Internet Draft
draft-shore-tist-prot-00.txt
May 2002
Expires November 2002

Melinda Shore
Cisco Systems

The TIST (Topology-Insensitive Service Traversal) Protocol <draft-shore-tist-prot-00.txt>

Status of this Memo

Abstract

TIST is an application of the RSVP protocol to the problem of communicating application requests, such as pinhole openings and NAT table entries, to NATs and firewalls. By using RSVP we avoid the problem of having to locate these devices in the network and establish trusted relationships with each one we would like to influence.

1. Introduction

Network transparency was one of the original design goals of the IP protocol suite. Over time, as the economic and social drivers

behind network have changed, services are appearing within the network that undermine transparency. Examples of these services include security services (such as security gateways and firewalls), address translation services, data compression, and QoS. Applications are increasingly needing to influence the behavior of these services in order improve their function on the network and sometimes, in dire cases, to be able to function at all.

Several different approaches to solving this problem have been proposed in different contexts. The IETF Middlebox Communication (midcom) working group [Srisuresh] is developing a protocol to run between an application endpoint or proxy and a "middlebox," such as a firewall or NAT, in the network. Cisco's Tunnel Endpoint Discovery Protocol (TED) [Fluhrer] is based on an IKE extension using the Vendor ID to find IPSec peers and negotiate proxies.

Aspects of the problem include:

+ Location

+ Provisioning

+ Dynamic state installation and maintenance

+ Routing

Policy-related aspects of the problem include authorization and admission control. Policy can be statically provisioned or it can be determined through consultation with a policy server (also known as a "policy decision point"). While the protocol between a middlebox and a policy server is outside the scope of this document, we need to ensure that sufficient information is carried in the TIST protocol for a middlebox to be able to consult with a policy server.

2. TIST

TIST (Topology-Insensitive Service Traversal) is based on a framework first outlined in [Shore]. Like TED, TIST relies on the network to deal with network-layer issues, including routing and discovery. Where midcom assumes that a layer-violating explicit connection will be established between each application endpoint needing to influence a middlebox and each middlebox, TIST simply sends a request from a source host to a destination host with an "attention" flag set (whether protocol number, port number, or IP option). Applications no longer need knowledge of network

topology, and solutions to the location and routing problems are inherent in this approach. We refer to it as "topology-insensitive" because it expected to be robust across complex, varied network topologies and in the face of changing network topologies.

As evaluation of requirements for a network-transparent protocol progressed it became clear that RSVP [RFC2205] was a very close match and that it constains the necessary protocol machinery to install and maintain middlebox state. RSVP relies on network-layer routing to find RSVP-capable routers. It has the considerable advantage of being widely available on routers and reasonably widely available on host operating systems. RSVP has the following attractive attributes [RFC2205]:

+ It recovers from routing changes

+ RSVP operates on unidirectional data flows

+ RSVP operates independently of network routing protocols

+ RSVP allows the transport of data elements that are opaque to the protocol

+ RSVP provides transparent operation through routers that don't support it

+ RSVP supports both IPv4 and IPv6

+ RSVP messages can be processed both by endpoints (host filters, for example) and by intermediate devices

This document assumes familiarity with the RSVP protocol and with its message processing rules [RFC2209].

3. Terminology

Downstream
: Away from the sender (see below) towards the receiver

Middlebox
: A network intermediate devices that implements one or more of the middlebox services, such as NAT or firewall. See [Srisuresh].

Receiver
: The entity towards which a Path message is ultimately directed, and which generates a Resv message.

Resource
: The state being manipulated in TIST requests, such as a NAT table mapping or a firewall pinhole.

Sender
The entity generating the original TIST Path message.

Upstream
Towards the sender and away from the receiver.

4. Protocol

4.1. Requesting Resources

TIST is based on the RSVP protocol, in particular its flow model and the use of the two main messages, Path and Resv.

Each data flow arrives from a previous hop node through a corresponding incoming interface and after processing departs through one or more outgoing interfaces. For the purposes of firewall configuration, this represents a significant advantage in that it allows a firewall or NAT to use the IP-layer routing information it already has in order to correctly identify the interface to which the TIST request should be applied. Unlike midcom, information on interfaces and routing table contents need not be exported to clients or agents.

