International Journal of Future Generation Communication and Networking
Vol. 8, No. 2 (2015), pp. 105-114
http://dx.doi.org/10.14257/ijfgcn.2015.8.2.09

A Throughput Improved Path Selection Method Based on Throughput Prediction Model and Available Bandwidth for MPTCP

Xiaomin Jin[1, 2], Yuanan Liu[1, 2], Wenhao Fan[1, 2], Fan Wu[1, 2] and Hongguang Zhang[1, 2]

[1]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Beijing Key Laboratory of Work Safety Intelligent Monitoring, Beijing University of Posts and Telecommunications, Beijing 100876, China
jxm@bupt.edu.cn

Abstract

Nowadays, many network devices have more than one network interface. Meanwhile, many protocols have been proposed to utilize multiple paths simultaneously. MPTCP (Multipath TCP) is a transport layer protocol that transfers data by multiple paths simultaneously, which is designed to improve the throughput and increase robustness of end to end transmission. However, when we use MPTCP, we find that MPTCP can not improve the throughput but even reduce the throughput in some cases. In this paper, we study this issue. Firstly, we study the effects of path characteristics for throughput of MPTCP, and find that path characteristics especially those bad paths have a great impact on the performance of MPTCP. Secondly, we analyze the reasons why the bad path characteristics have a serious impact on throughput of MPTCP, and propose a path selection method based on throughput prediction model and available bandwidth to improve the throughput. Finally, we validate our path selection method by simulations, and the results show that our method can improve the throughput of MPTCP.

Keywords: *MPTCP, path selection, throughput, path characteristics*

1. Introduction

With the development of wired and wireless network technology, more and more network devices are equipped with multiple interfaces (Ethernet, Wi-Fi, 3G/4G, GPRS, *etc.*). The standard Internet transport protocols can use only one network interface at the same time, which makes a waste of network resources. Many CMT (Concurrent Multipath Transport) methods and protocols have been proposed on different layers to utilize multiple paths simultaneously, such as application layer, transport layer and network layer. Compared with the application layer and network layer, the transport layer is more suitable for the deployment of CMT. MPTCP is a set of extensions to regular TCP, which enables a TCP connection to transport data through multiple paths at the same time [2]. MPTCP is designed to be backward compatible. The existing applications can use MPTCP to transport data without any modifications, and developers can use the original APIs to develop new applications. MPTCP is compatible with the Internet as it exists now, which makes it easier to be deployed.

MPTCP will increase the resilience of the connectivity by using multiple paths at the same time, and will get higher throughput by increasing the efficiency of the network resource usage. Some CMT protocols like [1] run regular TCP congestion control on each path, which is easy to implement relatively and can get high throughput added by each TCP connection theoretically. But in reality, it is not fair to standard TCP connection when they share a bottleneck link, for the CMT connection would take more bandwidth than a standard TCP flow would take at a shared bottleneck. MPTCP handles the fairness issue by using its coupled congestion control [3].The coupled congestion control algorithms do no harm to other network users and balance congestion by moving traffic away from the most congested paths [4].

MPTCP creates TCP sub flow for each path, and presents a standard socket interface to the upper layer. All those sub flows share the same receive buffer at the receiving end. Due to different parameters on different paths, path characteristics will affect the performance of MPTCP when those paths share the same receive buffer. Although MPTCP aims to improve the throughput of TCP connection, it does not meet the expected demands when we use MPTCP on our embedded network devices. The results of [5] also confirm this phenomenon, namely using multiple paths would get lower throughput than using single path in some cases.

In this paper we study the impact of path characteristics such as RTT, loss rate and available bandwidth on the throughput of MPTCP. We find that path characteristics seriously affect the throughput of MPTCP without other optimization methods such as data scheduling algorithm and packet reordering algorithm. We analyze this phenomenon and propose a path selection method based on throughput prediction model and available bandwidth to improve the throughput of MPTCP. To obtain greater throughput, more data should be scheduled to paths with best characteristics. But when bandwidth becomes a limiting factor to throughput, we need to select some second best paths to aggregate bandwidth to obtain greater throughput. The path selection method we propose is a kind of static scheduling methods to suspend data sending on bad paths to get greater throughput.

