Multipath TCP Meets Reinforcement Learning: A Novel Energy-Efficient Scheduling Approach in Heterogeneous Wireless Networks

Pingping Dong, Rongcheng Shen, Qian Wang, Dian Zhang, Yajing Li, Yuning Zuo, Wenjun Yang, and Lianming Zhang

Abstract

Multipath TCP (MPTCP) has been standardized by the IETF as an extension of conventional TCP and it allows the system to utilize multiple paths simultaneously, which can aggregate bandwidth to improve network throughput. However, MPTCP needs to open multiple interfaces at the same time, which makes MPTCP consume more energy to maintain multiple interface connections. Thus, how to manage subflows with the MPTCP's scheduling system to determine which paths should be used for data transmission is of critical importance to reduce energy consumption and ensure network throughput. Due to the path heterogeneity and random packet losses in wireless networks, existing scheduling systems, and selecting paths based on the path's delay or energy cost, may suffer from performance degradation. In this article, we propose a reinforcement learning-based multipath scheduler called MPTCP-RL to determine the optimal path set for different flows. MPTCP-RL adopts deep reinforcement learning as well as MPTCP transmission model to manage path usage among multiple connections to make sure that the sender can adaptively select the optimal path set for a certain application according to the current network environment. MPTCP-RL is an asynchronous reinforcement learning framework, which separates the processes of offline training and online decision to ensure that the learning process will not introduce extra delay and overhead on the decision making process in MPTCP path management. The extensive experimental results show that MPTCP-RL can improve the aggregate throughput and reduce energy consumption significantly compared to the state-of-the-art mechanisms in a variety of network scenarios.

Introduction

Benefiting from the continuous advancement of network technology, lots of terminal devices as well as servers can be accessed to the network by multiple interfaces, such as 5G, WiFi, and Bluetooth. There are multiple paths for end-to-end transmission between these devices. However, regular TCP mainly utilizes a single interface, which cannot effectively utilize multiple ports, resulting in the waste of resources [1]. MPTCP extends the standard TCP and allows data streams to be delivered across multiple simultaneous connections and consequently paths [2]. Nowadays, the MPTCP has been successfully deployed in data center networks as well as in practical applications. Apple released the first iOS 7 commercial operating system that officially deployed the MPTCP protocol, which enabled Siri's intelligent voice service to simultaneously use WiFi networks and cellular mobile networks to transmit data, improving the fluency of the system. In addition, South Korean Samsung took the lead in deploying the MPTCP protocol in the Android system smartphone Galaxy S6. The smartphone can use LTE and WiFi networks in parallel with a network connection speed of up to 1.17 Gb/s [3].

Although MPTCP can improve bandwidth utilization, increase network throughput [4] and improve reliability, MPTCP utilizes multiple network interfaces at the same time for data transmission, which makes MPTCP consume more energy. We hope that the nodes, such as the client or the server, can transmit more data at the same energy consumption, thereby saving network energy while saving energy consumption.

The researchers have proposed several algorithms to tackle the problem of MPTCP energy optimization. Palash et al. [5] defined three steps to work in cycles to continuously help mobile devices in balancing bandwidth requirements and energy efficiency. eMTCP [6] set up a subflow interface state detector and scheduled packets to the low energy interface whenever it was idle. Lim et al. [7] proposed an energy-efficient MPTCP that used all interfaces or WiFi-only for packet transfer based on the estimated available throughput of each interface. Literatures [8, 9] only selected the most energy-efficient path to use and muted the other paths. Pluntke et al. [8] proposed a multi-path scheduler by formalizing the multi-path schedulers as Markov decision processes. MPTCP-QE [10] provided an application rate-aware energy-saving subflow management strategy to trade-off the throughput performance and energy consumption for mobile phones. Above all, most of the existing algorithms mainly use certain criterions, such as energy cost or throughput, for path selection and shift traffic from the low-quality path to the high-quality path. However, the network is dynamic and the factors

Digital Object Identifier:
10.1109/MWC.013.2100658

Pingping Dong, Rongcheng Shen, Qian Wang, Dian Zhang, Yajing Li, Yuning Zuo and Lianming Zhang are with Hunan Normal University, China; Wenjun Yang is with the University of Victoria, Canada. Corresponding author is Lianming Zhang.

1536-1284/23/$25.00 © 2023 IEEE

that influence the performance of MPTCP are complicated and diverse, especially when the paths are asymmetric and random packet loss exists. The problem of designing an efficient path management algorithm that is simultaneously energy-aware and throughput-guaranteed is challenging and existing solutions may not achieve this aim.

The prevailing machine learning (especially deep reinforcement learning [11]) techniques motivate us to seek a learning-based scheduling algorithm to tackle the above issues. Li *et al.* proposed SmartCC [12], a Q-learning-based framework for MPTCP congestion control. Zhang *et al*. [13] proposed a neural adaptive multipath scheduler called ReLeS based on the normalized advantage function (NAF) model. These mechanisms can improve the performance of MPTCP. However, they are not optimized for overall energy.

This article takes the first step toward utilizing deep reinforcement learning for MPTCP's energy efficiency and proposed MPTCP-RL. MPTCP-RL regards the path selection as a learning task and adopts the proximal policy optimization (PPO) model to generate the set of available paths and the usage probability of each path belonging to an MPTCP connection to maximize the reward function. Then, considering that if an application can be transmitted faster with several higher quality paths, only using these set of paths can achieve higher throughput and lower flow completion time compared to the situation that all paths are taken into consideration, MPTCP-RL utilizes our previously proposed MPTCP transmission model [3] to select different path sets for different applications based on the usage probability output from the reinforcement learning model. In addition, MPTCP-RL is a scheduling policy deployed on the server side and aims to save the energy of the battery-limited mobile devices by controlling the number of utilized paths as well as the transferred packets over each path. The performance and efficiency of this proposed MPTCP-RL are analyzed and verified by numerical experiments with NS3.

The remainder of this article is organized as follows: The next section gives the background for our work. We then introduce the challenges of our investigated problem. Following that, we describe the details of MPTCP-RL. Extensive experiments are conducted. Finally, we conclude the article.

Background

The newly improved MPTCP can fully utilize all available interfaces to meet the requirements of higher throughput, better robustness, and greater flexibility. How to achieve green transmission with MPTCP is quite worthy of investigation. In this section, we review the background of energy consumption and MPTCP.

