Seamless Handoffs in Wireless HetNets: Transport-Layer Challenges and Multi-Path TCP Solutions with Cross-Layer Awareness

Hassan Sinky*, Bechir Hamdaoui§, and Mohsen Guizani†
* Umm Al-Qura University, Email: hhsinky@uqu.edu.sa
§ Oregon State University, Email: hamdaoub@onid.orst.edu
† Qatar University, Email: mguizani@ieee.org

***Abstract*—Complications and performance issues resulting from handoffs have widely been overlooked by transport layer protocols. In mobile scenarios layer 2 protocols begin to see issues especially when multiple handoffs are imminent. Differentiating between delay caused by actual packet loss and congestion on the current link and delay simply caused by handoffs is an important distinction where transport layer protocols fall short. With current advancement in technology traditional TCP reform is needed to accommodate for a growing mobile culture. Multi-Path TCP (MPTCP) is a new evolution of TCP that enables multiple paths or subflows and connections to be used transparently to applications. This is essential in a dynamically changing network as each subflow runs independently allowing the connection to be maintained. In this article, we present our findings on transport layer handoff issues in currently deployed networks. We then discuss the use of MPTCP as a potential solution to addressing handoff- and mobility-related service continuity issues. Finally, we propose cross-layer techniques for potential solutions to consider when designing a handoff aware MPTCP protocol.**

I. Introduction

The growing demand for access to information has grown exponentially throughout the years. We live in a culture that demands to be connected to communications services anywhere and anytime. This demand has caused an accelerated growth and directed researcher's focus towards integrating the various wireless access technologies to provide higher data rates, more services and a global roaming culture. Naturally, in heterogeneous wireless networks mobile terminals are bound to undergo multiple handoffs. In turn, mobile terminals typically have multiple network interfaces to deal with the various technologies they encounter. Users are increasingly mobile, relying heavily on their mobile devices that in some cases have replaced laptops and desktop computers. Due to the improvements in technology and overall power of mobile devices users are increasingly relying on them for everyday tasks. In other words, end users are put in more mobile situations and facing the consequences. Thus, the impact of mobility on users is of growing concern. As users become more mobile drawbacks with TCP and other traditional protocols begin to arise. Although technology is constantly evolving, TCP has remained mostly the same for more than 20 years [1]. In fact, 95% of all internet traffic is TCP [2]. With current advancement in technology traditional TCP reform is needed to accommodate for a growing mobile culture. This motivated the introduction of Multi-Path TCP (MPTCP) as a possible solution [3].

It is essential to analyze and hopefully resolve some of the issues affecting end users in a highly mobile environment. Our goal is to shed light on these overlooked issues arising from handoffs and open a discussion on potential solutions. In Section II we conduct an experiment to analyze handoff issues in currently deployed wireless networks and present our findings. We then discuss MPTCP as a potential solution to handoff issues in Section III and propose additional features that MPTCP must consider in order to provide a much more robust user experience in mobile scenarios. Open issues are briefly discussed in Section IV. Finally we conclude our article in Section V.

II. Handoff Issues in Current HetNets

Little work has been done in investigating transport layer issues during handoffs. For instance, traditional TCP has many handoff issues in terms of mobile data transfers, connection glitches and service continuity [4]. We aim to investigate these issues in currently deployed heterogeneous networks (HetNets). HetNets consist of multiple access technologies providing a variety of services. Alternatively, in terms of the transport layer, HetNets can be viewed as a collection of access networks providing different qualities of service and data rates. That is, although HetNets have been strictly associated with vertical handoffs (switching between networks of different technologies), they can include horizontal handoffs (switching between networks of the same technology) as the switch between say WiFi access points can cause different qualities of service and sudden performance issues. Currently deployed networks struggle to maintain service continuity, consistency and performance in mobile scenarios resulting in a frustrating user experience.

