Handoff-Aware Cross-Layer Assisted Multi-Path TCP for Proactive Congestion Control in Mobile Heterogeneous Wireless Networks

Hassan Sinky*, Bechir Hamdaoui*, and Mohsen Guizani†
*Oregon State University, Email: sinkyh,hamdaoub@onid.orst.edu
† Qatar University, Email: mguizani@ieee.org

***Abstract*—Multi-Path TCP (MPTCP) is a new evolution of TCP that enables a single MPTCP connection to use multiple TCP subflows transparently to applications. Each subflow runs independently allowing the connection to be maintained if end-points change; essential in a dynamic network. Differentiating between congestion delay and delay due to handovers is an important distinction overlooked by transport layer protocols. Protocol modifications are needed to alleviate handoff induced issues in a growing mobile culture. In this article, findings are presented on transport layer handoff issues in currently deployed networks. MPTCP as a potential solution to addressing handoff- and mobility-related service continuity issues is discussed. Finally, a handoff-aware cross-layer assisted MPTCP (CLA-MPTCP) congestion control algorithm is designed and evaluated.**

I. Introduction

The recent growth in data demand has prompted researchers to come up with new wireless techniques (e.g., MIMO [1], co-operative communication [2], femtocells [3], etc.) and develop new technologies (e.g., cognitive radio [4], [5], LTE [6], etc.) to be able to meet this high demand. With the emergence of such many access technologies, there has clearly been a need for integrating these various wireless technologies to enable a global roaming culture and a smooth handoff across the various technologies. Naturally, in heterogeneous wireless networks (HetNets), mobile terminals are bound to undergo multiple handovers and thus are equipped with multiple network interfaces for the technologies they encounter. As users become more mobile, drawbacks with TCP and other traditional protocols begin to arise. Disruptions in service continuity due to handovers is an unacceptable outcome, and it is therefore important to address them. Although technology is constantly evolving, TCP has remained mostly the same for more than 20 years. These reasons have motivated the introduction of Multi-Path TCP (MPTCP) as a possible solution.

This work sheds light on overlooked handoff issues and proposes a Cross-Layer Assisted MPTCP (CLA-MPTCP) to alleviate some of these issues. Contributions of this work are:

- Establishing that reducing congestion windows at time of handoff is essential for ensuring service continuity and minimizing disruption and glitches.
- Design of a handoff-aware cross-layer assisted MPTCP coupled congestion control algorithm.
- To our knowledge, this work is the first to investigate handoff disruptions and glitches with MPTCP, and to improve service continuity during handovers through proactive congestion window adjustments.

This work was supported in part by the US National Science Foundation under NSF award CNS-1162296.

The rest of this paper is organized as follows. In Section II our experiment findings are presented to analyze handoff issues in currently deployed wireless networks. MPTCP as a potential solution to handoff issues is discussed in Section III. CLA-MPTCP is presented and evaluated in Sections IV and V. Finally, we conclude in Section VI.

II. Handoff Induced Issues in Current HetNets

Although many works have focused on addressing handoff issues at the network-layer and below [7], [8], little focus has been given to issues arising at the transport layer. For instance, traditional TCP has many handoff issues in terms of mobile data transfers, connection glitches and service continuity [9]. The authors in [9] were the first to examine these issues using the New Reno variant of TCP which is the most widely used variant today. The challenge is to seamlessly maintain a mobile user's connection preferably without perceivable interruption in the event of a handoff. These issues are among many that have motivated the introduction of MPTCP. Since MPTCP utilizes multiple TCP subflows it is essential to understand basic TCP handoff induced issues.

Different TCP problems arise depending on the type of handoff that occurs [10]:

1) High-delay network to low-delay network:

- **Packet reordering:** This issue occurs when the source node switches to a considerably faster link. Packets sent on the new link overtake packets that have been sent on the old link which causes out of order packets to arrive at the destination, resulting in duplicate acknowledgments (dupacks) to be sent. TCP would misinterpret 3 dupacks as being caused by congestion or loss and re-transmit packets incorrectly assumed to be lost when in reality this was caused by a handoff.
- **Inflated re-transmission timeout (RTO):** Naturally, a high-delay link will have high RTT and RTO values. Since the RTO is updated once every RTT, when a handoff occurs, the RTO will converge slowly to the new low RTO value. This is an issue because invoking RTO recovery to recover lost packets will take longer. In other

words, in the event of lost packets on a low-delay link, TCP would not recover quickly due to the high RTO value of the high-delay link. Thus, TCP must be able to recover lost packets faster using a smaller RTO representing the new low-delay link.

