Path-Selection Method Based on the Available Bandwidth of Interfaces

Path-Selection Method Based on the Available Bandwidth of Interfaces

Daiki Ishida, Daiki Nobayashi, Takeshi Ikenaga
Dept. of Electrical and Electronic Engineering
Kyushu Institute of Technology
Fukuoka, Japan
ishida.daiki700@mail.kyutech.jp, nova@ecs.kyutech.ac.jp, ike@ecs.kyutech.ac.jp

***Abstract*—The number of end devices equipped with multiple interfaces is increasing, owing to the spread of the Internet and the development of wireless communication technologies. Many applications work and communicate simultaneously in one end device. However, such end devices are difficult to utilize multiple interfaces and multiple paths effectively. Traditional transport protocols, such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), support only single-path communication, so end device applications tend to use only single interface and path. Application' flows become unbalanced and compete; hence, their performance degrades. There are many related works using multiple interfaces and paths simultaneously, such as Multipath Transmission Control Protocol (MPTCP) and Multipath Internet Protocol (MPIP). However, in some cases, these methods cannot improve the communication performance because scheduling to multipath for concurrent applications is not effective. In this paper, we propose the path-selection method for each application based on the available bandwidth of interfaces of a device. This method can utilize network resources and improve the performance of all applications when many applications work and communicate on a device equipped with multiple interfaces. We evaluate this method using network simulations and show its usability.**

***Keywords*-multiple paths, multiple interfaces, load balancing, MPTCP**

I. Introduction

End devices equipped with multiple interfaces have become widespread and the cost of network interfaces has been greatly reduced because of the development of Internet and wireless technologies. For example, PCs, such as laptops, have Ethernet and Wi-Fi, and smartphones have 3G/4G, Wi-Fi and so on, as shown in Fig. 1. Many devices are available to multipath communication by using multiple interfaces. Multipathing provides many advantages (e.g., load balance and good performance) by increasing the available bandwidth [1].

To utilize multiple paths effectively, many methods are investigated to realize better throughput in multipathing. In the transport layer, Multipath Transmission Control Protocol (MPTCP) is well known and implemented to various operating systems [2]. MPTCP divides the flows of each application into available multiple paths in the end-to-end connection. Hence, performance improvement is possible because of the simultaneous establishment of TCP flows on different paths. However, current MPTCP path manager does not consider the bandwidth of paths; hence each subflow uses the bandwidth of available paths greedily. Thus, fairness of communication for multiple applications on a device is difficult because MPTCP flows affect the performance of other communication networks with many connections [3]. Meanwhile, a Multipath Internet Protocol (MPIP) is proposed to address in the network layer [4]. Its multipathing works in the network layer, thereby indicating that any application or transport layer protocols can utilize the multipath. However, MPIP requires path-management mechanisms or needs to feed back the availability and performance of each end-to-end path, and these mechanisms are not available in a traditional network layer. Thus, it needs major changes from existing network layer.

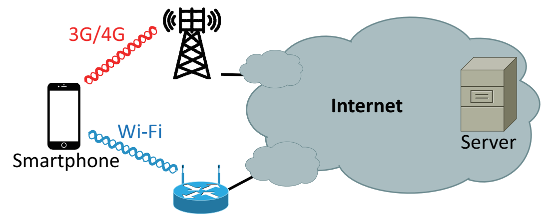


Fig. 1. Multiple interfaces, multipath environment.

In this paper, we propose a path-selection method to improve the communication performance by controlling the paths in each application of end devices. This method assigns a flow of each application to a certain interface based on the available bandwidth of the devices' interfaces. The path-selection method can effectively utilize resources and contribute to performance improvement of all applications and fairness.

The remainder of this paper is organized as follows. Related works and their challenges are presented in Section II. Our proposed method is explained in Section III. The usefulness of the proposed method based on the simulation evaluation is discussed in Section IV. Finally, the conclusion of this study, as well as future works, is provided in Section V.

