

Network Working Group  
INTERNET-DRAFT  
Expires in: January 2006

Jerry Perser  
  
Scott Poretsky  
Reef Point Systems

Shobha Erramilli  
Qnetworx

Sumit Khurana  
Telcordia

July 2005

## **Terminology for Benchmarking Network-layer Traffic Control Mechanisms**

[<draft-ietf-bmwg-dsmterm-11.txt>](#)

Intellectual Property Rights (IPR) statement:

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Copyright Notice

Copyright (C) The Internet Society (2005). All Rights Reserved.

Abstract

This document describes terminology for the benchmarking of devices that implement traffic control based on IP precedence or diff-serv code point criteria. The terminology is to be applied

to measurements made on the data plane to evaluate IP traffic control mechanisms.

## Table of Contents

|  |                    |
|--|--------------------|
| <a href="#">1. Introduction</a>                              | <a href="#">3</a>  |
| <a href="#">2. Existing definitions</a>                      | <a href="#">3</a>  |
| <a href="#">3. Term definitions</a>                          | <a href="#">4</a>  |
| 3.1 Configuration Terms                                      |                    |
| <a href="#">3.1.1 Classification</a>                         | <a href="#">4</a>  |
| <a href="#">3.1.2 Codepoint Set</a>                          | <a href="#">4</a>  |
| <a href="#">3.1.3 Forwarding Congestion</a>                  | <a href="#">5</a>  |
| <a href="#">3.1.4 Congestion Management</a>                  | <a href="#">6</a>  |
| <a href="#">3.1.5 Flow</a>                                   | <a href="#">7</a>  |
| 3.2 Measurement Terms  |                    |
| <a href="#">3.2.1 Forwarding Capacity</a>                    | <a href="#">7</a>  |
| <a href="#">3.2.2 Conforming Packet</a>                      | <a href="#">8</a>  |
| <a href="#">3.2.3 Nonconforming Packet</a>                   | <a href="#">9</a>  |
| <a href="#">3.2.4 Forwarding Delay</a>                       | <a href="#">9</a>  |
| <a href="#">3.2.5 Jitter</a>                                 | <a href="#">11</a> |
| <a href="#">3.2.6 Undifferentiated Response</a>              | <a href="#">11</a> |
| 3.3 Sequence Tracking  |                    |
| <a href="#">3.3.1 In-sequence Packet</a>                     | <a href="#">12</a> |
| <a href="#">3.3.2 Out-of-order Packet</a>                    | <a href="#">12</a> |
| <a href="#">3.3.3 Duplicate Packet</a>                       | <a href="#">13</a> |
| <a href="#">3.3.4 Stream</a>                                 | <a href="#">14</a> |
| <a href="#">3.3.5 Test Sequence number</a>                   | <a href="#">15</a> |
| <a href="#">3.4 Vectors</a>                                  | <a href="#">15</a> |
| <a href="#">3.4.1 Intended Vector</a>                        | <a href="#">15</a> |
| <a href="#">3.4.2 Offered Vector</a>                         | <a href="#">16</a> |
| 3.4.3 Expected Vectors                                       |                    |
| <a href="#">3.4.3.1 Expected Forwarding Vector</a>           | <a href="#">16</a> |
| <a href="#">3.4.3.2 Expected Loss Vector</a>                 | <a href="#">17</a> |
| <a href="#">3.4.3.3 Expected Sequence Vector</a>             | <a href="#">18</a> |
| <a href="#">3.4.3.4 Expected Instantaneous Delay Vector</a>  | <a href="#">18</a> |
| <a href="#">3.4.3.5 Expected Average Delay Vector</a>        | <a href="#">19</a> |
| <a href="#">3.4.3.6 Expected Maximum Delay Vector</a>        | <a href="#">10</a> |
| <a href="#">3.4.3.7 Expected Minimum Delay Vector</a>        | <a href="#">20</a> |
| <a href="#">3.4.3.8 Expected Instantaneous Jitter Vector</a> | <a href="#">21</a> |
| <a href="#">3.4.3.9 Expected Average Jitter Vector</a>       | <a href="#">22</a> |
| <a href="#">3.4.3.10 Expected Peak-to-peak Jitter Vector</a> | <a href="#">22</a> |
| 3.4.4 Output Vectors   |                    |
| <a href="#">3.4.4.1 Forwarding Vector</a>                    | <a href="#">23</a> |
| <a href="#">3.4.4.2 Loss Vector</a>                          | <a href="#">24</a> |
| <a href="#">3.4.4.3 Sequence Vector</a>                      | <a href="#">24</a> |
| <a href="#">3.4.4.4 Instantaneous Delay Vector</a>           | <a href="#">25</a> |
| <a href="#">3.4.4.5 Average Delay Vector</a>                 | <a href="#">26</a> |
| <a href="#">3.4.4.6 Maximum Delay Vector</a>                 | <a href="#">27</a> |
| <a href="#">3.4.4.7 Minimum Delay Vector</a>                 | <a href="#">28</a> |
| <a href="#">3.4.4.8 Instantaneous Jitter Vector</a>          | <a href="#">28</a> |
| <a href="#">3.4.4.9 Average Jitter Vector</a>                | <a href="#">29</a> |
| <a href="#">3.4.4.10 Peak-to-peak Jitter Vector</a>          | <a href="#">30</a> |

|   |                    |
|---|--------------------|
| <a href="#">4. IANA Considerations.....</a>     | <a href="#">31</a> |
| <a href="#">5. Security Considerations.....</a> | <a href="#">31</a> |
| <a href="#">6. Acknowledgments.....</a>         | <a href="#">31</a> |

|  |                    |
|--|--------------------|
| <a href="#">7. References.....</a>               | <a href="#">32</a> |
| <a href="#">8. Author's Address.....</a>         | <a href="#">33</a> |
| <a href="#">9. Full Copyright Statement.....</a> | <a href="#">34</a> |

## [1. Introduction](#)

New terminology is needed because most existing measurements assume the absence of congestion and only a single per-hop-behavior. This document introduces several new terms that will allow measurements to be taken during periods of congestion.