2) Low-delay network to high-delay network:

- **Spurious or false re-transmission timeout (RTO):** This issue arises when a handoff occurs to a slower link. The low-delay link before the handoff has a low RTO value. Packet acknowledgments before the handoff take the high-delay link after the handoff. This causes TCP to falsely or spuriously timeout assuming packets have been lost due to a low RTO value from the low-delay link.
- **Link overshoot:** In high-delay networks TCP requires several RTTs to probe available bandwidth. Specifically, after a handoff to a high-delay link the sender may inject more data onto the link than can be handled resulting in dropped packets. In addition, when TCP is in the slow start phase the congestion window grows exponentially. If a handoff occurs during the slow start phase a congestion window increase may result in packet losses and timeouts. This is referred to as a slow start overshoot. TCP may interpret this incorrectly and enter its congestion avoidance phase with a very small congestion window resulting in suboptimal TCP performance.

Transport layer protocols are neutral to the varying technologies of HetNets; rather, they react to the qualities of service and data rates available. Current deployed networks struggle to maintain service continuity, consistency and performance in mobile scenarios resulting in a frustrating user experience which is illustrated in II-A.

A. Transport Layer Analysis

A preliminary experiment was designed to force a mobile device into multiple handovers. The HetNet in the EECS department at Oregon State University consists of 41 access points using both the 2.4 and 5 GHz bands and 802.11 a, b, g and n access technologies. The experiment was limited to the third floor, which comprised of 12 access points. A python script logged network measurements while following a path involving multiple access points as shown in Figure 1a. Additionally, a one gigabyte file was downloaded by the mobile terminal. Figure 1 uses one of the experimental runs to illustrate these findings.

Access points throttle down transmit rates while mobile devices increase their power in an effort to maintain the connection resulting in performance degradation. Mobile terminals tend to cling to their current access point unless a handoff is necessary. This causes performance issues when multiple handovers take place as shown by the erratic signal-to-noise ration (SNR), throughput and RTT behavior in Figure 1. Each vertical line indicates a channel switch by the mobile terminal as it moves across the path.

A mobile user can undergo multiple handovers during an active connection's lifetime. This involves experiencing the quality degradation of a depleting data connection until a handoff is complete. This is followed by a period of satisfactory connection quality until the connection begins to deplete again usually ending when a mobile user becomes stationary. This cycle greatly affects performance for mobile users.

Transport layer protocols **must** be able to differentiate between delay caused by actual packet loss and congestion on a link and delay simply caused by handovers. A combination of metrics must be considered for an accurate representation of network conditions. Section III briefly introduces MPTCP and some of its drawbacks.

III. Multi-Path TCP (MPTCP)

Despite being a relatively new protocol, MPTCP is becoming increasingly popular. With the release of iOS 7 in 2013, Apple became the first to implement MPTCP commercially. The novelty behind MPTCP is its ability to decouple TCP and IP to simultaneously use multiple TCP subflows and interfaces transparent to the application [11].

The main modes available to MPTCP are `fullmesh` and `backup` [12], [13]. In `fullmesh` 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 (lowest RTT) [14]. In `backup` mode slave subflows that have joined the current MPTCP connection are only used when the master subflow, used for initiating the MTPCP connection, fails.

