Authors: J. Kang   Q. Liang   S. Deng, Ed.
         Huawei   Huawei   Huawei

Subtype Capability Exchange During MPTCP Handshake

Abstract

Multipath TCP provides the ability to simultaneously use multiple paths between peers. MPTCP protocol defines seven subtypes in MPTCP v0 [RFC6824] and ten subtypes in MPTCP v1 [RFC8684] to differentiate message types and implement some additional functions during a session.

This draft proposes an enhancement to support Subtype Capability Exchange during MPTCP connection establishment in order to improve elastic scalability of MPTCP protocol. It includes: 1) requirements for which this kind of capability exchange during handshake is important for a MPTCP session; 2) a typical flow for Subtype Capability Exchange between peers; 3) a feasible solution on protocol design is suggested.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. 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 RFC 2119 [RFC2119].

1.2. Background

Table 1 lists all subtypes that have been specified in current MPTCP
versions. Besides version negotiation, MPTCP peers can not interact
with each other on the granularity of subtype capability. This
feature may cause inflexible protocol extension. For example, if a
new message type A is added in future extension, a higher version
should be released to import it and a new subtype may need to be
allocated. Another case is that if a sender does not know the
subtypes supported by a receiver in a MPTCP session, as a result,
invalid data packets may been sent from the sender during data
transmission and the receiver will discard it which causes system
overhead on receiver side.

Value	Symbol	Name	MPTCPv0	MPTCPv1
0x0	MP_CAPABLE	Multipath Capable	Supported	Supported
0x1	MP_JOIN	Join Connection	Supported	Supported
0x2	DSS	Data Sequence Signal (Data ACK and Data Sequence Mapping)	Supported	Supported
0x3	ADD_ADDR	Add Address	Supported	Supported
0x4	REMOVE_ADDR	Remove Address	Supported	Supported
0x5	MP_PRIO	Change Subflow Priority	Supported	Supported
0x6	MP_FAIL	Fallback	Supported	Supported
0x7	MP_FASTCLOSE	Fast Close	Supported	Supported
0x8	MP_TCPRST	Subflow Reset	/	Supported
0xf	MP_EXPERIMENTAL	Reserved for Private Use	/	Supported

Table 1: Overview MPTCP Subtypes

This document suggests a new function of Subtype Capability Exchange during MPTCP handshake in the scenario that MPTCP peers in a session support same MPTCP protocol version but with different subtype sets.

