Design, Implementation, and Evaluation of Energy-Aware Multi-Path TCP

Yeon-sup Lim[1], Yung-Chih Chen[1], Erich M. Nahum[2], Don Towsley[1], Richard J. Gibbens[3], and Emmanuel Cecchet[1]

[1]College of Computer and Information Sciences, University of Massachusetts Amherst, MA, USA
[2]IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA
[3]Computer Laboratory, University of Cambridge, UK

{ylim, yungchih, towsley, cecchet}@cs.umass.edu[1], nahum@us.ibm.com[2], richard.gibbens@cl.cam.ac.uk[3]

ABSTRACT

Multi-Path TCP (MPTCP) is a new transport protocol that enables systems to exploit available paths through multiple network interfaces. MPTCP is particularly useful for mobile devices, which usually have multiple wireless interfaces. However, these devices have limited power capacity and thus judicious use of these interfaces is required.

In this work, we design, implement, and evaluate an energy-aware variant called *eMPTCP*, which seeks to reduce power consumption compared to standard MPTCP, with minimal impact on download latency. eMPTCP uses a combination of power-aware subflow management and delayed subflow establishment to accomplish its goals. Power-aware subflow management allows eMPTCP to choose paths dynamically to maximize per-byte energy efficiency, using runtime measurements and a parameterized energy consumption model that accounts for multiple interfaces. Delayed subflow establishment lets eMPTCP avoid heavy power consumptions in cellular interfaces for small transfers.

We implement eMPTCP on Android mobile devices and evaluate it across several scenarios, both in the lab and in the wild. We measure both energy consumption and download times, varying network bandwidth, background traffic, user mobility, client and server location, and download size. Our results show that eMPTCP reduces power consumption compared to MPTCP by up to 90% for small file downloads and up to 50% for large file downloads.

CoNEXT'15 December 01-04, 2015, Heidelberg, Germany
© 2015 ACM. ISBN 978-1-4503-3412-9/15/12…$15.00
DOI: http://dx.doi.org/10.1145/2716281.2836115

CCS Concepts

•Networks → Transport protocols; Network experimentation; Mobile networks; Wireless local area networks;

Keywords

Multi-Path TCP; Energy Efficiency; Measurements

1. INTRODUCTION

Multi-Path TCP (MPTCP) is a new standardized transport protocol that enables end hosts to take simultaneous advantage of multiple network interfaces and utilize path diversity in the network [10, 29, 32]. MPTCP can achieve greater throughput, robustness and availability than standard single-path TCP, all while maintaining compatibility with existing applications. One natural fit for MPTCP is on mobile devices, such as smartphones, which typically include both cellular and WiFi interfaces.

Applying MPTCP to mobile devices introduces a new concern, namely, the additional energy consumed from operating multiple network interfaces. Mobile devices are frequently constrained by the amount of available battery power. Thus, power consumption is an important area of research, particularly in mobile devices such as smartphones. In many environments, standard MPTCP does not reduce power consumption compared to single-path TCP. Standard MPTCP does not consider the per-byte energy efficiency of a network path, which depends on several factors, including available bandwidth and the interface type. It also ignores the high fixed energy costs of activating cellular interfaces, known as the promotion and tail costs [1].

This paper makes the following contributions:

- We design and implement eMPTCP, a multi-path TCP for power-constrained mobile devices. The goal of eMPTCP is to improve the energy efficiency of MPTCP while having minimal impact on download latency. eMPTCP is the first MPTCP implementation that monitors path characteristics at run time and *dynamically* chooses paths based on *per-byte energy efficiency*. Energy efficiency is provided by extending a standard power model [1, 14] to include multiple interfaces [17]. It is the first MPTCP implementation that addresses high fixed cellular overheads by using *delayed subflow establishment* to avoid those overheads when possible. It requires no user intervention and no changes to applications.

- We evaluate eMPTCP in a controlled lab setting (§4), showing how it naturally chooses the most energy efficient path. We show that it is robust to changing bandwidths and interfering nodes, and dynamically adapts in response to mobility.
- We evaluate eMPTCP in the wild (§5), using a variety of client environments and server locations to vary network conditions and determine how well it performs in real-world environments. eMPTCP reduces energy consumption by up to 90% for small downloads (§5.2) and up to 50% for large downloads (§5.3). In a more representative Web-site download consisting of multiple URLs and connections, power is reduced by 40% with no reduction in download speed (§5.4)

A preliminary version of this work appeared in [17]. In this paper, we refine and enhance the preliminary approach of [17], and perform comprehensive evaluations. Other previous approaches in this area have either relied only on analysis and simulation [22, 24] or have studied a much more restricted range of operating environments [21, 28]. Our approach is implemented on the phone and requires no offloaded computation at run time in contrast to [24]. In addition, none of the above attempt to reduce cellular fixed overheads. Ours is the first broad evaluation of an implemented, deployable energy-aware MPTCP.

The remainder of this paper is organized as follows. Section 2 provides the background context for our work. We present our energy-aware MPTCP in Section 3. Section 4 evaluates the performance of eMPTCP in a controlled lab setting, while Section 5 evaluates eMPTCP in the wild. Related work is reviewed in Section 6, and we conclude in Section 7.

2. BACKGROUND

2.1 Multi-path TCP

MPTCP splits a single data stream across multiple paths known as *subflows*, which are defined logically by all end-to-end interface pairs. For example, if each host has two interfaces, an MPTCP connection consists of four subflows. These subflows are exposed to the application layer as one standard TCP connection.

The benefits of leveraging MPTCP in mobile devices are three-fold: First, by utilizing the available bandwidth of each subflow, an MPTCP connection can achieve higher throughput than a standard TCP connection [27]. Second, while connectivity in one network can degrade or disappear, MPTCP offers a seamless TCP connection by using paths (subflows) through another network [28]. Finally, the MPTCP layer is hidden from user applications by providing a standard TCP socket. Existing TCP applications need not be modified to support MPTCP [9].

MPTCP has three modes of operation to control subflow usage: Full-MPTCP, Single-Path and Backup mode [21]. Full-MPTCP is the standard mode of operation, described above, that utilizes all interfaces. In Single-Path mode, MPTCP uses only one path at a time, establishing a new subflow only after the interface of the active current subflow goes down. In Backup mode, MPTCP opens TCP subflows over all interfaces, but uses only a subset of them for packet transmission. The remaining interfaces are kept idle as backups. If a user sets a particular interface to backup mode, MPTCP sends no traffic through the associated subflows unless all other subflows break. By setting the mode of each interface, a user can manually decide path usage considering traffic pricing or battery life [21]. MPTCP currently allows manual configuration of interface usage, but does not support *automatic* use of the most energy-efficient path.

2.2 Energy in Mobile and Wireless

Due to the limited battery life of mobile devices, understanding and reducing energy consumption in mobiles has been an active area of research. Researchers have observed high power overheads in cellular networking [1, 14], which contributes to cellular interfaces being less power-efficient than WiFi interfaces [14]. Others have discovered that more energy is consumed when the signal is weak [7, 31], and that multiple access points can cause extra power loss [19]. Researchers have examined WiFi in particular, reducing power consumption via rate adaptation [16], sleeping short periods [15], using fewer antennas [11], or lowering the clock rate [18].

Techniques to improve energy efficiency have included utilizing the most efficient available interface [23, 26], deferring transmissions until the signal is strong [3, 31], or until a more power-efficient interface is available [25].

