TCP Maintenance and Minor Extensions
Internet-Draft
Intended status: Experimental
Expires: 8 September 2022
M. Amend
DT
J. Kang
Huawei
7 March 2022

Multipath TCP Extension for Robust Session Establishment
draft-amend-tcpm-mptcp-robe-02

Abstract

Multipath TCP extends the plain, single-path limited, TCP towards the capability of multipath transmission. This greatly improves the reliability and performance of TCP communication. For backwards compatibility reasons the Multipath TCP was designed to setup successfully an initial path first, after which subsequent paths can be added for multipath transmission. For that reason the Multipath TCP has the same limitations as the plain TCP during connection setup, in case the selected path is not functional.

This document proposes a set of implementations and possible combinations thereof, that provide a more Robust Establishment (RobE) of MPTCP sessions. It includes RobE_TIMER, RobE_SIM, RobE_eSIM and RobE_IPS.

RobE_TIMER is designed to stay close to MPTCP in that standard functionality is used wherever possible. Resiliency against network outages is achieved by modifying the SYN retransmission timer: If one path is defective, another path is used.

RobE_SIM and RobE_eSIM provides the ability to simultaneously use multiple paths for connection setup. They ensure connectivity if at least one functional path out of a bunch of paths is given and offers beside that the opportunity to significantly improve loading times of Internet services.

RobE_IPS provides a heuristic to select properly an initial path for connection establishment with a remote host based on empirical data derived from previous connection information.

In practice, these independent solutions can be complementary used. This document also presents the design and protocol procedure for those combinations in addition to the respective stand-alone solutions.

Status of This Memo

This Internet-Draft will expire on 8 September 2022.

Copyright Notice

Table of Contents

1. Introduction

Multipath TCP Robust Session Establishment (MPTCP RobE) is a set of extensions to regular MPTCP [RFC6824] and its next version [RFC8684], which releases single path limitations during the initial connection setup. Several scenarios require and benefit from a reliable and in time connection setup which is not covered by [RFC6824] and [RFC8684] so far. MPTCP was designed to be compliant with the TCP standard [RFC0793] and introduced therefore the concept of an initial TCP flow while adding subsequent flows after successful multipath negotiation on the initial path. While fulfilling its purpose, MPTCP is however fully dependent on the transmission characteristics of the communication link selected for initiating MPTCP.

Figure 1 shows the traditional way of MPTCP handshaking with an MP_CAPABLE exchanged first, followed when successfully negotiated by additional flows engaging MP_JOIN. [RFC6824] and the next MPTCP [RFC8684] differ in that a Key-A is sent with the first MP_CAPABLE or not.


```
               Host A                                  Host B
        ------------------------                       ----------
        Address A1    Address A2                       Address B1
        ----------    ----------                       ----------
            |             |                                |
            |            SYN + MP_CAPABLE(Key-A[*])        |
            |---------------------------------------------->|
            |<----------------------------------------------|
            |          SYN/ACK + MP_CAPABLE(Key-B)         |
            |             |                                |
            |        ACK + MP_CAPABLE(Key-A, Key-B)        |
            |---------------------------------------------->|
            |             |                                |
            |             |   SYN + MP_JOIN(Token-B, R-A)  |
            |             |------------------------------->|
            |             |<-------------------------------|
            |             | SYN/ACK + MP_JOIN(HMAC-B, R-B) |
            |             |                                |
            |             |     ACK + MP_JOIN(HMAC-A)      |
            |             |------------------------------->|
            |             |<-------------------------------|
            |             |             ACK                |
```

[*] Key-A in the first MP-capable is related to RFC6824 only and does not exist in RFC8684.

Figure 1: MPTCP connection setup

Multipath TCP itself enables hosts to exchange packets belonging to a single connection over several paths. Implemented in mobile phones (UEs), these paths are usually assigned to different network interfaces within the UE and correspond to different access networks such as cellular and WiFi. The path or network interface for initiating the initial subflow setup is most often provided by the operation system of the UE. For example, if both a cellular connection and WiFi are present in a mobile phone, WiFi is usually the interface offered to initiate the MPTCP session.

This design falls short in situations where the default path does not provide the best performance compared to other available paths. In a worst case the default path is not even capable of setting up the initial flow letting any other functional path unused. For example, if the WiFi signal is weak, broken or cannot forward traffic to the destination, the establishment of the subflow will be delayed or impossible. This in turn, leads to a longer startup delay or no communication at all for services using MPTCP even if other functional paths are available. Even in scenarios where all paths are functional but services would benefit from a setup over the path with the lowest latency, MPTCP has no mean to support this demand.

It can be concluded, that sequential path establishment relying with an initial path establishment over an externally given default route will result in experience reduction when using MPTCP. So this document proposes solutions to overcome the aforementioned limitations and provides a more robust connection setup compared to traditional MPTCP.

