2016 IEEE 41st Conference on Local Computer Networks Workshops

An Enhancement of Multipath TCP Performance in Lossy Wireless Networks

Kien Nguyen*†, Gabriel Porto Villardi*, Mirza Golam Kibria*, Kentaro Ishizu*†, Fumihide Kojima*, Hiroyuki Shinbo†
* National Institute of Information and Communications Technology
† Advanced Telecommunications Research Institute International
Email: {kienng, gpvillardi, mirza.kibria, ishidu, f-kojima}@nict.go.jp; shinbo@atr.jp

***Abstract*—Multipath TCP (MPTCP) is an evolvable technology for bandwidth aggregation on a mobile device. MPTCP naturally and concurrently exploits wireless links via different interfaces (i.e., Wi-Fi and cellular) for data transferring. Theoretically, the MPTCP's aggregated throughput is better or at least equals to the TCP throughput over a link. However, our investigation of MPTCP performance in a lossy wireless environment shows that the theoretical statement does not always hold. Specifically, the aggregation benefit of MPTCP (i.e., in a comparison to TCP) becomes negative when the loss over a link surpasses specific levels. In this paper, we focus on finding the negative region of MPTCP's aggregation benefit by experimental study. Additionally, we propose a loss-aware disabling mechanism to bypass the negativity. The mechanism temporarily switches a wireless link to a backup mode when an observed loss value reaches a threshold. We integrate the mechanism with a state-of-the-art MPTCP implementation and compare the two MPTCPs in various lossy conditions. The evaluation results confirm the effectiveness of the newly proposed mechanism in terms of enhancing goodput and saving resources.**

Keywords: *MPTCP, loss awareness, experimental study*

I. Introduction

In recent years, along with the popularity of mobile devices and the ubiquitous deployment of cellular networks, the demand on Internet access by mobile users has increased dramatically. Accordingly, there are many research and development activities aiming to improving achievable throughput on mobile devices. Since the device commonly uses two wireless interfaces (e.g., Wi-Fi and cellular) for the Internet access, an intuitive approach of throughput improvement is concurrently transferring data over the interfaces. One of the most successful and widely-adopted solutions is the multipath TCP (MPTCP) technology, which exploits the concurrency in the transport layer of networking stack. MPTCP has been recently standardized by IETF [1] with an increasing number of new MPTCPs (e.g., *coupled* [2], *olia* [3], *wvegas* [4], *balia* [5]). Besides, various mobile platforms including Android [6] and Apple iOS [7] are capable of MPTCP.

The popularity and readiness of MPTCP originate from its evolving capability that requires zero modification on the existing networks and applications. In a normal condition, MPTCP on a device separates application traffic into different flows, each of which traverses over a wireless link. The diverse flows are consequently aggregated by MPTCP on another end of communication (i.e., a server). By doing so, MPTCP theoretically improves the application's throughput, availability, and resilience in a comparison to TCP. However, in a lossy condition that is popular in wireless networks, MPTCP potentially has a negative impact. That is because MPTCP generalizes all types of loss which normally require a period of several round trip times to be recognized by the device. Therefore, it is worthy to investigate and enhance (if necessary) MPTCP performance in the such condition.

Addressing the issue, we first verify the important criteria in MPTCP's design, which is the MPTCP's goodput is higher or at least as similar as the TCP's one over a single link, by experimental study. We set up a scenario where a mobile devices uses Wi-Fi and cellular (i.e., LTE) interfaces for communication with an MPTCP-capable server. Under various wireless loss conditions, the value of a so-called *aggregation benefit* function is extensively evaluated. The results show that MPTCP degrades its aggregation benefit when the wireless loss increases. The benefit even becomes negative when a link's loss reaches a threshold value. In order to bypass that negativity, we propose a loss-aware disabling mechanism which switches a wireless link to a standby mode when it experiences the threshold loss. We implement and integrate the mechanism to the MPTCP version 0.90. We then compare the performances of two MPTCPs in the lossy environments. The results show that the newly proposed mechanism enhances the goodput, as well as, saves the resources.

