TRAM
Internet-Draft
Intended status: Standards Track
Expires: February 20, 2021

M. Petit-Huguenin
Impedance Mismatch
G. Salgueiro
F. Garrido
Cisco

August 19, 2020

Packetization Layer Path MTU Discovery (PLMTUD) For UDP Transports Using Session Traversal Utilities for NAT (STUN)

draft-ietf-tram-stun-pmtud-18

Abstract

The datagram exchanged between two Internet endpoints have to go through a series of physical and virtual links that may have different limits on the upper size of the datagram they can transmit without fragmentation. Because fragmentation is considered harmful, most transports and protocols are designed with a mechanism that permits dynamic measurement of the maximum size of a datagram. This mechanism is called Packetization Layer Path MTU Discovery (PLPMTUD). But the UDP transport and some of the protocols that use UDP were designed without that feature. The Session Traversal Utilities for NAT (STUN) Usage described in this document permits retrofitting an existing UDP-based protocol with such a feature. Similarly, a new UDP-based protocol could simply reuse the mechanism described in this document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on February 20, 2021.

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. Many application layer protocols based on the transport layer protocol UDP do not implement the Path MTU discovery mechanism described in [RFC4821]. These application layer 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 Packetization Layer Path MTU Discovery (PLPMTUD) between a client and a server permits proper measurement of the Path MTU for application layer protocols based on the transport layer protocol UDP 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]). In addition, [I-D.ietf-tsvwg-datagram-plpmtud] provides a robust method for Path MTU Discovery for a broader range of protocols and applications.

2. Overview of Operations

This section is meant to be informative only and is not intended as a substitute 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 decrease in size

until transactions succeed, indicating that the Path MTU has been discovered. It then uses that information to update the Path MTU.

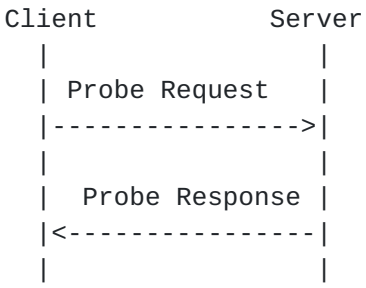


Figure 1: Simple Probing Example

If the Complete Probing mechanism (as described in Section 4.2) is chosen, then the client sends Probe Indications of various sizes (as specified in [RFC4821]) 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 examines the received reports to determine which probes were successful. When a probe succeeds with a larger size than the current PMTU, the PMTU is increased. When the probes indicate the current PMTU is not supported the size is decreased. This mechanism acts to detect that traffic is being back holed.

Because of the possibility of amplification attack, the Complete Probing mechanism must be authenticated as specified in Section 5.1. Particular care must be taken to prevent amplification when an external mechanism is used to trigger the Complete Probing mechanism.

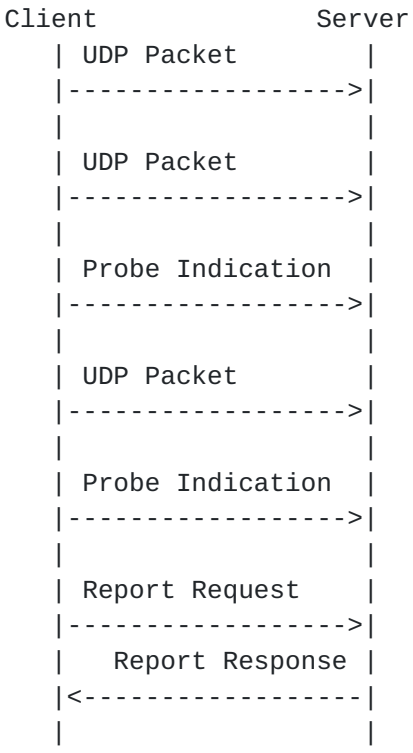


Figure 2: Complete Probing Example

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

4. Probing Mechanisms

The Probing mechanism is used to discover the Path MTU in one direction only: from the client to the server. Both endpoints MAY behave as a client and a server to achieve bi-directional path discovery.

