Machine Learning based Path Management for Mobile Devices over MPTCP

Jonghwan Chung
Seoul National University
chungjohn@snu.ac.kr

Dahyeon Han
Seoul National University
hann862@snu.ac.kr

Jiyoung Kim
Seoul National University
beyoundj@popeye.snu.ac.kr

Chong-kwon Kim
Seoul National University
ckim@snu.ac.kr

***Abstract*—Recent mobile devices are equipped with multiple network interfaces such as LTE and Wi-Fi. Transport protocols that can transfer data over multiple paths, especially MPTCP (Multipath TCP), allows the devices like smartphones and tablets to exploit both interfaces concurrently. However, in real environments, wireless devices abound and network quality changes frequently. It makes network connection affect the MPTCP performance negatively. In this paper, we propose a novel path management scheme called MPTCP-ML (MPTCP based on Machine Learning) to make MPTCP troubleshoot the problem. It manages path usage among multiple connections based on decision computed by machine learning model. For accurate capturing of path quality, we utilize various quality metrics including signal strength, data rate, TCP throughput, the number of interference APs, and RTT (Round Trip Time). We have implemented MPTCP-ML in Android and conducted experiments for various and dynamic environments. The results show that MPTCP-ML outperforms generic MPTCP, especially for mobile environments.**

***Index Terms*—MPTCP, Wi-Fi, Path Management, Smartphone**

I. Introduction

Today, mobile devices like smartphones and tablets are normally equipped with multiple wireless radio interfaces such as cellular (3G/4G/LTE) and Wi-Fi (IEEE 802.11). The legacy devices use one of them for data communication between two endpoints. In other words, the device transfers and receives data traffic through only one wireless network interface at a time. It is not a new argument that the conventional policy for network usage is crude. For example, when Wi-Fi interface comes up, it automatically associates a specific Wi-Fi AP only considering connection history and signal strength. Under the policy, the device cannot get out of performance degradation problem which results from various reasons including signal interference and congestion.

MPTCP (Multipath TCP) [2], [4], an extension of TCP, is one of possible solutions to cope with the trouble. The key idea is to split a single byte stream to multiple byte streams and transfer them over multiple disjoint network paths. It adds path diversity to a traditional TCP in order to expedite throughput and achieve robustness. However, the gains cannot be achieved in the cases where there exist active paths with different quality such as signal coverage, loss rate, performance of involving links, and connectivity of access networks [7], [12]. In real environments, mobile device has high probability to encountering such cases [9], [13]. As long as the connection is alive, MPTCP does not completely get rid of data traffic from it (i.e. a small amount of data is on the path), which hurts the performance of generic MPTCP. Absence of intelligent path management causes this problematic situation in current MPTCP.

We propose a novel path management scheme called MPTCP-ML which controls path usage based on machine learning mechanism. It samples features of path quality periodically and detects status of active paths in real-time by utilizing pre-built random forests model which relies on insight based on historical patterns discoverable in collected data. When it detects a path which experiences poor performance, MPTCP-ML suspending use of the path, not keeping it. We conducted experiment in real environments, including the places where network traffic jams, signal interference are commonplace, and signal strength of the connected access point is weak. And we also consider mobile environment. The results show that MPTCP-ML outperforms generic MPTCP with accurate detection of path quality in mobile environments.

The remainder of this paper is organized as follows. Section II briefly provides background and reviews related work. The design of our proposed path management scheme is elaborated in Section III, followed by performance evaluation in Section IV. We conclude this paper in Section V.