There are several works targeting to investigate MPTCP performance on a mobile device with cellular and Wi-Fi interfaces. In [8], the authors shows the benefit and capability of MPTCP in achieving handover between 3G/Wi-Fi links. The work in [9] proposes a solution for saving energy consumption that temporarily mutes a wireless links and shifts traffic to a more energy efficient path. In [10], a disabling mechanism has been introduced for a MPTCP-capable mobile device with two Wi-Fi links. The mechanism is triggered following observed RSSI values on the device. Different to previous works, we aim to verify the beneficial working region of MPTCP in a lossy wireless network. Moreover, the proposed mechanism, which relies on the observed loss information on devices, is adopted to enhance MPTCP.

The remainder of this paper is organized as follows. Section II includes the background information. Section III introduces our evaluation methodology and the mechanism. Section IV,

DOI 10.1109/LCNW.2016.36

Section V presents results and related work, respectively. Finally, Section VI concludes the paper.

II. Background

A. Wi-Fi and Cellular networks

Nowadays, Wi-Fi has been equipped on many devices ranging from electric appliances to personal gadgets. On mobile devices, Wi-Fi is normally selected as a default Internet connection. Besides that, there are continuous Wi-Fi related technologies standardized by IEEE (802 a/b/g/n/ac). The achievable throughput is increased accordingly (from 11 Mbps up to 300 Mbps). A Wi-Fi link generally has high throughput with short delay, but it experiences high loss. On the other hand, data transmission over cellular connection tremendously grows due to an emerging population of smart phones. It leads to the upgrades of cellular operators' infrastructures. We have seen the transformation from the different generation of cellular network in recent years (e.g., third generation (3G) to the fourth generation (4G/LTE)). In a 4G network, the theoretical peak speed for services is about 100 Mbps in high mobility environments, and 1 Gbps in low mobility environments. However, the real measurement shows the cellular throughput is as similar as the Wi-Fi's one [11]. The difference between the two technologies is that the cellular has a broader signal coverage than Wi-Fi. The difference is represented by different values of latency in our evaluation.

B. Multipath TCP

The scopes of Wi-Fi and LTE technologies are about MAC/PHY layers of a link between a device and a base station. In fact, there are the MAC/PHY related technologies for bandwidth aggregation. However, that is not enough since an application performance depends on many end-to-end parameters. It hence motivates higher-layer approaches for aggregation, such as MPTCP. Through a default link, MPTCP initializes a connection in a similar method as TCP does (using SYN, SYN/ACK, ACK). However, it has an MP_CAPABLE option in the SYN packet. The packet contains several additional flags that are for the usage of checksum and cryptographic. It contains an authentication key to authenticate a later added subflows to the MPTCP connection. If the SYN receiver is MPTCP capable, it sends a MP_CAPABLE signal (i.e., with authentication key and flags). After the signal is successfully received, the initialization is completed. If the SYN receiver is not MPTCP capable, the following progress is as same as in TCP. When the MPTCP-capable device has an extra interface, MPTCP will send a new SYN with a JOIN option. If the flow establishment is successful, a new TCP flow is created for data transmission. Consequently, the device concurrently sends data through two links. MPTCP supports several working modes which defines a subset of or whole available paths between a two ends of MPTCP communication are used. However, the configuration is necessarily activated before the MPTCP initialization.

