Multi-Path TCP with Network Coding for Mobile Devices in Heterogeneous Networks

Jason Cloud†*, Flávio du Pin Calmon†*, Weifei Zeng†, Giovanni Pau‡, Linda M. Zeger**, and Muriel Médard†

†Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA

‡Computer Science Department, University of California, Los Angeles, CA

*MIT Lincoln Laboratory, Lexington, MA

**Auroral LLC

Email: {jcloud, flavio, weifei}@mit.edu, gpau@cs.ucla.edu, zeger@auroral.biz, medard@mit.edu

Abstract—Existing mobile devices have the capability to use multiple network technologies simultaneously to help increase performance; but they rarely, if at all, effectively use these technologies in parallel. We first present empirical data to help understand the mobile environment when three heterogeneous networks are available to the mobile device (i.e., a WiFi network, WiMax network, and an Iridium satellite network). We then propose a reliable, multi-path protocol called Multi-Path TCP with Network Coding (MPTCP/NC) that utilizes each of these networks in parallel. An analytical model is developed and a mean-field approximation is derived that gives an estimate of the protocol's achievable throughput. Finally, a comparison between MPTCP and MPTCP/NC is presented using both the empirical data and mean-field approximation. Our results show that network coding can provide users in mobile environments a higher quality of service by enabling the use of multiple network technologies and the capability to overcome packet losses due to lossy, wireless network connections.

I. Introduction

Simultaneous use of multiple network interfaces on a single mobile device has the potential to increase quality of service, seamlessly offload traffic from expensive networks to cheaper ones, increase session reliability, etc.; yet current technology does not utilize the available resources efficiently to meet these objectives. Instead, only a single network interface is preferred while the others are left unused. For example, consider a standard smart phone that has a cellular data connection, such as 3G or LTE, and a WiFi connection. Data is sent over either one or the other, but not both. Given that existing infrastructure currently supports the use of both WiFi and cellular technologies ([1], [2]), new techniques must be developed to properly leverage all available resources, regardless of their quality, to increase mobile user performance.

A significant amount of research has been performed that attempts to utilize these heterogeneous network connections. For example, Multi-Path TCP (MPTCP) is a new protocol currently in the working group level of the IETF [3]. The protocol adds a new layer above the transport layer which provides packet scheduling across multiple TCP sub-flows and guarantees packet delivery through the use of a somewhat complex management scheme. Furthermore, MPTCP uses TCP as its primary flow control mechanism on each of the sub-flows. While the use of TCP ensures fairness with other TCP flows, the performance of TCP over lossy networks (e.g., wireless networks) is known to be poor [4]. Network coding is one possible solution that both reduces the need for a complex management scheme and can increase TCP's performance over lossy networks.

Several suggestions on how to incorporate network coding with MPTCP have been proposed. Gheorgiu *et. al.* [5] propose a protocol called CoMP that uses network coding for multi-path transmission that incorporates only some aspects of TCP. [6] and [7] add a multi-path scheduler below the TCP, network coding, and IP layers negating the congestion control benefits of TCP over single paths.

Finally, ParandehGheibi *et. al.* [8], and implemented by [9] in OpNet, provide a sub-flow selection control policy for network coded packets over heterogeneous networks that optimizes the trade-offs between the network usage costs and the Quality of user Experience (QoE) for media-streaming applications. Many approaches have also been proposed to increase TCP's performance in the presence of high losses ([10], [11], [12], [13] to name a few). One promising approach is TCP/NC proposed by Sundararajan *et. al.*, [14]. TCP/NC introduces a layer between TCP and IP that uses random linear network coding [15] to produce linear combinations of all packets contained in the TCP congestion control window. These coded packets are then transmitted over the network and decoded by a client. As shown in [14], network coding helps to alleviate the effects of packet loss due to poor channels while preserving the congestion control and fairness mechanisms provided by TCP.

