

Path MTU Discovery Using STUN

draft-petithuguenin-tram-stun-pmtud-01

IETF-93

Prague, July20, 2015

Marc Petit-Huguenin, Gonzalo Salgueiro



History

- Original draft was proposed in BEHAVE WG back in 2008
- Now resurrected as a TRAM draft that fits in nicely with other proposed STUN-based measurement/troubleshooting drafts
- Intended as a generic solution for any UDP based protocol that does not already have a way to measure the PMTUD

Overview (1 of 6)

- Describes a STUN usage for PMTUD between a client and a server
- Document only describes how probing mechanism is implemented with STUN; the actual algorithm to determine the path MTU is described in RFC4821
- Probe mechanism used to discover PMTU in one direction only (client to server)

Overview (2 of 6)

- Two probing methods defined:
 - 1. Simple Probing Mechanism:
 - 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.

Overview (3 of 6)

- 2. Complete Probing Mechanism:
 - Implemented by sending one or more Probe Indications with PADDING attribute and DF bit set over UDP then a Report Request to same server.
 - A router on the path to the server can reject this indication with an ICMP message or drop it.

Overview (4 of 6)

- 2. Complete Probing Mechanism (cont'd):
 - 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.

Overview (5 of 6)

- 2. Complete Probing Mechanism (cont'd):
 - Because UDP packets do 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.

Overview (6 of 6)

- First packet identifier mechanism: the server computes a checksum over each packet received and sends back to client the ordered list of checksums. The client compares this list to its own list of checksums.
- Second packet identifier mechanism: client adds a sequential number in front of each UDP packet sent. Server sends back ordered list of sequential numbers received that client compares to its list.

Next Steps

- Does WG have an interest to work on this?
- This is one of three proposed STUN-based measurement/troubleshooting drafts. Do we create a unified framework?
- Do we merge all three into a single document? Or are they best kept separate?
- Adopt (1 or 3 docs) as WG items?