Sizing Up User Traffic: Flow-based Mobile Data Offloading over WiFi

Delia Ciullo °, Thrasyvoulos Spyropoulos ⋆, Navid Nikaein ⋆, Bruno Jechoux ‡, Giannis Sarantidis *
° Groupe Renault, Sophia Antipolis, France ⋆ EURECOM, Sophia Antipolis, France
‡ TCL Communication, France * Nokia Networks, Greece

***Abstract*—We propose a smart offloading policy that dynamically assigns data flows to the WiFi and cellular interfaces, so as to minimize a given cost function (related to energy consumption and cellular plan usage), while keeping the average per-flow delay bounded. The basic insight of the proposed Threshold Policy is to assign larger flows to the network that provides the best rate (often WiFi), and smaller flows to the other, since energy is generally related to the time needed to send/receive data. However, choosing the size cutoff optimally must also consider load-balancing and queueing aspects, WiFi availability, flow size statistics, and user/application preferences. We validate our model against simulations, and show that our policy outperforms other standard or smart policies, achieving considerably better energy-delay trade-offs, while only offloading a small percentage of (large) flows. Initial measurements performed on an Android-based offloading prototype further support our findings.**

I. Introduction

The explosive growth of mobile data traffic, e.g. video on-demand, YouTube, personalized radio [1], is raising concerns both for mobile operators and for users. WiFi offloading is seen as one potential solution for both. Operators can alleviate congested base stations, and users can avoid using their data plan and face potential charges. What is more, studies suggest that WiFi-based offloading could save device battery power (e.g. 55% reported in [6]), especially when WiFi offers better rates than cellular.

Yet, a key disadvantage of WiFi is its sporadic availability. At the moment, "offloading" consists of simply switching *all* traffic from cellular to WiFi, when WiFi is available. This type of offloading, referred to hereafter as *On-the-spot* offloading [5], [25], can be chosen manually by the user or even enforced by the operator [4]. The actual benefits of course strongly depend on the WiFi availability.

To offload more data, *Delayed* offloading has been also proposed, where data transfers can be delayed until WiFi is available. The majority of proposals in this area are aggresive policies that send *all data* initially to the WiFi interface, [6], [7], [8], and associate each flow with a deadline: if the transfer does not finish within its deadline, data is sent over the cellular interface. A larger amount of data could be thus offloaded, if long enough deadlines are chosen [6], [7], but this could also lead to long delays for many flows.

Various studies have shown that users are willing to delay some traffic if incentives are provided [9], [10] and a growing number of applications are tolerant to longer delays, e.g. episodic Video-On-Demand, or bulk data transfers such as cloud synchronization and software updates, social network feeds, etc. Nevertheless, long delays are generally considered unwelcome for a user. Hence, aggresive policies that can potentially delay *most* flows might impede their adoption. It is preferable to selectively offload as few as possible.

The above two classes of policies, delaying all flows by some deadline (delayed offloading) or no flows (on-the-spot offloading), represent two extremes. A smart offloading policy should *choose* which flows to assign to the cellular interface and which to WiFi. While an abundance of criteria could be imagined for this choice (e.g. application type, user preferences, network rates), a good policy would ideally offload enough data to have an impact on performance metrics of interest (e.g. on congestion, data plan usage, device battery), but should not delay many user activities or data transfers.

The main argument in this paper is that the above seemingly conflicting goals could be achieved by offloading only the larger flows to WiFi, and keeping smaller flows, constituting the majority, on cellular. This is reminiscent of task assignment problems in queueing theory and server farms, where sending large flows to a different server can greatly reduce average job delay [12]. While the standard task assignment problem is to minimize job delay, things are more complicated in the case of data offloading. First, data plan usage costs and battery consumption are equally, if not more important for users. Second, WiFi is only intermittently available, and the ability to delay some flows adds another important dimension to the flow assignment decision. Finally, while size-based assignment seems like an evident solution to the above problem, choosing the actual size cutoff is non-trivial, and depends on the exact objective and key network parameters (e.g. transmission rates, traffic patterns, user mobility, etc.).

To this end, in this paper we propose an analytical framework for the above problem, and use it to derive an optimal size-based flow assignment policy, called "Threshold Policy" (TP for short) Summarizing, the key contributions are:

(1) An analytical formulation of the problem of flow assignment over two heterogeneous network interfaces, and the derivation of an optimal policy, called TP, in closed form, as a function of key network parameters (Section II).

(2) An extension of the basic TP policy, to include strict per flow deadlines (Section III).

(3) A detailed simulation-based validation of TP, showing that the main results hold even when assumptions made in the theoretical model are relaxed. (Section IV)

(4) An Android-based prototype of TP, and preliminary experimental results supporting our findings (Section V)

To our best knowledge, this is the first work to consider analytically the problem of optimal per-flow offloading, making a useful first step beyond existing works that treat all flows equally [6], [7], [24]. Finally, we note that TP could be implemented both on the user device or on the network side (e.g. at a BS offering both cellular and WiFi access). Due to space limitations, we focus our discussion on the former case.

II. Threshold policy

We introduce now the optimal flow assignment problem more formally. The following assumptions are made:

A.1: We define a *flow* as a concatenation of uploaded (or downloaded) packets corresponding to the same application request (e.g., a downloaded file, a photo uploaded on FaceBook). Identifying and delimiting flows is a well researched problem [13]. A user's activity generates flow requests according to a Poisson Process with parameter λ.

A.2: Each flow has a size S, drawn from a *generic* probability distribution $F(s)$, $s \in [S_m, S_M]$, (with density $f(s)$ when F is continuous). Flow size distributions are often highly skewed, exhibiting decreasing failure rate (DFR) [23].

A.3: A UE is equipped with two network interfaces, cellular and WiFi, that can be up and used concurrently *for different data*. This is not currently the case for most smartphones, but can already be manually implemented, as shown in Section V. A flow A could be transmitted over WiFi while another flow B sent, in parallel, over the cellular interface[1].

A.4: Each interface is associated with a mean transmission rate, denoted as R^C and R^W for cellular and WiFi, respectively. While the instantaneous rates might fluctuate, depending on the location of the UE and load on the specific AP or BS (i.e. other users), we assume that the above values correspond to measured estimates, averaged over a longer time-window to ensure stability. A potential implementation is discussed in Section V.

A.5: W.l.o.g. we assume that cellular connectivity is always available. WiFi availability is modeled as an ON-OFF alternating renewal process: "ON" are periods with WiFi connectivity, and "OFF" are periods without. The durations of these periods, T_{ON} and T_{OFF}, are random variables, drawn from generic distributions. These parameter values and distributions will often be chosen according to the observations in two real measurement studies found in [6], [7].

A.6 A cost per bit L_C is associated with the cellular interface. A cost per bit L_W is associated with WiFi. These costs depend on the objective considered, and might relate to data plan usage, mean energy per transmitted/received bit, or a combination. We discuss some examples later.

