공학석사 학위논문

기계 학습 기반 무선망 선택 기법

A Wireless Network Selection Scheme
based on Machine Learning

2017년 2월

서울대학교 대학원

컴퓨터공학부

정 종 환

기계 학습 기반 무선망 선택 기법

A Wireless Network Selection Scheme based on Machine Learning

지도교수 김 종 권

이 논문을 공학석사 학위논문으로 제출함

2017년 2월

서울대학교 대학원

컴퓨터 공학부

정 종 환

정종환의 공학석사 학위논문을 인준함

2017년 2월

위 원 장 : 전 화 숙 (인)

부위원장 : 김 종 권 (인)

위 원 : 권 태 경 (인)

Abstract

A Wireless Network Selection Scheme based on Machine Learning

Jonghwan Chung
Department of Computer Science & Engineering
The Graduate School
Seoul National University

Today, more and more public Wi-Fi APs are deployed for user convenience in response to the growing number of smartphone users. However, when both Wi-Fi and LTE interface are on, mobile devices usually select Wi-Fi interface regardless of its connection quality or stability. The blind preference of Wi-Fi interface can degrade user experience called QoE (Quality of Experience).

We present a decision tree based network connection management named SISA (Smart Interface Selection Algorithm) which takes account of user context, contents and network features such as RSSI, Link Speed, TCP throughput and Gateway RTT (RTT to gateway) to efficiently select wireless radio interfaces. We observed the effect of intermittent connection or continuous connection with low quality Wi-Fi AP on the TCP throughput. Through extensive measurements and experiments in real-field, especially public places, we confirmed that the mobile devices embedded our platform outperforms commodity devices for various scenarios. We also discovered that there is no correlation between network features measured in real-field.

We extend the problem to MPTCP (Multipath TCP) that allows the devices like smartphones and tablets to exploit both interfaces concurrently. Similar with phenomenon over SPTCP (Single Path TCP), in real mobile environments where wireless devices abound and quality of active paths changes frequently, it makes Wi-Fi connection affect the MPTCP performance negatively. To solve the problem, we present a novel path management called MPTCP-ML (MPTCP based on Machine Learning). It manages the usage among multiple paths based on decision calculated by machine learning model. We use path quality metrics as inputs for machine learning model. For accurate capturing of path quality, we utilize different quality metrics including signal strength, data rate, TCP throughput, the number of APs on the same and adjacent channel, and RTT (Round Trip Time). We have implemented MPTCP-ML in Android and conducted experiments for various and dynamic mobile environments. The results show that MPTCP-ML outperforms generic MPTCP, especially for mobile environments.

Keyword : Public Wi-Fi, MPTCP, Path Management, Network Interface, Smartphone, QoE

Student Number : 2015-21263

Contents

Abstract i

Contents iii

List of Figures V

List of Tables VI

Chapter 1. Introduction 1
1.1 Motivation 1
1.2 Background and Related Work 3
1.3 Goal and Contribution 5
1.4 Thesis Organization 7

Chapter 2. Problem Motivation 8
2.1 Problem Analysis 8
2.2 Correleation 12
2.3 Multipath TCP 13

Chapter 3. StreetSmart: Select the Network Best for You 14
3.1 Introduction 14
3.2 Interface Selection Algorithm over SPTCP 16
3.2.1 Decision Tree 16
3.2.2 Learning and Decision Boundary 17
3.2.3 System Design 18
3.3 Performance Evaluation 20
3.3.1 Experiment Setup 20
3.3.2 Application Layer Performance 21
3.3.3 Transport Layer Performance 27
3.4 Summary 29

Chapter 4. Machine Learning-based Path Management for Mobile Devices over MPTCP 30
4.1 Introduction 30
4.2 Path Management Algorithm over MPTCP 32
4.2.1 Random Decision Forests 32
4.2.2 Tree Learning and Modeling 33
4.2.3 System Design 35

4.3 Performance Evaluation 36
4.3.1 Experiment Setup 36
4.3.2 Performance under Static Environment 37
4.3.3 Performance under Mobile Environment 39
4.4 Summary 41

Chapter 5. Conclusion 42
5.1 Research Contributions 42
5.2 Future Research Directions 43

Bibliography 44

Abstract in Korean 46

List of Figures

Figure 2.1: One snapshot of wireless traces in street 9

Figure 2.2: One snapshot of wireless traces on bus.....10

Figure 2.3: Downlink throughput of Multipath TCP (over LTE and public Wi-Fi) and Single path TCP (over LTE) 13

Figure 3.1: Web Browsing.....22

Figure 3.2: Video Streaming 24

Figure 3.3: Music Streaming 26

Figure 3.4: Throughput Comparison.....28

Figure 4.1: MPTCP Performance Comparison 38

Figure 4.2: MPTCP and Wi-Fi throughput over time when using Wi-Fi and LTE 40

List of Tables

Table 2.1: Correlations 12

Table 3.1: Switching Threshold 17

Table 4.1: Classification Performance 34

Chapter 1

Introduction

1.1. Motivation

Aided by rapid technology advancements, more and more mobile devices are supposed to connect to public Wi-Fi network via nearby AP. In the view of cellular operators, it is a most important goal to satisfy potential requirements of subscribers. Operators planned and deployed Wi-Fi hotspots in public places such as street and bus stop and on public transportations such as subway and bus to extend signal coverage. Now at the time of writing this paper, there are more than 285 million Wi-Fi APs are deployed globally [1]. However, like other capitals or large cities, the public places and transportations are crowded with passengers or pedestrians during rush hour. For example, the number of passengers taking buses and subways are 5.8 million and 7.9 million per day respectively and the number of them is peek during rush hour in Seoul [2, 3]. As a result, in spite of the proliferation of Wi-Fi APs, user experience can be deteriorated at the places. It is caused by encountering performance anomalies, cross technology interference, shadowing, node density and etc. In this environments, we are not able to pinpoint the reasons of unexpected QoE (Quality of Experience) precisely.

Many 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 devices transfer and receive data traffic through only one wireless network interface at a time. Moreover, when both these interfaces are on, they usually select Wi-Fi interface regardless of its connection quality or stability. It is not a new argument that the conventional policy for network usage is crude. For example, when Wi-Fi interface comes up, the device tries to connect a specific Wi-Fi AP among connectable APs. It automatically selects a Wi-Fi AP for association only considering connection history and signal strength. Under the policy and blind use of network interfaces, the device cannot get out of performance degradation problem which results from various reasons including signal interference and congestion (i.e. it cannot efficiently cope with the case in which it is trapped in poor quality network path).

MPTCP (Multipath TCP) [6, 10] an extension of TCP, is one of possible solutions to cope with the trouble. This protocol adds path diversity to a traditional TCP in order to expedite throughput and achieve robustness. The key idea is to split a single byte stream to multiple byte streams and transfer them over multiple disjoint network paths. Prior performance studies of MPTCP over LTE and Wi-Fi networks corroborated MPTCP expedites TCP throughput by leveraging path diversity [15, 16, 21].