The remainder of this paper is organized as follows. In Section 2 we describe the related work. In Section 3 we study the impact of path characteristics on throughput, and then we simply analyze the reasons. In Section 4 we describe our MPTCP path selection method. In Section 5 we verify our path method with simulations. In Section 6 we conclude this paper.

2. Related Work

There are already a number of studies on improving the performance of CMT and path selection, such as [6], [7] and [8]. [6] proposes a range-based path selection method (RPS) to promote the throughput of a CMT protocol[1] modified from SCTP (Stream Control Transmission Protocol). [6] models the throughput of the CMT, and analyzes different path selection schemes based on that model. However, the protocol in [6] is not TCP-Friendly, for the throughput of that protocol is the sum of each path's throughput, so the path selection method is not suitable for MPTCP. In [7], the authors propose a MPTCP path selection strategy based on improved grey relational analysis, they choose paths according to the similarity of path characteristics calculated by the grey relational analysis. [7] provides an idea to us that grey relational analysis can be used to measure the similarity between two paths, but the authors do not consider the impact of available bandwidth and their method need to calculate a three-dimensional variable which will take more time and resources. A typical throughput model of TCP is proposed in [9], which gives a function to predict throughput by RTT and loss rate. The throughput computed from [9] is used as an important parameter for our path selection method. [10] describes an method to measure available bandwidth, which can be used to get another important parameter for our path selection method.

3. Problem and Simple Analysis

We use the code that implements multipath TCP on NS-3 to simulate. The topology is shown in Figure 1. Here, we use RTTi, Pi and BWi to denote the round trip time, the loss rate, and the available bandwidth of pathi respectively. We choose parameters according to [11, 6], which found that in most cases, loss rate is between [0.1%, 10%] and the delay is between [25ms, 360 ms].

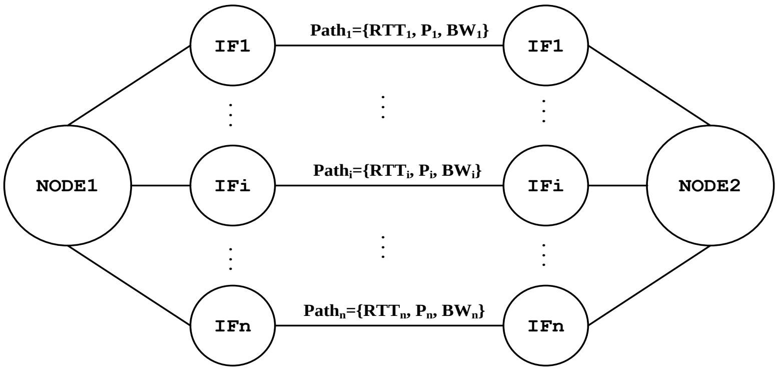

Figure 1. Simulation Topology

3.1. Problem

We once built a test bed like [12] using our embedded network devices, and observed that when both use Ethernet and Wi-Fi, the throughput is lower than only use Ethernet. In [5], the authors analyze the impact of bandwidth and RTT on MPTCP. Their results confirm that using multiple paths will reduce the throughput under some circumstances.

We do three experiments to further study the relationship between path characteristics and the throughput of MPTCP.

(a) To explore the impact of RTT on the performance of MPTCP, two paths are used to simulate. The parameters of path1 are: RTT1=100ms, P1=0.001, BW1=5Mbps. The parameters of path2 are: P2=0.001, BW2=5Mbps, RTT2=RTT1 + δ, where δ increases from 0ms to 200ms. The simulation result is shown in Figure 2.

