TRAM
Internet-Draft
Intended status: Standards Track
Expires: July 29, 2016

M. Petit-Huguenin
Impedance Mismatch
G. Salgueiro
Cisco

January 26, 2016

Path MTU Discovery Using Session Traversal Utilities for NAT (STUN) draft-ietf-tram-stun-pmtud-01

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

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 http://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 July 29, 2016.

Table of Contents

1. Introduction

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

This document only describes how probing mechanisms are implemented with Session Traversal Utilities for NAT (STUN). The algorithm to find the path MTU is described in [RFC4821].

The STUN usage defined in this document for Path MTU Discovery (PMTUD) between a client and a server simplifies troubleshooting and has multiple applications across a wide variety of technologies.

Additional network characteristics like 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]) can be discovered using complementary techniques.

2. 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.

3. Probing Mechanisms

A client MUST NOT send a probe if it does not have knowledge that the server supports this specification. This is done by an external mechanism specific to each UDP protocol. Section 6 describes some of this mechanisms.

The probe 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.

The simple probing mechanism is implemented by sending a Probe Request with a PADDING [RFC5780] attribute and the DF bit set over UDP. A router on the path to the server can reject this request with an ICMP message or drop it. The client SHOULD cease retransmissions after 3 missing responses.

The complete probing mechanism is implemented by sending one or more Probe Indication with a PADDING attribute and the DF bit set over UDP then 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 time ordered list of identifiers of 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 does not contain an identifier, the complete probing mechanism needs a way to identify each packet received. While there are other possible packet identification schemes, this document describes two different ways to identify a specific packet.

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

In the second packet identifier mechanism, the client adds a sequential number in front of each UDP packet sent. The server sends back the ordered list of sequential numbers received that the client compares to its own list

4. Simple Probing Mechanism

4.1. Sending a Probe Request

A client forms a Probe Request by following the rules in Section 7.1 of [RFC5389]. No authentication method is used. 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.

Then the client sends the Probe Request to the server over UDP with the DF bit set. The client SHOULD stop retransmitting after 3 missing responses.

4.2. Receiving a Probe Request

A server receiving a Probe Request MUST process it as specified in [RFC5389]. The server MUST NOT challenge the client.

The server then creates a Probe Response. The server MUST add the FINGERPRINT attribute. The server then sends the response to the client.

4.3. 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 transactions fails in timeout, then this is interpreted as a Probe Inconclusive as defined in [RFC4821] section 7.6.4.

5. Complete Probing Mechanism

5.1. Sending the Probe Indications and Report Request

A client forms a Probe Indication by 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. The client MUST add the FINGERPRINT attribute.

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. No authentication method is used. The client MUST add the FINGERPRINT attribute.

Then the client waits half the RTO if it is known or 50 milliseconds after sending the Probe Indication and sends the Report Request to the server over UDP.

5.2. 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.

5.3. Receiving a Probe Indication and Report Request

A server supporting this specification and knowing that the client also supports it will keep the identifiers of all packets received in a list ordered by receiving time. The same identifier can appear multiple times in the list because of retransmission. The maximum size of this list is calculated so 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 already full.

A server receiving a Report Request MUST process it as specified in [RFC5389]. The server MUST NOT challenge the client.

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

5.4. 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 other packets identifier sent before or after the Probe Indication cannot also be found, this is interpreted as a Probe Inconclusive as defined in [RFC4821] Section 7.5. If the Report Transaction fails in timeout, this is interpreted as a Full-Stop Timeout as defined in [RFC4821] Section 3.

