Abstract

This document describes P2PSIP diagnostics. It extends the methods defined in the base protocol for the diagnostics in P2PSIP overlay network. A new method Echo is an efficient method used in the trusted overlays for retrieving useful diagnostic information. The methods and message formats are consistent with P2PSIP base protocol RELOAD, and the diagnostic information mainly comes from the previous version-02 of this document.
Table of Contents

1. Introduction..................................................3
  1.1. Usage Scenarios...........................................3
2. Terminology...................................................4
3. Overview of Functions.........................................4
4. Motivation....................................................5
5. Packets Formats...............................................6
  5.1. Message Codes............................................7
  5.2. Message payloads.........................................8
    5.2.1. Error Codes and Sub-Codes............................8
    5.2.2. diagnostics information..............................9
6. Message........................................................10
  6.1. Ping......................................................10
  6.2. Route_Query...............................................11
  6.3. Echo......................................................13
  6.4. Error responses.........................................15
7. Security Considerations........................................16
8. IANA Considerations...........................................16
9. Acknowledgments...............................................17
10. References..................................................17
    10.1. Normative References................................17
    10.2. Informative References...............................19
Author’s Addresses..............................................20
1. Introduction

P2P systems are self-organizing and ideally require no network management in the traditional sense to set up and to configure individual P2P nodes. P2P service providers may however contemplate usage scenarios where some diagnostics are required. We present a simple connectivity test and some useful diagnostic information that may be used in such diagnostics.

1.1. Usage Scenarios

The common usage scenarios for P2P diagnostics can be broadly categorized in three classes:

a. Automatic diagnostics built into the P2P overlay routing protocol. Nodes perform periodic checks of known neighbors and remove those nodes from the routing tables that fail to respond to connectivity checks [Handling Churn in a DHT]. The unresponsive nodes may however be only temporarily disabled due to some local cryptographic processing overload, disk processing overload or link overload. It is therefore useful to repeat the connectivity checks to see if such nodes have recovered and can be again placed in the routing tables. This process is known as ‘failed node recovery’ and it can be optimized as described in the reference [Handling Churn in a DHT].

b. P2P system diagnostics to check the overall health of the P2P overlay network, the consumption of network bandwidth, problem links and also checks for abusive or malicious nodes. This is not a trivial problem and has been studied in detail for content and streaming P2P overlays, such as for example in [Diagnostic Framework].

Similar work has been reported more recently for P2PSIP overlays as applied to the P2PP protocol [Diagnostics and NAT traversal in P2PP].

c. Diagnostics for a particular node to follow up an individual user complaint. In this case a technical support person may use a desktop sharing application with the permission of the user to determine remotely the health and possible problems with the malfunctioning node. Part of the remote diagnostics may consist of simple connectivity tests with other nodes in the P2PSIP overlay. The simple connectivity tests are not dependent on the type of P2PSIP overlay and they are the topic of this memo. Note however that other tests may be required as well, such as checking the health and performance of the user’s computer or mobile device and also checking the link bandwidth connecting the user to the Internet.
2. Terminology

The concepts used in this document are compatible with "Concepts and Terminology for Peer to Peer SIP" [I-D.ietf-p2psip-concepts] and the P2PSIP base protocol RELOAD[I-D.ietf-p2psip-base].

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 RFC-2119 [1].

3. Overview of Functions

As one diagnostics protocol, P2PSIP diagnostics protocol is mainly used to detect and localize failures in P2PSIP overlay network. It provides mechanisms to detect and localize malfunctioning or badly behaving peers including disabled peers, congested peers and misrouting peers. It provides a mechanism to detect connectivity to the specified peer, a mechanism to detect availabilities of specified resource records and a mechanism to discover P2PSIP overlay topology and the underlay topology failures.

The P2PSIP diagnostics protocol described here extends Ping and Route_Query methods in the P2PSIP base protocol, as well as the Error response to these two methods. Essentially it reuses P2PSIP base protocol specification and then introduces one new method (i.e., Echo method). P2PSIP diagnostics protocol strictly follows the P2PSIP base protocol specification on the messages routing, transporting and NAT traversal etc. The diagnostic methods are however P2PSIP protocol independent.

