IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 19, 2017, accepted July 11, 2017, date of publication July 25, 2017, date of current version November 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2731899

A LDDoS-Aware Energy-Efficient Multipathing Scheme for Mobile Cloud Computing Systems

YUANLONG CAO[1], FEI SONG[2], QINGHUA LIU[1], MINGHE HUANG[1], HAO WANG[1], AND ILSUN YOU[3], (Senior Member, IEEE)

[1] School of Software, Jiangxi Normal University, Nanchang 330022, China
[2] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[3] Department of Information Security Engineering, Soonchunhyang University, Asan 31538, South Korea

Corresponding author: Ilsun You (ilsunu@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61562044 and in part by the Soonchunhyang University Research Fund. *(Yuanlong Cao and Fei Song contributed equally to this work.)*

ABSTRACT The multi-homed mobile devices in the mobile cloud computing (MCC) systems can improve their throughput by allocating the application data over several paths simultaneously, enabled by the promising Multipath TCP (MPTCP) technology. Meanwhile, network attacks against current Internet infrastructures are likely to increase, especially with the widely deployment of MCC systems. When a MPTCP connection is under network attacks and becomes a poor-performing path or a broken path, it can significantly affect other stable paths and in the absence of related schemes to handle this, MPTCP will undoubtedly suffer from serious performance degradation. Moreover, applying MPTCP to cloud data delivery may generally lead to higher energy consumption and is not favorable to a power-constrained mobile device. In this paper, we propose MPTCP-La/E^2, a low-rate distributed denial-of-service (LDDoS) attack-aware energy-efficient MPTCP solution aiming at: 1) avoiding the LDDoS-caused performance degradation of cloud multipath transmission, which has been seldom considered in existing MPTCP solutions and 2) optimizing the energy usage while still maintaining user's perceived quality of cloud multipathing services. The simulation results show that MPTCP-La/E^2 outperforms the baseline MPTCP in terms of QoS and energy-savings in a multi-homed MCC network environment.

INDEX TERMS Mobile cloud applications, Multipath TCP, low-rate DDoS attacks, failure detection, energy usage.

I. INTRODUCTION

With the ever-growing demand for computation-intensive mobile applications, and the widespread availability of both fixed wireless and mobile cellular access systems, the convergence of several heterogeneous wireless access networks (e.g., WiMax, Wi-Fi, LTE, etc.) with high-quality mobile cloud computing (MCC) and ubiquitous Internet access services are important characteristics of next-generation mobile Internet [1], [2]. Meanwhile, promoted by the dramatic development of wireless and mobile cellular technologies, growing numbers of MCC devices will be embedded with more than one network interface and attached heterogeneous multi-access capability at the same time [3], [4]. Such multi-homed mobile devices can boost the download speed of large cloud data and improve their throughput by using of several network links simultaneously, enabled by the emerging Multipath TCP (MPTCP) technology [5].

The MPTCP is a TCP extension that supports concurrent use of multiple network interfaces and allows the cloud applications to simultaneously exploit multiple access links for data transmission [6], [7]. Fig. 1 illustrates a typical MPTCP use case in a MCC system that involves a multi-homed mobile terminal (an MPTCP receiver) and a multi-homed cloud application server (an MPTCP sender), who communicate with each other using both cellular data network and wireless Wi-Fi network simultaneously. Such MPTCP-based multipathing and bandwidth aggregation features are not only beneficial for a MCC system to improve the application-level goodput performance, but also able to enhance network-level connectivity robustness [8]. Moreover, MPTCP presents the same socket APIs as the regular TCP and is backward-compatible with today's cloud applications. The backward-compatibility nature ensures the success of MPTCP in the current and future cloud networks [9].

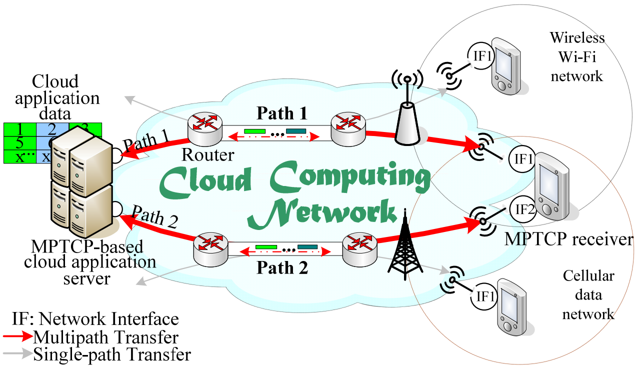


FIGURE 1. A typical MPTCP use case in a mobile cloud network.

Despite MPTCP brings many attractive benefits to MCC applications and is recognized as a promising technology for data delivery, it still has many concerns and challenges to be addressed. The first important concern of MPTCP-based data delivery in cloud computing environment is related to preventing the usage of poor-performing paths in the multipath transmission. When applying MPTCP to cloud multipath transmission, each path independently performs cloud traffic delivery according to its own QoS-related networking parameters; nevertheless, allocating cloud application data on a poor-performing path can cause transmission interruptions in the stable paths [10]. That is, a poor-performing path within the multipath transmission session can significantly affect other paths and degrades the application-level performance [11]. Therefore, how to declare a poor-performing path and constrain it from transmitting cloud application data is an important topic for MPTCP-based cloud multipathing services.

Recently, more and more researchers have devoted themselves to the research of poor-performing path detection and multipath management mechanisms [12]–[19]. However, their proposers have ignored that the MPTCP performance can be largely degraded due to cyber attacks, especially low-rate distributed denial-of-service (LDDoS) attacks, who can exploit the vulnerability in MPTCP's TCP-like retransmission timeout (RTO) mechanism to attack a MPTCP subflow (a TCP connection) [20]. Especially with the development of cloud computing, more and more network security studies show that the LDDoS attacks will likely to become more prevalent on future Internet due to its intrinsic natures of relative low-rate and ingenious concealment [21]. Therefore, the design of multipath protocols without considering the LDDoS attacks does not fully investigate the MPTCP performance that are likely to be achieved when it is applied to a MCC system [22].

Another important concern when applying MPTCP to MCC is the high energy overhead caused by the multipath data transmission. By making use of multiple network resources, MPTCP provides the MCC devices not only with a good goodput performance but also with a high energy cost. Since today's mobile devices have in general very limited power capacity in their batteries, the energy usage becomes an urgent topic worth in-depth investigation. More recently, many academic researchers have concentrated their efforts on the MPTCP energy usage optimization [23]–[28]. To summarize, their solutions mostly trade goodput performance for lower energy consume and longer battery life by shifting the packets required to be sent from a higher energy cost path onto a lower one. However, all well-known solutions do not take into account network attacks-caused transmission behaviors and path states, as we will discuss later.

In this paper, we propose a novel LDDoS attack-aware energy-efficient MPTCP solution (dubbed as MPTCP-La/E^2) for multi-homed MCC systems in order to address the issues of LDDoS attacks and mobile energy consumption. The goals of MPTCP-La/E^2 are: (i) to avoid the performance degradation of cloud multipath transmission caused by LDDoS attacks, and (ii) to optimize the energy usage while still maintaining user's perceived quality of cloud multipathing services. The proposed MPTCP-La/E^2 has been evaluated with a wide range of performance metrics. The simulation results demonstrate how MPTCP-La/E^2 outperforms the baseline MPTCP in terms of QoS and energy-savings. More specifically, MPTCP-La/E^2 makes important contributions in the following aspects:

- It is the first study to explore the influence of the famous LDDoS attacks on the MPTCP performance and provides MPTCP with a LDDoS-aware multipath management mechanism.
- It takes into consideration LDDoS-caused transmission behaviors and path states, and thereby introduces an energy-efficient MPTCP-based cloud data scheduling algorithm.

