Network Working Group	B. Carpenter
Internet-Draft	Univ. of Auckland
Intended status: BCP	S. Amante
Expires: April 10, 2011	Level 3
	October 7, 2010

TOC

Using the IPv6 flow label for equal cost multipath routing and link aggregation in tunnels
draft-carpenter-flow-ecmp-03

Abstract

The IPv6 flow label has certain restrictions on its use. This document describes how those restrictions apply when using the flow label for load balancing by equal cost multipath routing, and for link aggregation, particularly for tunneled traffic.

Status of this Memo

Copyright Notice

Table of Contents

---

1. Introduction

TOC

When several network paths between the same two nodes are known by the routing system to be equally good (in terms of capacity and latency), it may be desirable to share traffic among them. Two such techniques are known as equal cost multipath routing (ECMP) and link aggregation (LAG) [IEEE802.1AX] (Institute of Electrical and Electronics Engineers, “Link Aggregation,” 2008.). There are of course numerous possible approaches to this, but certain goals need to be met:

- Roughly equal share of traffic on each path.
- Work-conserving method (no idle time when queue is non-empty).
- Minimize or avoid out-of-order delivery for individual traffic flows.

There is some conflict between these goals: for example, strictly avoiding idle time could cause a small packet sent on an idle path to overtake a bigger packet from the same flow, causing out-of-order delivery.
One lightweight approach to ECMP or LAG is this: if there are N equally good paths to choose from, then form a modulo(N) hash [RFC2991] (Thaler, D. and C. Hopps, “Multipath Issues in Unicast and Multicast Next-Hop Selection,” November 2000.) from a consistent set of fields in each packet header, and use the resulting value to select a particular path. If the hash function is chosen so that the hash values have a uniform statistical distribution, this method will share traffic roughly equally between the N paths. If the header fields included in the hash are consistent, all packets from a given flow will generate the same hash, so out-of-order delivery will not occur. Assuming a large number of unique flows are involved, it is also probable that the method will be work-conserving, since the queue for each link will remain non-empty.

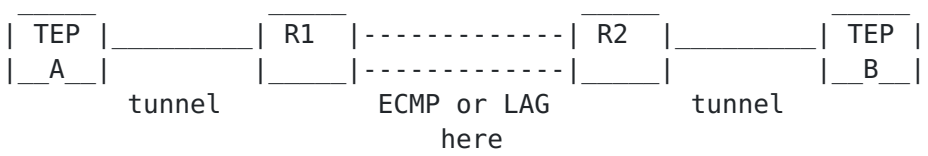
The question with such a method is which IP header fields are chosen to identify a flow and, consequently, are used as input keys to a modulo(N) hash algorithm.

In the remainder of this document, we will use the term "flow" to represent a sequence of packets that may be identified by either the source and destination IP addresses alone {2-tuple} or the source and destination IP addresses, protocol and source and destination port numbers {5-tuple}. It should be noted that the latter is more specifically referred to as a "microflow" in [RFC2474] (Nichols, K., Blake, S., Baker, F., and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” December 1998.), but this term is not used in connection with the flow label in [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, “IPv6 Flow Label Specification,” March 2004.).

The question with such a method, then, is which IP header fields to include to identify a flow. A minimal choice in the routing system is simply to use a hash of the source and destination IP addresses, i.e., the 2-tuple. This is necessary and sufficient to avoid out-of-order delivery, and with a wide variety of sources and destinations, as one finds in the core of the network, sometimes sufficient to achieve work-conserving load sharing. In practice, implementations often use the 5-tuple {dest addr, source addr, protocol, dest port, source port} as input keys to the hash function, to maximize the probability of evenly sharing traffic over the equal cost paths. However, including transport layer information as input keys to a hash may be a problem for IPv4 fragments [RFC2991] (Thaler, D. and C. Hopps, “Multipath Issues in Unicast and Multicast Next-Hop Selection,” November 2000.). In addition, protocol and destination port numbers in the hash will not only make the hash slightly more expensive to compute, but will not particularly improve the hash distribution, due to the prevalence of well known port numbers and popular protocol numbers. Ephemeral ports, on the other hand, are quite well distributed [Lee10] (Lee, D., Carpenter, B., and N. Brownlee, “Observations of UDP to TCP Ratio and Port Numbers,” May 2010.). In the case of IPv6, protocol numbers are particularly inconvenient due to the variable placement of and variable length of next-headers. In addition, [RFC2460] (Deering, S. and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” December 1998.) recommends that all next-headers, except hop-by-hop options, should not be inspected by intermediate nodes in the network, presumably to make introduction of new next-headers more straightforward.