2. One Typical Flow

Figure 1 illustrates a typical flow for this Subtype Capability Exchange during MPTCP connection setup. The field of Subtype Capability is used to indicate whether these subtypes are supported by the sender, for example, Host A Subtype Capabilities indicates the status of the subtypes on Host A and Host B Subtype Capabilities indicates that on Host B. Through the transmission of this information between both parties, a sender can determine whether a message can be properly processed by its receiver and only send the message that can be supported by the receiver during data transmission.

```
            Host A                              Host B
      ------------------------             ----------
      Address A1     Address A2             Address B1
      ----------     ----------             ----------
           |              |                      |
           |       SYN + Host A Subtype Capability      |
           |------------------------------------------->|
           |                                   Determine and
           |                               Cache the capabilities
           |                                    of Host A
           |<-------------------------------------------|
           |       SYN/ACK + Host B Subtype Capability   |
   Determine and Cache the                             |
    capabilities of Host B                             |
           |                        ACK                 |
           |------------------------------------------->|
           |              |                      |
           |              |        SYN + MP_JOIN        |
           |              |----------------------------->|
           |              |<-----------------------------|
           |              |        SYN/ACK + MP_JOIN     |
           |              |                      |
           |              |        ACK + MP_JOIN        |
           |              |----------------------------->|
           |              |<-----------------------------|
           |              |             ACK            |
           |              |                      |
           |     Data Transmission(with subtype messages)   |
           |<-------------------------------------------->|
           |              |                      |
           |              | Data Transmission(with subtype |
           |              |<----------------------------->|
           |              |             messages)       |
           |              |                      |
```
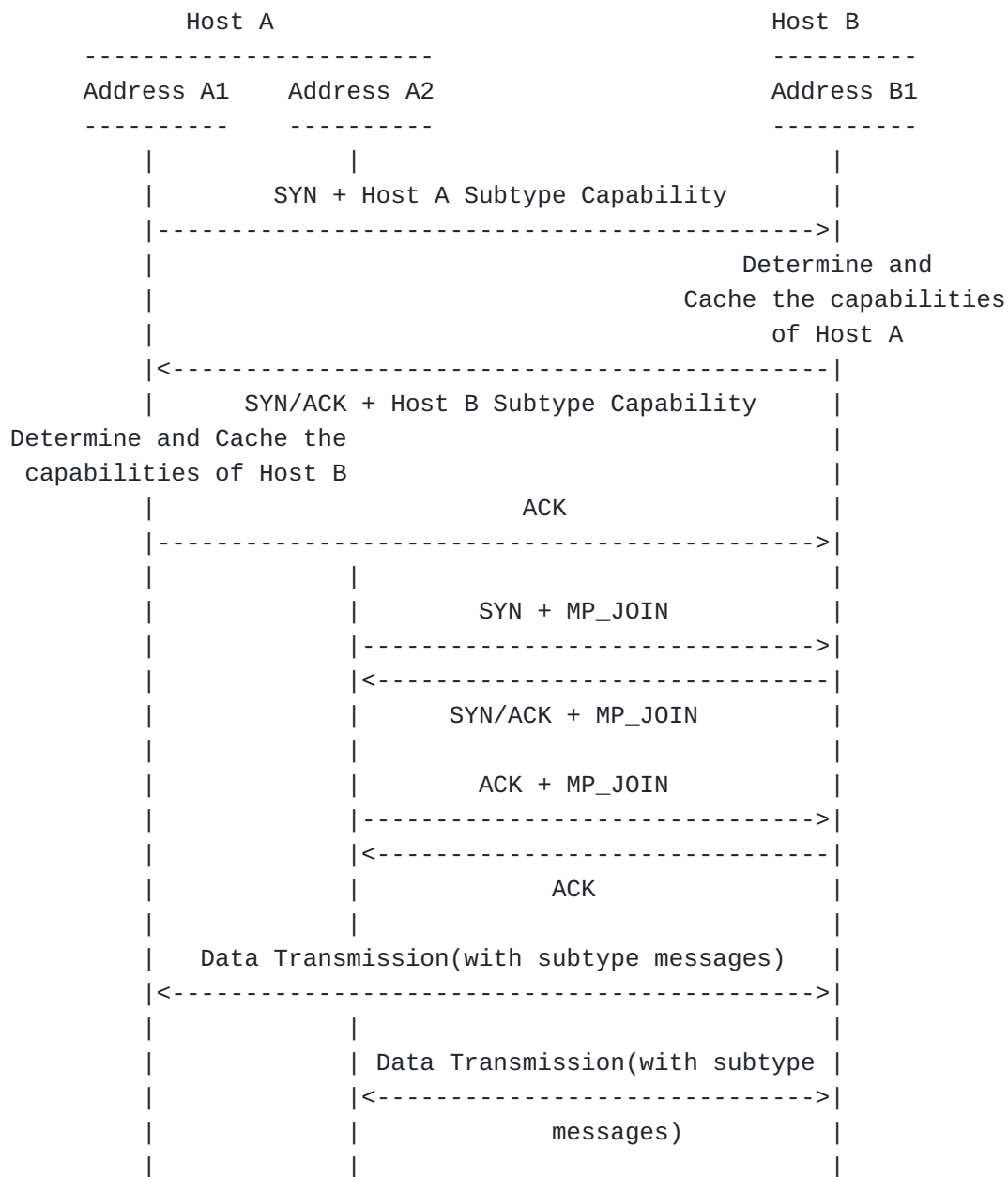
Figure 1: MPTCP Subtype Capability Exchange

In practice, another possible implementation is as follows: after
receiving the subtype capability information sent by Host A, Host B
determines the common subtype sets supported by both parties, and
returns this common subtype sets in the reponse. Host A caches this
common subtype sets locally. In data transmission phase, Host A
sends the specified subtype messages to Host B that are included in
the common subtype sets. As an alternative solution, its protocol
design on MPTCP will be considered and updated in later versions.

3. Protocol Implementation

This document describes one solution on the modifications to MPTCP
protocol to support this mechanism.In this solution, MP_CAPABLE
option is used and extended to add bits to carry subtype
capabilities information. There should be other possible solutions
that can be defined in subsequent discussions.