A. Handoffs

A handoff is the process of transitioning a user's connection from one access network to another. The challenge of such handoffs is to seamlessly maintain a mobile user's connection preferably without any perceivable interruption. Problems occur when we switch not only between networks

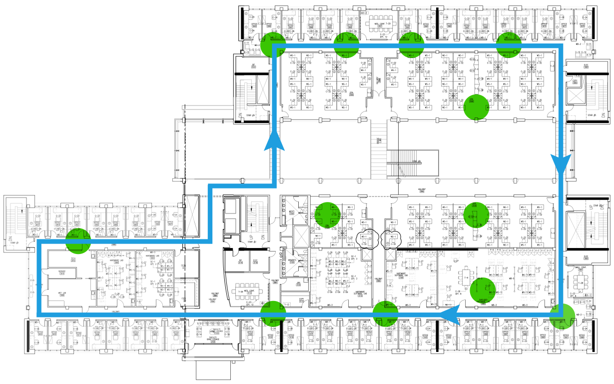


Figure 1: Experiment floor layout showing path of mobile terminal and access points

Table I: Experiment parameters

Parameter	Description
Access points	12 (Aruba Networks)
Download file size	1 Gigabyte
Duration	300 seconds
Time-step	100 milliseconds
Mobile pace	Stationary + slow walk
Conditions monitored	RTT, RSS, Distance Throughput
Number of experiments	55
Access technologies	802.11 a, b, g and n
Wireless bands	2.4 and 5 Gigahertz
Mobile device	MacBook Pro, 2.6 Ghz quad-core, Intel Core i7, 16 GB ram

with drastically different qualities or data rates but also when network qualities are somewhat similar [4], [5].

The general understanding of when a handoff occurs relies on a few characteristics mainly a user's mobility in relation to the coverage area of the access network. Naturally as a user moves away from an access network the received signal strength (RSS) decreases as well as its perceived quality. Once a threshold is reached and there exists a geographically adjacent access network with a higher RSS a handoff takes place. A mobile user may undergo multiple handoffs during an active connection's lifetime. This involves experiencing the quality degradation of a depleting data connection until a handoff is complete. This is typically followed by a relatively short period of satisfactory connection quality until the connection begins to deplete again and the cycle continues usually ending when a mobile user becomes stationary. This cycle greatly affects performance for mobile users illustrated in our findings.

B. The Experiment

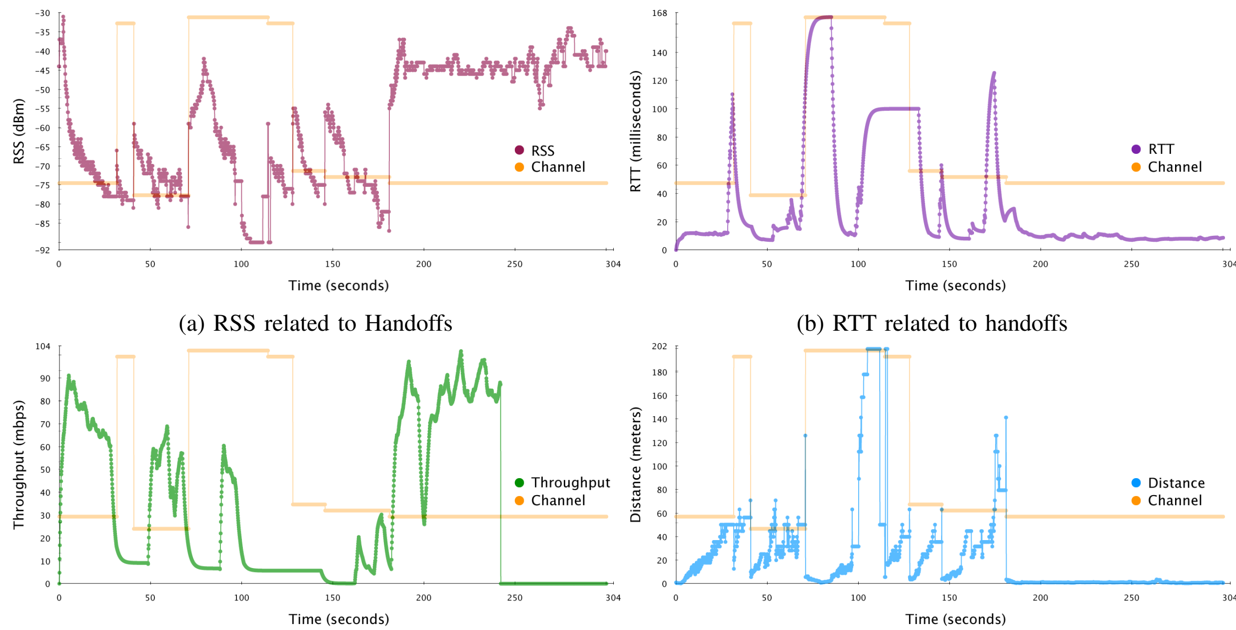
Naturally, when attempting to tackle handoff issues, the question we ask is what really happens in currently deployed wireless networks when faced with multiple mobile handoff scenarios? Fully answering this question requires the summation of a few parts. First, we developed a python script to measure and log various network metrics such as RSS, round-trip time (RTT), packet loss and throughput. Second, we designed an experiment to exploit the handoff issue. In order to investigate the drawbacks of transport layer protocols in a mobile environment and to better understand the potential issues experienced by MPTCP subflows our experiment involved monitoring network conditions while a mobile device experiences frequent handoffs. We ran our script on our mobile device as it followed the path illustrated in Figure 1 and encountering multiple WiFi access points along the way. Naturally, the mobile device frequently connects to and disconnects from multiple access points across its path resulting in a fluctuating connection and disruptive user experience. Experiment and network parameters are shown in Table I