Two Probing mechanisms are described: a Simple Probing mechanism and a more complete mechanism that can converge more quickly and find an appropriate Path MTU in the presence of congestion. Additionally, the Simple Probing mechanism does not require authentication except where used as an implicit signaling mechanism, 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 attribute over UDP with the DF bit set in the IP header for IPv4 packets and IPv6 packets without the Fragment Header included.

Note: Routers might be configured to clear the DF bit or ignore the DF bit which can be difficult or impossible to detect if reassembly occurs prior to receiving the packet.

4.1.1. Sending a Probe Request

A client forms a Probe Request by using the Probe Method and following the rules in Section 6.1 of [I-D.ietf-tram-stunbis].

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 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 PADDING bits MUST be set to zero. The client MUST add the FINGERPRINT attribute so the STUN messages are disambiguated from the other protocol packets as specified in Section 7 of [I-D.ietf-tram-stunbis].

Then the client sends the Probe Request to the server over UDP with the DF bit set for IPv4 packets and IPv6 packets without the Fragment Header included. For the purpose of this transaction, the Rc parameter is set to 3 and the initial value for RTO stays at 500 ms as specified in Section 6.2.1 of [I-D.ietf-tram-stunbis]

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 [I-D.ietf-tram-stunbis].

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 as specified in Section 7 of [I-D.ietf-tram-stunbis]. 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 section 6.3.1 of [I-D.ietf-tram-stunbis] and MUST ignore the PADDING attribute. 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" or "Packet Too Big" 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. Validation MUST be performed on the ICMP packet as specified in [I-D.ietf-tsvwg-datagram-plpmtud].

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 for IPv4 packets and IPv6 packets without the Fragment Header included 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 when no application layer protocol-specific identification mechanism is available.

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 a Probe Indications and Report Request

A client forms a Probe Indication by using the Probe Method and following the rules in [I-D.ietf-tram-stunbis] Section 6.1. The client adds to a 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. The PADDING bits MUST be set to zero. 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 a Probe Indication to the server over UDP with the DF bit set for IPv4 packets and IPv6 packets without the Fragment Header included.

Then the client forms a Report Request by following the rules in [I-D.ietf-tram-stunbis] Section 6.1. 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 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" or "Packet Too Big" is received then this is interpreted as a Probe Failure, as defined in [RFC4821] Section 7.5. Validation MUST be performed on the ICMP packet as specified in [I-D.ietf-tsvwg-datagram-plpmtud].

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 given flow's identifier 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 [I-D.ietf-tram-stunbis] Section 6.1. 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 [I-D.ietf-tram-stunbis] and MUST ignore the PADDING attribute.

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 [I-D.ietf-tram-stunbis]. If the response IDENTIFIERS attribute contains the identifier of a Probe Indication, then this is interpreted as a Probe Success for this probe, as defined in [RFC4821] Section 7.5. If a 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 a 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 keeps a chronologically ordered list of the packets it transmits, along with an associated checksum value. For STUN Probe Indication or Request packets, the associated checksum value is the FINGERPRINT value from the packet; for other packets a checksum value is computed. The value of the checksum is computed as the CRC-32 of the UDP payload, as defined by the Length field of the UDP datagram [RFC4821], XOR'ed

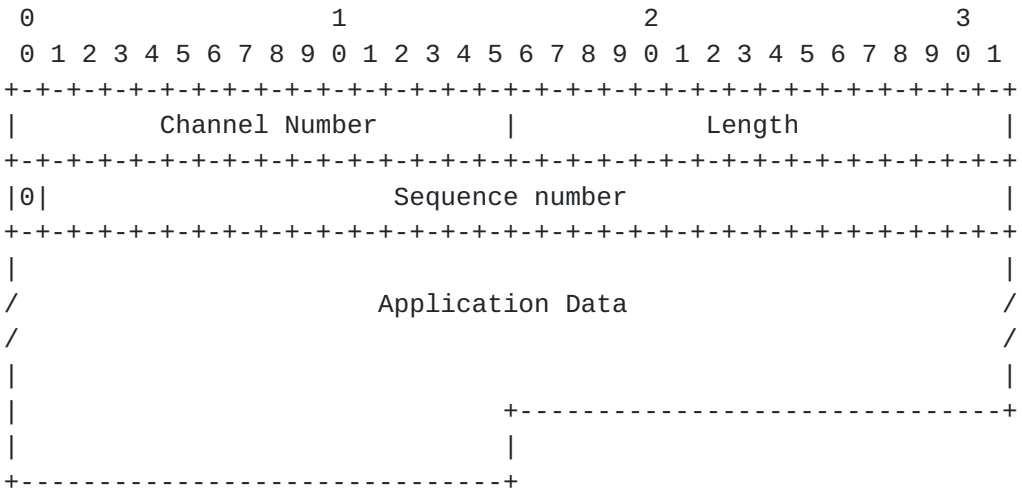
with the 32-bit value 0x5354554e. The 32-bit CRC is the one defined in ITU V.42 [[ITU.V42.2002], which has a generator polynomial of x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 + x^10 + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.