II. PROBLEM STATEMENT

In a MPTCP-based cloud multipathing system, each path has its own congestion window (*cwnd*), and the sender runs the classic TCP NewReno congestion control mechanisms for each path separately. However, these paths within the MPTCP connection do not work alone but influence each other due to MPTCP's fully-ordered data delivery services. At the same time, it is widely recognized that LDDoS attacks against current network infrastructures are likely to increase, especially when the cloud computing is widely applied [29]. If a network path is under LDDoS attacks, it can frequently and abruptly experience transmission interruptions. This situation can be even worse in MPTCP because other stable paths within the connection are likewise disturbed because of the transmission interruption in the attacked path. Correspondingly, significant application-level performance degradations will occur.

To investigate the impact of LDDoS attacks on the performance of MPTCP, a basic dual-dumbbell simulation topology with reasonable LDDoS attack traffic is developed in Network Simulator 3 version 15 (NS-3) [30], which is illustrated in Fig. 2. In the topology, the MPTCP sender and receiver connect to the MCC network through two network interfaces. Each router on path A (namely, $R_{1,1}$ and $R_{1,2}$) is attached

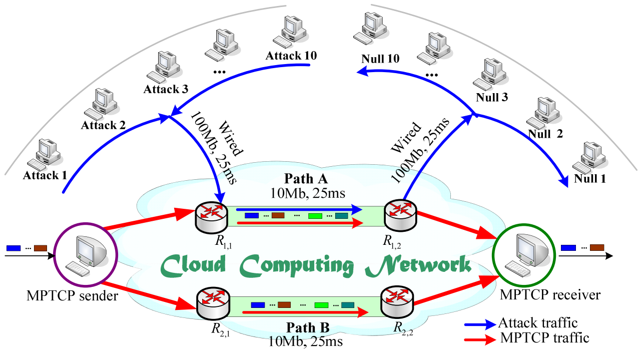


FIGURE 2. A basic dual-dumbbell simulation topology with LDDoS attacks.

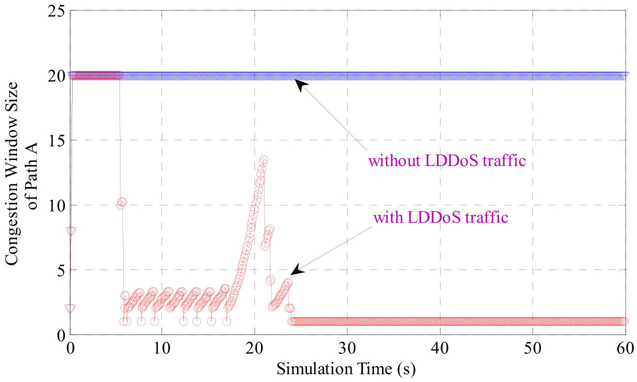


FIGURE 3. The congestion window size of Path A with or without a LDDoS attack.

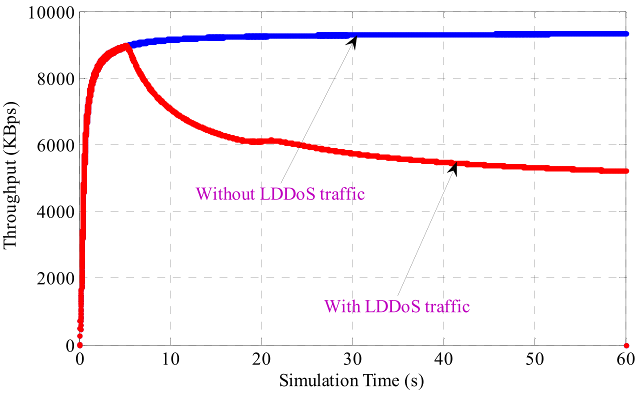


FIGURE 4. The throughput comparison with or without a LDDoS attack.

with 10 edge nodes. These edge nodes are equipped with a single network interface and connect to the routes to generate the LDDoS traffic. The bandwidth between each edge node and its connected router is set to 100Mb with 25ms of propagation delay. The bandwidth between $R_{1,1}$ and $R_{1,2}$, as well as $R_{2,1}$ and $R_{2,2}$ is set to 10Mb with 25ms of propagation delay. The total simulation time is 60 seconds.

Since a LDDoS attack usually leverages the UDP protocol with the Constant Bit Rate (CBR) traffic, all the attackers (*Attacker* 1, *Attacker* 2, $\cdots$, *Attacker* 10) generate UDP/CBR packets and begin their attack at 5.1th second of simulation time. The following triple is used to describe the characteristics of LDDoS attacks,

$$LDDoS\,(T,\ L,\ R) = LDDoS\,(100\text{ ms}, 100\text{ ms}, 1\text{ Mbps})\,, \tag{1}$$

which $T, L,$ and$R$ are the *attack period*, the width of attack pulse (*attack duration*), and the intensity of attack pulse (*attack rate*), respectively. The settings of T, L, and R are reasonable and ensure the LDDoS traffic can deny bandwidth to normal MPTCP flows while avoiding being detected by counter-DoS solutions. The readers who are interested in the LDDoS attacks can read and get more information from [31].

Fig. 3 shows the *cwnd* size of path A with and without the LDDoS traffic, respectively. As the figure shows, the *cwnd* value of path A decreases sharply when LDDoS attacks have been launched (after 5.1 seconds of simulation). This is because that the LDDoS attacks can exploit the MPTCP's TCP-like RTO mechanism and make the MPTCP sender repeatedly experience a RTO event on path A. And what's worse is, the path A's *cwnd* is set to one segment frequently due to the occurrence of timeout. Fig. 4 presents the overall throughput performance of MPTCP when the LDDoS attackers are enabled and disabled, respectively. From the figure, we can see that the MPTCP throughput performance decreases sharply at the start of LDDoS attacks. This is because that the LDDoS packets attempt to deny bandwidth to the MPTCP subflow (a TCP connection) on path A, and cause a huge path dissimilar between the attacked path A and other stable paths. Unfortunately, MPTCP has not a related mechanism to prevent the usage of poor-performing paths (the attacked paths) in multipath transmission.

Motivated by the fact that an effective multipath management mechanism including a poor-performing path declaring scheme is very beneficial to the MPTCP-based cloud multipathing services, this paper introduces a LDDoS-aware multipath management mechanism by jointly considering the intrinsic characteristics of MPTCP and LDDoS attacks. The goals of our proposed mechanism are (i) to possibly declare a poor-performing (attacked) path and prevent the usage of these paths in the multipath transmission timely, (ii) to possibly enhance the MPTCP throughput performance in a LDDoS-recurrent mobile cloud network environment. This paper also includes a proposal for saving the energy of MPTCP-based MCC systems.

III. THE DETAILED DESIGN OF MPTCP-La/E^2

Fig. 5 shows the architecture of the proposed MPTCP-La/E^2, which includes a multi-homed cloud application server (an MPTCP sender), a multi-homed mobile device (an MPTCP receiver), and multiple asymmetric paths. On the MPTCP receiver side, the arrival packets will be buffered and re-ordered if the cloud application data is split into a number of segments by the sender. While on the sender

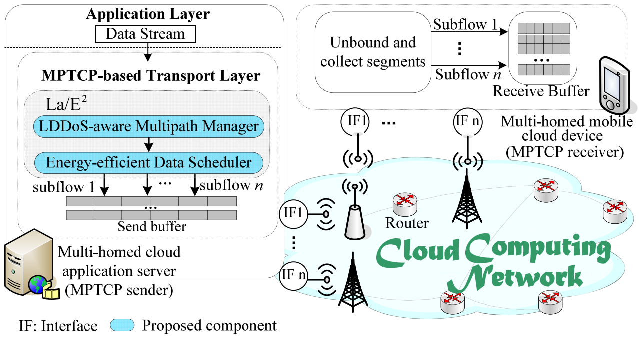

Application Layer
Data Stream
MPTCP-based Transport Layer
La/E^2
LDDoS-aware Multipath Manager
Energy-efficient Data Scheduler
subflow 1
…
subflow n
…
Send buffer
Multi-homed cloud application server (MPTCP sender)
Unbound and collect segments
Subflow 1
Subflow n
…
Receive Buffer
IF1
…
IF n
Multi-homed mobile cloud device (MPTCP receiver)
IF1
IF n
Router
Cloud Computing Network
IF: Interface
Proposed component

FIGURE 5. The architecture of MPTCP-La/E^2.

side, there are two newly added components, which are LDDoS-aware multipath manager (LaM^2) and energy-efficient data scheduler (E^{2}DS). The functions of the two components are outlined below:

- The LaM^2 is dedicated to monitoring the transmission quality of each MPTCP path, switching a path to a proper state, and choosing a subset of suitable paths for multipathing
- The E^{2}DS is devoted to detecting the energy cost of each path, adaptively achieving bandwidth aggregation and energy-savings by jointly considering per-path's transmission state and energy cost.

