TCPM Working Group
Internet-Draft
Intended status: Informational
Expires: January 17, 2018

B. Pithawala
U. Chunduri, Ed.
Huawei Technologies
July 16, 2017

TCP Mobility Option with Identifiers
draft-pc-tcpm-tcp-seamless-mobility-option-00

Abstract

This document specifies the TCP Seamless Mobility Option (TCP-SMO) with variable length identifiers. TCP-SMO allows mobility with session continuity, when either of the TCP endpoint change its IP address. This is being done securely and without any additional round trips during mobility event.

Requirements Language

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 [RFC2119].

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 January 17, 2018.

Copyright Notice

Table of Contents

1. Introduction

When a TCP [RFC0793] connection is opened with a peer, it (TCP Session or Socket) is identified by Source port, Destination port, Source IP address and Destination IP address by both initiator and responder of the connection. This connection is identified by the socket parameters and stored in Transmission Control Block (TCB) at TCP stack. Any of these connection identifier changes at one end cause either TCP RST by peer or the session times out eventually.

The above is per [RFC0793], Section 2.7- "To identify the separate data streams that a TCP may handle, the TCP provides a port identifier. Since port identifiers are selected independently by each TCP they might not be unique. To provide for unique addresses within each TCP, we concatenate an internet address identifying the

TCP with a port identifier to create a socket which will be unique throughout all networks connected together."

There are more mobile devices today than a decade earlier. Hence it is imperative that we have a mechanism to provide Session or Transport layer connectivity that is impervious to network layer changes. Some of the Identifier/Location protocols which has inherent support for seamless mobility and relationship to this work is discussed in Section 8.

This document provides an option towards providing resilience in the TCP transport layer as either end of the connection moves and changes its location (and thereby its IP address). Some of the previous efforts with TCP mobility have been discussed in Section 9

2. Acronyms

DH - Diffie-Hellman Key Agreement

HIP - Host Identity Protocol

LISP - The Locator/ID Separation Protocol

TCP - Transmission Control Protocol

3. TCP Seamless Mobility Option with Identifiers

For seamless mobility with an uninterrupted TCP connection, this document proposes a TCP Seamless Mobility Option (TCP-SMO) as described in Figure 1 . This new TCP option introduces connection identifiers and proposes changes to TCP to use these identifiers for TCB or connection identification.

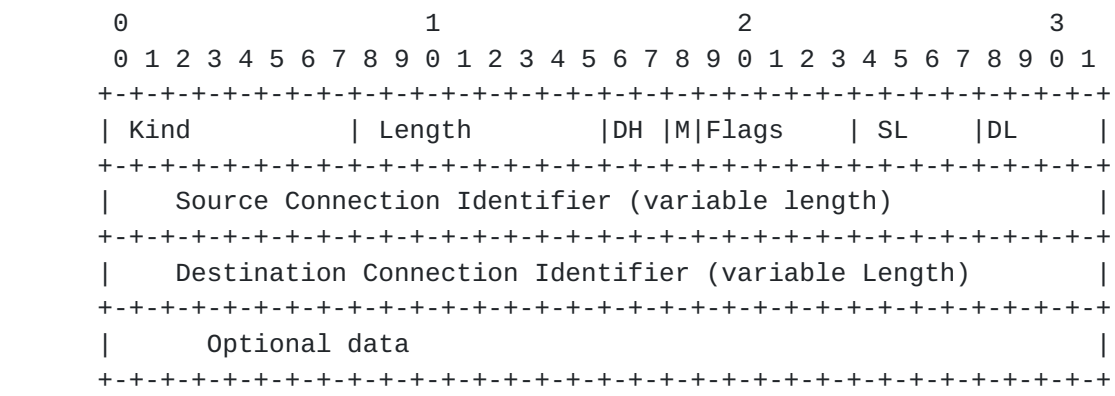

Figure 1: TCP Seamless Mobility Option (TCP-SMO)

Kind - TBD (TCP IANA).

