

Article

# Towards Adaptive Multipath Managing: A Lightweight Path Management Mechanism to Aid Multihomed Mobile Computing Devices

Yuanlong Cao <sup>1</sup>, Mario Collotta <sup>2</sup>, Siyi Xu <sup>1</sup>, Longjun Huang <sup>1</sup>, Xueqiang Tao <sup>3</sup> and Zhichao Zhou <sup>4,\*</sup>

<sup>1</sup> School of Software, Jiangxi Normal University, Nanchang 330022, China; ylcao@jxnu.edu.cn (Y.C.); 201726705017@jxnu.edu.cn (S.X.); Huanglong98614@163.com (L.H.)

<sup>2</sup> Faculty of Engineering and Architecture, Kore University of Enna, 94100 Enna, Italy; mario.collotta@unikore.it

<sup>3</sup> Institute of Medical Support Technology, Academy of Systems Engineering, Tianjin 300161, China; taoxue2437@126.com

<sup>4</sup> AsiaInfo Technology Co., Ltd., Beijing 100193, China

\* Correspondence: zhouzc@asiainfo.com

Received: 17 December 2019; Accepted: 30 December 2019; Published: 4 January 2020



**Abstract:** With the large scale deployment of multihomed mobile computing devices in today's Internet, the Multipath TCP (MPTCP) is being considered as a preferred data transmission technology in the future Internet due to its promising features of bandwidth aggregation and multipath transmission. However, MPTCP is more likely to be vulnerable to the transmission quality differences of multiple paths, which cause a “hot-potato” out-of-order arrival of packets at the receiver side, and in the absence of a related approach to fix this issue, serious application level performance degradations will occur. In this paper, we propose MPTCP-LM<sup>3</sup>, a Lightweight path Management Mechanism to aid Multihomed MPTCP based mobile computing devices towards efficient multipath data transmission. The goals of MPTCP-LM<sup>3</sup> are: (i) to offer MPTCP a promising path management mechanism, (ii) to reduce out-of-order data reception and protect against receiver buffer blocking, and (iii) to increase the throughput of mobile computing devices in a multihomed wireless environment. Simulations show that MPTCP-LM<sup>3</sup> outperforms the current MPTCP schemes in terms of performance and quality of service.

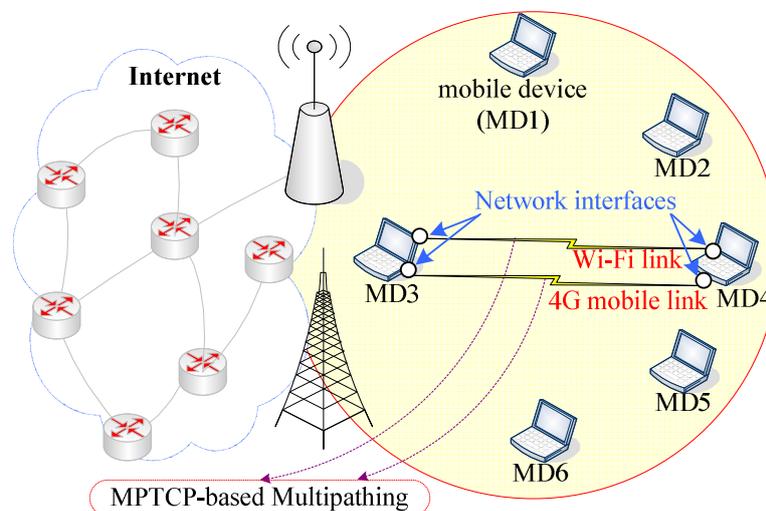
**Keywords:** wireless multipathing; multihomed mobile computing device; multipath TCP; receiver buffer blocking; path management

## 1. Introduction

In the past several years, wireless communications systems and networks have experienced a dramatic development [1]. These dramatic advances in wireless communications and networking technologies provide ubiquitous Internet connectivity to a mobile user [2]. Moreover, promoted by the increasing popularity of various wireless networks, mobile devices (i.e., smart phones, laptop computers, and netbooks) are already embedded with multiple Wireless Network Interface Cards (WNICs) and enabled multiple IP features [3,4]. With the multiple IP features (also known as multihoming), the devices can connect to the Internet via multiple IP addresses to provide enhanced and reliable Internet connectivity; they also can simultaneously use several IP addresses to spread data across several end-to-end IP paths. For example, the Apple iOS based products (i.e., iPhone and iPad) [5] and Samsung Galaxy mobile phones (i.e., S9 and S9 Plus) [6] can use both the Wi-Fi and cellular network links to download large files. Such multihomed devices can speed up the transmission

data rate and increase their throughput performance by making use of multiple wireless network links simultaneously, enabled by the emerging Multipath TCP (MPTCP) technology [7].

The MPTCP is an extension to traditional single path TCP that enables a multihomed device to allocate data traffic across multiple independent end-to-end paths concurrently through multiple network interfaces [8]. Figure 1 presents a typical MPTCP based multipath communication scenario that involves two multihomed Mobile Devices (MD3 and MD4), which communicate with each other by using both the Wi-Fi and 4G mobile links simultaneously. Such a multipathing feature would be beneficial to the mobile devices to aggregate bandwidth, increase throughput, and enhance communication robustness [9]. Furthermore, MPTCP inherits the standard socket Application Programming Interfaces (APIs) from TCP, which means an MPTCP based device can initiate and establish an MPTCP connection with other devices without making any modifications to today's Internet applications [10]. With its multipath transmission and backwards compatibility features, the MPTCP is being considered as a preferred transmission technology for data delivery over the current and future Internet [11].



**Figure 1.** A typical Multipath TCP (MPTCP) based multipath communication among multihomed mobile devices.

### 1.1. Motivation

Although applying MPTCP to the multihomed mobile computing devices towards multipath data transmission provides numerous potential benefits for data delivery [12–19], the MPTCP path management mechanism is very simple in RFC6824 [7], and this, in turn, leads to MPTCP being vulnerable to the path quality differences of the multiple paths [20]. With the regular path management mechanism, the MPTCP data traffic can be split and assigned to all the available end-to-end paths for concurrent multipath transmission. Such a full-MPTCP mode may be a useful way of providing a “best-effort” bandwidth aggregation service, but this, in turn, will determine data to be received out-of-order because asymmetric paths in an MPTCP connection are commonly encountered with different transmission characteristics. The out-of-order data arrival phenomenon can be especially problematic when applying MPTCP to a multihomed mobile computing device because today's mobile computing devices commonly have a poor receiver buffer, due to their very limited memory space. When a large number of out-of-order packets are maintained in the overloaded receiver buffer for reordering, buffer blocking will occur, and thereby, the MPTCP's performance will encounter a serious degradation (as we will discuss in more detail in the next section).

### 1.2. Related Works

In recent years, the growing interest in the MPTCP technology has resulted in thousands of peer reviewed publications. In this paper, we categorize the most relevant literature into two groups: MPTCP out-of-order packet arrival cases and MPTCP path management cases.

MPTCP out-of-order packet arrival cases: Xu et al. [9] presented a novel pipeline network coding based MPTCP solution to solve the out-of-order packet arrival problem and thus reduce the packet reordering overhead in multipath transmission. Yang et al. [12] introduced a new scheduling algorithm for MPTCP that mitigates jitter by transmitting packets out-of-order on different MPTCP sub-flows such that they can arrive in-order at the receiver. Alheid et al. [13] compared some different MPTCP congestion control mechanisms used in conjunction with the current TCP packet reordering solutions. Their results demonstrated that the performance of MPTCP could be influenced by the delay difference of multipath paths. Their study also identified combinations of congestion control and packet reordering solutions that give better aggregate goodput performance for different path delay differences. Ou et al. [16] designed a novel out-of-order transmission enabled congestion control and packet scheduling strategy for MPTCP. Xue et al. [17] presented a novel forward prediction based dynamic packet scheduling algorithm for MPTCP, by utilizing maximum likelihood estimation in TCP modeling and taking the packet loss rate and time offset into consideration. Le et al. [18] developed a new forward-delay based packet scheduling algorithm to solve the out-of-order received packet problem in the MPTCP based multipath transmission.

MPTCP path management cases: Wang et al. [20] investigated the performance of MPTCP in real-world Internet scenarios based on the NorNet Testbed. Their study revealed that path management and congestion control settings can present a significant impact on MPTCP's performance. Kim et al. [21] designed a novel receive buffer based path management mechanism, which managed multiple paths by using both the available receive buffer size and dissimilar characteristics of multiple paths. This path management mechanism was devoted to possibly reducing the out-of-order packet arrival problem and alleviating receive buffer blocking in MPTCP based multipath transmission. Oh et al. [22] proposed a feedback based path failure detection and buffer blocking protection approach for MPTCP. This approach aimed to (i) prevent the usage of underperforming sub-flows in the MPTCP, by using a path failure detection method, and (ii) prevent goodput degradation due to delay differences between paths, by detecting buffer blocking and closing underperforming sub-flows. Our previous work, MPTCP-La/E<sup>2</sup> [4], presented an LDDoS (Low-rate Distributed Denial-of-Service) attack aware energy saving oriented MPTCP extension aiming at optimizing the energy usage while still maintaining user's perceived quality of cloud multipathing services, by selecting and managing a subset of suitable paths for multipathing. Our previous work (PU)<sup>2</sup>M<sup>2</sup> [14] introduced a "Potentially Underperforming" (PU) concept to MPTCP and further designed a novel PU aware multipath management mechanism for MPTCP. The main goals of (PU)<sup>2</sup>M<sup>2</sup> were: (i) to prevent the use of potentially failed paths in multipath transmission and (ii) to manage multiple paths in MPTCP adaptively.