Another key difference from existing terminology is the definition of measurements as observed on egress as well as ingress of a device/system under test. Again, the existence of congestion requires the addition of egress measurements as well as those taken on ingress; without observing traffic leaving a device/system it is not possible to say whether traffic-control mechanisms effectively dealt with congestion.

The principal measurements introduced in this document are vectors for rate, delay, and jitter, all of which can be observed with or without congestion of the DUT/SUT.

This document describes only those terms relevant to measuring behavior of a device or a group of devices using one of these two mechanisms. End-to-end and service-level measurements are beyond the scope of this document.

## [2. Existing definitions](#)

[RFC 1242](#) "Benchmarking Terminology for Network Interconnect Devices" and [RFC 2285](#) "Benchmarking Terminology for LAN Switching Devices" SHOULD be consulted before attempting to make use of this document.

[RFC 2474](#) "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers" [section 2](#), contains discussions of a number of terms relevant to network-layer traffic control mechanisms and SHOULD also be consulted.

For the sake of clarity and continuity this RFC adopts the template for definitions set out in [Section 2 of RFC 1242](#). Definitions are indexed and grouped together in sections for ease of reference.

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](#) [Br97]. [RFC 2119](#) defines the use of these key words to help make the intent of standards track documents as clear as possible. While this

document uses these keywords, this document is not a standards track document.

Perser, Poretsky, Khurana, Erramilli

[Page 3]

### **3. Term definitions**

#### **3.1 Configuration Terms**

##### **3.1.1 Classification**

**Definition:**

Selection of packets based on the contents of packet header according to defined rules.

**Discussion:**

Packets can be selected based on the DS field or IP Precedence in the packet header. Classification can also be based on Multi-Field (MF) criteria such as IP Source and destination addresses, protocol and port number.

Classification determines the per-hop behaviors and traffic conditioning functions such as shaping and dropping that are to be applied to the packet.

Measurement units: n/a

See Also:

##### **3.1.2 Codepoint Set**

**Definition:**

The set of all DS Code-points or IP precedence values used during the test duration.

**Discussion:**

Describes all the code-point markings associated with packets that are input to the DUT/SUT. For each entry in the codepoint set, there are associated vectors describing the rate of traffic, delay, loss, or jitter containing that particular DSCP or IP precedence value.

The treatment that a packet belonging to a particular code-point gets is subject to the DUT classifying packets to map to the correct PHB. Moreover, the forwarding treatment in general is also dependent on the complete set of offered vectors.

Measurement Units: n/a

See Also: None





### 3.1.3 Forwarding Congestion

Definition:

A condition in which one or more egress interfaces are offered more packets than are forwarded.

Discussion:

This condition is a superset of the overload definition [Ma98]. Overload [Ma98] deals with overloading input and output interfaces beyond the maximum transmission allowed by the medium. Forwarding congestion does not assume ingress interface overload as the only source of overload on output interfaces.

Another difference between Forwarding Congestion and overload occurs when the SUT comprises multiple elements, in that Forwarding Congestion may occur at multiple points. Consider an SUT comprising multiple edge devices exchanging traffic with a single core device. Depending on traffic patterns, the edge devices may induce Forwarding Congestion on multiple egress interfaces on the core device.

Packet Loss, not increased Delay, is the metric to indicate the condition of Forwarding Congestion. Packet Loss is a deterministic indicator of Forwarding Congestion. While increased delay may be an indicator of Forwarding Congestion, it is a non-deterministic indicator of Forwarding Congestion. External observation of increased delay to indicate congestion is in effect external observation of Incipient Congestion.

[Ra99] implies that it is impractical to build a black-box test to externally observe Incipient Congestion indicated by increased delay in a router. [Ra99] introduces Explicit Congestion Notification (ECN) as the externally observable, deterministic method for indicating Incipient Congestion. Because [Ra99] is an Experimental RFC with limited deployment, ECN is not used for this particular methodology. For the purpose of "black-box" testing a DUT/SUT, Packet Loss as the indicator of Forwarding Congestion is used.

Throughput [Br91] defines the lower boundary of Forwarding Congestion. Throughput is the maximum offered rate with no Forwarding Congestion. At offered rates above throughput, the DUT/SUT is considered to be in a state of Forwarding Congestion.



Ingress observations alone are not sufficient to cover all cases in which Forwarding Congestion may occur. A device with an infinite amount of memory could buffer an infinite number of packets, and eventually forward all of them. However, these packets may or may not be forwarded during the test duration. Even though ingress interfaces accept all packets without loss, Forwarding Congestion is present in this hypothetical device.

Forwarding Congestion, indicated by occurrence of packet loss, is one type of congestion for a DUT/SUT. Congestion Collapse [[Na84](#)] is defined as the state in which buffers are full and all arriving packets MUST be dropped across the network. Incipient Congestion [[Ra99](#)] is defined as congestion that produces increased delay without packet loss.

The definition presented here explicitly defines Forwarding Congestion as an event observable on egress interfaces. Regardless of internal architecture, any device that cannot forward packets on one or more egress interfaces is under Forwarding Congestion.

Measurement units:  
none

See Also:  
Gateway Congestion Control Survey [[Ma91](#)]

### 3.1.4 Congestion Management

Definition:  
An implementation of one or more per-hop-behaviors to avoid or minimize the condition of congestion.

Discussion:  
Congestion management may seek either to control congestion or avoid it altogether. Such mechanisms classify packets based upon IP Precedence or DSCP settings in a packets IP header.

Congestion avoidance mechanisms seek to prevent congestion before it actually occurs.