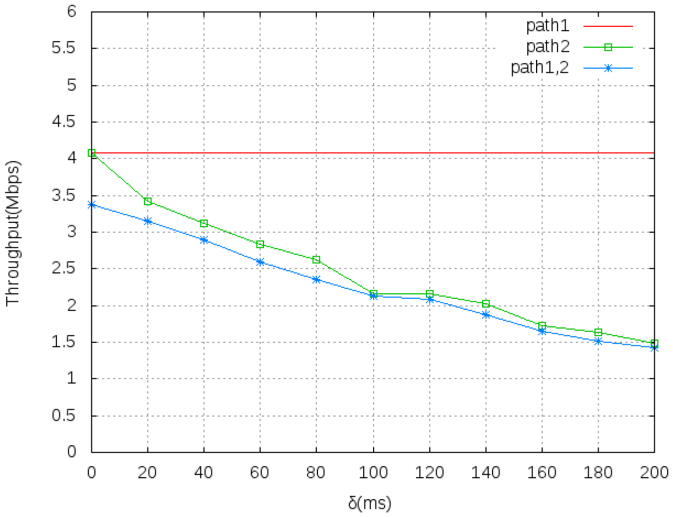

Figure 2. Impact of RTT

(b) When we explore the impact of loss rate, the parameters of path1 are: RTT1=100ms, P1 =0.001, BW1=5Mbps. The parameters of path2 are: RTT2=100ms, BW2=5Mbps, P2=ε, where ε increases from 0.001 to 0.05. The simulation result is shown in Figure 3.

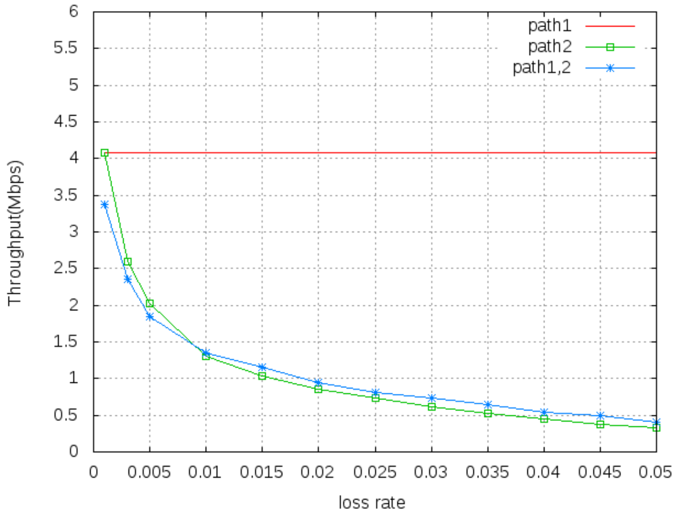

Figure 3. Impact of Loss Rate

(c) We use N (ranges from 1 to 10) bandwidth limited paths to investigate the impact of bandwidth, and the parameters of those paths are: RTT=100ms, P=0.001, BW=1Mbps. The simulation result is shown in Figure 4.

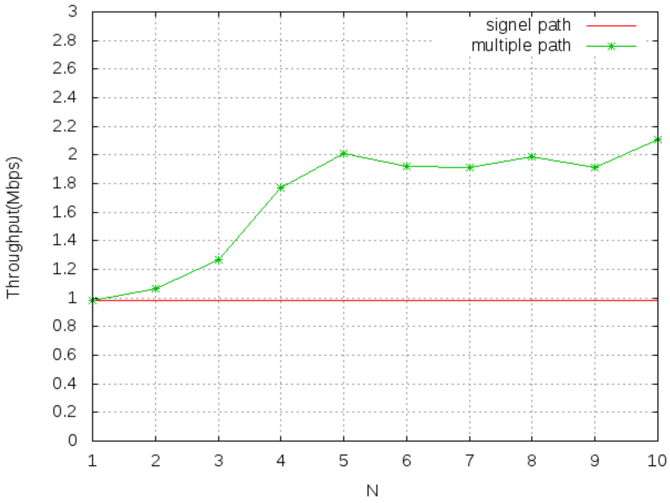

Figure 4. Impact of Available Bandwidth

From Figure 2, it can be seen that the throughput of MPTCP reduces as RTT2 increases, and the throughput of MPTCP is close to that of path2. Figure 3 illustrates that the throughput of MPTCP reduces as P2 increases, and the throughput of MPTCP is also close to that of path2. Figure 4 shows that the throughput of MPTCP increases as the number of paths increases, and will reach a stable stage.