A.7 We consider two types of user quality of service (QoS) requirements: In this section, we assume that a user only has a maximum value for the average delay over all her flows, denoted as D^M. In Section III, the user or application also sets a strict, possibly different delay requirement for each flow.

[1]We assume for simplicity that all packets of a flow must go over the same interface. Interface aggregation, e.g. [2], [3], is beyond the scope of this paper. Furthemore, in practice a few flows might be interrupted, if WiFi connectivity is lost, and restarted over the cellular connection.

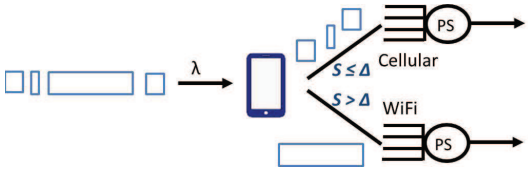

Fig. 1: Queueing system model.

The system model is shown in Fig. 1: A controler observes incoming flow requests, and assigns each flow to the WiFi or to the cellular interface queue, in order to minimize a cost function (related to A.6) subject to QoE constraints (related to A.7). Each interface serves all ongoing flows using Processor-Sharing (PS), i.e. the "server" (either WiFi or cellular) capacity is equally shared among all the current flows. While PS is only approximately implemented in practice (e.g. round-robin with a finite quota) it is very often used to describe the bandwidth sharing of elastic traffic in TCP-based networks and has been successfully applied to model the flow-level behavior in various networks, including WLAN, UMTS-HSDPA [19]. It is also known to have good delay and fairness properties [12].

Our proposed policy, called *Threshold Policy (TP)*, is to assign all flows larger than a certain threshold, namely Δ, to the server with the lower cost per bit and all flows smaller than Δ to the other (in the remainder, the terms interface and server are used interchangeably). In most cases, $L_C > L_W$, i.e. the WiFi cost is smaller than the cellular. Hence, without loss of generality, the subsequent analysis will be presented for the case of larger flows assigned to WiFi, as depicted in Fig. 1. However, the framework is applicable to the case of $L_C < L_W$, as well, (in which case the rule is flipped) and we'll refer back to it, where necessary.

To gain some insight, assume that the objective is monetary cost. Then, L_W is often 0, or in general $L_C > L_W$. We would thus like to maximize the amount of data transmitted over WiFi, or similarly, minimize the average cost per bit over a long time horizon. Nevertheless, aggressively assigning all flows to the interface with the lowest cost, might lead to excessive delays, violating the average user requirement D^M. The delays incurred are of two types:

(*Queueing/Transmission*) If the load on an interface approaches or exceeds its capacity, queueing theory predicts that the mean delay to transmit a flow grows to infinity.

(*WiFi unavailability*) If a flow is assigned to the WiFi interface, and no WiFi is currently available (as in the Delayed Offloading case), that flow will incur an extra delay, until the next WiFi availability period.

Consequently, to satisfy the delay requirement, some data will have to be sent over the more expensive interface, even if this increases the total cost. The following theorem resolves this tradeoff optimally.

Theorem 1. *Among all the flow-assignment policies (not*

necessarily size-based) for the system represented in Fig. 1, the size-based policy with threshold Δ given by the following optimization problem, gives the minimum possible cost per flow, subject to an average delay constraint of D^M.

$$\begin{array}{ll} min_{\Delta} & L_W \int_{\Delta}^{\infty} d\bar{F}(s) + L_C \int_0^{\Delta} d\bar{F}(s) \\ s.t. & \frac{1}{\frac{R^W}{\int_{\Delta}^{\infty} d\bar{F}(s)} - \lambda} + \frac{1}{\frac{R^C}{\int_0^{\Delta} d\bar{F}(s)} - \lambda} + \bar{F}(\Delta) \cdot D^{WF} \leq D^M \\ & \frac{\lambda \int_{\Delta}^{\infty} d\bar{F}(s)}{R^W} < 1, \frac{\lambda \int_0^{\Delta} d\bar{F}(s)}{R^C} < 1 \\ & S_m \leq \Delta \leq S_M \end{array} \tag{1}$$

where $D^{WF} = 0$ for on-the-spot offloading, and $D^{WF} = \frac{\mathbb{E}[T_{OFF}^2]}{2(\mathbb{E}[T_{ON}]+\mathbb{E}[T_{ON}])}$ for delayed offloading.

Proof. We define $X = \mathbb{1}_{[S \geq \Delta]}$ as an indicator random variable that is equal to one if $S \geq \Delta$, and zero if $S < \Delta$. The expected value of X is: $E[X] = 1 - F(\Delta) \doteq p$ for each flow i. Thus, each flow i is assigned to WiFi with probability p, and to cellular with probability $1 - p$. The objective function of (1) represents the average transmission (reception) energy in the system. Assuming that N flows are sent over a certain time window, the total transmission (reception) cost is then

$$L_W \cdot \sum_{i=1}^{N} s_i \cdot x_i + L_C \cdot \sum_{i=1}^{N} s_i \cdot (1 - x_i), \tag{2}$$

where s_i, x_i are instances of S, X for each flow i. Taking expectations, we can compute the average cost per bit $\mathbb{E}[L] = L_W \cdot \mathbb{E}[S \cdot X] + L_C \cdot \mathbb{E}[S \cdot (1 - X)]$, where

$$\mathbb{E}[S \cdot X] = \mathbb{E}[S|S \geq \Delta] \cdot p = \frac{\int_{\Delta}^{\infty} sf(s)ds}{p} \cdot p = \int_{\Delta}^{\infty} sf(s)ds,$$
$$\mathbb{E}[S \cdot (1 - X)] = \mathbb{E}[S|S < \Delta] \cdot (1 - p) = \int_0^{\Delta} sf(s)ds.$$

We next consider the delay constraint in (1). . Flows arrive as a Poisson process and go to WiFi, independently, with a certain probability p or cellular interface with $1 - p$. Hence, according to the Poisson thinning theorem [12], the Poisson nature of the arrival process to each queue is maintained. We can thus use the mean response time for M/G/1/PS to derive the mean per flow delay at the WiFi and cellular queues as

$$\mathbb{E}[T^{WF}] = \frac{1}{\mu^{WF} - \lambda p}, \ \mathbb{E}[T^C] = \frac{1}{\mu^C - \lambda(1-p)}, \tag{3}$$

where $\mu^{WF} = \frac{KR^C}{\mathbb{E}[S|S \geq \Delta]}$ and $\mu^C = \frac{R^C}{\mathbb{E}[S|S < \Delta]}$ are the respective service rates, and $\mathbb{E}[S|S \geq \Delta] = \int_{\Delta}^{\infty} d\bar{F}(s)/p$, and $\mathbb{E}[S|S < \Delta] = \int_0^{\Delta} d\bar{F}(s)/(1-p)$.