However, the gains cannot be achieved in the cases where there exist different quality of active paths used for communication such as signal coverage and loss rate, performance of involving links, and stability of access networks [8, 19]. [8] showed that poor quality Wi-Fi, the norm at public places in Seoul, may assert negative effects on MPTCP unless clever load balancing mechanisms are accompanied. [9, 14] studied that mobile device has high probability to encountering such cases, which hurts the performance of generic MPTCP. As long as connection of the path is alive, MPTCP does not completely get rid of data traffic from it (i.e. a small amount of data is on the path). Absence of method to intelligently manage the path in current MPTCP causes this problematic situation.

1.2. Background and Related Work

This section shows background material related to MPTCP. Multipath TCP (MPTCP) was standardized at [6]. The purpose of MPTCP is maximizing resource usage by enabling multiplexing over multiple wireless networks, designed for mobile devices.

Since it is able for MPTCP to create multiple network paths within a connection as far as possible, path management (i.e. making a decision when and how paths are created or destroyed) is needed to manage the paths for data communication. At the time of writing, two primary schemes of path management are implemented in MPTCP Linux kernel [5]: *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. A path is created from each IP address owned by client to each IP address advertised by the server. At the beginning of the connection, after the initial path has been validated, it creates paths using each pair of the addresses. 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 [7]. 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. [9, 17, 21] 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 [8, 18, 19].

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. [19] 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. [20] 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 computer networking and reflect varied path qualities which can be obtained from wireless networks and mobile device to get an acceptable performance of identifying path quality.

1.3. Goal and Contribution

In spite of the proliferation of Wi-Fi Aps, user experience can be deteriorated at the place. It is caused by encountering performance anomalies, cross technology interference, shadowing, node density and etc. And under the crude network policy and blind use of wireless radio interfaces mobile devices cannot get out of performance degradation problem which results from various reasons including signal interference and congestion (i.e. they cannot efficiently cope with the case in which it is trapped in poor quality network path). To address these problems and further improve the performance, we propose two schemes which make the devices exploit their wireless radio interfaces effectively.

One is a novel platform for SPTCP which can be aware of user context and contents. In addition, it has low memory and processing requirements. The goal of our scheme is to utilize network interfaces of mobile device, especially smartphone, expeditiously based on machine learning mechanism in order to elevate the network performance, ultimately user experience. Since, our platform is trained from real-field data during six months, it reflects real field well. Hence with our platform user can enjoy the high quality of experience compared to legacy and MPTCP devices.

The other is a novel path management for MPTCP called MPTCP-ML which controls path usage based on machine learning mechanism. MPTCP-ML samples features of path quality periodically and detects status of active paths in real-time utilizing pre-built random forests model which relies on insight based on historical patterns discoverable in collection 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 connected access point is weak. And we also consider user mobility environment. The results show that MPTCP-ML outperforms generic MPTCP with accurate detection of path quality in mobile

environments.

We also conducted real-field measurements and proved that unlike general thoughts, particularly in complex environment there is not any correlation between network features such as RSSI, Link Speed and TCP throughput.

1.4. Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we show problem motivation and analyze the root cause of performance degradation problem. In Chapter 3, we present a decision tree based connection management scheme for SPTCP and evaluate the performance in real-field environments. In Chapter 4, we present a path management scheme for MPTCP that relies on machine learning mechanism and conduct performance comparison in mobile environments. In Chapter 5, we conclude this thesis.

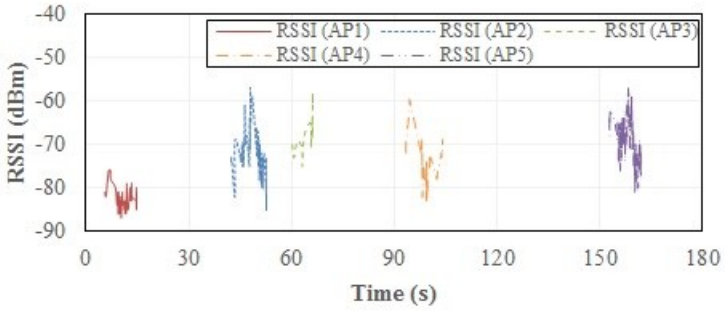
Chapter 2

Problem Motivation

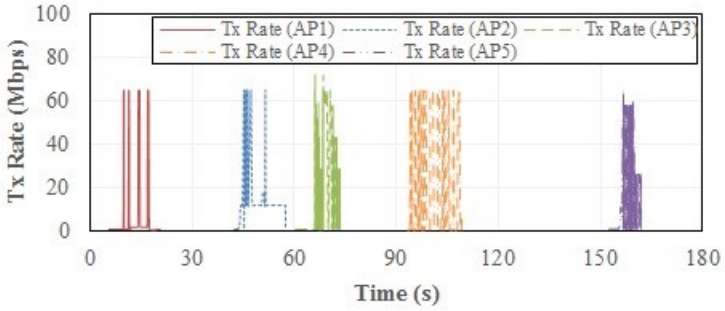
2.1. Problem Analysis

We now scrutinize the problem in detail delving into Data link and TCP layer performance. Since environment of users is changed as they move, network condition is different. For instance, pedestrians walking in public area while they use the Wi-Fi network face the problem due to intermittent connection of nearby fixed APs. As they stop and go, the APs actualize intermittent connection opportunities. However, the connection durations are too short such that the APs are a nuisance rather than a convenience. Such a scenario is depicted in Figure 2.1.

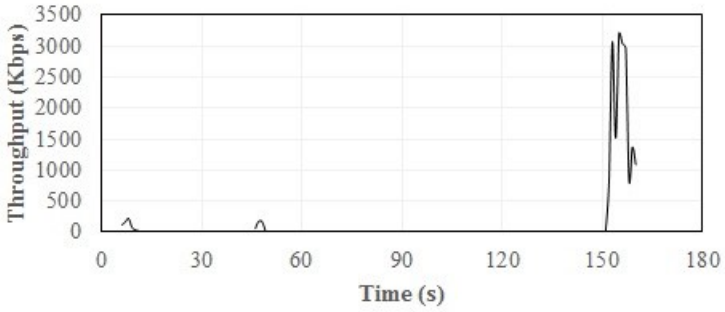
Figure 2.1 (a) and (b) presents signal strength and downlink transmission rate from nearby fixed APs installed on street when pedestrian download a 256MB file from file hosting server. During this period, the smartphone made connections to five different APs. The signal strengths and downlink transmission rates from different APs are represented in different colors. Figure 2.1 (c) shows that the short-lived TCP connections affects the performance negatively. At 6 and 42 second, the IP address of the smartphone is changed as its Wi-Fi connections are switched. At the moment, the new connections fail to utilize the available bandwidth and the throughput of Wi-Fi increases from the minimum (slow start).