Congestion control mechanisms give one or more flows (with a discrete IP Precedence or DSCP value) preferential treatment over other classes during periods of congestion.

Measurement units:  
n/a

See Also:

Perser, Poretsky, Khurana, Erramilli

[Page 6]

### 3.1.5 Flow

Definition:

A flow is a one or more of packets sharing a common intended pair of source and destination interfaces.

Discussion:

Packets are grouped by the ingress and egress interfaces they use on a given DUT/SUT.

A flow can contain multiple source IP addresses and/or destination IP addresses. All packets in a flow MUST enter on the same ingress interface and exit on the same egress interface, and have some common network layer content.

Microflows [[Ni98](#)] are a subset of flows. As defined in [[Ni98](#)], microflows require application-to-application measurement. In contrast, flows use lower-layer classification criteria. Since this document focuses on network-layer classification criteria, we concentrate here on the use of network-layer identifiers in describing a flow. Flow identifiers also may reside at the data-link, transport, or application layers of the OSI model. However, identifiers other than those at the network layer are out of scope for this document.

A flow may contain a single code point/IP precedence value or may contain multiple values destined for a single egress interface. This is determined by the test methodology.

Measurement units:

n/a

See Also:

Microflow [[Ni98](#)]  
Streams

## 3.2 Measurement Terms

### 3.2.1 Forwarding Capacity

Definition:

The number of packets per second that a device can be observed to successfully transmit to the correct destination interface in response to a specified offered load while the device drops none of the offered packets.



**Discussion:**

Forwarding Capacity measures the packet rate at the egress interface(s) of the DUT/SUT. In contrast, throughput as defined in [RFC 1242](#) measures the frame rate at the ingress interface(s) of the DUT/SUT.

Ingress-based measurements do not account for queuing of the DUT/SUT. Throughput rates can be higher than the Forwarding Capacity because of queuing. The difference is dependent upon test duration, packet rate, and queue size. Forwarding Capacity, as an egress measurement, does take queuing into account.

Understanding Forwarding Capacity is a necessary precursor to any measurement involving Traffic Control Mechanisms. The accompanying methodology document MUST take into consideration Forwarding Capacity when determining the expected forwarding vectors. When the sum of the expected forwarding vectors on an interface exceeds the Forwarding Capacity, the Forwarding Capacity will govern the forwarding rate.

This measurement differs from forwarding rate at maximum offered load (FRMOL) [[Ma98](#)] in that Forwarding Capacity requires zero loss.

**Measurement units:**

N-octet packets per second

**See Also:**

Throughput [[Br91](#)]

Forwarding Rate at Maximum Offered Load [[Ma98](#)]

### 3.2.2 Conforming Packet

**Definition:**

Packets which lie within specific rate, delay, or jitter bounds.

**Discussion:**

A DUT/SUT may be configured to allow a given traffic class to consume a given amount of bandwidth, or to fall within predefined delay or jitter boundaries. All packets that lie within specified bounds are then said to be conforming, whereas those outside the bounds are nonconforming.

**Measurement units:**

n/a





See Also:

- Expected Vector
- Forwarding Vector
- Offered Vector
- Nonconforming

### 3.2.3 Nonconforming Packet

Definition:

Packets that do not lie within specific rate, delay, or jitter bounds.

Discussion:

A DUT/SUT may be configured to allow a given traffic class to consume a given amount of bandwidth, or to fall within predefined delay or jitter boundaries. All packets that do not lie within these bounds are then said to be nonconforming.

Measurement units:

n/a

See Also:

- Expected Vector
- Forwarding Vector
- Offered Vector
- Conforming

### 3.2.4 Forwarding Delay

Definition:

The time interval starting when the last bit of the input IP packet is offered to the input port of the DUT/SUT and ending when the last bit of the output IP packet is received from the output port of the DUT/SUT.

Discussion:

The delay time interval MUST be externally observed. The delay measurement MUST NOT include delays added by test bed components other than the DUT/SUT, such as propagation time introduced by cabling or non-zero delay added by the test instrument.

Forwarding Delay differs from latency [[Br91](#)] and one-way delay [[A199](#)] in several key regards:

1. Latency [[Br91](#)] assumes knowledge of whether the DUT/SUT uses "store and forward" or "bit forwarding" technology.

Forwarding Delay is the same metric, measured the same way,  
regardless of the architecture of the DUT/SUT.

2. Forwarding Delay is a last-in, last-out (LILO) measurement, unlike the last-in, first-out method [[Br91](#)] or the first-in, last-out method [[A199](#)].

The LILO method most closely simulates the way a network-layer device actually processes an IP datagram. IP datagrams are not passed up and down the stack unless they are complete, and processing begins only once the last bit of the IP datagram has been received.

Further, the LILO method has an additive property, where the sum of the parts MUST equal the whole. This is a key difference from [[Br91](#)] and [[A199](#)]. For example, the delay added by two DUTs MUST equal the sum of the delay of the DUTs. This may or may not be the case with [[Br91](#)] and [[A199](#)].

3. Forwarding Delay measures the IP datagram only, unlike [[Br91](#)], which also includes link layer overhead.

A metric focused exclusively on the Internet protocol relieves the tester from specifying the start/end for every link layer protocol that IP runs on. This avoids the need to determine whether the start/stop delimiters are included. It also allows the use of heterogeneous link layer protocols in a test.

4. Forwarding Delay can be measured at any offered load, whereas the latency methodology [[Br99](#)] recommends measurement at, and only at, the throughput level. Comparing the Forwarding Delay below the throughput to Forwarding Delay above the Forwarding Capacity will give insight to the traffic control mechanisms.

For example, non-congested delay may be measured with an offered load that does not exceed the Forwarding Capacity, while congested delay may involve an offered load that exceeds Forwarding Capacity.