A host requiring firewall or NAT services (say, a pinhole opened or a NAT table mapping installed) sends a Path message downstream towards the host with which it intends to communicate. Path messages include a service request and parameters and the unicast address of the previous node. It also includes a Sender Template [RFC2205]. For the purposes of this application, we will use only a Fixed Filter reservation style, and we will NOT include a Sender Tspec.

When the downstream host receives the Path message, it returns a reservation request (Resv) upstream towards the original sender. This message MUST exactly follow the reverse of the path it took downstream. It is upon receipt of the Resv message that the middlebox confirms and finally installs new state.

4.2. State Maintenance

As with RSVP, TIST uses a soft state approach to managing state in firewalls and NATs. State is created and refreshed through the use of idempotent Path and Resv messages, which 1) provides automatic cleanup of stale state, and 2) provides robustness across topology and routing changes. See [RFC2205] for state maintenance details.

4.3. Path Teardown

In support of more efficient resource management TIST senders SHOULD explicitly tear down the resources they have requested, using the RSVP PathTear message. TIST receivers SHOULD NOT use the ResvTear message to delete installed resources (and cannot, when request authentication is being used).

If a middlebox wishes to delete an installed resource it SHOULD send a ResvErr back to the TIST sender, which may then send a PathTear message to remove the resource. [There's an issue here, and that's whether or not a middlebox wishes to conceal its existence.]

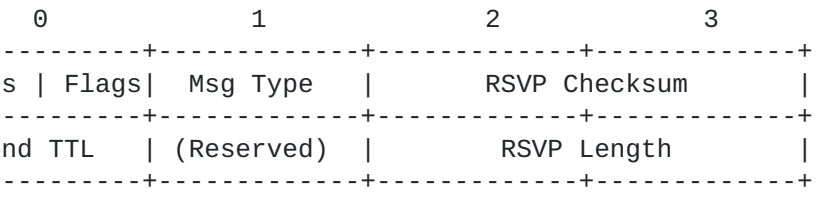
5. Traversing TIST-unaware Middleboxes

When RSVP messages traverse non-RSVP-capable routers, traffic may be perturbed but the RSVP protocol will itself function properly. Traffic that has been admitted for privileged service may be perturbed as it traverses a non-RSVP-capable router but it will likely receive best-effort service. When TIST messages traverse non-TIST-capable middleboxes the potential impact is quite different.

If a TIST message is passed along without being processed by a non-TIST-capable firewall, it may appear to the originator of the Path message that the TIST request was successful while, in fact, a firewall along the path did not process the request at all and therefore did not open an appropriate pinhole. Consequently care must be taken to craft the protocol to maximize the likelihood that TIST requests will be blocked by non-TIST-capable firewalls. Potential approaches might include setting IP options bits, such as the Router Alert option [RFC2113], and/or carrying the messages in raw IP packets rather than over a standard transport protocol such as TCP or UDP. See "Transport Considerations."

6. TIST Messages

TIST uses existing RSVP messages. It introduces some new classes and modifies the use of some existing classes. Because TIST is based on RSVP, it shares the RSVP common header format:



where:
Vers: 4 bits
Protocol version number. Currently 1

Flags: 4 bits
Not used

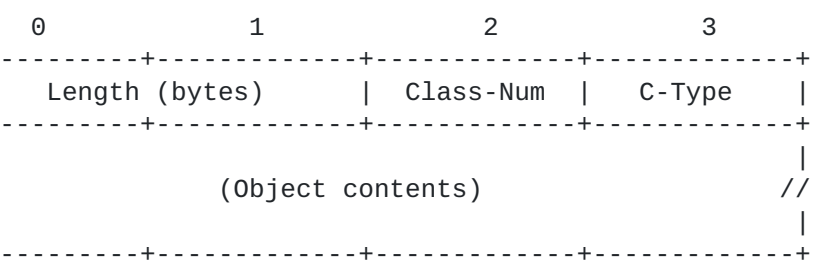
Msg Type: 8 bits
1 = Path
2 = Resv
3 = PathErr
4 = ResvErr
5 = PathTear
6 = ResvTear
7 = ResvConf

RSVP Checksum: 16 bits
See [RFC2205]

Send_TTL: 8 bits
The original IP TTL with which the message was sent.

RSVP Length: 16 bits
Total RSVP message length, including headers.