In this document, we mainly describe how to detect and localize those failures including disabled peers, congested peers, misrouting behaviors and underlying network faults in P2PSIP overlay network through a simple and efficient mechanism. This mechanism is modeled after the ping/traceroute paradigm: ping (ICMP echo request [RFC792]) is used for connectivity checks, and traceroute is used for hop-by-hop fault localization as well as path tracing. This document specifies a "ping" mode (by extending the Ping method defined in Reload) and a "traceroute" mode (implement differently with trusted such as operator deployed P2PSIP overlays, compared with untrusted overlays) for diagnose P2PSIP overlay network.

A Ping request message is forwarded by the intermediate peers along the path and then terminated by the responsible peer, and after local
diagnostics, the responsible peer returns a Ping response message. If an error is found when routing, an Error response is sent to the initiator node by the intermediate peer.

In "Traceroute" mode, we classify the diagnostics into two application scenarios. (1) In trusted p2p overlays, we use an Echo request message, it is received and disposed by each peer along the routing path, and each peer along the path returns an Echo response message with local diagnostics information including the result and causes if existing. (2) In untrusted p2p overlays, we extend the Route_Query method for retrieving diagnostics information iteratively. Firstly, the initiator node asks its neighbor A which is closest to the destination ID, and then retrieve the next hop node B information, along with other diagnostic information of A, to the initiator node. Then the initiator node asks the next hop node B (directly or symmetric routing) to get the further next hop node C information and diagnostic information of B. This step can be iterative until the request reaches responsible node D for the destination ID, and retrieve diagnostic information of node D, or terminates by some failures that prevent the process.

One approach these tools can be used is to detect the connectivity to the specified peer or the availability of the specified resource-record through P2PSIP Ping operation once the overlay network receives some alarms about overlay service degradation or interruption, if the ping fails, one can then send a P2PSIP Traceroute (iterative Route_Query or Echo) to determine where the fault lies.

The diagnostic information must only provide to the authorized peers. Some diagnostic information can be authorized to all the participants in the P2PSIP overlay, and some other diagnostic information can only be provided to the authorization peer list of each diagnostic information according to the local or overlay policy. The authorization mainly depends on the kinds of the diagnostic information and the administrative considerations.

4. Motivation

In the last few years, overlay networks have rapidly evolved and emerged as a promising platform to deploy new applications and services in the Internet. One of the reasons overlay networks are seen as an excellent platform for large scale distributed systems is their resilience in the presence of failures. This resilience has
three aspects: data replication, routing recovery, and static resilience. Routing recovery algorithms are used to repopulate the routing table with live nodes when failures are detected. Static resilience measures the extent to which an overlay can route around failures even before the recovery algorithm repairs the routing table. Both routing recovery and static resilience relies on accurate and timely detection of failures.

As descriptions in "Security requirements in P2PSIP" [I-D.matuszewski-p2psip-security-requirement] and "Security Mechanisms for Peer to Peer SIP"[I-D.jennings-p2psip-security-mechanisms], there are some malfunctioning or badly behaving peers in P2PSIP overlay, those peers may be disabled peers, congested peers or peers behaving with misrouting, and the impact of those peers in the overlay network is degradation of quality of service provided collectively by the peers in the overlay network or interruption of those services. It is desirable to identify malfunctioning or badly behaving peers through some diagnostics tools, and exclude or reject them from the P2PSIP system. Besides those faults, node failures may be caused by underlying failures, for example, when the IP layer routing failover speed after link failures is very slow, then the recovery from the incorrect overlay topology may also be slow. Moreover, if a backbone link fails and the failover is slow, the network may be partitioned, which may lead to partitions of overlay topologies and inconsistent routing results between different partitioned components.

Some keep-alive algorithms based on periodically probe and acknowledge enable accurate and timely detection of failures of one peer's neighbors [Overlay-Failure-Detection], but those algorithms only can detect the disabled neighbors using the periodical method, it may not be enough for operating the overlay network by service providers.

One general P2PSIP overlay diagnostics protocol supporting periodical method and on-demand method for node failures and network failures is desirable. This document describes one general P2PSIP overlay diagnostics protocol useful for P2PSIP base protocol and it is a good complementation for some keep-alive algorithms in the P2P or P2PSIP overlay itself.