In the case of WiFi, in addition to transmission delay, we need to consider the extra delay experienced by flows due to WiFi unavailability periods. If we define p_{off} as the probability that a flow arrives during the WiFi OFF period, then $p_{off} = \frac{\mathbb{E}[T_{OFF}]}{\mathbb{E}[T_{ON}]+\mathbb{E}[T_{OFF}]}$. Thus, a percentage p_{off} of all flows sent to WiFi will have to wait until the next ON period to start transmitting. This is the expected residual time of an OFF period, which by the renewal-reward theorem can be found to be $\frac{\mathbb{E}[T_{OFF}^2]}{2\mathbb{E}[T_{OFF}]}$ [14]. Putting everything together, the mean per-flow time in the system is: $\mathbb{E}[T] = p \cdot \mathbb{E}[T^{WF}] + (1 - p) \cdot \mathbb{E}[T^C] + p \cdot D^{WF}$, which gives the first constraint in (1).

The last two constraints represent simply the allowed values for Δ, and the stability condition for the two queues ($\rho^{WF} < 1$ and $\rho^C < 1$), and the allowed values for Δ. □

The above optimization problem applies to both on-the-spot offloading (when $D^{WF} = 0$) and delayed offloading ($D^{WF} > 0$), when the use of both interfaces in parallel (when available) is allowed. It also applies to both cases where WiFi is faster or cellular is faster (e.g. 4G/4G+ or femto-cell cases).

As mentioned earlier, if a user is interested in minimizing her cellular plan usage, e.g., while roaming, one should set $L_W < L_C$ or simply $L_W = 0$, in the above formulas. In the case of energy consumption, although the cellular and WiFi interfaces have multiple power states [15], a simple "1st order" model where L_W and L_C capture the energy/bit for each interface and are given by $L_W = \frac{P^W}{R^W}$ and $L_C = \frac{P^C}{R^C}$. P^W and P^C denote the average power consumption for the WiFi and cellular interfaces, respectively[2], and could be running estimates, maintained by the device, in practice. In Section V, we consider in more detail the potential impact of detailed power transitions and potential "tail energy" issues [16].

A. Optimal Threshold

In the following, we show how we can compute the optimal solution of (1). Due to space limitations, we only consider the case of $L_C > L_W$. The opposite case is symmetric, assigning large flows to cellular, and proceeding accordingly.

Proposition 1. *Let $K = R_W/R_C$. Under any flow size distribution, the objective function in (1), namely $g(\Delta)$, is monotonic in the threshold Δ. Thus, the optimal threshold that minimizes $g(\Delta)$, namely Δ^*, is either $\Delta^* = S_m$ (for $K > 1$) or $\Delta^* = S_M$ (for $K < 1$). If $K = 1$, Δ^* can take any value in (S_m, S_M).*

It is sufficient to show that $g'(\Delta) = \frac{\partial g}{\partial \Delta} > 0$ if $K > 1$, $g'(\Delta) < 0$ if $K < 1$, and $g'(\Delta) = 0$ if $K = 1$.

$$\begin{array}{ll} g'(\Delta) & = \frac{P}{R^C} \left[\frac{1}{K} \frac{\partial}{\partial \Delta} \int_{\Delta}^{S_M} sf(s)ds + \frac{\partial}{\partial \Delta} \int_0^{\Delta} sf(s)ds \right] \\ & = \frac{P}{R^C} \left[\frac{K-1}{K} \Delta f(\Delta)) \right] = c \cdot (K - 1), \end{array} \tag{4}$$

where $c = \frac{P\Delta f(\Delta)}{R^C K}$ is always positive. Thus, if $K \geq 1$ (WiFi faster), then $g'(\Delta) \geq 0$, and the objective is minimized at $\Delta^* = S_m$. Similarly, if $K < 1$, then $g'(\Delta) < 0$, and the objective is minimized at $\Delta^* = S_M$. If $K = 1$, then the objective function is constant, and any assignment leads to the same consumption. An illustration of $g(\Delta)$ is shown in Fig. 2 (left plot).

Proposition 2. *Under any flow size distribution, the delay constraint in (1) has a unique minimum in $[S_m, S_M]$.*

Proof. We focus on the case of $K = R_W/R_C > 1$, but the argument is similar for $K \leq 1$. Consider the behavior of the delay function as $\Delta \to S_m$ (i.e., all flows are assigned to WiFi). We can distinguish the following cases: (Case 1) If the

[2]Studies suggest that this average power is similar for WiFi and cellular [6], [15], and this was also confirmed on our platform.

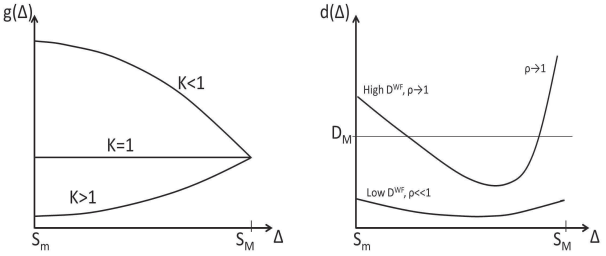

Fig. 2: Optimization function (left plot) and delay function for $K > 1$ (right plot) of the problem in (1).

system load is high (i.e. $\rho^{WF} \to 1$) the delay is high due to queueing. (Case 2) If the system load is low but D^{WF} is large, the delay for Δ values near S_m is still high. (Case 3) Both system load and D^{WF} are high. (Case 4) If both system load and D^{WF} are low then $d(S_m)$ is low. (Case 5) If both system load and D^{WF} are low, and $K \gg 1$ (cellular interface is very slow) then $d(S_m)$ is the lowest delay value. Cases 1-3 correspond to the top curve on Fig. 2 (right plot), while Case 4 to the bottom curve (Case 5 is not shown in the figure). Let us now examine potential local minima and prove that they also constitute global minima:

(*Local minimum at* $\Delta = S_m$) Then, sending a few flows to the cellular interface increases the total delay. This is only possible if the cellular interface is much slower than WiFi (Case 5). But increasing the number of flows send to the cellular interface, i.e., increasing Δ will increase the cellular interface utilization, and monotonically increase the delay. Hence, $\Delta = S_m$ is a global minimum.

(S_m *not a local minimum*) This implies that by assigning some flows to the cellular interface, we are reducing the total delay. This is either because D^{WF} is quite large (Case 2 or 3), or the WiFi interface is congested (Case 1). We can keep increasing Δ, i.e., sliding to the right along the curve $d(\Delta)$, until we find a local minimum. If this occurs at $\Delta = S_M$, then clearly S_M is a global minimum. If we stopped at an intermediate Δ value, then it means that sending one more flow to cellular would actually lead to worst total delay. Since we have assumed that the WiFi interface is faster ($K > 1$), for this local minimum to occur the marginal increase on the queueing delay of the cellular interface should outweigh the marginal decrease on the queueing plus unavailability delay of the WiFi. In other words, the delay due to the cellular load is the dominant component at that point, and the total delay will keep increasing as we further move to the right on the plot, towards S_M. Hence, this is a unique minimum. □

The following theorem, stems from Proposition 1 and 2, and gives the optimal solution Δ^* to the problem of Eq.(1).

