

MPTCP
Internet Draft
Intended status: Informational
Expires: September 22, 2018

J. Zuo
J. Zhu
W. Liu
Huawei
March 21, 2018

A Path-Aware Scheduling Scheme for Multipath Transport Protocols
draft-zuo-mptcp-scheduler-01.txt

Abstract

Scheduling schemes play a key role in the performance of multipath transport protocols. A path-aware scheduling scheme for latency-sensitive applications is proposed, which always schedules data in the path with the lowest One-Way Latency (OWL). The OWLs of all paths are obtained by employing the redundant transmission periodically.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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/1id-abstracts.html>

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

Copyright and License Notice

Copyright (c) 2017 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.

Table of Contents

1.	Introduction	2
2.	Acronyms and Terminology	3
3.	A New Path-Aware Scheduling Scheme	3
3.1	The definition of OWL	3
3.2	The Operations of the Path-Aware Scheduling Scheme	4
3.2.1	Initialization	4
3.2.2	Data scheduling	4
3.2.3	Periodically Redundant Transmission	4
4.	Implementation Considerations	5
4.1	Packet Format	5
4.2	Negotiation	6
4.3	The Steps for OWL Calculation	6
5.	Security Considerations	7
6.	References	7
6.1	Normative References	7
6.2	Informative References	7
	Author's Addresses	7

[1.](#) Introduction

For latency-sensitive applications, such as games, latency is the primary goal to make the interactive data exchanged in time. A suitable scheduling scheme will influence the latency performance of multipath transport protocols [[RFC6824](#)], since the packets with consecutive sequence numbers may arrive at the receiver out of order due to the different RTTs of multiple paths. Lots of scheduling schemes are proposed to select the path with the lowest latency to send data. RTT is commonly used as the condition for scheduling. However, it's found that the delays of the forward path and the reverse path are usually different in many cases, such as the asymmetric uplink delay and the downlink delay in wireless environment. Especially, the congestion may not happen in the forward path and the reverse path at the same time, resulting in the different queue delays. Therefore, scheduling data on the path with the lowest RTT cannot guarantee the lowest interactive latency required by the latency-sensitive applications.

A new path-aware scheduling scheme for multiple transport protocols

is proposed, which considers the One-Way Latency (OWL) as a scheduling condition. The data will be always sent in the path with the lowest OWL.

It is intended that the scheduling scheme presented in this draft can be applied to other multipath transport protocols, such as alternative multipath extensions to TCP [[RFC793](#)], UDP, QUIC, or indeed any other multipath protocols. However, for the purposes of example, this document will, where appropriate, refer to the MPTCP [[RFC6182](#)].

2. Acronyms and Terminology

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. A New Path-Aware Scheduling Scheme

A new path-aware scheduling scheme is proposed to always send data in the path with the lowest OWL.

3.1 The definition of OWL

OWL is defined as the latency of a data successfully delivered from the sender to the receiver, which is calculated as

$$\text{OWL}(i) = T_{\text{recv}}(i) - T_{\text{send}}(i),$$

where i means the i -th path, T_{recv} is the time when the data arrives at the receiver and T_{send} is the time when the data is sent from the sender.

Compared to the One-Way Delay (OWD) defined in [[RFC7679](#)], OWL can be deemed as

$$\text{OWL}(i) = \text{OWD}(i) + dT,$$

where dT means the time difference caused by the absolute clock time at the sender and the receiver. When the scheduler tries to select the path with the lowest latency to send data, it will compare the latencies of multiple paths. Take two paths for example,

$$d\text{OWL} = \text{OWL}(1) - \text{OWL}(2),$$

where $d\text{OWL}$ indicates the OWL difference of path 1 and path 2. If $d\text{OWL} > 0$, the path 2 is selected or else the path 1 is selected. We could see that dT will not influence the value of $d\text{OWL}$, since it will be

subtracted when path 1 and path 2 are connected between the same sender and the same receiver. Hence, OWL is more suitable than OWD as the scheduling condition in multipath environment, since it does not need to consider the issue of time synchronization.