Note: Forwarding Delay SHOULD NOT be used as an absolute indicator of DUT/SUT Forwarding Congestion. While Forwarding Delay may rise when offered load nears or exceeds Forwarding Capacity, there is no universal point at which Forwarding Delay can be said to indicate the presence or absence of Forwarding Congestion.

Measurement units:  
Seconds.



See Also:

Latency [[Br91](#)]

Latency [[A199](#)]

One-way Delay [[Br99](#)]

### 3.2.5 Jitter

Definition:

The absolute value of the difference between the arrival delay of two consecutive received packets belonging to the same stream.

Discussion:

The delay fluctuation between two consecutive received packets in a stream is reported as the jitter. Jitter can be expressed as  $|D(i) - D(i-1)|$  where  $D$  equals the delay and  $i$  is the order the packets were received.

Under loss, jitter can be measured between non-consecutive test sequence numbers. When IP Traffic Control Mechanisms are dropping packets, fluctuating Forwarding Delay may be observed. Jitter MUST be able to benchmark the delay variation independent of packet loss.

Jitter is related to the ipdv [[De02](#)] (IP Delay Variation) by taking the absolute value of the ipdv. The two metrics will produce different mean values. Mean Jitter will produce a positive value, where the mean ipdv is typically zero.

Measurement units:

Seconds

See Also:

Forwarding Delay

Jitter variation [[Ja99](#)]

ipdv [[De02](#)]

interarrival jitter [[Sc96](#)]

### 3.2.6 Undifferentiated Response

Definition:

The vector(s) obtained when mechanisms used to support diff-serv or IP precedence are disabled.

Discussion:

Enabling diff-serv or IP precedence mechanisms may impose additional processing overhead for packets. This overhead may degrade performance even when traffic belonging to only one class, the best-effort class, is offered to the device.



Measurements with "undifferentiated response" SHOULD be made to establish a baseline.

The vector(s) obtained with DSCP or IP precedence enabled can be compared to the undifferentiated response to determine the effect of differentiating traffic.

Measurement units:  
n/a

### 3.3 Sequence Tracking

#### 3.3.1 In-sequence Packet

Definition:

A received packet with the expected Test Sequence number.

Discussion:

In-sequence is done on a stream level. As packets are received on a stream, each packets Test Sequence number is compared with the previous packet. Only packets that match the expected Test Sequence number are considered in-sequence.

Packets that do not match the expected Test Sequence number are counted as "not in-sequence" or out-of-sequence. Every packet that is received is either in-sequence or out-of-sequence. Subtracting the in-sequence from the received packets (for that stream) can derive the out-of-sequence count.

Two types of events will prevent the in-sequence from incrementing: packet loss and reordered packets.

Measurement units:  
Packet count

See Also:

Stream  
Test Sequence number

#### 3.3.2 Out-of-order Packet

Definition:

A received packet with a Test Sequence number less than expected.





**Discussion:**

As a stream of packets enters a DUT/SUT, they include a Stream Test Sequence number indicating the order the packets were sent to the DUT/SUT. On exiting the DUT/SUT, these packets may arrive in a different order. Each packet that was re-ordered is counted as an Out-of-order Packet.

Certain streaming protocol (such as TCP) require the packets to be in a certain order. Packets outside this are dropped by the streaming protocols even though there were properly received by the IP layer. The type of reordering tolerated by a streaming protocol varies from protocol to protocol, and also by implementation.

Out-of-order Packet count is based on the worst case streaming protocol. It allows for no reordering.

Packet loss does not affect the Out-of-order Packet count. Only packets that were not received in the order that they were transmitted.

**Measurement units:**

Packet count

**See Also:**

Stream

Test Sequence number

Packet Reordering Metric for IPPM [[Mo03](#)]

### 3.3.3 Duplicate Packet

**Definition:**

A received packet with a Test Sequence number matching a previously received packet.

**Discussion:**

A Duplicate Packet is a packet that the DUT/SUT has successfully transmitted out an egress interface more than once. The egress interface has previously forwarded this packet.

A Duplicate Packet SHOULD be a bit for bit copy of an already transmitted packet (including Test Sequence number). If the Duplicate Packet traversed different paths through the DUT/SUT, some fields (such as TTL or checksum) may have changed.



A multicast packet is not a Duplicate Packet by definition. For a given IP multicast group, a DUT/SUT SHOULD forward a packet once on a given egress interface provided the path to one or more multicast receivers is through that interface. Several egress interfaces will transmit the same packet, but only once per interface.

To detect a Duplicate Packet, each offered packet to the DUT/SUT MUST contain a unique packet-by-packet identifier.

Measurement units:

Packet count

See Also:

Stream

Test Sequence number

#### 3.3.4 Stream

Definition:

A group of packets tracked as a single entity by the traffic receiver. A stream may share a common content such as type (IP, UDP), packet size, or payload.

Discussion:

Streams are tracked by test sequence number or "unique signature field" [[Ma00](#)]. Streams define how individual packets statistic are grouped together to form an intelligible summary.

Common stream groupings would be by egress interface, destination address, source address, DSCP, or IP precedence. A stream using test sequence numbers can track the ordering of packets as they transverse the DUT/SUT.

Streams are not restricted to a pair of source and destination interfaces as long as all packets are tracked as a single entity. A multicast stream can be forward to multiple destination interfaces.

Measurement units:

n/a

See Also:

Flow

Microflow [[Ni98](#)]

Test sequence number



### 3.3.5 Test Sequence number

Definition:

A field in the IP payload portion of the packet that is used to verify the order of the packets on the egress of the DUT/SUT.

Discussion:

The traffic generator sets the test sequence number value and the traffic receiver checks the value upon receipt of the packet. The traffic generator changes the value on each packet transmitted based on an algorithm agreed to by the traffic receiver.

The traffic receiver keeps track of the sequence numbers on a per stream basis. In addition to number of received packets, the traffic receiver may also report number of in-sequence packets, number of out-of-sequence packets, number of duplicate packets, and number of reordered packets.