II. Background and Related Work

Multipath TCP (MPTCP), designed for mobile devices, was standardized at [2]. The purpose of it is maximizing resource usage by enabling multiplexing over multiple wireless networks. Since it is able for MPTCP to create multiple network paths within a connection as far as possible, path management scheme (i.e. making a decision when and how paths are created or destroyed) is needed. At the time of writing, two primary schemes are implemented in MPTCP Linux kernel [1]: *Fullmesh* and *Ndiffports*. The first scheme is used as default. It is possible to create a full-mesh of paths among all available active paths. At the beginning of connection, after the initial path has been validated, it creates paths using each pair of IP addresses owned by client and advertised by server. If the client learn a new IP address (e.g. a mobile device connects to a new Wi-Fi AP), it automatically generates a new path over this interface. The second scheme is designed for single-homed hosts in datacenters. With this scheme, MPTCP initiates a connection consisting of a specific number of paths across the same pair of IP addresses that use different source port.

In conjunction with path management schemes, MPTCP has three modes of operation to control path usage [3]. Primary mode is a standard mode of operation that utilizes all available interfaces. In Backup mode, MPTCP selects only a subset of active paths for data transmission. The remaining paths are kept idle as backups. If all the established paths in primary mode become unavailable, MPTCP transfer traffic over the paths in backup mode. In Single-path mode, MPTCP utilizes only one path at a time like the behavior of a traditional TCP.

To achieve gain of MPTCP, we consider the effect of user mobility on the performance. [5], [9], [10] conducted the extensive experiments to show that MPTCP can brings advantages to mobile devices in various environments. However, there are some cases where mobile users face some severe performance degradation due to poor and intermittent connectivity of Wi-Fi APs [6], [7], [12]. In order to solve the problem, some researches have been tried to design intelligent path management scheme of which explicit goal is controlling the path usage efficiently. [7] proposed cross-layer path management which controls path usage based on link layer status such as signal strength, data rate and ratio of the number of frame retransmission to the number of successful frame transmission. Based on the status, it suspends data transmission on intermittently connected path and releases the path. [8] implemented a path control plane including useful functions related to path management. By using the functions, applications is able to control the utilization of the different paths.

Unlike the previous researches, we apply machine learning technique to handling performance degradation problem of MPTCP. In order to get an acceptable performance for identifying path quality, we utilize diverse quality features which can be obtained from wireless network and mobile device.

III. Path Management Algorithm

A. Random Decision Forests

There have been a number of machine learning techniques to learn from observed data and make predictions on data. Random decision forests [11], an extension of decision tree that constructs multitude of random decision trees and amalgamates them together, fits the requirements of our approach among the diverse techniques: (a) It can support high classification accuracy and stable prediction. (b) It can ease off overfitting to its training data set by building a forest composed of multiple decision trees at training time. (c) It is a fast and light-weight method like decision tree, thus making it possible to embed our scheme in commodity devices.

B. Tree Learning and Modeling

There are a lot of factors influencing Wi-Fi performance. They include signal strength from the associated Wi-Fi AP, signal interferences from nearby wireless devices, and network congestion. To reflect these features, we select channel status, interference degree, and network performance as classification features. The channel status is comprised of RSSI and data rate. RSSI is the signal strength from the associated Wi-Fi AP and data rate is the rate of the communication channel including both uplink and downlink. The network performance consists of TCP throughput, gateway RTT and end-to-end RTT. TCP throughput is the amount of data received successfully from sender given time period. Gateway RTT and end-to-end RTT means that the delay from when a signal is sent to when its response is received from gateway and remote endpoint, respectively. We define the degree of interference as the number of nearby Wi-Fi APs on same and adjacent channel of which the signal strength is greater than -80 dBm that is the minimum value for basic connectivity.

TABLE I
Classification Performance

	Classification Performance Metrics				
Model	TP Rate	FP Rate	Precision	Recall	OOB error
RF-2C	0.98	0.03	0.98	0.98	0.04
RF-4C	0.93	0.07	0.93	0.93	0.07

To extract the peculiarities of each case where a path experiences poor performance, we collected enough data for one week. Making full use of the data, we generate two random forests models for path quality classification. One is RF-2C (Random Forests with two Class) that classifies output in two classes: enable, disable. The other is RF-4C (Random Forests with four Class) that subdivide the disable class into three classes: Weak signal strength, interference, and congestion.