(a) RSSI



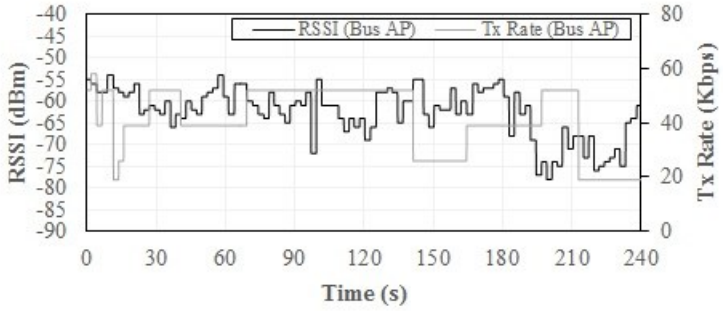
(b) Tx Rate



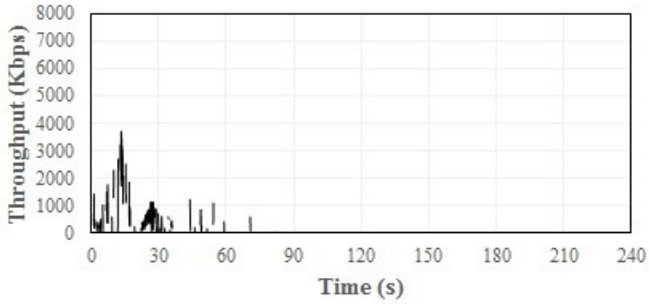
(c) TCP Throughput

Figure 2.1: **One snapshot of wireless traces in street.** (a) RSSI, (b) Tx rate and (c) TCP Throughput from nearby fixed APs.

In reference with bus stops, passengers waiting at the bus stops are provided with the intermittent connection opportunities by buses with 802.11 APs like behaviors of the pedestrians and the nearby APs in street.



(a) RSSI and Tx Rate



(b) TCP Throughput

Figure 2.2: **One snapshot of wireless traces on bus.** (a) RSSI and Tx rate from AP installed in a bus. (b) TCP Throughput.

In other example, recent public transportations such as buses and subways start to equip Wi-Fi APs. To support mobility like quick handovers between heterogeneous networks, KT (Korea Telecom), a major cellular operator in Korea, connects the 802.11 APs to a P-GW (Packet data network Gateway) over an LTE connection. The installed APs supply the connected devices with relative stable connectivity. However, riders who use the Wi-Fi network on public transport suffer from the similar problem due to congestion. Especially during rush hour the congestion occurs more frequently in components of the network such as routers, switches and APs.

Figure 2.2 shows an experiment for the problem expressed as a time series. Figure 2.2 (a) presents signal strength and downlink transmission rate from an

AP installed in public bus. The AP provisions stable connection to the connected device but there is no valid relation between the variation of transmission rate and signal strength. Figure 2.2 (b) shows that the congestion exert a bad influence on the performance. After 30 second, the throughput is dropped rapidly due to congestion which leads to poor user experience.

2.2. Correlation

Wireless networks are very intricate, and the quality of them, especially for Wi-Fi, is influenced by various factors in mobile environments. To investigate if there truly exist any correlation between network features, we measured RSSI, Link Speed and TCP throughput at the same time. Their correlations for each environment are shown in Table 2.1.

From Table 2.1 we can see that among three of correlations, the pair of link speed and throughput has the most explicit correlation, and it is the only one which has all positive values that are around 0.5. In other words, there exists somewhat correlation between link speed and throughput. However, even though the pair of link speed and throughput has some correlation, the value of link speed is inappropriate simply to predict throughput directly. Others represents almost no correlation, even some of them are negative. This means that it does not make sense to use RSSI and Link Speed to predict Wi-Fi throughput (or QoE).

Table 2.1: **Correlations**

	RSSI-Throughput	**RSSI-Link Speed**	**Link Speed-Throughput**
Street	0.19	0.15	0.41
Bus	- 0.06	0.17	0.34
Subway	0.25	0.19	0.54
Bus Stop	0.33	0.38	0.56

2.3. Multipath TCP

To identify that poor Wi-Fi connections pervert MPTCP, we measured the throughputs of both SPTCP (Single Path TCP) and MPTCP while pedestrian downloads a 512MB file in street. Since MPTCP communicates via multiple network paths, packets transmitted over paths which provide poor quality arrive in out of order at receiver side. The packets may bring about HoL (Head of Line) Blocking. Figure 2.3 represents the cumulative distribution function of the throughputs. As expected, MPTCP performs worse than the generic SPTCP over the LTE network because the congestion window of good quality path cannot be increased due to the out-of-order packets.

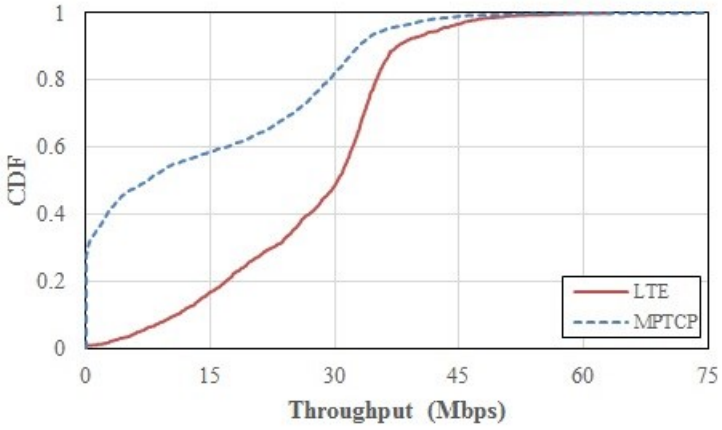


Figure 2.3: **Downlink throughput of Multipath TCP (over LTE and public Wi-Fi) and Single path TCP (over LTE).**

Chapter 3

StreetSmart: Select the Network Best for You

3.1. Introduction

Nowadays lots of public Wi-Fi APs are deployed for user convenience in response to the growing number of smartphone users. However, when both Wi-Fi and LTE interface are on, smartphones usually select Wi-Fi interface regardless of its connection quality or stability. The blind preference of Wi-Fi interface can degrade user experience called QoE (Quality of Experience).

We design a novel platform which can be aware of user context and contents. It takes account of user context, contents and network features such as RSSI, Link Speed, TCP throughput and Gateway RTT (RTT to gateway) to identify network quality of current path. In addition, it has low memory and processing requirements.

The goal of our scheme is to utilize network interfaces of mobile device, especially smartphone, expeditiously based on machine learning mechanism in order to elevate the network performance, ultimately user experience. Since, our platform is trained from real-field data during six months, it reflects real field well. Hence with our platform user can enjoy the high quality of experience compared to legacy and MPTCP devices.

Through extensive measurements and experiments in real-field, especially

public places, we confirmed that the mobile devices embedded our platform outperforms commodity devices for various scenarios.

3.2. Interface Selection Algorithm over SPTCP

3.2.1 Decision Tree

As mentioned in Section 2.2, there is almost no correlation between TCP throughput and other network features such as signal strength and link speed. Hence, we apply machine learning mechanism to solve the obstacle. There have been a number of classification methods to learn from data and make predictions on data. Among the diverse methods, decision tree better fits the requirements of our approach.

