Network Working Group
Internet Draft
Intended status: Experimental
Expires: October 2015

Changqiao Xu
BUPT
Hui Huang
BUPT
Hongke Zhang
BUPT
Chunshan Xiong
Huawei Technologies Co., Ltd
Lei Zhu
Huawei Technologies Co., Ltd
April 3, 2015

# Multipath Transmission Control Protocol (MPTCP) Partial Reliability Extension
draft-xu-mptcp-prmp-00.txt

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on October 3, 2015.

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 carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

This memo presents an extension to the Multipath Transmission Control Protocol (MPTCP) that allows MPTCP endpoints in which case both sender side and receiver side support this function to provide partially reliable data transmission service to the upper layer applications. In order to achieve the above goal, this memo extents MPTCP by adding two new subtypes which are expressed as "PR_CAPABLE" and "ACK_NOTIFY" and the corresponding processes are also introduced. The extension can provide the backward-compatibility with MPTCP if the new features are not available.

Table of Contents

1. Introduction.....3
   1.1. Motivation.....3
   1.2. Overview of PR-MPTCP.....3
2. Conventions.....3
3. New Options.....3
   3.1. PR_CAPABLE Option.....4
   3.2. ACK_NOTIFY Option.....5
4. PR-MPTCP Workflow.....6
   4.1. Connection Initialization.....6
   4.2. Process of Forced Acknowledged Packets.....7
      4.2.1. Sender Side Implementation.....7
      4.2.2. Receiver Side Implementation.....8
5. Impact on Congestion Control Algorithm.....8
6. Security Considerations.....8

7. IANA Considerations.....8
8. References.....9
8.1. Normative References.....9
8.2. Informative References.....9
9. Acknowledgments.....9

## 1. Introduction

PR-MPTCP is the extension of MPTCP which can provide partially reliable services when both sender side and receiver side support this function. It can meet the requirements of real time transport services, such as streaming media.

### 1.1. Motivation

As an extension of traditional TCP for multipath operation with multiple addresses, Multipath Transmission Control Protocol (MPTCP) provides a reliable and in-order delivery service to the upper layer applications. However, with the development of internet, more and more applications seek for the mechanisms which can transport data with different reliability level in different ways. This memo intends to fill the gap with the extension of MPTCP.

### 1.2. Overview of PR-MPTCP

This demo mainly describes the following two changes to MPTCP to provide the partial reliable function:

1. The negotiation of partial reliability function in the initialization phase.

2. The maintaining of partial reliable processing, including sender side and receiver side cooperation.

## 2. Conventions

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].