A. LDDoS-AWARE MULTIPATH MANAGER

As analyzed in Section II, a path under a LDDoS attack can frequently encounter timeout and easily become a broken path. However, MPTCP path management is very simple and just inherits TCP operations for broken path detection. As a result, the attacked and broken path may still be considered as "active" and used for data (re)transmission until the TCP keep-alive time has expired, which needs quite a long time to declare a broken path and thus leads to serious problems: (i) *unnecessary retransmissions*. When new data is allocated to a broken path for transmission, the sender will inevitably perform unnecessary retransmissions via the broken path; and (ii) *performance degradation*. The transmission interruptions in the broken path will undoubtedly affect the transmission efficiency of other stable paths and degrade the performance of MPTCP.

MPTCP can apply the maximum number of retransmissions (*TcpMaxDataRetransmissions*) to monitor each path's condition [5], [32]. The *TcpMaxDataRetransmissions* controls the number of times that a data segment can be retransmitted by the sender. If the retransmission count (*rc*) on a path reaches the value of *TcpMaxDataRetransmissions*, then MPTCP changes the path to the "inactive" state. However, the *rc*-based failure detection mechanism, also called as "*single sample-based failure detection mechanism*", is bound to two "hot potato" problems: (i) the time required to declare a broken path need at least $1+3+6+12+24 = 46$ s (when TcpMaxDataRetransmissions $= 5$ and RTO $= 1$ s) or more, which may be too long; and (ii) it can only detect continuous timeout events and complete path failures. Fig. 6 presents the pseudo code of *rc*-based path failure detection algorithm.

```
set retransmission count ($rc$) =0;
set path state ($ps$) ="active";

for each path within the MPTCP connection do
    if the retransmission timer expires then
        set $rc$=$rc$+1;
    end if
    if ($rc \geq TcpMaxDataRetransmissions$) then
        mark $ps$="inactive";
    end if
end for
```

FIGURE 6. The retransmission count-based path failure detection algorithm.

The main idea of LaM^2 is to improve on MPTCP by identifying a broken path as quickly as possible and preventing it from being further utilized in the multipath transmission. To this end, LaM^2 introduces a "path error count" (*pec*) concept and a new "*potentially broken*" state to MPTCP. The idea of pec calculation in LaM^2 comes from TCP's *exponential backoff* mechanism [33], in which the time between retransmissions is doubled each time. That is, each time when TCP retransmits a packet, it sets:

$$RTO_i = 2 \times RTO_{i-1}, \quad \text{subject to } 0 \leq RTO_i \leq M, \qquad (2)$$

which $RTO_{i\text{ - }1}$andRTO_i are the $(i-1)^{th}$ and i^{th} timeout interval, respectively. M is a specified upper bound of the RTO. The recommended value for this parameter is 64 [33].

Since the TCP's *exponential backoff* mechanism increases the retransmission interval (the RTO value) at double once having a retransmission to be launched, inspired by this "*back off the timer*" feature, LaM^2 also increases the *pec* value at double each time when the retransmission timer expires. Therefore, the use of pec can cause a path under LDDoS attacks and with continuous or burst timeout to quickly reach the *TcpMaxDataRetransmissions* value, which the standard TCP, also MPTCP does not do:

$$pec = 2 \times \text{retransmissioncount}. \qquad (3)$$

In MPTCP-La/E^2, when the *pec* value on a path reaches the value of *TcpMaxDataRetransmissions*, namely,

$$pec \geq TcpMaxDataRetransmissions, \qquad (4)$$

the path will be marked as "*potentially broken*" state. MPTCP-La/E^2 does not use any *potentially broken* path for application data transmission, it only sends probe packets (e.g., TCP keep-alive probe packet [33]) to further detect the connectivity of the *potentially broken* paths. In this way, MPTCP-La/E^2 can not only timely detect different kinds (continuous or burst) of timeout, but also possibly shorten the time for path state detection and transition.

Suppose there are n possible paths ($p_1, p_2, \cdots, p_n$) within the MPTCP connection, and let us take path p_ψ ($1 \leq \psi \leq n$) for example, LaM^2 declares and transits the state of each path by following the operations below:

1) When MPTCP is initiated, the state of p_ψ is marked as "active", the *rc* and *pec* are set to zero;
2) Each time when the retransmission timer expires, the *rc* value is incremented by 1, and the *pec* value is multiplied by 2, respectively;
3) When the *pec* value reaches the *TcpMaxDataRetransmissions*, but the *rc* value does not exceed the value of *TcpMaxDataRetransmissions*, the state of p_ψ is changed from "active" to "potentially broken";
4) When the *rc* value reaches the value of *TcpMaxDataRetransmissions*, the *pec* value returns to zero, and the state of p_ψ is changed from "potentially broken" to "inactive";
5) If the sender receives an ACK for any probe packet, it switches the state of p_ψ from "potentially broken" or "inactive" to "active".

Fig. 7 shows the *pec*-based path failure detection algorithm in LaM^2.

```
set retransmission count (rc) =0;
set path error count (pec) =0;
set path state (ps) =“active”;
for each path within the MPTCP connection do
    if the retransmission timer expires then
        set rc=rc+1;
        set pec=2×rc;
    end if
    if ( $pec \geq TcpMaxDataRetransmissions$ ) &
       ( $rc < TcpMaxDataRetransmissions$ ) then
        mark ps=“potentially broken”;
    end if
    if ($rc \geq TcpMaxDataRetransmissions$) then
        mark ps=“inactive”;
    end if
end for
```

FIGURE 7. The path error count-based path failure detection algorithm.

B. ENERGY-EFFICIENT DATA SCHEDULER

As mentioned previously, applying MPTCP to a MCC mobile device raises a new concern, namely, the high energy consumption for concurrent use of multiple network interfaces. Meantime, today's mobile devices have very limited supply of power due to the battery storage technology [22]. Unfortunately, the baseline MPTCP only knows the connectivity of paths (i.e., TCP keep alive) and not their energy cost [23]–[25], it just simply makes fully use of multiple network interfaces for concurrent multipath transmission, without considering the energy consumption of each path.

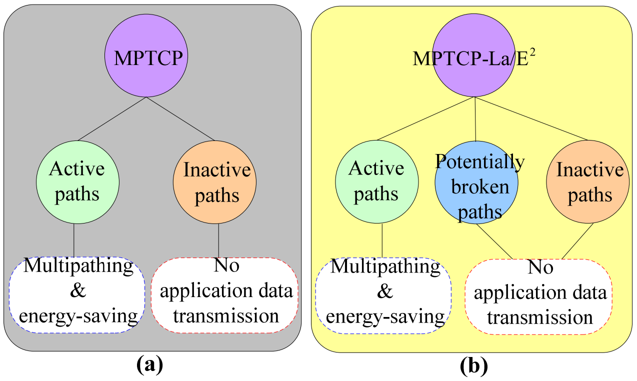

MPTCP
Active paths
Inactive paths
Multipathing & energy-saving
No application data transmission
(a)
MPTCP-La/E^2
Active paths
Potentially broken paths
Inactive paths
Multipathing & energy-saving
No application data transmission
(b)

FIGURE 8. Multiple paths which are used in (a) MPTCP and (b) MPTCP-La/E^2 for multipathing and energy-saving.