3.2. Simple Analysis

Firstly, we introduce the congestion control algorithm of MPTCP. The congestion control algorithm proposed by [3] is as below:

- Each ACK on sub flow *r*, increase the window *wr* by min (a/*wtotal*,1/*wr*).
- Each loss on sub flow *r*, decrease the window *wr* by *wr*/2.

Here

$$a = w_{total} \frac{\max_r w_r / RTT_r^2}{(\sum_r w_r / RTT_r)^2} \quad (1)$$

According to this congestion control algorithm, the paths of higher RTT are used less often than those with lower RTT[5].Those paths with higher RTT or large loss rate will reduce the throughput of MPTCP as the upper layer would not read data from receive buffer because of the lost packets or out-of-order packets.

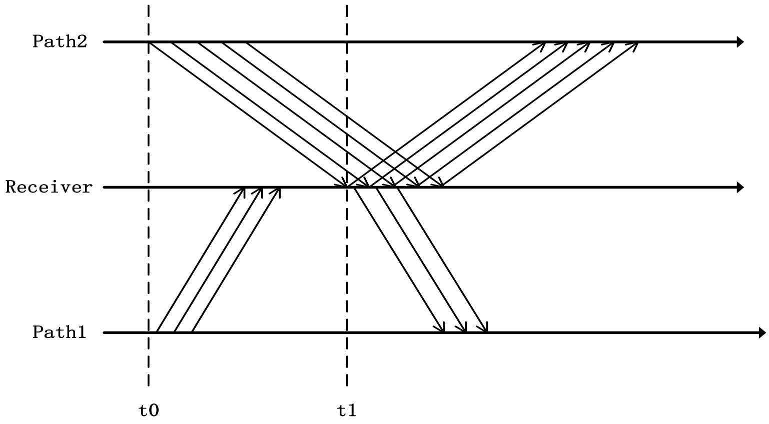

Figure 5. Packets Transmission Through Two Paths

In Figure 5, RTT2 > RTT1 and P2 > P1, at time t0 the packet (sequence number is SN1) is sent by path2, then other packets (sequence numbers are larger than SN1) are sent by path1. When MPTCP distributes data among sub flows with round-robin algorithm, the packets sent by path1 and path2 are adjacent. The application layer cannot read data from receive buffer before time t1 because packet SN1 is not arriving. The send window of path1 will gradually decrease, so more and more data are sent from path2. When there are only two paths, the throughput of MPTCP will be close to that of path2.

We find that the path characteristics have a great impact on throughput of MPTCP through experiments and analysis, especially those paths with bad path characteristics. We propose a path selection method based on path characteristics to improve throughput of MPTCP in Section 4.

4. Path Selection Method

Our path selection method is proposed based on a throughput prediction model and the available bandwidth. [9] infers a throughput model of RTT and loss rate:

$$T = \frac{1}{RTT} * \sqrt{\frac{3}{2bp}} * packet\_size \quad (2)$$

Where *RTT* is the round trip time, *p* is the loss rate and *b* is the number of packets that are acknowledged by a received ACK, here *b*=1. [13] introduces a method to measure the similarity of the two quantities in a system, and this method is used to measure the similarity between the best path and other alternative paths in this paper.