The RECOMMENDED algorithm to use to change the sequence number on sequential packets is an incrementing value.

Measurement units:

n/a

See Also:

Stream

### 3.4 Vectors

A vector is a group of packets all containing a specific DSCP or IP precedence value. Vectors are expressed as a pair of numbers. The first is being the particular diff-serv value. The second is the metric expressed as a rate, loss percentage, delay, or jitter.

#### 3.4.1 Intended Vector

Definition:

A vector describing the attempted rate at which packets having a specific code-point (or IP precedence) are transmitted to a DUT/SUT by an external source.

Discussion:

Intended loads across the different code-point classes determine the metrics associated with a specific code-point traffic class.

Measurement Units:

N-octets packets per second

Perser, Poretsky, Khurana, Erramilli

[Page 15]

## See Also:

- Offered Vector
- Expected Forwarding Vector
- Expected Loss Vector
- Expected Sequence Vector
- Expected Delay Vector
- Expected Jitter Vector
- Forwarding Vector
- Loss Vector

## 3.4.2 Offered Vector

## Definition:

A vector describing the measured rate at which packets having a specific DSCP or IP precedence value are offered to the DUT/SUT.

## Discussion:

Offered loads across the different code-point classes, constituting a code-point set, determine the metrics associated with a specific code-point traffic class.

## Measurement Units:

N-octets packets per second

## See Also:

- Expected Forwarding Vector
- Expected Loss Vector
- Expected Sequence Vector
- Expected Delay Vector
- Expected Jitter Vector
- Forwarding Vector
- Codepoint Set

## 3.4.3 Expected Vectors

## 3.4.3.1 Expected Forwarding Vector

## Definition:

A vector describing the expected output rate of packets having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of the DUT.





**Discussion:**

The DUT is configured in a certain way in order that service differentiation occurs for a particular DSCP or IP precedence value when a specific traffic mix consisting of multiple DSCPs or IP precedence values are applied. This term attempts to capture the expected forwarding behavior when subjected to a certain offered vectors.

The actual algorithm or mechanism the DUT uses to achieve service differentiation is not important in describing the expected forwarding vector.

**Measurement units:**

N-octet packets per second

**See Also:**

- Intended Vector
- Offered Vector
- Output Vectors
- Expected Loss Vector
- Expected Sequence Vector
- Expected Delay Vector
- Expected Jitter Vector

**3.4.3.2 Expected Loss Vector****Definition:**

A vector describing the percentage of packets, having a specific DSCP or IP precedence value that SHOULD not be forwarded. The value is dependent on the set of offered vectors and configuration of the DUT.

**Discussion:**

The DUT is configured in a certain way in order that service differentiation occurs for a particular DSCP or IP precedence value when a specific traffic mix consisting of multiple DSCPs or IP precedence values are applied. This term attempts to capture the expected forwarding behavior when subjected to a certain offered vector.

The actual algorithm or mechanism the DUT uses to achieve service differentiation is not important in describing the expected loss vector.

**Measurement Units:**

Percentage of intended packets that is expected to be dropped.



## See Also:

- Intended Vector
- Offered Vector
- Expected Forwarding Vector
- Expected Sequence Vector
- Expected Delay Vector
- Expected Jitter Vector
- One-way Packet Loss Metric [[Ka99](#)]

## 3.4.3.3 Expected Sequence Vector

## Definition:

A vector describing the expected in-sequence packets having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of the DUT.

## Discussion:

The DUT is configured in a certain way in order that service differentiation occurs for a particular DSCP or IP precedence value when a specific traffic mix consisting of multiple DSCPs or IP precedence values are applied. This term attempts to capture the expected forwarding behavior when subjected to a certain offered vectors.

The actual algorithm or mechanism the DUT uses to achieve service differentiation is not important in describing the expected sequence vector.

## Measurement Units:

N-octet packets per second

## See Also:

- Intended Vector
- Offered Vector
- Output Vectors
- Expected Loss Vector
- Expected Forwarding Vector
- Expected Delay Vector
- Expected Jitter Vector

## 3.4.3.4 Expected Instantaneous Delay Vector

## Definition:

A vector describing the expected delay for packets having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of the DUT.



**Discussion:**

The DUT is configured in a certain way in order that service differentiation occurs for a particular DSCP or IP precedence value when a specific traffic mix consisting of multiple DSCPs or IP precedence values are applied. This term attempts to capture the expected forwarding behavior when subjected to a certain offered vectors.

The actual algorithm or mechanism the DUT uses to achieve service differentiation is not important in describing the expected delay vector.

**Measurement units:**

Seconds.

**See Also:**

- Forwarding Delay
- Intended Vector
- Offered Vector
- Output Vectors
- Expected Loss Vector
- Expected Sequence Vector
- Expected Forwarding Vector
- Expected Jitter Vector

#### 3.4.3.5 Expected Average Delay Vector

**Definition:**

A vector describing the expected average delay for packets having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of the DUT.

**Discussion:**

The DUT is configured in a certain way in order that service differentiation occurs for a particular DSCP or IP precedence value when a specific traffic mix consisting of multiple DSCPs or IP precedence values are applied. This term attempts to capture the expected forwarding behavior when subjected to a certain offered vectors.

The actual algorithm or mechanism the DUT uses to achieve service differentiation is not important in describing the expected average delay vector.

**Measurement units:**

Seconds.



## See Also:

- Intended Vector
- Offered Vector
- Output Vectors
- Expected Loss Vector
- Expected Sequence Vector
- Expected Forwarding Vector
- Expected Jitter Vector

## 3.4.3.6 Expected Maximum Delay Vector

## Definition:

A vector describing the expected maximum delay for packets having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of the DUT.

