Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2009

M. Petit-Huguenin
8x8, Inc.

November 3, 2008

Path MTU Discovery Using Session Traversal Utilities for NAT (STUN) draft-petithuguenin-behave-stun-pmtud-02

Status of this Memo

Abstract

This document describes a Session Traversal Utilities for NAT (STUN) usages for discovering the path MTU between a client and a server.

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 describe how probing mechanisms are implemented with STUN. The algorithm to find the path MTU is described in [RFC4821].

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 [I-D.ietf-behave-nat-behavior-discovery] 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. As example, 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

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].

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.

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 [I-D.ietf-behave-nat-behavior-discovery] 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.

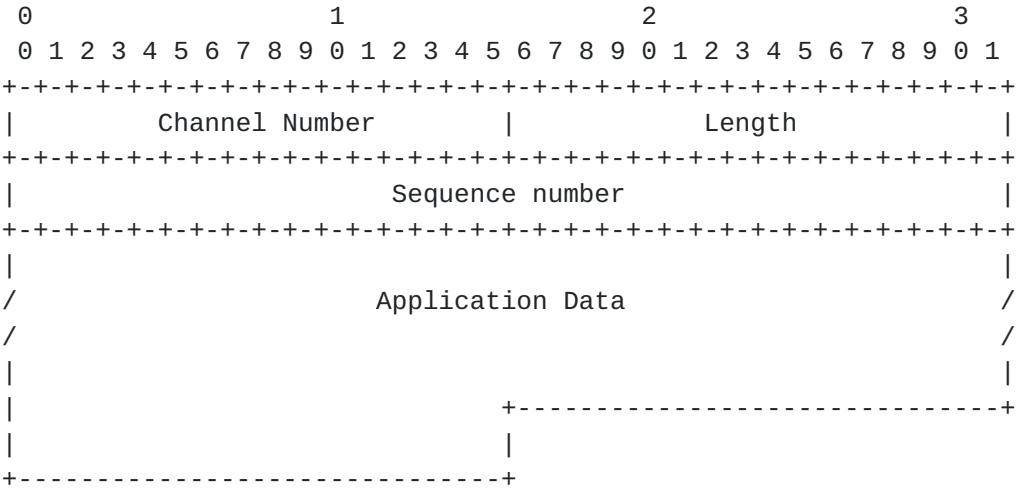
5.5. Using Checksum as Packet Identifiers

When using checksum as packet identifiers, the client calculate the checksum for each packets 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 [I-D.ietf-mmusic-ice] 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. New STUN Method

This specification defines the following new STUN methods:

0x801 : Probe
0x802 : Report

8. New STUN Attributes

This specification defines the following new STUN attributes:

0x4001 : IDENTIFIERS
0xC001 : PMTUD-SUPPORTED

8.1. IDENTIFIERS

The IDENTIFIERS attribute is used in Report Response. It contains a list of UDP packet identifiers.

8.2. PMTUD-SUPPORTED

The PMTUD-SUPPORTED attribute is used in STUN usages and extensions to signal the support of this specification. This attribute has no content.

9. Security Considerations

TBD

10. IANA Considerations

TBD

11. Acknowledgements

Thanks to Dan Wing and Eilon Yardeni for their comments, suggestions and questions that helped to improve this document.

This document was written with the xml2rfc tool described in [RFC2629].

12. References

12.1. 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.

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-19 (work in progress), October 2007.

12.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[I-D.ietf-behave-nat-behavior-discovery]
MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using STUN", draft-ietf-behave-nat-behavior-discovery-05 (work in progress), November 2008.

Appendix A. Release notes

This section must be removed before publication as an RFC.

A.1. Modifications between -02 and -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.2. Modifications between -01 and -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.

Author's Address

Marc Petit-Huguenin
8x8, Inc.
3151 Jay Street
Santa Clara, CA 95054
US

Phone: +1 408 654 0875
Email: marc@8x8.com

Full Copyright Statement

Intellectual Property