Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 03, 2013

P. Eardley
BT

May 02, 2013

Survey of MPTCP Implementations (blank version for implementers to fill in)

draft-eardley-mptcp-implementations-survey-01

Abstract

This survey gathers information from people who have implemented MPTCP, in particular to help progress the protocol from Experimental to Standards track.

It is ready to be filled in, if you are an implementer. Thank-you!

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 November 03, 2013.

Table of Contents

1. Introduction

The goal of this survey is to gather information from people who have implemented MPTCP, in particular to help progress the protocol from Experimental to Standards track.

The MPTCP WG's current charter (2013-03-14) states: "The primary goal of the working group is to create a bis version of the protocol document on the Standards track. This develops the current

Experimental document ... incorporating experience from (for example) implementations, interoperability events, experiments, usage scenarios, protocol corner cases, and feedback from TCPM ... [Also, the] working group will document implementation advice. The current documents have several points where an implementer may benefit from

guidance, for example about heuristics such as buffer sizing, or from advice about alternative implementations such as bump-in-the-stack."

The survey gathers relevant information to help this, for example about: the existence of running code (especially several independent, interoperable implementations); the parts of the experimental spec that have not been implemented (or not been used); the parts that need improvement, in terms of functionality or clarity.

The survey also takes the opportunity to gather some limited information about operational experiences and deployments.

The WG is very grateful to those implementers taking the trouble to fill in this survey - as well, of course, as their effort in creating the implementation!

The intention is to produce a revised version of this draft that incorporates a summary of the replies and an appendix with all the full responses.

2. Guidelines for filling in the survey

Please could each implementation team fill in the survey. If you cannot answer a question just skip it, as partial responses are also very valuable.

Please return your completed survey by email attachment to the MPTCP WG Chairs: Philip Eardley <philip.eardley@bt.com> and Yoshifumi Nishida <nishida@sfc.wide.ad.jp>.

If you wish to remain anonymous, please indicate -- we'll replace your answers in Question 1 by "Anonymous Implementation #n", but publish the rest of your answers unaltered, unless you ask for further obfuscation. Please note that Philip Eardley works for BT and Yoshifumi Nishida works for GE; if you wish to remain anonymous from one of us, please return it to the other.

3. Survey

3.1. Question 1: Your details

Question 1 gathers some information about the team that has implemented MPTCP.

1. Your institution:

2. Name(s) of people in your implementation and test teams:

3. Do you want your answers to Question 1.1 and 1.2 above to be anonymised?

3.2. Question 2: Preliminary information about your implementation

Question 2 gathers some preliminary information.

1. What OS is your implementation for? (or is it application layer?)

2. Do you support IPv4 or IPv6 addresses or both?

3. Is it publicly available (or will it be?) (for free or a fee?)

4. Overall, what are you implementation and testing plans? (details can be given against individual items later)

5. Is it an independent implementation? Or does it build on another MPTCP implementation -which one?

6. Have you already done some interop tests, for example with UCLouvain's "reference" Linux implementation?

7. Would you be prepared to take part in an interop event, for example adjacent to IETF-87 in Berlin?

3.3. Question 3: Support for MPTCP's Signalling Functionality

Question 3 asks about support for the various signalling messages

that the MPTCP protocol defines.

*** For each message, please give a little information about the status of your implementation: for example, you may have implemented it and fully tested it; the implementation may be in progress; you have not yet implemented it but plan to soon (timescale?); you may you have no intention to implement it (why?); etc.

1. Connection initiation (MP_CAPABLE) [Section 3.1 RFC6824]

    a. What is the status of your implementation?

    b. Any other comments or information?

2. Starting a new subflow (MP_JOIN) [Section 3.2 RFC6824]

    a. What is the status of your implementation?

    b. Can either end of the connection start a new subflow (or only the initiator of the original subflow)?

    c. What is the maximum number of subflows your implementation can support?

    d. Any other comments or information?