Every RSVP object consists of one or more 32-bit words with a one-word header, with the following format:



Length: 16 bits

Total object length in bytes. Must be a multiple of 4 and at least 4.

Class-num

Identifies the object class. See [RFC2205] for details. The RSVP messages which MUST be supported by a TIST implementation include:

NULL

SESSION

RSVP_HOP

TIME_VALUES

SENDER_TEMPLATE

ERROR_SPEC

POLICY_ DATA

INTEGRITY

6.1. TIST Path Messages

While TIST Path messages are sent by a sender in order to initiate request processing. Each sender periodically sends a Path message downstream towards a receiver. The Path message contains NAT and/or firewall requests, along with associated data such as policy objects or integrity objects.

A Path message travels from a sender to a receiver. The IP source address in the packet header MUST be the address of the sender, and the destination address MUST be the address of the receiver. This allows TIST to leverage existing routing state for the location of and communication with middleboxes along the data path. Note, however, that by including addresses in TIST objects we allow the possibility of 3rd-party requests, where appropriate. Therefore TIST-capable middleboxes MUST act on the addresses in the TIST objects and not on the addresses in the IP headers of the TIST requests.

The TIST Path message format is:

<Path Message> ::= <Common Header> [<INTEGRITY>]
<SESSION> <RSVP HOP>

```
                        <TIME_VALUES>

                        [ <POLICY_DATA> ... ]
                        [ <SENDER_TEMPLATE> ]
                        [ <FW> ... ]
                        [ <NAT> ... ]
```

All RSVP objects in the Path message MUST obey the behavior defined in [RFC2205]. When a NAT table mapping is being requested, after the mapping is installed the node MUST write the address/port tuple of the new mapping into the Previous Hop IP Address field in the NAT object.

6.2. TIST Resv Messages

TIST Resv messages are returned by the receiver to the sender in response to a Path message. These are the messages that actually confirm and install soft state, and they are returned along the reverse paths of data flows for the session. As with RSVP, the destination address of a Resv message is the unicast address of the previous-hop node. The source address is that of the node sending the message (not the TIST receiver).

The Resv format is:

```
<Resv Message> ::= <Common Header> [ <INTEGRITY> ]
                        <SESSION>  <RSVP_HOP>
                        <TIME_VALUES>
                        [ <RESV_CONFIRM> ]
                        [ <FW> ... ]
                        [ <NAT> ... ]
                        [ <POLICY_DATA> ... ]
```

All objects MUST obey the same behavior as they do in RSVP. The NAT object carries the address(es) which are being NATted-to by NATs in the data path. In addition, each TIST-capable NAT that chooses to honor the NAT request MAY add the address and port to which it will be mapping the requested address and port, as a NAT object. The NAT objects are ordered, so that the first NAT added will be first, the second will follow, and so on. Each NATting node adding a NAT object MUST replace the contents of Previous Hop IP Address field with the current contents of IP Address field; this allows the selection of the correct IP address and port number for installation in a firewall pinhole rule.

Note that a NAT may, according to site-local policy, choose not to add a NAT object. The primary reason for choosing not to do so would be to obscure the presence of a particular NAT, but given that this would likely cause protocol failures and that there are other options, such as STUN [Rosenberg], for finding NATted-to addresses, this is not recommended.

Because the IP transport address/port pair may be modified by NATs along the path and because there may be more than one NAT, firewalls MUST install pinhole informormation based on the most recently-assigned address - that is to say, the last NAT object.

[The ability of receivers to send back the address they see (i.e. that of the outermost NAT) is an open question and needs further consideration.]

6.3. TIST PathTear Messages

As with RSVP, the recipient of a PathTear message MUST delete matching path state, whether it is a firewall pinhole, a NAT table entry, or both. State is matched on the SESSION, SENDER_TEMPLATE, and PHOP objects, and MAY match on a FW or NAT object as appropriate. Unlike RSVP, however, non-matching PathTear messages SHOULD be forwarded. The TIST PathTear message format is:

```
<PathTear Message> ::= <Common Header> [ <INTEGRITY> ]
                            <SESSION> <RSVP_HOP>
                            <SENDER_TEMPLATE>
                            [ <FW> ] [ <NAT> ]
```

6.4. TIST ResvTear Messages

Receipt of a ResvTear message deletes matching reservation state. State is matched on the SESSION and RSVP_HOP objects. ResvTear messages are initiated by receivers (for example, upon call termination in a VoIP application).