In this paper, we first present empirical measurements for the simultaneous use of three heterogeneous network connections (e.g., WiFi, WiMax, and an Iridium satellite network) in a mobile environment. These measurements highlight the fact that none of the networks provide 100% reliable communication; but in combination, the simultaneous use of these networks can provide significant gains over the use of only one at a time. We then present a model based on [16] and [17], as well as derive a mean-field approximation for the throughput of both MPTCP and MPTCP with network coding (MPTCP/NC). MPTCP/NC uses network coding prior to packet sub-flow scheduling to simplify the MPTCP management scheme, and uses network coding a second time below TCP to provide a mechanism to overcome packet losses induced by lossy, wireless networks. We conclude with a comparison of MPTCP and MPTCP/NC using both the mean-field approximation and the experimentally collected data to show that network coding can provide beneficial enhancements to the existing protocols.

The remainder of the paper is organized as follows. In Section II, we outline the experimental measurement setup and present an overview of the collected data. Section III provides a brief description of MPTCP and MPTCP/NC, develops the analytical models used, and derives the mean-field approximations for both protocols. We then compare the performance of both MPTCP and MPTCP/NC in Section IV and conclude in Section V.

II. Empirical Measurements

Using a WiMax base station, a WiFi mesh network, and an Iridium satellite data modem [18], simultaneous network traces were collected between the Network Research Laboratory (NRL), Department of Computer Science, UCLA and a vehicle driving a fixed route around the UCLA campus. Each experiment sent packets, varying between 64 bytes, 512 bytes , and 1,350 bytes in size, at rates based on the direction of travel. For example, traffic generated by the computer in the NRL and sent to the vehicle, referred to as downlink (D/L) traffic, was sent at rates determined by the individual network (WiMax: 20 Mbps, WiFi: 20 Mbps, and Iridium: 1 kbps). Traffic generated by

(a) Network Configuration (b) WiMax/WiFi base station placement and vehicle route.
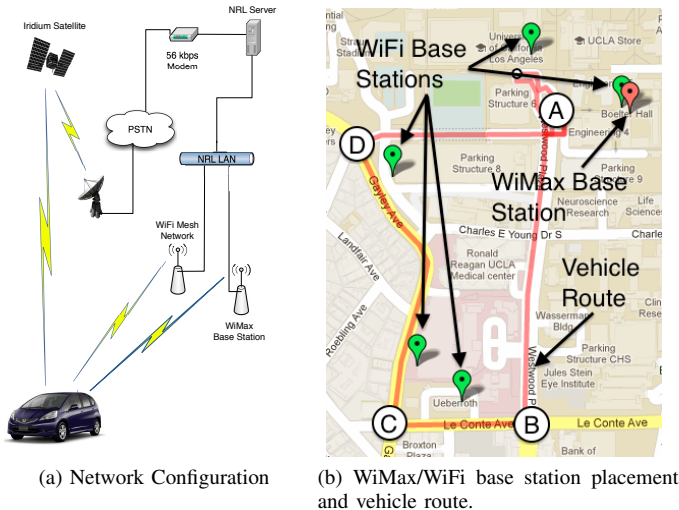
Fig. 1: Empirical measurement collection setup.

the computer in the vehicle and sent to the computer in the NRL, referred to as uplink (U/L) traffic, was also sent at rates determined by the individual network (WiMax: 1 Mbps, WiFi: 20 Mbps, and Iridium: 1 kbps). In each experiment, only D/L traffic or U/L traffic was generated.

A. Testbed Configuration

Measurements were taken between a mobile commodity laptop and a fixed server located within the NRL. The computer in the NRL was connected to the NRL LAN which has gateways to both the WiMax base-station and WiFi mesh network. A 56 kbps modem was used to connect the computer to the public switched telephone network (PSTN) in order to utilize the Iridium satellite network. In the vehicle, a single computer with separate WiMax and WiFi cards, as well as a connection to an Iridium data modem, was used to transmit and receive data. A diagram of the setup is shown in Figure 1(a). UDP network traffic was generated using Iperf [19] and network traces were collected using tshark (a command line version of Wireshark) [20] on both the computers.

The vehicle containing the mobile computer travelled a fixed route through the UCLA campus chosen so that the vehicle passed in and out of the coverage areas of all three networks. For example, connections through all three networks was established prior to each experiment. The vehicle would then drop from and reconnect to each of the individual networks, depending on the location of the vehicle and coverage of the specific network, throughout the duration of the experiment. Figure 1(b) provides the vehicle route and placement of the WiMax and WiFi mesh base stations on the UCLA campus.

