TRAM
Internet-Draft
Intended status: Standards Track
Expires: June 26, 2015

T. Reddy
D. Wing
P. Martinsen
Cisco
V. Singh
callstats.io

December 23, 2014

Discovery of path characteristics using STUN
draft-reddy-tram-stun-path-data-00

Abstract

A host with multiple interfaces needs to choose the best interface for communication. Oftentimes, this decision is based on a static configuration and does not consider the path characteristics, which may affect the user experience.

This document describes a mechanism for an endpoint to discover the path characteristics using Session Traversal Utilities for NAT (STUN) messages. The measurement information can then be used to influence the endpoint's Interactive Connectivity Establishment (ICE) candidate pair selection algorithm.

Table of Contents

1. Introduction

The ICE [RFC5245] mechanism uses a prioritization formula to order the candidate pairs and perform connectivity checks, in which the most preferred address pairs are tested first and when a sufficiently good pair is discovered, that pair is used for communications and further connectivity tests are stopped. This approach works well for an endpoint with a single interface, but is too simplistic for endpoints with multiple interfaces, wherein a candidate pair with a lower priority might infact have better path characteristics (e.g., round-trip time, loss, etc.). The ICE connectivity checks can be used to measure the path characteristics but the issue is that STUN responses to re-transmitted requests are indistinguishable from each other.

This draft extends STUN [RFC5389] to distinguish STUN responses to re-transmitted requests and assists the client in determining the path characteristics like round-trip time (RTT) and packet loss in each direction between endpoints. These metrics can be used by the controlling agent to influence the ICE candidate pair priorities.

2. Notational Conventions

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].

This note uses terminology defined in ICE [RFC5245] and STUN [RFC5389].

3. Path characteristics determination mechanism

When multiple paths are available for communication, the endpoint sends ICE connectivity checks across each path and perhaps chooses the path with the lowest round trip time. Choosing the path with the lowest round trip time is a reasonable approach, but re- transmits can cause an otherwise-good path to appear flawed. However, STUN's retransmission algorithm [RFC5389] cannot determine the round-trip time if a STUN request packet is re-transmitted, because each packet request and retransmission is identical, further, several STUN requests may be sent before the connectivity between pairs is ascertained (see Section 16 of [RFC5245]). To resolve the issue of identical request and response packets in a STUN transaction, this document changes that retransmission behavior for idempotent packets. In addition to determining RTT, it is also desirable to detect which path direction caused packet loss, described as "bi-directional path characteristics," below. This is achieved by defining a new STUN attribute and requires compliant STUN (TURN, ICE) servers to count retransmitted request packets.

This specification defines a new comprehension-optional STUN attribute PATH-CHARACTERISTIC. PATH-CHARACTERISTIC will have a STUN Type TBD-CA. This type is in the comprehension-optional range, which means that STUN agents can safely ignore the attribute if they do not understand it.

If a client wishes to determine the path characteristics, it inserts the PATH-CHARACTERISTIC attribute in a STUN request. In the PATH-CHARACTERISTIC attribute client sends the number of times the STUN request is retransmitted with the same Transaction ID. The server would echo back the retransmission count in the response so that client can distinguish STUN responses from the re-transmitted requests. Hence, the endpoint can use the STUN requests and responses to determine the round-trip time (RTT). The server may also convey the number of times it received the request with the same Transaction ID and the number of responses it has sent for the STUN request to the client. Further, this information enables the client to determine packet loss in each direction.

3.1. The PATH-CHARACTERISTIC attribute in request

The PATH-CHARACTERISTIC attribute in a STUN request takes a 1-byte Value, which means that the Length is 1 and 3 bytes of padding are required after the value (Section 15 of [RFC5389]). When sending a STUN request, the PATH-CHARACTERISTIC attribute allows a client to indicate to the server that it wants to determine path characteristics. If the client receives a STUN response with error code 420 (Unknown Attribute) and PATH-CHARACTERISTIC is listed in the UNKNOWN-ATTRIBUTE attribute of the message, the client SHOULD retransmit the original request without the PATH-CHARACTERISTIC attribute. However this case is not expected to occur, due to the use of the comprehension-optional attribute type.