The "blind" energy usage in MPTCP can increase the energy consumption of mobile devices and shorten their life cycle. In order to better run MPTCP on a battery power-limited mobile device, it is necessary to optimize the multipath usage and make MPTCP more energy-efficient.

In fact, timely preventing the usage of poor-performing paths (e.g., the paths under LDDoS attacks) and muting the corresponding network interfaces may possibly reduce the energy consumption in the MPTCP. However, the standard MPTCP only knows an *active* or *inactive* path, it cannot distinguish the poor-performing paths, and thus further cannot mute the corresponding network interfaces for energy-saving. The goal of E^{2}DS is to optimize MPTCP's scheduler and help MPTCP be more energy efficient, by jointly considering the transmission state (*active*, *potentially broken*, or *inactive*) and the energy cost of each path. More precisely, in addition to the paths marked as *inactive*, E^{2}DS also prevents the paths with *potentially broken* state from being further used by MPTCP.

Fig. 8 shows multiple paths that are used by (a) the MPTCP scheduler, and (b) the MPTCP-La/E^2 scheduler for multipathing and energy-saving. To explain the figure in more detail, let us also suppose that there are n paths ($p_1, p_2, \cdots, p_n$) within the MPTCP connection, and let PS_{active}, $PS_{potentially\ broken}$ and $PS_{inactive}$ be the set of *active*, *potentially broken*, and *inactive* paths, respectively. In the existing MPTCP solutions, the scheduler generally uses the following subset of paths for bandwidth aggregation and assigns them appropriate application data traffic for energy saving,

$$PS_{multipathing} = PS_{active} \cup PS_{potentially\ broken}, \tag{5}$$

which $PS_{multipathing}$ is a subset of paths that is used by the sender for multipathing. While in the proposed MPTCP-La/E^2 solution, the scheduler can use the following subset of paths for bandwidth aggregation and energy saving,

$$PS_{multipathing} = PS_{active}. \tag{6}$$

In addition to muting the potentially broken network interfaces for application data delivery and energy saving, E^{2}DS also moves application data as much as possible from a higher

```
/* Once having application data to be sent */
for $(i=1, i \le count(p_{list}), i++)$ do
    /* use LaM$^2$ to confirm the state of path $p_i$ */
    /* ignore the inactive paths */
    if $(pec_{p_i} < TcpMaxDataRetransmissions)$ then
        put $p_i$ into $p_{list}^{active}$;
    end if
    if$(pec_{p_i} \ge TcpMaxDataRetransmissions)$&
      $(rc_{p_i} < TcpMaxDataRetransmissions)$ then
        put $p_i$ into $p_{list}^{PB}$ ;
    end if
end for
for $(j=1, j \le count(p_{list}^{active}), j++)$ do
    sort $p_j$ in an ascending order according to its $p_j^{ec}$ value;
    if $(ABcwnd_{p_{list(0)}^{active}} > 0)$ then
        offload $\min(ABcwnd_{p_{list(0)}^{active}}, SD_{size})$ amount of data to $p_{list(0)}^{active}$;
    else
        /*use the second lowest energy cost path for data delivery*/
        set $p_{list(0)}^{active} = p_{list(0)}^{active} \rightarrow next$;
    end if
end for
/*in order to avoid an exception*/
if count$(p_{list}^{active})=0$ then
    use the lowest energy-cost path within $p_{list}^{PB}$ for data delivery;
end if
```

FIGURE 9. The E^{2}DS-based energy-efficient data scheduling algorithm.

energy-consuming path to a lower one in order to attain the purpose of energy-savings. The amount of application data that can be offloaded to the energy-efficient paths depends on each energy-efficient path's own *cwnd* value. In MPTCP, as we have already known, each of the individual path has its own *cwnd* to limit the total number of data the sender can assign to it (for congestion control). In MPTCP-La/E^2, the sender controls the maximum amount of data that can be assigned to a particular path p_τ, by using the following equation,

$$ABcwnd_{p_\tau} = cwnd_{p_\tau} - outstanding_{p_\tau}, \tag{7}$$

which $outstanding_{p_\tau}$ is the amount of sent but not yet acknowledged data on path p_τ. $ABcwnd_{p_\tau}$ is the available *cwnd* value of p_τ. E^{2}DS uses this parameter to control the data amount that allowed allocating or offloading to p_τ.

By jointly considering both the transmission state and *ABcwnd* value of each path, E^{2}DS can achieve potential energy-savings, while possibly avoiding the occurrence of network congestion in the multipath transmissions. Moreover, in order to avoid an exception, if all the paths are marked as "*potentially broken*", the path with the lowest energy cost will be used by the sender for data delivery. This operation can ensure that MPTCP-La/E^2 achieves better energy-savings compared to MPTCP when all paths within the MPTCP connection have possibly been attacked. Fig. 9 presents the E^{2}DS-based energy-efficient data scheduling algorithm used in MPTCP-La/E^2. We also define some notations in order to make convenience for understanding the algorithm, as illustrated in Table 1.

TABLE 1. Notations used in the E^{2}DS-based energy-efficient data scheduling algorithm.

Notation	Description
p_{list}	All the paths within the MPTCP connection
p_i	The i^{th} path within the MPTCP connection
pec_{p_i}	The path error count on p_i
rc_{p_i}	The retransmission count on p_i
p_{list}^{active}	The active path collection in the MPTCP
$p_{list(0)}^{active}$	The first path within the p_{list}^{active}
p_{list}^{PB}	The potentially broken path collection
$ABcwnd_{p_{list(0)}^{active}}$	The available *cwnd* size of $p_{list(0)}^{active}$
SD_{size}	The size of remaining application data in the buffer

IV. PERFORMANCE EVALUATION

In this section, we evaluate and analyze the performance and QoS of our MPTCP-La/E^2 solution for traditional FTP-like application data transmission and video content delivery, respectively. In the simulations, we have evaluated two versions of the MPTCP-La/E^2 solution against the baseline MPTCP. The two versions of our solution are: MPTCP-La, in which only *LDDoS-aware multipath manager* component is used and MPTCP-La/E^2, in which both the *LDDoS-aware multipath manager* and *Energy-efficient data scheduler* are deployed.

A. SIMULATION TOPOLOGY

The performance of our solution has been evaluated by using the NS-3 [30]. The simulation topology considered a multi-homed heterogeneous cloud network condition which is presented in Fig. 10. Both the two MPTCP endpoints access the cloud system through three asymmetric wireless access links and then intercommunicate with each other via the wired link. The three asymmetric paths (denoted A, B and C) with different network-related parameters. Path A is set with 10Mbps bandwidth, 10-20ms propagation delay (representative for an IEEE 802.16 link), path B is set with 11Mbps bandwidth with 10-20ms propagation delay (representative for an IEEE 802.11b link), and path C is set with 2Mbps bandwidth with 50-60ms propagation delay (representative for an IEEE 802.11 link). Table 2 shows the major configurations of the three paths. The configurations of MPTCP are set with default parameter values provided by the NS-3 MPTCP patch.