### 1.3. Contributions

In this paper, we propose MPTCP-LM<sup>3</sup>, a Lightweight, but highly promising Multipath Management Mechanism for MPTCP. The goals of MPTCP-LM<sup>3</sup> are (i) to possibly mitigate the out-of-sequence packet reception and reduce the delay of packet reordering in multipath transmission, (ii) to aggregate the bandwidth of multiple network paths adaptively and enhance the performance of MPTCP, and (iii) to preserve backward compatibility with the current Internet applications, as MPTCP does. We present simulation results that show that MPTCP-LM<sup>3</sup> can outperform the baseline MPTCP scheme in terms of data sending and receiving times, out-of-order data arrival, and throughput performance. The proposed MPTCP-LM<sup>3</sup> scheme makes fundamental contributions against the state-of-the-art in the literature, in the following aspects:

- (i) It reveals the fact that although each transmission path in MPTCP independently performs data delivery, a poorly performing path (with large delay) can lower the performance of MPTCP, by causing out-of-order data arrival and even receiver buffer blocking in multipath transmission.
- (ii) It introduces a lightweight multipath management mechanism for MPTCP, which allows a path to be removed and reconnected in multipath transmission adaptively, in order to reduce out-of-order data reception and protect against receiver buffer blocking in MPTCP.

Especially, it is worth pointing out that both our previous work (PU)<sup>2</sup>M<sup>2</sup> [14] and the MPTCP-LM<sup>3</sup> solution proposed in this paper were devoted to helping MPTCP adaptively manage multiple paths, by removing and reconnecting a transmission path accordingly. However, the (PU)<sup>2</sup>M<sup>2</sup> solution was mainly devoted to detecting path failure problems in MPTCP, while the MPTCP-LM<sup>3</sup> solution is mainly devoted to identifying which path would cause the performance degradations in multipath transmission. In addition, (PU)<sup>2</sup>M<sup>2</sup> prevented the usage of a PU path according to the path's delay variations, while in MPTCP-LM<sup>3</sup>, a path can be removed/reconnected according to not only the path's delay performance, but also the path's congestion window size. This help MPTCP-LM<sup>3</sup> address the limitation of the (PU)<sup>2</sup>M<sup>2</sup> solution, avoiding performing worse than the baseline MPTCP if a path has a high level of transmission capacity, but its delay performance becomes worse accidentally or suddenly.

## 2. Problem Statement

The asymmetric paths in a practical heterogeneous wireless environment with different transmission characteristics are more common and very sensitive to the variations of the real-time wireless condition. Because of this, the out-of-order data arrival will be an inevitable phenomenon in multipath transmission. What is worse, large numbers of out-of-order packets buffered in the constrained receiver buffer will result in the "hot-potato" receiver buffer blocking problem and thereby cause serious throughput performance degradations [22].

Several real-world measurement based studies on MPTCP reported that a huge path quality difference is a very important cause for MPTCP's low performance. Li et al. [23] performed a comprehensive large scale measurement on MPTCP in high speed mobility scenarios. Their study demonstrated that when Round-Trip Times (RTTs) in separate MPTCP paths differed, a significant out-of-order problem would arise, and the efficiency of MPTCP would be far from satisfactory. Chen et al. [24] presented a measurement based study of MPTCP performance with real wireless settings and background traffic. Their study revealed that the MPTCP performance was affected by path characteristic differences in terms of RTTs and loss rates. In this paper, we are especially interested in understanding how the performance of MPTCP is affected when path characteristics are diverse and caused by path failure. Thus, we present a group of reasonable simulations to answer this question. We also seek to answer why an effective multipath management mechanism is necessary for MPTCP.

To achieve the above purposes, we implemented a sample dual dumbbell network topology in Network Simulator 2 Version 2.35 (NS-2) [25], in which the MPTCP module [26] was included. The network topology is shown in Figure 2. An MPTCP sender and an MPTCP receiver were connected to each other through two independent end-to-end paths (A and B). The propagation delay of the edge links between the sender (or the receiver) and the routers was set to one microsecond with 100 Mbps of bandwidth. The bottleneck link between  $R_{1,0}$  and  $R_{1,1}$  ( $R_{2,0}$  and  $R_{2,1}$ ) was set with a 10 Mbps bandwidth and a 50 milliseconds propagation delay. Path B failed after 30 s of simulation time (we simulated Path B failure by bringing down the bottleneck link between routers  $R_{2,0}$  and  $R_{2,1}$ ). The total simulation time was 120 s with infinite FTP flows. Table 1 shows the parameters used in the simulations. Moreover, we performed two test cases named Case 1 and Case 2, which were examined as follows to investigate the effect of path transmission stability differences on the performance of MPTCP.

Case 1. The sender turns on its first interface and turns off its second interface at the beginning of simulation, which means always only Path A is used in MPTCP for data transmission.

Case 2. The sender turns on both of its interfaces, which means both Paths A and B are used in MPTCP. However, Path B suffered from network failure after 30 s of simulation time.

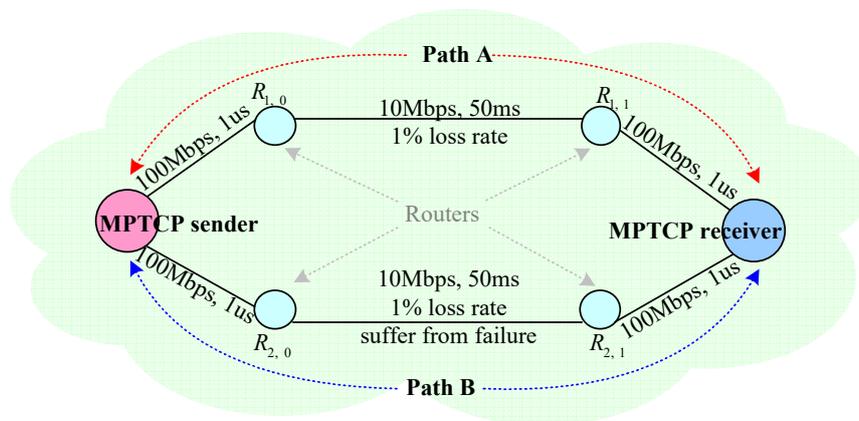


Figure 2. A sample dual dumbbell simulation topology.

Table 1. The parameters used in the simulations.

Parameter	Value
Application traffic type	FTP traffic
Number of paths	2
Access link bandwidth	100 Mb
Access link delay	1 microsecond
Bottleneck link bandwidth	10 Mb
Bottleneck link delay	50 ms
The link queue type	Droptail
The link queue limit	500 packets
Window size on each path	100
Packet loss on each path	1%

Figures 3–5 illustrate the performance comparisons in terms of throughput, congestion window (*cwnd*), and out-of-order Data Sequence Number (DSN) when Case 1 and Case 2 were executed, respectively. As Figure 3 shows, compared with the MPTCP’s throughput in Case 1, the MPTCP’s throughput in Case 2 was gradually lowered from the former (before 30 s of simulation time) to backward (after 30 s of simulation time) due to failure on Path B. When a failure occurred on Path B from the 30th s, it caused back-to-back timeouts on Path B (see Figure 4; the *cwnd* size on Path B was always set to one after 30 s). As in the failure case, the MPTCP sender tried to transmit data after each of these timeouts on Path B (namely, unnecessary retransmission via the failure path [27]). The resulting receiver buffer blocking in MPTCP, with a great number of out-of-order data (see Figure 5) held in the constrained receiver buffer, prevented the efficient use of Path A for data transmission and resulted in serious throughput performance degradations.

From the above experimental results, we can observe that the MPTCP could achieve lower application level performance even than using a single path when it simultaneously exploited multiple paths that had huge path quality differences. When a path experienced network failure or multiple paths had huge quality differences, the receiver buffer blocking may become an inevitable phenomenon in MPTCP. Unfortunately, the standard MPTCP path management mechanism was very simple; it used all the available paths in the MPTCP connection for concurrent multipath data transmission, without considering that wireless network latency and link failure tended to occur frequently due to mobility, signal fading, and wireless interference. To remedy this situation, this paper proposes a lightweight multipath management mechanism to help MPTCP know not only when, but also how to prevent the use of an underperforming or failure path in multipath transmission. This approach could possibly alleviate the receiver buffer blocking and optimize the throughput performance of MPTCP.