Introduction of RobE_SIM and RobE_eSIM aims to overcome the limitations of [RFC6824] and [RFC8684], using one initial flow and introduces the concept of multiple potential initial flows triggered simultaneously.Potential initial flows give the freedom to use more than one path to request multipath capability and select the initial flow at a later point. Potential initial flow mechanisms and the gain of robustness and performance over the traditional MPTCP connection setup are evaluated in [RobE_slides] and [RobE_paper]. RobE_SIM is a break-before-make mechanism, guaranteeing at least the robust connection establishment, however the RobE_eSIM reuses every potential initial flow request to combine it with less overhead and accelerated multipath availability, leveraging a new MPTCP option MP_JOIN_CAP. From a standardization perspective, the RobE_SIM is fully compliant with [RFC6824] and [RFC8684] and is herein more of a descriptive and procedural nature. The RobE_eSIM requires a new MPTCP option but offers the potential to significantly improve the MPTCP experience.

For the limitation of the default initial path, RobE_IPS makes no changes to standard MPTCP procedure and improves the performance of connection establishment by introducing an initial path selection strategy and required algorithms. The input for strategy and algorithms is the transmission status information which represents the transmission performance of each available path or network interface. The transmission status information is characterized by at least one of the parameters: signal strength, throughput, round-trip time (RTT), and link success rate. In this way, a path with better transmission performance can be learned and determined and the respective network interface can be used for connection establishment.

The most simple approach for a robust MPTCP session establishment is RobE_TIMER, iterating the process of initial path establishment over all available paths, if the previous try has failed. Triggering a new try on a next path is depending on an expiration timer, preferably re-use TCP's in-built expiration timer.

Table 1 summarizes the impact of RobE_TIMER, RobE_SIM, RobE_eSIM, and RobE_IPS compared to [RFC6824] and [RFC8684].

Scenario	MPTCP	RobE_TIMER	RobE_SIM	RobE_eSIM	RobE_IPS
IP packet loss	Delayed connection	In the scope of timer	No impact	No impact	Delayed connection
IP	No	In the scope	No impact	No impact	No

broken	connection	of timer			connection
IP setup duration de- pendency	Default route	Default route (+ path 1..n)	Fastest path	Fastest path	Selected path
MP avail- ability duration	MP_CAPABLE HS + MP_JOIN HS	sum_1..n(MP_CAPABLE_n HS) + MP_JOIN HS	MP_CAPABLE HS + MP_JOIN HS	max(MP_CAPABLE_1 .. MP_CAPABLE_n HS)	MP_CAPABLE HS + MP_JOIN HS
Guaran- teeing session setup	Depends on the default route	Yes	Yes	Yes	Depends on selection

Table 1: Overview RobE features during initial connection setup

> IP: Initial Path; MP: Multi-Path; HS: Handshake

1.1. Terminology

This document makes use of a number of terms that are either MPTCP-specific or have defined meaning in the context of MPTCP, as follows:

Path: A sequence of links between a sender and a receiver, defined in this context by a 4-tuple of source and destination address/port pairs.

Subflow: A flow of TCP segments operating over an individual path, which forms part of a larger MPTCP connection. A subflow is started and terminated similar to a regular TCP connection.

2. Implementation without MPTCP protocol adaptation

RobE_TIMER, RobE_SIM, and RobE_IPS are compatible with the current MPTCP protocol definitions in [RFC6824] and [RFC8684] but may lack of the full optimization potential which requires protocol adaptation as detailed in Section 3. Following sections will describe the newly introduced mechanisms in detail.

2.1. Re-transmission Timer(RobE_TIMER)

In RobE_TIMER, a new connection is initiated by sending a SYN+MP_CAPABLE along the initial path. If this path is functional, the solution will perform in the same way as classic MPTCP: the initial flow will be established, and subsequent flows can be created afterwards. If however the initial path is faulty, the retransmission will be triggered on another path. This path might circumvent the dysfunctional network, and allow the client to create an initial subflow. The first path is now seen as a subsequent path and the client sends SYN+MP_JOIN messages to create a subsequent flow.