Table I shows classification performances of the two models. As shown in Table I, both models support high accuracy and low error rate for prediction task. The accuracy and error rate of RF-2C are 98% and 4%, respectively. And both values of RF-4C are 0.93 and 7%, respectively. RF-4C can classify path quality in more detail, but it needs higher memory requirement than RF-2C since the number of result classes affects complexity of random forests. As a result, RF-2C is more suitable for our work.

C. Our Algorithm

The aim of our algorithm is to control Wi-Fi path efficiently based on machine learning mechanism. It consists of three components: Interference Scanner, Feature Sampler, and MPTCP-ML Core.

Interference Scanner performs interference checking task before a mobile device associates with a Wi-Fi AP. It scans nearby Wi-Fi APs available to connect and then extracts information about channel number and signal strength of each AP. After acquiring the information, it computes the interference degree using the number of APs whose signal strength is greater than -80 dBm. During use of Wi-Fi interface, the calculation of interference degree takes into account the signal strength and channel number of the connected AP. If the interference degree is lower than the threshold determined by random forest model, it allows the devices to associate with a Wi-Fi AP.

Features Sampler collects feature data of path quality every 100 ms after association and feeds the data into MPTCP-ML Core. The collected data (i.e. signal strength from the associated Wi-Fi AP, data rate, TCP throughput, gateway RTT, end-to-end RTT and interference degree) is combined to constitute a data set. Since the end-to-end RTT is not always obtained within the sampling period, it records the value as empty only when the delay between two endpoints becomes longer than the sampling interval.

MPTCP-ML Core predicts path quality based on pre-built random forests model and handles Wi-Fi path. The model takes as input the data set obtained from the sampler and outputs the predicted quality of Wi-Fi path. It determines use of the path according to the result. If the result is that the path experiences poor performance, MPTCP-ML suspends data communication over the path, not keeping it.

The above process is repeated to identify the path quality of the associated Wi-Fi AP continuously.

IV. Performance Evaluation

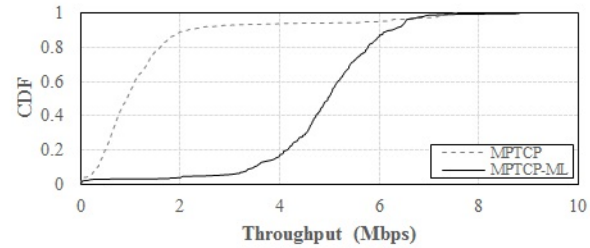
A. Experiment Setup

Our testbed comprises a wired server, residing at the SNU (Seoul National University) and two mobile clients. The server is connected through a single Gigabit Ethernet interface to SNU network and is running Ubuntu Linux 12.10 with Kernel version 3.5.7 using the recent stable release of MPTCP Linux Kernel implementation version v0.90. We used two android smartphone (Google Nexus 5) as clients where MPTCP stack is ported. We implemented MPTCP-ML at application layer because of portability and convenience about getting a variety of information from wireless networks and mobile device. The pre-built random forests model is embedded in the application. We use LTE network provided by KT (Korea Telecom) and free Wi-Fi network for experiment. To fairly compare MPTCP-ML to generic MPTCP, we conducted the experiments using the two smartphones at the same time. We first demonstrate the better performance of MPTCP-ML under static environments and then conducts experiments in mobile environments.

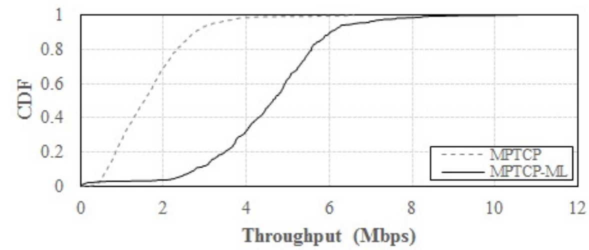
B. Performance under Static Environment

Before evaluating MPTCP-ML performance in mobile environments, we first measured two MPTCPs (MPTCP and MPTCP-ML) performances in static environments to identify effect of poor quality Wi-Fi path on the performance. We focused on the environments where mobile device is affected negatively by weak signal strength from an associated AP, signal interference, and network congestion.

