TRAM
Internet-Draft
Intended status: Standards Track
Expires: September 4, 2018
M. Petit-Huguenin
Impedance Mismatch
G. Salgueiro
Cisco
March 3, 2018

Path MTU Discovery Using Session Traversal Utilities for NAT (STUN)
draft-ietf-tram-stun-pmtud-07

Abstract

This document describes a Session Traversal Utilities for NAT (STUN) Usage for Path MTU Discovery (PMTUD) between a client and a server.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Packetization Layer Path MTU Discovery (PMTUD) specification [RFC4821] describes a method to discover the Path MTU but does not describe a practical protocol to do so with UDP.

Not all UDP-based protocols implement the Path MTU discovery mechanism described in [RFC4821]. These protocols can make use of the probing mechanisms described in this document instead of designing their own adhoc extension. These probing mechanisms are implemented with Session Traversal Utilities for NAT (STUN), but their usage is not limited to STUN-based protocols.

The STUN usage defined in this document for Path MTU Discovery (PMTUD) between a client and a server permits proper operations of UDP-based applications in the network. It also simplifies troubleshooting and has multiple other applications across a wide variety of technologies.

Complementary techniques can be used to discover additional network characteristics, such as the network path (using the STUN Traceroute mechanism described in [I-D.martinsen-tram-stuntrace]) and bandwidth availability (using the mechanism described in [I-D.martinsen-tram-turnbandwidthprobe]).

2. Overview of Operations

This section is meant to be informative only. It is not intended as a replacement for [RFC4821].

A UDP endpoint that uses this specification to discover the Path MTU over UDP and knows that the endpoint it is communicating with also supports this specification can choose to use either the Simple Probing mechanism (as described in Section 4.1) or the Complete Probing mechanism (as described in Section 4.2). The selection of

which Probing Mechanism to use is dependent on performance and security and complexity trade-offs.

If the Simple Probing mechanism is chosen, then the Client initiates Probe transactions, as shown in Figure 1, which increase in size until transactions timeout, indicating that the Path MTU has been exceeded. It then uses that information to update the Path MTU.

```
                       Client           Server
                         |                  |
                         | Probe Request    |
                         |----------------->|
                         |                  |
                         |  Probe Response  |
                         |<-----------------|
                         |                  |
```

Figure 1: Simple Probing Example

If the Complete Probing mechanism (as described in Section 4.2) is chosen, then the Client sends Probe Indications of various sizes interleaved with UDP packets sent by the UDP protocol. The Client then sends a Report Request for the ordered list of identifiers for the UDP packets and Probe Indications received by the Server. The Client then compares the list returned in the Report Response with its own list of identifiers for the UDP packets and Probe Indications it sent. The Client then uses that comparison to find which Probe Indications were dropped by the network as a result of their size. It then uses that information to update the Path MTU.

```
                        Client              Server
                           | UDP Packet         |
                           |------------------->|
                           |                    |
                           | UDP Packet         |
                           |------------------->|
                           |                    |
                           | Probe Indication   |
                           |------------------->|
                           |                    |
                           | UDP Packet         |
                           |------------------->|
                           |                    |
                           | Probe Indication   |
                           |------------------->|
                           |                    |
                           | Report Request     |
                           |------------------->|
                           |   Report Response  |
                           |<-------------------|
                           |                    |
```

Figure 2: Complete Probing Example

3. Terminology

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 [RFC2119]. When these words are not in ALL CAPS (such as "must" or "Must"), they have their usual English meanings, and are not to be interpreted as RFC 2119 key words.

4. Probing Mechanisms

The Probing mechanism is used to discover the Path MTU in one direction only, from the client to the server.

Two Probing mechanisms are described, a Simple Probing mechanism and a more complete mechanism that can converge quicker and find an appropriate PMTU in the presence of congestion. Additionally, the Simple Probing mechanism does not require authentication, whereas the complete mechanism does.

Implementations supporting this specification MUST implement the server side of both the Simple Probing mechanism (Section 4.1) and the Complete Probing mechanism (Section 4.2).

Implementations supporting this specification MUST implement the client side of the Complete Probing mechanism. They MAY implement the client side of the Simple Probing mechanism.