Energy Consumption Model

Due to the limited battery capacity of mobile devices, we mainly focus on the energy consumption of mobile devices when using MPTCP for wireless data transmissions, as it determines the lifetime and reliability of an MPTCP connection.

The wireless interface of mobile devices works in three states: IDLE state, CONNECTED state, and tail state. Generally, the device stays in the IDLE state to keep lower power consumption when there is no data transmission, and promotes to CONNECTED state for sending/receiving data. The interface enters the tail state if it finishes the transmission. Then, it waits for a time interval of tail length. If there is no data transmission during the tail length, the interface switches to the IDLE state again. Among all the three states, the CONNECTED state consumes the highest power, followed by the tail state and the IDLE state consumes the lowest power.

In this article, we adopt the power consumption model proposed in [14] which has been widely adopted in the current research to obtain the energy consumption for mobile devices and is suitable for most wireless interfaces, such as 4G, 3G, and WiFi. Assuming the uplink and the downlink throughput are t_u (Mb/s) and t_d (Mb/s), respectively, the instant power (mW) consumption over path r denoted by P_r can be calculated as the sum of $\alpha_u t_u$, $\alpha_d t_d$, β. α_u and α_d represent the power consumed per bit for sending and receiving data, respectively. β denotes the sunk power cost for keeping the interface active. The energy consumption parameters are set to be 283 mW/Mb/s, 137 mW/Mb/s and 132.9 mW, 438 mW/Mb/s, 52 mW/Mb/s and 1288 mW, 869 mW/Mb/s, 122.1 mW/Mb/s and 817.9 mW for WiFi, and 4G and 3G, respectively.

Thus, if a transmission lasts for a duration T, the energy consumption on path r for this transmission can be get by multiplying P_r by T. Finally, the total energy consumption of all multiple interfaces can be obtained by adding up the energy consumption of all paths.

MPTCP

MPTCP is one of the most popular multipath transmission protocols [10] and it allows a single data stream to be transmitted across multiple paths simultaneously. Each path is called a subflow. The MPTCP path scheduler is responsible for splitting data packets over available paths. More specifically, if several paths have available congestion window (cwnd), the MPTCP scheduler should first select on which to send data, and then determines how much data should be sent over the selected path. There are many scheduling algorithms and the default schedulers in NS3 and Linux are Round-Robin and minRTT, respectively. Round-Robin simply selects paths in turn to utilize all paths. minRTT always selects the path with the lowest RTT.

Different scheduling algorithms may have different performances for MPTCP. In the next section, we analyze the challenges when designing new schedulers, especially when considering energy efficiency.

Problems and Challenges

As multiple factors can influence MPTCP performance, it is quite challenging to design an efficient MPTCP scheduling algorithm in today's dynamic network environment, particularly in heterogeneous networks where the path characteristics are asymmetry and random packet loss exists.

Heterogeneous Characteristics of Links

The process of MPTCP scheduling algorithms selects one path after the other among all available paths based on certain criterions that can reflect the path quality. However, when the network is heterogeneous and the characteristics between

The newly improved MPTCP can fully utilize all available interfaces to meet the requirements of higher throughput, better robustness, and greater flexibility. How to achieve green transmission with MPTCP is quite worthy of investigation. In this section, we review the background of energy consumption and MPTCP.

The process of MPTCP scheduling algorithms selects one path after the other among all available paths based on certain criterions that can reflect the path quality. However, when the network is heterogeneous and the characteristics between different paths are asymmetric, utilizing all available paths simultaneously may inversely impair the transmission performance for short flows.

different paths are asymmetric, utilizing all available paths simultaneously may inversely impair the transmission performance for short flows.

Considering a simple scenario where an application with 16 packets is transferred across two disjoint paths. The RTT of subflow1 and subflow2 is set to 20 ms and 80 ms, respectively. The workflow of MPTCP is analyzed as follows.

In MPTCP, each subflow is a regular TCP connection and is initialized through a three-way handshake at the start time simultaneously. That is why subflow1 starts to send data at 20 ms and subflow2 is established and starts to transfer the first packet at 80 ms. As the RTT of subflow2 is four times that of subflow1, subflow1 is at the beginning of the fourth round with the congestion window size of eight when subflow2 begins to transmit the first round with the congestion window size of one. In the slow start phase, the congestion window doubles every round with an initial value of one. During the transmission of the first three rounds, seven packets have been delivered. The other nine packets are spread to subflow1 with eight packets and subflow2 with one packet simultaneously. Finally, the whole transmission ends at 160 ms. However, if only the fast subflow, that is, subflow1, is used, subflow1 can finish the transmission within five rounds and can finish the transmission at 100 ms.

Thus, utilizing all available paths to transmission in such heterogeneous networks may hurt MPTCP performance. The proposed MPTCP-RL resolves this problem by selecting different path sets for different applications based on the estimated data amount according to our MPTCP transmission model.

Random Packet Loss in Wireless Networks

The random packet loss in wireless networks makes it challenging to define the path quality when selecting a path and ignoring it can certainly harm MPTCP performance. Take the state-of-the-art scheduling algorithm — BLEST [4], as an example.

BLEST aims to make segments arrive in order to avoid head-of-line (HOL) blocking caused by out-of-order (OFO) packets. If an MPTCP connection has several subflows, the RTT of one subflow is far larger than the other flows and a packet with a small sequence number is transmitted over this subflow. This will cause a large number of packets with big sequence numbers transferred over other fast subflows to arrive at the client earlier. The client has to put these out-of-order packets in the buffer. When the number of these packets is large enough, the buffer will become full and the receive window will become zero. In this case, the MPTCP cannot send packets anymore even if the network is capable. This phenomenon is called HOL-Blocking.

To alleviate this issue, BLEST pre-allocates data packets out of order. Take the scenario that the value of the RTT of $subflow_i$ and $subflow_j$ are 10 ms and 20 ms, respectively, as an example. The current congestion window size of the two subflows are both five packets and they are in the congestion avoidance phase where the congestion window increase by one every round. As the RTT of $subflow_j$ is twice that of $subflow_i$, $subflow_i$ can complete two rounds of transmission with 11 packets (five packets in the first round and six packets in the second round) during RTT_j. Then the packets with the sequence number from one to 11 will be allocated to $subflow_i$, and packets from 12 to 16 will be transmitted on $subflow_j$. If no packet loss occurs, all the packets can arrive at the client simultaneously after 20 ms.

