Internet Engineering Task Force (IETF)
Request for Comments: 8016
Category: Standards Track
ISSN: 2070-1721

T. Reddy
Cisco
D. Wing
P. Patil
P. Martinsen
Cisco

November 2016

Mobility with Traversal Using Relays around NAT (TURN)

Abstract

It is desirable to minimize traffic disruption caused by changing IP address during a mobility event. One mechanism to minimize disruption is to expose a shorter network path to the mobility event so that only the local network elements are aware of the changed IP address and the remote peer is unaware of the changed IP address.

This document provides such an IP address mobility solution using Traversal Using Relays around NAT (TURN). This is achieved by allowing a client to retain an allocation on the TURN server when the IP address of the client changes.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc8016.

Table of Contents

1. Introduction

When moving between networks, the endpoint's IP address can change or, due to NAT, the endpoint's public IP address can change. Such a change of IP address breaks upper-layer protocols such as TCP and RTP. Various techniques exist to prevent this breakage, all tied to making the endpoint's IP address static (e.g., Mobile IP, Proxy Mobile IP, Locator/ID Separation Protocol (LISP)). Other techniques exist, which make the change in IP address agnostic to the upper-layer protocol (e.g., Stream Control Transmission Protocol (SCTP)). The mechanism described in this document is in that last category.

A server using Traversal Using Relays around NAT (TURN) [RFC5766] relays media packets and is used for a variety of purposes, including overcoming NAT and firewall traversal issues. The existing TURN specification does not permit a TURN client to reuse an allocation across client IP address changes. Due to this, when the IP address of the client changes, the TURN client has to request a new allocation, create permissions for the remote peer, create channels, etc. In addition, the client has to re-establish communication with its signaling server and send an updated offer to the remote peer conveying the newly relayed candidate address. Then, the remote side has to re-gather all candidates and signal them to the client, and the endpoints have to perform Interactive Connectivity Establishment (ICE) [RFC5245] checks. If the ICE continuous nomination procedure [NOMBIS] is used, then the newly relayed candidate address would have to be "trickled" (i.e., incrementally provisioned as described in [TRICKLE-SIP]), and ICE checks would have to be performed according to [TRICKLE-ICE] by the endpoints to nominate pairs for selection by ICE.

This specification describes a mechanism to seamlessly reuse allocations across client IP address changes without any of the hassles described above. A critical benefit of this technique is that the remote peer does not have to support mobility or deal with any of the address changes. The client, which is subject to IP address changes, does all the work. The mobility technique works across and between network types (e.g., between 3G and wired Internet access), so long as the client can still access the TURN server. The technique should also work seamlessly when (D)TLS is used as a transport protocol for Session Traversal Utilities for NAT (STUN) [RFC5389]. When there is a change in IP address, the client uses (D)TLS Session Resumption without Server-Side State as described in [RFC5077] to resume secure communication with the TURN server, using the changed client IP address.

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 document uses terminology defined in [RFC5245] and the following additional terminology:

Break Before Make: The old communication path is broken ("break") before new communication can be created ("make"). Such changes typically occur because a network's physical cable is disconnected, radio transmission is turned off, or a client moves out of radio range.

Make Before Break: A new communication path is created ("make") before the old communication path is broken ("break"). Such changes typically occur because a network is reconnected with a physical cable, radio transmission is turned on, or a client moves into radio range.

3. Mobility Using TURN

To achieve mobility, a TURN client should be able to retain an allocation on the TURN server across changes in the client IP address as a consequence of movement to other networks.

When the client sends the initial Allocate request to the TURN server, it will include a new STUN attribute MOBILITY-TICKET (with zero length value), which indicates that the client is capable of mobility and desires a ticket. The TURN server provisions a ticket that is sent inside the new STUN attribute MOBILITY-TICKET in the Allocate success response to the client. The ticket will be used by the client when it wants to refresh the allocation but with a new client IP address and port. This ensures that an allocation can only be refreshed by the same client that allocated the relayed transport address. When a client's IP address changes due to mobility, it presents the previously obtained ticket in a Refresh request to the TURN server. If the ticket is found to be valid, the TURN server will retain the same relayed address/port for the new IP address/port allowing the client to continue using previous channel bindings -- thus, the TURN client does not need to obtain new channel bindings. Any data from the external peer will be delivered by the TURN server to this new IP address/port of the client. The TURN client will continue to send application data to its peers using the previously allocated channelBind Requests.