4.1. Simple Probing Mechanism

The Simple Probing mechanism is implemented by sending a Probe Request with a PADDING [RFC5780] attribute over UDP with the DF bit set in the IP header. A router on the path to the server can reject this request with an ICMP message or drop it.

4.1.1. Sending a Probe Request

A client forms a Probe Request by using the Probe Method and following the rules in Section 7.1 of [RFC5389].

The Probe transaction MUST be authenticated if the Simple Probing mechanism is used in conjunction with the Implicit Probing Support mechanism described in Section 5.2. If not, the Probe transaction MAY be authenticated.

The client adds a PADDING [RFC5780] attribute with a length that, when added to the IP and UDP headers and the other STUN components, is equal to the Selected Probe Size, as defined in [RFC4821] Section 7.3. The client MUST add the FINGERPRINT attribute so the STUN messages are disambiguated from the other protocol packets.

Then the client sends the Probe Request to the server over UDP with the DF bit set. For the purpose of this transaction, the Rc parameter specified in Section 7.2.1 of [RFC5389] is set to 3. The initial value for RTO stays at 500 ms.

A client MUST NOT send a probe if it does not have knowledge that the server supports this specification. This is done either by external signalling or by a mechanism specific to the UDP protocol to which PMTUD capabilities are added or by one of the mechanisms specified in Section 5.

4.1.2. Receiving a Probe Request

A server receiving a Probe Request MUST process it as specified in [RFC5389].

The server then creates a Probe Response. The server MUST add the FINGERPRINT attribute so the STUN messages are disambiguated from the other protocol packets. The server then sends the response to the client.

4.1.3. Receiving a Probe Response

A client receiving a Probe Response MUST process it as specified in [RFC5389]. If a response is received this is interpreted as a Probe Success, as defined in [RFC4821] Section 7.6.1. If an ICMP packet "Fragmentation needed" is received then this is interpreted as a Probe Failure, as defined in [RFC4821] Section 7.6.2. If the Probe transaction times out, then this is interpreted as a Probe Inconclusive, as defined in [RFC4821] Section 7.6.4.

4.2. Complete Probing Mechanism

The Complete Probing mechanism is implemented by sending one or more Probe Indications with a PADDING attribute over UDP with the DF bit set in the IP header followed by a Report Request to the same server. A router on the path to the server can reject this Indication with an ICMP message or drop it. The server keeps a chronologically ordered list of identifiers for all packets received (including retransmitted packets) and sends this list back to the client in the Report Response. The client analyzes this list to find which packets were not received. Because UDP packets do not contain an identifier, the Complete Probing mechanism needs a way to identify each packet received.

Some application layer protocols may already have a way of identifying each individual UDP packet, in which case these identifiers SHOULD be used in the IDENTIFIERS attribute of the Report Response. While there are other possible packet identification schemes, this document describes two different ways to identify a specific packet.

In the first packet identification mechanism, the server computes a checksum over each packet received and sends back to the sender the list of checksums ordered chronologically. The client compares this list to its own list of checksums.

In the second packet identification mechanism, the client prepends the UDP data with a header that provides a sequence number. The server sends back the chronologically ordered list of sequence numbers received that the client then compares with its own list.

4.2.1. Sending the Probe Indications and Report Request

A client forms a Probe Indication by using the Probe Method and following the rules in [RFC5389] Section 7.1. The client adds to the Probe Indication a PADDING attribute with a size that, when added to the IP and UDP headers and the other STUN components, is equal to the Selected Probe Size, as defined in [RFC4821] Section 7.3. If the

authentication mechanism permits it, then the Indication MUST be authenticated. The client MUST add the FINGERPRINT attribute so the STUN messages are disambiguated from the other protocol packets.

Then the client sends the Probe Indication to the server over UDP with the DF bit set.

Then the client forms a Report Request by following the rules in [RFC5389] Section 7.1. The Report transaction MUST be authenticated to prevent amplification attacks. The client MUST add the FINGERPRINT attribute so the STUN messages are disambiguated from the other protocol packets.

Then the client waits half the RTO, if it is known, or 250 ms after sending the last Probe Indication and then sends the Report Request to the server over UDP.

4.2.2. Receiving an ICMP Packet

If an ICMP packet "Fragmentation needed" is received then this is interpreted as a Probe Failure, as defined in [RFC4821] Section 7.5.