In high latency networks, the initial SYN+MP_CAPABLE messages might be delayed until the client retries sending them on another path. Once the second SYN arrives at the server, it will try to complete the three-way handshake. If the first SYN was delayed by more than the retransmission time plus half a Round Trip Time (RTT) of the second path, it will arrive at the server after the second SYN. The server could now treat the segment as obsolete and drop it.

```
             Host A                                   Host B
   ------------------------                        ----------
   Address A1     Address A2                       Address B1
   ----------     ----------                       ----------
       |              |                                 |
       |             SYN + MP_CAPABLE(Key-A[*])         |
       |Timer-------------------------------------------->|
       |              |   SYN + MP_CAPABLE(Key-A'[*])   |
       |              |-------------------------------->|
       |              |   SYN/ACK+MP_CAPABLE(Key-B')    |
       |              |<--------------------------------|
       |              | ACK + MP_CAPABLE(Key-A',Key-B') |
       |              |-------------------------------->|
       |              |   SYN + MP_JOIN(Token-B',R-A)   |
       |------------------------------------------------>|
       |   Subflow will be set up as normal MPTCP       |
       |                                                |

            [*] Key-A in the first MP-capable is related to
             RFC6824 only and does not exist in RFC8684.
```

Figure 2: The RobE_TIMER Solution

Immediately after sending the final ACK of the initial handshake, subflows are established on the remaining paths as defined in [RFC6824] and [RFC8684]

[Notes: How to set the Timer is TBD. If there is the case that the first SYN on default path arrives earlier than that from the second path, the MPTCP connection will be initialized on the path of the first SYN. The server could treat the second SYN as obsolete and drop it.]

2.2. Simultaneous Initial Paths Simple Version (RobE_SIM)

RobE_SIM is a sender only implementation and no prior negotiation with the receiver side is required. In RobE_SIM, the MPTCP connection setup benefits from the fastest path. As shown in Figure 3, host A initiates the connection handshake on more than one path independently (SA1 and SA2). The paths selected for RobE_SIM and referred to as potential initial flows, can belong to the number of interfaces on the device or a subset selected on experience. When Host A receives the first SYN/ACK back from Host B (SA3), the path carrying this message is identified as the normal initial path. Host A sends then immediately a TCP RST message (SA6.1) on any other path used for simultaneous connection setup causing an immediate termination of assigned flows (break-before-make). The terminated ones are merged as subsequent subflows following the JOIN procedure

described in [RFC6824] and [RFC8684]. The process is equivalent to any other scenario where the SYN/ACK arrives on an other path than depicted in Figure 3.

```
                  Host A                              Host B
          ------------------------                 ----------
          Address A1    Address A2                 Address B1
          ----------    ----------                 ----------
              |             |                          |
              |           SYN + MP_CAPABLE(Key-A[*])   |
    (SA1)     |--------------------------------------->|  (SB1)
              |             |    SYN + MP_CAPABLE(Key-A'[*]) |
    (SA2)     |             |--------------------------->|  (SB2)
              |             |                          |
    (SA3)     |<---------------------------------------|  (SB3)
              |         SYN/ACK + MP_CAPABLE(Key-B)    |
    (SA4)     |             |<---------------------------|  (SB4)
              |             |   SYN/ACK + MP_CAPABLE(Key-B') |
              |             |                          |
              |       ACK + MP_CAPABLE(Key-A, Key-B)   |
    (SA5)     |--------------------------------------->|  (SB5)
              |             |            RST           |
    (SA6.1)   |             |--------------------------->|  (SB6.1)
    RobE SIM  |             |   SYN + MP_JOIN(Token-B, R-A) |
    (robust)  |             |--------------------------->|
              |             |        MP_JOIN Process...  |

            [*] Key-A in the first MP-capable is related to
                RFC6824 only and does not exist in RFC8684.
```

Figure 3: MPTCP RobE_SIM Connection Setup

2.3. Heuristic Initial Path Selection (RobE_IPS)

2.3.1. Architecture

Figure 4 provides the architecture for RobE_IPS and employs an "Initial Path Selection" logic which can be integrated into the MPTCP stack or exists as an isolated module in the terminal. The IPS logic has access to a set of transmission status information for each available path or its belonging network interfaces. When an application starts a first communication, IPS selects based on the available path transmission characteristics the path with the highest probability to succeed.


```
         +-------------------+              +-------------------+
         |     Terminal      |              |      Server       |
         |  +-------------+  |              |  +-------------+  |
         |  |Application n|  |              |  |Application n|  |
         |  +-------------+  |              |  +-------------+  |
         |        |          |              |         |         |
         |  +-------------+  |              |         |         |
         |  |Initial-path |  |-------+      |         |         |
         |  |  Selection  |  |       |      |         |         |
         |  +-------------+  |       |      |         |         |
         |        |          |  +--------+  |         |         |
         |  +-------------+  |--|Internet|--|  +-------------+  |
         |  | MPTCP Stack |  |--+--------+--|  | MPTCP Stack |  |
         |  +-------------+  |              |  +-------------+  |
         +-------------------+              +-------------------+
```