5. Packets Formats

This document reuses the P2PSIP base protocol to carry diagnostics information. Considering special usage of diagnostics, this document

extends the P2PSIP base protocol Ping and Route_Query methods, as well as the Error response, and introduces one new type of message method Echo and some diagnostics information.

As described in the P2PSIP base protocol, each message has three parts. This specification is consistent with the format.

```
+-------------------------+   +-------------------------+   +-------------------------+
|    Forwarding Header    |   |    Message Contents     |   |       Signature         |
+-------------------------+   +-------------------------+   +-------------------------+
```

5.1. Message Codes

The mechanism defined in this document follows P2PSIP base protocol specification, the introduced message whatever request or response adopts the same message format with existing P2PSIP base protocol messages. Different types of messages convey different message contents following the forwarding header according to the protocol design. Please refer to P2PSIP base protocol [I-D.ietf-p2psip-base] for the detailed format of forwarding header.

This document extends the following message methods and related message codes by defining additional message payloads.

```
+-------------------+----------------+----------+
| Message Code Name |     Code Value |      RFC |
+-------------------+----------------+----------+
| route_query_req   |             21 | RFC-AAAA |
| route_query_ans   |             22 | RFC-AAAA |
| ping_req          |             23 | RFC-AAAA |
| ping_ans          |             24 | RFC-AAAA |
| error             |         0xffff | RFC-AAAA |
+-------------------+----------------+----------+
```

This document introduces one new type of message as below:
5.2. Message payloads

As P2PSIP base protocol, a P2PSIP diagnostics protocol message content contains one message code following by its payloads. Please refer to P2PSIP base protocol [I-D.ietf-p2psip-base] for the detailed format of Message Contents.

In addition to the newly introduced Echo method, this document extends the Error codes defined in P2PSIP base protocol specification.

5.2.1. Error Codes and Sub-Codes

This document extends the Error response method defined in the P2PSIP base protocol specification to describe the result of diagnostics.

This document introduces new Error Codes as below:

<table>
<thead>
<tr>
<th>Code Value</th>
<th>Error Code Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Underlay Destination Unreachable</td>
</tr>
<tr>
<td>9</td>
<td>Underlay Time exceeded</td>
</tr>
<tr>
<td>10</td>
<td>Message Expired</td>
</tr>
<tr>
<td>11</td>
<td>Upstream Misrouting</td>
</tr>
<tr>
<td>12</td>
<td>Loop detected</td>
</tr>
<tr>
<td>13</td>
<td>TTL hops exceeded</td>
</tr>
</tbody>
</table>

This document introduces Error Sub-Codes in the error_info for Error Code 8 as below:

<table>
<thead>
<tr>
<th>Error Sub-Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>net unreachable</td>
</tr>
<tr>
<td>1</td>
<td>host unreachable</td>
</tr>
<tr>
<td>2</td>
<td>protocol unreachable</td>
</tr>
<tr>
<td>3</td>
<td>port unreachable</td>
</tr>
<tr>
<td>4</td>
<td>fragmentation needed</td>
</tr>
<tr>
<td>5</td>
<td>source route failed</td>
</tr>
</tbody>
</table>
5.2.2. diagnostics information

This document introduces some new diagnostics information conveyed in the message payload, including: the number of hops that the message traverses, the underlay TTL specified, the timestamp of initiating the request message, the timestamp of receiving the request message, and the expiration time of the request message, the processing power, the bandwidth, the number of entries in one’s neighbor table, etc. They are defined as below.

HopCounter (8 bits): This byte only appears in diagnostic responses. It must be exactly copied from the TTL field of the message header in the received Echo request. Then this information is sent back to the request initiator to compute the hops that the message traverses in the overlay.

UnderlayTTL (8 bits): It indicates the underlay TTL which the intermediate peer must adopt when forwarding the diagnostic requests, it is specified by the initiator. If the value is 0, then the intermediate peer must ignore this field, and use the underlay TTL with its local configuration.