B. Collected Data

Ten mobile experiments were conducted over a period of five days in August 2011. For each of these experiments, traces were collected and compared for each of the different networks. A sample of the collected traces are shown in Figure 2. These traces show the UDP throughput for each network when all traffic is sent either to (D/L) or from (U/L) the vehicle.

The round-trip time (RTT) and packet loss probability for each network was also collected. The CDFs for both the RTT and packet loss probability during the D/L experiment where 1,350 byte packets were used is shown in Figure 3. The RTT was measured using ping messages that were sent throughout the experiment on both the WiFi and WiMax networks, while ping messages were only sent for approximately 60 seconds at the beginning of the experiment on the Iridium network due to the bandwidth constraints of the network. The
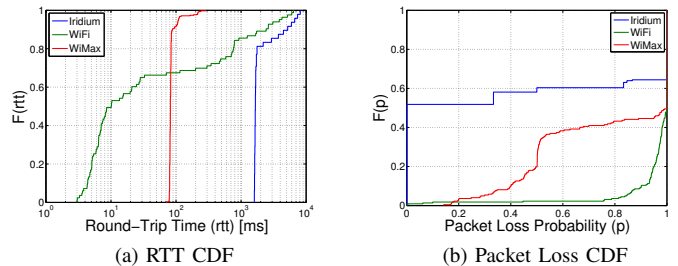

(a) RTT CDF (b) Packet Loss CDF

Fig. 3: CDFs of the RTT and packet loss probabilities during the D/L experiment using 1,350 byte packets.

packet loss probabilities were determined by comparing the trace files on both the NRL server and the vehicle computer.

C. Discussion and Comments on the Experimental Results

Data collected during each of the experiments provides information about the expected environment that mobile users are likely to experience. However, there are caveats concerning the methods in which the data was collected that must be noted. First, the only user on each network was the vehicle; although the WiFi mesh network performance was affected by significant interference from adjacent WiFi networks and the Iridium network was setup over an operational system. As a result, the WiMax throughput presented in Figure 2 is much larger than what would be expected when fully loaded, while the WiFi and Iridium throughput is close to what we would expect in the real-world. The RTT shown in Figure 3(a) is also affected by this situation. Since only one user has access to the WiMax network, the RTT is fairly consistent throughout each of the experiments. The WiFi RTT is largely affected by contention with adjacent WiFi networks resulting in a wide range of possible RTTs, and the Iridium RTT is consistent except for periods where we believe horizontal handoffs between satellites occurred. Second, the data collection methods were designed so that data could be used to replay each experiment off-line enabling easy evaluation of future protocol designs. This prevented us from collecting reliable statistics on the packet loss probabilities. However, Figure 3(b) shows the overall reliability of each network and indicates that the satellite network provides the most reliability and the WiFi network provides the least. Finally, the use of a modem and the PSTN for the Iridium network (and consequently the low throughput) is due to the Iridium system design. Iridium was developed for world-wide voice communications. Modern satellite systems do provide higher bandwidth, and therefore better performance for packet based communication. Unfortunately, the use of these systems was prohibitively expensive.

Regardless, the traces shown in Figure 2 indicate that using a multi-path solution can potentially provide significant performance gains over that of using only one of the networks exclusively. Throughout each experiment, the vehicle was connected to at least one network the majority of the time; and in many cases, it was connected to two or more networks. Leveraging this connectivity can help ensure that reliable, continuous data transport is an option in mobile environments. The benefits of leveraging simultaneous networks for data transport will be quantified in the following sections using the collected data. Specifically, packet loss and RTT statistics will be used to provide a comparison between the performance of MPTCP and MPTCP/NC in multi-path, wireless scenarios.

