Improving Multipath TCP Backup Subflows
draft-bonaventure-mptcp-backup-00

Abstract

This document documents some issues with the current definition of the backup subflows in [RFC6824]. The solution proposed in [RFC6824] works well when a subflow completely fails. However, if a subflow suffers from huge packet losses, but still remains up, then the delay to switch to the backup subflow may be very long. We propose to measure the evolution of the retransmission timer (RTO) to detect the bad performance of subflows.

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 7, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
1. Introduction

Multipath TCP is an extension to TCP [RFC0793] that was specified in [RFC6824]. A Multipath TCP connection is composed of one or more subflows. Each subflow is a TCP connection that is established by using the classical TCP three-way handshake. The subflows that compose a Multipath TCP connection are not all equal. [RFC6824] defines two types of subflows:

- the regular subflows
- the backup subflows

The regular subflows can be used to transport any data. The backup subflows are intended to be used only when all the regular subflows have failed. Section 2.5 of [RFC6824] defines them by using the following sentence: "Hosts can indicate at initial subflow setup whether they wish the subflow to be used as a regular or backup path - a backup path only being used if there are no regular paths available."

Intuitively, a user expects that the backup subflow will be used when the regular subflow fails to continue the data transfer and minimize the impact of the failure on the Multipath TCP connection.

In this document, we first describe in Section 2 how Multipath TCP operates when backup subflows are used and some of the operational problems that this causes. Backup subflows work well when subflows
completely fail due to, for example, the reception of a RST segment or the invalidity of the IP address associated to the subflow (expired lease time, de-attachment from network, etc.). However, there are many practical situations where the failure of a regular subflow cannot be quickly detected and the user experience suffers. We then propose in Section 3 a slight modification to the handling of the backup subflows in Multipath TCP.

2. What is a Subflow Failure?

Experience with Multipath TCP shows that the backup subflows that are only used when all the other subflows have failed works well on fixed hosts where the loss of connectivity can be quickly detected by the affected host. However, there are many situations where it can be difficult to detect the failure of a regular subflow.

To understand the situation, let us consider the simple network shown in Figure 1. In this network, the client has established two subflows:

- a regular subflow passing through router1 and router2
- a backup subflow passing through router3 and router4

[RFC6824] supports two methods to signal that a subflow is a backup subflow:

- setting the B bit in the MP_JOIN option that is used to create the subflow
- sending the MP_PRIO option with the B bit set

Note that in both cases, when a host sets the B bit in the MP_JOIN or sends an MP_PRIO option, it requests the other host to only use the subflow if the other regular subflows have failed. Setting the B bit...
in the MP_JOIN option or sending the MP_PRIO option does not affect
the data sent by the host that sends this option [RFC6824].

Let us now consider three different failure scenarios. For
simplicity, we assume that all the data flows from the Server to the
Client and that the top subflow is the primary subflow while the
bottom subflow was signaled as a backup subflow.

Our first failure scenario is the simplest one: the failure of link1.
In this case, the Client detects the failure locally. This detection
can be fast with wired link layer technologies and slower with some
wireless technologies. Once the failure has been detected, the
Client can either send a REMOVE_ADDR option to indicate the failure
of its address attached to link1 or send an MP_PRIO option with the B
bit reset over the backup subflow. In both cases, a single segment
sent over the backup subflow is sufficient to inform the Server of
the failure of the primary subflow. Note that the REMOVE_ADDR and
the MP_PRIO options are sent unreliably. This implies that any loss
of these options will further delay the recovery on the Server.

Our second failure scenario is the symmetric scenario: the failure of
link2. In this case, the Server will react by sending a REMOVE_ADDR
option over the backup subflow to indicate the loss of the address
attached to this link. Since the Server knows that the primary
subflow has failed, it can immediately start to use the backup
subflow to send data to the Client. Experiments show that these two
failure scenarios work well [Cellnet12].