Length - Variable. Includes Kind and Length field in bytes. Minimum value is 12 and a maximum value is 40 bytes. When the Length is less than 12 TCP MUST discard the segment.

Flags

'D H' - (2 bits) When its non-zero Diffie-Helman value is being exchanged

D H Bits in Flags

0 0 NO Diffie-Hellman values exchanged

0 1 1024-bit ModP Group (mandatory)

1 0 2048-bit ModP Group/elliptic curve groups which has 200bits ModP

1 1 Reserved

Two tiers of identification are needed, with identifiers anchored at the identity.

'M' - mobility event, optional data (4 bytes) contains the keyed hash of the Source Connection Identifier concatenated with a fixed 12 byte String 0xFEFEFEFEABABABABDEDEDEDE. The fixed string will help normalized the total data for the hash function as connection identifiers length can be as low as 4 bytes.

In Case of Security Option-1 - the hash function is a simple SHA1-96 hash of the above concatenated data. Optional 4 byte data is the checksum of this hash data.

In Case of Security Option-2 Hash function MUST use AES-128-CMAC-6 [NIST-SP800-38B][FIPS197] and Optional 4 byte data is the checksum of this "keyed" hash data.

Other flags are reserved and undefined.

SL - Source CID Length (4 bits) - Length of the source identifier in bytes. Minimum is 4 bytes and a maximum of 16 bytes.

DL - Source CID Length (4 bits) - Length of the destination identifier in bytes. Minimum is 4 bytes and a maximum of 16 bytes.

SCID: Source Connection identifier - Length minimum of 4 bytes as indicated by CID length. It can be generated by TCP module or can be supplied by application.

DCID: Destination Connection identifier - Length minimum of 4 bytes as indicated by CID length. It can be generated by TCP module or can be supplied by application.

Optional data (4 Bytes) - MUST be present only when M bit is set and MUST contain the 4 bytes of Checksum of Generated hash of the Source Connection Identifier + Fixed Data.

The Checksum and Hash is generated on the side of communication which detected an IP address change and needs to provide seamless mobility.

Source CID (SCID) and Destination CID (DCID) can be provided to TCP by an application or other protocol (LISP, HIP or any other ID related protocol with in the constrains of this option) while opening the TCP socket or TCP stack can generate unique pair of CIDs to be used in this option.

4. Elements of Procedure

4.1. Flow of TCP Connection with SMO And IP Address Change

Consider IP address of Source is IP1 and Destination is IP2. TCP-SMO can be explained with a sequence of TCP segments as specified below:

1. Source sends TCP-SYN message with TCP-SMO. In the Seamless Mobility Option: If CID's (source and destination) are chosen by application and given to TCP then both can be present in the TCP-SYN segment. If TCP has to choose the CID's, source chooses the unique

(local to the TCP stack) connection Identifier and keeps the destination connection identifier as 0x0 with minimum length 4 bytes in the TCP-SYN segment. When destination receives this segment, it allocates another unique identifier and this value would be kept in the SYN-ACK segment sent to the source.

2. Destination receives the SYN segment and it supports this option. It responds back with SYN-ACK with the mobility options. If destination CID is 0x0, as specified above it Chooses a unique CID.

From this point TCB is looked-up by Source CID, Destination CID instead of [RFC0793] Section 2.7.

3. Source sends Sync-Ack-Ack segment and at source side TCB is established with look up parameters by Source CID, Destination CID, Source Port and Destination port instead of [RFC0793] Section 2.7.

4. TCP data segments exchanged from both Source and destination and these segments will have the TCP-SMO.

5. In the middle of the connection, a mobility event happened at the source and IP address is changed from IP1 to IP11. When there is data to be sent immediately by source, data segment will be sent with the new Source IP address IP11 and TCB at both ends update the change in the Source IP address. If there is no data segment to be sent by source during mobility event either a TCP keep alive segment or empty data packet needs to be sent with this new IP to allow TCB's at both end to know the new Source IP address.

The security implications of continuing to use the same session after an IP address change are discussed in Section 4.2.