II. Related Works

Recently, MPTCP has become an active research topic as a transmission method that can simultaneously utilize multiple paths. It was proposed as RFC 6182 at Internet Engineering Task Force (IETF) in 2011 and standardized as RFC 6824 in 2013, which is new transport protocol [4] [5]. MPTCP establishes multiple IP paths in end-to-end and transports data by using TCP flows called subflows. Hence, it can improve the throughput and resilience. MPTCP has another significant advantage: being compatible with traditional network infrastructures, such as middle boxes and socket APIs, because each subflow is handled as a normal TCP flow. Thus, it has elicited considerable attention as the most used Internet architecture of the future. However, UDP-based applications cannot benefit from this advantage because MPTCP is only an expansion of TCP. MPTCP does not have a function to distribute traffic to interfaces intelligently; hence, it disaggregates flows of all applications to each interface. These disaggregated flows use available subflows at a maximum. Thus, MPTCP seems to have a crucial influence on other applications using UDP or single-path connections.

MPIP controls the multipath transmission in the network layer of end devices. MPTCP has a benefit for TCP-based applications because it is an expansion of TCP. However, MPIP is also compatible with UDP-based applications. Flexible routing is possible in MPIP because it controls multiple paths in the network layer. However, it requires additional mechanisms to feed back end-to-end path information, a design to pass through the middle boxes, and a multipath IP routing method.

Concurrent multipath data transfer using stream control transmission protocol (CMT-SCTP) is an expansion of stream control transmission protocol (SCTP), which can use multiple paths simultaneously. However, the efficiency of CMT-SCTP deteriorates and the data cannot be sent to the receiver in order when the delay difference between paths is large. Thus, one study assigns the transmission data by considering the differences between multiple paths when each path has a heterogeneous network environment [6]. However, this method only considers the delays as path characteristics.

Applications, such as music and videos, often have extra quality requirements compared to system performance. Because it may be related to link scheduling, the scheduling policy called remaining time based maximal (RTBM) is proposed [7].

As described above, research and development of multipathing, which can transport data by using multiple interfaces, become popular, in response to an environment in which many devices have multiple interfaces. However, in existing methods that use multiple interfaces, several problems remain (e.g., MPTCP divides the flows of all applications to subflows and uses resources at a maximum, and MPIP requires major change to an existing network infrastructure).

III. Proposed Method

In this research, we focus on the two facts. Firstly, various works are researched for multipathing but they cannot decide the path based on the state of the path. Secondly, many applications work simultaneously on one device, such as a smartphone, and connect various destinations. We propose a path-selection method that can assign paths for each application based on the available bandwidth of the device's own interfaces. This method monitors the number of flows

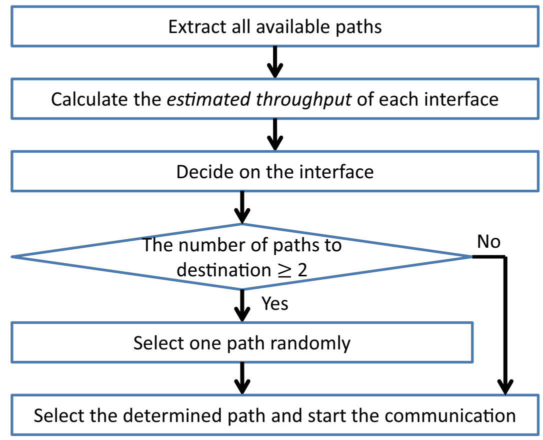


Fig. 2. Diagram of the proposed method.

through each interface of a device, and calculates the estimated bandwidth for each interface. The newly generated application uses the path with larger estimated bandwidth.

The method has the following advantages:

- It can utilize resources effectively by switching and using the interfaces of the device.
- It does not unnecessarily divide all flows into subflows because it allocates a flow to the path *for each application*.
- It does not require major changes of networks to obtain path information because it uses only the interface information of the devices.

The path selection steps are described in the following and are shown in Fig. 2:

1) Extract all available paths.
   - When a new communication occurs, the device lines up all candidate paths based on its own and destination host's interface information.
2) Calculate the *estimated throughput* of each interface.
   - The device counts the number of flows streaming through each interface. Then it calculates the estimated throughput, assuming a case when new communications begin by using the following equation:

$$E_i = \frac{B_i}{N_i + 1} \quad (1)$$

   where i is the interface ID ($i = 1, 2, ...$), E_i is the estimated throughput of the interface i, B_i is the bandwidth of i, and N_i is the current number of flows of i.
3) Decide on the interface.
   - The device decides on the interface that has the maximum estimated throughput E_i.
4) Select the determined path and start the communication.
   - In deciding a unique path, a new application starts the communication by using the path. If more than one path to the destination exists in the selected