2.3 Cellular Promotion and Tail

Multiple works have identified and addressed the power overheads in cellular interfaces, frequently called the *promotion* and the *tail* [1, 14]. The 3rd Generation Partnership Project (3GPP) standard defines a state machine for cellular interfaces, and describes the possible power states of each device connected to the network. Idle interfaces typically remain in a low-power state to save energy. Before a packet can be sent or received, an idle interface must switch from a low power state to a high power one, which takes additional time, called the *promotion*. After a transmission is complete, however, a cellular interface does not immediately

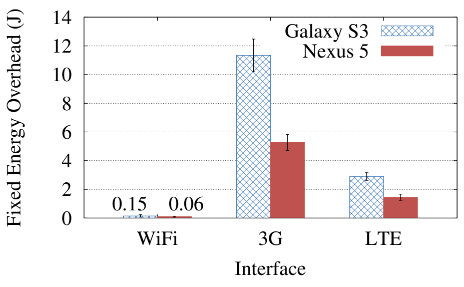


Figure 1: Fixed Energy Cost: WiFi and Cellular

Name	Samsung Galaxy S3	LG Nexus 5
Release Date	May 2012	Nov 2013
App. Processor	Qualcomm MSM8960	Qualcomm 8974-AA
Semiconductor	28nm LP	28nm HPM
Android Version	4.1.2 (Jelly Bean)	4.4.4 (KitKat)
Kernel Version	3.0.48	3.4.0
WiFi chipset	Broadcom BCM4334	Broadcom BCM4339

Table 1: Mobile Devices

return to the lower power state. Instead, ostensibly to save energy, the interface remains in the high power state for a period of time. If there is no further packet transmission during that period, the interface then returns to the low power state. This period is called the *tail*, and can last 6–12 seconds depending on the provider. Given its length, the tail contributes a significant portion of the energy cost for transmission, especially for short transfers. We refer to these as the fixed energy overheads. Our mobile devices share these overheads, as shown in Figure 1. Device specifications are given in Table 1.

3. ENERGY AWARE MPTCP

In this section, we present the design and implementation of eMPTCP.

3.1 Overview

eMPTCP's goal is to reduce energy consumption compared to standard MPTCP, while minimizing additional delay. It does this by dynamically choosing paths based on estimated energy efficiency. It also tries to avoid unnecessary cellular fixed costs by delaying subflow establishment over cellular interfaces, unless it believes it is more energy efficient to do so.

To this end, eMPTCP requires an energy model that captures power consumption according to network interface usage. There is a large body of literature that considers mobile devices' energy consumption based on bandwidth, signal strength, simultaneous use of interfaces, and so on [1, 7, 14, 17, 20]. We utilize our existing parameterized energy model described in [17] to generate the required information for eMPTCP.

Figure 2 presents the architecture of the eMPTCP

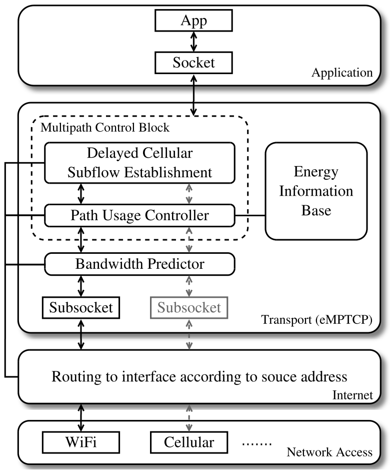


Figure 2: eMPTCP Architecture

implementation. Four core components are depicted that extend the regular MPTCP implementation at the transport layer: The *bandwidth predictor* (§3.2) monitors bandwidths over all active subflows and predicts future bandwidths. The *energy information base* (§3.3) holds the transition thresholds, indexed by bandwidth, for selecting which interfaces to use. The *path usage controller* (§3.4) makes decisions about which subflows to utilize, based on information retrieved from the bandwidth predictor and the energy information base. The *delayed subflow* (§3.5) module manages subflow establishment requests and delays them if necessary, based on information from the other two components. We describe each component in turn.

3.2 Bandwidth Predictor

The bandwidth predictor samples all active subflow throughputs and predicts their future values. By querying the routing information at the Internet layer, the bandwidth predictor identifies the interfaces associated with active subflows and categorizes predictions per interface. The sampling interval δ for each subflow is chosen based on its measured round-trip time (RTT) during subflow establishment, i.e., the elapsed time for the TCP three way handshake. Throughput predictions are made using a Holt-Winters time-series forecasting algorithm [30], which is known to be more accurate than formula-based predictors [13].

If an interface is inactive, the bandwidth predictor will not have any current throughput information about that interface. This can occur for two reasons: because the interface was activated but is now deactivated, or because it has never been activated. If the interface was deactivated, the bandwidth predictor uses old observed samples together with new sampled throughputs. If the interface has *never* been activated, the predictor assumes non-zero throughput (e.g., 5 Mbps) as an initial bandwidth for the interface to allow eMPTCP to probe the path through the interface.

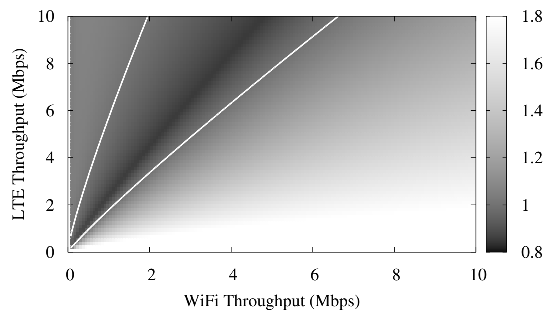


Figure 3: Energy Efficiency per Downloaded Byte (Samsung Galaxy S3)

3.3 Energy Information Base

The Energy Information Base (EIB) contains the data required for eMPTCP to decide which interface(s) to use to maximize energy efficiency. Since we cannot predict the amount of data remaining to be transferred, eMPTCP assumes a large transfer and defines efficiency in terms of per-byte energy consumption. This information is computed offline using our parameterized model for energy consumption that accounts for multiple interfaces [17] and generates the EIBs for two mobile devices listed in Table 1. Note that any energy model generated by state-of-art techniques such as [33, 34] can be used to populate the EIB.

Given the energy model, we are able to characterize the operating region where MPTCP is more energy efficient than standard single-path TCP over Cellular or WiFi. For example, Figure 3 shows the *relative* energy consumption per downloaded byte over both WiFi and LTE on the Samsung Galaxy S3, normalized by the amount of energy consumed using the best single interface (WiFi or LTE). The Figure is a grey-scale heat map where the darker the area, the more energy-efficient MPTCP is. At the left side of the region, TCP over LTE is the most energy efficient, whereas on the right side TCP over WiFi is the most efficient. In the region defined by the "V" shape, inside the solid white curves, using both interfaces consumes the smallest amount of energy to download a byte. This heat map is instantiated in the EIB.

The EIB represents this data as an array, indexed by the observed LTE throughput, where each entry includes two WiFi throughputs. This pair of throughputs specifies the transition points where eMPTCP should switch from a single interface to multiple interfaces and vice versa. For example, in the second row in Table 2, given an observed LTE throughput of 1 Mbps, if the observed WiFi throughput < 0.134 Mbps, LTE-only should be used; if the observed WiFi throughput ≥ 0.502 Mbps, WiFi-only should be used; otherwise, both should be used.

LTE Thpt.	WiFi throughput (Mbps)	
(Mbps)	LTE Only Threshold	WiFi Only Threshold
0.5	< 0.043	≥ 0.234
1.0	< 0.134	≥ 0.502
1.5	< 0.209	≥ 0.803
2.0	< 0.304	≥ 1.070
...	......	