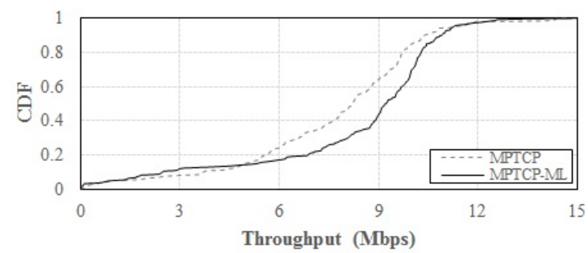
Figure 1 presents the cumulative distribution of throughput for the different scenarios where the two smartphones exploit both LTE and Wi-Fi network and the surroundings have a negative influence on the Wi-Fi path. We observe that MPTCP-ML achieves higher aggregate TCP throughput than MPTCP in these scenarios. As shown in Figure 1 (a) and (b), there exists a significant difference in performance between MPTCP and MPTCP-ML. The average aggregate throughputs of the MPTCP and MPTCP-ML for weak signal strength scenario are about 1.5 Mbps and 4.8 Mbps, respectively. In signal interference scenario, the values are about 1.6 Mbps and 4.5 Mbps. The throughput of MPTCP-ML is improved by 3.2 times (under weak signal strength) and 2.8 times (under interference). The values can be varied by experiment settings. This results are related with the signal quality influenced by distance and density of wireless devices and access points. To sum up, MPTCP-ML outperforms MPTCP since it considers signal information. Figure 1 (c) presents congestion has less impact on the aggregate throughput. Owing to the default packet scheduler of MPTCP at a sender side, the amount of packets can be adjusted according to end-to-end RTT of each active path, which brings advantages for MPTCP.



(a) Weak signal strength



(b) Signal interference

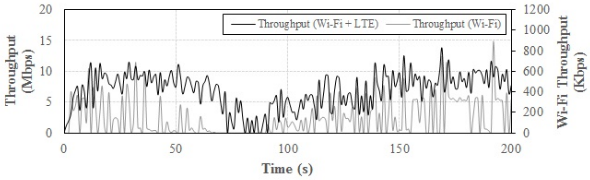


(c) Congestion

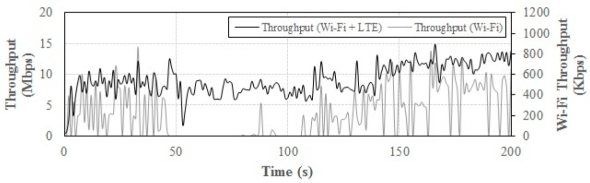
Fig. 1. MPTCP Performance Comparisons. (a) Weak signal strength from the connected Wi-Fi AP, (b) Signal interference, and (c) Network congestion

C. Performance under Mobile Environment

In this section, we compare the performance of MPTCP-ML with that of MPTCP in real mobile environments where the mobile device experiences severe performance degradation. To



(a) MPTCP



(b) MPTCP-ML

Fig. 2. Throughput over time when using LTE and Wi-Fi

analyze the performance and behavior of MPTCP-ML, we let the device to download a large file from our server. The most obvious performance improvement that can be expected from the use of MPTCP-ML is an increase in throughput, since it accompanied intelligent path management mechanism, which is depicted in Figure 2.