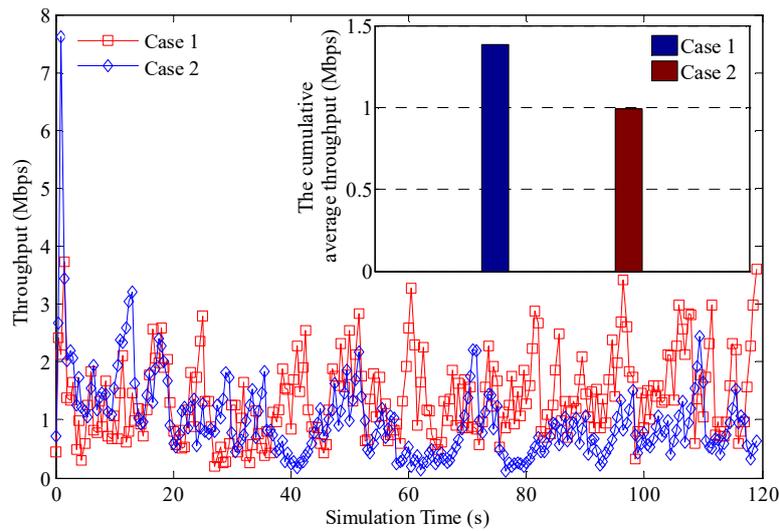


Figure 3. The throughput comparison.

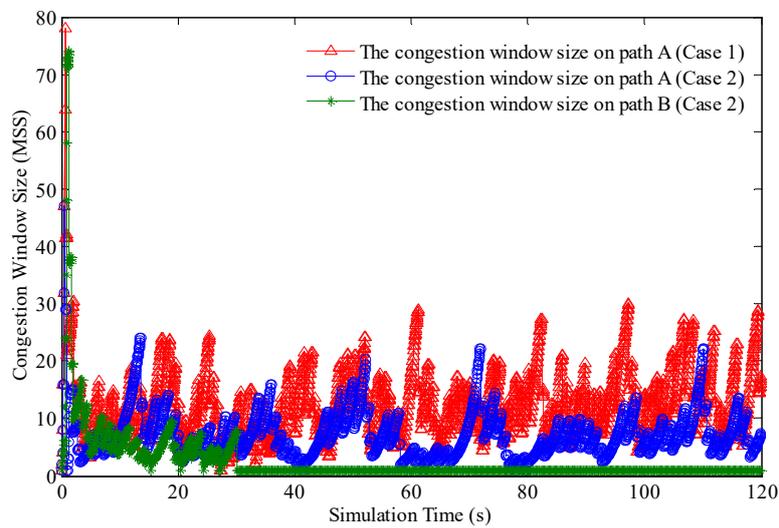


Figure 4. The congestion window comparison.

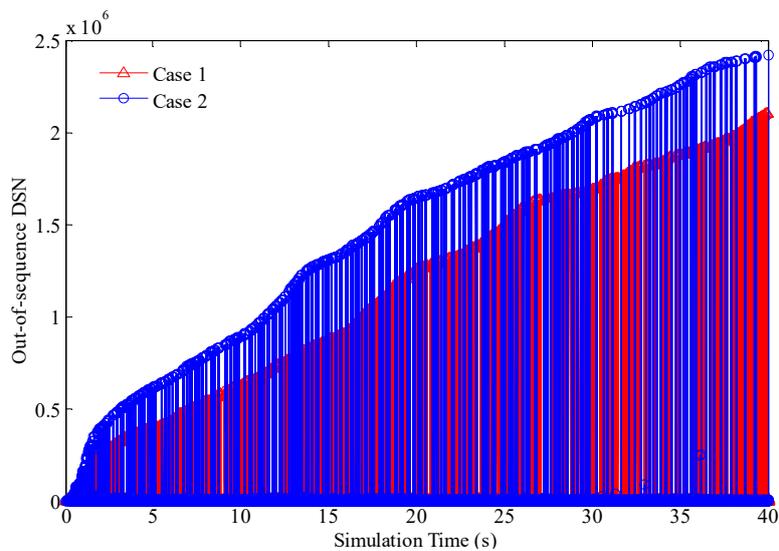


Figure 5. The out-of-order data comparison.

### 3. System Detail Design

Figure 6 illustrates the architecture of our MPTCP-LM<sup>3</sup> system, which involves an MPTCP sender, an MPTCP receiver, and  $n$  multiple paths. As an extension of the standard MPTCP, the MPTCP-LM<sup>3</sup> receiver was responsible for the path collection and reassembling if any MPTCP segment data arrived in the receiver buffer. At the MPTCP-LM<sup>3</sup> sender side, there was an additional MPTCP-LM<sup>3</sup> module compared with the standard MPTCP, which was a Lightweight Multipath Management Mechanism (LM<sup>3</sup>) that aimed to (i) prevent the use of an underperforming or failure path in multipath transmission, (ii) collect appropriate paths into MPTCP for smart bandwidth aggregation and parallel transmission, and (iii) reduce the receiver buffer blocking and optimize the quality of service when deploying MPTCP on the multihomed mobile computing devices. Following is the detail of the design of the proposed MPTCP-LM<sup>3</sup> solution.

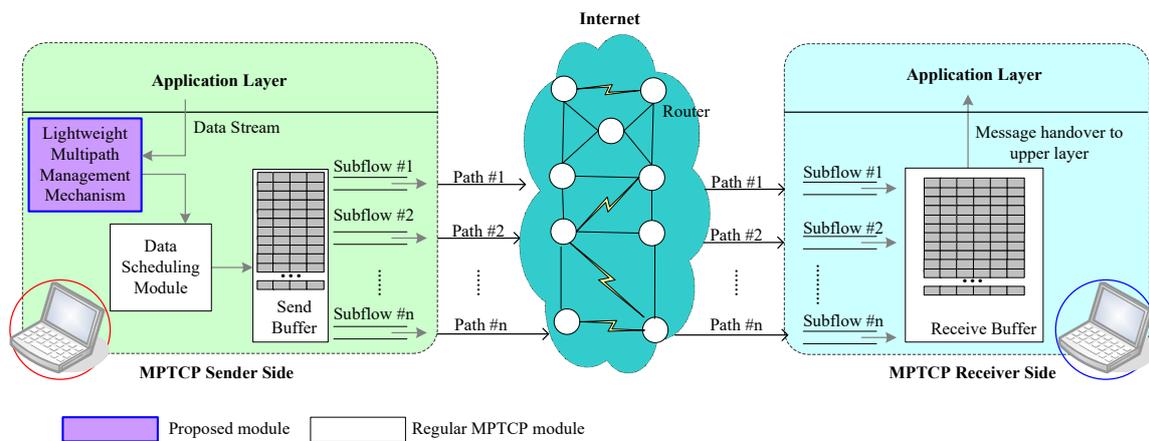


Figure 6. The architecture of MPTCP-Lightweight Multipath Management Mechanism (LM<sup>3</sup>).

Simply, in MPTCP-LM<sup>3</sup>, the receiver buffer maintaining the out-of-order data for reordering is viewed as a router queuing packets for forwarding. Further, a large number of out-of-order data held in the limited receiver buffer and resulting in receiver buffer blocking can be viewed as a large number of packets held in the overloaded router and resulting congestion. In view of this, MPTCP-LM<sup>3</sup> thinks of buffer blocking occurring at the receiver side as “congestion” occurring in the receiver buffer, like network congestion caused by an overloaded router (the router buffer gets full). Therefore, we borrowed ideas from the classic TCP’s congestion Avoidance (CA) mechanism [28] to design the LM<sup>3</sup> mode for the receiver buffer blocking avoidance. In order to make the paper self-contained, we introduce the general ideas of the CA mechanism as follows.

The CA mechanism is maintained by a TCP sender to adjust the *cwnd* size and alleviate network congestion. The main operations of the TCP’s CA mechanism running in the congestion avoidance state can be summarized as follows [28]: (i) for every packet acknowledged, the *cwnd* is additively increased by one Maximum Segment Size (MSS) each RTT; (ii) for any packet loss (three duplicate ACKs are received), set the *cwnd* value to half the current *cwnd* size; (iii) on timeout, the *cwnd* size is reset to one MSS. More simply, the TCP’s CA operations can be expressed by using the following mathematical model:

$$cwnd_{new} = \begin{cases} \frac{1}{2} \times cwnd_{old}, & \text{if a packet loss event is detected;} \\ 1 \text{ MSS}, & \text{if a timeout event is detected,} \end{cases} \quad (1)$$

Inspired by the idea of the above TCP’s CA mechanism, MPTCP-LM<sup>3</sup> was designed to adjust the number of transmission paths intelligently, like the *cwnd* adjustment in TCP’s CA mechanism, to protect possibly against receiver buffer blocking. To this end, in the connection establishment phase, MPTCP-LM<sup>3</sup> set a buffer blocking counter (*buf\_bloc\_count*) and triggered these counters with receiver

buffer blocking events. The *buf\_bloc\_count* was set with a default value of zero and was incremented by one when a receiver buffer blocking occurred in the receiver buffer. After three consecutive receiver buffer blocking events, the *buf\_bloc\_count* was reset back to its default value. According to the *buf\_bloc\_count* value, MPTCP-LM<sup>3</sup> could adaptively abandon or exploit multiple paths, similar to the *cwnd* adjustment in TCP’s CA mechanism, following the principles below:

- (i) For a case in which *buf\_bloc\_count* = 1 (or *buf\_bloc\_count* = 2), namely there is a single buffer blocking event occurring in the receiver buffer, MPTCP-LM<sup>3</sup> views this case as “a packet loss event in the TCP congestion avoidance phase”. In this case, the number of paths within the MPTCP connection (denoted as *path\_num*) will be halved, and the half of paths that has high performing transmission quality will be used in multipath transmission.
- (ii) For a case in which *buf\_bloc\_count* = 3, namely three consecutive buffer blocking events occurred in the receiver buffer, MPTCP-LM<sup>3</sup> views this case as “a timeout event in the TCP congestion avoidance phase”. In this case, only one path that has the highest transmission quality is used in MPTCP (*path\_num* is set to one). In addition, the value of *buf\_bloc\_count* is reset back to zero.