Figure 4: Architecture for Initial-path Selection

2.3.2. Typical Scenarios

Two typical RobE_IPS scenarios are presented in this section. Figure 5 shows the "Initial Path Selection" logic executed for each MPTCP connection establishment. On the other hand Figure 6 describes that "Initial Path Selection" in case no path information is available. Considering the fact that no heuristics are given before a recent MPTCP connection was established, the default initial path can be adopted. Further combinations and implementations with more or less sophisticated heuristics are possible.

```
                 +---------------+
                 |  Application  |
                 |    Request    |
                 +---------------+
                         |
                         V
                 +---------------+
            +--->| Initial-path  |<---+
            |    |   Selection   |    |
            |    +---------------+    |
            |            |            |
            |            V            |Info
            |    +---------------+    |
            |    |  Set initial  |----+
            |    |     path      |
            |    |   for MPTCP   |
            |    +---------------+
            |            |
            |            V
            |    +---------------+
            |No  |Establish MPTCP|
            +----|  Connection   |
                 +---------------+
                         |Yes
                         V
```

Figure 5: RobE_IPS for each connection establishment


```
                        +--------------+
                        |  Application |
                        |    Request   |
                        +--------------+
                                |
                                V
                        +--------------+Yes
                        |     First    |------------+
                        |  Connection? |            |
                        +--------------+            |
                                |No                 |
                                V                   |
                        +--------------+            V
                 +----->| Initial-path |<-+     +--------+
                 |      |   Selection  |  |     |Default|
                 |      +--------------+  |     |initial|
                 |              |         |     |  path |
                 |              |         |     +--------+
```

```
          |              V         |Info     |
          |      +--------------+  |         |
          |      |  Set initial |--+         |
          |      |     path     |            |
          |      |   for MPTCP  |            |
          |      +--------------+            |
          |              |                   |
          |              V                   |
          |No    +--------------+            |
          +------| Successful? |<-----------+
                 +--------------+
                         |Yes
                         V
```

Figure 6: RobE_IPS using default route when no meaningful heuristic available

Figure 7 shows the process flow of "Initial Path Selection". Upon a request from an application, the IPS logic will acquire transmission status information which represents the transmission performance of each available path or network interface and evaluate it. The transmission status information is characterized by at least one of the parameters: signal strength, throughput, round-trip time (RTT), and link success rate. In this way, the path with the best transmission performance can be determined and used for connection establishment.


```
                 |
                 V
    +----------------------------+
    |Acquire transmission status|
    | info for available paths  |
    +----------------------------+
                 |
                 V
    +----------------------------+
    |   Evaluating the status   |
    |    for available paths    |
```

```
+---------------------------+
             |No
             V
+---------------------------+
| Determining an available  |
|     path with better      |
|      transmission         |
|      performance          |
+---------------------------+
             |
             V
+---------------------------+
|     Using the network     |
|         interface         |
|corresponding to the path |
| with better transmission  |
|performance for connection |
|       establishment       |
+---------------------------+
             |
             V
```

Figure 7: Implementation process for Initial Path Selection

2.3.3. Path decision information

The level of heuristic can be mainly divided into three layers: application level, transport-layer level and link-layer level based on the information acquisition method. For example, RTT can be calculated for each path within an MPTCP connection and belongs thereof to the transport-layer level. The transmission status information for each available path SHOULD be characterized by at least one of the parameters: signal strength, throughput, RTT, and link success rate. Application level information are more seen for statistical purposes.

* Application level: application name, domain name, port number, and location.

* Transport-layer level: RTT, CWND, Error rate.

2.3.4. Initial Path Selection use local RTT information

Figure 8 presents an "Initial Path Selection" logic based on RTT, e.g. assuming two paths over LTE and WiFi access. RTT calculation on the transport layer usually reflects the time when an information is sent and a related acknowledgment received. For an asymmetric usage (e.g. download only) of a communication it might happen that recent RTT calculation is only available on sender side which is possibly not the side which employs the IPS logic. A solution for this can be found in Section 3.2. Instead of using the most recent RTT value of a path a filtered value consisting of several measured RTTs can be used. A RTT can also be derived from link layer information but may have a limited meaning only when it does not represent the end-to-end latency.

```
+--------------------+
|    New Session     |
+--------------------+
          |
          V
+--------------------+ No
|Running Connections|-----------+
|(LTE.RTT<WiFi.RTT) |           |
+--------------------+           |
          |Yes                   |
          V                      V
+--------------------+   +-----------------+
|     Set LTE as     |   |   Set WiFi as   |
|    initial path    |   |  initial path   |
+--------------------+   +-----------------+
```