TimestampInitiated (64 bits): The time-of-day (in seconds and microseconds, according to the sender’s clock) in NTP format [RFC2030] when the P2PSIP Overlay diagnostic request is sent. It can be carried in the diagnostic response message from the receiver; certainly it first appears in the diagnostic request message;

TimestampReceived (64 bits): it is in a diagnostic response message and the time-of-day (according to the receiver’s clock) in NTP format [RFC2030] that corresponding to the P2PSIP Overlay diagnostic request was received;

Expiration (64 bits): the expiration time of the request message, it is the time-of-day in NTP format [RFC2030]. It can be used to mitigate the replay attack to the destination peer and overlay network.

ProcessPower (32 bits): it appears in a diagnostic response message. The processing power of the node is described by ProcessPower value in unit of MIPS.

Bandwidth (32 bits): It appears in a diagnostic response message. The Bandwidth of the node is described by Bandwidth value in unit of Kbps.
NumEntries (32 bits): It indicates the number of entries in the peer's neighbor table. The administrator of the overlay may be interested in statistics of this value for the consideration such as routing efficiency.

StatusInfo (8 bits): It indicates whether or not the node is in congestion status.

6. Message

All P2PSIP base protocol requests and responses use the common forwarding header after which the message contents follow.

This document extends Ping and Route_Query methods, and introduces the new Echo message to detect and localize failures in P2PSIP overlay network. The Error Codes and their sub-codes to these requests can refer to section 5.2.1 of this spec.

6.1. Ping

In P2PSIP base protocol, Ping is used to test connectivity along a path. However, connectivity quality can not be measured well without some useful information, such as the timestamp and HopCounter. Here we extend the Ping message payloads with a few kinds of useful information for connectivity quality check purposes. See below for the Ping formats.

Ping Request:

struct {
    uint8  UnderlayTTL;
    uint64 TimestampInitiated;
    uint64 Expiration;
} PingReq;

Ping Response:

struct {
    uint8  HopCounter;
    uint64 TimestampReceived;
    uint64 Expiration;
} PingAns;

Any intermediate node which receives the Ping request must adopt the UnderlayTTL for the next hop forwarding. If the value equals to 0, the intermediate peer must ignore this value and determine the underlay TTL using local configuration.
Each peer receiving the Ping request/response should check the Expiration value (NTP format) to determine if the message expires. If the message expires, the intermediate peer should generate a "Message Expired" Error to the initiator node, and discard the message. The responsible node for the request or response must check this value to determine if this message should be ignored.

The responsible peer of the Ping request must copy the TTL field value in the forwarding header to the HopCounter value in the response, meanwhile, it should generate a NTP format timestamp to indicate the received time.

The initiator node, as well as the response peer, can compute the overlay RTT time through the value in TimestampReceived and the TimestampInitiated field.

The initiator node receiving the Ping response should compute the overlay hops to the destination peer for the statistics of connectivity quality from the perspective of overlay hops.

Ping is also used to detect possible failures in the specified path of P2PSIP overlay network. If disabled peers, misrouting behavior and underlying network faults are detected during the routing process, the Error responses with Error codes and descriptions, must be sent to the initiator node immediately. See section 6.4 for the details.

6.2. Route_Query

We simply extend Route_Query to retrieve the diagnostic information from the intermediate peers along the routing path. At each step of the Route_Query request, the responsible peer responds to the initiator node with the status information of itself whether or not congested, its processing power, its available bandwidth, the number of entries in its neighbor table, its uptime, the address information of itself, the next hop peer information.
A Route_Query request must specify which diagnostic information he is interested in by setting different bits in a flag contained in the Route_Query request payload. If the flag is cleared, then the Route_Query request is only used for asking the next hop information, then the iterative mode of Route_Query is degraded only for checking the liveness of the peers along the routing path. The Route_Query request can be routed directly or through the overlay due to the local policy of the initiator node.

**Route_Query request:**
```
struct {
    Boolean    send_update;
    Destination destination;
    uint16     dFlag;
    opaque     overlay_specific_data<0..2^16-1>;
} RouteQueryReq;
```

`send_update`: A single byte. This may be set to "true" to indicate that the requester wishes the responder to initiate an Update request immediately. Otherwise, this value must be set to "false".

`Destination`: The destination which the requester is interested in. This may be any valid destination object, including a Node-ID, compressed ids, or Resource-ID.