Table 2: Example of Energy Information Base

3.4 Path Usage Controller

The path usage controller dynamically decides which paths to use based on energy efficiency. Once a multipath connection is established, eMPTCP dynamically decides which interfaces to use by retrieving the current estimates from the bandwidth predictor and using them to query the EIB. If data needs to be queued on a subflow, the subflow with the highest per-byte energy efficiency is chosen. Thus, eMPTCP seeks a local optimum in terms of total energy consumption.

To prevent oscillations, our algorithm uses a 10% 'safety factor' when switching from one state to another. Continuing the earlier example (second row of Table 2), if eMPTCP is using both interfaces, it requires a predicted WiFi throughput of 0.552 Mbps, not 0.502, to transition to WiFi-only. Similarly, if eMPTCP is using WiFi-only, it would require a predicted WiFi throughput of 0.452 Mbps to switch to both interfaces. This adds some hysteresis to the system and prevents it from switching states too frequently.

Note that, eMPTCP does not typically switch to using a cellular interface only, since the expected gain is not much more than using both.

3.5 Delayed Subflow Establishment

When data transfers are small, we wish to avoid any unnecessary expenditure of power establishing a cellular subflow, with the attendant energy costs of the promotion and tail states. To this end, if a cellular interface is not already active, eMPTCP introduces a delay between WiFi and 3G/LTE subflow establishment. The cellular subflow is not started until after eMPTCP receives κ bytes through the WiFi interface, thus avoiding these fixed overheads when transferring data fewer than κ bytes.

Even after κ bytes of data have been transferred, eMPTCP still postpones a cellular subflow establishment if the measured WiFi throughput is large enough such that using WiFi-only is more energy-efficient than using both interfaces, as indicated by the energy information base.

If the available WiFi throughput is poor, solely using the WiFi subflow until κ bytes are transferred can be less energy efficient than using both interfaces. Indeed, κ may never be reached on a slow WiFi subflow. To preclude this scenario, eMPTCP also uses a timer to trigger cellular subflow establishment. If the timer expires after τ seconds, eMPTCP establishes a cellular subflow, even if fewer than κ bytes have been transferred. The time threshold τ needs to be tuned to correctly estimate WiFi throughput: τ must be larger than or equal to the time required to collect enough samples after the WiFi subflow throughput stabilizes. Suppose B_W is the available throughput over WiFi, W_{init} the initial congestion window size, R_W the WiFi RTT, and ϕ the number of required samples. Then the condition for τ is:

$$\tau \geq R_W \times \left(\log_2 \frac{B_W \times R_W + W_{init}}{W_{init}} + \phi\right) \quad (1)$$

eMPTCP also postpones cellular subflow establishment if the current MPTCP connection is in an idle state with no transmission activity, even if the timer τ expires. This avoids unnecessary cellular subflow establishment for idle connections, as some applications (e.g., HTTP [8]) hold connections open in idle states even after completing data transfer. eMPTCP regards a connection as idle if it does not send or receive any packets during an estimated RTT.

3.6 Implementation

We have implemented eMPTCP in the Android versions specified in Table 1. We focus on downloads over the WiFi and LTE interfaces since they are more common. Each component of eMPTCP utilizes the routing information at the Internet layer to identify the interface associated with a subflow (`subsocket`). eMPTCP checks the destination entry of the socket. The destination entry (`struct dst_entry`) includes a pointer to the related network device (`struct net_device *dev`) which has a name of the interface and a pointer to the wireless device information (`struct wireless_dev *ieee80211_ptr`). Thus, eMPTCP can identify whether a socket is associated with a WiFi interface or not by checking if `dev` has a valid `ieee80211_ptr`.

MPTCP requires a default primary interface with which to initiate and receive transfers. Since WiFi is more energy efficient than cellular on our equipment and has negligible promotion and tail costs, as shown in Figure 1, we use WiFi as the default interface.

Once eMPTCP decides path usage for better energy efficiency, eMPTCP accordingly adds an MP_PRIO option [10], which changes the priority of subflows, to the next packet to be transmitted; for example, MP_PRIO indicating that LTE subflow is in low priority is added in the next packet if using LTE is energy inefficient.

When the device re-uses a path over an interface suspended by the path usage controller, the sender needs to utilize the additional subflow quickly. To this end, eMPTCP disables the CWND reset after an idle period longer than the retransmission timeout in RFC2861 [12] to ensure that a subflow avoids unnecessary slow-start when eMPTCP starts re-using the subflow. In addition, eMPTCP sets the measured round trip time (RTT) of the new subflow to zero. This modification enables a renewed subflow to be quickly probed by the MPTCP scheduler, since it selects a subflow with the lowest RTT for packet transmission [29]. Note that this is only for *re-used* subflows; new subflows behave the same as standard MPTCP in terms of CWND and RTT.

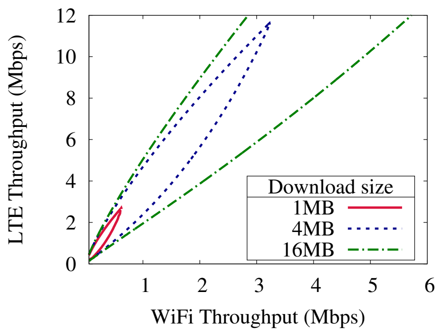


Figure 4: Operating Region where MPTCP is the Most Energy Efficient to Complete an Entire Transfer (Samsung Galaxy S3)

4. EVALUATION IN A CONTROLLED LAB

The goal of our first set of experiments is to investigate the behavior of eMPTCP in an environment where we can control wireless conditions. We first examine a static configuration (§4.2), then vary the WiFi bandwidth (§4.3). We examine the impact of background traffic (§4.4), and then the impact of mobility(§4.5). We explore how well eMPTCP adaptively controls path usage to obtain greater energy efficiency than standard MPTCP, as well as the impact on download time.

4.1 Setup

The mobile devices access a wired server, running Ubuntu Linux 12.04 with the MPTCP implementation [29]. The server is connected to our campus network through a single Gigabit Ethernet interface. Mobile devices can communicate with the server over the Internet using a WiFi access point (IEEE 802.11g), 3G or 4G cellular interfaces from AT&T. Energy traces are collected using the techniques described in [17].

For the controlled experiments with bandwidth changes, the mobile device downloads a large file while experiencing changing network conditions. We compare eMPTCP to standard MPTCP and TCP over WiFi in several scenarios, using the Samsung Galaxy S3.

We set eMPTCP parameters as follows. The download amount threshold κ to postpone an LTE subflow is set to one MB. This is because MPTCP is rarely more energy efficient than single path TCP when downloading a file smaller than this size. Figure 4 presents the calculated throughput region where MPTCP is more energy efficient than TCP over WiFi-only and LTE-only, for 1 MB, 4 MB, and 16 MB downloads. Based on our energy model in [17], the regions inside the curves of each size correspond to the throughput values where

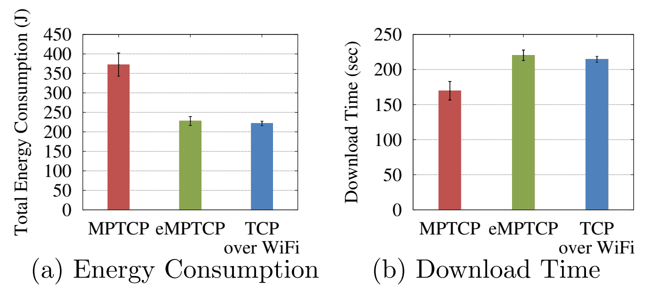


Figure 5: Static Good WiFi Comparison

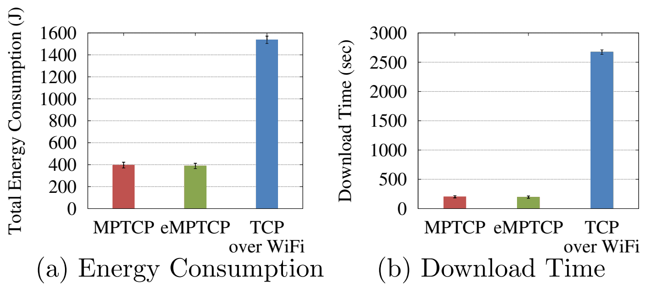


Figure 6: Static Bad WiFi Comparison

MPTCP is more energy efficient than single-path TCP to download a file of that size. The timer threshold τ is set to three seconds: given our experimental setting, the estimated condition based on equation (1) to guarantee ten bandwidth samples is $\tau \geq 2.67s$. While these values have worked well for our experiments, refining them to improve performance remains a subject for future work.