## Discussion:

The DUT is configured in a certain way in order that service differentiation occurs for a particular DSCP or IP precedence value when a specific traffic mix consisting of multiple DSCPs or IP precedence values are applied. This term attempts to capture the expected forwarding behavior when subjected to a certain offered vectors.

The actual algorithm or mechanism the DUT uses to achieve service differentiation is not important in describing the expected maximum delay vector.

## Measurement units:

Seconds.

## See Also:

- Intended Vector
- Offered Vector
- Output Vectors
- Expected Loss Vector
- Expected Sequence Vector
- Expected Forwarding Vector
- Expected Jitter Vector

## 3.4.3.7 Expected Minimum Delay Vector

## Definition:

A vector describing the expected minimum delay for packets having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of

the DUT.



**Discussion:**

The DUT is configured in a certain way in order that service differentiation occurs for a particular DSCP or IP precedence value when a specific traffic mix consisting of multiple DSCPs or IP precedence values are applied. This term attempts to capture the expected forwarding behavior when subjected to a certain offered vectors.

The actual algorithm or mechanism the DUT uses to achieve service differentiation is not important in describing the expected minimum delay vector.

**Measurement units:**

Seconds.

**See Also:**

- Intended Vector
- Offered Vector
- Output Vectors
- Expected Loss Vector
- Expected Sequence Vector
- Expected Forwarding Vector
- Expected Jitter Vector

### 3.4.3.8 Expected Instantaneous Jitter Vector

**Definition:**

A vector describing the expected jitter between two consecutive packets arrival times having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of the DUT.

**Discussion:**

Instantaneous Jitter is the absolute value of the difference between the delay measurement of two packets belonging to the same stream.

The delay fluctuation between two consecutive packets in a stream is reported as the "Instantaneous Jitter".

Instantaneous Jitter can be expressed as  $|D(i) - D(i-1)|$  where  $D$  equals the delay and  $i$  is the test sequence number. Packets lost are not counted in the measurement.

Forwarding Vector may contain several Jitter Vectors. For  $n$  packets received in a Forwarding Vector, there is a total of  $(n-1)$  Instantaneous Jitter Vectors.

**Measurement units:**

Seconds

Perser, Poretsky, Khurana, Erramilli

[Page 21]

## See Also:

- Delay
- Jitter
- Offered Vector
- Output Vectors
- Expected Average Jitter Vector
- Expected Peak-to-peak Jitter Vector
- Stream

## 3.4.3.9 Expected Average Jitter Vector

## Definition:

A vector describing the expected jitter in packet arrival times for packets having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of the DUT.

## Discussion:

Average Jitter Vector is the average of all the Instantaneous Jitter Vectors measured during the test duration for the same DSCP or IP precedence value.

## Measurement units:

Seconds

## See Also:

- Intended Vector
- Offered Vector
- Output Vectors
- Expected Instantaneous Jitter Vector
- Expected Peak-to-peak Jitter Vector

## 3.4.3.10 Expected Peak-to-peak Jitter Vector

## Definition:

A vector describing the expected maximum variation in the delay of packet arrival times for packets having a specific DSCP or IP precedence value. The value is dependent on the set of offered vectors and configuration of the DUT.

## Discussion:

Peak-to-peak Jitter Vector is the maximum delay minus the minimum delay of the packets (in a vector) forwarded by the DUT/SUT.

Peak-to-peak Jitter is not derived from the Instantaneous Jitter Vector. Peak-to-peak Jitter is based upon all the

packets during the test duration, not just two consecutive packets.

Measurement units:

Seconds

See Also:

Intended Vector

Offered Vector

Output Vectors

Expected Instantaneous Jitter Vector

Expected Average Jitter Vector

### 3.4.4 Output Vectors

#### 3.4.4.1 Forwarding Vector

Definition:

The number of packets per second for all packets containing a specific DSCP or IP precedence value that a device can be observed to successfully forward to the correct destination interface in response to an offered vector.

Discussion:

Forwarding Vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND the packets per second value combine to make a vector.

The Forwarding Vector represents packet rate based on its specific DSCP (or IP precedence) value. It is not necessarily based on a stream or flow. The Forwarding Vector may be expressed as per port of the DUT/SUT. However, it MUST be consistent with the Expected Forwarding Vector.

Forwarding Vector is a per-hop measurement. The DUT/SUT may change the specific DSCP (or IP precedence) value for a multiple-hop measurement.

Measurement units:

N-octet packets per second

See Also:

Intended Vector

Offered Vector

Expected Vectors

Loss Vector

Sequence Vector

Delay Vectors



#### 3.4.4.2 Loss Vector

Definition:

The percentage of packets containing a specific DSCP or IP precedence value that a DUT/SUT did not transmit to the correct destination interface in response to an offered vector.

Discussion:

Loss Vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND the percentage value combine to make a vector.

The Loss Vector represents percentage based on a specific DSCP or IP precedence value. It is not necessarily based on a stream or flow. The Loss Vector may be expressed as per port of the DUT/SUT. However, it MUST be consistent with the Expected Loss Vector

Loss Vector is a per-hop measurement. The DUT/SUT may change the specific DSCP or IP precedence value for a multiple-hop measurement.

Measurement Units:

Percentage of offered packets that is not forwarded.

See Also:

Intended Vector  
Offered Vector  
Expected Vectors  
Forwarding Vector  
Sequence Vector  
Delay Vectors  
One-way Packet Loss Metric [[Ka99](#)]

#### 3.4.4.3 Sequence Vector

Definition:

The number of packets per second for all packets containing a specific DSCP or IP precedence value that a device can be observed to transmit in sequence to the correct destination interface in response to an offered vector.

Discussion:

Sequence Vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND the packets per second value combine to make a vector.





The Sequence Vector represents packet rate based on its specific DSCP or IP precedence value. It is not necessarily based on a stream or flow. The Sequence Vector may be expressed as per port of the DUT/SUT. However, it MUST be consistent with the Expected Sequence Vector.