CMT promotes throughput by aggregating idle bandwidth. MPTCP will give a connection at least as much throughput as it would get with single-path TCP on the best of its paths and will not unduly harm other flows at a bottleneck link[3].MPTCP would not promote throughput endlessly for its TCP-Friendly congestion control algorithm. When the available bandwidth of the best single path is enough, the throughput of MPTCP will not increase any more than the best single path, it is useless to aggregate bandwidth to improve throughput. It is obvious that multiple paths should not be used in this case from the experiments and analysis in Section 3. When the available bandwidth of the best single path is limited, multiple paths can be used to aggregate bandwidth. From Section 3, we know that not all paths are suitable. If a path has bad characteristics, the throughput increment brought by aggregating its bandwidth is smaller than the reduction brought by its RTT and loss rate, thus this path should not be selected.

Let *I*={1, 2, …, n} represent all the available paths, and the parameters of those paths are: pathi={RTTi, Pi, BWi}.(2) is used to predict the throughput of each single path under standard TCP protocol. We define the maximum throughput *Tmax* as:

$$T_{\max} = \max_{i \in I} T_i \quad (3)$$

(1) If the available bandwidth of the best paths is not limited, it can be assumed:

$$T_k = T_{\max} \text{ and } T_k < BW_k \quad (4)$$

In this case, the set *S*={*k*|*Tk* = *Tmax* and *Tk* < *BWk*} is the best path selection scheme.

(2) If the available bandwidth of the best paths is limited, it can be assumed:

$$T_k = T_{\max} \text{ and } T_k > BW_k \quad (5)$$

In this case, the set *Sinit* = {*k*|*Tk* = *Tmax* and *Tk* > *BWk*} is the initial path selection scheme, then the method in [13] is used to choose other paths.

Let *X0* be the maximum of the available bandwidth in *Sinit*, and *Xm* be the throughput of best paths. *Xi* represents the throughput of pathi, and pathi does not belong to the initial path selection scheme.

$$X_0 = \max_{k \in S_{init}} BW_k \quad (6)$$

$$X_m = T_{\max} = T_k \quad (7)$$

$$X_i = T_i (\mathrm{i} \notin S_{init}) \quad (8)$$

When $j \notin S_{init}$ and $j \neq 0$,$\Delta_{m,j}$ represents the difference between *Tmax* and throughput of other paths. When $j = 0$,$\Delta_{m,j}$ represents the difference between *Tmax* and the biggest bandwidth of the best paths.

$$\Delta_{m,\mathrm{j}} = \mathrm{X}_m - X_j \quad (9)$$

$\varepsilon_{m,j}$ represents the similarity between X_m and X_j. The bigger $\varepsilon_{m,j}$, the closer X_m and X_j. $\zeta \in [0,1]$ is a distinguishing coefficient, the purpose of which is to weaken the effect of $\Delta_{\max}$ when it gets too big[13], here $\zeta = 0.5$.

$$\Delta_{\max} = \max \Delta_{m,j}, \Delta_{\min} = \min \Delta_{m,j} \quad (10)$$

$$\varepsilon_{m,\mathrm{j}} = \frac{\Delta_{\min} + \zeta \Delta_{\max}}{\Delta_{m,\mathrm{j}} + \zeta \Delta_{\max}} \quad (11)$$

ε_{thr} is the threshold which determines whether a path should be selected or not. When *X0* is bigger, it needs fewer paths to aggregate bandwidth. Conversely, it needs more paths to aggregate bandwidth. So, we set ε_{thr} to equal $\varepsilon_{m,0}$, when *X0* is bigger, ε_{thr} will also be bigger, less paths will be selected. Conversely, more paths will be selected. When $\varepsilon_{m,i} \geq \varepsilon_{thr}$, pathi will be selected.

$$\varepsilon_{thr} = \varepsilon_{m,0} \quad (12)$$

The path selection scheme is:

$$S = S_{init} \cup S_{oth} \quad (13)$$

where $S_{oth} = \{ \text{i} \mid \varepsilon_{m,i} \geq \varepsilon_{thr} \}$.

In a real network environment, path characteristics are constantly changing. Path selection method is a static scheduling method, we have to monitor the network parameters and rerun the path selection method and adjust path selection scheme according to it[6]. We can set a time interval Δt, and run our path selection method every Δt. MPTCP has socket APIs to add or remove paths to the existing connection, and those socket APIs can be used to add the good paths and remove the bad paths to implement the path selection method in reality.