C. Our Results Findings

Our experiment was designed to force our mobile device into multiple handoffs. Each vertical orange line in Figure 2a indicates that the mobile terminal switched between one channel to another as it moved across the path. Access points throttle down transmit rates, referred to as rate adaptation, in an effort to maintain the connection resulting in performance degradation. In addition, mobile devices increase their transmit power to remain connected resulting in decreased battery life. Thus mobile terminals tend to cling to the access point they are currently connected to unless a handoff is absolutely necessary. This causes mobile performance issues across the board when multiple handoffs take place.

Overlaying our measurements helps emphasize the relationships between them. In Figure 2a we can see RSS as a result of the handoffs experienced. Prior to each handoff we have a gradual drop in RSS as the mobile terminal moves away from the current access point. Once a particular threshold (vendor specific) is reached a handoff to a geographically adjacent access point takes place depicted as a vertical spike in the RSS. The gradual RSS drop also depends on the speed of the mobile user. The faster a user moves away from the access point the more rapid of a drop in RSS. The usage of an RSS threshold is necessary as to avoid ping ponging between access networks.

Similar behavior affecting performance and service continuity can be seen in Figures 2b and 2c which show RTT and throughput related to handoffs respectively. Mobile terminal throughput performance is measured by downloading a one gigabyte file. That is, prior to a handoff we experience a gradual throughput drop. Once a handoff is complete a gradual throughput increase is experienced. However, the throughput increases are short lived in highly mobile scenarios lasting only tens of seconds. This is disappointing as the mobile terminal's speed was not excessive and was representative of a slow walk. Unsurprisingly, stable performance is only really seen once the mobile terminal is almost stationary at around 200 seconds and near the starting access point. The download finishes at around 240 seconds. In similar fashion,



(a) RSS related to Handoffs

(b) RTT related to handoffs

(c) Throughput related to handoffs

(d) Distance related to handoffs

Figure 2: Performance measurements related to multiple handoffs

the mobile terminal experiences RTT increases prior to and as a result of handoffs and experiences more stable RTT values following the completion of a handoff. Noticeable fluctuations can be seen as the RSS drops prior to handoffs with lower RSS values resulting in higher RTT values. Naturally, as RTT increases we see a drop in throughput as well due to the handoffs. In addition it appears the residual affects of handoffs can be seen on the RTT and throughput even after a handoff is complete. In Figure 2d we graph the distance in meters from the mobile terminal to the currently connected access point. Rapid variations in distance can be seen as the mobile user undergoes multiple handoffs across the path. When the distance of the mobile terminal from its current access point is at its peak a handoff takes place. Overall, the relationships between metrics such as RSS, RTT, throughput and distance are adequate determinants of an impending handoff.