The third failure scenario is a failure of the link between router1
and router2. Different types of failures are possible on this link.
We consider two extreme cases. The first case is a pure link failure
that is detected by the two routers. Since there is no alternate
path between router1 and router2 in our example network, the Client
cannot reach the Server anymore over the top path. Once router1 and
router2 have detected the failure, they will return ICMP destination
unreachable messages to the Client and the Server. This error
message could suggest a failure of the primary subflow. According to
[RFC1122], this ICMP message should cause the termination of the top
subflow. However, according to [RFC5461], current TCP
implementations do not follow this recommendation and ignore the
received ICMP messages. This is motivated by the risk of denial of
service attacks that could disrupt existing TCP connections by
sending spoofed ICMP messages. A Multipath TCP implementation could
react differently and for example consider the subflow over which the
ICMP message was received as temporarily unusable to cause the
utilization of other (possibly backup) subflows.
If a Multipath TCP implementation does not react to ICMP messages, the last resort method to detect the failure of the top path is the retransmission timer (RTO). TCP implementations apply an exponential backoff algorithm to the retransmission timeout [RFC6298]. If the primary path fails, the retransmission timeout associated to this path will double until it reaches the maximum value configured on the TCP stack. On many stacks, this limit is in the order of tens of minutes which does not match the expectations of the Multipath TCP user who expects that her backup subflow will be used earlier than that. A similar situation occurs when the link between the two routers remains up but is so congested that packets sent on the regular subflow rarely traverse the link [BD2015]. In this case, the user also expects to be able to quickly use the backup subflow to preserve the end-to-end connectivity.

3. Detecting Underperforming Subflows

As explained in the previous section, users cannot accept a too long delay to detect the failure of a regular subflow and the switch to an existing backup subflow. [RFC6824] allows a host to specify that a subflow is a backup subflow, but there is no definition of underperforming subflows and no mechanism to allow applications to specify a switchover time to a backup subflow.

Various techniques exist to detect failures. Shim6 [RFC5533] includes the REAP protocol [RFC5534] to verify the reachability of addresses. BFD [RFC5880] is used to detect link failures between routers and also over multihop paths [RFC5883]. Depending on the chosen parameters, these protocols can achieve fast detection and/or low overhead. We do not believe that additional protocols are required to quickly detect the failure of a subflow. With its retransmission timer that doubles after each unsuccessful retransmission, Multipath TCP already has the ability to detect underperforming subflows. If data is transmitted over a broken subflow, the retransmission timer of this subflow will quickly increase. These successive retransmissions are an appropriate mechanism to detect the failure of a subflow and switch to a backup one provided that the TCP retransmission timer does not become too high.

[RFC0793] specifies an abstract API that allows user applications to indicate bounds on the retransmission timer. [RFC5482] goes further in by proposing a TCP option that can be used to signal a proposed maximum value for the TCP retransmission timeout through the User Timeout option [RFC5482]. This option specifies the maximum time that some data can remain unacknowledged before considering the connection to have failed. In [RFC5482], the User Timeout is encoded as a 15 bits field that represents seconds or minutes. This implies
that the User Timeout option cannot be used to signal a bound smaller than 1 second.

With the User Timeout option, the TCP connection must be terminated once its RTO reaches the signaled maximum value.

[RFC5482] defines the following parameters for the RTO:

- U_LIMIT: the upper limit on the USER TIMEOUT
- L_LIMIT: the lower limit on the USER TIMEOUT

In addition, the application can specify, e.g. through a socket option, the USER TIMEOUT that it wishes to use and advertise to the peer: ADV_UTO. Similarly, the REMOTE_UTO is the User Timeout option received from the peer. Then, [RFC5482] defines the USER TIMEOUT with the following formula:

USER_TIMEOUT = min(U_LIMIT, max(ADV_UTO, REMOTE_UTO, L_LIMIT))

[RFC6824] does not discuss precisely how the User Timeout option should be handled if received over a Multipath TCP connection. If this option is set through the regular socket API that does not expose any information about the subflows, it must apply on the overall Multipath TCP connection.

In this document, we envision an API that exposes some parts of Multipath TCP to the application to enable them to make a better utilisation of the features of the protocol. Such an API would expose some information about the subflows to the applications.

A first possibility to control the performance of the subflows could be to specify a USER_TIMEOUT on a per subflow basis and terminate the subflows whose RTO has reached the USER_TIMEOUT. However, terminating an underperforming subflow may be too severe in environments where there are transient losses such as wireless networks. An alternative approach is to tag the subflow as underperforming and modify the operation of Multipath TCP.

According to [RFC6824], an established subflow can operate in two modes:

- primary mode
- backup mode

The initial subflow is always created in primary mode. When a subflow is created, its mode depends on the B bit of the received
MP_JOIN option. The reception of the MP_PRIO option changes the mode of the corresponding subflow. We a Multipath TCP implementation sends data, it always selects one of the available primary subflows to transmit the data. The backup subflows are only selected if there is no established subflow in primary mode.

