Network Working Group
Request for Comments: 3697
Category: Standards Track

J. Rajahalme
Nokia
A. Conta
Transwitch
B. Carpenter
IBM
S. Deering
Cisco

March 2004

IPv6 Flow Label Specification

Status of this Memo

Copyright Notice

Abstract

This document specifies the IPv6 Flow Label field and the minimum requirements for IPv6 source nodes labeling flows, IPv6 nodes forwarding labeled packets, and flow state establishment methods. Even when mentioned as examples of possible uses of the flow labeling, more detailed requirements for specific use cases are out of scope for this document.

The usage of the Flow Label field enables efficient IPv6 flow classification based only on IPv6 main header fields in fixed positions.

1. Introduction

A flow is a sequence of packets sent from a particular source to a particular unicast, anycast, or multicast destination that the source desires to label as a flow. A flow could consist of all packets in a specific transport connection or a media stream. However, a flow is not necessarily 1:1 mapped to a transport connection.

Traditionally, flow classifiers have been based on the 5-tuple of the source and destination addresses, ports, and the transport protocol type. However, some of these fields may be unavailable due to either fragmentation or encryption, or locating them past a chain of IPv6 option headers may be inefficient. Additionally, if classifiers depend only on IP layer headers, later introduction of alternative transport layer protocols will be easier.

The usage of the 3-tuple of the Flow Label and the Source and Destination Address fields enables efficient IPv6 flow classification, where only IPv6 main header fields in fixed positions are used.

The minimum level of IPv6 flow support consists of labeling the flows. IPv6 source nodes supporting the flow labeling MUST be able to label known flows (e.g., TCP connections, application streams), even if the node itself would not require any flow-specific treatment. Doing this enables load spreading and receiver oriented resource reservations, for example. Node requirements for flow labeling are given in section 3.

Specific flow state establishment methods and the related service models are out of scope for this specification, but the generic requirements enabling co-existence of different methods in IPv6 nodes are set forth in section 4. The associated scaling characteristics (such as nodes involved in state establishment, amount of state maintained by them, and state growth function) will be specific to particular service models.

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 BCP 14, RFC 2119 [KEYWORDS].

2. IPv6 Flow Label Specification

The 20-bit Flow Label field in the IPv6 header [IPv6] is used by a source to label packets of a flow. A Flow Label of zero is used to indicate packets not part of any flow. Packet classifiers use the triplet of Flow Label, Source Address, and Destination Address fields to identify which flow a particular packet belongs to. Packets are processed in a flow-specific manner by the nodes that have been set up with flow-specific state. The nature of the specific treatment and the methods for the flow state establishment are out of scope for this specification.

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

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

Nodes keeping dynamic flow state MUST NOT assume packets arriving 120 seconds or more after the previous packet of a flow still belong to the same flow, unless a flow state establishment method in use defines a longer flow state lifetime or the flow state has been explicitly refreshed within the lifetime duration.

The use of the Flow Label field does not necessarily signal any requirement on packet reordering. Especially, the zero label does not imply that significant reordering is acceptable.

If an IPv6 node is not providing flow-specific treatment, it MUST ignore the field when receiving or forwarding a packet.

3. Flow Labeling Requirements

To enable Flow Label based classification, source nodes SHOULD assign each unrelated transport connection and application data stream to a new flow. The source node MAY also take part in flow state establishment methods that result in assigning certain packets to specific flows. A source node which does not assign traffic to flows MUST set the Flow Label to zero.

To enable applications and transport protocols to define what packets constitute a flow, the source node MUST provide means for the applications and transport protocols to specify the Flow Label values to be used with their flows. The use of the means to specify Flow Label values is subject to appropriate privileges (see section 5.1). The source node SHOULD be able to select unused Flow Label values for flows not requesting a specific value to be used.

A source node MUST ensure that it does not unintentionally reuse Flow Label values it is currently using or has recently used when creating new flows. Flow Label values previously used with a specific pair of source and destination addresses MUST NOT be assigned to new flows with the same address pair within 120 seconds of the termination of the previous flow. The source node SHOULD provide the means for the applications and transport protocols to specify quarantine periods longer than the default 120 seconds for individual flows.