Like [34], each of the three access links is attached with two loss models, which are Uniform loss model that is used to represent the infrequent continuous loss caused by random contention, and two-state Markov loss model that is used to represent the distributed loss caused by transient failure or stream burst. In addition, the access links of path A and B are attached with a Variable Bit Rate (VBR) traffic

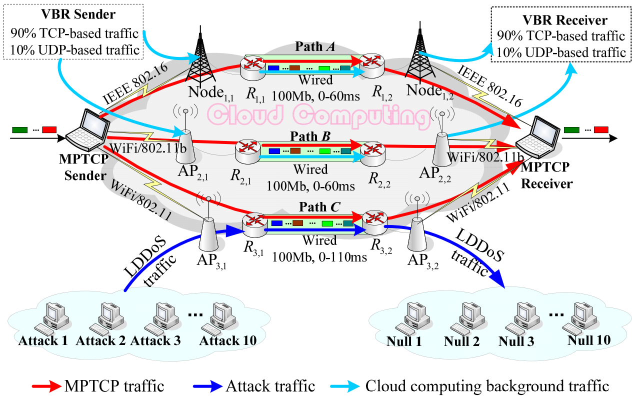


FIGURE 10. A heterogeneous cloud network simulation topology with LDDoS attacks.

TABLE 2. Path parameters used in the simulation.

Parameters	Path A	Path B	Path C
Access link standard	802.16	802.11b	802.11
Access link bandwidth	10Mbps	11Mbps	2Mbps
Access link delay	0-30ms	0-30ms	50-60ms
Access link queue type	Droptail	Droptail	Droptail
Access link queue limit	50 packets	50 packets	50 packets
Cloud network bandwidth	100Mbps	100Mbps	100Mbps
Cloud network delay	0-60ms	0-60ms	0-110ms
Uniform loss-rate	1%-3%	2%-4%	1%-10%
Markov loss rate	1%	1%	1%

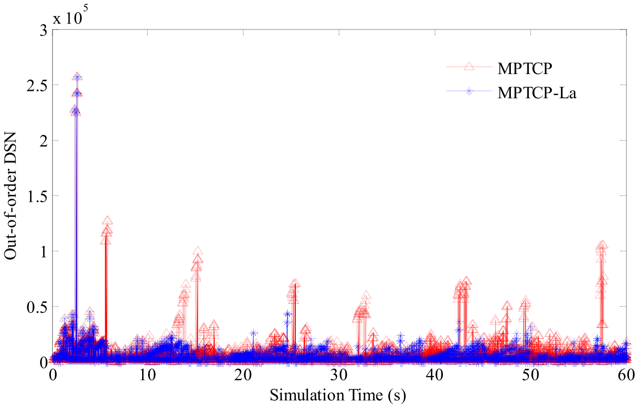


FIGURE 11. Comparison of out-of-order DSN.

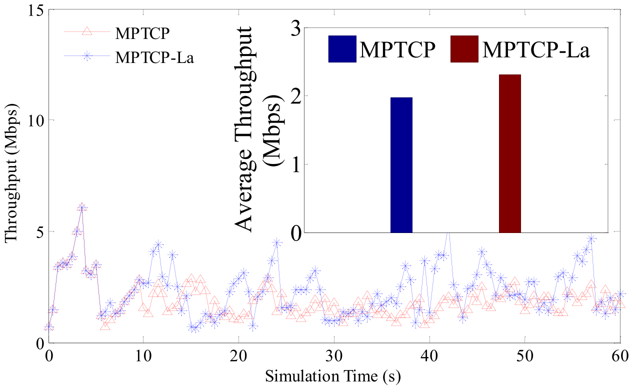


FIGURE 12. Comparison of throughput.

generator in order to simulate the cloud background traffic. The packet size of VBR traffic is as follows: 46 percent are 1500 bytes long, 1.2 percent have 576 bytes, 1.7 percent are 1300 bytes, 2.1 percent are 628 bytes, and the other 49 percent are 44 bytes long, in which 90% of the background traffic are carried by TCP and the rest 10% are sent by UDP connections [34]. The background traffic on the two paths occupies randomly between 0-50% of the access bandwidth. The access link of path C is attached with ten CBR traffic generators in order to simulate the LDDoS attack traffic. All the generators begin their attack at 0.1th second of simulation time and inject the LDDoS traffic with $T = 400$ms, and $L = 200$ms. For the attack pulse R, it varies randomly between 0.15-0.25Mbps. The total simulation time is 60 seconds.

B. SIMULATION RESULTS

1) OUT-OF-ORDER DATA SEQUENCE NUMBER

The out-of-order data sequence number (DSN) can be estimated by the offset between the DSNs of two segments consecutively received by the receiver. Fig. 11 presents the comparison of out-of-order DSN when MPTCP and MPTCP-La are used, respectively. As shown in the figure, the baseline MPTCP generates more out-of-order DSN and thus requires larger packet reordering delay than MPTCP-La. This is because the data scheduler of MPTCP splits application packets over all paths within the connection, it cannot declare a broken-prone path (e.g., a path under LDDoS attacked) and further constrain it from transmitting application data. The data transmission delay differences between the broken-prone path and the stable paths will cause large numbers of out-of-order data arrivals and increase the data reordering delay. In contrast, MPTCP-La only transmit application data over the selected paths while constrain the potentially broken paths from data delivery. Therefore, MPTCP-La can provide stable performance and ensure data possible in-order arrive. The average out-of-order data reception at the receiver side is about 4544 and 3657 when using MPTCP and MPTCP-La, respectively.

2) AVERAGE THROUGHPUT

Fig. 12 presents the comparison of throughput performance when the MPTCP and MPTCP-La solutions are used, respectively. Since MPTCP utilizes all the available paths for data transmission, the overall application-level throughput might be degraded if one or more paths are unstable. In contrast, MPTCP-La includes a LDDoS-aware path failure detection mechanism, it can quickly determine a potentially path and timely prevent a potentially broken path from being further used in the multipath transmission. These features make

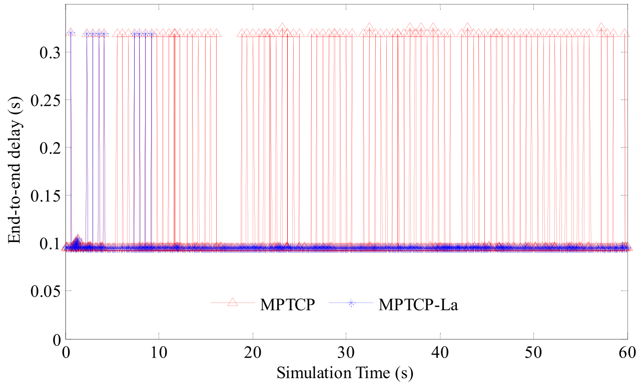


FIGURE 13. Comparison of end-to-end delay.

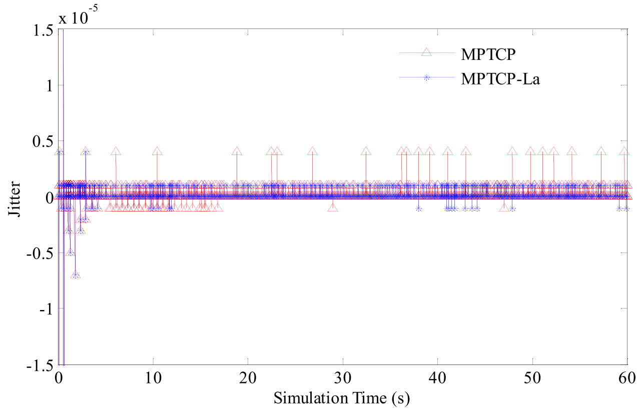


FIGURE 14. Comparison of jitter.

MPTCP-La avoid the throughput performance degradation caused by LDDoS attacks. The subfigure in Fig. 12 shows the cumulative average throughput of the two solutions with a total 60 seconds simulation time. As the subfigure shows, the cumulative average throughput of MPTCP-La is about 22.14% higher than that of MPTCP.