As for path transmission quality, in MPTCP, there are two recommended schedulers used to distinguish multiple paths and assign data to these paths for transmission accordingly, which are the lowest-RTT-first scheduler and the largest-*cwnd*-first scheduler, which differentiate the transmission quality of multiple paths by using each path’s own RTT and *cwnd* [29], respectively. In MPTCP-LM<sup>3</sup>, the transmission quality of multiple paths is differentiated by a function of two transport layer networking variables (namely RTT and *cwnd*), enabled by the Simple Additive Weighting (SAW) method [30], which is probably the most popular choice for multiple attribute decision making systems due to its simplicity. By using SAW, the transmission quality of each path in MPTCP-LM<sup>3</sup> is determined by the weighted sum of both the RTT and *cwnd* values. Assume that path *p<sub>i</sub>* is one of multiple paths within the MPTCP-LM<sup>3</sup> connection; the MPTCP-LM<sup>3</sup>’s SAW based path quality evaluation method calculates the transmission quality of *p<sub>i</sub>* by using the following equation,

$$\left. \begin{aligned} \mathbb{R}_{p_i} &= \sigma \times cwnd'_{p_i} + \ell \times RTT'_{p_i}; \\ \text{s.t. } \quad \sigma + \ell &= 1, \end{aligned} \right\} \tag{2}$$

where  $\mathbb{R}_{p_i}$  represents the transmission quality of *p<sub>i</sub>*. In MPTCP-LM<sup>3</sup>, the path with the largest  $\mathbb{R}$  value has the best transmission quality. Since SAW mostly reaches an optimal performance when the weighting factors of the system variables are fair [31], therefore, both weighting factors  $\sigma$  and  $\ell$  used in Equation (2) were set to a fairness value  $\frac{1}{2}$  by default. For real-world networking, the two weighting factors could be changed to other suitable values. Moreover, in Equation (2),  $cwnd'_{p_i}$  and  $RTT'_{p_i}$  are the normalized *cwnd* and RTT values of *p<sub>i</sub>*, which can be calculated by using the following Equations (3) and (4), respectively,

$$cwnd'_{p_i} = (cwnd_{p_i} - H_{\min}) \times \frac{1}{H_{\max} - H_{\min}}, \tag{3}$$

$$RTT'_{p_i} = (\Pi_{\max} - RTT_{p_i}) \times \frac{1}{\Pi_{\max} - \Pi_{\min}}, \tag{4}$$

where  $H_{\max}$  and  $H_{\min}$  are the upper bound and the lower bound of the *cwnd* size measured on *p<sub>i</sub>*, respectively.  $\Pi_{\max}$  and  $\Pi_{\min}$  are the upper bound and the lower bound of the RTT values measured on *p<sub>i</sub>*, respectively.

Let us suppose *n* possible paths (*p<sub>1</sub>*, *p<sub>2</sub>*, ..., *p<sub>n</sub>*) within the MPTCP connection, and let *p<sub>list</sub>*, *p<sub>A\_list</sub>*, and *p<sub>R\_list</sub>* be the collection of all the available paths in the MPTCP connection, the collection of paths activated for multipathing, and the collection of paths removed from multipathing, respectively, in which (*p<sub>A\_list</sub>*, *p<sub>R\_list</sub>* ∈ *p<sub>list</sub>*) and *p<sub>A\_list</sub>* ∩ *p<sub>R\_list</sub>* = ∅. The main MPTCP-LM<sup>3</sup> operations, each of which is explained in detail, are as follows:

- (i) Estimating each path’s transmission quality ( $\mathbb{R}$ ) periodically (per RTT) by jointly considering its own *cwnd* and RTT values;
- (ii) Sorting all the paths in  $p_{list}$  in descending order according to their own  $\mathbb{R}$  values;
- (iii) If there is no receiver buffer blocking occurring at the receiver side, assigning traffic over all the paths in  $p_{list}$ , as the regular MPTCP does;
- (iv) If there is a receiver buffer blocking event detected, but  $buf\_bloc\_count < 3$ , halving the number of paths in multipath transmission, and then incrementing the value of  $buf\_bloc\_count$  by one, by using the following Equations (5) and (6),

$$path\_num \leftarrow \left\lfloor \frac{1}{2} \times path\_num \right\rfloor . \tag{5}$$

$$buf\_bloc\_count \leftarrow buf\_bloc\_count + 1 . \tag{6}$$

In this case, the top half of paths in the  $p_{A\_list}$  will be used in multipath transmission (there is  $p_{A\_list} = p_{list}$  at the beginning of MPTCP based multipathing).

- (v) If there is a receiver buffer blocking detected and  $buf\_bloc\_count = 3$ , reducing the number of paths to one, then resetting the value of  $buf\_bloc\_count$  back to zero, by using the following Equations (7) and (8),

$$path\_num \leftarrow 1 . \tag{7}$$

$$buf\_bloc\_count \leftarrow 0 . \tag{8}$$

In this case, only the first path in  $p_{A\_list}$  can be used for data transmission.

By using the above operations, MPTCP-LM<sup>3</sup> can intelligently aggregate the bandwidth of multiple paths for data transmission while possibly mitigating the receiver buffer blocking problem. Moreover, in order to maximize the network resource utilization possibly, in MPTCP-LM<sup>3</sup>, a path  $p_j$  in  $p_{R\_list}$  can be put into  $p_{A\_list}$  for multipath transmission, if its current transmission quality,  $\mathbb{R}_{p_j}$ , is not lower than the average transmission quality of other high performing paths (in order to make sure  $p_j$  has little transmission quality difference compared to other high performing paths), namely,

$$\mathbb{R}_{p_j} \geq \frac{1}{\text{count}(p_{A\_list})} \times \sum_{k=1}^{\text{count}(p_{A\_list})} \mathbb{R}_{p_k} , \tag{9}$$

where  $\text{count}(p_{A\_list})$  is the number of paths in  $p_{A\_list}$ .  $p_k$  is the  $k^{\text{th}}$  path in  $p_{A\_list}$ . The pseudocode of multipath management mechanism used in MPTCP-LM<sup>3</sup> is presented in Algorithm 1. For convenience, we adopted the following notations used in Algorithm 1 (as shown in Table 2).

For Algorithm 1, the time complexity is primarily influenced by the first “for” statement (from Line 2 to Line 4), the second “for” statement (from Line 16 to Line 21), and the sort algorithm in Line 5. For the two “for” statements, the time complexity of each “for” statement is  $O(n)$  (we assumed the second “for” statement was executed). As for the sort algorithm, the time complexity was  $O(n \log_2 n)$  because the QuickSort algorithm was used for sorting the paths. Since there was  $f(n) = 2n + n \log_2 n$ , therefore the time complexity of this algorithm was  $O(n \log_2 n)$ . More simply, the total number of paths collected by Algorithm 1 can be expressed by using the following mathematical model:

$$path\_num = \begin{cases} path\_num, & \text{if } buf\_bloc\_count = 0; \\ \left\lfloor \frac{1}{2} \times path\_num \right\rfloor, & \text{if } buf\_bloc\_count < 3; \\ 1, & \text{if } buf\_bloc\_count=3. \end{cases} \tag{10}$$

**Table 2.** Notations used in the MPTCP-LM<sup>3</sup> based multipath management algorithm.

Notation	Description
$p_{list}$	The collection of all the available paths in the MPTCP connection
$p_{A\_list}$	The collection of paths activated for multipathing
$p_{R\_list}$	The collection of paths removed from multipathing
$p_{list(i)}$	The $i^{th}$ path within $p_{list}$
$p_{A\_list(j)}$	The $j^{th}$ path within $p_{A\_list}$
$p_{R\_list(k)}$	The $k^{th}$ path within $p_{R\_list}$

**Algorithm 1.** MPTCP-LM<sup>3</sup> based multipath management algorithm.

---

**Initialization:**  
 $p_{A\_list} = p_{list}$ ;  
 $p_{R\_list} = \emptyset$ ;  
 $path\_num = \text{count}(p_{list})$ ;  
 $buf\_bloc\_count = 0$ .