5. Method Validation

We validate our path selection method by NS-3 simulations. The simulation topology and the range of path parameters are introduced in Section 3. The validation of our path selection method is divided into two cases based on bandwidth-limited or not.

In the first case, there are four paths:

path$_1$= {100ms, 0.001, 5Mbps}, path$_2$={150ms, 0.001, 5Mbps},path$_3$={80ms, 0.004, 3Mbps}, path$_4$={100ms, 0.006, 1Mbps}. The path selection scheme is S={1} according to our path selection method.

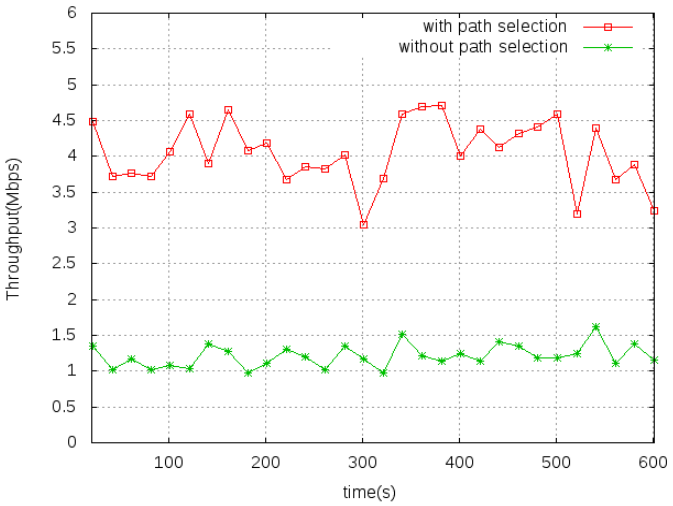

Figure 6. Throughput Comparison with Unrestricted Bandwidth

In another case where the bandwidth is limited, there are four paths:

path$_1$= {100ms, 0.001, 2Mbps}, path$_2$={300ms, 0.001, 5Mbps}, path$_3$={80ms, 0.002, 2Mbps}, path$_4$={200ms, 0.003, 5Mbps}. The path selection scheme is S={1, 3} according to our path selection method.

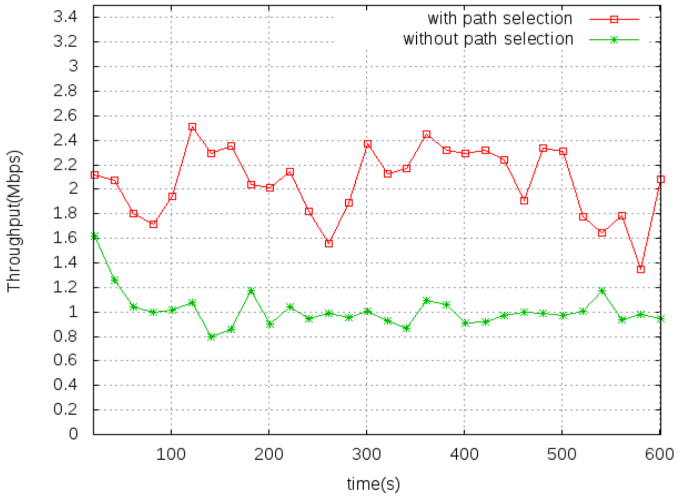

Figure 7.Throughput Comparison with Restricted Bandwidth

We use a long simulation time to get accurate results, and the throughput in figures is the average throughput. Figure 6 and Figure 7 illustrate that our path selection method improves the throughput of MPTCP, which proves that our method is useful.

6. Conclusions

In this paper we study the impact of path characteristics on throughput of MPTCP and analyze the reasons why path parameters especially bad path parameters will reduce the throughput of MPTCP, then we propose a path selection method based on throughput prediction model and available bandwidth to improve the throughput of MPTCP and validate it by simulations, and the results show that our method can improve the throughput of MPTCP. The method we propose is concise and efficient. It requires less calculation amount, which makes it feasible that our path selection method can be used in the real protocol stack.

