SmartCut: Mitigating 3G Radio Tail Effect on Smartphones

Guangtao Xue, *Member, IEEE*, Hongzi Zhu, *Member, IEEE*, Zhenxian Hu, *Student Member, IEEE*, Jiadi Yu, *Member, IEEE*, Yanmin Zhu, *Member, IEEE*, and Gong Zhang, *Member, IEEE*

Abstract—3G technology has stimulated a wide variety of high-bandwidth applications on smartphones, such as video streaming and content-rich web browsing. Although having those applications mobile is quite appealing, high data rate transmission also poses huge demand for power. It has been revealed that the tail effect in 3G radio operation results in significant energy drain on smartphones. Recent fast dormancy technique can be utilized to remove tails but, without care, can degrade user experience. In this paper, we propose a novel scheme *SmartCut*, which effectively mitigates the tail effect of radio usage in 3G networks with little side-effect on user experience. The core idea of SmartCut is to utilize the temporal correlation of packet arrivals to predict upcoming packets, based on which unnecessary high-power-state tails of radio are cut out leveraging the Fast Dormancy mechanism. Both prototype experiment and extensive trace-driven simulation results demonstrate the efficacy of SmartCut design. On average, SmartCut can save up to 43 percent network energy while having little side-effect to user experience.

Index Terms—3G networks, radio tail, fast dormancy mechanism, temporal correlation, smartphones, user experience

1 Introduction

Smartphones equipped with powerful CPUs and 3G wireless communication modules have gained a large popularity nowadays, which stimulate a large spectrum of high-bandwidth applications such as video streaming, online games and content-rich web browsing booming on smartphones. As the capability of smartphones keeps soaring, battery technology remains a bottleneck. Having those applications mobile is fun but how to preserve energy on smartphones is of great importance. Recently, the tail effect of 3G interface has gained much attention, which refers to the interface will keep at high-power states for a long time even after the completion of data transmission. Studies [1], [2] have reported that the tail effect of 3G interface can consume up to 60 percent of the total energy.

To eliminate the tail effect in 3G networks, however, is very challenging due to two reasons. First, the tails are designed to achieve fast response to upcoming data transmissions. Simply cutting the tails without care would cause non-negligible delays since the interface needs time to switch from low-power (idle) states to high-power (data-transmission-ready) states each time when the interface is not ready for the required data transmission. Long promotion delay would seriously degrade user experience. Second, in order to determine energy-efficient tails, it is inevitable to know the precise data workload in the future. In practise, this is very hard, if not impossible.

In the literature, several schemes have been proposed to reduce the impact of tail effect, trading off between minimal energy waste and good user experience. Basically, those schemes fall into two categories: *tail-cutting* and *tail-sharing*. Given the knowledge of future packets, tail-cutting schemes [3] remove the high-power-state tail by adopting the fast dormancy mechanism, under which a smartphone can ask for an immediate release of radio links and leave the high-power-states rapidly. Knowing the information of future packets, however, is infeasible in practice. In contrast, tail-sharing schemes [1], [4] re-schedule packets from different applications and pack packets together in order to share tails. These schemes can cause packets delayed and therefore are effective only when the applications running on a smartphone are delay tolerant but fail for real-time or time sensitive applications. As a result, there exists no successful solution, as far as we know, to resolving the problem of tail effects in 3G networks.

In this paper, we first collect a real trace of 3G network traffic from a metropolis in China, involving more than 65,000 users over a month. By analyzing the empirical data, we confirm that the total tail duration accounts for nearly 50 percent of the channel holding time and the tail effect contributes more than 60 percent to the total power consumption used for data communication. Furthermore, in order to find whether there exist clear patterns embedded in 3G traffic, we examine the packet arrival time from traffic generated by three categories of popular applications, i.e., video streaming, web browsing and instant messaging. We find that 3G user traffic shows strong temporal correlation.