4.2 Experiments with Static Configuration

The purpose of these experiments is to show that, in relatively simple static environments, eMPTCP makes the proper path decisions to reduce power consumption. We measure energy consumption and download time of eMPTCP, MPTCP, and TCP over WiFi for two extreme cases: persistent high (>10Mbps) and low (<1Mbps) WiFi bandwidth while the device downloads a 256 MB file at a fixed location. Figures 5 and 6 compare the energy consumption and download times averaged over five runs for the two cases. In the first case, WiFi bandwidth is high, and thus using it is more power efficient than using either LTE or both interfaces. Figure 5 shows that eMPTCP chooses WiFi-only, effectively behaving similar to single-path TCP over WiFi. In contrast, when WiFi bandwidth is small (<1Mbps), and thus less energy-efficient than LTE, Figure 6 shows that eMPTCP yields almost the same performance as MPTCP by using both interfaces (after the LTE startup delay determined by parameters κ and τ). This illustrates that eMPTCP automatically seeks the most energy efficient path usage without user involvement.

4.3 Experiments with Bandwidth Changes

Next we examine how robust eMPTCP is to changes

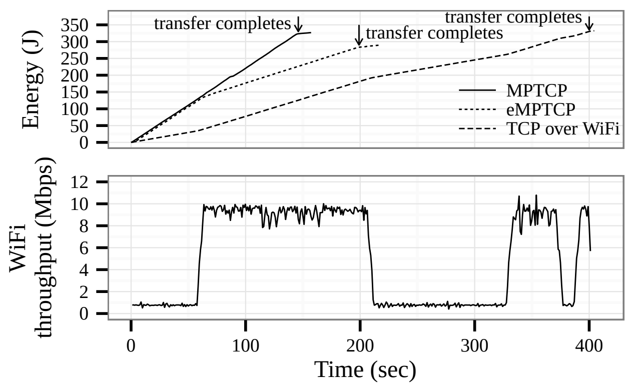


Figure 7: Accumulated Energy Consumption Example with Random WiFi Bandwidth Change

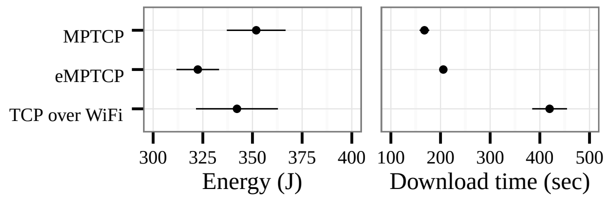


Figure 8: Random WiFi Bandwidth Changes

in network bandwidth. Here, WiFi link bandwidth is modulated by a two state on-off process with exponentially distributed times spent in the on or off state with a mean of 40 seconds. The bandwidth provided by the AP is $\leq$1Mbps or $\geq$10Mbps, depending on its state. Measurements are taken while the device downloads a 256 MB file at a fixed location.

Figure 7 presents an example time series trace of the accumulated energy consumption of eMPTCP, MPTCP, and TCP over WiFi for a single run with random WiFi bandwidth changes. At the beginning of the trace, after the LTE startup delay, eMPTCP uses both interfaces since WiFi throughput is too small to be energy efficient. However, eMPTCP suspends the LTE subflow after the WiFi bandwidth increases at around time 60, while MPTCP continues to use both interfaces. By suspending the LTE subflow, eMPTCP spends approximately 20% less energy than standard MPTCP at the cost of a 40% larger download time. Compared to single-path TCP over WiFi, eMPTCP both completes the download sooner (by about 50%) and consumes less energy (by about 15%).

Figure 8 compares energy consumption and download time averaged over ten runs. This figure, as well as similar ones to follow, uses a symbol to show the sample mean together with horizontal bars of length twice the standard error of the mean (SEM). The quantity SEM for a sample $x_1, x_2, \ldots, x_n$ is given by the expression $s/\sqrt{n}$ where

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})} \qquad (2)$$

is the sample standard deviation and $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean. In this set of experiments, eMPTCP consumes approximately 8% and 6% less energy on average than MPTCP and TCP over WiFi, respectively. However, eMPTCP is approximately 22% slower on average than MPTCP since it utilizes the LTE subflow only when the LTE subflow can improve energy efficiency. In contrast to single-path TCP over WiFi, by utilizing an LTE subflow when WiFi throughput is $\leq$1 Mbps, eMPTCP completes downloads twice as fast and consumes less energy. Note that the performance gain achieved by eMPTCP differs according to how WiFi bandwidth changes. If WiFi bandwidth changes frequently, the switching overhead in eMPTCP may become noticeable as an LTE interface triggers a promotion and tail state each time that it starts to be used again.

4.4 Experiments with Background Traffic

In this section we investigate how well eMPTCP copes with random background traffic. It is well known that multiple WiFi nodes can contend for the air channel, causing interference and loss (e.g., [19]). Background traffic causes contention and interference in the communication channel, resulting in throughput changes similar to link bandwidth changes. In these experiments, we utilize $n = 2$ or $n = 3$ interfering nodes, which use the same WiFi channel as the mobile device. While the device downloads a 256 MB file at a fixed location, each node generates UDP traffic according to a two state Markov on-off process, with rates (per second) λ_{on} and λ_{off}. We fix $\lambda_{\text{on}} = 0.05$, and then perform experiments with $\lambda_{\text{off}} = 0.025$ and $\lambda_{\text{off}} = 0.05$. As in Section 4.3, we control background traffic for the WiFi channel only.

Figure 9 shows sample throughput traces of MPTCP and eMPTCP when two interfering nodes turn traffic on and off with $\lambda_{\text{on}} = 0.05$ and $\lambda_{\text{off}} = 0.025$. We observe that standard MPTCP avoids aggressive use of the LTE subflow when the WiFi subflow provides high bandwidth, e.g., during the interval 10–60 seconds in Figure 9. This is because the MPTCP subflow scheduler chooses the WiFi subflow for packet transmission because it has a large CWND and the smallest RTT [29]. However, MPTCP consumes energy utilizing the LTE subflow even though the throughput gain is small. In contrast, eMPTCP suspends the LTE subflow when WiFi bandwidth is sufficiently large in order to avoid energy inefficient path usage. Note that at around time 60, the LTE throughput of MPTCP increases more slowly than that of eMPTCP. This is because the LTE subflow of MPTCP is in congestion avoidance after experiencing a loss at time 10 while that of eMPTCP is in slow-start