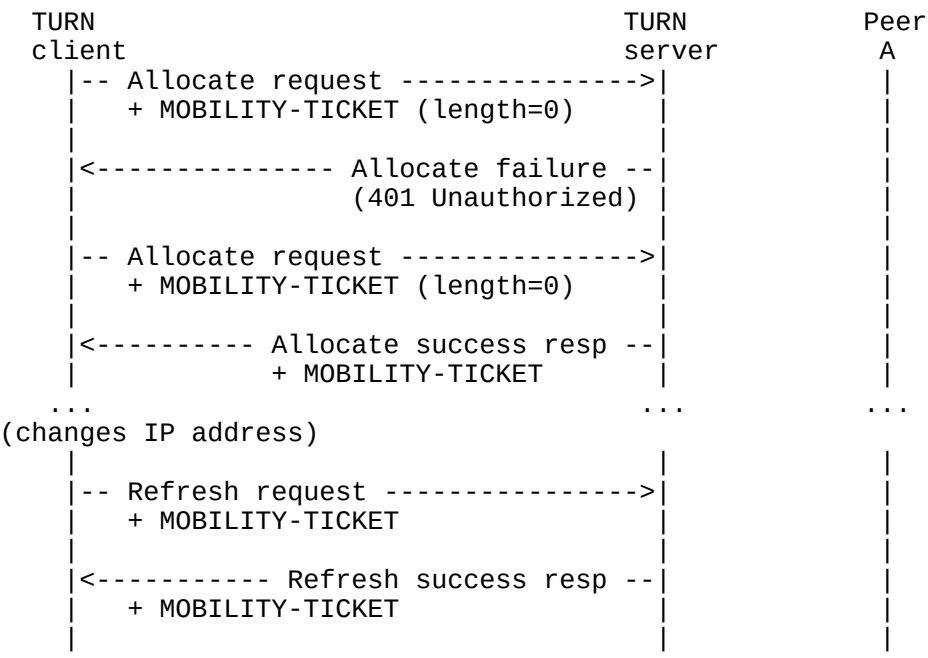

Figure 1: Mobility Using TURN

In Figure 1, the client sends an Allocate request with a MOBILITY-TICKET attribute to the server without credentials. Since the server requires that all requests be authenticated using STUN's long-term credential mechanism, the server rejects the request with a 401 (Unauthorized) error code. The client then tries again, this time including credentials (not shown). This time, the server accepts the Allocate request and returns an Allocate success response and a ticket inside the MOBILITY-TICKET attribute. Sometime later, the client IP address changes, and the client decides to refresh the allocation, and thus sends a Refresh request to the server with a MOBILITY-TICKET attribute containing the ticket it received from the server. The refresh is accepted, and the server replies with a Refresh success response and a new ticket inside the MOBILITY-TICKET attribute.

3.1. Creating an Allocation

3.1.1. Sending an Allocate Request

In addition to the process described in Section 6.1 of [RFC5766], the client includes the MOBILITY-TICKET attribute with a length of zero. This indicates that the client is a mobile node and wants a ticket.

3.1.2. Receiving an Allocate Request

In addition to the process described in Section 6.2 of [RFC5766], the server does the following:

If the MOBILITY-TICKET attribute is included, and has a length of zero, but TURN session mobility is forbidden by local policy, the server will reject the request with the new error code 405 (Mobility Forbidden). If the MOBILITY-TICKET attribute is included and has a non-zero length, then the server will generate an error response with an error code of 400 (Bad Request). Following the rules specified in [RFC5389], if the server does not understand the MOBILITY-TICKET attribute, it ignores the attribute.

If the server can successfully process the request and create an allocation, the server replies with a success response that includes a STUN MOBILITY-TICKET attribute. The TURN server can store system-internal data in the ticket that is encrypted by a key known only to the TURN server and sends the ticket in the STUN MOBILITY-TICKET attribute as part of the Allocate success response. An example of ticket construction is discussed in Appendix A. The ticket is opaque to the client, so the structure is not subject to interoperability concerns, and implementations may diverge from this format. The client could be roaming across networks with a different path MTU and from one address family to another (e.g., IPv6 to IPv4). The TURN server to support mobility must assume that the path MTU is unknown and use a ticket length in accordance with the published guidance on STUN UDP fragmentation (Section 7.1 of [RFC5389]).

Note: There is no guarantee that the fields in the ticket are going to be decodable to a client, and therefore attempts by a client to examine the ticket are unlikely to be useful.

3.1.3. Receiving an Allocate Success Response

In addition to the process described in Section 6.3 of [RFC5766], the client will store the MOBILITY-TICKET attribute, if present, from the response. This attribute will be presented by the client to the server during a subsequent Refresh request to aid mobility.

3.1.4. Receiving an Allocate Error Response

If the client receives an Allocate error response with error code 405 (Mobility Forbidden), the error is processed as follows:

> 405 (Mobility Forbidden): The request is valid, but the server is refusing to perform it, likely due to administrative restrictions. The client considers the current transaction as having failed.
>
> The client can notify the user or operator. The client SHOULD NOT retry sending the Allocate request containing the MOBILITY-TICKET with this server until it believes the problem has been fixed.