C. MPTCP balia

MPTCP balia is the state-of-the-art MPTCP using the scheduler named balia [5]. The *balia* algorithm is carefully designed by using a mathematical fluid model. It has been proven to be more effective than the other MPTCP schedulers. More importantly, the *balia* algorithm has been implemented in the MPTCP kernel on Linux. Therefore, we choose MPTCP balia as the baseline in this work. To briefly present the scheduler's behavior, we denote by w_i, rtt_i, and w the size of congestion window size, round trip time of subflow i, and the total congestion window size over all the subflows, respectively. Each *balia* subflow behaves as follows

- For each ACK on flow i:
$$w_i \leftarrow w_i + \frac{w_i/rtt_i}{\sum_{p\in R}(w_p/rtt_p)^2} * \frac{1+\alpha_r}{2} * \frac{4+\alpha_r}{5}$$
in which $\alpha_r = \frac{max \frac{w_r}{rtt_r}}{w_r} * rtt_r$
- For each loss on flow i:
$$w_i \leftarrow \frac{w_i}{2} * min\{\alpha_r, 1.5\}$$

III. Aggregation Benefit and Disabling Mechanism

A. Aggregation benefit

The efficiency of multipath communication could be verified via an aggregation benefit function as proposed in [12]. In the case of MPTCP, we denote $Ben(M)$ as its aggregation benefit function, which is defined as follow

$$Ben(M) = \begin{cases} \frac{G-C_{max}}{\sum\limits_{i=1}^{n} C_i - C_{max}}, & \text{if } G \geq C_{max} \\ \frac{G-C_{max}}{C_{max}}, & \text{if } G < C_{max} \end{cases} \tag{1}$$

where M be a multipath scenario, with n paths, and C_i is the capacity (i.e., TCP goodput) of the path i , C_{max} is the highest capacity among all paths, G is the measured value of MPTCP goodput.

We can observe that the values of $Ben(M)$ are in the range $[-1, 1]$ that makes measurements in different scenarios comparable. If $Ben(M) \geq 0$ or $G \geq C_{max}$, the mentioned criteria holds. Specifically, MPTCP has a better performance (i.e., positive benefit) than TCP over a single path. However, if $Ben(M)$ is negative, the device should be aware of the condition. When MPTCP has the negative benefit values, the device should transmit data over a path instead (e.g., by temporarily disabling an interface).

B. Wireless loss-aware disabling mechanism

As a pure transport layer protocol, MPTCP generalizes all types of loss which are generated by many reasons in wireless networks (e.g., fading). The wireless loss information could be observed by the mobile device from either probing network or management information [13]. In this work, we assume that the device is able to recognize the loss of wireless link. If the device is aware of a loss level of a wireless link, which will degrade the MPTCP performance, the device temporarily switch the lossy link to a standby mode. The benefits of switching include not only the MPTCP enhancement but also resource efficiency (e.g., saving energy).

Our implementation of the loss-aware switching mechanism on Linux system is as follows. Assuming the device knows the loss threshold via an interface (i.e., denoted as γ), which will be harmful for MPTCP. We add two new variables (i.e., *mptcp_loss_cnt* and *mptcp_send_cnt*) into the system's *struct tcp_sock*. As the names indicate, the former variable is for counting the number of MPTCP loss packets, and the latter is for MPTCP sent packet, respectively. The variables are used for calculating and comparing the loss over a link, but we only consider the MPTCP packets. Specifically, when a packet from *mptcp socket* is dropped, the number of loss count (*mptcp_loss_cnt*) is incremented. The similar process is happened with *mptcp_send_cnt*. A switching action will be activated when $\frac{mptcp\_loss\_cnt}{mptcp\_send\_cnt} \geq \gamma$. We add a new flag in *struct tcp_sock*, which is used to toggle the appropriate interface to the MPTCP backup mode. In order to let the default MPTCP compatible with our implementation, we add a new mode named *mptcp_threshold* for MPTCP. The configuration of new mode requires an input parameter of loss threshold for each interface. The threshold could be determined by the experimental study as in the following section. In the case there is only one active interface, the mechanism has no effect.