- It provides the clear relationship between input data (the collected data from real-field) and output prediction.

- It is a simple and light-weight method, thus making it possible to embed our scheme in commodity devices.

- It has lots of potential to be enhanced. Decision tree has been researched in wide area [11, 12].

3.2.2 Learning and Decision Boundary

We choose user context like location and user contents like the type of application as decision features. We also consider channel status of Wi-Fi. The channel status is comprised of RSSI, link speed, throughput and gateway RTT (Round Trip Time). RSSI is a signal strength from connected AP and link speed is a rate of the communication channel including both uplink and downlink. Gateway RTT means that the time delay from when a signal is sent to when its response is received at device. In order to build a well-trained decision tree, we collected enough data from real-field for six months. Hence, we are able to avoid from overfitting (unrealistic data properties and biased).

We decided LTE performances as training boundaries of Wi-Fi performances. Making full use of the data, we obtained switching thresholds for the varied applications: Web browser, Video streaming, Music streaming, SNS (Social Network Service) and Online game. We use Chrome web browser to load web page. We use YouTube app for video streaming and SoundCloud app for music streaming. We select Facebook app and HearthStone app as the representative of SNS and online game respectively. We compute the finish time of each operation and then express the values in seconds. The values are shown in Table 3.1. According to user preference, the values can be adjusted in the range of them. We use equal time length from both the thresholds in our experiments. The overall precision and recall of decision tree are 80.6% and 81.3%, respectively.

Table 3.1: **Switching Threshold**

	Application				
Switching Threshold	Web	Video	Music	SNS	Game
From LTE to Wi-Fi	3.19	4.56	3.30	2.50	27.30
From Wi-Fi to LTE	7.20	6.50	6.80	5.00	38.80

3.2.3 System Design

The goal of our algorithm is to improve user experience called QoE (Quality of Experience), especially in public places. It consists of three detection mechanisms: context detection, content detection and Wi-Fi quality detection.

First of all, we consider both traffic patterns of applications and user mobility to reflect the user environment. To perform the detection functionalities, it gathers data periodically from built-in sensors. To classify the user context, it samples sensor data from barometer and accelerometer at a frequency of 1Hz and processes them using smoothing filter. According to user moving speed obtained from accelerometer sensor, it distinguish bus stop, street and vehicle. To classify the vehicle as bus and subway, it uses the characteristic of passengers going to take subway. For example, user is walking and the difference of height obtained from barometer sensor is more than 6m during 30 seconds. The output is location with high overall accuracy (96.17%): street, bus, subway and bus stop. The next task is to detect user content which is the type of application in progress. It categorizes applications according to pre-defined table and then reports the result to decision tree at same frequency with context detection. The detection of Wi-Fi quality is based on the features described in Section III-B. For every 100ms, Wi-Fi quality detection obtains signal strength and link speed from the connected AP and measures the gateway RTT using ICMP (Internet Control Message Protocol) packet. It considers the TCP throughput only when the device use Wi-Fi network. Decision tree takes the outputs from the three detection mechanisms as input, and make a decision whether the smartphone uses Wi-Fi network or not. The above process is repeated in the course of communication.

A challenge in the design is that overhead occurs when the switching is operated between heterogeneous radio interfaces (switching overhead). To address the difficulty, we modified the platform of smartphone that can switch between them in real time with small overhead by enabling both the interfaces

and notifying the Wi-Fi NIC driver of the time when it operates. In addition, due to high priority of Wi-Fi, legacy devices can transfer data over only the Wi-Fi interface even though multiple interfaces are on. To solve the problem, we also modified the platform that is able to transfer data over LTE when Wi-Fi interface is on.

3.3. Performance Evaluation

3.3.1 Experiment Setup

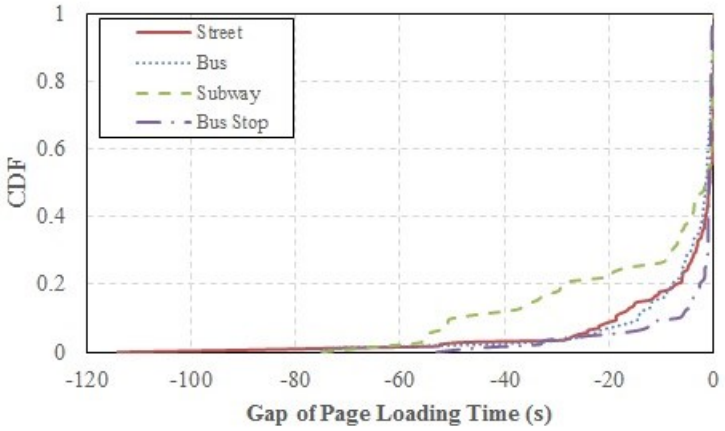
In order to provide satisfactory performance to subscribers, cellular operators planned and deployed Wi-Fi APs at any location. We used the APs and LTE network provisioned by KT (Korea Telecom), a major cellular operator in Korea. We have conducted extensive measurements and experiments at various public places located in Seoul (Gangnam and Yangjae) for six months. These include many streets, bus stops, and public transportations such as subway (Seoul Metro) and bus. Our experiment scenario includes these places where network traffic jams and wireless interference are commonplace and signal strength of Wi-Fi AP is weak due to crowd. To reflect these environments, we collected the data during rush hour. We implemented our algorithm in Android platform [4] (Android version 4.4.4; KitKat). To evaluate the performance of our algorithm named SISA, we used two smartphones (Google Nexus 5). One is the smartphone that is commonly used, and the other is the smartphone where the novel platform we developed is ported. To fairly compare our scheme to original one, we conducted the measurements using the smartphones at the same time. As we mentioned before, we concentrate on environments where the Wi-Fi AP provides poor quality to the connected users which leads to affect user experience negatively.

3.3.2 Application Layer Performance

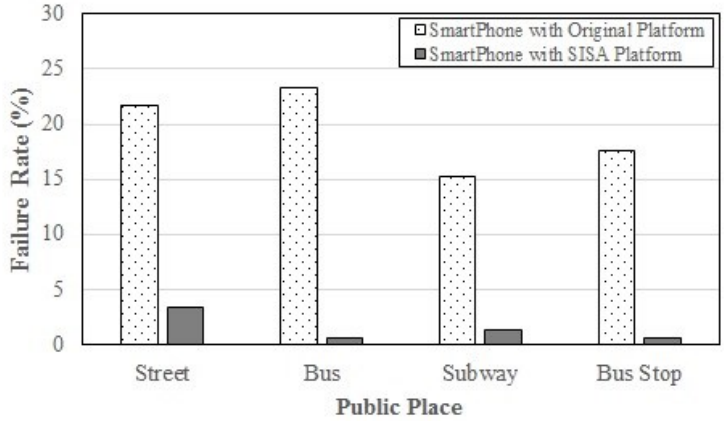
For Internet applications, PLT (Page Loading Time), page loading failure rate and latencies are important application level quality metrics. We first measured the time taken to load the Yahoo home page (https://www.yahoo.com). The PLT is defined to be the time difference from when the request is submitted to finish time. We measure the time for both smartphones at the same time and then compare both PLT.

