# Observing Network Handovers with Multipath TCP

Quentin De Coninck*
UCLouvain, Belgium

Olivier Bonaventure
UCLouvain, Belgium

## ABSTRACT

Multipath TCP is a recent TCP extension that enables the usage of multiple networks for a single connection. Since September 2017, Apple has enabled Multipath TCP on iOS11 for all applications. The main dedicated use case of multipath usage resides in network resiliency. We evaluate how the "interactive" mode of Multipath TCP behaves under user mobility scenarios using bidirectional constant bit rate traffic. Our results show that the passage from WiFi to cellular is not abrupt, justifying the usage of multiple paths. However, there is still some room for improvement for Multipath TCP in such cases.

**ACM Reference Format:**
Quentin De Coninck and Olivier Bonaventure. 2018. Observing Network Handovers with Multipath TCP. In *SIGCOMM Posters and Demos '18: ACM SIGCOMM 2018 Conference Posters and Demos, August 20–25, 2018, Budapest, Hungary.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3234200.3234214

## 1 INTRODUCTION

Multipath TCP [3] is a recent TCP extension that enables the transmission of data packets belonging to a single connection over multiple paths. Today, the largest deployment of Multipath TCP is on iPhones. The initial use case for Multipath TCP on iPhones was to support the Siri voice recognition application and more precisely to enable smooth handover between WiFi and cellular networks while the user interacts with the application. In September 2017, Apple released iOS11 that enables any application to use Multipath TCP. As of today, 75% of the iPhones have been upgraded to iOS11.

Although smooth handovers are one of the key benefits of Multipath TCP, its performance under those conditions have not yet been analysed in details. The first study of those handovers [5] confirmed that the Linux implementation of Multipath TCP could provide smooth handovers but it did not analyse various networks environments. Most of the scientific literature on Multipath TCP focused on its bandwidth aggregation capabilities  [1, 2, 4], despite the importance of smooth handovers for mobile applications.

To fill this gap, we design and implement an iOS11 application that uses sends and receives different types of data by using Multipath TCP while users walk away from one Wifi access point to force handovers to other access points or cellular networks. We collect and analyse 187 measurements in 30 WiFi and 13 cellular networks, collected from 23rd April to 22nd June 2018, that provide a first insight on how Multipath TCP reacts to network handovers.

## 2 METHODOLOGY

Before describing the measurement methodology, it is important to understand how Apple's API can be used by application to controls the utilisation of the Multipath TCP protocol. Apple supports two modes of operation[1] for Multipath TCP: *handover* and *interactive*. We use the *interactive* mode that was designed for applications requiring low-latency such as Siri. We infer the internals of this mode of operation from the open-source iOS source code [2]. The interactive mode prioritises the WiFi subflow over the cellular one. The packet scheduler used by iOS sends data only on the WiFi subflow, unless one of the following conditions occurs : (*i*) the smoothed RTT of the WiFi subflow is above a threshold initially set to 600 ms, while the cellular path is not, (*ii*) the WiFi path is under RTO, i.e., the timer has fired and no acknowledgement was received since that event, and the phone wants to push new data, (*iii*) the WiFi RTO value is over a threshold initially set to 1500 ms, while the cellular path is not. Notice that the Apple WiFiAssist can detect the WiFi network as bad and decrease both RTT and RTO thresholds.

We conducted our experiments with an iOS11 application called MultipathTester [3] that has been downloaded by more than hundred users. This app allows user to evaluate the distance at which their WiFi network remains reachable while walking. To start a measurement, the iPhone must be connected to both WiFi and cellular networks. When the test starts, the app creates two Multipath TCP connections

*FNRS Research Fellow

*SIGCOMM Posters and Demos '18, August 20–25, 2018, Budapest, Hungary*
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5915-3/18/08…$15.00
https://doi.org/10.1145/3234200.3234214

[1] A third one, *aggregate*, is only available for developers and cannot be used on regular devices.

[2] https://opensource.apple.com/source/xnu/

[3] https://itunes.apple.com/us/app/multipathtester/id1351286809?mt=8

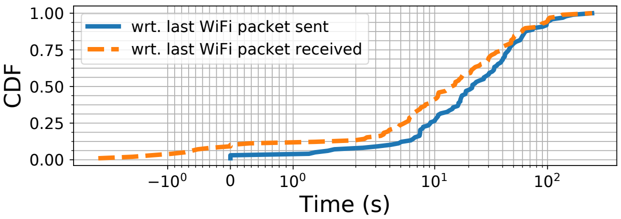

**Figure 1: Duration of the WiFi/cellular handover.**

to one of our servers using the interactive mode. The iPhone sends data at 20 KBytes/sec over the *upload* connection by sending 2 KB of data every 100 ms.[4] This traffic pattern could represent a user talking to a voice-activated application like Siri. On the second connection, called *download*, the server sends data at the same rate. The latter connection could correspond to a radio stream. On both connection, the receiving host returns a short application-level acknowledgement after the reception of each 2 KB chunk. Both the iPhone and our server compute the delay between the transmission of the first byte of the data chunk sent and the last byte of the acknowledgement relative to the data chunk received. The test completes when either the system detects the WiFi as lost or the SSID of the WiFi changes. A run might use several access points if those provide the same SSID. During the runs, we collect the internal state of TCP subflows such as the estimated RTT, the current RTO value or the number of packets sent and received. The client is an iOS11 iPhone device while the server runs MPTCP v0.94 in the Linux kernel using the default lowest-latency scheduler [6].