IV. Evaluation

A. Setup

We setup an indoor testbed to investigate the MPTCP performance and the proposed mechanism. The testbed emulates a heterogeneous network with Wi-Fi and LTE as shown in Fig. 1. The mobile device has two wireless interfaces, each of which is appropriately routed to reach an application server. In the testbed, we choose the baseline delay parameters of two wireless link as suggested from the previous works [11].

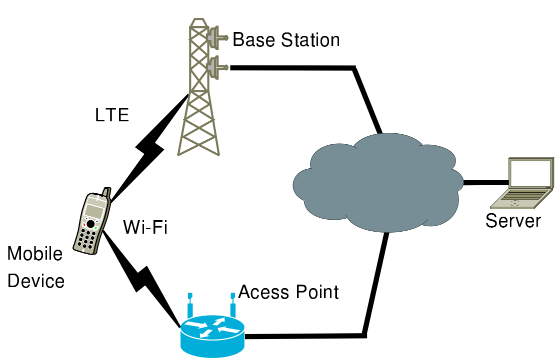


Fig. 1: Connection diagram

TABLE I: Link parameters

	Delay (ms)	Jitter (ms)	Bandwidth (Mbps)
Wi-Fi	10	[0, 5]	(10, 20, 30, 40, 50)
LTE	75	[0, 35]	(10, 20, 30, 40, 50)

Additionally, the jitter value of each path in one run is randomly selected in a range to make the delays dynamic. Those parameters and the set of bandwidth values are shown in Table I. The mobile device and the server use Linux kernel version 3.18.20 with the MPTCP version 0.90.0 [14]. The MPTCP *balia* is selected in the evaluations. To form the network, the device and the server are connected with other Linux machine on which the path settings are configured by the tool named *tc*. To emulate lossy links, we add a new component on the device's system (i.e., Linux). Each interface has a new configuration file under the Linux's *proc* that contains a value of loss probability. When an interface is configured with a loss value, it drops packet following the appropriate probability (i.e., in */net/ipv4/ip_output.c*). To reduce interferences caused by the environment, we disable route caching, buffer autotuning on the Linux machines. In each run, iperf3 generates the TCP flows between the device and server.

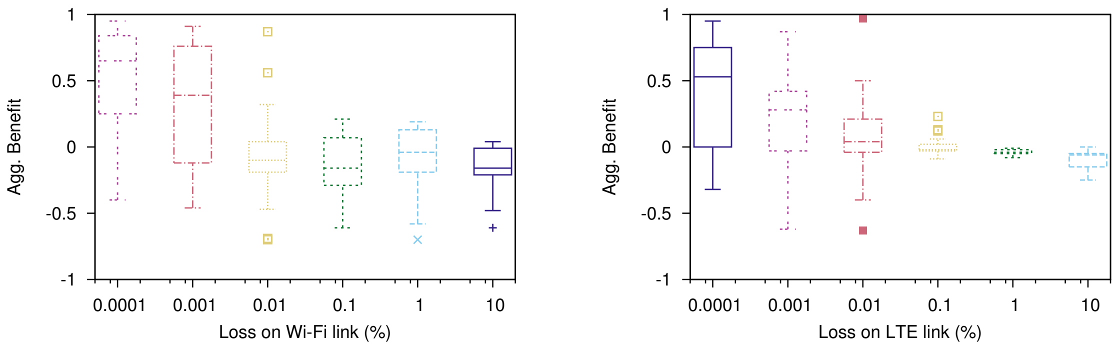
B. Investigate aggregation benefit

As in [15], we also adopt the experimental design, which is useful in protocol design analysis in our investigation. However, different to [15], our aim is to make a decision following the investigated parameter (i.e., the value of aggregation benefit function). Since the loss parameter of wireless link is introduced, it is necessary to determine the evaluation time in order to guarantee the wireless link incurring such the loss level. We need to specify a large enough number of sending packets. The MTU of iperf3 flow is 1500 bytes, hence the resulting evaluation time with each bandwidth values is calculated and shown in Table II. Note that, the real calculation time values for loss (10%, 1%, 0.1%) are much smaller than the values in the table. However, we use 120 seconds for the convenience in our evaluation setup.