Inspired by this observation, we propose an innovative scheme, called SmartCut, which integrates three key techniques to mitigate the tail effect of 3G networks while keep

- *G. Xue and Y. Zhu are with the Shanghai Key Lab of Scalable Computing and Systems, Shanghai, China, and the Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China. E-mail: {gt_xue, yzhu}@sjtu.edu.cn.*
- *H. Zhu, Z. Hu and J. Yu are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China. E-mail: {hongzi, zhenxian, jdyu}@sjtu.edu.cn.*
- *G. Zhang is with Huawei Research, Huawei Technologies co., Shenzhen, China, 518129. E-mail: nicholas.zhang@huawei.com.*

Manuscript received 25 Mar. 2013; revised 10 Dec. 2013; accepted 12 Mar. 2014. Date of publication 23 Mar. 2014; date of current version 26 Nov. 2014. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2014.2313339

most applications running on smartphones un-affected. The core idea of SmartCut is to train an autoregressive move average (ARMA) model using historical 3G traffic trace, based on which SmartCut consecutively predicts the arrival time of upcoming packets. With the estimated packet arrival time, SmartCut first adopts the fast dormancy mechanism to cut the unnecessary high-power-state tails off. Moreover, it also promotes the 3G interface in advance before the next packet arrives. In the case that SmartCut makes a wrong prediction, variation of past estimations is utilized to reduce the influence. The advantage of SmartCut is two-fold. First, SmartCut can significantly reduce the energy waste caused by tail with little side-effect on user experience. Second, SmartCut is lightweight and easy to implement without the requirement of any changes to upper layer applications. We implement a prototype on Android-based smartphones and conduct small-scale experiments. The experiment results on the prototype demonstrate the feasibility of SmartCut. Furthermore, we conduct extensive trace-driven simulations based on the large 3G trace that we have collected and results show that SmartCut can save 43 percent network energy on average among different applications.

We highlight our main contributions in this paper as follows:

- We have collected a large 3G traffic trace of over 65,000 users for a month and analyzed the packet arrival patterns. We find that the data traffic of 3G mobile applications have clear temporal correlation.
- We have proposed a novel scheme, SmartCut, which effectively utilizes the temporal correlation of packet arrivals to reduce the energy waste caused by the tail effect while has little side-effect on user experience.
- We have implemented a prototype on Android-based smartphones adopting the fast dormancy mechanism. The prototype implementation has verified that SmartCut is lightweight and easy to implement without the requirement of any changes to upper layer applications.
- We have conducted extensive trace-driven simulations with our trace, the results show that SmartCut can achieve a significant power saving up to 43 percent on average.

The remainder of the paper is organized as follows. Section 2 presents the related work. Section 3 introduces the system model. In Section 4, we describe the collected 3G data trace and then analyze the impact of the tail effect. We also characterize the features of the trace. The SmartCut scheme design is presented in Section 5. Section 6 describes the prototype implementation. In Section 7, we describe the methodology to evaluate the performance of SmartCut and present results. Finally, we give a conclusion of our work in Section 8.

2 Related Work

A thorough understanding of cellular radio energy characteristics helps to achieve more energy-efficient fashions for utilizing cellular radio in smartphone. The tail effect, where the cellular radio remains in a high power state for a long duration, contributes a large fraction of total radio energy consumption. How to decrease the tail effect has received significant attention.

Radio resource control (RRC) state machine and energy Modeling. A cellular radio switches its states according to Radio Resource Control protocol maintained by smartphones and the cellular network [5]. Different states denote different capabilities with diverse power characteristics. 3GPP gives the definite RRC specification for each cellular system while each operator has its own implementation or settings [6]. Qian et al. [2] used an active measurement to infer parameters of RRC state machine and showed two operators with different promotion or demotion patterns and widely varying timer values. These measurements showed that the tail time results in significant energy drain on smartphones. Huang et al. [7] extended its investigation to 4G LTE network and discovered the tail time was nearly 12 seconds.

Analytic models have been proposed to compute the energy consumption of 3G radios. Authors in [8] compute the energy cost of using different inactive timers by modeling traffic and 3G radio characteristics. Pathak et al. [9] use system call tracing to perform fine-grained modeling of energy consumption.