3) END-TO-END DELAY

Fig. 14 presents the comparison of end-to-end delay performance when the standard MPTCP and MPTCP-La solutions are used, respectively. As we discussed earlier, MPTCP-La monitors each path's *pec* value and prevents the usage of a broken-prone path in the multipath transmission if the *pec* value of the path reaches the value of *TcpMaxDataRetransmissions*, instead of waiting for five consecutive retransmission timeout expirations. Therefore, MPTCP-La can timely deactivate a broken-prone path for multipath transmission and possibly alleviate unnecessary retransmissions via the broken path. These features help MPTCP-La avoid allocating application data over the broken paths and reduce the overall transmission delay in multipath transmission. From Fig. 14, it can be observed that MPTCP-La achieves a low-level end-to-end delay performance compared to MPTCP. The cumulative average delay of MPTCP-La is approximately 15.76% lower than that of MPTCP.

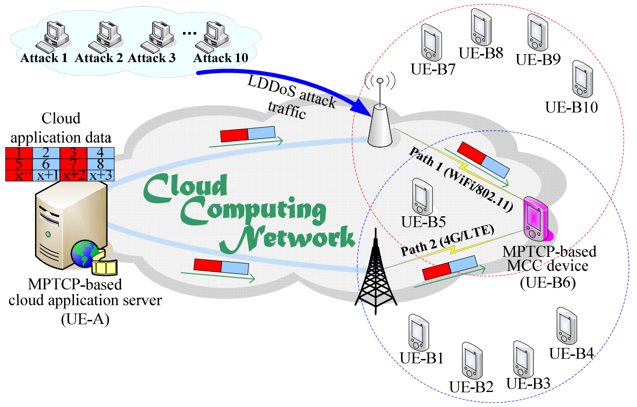


FIGURE 15. A heterogeneous wireless cloud network topology with LDDoS attacks.

4) JITTER COMPARISON

Jitter is a data delivery latency variation caused by network attacks, network failures, or other networking activities effects on transmission performance. It can be calculated by the latency variation between the DSNs of two segments consecutively received by the receiver. Jitter has been widely recognized as one key metric and should be considered when evaluating and analyzing the performance of multipath transport technologies. Higher level of jitter is more likely to occur on a more underperforming multipath technology and vice versa. Fig. 13 shows the comparison results of the jitter performance when using the MPTCP and MPTCP-La solutions, respectively. Because of its efficient *path error count*-based path failure detection mechanism, MPTCP-La can detect a broken path and choose a subset of stable paths for multipath transmission. Correspondingly, it maintains the latency variation at a lower level and thus outperforms the standard MPTCP scheme in terms of jitter performance.

V. ENERGY EFFICIENCY TESTING

A. SIMULATION TOPOLOGY

Since modern mobile devices (e.g., smartphones) are already embedded with wireless Wi-Fi and 4G LTE cellular interfaces simultaneously [35], [36], thus we consider that a multi-homed mobile device (referred to as UE-B6) is communicating with a cloud application server (referred to as UE-A) by concurrently using two asymmetric paths, which are path 1 that is a Wi-Fi/IEEE 802.11 link with 2Mbps bandwidth and 5-15ms propagation delay, and path 2 that is a 4G/LTE link with 100Mbps bandwidth and 100ms propagation delay [37], [38]. The distance between the Wi-Fi base station and UE-B6 as well as the LTE base station and UE-B6, is set to 60 meters. The initial battery energy of UE-B6 is 300 Joules. Moreover, we adopt the Log-normal Shadowing Model [39] to simulate the signal attenuations occurred in wireless transmission channels. Fig. 15 presents the simulations topology.

Like previous work [40], we inject Internet cross traffic generated by nine single-interface mobile terminals

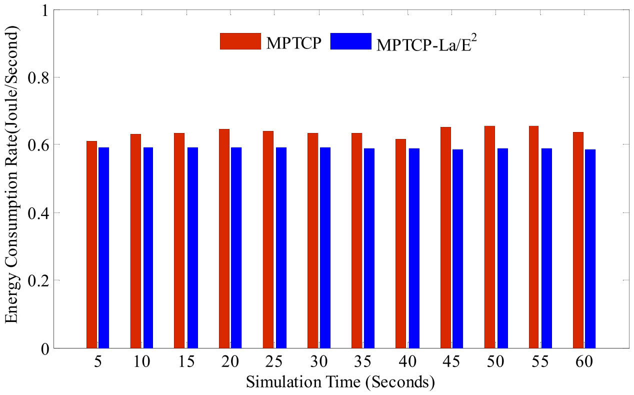


FIGURE 16. Comparison of energy consumption rate.

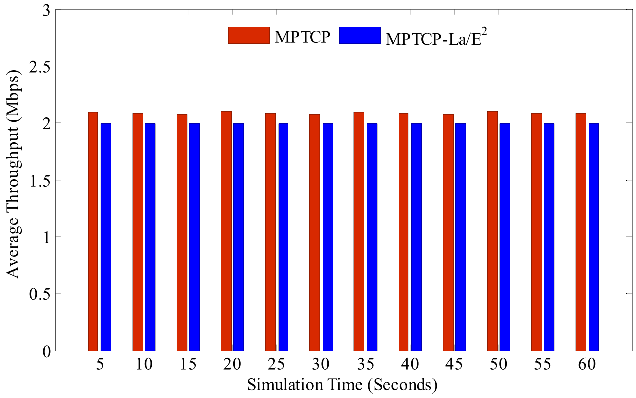


FIGURE 17. Comparison of average throughput.

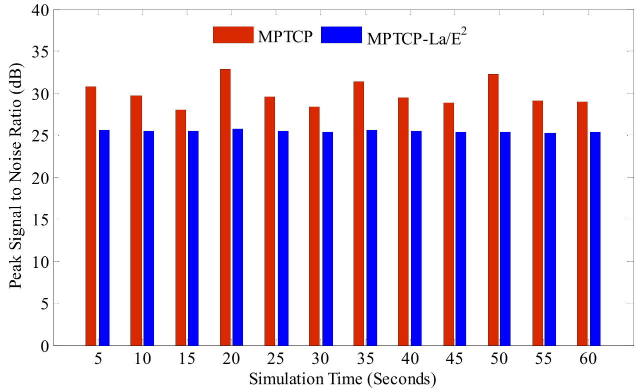


FIGURE 18. Comparison of peak signal-to-noise ratio.

(denoted UE-B1, UE-B2, $\cdots$, UE-B10, respectively), in which ($\mathrm{UE}-\mathrm{B}1, \cdots, \mathrm{UE}-\mathrm{B}4$) are located in the coverage of LTE network, ($\mathrm{UE}-\mathrm{B}7, \cdots, \mathrm{UE}-\mathrm{B}10$) are located in the coverage of Wi-Fi network, and UE-B5 is located in the overlapping coverage of Wi-Fi and LTE networks. All the single-interface mobile terminals and UE-A are intercommunicated via traditional TCP connections. Moreover, path 1 is attached with ten CBR traffic generators in order to simulate the LDDoS traffic. All the generators start their attack at 0.1^{th} second and inject the LDDoS traffic with the characteristics of ($T = 400\text{ms}, L = 200\text{ms}$). In order to convince the E^2DS component is good and effective, the attack pulse R of these LDDoS attacks only varies randomly between 0-0.2Mbps.