Sequence Vector is a per-hop measurement. The DUT/SUT may change the specific DSCP or IP precedence value for a multiple-hop measurement.

Measurement Units:

N-octet packets per second

Issues:

See Also:

In-sequence Packet  
Intended Vector  
Offered Vector  
Expected Vectors  
Loss Vector  
Forwarding Vector  
Delay Vectors

#### 3.4.4.4 Instantaneous Delay Vector

Definition:

The delay for a packet containing a specific DSCP or IP precedence value that a device can be observed to successfully transmit to the correct destination interface in response to an offered vector.

Discussion:

Instantaneous Delay vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND delay value combine to make a vector.

The Instantaneous Delay Vector represents delay on its specific DSCP or IP precedence value. It is not necessarily based on a stream or flow. The Delay vector may be expressed as per port of the DUT/SUT. However, it MUST be consistent with the Expected Delay vectors.

Instantaneous Delay Vector is a per-hop measurement. The DUT/SUT may change the specific DSCP or IP precedence value for a multiple-hop measurement.

Instantaneous Delay vector can be obtained at any offered load. RECOMMEND at or below the Forwarding Capacity in the

absence of forwarding congestion. For congested delay, run the offered load above the Forwarding Capacity.

Forwarding Vector may contain several Instantaneous Delay Vectors. For every packet received in a Forwarding Vector, there is a corresponding Instantaneous Delay Vector.

Measurement Units:

Seconds

See Also:

Delay  
Intended Vector  
Offered Vector  
Expected Delay Vectors  
Average Delay Vector  
Maximum Delay Vector  
Minimum Delay Vector

#### 3.4.4.5 Average Delay Vector

Definition:

The average delay for packets containing a specific DSCP or IP precedence value that a device can be observed to successfully transmit to the correct destination interface in response to an offered vector.

Discussion:

Average Delay vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND delay value combine to make a vector.

The Delay Vector represents delay on its specific DSCP or IP precedence value. It is not necessarily based on a stream or flow. The Delay vector may be expressed as per port of the DUT/SUT. However, it MUST be consistent with the Expected Delay vector.

The Average Delay Vector is computed by averaging all the Instantaneous Delay Vectors for a given vector.

Average Delay Vector is a per-hop measurement. The DUT/SUT may change the specific DSCP or IP precedence value for a multiple-hop measurement.

Average Delay vector can be obtained at any offered load. Recommend at or below the Forwarding Capacity in the absence of congestion. For congested delay, run the offered load above the Forwarding Capacity.

Measurement Units:

Seconds

Perser, Poretsky, Khurana, Erramilli

[Page 26]

## See Also:

- Delay
- Intended Vector
- Offered Vector
- Expected Delay Vectors
- Instantaneous Delay Vector
- Maximum Delay Vector
- Minimum Delay Vector

## 3.4.4.6 Maximum Delay Vector

## Definition:

The maximum delay from all packets containing a specific DSCP or IP precedence value that a device can be observed to successfully transmit to the correct destination interface in response to an offered vector.

## Discussion:

Maximum Delay vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND delay value combine to make a vector.

The Maximum Delay Vector represents delay on its specific DSCP or IP precedence value. It is not necessarily based on a stream or flow. The Maximum Delay vector may be expressed as per port of the DUT/SUT. However, it MUST be consistent with the Expected Delay vector.

Maximum Delay Vector is based upon the maximum Instantaneous Delay Vector of all packets in a Forwarding Vector.

Maximum Delay Vector is a per-hop measurement. The DUT/SUT may change the specific DSCP or IP precedence value for a multiple-hop measurement.

## Measurement Units:

Seconds

## See Also:

- Delay
- Intended Vector
- Offered Vector
- Expected Delay Vectors
- Instantaneous Delay Vector
- Forwarding Vector
- Average Delay Vector
- Minimum Delay Vector



#### 3.4.4.7 Minimum Delay Vector

Definition:

The minimum delay from all packets containing a specific DSCP or IP precedence value that a device can be observed to successfully transmit to the correct destination interface in response to an offered vector.

Discussion:

Delay vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND delay value combine to make a vector.

The Minimum Delay Vector represents delay on its specific DSCP or IP precedence value. It is not necessarily based on a stream or flow. The Minimum Delay vector may be expressed as per port of the DUT/SUT. However, it MUST be consistent with the Expected Delay vector.

Minimum Delay Vector is based upon the minimum Instantaneous Delay Vector of all packets in a Forwarding Vector.

Minimum Delay Vector is a per-hop measurement. The DUT/SUT may change the specific DSCP or IP precedence value for a multiple-hop measurement.

Minimum Delay vector can be obtained at any offered load. Recommend at or below the Forwarding Capacity in the absence of congestion. For congested delay, run the offered load above the Forwarding Capacity.

Measurement Units:

Seconds

See Also:

Delay  
Intended Vector  
Offered Vector  
Expected Delay Vectors  
Instantaneous Delay Vector  
Forwarding Vector  
Average Delay Vector  
Maximum Delay Vector

#### 3.4.4.8 Instantaneous Jitter Vector

Definition:

The jitter for two consecutive packets containing a specific DSCP or IP precedence value that a device can be observed to

successfully transmit to the correct destination interface in response to an offered vector.



**Discussion:**

Instantaneous Jitter is the absolute value of the difference between the delay measurement of two packets belonging to the same stream.

Jitter vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND jitter value combine to make a vector.

The delay fluctuation between two consecutive packets in a stream is reported as the "Instantaneous Jitter".  
Instantaneous Jitter Vector can be expressed as  $|D(i) - D(i-1)|$  where D equals the delay and i is the test sequence number. Packets lost are not counted in the measurement.

Instantaneous Jitter Vector is a per-hop measurement. The DUT/SUT may change the specific DSCP or IP precedence value for a multiple-hop measurement.