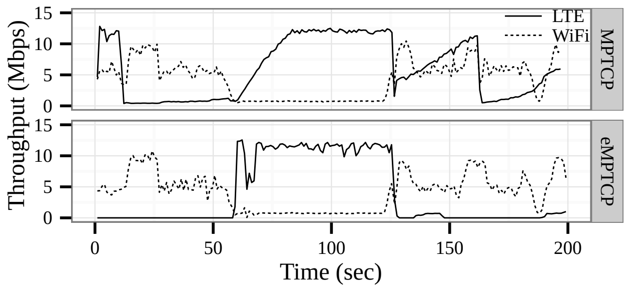


Figure 9: Example Throughput Trace with Random WiFi Background Traffic ($n = 2$, $\lambda_{\text{on}} = 0.05$, and $\lambda_{\text{off}} = 0.025$)

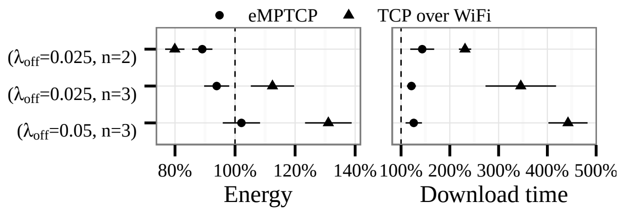


Figure 10: Comparison normalized relative to MPTCP in Random WiFi Background Traffic

since it is used for the first time. In Figure 9, eMPTCP incorrectly uses an LTE subflow in the intervals [140, 150] and [190, 200]. This is due to sudden decreasing WiFi throughputs, which result in incorrect throughput predictions. However, eMPTCP stops using the LTE subflow after the WiFi estimates improve.

Figure 10 presents the average energy consumption and download times for different values of n and λ_{off}, in percentage terms relative to MPTCP, i.e., smaller numbers are better, and numbers lower than 100% are better than standard MPTCP. MPTCP's place on the Figure is denoted by the red dashed line. Values are averaged over five experiments.

Figure 10 shows that eMPTCP consumes less energy than MPTCP as n and λ_{off} decreases. When $\lambda_{\text{off}} = 0.025$, we observe that the eMPTCP and WiFi-only are more efficient that MPTCP when $n = 2$. Recall that the energy efficiency of eMPTCP improves when it can suspend an LTE subflow in situations where using WiFi only is more energy efficient. Larger numbers of interfering WiFi nodes result in more losses caused by collisions when background traffic is present, resulting in more CWND decreases. Thus, the device is likely to obtain a larger TCP throughput with a larger CWND when there is no background traffic, resulting in a greater energy efficiency of TCP over WiFi and eMPTCP. Note that TCP over WiFi is most energy efficient when $n = 2$ and $\lambda_{\text{off}} = 0.025$. However, as shown in Figure 10, in that setting, TCP over WiFi requires twice as much time to complete a download as eMPTCP, while it consumes just 11% less energy.

As shown in Figure 10, MPTCP provides the smallest

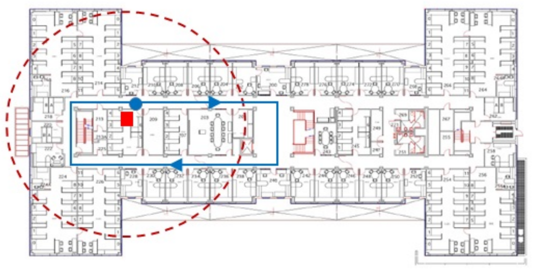


Figure 11: Mobile Scenario inside UMass CS building. Route starts at the blue point. The red square is the AP. The red dashed circle is the estimated usable access range of the AP.

download times, regardless of the values of n and λ_{off}, since it always utilizes the LTE subflow. In addition, as expected, the download time of TCP over WiFi becomes significantly larger as n and λ_{off} increase. Compared to MPTCP, download times under eMPTCP are 20-40% larger while energy consumption is 9-11% lower. This may be because eMPTCP sometimes poorly predicts available bandwidth due to fluctuating throughputs, as illustrated in Figure 9, and it also incurs additional energy overhead when suspending and resuming an LTE subflow due to the promotion and tail costs. Compared to single-path TCP over WiFi, eMPTCP reduces download time by up to 70%, while at the same time using less energy.

4.5 Experiments with Mobility

The focus of this section is to determine how well eMPTCP performs and adapts in a mobile scenario. We take measurements while moving for 250 seconds along the route shown in Figure 11. To make our comparison between MPTCP and eMPTCP as fair as possible, we use the same route for the experiments.

Figure 12 presents example traces of accumulated energy consumption from our mobile scenario. In this experiment, the device is sometimes within WiFi communication range, and sometimes outside it, depending on its location. As the device moves outside the communication range, WiFi throughput decreases, e.g., the duration around 25-40 seconds in Figure 12. At the beginning of the route, MPTCP starts by establishing both subflows, whereas eMPTCP postpones establishing an LTE subflow, since WiFi throughput is high enough to be more energy efficient than using both interfaces. However, eMPTCP establishes an LTE subflow after the WiFi bandwidth decreases when the device is leaving the AP communication range (at around 25 seconds in Figure 12). Then, whenever the device cannot obtain enough WiFi bandwidth to be more energy efficient than using both interfaces, eMPTCP utilizes the LTE subflow rather than only using the bad WiFi subflow. In this experiment, we observe that since

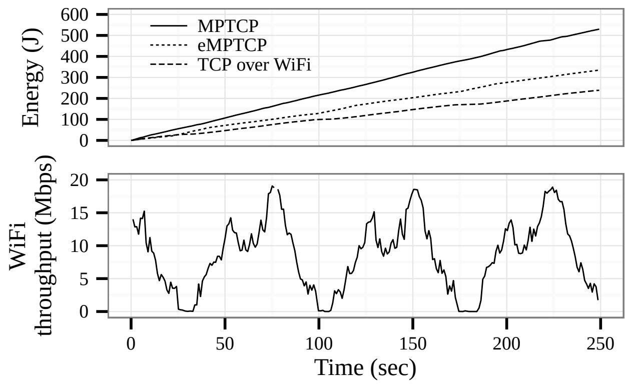


Figure 12: Example of Accumulated Energy Consumption with Mobile Scenario

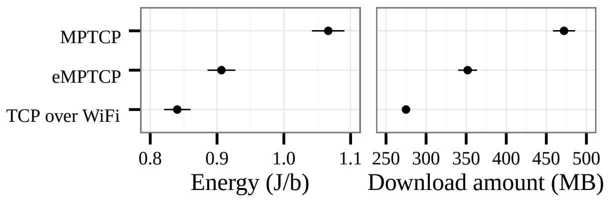


Figure 13: Mobile Scenario Comparison

the device is inside WiFi communication range most of the time, eMPTCP utilizes the LTE subflow only for a few short periods. Therefore, as shown in Figure 12, the slope of eMPTCP's accumulated energy consumption (the energy consumption per second) is larger than that of TCP over WiFi, but smaller than that of MPTCP.

We now examine the per-byte energy efficiencies of MPTCP, eMPTCP, and single-path TCP over WiFi. Figure 13 compares the energy consumption per byte and download amount for 250 seconds averaged over five runs. eMPTCP's energy consumption per byte is 22% smaller than that of MPTCP and 15% larger than that of TCP over WiFi, since eMPTCP utilizes the LTE subflow for only several short periods. Because WiFi throughput degradation is due only to the distance between the AP and device (as there is no WiFi background traffic in these experiments), TCP over WiFi is slightly better in terms of energy efficiency than eMPTCP, corresponding to the case when $n = 2$ and $\lambda_{\text{off}} = 0.025$ in Figure 10.