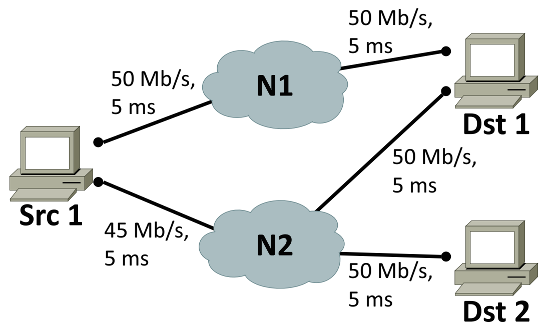


Fig. 3. Simulation topology.

TABLE I
SIMULATION PARAMETER.

Network Simulator	ns3
Congestion Control (TCP)	NewReno
Congestion Control (MPTCP)	LIA
Number of Flows	20
Amount of Data per Flow	50 MB
Flow Generation Interval	Random from 0.0 s to 0.1 s
Number of Trials	10

interface i, then the application randomly selects one among the paths.

The method repeats the above steps every time new communication occurs and realizes a throughput improvement of the entire device.

IV. SIMULATION EVALUATION

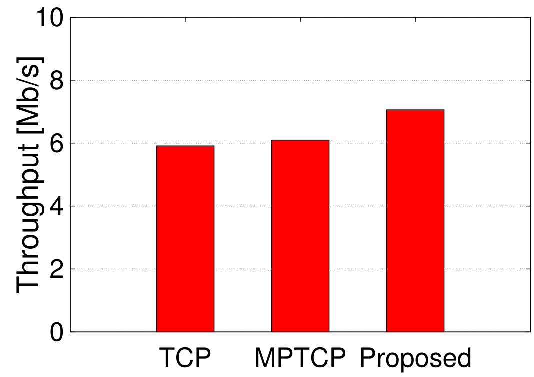
A. Simulation-Evaluation Scenario

By using the network simulator ns3, we evaluate the performance of the proposed method above. The simulation network topology is shown in Fig. 3. In this evaluation, we assume that destinations corresponding to multipath and single-path transmission are present and that these communications compete on a certain path. Fifteen flows arise in the path from Src 1 to Dst 1, whereas five flows arise in the path from Src 1 to Dst 2. Hence, we start these 20 flows at random time intervals from 0.0 seconds to 0.1 seconds. Each flow transfers 50 MB of data. The number of trials is 10. We compare the results of three methods i.e. TCP NewReno, MPTCP, and the proposed method. The simulation parameter is presented in Table I.

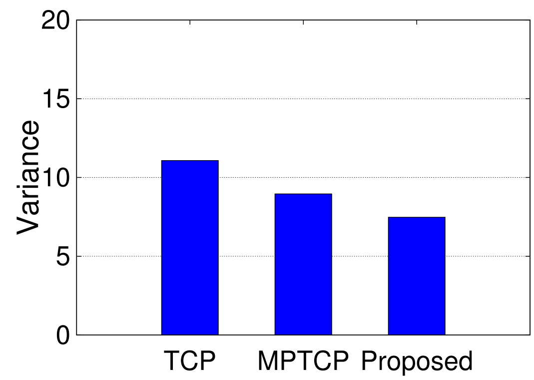
B. Simulation Results

We show the simulation results in Figs. 4 – 5, Tables II and III. Figure 4(a) and Table II show the comparison of the three methods related to the average throughput of 20 flows on Src 1. Figure. 4(b) and Table III present the variance of per-flow throughput. The "Rate" in the tables are based on TCP NewReno. Figure 5 presents the throughput of each flow in the proposed method case.

Then, we compare MPTCP to TCP (Fig. 4(a) and Table II). MPTCP has very minimal throughput improvement compared with the TCP, although it uses multipathing. This is because single-path and multipath transmissions compete on the path



(a) Average throughput of all flows.



(b) Variance of all flows.

Fig. 4. Comparison of the three methods.

TABLE II
AVERAGE THROUGHPUT OF ALL FLOWS.