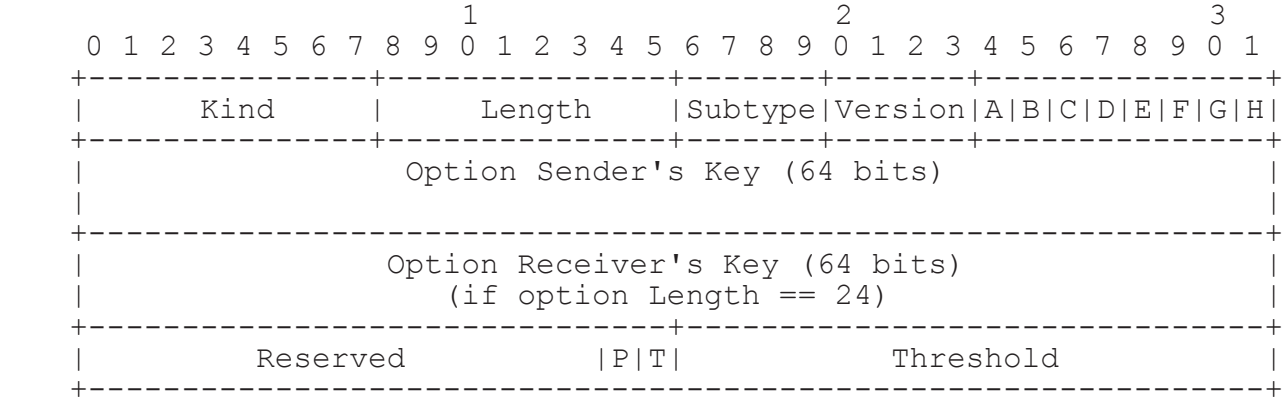
## 3. New Options

Similar with the manner MPTCP enhances TCP functionality, PR-MPTCP extends MPTCP by utilizing the TCP Options field and by defining new options. Two new subtype MPTCP options are introduced, named "PR_CAPABLE" and "ACK_NOTIFY", which are described next.

### 3.1. PR_CAPABLE Option

Before transmitting any data, the communicating endpoints exchange information to negotiate supported functions, including the partial reliability function. This function can be used only if both communicating sides are partial reliability capable. In order to negotiate the availability of the partially reliable mechanism during connection establishment, we define a new subtype of MPTCP option named PR_CAPABLE. This subtype extends the MP_CAPABLE option fields described in MPTCP by appending partial reliability parameters setting information to the end of the MP_CAPABLE option data.

The detailed format of PR_CAPABLE option is as follows:




The new fields include two flags P and T and Threshold, which will be described next. Naturally the value in the Length field of the PR-MPTCP header will also be larger with 4 than that in the similar MPTCP header.

P: 1 bit

This flag bit indicates whether the sender wishes to use the packet-based partial reliability transmission or not. By setting P to 1, the sender requests the receiver to enable packet-based partial reliability transmission. If P equals 0, the receiver performs no action and classic MPTCP transmission takes place by default.

T: 1 bit

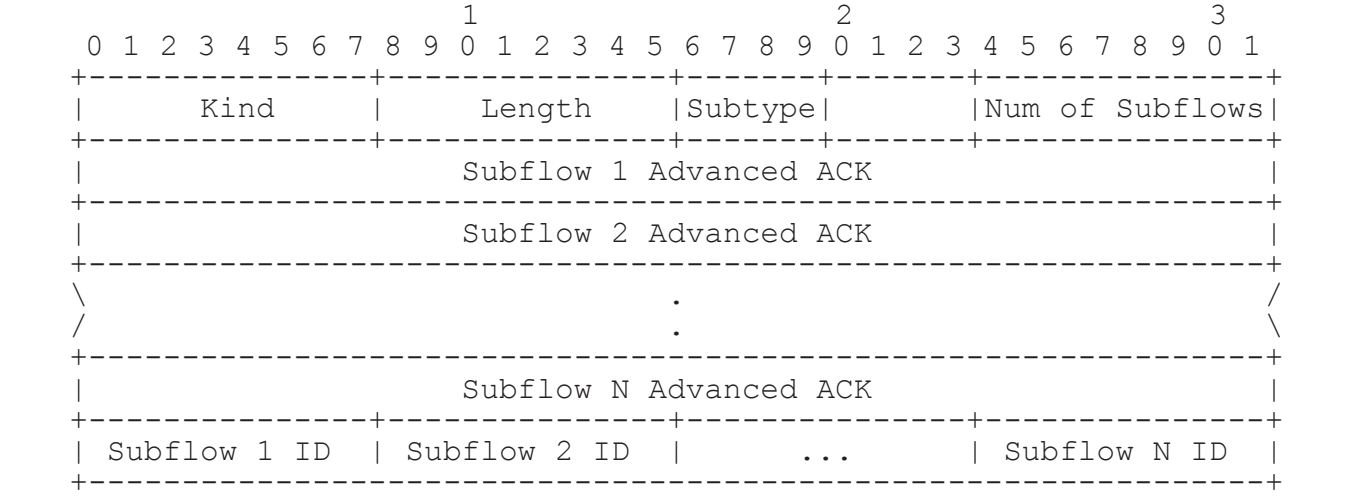
This flag bit indicates whether the sender wishes to use the time-based partial reliability transmission. By setting T to 1, the sender requests the receiver to enable time-based partial reliability transmission. If T equals 0, classic MPTCP transmission takes place by default.