Figure 8: Initial-path Selection based on RTT

2.4. Combination of RobE_SIM and RobE_IPS

In an implementation, a single solution may not be sufficient to achieve an expected behavior. Combination of approaches to improve robustness is recommended therefore. Figure 9 shows the combination of RobE_SIM and RobE_IPS. RobE_SIM can be used at the very beginning when the sender is without any path information followed by RobE_IPS for consecutive connections.

```
       +--------------+
       |  Application |
       |    Request   |
       +--------------+
               |
               V
       +--------------+
+------>| Any data for | No
|      | Initial Path |----------+
|      |  Selection?  |          |
|      +--------------+          |
|              |                 |
|              V                 V
|      +--------------+     +--------+
|      | Initial-path |     |RobE_SIM|
|      |  Selection   |<-+  +--------+
|      +--------------+  |         |
|              |         |         |
|              V         |Info     |
|      +---------------+ |         |
|No    |Establish MPTCP|-+         |
+------|   Connection  |<----------+
       +---------------+
               |
               V
 No    +---------------+
<------|  Successful ? |
Network+---------------+
Problem        |Yes
               V
```

Figure 9: Combination of RobE_SIM and RobE_IPS

2.5. Combination of RobE_TIMER and RobE_IPS

Since RobE_IPS solely does not guarantee that a session can be set up based on the selection of initial path, it can also be combined with RobE_TIMER which generates less overhead compared to the combination with RobE_SIM in Section 2.4 and guarantees session setup. RobE_TIMER can be introduced to optimize the control of path switching when the initial path selected by RobE_IPS is dysfunctional. When the system enables RobE_IPS and uses the selected initial path for session establishment, it sets the timer for path switching. When timer is expired, the system will change to another path to re-establish connection according to Section 2.1.

```
                                +---------------+
                                |  Application  |
                                |    Request    |
                                +---------------+
                                        |
                                        V
                                +---------------+
                                |  Initial Path |
                         |----->|   Selection   |
                         |      | and Set Timer |
                         |      +---------------+
                         |              |
                         |              V
                         |Yes   +---------------+
                         +------| Timer is up?  |
                                +---------------+
                                        |No
                                        V
                                +---------------+
                                |Establish MPTCP|
                                |  Connection   |
                                +---------------+
                                        |
                                        V
                          No    +---------------+
                         <------|  Successful?  |
                         Network+---------------+
                         Problem        |Yes
                                        V
```

Figure 10: Combination of RobE_Timer and RobE_IPS

3. Implementation with Bi-directional MPTCP Support

Solutions which requires bi-directional support between two MPTCP hosts promise to have better and possibly more features. However, they cannot be defined without extending current standards in [RFC6824] and [RFC8684]. The RobE_SIM and RobE_IPS approach are both capable of profiting from an explicit support of the remote end host and will be defined within this section.

3.1. Simultaneous Initial Paths Extended Version (RobE_eSIM)

RobE_eSIM extends RobE_SIM by reusing the potential initial flows. This eliminates the overhead from RobE_SIM by introducing a new option MP_JOIN_CAP and accelerate the transmission speed by early availability of multiple paths. Further it relaxes the dependency on a reliable third ACK of the 3-way handshake in [RFC8684]. Remote endpoint support can be negotiated in two ways, an implicit one described in Section 3.1.1 or an explicit on which is described in Section 3.1.2.

3.1.1. RobE_eSIM implicit Negotiation and Procedure

Similar to RobE_SIM in Section 2.2, the establishment process of [RFC6824] or [RFC8684] is applied independently on multiple paths simultaneously. In Figure 11 this is shown in SA1 and SA2. The first path which returns a SYN/ACK (e.g. SA3) is selected as the initial path and proceeds with the traditional establishment process (SA5). Any other path which has to send the final ACK of the 3-way handshake includes a new option MP_JOIN_CAP (see definition in Section 3.1.3.2) instead of an MP_CAPABLE (SA6.2).

```
                 Host A                                Host B
         ------------------------                   ----------
         Address A1    Address A2                   Address B1
         ----------    ----------                   ----------
             |             |                             |
             |          SYN + MP_CAPABLE(Key-A[*])       |
     (SA1)   |------------------------------------------>|   (SB1)
             |             |      SYN + MP_CAPABLE(Key-A'[*]) |
     (SA2)   |             |------------------------------>|   (SB2)
             |             |                             |
     (SA3)   |<------------------------------------------|   (SB3)
```

```
               |          SYN/ACK + MP_CAPABLE(Key-B)         |
       (SA4)   |             |<-------------------------------|  (SB4)
               |             |   SYN/ACK + MP_CAPABLE(Key-B') |
               |             |                                |
               |        ACK + MP_CAPABLE(Key-A, Key-B)        |
       (SA5)   |--------------------------------------------->|  (SB5)
               |             |                                |
       (SA6.2) |             |                                |  (SB6.2)
      RobE EXT |             | ACK + MP_JOIN_CAP(Key-A, HMAC) |
      (+fast)  |             |------------------------------->|

               [*] Key-A in the first MP-capable is related to
                   RFC6824 only and does not exist in RFC8684.
```

Figure 11: MPTCP RobE_eSIM implicit Connection Setup

Following the possible process in Figure 11, two further constellations are imaginable and elaborated below.