Theorem 2. *The optimal policy for* $K = R_W/R_C > 1$ ($K < 1$) *is:*

1) *If* $D^M > d(S_m)$ ($D^M > d(S_M)$) *then* $\Delta^* = S_m$ ($\Delta^* = S_M$)*;*
2) *If* $D^M < min\{d(\Delta)\}$ *then the problem is infeasible;*
3) *Otherwise, it is the solution of equation* $d(\Delta) = D^M$*. If there are two solutions, then* Δ^* *is the smaller (larger) one, if* $K > 1$ ($K < 1$)*.*

The first case corresponds to the unconstrained problem and the optimal solution follows from Proposition 1. To find the optimal solution in the third case, when $K > 1$, Proposition 2 says that it suffices to start from $\Delta = S_m$ (where $d(S_m) > D^M$) and move down the delay constraint until $d(\Delta) = D^M$. If there are two solutions $d(\Delta) = D^M$ (see for example Fig. 2 (right plot)), the above methods will give us the smaller of the two. Similarly, for $K < 1$, but starting from $\Delta = S_M$, it picks the larger of two solutions. Note that, if $K = 1$, any feasible solution Δ (such that $d(\Delta) \leq D^M$) will do. Moreover, note that if the delay constraint is satisfied, also the load constraints in (1) are satisfied (since otherwise the delay would be infinite).

1) Case-study: Pareto distributed flow sizes: Real flow size distributions have been shown to be heavy-tailed [13]. To this end, we consider the Pareto distribution (Bounded Pareto in the simulations) as a concrete example for our optimization problem. We use S_m as the scale parameter, and α as the shape parameter (the variance is finite if $\alpha > 2$). The ccdf is $p = \mathbb{P}(S \geq \Delta) = \left(\frac{S_m}{\Delta}\right)^{\alpha}$, for $\Delta \geq S_m$. Thus, we obtain $\mathbb{E}[S \cdot X] = \frac{\alpha S_m^{\alpha} \Delta^{1-\alpha}}{\alpha - 1}$, and $\mathbb{E}[S \cdot (1 - X)] = \frac{\alpha}{\alpha - 1}(S_m - S_m^{\alpha} \Delta^{1-\alpha})$. Hence, we can derive the objective function and constraint for (1) in closed form:

$$
\begin{aligned}
min_{\Delta} \quad & L_W \cdot \tfrac{\alpha S_m^{\alpha} \Delta^{1-\alpha}}{\alpha - 1} + L_C \cdot \tfrac{\alpha}{\alpha - 1} \cdot \left(S_m - S_m^{\alpha} \cdot \Delta^{1-\alpha}\right) \\
s.t. \quad & \tfrac{\alpha S_m^{\alpha} \Delta^{1-\alpha}}{KR^C(\alpha-1) - \lambda \alpha S_m^{\alpha} \Delta^{1-\alpha}} + \tfrac{\alpha (S_m - S_m^{\alpha} \Delta^{1-\alpha})}{R^C(\alpha-1) - \lambda \alpha (S_m - S_m^{\alpha} \Delta^{1-\alpha})} \\
& + \left(\tfrac{S_m}{\Delta}\right)^{\alpha} D^{WF} \leq D^M, \\
& \tfrac{\lambda \alpha S_m^{\alpha} \Delta^{1-\alpha}}{KR^C(\alpha-1)} < 1, \tfrac{\lambda \alpha (S_m - S_m^{\alpha} \Delta^{1-\alpha})}{R^C(\alpha-1)} < 1, \\
& \Delta \in [S_m, \infty)
\end{aligned}
$$

III. Threshold Policy with hard deadline

So far we have derived the optimal size-based policy subject to a constraint on the average per-flow delay. While the formulation in Eq.(1) guarantees a desired average delay, it does not guarantee that some individual flows will not be excessively delayed, e.g. if a UE stays without WiFi connectivity for a long period. To address this issue, each flow assigned to WiFi could also have a strict deadline, e.g. user defined or application specific. Some flows could even have a deadline of 0, if they cannot be offloaded. Each new flow assigned to WiFi, will wait in the WiFi queue only up to its deadline. If this expires, it will be sent through the cellular interface. Such per flow strict deadlines are used, for example, in [6], [8].

If such strict deadlines are implemented, Eq.(1) needs to be revisited, in order to account for the potential re-routing. Let d^{WF} be a random variable denoting the delay a flow experiences when assigned to the WiFi interface. This delay depends on the duration of the WiFi unavailability periods as well as the queueing delay. If d_i^T denotes the strict deadline for some flow i, then the probability that this flow is sent back to cellular is clearly equal to $P(d^{WF} > d_i^T)$. Since, we are optimizing performance over a time window with many flows, we need

an estimate of the above probability. If D^T is the average over all individual deadlines d_i^T, then the expected ratio of flows rerouted to cellular will be $1 - p_d = P(d^{WF} > D^T)$.

Probability p_d can be approximated, in closed form, if we further assume that transmission delay on the WiFi is much smaller than the average unavailability period (this is a reasonable assumption in most scenarios, including the ones considered in our simulations and experimental validation). In that case, d^{WF} is the "excess" or "residual" time of a WiFi OFF period, distributed according to some $F_{off}(x)$ with mean value $\mathbb{E}[T_{off}]$. Using the formula for excess time distribution [14] we get

$$p_d = \left(\frac{1}{\mathbb{E}[T_{off}]}\right) \int_0^{D^T} (1 - F_{off}(x))\, dx. \tag{5}$$

Based on this probability, we can now reformulate the optimal TP policy for scenarios with strict per flow deadlines.

Theorem 3. *Assume that flows arriving in the system of Fig. 1 have an average delay requirement D^M. Assume further that each flow additionally has a strict deadline, as described earlier, and that the average value among all such deadlines is D^T. Then, among all the flow-assignment policies, the size-based policy with threshold Δ given by the following optimization problem, gives the minimum possible cost per flow among all policies satisfying both the average and individual deadlines.*

$$\begin{aligned}
min_{\Delta} \quad & L_W \cdot p_d \textstyle\int_{\Delta}^{\infty} d\bar{F}(s) + L_C \cdot \left[(1-p_d)\int_{\Delta}^{\infty} d\bar{F}(s) + \int_0^{\Delta} d\bar{F}(s)\right] \\
s.t. \quad & \frac{p_d}{\frac{R^W}{\int_{\Delta}^{\infty} d\bar{F}(s)} - \lambda \cdot p_d} + \frac{1-\bar{F}(\Delta)\cdot p_d}{\frac{R^C}{\int_0^{\Delta} d\bar{F}(s) - p_d \int_{\Delta}^{\infty} d\bar{F}(s)} - \lambda(1-\bar{F}(\Delta)\cdot p_d)} \\
& + \bar{F}(\Delta)\left[(1-p_d)\cdot D^T + p_d \cdot p_{off} \cdot D^T\right] \leq D^M, \\
& \frac{\lambda \cdot p_d \cdot \int_0^{\infty} d\bar{F}(s)]}{R^W} < 1, \frac{\lambda(1-\bar{F}(\Delta)\cdot p_d)\cdot[\int_0^{\Delta} d\bar{F}(s) - p_d \cdot \int_{\Delta}^{\infty} d\bar{F}(s)]}{R^C} < 1 \\
& S_m \leq \Delta \leq S_M
\end{aligned} \tag{6}$$