Threshold: 16 bits

The meaning of the value in this field depends on the value of flag P and T. If P flag is set to 1, the value in this field is used as the maximum number of transmission attempts for each packet. If T flag is set to 1, the value in this field is used as the maximum delay of transmission for each packet, expressed in milliseconds.

## 3.2. ACK_NOTIFY Option

The detailed format of ACK_NOTIFY option is as follows:




When the sender identifies a packet exceeding its maximum number of retransmission attempts or its delay limit as set by the Threshold, the packet is not transmitted anymore. The sender has to inform the receiver of the not-transmitted packet's sequence number, so it will not wait for its arrival anymore. This is performed by sending a forced acknowledgement with the next data packet with a new option ACK_NOTIFY in it. Upon receiving this information, the receiver updates its local ACK point and then sends a new ACK as response. When receiving this ACK packet indicating the new ACK point by passing the not-transmitted packet, the sender releases this packet from the sender buffer just like when it is received successfully. This process can involve more than one packet on multiple subflows.

Num of Subflows: 8 bits

This field indicates that how many subflows need to send advanced ACK notifications.

Subflow # Advanced ACK: 32 bits

This field indicates the sequence number of the packet to be forced acknowledged in subflow #. If more than one data packets need to be forced acknowledged in the same subflow, only the largest sequence number will be indicated in this field.

Subflow # ID: 8 bits

The address ID of the destination to which the notification should be sent.

If a single option can't contain all forward ACK information for all subflows due to the limitation of the TCP option size, you SHOULD wait for another chance to send these forward information.

# 4. PR-MPTCP Workflow

## 4.1. Connection Initialization

The PR-MPTCP communication initiator sends PR_CAPABLE option instead of sending the MP_CAPABLE option, as in the classic MPTCP connection initialization case. If the other side is not partial reliability capable, but supports MPTCP, the connection initialization will follow the regular MPTCP connection establishment process. If the other side is not MPTCP capable, TCP connection establishment will be performed instead.

At the beginning, the initiator SHOULD add the PR_CAPABLE option into the SYN request packet to declare that it is PR-MPTCP capable.

Upon receipt of a SYN that contains PR_CAPABLE option, the receiver SHOULD send a SYN/ACK containing PR_CAPABLE, if it is PR-MPTCP capable; or ignore the PR_CAPABLE option in the SYN and continue to act as MPTCP does, if it is not PR-MPTCP capable but MPTCP capable.

If a connection initiator which is PR-MPTCP capable received a SYN/ACK containing PR_CAPABLE option, the transmission will adopt the partially reliable approach. Otherwise, if the received SYN/ACK does not contain PR_CAPABLE option (maybe the other side is not PR-MPTCP capable or the PR_CAPABLE option is stripped off by the middle

box) but a MP_CAPABLE option contained, the connection will fall back to MPTCP connection, or TCP connection will be used.

The other process such as exchange of address information are the same with the operation mentioned in [RFC6824].

### 4.2. Process of Forced Acknowledged Packets

In the implementation of partially reliable transmission, the sender gives up the retransmission of over-limited packets to ensure the requirements of some upper layer applications.