- 1: use all the paths in  $p_{A\_list}$  for multipathing;
- 2: **for** ( $i = 1, i \leq \text{count}(p_{A\_list}), i + +$ ) **do**
- 3:   calculate the  $\mathbb{R}$  value of  $p_{A\_list(i)}$  by using Equation (2);
- 4: **end for**
- 5: sort the paths in  $p_{A\_list}$  in descending order according to their own  $\mathbb{R}$  values;
- 6: **if** there is a receiver buffer blocking detected **then**
- 7:   set  $buf\_bloc\_count \leftarrow buf\_bloc\_count + 1$ ;
- 8:   **if**  $buf\_bloc\_count < 3$ , **then**
- 9:     set  $path\_num \leftarrow \lceil \frac{1}{2} \times path\_num \rceil$ ;
- 10:   **end if**
- 11:   **if**  $buf\_bloc\_count = 3$ , **then**
- 12:     set  $path\_num \leftarrow 1$ ;
- 13:   **end if**
- 14: **end if**
- 15: **if** ( $\text{count}(p_{R\_list}) > 0$ ), **then**
- 16:   **for** ( $j = 1, j \leq \text{count}(p_{R\_list}), j + +$ ) **do**
- 17:     calculate the  $\mathbb{R}$  value of  $p_{R\_list(j)}$  by using Equation (2);
- 18:     **if**  $\mathbb{R}_{p_{R\_list(j)}} \geq \frac{1}{\text{count}(p_{A\_list})} \times \sum_{i=1}^{\text{count}(p_{A\_list})} \mathbb{R}_{p_{A\_list(i)}}$
- 19:       move  $p_{R\_list(j)}$  into  $p_{A\_list}$ ;
- 20:     **end if**
- 21:   **end for**
- 22: **end if**
- 23: allocate the MPTCP data across the path(s) in  $p_{A\_list}$ .

---

## 4. Performance Evaluation

### 4.1. Simulation Topology

The performance evaluation was carried out on Network Simulator 2 Version 2.35 (NS-2) [25] with the MPTCP module [26]. The experiments considered a typical MPTCP based device-to-device communication scenario presented in Figure 7. As the figure shows, an MPTCP sender and an MPTCP receiver are connecting with each other via three independent transmission paths (denoted A, B, and C). Path A's bottleneck has a 10 Mbps bandwidth and 10–20 ms propagation delay. Path B's bottleneck has a 11 Mbps bandwidth and 10–20 ms propagation delay. Path C's bottleneck has a 2 Mbps bandwidth and 10–60 ms propagation delay. In the simulation, Path C will experience a short term failure from 10 to 30 s and a complete failure after 50 s of simulation time (we simulated Path C failure by bringing down the bottleneck link between routers  $R_{3,1}$  and  $R_{3,2}$ ). The parameters shown in Table 3 were used for configuring multiple paths. The other parameters of the MPTCP used the default values specified in NS-2.

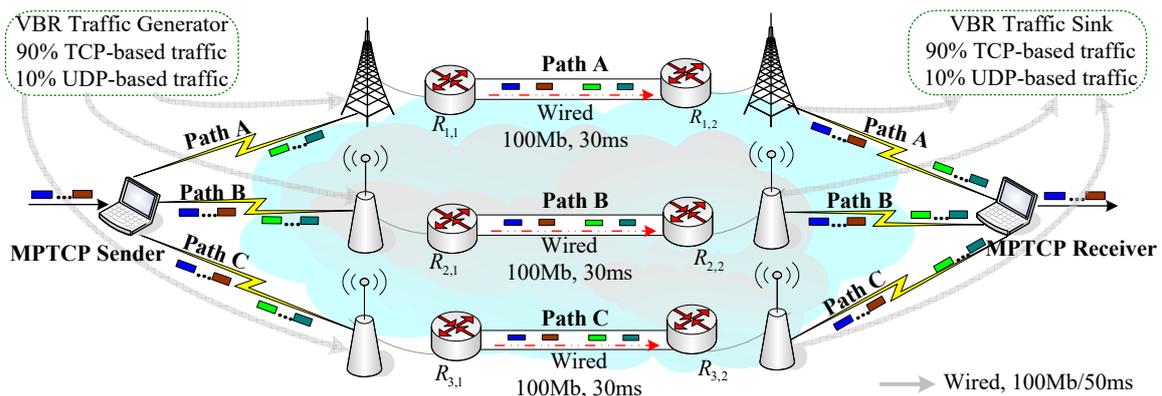


Figure 7. Simulation topology.

In the experiment, all the wireless access links were attached to a Uniform Path-Loss (UPL) model and a Two-state Markov Loss (TML) model in order to simulate the highly frequent and bursty frame loss in the data link layer. The UPL and TML models were used to represent the distributed loss caused by wireless noise interference and the infrequent continuous loss caused by wireless signal fading, respectively [32]. In addition, in order to simulate the complex behaviors of Internet traffic [33], we attached each of the three paths to a VBR (Variable Bit-Rate) traffic generator to send the VBR cross-traffic to its corresponding sink. The packet size of the VBR cross-traffic utilized in the simulation was chosen as follows [34]: 49% were 44 bytes in length, 1.2% 576 bytes, 2.1% 628 bytes, 1.7% 1300 bytes, and the other 46% 1500 bytes in length, in which 90% of the VBR cross-traffic was transmitted by TCP technology and the remaining 10% by UDP technology. The VBR cross-traffic on each of the three paths consumed 0–50% of the access link bandwidth. The simulation time was set to 100 s with infinite FTP traffic.

Table 3. Path parameter configuration used in the simulation.

Network Parameters	Path A	Path B	Path C
Wireless access bandwidth	10 Mbps	11 Mbps	2 Mbps
Wireless link delay	10–20 ms	10–20 ms	10–60 ms
Wireless link queue type	Droptail	Droptail	Droptail
Core network bandwidth	100 Mbps	100 Mbps	100 Mbps
Core network delay	30 ms	30 ms	30 ms
Uniform loss rate	1–2%	1–4%	1–6%
Markov loss rate	1%	1%	1%

#### 4.2. Simulation Results

We here compare the performance of our MPTCP-LM<sup>3</sup> with the baseline MPTCP [7] and our previous work (PU)<sup>2</sup>M<sup>2</sup> [14]. For convenience, we portray the results of MPTCP-LM<sup>3</sup> as “MPTCP-LM<sup>3</sup>” in the result figures, and the results when applying the baseline MPTCP and (PU)<sup>2</sup>M<sup>2</sup> schemes are portrayed as “the baseline MPTCP” and “MPTCP + (PU)<sup>2</sup>M<sup>2</sup>”, respectively.

(1) Sending and receiving data DSN: Figure 8 shows the comparison results of data sending and receiving times when the baseline MPTCP, MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, and MPTCP-LM<sup>3</sup> were used, respectively. In order to better illustrate the comparison, the results between  $t = 30$  s and  $t = 100$  s are illustrated (including the no path failure phase (between 30 s to 50 s) and path failure phase (after 50 s)), representative for all the simulation results. From the figure, we can observe that between 30 and 50 s of simulation time, MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> performed at a lower level of data sending and receiving times than the baseline MPTCP. This was because the baseline MPTCP fully utilized the three paths for data transmission (Path C was available and reused in multipath transmission after 30 s). In both MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, Path C may be prevented from multipathing due to the quality gap (compared with Paths A and B).

However, MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> could achieve a higher level of data sending and receiving times than the baseline MPTCP after 50 s. This was because in both MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, the sender only selected the stable paths for multipath transmission, while in the baseline MPTCP, the path failure declaring was inherited from TCP's operations (in the TCP, the time required to declare a broken path needs at least 1 + 3 + 6 + 12 + 24 = 46 s (*TcpMaxDataRetransmissions* = 5 and *RTO* = 1 s)). That is, between 50 and 100 s, the broken Path C could also be used in multipath transmission, which caused the interruptions in Paths A and B and, thus, constrained the sender from transmitting data traffic.

Besides, we can also see that between 50 and 70 s of simulation time, MPTCP + LM<sup>3</sup> achieved a lower level of data sending and receiving times than the MPTCP + (PU)<sup>2</sup>M<sup>2</sup> scheme. This was because MPTCP + LM<sup>3</sup> detected path failures based on the counts of receiver buffer blocking, while MPTCP + (PU)<sup>2</sup>M<sup>2</sup> was based on the path's delay variations, which means MPTCP + LM<sup>3</sup> could possibly find out and prevent the usage of a broken path or a poorly performing path more slowly than MPTCP + (PU)<sup>2</sup>M<sup>2</sup>. However, in MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, a path could be removed/reconnected frequently if its delay performance was getting worse accidentally or suddenly. This limitation made MPTCP + (PU)<sup>2</sup>M<sup>2</sup> not able to provide stable transmission efficiency and perform worse than MPTCP + LM<sup>3</sup> in the rest of the simulation time (between 70 and 100 s).