However, suppose a packet (e.g., packet 1) is lost in the lossy heterogeneous network. It should be retransmitted in the next round. Then not all packets numbered from one to 12 can reach the client earlier than packets numbered from 12 to 16. BLEST cannot avoid out-of-order packets in this situation.

Based on the above analysis, we can conclude that multiple factors should be taken into consideration when designing a path management algorithm. This prompts us to propose a new kind of MPTCP path management scheme, which should consider various network statuses, such as packet loss rate, throughput, and delay. It should adapt to network dynamics and it must reduce energy consumption while ensuring network throughput. The proposed MPTCP-RL is based on reinforcement learning, which uses multi-dimensional network features as input and generates different optimal path sets for different applications.

Model and Solution

In this section, we introduce the proposed MPTCP-RL which first calculates the usage probability of each path based on deep reinforcement learning to maximize the value of the reward, and then selects different paths sets for different applications to enhance the performance of MPTCP.

The Structure of the Agent in MPTCP-RL

With the development of deep learning, many new algorithms are evolved in recent years, such as deep Q network (DQN) and PPO. As an online strategy gradient method, PPO has the advantage of unbiased gradient estimation. The parameters involved in PPO include l_1, l_2, τ, γ, and ε, where l_1 and l_2 are learning late, τ and γ are PPO chip number and discount coefficient, respectively, ε is for the ε-greedy algorithm. Through the importance sampling, offline learning is possible. This means PPO can be directly applied to the asynchronous learning framework. This framework can predict the future status of each subflow, give the optimal scheduling strategy according to historical data and better adapt to the network dynamics.

In addition, our reinforcement learning model aims to generate a scheduling rule table for MPTCP path selection. This process is a continuous action space where every path can be selected. PPO has the characteristics of high efficiency, easy deployment, and high reliability, which can handle continuous action spaces. Thus, the proposed MPTCP-RL adopts PPO in this article. MPTCP-RL captures the current state of each path and derives the optimal subflow set for each application to be transmitted. To achieve this aim, we define three key elements for the deep reinforcement learning model in MPTCP-RL.

State: It refers to the current state information of all subflows, such as the throughput, delay, energy consumption, and packet loss rate. The state is represented as s in the article. What is more, the information on the network state is not only a precondition for MPTCP subflow scheduling, but also

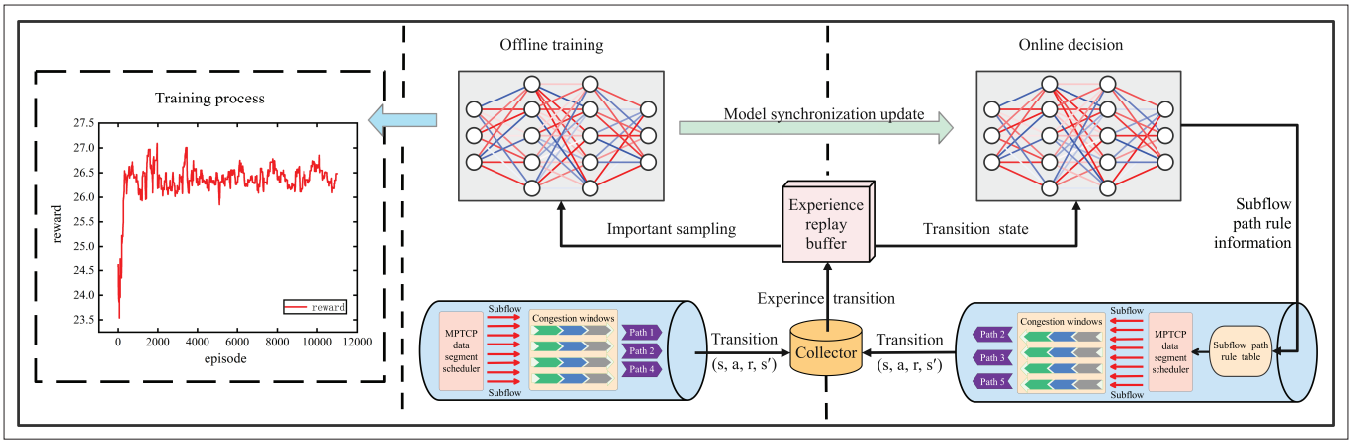

FIGURE 1. The MPTCP-RL learning framework.

a learning experience for further optimization of our network model.

Action: The output layer data of the neural network model is used as the output action, which is represented as a tuple of available paths and the probability that they may be selected. This probability is further used to select different subflow sets for different applications. In the article, *a* represents action. In order to make the proposed MPTCP-RL can deal with a certain range of changes in the number of available paths without retraining, we handle actions in a similar way to the action mask. That's, firstly, the action mask vector corresponding to each path is constructed based on the path information collected by the server. The vector is composed of zero and one, where zero denotes that the path is invalid and should be blocked, and one represents that the path is available. Then, by doing the dot product of the output action vector of the MPTCP-RL model and this action mask vector, we can block invalid paths or activate valid paths.

Reward: It refers to the obtained reward when taking a possible action in each step. During the training of the PPO model, the value of the reward is calculated by the information of all subflows as needed. In the decision-making stage of the model, QoS targets are taken into consideration, which is maximizing the total throughput B for all paths, decreasing the total energy consumption E for all paths, reducing the end-to-end delay D on all paths, and alleviating the jitter J for all paths to keep the transmission more stable. And all pieces of information are normalized.

To this end, we define the reward function $r = w_1 \cdot B - w_2 \cdot E - w_3 \cdot D + w_4 \cdot (J + 1)^{-1}$, where $w_i \in [0, 1]$ is the weight coefficients that represent the importance of the QoS targets. Based on relevant studies [12] and considering the impact of each QoS parameter, w_i are set to one.

The MPTCP-RL Learning Framework

The framework of the proposed MPTCP-RL is shown in Fig. 1. It is designed as an asynchronous learning framework and is divided into offline training and online decision-making.

Offline Training: The PPO model is first trained offline on the server side by collecting the historical information of the traditional MPTCP scheduling algorithm. Then, in the operation of the interaction between the online decision scheduler and the environment, the local collector starts collecting the network information (*s*, *a*, *r*, *s*′) and storing them in the experience pool in the server. This collected information reflects whether the subflow scheduling strategy is good or not. Therefore, the server can judge whether or not to retrain the PPO model based on the gathered information to ensure that the model can adapt to the network dynamics. The whole training process is shown at the leftmost end of Fig. 1, in which we can see that the proposed MPTCP-RL exhibits good convergence.