These measurements represent fundamental characteristics of wireless networks. We analyzed real world scenarios to better understand the relationship between network metrics and handoffs. Transport layer performance and network conditions were erratic in mobile scenarios undergoing multiple handoffs. TCP's lack of ability to differentiate between congestion and handoff negatively influenced connection performance. Ideally, more efficiency and improved performance in mobile scenarios is left to be desired. We can deduce that network conditions such as RSS, RTT, throughput and distance are closely related in detecting when handoffs are imminent. A combination of these metrics is necessary for an accurate representation of network conditions. This leads us to our next question and that is, given our findings, how can we provide normalcy and stability in a highly dynamic environment?

III. Potential Solutions: MPTCP

Naturally, in HetNet environments, mobile terminals are bound to undergo multiple handoffs. Mobile terminals typically have multiple network interfaces to deal with the various technologies they encounter. MPTCP is a relatively new protocol introduced by the International Engineering Task Force (IETF) in 2009, underwent 12 RFC revisions and was finally published in early 2013. MPTCP aims to decouple TCP and IP by allowing an application to send data over multiple IPs and interfaces.

Little work has been done in investigating transport layer issues during handoffs. Most researchers have focused on the physical layer aspects of handoffs and have overlooked the consequences experienced by layer 2 protocols. For instance, traditional TCP has many issues with data transfers and service continuity during mobile handoff scenarios [4], [6]. MPTCP, however, is becoming increasingly popular. With the release of iOS 7 in 2013, Apple became the first to implement MPTCP commercially. However, Apple has only considered the backup mode implementation of MPTCP which we will discuss later on [7]. In addition, an active developer community at the IP Networking Lab aim to implement MPTCP in the Linux kernel [8]. Thus, focus has been geared towards MPTCP as a promising solution for handoffs and mobile scenarios in general providing resilient and efficient data connections. Both the host and server must support MPTCP in order to work.

A. *Overview of Current MPTCP*

MPTCP is an evolution of TCP that enables multiple paths and connections to be used transparently to applications [1]. MPTCP has the ability to maintain a connection even when

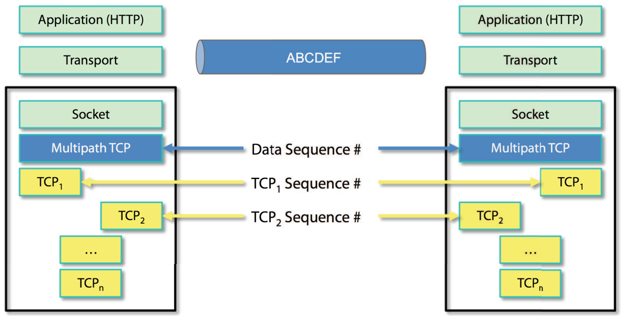


Figure 3: MPTCP architecture

endpoints of the connection change. That is, it allows for a single connection to use several subflows and interfaces simultaneously. Whereas, if an endpoint were to change in traditional TCP the connection disconnects and would need to be reestablished to resume or restart a data transfer [9]. MPTCP removes the strong coupling of TCP and IP by using multiple TCP subflows. MPTCP's usage of multiple TCP subflows helps for redundancy, improved performance and fault tolerance, which is essential for a mobile environment. A simplified overview of MPTCP can be seen in Figure 3.

1) Initiating and Joining a Connection: Similar to TCP, a MPTCP connection is initiated using a three-way handshake [10]. Both endpoints need to support MPTCP otherwise traditional TCP is used. In addition, both endpoints know the addresses available to them and can be notified of the addresses available to the other through signaling. Let's walk through the establishment of a MPTCP connection. Assume that a smartphone chooses its cellular interface to open the connection. The smartphone first sends a SYN segment to the server. This segment contains a *MP_CAPABLE* TCP option indicating that the smartphone supports MPTCP. This option also contains a token chosen by the smartphone. The server then replies with a *SYN+ACK* segment containing the *MP_CAPABLE* option and token chosen by the server. These tokens are used to prevent a malicious third party from joining the connection. Finally, the smartphone completes the handshake by sending an *ACK* segment establishing a multi-path session. The client and server can now exchange TCP segments via the cellular interface. However, the MPTCP connection cannot use the WiFi interface just yet. A full *SYN+ACK* handshake on the WiFi interface needs to be performed before sending any data packets. The *SYN* segment contains a *MP_JOIN* TCP option, along with the client's token so that the server can securely identify the correct MPTCP connection to associate this additional subflow with. Finally, the server replies with an *MP_JOIN* TCP option in the *SYN+ACK* fully establishing the new subflow. Each subflow operates independently of others with their own congestion windows and sequence numbers. A subflow is represented by an IP address and port number.