We start the evaluation by benchmarking the capability of each link with TCP traffic in both normal and loss conditions. We experiment each set of parameters ten times. Then the maximum values of TCP goodput are recorded. Those benchmarking values are filled in the aggregation function (i.e., C_i, C_{max}). In the MPTCP evaluation, we vary the loss and bandwidth parameters on the Wi-Fi while the LTE link has no loss, and vice versa. All the possible combinations in those cases have been evaluated. After each run, the measured goodput of MPTCP is collected and the value of aggregation benefit is calculated. The resulting values are plotted in Fig. 2. In the figure, Fig. 2a and Fig. 2b show the values of aggregation benefit function while the Wi-Fi link and the LTE

TABLE II: Evaluation time (seconds)

	Link Bandwidth (Mbps)				
Loss(%)	10	20	30	40	50
10	120	120	120	120	120
1	120	120	120	120	120
0.1	120	120	120	120	120
0.01	120	120	120	120	120
0.001	1200	6000	400	300	240
0.0001	12000	6000	4000	3000	2400



(a) Aggregation benifit in the case of lossy Wi-Fi link

(b) Aggregation benifit in the case of lossy LTE link

Fig. 2: Investigate the aggregation benefit of MPTCP in lossy wireless environment

link experience loss, respectively. In each sub-figure, the x-axis shows the log scale of loss values and the y-axis presents the values of benefit function. The boxplots show the average, minimum, maximum values and the distribution of values. We also plot the outliers in the figures.

Except the outliers, we observe that the aggregation benefit of MPTCP decreases when the loss over a wireless link increases. Figure 2a shows when the loss of Wi-Fi link equals to or larger than 0.01 %, the benefit function tends to be zero or negative. In the such scenarios, even though MPTCP shifts the traffic to the less lossy link (i.e., LTE), however the MPTCP goodput is worse than the TCP goodput over the LTE link. Therefore, it is worthy to apply the disabling mechanism the Wi-Fi link when the loss reaches a threshold of 0.1%. In the case of LTE link, the same trend could be observed, however the loss threshold is 1% as shown in Fig. 2b. That is because the latency on LTE link is larger than the one via the Wi-Fi link.

C. *MPTCP with the loss-aware mechanism*

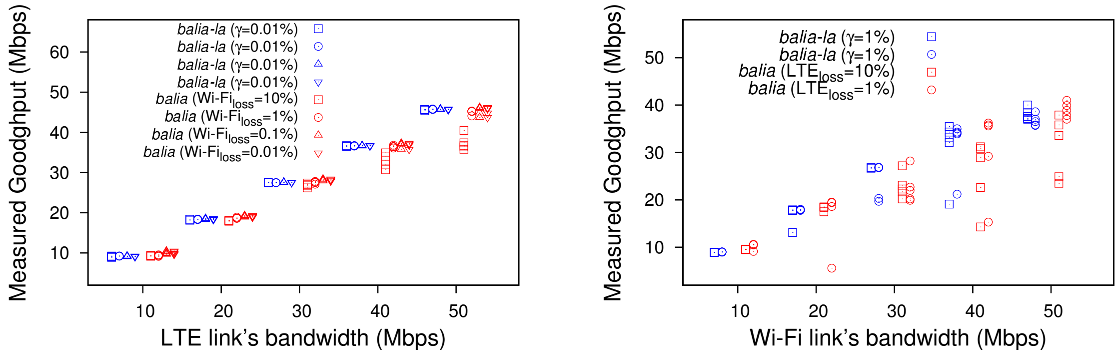
Inheriting the threshold values from the previous section, we configure the new mode (i.e., *mptcp_threshold*) for MPTCP on the device. We then repeat all the experiments for comparison. The new MPTCP, which is additionally equipped the proposed loss-aware mechanism, has the threshold $\gamma = 0.01\%$, $\gamma = 1\%$ for Wi-Fi and LTE, respectively. The comparison of measured goodput of the two *balia* implementations is shown in Fig. 3. In the figures, balia is for the default and *balia-la* is for the *balia* with loss-awareness.