Online Decision-Making: With the current network state information as the input, MPTCP-RL can easily get the optimal rule table by real-time computing. Based on this table, subflow scheduling strategy can be carried out by the scheduler. The decision-making process is as follows. The scheduler under the state s executes the optimal action *a*, which generates a rule table. According to the table, path selection is optimized and rules are delivered. After that, the reward r and the next network state s′ are produced.

Using this framework, offline training and online decision-making can be run asynchronously, and after offline training, the parameters of the neural network model for online decision-making are synchronously updated to adapt to the environmental changes, so that the scheduling policy is optimal. It is to note that the PPO model is placed on the service side, there are no side effects on the client's energy consumption.

Paths Selection Based on the MPTCP Transmission Model

Based on the output of PPO, which describes all available subflows and the probability that they should be selected, we describe how to determine different subflow sets for different applications in this subsection.

We first sort all paths according to the probability in the descending order, then try to select the first k subflows. To determine k for a certain application, our earlier proposed MPTCP transmission model [3] is taken into consideration. The model considers the MPTCP's four congestion control algorithms over each subflow, for example, slow start, the coupled congestion avoidance, fast retransmit, as well as fast recovery to estimate the data amount that can be transferred during n rounds. Its accuracy has also been validated in [3].

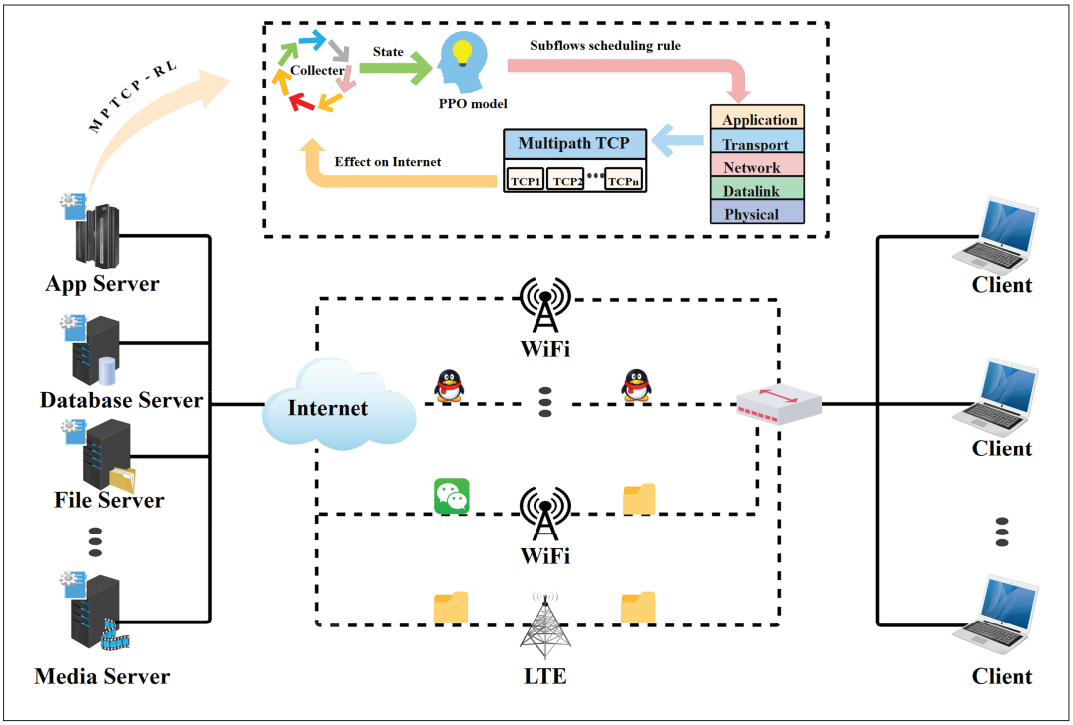

FIGURE 2. The topology of the experiments.

The detailed description of the process for determining k is as follows.

We sort the k path again based on the round-trip time of each path i, RTT_i, in ascending order. Then the number of rounds η_i that can be transferred on the faster path i compared with the slowest path k among the k subflows is calculated as the ratio of RTT_k and RTT_i. Further, the amount of data that path i can transmit during the η_i rounds can be estimated according to the MPTCP transmission model. Then, we can obtain the total data amount C_k of all the k paths during RTT_k by summing these values. Finally, based on the idea that if an application can be successfully transmitted by the first k subflows with higher quality, the other paths can be ignored. To this end, we compare C_k with the flow size. If C_k is smaller than the flow size, more subflow whose utilization probability is highest among the rest of subflows will be taken into consideration. Otherwise, these k subflows are selected.

Evaluation

Experimental Settings

We conduct simulation experiments based on NS 3.14 and adopt the MPTCP code provided by the Google MPTCP group. We rewrote the scheduling mechanism of the simulator to support MPTCP-RL and compare its performance with the other four existing algorithms: the default Round-Robin, eMPTCP [15], DMPTCP [3], and ReLeS [13]. Among them, eMPTCP selects the path used according to the energy efficiency of the WLAN and LTE paths. ReleS schedules data packets based on the split ratio of each subflow generated from the LSTM + NAF model. Similar to MPTCP-RL, DMPTCP also selects subflows based on data estimation according to the MPTCP transmission model.

The experiments are simulated in a heterogeneous network with the coexistence of WiFi and LTE links, which is illustrated in Fig. 2 where multiple concurrent flows run between the MPTCP client and the MPTCP server with up to four disjoint subflows. Each client randomly downloads some files from the server. Both the web-like short flows and the long-lived FTP flows consist of the traffic pattern in our simulation. The transfer size of the short flows obeys the Pareto distribution with an average value of 57 KB according to the real-world Web traffic model. The size of long-lived TCP flows is 15 MB. All flows burst synchronously at 0.1 s.

The parameters of WiFi and LTE are set following the measurements of [12]. Specifically, the WiFi link capacity is in 5–10 Mb/s with a typical RTT of 40–60 ms. The LTE link has a capacity of 15–20 Mb/s with an RTT value of 100 — 200 ms. In the offline training phases, we run MPTCP-RL on a variety of network parameter settings to learn the optimal rules. In our PPO model, let $l_1 = 0.0003$, $l_2 = 0.001$, $\tau = 0.2$, $\gamma = 0.99$, and $\varepsilon = 0.05$.