2) Data Transfers: MPTCP subflows may appear or disappear at any time during an active connection. Generally, a MPTCP connection will split a data input stream among one or more of its available subflows (i.e. WiFi and Celluar). To ensure reliable, in-order data delivery over multiple dynamic subflows, MPTCP uses two sequence numbers; a data connection level sequence number (DSN) and a per-subflow sequence number (SSN) [10]. The DSN is used to number all data sent over a MPTCP connection. Whereas, each SSN has its own space independent of the DSN. This allows MPTCP to setup and teardown subflows independent of the DSN since separate sequence number spaces are used. Each SSN includes an MPTCP option that maps the SSN to the DSN, called a Data Sequence Mapping (DSM). This allows for the same data (DSN) to be mapped to and re-transmitted on different subflows in the event of packet loss or failure. The DSM consists of the SSN, the DSN and length for which this mapping is valid. The mapping is fixed once it has been declared and carried along with the segment being sent. Received DSN are acknowledged using a data connection level acknowledgment (Data ACK). Using the DSM, the connections data is striped among all available subflows where most of the data is sent on the least congested subflow [11], [12].

B. Suitability of MPTCP for HetNets

The novelty of MPTCP is its ability to use multiple TCP connections through a single or multiple interfaces transparently to the application. This is realized by the concept of subflows. The first subflow created is the default subflow and is referred to as the master subflow. Subsequent subflows are referred to as slaves. MPTCP subflow policies are implementation specific and can generally be, but not limited to, three types [9], [8]:

- **Alternate:** this policy uses all available subflows. The default version of MPTCP uses the least congested subflow represented by the lowest RTT [12]. Naturally, the least congested path will not remain the least congested. Thus, a second subflow will be used if it is less congested by a certain threshold controlled by an aggressiveness factor. The default version of MPTCP aims to balance the flapping between subflows [1]. Thus, MPTCP alternates between master and slave subflows based on the coupled congestion control mechanism in [13]
- **Backup:** in this policy MPTCP opens connections among all available subflows. Subflows can be prioritized and used only when needed as backups if the master subflow fails. In backup mode additional subflows are used only when the master subflow has failed, an all-or-nothing approach. Additionally, a backup subflow can be used if other subflows are overflowed. Also worth noting is a similar policy to backup mode called single-path mode which only opens a new subflow if the current subflow fails
- **Simultaneous:** in this policy MPTCP simultaneously sends the same data across all subflows. This achieves redundancy and robustness in the event of failure. However, this policy can be viewed as a waste of resources. Data can also initially be divided or mapped equally or conversely among available subflows and sent simultaneously

MPTCP's inherent architecture provides a potential solution to the handoff problem. MPTCP's usage of multiple subflows helps alleviate some of the transport issues arising from handoffs. A mobile device running MPTCP with a single cellular interface and WiFi interface would be able to use both simultaneously. As the mobile user comes in or out of range of a particular network additional subflows can be used interchangeably. This ability to simultaneously use multiple subflows allows for a MPTCP connection to remain active in case of a handoff. For instance, a smartphone has two network interfaces; a WiFi and cellular interface each with its own IP address. MPTCP would allow an application on the smartphone to use a single MPTCP connection that can use both WiFi and cellular interfaces to communicate with the server even if the connection endpoints change, whereas traditional TCP would break. Thats is, normally, when the current connection reaches an unusable level the mobile device will either need to establish a new connection using its cellular interface or wait for the current WiFi connection to improve, often times disconnecting, which drastically reduces the user's experience. With the ability to stripe its data among available subflows the MPTCP connection is able to be maintained in case of a handoff on the WiFi interface without the need to re-establish the connection.