## 3 RESULTS

We show here preliminary results showing how Multipath TCP reacts while users walk. We first observe the duration of the network handover, then we identify its cause and finally analyse the latency observed by our test traffic.

**How long does a handover last ?** During a handover, there is a period of time where Multipath TCP uses both networks. This concurrent usage of the two networks enables a smooth handover. We measure this period as the time between the first data packet sent on the cellular by the iPhone and the last activity observed on the WiFi network. Figure 1 shows that it is frequent to observe clients that start to use the cellular network while it still sends or receives packets over the WiFi network. This shows that in mobile scenarios, the network handover is not an abrupt process. It can takes tens of seconds before switching from the WiFi to the cellular. Notice that about 10% of our tests, the WiFi has stopped working before the device started to use the cellular path, hence the negative values for the network handover.

| **Reason** | RTT Thres. | Under RTO | RTO Thres. | Other |
|---|---|---|---|---|
| **Test (%)** | 1.2% | 66.7% | 11.9% | 20.2% |

**Table 1: Reason of the beginning of the handover.**

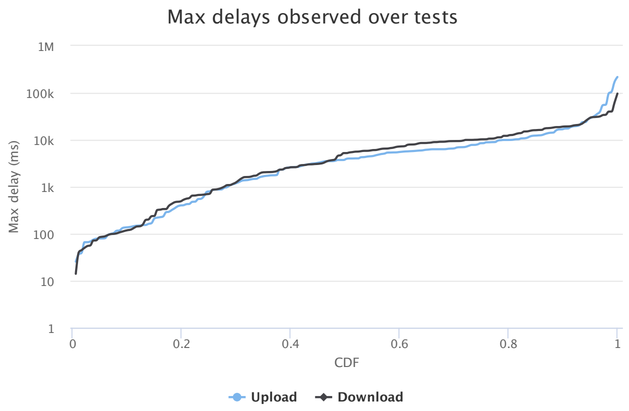

**Figure 2: Maximum observed delays.**

**Which conditions trigger the utilisation of the cellular network ?** Based on the description presented in Sect. 2 and the collected TCP subflow kernel states during the runs, we infer the reasons that triggered the smartphone to perform a handover to the cellular network by following the iOS Multipath TCP sending algorithm. Table 1 identifies that two-third of the tests started to use the cellular because new data arrived while the WiFi subflow was experiencing a RTO. This might be related to our measurement traffic that generates new data every 100 ms. In comparison, the handover due to high smoothed RTT is rare and might be related to the high threshold of 600 ms. Notice that our approach cannot directly classify the cause of the cellular usage for 20% of the tests. These are very likely triggered by WifiAssist that considerably decreases the RTO and RTT thresholds when it considers the WiFi as bad.

**What is the impact of the handover on the perceived latency ?** As the goal of the interactive mode of Multipath TCP is to provide low-latency in mobile situations, we focus on the maximum delay observed by the sending host on our two test connections. Figure 2 shows that despite the utilisation of the interactive mode, it is frequent to observe application delays of a couple of seconds. In 30% of the tests delays can exceed ten seconds. Upload connections tend to have lower delays than download ones (between 1 and 3 seconds of difference between 50th and 90th percentiles). This is likely because it is easier for the iPhone to detect a weak WiFi thanks to its local information than for the server.

[4] We do not expect issues with congestion nor receive windows.

## REFERENCES

[1] Yung-Chih Chen et al. 2013. A measurement-based study of multipath tcp performance over wireless networks. In *IMC’13*. ACM, 455–468.

[2] Shuo Deng et al. 2014. Wifi, lte, or both?: Measuring multi-homed wireless internet performance. In *IMC’14*. ACM, 181–194.

[3] A. Ford et al. 2013. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824. (January 2013).

[4] Ashkan Nikravesh et al. 2016. An in-depth understanding of multipath TCP on mobile devices: Measurement and system design. In *Mobicom’16*. ACM, 189–201.

[5] Christoph Paasch et al. 2012. Exploring mobile/WiFi handover with multipath TCP. In *CellNet’12*. ACM, 31–36.

[6] Christoph Paasch, Sebastien Barre, et al. 2009-2018. Multipath TCP in the Linux Kernel. (2009-2018). http://www.multipath-tcp.org.