Method	Throughput [Mb/s]	Rate [%]
TCP NewReno	5.91143	100.0
MPTCP	6.09602	103.1
Proposed Method	7.05805	119.4

TABLE III
AVERAGE THROUGHPUT OF ALL FLOWS.

Method	Variance	Rate [%]
TCP NewReno	11.0716	100.0
MPTCP	8.95648	80.9
Proposed Method	7.47498	67.5

Src 1 – Dst 2. MPTCP cannot display the performance well because the capabilities of the subflows between N1 and N2 become unbalanced. Meanwhile, the proposed method does not use multiple path greedily. In this case, the applications to Dst 2 can use only one path, so some flows of applications to Dst 1 used path through N1 not N2. Thus, the proposed method can avoid the flow contentions and achieve high throughput by selecting the appropriate path.

As shown in Fig. 4(b) and Table III, MPTCP slightly reduces the variance compared with the single-path TCP. In the single-path TCP, the 15 flows of application to Dst 1 used only one path through N1 and share the link bandwidth. Hence, the performance difference between applications was marked.

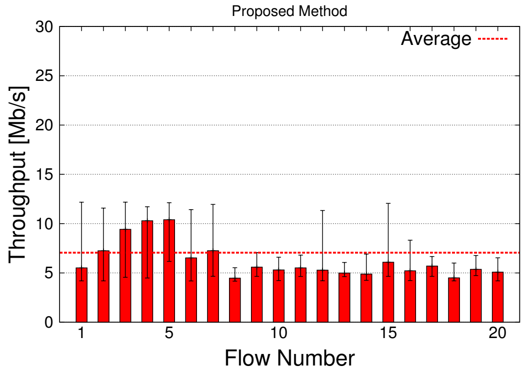


Fig. 5. Throughput of each flow in the proposed method.

This is slightly improved in MPTCP. The proposed method has higher variance reduction. Hence, the performance difference of each flow is minimal, thereby confirming that applications can effectively utilize network resources by using proposed method (Fig. 5).

As described above, several cases exist in which MPTCP cannot improve the throughput compared with TCP. Meanwhile, the proposed method selects the appropriate path. The result shows that improving both the throughput and variance is possible.

V. Conclusion

Many end devices have multiple interfaces. However, we cannot fully use its environment currently. In our research, we focus on an environment in which the devices are equipped with multiple interfaces and many applications transport data simultaneously on one device. The goal of this research is to achieve performance improvement of the applications by efficiently using network resources. In this paper, we propose a path-selection method based on the availability of the device's own interfaces. This method monitors the number of flows through each interface of the device and calculates the estimated bandwidth. The new generated application selects the path with larger estimated bandwidth. We revealed usefulness of this path-selection method by simulation.

In the future, we will propose a method that can select and use not only a single path but also multiple paths. Moreover, we will consider the end-to-end characteristics of a network as a criterion for the path selection.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 16K00131.

References

[1] Sana Habib, et al., "The past, present, and future of transport-layer multipath," Journal of Network and Computer Application 75, pp. 236–258, Sep. 2016.

[2] Tomohiko HANAI, Katsuyoshi IIDA, "Performance Evaluation of UDP and TCP with Subflow Control Policies using Multipath TCP under Different RTTs," IEICE Technical Report NS2016-9 (2016-04), pp. 41–46, Sep. 2016.

[3] Allen L. Ramaboli, et al., "Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues," Journal of Network and Computer Applications 35, pp. 1674–1690, Jun. 2012.

[4] Liyang Sun, et al., "Multipath IP Routing on End Devices: Motivation, Design, and Performance," arXiv:1709.05712v1, pp. 1–13, Sep. 2017.

[5] Yu Okada, et al., "System Investigation of a Gateway Implementing Subflow Control Policies using a Multipath TCP Proxy," 2015 Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), pp. 1–6, Aug. 2015.

[6] Xiaokai Li, Xiangli Zhang, "A Concurrent Multi-Path Transmission of Data Allocation Algorithm Based on Heterogeneous Networks," The 2016 3rd International Conference on System s and Informatics (ICSAI 2016), pp. 487–491, Nov. 2016.

[7] Xu Zheng, et al., "A Study on Application-Aware Scheduling in Wireless Networks," IEEE Transactions on Mobile Computing, Vol. 16, No. 7, pp. 1787–1801, Jul. 2017.