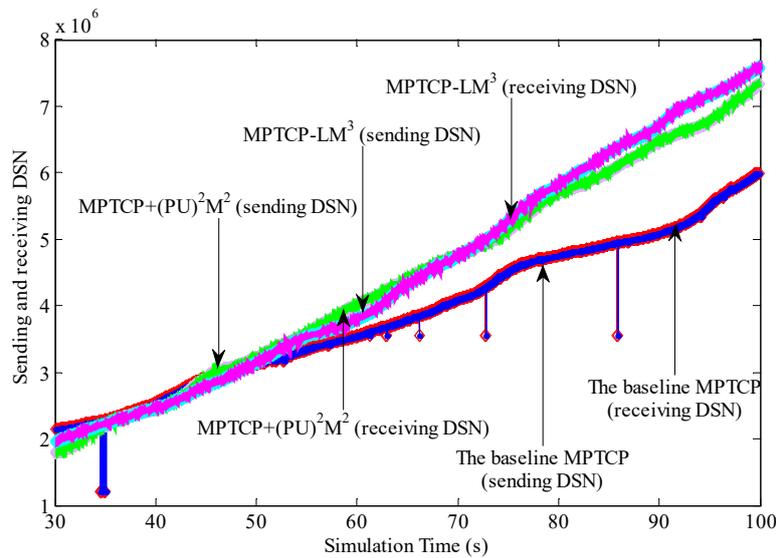
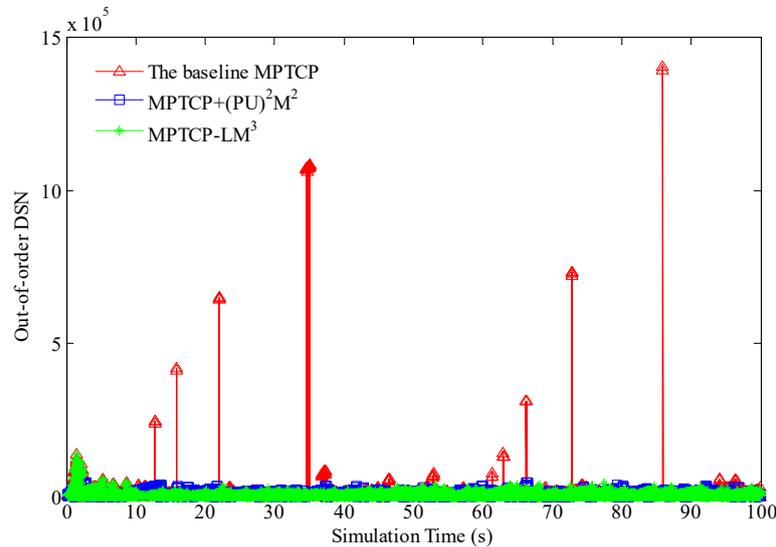


Figure 8. The comparison of data sending and receiving times.

(2) Out-of-order DSN: The Out-Of-Order DSN (OOO DSN) is a good metric to select for studying and analyzing the performance of multipath protocols. The OOO DSN can be attained by the offset between the DSNs of two consecutively received MPTCP data chunks (the difference between the DSN of the current MPTCP chunk and that of the latest received MPTCP chunk). Figure 9 shows a comparison of out-of-order chunks among the baseline MPTCP, MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, and MPTCP-LM<sup>3</sup>. As the figure shows, the baseline MPTCP scheme generated more out-of-order chunks and required increased unnecessary packet reordering than both MPTCP + (PU)<sup>2</sup>M<sup>2</sup> and MPTCP-LM<sup>3</sup>. This was because the baseline MPTCP's scheduler assigned data traffic over all the paths, without considering that a poorly performing path or a failure prone path would cause extremely great numbers of out-of-order chunks for reordering.

In contrast, both MPTCP + (PU)<sup>2</sup>M<sup>2</sup> and MPTCP-LM<sup>3</sup> could detect a broken/poorly performing path and select a group of stable paths for data transmission. In this way, they could possibly allocate data traffic over the stable and high performing paths and thus reduce the out-of-order data reception. However, it was noted that MPTCP + (PU)<sup>2</sup>M<sup>2</sup> often generated more out-of-order data chunks than MPTCP-LM<sup>3</sup>. This was because in MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, the paths selected for multipath transmission

would be frequently changed and sensitive to paths' delay variations, while in MPTCP-LM<sup>3</sup>, the path group, which was jointly determined by the paths' delay and *cwnd* characteristics, was relatively stable and changed little. When comparing the three schemes, we can see that the peak out-of-order data reception at the receiver was up to  $1.40 \times 10^6$  when using the baseline MPTCP, while it was approximately  $1.39 \times 10^5$  when using both MPTCP + (PU)<sup>2</sup>M<sup>2</sup> and MPTCP-LM<sup>3</sup>.



**Figure 9.** The comparison of out-of-order Data Sequence Number (DSN).

(3) End-to-end delay: Figure 10 shows a comparison of end-to-end delay performance when using the baseline MPTCP, MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, and MPTCP-LM<sup>3</sup> schemes, respectively. As we analyzed earlier, both MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> could prevent an underperforming path (or a broken path) from multipathing, and they could thereby prevent unnecessary retransmission in the underperforming or broken paths. This made MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> reduce the unnecessary retransmission delay. Moreover, both MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> could possibly ensure that the MPTCP segments arrive at the receiver in the right order. This helped the two schemes reduce the packet reordering delay. As a result, MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> obtained a lower level of delay than the baseline MPTCP.

However, MPTCP + (PU)<sup>2</sup>M<sup>2</sup> performed worse than MPTCP-LM<sup>3</sup> because more out-of-order data reception in MPTCP + (PU)<sup>2</sup>M<sup>2</sup> required increased reordering than MPTCP-LM<sup>3</sup>. Moreover, in MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, the paths used in multipath transmission often changed, and as a result, the data traffic could be offloaded from one path to another frequently, which incurred additional processing time overhead. When comparing the three methods in terms of the cumulative average delay (with a total of 100 s of simulation time), the MPTCP-LM<sup>3</sup> scheme's cumulative average delay was approximately 4.35% lower than that of the baseline MPTCP and 1.03% lower than that of MPTCP + (PU)<sup>2</sup>M<sup>2</sup>.

(4) Jitter comparison: Jitter can be defined as the amount of packet delay variation in the end-to-end data traffic transmission time. Jitter is a well suited metric to evaluate and analyze the temporal performance of a multipath transmission protocol. A higher level of jitter is more likely to occur on a poorly performing transport technology and vice versa. Figure 11 shows a comparison of the jitter performance when the three schemes were used. Although the path management behaviors of both MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> (removing or reconnecting a path in multipath transmission) influenced the jitter performance, which was occasionally less than that of the baseline MPTCP, we argue that this was acceptable and worthy of achieving high level performance since MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> could adaptively select appropriate paths for multipathing. In contrast, the baseline MPTCP, as previously mentioned, was bound to out-of-order data reception (due to path

quality characteristics) and unnecessary retransmission (due to path failures), resulting in a lower quality of service than both MPTCP-LM<sup>3</sup> and MPTCP + (PU)<sup>2</sup>M<sup>2</sup>.

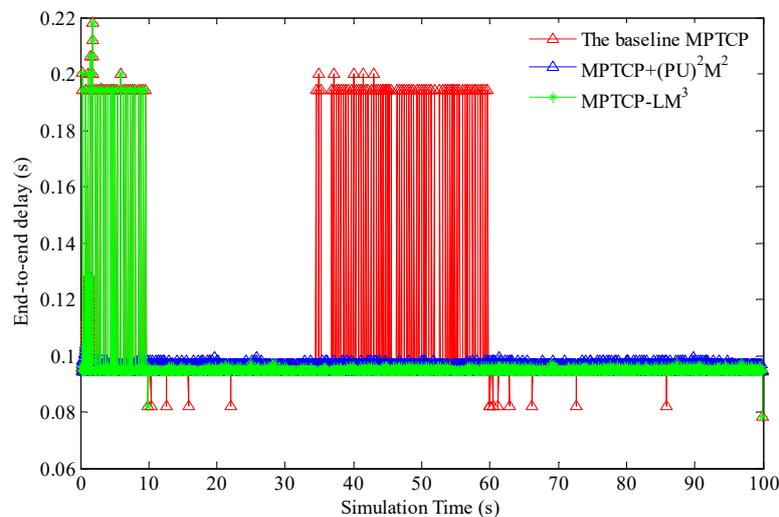


Figure 10. The comparison of end-to-end delay.

When comparing the MPTCP + (PU)<sup>2</sup>M<sup>2</sup> and MPTCP-LM<sup>3</sup> methods, it was noted that MPTCP + (PU)<sup>2</sup>M<sup>2</sup> more or less generated a higher level of jitter than MPTCP-LM<sup>3</sup>. This was because MPTCP + (PU)<sup>2</sup>M<sup>2</sup> frequently deactivated or reactivated a path for data transmission, which incurred the additional overhead for data offloading and reordering and, thus, increased the amount of packet delay variations. In contrast, MPTCP + (PU)<sup>2</sup>M<sup>2</sup> provided a more stable path group for data transmission than MPTCP + (PU)<sup>2</sup>M<sup>2</sup>, and it could possibly reduce the overhead of data offloading and reordering and correspondingly generate a lower level of jitter performance than MPTCP + (PU)<sup>2</sup>M<sup>2</sup>.

(5) Throughput performance: Figure 12 shows a comparison of throughput performance when using the three schemes. MPTCP + (PU)<sup>2</sup>M<sup>2</sup> and MPTCP-LM<sup>3</sup> prevented the usage of underperforming or broken paths in multipath transmission, which could not only reduce the probability of receiver buffer blocking, but also reduce transmission interruptions in the other high performing paths. Therefore, both MPTCP + (PU)<sup>2</sup>M<sup>2</sup> and MPTCP-LM<sup>3</sup> performed better than the baseline MPTCP. When comparing MPTCP + (PU)<sup>2</sup>M<sup>2</sup> and MPTCP-LM<sup>3</sup>, it was noted that the MPTCP-LM<sup>3</sup> achieved higher association average throughput than the MPTCP + (PU)<sup>2</sup>M<sup>2</sup>. This was because MPTCP + (PU)<sup>2</sup>M<sup>2</sup> detected an underperforming or a broken path by monitoring the delay changes of paths. This path estimation method was very sensitive to variations of delay and could cause throughput performance degradation because of frequently changing the paths in multipath transmission. In contrast, MPTCP-LM<sup>3</sup> declared an underperforming or a broken path by monitoring the receiver buffer blocking, and in this way, it helped MPTCP-LM<sup>3</sup> provide stable transmission performance.