Figure 2 (a) and (b) shows a sample trace for each MPTCP expressed as a time series where the two MPTCPs are using LTE with Wi-Fi. The figure presents the aggregate throughput as a function of time when each MPTCP uses both LTE and Wi-Fi, in black, on the left Y-axis. The figure also shows the Wi-Fi throughput, in light gray, on the right Y-axis. Since different experimental conditions such as status of wireless network and routing path can served to confuse the analysis of performance and behavior, we conducted the experiments using the two smartphones at the same time. During the interval (50, 100), the associated Wi-Fi path provides poor performance. It affects the aggregate throughput of MPTCP negatively, which leads to performance degradation of MPTCP. However, MPTCP-ML suspends traffic over it at the moment. As a result, the aggregate throughput of MPTCP-ML is higher than that of MPTCP. From this experiment, we identify how effectively MPTCP-ML works with user mobility. In short, current MPTCP does not quickly detect and handle the path that experiences poor performance. In contrast, we can observe that MPTCP-ML quickly detects the path using the path quality metrics and suspends use of the path.

V. Conclusion

In this paper, we propose a novel path management mechanism named MPTCP-ML that relies on machine learning to handle active paths efficiently in real-time, especially for Wi-Fi path. We build a well-trained random forests model using the various path quality metrics obtained from wireless network and mobile device. It determines whether a path suspends or not according to the result from the prediction model. After implementing it in a mobile device, we perform experiments in an actual environments and have demonstrated that MPTCP-ML achieves significantly better performance than the generic MPTCP with high prediction accuracy. For future work, we will design more adaptive path management scheme to extend our work to cover wide and complex real environments and are also interested in energy consumption to reduce energy which is used to utilize multiple network interfaces.

Acknowledgment

This research was supported by the MSIP (Misistry of Science, ICT and future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-R0992-16-1023) supervised by the IITP (Institute for Information and communications Technology Promotion) grant funded by the Korea government (MSIP) (No.B0190-16-2017, Resilient/Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices) and partly supported by the Institute for Industrial Systems Engineering of Seoul National University.

References

[1] Multipath TCP: Linux Kernel Implementation http://multipath-tcp.org/

[2] A.Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, TCP Extension for Multipath Operation with Multiple Address, IETF RFC 6824, 2013.

[3] O. Bonaventure, Q. De Coninck, M. Baerts, F. Duchene, and B. Hesmans. Improving Multipath TCP Backup Subflows, IETF RFC 6824, 2015.

[4] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, et al., How hard can it be? designing and implementing a deployable multipath TCP. *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, 2012.

[5] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley. Opportunistic mobility with multipath TCP. *Proceedings of the sixth international workshop on MobiArch*, 2011.

[6] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure. Exploring mobile/WiFi handover with multipath TCP. *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, 2012.

[7] Y.-s. 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. *Proceedings of the IEEE INFOCOM 2014 Conference on Computer Communications*, 2014.

[8] B. Hesmans, G. Detal, R. Bauduin, and O. Bonaventure. SMAPP: Towards Smart Multipath TCP-enabled APPlications. *Proceedings of the 2015 ACM Conference on emerging Networking Experiments and Technologies*, 2015.

[9] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan. Wifi, lte, or both?: Measuring multi-homed wireless internet performance. *Proceedings of the 2014 Conference on Internet Measurement Conference*, 2014.

[10] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley. A measurement-based study of multipath tcp performance over wireless networks. *Proceedings of the 2013 conference on Internet measurement conference*, 2013.

[11] L. Breiman and A. Cutler. Random forests-classification description. Department of Statistics, Berkeley, vol. 2, 2007.

[12] S. Bae, D. Ban, D. Han, J. Kim, K.-h. Lee, S. Lim, et al., StreetSense: Effect of Bus Wi-Fi APs on Pedestrian Smartphone. *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, 2015.

[13] B. Han, F. Qian, S. Hao, L. Ji, and N. Bedminster. An Anatomy of Mobile Web Performance over Multipath TCP. *Proceedings of the 2015 ACM Conference on emerging Networking Experiments and Technologies*, 2015.