#### 4.2.1. Sender Side Implementation

Apart from the standard processing in MPTCP, in order to achieve the partial reliability goal, the following extra actions MUST be taken:

1) The sender maintains a mandatory acknowledgment points (Advanced ACK Point) for each subflow and MUST update it when the judgment sub-processes determine to advance the cumulative ACK point, this means that all data with sequence numbers less than the Cumulative ACK Point is regarded as having been ACKed;

2) When receiving a SACK, the sender side firstly processes the SACK as MPTCP does, namely updates the Cumulative ACK Point;

3) Then the judgment sub-processes SHOULD compare the cumulative ACK with Advanced ACK Point. If Advanced ACK Point is less than the cumulative ACK, then update Advanced ACK Point to be equal to the cumulative ACK;

4) After processing SACK, the sender SHOULD try to advance the Advanced ACK Point for each subflow, if Advanced ACK Point is greater than the cumulative ACK, the judgment sub-processes SHOULD inform the receiver of updating its local cumulative ACK Point by sending a packet with ACK_NOTIFY option;

5) After sending a packet with ACK_NOTIFY option, the sender MUST be sure that at least a RTX timer is running, if the timer timeout occurs, the packet with the ACK_NOTIFY option will be retransmitted.

The default policy to send ACK_NOTIFY option in this document is bundling the notification of each subflow in one TCP option space as long as the total size of the option would not exceed the limitation

of the TCP option size, in addition, the total size of the packet SHOULD NOT exceed the MTU.

#### 4.2.2. Receiver Side Implementation

When receiving a packet with an ACK_NOTIFY option, the receiver compares the local Cumulative ACK Point with the notified ACK contained in ACK NOTIFY option for the corresponding subflow, and releases the forced acknowledged packets from the receiver buffer.

When the forced acknowledged packets arrive at the receiver side, these packets SHOULD be treated as duplicate packets, and the receiver processes then as MPTCP does, (i.e. drop).

## 5. Impact on Congestion Control Algorithm

When a packet is forcedly acknowledged, it SHOULD be treated as loss and trigger the congestion control algorithms. If more than one packets are forcedly acknowledged, the congestion window adjustment SHOULD be triggered only once in a short specified duration, such as a RTT.

## 6. Security Considerations

This memo develops no new security scheme for MPTCP. PR-MPTCP share the same security issues discussed in [RFC6824] with MPTCP.

## 7. IANA Considerations

| Value | Symbol | Name | Reference |
|---|---|---|---|
| 0xa | PR_CAPABLE | Multipath Partial Reliability Capable | This document, Section 3.1 |
| 0xb | ACK_NOTIFY | Acknowledgement Notification | This document, Section 3.2 |

Table 1: MPTCP Option Subtypes

The 4-bit MPTCP subtype sub-registry ("MPTCP Option Subtypes" under the "Transmission Control Protocol (TCP) Parameters" registry) was defined in [RFC6824]. This document defines two additional subtype (PR_MPCABLE and ACK_NOTIFY). The updates are listed in the above table.

# 8. References

## 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

## 8.2. Informative References

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

# 9. Acknowledgments

This Internet Draft is the result of a great deal of constructive discussion with several people, notably Man Tang, Jiuren Qin, and Peng Wang.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Changqiao Xu
Beijing University of Posts and Telecommunications
Institute of Network Technology, No. 10, Xitucheng Road,
Haidian District, Beijing
P.R. China

Email: cqxu@bupt.edu.cn

Hui Huang
Beijing University of Posts and Telecommunications
Institute of Network Technology, No. 10, Xitucheng Road,
Haidian District, Beijing
P.R. China

Email: hh1126@bupt.edu.cn

Hongke Zhang
Beijing University of Posts and Telecommunications
Institute of Network Technology, No. 10, Xitucheng Road,
Haidian District, Beijing
P.R. China

Email: hkzhang@bupt.edu.cn

Chunshan Xiong
Huawei Technologies Co., Ltd
Science and Technology Demonstration Garden, No. 156, Zhongguancun
North Qing Road,
Haidian District, Beijing
P.R. China

Email: sam.xiongchunshan@huawei.com

Lei Zhu
Huawei Technologies Co., Ltd
Science and Technology Demonstration Garden, No. 156, Zhongguancun North Qing Road,
Haidian District, Beijing
P.R. China

Email: lei.zhu@huawei.com