Forwarding Vector may contain several Instantaneous Jitter Vectors. For n packets received in a Forwarding Vector, there are exactly (n-1) Instantaneous Jitter Vectors.

**Measurement units:**

Seconds

**See Also:**

Delay  
Jitter  
Forwarding Vector  
Stream  
Expected Vectors  
Average Jitter Vector  
Peak-to-peak Jitter Vector

**3.4.4.9 Average Jitter Vector****Definition:**

The average jitter for packets containing a specific DSCP or IP precedence value that a device can be observed to successfully transmit to the correct destination interface in response to an offered vector.

**Discussion:**

Average Jitter Vector is the average of all the Instantaneous Jitter Vectors of the same DSCP or IP precedence value, measured during the test duration.

Average Jitter vector is expressed as pair of numbers. Both

the specific DSCP (or IP precedence) value AND jitter value  
combine to make a vector.

Average Jitter vector is a per-hop measurement. The DUT/SUT may change the specific DSCP or IP precedence value for a multiple-hop measurement.

Measurement units:

Seconds

See Also:

Jitter

Forwarding Vector

Stream

Expected Vectors

Instantaneous Jitter Vector

Peak-to-peak Jitter Vector

#### 3.4.4.10 Peak-to-peak Jitter Vector

Definition:

The maximum possible variation in the delay for packets containing a specific DSCP or IP precedence value that a device can be observed to successfully transmit to the correct destination interface in response to an offered vector.

Discussion:

Peak-to-peak Jitter Vector is the maximum delay minus the minimum delay of the packets (in a vector) forwarded by the DUT/SUT.

Jitter vector is expressed as pair of numbers. Both the specific DSCP (or IP precedence) value AND jitter value combine to make a vector.

Peak-to-peak Jitter is not derived from the Instantaneous Jitter Vector. Peak-to-peak Jitter is based upon all the packets during the test duration, not just two consecutive packets.

Measurement units:

Seconds

See Also:

Jitter

Forwarding Vector

Stream

Expected Vectors

Average Jitter Vector

Peak-to-peak Jitter Vector



#### **4. IANA Considerations**

This document requires no IANA considerations.

#### **5. Security Considerations**

Documents of this type do not directly affect the security of the Internet or of corporate networks as long as benchmarking is not performed on devices or systems connected to production networks.

Packets with unintended and/or unauthorized DSCP or IP precedence values may present security issues. Determining the security consequences of such packets is out of scope for this document.

#### **6. Acknowledgments**

The authors gratefully acknowledge the contributions of the IETF's benchmarking working group members in reviewing this document. The authors would like to express our thanks to David Newman for his consistent and valuable assistance throughout the development of this document. The authors would also like to thank Al Morton ([acmorton@att.com](mailto:acmorton@att.com)) and Kevin Dubray ([kdubray@juniper.net](mailto:kdubray@juniper.net)) for their ideas and support.

#### **7. References**

##### **7.1 Normative References**

- [Br91] Bradner, S., "Benchmarking Terminology for Network Interconnection Devices", [RFC 1242](#), July 1991.
- [Br97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997
- [Ma98] Mandeville, R., "Benchmarking Terminology for LAN Switching Devices", [RFC 2285](#), July 1998.
- [Ni98] Nichols, K., Blake, S., Baker, F., Black, D., "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", [RFC 2474](#), December 1998.



## **7.2 Informative References**

- [Al99] Almes, G., Kalidindi, S., Zekauskas, M., "A One-way Delay Metric for IPPM", [RFC 2679](#), September 1999
- [Bl98] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W., "An Architecture for Differentiated Services", [RFC 2475](#), December 1998.
- [Br99] Bradner, S., McQuaid, J. "Benchmarking Methodology for Network Interconnect Devices", [RFC 2544](#), March 1999
- [De02] Demichelis, C., Chimento, P., "IP Packet Delay Variation Metric for IPPM", [RFC 3393](#), November 2002
- [Ja99] Jacobson, V., Nichols, K., Poduri, K., "An Expedited Forwarding PHB", [RFC 2598](#), June 1999
- [Ka99] Almes, G., Kalidindi, S., Zekauskas, M., "A One-way Packet Loss Metric for IPPM", [RFC 2680](#), September 1999
- [Ma91] Mankin, A., Ramakrishnan, K., "Gateway Congestion Control Survey", [RFC 1254](#), August 1991
- [Ma00] Mandeville, R., Perser, J., "Benchmarking Methodology for LAN Switching Devices", [RFC 2889](#), August 2000
- [Mo03] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., Perser, J., "Packet Reordering Metric for IPPM", Work in Progress
- [Na84] Nagle, J., "Congestion Control in IP/TCP Internetworks", [RFC 896](#), January 1984.
- [Ra99] Ramakrishnan, K. and Floyd, S., "A Proposal to add Explicit Congestion Notification (ECN) to IP", [RFC 2481](#), January 1999.
- [Sc96] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", [RFC 1889](#), January 1996





**8. Authors' Addresses**

Jerry Perser  
USA

EMail: [jerry@perser.org](mailto:jerry@perser.org)

Scott Poretsky  
Reef Point Systems  
8 New England Executive Park  
Burlington, MA 01803  
USA

Phone: + 1 508 439 9008  
EMail: [sporetsky@reefpoint.com](mailto:sporetsky@reefpoint.com)

Shobha Erramilli  
QNetworx Inc  
1119 Campus Drive West  
Morganville, NJ 07751  
USA

EMail: [shobha@qnetworx.com](mailto:shobha@qnetworx.com)

Sumit Khurana  
Telcordia Technologies  
445 South Street  
Morristown, NJ 07960  
USA

Phone: + 1 973 829 3170  
EMail: [sumit@research.telcordia.com](mailto:sumit@research.telcordia.com)



## Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

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