There are many factors that can affect the throughput of MPTCP, such as data distribution algorithm, send buffer size, receive buffer size and packet reordering algorithm. One idea to improve the throughput of MPTCP is transmitting more data on the good paths and less data on the bad paths. The path selection method proposed in this paper is a kind of static scheduling methods to send no data on bad paths. In the future, we are ready to deploy this path selection method to our embedded network devices, and further study the data distribution algorithm to improve the throughput of MPTCP.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61170275/61272518), Fundamental Research Funds for the Central Universities, Beijing Higher Education Young Elite Teacher Project and Director Foundation of Beijing Key Laboratory of Work Safety Intelligent Monitoring.

References

[1] J. R. Iyengar, P. D. Amer and R. Stewart, “Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths”, IEEE/ACM Transactions on Networking, vol. 14, no. 5, **(2006)**.

[2] A. Ford, C. Raiciu, M. Handley and O. Bonaventure, “TCP Extensions for Multipath Operation with Multiple Addresses”, IETF RFC, vol. 68, no. 24, **(2013)** January.

[3] D. Wischik, C. Raiciu, A. Greenhalgh and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP", Proceedings of the 8th USENIX conference on Networked systems design and implementation, Boston, MA, **(2011)**.

[4] A. Ford, C. Raiciu, M. Handley, S. Barre and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", IETF RFC 6182, **(2011)** March.

[5] B. Arzani, A. Gurney, S. Cheng, R. Guerin and B. T. Loo, "Impact of path characteristics and scheduling policies on MPTCP performance", Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications Workshops (IEEE WAINA 2014), Victoria, BC, Canada, **(2014)** May 13-16.

[6] W. Yang, H. Li, F. Li, Q. Wu and J. Wu, "RPS: Range-based path selection method for Concurrent Multipath Transfer", Proceedings of the 6th International Wireless Communications and Mobile Computing Conference (IWCMC 2010), Caen, France, **(2010)** June 28 - July 2.

[7] L. Kou, S. R. Chen and R. Wang, "A MPTCP path selection strategy based on improved grey relational analysis", Proceedings of the 3rd International Conference on Frontiers of Manufacturing Science and Measuring Technology (ICFMM 2013), Li Jiang, China, **(2013)** July 30-31.

[8] P. Key, L. Massoulie and D. Towsley, "Path selection and multipath congestion control", Proceedings of the 26th IEEE International Conference on Computer Communications, Anchorage, AK, United states, **(2007)** May 6-12.

[9] J. Padhye, V. Firoiu, D. F. Towsley and J. F. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation", IEEE/ACM Transactions on Networking, vol. 8, no. 2, **(2000)**.

[10] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput", Proceedings of ACM SIGCOMM 2002 Conference, Pittsburgh, PA, United states, **(2002)** August 19-23.

[11] S. Shakkottai, N. Brownlee, A. Broido and K. Claffy, "The rtt distribution of tcp flows in the internet and its impact on tcp -based flow control", Cooperative Association for Internet Data Analysis (CAIDA), **(2004)**.

[12] K. Nguyen, Y. Ji and S. Yamada, "Improving Wi-Fi networking with concurrent connections and multipath TCP", Proceedings of the IEEE 14th International Symposium and Workshops World of Wireless, Mobile and Multimedia Networks (Wow Mom), Madrid, **(2013)** June 4-7.

[13] K. W. David, "Grey system and grey relational model", ACM SIGICE Bulletin, vol. 20, no. 2, **(1994)**.

Author

Xiaomin Jin, he received his B.E. from BUPT(Beijing University of Posts and Telecommunications), China, in 2012. In the year of 2012 and 2013, he was working on his master's degree in BUPT. Now, he is work-ing towards his Ph.D. in School of Electronic Engineering, BUPT. His current research interests include wireless networks and mobile devices.