Figure 3.1 (a) shows the gap between the smartphone with our platform and the smartphone with original one. It can happen that the smartphone with our platform results in a similar PLT with the smartphone with original one. The results may result from at least two reasons. First, the public Wi-Fi networks does not always provide poor quality. Second, false positives can occur if several cases betide. We can see however that the gains are greater for our scheme with more than 60% of successful web page loading with less than 10 seconds.

We define a trial as a failure if the web page loading does not finish for a long time and error message is printed on the web page. Figure 3.1 (b) illustrates the page loading failure rates when Wi-Fi or LTE connections are used to download the web page in street, at bus stop, on subway and bus, respectively. The failure rates of our scheme are much less than those of original one because it tries to exploit the good and stable network, not the poor and unstable network, when the poor quality network is connected.



(a) PLT gap

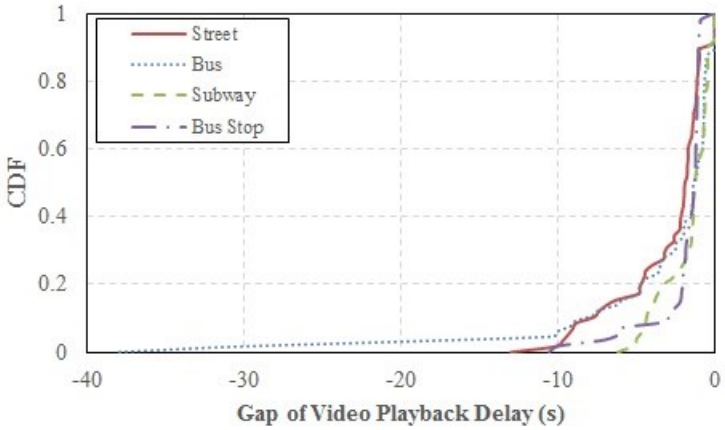


(b) Failure rate

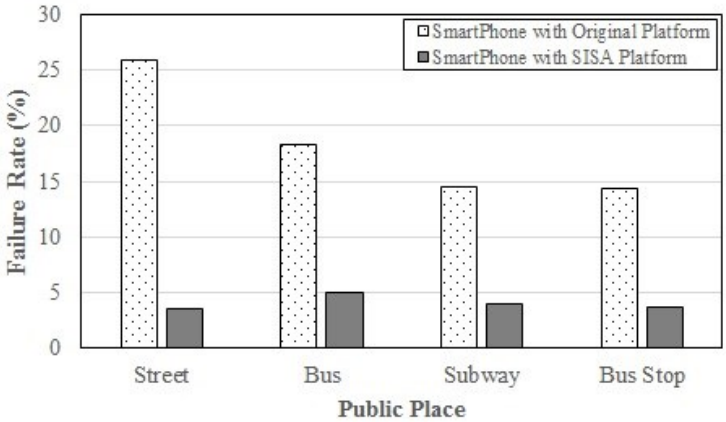
Figure 3.1: **Web Browsing.** (a) PLT gap between the smartphone with SISA platform and the smartphone with original platform (b) Failure rate of loading a web page.

We now compare our platform to original platform for video streaming. We use both video playback delay and failure rate as the user experience. The video playback delay is defined to be the time from when the video request is submitted to when video starts playing. Again, we define a trial as a failure if the video streaming application waits more than 60 seconds. We measured these performance metrics using YouTube application. With respect to streaming service, reducing the delay is one of the important factors affecting the performance because the service is sensitive to it. The large variation of the delay before the video is decoded and displayed can result from some network problem such as packet loss or long round trip time.

We represent the gap of the video playback delay in Figure 3.2 (a). Except for the small portion of similar performance between our scheme and legacy one, the smartphone with our platform reduces the playback delay about 3 seconds compared with the smartphone with no modified one for all cases. This results show that our scheme mitigates the adverse impact of high variation of the delay on user experience. Figure 3.2 (b) shows failure rates of playing video streaming content. While the failure rates of the smartphone with no modified platform are more than 14% for all cases, those of the smartphone with our platform are less than 5%. It means that our platform significantly uplifts the performance versus stability.



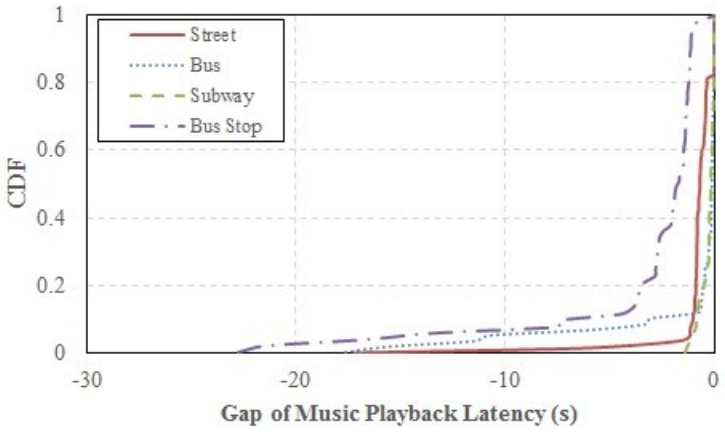
(a) Playback delay gap



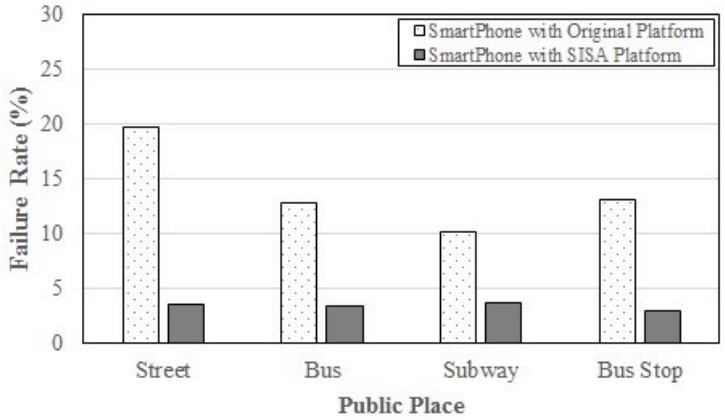
(b) Failure Rate

Figure 3.2: **Video Streaming.** (a) Playback delay gap between the smartphone with SISA platform and the smartphone with original platform (b) Failure rate of loading video content.

We also compare our platform and original platform for music streaming. The music playback latency is defined to be the difference between the time the client submits the music request and the time the music starts playing. We define a failure rate as a ratio of the number of the fails of streaming start to the number of successes anew. We collected the data using SoundCloud application. We present in Figure 3.3 (a) and Figure 3.3 (b) the gap of music streaming latency and failure rate, respectively. Similar with video streaming, the overall latency of starting music streaming is shorted, especially for bus stop, and latency variation is decreased in the others. The failure rate of street, bus, subway and bus stop are declined 16.2%, 9.5%, 6.5% and 10.2%, respectively. According to the results, our scheme utilizes the network interface efficiently to offer a better user experience for the smartphone users.