Optimizing the tail. The inactivity timer values are chosen in an ad hoc manner, trading off the user experience, energy consumption and signalling overhead. Several papers have also proposed to dynamically choose idle timer values based on traffic characteristics instead of static idle timer values used currently [10]. Falaki et al. [11] show that 95 percent of inter-packet arrival times lie within 4.5 seconds by analyzing smartphone traffic from a large-scale data set, indicating a shorter idle timer value. Tailender [1] combined delaying transfer of time insensitive sync application and prefetching of search query results in order to reduce energy. Bartendr [12] predicted an appropriate time to transfer data under better signal strength. For delay-tolerant applications such as email and video streaming, which permit flexible communication scheduling, we can pack several data packets into a single one and let them share the same tail time. However, this kind of approaches is applicable only under few scenarios. Users may encounter a number of deferred responses when using interactive applications such as web browsing and IM.

Applying fast dormancy. Fast dormancy had been discussed through several editions of 3GPP and implemented by some handset manufacturers [6], [13]. Several recent smart phones use an idle timer ranging from 3 to 10s to invoke fast dormancy [14], a much smaller value than the default 12-20 s tail duration settings [2]. TOP [3] modified applications to leverage fast dormancy in order to save energy. A more recent paper [15] mines program execution traces to predict the end of communication spurts, thus accurately invoking fast dormancy. Still,we find big room for further improvement. If the time next packet arrives can be predicted, we can promote the 3G interface in advance.

3 System Model

In 3G networks, the establishment and release of a radio link between the interface of user equipment (UE) and the base

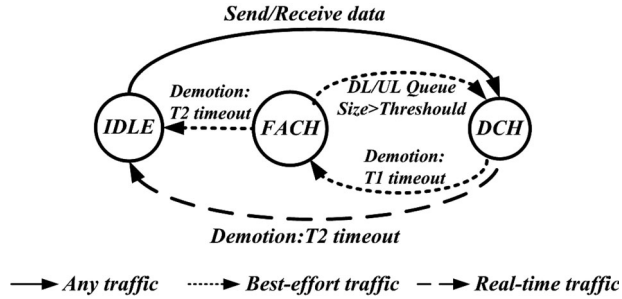


Fig. 1. State transition graph.

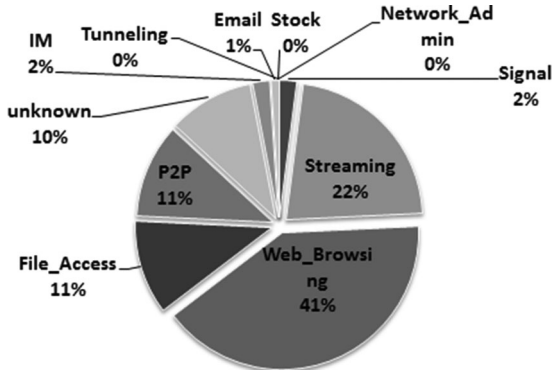


Fig. 2. Throughput pie chart for all applications.

station (BS) is controlled by the Radio Resource Control protocol. In RRC, a state machine [16] is established to manage the status of a physical radio link, which consists of three states, i.e., IDLE, FACH and DCH. The RRC state transition graph is shown in Fig. 1.

IDLE. If there is no data to transfer for a sufficiently long time, the interface will switch to the IDLE state. At this state, there is no radio link established between the device and the base station. Therefore, the interface is unable to transfer any data and does not consume any energy either.

DCH. When the interface requires to transmit data, it will switch to the DCH state. At this state, the interface is allocated with a dedicated transport channel. It can make full use of the radio link for high speed transmission but the power consumption at this state is also the highest, e.g., about 800 mW.

FACH. When the interface completes a data transmission at the DCH state, it will switch to the FACH state. At this state, the interface is allocated a shared transport channel and has to compete with other devices for the use of the radio link. The power consumption at this state is about 400 mW.