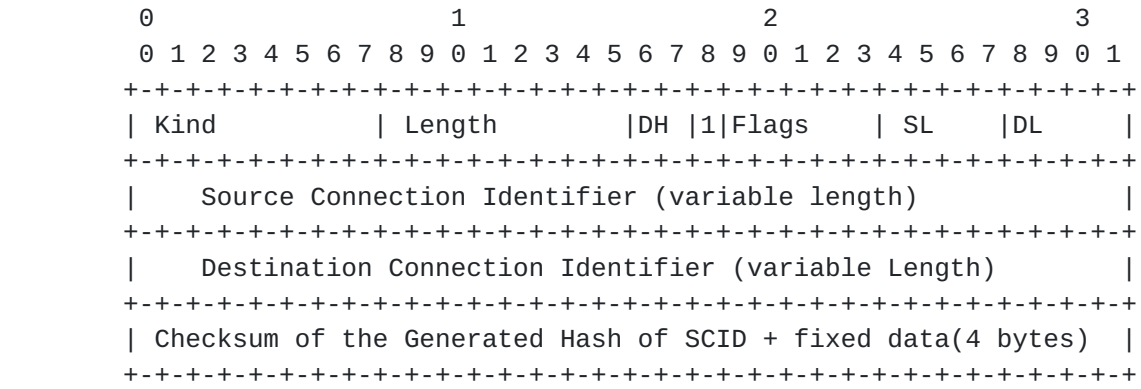

Figure 2: TCP Seamless Mobility Option Sent at IP Change Event

6. Destination receives TCP-SMO segment. Verifies that IP1 sent the data for change to IP11 thru the Generated Hash, updates its TCB for

IP1 connections with Mobility flag set and starts to receive packets on source IP11 TCP segment sent by destination with new Source IP address. If Destination cannot verify then it terminates the session, which is the same as existing behavior.

7. Data sent by source but a mobility event happens at destination before this event can be updated to source. In this case the data is lost and would be retransmitted by source when changed IP address is notified by destination.

8. Either end can terminate the session by sending the TCP-FIN sequence which also uses TCP-SMO regardless who initiates the socket close.

4.2. Security mechanisms during IP address change

4.2.1. Security Option-1

The TCP-SMO discussed does introduce a security issue as anybody can hijack the existing TCP connection by changing the source IP address (Connection Hijacking). It is recommended to use security at IP (IPsec) or transport layer-Authentication Option (TCP-AO) [RFC5925] for securing the TCP connection including the TCP header. When [RFC5925] is used TCP-SMO size MUST not be exceeded more than 12 bytes to accommodate TCP-AO [RFC5925].

Implication on Connection Identifiers:

With the TCP AO option for authentication of an IP Address change, the Source and Destination connection Identifiers have to be fixed at 4 bytes each for a valid TCP-SMO.

4.2.2. Security Option-2

In this option, security for TCP-SMO is done in-line with in the newly defined option by exchanging a DH public value and computing a shared secret at both ends, which can be used as a key to generate a hash value (pre-defined has algorithm) during mobility. DH groups and the hash algorithms are pre-defined and MUST be implemented to use this mechanism to avail the needed security.

This option relies on the "data" to be sent during 3 way handshake. This data is not application data but the ModP (DH public value) for selected DH group. As the size of this is anywhere from 1024 bits (DH Group-1) to 2048 bits (DH Group-2)/200 bits (Elliptic curve DH groups) sending this as part of option data is not possible (limited option space in the TCP header).

Section 3.4 of [RFC0793] specifies and allows a mechanism to send application data during TCP 3 way hand shake but this "data" should not be sent to application until completion of handshake. The data we send is not application data and will be consumed by TCP end host stack until handshake completion.

The Sequence of data packet exchange:

1. The TCP-SMO DH flags will have non-zero value and particular DH group used by source to generate the ModP value. This is sent as part of the data in the TCP SYN-Segment.

2. When destination receives this data and generates the ModP value of the destination and sends back in SYN-ACK segment data portion. At this point destination can compute the DH shared secret (would be used during mobility as a symmetric key for authentication information).