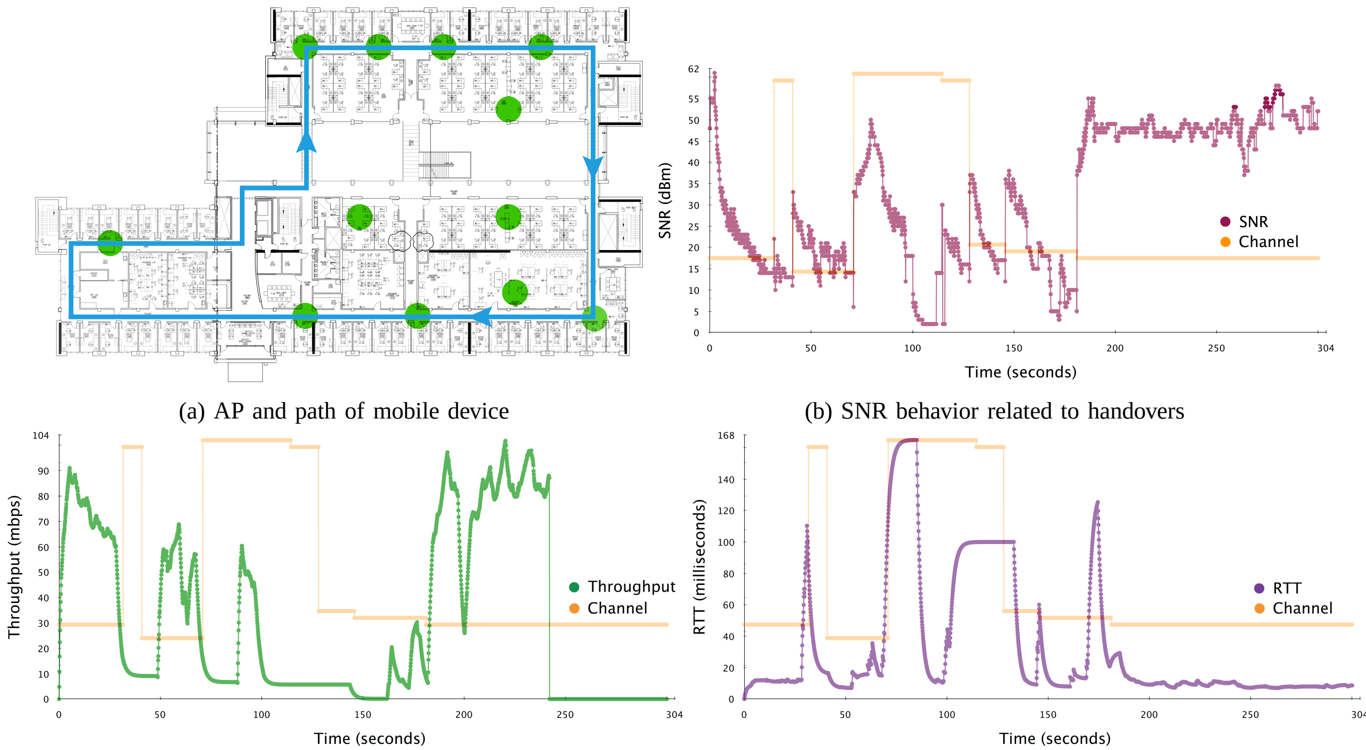
A. MPTCP Architecture

A simplified overview of MPTCP can be seen in Figure 2. MPTCP's inherent architecture provides a potential solution to the aforementioned handoff issues. For instance, an application on a smartphone may use a single MPTCP connection to utilize both Wi-Fi and cellular interfaces to communicate with a server even if endpoints were to change, whereas traditional TCP would break.

MPTCP subflows may appear or disappear at any time during an active connection. Similar to TCP, each subflow is initiated using a three-way handshake and operates independently of others with their own congestion states (i.e. cwnd) [15]. For reliable, in-order data delivery MPTCP uses a data sequence number (DSN) to number all data sent over a MPTCP connection and a per subflow sequence number (SSN) mapped to the DSN. This allows for the same data (DSN) to be re-transmitted on different subflows in the event of packet loss or failure. The mapping, once declared, is fixed and carried with the SSN sent. DSN are acknowledged using a data connection level acknowledgment.

MPTCP aims to simultaneously pool available resources, appearing as a single resource to the end user application, whilst greatly improving user experience. This is realized through three MPTCP goals listed in [16]:

- **Goal 1** (Improve throughput) Perform at least as well as single-path TCP would on its best subflow.
- **Goal 2** (Do no harm) Do not use more capacity than if it was single-path TCP on its best subflow.



(a) AP and path of mobile device

(b) SNR behavior related to handovers

(c) Throughput performance related to handovers

(d) RTT behavior related to handovers

Figure 1: Transport layer performance measurements related to multiple handoffs

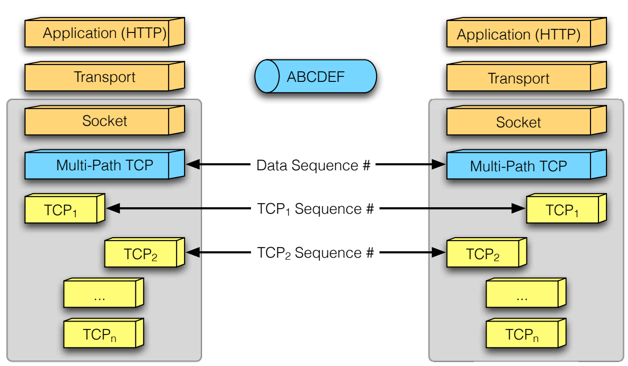


Figure 2: MPTCP architecture

- **Goal 3** (Balance congestion) MPTCP should move as much traffic as possible off its most congested paths.