III. ANALYTICAL MODELS FOR MULTI-PATH TCP AND MULTI-PATH TCP WITH NETWORK CODING

Approaches similar to that of [16] and [17] are used to provide a mean-field approximation of the throughput for both MPTCP and MPTCP/NC. The MPTCP analysis will assume the standard

(a) U/L, Packet Size: 512B   (b) U/L, Packet Size: 1350B   (c) D/L, Packet Size: 64B   (d) D/L, Packet Size: 1350B
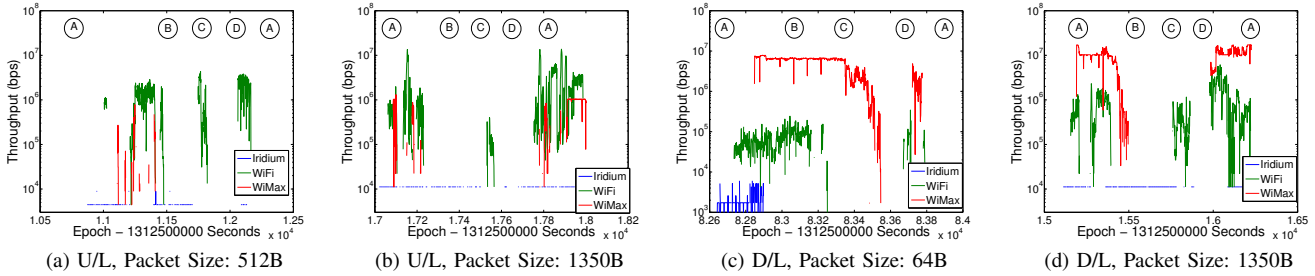
Fig. 2: Sample traces showing the UDP throughput for two U/L and two D/L experiments with varying packet sizes. The labels A, B, C, and D provide the approximate location of the vehicle when compared with Figure 1(b).
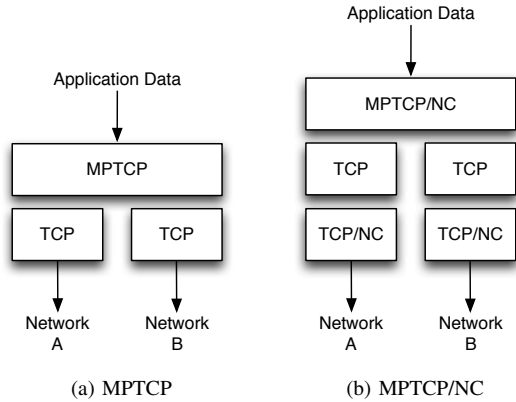

(a) MPTCP   (b) MPTCP/NC

Fig. 4: Assumed network stack configuration for both MPTCP and MPTCP/NC.

implementation as shown in Figure 4(a) and defined by [3]. The MPTCP/NC analysis will assume that the MPTCP/NC layer shown in Figure 4(b) provides a first layer of network coding before packets are injected into a TCP sub-flow, and the TCP/NC layer provides a second layer of network coding, similar to [14], in order to overcome random packet losses due to lossy networks.

The analysis for MPTCP will use the model presented by [16] while assuming that perfect scheduling of packets across various TCP sub-flows takes place. Once the analytical throughput for each individual sub-flow is determined, the results can be summed to determine MPTCP's overall throughput. In reality, perfect scheduling is not possible due to packet losses, termination of a specific sub-flow, etc. This, in-turn, results in the need to collect feedback regarding which packets were lost, retransmit each lost packet on a second (or third) TCP sub-flow, and verify receipt of that packet by the receiver. This process significantly decreases the efficiency of MPTCP by both lowering the throughput and increasing the transport time. With this in mind, the analytical results presented later will over estimate the performance of MPTCP.

In the case of MPTCP/NC, network coding can be used to aid in the sub-flow scheduling problem by eliminating the need to track specific packets sent over the network. With respect to the analysis, we will assume that network coding is performed prior to a packet's injection into a sub-flow. If the coding operations are carried out properly, the receiver only needs to collect enough coded packets, or degrees of freedom (DOF), in order to successfully transfer data over multiple sub-flows without the need to track individual packets through the multiple networks. Not only does this significantly decrease the complexity of the protocol, but also provides greater freedom for determining how to allocate packets among the collection of sub-flows. We will also assume that a second layer of network coding occurs below TCP and redundant packets are transmitted to overcome

random packet losses. In general, the number of transmitted packets for every DOF sent should be $R \geq 1/1-p$ where R is the redundancy and p is the packet loss probability of the network path. [14] provides a full description of the network coding operations and gains that can be achieved using network coding in this manner.