The situation is different in tunneled scenarios. Identifying a flow inside the tunnel is more complicated, particularly because nearly all hardware can only identify flows based on information contained in the outermost IP header. Assume that traffic from many sources to many destinations is aggregated in a single IP-in-IP tunnel from tunnel end point (TEP) A to TEP B (see figure). Then all the packets forming the tunnel have outer source address A and outer destination address B. In all probability they also have the same port and protocol numbers. If there are multiple paths between routers R1 and R2, and ECMP or LAG is applied to choose a particular path, the 5-tuple and its hash will be constant and no load sharing will be achieved. If there is much tunnel

traffic, this will result in a high probability of congestion on one of the paths between R1 and R2.



Also, for IPv6, the total number of bits in the 5-tuple is quite large (296), as well as inconvenient to extract due to the next-header placement. This may be challenging for some hardware implementations, raising the potential that network equipment vendors might sacrifice the length of the fields extracted from an IPv6 header. The question therefore arises whether the 20-bit flow label in IPv6 packets would be suitable for use as input to an ECMP or LAG hash algorithm. If it could be used in place of the port numbers and protocol number in the 5-tuple, the hash calculation would be simplified.
The flow label is left experimental by [RFC2460] (Deering, S. and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” December 1998.) but is better defined by [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, “IPv6 Flow Label Specification,” March 2004.). We quote three rules from that RFC:

1. "The Flow Label value set by the source MUST be delivered unchanged to the destination node(s)."

2. "IPv6 nodes MUST NOT assume any mathematical or other properties of the Flow Label values assigned by source nodes."

3. "Router performance SHOULD NOT be dependent on the distribution of the Flow Label values. Especially, the Flow Label bits alone make poor material for a hash key."

These rules, especially the last one, have caused designers to hesitate about using the flow label in support of ECMP or LAG. The fact is today that most nodes set a zero value in the flow label, and the first rule definitely forbids the routing system from changing the flow label once a packet has left the source node. Considering normal IPv6 traffic, the fact that the flow label is typically zero means that it would add no value to an ECMP or LAG hash. But neither would it do any harm to the distribution of the hash values. If the community at some stage agrees to set pseudo-random flow labels in the majority of traffic flows, this would add to the value of the hash.
However, in the case of an IP-in-IPv6 tunnel, the TEP is itself the source node of the outer packets. Therefore, a TEP may freely set a flow label in the outer IPv6 header of the packets it sends into the tunnel. In particular, it may follow the [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, “IPv6 Flow Label Specification,” March 2004.) suggestion to set a pseudo-random value.

The second two rules quoted above need to be seen in the context of [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification," March 2004.), which assumes that routers using the flow label in some way will be involved in some sort of method of establishing flow state: "To enable flow-specific treatment, flow state needs to be established on all or a subset of the IPv6 nodes on the path from the source to the destination(s)." The RFC should perhaps have made clear that a router that has participated in flow state establishment can rely on properties of the resulting flow label values without further signaling. If a router knows these properties, rule 2 is irrelevant, and it can choose to deviate from rule 3.
In the tunneling situation sketched above, routers R1 and R2 can rely on the flow labels set by TEP A and TEP B being assigned by a known method. This allows a safe ECMP or LAG method to be based on the flow label without breaching [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification," March 2004.).

---

2. Normative Notation

TOC

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] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

---

3. Guidelines

TOC

We assume that the routers supporting ECMP or LAG (R1 and R2 in the above figure) are unaware that they are handling tunneled traffic. If it is desired to include the IPv6 flow label in an ECMP or LAG hash in the tunneled scenario shown above, the following guidelines apply:

*Inner packets MUST be encapsulated in an outer IPv6 packet whose source and destination addresses are those of the tunnel end points (TEPs).