Sketch of Proof. While the above formulation appears less intuitive than the simpler case without strict deadlines, we sketch here the main differences and refer the reader to the proof of Theorem 1 for comparison: A percentage $(1 - p_d)$ of "large" flows get rerouted to cellular, adding an additional $(1 - p_d) \cdot \mathbb{E}[S \cdot X]$ bits to the cellular interface, where $\mathbb{E}[S \cdot X] = \int_{\Delta}^{\infty} d\bar{F}(s)$ is the average size for flows routed to WiFi initially. This is easy to see in the objective. Furthermore, when calculating the expected delays, these rerouted flows must be discounted from the load of the WiFi. The total arrival rate is now $\lambda \cdot p_d \cdot \bar{F}(\Delta)$ which is smaller than the WiFi arrival rate $\lambda \cdot \bar{F}(\Delta)$ in the original formulation of Theorem 1. The service rate μ^{WF} remains unchanged, since the average flow size assigned to WiFi is still $\mathbb{E}[S|S \geq \Delta]$. Finally, the queueing delay of the WiFi interface $(\mu^{WF} - \lambda \cdot p_d \cdot \bar{F}(\Delta))^{-1}$ must be weighed by $p_d \cdot \bar{F}(\Delta)$, since this is the number of flows finally served by WiFi (Note that $\bar{F}(\Delta)$ is cancelled out from both the numerator and the denominator, as in the case of Eq.(1)). The queueing delay for the cellular interface is slightly more difficult to calculate, because the service rate μ^C now also changes: a percentage $F(\Delta)$ of files served are small ($\mathbb{E}[S|S < \Delta]$), as before, but an additional percentage $p_d \cdot \bar{F}(\Delta)$ are large files ($\mathbb{E}[S|S \geq \Delta]$) that were initially assigned to WiFi, but their deadline expired. Finally, we approximate the average WiFi unavailability delay, $D^{WF} = \mathbb{E}[d^{WF}|d^{WF} \leq D^T]$, with its upper bound $p_{off} \cdot D^T$. □

Remark: As a final note, we stress here that using D^T in the formulation does not mean we require all flows to have the same strict deadline D^T , or that the policy only guarantees yet again an average delay: a flow will always get rerouted to cellular as soon as its deadline expires, no matter what the choice of Δ is. This is also visible in the simulation results. What the above policy does is to estimate that percentage of flows rerouted (for which it uses the average value D^T), and its impact on the objective and mean delay constraint, in order to still choose the best threshold Δ, costwise.

IV. Performance evaluation

In this section, well use simulations to study the performance of the Threshold Policy (TP), considering several scenarios and network conditions. To focus on realistic scenarios, we will use the results and observations made in two studies of real users and measured WiFi and cellular connectivity: a study of (mostly) pedestrian, low mobility users performed in [6], which we will be the basis of most our scenarios, and a second one studying a trace of vehicular users [7] that we consider in Section IV-C.

Specifically, unless otherwise stated, we assume that the *average* duration of WiFi availability period (ON) is 110 min, while the average duration of OFF period is 45 min [6].[3] Thus, we obtain $p_{off} \simeq 0.3$ and $D^{WF} \simeq 13$ min (assuming that OFF periods are exponentially distributed). Throughout this section, we focus on the specific cost function of energy comsumption, due to its importance for modern smartphones, where we use the simple 1st order energy model described earlier: $L_W = P^W/R^W$ and $L_C = P^C/R^C$, measured in energy/bit. Note however, that depicted results could be easily read in terms of other costs (e.g. monetary cost) with an appropriate change in units and relative costs. We set: $P = 1$ Watt [20], $R^C = 0.5$ Mbit/s, as reported in [6]. This value depends on the cellular technology and we are targetting scenarios where cellular cannot always offer peak rates (e.g., we have measured higher values in our experimental platform, but we did so using a business-class plan).

The flow arrival rate is $\lambda = 0.004$ flows/s, and $D^M = 6$ min (or 360 sec). We consider flow sizes following the Bounded Pareto (BP) distribution with support in the interval $[S_m, S_M]$, with $\mathbb{E}[S] = 10$ MB. We compare our TP policy with some other baseline policies. Specifically, we consider the following ones:

[3]These values capture "nomadic" groups of users. We stress that we have tried a number of ON/OFF value combinations, with similar results. We thus prefer to focus on the values reported in that real study, and juxtapose it to the high mobility scenario of Section IV-C.

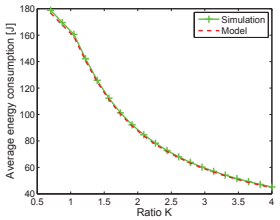

Fig. 3: Average energy vs K.

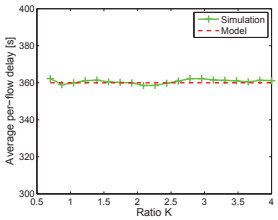

Fig. 4: Average delay vs K.

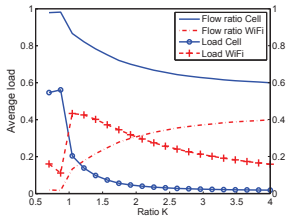

Fig. 5: Average load and flow ratio vs K ($D^M = 360$ s).

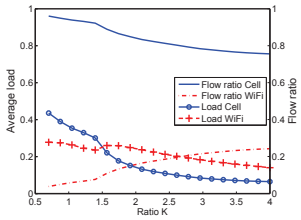

Fig. 6: Average load and flow ratio vs K ($D^M = 260$ s).

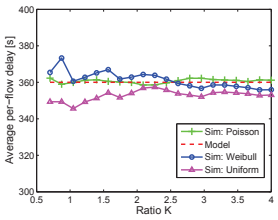

Fig. 7: Average delay vs K for different flows arrival distributions.

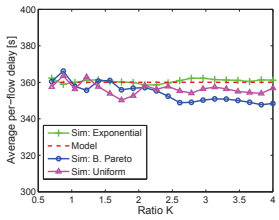

Fig. 8: Average delay vs K under different WiFi ON-OFF times distributions.

1) "Cell-only": all flows are sent over the cellular interface;
2) "WiFi-only": all flows are sent over the WiFi interface;
3) "On-the-spot": flows go to WiFi when it is available, as in the existing offloading policy;
4) "FB" (Flow Balancing): it balances the number of flows per interface regardless of the size, thus exploiting the parallel availability of the interfaces;
5) "LB" (Load Balancing): it balances the load between the two interfaces, thus improving the queueing delay [12].