Finally, we will assume that both protocols use a TCP Reno style of congestion control on each sub-flow. This assumption keeps the results presented here in line with those presented by [16] and also simplifies the analysis for MPTCP/NC. Because we assume that network coding is performed below TCP on each sub-flow, network coding eliminates the need to consider the effects of triple-duplicates on TCP's window size. A more detailed discussion will be provided in subsequent sections.

A. MPTCP Analytical Throughput

The analytical throughput for MPTCP follows directly from [16]. In [16], the analytical throughput, $B(p)$, of a single TCP connection was derived where p is the independent and identically distributed (i.i.d) loss probability of a single packet. The equation for $B(p)$ can be found in equation (32) of [16]. We extend this analysis to the multi-path case by taking the calculated $B_j(p_j)$ for each sub-flow, $j = \{1, \ldots, n\}$, and summing them together to form the MPTCP throughput:

$$B(p_1, \ldots, p_n) = \sum_{j=1}^{n} B_j(p_j). \tag{1}$$

As noted earlier, this does not let us take into account the inefficiencies introduced by the MPTCP layer and will over-estimate the achievable MPTCP throughput.

B. Modeling MPTCP/NC's End-to-End Throughput

Two metrics will be used to develop the MPTCP/NC mean-field approximation: the average throughput $\mathcal{T}$, and the expected MPTCP/NC congestion window evolution $\mathbb{E}[W]$. We model MPTCP/NC's behavior in terms of rounds. The natural choice for determining the duration of a round is to use the RTT from the sender to the receiver (i.e., $t_{rnd} = RTT$). While this works if there is a single TCP connection, each sub-flow is expected to have different round trip times making it difficult to determine which RTT to use. This is accounted for by setting the duration of each round, t_{rnd}, equal to the greatest common divisor (GCD) of the sub-flows' RTTs. Figure 5 provides an illustration of this concept.

1) MPTCP/NC Sub-Flow Analysis: We now use the most basic implementation of TCP in our analysis and initially assume that each round's duration is equal to the RTT of sub-flow j. We assume that the congestion window size during round i is determined by the number of acknowledgements a indicating successfully transmitted packets obtained during round $i-1$:

$$W_i^{(j)} = W_{i-1}^{(j)} + \frac{a_{i-1}^{(j)}}{W_{i-1}^{(j)}}. \tag{2}$$
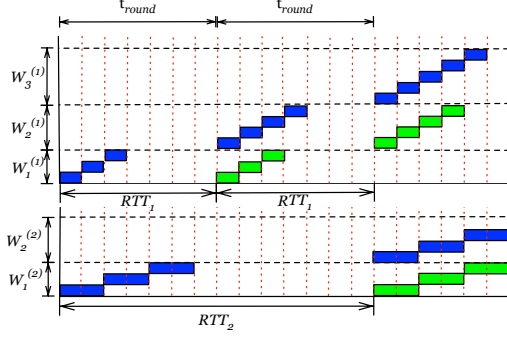
Fig. 5: Round duration used for MPTCP/NC for two sub-flows. The blue blocks indicate packets and the green blocks indicate acknowledgements.

This concept is also shown in Figure 5 where the congestion window size of each sub-flow grows as a function of the number of acknowledgements received. We now assume that R_j linearly independent, redundant, network coded packets are sent for each uncoded packet contained the TCP congestion window, there are i.i.d. packet losses, and a packet loss rate of p_j. Taking the expectation of the window size, $\mathbb{E}[W_i^{(j)}]$, we obtain:

$$\mathbb{E}\left[W_i^{(j)}\right] = \mathbb{E}\left[W_{i-1}^{(j)}\right] + \min\left(1, (1-p_j)R_j\right) \quad (3)$$

$$= \mathbb{E}\left[W_1^{(j)}\right] + (i-1)\min\left(1, (1-p_j)R_j\right), \quad (4)$$

where the minimization is required because the window size can only increase by a maximum of one packet per round.

Since the throughput $\mathcal{T}_i^{(j)}$ per round is related to the number of packets sent in that round,

$$\mathcal{T}_i^{(j)} = \frac{\mathbb{E}\left[W_i^{(j)}\right]}{RTT_j}\min\left(1, (1-p_j)R_j\right). \quad (5)$$

The minimization in this equation is necessary to account for packets that are received that do not deliver new degrees of freedom. Since the TCP/NC layer codes all packets within the TCP congestion window, delivered packets 1 through $W_i^{(j)}$ contain new degrees of freedom. If more than $W_i^{(j)}$ packets are received in the round, the MPTCP/NC layer disregards them since they contain no new information.