4.2.3. Receiving a Probe Indication and Report Request

A server supporting this specification will keep the identifiers of all packets received in a chronologically ordered list. The packets that are to be associated to a list are selected according to Section 5.2 of [RFC4821]. The same identifier can appear multiple times in the list because of retransmissions. The maximum size of this list is calculated such that when the list is added to the Report Response, the total size of the packet does not exceed the unknown Path MTU, as defined in [RFC5389] Section 7.1. Older identifiers are removed when new identifiers are added to a list that is already full.

A server receiving a Report Request MUST process it as specified in [RFC5389].

The server creates a Report Response and adds an IDENTIFIERS attribute that contains the chronologically ordered list of all identifiers received so far. The server MUST add the FINGERPRINT attribute. The server then sends the response to the client.

The exact content of the IDENTIFIERS attribute depends on what type of identifiers have been chosen for the protocol. Each protocol adding PMTUD capabilities as specified by this specification MUST describe the format of the contents of the IDENTIFIERS attribute, unless it is using one of the formats described in this

specification. See Section 6.1 for details about the IDENTIFIERS attribute.

4.2.4. Receiving a Report Response

A client receiving a Report Response processes it as specified in [RFC5389]. If the response IDENTIFIERS attribute contains the identifier of the Probe Indication, then this is interpreted as a Probe Success for this probe, as defined in [RFC4821] Section 7.5. If the Probe Indication identifier cannot be found in the Report Response, this is interpreted as a Probe Failure, as defined in [RFC4821] Section 7.5. If the Probe Indication identifier cannot be found in the Report Response but identifiers for other packets sent before or after the Probe Indication can all be found, this is interpreted as a Probe Failure as defined in [RFC4821] Section 7.5. If the Report Transaction times out, this is interpreted as a Full-Stop Timeout, as defined in [RFC4821] Section 3.

4.2.5. Using Checksums as Packet Identifiers

When using a checksum as a packet identifier, the client calculates the checksum for each packet sent over UDP that is not a STUN Probe Indication or Request and keeps this checksum in a chronologically ordered list. The client also keeps the checksum of the STUN Probe Indication or Request sent in that same chronologically ordered list. The algorithm used to calculate the checksum is similar to the algorithm used for the FINGERPRINT attribute (i.e., the CRC-32 of the payload XOR'ed with the 32-bit value 0x5354554e).

For each STUN Probe Indication or Request, the server retrieves the STUN FINGERPRINT value. For all other packets, the server calculates the checksum as described above. It puts these FINGERPRINT and checksum values in a chronologically ordered list that is sent back in the Report Response.

The contents of the IDENTIFIERS attribute is a list of 4 byte numbers, each using the same encoding that is used for the contents of the FINGERPRINT attribute.

It could have been possible to use the checksum generated in the UDP checksum for this, but this value is generally not accessible to applications. Also, sometimes the checksum is not calculated or is off-loaded to network hardware.