We propose a new mode of operation: the underperforming mode. Subflows are still established in the primary or backup mode as explained above. A subflow enters the underperforming mode as soon as its retransmission timer (RTO) reaches a configurable limit. At this point, the subflow is considered to be underperforming. An underperforming subflow cannot be selected for data transmission if there exists another subflow in primary or backup mode. Once a subflow has been tagged as underperforming, it remains in this mode as long as there are unacknowledged data on this subflow. Once all data has been acknowledged, it may return to the primary or backup mode. Further experimentation is required to evaluate how quickly an underperforming subflow should leave the underperforming mode once all data has been acknowledged.

System administrators and/or application developers (e.g. through a socket option) should be able to specify the maximum RTO that causes a Multipath TCP subflow to be tagged as underperforming. For this, we propose two new parameters:

- **UPERF_ADV_TO**: the upper threshold on the RTO that forces the subflow to be considered as underperforming
- **UPERF_REMOTE_TO**: the upper threshold on the RTO received from the remote peer

The **UPERF_ADV_TO** is configured locally on the host. It could be configured globally or on a per connection basis. The configuration applies to all subflows of a Multipath TCP connection.

The **UPERF_REMOTE_TO** is received in a Multipath TCP option. This value applies only on the subflow over which it has been received.

The **UPERF_TIMEOUT** that is used to detect underperforming subflows is then computed by using the following formula:

\[
UPERF_TIMEOUT = \min(U\_LIMIT, \max(UPERF\_ADV\_TO, \UPERF\_REMOTE\_TO, L\_LIMIT))
\]

If a **USER\_TIMEOUT** is defined for the Multipath TCP connection, its value MUST be larger than the **UPERF\_TIMEOUT**.
The UPERF_REMOTE_TO can be signaled by using a Multipath TCP option to the remote peer. This document proposes the following experimental option to encode this information (Figure 2:

```
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| Flags |  Experiment   |
+---------------+---------------+-------+-------+---------------+
| Id. (16 bits) |       Maximum  RTO  (milliseconds) |
+---------------------------------------------------------------+
```

Figure 2: The UPERF Maximum RTO experimental Multipath TCP option

We do not use the same encoding as [RFC5482] because the encoding for the USER_TIMEOUT option cannot support maximum RTOs that are smaller than one second. There are already use cases where users do not accept to wait such a long time before switching to a backup subflow.

The Experiment Identifier should be TBD and the flags must be used as defined in [I-D.bonaventure-mptcp-exp-option].

If experiments conducted with this option show positive results, it could be possible to update the MP_PRIO option to encode the maximum RTO information as shown in Figure 3.

```
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|     |B| AddrID (opt) |
+---------------+---------------+-------+-----+-+--------------+
|            Maximum RTO  (milliseconds)        |
+-----------------------------------------------+
```

Figure 3: The UPERF Maximum RTO Multipath TCP option

4. Security considerations

This document does not modify the security considerations for Multipath TCP.

5. IANA considerations

This document proposes the UPERF experimental Multipath TCP option whose experiment identifier is TBD.
If experiments are successful, an update to this document will propose a new format for the MP_PRIO option defined in [RFC6824].

6. Conclusion

In this document, we have first explained some issues with the handling of backup subflows by Multipath TCP. Multipath TCP meets the expectations of its uses when subflows fail completely. In this case, Multipath TCP moves the traffic over the backup subflows. However, if the primary subflows underperform, Multipath TCP implementations may try to retransmit data over such subflows for a long period of time instead of switching quickly to the backup subflow. We have then proposed to set an upper bound on the retransmission timer (RTO) to detect underperforming subflows. This bound can be set locally or exchanged through the proposed UPERF Multipath TCP option.

7. Acknowledgements

This work was partially supported by the FP7-Trilogy2 project. We would like to thank Mohamed Boucadair for his useful suggestions and comments on this document.

8. References

8.1. Normative References


8.2. Informative References


[I-D.bonaventure-mptcp-exp-option]
Bonaventure, O., benjamin.hesmans@uclouvain.be, b., and M. Boucadair, "Experimental Multipath TCP option", draft-bonaventure-mptcp-exp-option-00 (work in progress), June 2015.


Authors’ Addresses

Olivier Bonaventure
UCLouvain
Email: Olivier.Bonaventure@uclouvain.be

Quentin De Coninck
UCLouvain
Email: Quentin.Deconinck@student.uclouvain.be
Mathieu Baerts
UCLouvain
Email: Matthieu.Baerts@student.uclouvain.be

Fabien Duchene
UCLouvain
Email: Fabien.Duchene@uclouvain.be

Benjamin Hesmans
UCLouvain
Email: Benjamin.Hesmans@uclouvain.be