`dFlag`: A flag indicating which kind of diagnostic information the initiator interested in. The initiator sets different bits to
retrieve different kinds of diagnostic information. If this flag is cleared, then no diagnostic information is conveyed in the Route_Query response. The kinds of diagnostic information including: status information, its processing power, its available bandwidth, the number of entries in its neighbor table, its uptime, etc. The mapping between the bits in the dFlag and the diagnostic information kind is TBD.

Overlay_specific_data : Other data as appropriate for the overlay.

A response to a successful RouteQueryReq request is a RouteQueryAns message. This is completely overlay specific.

As for the Route_Query responses, whether or not sending back certain kind of diagnostic information to the initiator node depends on (1) the dFlag (2) the authorization policy.

Failures may be detected during the process, after that an Error response should be reported to the initiator node immediately.

6.3. Echo

An Echo request message is used to retrieve the diagnostic information of the specified path in administrative p2p overlays where all the peers in the overlay are trusted. For example, it can be used in a p2p overlay that all peers deployed by the operator to provide services to the customers(clients), where the diagnostics happens between peers in the p2p overlay. For the untrusted p2p overlays. The Echo method must be used with care for the consideration of potential DoS attack.

An Echo request is normal P2PSIP base protocol message; it can be initiated by any node in the administrative p2p overlay which supports P2PSIP base protocol specification.

An Echo request must specify which diagnostic information he is interested in by setting different bits in the dFlag contained in the request payload. If the flag is cleared, the response is a simple Echo response containing no additional diagnostic information.

Any intermediate peer along the Echo request path should forward the Echo request to the next hop, and then returns an Echo response to the initiator node, along with the diagnostic information indicated by the dFlag in the Echo request.
As for the Echo responses, whether or not sending back certain kind of diagnostic information to the initiator node depends on (1) the dFlag (2) the authorization policy.

Any Echo request/response that expires the time in the Expiration field should be ignored.

Echo request:

Struct {
  uint64 Expiration
  uint16 dFlag;
}EchoReq

dFlag (16 bits): A flag indicating which kind of diagnostic information the initiator interested in. The initiator sets different bits to retrieve different kinds of diagnostic information. If this flag is cleared, then no diagnostic information is conveyed in the Echo response. The kinds of diagnostic information including: status information, its processing power, its available bandwidth, the number of entries in its neighbor table, its uptime, etc. The mapping between the bits in the dFlag and the diagnostic information kind is TBD.

Expiration: See section 5.2.2 for the meaning.
Echo response:

Struct {
    uint64 Expiration;
    Opaque overlay_specific_data<0..2^16-1>;
} EchoAns

The peer may find misrouting behaviors or the underlay failures during the Echo process, then an Error response should be generated and send back to the initiator node. See section 6.4 for details.

6.4. Error responses

In p2psip overlay, the error response can be generated by the intermediate peer or responsible peer, to a diagnostic message or other messages. All error responses contain the Error code followed by the subcode and descriptions if existed.

When a request arrives at a peer, if the peer’s responsible ID space does not cover the destination ID of the request, then the peer continues to forward this request according to the overlay specified routing mode.

When a request arrives at a peer, the peer may find some connectivity failures or malfunction peers through the analysis of via list or underlay error messages. The peer should report the error responses to the initiator node. The malfunction node information should also be reported to the initiator node in the error message payload.

The peer should return an Error response with the Error Code 8 "Underlay Destination Unreachable" when it receives an ICMP message with "Destination Unreachable" information after forwarding the received request.

The peer should return an Error response with the Error Code 9 "Underlay Time Exceeded" when it receives an ICMP message with "Time Exceeded" information after forwarding the received request.

The peer should return an Error response with the Error Code 10 "Message Expired" when it finds that the message expires the time field Expiration contained in the message payload.

The peer should return an Error response with Error Code 11 "Upstream Misrouting" when it finds its upstream peer disobeys the routing
rules defined in the overlay. The immediate upstream peer information should also be conveyed to the initiator node.

The peer should return an Error response with Error Code 12 "Loop detected" when it finds a loop through the analysis of via list.