(a) Playback latency gap



(b) Failure rate

Figure 3.3: **Music Streaming.** (a) Playback latency gap between the smartphone with SISA platform and the smartphone with original platform (b) Failure rate of starting music streaming.

3.3.3 Transport Layer Performance

In this section, we investigate the effects of our algorithm on transport layer performance. User perceived performance called QoE (Quality of Experience) may be the most important performance measure but network level performance is also important. In Figure 3.4, we compare the throughputs of the legacy smartphone, the smartphone with our platform and the smartphone ported MPTCP. Due to lack of space, we have only shown the comparison of throughput to take download a 512MB file through these devices in street.

In public environments, the smartphone with SPTCP is easily trapped in poor quality Wi-Fi APs installed in public areas and transportations and establishes new connections to nearby APs frequently. Since there is no method to handle the problem in the legacy smartphone, the intermittent connections with Wi-Fi APs installed in street affects throughput of SPTCP negatively. The average throughput of our scheme, as expected, is significantly greater than that of SPTCP with normal policy of network interface usage; 20.9 Mbps for LTE, 320 Kbps for Wi-Fi on average. Different from SPTCP, our algorithm avoids the connections using machine learning (decision tree).

Through this experiment, we observe that the smartphone with our scheme outperforms the legacy smartphone. And our scheme also achieves more aggregate throughput than MPTCP even though MPTCP utilizes both Wi-Fi and LTE simultaneously.

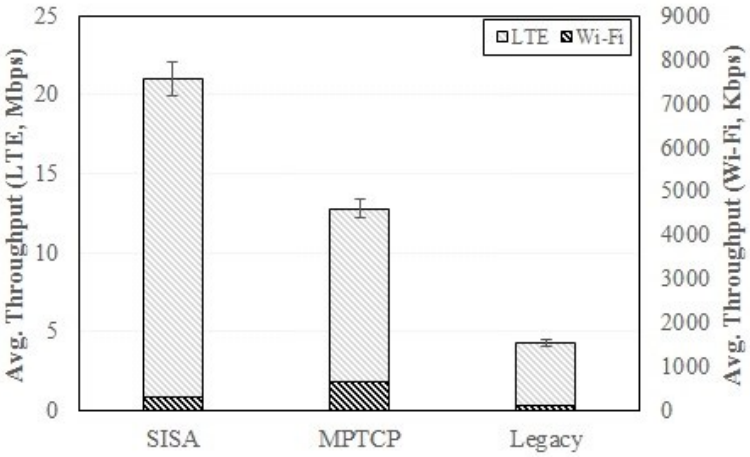


Figure 3.4: **Throughput Comparison.** The legacy smartphone and smartphone with SISA platform download the file over public Wi-Fi or LTE. The MPTCP smartphone downloads the file over public Wi-Fi and LTE.

3.4. Summary

In this chapter, we deal with the improvement of user experience in public places. We introduce a novel network interface selection algorithm named SISA which exploits multiple network interfaces efficiently based on the predicted quality of Wi-Fi. We assume that Wi-Fi quality is vary over time in various environments for different context. Since our decision tree boundary is determined by real-field measurement, significant gains can be achieved by the implementation of our platform. We also showed and proved that it is illogical and stereotypical to predict stability of Wi-Fi using signal strength and transmission rate of link by analyzing correlation between network features. As shown in Section 3.3, our scheme is efficient in public places but false positive of decision can be happen. The results shows that our scheme outperforms both SPTCP with normal network usage policy and generic MPTCP in real-field environments.

Chapter 4

Machine Learning-based Path Management for Mobile Devices over MPTCP

4.1. Introduction

Recent mobile devices are equipped with multiple network interfaces such as Wi-Fi and LTE. 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, wireless devices abound and quality of active paths changes frequently in real mobile environments. It makes Wi-Fi connection affect the MPTCP performance negatively.

We design a novel path management called MPTCP-ML which controls path usage based on machine learning mechanism. We use various path quality metrics as inputs for machine learning model. For accurate capturing of path quality, we utilize different quality metrics including signal strength, data rate, TCP throughput, the number of APs on the same and adjacent channel, and RTT (Round Trip Time).

MPTCP-ML samples the features of path quality periodically and detects status of active paths in real-time utilizing pre-built random forests model which relies on insight based on historical patterns discoverable in collection

data. When it detects a path which experiences poor performance, MPTCP-ML suspending use of the path, not keeping it.

We have implemented MPTCP-ML in Android and conducted experiments for various and dynamic mobile environments. The environments include the places where network traffic jams, signal interference are commonplace, and signal strength of the connected access point is weak. And we also consider user mobility environment. The results show that MPTCP-ML outperforms generic MPTCP with accurate detection of path quality in mobile environments.

4.2. Path Management Algorithm over MPTCP

4.2.1 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 [22], an extension of decision tree that constructs multitude of random decision trees and amalgamate them together, fits the requirements of our approach among the diverse techniques.

- It can support high classification accuracy and stable prediction.

- It eases off overfitting to its training data set by building a forests composed of decision trees at training time.

- It is a fast and light-weight method like decision tree, thus making it possible to embed our scheme in commodity devices.

4.2.2 Tree Learning and Modeling

There are a lot of factors influencing path quality of Wi-Fi that includes signal strength from associated Wi-Fi AP, signal interferences from nearby Wi-Fi APs on same and adjacent channel, and network congestion.

To reflect these features, we choose channel status, network performance and interference degree of Wi-Fi network as classification features. The channel status is comprised of RSSI and data rate. RSSI is the signal strength from 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 -80dBm that is the minimum value for basic connectivity.

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 classification of path quality. 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.

TP (True Positive) rate, FP (False Positive) rate, Precision, Recall, F-Measure, and OOB (Out-of-Bag) error rate are performance metrics for classification. TP rate is the rate that the predicted result is positive when real condition is positive (i.e. the rate of correct identification). FP rate is the rate that the predicted result is negative when real condition is positive (i.e. the rate of incorrect identification). Precision is the ratio of the number of correct positive results to the number of all positive results, and recall is the ratio of

the number of correct positive results to the number of positive results that should have been returned. F-measure represents the classifier accuracy considering both precision and recall. Its value reaches its best value at 1 and worst at 0. OOB error rate is the mean prediction error rate of classifier on each training data at each bootstrap iteration, using only the trees that did not have the data. For some metrics, lower values indicate better performance. For others, higher values are better.

Table 4.1 shows classification performances of the two models. As shown in Table 4.1, both models support high accuracy and low OOB error rate. The F-measure and OOB error rate of RF-2C model are 0.98 and 4%, respectively. And both values of RF-4C model are 0.93 and 7%, respectively. The results demonstrates both models perform prediction task with high accuracy and low prediction error rate. 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 little bit more suitable for our work.