To avoid accidental Flow Label value reuse, the source node SHOULD select new Flow Label values in a well-defined sequence (e.g., sequential or pseudo-random) and use an initial value that avoids

reuse of recently used Flow Label values each time the system restarts. The initial value SHOULD be derived from a previous value stored in non-volatile memory, or in the absence of such history, a randomly generated initial value using techniques that produce good randomness properties [RND] SHOULD be used.

4. Flow State Establishment Requirements

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 methods for the state establishment, as well as the models for flow-specific treatment will be defined in separate specifications.

To enable co-existence of different methods in IPv6 nodes, the methods MUST meet the following basic requirements:

(1) The method MUST provide the means for flow state clean-up from the IPv6 nodes providing the flow-specific treatment. Signaling based methods where the source node is involved are free to specify flow state lifetimes longer than the default 120 seconds.

(2) Flow state establishment methods MUST be able to recover from the case where the requested flow state cannot be supported.

5. Security Considerations

This section considers security issues raised by the use of the Flow Label, primarily the potential for denial-of-service attacks, and the related potential for theft of service by unauthorized traffic (Section 5.1). Section 5.2 addresses the use of the Flow Label in the presence of IPsec including its interaction with IPsec tunnel mode and other tunneling protocols. We also note that inspection of unencrypted Flow Labels may allow some forms of traffic analysis by revealing some structure of the underlying communications. Even if the flow label were encrypted, its presence as a constant value in a fixed position might assist traffic analysis and cryptoanalysis.

5.1. Theft and Denial of Service

Since the mapping of network traffic to flow-specific treatment is triggered by the IP addresses and Flow Label value of the IPv6 header, an adversary may be able to obtain better service by modifying the IPv6 header or by injecting packets with false addresses and/or labels. Taken to its limits, such theft-of-service becomes a denial-of-service attack when the modified or injected traffic depletes the resources available to forward it and other

traffic streams. A curiosity is that if a DoS attack were undertaken against a given Flow Label (or set of Flow Labels), then traffic containing an affected Flow Label might well experience worse-than-best-effort network performance.

Note that since the treatment of IP headers by nodes is typically unverified, there is no guarantee that flow labels sent by a node are set according to the recommendations in this document. Therefore, any assumptions made by the network about header fields such as flow labels should be limited to the extent that the upstream nodes are explicitly trusted.

Since flows are identified by the 3-tuple of the Flow Label and the Source and Destination Address, the risk of theft or denial of service introduced by the Flow Label is closely related to the risk of theft or denial of service by address spoofing. An adversary who is in a position to forge an address is also likely to be able to forge a label, and vice versa.

There are two issues with different properties: Spoofing of the Flow Label only, and spoofing of the whole 3-tuple, including Source and Destination Address.

The former can be done inside a node which is using or transmitting the correct source address. The ability to spoof a Flow Label typically implies being in a position to also forge an address, but in many cases, spoofing an address may not be interesting to the spoofer, especially if the spoofer's goal is theft of service, rather than denial of service.

The latter can be done by a host which is not subject to ingress filtering [INGR] or by an intermediate router. Due to its properties, such is typically useful only for denial of service. In the absence of ingress filtering, almost any third party could instigate such an attack.

In the presence of ingress filtering, forging a non-zero Flow Label on packets that originated with a zero label, or modifying or clearing a label, could only occur if an intermediate system such as a router was compromised, or through some other form of man-in-the-middle attack. However, the risk is limited to traffic receiving better or worse quality of service than intended. For example, if Flow Labels are altered or cleared at random, flow classification will no longer happen as intended, and the altered packets will receive default treatment. If a complete 3-tuple is forged, the altered packets will be classified into the forged flow and will receive the corresponding quality of service; this will create a denial of service attack subtly different from one where only the

addresses are forged. Because it is limited to a single flow definition, e.g., to a limited amount of bandwidth, such an attack will be more specific and at a finer granularity than a normal address-spoofing attack.