If a middlebox wishes to delete an installed resource it SHOULD send a ResvErr back to the TIST sender, which may then send a PathTear message to remove the resource.

A ResvTear message must be routed like the corresponding Resv message, and its IP destination address MUST be the unicast adress of a previous hop.

The TIST ResvTear message format is:

```
<ResvTear Message> ::= <Common Header> [<INTEGRITY>]
                           <SESSION> <RSVP_HOP>
```

6.5. TIST PathErr Messages

PathErr messages are error messages in response to errors in processing Path messages. They are returned to the sender and they are routed hop-by-hop using path state. At each hop, the IP destination address is the unicast address of the previous hop. PathErr messages are not processed by intermediate nodes, but only by the sender application.

The format for PathErr messages is:

```
<PathErr message> ::= <Common Header> [ <INTEGRITY> ]
                           <SESSION> <ERROR_SPEC>

                           [ <POLICY_DATA> ...]
```

The ERROR_SPEC object specifies the error being reported and includes the IP address of the node that encountered the error. The POLICY_ DATA object(s) may carry information about policy errors.

6.6. TIST ResvErr Messages

ResvErr messages report Resv processing errors or disruption in reservation (pinhole/NAT table mapping) state. ResvErr messages are returned towards the receiver that initiated the Resv message and are routed hop-by-hop using the reservation state. At each hop the IP destination address is the unicast address of the next-hop node.

The format for a ResvErr message is:

```
<ResvErr Message> ::= <Common Header> [ <INTEGRITY> ]
                           <SESSION>  <RSVP_HOP>
                           <ERROR_SPEC>  [ <SCOPE> ]
                           [ <POLICY_DATA> ...]
```

The ERROR_SPEC object specifies the error being reported and includes the IP address of the node that encountered the error. The POLICY_DATA object(s) may carry information about policy errors.

7. TIST Classes

TIST adds these new classes:

NAT

Carries NAT-specific requests.

FW

Carries firewall-specific requests.

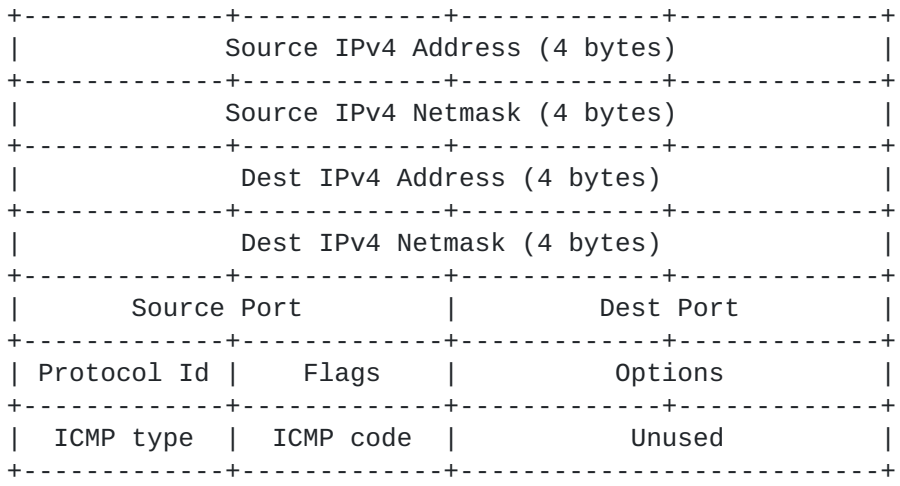
7.1. Object Definitions

7.1.1. FW Class

FW Class = <to be assigned by IANA>.

7.1.1.1. IPv4 FIREWALL object: Class = <xxx>, C-Type = 1

[This bit needs major tweaking]



The FW object describes a filter rule to be installed. It is derived from the Diameter IPFilterRule AVP [Diameter].

The Netmask fields are provided to allow arbitrary-length network wildcarding, such as CIDR addresses or locally-visible subnets, to be filtered upon.

If the port is set to zero, it is treated as a wildcard matching any port number.

The Protocol ID is any IP protocol number.