A. Model validation

In this section, we want to validate our model. Wherever we refer to "model" or "analytical" plots, these plots correspond to the analytical predictions, for the average cost and per flow delay, derived in Theorem 1. For simulations, the optimal threshold Δ is calculated with the methodology of Theorem 2, and this threshold is used, simulating the queueing system of Fig. 1, and whose assumptions might depart from those made in the model (we will explain how in each scenario).

Fig. 3 compares the model-based and simulated average energy consumption vs. different values of K in the range $[0.7 - 4]$. As expected, we can observe that the energy consumption decreases as K increases, as higher WiFi rates imply less time to transmit the same amount of data, and thus less energy. Fig. 4 further reports the corresponding average delay and compares it to the theoretical delay requirement $D^M = 360$ s. It is important to note that the simulated results match well the analytical model predictions.

To better understand the energy consumption trend versus K, we report the average load (ρ^W and ρ^C) at the two interfaces in Fig. 5 (left y-axis). Observe how, for $K < 1$, the TP uses more the cellular network (cellular load is higher than the WiFi one), while for $K > 1$ it exploits more the WiFi network. Fig. 5 (right y-axis) also reports the corresponding flow ratio, i.e., the number of flows that go over WiFi (cellular) divided the total number of flows. In the model, this flow ratio corresponds to the probability p_d and $1-p_d$, respectively. Interestingly, we can note that for $K > 1$ only a few large flows are sent/received over WiFi contributing the most to load. E.g, for $K = 2$, about 20% of the (WiFi) flows contribute the 80% of the system load (total load is $\rho^W + \rho^C$), thus exhibiting a 80-20 rule.

This behavior, which is a key reason why the TP policy outperforms other policies, as we will see, becomes even more pronounced as the delay requirement D^M decreases: the tighter the user delay requirement, the lower will be the number of flows sent through WiFi. As an example, Fig. 6 shows the load and flow ratio for a smaller delay constraint, $D^M = 260$s. Observe that the percentage of flows assigned to WiFi is only 24%, for $K = 4$. Nevertheless, considerable cost savings can be achieved, because these few flows amount for a large chunk of the total bytes transmitted.

1) Validation versus assumptions: In Section II, we assumed flows arrive according to a Poisson process. Here we depart from this assumption and simulate our system for (a) flow arrivals according to a Weibull distribution (shape parameter $h = 0.7$), and (b) uniformly distributed arrivals. Weibull is used to emulate more burst arrival patterns, and was found to be a good model for traffic arrivals in [6], while uniform has lower variability than both Poisson and Weibull ("regular" arrivals). Fig. 7 shows the average delay under the TP for Poisson, Weibull (the shape parameter is $h = 0.7$) and Uniform flow arrival distributions. The average delay is comparable in all cases, suggesting that our model does not underestimate the predicted delay, leading perhaps to major constraint violations, if the arrival model is different.

Furthermore, we simulate the system under different WiFi ON-OFF time distributions: Exponential, Uniform and Bounded Pareto (as in [6]). Fig. 8 shows the average delay under the considered WiFi availability patterns. Observe how simulation results under different distributions are comparable, with the average delay sligthly smaller for the Uniform and Bounded Pareto cases.

B. Threshold Policy gains

We will now compare the performance of TP against other policies. The relative energy savings are depicted in Fig. 9. Observe how the TP outperforms all the others in terms of energy except of WiFi-only policy. This is expected, as the latter is the optimal *unconstrained* policy for $K > 1$. However, the WiFi-only policy, as well as the rest of the policies *violate the delay constraint, sometimes significantly*. Fig. 10 reports

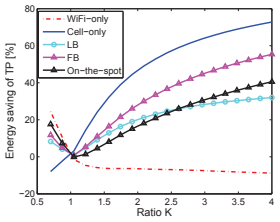

Fig. 9: Energy saving of TP vs K.

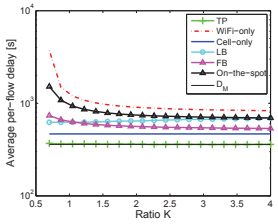

Fig. 10: Average delay of TP vs K.

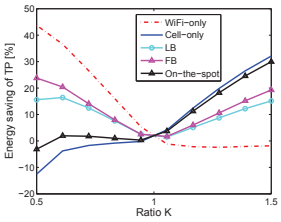

Fig. 11: High-mobility scenario: Saving vs K.

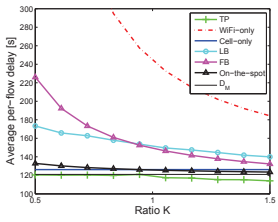

Fig. 12: High-mobility scenario: Average delay vs K.

the average delay for each policy. Note the logarithmic scale on the y-axis, implying up to an order of magnitude worse delays. Note also that the FB and LB policies outperform the baseline options (i.e., Cell-only, WiFi-only and On-The-Spot) in terms of energy-delay trade-off, since they make an attempt to balance the load between the two interfaces. However, as proven in Section II, our policy can do better, by also taking into account WiFi unavailability statistics and flow size variability. In fact, the higher the flow size variability the more TP outperforms the other policies.

Summarizing the results of Sections IV-A and IV-B, they show that: 1) for $K > 1$, the TP policy alleviates the cellular load, dropping utilization from 0.6 to less than 0.1 (more than a $6\times$ improvement); 2) energy consumption compared to the Cell-only policy is very low (e.g. 70% of energy saved for $K = 4$), and always lower than other load-balancing policies; 3) the above savings can be achieved by offloading to WiFi only a small subset of (large) flows (as little as 10%, in the scenarios considered); even fewer of them will actually experience an extra delay due to WiFi unavailability, since only a subset of flows arrive during an OFF period.

C. High-mobility case-study

To complement our results, we also study the performance of TP in a high-mobility scenario (i.e., lower duration for ON and OFF periods), motivated by the measurement study of [7]. In addition to the higher mobility, this scenario is interesting because of the more pessimistic conditions for offloading reported there, due to lower data rates for WiFi and lower availability (this is also partly due to slow WiFi rate adaptation and association mechanisms, not optimized for vehicular mobility). Specifically, reported downlink rates of WiFi and cellular (3G) are 280 Kbps and 600 Kbps, respectively. Thus, $K \sim 0.5$. Moreover, WiFi availability can drop to 11% (e.g., $p_{off} \simeq 0.9$, $D^{WF} = 132$ s). For this scenario, we assume $\mathbb{E}[S] = 5$ MB (variance of 650 MB) and a shorter delay constraint, namely $D^M = 121$ s. Fig. 11 and 12 show the average energy saving and the corresponding delay for this scenario, as a function of K, ranging from $K = 0.5$ (WiFi slightly worse) to $K = 1.5$ (WiFi slightly better). Note that, to improve the readability of Fig. 12 the y-axis is upper limited (e.g., the delay of WiFi-only is equal to 1000 s for $K = 0.5$).