Figure 3a shows the scenario of lossy Wi-Fi link. In the figure, the same shape of point shows the same link conditions, while the *balia*'s results and the *balia-la*'s ones are in red and blue, respectively. We can see that the Wi-Fi link causes dropped packets, the MPTCP traffic is seemingly conveyed only via the LTE link. When the bandwidth of Wi-Fi link is small, the two *balia*(s) have comparable performances. Even in that case, the disabling mechanism could save the energy consumption and radio resource of the LTE interface (i.e., from a switching moment to an end of experiment). However, when the bandwidth value of Wi-Fi link is larger than 20 Mbps, it is clear that the *balia* goodput is much lower than the *balia-la*'s. It is because there are sending packets over the lossy link which confuse *balia* and cause retransmissions. In the such case, the effectiveness of the proposed mechanism is even more significant. It not only saves the resources but also enhances the goodput. We can draw the same conclusion when the LTE link experiences loss over the threshold as in Fig. 3b.

V. Related Works

MPTCP has been actively researched in various networks from existing datacenters [16] to future software defined wireless networks [17], etc. Besides that, there are ongoing efforts in spreading MPTCP to popular platforms not only Android, iOS but also Linux [14], FreeBSD [18], etc. An important use case of MPTCP is the MPTCP-capable mobile device that has multiple wireless interfaces (e.g., Wi-Fi/3G [8], [9], Wi-Fi/Wi-Fi [10], [19]). MPTCP enables concurrent transmissions over wireless links for the throughput and resilience improvements. Due to the dynamic nature of mobile wireless, the MPTCP's efficiency on mobile devices needs to be carefully investigated and optimized. Accordingly, there are many related researches in recent years.

In [8], the authors shows the capability of soft-switchover between Wi-Fi/3G using MPTCP. The same positive conclusion is drawn from [9], where the authors focus on the capability of handing mobility in MPTCP. In [20], the authors study the MPTCP performance in the wild, with real wireless settings and background traffic. The results show that similar to the theory MPTCP's throughput is always better than the best single-path TCP. In [11], a similar measurement research has been conducted, it however shows different conclusions. The results indicate that MPTCP is worse than the single-path TCP in several scenarios, especially the ones with loss. In [15], the authors show the benefit of applying experimental methods to find the misbehaviors of MPTCP designs and implementations. Similar to [15], we also adopt experimental study to find



(a) Througput comparison of lossy Wi-Fi link

(b) Througput comparison of lossy LTE link

Fig. 3: Comparing the loss-aware balia and balia

the behaviours of MPTCP, however we focus on lossy wireless conditions. Another difference in our work is that we aim to determine the working region where the MPTCP performance is worse than the TCP's one. We then propose the threshold-based switching mechanism for temporarily disabling the link that causes the performance degradation.

VI. Conclusion

MPTCP is an effective tool for bandwidth aggregation over diverse wireless links of mobile devices. Due to the aggregation benefit, the achievable goodput of MPTCP is theoretically better than the one of TCP over one link. However, the high loss on the wireless link potentially degrades the MPTCP performance and eliminates the benefit. In this paper, we have conducted an extensive evaluation of MPTCP performance on a lossy heterogeneous wireless environment with Wi-Fi and LTE links. The results show that the default MPTCP *balia* has lower values of measured goodput than TCP when a wireless link incurs high loss levels. To bypass that issue, we propose a loss-aware disabling mechanism that temporarily switch a wireless interface to a backup mode. The mechanism is toggled when the observed loss value exceeds a threshold. We implement and integrate the mechanism with the MPTCP *balia*. The evaluation results show that the mechanism significantly enhances MPTCP goodput as well as saves resources in the lossy environment.