3.2. The Operations of the Path-Aware Scheduling Scheme

3.2.1. Initialization

As in [RFC6824](#), MPTCP first sets up the connection in the primary path and then subsequently sets up the connections of other paths. After the handshake stage of the primary path, data is immediately sent in this path. Then when the connection of the second path is built up, the OWLs of these two paths could be obtained. The following data could be scheduled in the path with the lowest OWL. However, network is unstable at the moment, the value of the obtained OWL may change in the next round of data transmission, especially when there are other flows co-existed or packet loss is very high. For avoiding the frequent switching amongst the multiple paths due to the variant instant OWLs, the data are scheduled redundantly in the multiple paths for a period of time [[CONEXT17](#)]. The redundant scheduling means the same data are sent in all paths concurrently [[MPTCP.org](#)]. A Smoothed OWL (SOWL) is defined as the effective OWL of a path, as referred to the smoothed RTT, which is defined in [[RFC793](#)]:

$$\text{SOWL} = (\text{ALPHA} * \text{SOWL}) + ((1-\text{ALPHA}) * \text{OWL}).$$

Alternatively, other methods which could calculate the effective OWL during the period of time can be considered. The period of time could be set to a fixed number, e.g. 1s, or $\max\{200\text{ms}, \text{SRTT}\}$.

3.2.2. Data scheduling

After the effective OWLs of all paths are obtained, the sender will schedule the following data from application to the path with the lowest OWL.

3.2.3. Periodically Redundant Transmission

Once the path with the lowest OWL is selected to send the data henceforth, and no data will be scheduled to the other paths. The result is that there is no chance to know the future OWLs of the other paths. For making the scheduling more responsive to the variant network environment and always sending data in the path with the lowest OWL, sending probe packets in other paths may be one of the potential solutions. However, it introduces extra packets which

consume extra bandwidth of the path. Therefore, the periodically redundant transmission is proposed, which sends data redundantly for obtaining the effective OWLs of all paths every a certain period of time (e.g. 10s). Figure 1 shows the process of the periodically redundant transmission.

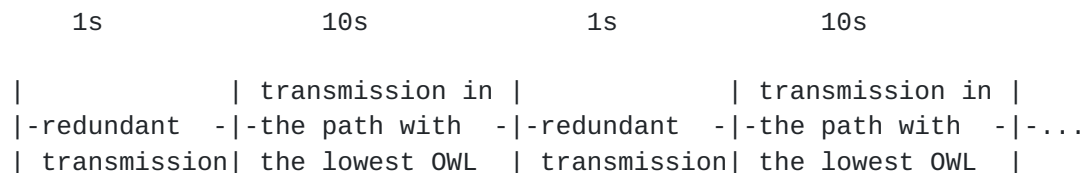


Figure 1: Periodically redundant transmission.

On the one hand, the periodically redundant transmission could obtain the OWLs of all paths at the same time without introducing extra packets; on the other hand, the same data are transmitted concurrently in all the paths, which could guarantee the lowest delivering time as long as the data arrives at the receiver no matter in which path it is sent.

Once the OWLs of all paths are updated, the following data are scheduled as described in [Section 3.2.2](#). During the data transmission in the path with the lowest OWL, the OWL of the selected path is continually updated. Once the updated OWL has increased so much that it may not be the lowest OWL any more, the redundant transmission is immediately activated to obtain the updated OWLs of all paths, as shown in Figure 2.

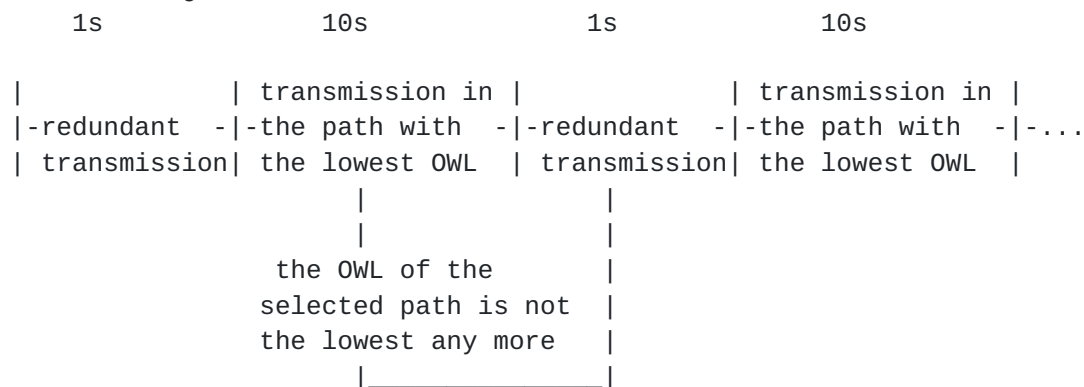


Figure 2: Redundant transmission activated when the OWL of the selected path is not the lowest any more.