B. *MPTCP Congestion Control*

Simply running standard TCP congestion control on each subflow gives MPTCP an unfair share of capacity when its subflows share a bottleneck [16]. By default, MPTCP implements a coupled congestion control algorithm that links subflow increase functions. For each ACK on subflow i, congestion window w_i is increased by $min(\frac{\alpha}{w_{total}}, \frac{1}{w_i})$. For each loss $w_i = \frac{w_i}{2}$. The α parameter controls the aggressiveness of a MPTCP connection and is chosen so that the MPTCP aggregate flow is equal to the achievable throughput of a single-path TCP flow on the best path [16]:

$$\alpha = w_{total} \times \frac{\max\left(w_i/RTT_i^2\right)}{\left(\sum w_i/RTT_i\right)^2}$$

C. *Drawbacks of MPTCP Congestion Control*

Handoff issues presented in Section II still apply to the TCP subflows of MPTCP. For example, MPTCP reordering occurs both at the subflow and data levels which can cause added complexity with duplicate ACKs and spurious RTOs. Common MPTCP drawbacks that may arise from handovers are listed below:

- **Responsiveness** MPTCP relies on α to adapt to network conditions. However, to reduce computational costs, α is only updated when there is a packet drop or once per RTT [17]. In addition, sender RTT estimations can be inaccurate due to TCP's delayed ACK mechanism and smoothed RTT values (SRTT) especially when a network is over-buffered. This results in slow responsiveness to changes in subflow congestion windows resulting in an underestimation of α affecting MPTCP's increase rate. 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.

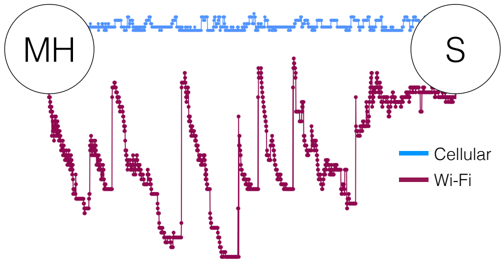


Figure 3: Subflow SNR behavior between mobile and server

- **Fullmesh mode** 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. Delay is increased even more if RTOs are high forcing subflows to be momentarily inactive.
- **Backup mode** 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 [12]

Packet loss and RTT alone are not enough for a MPTCP connection to quickly adapt to handovers. Thus, balancing between network congestion and impending handoff as a congestion control mechanism is necessary. A solution is required that incorporates not only real-time network monitoring but also cross-layer assistance in MPTCP. In the following section a Cross-Layer Assisted MPTCP (CLA-MPTCP) is proposed to alleviate handoff induced glitches and disruptions in service continuity.

IV. Cross-Layer Assisted Congestion Control

MPTCP requires the ability to differentiate between fluctuations in network conditions caused by congestion and those caused by mobility and network handoffs. Figure 3 shows subflow SNR of a typical MPTCP connection (cellular + Wi-Fi) as a mobile host moves across a path while communicating with a server. The subflow undergoing multiple handovers i.e., Wi-Fi, experiences visible SNR fluctuations that must be accounted for by MPTCP in order to provide adequate service continuity. As mentioned before, relying on RTT alone as a measure of link quality does not allow MPTCP to adapt quickly enough especially when RTT and RTO values are high.

In order to minimize connection glitches and maintain adequate service continuity throughout the handoff process the bandwidth delay product (BDP) and subflow SNR of a link must be considered [9]. The BDP refers to the maximum amount of data that can be sent on a network link at any given time. In other words, it represents the amount of data that is sent before the first ACK is received [18]. The BDP is represented by the product of a link's capacity and its RTT. By definition then the path BDP, BDP_{path}, is the product of the path bottleneck throughput, R_{min}, and the path round-trip time, RTT_{path}[1]; i.e., $BDP_{path} = R_{min} \times RTT_{path}$. When propagation and queuing delays are neglected, we can write

$$BDP_{path} = R_{min}\left(\sum_{i=1}^{n}\frac{S}{R_i} + \sum_{i=1}^{m}\frac{S}{R'_i}\right) \quad (1)$$

where n and R_i are the number of hops and throughput on the forward path, m and R'_i are the number of hops and throughput on the backward path, and S is the packet size. As done in [18], the path BDP can be upper bounded by

$$R_{min}\left(n\frac{S}{R_{min}} + m\frac{S}{R'_{min}}\right) \quad (2)$$

where R'_{min} is the bottleneck throughput in the backward path. Now considering that the forward and backward paths are likely to be the same, the upper bound given in Equation (2) can be replaced by $S \times N$ where $N = n + m$; i.e.,

$$BDP_{path} \leq S \times N \quad (3)$$

Note that having the path BDP exceed $S \times N$ results in packets being queued at the bottleneck [18]. In what follows, let $BDP_{UB} = S \times N$.

A. Design

Modifications to MPTCP are proposed to alleviate the issues presented in Section II. A fourth MTPCP design goal is presented:

- **Goal 4** (Cross-layer assistance) A multi-path flow should consider both a subflow's SNR and BDP to anticipate and alleviate handoff induced network conditions to maintain and improve service continuity.