Focusing on the amount of data downloaded during the experiments, we observe that eMPTCP downloads 25% less data than MPTCP, while exhibiting 22% lower per-byte energy costs. Here, eMPTCP roughly loses about the same in performance that it saves in energy, due to the overhead of switching between WiFi-only and using both interfaces. eMPTCP downloads 28% more data than single-path TCP over WiFi even though it yields almost as good per-byte energy efficiency (just 8% more energy consumption per byte).

4.6 Comparison with existing approaches

Raiciu *et. al* [28] propose a simple strategy called MPTCP with WiFi First, where MPTCP only uses WiFi when available, and only uses the cellular network otherwise. This is accomplished by placing the cellular subflow in backup mode and activating it only when WiFi is not available. This simple strategy may seem similar to eMPTCP, however, it cannot take advantage of dynamic situations where TCP over LTE or MPTCP is more energy-efficient than TCP over WiFi. MPTCP with WiFi First can only utilize an LTE subflow when the WiFi subflow explicitly breaks, such as due to a WiFi AP disassociation. For example, in our mobile scenario in Section 4.5, MPTCP with WiFi First would not use the LTE subflow even when the WiFi subflow becomes unusable, e.g., the duration around 25∼40 seconds in Figure 12, since the device does not lose the WiFi association. Therefore, if WiFi provides too low bandwidth to be more energy efficient than LTE while it is still associated, MPTCP with WiFi First degenerates into single-path TCP over WiFi, which yields inefficient energy usage as shown in subsections 4.2–4.5. It also needlessly activates the cellular interface at connection establishment.

Pluntke *et al.* [24] introduce a Markov Decision Process (MDP) based MPTCP path scheduler to minimize energy consumption. The MDP based scheduler cannot be computed in the Kernel at run time since it requires expensive computational overhead and a finite state machine of throughput changes with transition probabilities, which makes it hard to be practical in real deployments. Therefore, rather than directly implementing their approach in mobile devices, we generate the MDP based schedulers and simulate their behaviors given our experimental scenarios and energy model. Unit time for state transitions is set to one second as in [24]. Note that the authors consider only 3G and WiFi energy models in which 3G energy consumption becomes lower than WiFi for high data rates. In contrast, LTE energy consumption per *second* never becomes lower than WiFi in our energy model. We observe that the generated MDP schedulers choose WiFi-only for all scenarios, resulting in same energy performance (and limitations) as TCP over WiFi.

5. EVALUATION IN THE WILD

We next examine whether eMPTCP provides greater energy efficiency than standard MPTCP in more realistic environments. We deploy MPTCP enabled servers in Asia (Singapore - SNG), Europe (Amsterdam - AMS), and North America (Washington D.C. - WDC). Each server has one network interface on the public Internet. We investigate the performance of eMPTCP when downloading files of different sizes (256 KB for small file transfer and 16 MB for large file transfer) at three locations: a university building where the WiFi AP is connected to the campus network, student housing where the AP is connected to the campus network through Cisco Long-Reach Ethernet, and a personal residence where the AP connects to a cable network.

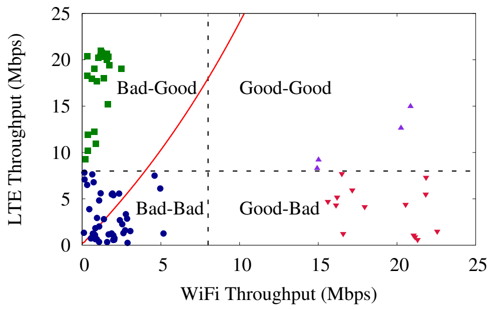


Figure 14: Trace Categories according to WiFi and LTE throughput - 16MB Downloads

We collect ten traces for each combination of file size, device and server locations. Since network conditions can vary over time, for each configuration we conduct ten iterations of a set of experiments. Each set consists of one run each of eMPTCP, original MPTCP, and single-path TCP over WiFi for the same configuration. The ordering within the set is randomized.

5.1 Trace Categorization

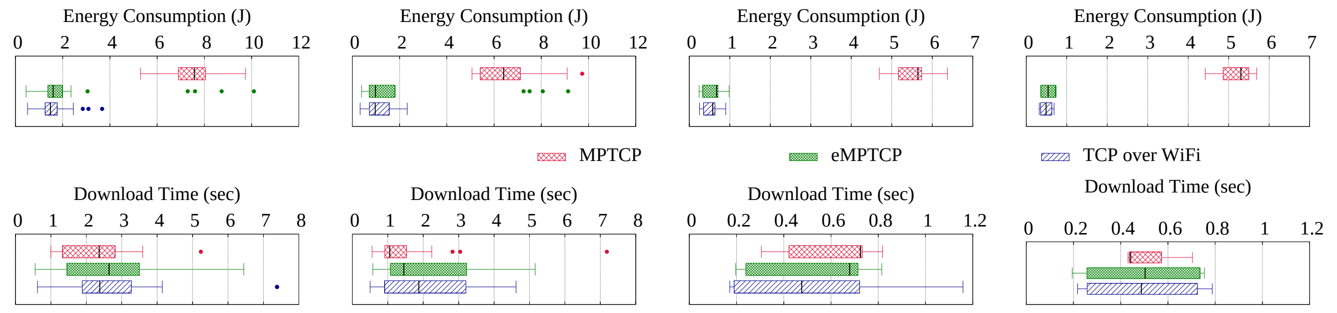
We group the collected traces into four categories based on the qualities of WiFi and LTE, either Good or Bad. We set 8 Mbps as a threshold to decide whether a network is good or bad. Figure 14 presents a scatterplot of measured WiFi and LTE throughputs for all experiments downloading 16 MB files. The scatterplot is divided into the four categories, and the red line indicates the boundary above which MPTCP is more energy efficient than TCP over WiFi for downloading.

5.2 Small File Transfers

Figure 15 presents Whisker plots showing first quartile (Q_1), median, third quartile (Q_3), and outliers of measured total energy consumptions and download times in the 256 KB traces for each of the four categories. Dots are outliers, which sit outside the range $[Q_1 - 1.5IQR, Q_3 + 1.5IQR]$, where IQR is the inter-quartile range defined as $(Q_3 - Q_1)$.

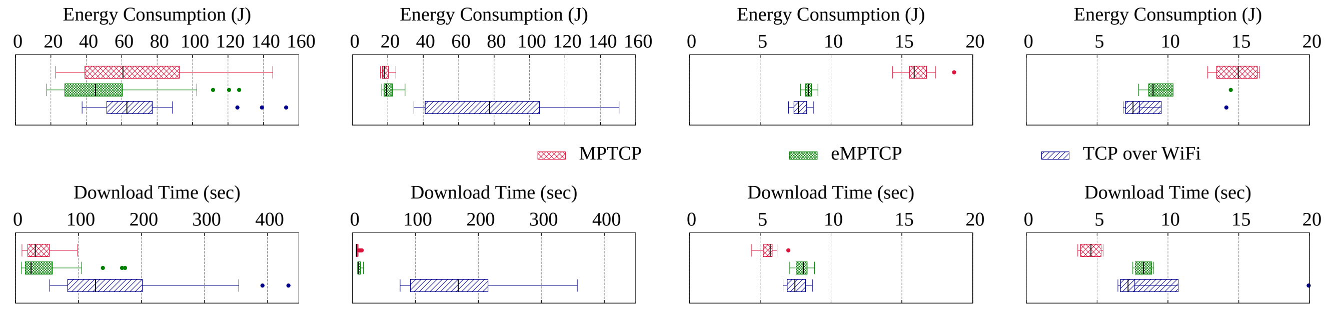
In this environment, eMPTCP almost always behaves the same as TCP over WiFi across all categories, yielding significantly less total energy consumption (from 75% to 90%) with statistically similar download times as MPTCP. This is because the transfer is short, can be completed over WiFi, and eMPTCP avoids the tail state power costs of the LTE interface by delaying LTE subflow establishment.

