7.2. Informative References

[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.

[RFC6294] Hu, Q. and B. Carpenter, "Survey of Proposed Use Cases for the IPv6 Flow Label", RFC 6294, June 2011.

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

[RFC6436] Amante, S., Carpenter, B., and S. Jiang, "Rationale for Update to the IPv6 Flow Label Specification", RFC 6436, November 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
HAProxy Technologies, Inc.
R&D Network Products
3 rue du petit Robinson
78350 Jouy-en-Josas
France

EMail: willy@haproxy.com