In designing CLA-MPTCP three assumptions are made:

- A mobile device is equipped with two interfaces (i.e. cellular and Wi-Fi)
- A MPTCP mobile user undergoes multiple handovers on its Wi-Fi interface across a path. This involve times where a mobile node is in range of multiple networks.
- Both endpoints of a MPTCP connection exchange, through signaling, cross-layer information such as handoff, mobility and SNR notifications.

CLA-MPTCP brings three modifications to MPTCP:

- Utilization of a mobility index MI and SNR metric Q to forecast handovers.
- Subflow congestion windows are set to BDP_{UB} and RTT and RTO estimations are reset on handoff notifications.
- Couples MI and Q with MPTCP's congestion control algorithm to preemptively adjust the aggressiveness of individual subflows in anticipation of handovers.

[1]RTT consists of propagation and transmission delays only; i.e., it does include queuing delays.

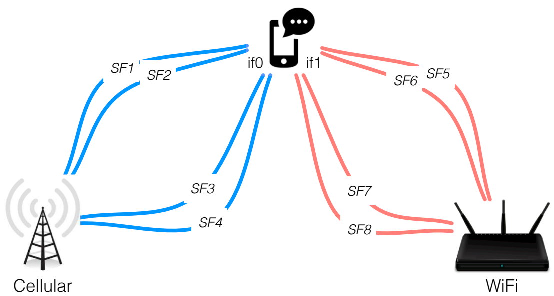


Figure 4: Subflows must go through same point of access

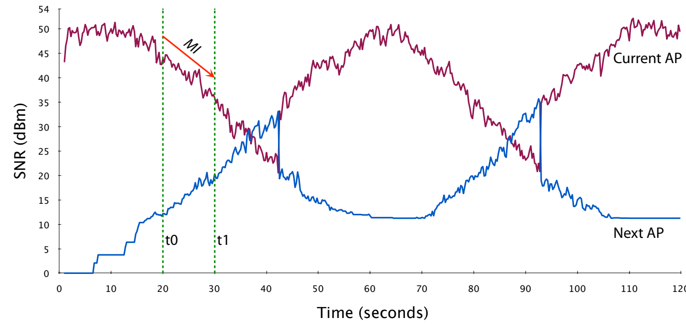


Figure 5: Mobile SNR readings across path

All subflows on a particular interface must go through the same cellular tower or Wi-Fi access point as shown in Figure 4. That is, a single interface cannot be connected to multiple cellular towers or multiple Wi-Fi access points. 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. To account for this CLA-MPTCP categorizes a mobile user into three distinct stages: stationary, mobile and handoff. For the stationary stage standard MPTCP is used. Algorithms 1 and 2 are applied by a CLA-MPTCP sender in the mobile and handoff stages respectively. The design of these stages are discussed next.

B. CLA-MPTCP Mobile Stage

In CLA-MPTCP a simple mobility index, $MI = \frac{\arctan(m)}{\pi/2}$, is maintained per interface. A forecasting module estimates the user's current SNR trend (slope m) and returns a normalized value between -1 and 1. An MI closer to -1 or 1 indicates how rapid a mobile user is moving from or to its respective access network, and an MI close to 0 depicts stationarity.

Mobile devices use a trigger threshold to initiate a scan for available access networks [19]. Once the difference in signal strength between current and candidate access networks reaches a particular threshold, the mobile device re-associates and a handoff ensues. CLA-MPTCP maintains an SNR based handoff metric Q that considers both an interface's current SNR_1 and next candidate SNR_2 at time t_0:

$$Q = \frac{SNR_1(t_0)}{SNR_1(t_0) + SNR_2(t_0)} \tag{4}$$

The analysis from II shows rapid variations in throughput and RTT around an SNR trigger threshold[2] of 20 dBm. From Equation 4, naturally as Q approaches $1/2$ the handover likelihood increases. This can easier be seen in Figure 5. Coupling MI and Q provides adequate information for MPTCP to anticipate handovers.

[2]SNR depends on received signal strength (RSS) and noise. Apple devices use an RSS trigger threshold of -70 dBm.

Due to immense bufferbloat in real networks [17], the amount of actual in-flight packets is much higher than the real path BDP. Thus, in order to reduce a subflow's throughput, in response to mobility and handovers, the congestion window needs to be proactively reduced below the path BDP. In algorithm 1 subflow MI and Q are exchanged between source and destination. First, handoff likelihood is checked. MPTCP subflow congestion windows on handoff interfaces are then tuned to their BDP_{path} using a new reduced rate $R_{new} = R_i(t_0) + R_{off}$ where $R_{off} = R_1(t_0) \times (1 - Q) \times MI$. Conversely, MPTCP subflow congestion windows on non-handoff interfaces are tuned using $R_{new} = R_i(t_0) - R_{off}$. MI is used to control aggressiveness depending on user mobility. An MI closer to -1 will cause the subflows on handoff interfaces to offload more data onto its other non-handoff subflows.