*The flow label in the outer packet SHOULD be set by the sending TEP to a pseudo-random 20-bit value in accordance with [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification," March 2004.). The same flow label value MUST be used for all packets in a single user flow, as determined by the IP header fields of the inner packet.

-Note that this rule is a SHOULD rather than a MUST, to permit individual implementers to take an alternative approach if they

wish to do so. Such an alternative MUST conform to [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification," March 2004.).

*The sending TEP MUST classify all packets into flows, once it has determined that they should enter a given tunnel, and then write the relevant flow label into the outer IPv6 header. A user flow could be identified by the ingress TEP most simply by its {destination, source} address pair (coarse) or by its 5-tuple {dest addr, source addr, protocol, dest port, source port} (fine). This is an implementation detail in the sending TEP.

-It might be possible to make this classifier stateless, by using a suitable 20 bit hash of the inner IP header's 2-tuple or 5-tuple as the pseudo-random flow label value.

*At intermediate router(s) that perform load distribution of tunneled packets whose source address is a TEP, the hash algorithm used to determine the outgoing component-link in an ECMP and/or LAG toward the next-hop MUST minimally include the triple {dest addr, source addr, flow label} to meet the [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification," March 2004.) rules.

-Intermediate router(s) MAY also include {protocol, dest port, source port} as input keys to the ECMP and/or LAG hash algorithms, to provide sufficient entropy in cases where the flow-label is currently set to zero.

---

4. Security Considerations

TOC

The flow label is not protected in any way and can be forged by an on-path attacker. Off-path attackers are unlikely to guess a valid flow label if a pseudo-random value is used. In either case, the worst an attacker could do against ECMP or LAG is to attempt to selectively overload a particular path. For further discussion, see [RFC3697] (Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification," March 2004.).

---

5. IANA Considerations

TOC

This document requests no action by IANA.

---

6. Acknowledgements

TOC

This document was suggest by corridor discussions at IETF76. Joel Halpern made crucial comments on an early version. We are grateful to Qinwen Hu for general discussion about the flow label. Valuable comments and contributions were made by Jarno Rajahalme, Brian Haberman, Sheng Jiang, and others.
This document was produced using the xml2rfc tool [RFC2629] (Rose, M., "Writing I-Ds and RFCs using XML," June 1999.).

---

7. Change log

TOC

draft-carpenter-flow-ecmp-03: clarifications after further comments, 2010-10-07
draft-carpenter-flow-ecmp-02: updated after IETF77 discussion, especially adding LAG, changed to BCP language, added second author, 2010-04-14
draft-carpenter-flow-ecmp-01: updated after comments, 2010-02-18
draft-carpenter-flow-ecmp-00: original version, 2010-01-19

---

8. References

TOC

8.1. Normative References

TOC

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2460]	Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998 (TXT, HTML, XML).
[RFC3697]	Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification," RFC 3697, March 2004 (TXT).

8.2. Informative References

TOC

[IEEE802.1AX]	Institute of Electrical and Electronics Engineers, “Link Aggregation,” IEEE Standard 802.1AX-2008, 2008.
[Lee10]	Lee, D., Carpenter, B., and N. Brownlee, “Observations of UDP to TCP Ratio and Port Numbers,” Fifth International Conference on Internet Monitoring and Protection ICIMP 2010, May 2010.
[RFC2474]	Nichols, K., Blake, S., Baker, F., and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” RFC 2474, December 1998 (TXT, HTML, XML).
[RFC2629]	Rose, M., “Writing I-Ds and RFCs using XML,” RFC 2629, June 1999 (TXT, HTML, XML).
[RFC2991]	Thaler, D. and C. Hopps, “Multipath Issues in Unicast and Multicast Next-Hop Selection,” RFC 2991, November 2000 (TXT).

Authors' Addresses

TOC

	Brian Carpenter
	Department of Computer Science
	University of Auckland
	PB 92019
	Auckland, 1142
	New Zealand
Email:	brian.e.carpenter@gmail.com
	Shane Amante
	Level 3 Communications, LLC
	1025 Eldorado Blvd
	Broomfield, CO 80021
	USA
Email:	shane@level3.net