There are two kinds of transitions among the three states, namely, *promotion* and *demotion*.

Promotion. It refers to that the interface switches from a state to another state with higher power consumption, such as from IDLE to DCH or from FACH to DCH. The promotion transition requests for more radio resources from the base station, which will introduce large delays before the transition is completed. In addition, promotions also bring additional energy consumption.

Demotion. It refers to that the interface switches from a state to another state with lower power consumption. Demotions between states are controlled by *inactivity timers*. Let T_1 denote the timer for the demotion from DCH to FACH and T_2 denote the timer for the demotion from FACH to IDLE or DCH to IDLE. The inactivity timers work as follows. First, when the interface completes a data transmission at DCH, the T_1 timer starts. For best-effort services (e.g., Web Browsing and Instant Messaging), if T_1 expires before any data transfer request, the interface will be demoted to the FACH state. After that, the T_2 timer starts and if it expires, the interface will drop to the IDLE state. Note that during the procedure, any data transmission request will reset both timers. For real-time services (VoIP and Streaming), when T_2 expires, the Radio Network Controller (RNC) demotes the state to IDLE from either DCH. The two timeouts are known *as tail time*.

There is a fundamentally difference between the RRC state machines reported in [2] and ours. In order to improve channel resource utilization for the real-time services, the transitions from the our state machines depend on the service type. The transition from DCH to IDLE never occurs in their state transitions, the RRC connection always switches to FACH before reaching IDLE.

Tail time is designed in 3G networks so that instant packets can be transferred immediately. On one hand, long tail time can provide good user experience of applications since no packets will experience the promotion delay if the interface is not staying at DCH or FACH. On the other hand, long tail time will also result in great waste of energy if there are no packets to send or receive. Moreover, long tail time also deceases the utility of radio resource since the link is held by the interface and cannot be used by other UEs. Therefore, the tail-time selection strategy is the tradeoff between good performance and energy efficiency. An optimal strategy should adaptively meet the data transfer requirements of applications.

4 Empirical 3G Data Analysis

4.1 Collecting 3G Trace

We collect 3G traffic trace data from a southern metropolis in China, which contain more than 65,000 3G users for one month from Nov. 25 to Dec. 24, 2011. A record in the trace contains the following information of a packet: User ID, time stamp, packet size, source and destination IP addresses, transportation-layer port, etc. Furthermore, we use a commercial Deep Packet Inspection (DPI) tool [17] to identify the application protocols. It maintains a database of signatures of hundreds of applications, especially many widely-used local applications, and maps each flow to an IP application based on the pre-defined payload signatures. With our trace, it can identify over 90 percent of the traffic.

In total, we have derived 14 categories of applications from the trace. Fig. 2 illustrates the throughput pie chart for all applications. It can be seen that the amount of web browsing and streaming applications can reach up to 62 percent of the total traffic. These applications are time sensitive. Any delays occurred during the application are noticeable and therefore should be avoided. We choose both categories to study since they contribute the majority of the traffic. In addition, we also notice that, although instant messaging only account for 2 percent of the total traffic, it consumes

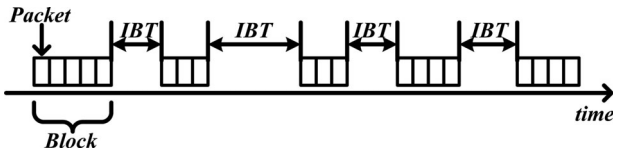


Fig. 3. An example of data blocks and inter-block times.

more than 30 percent of energy due to their periodical behavior of sending beacon messages [18]. We also take instant messaging into consideration.

4.2 Revealing the Impact of Tail Effect

4.2.1 Measuring the Inactivity Timers

We conduct controlled experiments to measure the Carrier's state machine transition parameters. These parameters were statically configured by the Radio Network Controller while different network operators may adopt different values. Inactivity timer T_1 denotes the time for the demotion from DCH to FACH and T_2 denotes the time for the demotion from FACH to IDLE. Our methodology is similar with the method using in [2], we infer T_1 and T_2 to be 5 and 10 sec, respectively.