C. Cross-Layer Assisted MPTCP: Proposed Solution

Although MPTCP is a promising protocol aimed to provide resilient and efficient data connections in a dynamic environment; there are still a few issues when faced with mobile handoff scenarios. When a user is mobile and a handoff is imminent packets begin to be lost and delay increases resulting in performance issues as shown in our findings. The TCP subflows of MPTCP will experience and suffer the same affects. Link congestion, packet retransmission and reordering are not new issues, however, their consequences are exacerbated in mobile scenarios as layer 2 protocols tend to overlook handoff occurrences. The default decision that governs MPTCP when to use a particular subflow is based on a coupled congestion control mechanism [13] that favors balancing the load among subflows by preferring the least congested path (lowest RTT). Larger congestion windows indicate a larger RTT. Naturally, as a path is used more congestion increases prompting MPTCP to choose the least congested path, alternating between subflows. However RTT alone is not always a good representation of link congestion as fluctuations in network conditions may simply be an indication of an impending handoff. Thus, a depleting connection due to handoff may be misinterpreted by MPTCP as being highly congested. The time needed for RTT updates to characterize a congested link is detrimental for a highly mobile user. The following are scenarios that emphasize the need for cross-layer assistance in MPTCP:

- In MPTCP's alternate mode the connection's data is striped among all subflows as space in the subflow windows becomes available where most of the data is sent on the least congested subflow. Data that was originally striped onto a particular subflow is delayed or lost at the time of handoff and would need to be remapped onto other subflows. The worst case scenario is a subflow's entire congestion window is lost resulting in a gap in the received data sequence numbers. These data sequence numbers would then need to be remapped to other available subflows causing glitches and affecting service continuity at the point of handoff for the end user. Delay is increased even more if retransmission timeouts are high forcing subflows to be momentarily inactive.
- In backup mode packets are transmitted on different subflows only if the master subflow fails. This is not ideal for handoff scenarios as it negatively affects service continuity by waiting for a connection to fail before using another. Single-path mode would experience even more performance issues as it suffers from a short period where no subflow is active [9]
- In simultaneous mode the same input data stream can be striped across all subflows. That is, mapping the same data sequence numbers for all subflows. That way, if one connection is depleting or fails, service continuity is maintained and connection performance preserved. However, this is a considerable waste of resources and increases overhead due to excessive duplicate packets and the processing and reordering of packets.

Packet losses and increased RTT alone are not always indicative that there exists congestion on a network as the MPTCP connection may be experiencing a handoff. Thus balancing between network congestion and impending handoff as a subflow policy is necessary. A solution is required that incorporates not only real-time network monitoring but also cross-layer assistance to be implemented into transport layer protocols.

Thus, a more reasonable solution is to implement an adaptive metric that accurately monitors network conditions searching for signs of a depleting connection due to either congestion or handoff. If a handoff is anticipated, the MPTCP connection maps the same data currently being sent (in-flight data) as well as subsequent data from the input and transmits it simultaneously across all subflows, for a short period of time. The simultaneous data transmission begins prior to and in anticipation of the handoff and ends once it is complete. Otherwise, if a depleting connection is due to network congestion, normal MPTCP operation is used. A potential solution would need to define a user mobility metric to help differentiate between delay caused by mobility and congestion. We define *Mobility Index (MI)* as the degree at which a user is mobile. The higher the *MI* the more mobile a user is. Figure 4 summarizes some of these points. For instance, as the user moves further away from its point of attachment network measurements such as RSS (low signal strength) or RTT (high delay) values may indicate increased mobility. That is, the more mobile a user is will require a more aggressive MPTCP congestion control policy where most data is sent across the least congested subflow in anticipation of a handoff. Thus, providing a more seamless transition.