Table 4.1: **Classification Performance**

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

4.2.3. System Design

The aim of our algorithm is to efficiently control Wi-Fi path usage based on machine learning mechanism. It consists of three components: Interference Scanner, Network 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 which can be available to connect and extracts information about channel number and signal strength of each AP. And then it computes the degree of interference by calculating 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 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.

Network 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 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 in 100 ms, it records the value as empty when the delay between two endpoints becomes longer than the sampling interval.

MPTCP-ML Core processes path quality prediction and handling Wi-Fi path. It predicts the quality of Wi-Fi path based on pre-built random forests model. The model takes as input the all data set obtained from the sampler and outputs path quality of Wi-Fi path that are predicted. MPTCP-ML 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 continuously check the path quality of the associated Wi-Fi AP.

4.3. Performance Evaluation

4.3.1 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 networks for experiment. To fairly compare MPTCP-ML to generic MPTCP, we conducted the experiments using the two smartphones at the same time. As mentioned before, we concentrate on mobile environments where the quality of wireless paths change frequently. We first demonstrate the better performance of MPTCP-ML under static environments and then conducts experiments in mobile environments.

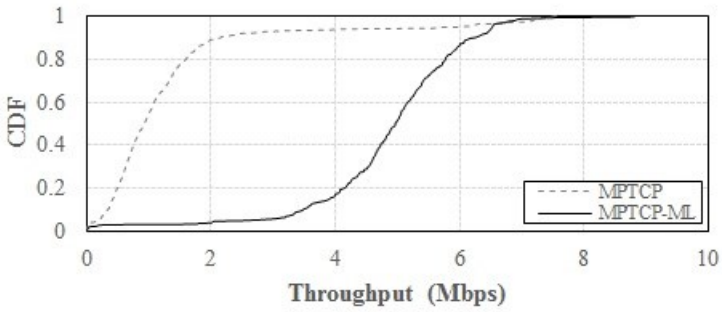
4.3.2 Performance under Static Environment

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 of associated AP, signal interference, and network congestion.

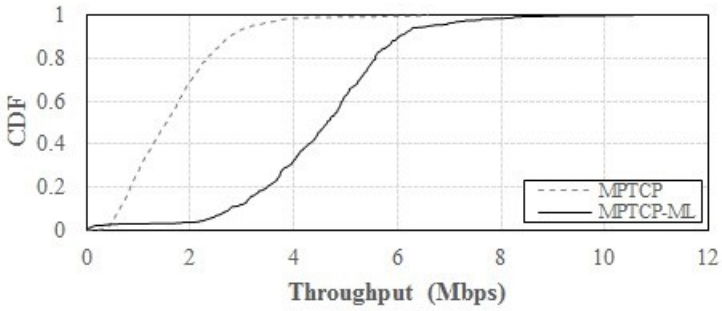
Figure 4.1 presents the cumulative distribution of throughput for the different scenarios where the two smartphones exploit both LTE and Wi-Fi network and the surroundings have a negative influence on the Wi-Fi path. We observe that MPTCP-ML achieves higher aggregate TCP throughput than MPTCP in the scenarios.

As shown in Figure 4.1 (a) and (b), there exists a significant difference in performance between MPTCP and MPTCP-ML. It is related with the signal quality influenced by distance and density of wireless devices and access points. Under the scenarios, MPTCP-ML outperforms MPTCP since it considers signal information. 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 signal interference). The results can be varied by experiment settings.

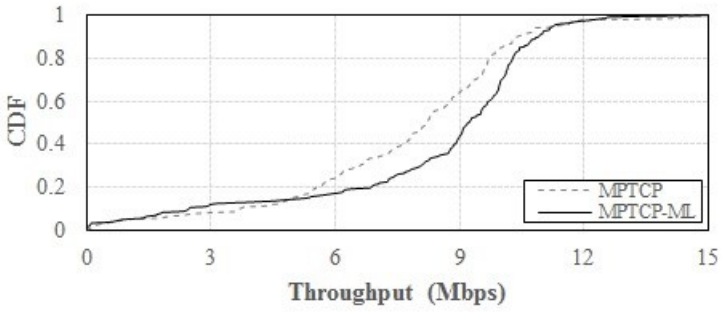
Figure 4.1 (c) presents congestion has less impact on the aggregate throughput. Owing to the default packet scheduler of MPTCP at a sender side, the amount of packets can be adjusted according to end-to-end RTT of each active path, which brings advantages for MPTCP.



(a) Weak signal strength



(b) Interference



(c) Congestion

Figure 4.1: **MPTCP Performance Comparisons.** (a) Weak signal strength from connected Wi-Fi AP, (b) Signal Interference, and (c) Network congestion.

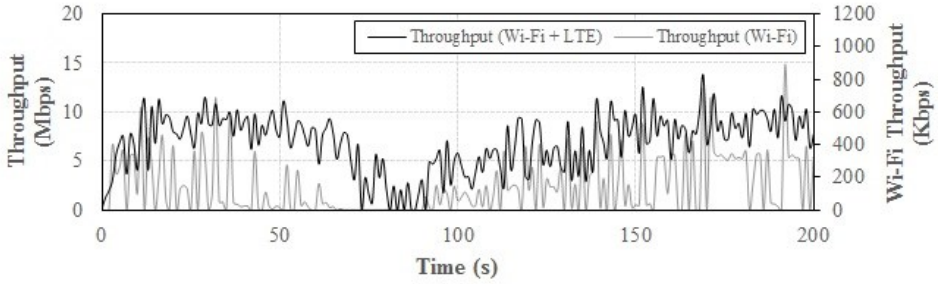
4.3.3 Performance under Mobile Environment

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 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 4.2.

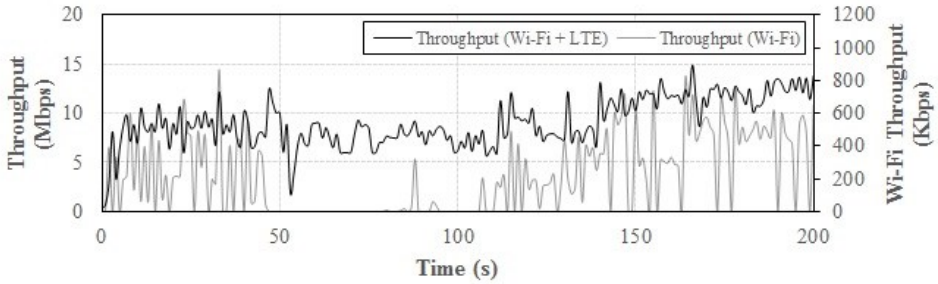
Figure 4.2 (a) and (b) shows a sample trace for each MPTCP expressed as a time series where the two MPTCPs are using LTE with Wi-Fi. The figure presents the aggregate throughput as a function of time when each MPTCP uses both LTE and Wi-Fi, in black, on the left Y-axis. The figure also shows the Wi-Fi 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.

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.

From this experiment, we identify how effectively MPTCP-ML works with user mobility.