In order to illustrate the throughput comparison better, we calculated the cumulative average throughput of the three schemes respectively, by averaging the total average throughput values in a total of 100 s of simulation time. The subfigure in Figure 12 shows a comparison of cumulative average throughput when the three schemes were used, respectively. As the subfigure shows, MPTCP-LM<sup>3</sup> performed the best among the three schemes compared. However, it is noteworthy that the cumulative average throughput of MPTCP-LM<sup>3</sup> was about 2.8 Mbps, which corresponded only to 12% of the aggregated bandwidths of the three access links (10 Mbps + 11 Mbps + 2 Mbps = 23 Mbps). We argue that this phenomenon was caused by several reasons: (i) the VBR competing traffic could consume a certain amount of each path's access bandwidth; (ii) the two wireless loss models attached to each of the three access links could lead to the degradation of MPTCP throughput performance; (iii) the path failures on Path C will inevitably caused transmission interruption in Paths A and B and, thus,

lowered the MPTCP throughput performance; and (iv) the out-of-order data arrival would actually degrade the performance in most cases because of the path quality differences. Still, the MPTCP-LM<sup>3</sup> scheme’s cumulative average throughput was 27.97% higher than that of the baseline MPTCP and 3.68% higher than that of MPTCP + (PU)<sup>2</sup>M<sup>2</sup>.

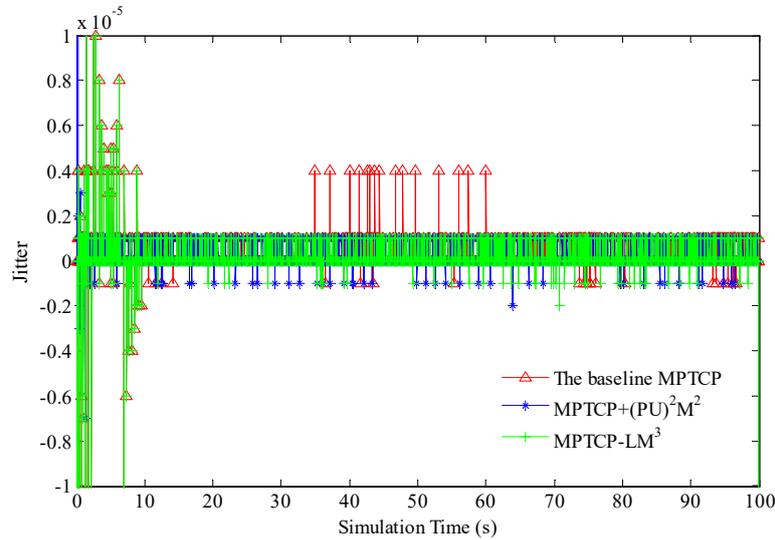


Figure 11. The comparison of jitter.

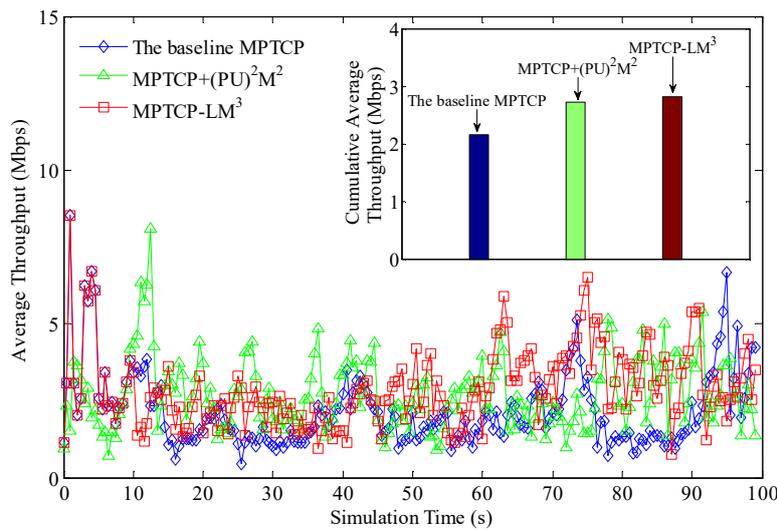


Figure 12. The comparison of throughput.

### 5. Discussion

In this paper, we proposed a lightweight path management mechanism to aid multihomed MPTCP based mobile computing devices towards efficient multipath data transmission, inspired by the TCP congestion avoidance mechanism. The proposed solution can be easily merged with the current MPTCP variants to remove and reconnect the connection in MPTCP adaptively. In this section, we highlight the limitations of our solution and give some interesting directions for future work. We encourage the researchers who are interested in this field to pay attention to and discuss the following interesting challenges.

- In practice, the number of connection interfaces on the end-devices may not be too large, which may constrain the success of our proposal in today’s Internet architectures. We argue that end-devices in the future Internet may be equipped with many network interfaces and can see multiple access links. The authors hope to attract more researchers to discuss this topic.

- In the Performance Evaluation Section, we discussed the possible reasons why the throughput performance of all the MPTCP variants was far from satisfactory. We noticed that in a sample dual dumbbell simulation topology (see Section 2), the cumulative average throughput of MPTCP was only close to 1.5 Mbps, which corresponded only to 15% of the bottleneck bandwidth of Path A (10 Mbps). For these negative simulation results, we argue that the NS-2 model of MPTCP may not be able to reflect the MPTCP implementation fully. We encourage more researchers to pay attention to and discuss this controversial problem.

## 6. Conclusions and Future Work

Mobile computing devices attached to more than one wireless network interface can improve their transmission performance by making use of the MPTCP technology. However, the path management mechanism in MPTCP is very simple and vulnerable to the quality differences of multiple paths in heterogeneous wireless environments. As a remedy, this paper presented a lightweight multipath management mechanism for MPTCP (MPTCP-LM<sup>3</sup>) necessitating the following aims: (i) optimizing MPTCP path management mechanism and possibly preventing an underperforming path from multipathing, (ii) reducing the out-of-order data reception and alleviating the receiver buffer blocking problems in MPTCP, (iii) improving the throughput performance and quality of service of multipath transmission. The simulation results demonstrated that MPTCP-LM<sup>3</sup> outperformed the baseline MPTCP and MPTCP + (PU)<sup>2</sup>M<sup>2</sup> in terms of data transmission service quality.

By carrying out simulations on top of NS-2, we noted a limitation that the throughput performance of all the three MPTCP variants was far from satisfactory, as we discussed in the Performance Evaluation Section. Recent research argues that there is an open-source implementation of multipath TCP that exists and is well maintained in the Linux kernel [35]. Our future work will develop a Linux kernel based hardware test-bed, then implement the proposed MPTCP-LM<sup>3</sup> solution in the test-bed, and provide real experimental results.

Furthermore, due to the strict layering principles, MPTCP-LM<sup>3</sup> can only use the networking parameters at the transport layer to estimate each path's transmission quality and thereby make a "best-effort" transmission behavior. Considering the fact that variations of the wireless channel, such as channel fading and co-channel interference, are the extremely influencing factors in the wireless transmission, our future work will be devoted to the optimization of our MPTCP-LM<sup>3</sup> solution by jointly considering the cross-layer design and "smart collaborative networking" method [36–39]. Then, a comprehensive wireless network environment including many and promising wireless technologies (i.e., Bluetooth, LoRaWAN, SigFox, etc.) will be considered, and the comparison with the state-of-the-art MPTCP schemes will be investigated in the performance evaluations.

