Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 23, 2010

B. Carpenter
Univ. of Auckland
January 19, 2010

Using the IPv6 flow label for equal cost multipath routing in tunnels
draft-carpenter-flow-ecmp-00

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, particularly for tunneled traffic.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

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. This is known as equal cost multipath routing (ECMP). 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 approach to ECMP is this: if there are N equally good paths to choose from, then form a hash code modulo(N) from each packet header, and use the resulting value to select a particular path. If the hash values have an even statistical distribution, this method will share traffic roughly equally between the N paths. If the header fields included in the hash are well chosen, all packets from a given flow will generate the same hash, so out-of-order delivery will not occur. Assuming a large number of flows from many sources 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 to include. A minimal choice in the routing system is simply to use a hash of the source and destination IP addresses. 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, probably sufficient to achieve work-conserving load sharing. In practice, implementations often use the 5-tuple {dest addr, source addr, protocol, dest port, source port}. However, including port and protocol 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.

The situation is different in tunneled scenarios. 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 tunnel packets have source address A and 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 is applied, the 5-tuple and its hash will be constant and no load sharing will be achieved.

```
   _____           _____               _____           _____
  | TEP |_________| R1  |-------------| R2  |_________| TEP |
  |__A__|         |_____|-------------|_____|         |__B__|
          tunnel           ECMP here          tunnel
```

Also, for IPv6, the total number of bits in the hash would then be quite large (296), which could be an issue for some hardware implementations. The question therefore arises whether the 20-bit flow label in IPv6 packets would be suitable for using in an ECMP hash.

The flow label is left experimental by [RFC2460] but is better defined by [RFC3697]. 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. The fact is today that most nodes do not set a non-zero value in the flow label, and the first rule definitely forbids the routing system from doing so 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 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] suggestion to set a pseudo-random value.

The second two rules quoted above need to be seen in the context of [RFC3697], 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 know, rather than assume, properties of the resulting flow label values. 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 method to be based on the flow label without breaching [RFC3697].

2. Guidelines

We assume that the routers supporting ECMP (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 hash in the tunneled scenario shown above, the following guidelines are suggested:

- Inner packets should 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]. The same flow label value should be used for all packets in a single user flow.
- The TEP will need a flow classifier for packets entering the tunnel. A user flow could be defined most simply by its {destination, source} address pair (coarse ECMP) or by its 5-tuple {dest addr, source addr, protocol, dest port, source port} (fine ECMP). This is an implementation detail in the TEP.
- In router(s) liable to perform ECMP for packets whose source address is a TEP, the ECMP hash should minimally include the triple {dest addr, source addr, flow label} to meet the [RFC3697] rules. In practice, since the routers are assumed to be unaware of tunneled traffic, this means adding the flow label to the existing 5-tuple hash.
    * For tunnel packets, it is harmless for the hash to also include {protocol, dest port, source port}, which will be constant.
    * For non-tunnel packets, it is harmless for the hash to also include the flow label, which is zero in normal traffic, and could only improve the hash if set.

3. Security Considerations

The flow label is not protected in any way and can be forged by an on-path attacker. Off-path attackers are extremely unlikely to guess a valid flow label. In either case, the worst an attacker could do

against ECMP is to selectively overload a particular path.

4. IANA Considerations

This document requests no action by IANA.

5. Acknowledgements

This document was suggest by corridor discussions at IETF76. Joel Halpern made crucial comments on an early version. The author is grateful to Qinwen Hu for general discussion about the flow label. Valuable comments and contributions were made by ..., and others.

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

6. Change log

draft-carpenter-flow-ecmp-00: original version, 2010-01-19

7. References

7.1. Normative References

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

[RFC3697] Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification", RFC 3697, March 2004.

7.2. Informative References

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

Author's Address

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

Email: brian.e.carpenter@gmail.com