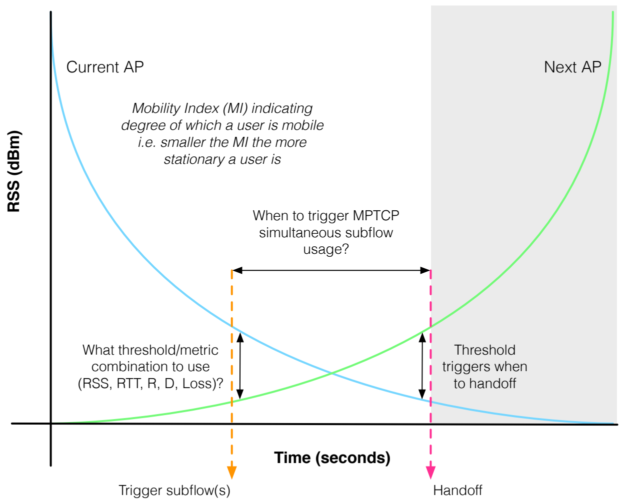


Figure 4: Cross-layer assisted handoff MPTCP

In addition, this will involve monitoring the aforementioned network conditions (RSS, RTT and throughput) and the rate at which they change while defining a threshold indicating that a handoff is imminent, hence, dynamically initiating MPTCP subflow usage. We propose the dimensioning of MPTCP into multiple stages based on a user's mobility and current network conditions. It is important to note that our goal is to develop an adaptive protocol that is capable of adjusting to fluctuating network conditions resulting from user mobility. Dimensioning MPTCP alone is not sufficient, however, and will require the design of an adaptive cross-layer congestion control model. Using measured network conditions, our work suggests partitioning MPTCP into three stages; stationary, mobile and handoff. Throughout the stages a throughput threshold is attempted to be maintained. During the stationary stage MPTCP is assumed to be achieving the capacity throughput provided by its current connection. During the mobility stage MPTCP assumes the user is moving away from its point of attachment and attempts to maintain the throughput threshold by adaptively using its available subflows. Finally, during the handoff stage, at the time a handoff occurs, MPTCP adjusts a subflow's current congestion window to the subflows path bandwidth delay product (BDP). The dimensioning of MPTCP allows for mobility induced issues to be addressed efficiently by attempting to maintain a connection level throughput threshold. Thus, our goal is to anticipate packet loss and delay due to handoff to trigger dynamic usage of additional subflows.

Our findings strongly suggest the need for a solution to transport layer issues resulting from handoffs. Thus, potential solutions must account for the following concepts in order to design a handoff aware MPTCP protocol:

- Ability to differentiate between fluctuations in network conditions caused by congestion and fluctuations caused by mobility and network handoffs
- Dimensioning MPTCP into multiple stages while adaptively adjusting to fluctuating network conditions due to mobility
- Cross-layer assisted subflow policy through handoff anticipation DSM: if a handoff is anticipated data sequence mapping can begin from the last received data acknowledgment and mapped to all available subflows for the duration of the handoff
- Cross-layer assisted address knowledge exchange: prior to handoff additional available addresses need to be exchanged between source and destination

The design and implementation of a cross-layer congestion control mechanism that leverages network conditions to dynamically adjust subflow usage is left for future work. The following section will discuss some of the open issues related to MPTCP and mobile handoff scenarios.

IV. Open Research Problems

Possible solutions in [14], [15] have been proposed. These solutions utilize MAC-layer frame retransmission and error rates to assess link quality. Physical layer frame sizes are very small where error rates are less thus reducing accuracy. In addition, the physical layer experiences much higher retransmissions not seen by the upper layers which leaves room for false positives. That is, high frame retransmissions may indicate congestion or a handoff. Also, exponential backoff in turn increases delay. Using this alone is not enough to anticipate handoffs.

Another open issue is the case where multiple subflows exist on a single interface. All subflows on a particular interface must go through the same cellular tower or same WiFi access point as shown in Figure 5. That is, a single interface cannot physically connect to multiple cellular towers or multiple WiFi access points. This is due to a hardware limitation where a single interface antenna can only be tuned to a single frequency at one time. In the event of a handoff all subflows on the current network must disconnect and reconnect on the next access network. Conversely, handoffs between access points within the same network can maintain their connection but still suffer from the consequences of a depleting connection. This requires more complex subflow policies in order to maintain service continuity. For this purpose, we focused on the practical scenario where a mobile device is equipped with both WiFi and cellular interfaces and encounters multiple access points during an active connection. However, a full investigation of this limitation and its impact on MTPCP is left for future work.