All other error responses must be handled as described in [RFC5766].

3.2. Refreshing an Allocation

3.2.1. Sending a Refresh Request

If a client wants to refresh an existing allocation and update its time-to-expiry or delete an existing allocation, it sends a Refresh request as described in Section 7.1 of [RFC5766]. If the client's IP address or source port has changed and the client wants to retain the existing allocation, the client includes the MOBILITY-TICKET attribute received in the Allocate success response in the Refresh request. If there has been no IP address or source port number change, the client MUST NOT include a MOBILITY-TICKET attribute, as this would be rejected by the server and the client would need to retransmit the Refresh request without the MOBILITY-TICKET attribute.

3.2.2. Receiving a Refresh Request

In addition to the process described in Section 7.2 of [RFC5766], the server does the following:

If the STUN MOBILITY-TICKET attribute is included in the Refresh request, and the server configuration changed to forbid mobility or the server transparently fails over to another server instance that forbids mobility, then the server rejects the Refresh request with a 405 (Mobility Forbidden) error and the client starts afresh with a new allocation.

If the STUN MOBILITY-TICKET attribute is included in the Refresh request, then the server will not retrieve the 5-tuple from the packet to identify an associated allocation. Instead, the TURN server will decrypt the received ticket, verify the ticket's validity, and retrieve the 5-tuple allocation using the ticket. If this 5-tuple obtained does not identify an existing allocation, then

the server MUST reject the request with a 437 (Allocation Mismatch) error. If the ticket is invalid, then the server MUST reject the request with a 400 (Bad Request) error.

If the source IP address and port of the Refresh request with the STUN MOBILITY-TICKET attribute is the same as the stored 5-tuple allocation, then the TURN server rejects the request with a 400 (Bad Request) error. If the source IP address and port of the Refresh request is different from the stored 5-tuple allocation, the TURN server proceeds with a MESSAGE-INTEGRITY validation to identify that it is the same user that had previously created the TURN allocation. If the above check is not successful, then the server MUST reject the request with a 441 (Wrong Credentials) error.

If all of the above checks pass, the TURN server understands that the client either has moved to a new network and acquired a new IP address (Break Before Make) or is in the process of switching to a new interface (Make Before Break). The source IP address of the request could be either the host transport address or the server-reflexive transport address. The server then updates its state data with the new client IP address and port but does not discard the old 5-tuple from its state data. The TURN server calculates the ticket with the new 5-tuple and sends the new ticket in the STUN MOBILITY-TICKET attribute as part of Refresh success response. The new ticket sent in the refresh response MUST be different from the old ticket.

The TURN server MUST continue receiving and processing data on the old 5-tuple and MUST continue transmitting data on the old-5 tuple until it receives a Send Indication or ChannelData message from the client on the new 5-tuple or a message from the client to close the old connection (e.g., a TLS fatal alert or TCP RST). After receiving any of those messages, a TURN server discards the old ticket and the old 5-tuple associated with the old ticket from its state data. Data sent by the client to the peer is accepted on the new 5-tuple and data received from the peer is forwarded to the new 5-tuple. If the refresh request containing the MOBILITY-TICKET attribute does not succeed (e.g., the packet is lost if the request is sent over UDP, or the server is unable to fulfill the request), then the client can continue to exchange data on the old 5-tuple until it receives the Refresh success response.

The old ticket can only be used for the purposes of retransmission. If the client wants to refresh its allocation with a new server-reflexive transport address, it MUST use the new ticket. If the TURN server has not received a Refresh request with the STUN MOBILITY-TICKET attribute but receives Send indications or ChannelData messages from a client, the TURN server MAY discard or queue those Send indications or ChannelData messages (at its discretion). Thus,

it is RECOMMENDED that the client avoid transmitting a Send indication or ChannelData message until it has received an acknowledgement for the Refresh request with the STUN MOBILITY-TICKET attribute.

To accommodate the potential loss of Refresh responses, a server must retain the old STUN MOBILITY-TICKET attribute for a period of at least 30 seconds to be able to recognize a retransmission of the Refresh request with the old STUN MOBILITY-TICKET attribute from the client.

3.2.3. Receiving a Refresh Response

In addition to the process described in Section 7.3 of [RFC5766], the client will store the MOBILITY-TICKET attribute, if present, from the response. This attribute will be presented by the client to the server during a subsequent Refresh request to aid mobility.

3.3. New STUN Attribute MOBILITY-TICKET

This attribute is used to retain an allocation on the TURN server. It is exchanged between the client and server to aid mobility. The value of the MOBILITY-TICKET is encrypted and is of variable length.

