MPTCP Version Negotiation
vs
MPTCP SubType Capability Exchange

draft-kang-tcpm-subtype-capability-exchange-in-mptcp-00.txt

Jiao Kang, Qiandeng Liang
IETF-110, TCPM WG, March, 2021
MPTCP Version Negotiation

Some underlying principles:

1. MPTCP v0 connection will likely be preferred over a TCP connection.
2. Currently, in a definite version, all subtype messages are mandatory and fixed.

Host A (Initiator, v0, v1), Host B (Receiver, v0) → the receiver will signal the version number it wishes to use

Host A (Initiator, v0, v1), Host B (Receiver, v1) → the receiver will signal the version number it wishes to use

Host A (Initiator, v1), Host B (Receiver, v0) → fallback to TCP
Requirements for Scalability and Performance Optimization

- 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.

- 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.

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

Table 1: Overview MPTCP Subtypes
MPTCP Subtype Capability Exchange Scenarios

• MPTCP peers in a session support same MPTCP protocol version including same subtype sets (covered by these slides)

• MPTCP peers in a session support same MPTCP protocol version but with different subtype sets (covered by these slides)

• MPTCP peers in a session support different MPTCP protocol version including same subtype sets (TBD)

• MPTCP peers in a session support different MPTCP protocol version with different subtype sets (TBD)
A typical flow for Subtype Capability Exchange between two endpoints

Host A
Address A1
SYN (Host A Subtype Capabilities)
SYN/ACK (Host B Subtype Capabilities)
Determine and Cache the capabilities of Host B
ACK
Data Transmission (with some Subtype Message)

Host B
Address B1
Determine and Cache the capabilities of Host A
ACK
SYN + MP_JOIN
SYN/ACK + MP_JOIN
ACK + MP_JOIN
ACK
Data Transmission (with some Subtype Message)
One Possible Solution on Protocol Design

Carrying Subtype Capabilities in MP_CAPABLE Option

Figure 2: OptionSupported Format
Next

• Is this requirement/scenario interesting and useful?
• If above is yes, go ahead to complete the draft.