The energy consumption is calculated according to the work of [14], as described earlier. Mobile devices mainly download data in our experiments. Thus, the instant power P_r (mW) over path r is calculated as the sum of $\alpha_d t_d$ and β. t_d denotes the throughput of the corresponding interface. The energy consumption on path r is calculated by multiplying P_r by transmission time T over path r.

For the simulation results, we consider the flow completion time and energy consumption as the main evaluation metrics.

Experimental Results

We first conduct experiments in the scenario where there are one WLAN path and one LTE path with varying numbers of concurrent short flows. The results are shown in Fig. 3. According to the figure, we can find that MPTCP-RL obtains the shortest average completion time and the lowest total energy consumption. However, the performance gains are not significant in this network scenario as the

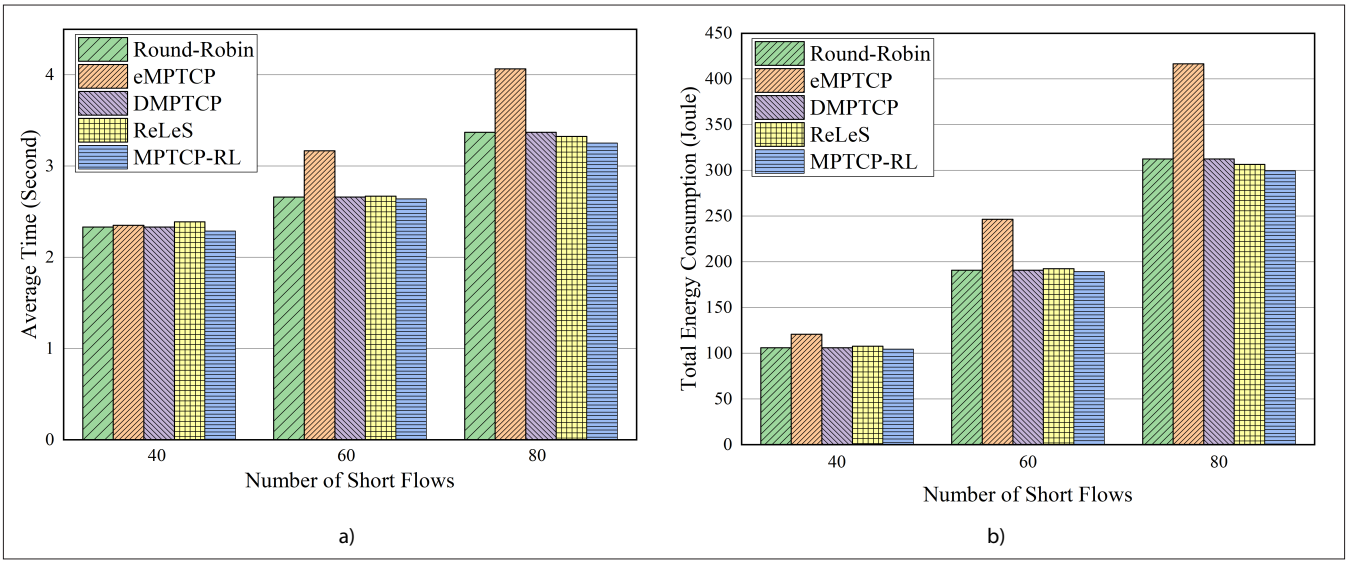

FIGURE 3. The performance of the five algorithms with different numbers of concurrent short flows in the scenario where there are one WLAN path and one LTE path: a) the average time with different numbers of short flows; b) the total energy consumption with different numbers of short flows.

size of the short flow is small and the network environment is relatively simple. All algorithms spend a little time to finish the transmission.

Therefore, we further explore the performance of the five algorithms with concurrent flows that have larger data volume. The results are shown in Fig. 4. As depicted in these figures, with the increase in the number of concurrent flows and the size of the flows, the performance difference between MPTCP-RL and the other four algorithms becomes more and more obvious. In the scenario of 30 concurrent long flows, the average completion time of MPTCP-RL compared to Round-Robin is reduced by 45.1 percent, and the total energy consumption is reduced by 44.7 percent. This is because MPTCP-RL considers energy consumption, delay, packet loss rate and other factors at the same time. It can evaluate each path more comprehensively and select a better transmission path set. Compared with the other four algorithms, MPTCP-RL is more adaptable in complex network environments.

To our surprise, eMPTCP performs worse than the default Round-Robin. To find the reasons, we first conduct experiments when there is only one short flow with the size of 10 kB. The flow completion time is 2.77 s and 3.17 s with eMPTCP and Round-Robin, respectively. Obviously, eMPTCP outperforms Round-Robin in this situation when the transmitted data is small. Furthermore, we calculate the amount of data transmitted over each subflow. The results reveal that eMPTCP almost only uses the WLAN path in most of the transmission time and cannot fully utilize the LTE path, causing its poor performance when there are concurrent flows.

However, in the 30 concurrent long flow scenarios, the completion time of Round-Robin is longer than eMPTCP. This is because both Round-Robin and eMPTCP don't consider the uncertainty in transmission caused by the packet loss rate on the WLAN link. We find that in the scenario of transmitting 30 concurrent long flows, the transmission time of several flows with Round-Robin is several times longer than other flows, resulting in an abnormally long transmission time.

In addition, we also evaluate the performance of the five algorithms with different numbers of subflows, including one LTE path and several WLAN paths. We take the traffic pattern with 20 concurrent media flows whose size is about 1 MB, as an example. The results are shown in Fig. 5. In this figure, the number of WLAN paths ranges from one to three, and the delay of the WLAN path is set to 40 to 60 ms. With the increase of WLAN paths, the average completion time and total energy consumption of the five algorithms are decreasing, and MPTCP-RL also outperforms other algorithms in this scenario.

Finally, we conduct experiments of each algorithm in two dynamic experimental scenarios: path performance changes (Scenario 1) and path number changes (Scenario 2). In these experiments, we use two WLAN paths and one LTE path to transmit a long flow. In Scenario 1, we increase the packet loss rate of path 1 to eight percent at 10 s. In Scenario 2, we directly cut off the connection of path 1 at 20 s.