Being a relatively new protocol, MPTCP leaves ample room for improvement. As the protocol is more widely adopted and popularity increases we will see such issues addressed and given the attention needed.

V. Conclusion

In this article we discussed issues dealing with handoffs in heterogeneous wireless networks. Specifically we focused on the weaknesses of transport layer protocols in mobile scenarios where multiple handoffs are imminent. We argue that from the perspective of layer 2 protocols heterogeneous networks can be viewed as simply a collection of networks providing

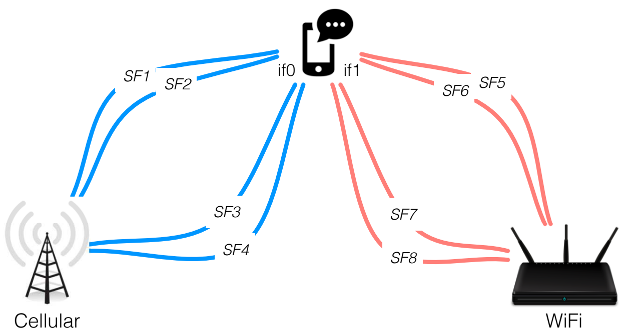


Figure 5: Subflows (SFx) must go through access network

different services and data rates. We conducted an experiment exploiting mobile handoff scenarios and discussed our findings in analyzing the resulting issues with traditional TCP in currently deployed wireless networks. We then discussed the suitability of MPTCP as a potential solution to the handoff issue. Finally, we expressed downfalls with MPTCP in highly mobile scenarios and emphasized the need for cross-layer assistance to be incorporated in a potential MPTCP solution.

References

[1] O. Bonaventure, M. Handley, and C. Raiciu, "An overview of multipath tcp," *USENIX;*, October 2012.
[2] S. Barre, C. Paasch, and O. Bonaventure, "Multipath tcp: From theory to practice," in *IFIP Networking, Valencia*, May 2011.
[3] H. Sinky, B. Hamdaoui, and M. Guizani, "Proactive multipath TCP for seamless handoff in heterogeneous wireless access networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4754–4764, 2016.
[4] H. Sinky and B. Hamdaoui, "Cross-layer assisted TCP for seamless handoff in heterogeneous mobile wireless systems," in *Globecom Workshops (GC Wkshps), 2013 IEEE*. IEEE, 2013, pp. 4982–4987.
[5] L. Daniel and M. Kojo, "Employing cross-layer assisted tcp algorithms to improve performance with vertical handoffs," in *International Journal of Communications Networks and Distributed Systems,*, 2008.
[6] H. Sinky, B. Hamdaoui, and M. Guizani, "Handoff-aware cross-layer assisted multi-path TCP for proactive congestion control in mobile heterogeneous wireless networks," in *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 2015, pp. 1–7.
[7] *http://support.apple.com/kb/ht5977*, 2014.
[8] *http://www.multipath-tcp.org/*, 2014.
[9] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath tcp," in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, ser. CellNet '12. New York, NY, USA: ACM, 2012, pp. 31–36. [Online]. Available: http://doi.acm.org/10.1145/2342468.2342476
[10] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, *https://tools.ietf.org/html/rfc6824*, 2013.
[11] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp," in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 99–112. [Online]. Available: http://dl.acm.org/citation.cfm?id=1972457.1972468
[12] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? designing and implementing a deployable multipath tcp," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 29–29. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228298.2228338
[13] C. Raiciu, M. Handley, and D. Wischik, *http://tools.ietf.org/html/rfc6356*, 2011.
[14] X. Zhang, T.-M.-T. Nguyen, and G. Pujolle, "A cross-layer approach to optimize the performance of concurrent multipath transfer in wireless transmission," in *Proceedings of the 2Nd IFIP Conference on Wireless Days*, ser. WD'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 107–111. [Online]. Available: http://dl.acm.org/citation.cfm?id=1834116.1834136
[15] Y. sup Lim, Y.-C. Chen, E. Nahum, D. Towsley, and K.-W. Lee, "Cross-layer path management in multi-path transport protocol for mobile devices," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 1815–1823.