3. SYN-ACK-ACK segment is sent and also source computes the DH shared secret.

4. TCP data segments exchanged from both side with TCP-SMO.

5. Before this message mobility event happens at source and IP1 changes to IP11.

In this case source sends the data segment with TCP-SMO 'M' bit set with the 4 byte optional data, which is the checksum of the hash computed using the DH shared secret with data being the new IP address.

Optional-data = Checksum (Hash((Source Connection Identifier + Fixed data normalizer), DH-Shared-Secret)).

Fixed data Normalizer: 0xFEFEFEFEABABABABDEDEDEDE

Length of optional data: first 4 bytes of the checksum.

Hash algorithm to be used is AES-128-CMAC-96.

6. When the above segment arrives with M bit set, destination computes the checksum of the hash for the new IP address received and compares the same with the received value of "optional data" in the TCP-mobility-option. If this matches, it ensures the mobility event is received indeed from Source and IP11 indicated is correct.

It sends the TCP data segment to the newly received IP address from Source. This time M bit will be cleared (only used for mobility).

4.3. Notifying IP Address Change Event

An IP address change in the local stack MUST be notified to the local TCP layer. This is to ensure that the local TCP layer can adjust to IP change and to indicate this change to the destination. There are 2 possibilities of change:

Delete: If any IP address (connected route) is deleted, TCP should be notified of this change and TCP should find another source IP address for this existing connection only if sockets that are bound to the deleted IP address had their Mobility Flag ON.

Modify: If IP address is changes, then both the old and new address should be updated to local TCP to identify the old connection and also to enable TCP to find new source IP for that connection (not necessarily the changed IP, but based on the route lookup on the destination IP).

Once the above is done to intimate the destination about this change there are 3 possible option and one of the option has to be fulfilled by local TCP.

Pending data from application: in this case TCB should use the new source IP address and send the segment with application data.

Sending a keep-alive: If keep-alive are enabled on the TCP connection and asynchronous keep-alive can be sent to the destination with the new IP address.

Sending zero-data segment: if the above 2 are not available, local TCP stack can send a zero-data segment to the destination with the new source IP address.

In all 3 cases, the packet is sent with the TCP-SMO as in Section 4 step 6.

4.4. Buffering the In-flight data

During IP address change local host where the change happened can have a handshake mechanism with TCP stack and during that time received data can be buffered. Though this is an implementation detail and it can be done many ways and this is out of scope for this document.

4.5. TCP-SMO with Unsupported Destination

If a destination does not support the TCP-SMO then it MUST send a regular SYN-ACK as per [RFC0793]. If SYN-ACK is received without any TCP-SMO, source falls back to sending SYN segment without any option. At this point this will be a regular TCP connection and seamless mobility is not possible for this connection.

5. Unsupported Middleboxes

Any new TCP options are prone to be dropped by middle boxes and this is generally applies to 5% of TCP connections per [RFC7413] section 7.1. To cater this case, after the initial time out of the SYN with TCP-SMO, source MUST send SYN segment without this option, thus resorting to native TCBs and hence not having seamless mobility.

6. Impact of NAT between Source and Destination

This solution does not have any impact of NAT/NAPT devices between source and destination because:

- TCB uses only Source and destination connection identifiers
- During mobility the authentication data uses stable Source identifier to verify the hash on both sides (does not include new IP address or port).

So even in the face NAT/NAPT device with M bit set peer can securely authenticate the change of IP address.

7. Changes to host TCP Stack and Backward Compatibility

This draft suggests changes to TCP host stack on both ends of the connection. Main aspect of the change includes how TCB (Transmission Control Block) as defined in TCP [RFC0793] Section 2.7 can be looked up and subsequently used. TCB is created when OPEN call is done at the host stack and it is a data structure which holds the state of the connection. During connection OPEN time a flag can be indicated which suggests to use TCP-SMO with additional parameters as required.