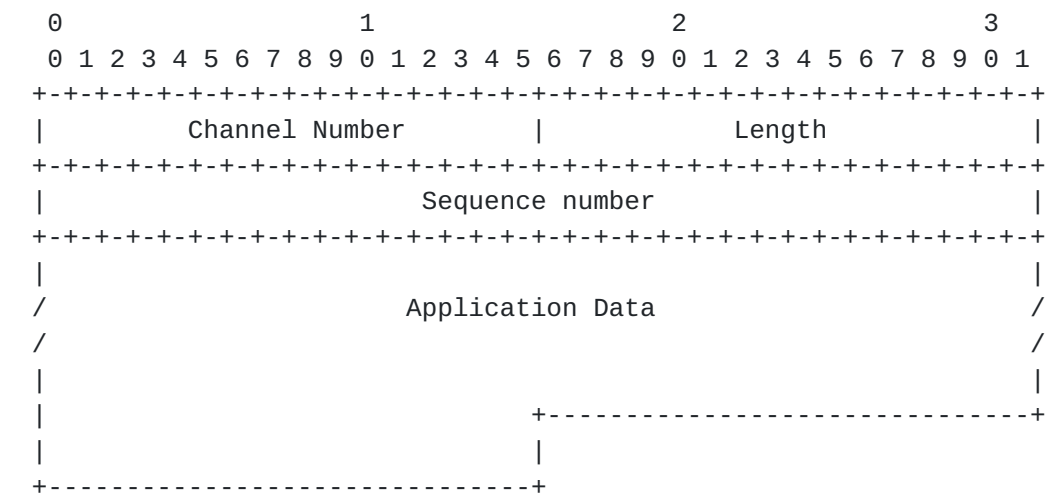
5.5. Using Checksum as Packet Identifiers

When using checksum as packet identifiers, the client calculate the checksum for each packet sent over UDP and keep this checksum in an ordered list. The server does the same thing and send back this list in the Report Response.

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 off-loaded to the network card.

5.6. Using Sequential Numbers as Packet Identifiers

When using sequential numbers, a small header similar to the TURN ChannelData header is added in front of all non-STUN packets. The sequential number is incremented for each packet sent. The server collects the sequence number of the packets sent.



The Channel Number is always 0xFFFF.

6. Probe Support Discovery Mechanisms

6.1. Implicit Mechanism

An endpoint acting as a client for the STUN usage described in this specification MUST also act as a server for this STUN usage. This means that a server receiving a probe can assumes that it can acts as a client to discover the path MTU to the IP address and port from which it received the probe.

6.2. Probe Support Discovery with TURN

A TURN client supporting this STUN usage will add a PMTUD-SUPPORTED attribute to the Allocate Request sent to the TURN server. The TURN server can immediately start to send probes to the TURN client on reception of an Allocation Request with a PMTUD-SUPPORTED attribute. The TURN client will then use the Implicit Mechanism described above to send probes.

6.3. Probe Support Discovery with ICE

An ICE [RFC5245] client supporting this STUN usage will add a PMTUD-SUPPORTED attribute to the Binding Request sent during a connectivity check. The ICE server can immediately start to send probes to the ICE client on reception of a Binding Request with a PMTUD-SUPPORTED attributed. Local candidates receiving Binding Request with the PMTUD-SUPPORTED flag must not start PMTUD with the remote candidate if already done so. The ICE client will then use the Implicit Mechanism described above to send probes.

7. Security Considerations

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

The attack described in [RFC4821] applies equally to the mechanism described in this document.

8. IANA Considerations

This specification defines two new STUN method 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 and Section 5 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 5. IANA has allocated the comprehension-required codepoint 0xXXXX for this attribute.

This document also defines the PMTUD-SUPPORTED STUN attribute, described in Section 6. IANA has allocated the comprehension-optional codepoint 0xXXXX for this attribute.

9. Acknowledgements

Thanks to Eilon Yardeni, Geir Sandbakken and Paal-Erik Martinsen 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.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

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.

[I-D.ietf-payload-flexible-fec-scheme]
Singh, V., Begen, A., Zanaty, M., and G. Mandyam, "RTP Payload Format for Flexible Forward Error Correction (FEC)", draft-ietf-payload-flexible-fec-scheme-01 (work in progress), October 2015.

[RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, May 2010.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, July 2013.

Appendix A. Release Notes

This section must be removed before publication as an RFC.

A.1. Modifications between draft-ietf-tram-stun-pmtud-01 and draft-ietf-tram-stun-pmtud-00

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

A.2. Modifications between draft-ietf-tram-stun-pmtud-00 and draft-petithuguenin-tram-stun-pmtud-01

- Adopted by WG - Text unchanged.

A.3. 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.4. Modifications between draft-petithuguenin-tram-stun-pmtud-00 and draft-petithuguenin-behave-stun-pmtud-03

- General refresh for republication.

A.5. Modifications between draft-petithuguenin-behave-stun-pmtud-03 and draft-petithuguenin-behave-stun-pmtud-02

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

A.6. Modifications between draft-petithuguenin-behave-stun-pmtud-02 and draft-petithuguenin-behave-stun-pmtud-01

- Replaced the transactions identifiers by packet identifiers
- 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.7. 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.

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