eMPTCP does use the LTE subflow in a few instances, namely the outliers in Figure 15(a) and (b), which exhibit similar values to MPTCP. In these cases, LTE subflow establishment is triggered via timer expiration



(a) Bad WiFi & Bad LTE (b) Bad WiFi & Good LTE (c) Good WiFi & Bad LTE (d) Good WiFi & Good LTE

Figure 15: Small File Transfers (256 KB Downloads)



(a) Bad WiFi & Bad LTE (b) Bad WiFi & Good LTE (c) Good WiFi & Bad LTE (d) Good WiFi & Good LTE

Figure 16: Large File Transfers (16 MB Downloads)

since the WiFi throughput obtained by eMPTCP is exceptionally small. Recall that after the timer expires, eMPTCP greedily chooses paths based on expected per-byte efficiency without knowing the length of the transfer remaining. Thus, even though the remaining data is too small for eMPTCP to take advantage of LTE's relative energy efficiency in this scenario, eMPTCP uses both because of the extremely slow WiFi.

5.3 Large File Transfers

Figure 16 presents Whisker plots of measured total energy consumption and download time of the 16 MB download experiments for each category. We observe the following:

Bad WiFi & Bad LTE: eMPTCP consumes 33% less energy than MPTCP and TCP over WiFi, while completing downloads in 20% less time. eMPTCP is the most energy efficient since it adaptively controls path usage according to expected per-byte efficiency. TCP over WiFi exhibits similar energy consumption to standard MPTCP but with roughly 6x longer download times.

Bad WiFi & Good LTE: eMPTCP yields similar energy consumption to MPTCP with slightly larger download times. eMPTCP behaves similarly to MPTCP, since the throughput obtained over WiFi is small. The slightly larger download times than MPTCP are due to the delayed LTE subflow establishment. TCP over WiFi performs the worst, both in terms of download time and efficiency, since the WiFi is poor.

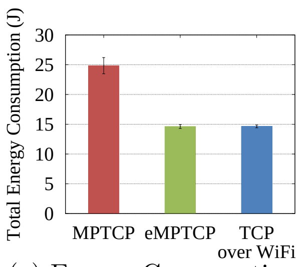
Good WiFi & Bad LTE: eMPTCP uses roughly 50% of the energy that MPTCP does, since it never utilizes the LTE subflow. TCP over WiFi behaves similarly. eMPTCP takes about 20% longer than MPTCP to complete downloads. eMPTCP essentially behaves the same as TCP over WiFi, and obtains similar results.

Good WiFi & Good LTE: This is similar to the above case, since the WiFi is fast and using WiFi-only is more power efficient than using both.

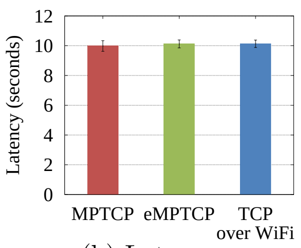
5.4 Case Study: Web Browsing

We now examine whether eMPTCP improves energy efficiency with a common application: Web browsing. To this end, we deploy a copy of CNN's home page (as of 9/11/2014), which consists of 107 Web objects, into our MPTCP server in Washington DC. At our department building, we measure the energy consumption and latency to download all objects in the page. We consider the environment in this setting to have good WiFi and good LTE.

Figure 17 shows the average measured energy consumption and latency of MPTCP, eMPTCP, and single-



(a) Energy Consumption



(b) Latency

Figure 17: Web browsing Comparison

path TCP over WiFi. In this experiment, the Android web browser establishes six parallel (MP)TCP connections to the server (12 subflows for MPTCP), with HTTP persistent connections. Note that the values are averaged over 10 runs and the power consumed for the Web browser application is included in the total energy consumption. As shown in Figure 17(a), MPTCP consumes 60% more energy (around 10J) than eMPTCP and TCP over WiFi. Since almost of all objects in the Web page are small (<256 KB), eMPTCP does not utilize the LTE network and achieves better energy efficiency. Figure 17(b) shows that eMPTCP yields almost the same latency as MPTCP with less energy consumption. This is because, for small downloads, it is hard for MPTCP to obtain gains by utilizing an LTE subflow in the case of small object downloads [4]. In contrast, eMPTCP automatically postpones an LTE subflow establishment unless a user initiates a large data transfer, such as video streaming. If a large transfer occurs, eMPTCP adaptively switches path usage between using both interfaces and WiFi-only, resulting in better energy efficiency than MPTCP. This example is meant to illustrate the potential benefits of eMPTCP for more complex workloads. More representative and exhaustive application scenarios remain as future work.

6. RELATED WORK

Pluntke *et al.* [24] are the first to introduce the concept of scheduling paths in MPTCP to minimize energy consumption. They use a scheduler based on a Markov decision process. Schedules are computationally expensive operations, hence they are computed in the cloud and downloaded periodically to the device, which is impractical for real deployment. They evaluate their scheduler via simulation, using models of device energy consumption. They find they can reduce energy consumption by almost 10% in one out of four scenarios. Our algorithm, in contrast, is evaluated experimentally using a real MPTCP implementation on a physical device, and across many scenarios. It also achieves much higher energy reductions, up to 90%. Pluntke *et al.* [24] is discussed in more detail in Section 4.6.

Raiciu *et al.* [28] look at a number of issues in using MPTCP for mobility, including power consumption. They propose and evaluate a simple strategy that periodically samples both paths for 10 seconds and then uses the more energy-efficient path for 100 seconds. Evaluating their approach via simulation, their method is more energy efficient than an energy-unaware MPTCP implementation, but achieves lower bandwidth. They also propose a strategy called "MPTCP with WiFi First", which exclusively uses WiFi when it is present, and only uses the cellular network when WiFi is unavailable. However, this approach cannot avoid inefficient use of energy due to automatic activation of the cellular interface and the attendant promotion and tail state costs. Raiciu *et al.* [28] are discussed in more detail in Section 4.6.

Paasch *et al.* [21] studied mobile/WiFi handover performance with MPTCP. They also measure energy consumption on a Nokia N950 smartphone for two download scenarios and find that using WiFi alone is more energy-efficient than base MPTCP. They also suggest *Single-Path Mode*, which establishes a new subflow only after the current interface goes down. By setting WiFi as the primary interface, Single-Path Mode MPTCP can avoid the fixed energy overhead for 3G/LTE when WiFi is available.

Compared to our approach, however, both strategies are unable to take advantage of more dynamic situations where TCP over LTE or MPTCP may become more energy efficient than TCP over WiFi. In addition, they can only utilize an LTE subflow when the WiFi subflow explicitly breaks, such as disassociating with an AP.

Peng *et al.* [22] study energy consumption in MPTCP via analysis as a global optimization problem. They present two algorithms, customized for real-time and file transfer, and evaluate them via simulation. While they do utilize an energy model, they do not consider cellular promotion and tail costs, and do not evaluate their approach experimentally.

Bui *et al.* [2] present GreenBag, a middleware system to aggregate bandwidth of asymmetric wireless links for video streaming. GreenBag estimates the available bandwidth of WiFi and LTE and determines the amount of traffic to allocate to each interface. The authors show that GreenBag reduces energy consumption by 14∼25% compared with a bandwidth aggregation for throughput maximization. Since GreenBag operates above the TCP layer as a system background process, it only works for modified HTTP requests sent not to original destinations but to GreenBag. Thus, each application needs to be modified to cooperate with GreenBag, making deployment difficult. MPTCP, in contrast, requires no application modifications and works with all TCP traffic.