4.2.6. Using Sequence Numbers as Packet Identifiers

When using sequence numbers, a small header similar to the TURN ChannelData header is added in front of all packets that are not a STUN Probe Indication or Request. The sequence number is monotonically incremented by one for each packet sent. The most significant bit of the sequence number is always 0. The server collects the sequence number of the packets sent, or the 4 first bytes of the transaction ID if a STUN Probe Indication or Request is sent. In that case, the most significant bit of the 4 first bytes is set to 1.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Channel Number        |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|                     Sequence number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
/                       Application Data                        /
/                                                               /
|                                                               |
|                               +-------------------------------+
|                               |
+-------------------------------+
```

The Channel Number is always 0xFFFF. The header values are encoded using network order.

The contents of the IDENTIFIERS attribute is a chronologically ordered list of 4 byte numbers, each containing either a sequence number, if the packet was not a STUN Probe Indication or Request, or the 4 first bytes of the transaction ID, with the most significant bit forced to 1, if the packet is a STUN Probe Indication or Request.

5. Probe Support Signaling Mechanisms

The PMTUD mechanism described in this document is intended to be used by any UDP-based protocols that do not have built-in PMTUD capabilities, irrespective of whether those UDP-based protocols are STUN-based or not. So the manner in which a specific protocol discovers that it is safe to send PMTUD probes is largely dependent on the details of that specific protocol, with the exception of the Implicit Mechanism described below, which applies to any protocol.

5.1. Explicit Probe Support Signaling Mechanism

Some of these mechanisms can use a separate signalling mechanism (for instance, an SDP attribute in an Offer/Answer exchange [RFC3264]), or an optional flag that can be set in the protocol that is augmented with PMTUD capabilities. STUN Usages that can benefit from PMTUD capabilities can signal in-band that they support probing by inserting a PMTUD-SUPPORTED attribute in some STUN methods. The decision of which methods support this attribute is left to each specific STUN Usage.

UDP-based protocols that want to use any of these mechanisms, including the PMTUD-SUPPORTED attribute, to signal PMTUD capabilities MUST ensure that it cannot be used to launch an amplification attack. For example, using authentication can ensure this.

5.2. Implicit Probe Support Signaling Mechanism

As a result of the fact that all endpoints implementing this specification are both clients and servers, a Probe Request or Indication received by an endpoint acting as a server implicitly signals that this server can now act as a client and MAY send a Probe Request or Indication to probe the Path MTU in the reverse direction toward the former client, that will now be acting as a server.

The Probe Request or Indication that are used to implicitly signal probing support in the reverse direction MUST be authenticated to prevent amplification attacks.

6. STUN Attributes

6.1. IDENTIFIERS

The IDENTIFIERS attribute carries a chronologically ordered list of UDP packet identifiers.

While Section 4.2.5 and Section 4.2.6 describe two possible methods for acquiring and formatting the identifiers used for this purpose, ultimately each protocol has to define how these identifiers are acquired and formatted. Therefore, the contents of the IDENTIFIERS attribute is opaque.

6.2. PMTUD-SUPPORTED

The PMTUD-SUPPORTED attribute indicates that its sender supports this specification. This attribute has no value part and thus the attribute length field is 0.

7. Security Considerations

The PMTUD mechanism described in this document does not introduce any specific security considerations beyond those described in [RFC4821].

The attacks described in Section 11 of [RFC4821] apply equally to the mechanism described in this document.

The Simple Probing mechanism may be used without authentication because this usage by itself cannot trigger an amplification attack as the Probe Response is smaller than the Probe Request. An unauthenticated Simple Probing mechanism cannot be used in conjunction with the Implicit Probing Support Signaling mechanism in order to prevent amplification attacks.

8. IANA Considerations

This specification defines two new STUN methods and two new STUN attributes. IANA added these new protocol elements to the "STUN Parameters Registry" created by [RFC5389].

8.1. New STUN Methods

This section lists the codepoints for the new STUN methods defined in this specification. See Sections Section 4.1 and Section 4.2 for the semantics of these new methods.

0xXXX : Probe

0xXXX : Report

8.2. New STUN Attributes

This document defines the IDENTIFIERS STUN attribute, described in Section 6.1. IANA has allocated the comprehension-required codepoint 0xXXXX for this attribute.