1. In the flow diagram Figure 11, A1<->B1 is assumed to be the initial flow. A2<->B1 shall be recycled and the ACK is sent with MP_JOIN_CAP. Furthermore, the MP_CAPABLE arrives first at Host B (SB5) and the MP_JOIN_CAP afterwards (SB6.2). When the MP_JOIN_CAP is received, Host B has to iterate over the connection list once (like MP_JOIN) and check for Key-A availability. If a Key-A connection is found, this one is validated against the HMAC value. The validation has two reasons: first, several Key-A can exist, because different hosts may choose the same Key-A by accident. Furthermore, no one can join a connection by just recording/brute-forcing Key-A and duplicating the request.

2. Like above, but MP_JOIN_CAP arrives before last MP_CAPABLE at Host B

   * [RFC8684]; Based on Key-A, Host B will iterate over the connection list, but it will not find a match, because Key-A of the previous selected initial flow (SA3, SA5) has not arrived yet. So it will continue with a fast iteration only over the connections which are still in establishment phase using the 10 bit Key-B fast hash (crc16(Key-B) & 0x3FF). If it matches against a (precomputed) existing Key-B_fast_hash in

the connection list, it will validate the request using the HMAC(Key-A+B+B') to ensure legitimation. If successful, both, the initial flow and the MP_JOIN_CAP flow, can be immediately established. This is true, because without the knowledge of Key-B, Host A could not calculate the HMAC. So it is clear, that Host A had received the SYN/ACK (SB3). This also mitigates the exchange of a reliable ACK during the handshake process. MPTCP sends the Key-A only with the last ACK and therefore prevents subsequent flow establishment until successful reception at Host B. Using RobE_EXT, the reception of an MP_JOIN_CAP ([RFC8684]) is sufficient to establish both, the path carrying Key-B and Key-B'.

   * [RFC6824]; Can match based on Key-A, same effort as for an MP_JOIN.

3. A2<->B1 is selected as initial flow, because the respective SYN/ ACK returns earlier at Host A. It is the same as above, just the other way round.

3.1.2. RobE_eSIM explicit Negotiation and Procedure

The process of an explicit negotiation of RobE_eSIM follows Figure 11 but uses the ROBE_eSIM_EN option Figure 13 additionally during the handshake procedure.

```
          Host A                                           Host B
  ------------------------                               ----------
  Address A1    Address A2                               Address B1
  ----------    ----------                               ----------
      |             |                                        |
      |      SYN+MP_CAPABLE+ROBE_eSIM_EN(Key-A[*])           |
      |----------------------------------------------------->|
      |             | SYN+MP_CAPABLE+ROBE_eSIM_EN(Key-A'[*]) |
      |             |--------------------------------------->|
      |      SYN/ACK+MP_CAPABLE+ROBE_eSIM_EN(Key-B)          |
      |<---------------------------------------------------->|
      |             | SYN/ACK+MP_CAPABLE+ROBE_eSIM_EN(Key-B')|
      |             |<---------------------------------------|
```

```
     |      ACK+MP_CAPABLE(Key-A,Key-B)                     |
     |----------------------------------------------------->|
     |             |                                        |
     |             |  ACK+MP_JOIN_CAP(Key-A,HMAC)           |
     |             |--------------------------------------->|
     |             |                                        |
```

[*] Key-A in the first MP-capable is related to RFC6824 only and does not exist in RFC8684.

Figure 12: MPTCP RobE_eSIM explicit Connection Setup

3.1.3. Protocol Adaptation

3.1.3.1. ROBE_eSIM_EN Option

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------+-------+---------------+
|     Kind      |    Length     |Subtype|    (reserved)         |
+---------------+---------------+-------+-------+---------------+
```

Figure 13: ROBE_eSIM_EN_OPTION

3.1.3.2. MP_JOIN_CAP Option


```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------+-------+---------------+
|     Kind      |    Length     |Subtype|       |    ADDR_ID    |
+---------------+---------------+-------+-------+---------------+
|                   Sender's Key-A (64 bits)                    |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
|                       HMAC (>=96 bits)                        |
|                                                               |
|                                                               |
```

```
   :                                                               :
   +---------------------------------------------------------------+
    Key-B_fast_hash = crc16(Key-B) & 0x3FF          -> (10bit)
    HMAC_keys =  HMAC(Key-A+Key-B+Key-B')           -> (>=96bit)
    HMAC =  (HMAC_keys & ~0x3FF) | Key-B_fast_hash  -> (size HMAC_keys)
```

Figure 14: MP_JOIN_CAP

Computational effort on receiver side is most often expected to be the same as with MP_JOIN. Key-A ensures identification of related flows Key-B_fast_hash enables MP session even when selected initial flow is not fully established yet (slight computational overhead). HMAC authenticates relationship of initial and potential initial flows.

3.1.4. Fallback Mechanisms

3.1.4.1. Fallback mechanism for implicit RobE_eSIM

[TBD]

3.1.4.2. Fallback mechanism for explicit RobE_eSIM

This mechanism considers that both sides support MPTCP capability but the receiver is not equipped with RobE_eSIM. MPTCP session with RobE_eSIM negotiation will seamlessly fallback to normal MPTCP process.

[Requires further check how an unaware Host B reacts on possible ROBE_eSIM_EN; Ignore or RST? See also RFC6824 Sec. 3.6 "Should fallback [...] the path does not support the MPTCP options"]