The results are depicted in Fig. 6. As shown in Fig. 6a and b, the throughput of each algorithm in the first 10 s is mainly concentrated on path 1. Due to random packet loss occurring on the WLAN path, MPTCP-RL shows low throughput at this stage. But after 10 s, as the packet loss rate of path 1 increases to eight percent, the throughput of each algorithm in path 1 drops sharply. MPTCP-RL can shift the traffic to path 2 and path 3 faster, reach the peak value fastest among the five algorithms, and finally complete the data transmission first in 89.8 s. As shown in Fig. 6c and d, the completion time and total energy consumption of MPTCP-RL are lower than the other four algorithms in both scenarios. Moreover, the completion time and the energy consumption of MPTCP-RL can be reduced by up to 61.03 percent and 59.81 percent respectively in scenario 1. This is because that MPTCP-RL considers various factors, such as throughput and energy consumption at the same

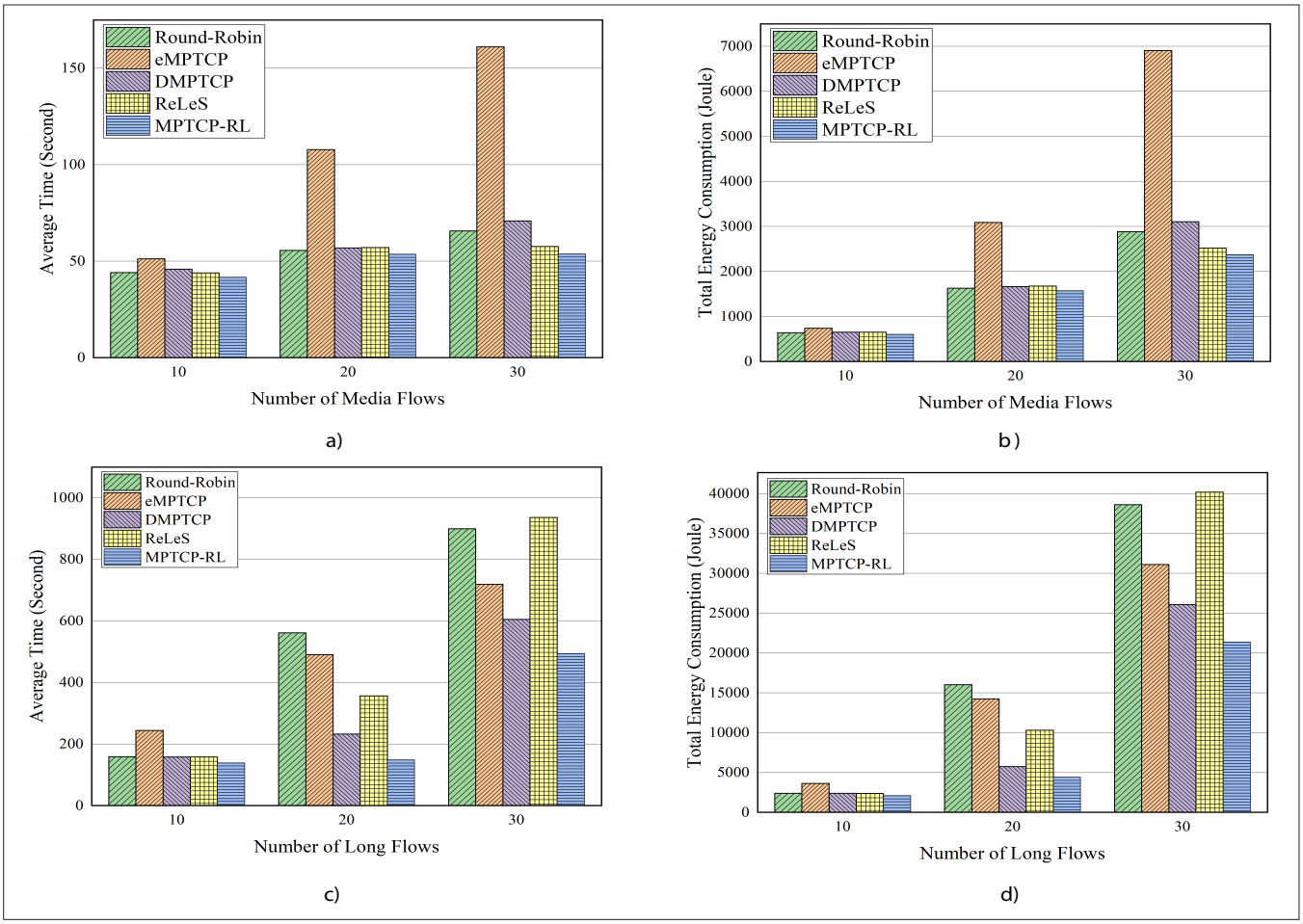

FIGURE 4. The performance of the five algorithms with different numbers of concurrent media and long flows in the scenario where there are one WLAN path and one LTE path, and the size of each media flow and long flow is about 1 MB and 15 MB, respectively: a) the average time with different numbers of media flows; b) the total energy consumption with different numbers of media flows; c) the average time with different numbers of long flows; d) the total energy consumption with different numbers of long flows.

time to comprehensively select path sets. Whether it is a change in path performance or a change in the number of paths, MPTCP-RL is keenly aware and transfers the traffic to the higher-quality path.

Conclusions

To improve the energy efficiency and network throughput of MPTCP, we propose a learning-based MPTCP scheduling algorithm named MPTCP-RL based on reinforcement learning to enable it to adapt to the dynamics of the operational network environment. MPTCP-RL employs a multi-objective utility function as the reward function which takes the overall throughput, the energy consumption, as well as the network delay into consideration. The input of the machine learning model is also multidimensional to ensure that all kinds of critical factors that have an important influence on the MPTCP performance are considered in MPTCP-RL. Based on the usage probability of each subflow output from the reinforcement learning model, MPTCP-RL further adopts our earlier proposed MPTCP transmission model to select different subflow sets for different applications to achieve the aim. Extensive experimental results demonstrate that MPTCP-RL outperforms existing scheduling algorithms in a variety of performance metrics, such as energy efficiency, flow completion time, and throughput.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Nos. 61572191 and 61602171), Hunan Provincial Natural Science Foundation of China (Nos. 2022JJ30398 and 2022JJ40277), Scientific Research Fund of Hunan Provincial Education Department (Nos. 22A0056 and 22B0102).