Algorithm 1 CLA-MPTCP Mobile Stage

for each subflow i **do**
  if $Q_i \leq \frac{1}{2}$ **then** ▷ subflow approaching handoff
    calculate $R_i(t_0)$
    let $R_{off} = R_i(t_0) \times (1 - Q_i) \times MI_i$
    let $R_{new} = R_i(t_0) + R_{off}$
  else ▷ non handoff subflows
    calculate $R_i(t0)$
    let $R_{new} = R_i(t_0) - R_{off}$
  end if
  calculate
$$BDP_{path} = \frac{R_{new} \times RTT_i}{MSS_i}[pkts]$$
  set $cwnd_i$ = BDP_{path}
end for
re-calculate α using adjusted cwnds

C. CLA-MPTCP Handoff Stage

A large congestion window during a handoff can be detrimental to service continuity. Maintaining a large congestion window increases the probability of contention, RTT, queuing and packet loss which can degrade MPTCP performance drastically. By setting the congestion window to the path BDP, that is the maximum number of bytes that can be in transit across the path, guarantees we are transmitting the maximum amount of bytes while avoiding packet queuing at intermediate links at the time of handoff.

When a MPTCP subflow goes through a network handoff algorithm 2 is applied to all subflows. The intuition behind this is that if the RTO is set too high MPTCP may be slow to recover lost packets. If set too low, spurious RTOs cause MPTCP to assume packets are lost when in reality they may simply be on their way. Analyses established that by resetting these values MPTCP converges quicker to the new network's RTT and RTO estimations. Depending on the subflow TCP variant, duplicate acknowledgments are temporarily disabled to avoid false dupacks caused by packet reordering. Finally, the congestion window of each subflow is set to their respective path BDP to avoid added queuing at the bottleneck as well as link overshoots. Using the path BDP ensures throughput is not compromised as shown in Section V.

Algorithm 2 CLA-MPTCP Handoff Stage

```
1: if Notification of handoff on subflow i then
2:     reset RTT_i and RTO_i estimations
3:     temporarily disable duplacks
4:     obtain path BDP_UB
5:         set cwnd_i = BDP_UB
6: end if
```

In Section V service continuity and performance measurements (i.e. throughput, RTT, re-transmissions and duplicate ACKs) are evaluated.

V. CLA-MPTCP Evaluation

To evaluate CLA-MPTCP the network topology in Figure 6a is built using NS3-DCE. DCE is a framework that allows for existing Linux `kernelspace` network protocols to be used within NS3 simulations [20]. In addition, a simple access point decision algorithm was implemented in NS3 using an SNR trigger threshold of 20 dBm. CLA-MPTCP modifications were implemented in version 0.87 of Linux MPTCP and used on NS3 nodes using DCE. Each CLA-MPTCP user is equipped with two interfaces, i.e. cellular and Wi-Fi. As done in Section II-A, the mobile client moves across a path and undergoes two Wi-Fi handovers while remaining connected to the LTE network. Data is sent from the source to the destination at 20 Mbps.

Figure 6c compares the achieved throughput between cross-layer assisted, uncoupled, coupled and olia (Opportunistic Linked Increases Algorithm) versions of MPTCP. Olia increases good quality subflows with small windows faster while subflows with maximum windows increase slower. In other words, Olia re-forwards traffic from fully used paths to paths that have free capacity [21]. Results show CLA-MPTCP quickly adapts to network conditions and proactively offloads capacity from the Wi-Fi subflow undergoing a handoff to the LTE subflow. CLA-MPTCP's ability to preemptively tune subflow congestion windows to the adjusted path BDP ensures throughput is improved, hence service continuity is maintained while conditions of the newly connected access point after handoff are quickly learned and utilized earlier than other other schemes. The data offloaded onto the LTE subflow increases aggregate throughput achieving a nearly 50% improvement near handovers. Drastic RTT improvements are shown in Figure 6d. Since MPTCP's default congestion control algorithm relies on a path's RTT when striping an input stream among the available subflows it does not react fast enough to sudden handovers across a mobile user's path. The RTT of a subflow at the point of handoff may still be less than the RTT on other subflows causing MPTCP to still prefer and cling to it even when undergoing a handoff. Finally, a 30% improvement in the number of packet re-transmissions and duplicate acknowledgments experienced by the Wi-Fi subflow around handovers is shown in Figure 6b.