4. Implementation Considerations

4.1 Packet Format

As described in [Section 3.1](#), the calculation of the OWL requires the time when the data is sent from the sender and the time when the data

arrives at the receiver. Since the sender needs to compare the OWLs of all the paths, the sender is assigned executing the OWL calculation in this draft. A new MPTCP option, called as MP_OWL Option, is defined to carry the timestamp of data receiving at the receiver. The format of MP_OWL option is shown in Figure 3, where the value of 'Kind' is '30' (decimal) [[RFC6824](#)].

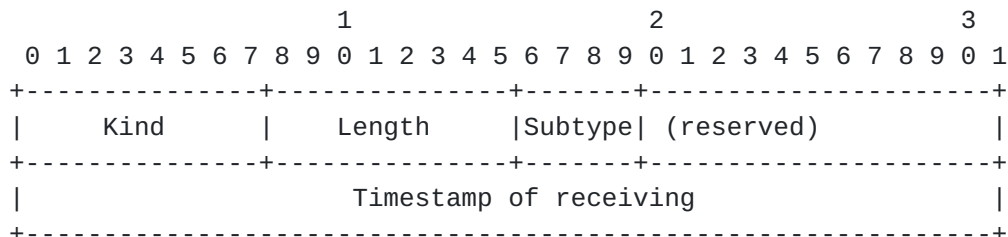


Figure 3: MP_OWL Option

The value of 'subtype' in MP_OWL option could be assigned as '0x8', since '0x0' through '0x7' have been used [[RFC5226](#)]. Table 1 gives the explanation of the value of '0x8'.

Value	Symbol	Name
0x8	MP_OWL	One-Way Latency

Table 1: The Subtype of MP_OWL option

4.2 Negotiation

The sender and the receiver need to negotiate with each other to see whether MP_OWL option is supported or not.

The details are to be defined.

4.3 The Steps for OWL Calculation

- 1) The sender sends each data and remember the sequence number as well as the sending time T_{send} of the data;
- 2) When the receiver receives the data, it responses an ACK with an MP_OWL option added, which carries the receiving time T_{recv} of the data;
- 3) When ACK gets back to the sender, the sender fetches the receiving time T_{recv} of the data from the ACK and subtracts the sending time T_{send} it has remembered to get the value of OWL, where $\text{OWL} = T_{\text{recv}} - T_{\text{send}}$.

5. Security Considerations

TBD.

6. References

6.1. Normative References

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

6.2. Informative References

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, [RFC793](#), DOI 10.17487/RFC0793, September 1981, <<http://www.rfc-editor.org/info/rfc793>>.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and Bonaventure, O., "TCP Extensions for Multipath Operation with Multiple Addresses", [RFC 6824](#), DOI 10.17487/RFC6824, January 2013, <<http://www.rfc-editor.org/info/rfc6824>>.

[RFC6182] Ford, A., Raiciu, C., Handley, M., Barre, S., Iyengar, J., "Architectural Guidelines for Multipath TCP Development", [RFC 6182](#), DOI 10.17487/RFC6182, March 2011, <<http://www.rfc-editor.org/info/rfc6182>>.

[RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., Morton, A., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", [RFC 7679](#), DOI 10.17487/RFC7679, January 2016, <<http://www.rfc-editor.org/info/rfc7679>>.

[RFC5226] Narten, T., Alvestrand, H., "Guidelines for Writing an IANA Considerations Section in RFCs", [RFC 5226](#), DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.

[MPTCP.org] Multipath TCP - Linux Kernel Implementation, <<http://multipath-tcp.org/pmwiki.php/Users/ConfigureMPTCP>>.

[CONEXT17] Coninck, Q. D., Bonaventure, O., "Multipath QUIC: Design and Evaluation", December 2017, In Proceedings of CoNEXT'17: The 13th International Conference on emerging Networking EXperiments and Technologies.

Author's Addresses

Jing Zuo
Huawei Technologies
Bantian, Longgang District,
Shenzhen 518129 P.R. China
EMail: jing.zuo@huawei.com

Jianjian Zhu
Huawei Technologies
Bantian, Longgang District,
Shenzhen 518129 P.R. China
EMail: zhujianjian1@huawei.com

Wei Liu
Huawei Technologies
Bantian, Longgang District,
Shenzhen 518129 P.R. China
EMail: liuwei57@huawei.com