The peer should return an Error response with Error Code 13 "TTL hops exceeded" when it finds that the TTL field value is no more than 0 when forwarding.

7. Security Considerations

The Echo method may cause DoS attack to the initiator, though this implementation is more efficient than using iterative mode of Route_Query operation.

An advice is to use the efficient Echo operation in administrated P2PSIP overlay and use the pacing-style Route_Query operation in the untrustworthy P2PSIP overlay network, certainly, the probability of this type of DoS attack is very low because the overlay is distributed and then it is very hard for the attacker to know the accurate Peer-IDs and attack most of all peers simultaneously.

8. IANA Considerations

Message Code: this document introduces a new type of message as below:

```
+-------------------+----------------+----------+
| Message Code Name |     Code Value |      RFC |
+-------------------+----------------+----------+
| Echo_req          |             29 | RFC-AAAA |
| Echo_ans          |             30 | RFC-AAAA |
+-------------------+----------------+----------+
```

Error Code: this document introduces some new Error Codes as below:

```
Code Value   Error Code Name
8            Underlay Destination Unreachable
9            Underlay Time exceeded
10           Message Expired
11           Upstream Misrouting
12           Loop detected
13           TTL hops exceeded
```

Error Sub-Code: this document defines response sub-codes for the Error Code 8 "Underlay Destination Unreachable" as below:
<table>
<thead>
<tr>
<th>Sub-Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>net unreachable</td>
</tr>
<tr>
<td>1</td>
<td>host unreachable</td>
</tr>
<tr>
<td>2</td>
<td>protocol unreachable</td>
</tr>
<tr>
<td>3</td>
<td>port unreachable</td>
</tr>
<tr>
<td>4</td>
<td>fragmentation needed</td>
</tr>
<tr>
<td>5</td>
<td>source route failed</td>
</tr>
</tbody>
</table>

9. Acknowledgments

We would like to thank Zheng Hewen for the contribution of the initial version of this draft, we would also like to thank Bruce Lowekamp, Salman Baset, Henning Schulzrinne and Jiang Haifeng for the email discussion and their valued comments. We would also like to thank Henry Sinnreich for contributing to the usage scenarios in the Introduction.

10. References

10.1. Normative References


10.2. Informative References

[I-D.ietf-behave-rfc3489bis] Rosenberg, J., Huitema, C., Mahy, R.,
and D. Wing, "Simple Traversal Underneath Network Address
Translator (NAT) (STUN)", draft-ietf-behave-rfc3489bis-08
(work in progress), July 2007.

[I-D.ietf-behave-turn] Rosenberg, J., Mahy, R., and C. Huitema,
"Obtaining Relay Addresses from Simple Traversal Underneath
NAT (STUN)", draft-ietf-behave-turn-04 (work in progress),

[I-D.ietf-mmusic-ice] Rosenberg, J., "Interactive Connectivity
Establishment (ICE): A Methodology for Network Address
Translator (NAT) Traversal for Offer/Answer Protocols",
draft-ietf-mmusic-ice-17 (work in progress), July 2007

Registration and Resource Location", draft-bryan-p2psip-dsip-00
(work in progress), February 2007.

draft-baset-p2psip-p2pp-00 (work in progress), July 2007.

[I-D.Jennings-p2psip-asp] C. Jennings, "Address Settlement by Peer to
Peer", draft-jennings-p2psip-asp-00 (work in progress),

[I-D.marocco-p2psip-xpp-pcan] Marocco, E. and E. Ivov, "XPP
Extensions for Implementing a Passive P2PSIP Overlay
Network based on the CAN Distributed Hash Table",
draft-marocco-p2psip-xpp-pcan-00 (work in progress), June 2007.

Author’s Addresses

Haibin Song
Huawei
10F, Huihong Mansion, No. 91 Baixia Rd., Nanjing, China
Phone: +86-25-84565867
Email: melodysong@huawei.com

Xingfeng Jiang
Huawei
10F, Huihong Mansion, No. 91 Baixia Rd., Nanjing, China
Phone: +86-25-84565868
Email: jiang.x.f@huawei.com

Full Copyright Statement

Copyright (C) The IETF Trust (0000).

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, THE IETF TRUST 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 Statement

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.