B. SIMULATION RESULTS

1) ENERGY EFFICIENCY COMPARISON

Fig. 16 shows the comparison of energy consumption rate (ecr) when using the baseline MPTCP and MPTCP-La/E^2, respectively. It is obvious from the figure that MPTCP achieves a higher ecr value than MPTCP-La/E^2. This is because the baseline MPTCP exploits the access diversity of multiple network interfaces for multipath data transmission but fails to consider the energy consumption and use cost of each interface. In contrast, MPTCP-La/E^2 exploits an energy-aware data scheduling mechanism to allocate application traffic to the most energy-efficient path. Moreover, compared with MPTCP-La/E^2, MPTCP requires additional energy for operating multiple network interfaces, regardless of whether these interfaces are active or potentially broken. Correspondingly, MPTCP requires higher energy consumption than MPTCP-La/E^2. With a total simulation time of 60 seconds, MPTCP-La/E^2's energy consumption rate is 19.46% lower than that of MPTCP.

2) THROUGHPUT PERFORMANCE COMPARISON

Fig. 17 presents the throughput performance comparison when the baseline MPTCP and MPTCP-La/E^2 are utilized, respectively. It can be observed from this figure that MPTCP performs better than MPTCP-La/E^2 in terms of average throughput performance. The reason is that MPTCP-La/E^2 needs to calculate both the energy consumption and the current available *cwnd* size of each path, and then offload a certain amount of data required to be sent from a high energy-cost path to an energy-efficient one. In contrast, MPTCP does not trade throughput performance for energy-savings, it just simply makes full use of the multipath resources for bandwidth aggregation and data delivery. Although MPTCP-La/E^2's throughput is lower than that of MPTCP, it should be noted that there is more potential for energy-savings in MPTCP-La/E^2 than the MPTCP scheme. After 60 seconds of simulation time, MPTCP-La/E^2's average throughput is only 4.3% lower than that of MPTCP.

3) USERS' QUALITY OF EXPERIENCE FOR VIDEO STREAMING

We utilize the Peak Signal-to-Noise Ratio (PSNR) to evaluate the video transmission performance. To this end, we compute the PSNR value by using the following Eq. (8) [39],

$$PSNR = 20\log_{10}\left(\frac{Bitrate_{\max}}{\sqrt{\left(Thr_{\exp} - Thr_{\text{crt}}\right)^2}}\right), \tag{8}$$

which $Bitrate_{\max}$, $Thr_{\exp}$, and Thr_{crt} are the average bitrate of the video required to be sent, the expected average throughput and the actual average throughput during video transmission, respectively. The values of both $Bitrate_{\max}$ and $Thr_{\exp}$ are 2Mbps in our tests.

Fig. 18 shows the PSNR (dB) comparison when MPTCP and MPTCP-La/E^2 are used, respectively. As shown in the figure, MPTCP attains a higher PSNR than that of the MPTCP-La/E^2 solution. However, as we discussed earlier, MPTCP does not take the energy consumption problem into account during in multipath transmission. We can also observe that the energy-aware scheduling mechanism of the MPTCP-La/E^2 affects directly the PSNR performance which is lower than that of MPTCP. This is actually a necessary 'cost' for applying an energy optimization operation to the MPTCP. We think that this is 'a cost worth paying' for energy consumption reduction and can be acceptable to a mobile cloud user since according to the mentioned relationship between PSNR and Mean Opinion Sore (MOS) in [42], MPTCP-La/E^2 achieves the same MOS as that of MPTCP does (the average PSNRs of MPTCP and MPTCP-La/E^2 are 29.95 dB and 25.24 dB, respectively, which belong to the same MOS level of '3-Fair').

VI. CONCLUSION

Motivated by the facts that a poor-performing path caused by LDDoS attacks can present a significant impact on MPTCP's performance and quality of service, this paper proposes a novel LDDoS attack-aware energy-efficient MPTCP solution dubbed as MPTCP-La/E2 for multi-homed MCC systems. MPTCP-La/E^2 mainly consists of two components, which are *LDDoS-aware multipath manager* (LaM^2) that is devoted to detecting the transmission quality of each MPTCP path, switching a path to a proper state, and choosing a subset of suitable paths for multipathing, and *energy-efficient data scheduler* (E^{2}DS) that is devoted to achieving bandwidth aggregation and energy-savings by jointly considering the transmission state and energy cost of each path. The simulation results demonstrate that MPTCP with LaM^2, the potentially broken can be timely detected and the application-level performance is enhanced. When applying E^{2}DS to MPTCP, the energy usage is optimized while the users' quality of experience is maintained.

REFERENCES

[1] H. Hu, Y. Wen, and D. Niyato, "Public cloud storage-assisted mobile social video sharing: A supermodular game approach," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 545–556, Mar. 2017.

[2] Y. Liu, M. J. Lee, and Y. Zheng, "Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2398–2410, Oct. 2016.

[3] C. Xu, P. Wang, C. Xiong, X. Wei, and G.-M. Muntean, "Pipeline network coding-based multipath data transfer in heterogeneous wireless networks," *IEEE Trans. Broadcast.*, vol. 63, no. 2, pp. 376–390, Jun. 2017.

[4] Y. Cao, C. Xu, J. Guan, and H. Zhang, "CMT-CC: Cross-layer cognitive CMT for efficient multimedia distribution over multi-homed wireless networks," *Wireless Pers. Commun.*, vol. 82, no. 3, pp. 1643–1663, 2015.

[5] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, *TCP Extensions for Multipath Operation With Multiple Addresses*, document IETF RFC 6824, Jan. 2013.

[6] Q. Peng, A. Walid, J. Hwang, and S. H. Low, "Multipath TCP: Analysis, design, and implementation," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 596–609, Feb. 2016.

[7] C. Xu, J. Zhao, and G.-M. Muntean, "Congestion control design for multipath transport protocols: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2948–2969, 4th Quart., 2016.

[8] H. Sinky, B. Hamdaoui, and M. Guizani, "Proactive multipath TCP for seamless handoff in heterogeneous wireless access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4754–4764, Jul. 2016.

[9] Q. de Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "Observing real smartphone applications over multipath TCP," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 88–93, Mar. 2016.

[10] B.-H. Oh and J. Lee, "Feedback-based path failure detection and buffer blocking protection for MPTCP," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3450–3461, Dec. 2016.

[11] Y. Cao *et al.*, "(PU)^{2}M^2: A potentially underperforming-aware path usage management mechanism for secure MPTCP-based multipathing services," *Concurrency Comput., Pract. Exp.*, to be published.

[12] P. Dong, J. Wang, J. Huang, H. Wang, and G. Min, "Performance enhancement of multipath TCP for wireless communications with multiple radio interfaces," *IEEE Trans. Commun.*, vol. 64, no. 8, pp. 3456–3466, Aug. 2016.

[13] Y. Zhang, H. Mekky, Z. Zhang, F. Hao, S. Mukherjee, and T. V. Lakshman, "SAMPO: Online subflow association for multipath TCP with partial flow records," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[14] K. Xue, J. Han, H. Zhang, K. Chen, and P. Hong, "Migrating unfairness among subflows in MPTCP with network coding for wired–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 798–809, Jan. 2017.

[15] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2345–2361, Sep. 2016.

[16] A. Ali, J. Qadir, A. Sathiaseelan, and K.-L. A. Yau, "MP-ALM: Exploring reliable multipath multicast streaming with multipath TCP," in *Proc. IEEE LCN*, Nov. 2016, pp. 138–146.

[17] B. Arzani, A. Gurney, S. Cheng, R. Guerin, and B. Loo, "Deconstructing MPTCP performance," in *Proc. IEEE ICNP*, Oct. 2014, pp. 269–274.

[18] Y. Cui, L. Wang, X. Wang, and Y. Wang, "FMTCP: A fountain code-based multipath transmission control protocol," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 465–478, Feb. 2015.

[19] Y. Lim, Y.-C. Chen, E. M. Nahum, D. Towsley, and K.-W. Lee, "Cross-layer path management in multi-path transport protocol for mobile devices," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1815–1823.

[20] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, and K. Long, "On a mathematical model for low-rate shrew DDoS," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 7, pp. 1069–1083, Jul. 2014.