References

[1] S. R. Pokhrel *et al.*, "Multipath TCP Meets Transfer Learning: A Novel Edge-based Learning for Industrial IoT," *IEEE Internet of Things J.*, vol. 8, no. 13, 2021, pp. 10299–07.
[2] B. Liao *et al.*, "Precise and Adaptable: Leveraging Deep Reinforcement Learning for Gap-Based Multipath Scheduler," *Proc. IFIP Networking Conf.*, 2020, pp. 154–62.
[3] P. Dong *et al.*, "Reducing Transport Latency for Short Flows with Multipath TCP," *J. Network and Computer Applications*, vol. 108, 2018, pp. 20–36.
[4] S. Ferlin *et al.*, "BLEST: Blocking Estimation-based MPTCP Scheduler for Heterogeneous Networks," *Proc. IFIP Networking Conf. and Workshops*, 2016, pp. 431–39.
[5] M. R. Palash *et al.*, "Bandwidth-Need Driven Energy Efficiency Improvement of MPTCP Users in Wireless Networks," *IEEE Trans. Green Commun. Networking*, vol. 3, no. 2, 2019, pp. 343–55.
[6] S. Chen *et al.*, "An Energy-Aware Multipath-TCP-Based Content Delivery Scheme in Heterogeneous Wireless Networks," *Proc. IEEE Wireless Commun. Networking Conf.*, 2013, pp. 1291–96.
[7] Y. Lim *et al.*, *Improving Energy Efficiency of MPTCP for Mobile Devices*, Nov. 2021; available: https://arxiv.org/pdf/1406.4463.

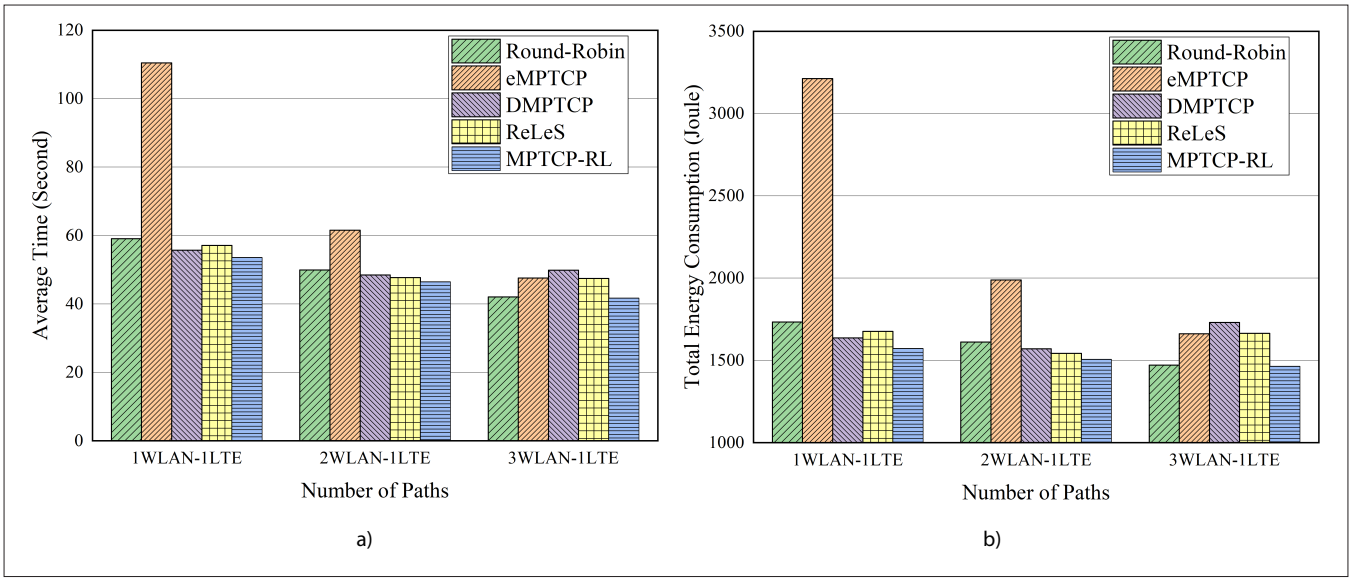

FIGURE 5. The performance of the five algorithms in scenarios with different numbers of paths where the number of LTE paths is 1 and the number of WLAN paths ranges from 1–3: a) the average time with different numbers of paths; b) the total energy consumption with different numbers of paths.