```
                   Host A                               Host B
         ------------------------                     ----------
         Address A1    Address A2                     Address B1
         ----------    ----------                     ----------
             |             |                              |
```

```
            |      SYN+MP_CAPABLE+ROBE_eSIM_EN            |
            |------------------------------------------->|
            |             | SYN+MP_CAPABLE+ROBE_eSIM_EN |
            |             |---------------------------->|
            |         SYN/ACK+MP_CAPABLE                  |
            |<------------------------------------------->|
            |             |    SYN/ACK+MP_CAPABLE       |
            |             |<----------------------------|
            |       ACK+MP_CAPABLE                        |
            |------------------------------------------->|
            |             |          RST                |
            |             |---------------------------->|
            |             |       SYN+MP_JOIN           |
            |             |---------------------------->|
            |             |     MP_JOIN Process...      |
            |             |                             |
```

Figure 15: Fallback to MPTCP when missing RobE_eSIM support

3.1.4.3. Fallback to regular TCP when missing MPTCP support

When the receiver is not MPTCP enabled, MPTCP session with RobE_eSIM negotiation will seamlessly fallback to regular process which is illustrated in this section.

```
                   Host A                                  Host B
          ------------------------                      ----------
          Address A1    Address A2                      Address B1
          ----------    ----------                      ----------
              |             |                               |
              |      SYN+MP_CAPABLE+ROBE_eSIM_EN            |
              |-------------------------------------------->|
              |             | SYN+MP_CAPABLE+ROBE_eSIM_EN   |
              |             |------------------------------>|
              |          SYN/ACK                            |
              |<------------------------------------------->|
              |             |    SYN/ACK                    |
              |             |<------------------------------|
              |            ACK                              |
              |-------------------------------------------->|
              |             |          RST                  |
              |             |------------------------------>|
              |             |   Regular TCP Process...      |
              |             |                               |
```

Figure 16: Fallback to TCP without MPTCP support

3.1.5. Comparison Robe_SIM and RobE_eSIM

Potential initial flows in RobE_SIM Section 2.2 and RobE_eSIM Section 3.1 guarantee MPTCP session establishment if at least one selected path for session establishment is functional. Figure 17 makes the differences between both approaches visible and points to the latest decision possibility during session setup when RobE_SIM or RobE_eSIM can be selected. Until SA5 in Figure 17 traditional MPTCP connection setup is independently applied on multiple paths simultaneously and offers to select the initial flow later (potential initial flows). The final decision which path is selected as the main one and the handling of the remaining flow(s) differs in SA6.1 when RobE_SIM is applied or instead SA6.2 RobE_eSIM.

```
                Host A                                  Host B
        ------------------------                       ----------
        Address A1    Address A2                       Address B1
        ----------    ----------                       ----------
            |             |                                |
            |            SYN + MP_CAPABLE(Key-A[*])        |
    (SA1)   |--------------------------------------------->|  (SB1)
            |             |    SYN + MP_CAPABLE(Key-A'[*]) |
    (SA2)   |             |------------------------------->|  (SB2)
            |             |                                |
    (SA3)   |<---------------------------------------------|  (SB3)
            |          SYN/ACK + MP_CAPABLE(Key-B)         |
    (SA4)   |             |<-------------------------------|  (SB4)
            |             |   SYN/ACK + MP_CAPABLE(Key-B') |
            |             |                                |
            |        ACK + MP_CAPABLE(Key-A, Key-B)        |
    (SA5)   |--------------------------------------------->|  (SB5)
            |             |             RST                |
    (SA6.1) |             |------------------------------->|  (SB6.1)
   RobE SIM |             |                                |
   (robust) |             |                                |
   -------------------------------------------------------------------
   RobE EXT |             |                                |
   (+fast)  |             | ACK + MP_JOIN_CAP(Key-A, HMAC) |
    (SA6.2) |             |------------------------------->|  (SB6.2)

            [*] Key-A in the first MP-capable is related to
                RFC6824 only and does not exist in RFC8684.
```

Figure 17: MPTCP RobE_SIM and RobE_eSIM connection setup

3.1.6. Security Consideration

[Tbd, however no differences to [RFC6824] and [RFC8684] are expected]

3.2. Heuristic Initial Path Selection with remote RTT Measurement

3.2.1. Description

Usually the path RTT can be determined by a time difference between sending a package and receiving an ACK and is integrated into the TCP protocol. For asymmetric transmission, the latest RTT for TCP flows is calculated by the side which sends data at latest and possible does not correspond to the site which employs RobE_IPS. This problem is already elaborated in Section 2.3.4 and can be solved by transmitting the RTT information per subflow. The negotiation procedure is depicted in Figure 18 and uses the MPTCP option L_RTT_EN defined in Section 3.2.2.