This specification updates one the STUN message structuring rules explained in Section 6 of [RFC5389] that resends of the same request reuse the same transaction ID and are bit-wise identical to the previous request. The ReTransCnt in the PATH-CHARACTERISTIC attribute will be incremented by 1 for every re-transmission and the re-transmitted STUN request MUST be bit-wise identical to the previous request except for the ReTransCnt value.

The format of the value in PATH-CHARACTERISTIC attribute in the request is:

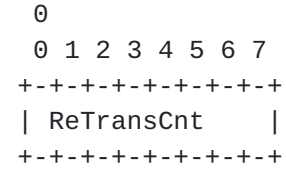

Figure 1: PATH-CHARACTERISTIC attribute in request

The field is described below:

ReTransCnt: Number of times request is re-transmitted with the same transaction ID to the server.

3.2. The PATH-CHARACTERISTIC attribute in response

When a server receives a STUN request that includes a PATH-CHARACTERISTIC attribute, it processes the request as per the STUN specification [RFC5389] plus the specific rules mentioned here. The server checks the following:

- If the PATH-CHARACTERISTIC attribute is not recognized, ignore the attribute because its type indicates that it is comprehension-

optional. This should be the existing behavior as explained in section 3.1 of [RFC5389].

- The server that supports PATH-CHARACTERISTIC attribute MUST echo back ReTransCnt in the response.

- If the server is stateless or does not want to remember the transaction ID then it would populate value 0 for the ReqTransCnt and RespTransCnt fields in PATH-CHARACTERISTIC attribute sent in the response .If the server is stateful then it populates ReqTransCnt with the number of times it received the STUN request with the same transaction ID and RespTransCnt with the number of responses it has sent for the STUN request.

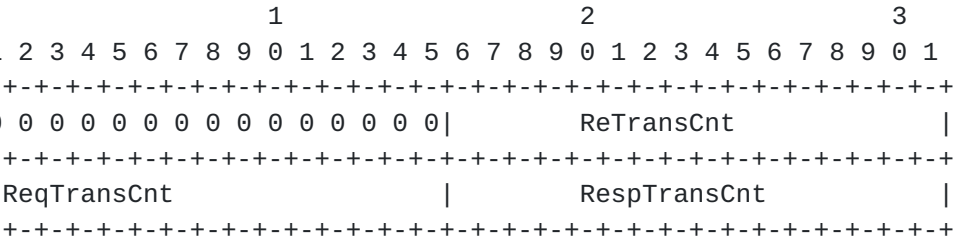

Figure 2: PATH-CHARACTERISTIC attribute in response

The fields are described below:

ReTransCnt: Copied from request.

ReqTransCnt: Number of times request is received from the client with the same transaction ID.

RespTransCnt: Number of responses sent to the client for the same transaction ID.

3.2.1. Example Operation

The operation is described in Figure 3. In the first case, all the requests and responses are received correctly. In the upstream loss case, the first request is lost, but the second one is received correctly, the client on receiving the response notes that while 2 requests were sent, only one was received by the server, also the server realizes that the RespTransCnt does not match the ReTransCnt, therefore 1 request was lost. This may also occur at startup in the presence firewalls or NATs that block unsolicited incoming traffic. In the downstream loss case, the responses get lost, client expecting multiple response notes that while the server responded to 3 requests but only 1 response was received. In the both loss case, requests and responses get lost in tandem, the server notes one request packet

was not received, while the client expecting 3 responses received only one, it notes that one request and response packets were lost.

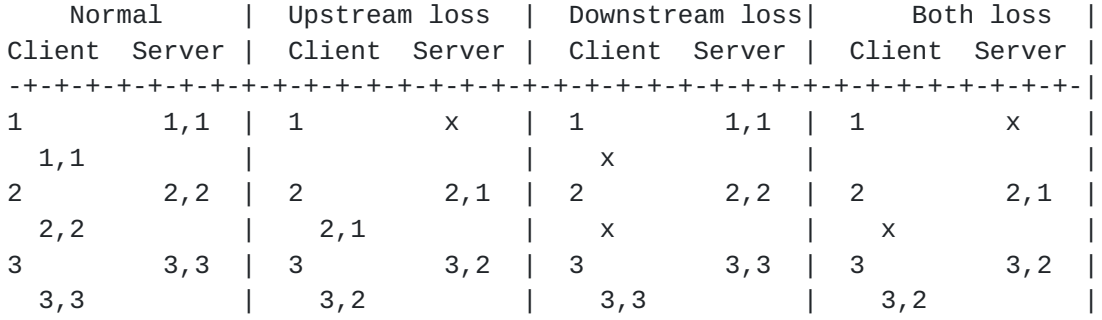

Figure 3: Retransmit Operation between client and Server

4. Usecases

The STUN attribute defined in this specification can be used by applications in the following scenarios:

- When an endpoint has multiple interfaces (for example 3G, 4G, WiFi, VPN, etc.), an ICE agent can choose the interfaces for media streams according to the path characteristics. After STUN responses to STUN checks are received, the ICE agent using regular nomination can sort the ICE candidate pairs according to the path characteristics discovered using STUN. The controlling agent can assign highest priority to candidate pair which best fulfills the desired path characteristics.

- [TODO: Add details of http://juberti.github.io/draughts/nombis/draft-uberti-mmusic-nombis.html that explains simplifying and improving the procedures for candidate nomination in ICE to make dynamic decisions.]

- When a host has multiple interfaces available an MPRTP [I-D.ietf-avtcore-mprtp] application can choose the interface for the primary subflow and interfaces for subsequent subflows according to the path characteristics discovered using STUN. For example, the scheduling algorithm described in [ACM-MPRTP] uses both path loss and latency for choosing the most suitable subset of paths.

- The STUN extension proposed in this specification can also be used to choose a TURN server that provides the best user experience (section 3.1 of [I-D.patil-tram-turn-serv-selection]).

5. IANA Considerations

[Paragraphs in braces should be removed by the RFC Editor upon publication]

[The PATH-CHARACTERISTIC attribute requires that IANA allocate a value in the "STUN attributes Registry" from the comprehension-optional range (0x8000-0xFFFF), to be replaced for TBD-CA throughout this document]

This document defines the PATH-CHARACTERISTIC STUN attribute, described in Section 3. IANA has allocated the comprehension-optional codepoint TBD-CA for this attribute.

6. Security Considerations

Security considerations discussed in [RFC5389] are to be taken into account. STUN requires the 96 bits transaction ID to be uniformly and randomly chosen from the interval 0 .. 2**96-1, and be cryptographically strong. This is good enough security against an off-path attacker. An on-path attacker can either inject a fake response or modify the values in PATH-CHARACTERISTIC attribute to mislead the client and server, this attack can be mitigated using STUN authentication. As PATH-CHARACTERISTIC is expected to be used between peers using ICE, and ICE uses STUN short-term credential mechanism the risk of on-path attack influencing the messages is minimal. However, an attacker could corrupt, remove, or delay an ICE request or response, in order to discourage that path from being used. Unauthenticated STUN message MUST NOT include the PATH-CHARACTERISTIC attribute in order to prevent on-path attacker from influencing decision-making.

7. Acknowledgements

Thanks to Brandon Williams for valuable inputs and comments.

8. References

8.1. Normative References

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

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

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

8.2. Informative References

[ACM-MPRTP]
Singh, V., Ahsan, S., and J. Ott, "MPRTP: multipath considerations for real-time media", in Proc. of ACM Multimedia Systems, MMSys, 2013.

[I-D.ietf-avtcore-mprtp]
Singh, V., Karkkainen, T., Ott, J., Ahsan, S., and L. Eggert, "Multipath RTP (MPRTP)", draft-ietf-avtcore-mprtp-00 (work in progress), December 2014.

[I-D.patil-tram-turn-serv-selection]
Patil, P., Reddy, T., and G. Salgueiro, "Traversal Using Relays around NAT (TURN) Server Selection", draft-patil-tram-turn-serv-selection-00 (work in progress), October 2014.

Authors' Addresses

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Paal-Erik Martinsen
Cisco Systems, Inc.
Philip Pedersens vei 22
Lysaker, Akershus 1325
Norway

Email: palmarti@cisco.com

Varun Singh
Nemu Dialogue System Oy
Espoo 02235
Finland

Email: varun@callstats.io