VI. Conclusion

In this paper, issues dealing with handovers in HetNets were discussed. Specifically weaknesses of transport layer protocols in mobile scenarios where multiple handoffs are imminent. A series of real world experiments in current deployed networks were conducted exploiting traditional TCP in mobile scenarios. MPTCP as a potential solution to the handoff issue is discussed. Possible drawbacks of MPTCP in highly mobile scenarios are presented. An additional MPTCP design goal is suggested emphasizing the need for cross-layer assistance. Finally, Cross-Layer Assisted MPTCP (CLA-MPTCP) is designed and evaluated. Results show the benefits of cross-layer assistance as MPTCP quickly adapts to network conditions and outperforms uncoupled, coupled and olia congestion control algorithms in scenarios where multiple handovers take place.

References

[1] B. Hamdaoui and P. Ramanathan, "Cross-layer optimized conditions for QoS support in multi-hop wireless networks with MIMO links," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 4, 2007.

[2] W. Su, A. K. Sadek, and K. R. Liu, "Cooperative communication protocols in wireless networks: performance analysis and optimum power allocation," *Wireless Personal Communications*, vol. 44, no. 2, pp. 181–217, 2008.

[3] N. Chakchouk and B. Hamdaoui, "Uplink performance characterization and analysis of two-tier femtocell networks," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 9, pp. 4057–4068, 2012.

[4] Y. Belkasmi, A. Kobbane, M. Koutbi, and J. Ben-Othman, "Channel allocation strategies in opportunistic-based cognitive networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*. IEEE, 2012, pp. 872–877.

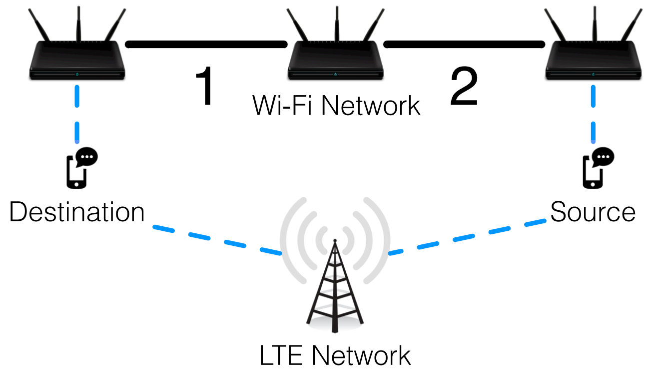
[5] P. Venkatraman, B. Hamdaoui, and M. Guizani, "Opportunistic bandwidth sharing through reinforcement learning," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 6, pp. 3148–3153, 2010.

[6] D. Dechene and A. Shami, "Energy-aware resource allocation strategies for lte uplink with synchronous harq constraints," *IEEE Transactions on Mobile Computing*, no. 99, pp. 1–1, 2014.

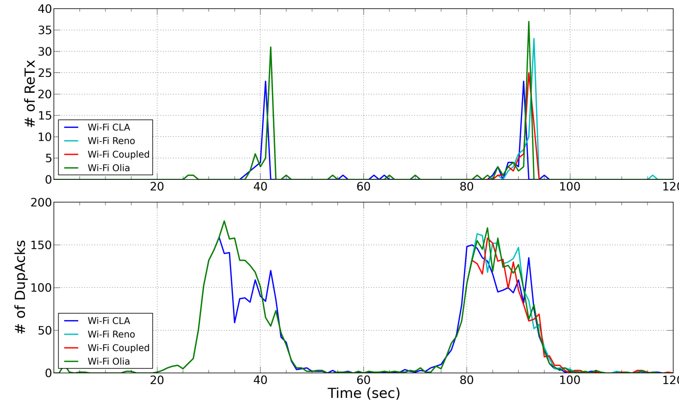
[7] B. Hamdaoui and P. Ramanathan, "A network-layer soft handoff approach for mobile wireless IP-based systems," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 4, pp. 630–642, 2004.

[8] M. Masud, F. Anwar, and S. Bari, "Enhanced handoff latency reduction mechanism in layer 2 and layer 3 of mobile ipv6 (mipv6) network." *Journal of Mathematics & Technology*, vol. 4, no. 1, 2013.

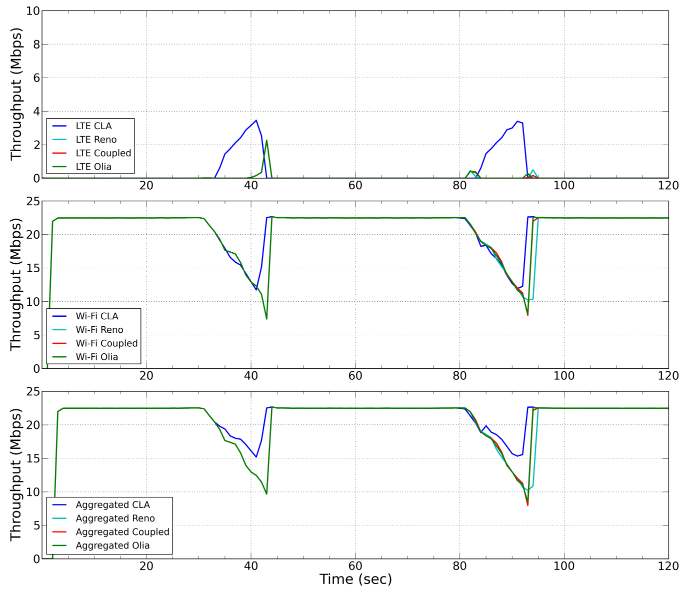
[9] H. Sinky and B. Hamdaoui, "Cross-layer assisted tcp for seamless handoff in heterogeneous mobile wireless systems," in *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2013, pp. 4982–4987.