The above analysis assumed that the RTTs for each sub-flow was the same. Because this is not necessarily the case, we must adjust equation (4) to account for the shorter round durations by defining $\alpha_j = RTT_j/t_{rnd}$ and substituting $\lceil i/\alpha_j \rceil$ for i,

$$\mathbb{E}\left[W_{\lceil i/\alpha_j \rceil}^{(j)}\right] = \mathbb{E}\left[W_1^{(j)}\right] + (\lceil i/\alpha_j \rceil - 1)\min\left(1, (1-p_j)R_j\right)$$

$$= \gamma_i^{(j)}. \quad (6)$$

The throughput for each TCP sub-flow j then becomes $\mathcal{T}_i^{(j)} = \gamma_i^{(j)}/(\alpha_j \cdot t_{rnd})\min\left(1, (1-p_j)R_j\right)$, which can be further reduced if we consider a large enough redundancy factor R_j. For $R_j > 1/(1-p_j)$, the instantaneous throughput becomes,

$$\mathcal{T}_i^{(j)} = \frac{1}{\alpha_j \cdot t_{rnd}}\left(\mathbb{E}\left[W_1^{(j)}\right] + \lceil i/\alpha_j \rceil - 1\right). \quad (7)$$

Finally, we account for the fact that the number of packets sent in each RTT is upper-bounded by TCP's maximum congestion window size, $W_{\max}^{(j)}$. This results in:

$$\mathcal{T}_i^{(j)} = \frac{1}{\alpha_j \cdot t_{rnd}}\left(\min\left(W_{\max}^{(j)}, \mathbb{E}\left[W_1^{(j)}\right] + \lceil i/\alpha_j \rceil - 1\right)\right). \quad (8)$$

The model we used in our analysis of the MPTCP/NC sub-flow performance makes several assumptions that, in practice, should be considered. First, we assume that packet losses are i.i.d. with loss probability p_i. Therefore, the analysis does not account for correlated packet losses due to congestion and other factors. Second, we assumed that R_j is sufficiently large enough to ignore the possibility of time-outs. While the probability of a time-out decreases with increasing R_j, time-outs still occur in practice and the impact of each time-out on the throughput is significant (i.e., the congestion window size is reset to $\mathbb{E}[W_1^{(j)}]$). Specifically, a time-out occurs when the sum of received acknowledgements over two rounds, i and $i+1$, is less than the window size during round i with probability $Pr(a_i + a_{i+1} < W_i)$. Generalizing equation (2) to account for time-outs, with respect to $W_{\max}$ and R_j, may allow for a bound on the decrease in throughput to be determined resulting in a more accurate approximation. Third, we assume that RTT_j remains constant. In practice, this is not true, and implementations of TCP generally use an averaged round-trip time often referred to as the "smoothed" round-trip time $SRTT$.

2) MPTCP/NC's Window Evolution and End-to-End Throughput: Using the above results, the average end-to-end MPTCP/NC throughput over k rounds is determined using a round duration of t_{rnd} and defining $\alpha_j = RTT_j/t_{rnd}$,

$$\mathcal{T}(k) = \frac{1}{k}\sum_{i=1}^{k}\mathcal{T}_i = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{n}\mathcal{T}_i^{(j)}. \quad (9)$$

Assuming that $k/\alpha_j \in \mathbb{Z}$, $\gamma_i^{(j)} \leq W_{max}^{(j)}$, and relaxing $\lceil i/\alpha_j \rceil$ so that it is i/α_j for all j,

$$\mathcal{T}(k) = \frac{1}{k}\sum_{j=1}^{n}\left(\frac{1}{\alpha_j \cdot t_{rnd}}\sum_{i=1}^{k}\gamma_i^{(j)}\right) \quad (10)$$

$$= \frac{1}{t_{rnd}}\sum_{j=1}^{n}\left(\frac{1}{\alpha_j}\mathbb{E}\left[W_1^{(j)}\right] + \frac{k+1}{2\alpha_j^2} - \frac{1}{\alpha_j}\right), \quad (11)$$

If $k/\alpha_j \notin \mathbb{Z}, \forall j$, the above equation will contain additional terms that contain packets sent in the rounds from $\lfloor k/\alpha_j \rfloor$ to k/α_j. Furthermore, the relaxation of $\lceil i/\alpha_j \rceil$ to i/α_j decreases the throughput since we are no longer accounting for $\lceil i/\alpha_j \rceil - i/\alpha_j$ packets sent per round. As k grows, these approximations have less of an effect on the throughput.