Options

```
FRAG                . . . . . . . . . . . . . . . x
SSRR                . . . . . . . . . . . . . . x . 
LSRR                . . . . . . . . . . . . . x . . 
RR                  . . . . . . . . . . . . x . . . 
TCPESTABLISHED      . . . . . . . . . . . x . . . . 
TCPSETUP            . . . . . . . . . . x . . . . . 
ICMP                . . . . . . . . . x . . . . . . 
```

When the FRAG bit is set, match if the packet is a fragment but not the first fragment of a datagram.

When the SSRR bit is set, match if the strict source route IP option is set. The LSRR bit is used to indicate the loose source route IP option, and RR is used to indicate the record route IP option.

TCPESTABLISHED is used to match TCP packets that have the RST or ACK bits set. It is meaningless for non-TCP packets and the middlebox SHOULD return a XXX error if it receives a request in which this bit is set but the protocol is not TCP.

TCPSETUP is used to match TCP packets. It is meaningless for non-TCP packets and the middlebox SHOULD return a XXX error if it receives a request in which this bit is set but the protocol is not TCP.

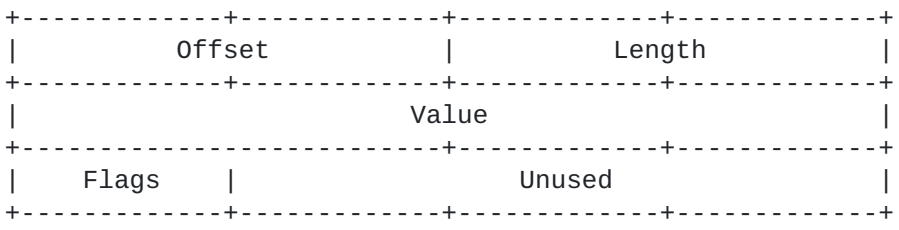
The ICMP type and ICMP code fields are ignored unless the ICMP bit is set in the options field.

Flags

```
ALLOW               . . . . . . . x
```

These flags affect how the rules should be processed by the middlebox. If the ALLOW bit is set, the request is that matching traffic should be allowed by the firewall. If it is not set, the request is that matching traffic should be denied.

7.1.1.2. IPv4 FIREWALL Offset object: Class = <xxx>, C-Type = 2



This object allows senders to request filtering on arbitrary values at arbitrary offsets from the start of the IP packet. For example, this could be used to carry an IPSec SPI, which would allow filtering requests on encapsulated packets.

Offset is a 16-bit value representing the number of octets from the start of the IP packet at which to begin the comparison. Length a 16-bit value representing number of length in octets of the value to be compared against, and Value is the Value itself.

Flags

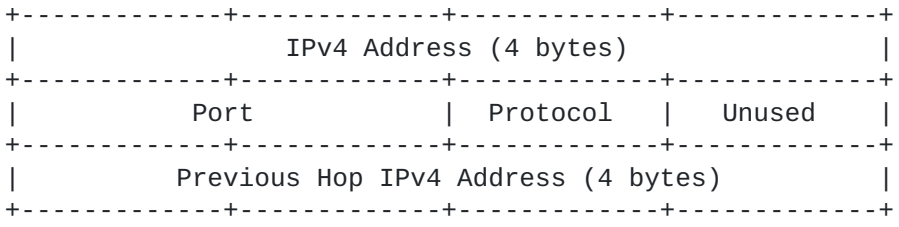
ALLOW x

These flags affect how the rules should be processed by the middlebox. If the ALLOW bit is set, the request is that matching traffic should be allowed by the firewall. If it is not set, the request is that matching traffic should be denied.

7.1.2. NAT Class

NAT Class = <to be assigned by IANA>

7.1.2.1. IPv4 NAT object: Class = <xxx>, C-Type = 1



The NAT object describes an address/port/protocol tuple for which a NAT table mapping is requested. The Previous Hop IPv4 address contains the most recent address assignment, so that, for example, if there has been no NAT in the path to this point it contains the address of the sending host, and if there has been a NAT in the path it contains the address/port tuple to which the sending host's address has most recently been mapped.

8. Error Codes and Values

TIST uses existing RSVP error codes. Usage details follow.

8.1. Error Code = 01: Admission control failure

Error code 01 would be sent in the case where there were insufficient resources on the middlebox, for example if the NAT table were full. The ss, u, and r bits are to be treated as specified in [RFC 2205].

8.2. Error Code = 05: Conflicting reservation style

This error code is not used.

8.3. Error Code = 06: Unknown reservation style

This error code is not used.