4.2.2 Analyzing Energy Waste on Tails

In order to reveal the impact of tail effect in real scenarios, we apply the above RRC model we have learnt to our trace data.

We first arrange packets of each application in a row according to the time when the packet was sent or received. We then consider the inter-packet time which refers to the time interval between any two consecutive packets in the row. Particularly, we also notice that packets are coming in burst in our trace. Since the tail between two immediate packets are pretty short and can be omitted, we refer to a block as a series of consecutive data packets which have small inter-packet time (less than 500 milliseconds in our analysis) and take blocks into consideration instead of individual packets. Accordingly, we define the inter-block time (IBT) as the time interval between any two consecutive blocks of packets (as illustrated in Fig. 3).

Given the RRC model, an IBT larger than the sum of two inactivity timers (i.e., T_1 and T_2) will cause the link between the UE and the BS to be released. In this case, when the next packet block arrives, the UE has to go through a promotion transition, introducing a promotion delay before the real data transfer happens. Besides the promotion delay, waiting for T_1 and T_2 to expire also waste much energy in vain. In addition, as mentioned in Section 3, promotion transition also generates extra power consumption. We consider both the power consumption caused by tails and that of promotions as extra energy cost, in contrast with effective energy cost which is used for the actual data transmission.

With the trace, we first extract IBTs by inferring whether a block of packets has been delayed due to link state promotions (i.e., from IDLE to DCH and from FACH to DCH). Specifically, if an observed IBT is no less than the sum of T_1, T_2 and the promotion delay from IDLE to DCH, denoted as $T_{I\rightarrow D}$, the upcoming block must have been experienced a link state promotion from IDLE to DCH. Thus, the $T_{I\rightarrow D}$ is subtracted from the IBT. Similarly, if an IBT is no less than the sum of T_1 and the promotion delay from FACH to DCH, denoted as $T_{F\rightarrow D}$, but less than the sum of T_1 and T_2, a link promotion from FACH to DCH happened and the corresponding $T_{F\rightarrow D}$ is removed from the IBT. We show the cumulative distribution function (CDF) of IBTs of IM, web browsing, streaming and mixed traffic of 1,000 randomly-chosen users in our trace on a workday and a weekend in Figs. 4 and 5, respectively. It can be seen that, in general, IBTs have smaller values in workdays than weekends and different types of applications have similar IBTs in workdays but show more diversity in weekends. It is clear that IM applications have the biggest IBT values than others and then comes streaming in the second place.

Given the tail distributions, we can estimate the energy consumption wasted on unnecessary tails by multiplying the duration of a tail and its power level. We plot the CDF of the ratio of extra energy cost to the total energy consumption over 1,000 randomly-selected users in Fig. 6. It is clear to see that, in general, most of the energy is extra energy cost. For example, over 80 percent users have more than 70 percent energy spent on tails. It can also be seen that mixed traffic has lighter tail effect than separated ones. The reason is that

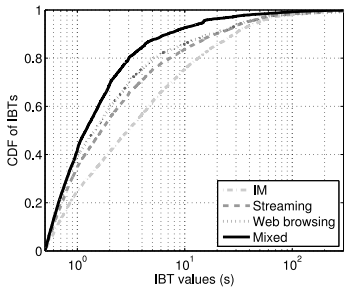


Fig. 5. CDF of IBTs during weekend.

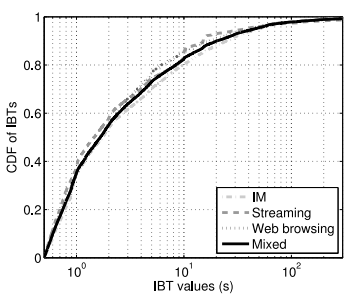


Fig. 4. CDF of IBTs during workday.