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.

4.2.6. Using Sequence Numbers as Packet Identifiers

When using sequence numbers, a small header similar to the TURN ChannelData header, as defined in Section 11.4 of [RFC5766], is added in front of all packets that are not a STUN Probe Indication or Request. The initial sequence number MUST be randomized and 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.



The Channel Number is always 0xFFFF. The Length field specifies the length in bytes of the sequence number and application data fields. 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.

An amplification attack can be prevented using techniques such as:

- Authentication, where the source of the packet and the destination share a secret.

- 3 way handshake with some form of unpredictable cookie.

- Make sure that the total size of the traffic potentially generated is lower than the size of the request that generated it.

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 mechanism, as incorporated into the STUN usage or protocol being used. This attribute has no value part and thus the attribute length field is 0.

6.3. PADDING

The PADDING attribute allows for the entire message to be padded to force the STUN message to be divided into IP fragments. The PADDING bits MUST be set to zero. PADDING can be used in either Binding Requests or Binding Responses.

PADDING MUST NOT be longer than the length that brings the total IP datagram size to 64K, minus the IP and UDP headers and the other STUN components. It SHOULD be equal in length to the MTU of the outgoing interface, rounded up to an even multiple of four bytes and SHOULD ensure a probe does not result in a packet larger than the MTU fo the outgoing interface. STUN messages sent with PADDING are intended to test the behavior of UDP fragmentation, therefore they are an exception to the usual rule that STUN messages need to be less than the PMTU for the path.

7. DPLPMTUD Considerations

This section specifies how the PMTUD mechanism described in this document conforms to Sections 3 and 6.1 of [I-D.ietf-tsvwg-datagram-plpmtud] and indicates where each requirement is addressed.

7.1. Features Required to provide Datagram PLPMTUD

This section covers Section 3 of [I-D.ietf-tsvwg-datagram-plpmtud] and refers back to sections in this document covering each of the feature requirements.

1. Managing the PLMPTU: This requirement is fulfilled by the Simple probing and Complete probing mechanisms as discussed in Section 2, Section 4.1 and Section 4.2 of this document.

2. Probe packets: This requirement is fulfilled by including a PADDING attribute which indicates that the DF bit is set in the IP header for IPv4 packets and not including the Don't Fragment header in IPv6 packets as discussed in Section 4.1 and Section 4.2 of this document.

3. Reception feedback: This requirement fulfilled by the Probe Response and Report Response in Section 2 of this document.

4. Probe loss recovery: This requirement is fulfilled by requiring that the PADDING bits MUST be set to zero as discussed in Section 4.1.1 and Section 4.2.1 of this document. No retransmission is required as there is no user data is being transmitted in the probe.

5. PMTU parameters: This requirement is fulfilled by setting the Selected Probe Size as defined in [RFC4821] and discussed in Section 4.1 and Section 4.2 of this document.

6. Processing PTB messages: This requirement is fulfilled by the Probe Response and Report Response in Section 4.1.3 and Section 4.2.2 of this document.

7. Probing and congestion control: This requirement is fulfilled by the Probe Request and Probe Indication discussed in Section 4.1.1 and Section 4.2.1 of this document. It conforms to Section 6.2.1 of [I-D.ietf-tram-stunbis].