3.4. New STUN Error Response Code

This document defines the following new error response code:

> 405 (Mobility Forbidden): Mobility request was valid but cannot be performed due to administrative or similar restrictions.

4. IANA Considerations

IANA has added the following attribute to the "STUN Attributes" registry [IANA-STUN]:

- MOBILITY-TICKET (0x8030, in the comprehension-optional range)

Also, IANA has added a new STUN error code "Mobility Forbidden" with the value 405 to the "STUN Error Codes" registry [IANA-STUN].

5. Security Considerations

The TURN server MUST always ensure that the ticket is authenticated and encrypted using strong cryptographic algorithms to prevent modification or eavesdropping by an attacker. The ticket MUST be constructed such that it has strong entropy to ensure that nothing can be gleaned by looking at the ticket alone.

An attacker monitoring the traffic between the TURN client and server can impersonate the client and refresh the allocation using the ticket issued to the client with the attacker's IP address and port. The TURN client and server MUST use the STUN long-term credential mechanism [RFC5389], the STUN Extension for Third-Party Authorization [RFC7635], or a (D)TLS connection to prevent malicious users from impersonating the client. With any of those three mechanisms, when the server receives the Refresh request with the STUN MOBILITY-TICKET attribute from the client, it identifies that it is indeed the same client but with a new IP address and port using the ticket it had previously issued to refresh the allocation. If (D)TLS is not used or the (D)TLS handshake fails, and authentication also fails, then the TURN client and server MUST fail and not proceed with TURN mobility.

Security considerations described in [RFC5766] are also applicable to this mechanism.

6. References

6.1. Normative References

[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>.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, DOI 10.17487/RFC5077, January 2008, <http://www.rfc-editor.org/info/rfc5077>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <http://www.rfc-editor.org/info/rfc5389>.

[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, <http://www.rfc-editor.org/info/rfc5766>.

6.2. Informative References

[IANA-STUN]
IANA, "Session Traversal Utilities for NAT (STUN) Parameters", <http://www.iana.org/assignments/stun-parameters>.

[NOMBIS] Uberti, J. and J. Lennox, "Improvements to ICE Candidate Nomination", Work in Progress, draft-uberti-mmusic-nombis-00, March 2015.

[RFC7635] Reddy, T., Patil, P., Ravindranath, R., and J. Uberti, "Session Traversal Utilities for NAT (STUN) Extension for Third-Party Authorization", RFC 7635, DOI 10.17487/RFC7635, August 2015, <http://www.rfc-editor.org/info/rfc7635>.

[TRICKLE-ICE]
Ivov, E., Rescorla, E., Uberti, J., and P. Saint-Andre, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", Work in Progress, draft-ietf-ice-trickle-04, September 2016.

[TRICKLE-SIP]
Ivov, E., Stach, T., Marocco, E., and C. Holmberg, "A Session Initiation Protocol (SIP) usage for Trickle ICE", Work in Progress, draft-ietf-mmusic-trickle-ice-sip-06, October 2016.

Appendix A. Example of Ticket Construction

The TURN server uses two different keys: one 128-bit key for Advance Encryption Standard (AES) in Cipher Block Chaining (CBC) mode (AES_128_CBC) and a 256-bit key for HMAC-SHA-256-128 for integrity protection. The ticket can be structured as follows:

```
struct {
    opaque key_name[16];
    opaque iv[16];
    opaque encrypted_state<0..2^16-1>;
    opaque mac[16];
} ticket;
```

Figure 2: Ticket Format

Here, key_name serves to identify a particular set of keys used to protect the ticket. It enables the TURN server to easily recognize tickets it has issued. The key_name should be randomly generated to avoid collisions between servers. One possibility is to generate new random keys and key_name every time the server is started.

The TURN state information (which is either self-contained or a handle) in encrypted_state is encrypted using 128-bit AES in CBC mode with the given Initialization Vector (IV). The Message Authentication Code (MAC) is calculated using HMAC-SHA-256-128 over key_name (16 octets) and IV (16 octets), followed by the length of the encrypted_state field (2 octets) and its contents (variable length).

Acknowledgements

Thanks to Alfred Heggestad, Lishitao, Sujing Zhou, Martin Thomson, Emil Ivov, Oleg Moskalenko, Dave Waltermire, Pete Resnick, Antoni Przygienda, Alissa Cooper, Ben Campbell, Suresh Krishnan, Mirja Kuehlewind, Jonathan Lennox, and Brandon Williams for review and comments.

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

Email: dwing-ietf@fuggles.com

Prashanth Patil
Cisco Systems, Inc.
Bangalore
India

Email: praspati@cisco.com

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

Email: palmarti@cisco.com