3.1. Carrying Subtype Capabilities in MP_CAPABLE Option

In Figure 2, a 32-bit "OptionSupported" is added to MP_CAPABLE
option to indicate whether the subtypes are supported by the sender.

```
                      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|Version|A|B|C|D|E|F|G|H|
 +---------------+---------------+-------+-------+---------------+
 |                   Option Sender's Key (64 bits)              |
 |                      (if option Length > 4)                  |
 |                                                              |
 +--------------------------------------------------------------+
 |                   Option Receiver's Key (64 bits)            |
 |                      (if option Length > 12)                 |
 |                                                              |
 +-----------------------------+--------------------------------+
 |                    OptionSupported (32 bits)                 |
 +-----------------------------+--------------------------------+
 | Data-Level Length (16 bits) |  Checksum (16 bits, optional)  |
 +-----------------------------+--------------------------------+
```
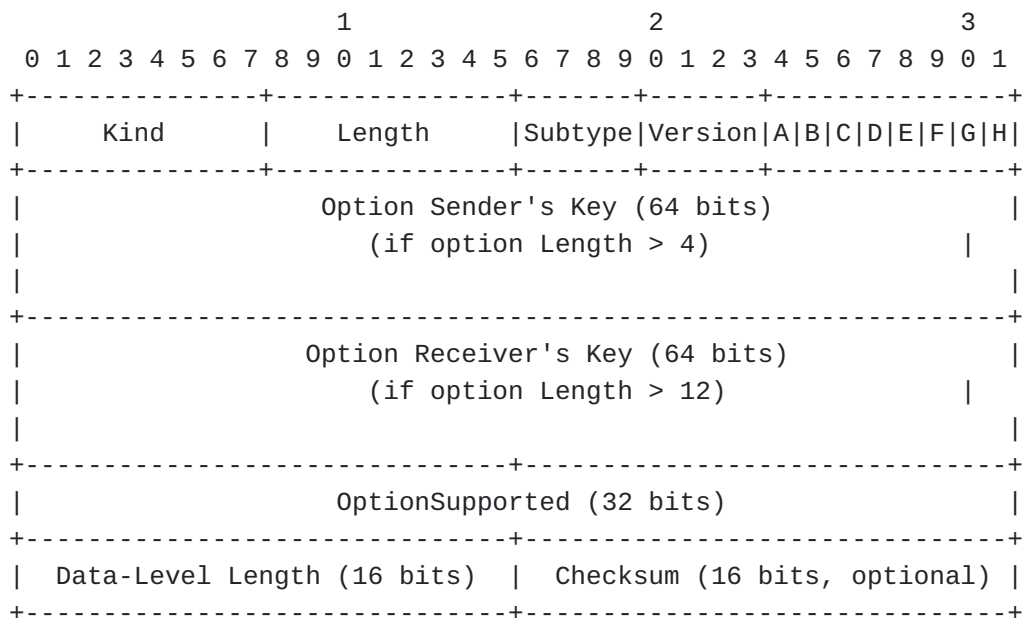
Figure 2: OptionSupported Format

For MPTCP v1, ten subtypes has been defined and applied in practice.
So the first 10-bits in OptionSupported field is used for indicating
whether these subtypes is supported by sender. The order is listed
below:

0: MP_CAPABLE

1: MP_JOIN

2: DSS

3: ADD_ADDR

4: REMOVE_ADDR

5: MP_PRIO

6: MP_FAIL

7: MP_FASTCLOSE

9: MP_TCPRST

10: MP_EXPERIMENTAL

11~31: Reserved for Future Use

Two values, that is 0 and 1, can be set to these bits in
OptionSupported field. The value of 0 indicates that the sender does
not support this subtype. The value of 1 indicates that the sender
supports this subtype.

4. Security Considerations

To be added.

5. IANA Considerations

To be added.

6. References

6.1. 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>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/
           RFC2119, March 1997, <https://www.rfc-editor.org/info/
           rfc2119>.

[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>.

6.2. Informative References

[RFC2629]

Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <https://www.rfc-editor.org/info/rfc2629>.

Authors' Addresses

Jiao Kang
Huawei

Email: jiao_kang2022@163.com

Qiandeng Liang
Huawei
No. 207, Jiufeng 3rd Road, East Lake High-tech Development Zone
Wuhan
China

Email: liangqiandeng@huawei.com

Shangling Deng (editor)
Huawei
D2-03,Huawei Industrial Base
Shenzhen
China

Email: dengshangling@huawei.com