Finally, we take into account the maximum window size of each sub-flow $W_{\max}^{(j)}$; but first, we define:

$$r^{(j)} = \alpha_j\left(W_{\max}^{(j)} - \mathbb{E}\left[W_1^{(j)}\right]\right). \quad (12)$$

Using equation (11) and assuming that $R_j > 1/(1-p_j)$, the average end-to-end throughput $\mathcal{T}_{e2e}$, in packets per second is:

$$\mathcal{T}_{e2e}(k) = \sum_{j=1}^{n}\mathcal{T}_{e2e}^{(j)}(k), \quad (13)$$

where

$$\mathcal{T}_{e2e}^{(j)}(k) = \begin{cases} \frac{1}{\alpha_j \cdot t_{rnd}}\left(\mathbb{E}\left[W_1^{(j)}\right] + \frac{k+1}{2\alpha_j} - 1\right) & \text{for } k \leq r^{(j)}, \\ \frac{\rho^{(j)}}{\alpha_j \cdot k \cdot t_{rnd}} & \text{for } k > r^{(j)}, \end{cases} \quad (14)$$

and

$$\rho^{(j)} = r^{(j)}\mathbb{E}\left[W_1^{(j)}\right] + \frac{r^{(j)}\left(r^{(j)} + 1 - 2\alpha_j\right)}{2\alpha_j} + W_{\max}^{(j)}\left(k - r^{(j)}\right). \quad (15)$$

It should be noted that as $k \to \infty$ for $R_j > 1/(1-p_j)$, the average end-to-end throughput $T_{e2e}(k) \to \sum_{j=1}^{n} W_{\max}^{(j)}/(\alpha_j \cdot t_{tnd})$.

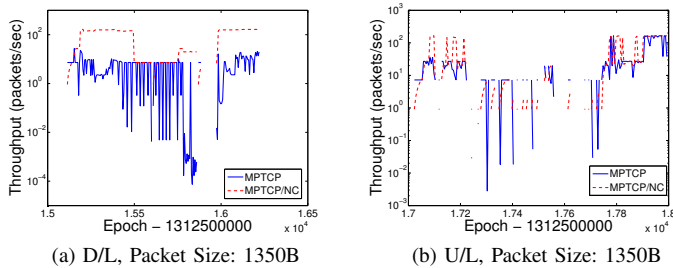(a) D/L, Packet Size: 1350B   (b) U/L, Packet Size: 1350B

Fig. 6: Comparison of the theoretical MPTCP and MPTCP/NC throughput using the data presented in Section II.

IV. COMPARISON OF MULTI-PATH TCP AND MULTI-PATH TCP WITH NETWORK CODING USING EMPIRICAL DATA

We now compare the theoretical throughput of MPTCP, equation (1), with that of the theoretical throughput of MPTCP/NC, equation (13). Figure 6 shows the performance of both protocols using the data presented in Section II as a baseline. The maximum window size for each TCP sub-flow was set to $W_{max} = 12$; and a mean RTT, based off of empirical data, was used for each network where $RTT_{Iridium} = 1.653s$, $RTT_{WiFi} = 0.607s$, and $RTT_{WiMax} = 0.087s$. Empirical packet loss data, averaged over 5s, on each separate path from two of the experiments was used as a baseline for determining both throughputs. It was assumed that the network capacity for each network was large enough to send W_{max} packets in the case of MPTCP and $R_j W_{max}$ packets in the case of MPTCP/NC where R_j is assumed to be large enough so that time-outs are very unlikely (i.e., R_j is much larger than the 5s average of the packet loss probability). In addition, Figure 6 uses the mean-field approximations developed in the last section and does not show a simulated behavior of each protocol.