Acknowledgement

This research was conducted under a contract of R&D for radio resource enhancement, organized by the Ministry of Internal Affairs and Communications, Japan.

References

[1] Alan Ford, Costin Raiciu, Mark Handley, and Olivier Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. Technical report, RFC 6824, 2013.

[2] Costin R., Christoph P., Sebastien B., Alan F., Michio H., Fabien D., Olivier B., and Mark H. How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP. In *Proc. USENIX NSDI*, 2012.

[3] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec. MPTCP Is Not Pareto-Optimal: Performance Issues and a Possible Solution. *IEEE/ACM ToN*, 21(5):1651–1665, 2013.

[4] Yu Cao, Mingwei Xu, and Xiaoming Fu. Delay-based Congestion Control for Multipath TCP. In *Proc. 20th IEEE ICNP*, pages 1–10.

[5] Q. Peng, A. Walid, J. Hwang, and S.H. Low. Multipath tcp: Analysis, design, and implementation. *IEEE/ACM ToN*, PP(99), 2015.

[6] MPTCP kernel for Android. http://multipath-tcp.org/pmwiki.php/Users/Android, 2015.

[7] iOS: Multipath TCP Support in iOS 7. https://support.apple.com/en-us/HT201373, 2015.

[8] Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, and Olivier Bonaventure. Exploring Mobile/WiFi Handover with Multipath TCP. In *Proc. ACM SIGCOMM CellNet*, pages 31–36, 2012.

[9] Costin Raiciu, Dragos Niculescu, Marcelo Bagnulo, and Mark James Handley. Opportunistic Mobility with Multipath TCP. In *Proc. 6th ACM MobiArch*, pages 7–12, 2011.

[10] 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. In *Proc. IEEE INFOCOM*, pages 1815–1823, 2014.

[11] Shuo Deng, Ravi Netravali, Anirudh Sivaraman, and Hari Balakrishnan. WiFi, LTE, or Both?: Measuring Multi-Homed Wireless Internet Performance. In *Proc. ACM IMC '14*, pages 181–194, 2014.

[12] D. Kaspa. Multipath Aggregation of Heterogeneous Networks. PhD thesis, University of Oslo, 2011.

[13] K. Shin, J. Kim, and S. B. Choi. Loss Recovery Scheme for TCP Using MAC MIB over Wireless Access Networks. *IEEE Communications Letters*, 15(10):1059–1061, 2011.

[14] S. Barre C. Paasch et al. Multipath TCP in the Linux Kernel, available from. http://www.multipath-tcp.org.

[15] Christoph Paasch, Ramin Khalili, and Olivier Bonaventure. On the Benefits of Applying Experimental Design to Improve Multipath TCP. In *Proc. ACM CoNEXT '13*, pages 393–398, 2013.

[16] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, and Mark Handley. Improving Datacenter Performance and Robustness with Multipath TCP. In *Proc. ACM SIGCOMM*, pages 266–277, 2011.

[17] K. Nguyen, K. Ishizu, H. Murakami, F. Kojima, and H. Yano. A Scalable and Robust OpenFlow Channel for Software Defined Wireless Access Networks. In *Proc. IEEE VTC-Fall*, 2015.

[18] Multipath TCP for FreeBSD. http://caia.swin.edu.au/newtcp/mptcp/.

[19] K. Nguyen, Y. Ji, and S. Yamada. A cross-layer approach for improving wifi performance. In *IEEE IWCMC 2014*, pages 458–463, 2014.

[20] Yung-Chih Chen, Yeon-sup Lim, Richard J. Gibbens, Erich M. Nahum, Ramin Khalili, and Don Towsley. A Measurement-based Study of MultiPath TCP Performance over Wireless Networks. In *Proc. ACM IMC*, pages 455–468, 2013.