8. Probing and flow control: This requirement is out of scope and is not discussed in this document.

9. Shared PLPMTU state: This requirement is out of scope and is not discussed in this document.

Datagram reordering: This requirement is fulfilled by the Report Response in Section 4.2 of this document.

Datagram delay and duplication: his requirement is fulfilled by the Report Response in Section 4.2 of this document.

When to probe: This requirement is discussed in Section 2 of this document.

7.2. Application Support for DPLPMTUD with UDP

This section covers Section 6.1 of [I-D.ietf-tsvwg-datagram-plpmtud] and refers back to which sections in this document covering each of the feature requirements.

6.1.1 Application Request: This requirement is fulfilled by the Simple probing and Complete probing mechanisms as discussed in Section 2, Section 4.1 and Section 4.2 of this document.

6.1.2 Application Response: This requirement is fulfilled by the Simple probing and Complete probing mechanisms as discussed in Section 4.1 and Section 4.2 of this document.

6.1.3 Sending Application Probe Packets: This requirement is fulfilled by requiring that the PADDING bits MUST be set to zero as discussed in Section 4.1.1 and Section 4.2.1 of this document.

6.1.4 Initial Connectivity: This requirement is fulfilled by the Implicit and Explicit Probe Support Signaling mechanisms as discussed Section 5 of this document.

6.1.5 Validating the Path: This requirement is fulfilled by the Report Request and Report Response mechanisms as discussed in Section 4.2 of this document.

6.1.6 Handling of PTB Messages: This requirement is fulfilled by the Probe Response and Report Response in Section 4.1.3 and Section 4.2.2 of this document.

8. Security Considerations

The PMTUD mechanism described in this document, when used without the signalling mechanism described in Section 5.1, does not introduce any specific security considerations beyond those described in [RFC4821] and [I-D.ietf-tsvwg-datagram-plpmtud].

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

The amplification attacks introduced by the signalling mechanism described in Section 5.1 can be prevented by using one of the techniques described in that section.

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 except when used in conjunction with the Implicit Probing Support Signaling mechanism.

9. IANA Considerations

This specification defines two new STUN methods and two new STUN attributes.

9.1. New STUN Methods

IANA is requested to add the following methods to the STUN Method Registry:

0xXXX : Probe

0xXXX : Report

See Sections Section 4.1 and Section 4.2 for the semantics of these new methods.

9.2. New STUN Attributes

IANA is requested to add the following attributes to the STUN Method Registry:

Comprehension-required range (0x0000-0x7FFF):
0xXXXX: IDENTIFIERS

Comprehension-optional range (0x8000-0xFFFF)
0xXXXX: PMTUD-SUPPORTED

0x0026: PADDING

The IDENTIFIERS STUN attribute is defined in Section 6.1, the PMTUD-SUPPORTED STUN attribute is defined in Section 6.2; the PADDING STUN attribute is defined in Section 6.3.

NOTE: TO BE DELETED BEFORE PUBLICATION. PLEASE NOTE THAT THE PADDING ATTRIBUTE ENTRY IS REPLACING THE ENTRY MADE BY RFC5780 (EXPERIMENTAL). THE SAME VALUE AND NAME ARE USED BUT THE REFERENCE SHOULD BE CHANGED TO THIS STANDARDS TRACK DOCUMENT.

10. References

10.1. Normative References

[I-D.ietf-tram-stunbis]
Petit-Huguenin, M., Salgueiro, G., Rosenberg, J., Wing, D., Mahy, R., and P. Matthews, "Session Traversal Utilities for NAT (STUN)", draft-ietf-tram-stunbis-21 (work in progress), March 2019.

[I-D.ietf-tsvwg-datagram-plpmtud]
Fairhurst, G., Jones, T., Tuexen, M., Ruengeler, I., and T. Voelker, "Packetization Layer Path MTU Discovery for Datagram Transports", draft-ietf-tsvwg-datagram-plpmtud-22 (work in progress), December 2019.

[ITU.V42.2002]
International Telecommunications Union, "Error-correcting Procedures for DCEs Using Asynchronous-to-Synchronous Conversion", ITU-T Recommendation V.42, 2002.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <http://www.rfc-editor.org/info/rfc8174>.