The figures show that MPTCP/NC provides a better throughput throughout the "simulated" experiment than MPTCP. While MPTCP is severely hindered by high packet losses as a result of poor channel conditions, MPTCP/NC is able to mask the majority of packet losses and maintain a high throughput. Scheduling of packets on each sub-flow is also easier with MPTCP/NC than with MPTCP due to the network coding operations performed immediately below the application layer. The throughput shown for MPTCP assumes that there is perfect scheduling among the sub-flows with no need to retransmit a packet on more than one sub-flow. This provides a best-case scenario for the achievable throughput. This assumption is not made in MPTCP/NC because each packet transmitted on a sub-flow is viewed as a degree of freedom. If a packet is lost, any packet sent on a different sub-flow can be used in the lost packet's place.

V. CONCLUSION

We have presented empirical measurements for the simultaneous use of three heterogeneous networks showing that the combined use of all three is needed in order to provide an improved level of performance in mobile environments. We then suggested the use of a multi-path protocol based on MPTCP that uses network coding to overcome the challenges of packet scheduling and lossy wireless networks. A mean-field approximation of the throughput for both MPTCP and MPTCP/NC was developed and used, along with the empirical data, to provide a comparison of the two protocols. This comparison showed that the use of network coding in multi-path, lossy scenarios can significantly increase the quality of service for mobile users.

REFERENCES

[1] X. Chen, R. Jin, K. Suh, B. Wang, and W. Wei, "Network performance of smart mobile handhelds in a university campus WiFi network," in *Proc. ACM Conference on Internet Measurement Conference (IMC)*, pp. 315–328, 2012.

[2] J. Sommers and P. Barford, "Cell vs. WiFi: on the performance of metro area mobile connections," in *Proc. ACM Conference on Internet Measurement Conference (IMC)*, pp. 301–314, 2012.

[3] "MPTCP IETF working group." https://datatracker.ietf.org/wg/mptcp/.

[4] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saaranen, "TCP performance issues over wireless links," *IEEE Communications Magazine*, vol. 39, no. 4, pp. 52–58, 2001.

[5] S. Gheorghiu, A. L. Toledo, and P. Rodriguez, "Multipath TCP with network coding for wireless mesh networks," in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1–5, 2010.

[6] X. Zhuoqun, C. Zhigang, Y. Hui, and Z. Ming, "An improved MPTCP in coded wireless mesh networks," in *Proc. IEEE International Conference on Broadband Network & Multimedia Technology (IC-BNMT)*, pp. 795–799, 2009.

[7] Z. qun Xia, Z. gang Chen, Z. Ming, and J. qi Liu, "A multipath TCP based on network coding in wireless mesh networks," in *Proc. IEEE International Conference on Information Science and Engineering (ICISE)*, pp. 3946–3950, 2009.

[8] A. ParandehGheibi, M. Médard, A. Ozdaglar, and S. Shakkottai, "Access-network association policies for media streaming in heterogeneous environments," in *CDC'10*, pp. 960–965, 2010.

[9] A. Kulkarni, M. Heindlmaier, D. Traskov, M.-J. Montpetit, and M. Médard, "An implementation of network coding with association policies in heterogeneous networks," in *Proc. IFIP TC International Conference on Networking (NETWORKING'11)*, pp. 110–118, 2011.

[10] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1300–1315, 2001.

[11] R. Caceres and L. Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 5, pp. 850–857, 1995.

[12] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 4, pp. 747–756, 2004.

[13] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback-based scheme for improving TCP performance in ad hoc wireless networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 34–39, 2001.

[14] J. K. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network coding meets TCP: theory and implementation," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 490–512, 2011.

[15] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[16] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP reno performance: a simple model and its empirical validation," *IEEE/ACM Trans. Netw.*, vol. 8, no. 2, pp. 133–145, 2000.

[17] M. Kim, M. Médard, and J. a. Barros, "Modeling network coded TCP throughput: a simple model and its validation," in *Proc. ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS)*, pp. 131–140, 2011.

[18] "Iridium 9522A satellite transceiver." http://www.iridium.com/products/Iridium-9522A-Satellite-Transceiver.aspx?productCategoryID=2.

[19] "Iperf." http://sourceforge.net/projects/iperf/.

[20] "tshark." http://www.wireshark.org.