It can be observed that, for $K > 1$ the TP always outperforms the LB, FB, and cell-only in terms of both energy and delay, and performs better than the WiFi-only policy in terms of delay. For $K < 1$, as expected, the best policy in terms of energy is the Cell-only one. However, we can note that, even if WiFi rate is smaller than the cellular one and WiFi availability is low, the TP still exploits the WiFi network by using both interfaces in parallel to reduce the transmission delay, thus keeping the average delay below the constraint D^M. In addition, the higher is the load, the more traffic will be offloaded from cellular to WiFi network, when available. Note that the high-mobility case is a worse mobility scenario, because of poorer WiFi availability and lower WiFi rates, due also to rate adaptation and association mechanisms.

D. Threshold Policy extensions

1) Hard-Deadline case: In Theorem 3, we considered a system where each WiFi flow has a hard deadline (assigned by the user or application) after which it is sent/received over the cellular interface, if not yet transmitted/received through WiFi. Hence, flows have an average delay requirement $D^M = 360$ s, as before, but now also have a maximum delay requirement D^T, which we set equal to 900 s for all flows (we stress again that our policy does not require equal flow deadlines, as explained in Section III). The former bounds the average delay among flows, while the latter bounds the tail. To illustrate this, Fig. 13 reports the CDF of the per-flow delay, for the policy of Theorem 1 without the deadline ("No HD" label) and the policy of Theorem 3 with hard-deadline ("HD" label), with $D^T = 900s$ depicted as a vertical line.

In the case of no hard deadline, while the respective assignment maintains the average delay bounded (to 360s), we can see that 30% of the flows exceed the maximum deadline D^T. Specifically, the 5% of flows with the highest delay have a mean value of about 9900 s, i.e., exceed D^T by more than $10\times$. In contrast, when the policy takes into account the hard-deadline (HD plot), we can observe that the tail of the distribution is correctly bounded by D^T.[4] To ensure this, the HD policy is forced to be a bit more conservative, and thus results in slightly lower average delay as well (334 s). Also, as more flows now go over the cellular interface, the energy consumption is 13% higher in the HD case.

Specifically, the 5% of flows with the highest delay has an average delay of 1212 s in case of HD, and this delay is about 9917 s (more than ten times higher than D^T) in case of no deadline. Finally, the average delay is slightly lower than D^M

[4]A small percentage of flows slightly exceeds this value due to the transmission delay approximation, explained in Section III.

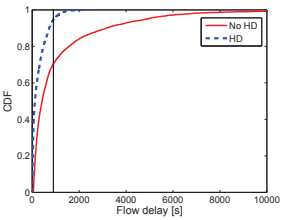

Fig. 13: Cumulative distribution function of the per-flow delay.

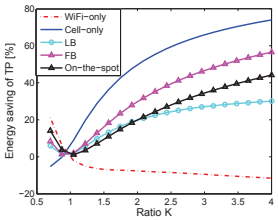

Fig. 14: Energy saving considering different network power models.

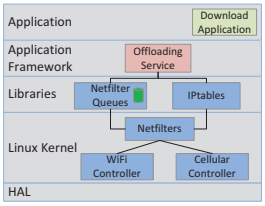

Fig. 15: System view of the prototype.

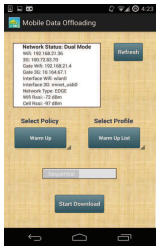

Fig. 16: Application GUI.

in the HD case (334 s), while it is equal to $D^M = 360$ s in the case with no deadline. Concerning the energy consumption, it is 13% higher in the HD case. In this example and in all the cases we considered, we noted that there is a trade-off between energy consumption and delay (our QoS metric). This is due to the fact that, in the HD case, the tighter is D_T (i.e. stricter QoS constraint) the lower is the average per-flow delay at the penalty of an higher energy consumption. However, this behavior depends on several factors: if the bottleneck is the cellular network service time (e.g. high load) then the stricter the deadline is the more the TP policy will exploit the WiFi network; otherwise, if the bottleneck is WiFi (e.g. high unavailability delay) the more the TP policy will exploit the cellular network.

2) Energy model approximation: As we have so far only considered a simple energy model, we conclude our simulation scenarios by considering a more realistic energy model, and its impact on TP. Specifically, we implemented the detailed 3G and WiFi power state and transition models represented in Fig.1 of [16]. We defer to future work the modeling of the more sophisticated 4G power management. The 3G model consists of three states: High-power state (DCH) in which the interface is transmitting/receiving, low-power state (FACH), and Idle state. WiFi model also consists of three states: Active, PSM (Power Saving Mode) and Idle. Transition timeouts and delays between states are also implemented, in order to investigate if tail energy phenomena might lead to largely different energy consumption than the one assumed by TP.

To this end, Fig. 14 reports the relative energy savings of the TP policy computed using the energy model from [16]. One can observe that simulation results are very similar to the ones obtained in Section IV-B (see Fig. 9 for comparison), thus corroborating the applicability of our model. Indeed, while our model does not account for tail/transition energy consumption, it can be seen as a worst-case model since it assumes the interface (cell./WiFi) is always either in high power (transmitting/receiving) state, or Idle.

V. Implementation and Experiments

To further validate the proposed TP in realistic conditions, we develop a prototype of the proposed offloading policy using an Android-based mobile OS, CyanogenMod (v10.1) [22], and modify the Connectivity Service to enable simultaneous usage of WiFi and cellular interfaces. Fig. 15 illustrates the offloading application prototype composed of a download application and an offloading service. The application will use the default interface to send and receive user traffic. To decouple the WiFi variability from the application, this is set to be the cellular interface, but it could also be a virtual interface as described in [3]. The offloading service determines the optimal threshold based on the passive rate measurements of both WiFi and cellular computed through a moving average based on estimates of past flow transmissions, then marks the outgoing/incoming traffics in Linux Netfilters using `iptables` utility, and add appropriate changes in the routing tables using `ip` utility to enable match-action of flows in post (re)routing phase. For the delayed offloading, when a filter is matched, packets will be stored in the Netfilter queues for later transmission.

In our experiments, we use Google Nexus 4 with superuser privilege downloading 400 files across 6 servers located in USA (with sizes in the range $[18.19Mbits, 358.45Mbits]$). The average rate measured with cellular connection (HSPA+) is about 5 Mbps, and the average WiFi rate is about 7.5Mbps, thus $K \simeq 1.5$. We note that this cellular rate is very high, due to access to a nearby underutilized BS with a business class plan, and is not representative of actual cellular connections that are typically much slower (as in Section IV). The arrival rate of new requests for downloads λ to 0.15 flows/second

In the application, whose GUI is shown in Fig. 16, we implement three policies, namely TP, WiFi-only, and Cell-only. When the TP policy is selected, upon arrival of a file request, the target interface will be selected based on the file size and the measured rates. While for uplink the file size is known to the user, in downlink this is obtained by sending an additional request to the server via the default interface (worst case scenario). For each policy, the downloading application collects the key performance indicators (KPI) in terms of percentage of battery consumption, average throughput, average per-flow delay, and connection rates. The average delay for TP includes the waiting time to get the file size from the server.