(a) MPTCP



(b) MPTCP-ML

Figure 4.2: **MPTCP and Wi-Fi throughput over time when using Wi-Fi and LTE.**

4.4. Summary

In this chapter, we deal with the improvement of MPTCP performance in mobile environments. We design a path management scheme that relies on machine learning to efficiently handle active paths 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 mobile environments. We have demonstrated that MPTCP-ML achieves significantly better performance than generic MPTCP with high prediction accuracy.

Chapter 5

Conclusion

5.1. Research Contributions

In this thesis, we addressed performance degradation problem of mobile device such as smartphone and tablet in public and mobile environments. First, we applied machine learning technique to computer networking and reflected various path quality features which can be obtained from wireless networks and mobile devices in order to get an acceptable performance of identifying path quality. Second, we designed two novel connection management schemes for SPTCP and MPTCP which can be aware of not only various network quality features not also user context and contents. Lastly, we proved that unlike general thoughts, particularly in complex real environment, there is not any correlation between network features such as signal strength, link speed and TCP throughput.

To summarize, SPTCP and MPTCP has opened new possibilities to manage the use of wireless radio interfaces and boost the achievable capacity to uses. Although there still remain some issues, it can appears to real network policy and solution. Beside the two solutions of this thesis, there remain many interesting problems to require further research. This thesis could help as the guideline to improve TCP performance and user experience.

5.2. Future Research Directions

There are a lot of remaining issues to efficiently handle the network connections and paths. We conducted data collection and measurements in real-field and show that device with our scheme outperforms legacy devices. However, our schemes can be improved by designing more sophisticated algorithm that takes apposite decision regarding the switching between network interfaces and finding a proper boundary between energy efficiency and QoE. While our schemes is based on real-field data, more adaptive path management scheme is needed to cover wide and complex real environments. In this research area, energy is also interesting topic. For example, efficient method which can reduce energy consumption used to obtain the diverse path quality metrics.

Bibliography

[1] WiGLE: Wireless Network Mapping. https://wigle.net/. Accessed: 2016-09-23.

[2] Seoul Bus Statistics. http://www.socialindex.seoul.go.kr.

[3] Seoul Subway Statistics. http://www.seoulmetro.co.kr.

[4] Android Open Source Project. https://source.android.com/

[5] MultiPath TCP: Linux Kernel Implementation http://multipath-tcp.org/

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

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

[8] 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.

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

[10] 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.

[11] G. Katz, A. Shabtai, L. Rokach, and N. Ofek. ConfDTree: Improving decision trees using confidence intervals. *IEEE 12th International Conference on Data Mining*, 2012.

[12] L. Fan. Accurate Robust and Efficient Error Estimation for Decision Trees. *The 33rd International Conference on Machine Learning*, 2016.

[13] 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 ACM conference on Internet Measurement Conference*, 2013

[14] 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

[15] M. Kheirkhah, I. Wakeman, and G. Parisis. MMPTCP: A Multipath Transport Protocol for Data Centers. *Proceedings of the IEEE INFOCOM 2016 Conference*, 2016.

[16] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving datacenter performance and robustness with multipath TCP. *ACM SIGCOMM Computer Communication Review*, 2011.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

요 약

오늘날, 스마트폰 이용자의 수가 증가함에 따라 사용자의 편의를 위해, 공공장소와 대중교통에 공용 Wi-Fi AP들이 점점 더 많이 설치되고 있다. 그러나, 모바일 디바이스의 Wi-Fi와 LTE 네트워크 인터페이스가 모두 켜진 상태에서, 모바일 디바이스는 통신하기 위해 연결 품질과 안정성에 상관없이 Wi-Fi 인터페이스를 사용한다. 이러한 네트워크 인터페이스 사용 방식은 사용자의 체감품질(QoE)를 감소시킨다.

먼저 사용자가 스마트폰의 무선 네트워크 인터페이스를 효율적으로 사용하도록 하기 위해, 우리는 사용자의 상황정보, 핸드폰 이용정보, 네트워크 성능 지표 (신호 세기, 링크 속도, TCP throughput, Gateway까지의 RTT)를 활용한 Decision Tree 기반의 네트워크 연결 관리 알고리즘을 제안하였다. 우리는 간헐적인 연결 혹은 낮은 성능을 제공하는 Wi-Fi AP와의 지속적인 연결이 TCP throughput에 어떤 영향을 주는지 실험하였다. 실제 환경에서의 광범위한 측정 및 실험을 통해, 우리는 우리가 새롭게 개발한 플랫폼을 탑재한 스마트폰이 다양한 환경에서 시중에 상품화된 스마트폰보다 높은 성능을 보임을 확인하였다. 또한, 네트워크 성능을 나타내는 지표들이 실제 복잡한 환경에서 이들간의 상호 연관성이 없음을 증명하였다.

다음으로, 우리는 낮은 성능을 제공하는 Wi-Fi AP로 인한 성능 저하 문제를 다중경로 전송 프로토콜(MPTCP)에 적용 및 확장하였다. 다중경로 전송 프로토콜은 스마트폰에서 Wi-Fi와 LTE 인터페이스를 동시에 사용 가능한 기법이다. 실제 환경에서는 무선 통신 장비들이 혼재되어 있고, 네트워크의 품질이 자주 변화하기 때문에 Wi-Fi AP와의 간헐적인 연결과 낮은 성능을 제공하는 Wi-Fi AP와의 지속적인 연결이 다중경로 전송 프로토콜의 성능에 부정적인 영향을 주는 것을 확인하였다. 이 문제를 해결하기 위해, 우리는 새로운 머신 러닝 기반의 경로 관리 기법(MPTCP-ML)을 개발하였다. 이 기법은 다중경로 전송 프로토콜이 사용하는 여러 개의 경로를 머신 러닝 모델을 통해 각 경로의 사용 유무를 관리하는 기법이다. 네트워크 품질을 나타내는

다양한 지표들을 머신 러닝에 사용하였다. 스마트폰에서 사용하는 네트워크 경로의 성능을 정확히 파악하기 위해, 다양한 정보들(신호 세기, 전송 속도, TCP throughput, 신호 간섭을 발생시키는 Wi-Fi AP의 수, 종단간 RTT, Gateway까지의 RTT)을 활용하였다. 우리는 이 기법을 안드로이드 기반 스마트폰에 탑재시키고 다양한 환경에서 성능 측정 실험을 수행하였다. 실험 결과, 낮은 성능을 제공하는 Wi-Fi AP에 연결될 경우, 우리가 새롭게 개발한 기법을 탑재한 스마트폰이 기존의 다중경로 통신 프로토콜을 탑재한 스마트폰과 비교해 이를 빠르게 이를 인지하고 해당 경로 상에 수립된 연결을 종료하여 스마트폰의 통신 성능 저하 문제를 해결할 수 있음을 확인하였다.

주요어 : 공용 와이파이, 다중경로 전송 프로토콜, 경로 관리, 네트워크 인터페이스, 스마트폰, 체감품질

학번 : 2015-21263