10.2. 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>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <https://www.rfc-editor.org/info/rfc5766>.

Appendix A. Release Notes

This section must be removed before publication as an RFC.

A.1. Modifications between draft-ietf-tram-stun-pmtud-18 and draft-ietf-tram-stun-pmtud-17

- Modifications to address DISCUSS and COMMENT from IESG review. updated section 7.

A.2. Modifications between draft-ietf-tram-stun-pmtud-17 and draft-ietf-tram-stun-pmtud-16

- Modifications to address DISCUSS and COMMENT from IESG review. Added section 7.

A.3. Modifications between draft-ietf-tram-stun-pmtud-16 and draft-ietf-tram-stun-pmtud-15

- Modifications to address DISCUSS and COMMENT from IESG review

A.4. Modifications between draft-ietf-tram-stun-pmtud-15 and draft-ietf-tram-stun-pmtud-14

- Modifications to address DISCUSS and COMMENT from IESG review

A.5. Modifications between draft-ietf-tram-stun-pmtud-14 and draft-ietf-tram-stun-pmtud-13

- Modifications to address COMMENTS from IESG review

A.6. Modifications between draft-ietf-tram-stun-pmtud-13 and draft-ietf-tram-stun-pmtud-12

- Modifications to address nits

A.7. Modifications between draft-ietf-tram-stun-pmtud-12 and draft-ietf-tram-stun-pmtud-11

- Modifications following IESG review. Incorporated RFC5780 PADDING attribute (Adam's Discuss) and added IPv6 language (Suresh's Discuss).

A.8. Modifications between draft-ietf-tram-stun-pmtud-11 and draft-ietf-tram-stun-pmtud-10

- Modifications following IESG review.

A.9. Modifications between draft-ietf-tram-stun-pmtud-10 and draft-ietf-tram-stun-pmtud-09

- Modifications following reviews for gen-art (Roni Even) and secdir (Carl Wallace).

A.10. Modifications between draft-ietf-tram-stun-pmtud-09 and draft-ietf-tram-stun-pmtud-08

- Add 3 ways of preventing amplification attacks.

A.11. Modifications between draft-ietf-tram-stun-pmtud-08 and draft-ietf-tram-stun-pmtud-07

- Updates following Spencer's review.

A.12. Modifications between draft-ietf-tram-stun-pmtud-07 and draft-ietf-tram-stun-pmtud-06

- Updates following Shepherd review.

A.13. Modifications between draft-ietf-tram-stun-pmtud-06 and draft-ietf-tram-stun-pmtud-05

- Nits.
- Restore missing changelog for previous version.

A.14. Modifications between draft-ietf-tram-stun-pmtud-05 and draft-ietf-tram-stun-pmtud-04

- Modifications following Brandon Williams review.

A.15. Modifications between draft-ietf-tram-stun-pmtud-04 and draft-ietf-tram-stun-pmtud-03

- Modifications following Simon Perreault and Brandon Williams reviews.

A.16. 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.17. Modifications between draft-ietf-tram-stun-pmtud-02 and draft-ietf-tram-stun-pmtud-01

- Cleaned up references.

A.18. Modifications between draft-ietf-tram-stun-pmtud-01 and draft-ietf-tram-stun-pmtud-00

- Added Security Considerations Section.
- Added IANA Considerations Section.

A.19. Modifications between draft-ietf-tram-stun-pmtud-00 and draft-petithuguenin-tram-stun-pmtud-01

- Adopted by WG - Text unchanged.

A.20. 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.21. Modifications between draft-petithuguenin-tram-stun-pmtud-00 and draft-petithuguenin-behave-stun-pmtud-03

- General refresh for republication.

A.22. Modifications between draft-petithuguenin-behave-stun-pmtud-03 and draft-petithuguenin-behave-stun-pmtud-02

- Changed author address.
- Changed the IPR to trust200902.

A.23. 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.24. 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, Tolga Asveren, Spencer Dawkins, Carl Wallace, and Roni Even 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

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

Email: fegarrid@cisco.com