We present here some preliminary results from our prototype. In the left plot of Fig. 17, we present the average flow delay for different size threshold values Δ. We observe the single minimum behavior discussed in Proposition 2 and shown in Fig. 2. The percentage of bytes sent on each interface is also depicted in Fig. 17(right plot). E.g., for the Δ value achieving minimum delay (highlighted with a red circle), around 40% are sent over the cellular, while 60% is sent on

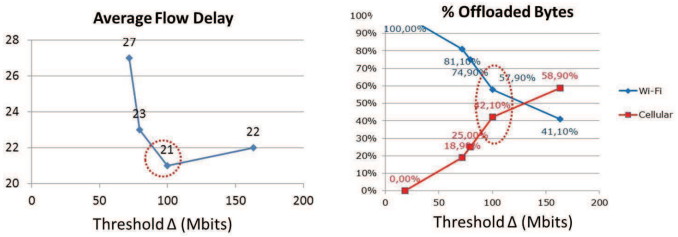

Fig. 17: Average flow delay (left) and % of offloaded data (right) for the Android experiment.

TABLE I: Experimental results

Policy	Mean delay [s]	Battery Consumption (%)
TP	23.23	1
WiFi-only	40	8
Cell-only	115	20

the WiFi. However, as predicted by the theory, this is achieved by only sending 27% of the total flows to WiFi.

For this setup, we ran our TP policy to calculate the optimal threshold. The average delay requirement was $D^M = 23$s), hence the optimal threshold from an energy point of view is slightly higher than the one minimizing the delay (we remind the reader that our goal is not to minimize delay, but rather the cost function, subject to the delay constraint). Table I compares (i) the total battery consumption to download all files, and (ii) the average per flow delay, for three policies: TP, WiFi-only, and Cell-only policies.

While these results are preliminary, sensitive to our crude online rate estimates, and other components consuming battery, we can already observe the main behaviors predicted in theory. Namely, TP manages to download all files with 50% energy savings compared to Cell-only. WiFi-only has somewhat lower energy consumption, as our theory predicts (given the higher WiFi rate, the unconstrained optimal policy is to download all files from WiFi.). However, WiFi-only had $2\times$ higher delay than TP, and Cell-only more than $5\times$.

VI. Discussion and Conclusion

We have proposed a threshold policy that assigns flows to both cellular and WiFi network interface based on their size, and shown that it can minimize a cost function related to battery consumption and plan usage at the UE, while keeping the average per-flow delay bounded. Results obtained from simulations and an Android-based prototype support our assumptions and validate our policy. Among a number of interesting future directions, we briefly state two here.

Algorithm convergence: While TP considers the impact of other users through variations on the measured interface rates, if TP is implemented on many UEs associated with the same BS-AP tuple, all UEs might switch *large* amounts of traffic, *at the same time*, possibly leading to oscillations. To avoid such phenomena one could introduce one or both of the following: (*low pass filter for* Δ) if conditions change such that TP suggests a different, much smaller Δ than the one used (e.g. better WiFi is found), the UE does not immediately use this new threshold, but only reduces its Δ by some constant amount; Also, some *randomization* can be introduced between when a UE measures radio conditions, and when it decides to apply its policy. While this improves convergence, a game-theoretic analysis could possibly reveal whether the point of convergence is indeed the optimal.

Network-side implementation: To achieve better load balancing, and resolve convergence issues, the policy could be implemented by the operator. The operator could observe all flows coming from all users connected in one or more $\{$BS, AP$\}$ sets, and calculate a threshold that is suggested (or enforced) for each user, dependent on both network-wide conditions but also user load and profile.

References

[1] http://www.shoelacewireless.com.
[2] B. Wang, W. Wei, Z. Guo, and D. Towsley, "Multipath Live Streaming via TCP: Scheme, Performance and Benefits", in ACM CoNEXT 2007.
[3] K.-K. Yap et al., "Making Use of All the Networks Around Us: A Case Study in Android", in CellNet 2012.
[4] http://www.heavyreading.com/document.asp?doc_id=192489.
[5] F. Mehmeti et al., "Performance Analysis of Mobile Data Offloading in Heterogeneous Networks", in IEEE Transactions on Mobile Computing, 2017.
[6] K. Lee et al., "Mobile Data Offloading: How Much Can WiFi Deliver?", in IEEE/ACM Transactions on Networking, 2013.
[7] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting Mobile 3G Using WiFi", in MobiSys 2010.
[8] F. Mehmeti, and T. Spyropoulos, "Is it Worth to be Patient? Analysis and Optimization of Delayed Mobile Data Offloading", in IEEE Infocom 2014.
[9] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang, "Incentivizing Time-Shifting of Data: A Survey of Time-Dependent Pricing for Internet Access", IEEE Communications Magazine, November 2012.
[10] Y. Im et al., "AMUSE: Empowering Users for Cost-Aware Offloading with Throughput-Delay Tradeoffs", in IEEE Infocom 2013.
[11] M. Laner et al., "A comparison between one-way delays in operating HSPA and LTE networks", in WINMEE 2012.
[12] M. Harchol-Balter, "Performance Modeling and Design of Computer Systems: Queueing Theory in Action", Cambridge University Press, 2013.
[13] X. Meng, S. Wong, Y. Yuan, and S. Lu, "Characterizing Flows in Large Wireless Data Networks", in ACM MOBICOM 2004.
[14] S. M. Ross, "Stochastic Processes", 2nd ed. John Wiley & Sons, 1996.
[15] A. Sharma et al., "Cool-Tether: Energy efficient on-the-fly wifi hot-spots using mobile phones", in ACM CoNEXT 2009.
[16] N. Ding et al., "Characterizing and Modeling the Impact of Wireless Signal Strength on Smartphone Battery Drain", in ACM SIGMETRICS 2013.
[17] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications", in ACM IMC 2009.
[18] A. De la Oliva et al., "IP Flow Mobility: Smart Traffic Offload for Future Wireless Networks", IEEE Communications Magazine, October 2011.
[19] G.J. Hoekstra et al., "Engineering Elastic Traffic in TCP-Based Networks: Processor Sharing and Effective Service Time", in ITC 25, 2013.
[20] Y. Xiao et al., "Modeling Energy Consumption of Data Transmission over Wi-Fi", IEEE Transactions on Mobile Computing, 2013.
[21] Shuo Deng, Hari Balakrishnan, "Traffic-aware techniques to reduce 3G/LTE wireless energy consumption," in ACM CoNEXT 2012.
[22] CyanogenMod, http://www.cyanogenmod.org/.
[23] I. Rai, G. Urvoy-Keller, and E. Biersack, Analysis of LAS scheduling for job size distributions with high variance, ACM SIGMETRICS Performance Evaluation Review, 2003.
[24] F. Mehmeti, and T. Spyropoulos,"Performance modeling, analysis, and optimization of delayed mobile data offloading for mobile users," in ACM/IEEE Trans. on Networking, 2017.
[25] H. Zhou, et al., "A Survey on Mobile Data Offloading Technologies," in IEEE Access, vol. 6, pp. 5101-5111, 2018.