3. Data transfer (DSS) [Section 3.3 RFC6824]

    a. What is the status of your implementation?

    b. The "Data ACK" field can be 4 or 8 octets. Which one(s) have you implemented?

    c. The "Data sequence number" field can be 4 or 8 octets. Which one(s) have you implemented?

    d. Does your implementation support the "DATA_FIN" operation to close an MPTCP connection?

    e. Does your implementation support the "Checksum" field (which

is negotiated in the MP_CAPABLE handshake)?

f. Any other comments or information?

4. Address management (ADD_ADDR and REMOVE_ADDR) [Section 3.4 RFC6824]

a. What is the status of your implementation?

b. Can your implementation do ADD_ADDRESS for addresses that appear *after* the connection has been established?

c. Any other comments or information?

5. Fast close (MP_FASTCLOSE) [Section 3.5 RFC6824]

a. What is the status of your implementation?

b. Any other comments or information?

3.4. Question 4: Fallback from MPTCP

Question 4 asks about action when there is a problem with MPTCP, for example due to a middlebox mangling MPTCP's signalling. The connection needs to fall back: if the problem is on the first subflow then MPTCP falls back to TCP, whilst if the problem is on an

additional subflow then that subflow is closed with a TCP RST, as discussed in [Section 3.6 RFC6824].

1. If the MP_CAPABLE option is removed by a middlebox, does your implementation fall back to TCP?

2. If the MP_JOIN option does not get through on the SYNs, does your implementation close the additional subflow?

3. If the DSS option does not get through on the first data segment(s), does your implementation fall back? (either falling back to MPTCP (if the issue is on the first subflow) or closing the additional subflow (if the issue is on an additional subflow))

4. Similarly, if the "DATA ACK" field does not correctly acknowledge the first data segment(s), does your implementation fall back?

5. Does your implementation protect data with the "Checksum" field in the DSS option [Section 3.3 RFC6824]? If the checksum fails (because the subflow has been affected by a middlebox), does your implementation immediately close the affected subflow (with a TCP RST) with the MP_FAIL Option? If the checksum fails and there is a single subflow, does your implementation handle this as a special case, as described in [Section 3.6 RFC6824]?

6. Does your implementation fall back to TCP by using an "infinite mapping" [Section 3.3.1 RFC6824] (so that the subflow-level data is mapped to the connection-level data for the remainder of the connection)?

7. Did you find any corner cases where MPTCP's fallback didn't happen properly?

8. Any other comments or information about fallback?

3.5. Question 5: Heuristics

Question 5 gathers information about heuristics: aspects that are not required for protocol correctness but impact the performance. We would like to document best practice so that future implementers can learn from the experience of pioneers. The references contain some initial comments about each topic.

1. Receiver considerations [S3.3.4, RFC6824]: What receiver buffer have you used? Does this depend on the retransmission strategy? What advice should we give about the receiver?

2. Sender considerations [S3.3.5, RFC6824]: How do you determine how much data a sender is allowed to send and how big the sender buffer is? What advice should we give about the sender?

3. Reliability and retransmissions [S3.3.6, RFC6824]: What is your retransmission policy? (when do you retransmit on the original subflow vs on another subflow or subflows?) When do you decide that a subflow is underperforming and should be reset, and what

do you then do? What advice should we give about this issue?

4. Port usage [S3.3.8.1, RFC6824]: Does your implementation use the same port number for additional subflows as for the first subflow? Have you used the ability to define a specific port in the Add Address option? What advice should we give about this issue?

5. Delayed subflow start [S3.3.8.2, RFC6824]: What factors does your implementation consider when deciding about opening additional subflows? What advice should we give about this issue?

6. Failure handling [S3.3.8.3, RFC6824]: Whilst the protocol defines how to handle some unexpected signals, the behaviour after other unexpected signals is not defined. What advice should we give about this issue?

7. Use of TCP options: As discussed in [Appendix A, RFC6824], the TCP option space is limited, but a brief study found there was enough room to fit all the MPTCP options. However there are constraints on which MPTCP option(s) can be included in packets with other TCP options - do the suggestions in Appendix A need amending or expanding?

8. What other heuristics should we give advice about? Any other comments or information?

3.6. Question 6: Security

Question 6 asks about Security related matters [Section 5 RFC6824].