Ding *et al.* [6] proposes a mobile traffic offloading architecture that utilizes WiFi to migrate mobile traffic from cellular while reducing energy consumption. The authors demonstrate that the proposed architecture obtains more than 80% energy savings, using a prototype on a Nokia N900 smartphone and their own music streaming application. However, they do not consider potential energy gain from simultaneous use of both interfaces. As with Bui *et al.* [2], applications must be modified for the approach to work.

Croitoru *et al.* [5] show that MPTCP can improve user experience over WiFi by associating with multiple APs simultaneously. By utilizing an MPTCP connection with subflows connected to all available APs, a mobile client can maintain seamless connectivity without having to consider a handover. The authors also investigate situations when connecting to multiple APs can degrade network performance. They propose estimating client-side AP efficiency and using ECN notification to direct traffic to subflows associated with the most efficient APs while avoiding use of subflows that experience poor throughput. Their study does not consider cellular usage or energy consumption.

7. CONCLUSION

This paper proposes, implements, and evaluates an enhanced MPTCP designed to improve energy efficiency, called eMPTCP. eMPTCP manages subflow usage based on expected per-byte energy efficiency given available bandwidth. Our experimental results using our implementation in real mobile devices show that eMPTCP is able to consume less energy than MPTCP while still providing MPTCP's benefits of multi-path such as improved performance, availability, and transparency. For future work, we plan to examine more statistically varied application traffic such as video streaming, Web downloads, and upload scenarios, as well as other wireless technologies such as Bluetooth.

Acknowledgements

This research was sponsored by the US Army Research Laboratory and the UK Ministry of Defence and was accomplished under Agreement No. W911NF-06-3-0001. We would like to thank the anonymous reviewers for their useful comments.

8. REFERENCES

[1] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: A measurement study and implications for network applications. In *Proc. of ACM IMC*, pages 280–293, 2009.

[2] D. H. Bui, K. Lee, S. Oh, I. Shin, H. Shin, H. Woo, and D. Ban. Greenbag: Energy-efficient bandwidth aggregation for real-time streaming in heterogeneous mobile wireless networks. In *Proc. of IEEE RTSS*, pages 57–67, 2013.

[3] A. Chakraborty, V. Navda, V. N. Padmanabhan, and R. Ramjee. Coordinating cellular background transfers using loadsense. In *Proc. of ACM MobiCom*, pages 63–74. ACM, 2013.

[4] Y.-C. Chen, Y.-S. Lim, R. J. Gibbens, E. Nahum, R. Khalili, and D. Towsley. A measurement-based study of multipath TCP performance in wireless networks. In *Proc. of ACM IMC*, pages 455–468, Nov 2013.

[5] A. Croitoru, D. Niculescu, and C. Raiciu. Towards WiFi mobility without fast handover. In *Proc. of USENIX NSDI*, pages 219–234, 2015.

[6] A. Ding, B. Han, Y. Xiao, P. Hui, A. Srinivasan, M. Kojo, and S. Tarkoma. Enabling energy-aware collaborative mobile data offloading for smartphones. In *Proc. of IEEE SECON*, pages 487–495, 2013.

[7] N. Ding, D. Wagner, X. Chen, Y. C. Hu, and A. Rice. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In *Proc. of ACM SIGMETRICS*, pages 29–40, 2013.

[8] R. Fielding and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. RFC 7230, 2014.

[9] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Architectural guidelines for multipath TCP development. RFC 6182 (Informational), Mar. 2011.

[10] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP extensions for multipath operation with multiple addresses. RFC 6824, 2013.

[11] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall. Demystifying 802.11n power consumption. In *Proceedings of the 2010 workshop on Power aware computing and systems (HotPower)*, page 1. USENIX Association, 2010.

[12] M. Handley, J. Padhye, and S. Floyd. TCP congestion window validation. *IETF RFC 2861*, 2000.

[13] Q. He, C. Dovrolis, and M. Ammar. On the predictability of large transfer TCP throughput. In *Proc. of ACM SIGCOMM*, pages 145–156, 2005.

[14] J. Huang, Q. Feng, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A close examination of performance and power characteristics of 4G LTE networks. In *Proc. of ACM MobiSys*, pages 225–238, 2012.

[15] K.-Y. Jang, S. Hao, A. Sheth, and R. Govindan. Snooze: Energy management in 802.11n WLANs. In *Proc. of ACM CoNEXT*, pages 12:1–12:12, 2011.

[16] C.-Y. Li, C. Peng, S. Lu, and X. Wang. Energy-based rate adaptation for 802.11n. In *Proc. of ACM MobiCom*, pages 341–352. ACM, 2012.

[17] Y.-S. Lim, Y.-C. Chen, E. M. Nahum, D. Towsley, and R. J. Gibbens. How green is multipath TCP for mobile devices? In *Proc. of ACM AllThingsCellular*, pages 3–8, 2014.

[18] F. Lu, G. M. Voelker, and A. C. Snoeren. SloMo: Downclocking WiFi communication. In *Proc. of USENIX NSDI*, pages 255–258, 2013.

[19] J. Manweiler and R. Roy Choudhury. Avoiding the rush hours: WiFi energy management via traffic isolation. In *Proc. of ACM MobiSys*, pages 253–266, 2011.

[20] A. Nika, Y. Zhu, N. Ding, A. Jindal, Y. C. Hu, X. Zhou, B. Zhao, and H. Zheng. Energy and performance of smartphone radio bundling in outdoor environments. In *Proc. of WWW*, pages 809–819, 2015.

[21] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure. Exploring mobile/WiFi handover with multipath TCP. In *Proc. of ACM Cellnet*, pages 31–36, 2012.

[22] Q. Peng, M. Chen, A. Walid, and S. Low. Energy efficient multipath TCP for mobile devices. In *Proc. of ACM MobiHoc*, pages 257–266, 2014.

[23] T. Pering, Y. Agarwal, R. Gupta, and R. Want. Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proc. of ACM MobiSys*, pages 220–232, 2006.

[24] C. Pluntke, L. Eggert, and N. Kiukkonen. Saving mobile device energy with multipath TCP. In *Proc. of ACM MobiArch*, pages 1–6, 2011.

[25] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely. Energy-delay tradeoffs in smartphone applications. In *Proc. of ACM MobiSys*, pages 255–270, 2010.

[26] A. Rahmati and L. Zhong. Context-for-wireless: Context-sensitive energy-efficient wireless data transfer. In *Proc. of ACM MobiSys*, pages 165–178, 2007.

[27] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving datacenter performance and robustness with multipath TCP. In *Proc. of ACM SIGCOMM*, pages 266–277, 2011.

[28] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley. Opportunistic mobility with multipath TCP. In *Proc. of ACM MobiArch*, pages 7–12, 2011.

[29] 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 *Proc. of USENIX NSDI*, pages 399–412, 2012.

[30] P. J. Rockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer, 1994.

[31] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan. Bartendr: A practical approach to energy-aware cellular data scheduling. In *Proc. of ACM MobiCom*, pages 85–96, 2010.

[32] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath TCP. In *Proc. of USENIX NSDI*, pages 99–112, 2011.

[33] F. Xu, Y. Liu, Q. Li, and Y. Zhang. V-edge: Fast self-constructive power modeling of smartphones based on battery voltage dynamics. In *Proc. of USENIX NSDI*, pages 43–55, 2013.

[34] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *CODES+ISSS*, pages 105–114, 2010.