8.4. Error Code = 12: Service preempted

This error code is not used.

8.5. Error Code = 21: Traffic Control Error

This error code is not used. Malformed or unreasonable requests would return error code 23.

8.6. Error Code = 22: Traffic Control System error

This error code is not used.

9. Transport Issues

The primary issue related to TIST transport is that we need to maximize the likelihood that TIST requests will be dropped by any non-TIST-capable firewall/NAT that they traverse. Note that this contrasts immediately with RSVP, where it is desirable that RSVP

requests be forwarded by non-RSVP-aware devices.

There are several options for doing this, the most promising of which are 1) setting an IP option, and 2) using a "surprising" transport-layer protocol (i.e. not TCP or UDP). The RSVP protocol requires that RSVP messages be sent with the Router Alert IP option [RFC2113] and RSVP messages be sent in "raw" IP packets (and only optionally be UDP-encapsulated). The risk is that a firewall may be configured to pass RSVP traffic and that a TIST packet may, therefore, be forwarded without having been processed. Because of this it may be desirable to assign a separate IP protocol number.

10. IANA Considerations

The following classes need RSVP class numbers assigned:

NAT

FW

[We may wish to have a new protocol number assigned.]

11. Security Considerations

Security is critically important to TIST because TIST is used to communicate with security devices such as firewalls. The threats must be well-understood and appropriate countermeasures taken. In particular, we need to protect against inappropriate manipulation of state both at the sender and receiver as well as at the intermediate nodes.

The use of the RSVP INTEGRITY object [RFC2747], in particular, will provide hop-by-hop message authentication and anti-replay protection, and it SHOULD be applied to all TIST messages. Unfortunately, while the specification provides a keying example based on Kerberos, keying is unspecified and will need to be addressed in future versions of this document.

Authorization (admissions) decisions must be made on the basis of the authenticated identify of the requesting user or application (the sender). TIST implementations MUST include the authentication policy element (AUTH_DATA) [RFC3182]. In order to protect against application-layer data hijacking attacks, all NAT responses from a NAT back to the sender SHOULD include AUTH_DATA elements. Note that this implies that the NAT device either participates in a PKI or in a Kerberos deployment.

12. Proxy considerations

In some cases it may be desirable to have proxies generate Path messages, or receive Path messages and/or generate Resv messages, on behalf of topologically-appropriate hosts. Whether or not this is possible will depend on the addressing and security models. A primary consideration will be location and addressing - a proxy providing an AUTH_DATA object in a Resv message may lead to that message being inappropriately discarded if the sender expects the Resv message to come from the host to which the Path message was originally addressed.

13. Multicast considerations

Multicast will be addressed in a future iteration of this document.

14. References

[Diameter] Calhoun, P. et al. "Diameter Base Protocol," work in progress. April 2002.

[Fluhrer] Fluhrer, S. "Tunnel Endpoint Discovery," work in progress. November 2001.

[RFC2205] Braden, R. et al. "Resource ReSerVation Protocol (RSVP -- Version 1 Functional Specification." RFC 2205.

[RFC2209] Braden, R. and Zhang, L. "Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules." RFC 2209.

[RFC2113] Katz, D. "IP Router Alert Option," RFC 2113.

[RFC2747] Baker, F., Lindell, B., and Talwar, M. "RSVP Cryptographic Authentication." RFC 2747.

[Rosenberg] Rosenberg, J. et al. "STUN - Simple Traversal of UDP Through NATs," work in progress. April 2002.

[Shore] Shore, M. "Towards a Network-Friendlier Midcom," work in progress. October 2001.

[Srisuresh] Srisuresh, S. et al. "Middlebox Communication Architecture and framework," work in progress. October 2001.

15. Acknowledgements

I would like to thank Adina Simu and Mahadev Somasunderam for their analysis and discussion.

16. Copyright

The following copyright notice is copied from RFC 2026 [RFC2026] Section 10.4, and describes the applicable copyright for this document.

17. Intellectual Property

The following notice is copied from RFC 2026 [Bradner, 1996], Section 10.4, and describes the position of the IETF concerning intellectual property claims made against this document.

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to

pertain to the implementation or use other technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication 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 Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Author's Address

Melinda Shore
Cisco Systems
809 Hayts Road
Ithaca, NY 14850
USA
mshore@cisco.com