1. Does your implementation use the hash-based, HMAC-SHA1 security mechanism defined in [RFC6824]?

2. Does your implementation support any other handshake algorithms?

3. It has been suggested that a Standards-track MPTCP needs a more secure mechanism. Do you have any views about how to achieve this?

4. Any other comments or information?

3.7. Question 7: IANA

Question 7 asks about IANA related matters.

1. Does your implementation follow the IANA-related definitions? [Section 8 RFC6824] defines: TCP Option Kind number (30); the sub-registry for "MPTCP Option Subtypes"; and the sub-registry for "MPTCP Handshake Algorithms"

2. Any other comments or information?

3.8. Question 8: Congestion control and subflow policy

Question 8 asks about how you share traffic across multiple subflows.

1. How does your implementation share traffic over the available paths? For example: as a spare path on standby ('all-or-nothing'), as an 'overflow', etc? Does it have the ability to send /receive traffic across multiple subflows simultaneously?

2. Does your implementation support "handover" from one subflow to another when losing an interface?

3. Does your implementation support the coupled congestion control defined in [RFC6356]?

4. Does your implementation support some other coupled congestion control (ie that balances traffic on multiple paths according to feedback)?

5. The MP_JOIN (Starting a new subflow) Option includes the "B" bit, which allows the sender to indicate whether it wishes the new subflow to be used immediately or as a backup if other path(s) fail. The MP_PRIO Option is a request to change the "B" bit - either on the subflow on which it is sent, or (by setting the optional Address ID field) on other subflows. Does your implementation support the "B" bit and MP_PRIO mechanisms? Do you think they're useful, or have another suggestion?

6. Any other comments or information or suggestions about the advice we should give about congestion control [S3.3.7 RFC6824] and subflow policy [S3.3.8 RFC6824]?

3.9. Question 9: API

Question 9 gathers information about your API. [RFC6897] considers the MPTCP Application Interface.

1. With your implementation, can legacy applications use (the existing sockets API to use) MPTCP? How does the implementation decide whether to use MPTCP? Should the advice in [Section 4, RFC6897] be modified or expanded?

2. The "basic MPTCP API" enables MPTCP-aware applications to interact with the MPTCP stack via five new socket options. For each one, have you implemented it? has it been useful?

   a. TCP_MULTIPATH_ENABLE?

   b. TCP_MULTIPATH_ADD?

   c. TCP_MULTIPATH_REMOVE?

   d. TCP_MULTIPATH_SUBFLOWS?

   e. TCP_MULTIPATH_CONNID?

3. Have you implemented any aspects of an "advanced MPTCP API"? ([Appendix A, RFC6897] hints at what it might include.)

4. Any other comments or information?

3.10. Question 10: Deployments, use cases and operational experiences

Question 10 takes the opportunity of this survey to gather some limited information about operational experiences and deployments. Any very brief information would be appreciated, for example:

1. What deployment scenarios are you most interested in?

2. Is your deployment on "the Internet" or in a controlled environment?

3. Is your deployment on end hosts or with a MPTCP-enabled proxy (at one or both ends?)?

4. What do you see as the most important benefits of MPTCP in your scenario(s)?

5. How extensively have you deployed and experimented with MPTCP so far?

6. MPTCP's design seeks to maximise the chances that the signalling works through middleboxes. Did you find cases where middleboxes blocked MPTCP signalling?

7. MPTCP's design seeks to ensure that, if there is a problem with MPTCP signalling, then the connection either falls back to TCP or removes the problematic subflow. Did you find any corner cases where this didn't happen properly?

8. Have you encountered any issues or drawbacks with MPTCP?

9. Any other comments or information?

3.11. Question 11: Improvements to RFC6824

1. Are there any areas where [RFC6824] could be improved, either in technical content or clarity?

2. Any other issues you want to raise?

4. IANA Considerations

This document makes no request of IANA.

5. Security Considerations

This survey does not impact the security of MPTCP, except to the extent that it uncovers security issues that can be tackled in a future version of the protocol.

6. Acknowledgements

7. Normative References

[RFC6356] Raiciu, C., Handley, M., and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols", RFC 6356, October 2011.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure,

"TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

Author's Address

Philip Eardley
BT