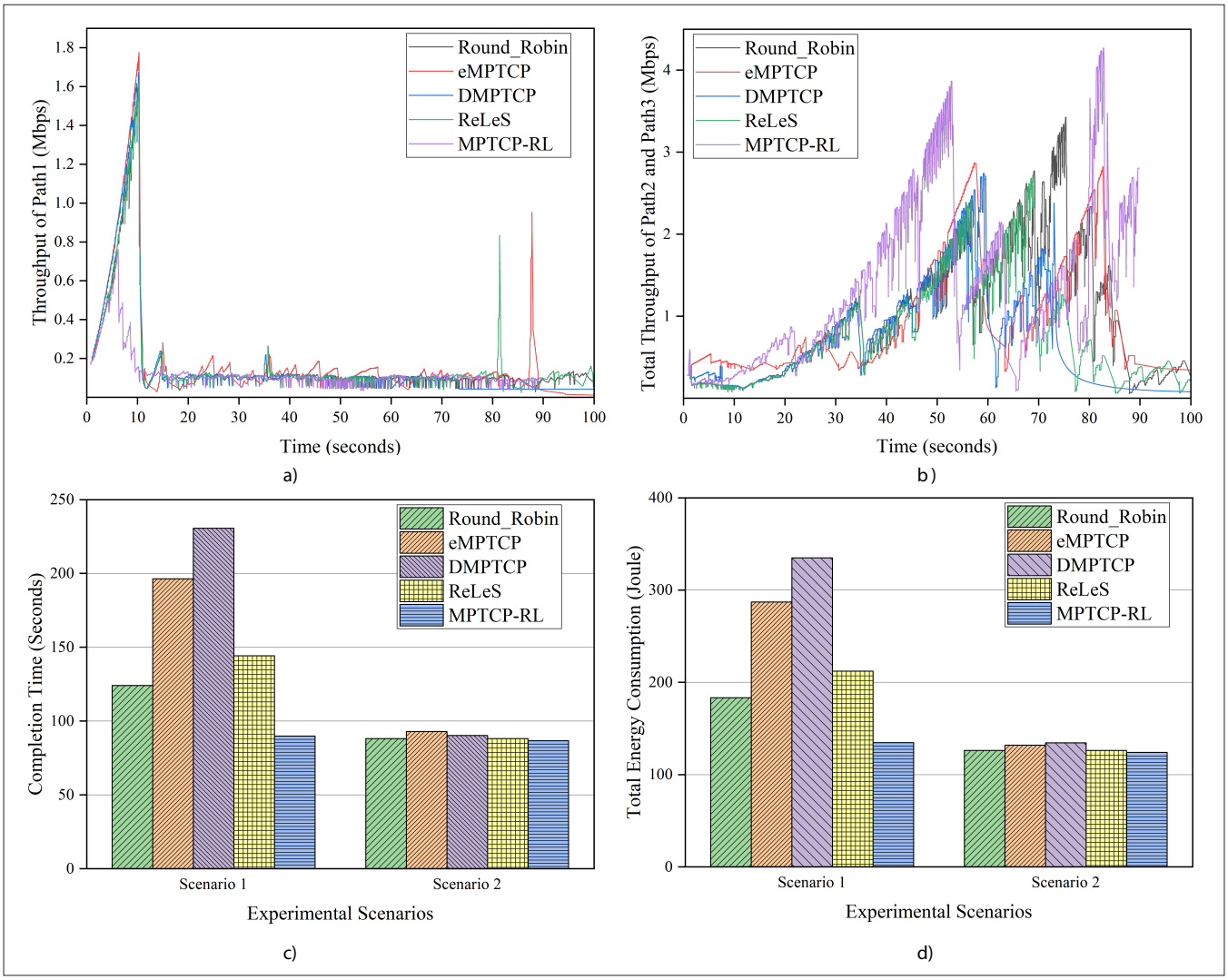

FIGURE 6. The performance of five algorithms in two dynamic scenarios: path performance change and path number change: a) Real-time throughput of path1 in scenario 1; b) real-time total throughput of path2 and path3 in scenario 1; c) The flow completion time; d) the total energy consumption.

[8] C. Pluntke *et al.*, "Saving Mobile Device Energy with Multipath TCP," *Proc. 6th Int'l Wksp. MobiArch*, 2011, pp. 1–6.
[9] C. Raiciu *et al.*, "Opportunistic Mobility with Multipath TCP," *Proc. 6th Int'l. Wksp. MobiArch*, 2011, pp. 7–12.
[10] Y. Cao *et al.*, "QoE-Driven Energy-Aware Multipath Content Delivery Approach for MPTCP-Based Mobile Phones," *China Commun.*, vol. 14, no. 2, 2017, pp. 90–103.
[11] J. Xu *et al.*, "Deep Reinforcement Learning for Handover-Aware MPTCP Congestion Control in Space-Ground Integrated Network of Railways," *IEEE Wireless Commun.*, vol. 28, no. 6, 2021, pp. 200–07.
[12] W. Li *et al.*, "SmartCC: A Reinforcement Learning Approach for Multipath TCP Congestion Control in Heterogeneous Networks," *IEEE JSAC*, vol. 37, no. 11, 2019, pp. 2621–33.
[13] H. Zhang *et al.*, "ReLeS: A Neural Adaptive Multipath Scheduler based on Deep Reinforcement Learning," *Proc. IEEE Conf. Computer Commun.*, 2019, pp. 1648–56.
[14] J. Huang *et al.*, "A Close Examination of Performance and Power Characteristics of 4G LTE Networks," *Proc. Int'l. Conf. Mobile Systems, Applications, and Services*, 2012, pp. 225–38.
[15] Y. Lim *et al.*, "Design, Implementation, and Evaluation of Energy-Aware Multi-Path TCP," *Proc. 11th ACM Conf. Emerging Networking Experiments and Technologies*, 2015, pp. 1–13.

Biographies

Pingping Dong (ppdong@hunnu.edu.cn) received her B.S., M.S., and Ph.D from the School of Information Science and Engineering at Central South University. Currently she is an Associate Professor at Hunan Normal University. Her research interests include protocol optimization and protocol design in wide area networks (WANs) and wireless local area networks (WLANs).

Rongcheng Shen (201920293209@hunnu.edu.cn) received his B.S. degree in Computer Science and Technology from the Hunan Normal University in 2019. He is currently pursuing his M.S. degree at Hunan Normal University. His current research interest includes multipath TCP, viewport prediction in 360 degree video, and deep learning.

Qian Wang (wangqian@hunnu.edu.cn) received his B.S. degree in Information and Computing Science from Hunan Agricultural University in 2019. He is currently pursuing his M.S. degree at Hunan Normal University. His research interests include deterministic networks, network calculus, tactile Internet, and multipath TCP.

Dian Zhang (202070291677@hunnu.edu.cn) received his B.S. degree in Logistics Management from the Changsha University in 2019. He is currently pursuing his M.S. degree at Hunan Normal University. His research interests include software-defined networks, route optimization, graph neural networks, and deep reinforcement learning.

Yajing Li (lyj@hunnu.edu.cn) received her B.S. degree in Internet of Things Engineering in 2019 and is currently pursuing her M.S. degree at Hunan Normal University. Her current research interest includes multipath TCP, intelligent transportation systems, and deep learning.

Yuning Zuo (YuningZuo@hunnu.edu.cn) received her B.S. degree in Applied Electronics from the Changsha Aeronautical Vocational and Technical College. She is currently pursuing her M.S. degree at Hunan Normal University. Her research interests include multipath transmission, automatic driving, and trajectory prediction.

Wenjun Yang (wenjunyang@uvic.ca) received his M.S. degree in Information Science and Engineering from the Hunan Normal University in 2019. He is currently pursuing his Ph.D. at University of Victoria. His current research interests include next generation of network architecture and related issues, such as multipath TCP, multihoming, and mobility.

Lianming Zhang (zlm@hunnu.edu.cn) received his B.S. and M.S. degrees at Hunan Normal University in 1997 and 2000, and his Ph.D. at Central South University in 2006. He is currently a professor at Hunan Normal University. His research interests include computer networks, software-defined networking, edge computing, and machine learning.