```
        Host A                                Host B
------------------------                    ----------
Address A1     Address A2                   Address B1
----------     ----------                   ----------
    |              |                             |
    |         SYN+MP_CAPABLE+L_RTT_EN            |
    |------------------------------------------->|
    |       SYN/ACK+MP_CAPABLE+L_RTT_EN          |
    |<-------------------------------------------|
    |             ACK+MP_CAPABLE                 |
    |------------------------------------------->|
    |    ACK+DSS+L_RTT_EN(latest RTT)+Data       |
    |<-------------------------------------------|
    |              |         SYN+MP_JOIN         |
    |              |---------------------------->|
    |              |      MP_JOIN Process...     |
    |              |                             |
```

Figure 18: Negotiation procedure for RTT exchange

A successful negotiation allows the exchange of the measured RTT value from one subflow of an MPTCP host to another using the "Latest RTT" field within the L_RTT_EN option.

3.2.2. Protocol Adaptation

Calculating the "Latest RTT" by a remote host in an asymmetry transmission scenario should be transferred from remote host to the client running RobE_IPS. So a new MPTCP subtype option named L_RTT_EN is allocated for this function. During the three-way handshake L_RTT_EN is used for negotiation of remote RTT measurement capability between client and server (in Section 3.2.1). When both parts support the usage of remote RTT measurement, the "Latest RTT" field in L_RTT_EN is applied for carrying the value of latest RTT computed by the remote host.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------+-------+---------------+
|     Kind      |    Length     |Subtype|      (reserved)       |
+---------------+---------------+-------+-------+---------------+
|                    Latest RTT (32 bits)                       |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
```

Figure 19: ROBE_L_RTT_EN OPTION

3.2.3. Fallback Mechanism

When the receiver is not L_RTT_EN capable, MPTCP session with L_RTT_EN negotiation will seamlessly fallback to normal MPTCP process.

[TBD, Need same checks as Section 3.1.4.2]

```
                Host A                           Host B
     ------------------------              ----------
     Address A1    Address A2              Address B1
     ----------    ----------              ----------
         |             |                        |
         |        SYN+MP_CAPABLE+L_RTT_EN       |
         |------------------------------------->|
         |         SYN/ACK+MPTCP_CAPABLE        |
         |<-------------------------------------|
         |           ACK+MPTCP_CAPABLE          |
         |------------------------------------->|
         |             |      SYN+MP_JOIN       |
         |             |----------------------->|
         |             |    MP_JOIN Process...  |
         |             |                        |
```

Figure 20: Fallback to MPTCP without RobE_IPS

3.2.4. Security Consideration

[Tbd]

4. IANA Considerations

This document defines three new values to MPTCP Option Subtype as following.

Value	Symbol	Name	Reference
TBD	ROBE_eSIM_EN	RobE_eSIM enabled	Section 3.1
TBD	MP_JOIN_CAP	Join connection directly in RobE_eSIM	Section 3.1
TBD	L_RTT_EN	Server RTT enabled	Section 3.2

Table 2: RobE Option Subtypes

5. References

5.1. Normative References

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

[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, <https://www.rfc-editor.org/info/rfc6824>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <https://www.rfc-editor.org/info/rfc8684>.

5.2. Informative References

[RobE_paper]

Amend, M., Rakocevic, V., Matz, A.P., and E. Bogenfeld, "RobE: Robust Connection Establishment for Multipath TCP", ANRW '18 p. 76-82, 16 July 2018, <http://doi.acm.org/10.1145/3232755.3232762>.

[RobE_slides]
Amend, M., Matz, A.P., and E. Bogenfeld, "A proposal for MPTCP Robust session Establishment (MPTCP RobE)", IETF 99 Multipath TCP WG session, 18 July 2017, <https://datatracker.ietf.org/meeting/99/materials/slides-99-mptcp-a-proposal-for-mptcp-robust-session-establishment-mptcp-robe-01>.

Authors' Addresses

Markus Amend
Deutsche Telekom
Deutsche-Telekom-Allee 9
64295 Darmstadt
Germany
Email: Markus.Amend@telekom.de

Jiao Kang
Huawei
D2-03,Huawei Industrial Base
Shenzhen
Guangdong, 518129
China
Email: kangjiao@huawei.com