Since flows are identified by the complete 3-tuple, ingress filtering [INGR] will, as noted above, mitigate part of the risk. If the source address of a packet is validated by ingress filtering, there can be a degree of trust that the packet has not transited a compromised router, to the extent that ISP infrastructure may be trusted. However, this gives no assurance that another form of man-in-the-middle attack has not occurred.

Only applications with an appropriate privilege in a sending host will be entitled to set a non-zero Flow Label. Mechanisms for this are operating system dependent. Related policy and authorization mechanisms may also be required; for example, in a multi-user host, only some users may be entitled to set the Flow Label. Such authorization issues are outside the scope of this specification.

5.2. IPsec and Tunneling Interactions

The IPsec protocol, as defined in [IPSec, AH, ESP], does not include the IPv6 header's Flow Label in any of its cryptographic calculations (in the case of tunnel mode, it is the outer IPv6 header's Flow Label that is not included). Hence modification of the Flow Label by a network node has no effect on IPsec end-to-end security, because it cannot cause any IPsec integrity check to fail. As a consequence, IPsec does not provide any defense against an adversary's modification of the Flow Label (i.e., a man-in-the-middle attack).

IPsec tunnel mode provides security for the encapsulated IP header's Flow Label. A tunnel mode IPsec packet contains two IP headers: an outer header supplied by the tunnel ingress node and an encapsulated inner header supplied by the original source of the packet. When an IPsec tunnel is passing through nodes performing flow classification, the intermediate network nodes operate on the Flow Label in the outer header. At the tunnel egress node, IPsec processing includes removing the outer header and forwarding the packet (if required) using the inner header. The IPsec protocol requires that the inner header's Flow Label not be changed by this decapsulation processing to ensure that modifications to label cannot be used to launch theft- or denial-of-service attacks across an IPsec tunnel endpoint. This document makes no change to that requirement; indeed it forbids changes to the Flow Label.

When IPsec tunnel egress decapsulation processing includes a sufficiently strong cryptographic integrity check of the encapsulated packet (where sufficiency is determined by local security policy), the tunnel egress node can safely assume that the Flow Label in the inner header has the same value as it had at the tunnel ingress node.

This analysis and its implications apply to any tunneling protocol that performs integrity checks. Of course, any Flow Label set in an encapsulating IPv6 header is subject to the risks described in the previous section.

5.3. Security Filtering Interactions

The Flow Label does nothing to eliminate the need for packet filtering based on headers past the IP header, if such filtering is deemed necessary for security reasons on nodes such as firewalls or filtering routers.

6. Acknowledgements

The discussion on the topic in the IPv6 WG mailing list has been instrumental for the definition of this specification. The authors want to thank Ran Atkinson, Steve Blake, Jim Bound, Francis Dupont, Robert Elz, Tony Hain, Robert Hancock, Bob Hinden, Christian Huitema, Frank Kastenholz, Thomas Narten, Charles Perkins, Pekka Savola, Hesham Soliman, Michael Thomas, Margaret Wasserman, and Alex Zinin for their contributions.

7. References

7.1. Normative References

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

[KEYWORDS] Bradner, S., "Key words for use in RFCs to indicate requirement levels", BCP 14, RFC 2119, March 1997.

[RND] Eastlake, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.

7.2. Informative References

[AH] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[ESP] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[INGR] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[IPSec] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

Authors' Addresses

Jarno Rajahalme
Nokia Research Center
P.O. Box 407
FIN-00045 NOKIA GROUP,
Finland

EMail: jarno.rajahalme@nokia.com

Alex Conta
Transwitch Corporation
3 Enterprise Drive
Shelton, CT 06484
USA

EMail: aconta@txc.com

Brian E. Carpenter
IBM Zurich Research Laboratory
Saeumerstrasse 4 / Postfach
8803 Rueschlikon
Switzerland

EMail: brc@zurich.ibm.com

Steve Deering
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.