It is important to note, if the TCP option proposed in this document is not present, TCP connection should continue to identify TCB with connection identifiers as specified in TCP [RFC0793]. So, implementations supporting this option will have both old way of identifying the connection/TCB per [RFC0793] and with connection identifiers in the TCP-SMO. How implementation can be done to have both these methods is beyond the scope of the document, while one

could easily perceive my maintaining two separate TCBs or harmonizing the keys to access the existing TCB with the proposal here.

When a device or User Equipment (UE) connects to a TCP server, in majority of the cases TCP server will not change its location but to support device or UE mobility server TCP stack also MUST be updated to support this option. One way to mitigate this update at server side is by using a TCP server proxy where this option is implemented.

8. Relationship with other Identifier Protocols

Proposal made in this document can be seen as a potential Data Plane alternative to existing Location/Identifier based protocols LISP [RFC6830] or HIP [RFC7401] with respective protocols control plane. This option doesn't specify any changes to control plane for existing identifier based protocols. However, existing Location/Identifier based protocols will have some restrictions to use the data plane proposed in this document for example, limitation on the length of Identifier in TCP-SMO, which is variable and a maximum of 16 bytes.

The mechanism proposed in this document also can be used with new Identity based common control plane protocol as specified in [I-D.padma-ideas-problem-statement] and this provides a data plane solution for applications using TCP transport. UDP based applications can use mobility solution with QUIC protocol [I-D.tsvwg-quic-protocol], which allows similar functionality with respect to mobility.

9. Previous and Related Efforts

There were earlier efforts for TCP Mobility and one of those is described in [I-D.eddy-tcp-mobility]. Two of the main differences in this proposal is, how mobility event is notified to the peer (here no-3-way-handshake) and the security mechanisms during that event part from others.

Multipath TCP (MPTCP) [RFC6182] [RFC6824] was originally invented to distribute TCP load across multiple TCP connections but later mobility has been introduced by deleting the old connection. The proposal in this draft can be seen as more simpler than the solution proposed with MPTCP. Some of the other aspects, this proposal can be seen as different from MPTCP are in the area of performance [I-D.khalili-mptcp-performance-issues] and the host's ability to have insight into network topology in order to create multiple paths in some cases.

10. Acknowledgements

TBD.

11. IANA Considerations

As specified in Section 3, this document requests new value for 'kind' assignment from TCP IANA registry.

12. Security Considerations

Mechanisms to protect IP address change event is covered in Section 4.2. Further security concerns TBD.

13. References

13.1. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[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>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <http://www.rfc-editor.org/info/rfc5925>.

13.2. Informative References

[I-D.eddy-tcp-mobility] Eddy, W., "Mobility Support For TCP", draft-eddy-tcp-mobility-00 (work in progress), April 2004.

[I-D.khalili-mptcp-performance-issues] Khalili, R., Gast, N., Popovic, M., and J. Boudec, "Performance Issues with MPTCP", draft-khalili-mptcp-performance-issues-06 (work in progress), July 2014.

[I-D.padma-ideas-problem-statement] Pillay-Esnault, P., Boucadair, M., Fioccola, G., Jacquenet, C., and A. Nennker, "Problem Statement for Identity Enabled Networks", draft-padma-ideas-problem-statement-03 (work in progress), July 2017.

[I-D.tsvwg-quic-protocol]
Hamilton, R., Iyengar, J., Swett, I., and A. Wilk, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2", draft-tsvwg-quic-protocol-02 (work in progress), January 2016.

[RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, DOI 10.17487/RFC2631, June 1999, <http://www.rfc-editor.org/info/rfc2631>.

[RFC6182] Ford, A., Raiciu, C., Handley, M., Barre, S., and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", RFC 6182, DOI 10.17487/RFC6182, March 2011, <http://www.rfc-editor.org/info/rfc6182>.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <http://www.rfc-editor.org/info/rfc6824>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <http://www.rfc-editor.org/info/rfc7401>.

[RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <http://www.rfc-editor.org/info/rfc7413>.

Authors' Addresses

Burjiz Pithawala
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: burjiz.pithawala1@huawei.com

Uma Chunduri (editor)
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: uma.chunduri@huawei.com