**Author Contributions:** Proposed the ideas, Y.C., M.C., S.X., L.H., X.T., and Z.Z.; writing, original draft preparation, Y.C., M.C., and Z.Z.; writing, reviews and editing, S.X.; investigation and validations, M.C. All authors read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61962026, by the National Key R&D Program of China under Grant No. 2017YFC0806404, and by the Natural Science Foundation of Jiangxi Province under Grant Nos. 20192ACBL21031 and 20171BAB212014.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wu, J.; Tan, R.; Wang, M. Energy-Efficient Multipath TCP for Quality-Guaranteed Video over Heterogeneous Wireless Networks. *IEEE Trans. Multimed.* **2018**. [CrossRef]
2. Song, F.; Ai, Z.; Zhou, Y.; You, I.; Choo, R.; Zhang, H. Smart Collaborative Automation for Receive Buffer Control in Multipath Industrial Networks. *IEEE Trans. Ind. Inform.* **2019**. [CrossRef]
3. Wu, J.; Cheng, B.; Wang, M. Improving Multipath Video Transmission with Raptor Codes in Heterogeneous Wireless Networks. *IEEE Trans. Multimed.* **2018**, *20*, 457–472. [CrossRef]

4. Cao, Y.; Song, F.; Liu, Q.; Huang, M.; Wang, H.; You, I. A LDDoS-Aware Energy-Efficient Multipathing Scheme for Mobile Cloud Computing Systems. *IEEE Access* **2017**, *5*, 21862–21872. [[CrossRef](#)]
5. Available online: <https://support.apple.com/lv-lv/HT201373> (accessed on 30 December 2018).
6. Available online: <https://www.samsung.com/uk/support/mobile-devices/what-is-the-download-booster-and-how-do-i-enable-it-on-my-samsung-galaxy-alpha/> (accessed on 30 December 2018).
7. Ford, A.; Raiciu, C.; Handley, M.; Bonaventure, O. TCP Extensions for Multipath Operation with Multiple Addresses. *IETF RFC 6824*. **2013**. Available online: <https://tools.ietf.org/html/rfc6824> (accessed on 30 December 2018).
8. Wu, J.; Cheng, B.; Wang, M.; Chen, J. Quality-Aware Energy Optimization in Wireless Video Communication with Multipath TCP. *IEEE/ACM Trans. Netw.* **2017**, *25*, 2701–2718. [[CrossRef](#)]
9. Xu, C.; Wang, P.; Xiong, C.; Wei, X.; Muntean, G.-M. Pipeline Network Coding-Based Multipath Data Transfer in Heterogeneous Wireless Networks. *IEEE Trans. Broadcast.* **2017**, *63*, 376–390. [[CrossRef](#)]
10. Wang, W.; Wang, X.; Wang, D. Energy Efficient Congestion Control for Multipath TCP in Heterogeneous Networks. *IEEE Access* **2018**, *6*, 2889–2898. [[CrossRef](#)]
11. Kimura, B.; Lima, D.; Loureiro, A. Alternative Scheduling Decisions for Multipath TCP. *IEEE Commun. Lett.* **2017**, *21*, 2412–2415. [[CrossRef](#)]
12. Yang, F.; Wang, Q.; Amer, P. Out-of-Order Transmission for In-Order Arrival Scheduling for Multipath TCP. In Proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 13–16 May 2014; pp. 749–752.
13. Alheid, A.; Kaleshi, D.; Doufexi, A. An Analysis of the Impact of Out-of-Order Recovery Algorithms on MPTCP Throughput. In Proceedings of the IEEE 28th International Conference on Advanced Information Networking and Applications, Victoria, BC, Canada, 13–16 May 2014; pp. 156–163.
14. Cao, Y.; Song, F.; Luo, G.; Yi, Y.; Wang, W.; You, I.; Hao, W. (PU)<sup>2</sup>M<sup>2</sup>: A Potentially Underperforming-aware Path Usage Management Mechanism for Secure MPTCP based Multipathing Services. *Concurr. Comput. Pract. Exp.* **2018**, *30*, e4191. [[CrossRef](#)]
15. Liu, Y.; Neri, A.; Ruggeri, A.; Vegni, A. A MPTCP-Based Network Architecture for Intelligent Train Control and Traffic Management Operations. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2290–2302. [[CrossRef](#)]
16. Ou, S.; Huang, C.; Lee, T.; Huang, C. Out-of-order transmission enabled congestion and scheduling control for multipath TCP. In Proceedings of the International Wireless Communications and Mobile Computing Conference, Paphos, Cyprus, 5–9 September 2016.
17. Xue, K.; Han, J.; Ni, D.; Wei, W.; Cai, Y.; Xu, Q.; Hong, P. DPSAF: Forward Prediction Based Dynamic Packet Scheduling and Adjusting with Feedback for Multipath TCP in Lossy Heterogeneous Networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 1521–1534. [[CrossRef](#)]
18. Le, T.; Bui, L. Forward Delay based Packet Scheduling Algorithm for Multipath TCP. *Mob. Netw. Appl.* **2018**, *23*, 4–12. [[CrossRef](#)]
19. Ferlin, S.; Kucera, S.; Claussen, H.; Alay, O. MPTCP Meets FEC: Supporting Latency-Sensitive Applications Over Heterogeneous Networks. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2005–2018. [[CrossRef](#)]
20. Wang, K.; Dreiholz, T.; Zhou, X.; Fa, F.; Tan, Y.; Cheng, X.; Tan, Q. On the Path Management of Multi-Path TCP in Internet Scenarios based on the NorNet Testbed. In Proceedings of the IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan, 27–29 March 2017; pp. 1–8.
21. Kim, J.; Oh, B.; Lee, J. Receive Buffer based Path Management for MPTCP in heterogeneous networks. In Proceedings of the IFIP/IEEE IM, Lisbon, Portugal, 8–12 May 2017; pp. 648–651.
22. Oh, B.; Lee, J. Feedback-Based Path Failure Detection and Buffer Blocking Protection for MPTCP. *IEEE/ACM Trans. Netw.* **2016**, *24*, 3450–3461. [[CrossRef](#)]
23. Li, L.; Xu, K.; Li, T.; Zheng, K.; Peng, C.; Wang, D.; Wang, X.; Shen, M.; Mijumbi, R. A Measurement Study on Multi-path TCP with Multiple Cellular Carriers on High Speed Rails. In Proceedings of the ACM SIGCOMM, Budapest, Hungary, 20–25 August 2018; pp. 455–468.
24. Chen, Y.; Lim, Y.; Gibbens, R.; Nahum, E.; Khalili, R.; Towsley, D. A measurement based study of multipath tcp performance over wireless networks. In Proceedings of the ACM IMC, Barcelona, Spain, 23–25 October 2013; pp. 455–468.

25. UC Berkeley. *LBL, USC/ISI and Xerox Parc, NS-2 Documentation and Software*; version 2.35; UC Berkeley: Berkeley, CA, USA, November 2011. Available online: <https://www.isi.edu/nsnam/ns/doc/index.html> (accessed on 30 December 2018).
26. Google Code Project. Multipath-TCP: Implement Multipath TCP on NS-2. Available online: <http://code.google.com/p/multipath-tcp/> (accessed on 30 December 2018).
27. Hwang, J.; Walid, A.; Yoo, J. Fast Coupled Retransmission for Multipath TCP in Data Center Networks. *IEEE Syst. J.* **2018**, *12*, 1056–1059. [[CrossRef](#)]
28. Jacobson, V.; Karels, M. Congestion Avoidance and Control. In Proceedings of the ACM SIGCOMM, Stanford, CA, USA, 16–18 August 1988; pp. 314–329.
29. Paasch, C.; Ferlin, S.; Alay, O.; Bonaventure, O. Experimental Evaluation of Multipath TCP Schedulers. In Proceedings of the the ACM SIGCOMM Workshop on Capacity Sharing Workshop, Chicago, IL, USA, 17–22 August 2014; pp. 27–32.
30. Tzeng, G.; Huang, J. *Multiple Attribute Decision Making: Methods and Applications*; CRC Press: Boca Raton, FL, USA, 2011; pp. 55–56.
31. Ramirez-Perez, C.; Ramos-R, V.M. On the effectiveness of multi-criteria decision mechanisms for vertical handoff. In Proceedings of the IEEE 27th International Conference on Advanced Information Networking and Applications, Barcelona, Spain, 25–28 March 2013; pp. 1157–1164.
32. Song, F.; Zhou, Y.; Liu, C.; Zhang, H. Modeling Space-Terrestrial Integrated Networks with Smart Collaborative Theory. *IEEE Netw.* **2019**, *33*, 51–57. [[CrossRef](#)]
33. Ai, Z.; Zhou, Y.; Song, F. A Smart Collaborative Routing Protocol for Reliable Data Diffusion in IoT Scenarios. *Sensors* **2018**, *18*, 1926. [[CrossRef](#)]
34. Xu, C.; Li, Z.; Li, J.; Zhang, H.; Muntean, G. Cross-layer Fairness-driven Concurrent Multipath Video Delivery over Heterogenous Wireless Networks. *IEEE Trans. Circ. Syst. Video Technol.* **2015**, *25*, 1175–1189.
35. Paasch, C.; Weber, D.; Baerts, M.; Froidcoeur T.; Hesmans B. Multipath TCP in the Linux Kernel. Available online: <https://www.multipath-tcp.org> (accessed on 30 December 2019).
36. Song, F.; Zhu, M.; Zhou, Y.; You, I.; Zhang, H. Smart Collaborative Tracking for Ubiquitous Power IoT in Edge-Cloud Interplay Domain. *IEEE Internet Things J.* **2019**. [[CrossRef](#)]
37. Ai, Z.; Liu, Y.; Song, F.; Zhang, H. A Smart Collaborative Charging Algorithm for Mobile Power Distribution in 5G Networks. *IEEE Access* **2018**, *6*, 28668–28679. [[CrossRef](#)]
38. Song, F.; Zhou, Y.; Wang, Y.; Zhao, T.; You, I.; Zhang, H. Smart Collaborative Distribution for Privacy Enhancement in Moving Target Defense. *Inf. Sci.* **2019**, *479*, 593–606. [[CrossRef](#)]
39. Quan, W.; Liu, Y.; Zhang, H.; Yu, S. Enhancing Crowd Collaborations for Software Defined Vehicular Networks. *IEEE Commun. Mag.* **2017**, *55*, 80–86. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).