This document also defines the PMTUD-SUPPORTED STUN attribute, described in Section 6.2. IANA has allocated the comprehension-optional codepoint 0xXXXX for this attribute.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <http://www.rfc-editor.org/info/rfc4821>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <http://www.rfc-editor.org/info/rfc5389>.

9.2. Informative References

[I-D.martinsen-tram-stuntrace] Martinsen, P. and D. Wing, "STUN Traceroute", draft-martinsen-tram-stuntrace-01 (work in progress), June 2015.

[I-D.martinsen-tram-turnbandwidthprobe] Martinsen, P., Andersen, T., Salgueiro, G., and M. Petit-Huguenin, "Traversal Using Relays around NAT (TURN) Bandwidth Probe", draft-martinsen-tram-turnbandwidthprobe-00 (work in progress), May 2015.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, DOI 10.17487/RFC5780, May 2010, <http://www.rfc-editor.org/info/rfc5780>.

Appendix A. Release Notes

This section must be removed before publication as an RFC.

A.1. Modifications between draft-ietf-tram-stun-pmtud-07 and draft-ietf-tram-stun-pmtud-06

- o Updates following Shepherd review.

A.2. Modifications between draft-ietf-tram-stun-pmtud-06 and draft-ietf-tram-stun-pmtud-05

- Nits.

- Restore missing changelog for previous version.

A.3. Modifications between draft-ietf-tram-stun-pmtud-05 and draft-ietf-tram-stun-pmtud-04

- Modifications following Brandon Williams review.

A.4. Modifications between draft-ietf-tram-stun-pmtud-04 and draft-ietf-tram-stun-pmtud-03

- Modifications following Simon Perreault and Brandon Williams reviews.

A.5. Modifications between draft-ietf-tram-stun-pmtud-03 and draft-ietf-tram-stun-pmtud-02

- Add new Overview of Operations section with ladder diagrams.

- Authentication is mandatory for the Complete Probing mechanism, optional for the Simple Probing mechanism.

- All the ICE specific text moves to a separate draft to be discussed in the ICE WG.

- The TURN usage is removed because probing between a TURN server and TURN client is not useful.

- Any usage of PMTUD-SUPPORTED or other signaling mechanisms (formerly knows as discovery mechanisms) must now be authenticated.

- Both probing mechanisms are MTI in the server, the complete probing mechanism is MTI in the client.

- Make clear that stopping after 3 retransmission is done by changing the STUN parameter.

- Define the format of the attributes.

- Make clear that the specification is for any UDP protocol that does not already have PMTUD capabilities, not just STUN based protocols.

- Change the default delay to send the Report Request to 250 ms after the last Indication if the RTO is unknown.

- Each usage of this specification must the format of the IDENTIFIERS attribute contents.

- Better define the implicit signaling mechanism.

- Extend the Security Consideration section.

- Tons of nits.

A.6. Modifications between draft-ietf-tram-stun-pmtud-02 and draft-ietf-tram-stun-pmtud-01

- Cleaned up references.

A.7. Modifications between draft-ietf-tram-stun-pmtud-01 and draft-ietf-tram-stun-pmtud-00

- Added Security Considerations Section.

- Added IANA Considerations Section.

A.8. Modifications between draft-ietf-tram-stun-pmtud-00 and draft-petithuguenin-tram-stun-pmtud-01

- Adopted by WG - Text unchanged.

A.9. Modifications between draft-petithuguenin-tram-stun-pmtud-01 and draft-petithuguenin-tram-stun-pmtud-00

- Moved some Introduction text to the Probing Mechanism section.

- Added cross-reference to the other two STUN troubleshooting mechanism drafts.

- Updated references.

- Added Gonzalo Salgueiro as co-author.

A.10. Modifications between draft-petithuguenin-tram-stun-pmtud-00 and draft-petithuguenin-behave-stun-pmtud-03

- General refresh for republication.

A.11. Modifications between draft-petithuguenin-behave-stun-pmtud-03 and draft-petithuguenin-behave-stun-pmtud-02

- Changed author address.

- Changed the IPR to trust200902.

A.12. Modifications between draft-petithuguenin-behave-stun-pmtud-02 and draft-petithuguenin-behave-stun-pmtud-01

- Defined checksum and sequential numbers as possible packet identifiers.

- Updated the reference to RFC 5389

- The FINGERPRINT attribute is now mandatory.

- Changed the delay between Probe indication and Report request to be RTO/2 or 50 milliseconds.

- Added ICMP packet processing.

- Added Full-Stop Timeout detection.

- Stated that Binding request with PMTUD-SUPPORTED does not start the PMTUD process if already started.

A.13. Modifications between draft-petithuguenin-behave-stun-pmtud-01 and draft-petithuguenin-behave-stun-pmtud-00

- Removed the use of modified STUN transaction but shorten the retransmission for the simple probing mechanism.

- Added a complete probing mechanism.

- Removed the PADDING-RECEIVED attribute.

- Added release notes.

Acknowledgements

Thanks to Eilon Yardeni, Geir Sandbakken, Paal-Erik Martinsen, Tirumaleswar Reddy, Ram Mohan R, Simon Perreault, Brandon Williams, and Tolga Asveren for their review comments, suggestions and questions that helped to improve this document.

Special thanks to Dan Wing, who supported this document since its first publication back in 2008.

Authors' Addresses

Marc Petit-Huguenin
Impedance Mismatch

Email: marc@petit-huguenin.org

Gonzalo Salgueiro
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: gsalguei@cisco.com