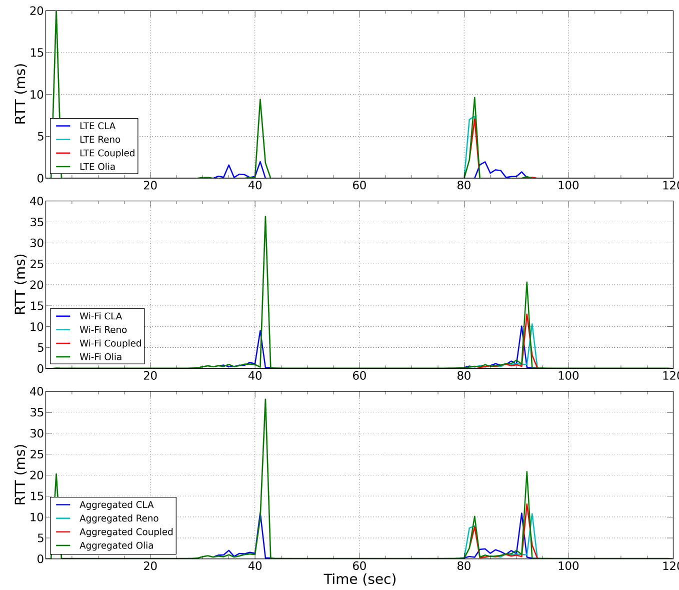
(a) NS3-DCE Network Topology



(b) Duplicate ACKs and Re-transmissions



(c) Throughput performance



(d) RTT performance

Figure 6: CLA-MPTCP Performance compared to Uncoupled Reno, Coupled, and Olia

[10] W. Hansmann and M. Frank, "On things to happen during a tcp handover," in *Proceedings of the 28th LCN Conference*. IEEE Computer Society, 2003.

[11] O. Bonaventure, M. Handley, and C. Raiciu, "An overview of multipath tcp," October 2012.

[12] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath tcp," in *SIGCOMM '12 CellNet*. New York, NY, USA: ACM, 2012, pp. 31–36.

[13] *http://www.multipath-tcp.org/*, 2014.

[14] 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*, ser. NSDI'12, Berkeley, CA, USA, 2012.

[15] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, *https://tools.ietf.org/html/rfc6824*, 2013.

[16] C. Raiciu, M. Handley, and D. Wischik, *http://tools.ietf.org/html/rfc6356*, 2011.

[17] Y.-C. Chen and D. Towsley, "On bufferbloat and delay analysis of multipath tcp in wireless networks," in *International Federation for Information Processing IFIP, Norway*, 2014.

[18] K. Chen, Y. Xue, S. H. Shah, and K. Nahrstedt, "Understanding bandwidth-delay product in mobile ad hoc networks," *Comput. Commun.*, 2004.

[19] "Ieee std 802.11k-2008 (amendment to ieee std 802.11-2007)," June 2008, pp. 1–244.

[20] D. Camara, H. Tazaki, E. Mancini, T. Turletti, W. Dabbous, and M. Lacage, "DCE: Test the real code of your protocols and applications over simulated networks," 2014.

[21] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, "Mptcp is not pareto-optimal: Performance issues and a possible solution," vol. 21, no. 5, Oct 2013, pp. 1651–1665.

[22] 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 Proc. IEEE*, April 2014, pp. 1815–1823.

[23] 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 *IFIP Conference*, ser. WD'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 107–111.

[24] S. F. Chien, H. Liu, A. L. Y. Low, C. Maciocco, and Y. L. Ho, "Smart Predictive Trigger for Effective Handover in Wireless Networks." *ICC*, pp. 2175–2181, 2008.

[25] W. Guan, X. Ling, X. S. Shen, and D. Zhao, "Handoff trigger table for integrated 3G/WLAN networks," in *IWCMC '06*, Jul. 2006.

[26] C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath TCP," in *SIGCOMM '11*. ACM Request Perm., Aug. 2011.