[21] Z. Wu, L. Zhang, and M. Yue, "Low-rate DoS attacks detection based on network multifractal," *IEEE Trans. Depend. Sec. Comput.*, vol. 13, no. 5, pp. 559–567, Sep./Oct. 2016.

[22] C. Pearce and S. Zeadally, "Ancillary impacts of multipath TCP on current and future network security," *IEEE Internet Comput.*, vol. 19, no. 5, pp. 58–65, Sep./Oct. 2015.

[23] J. Wu, B. Cheng, M. Wang, and J. Chen, "Energy-efficient bandwidth aggregation for delay-constrained video over heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 1, pp. 30–49, Jan. 2017.

[24] Q. Peng, M. Chen, A. Walid, and S. Low, "Energy efficient multipath TCP for mobile devices," in *Proc. ACM MobiHoc*, 2014, pp. 257–266.

[25] Y. Lim *et al.*, "How green is multipath TCP for mobile devices?" in *Proc. ACM SIGCOMM Workshop Things Cellular*, 2014, pp. 3–8.

[26] F. Kaup, M. Wichtlhuber, S. Rado, and D. Hausheer, "Can multipath TCP save energy? A measuring and modeling study of MPTCP energy consumption," in *Proc. IEEE LCN*, Oct. 2015, pp. 442–445.

[27] T. A. Le, C. S. Hong, M. A. Razzaque, S. Lee, and H. Jung, "ecMTCP: An energy-aware congestion control algorithm for multipath TCP," *IEEE Commun. Lett.*, vol. 16, no. 2, pp. 275–277, Feb. 2012.

[28] S. Chen, Z. Yuan, and G.-M. Muntean, "An energy-aware multipath-TCP-based content delivery scheme in heterogeneous wireless networks," in *Proc. IEEE WCNC*, Apr. 2013, pp. 1291–1296.

[29] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, M. Rajarajan, and R. Buyya, "Combating DDoS attacks in the cloud: Requirements, trends, and future directions," *IEEE Cloud Comput.*, vol. 4, no. 1, pp. 22–32, Jan./Feb. 2017.

[30] *MPTCP-NS3 Project*. Accessed: Jul. 2017. [Online]. Available: http://code.google.com/p/mptcp-ns3

[31] Y. Xiang, K. Li, and W. Zhou, "Low-rate DDoS attacks detection and traceback by using new information metrics," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 426–437, Jun. 2011.

[32] S. Shin, D. Han, H. Cho, J.-M. Chung, I. Hwang, and D. Ok, "TCP and MPTCP retransmission timeout control for networks supporting WLANs," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 994–997, May 2016.

[33] V. Paxson and M. Allman, *Computing TCP's Retransmission Timer*, document IETF RFC 2988, 2000.

[34] C. Xu, Z. Li, J. Li, H. Zhang, and G.-M. Muntean, "Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 7, pp. 1175–1189, Jul. 2015.

[35] A. Nikravesh, Y. Guo, F. Qian, Z. Mao, and S. Sen, "An in-depth understanding of multipath TCP on mobile devices: Measurement and system design," in *Proc. ACM MobiCom*, 2016, pp. 189–201.

[36] Y. Cao, Q. Liu, Y. Zuo, G. Luo, H. Wang, and M. Huang, "Receiver-assisted cellular/WiFi handover management for efficient multipath multimedia delivery in heterogeneous wireless networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, p. 229, Dec. 2016.

[37] F. Song, R. Li, and H. Zhou, "Feasibility and issues for establishing network-based carpooling scheme," *Pervas. Mobile Comput.*, vol. 24, pp. 4–15, Dec. 2015.

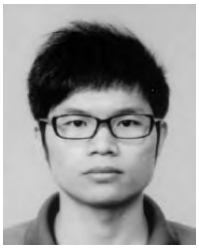
[38] F. Song, Y. Zhang, Z. An, H. Zhou, and I. You, "The correlation study for parameters in four tuples," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 19, nos. 1–2, pp. 38–49, 2015.

[39] S. Banerjee and A. Sen, "Impact of region-based faults on the connectivity of wireless networks in log-normal shadow fading model," in *Proc. IEEE ICC*, Jul. 2011, pp. 1–6.

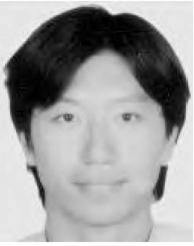
[40] Y. Cao, S. Chen, Q. Liu, Y. Zuo, H. Wang, and M. Huang, "QoE-driven energy-aware multipath content delivery approach for MPT CP-based mobile phones," *China Commun.*, vol. 14, no. 2, pp. 90–103, 2017.

[41] S.-B. Lee, G.-M. Muntean, and A. F. Smeaton, "Performance-aware replication of distributed pre-recorded IPTV content," *IEEE Trans. Broadcast.*, vol. 55, no. 2, pp. 516–526, Jun. 2009.

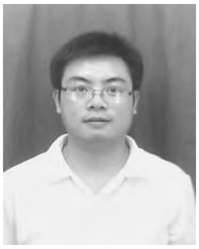
[42] *Methods for Subjective Determination of Transmission Quality*, document ITU-T Recommendation P.800, Aug. 1996.



YUANLONG CAO received the B.S. degree in computer science and technology from Nanchang University, China, in 2006, the M.S. degree in software engineering from the Beijing University of Posts and Telecommunications (BUPT), in 2008, and the Ph.D. degree from the Institute of Network Technology, BUPT, in 2014. He was an Intern/Engineer with BEA TTC, IBM CDL, and DT Research, Beijing, from 2007 to 2011. He is currently a Lecturer with the School of Software, Jiangxi Normal University, China. His research interests include multimedia communications and next-generation Internet technology.



FEI SONG is currently with the National Engineering Laboratory for Next Generation Internet Technology, School of Electronic and Information Engineering, Beijing Jiaotong University. His current research interests include network architecture, network security, protocols optimization, wireless communications, and cloud computing. He also serves as the Technical Reviewer for several journals, including the IEEE Transactions on Services Computing, the IEEE Transactions on Parallel Distribution System, and the IEEE Transactions on Emerging Topics in Computing.



QINGHUA LIU received the B.S. degree in software engineering and the M.S. degree in management science and engineering from Jiangxi Normal University (JXNU), China, in 2007 and 2011, respectively. He is currently an Experimentalist with the School of Software, JXNU. His research interests include multimedia networking and next-generation Internet technology.

MINGHE HUANG was the chairman of the School of Software, Jiangxi Normal University (JXNU). He was elected as a University Young and Middle-aged Academic Leaders in Jiangxi Province and Distinguished Teacher of Jiangxi Province. He is currently a Professor with the School of Software, JXNU. He also serves as the Jiangxi Provincial Government Counselor and the Standing Director of the Association of Fundamental Computing Education in Chinese Universities. He has authored over 50 research papers in computer networks, communications, and information theory.

HAO WANG was elected as a University Young and Middle-aged Academic Leaders in Jiangxi Province and Distinguished Teacher of Jiangxi Province. He is currently a Professor and the Chairman with the School of Software, Jiangxi Normal University. His major research interests include computer network congestion control and computer network management.

ILSUN YOU (SM'17) is currently with the Department of Information Security Engineering, Soonchunhyang University. He has served or is currently serving as a main organizer of international conferences and workshops, such as MobiWorld, MIST, SeCIHD, AsiaARES, and so forth. His main research interests include internet security, authentication, access control, and formal security analysis. He is a Fellow of the IET. He is the EiC of the *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. He is a member of the Editorial Board for *Information Sciences*, *Journal of Network and Computer Applications*, *International Journal of Ad Hoc and Ubiquitous Computing*, *Computing and Informatics*, *Journal of High Speed Networks*, *Intelligent Automation & Soft Computing*, and *Security and Communication Networks*.