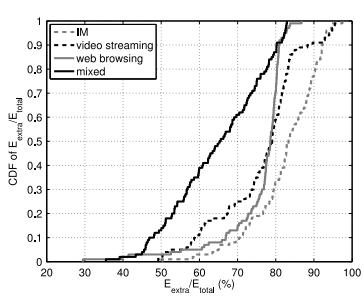
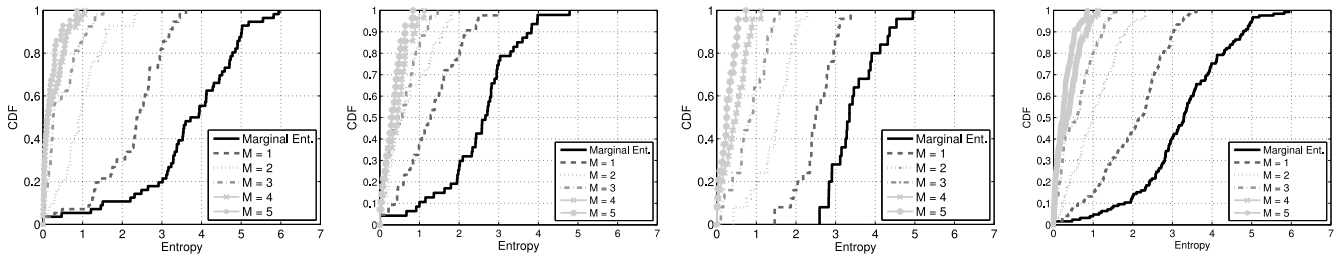


Fig. 6. Ratios of extra energy against total energy.



(a) IM (b) Streaming (c) Web browsing (d) Mixed traffic

Fig. 7. CDFs of marginal entropy and conditional entropy of inter-block times in 3G data set.

when several applications access the network simultaneously, the packet arrivals become more intensive, which implies shorter IBTs and duration of tail times. Therefore, the extra energy under mixed traffic is smaller than others.

4.3 Characterizing Temporal Correlation of 3G data

From the above analysis, the characteristics of the traffic load, especially the layout of packets, plays an important role in determining the amount of power consumed over 3G networks. To understand whether there exist patterns in 3G traffic load, we examine the temporal correlation between IBTs by calculating the marginal and conditional entropy in this section.

Let $\{t_i, i = 1..n-1\}$ represent the IBT series where t_i denotes the ith IBT and t_i has m different observed values $\{v_j, 1 \leqslant j \leqslant m\}$ throughout the series. Let x_j be the times of v_j appearing in the series. Then, the probability of $P_{v_j} = x_j/(n-1)$. Therefore, the marginal entropy of the series $\{t_i\}$ can be written as:

$$H(t_i) = -\sum_{1 \leqslant j \leqslant m} P_{v_j} * \ log_2 P_{v_j}. \quad (1)$$

Now, we calculate the conditional entropy of IBTs given their previous M values.

When $M = 1$ Let $\{(t_{i-1}, t_i), i = 2..n-1\}$ be the two-dimensional random variable for inter-block time t_i and its immediate predecessor. Suppose that the value space of (t_{i-1}, t_i) is Q. Therefore, the joint entropy of (t_{i-1}, t_i) is:

$$H(t_{i-1}, t_i) = -\sum_{(u,v) \in Q} P_{(u,v)} * \ log_2 P_{(u,v)}, \quad (2)$$

where $P_{(u,v)}$ is the probability of inter-block time t_i being v and its immediate predecessor being u. With $H(t_{i-1}, t_i)$ and $H(t_i)$, the conditional entropy of series $\{t_i\}$ given $\{t_{i-1}\}$ is:

$$H(t_i \,|\, t_{i-1}) = H(t_{i-1}, t_i) - H(t_{i-1}) = H(t_{i-1}, t_i) - H(t_i). \quad (3)$$

Similarly, when $M = 2$, the conditional entropy of series $\{t_i\}$ given $\{t_{i-1}\}$ and $\{t_{i-2}\}$ can be written as:

$$H(t_i \,|\, t_{i-1} t_{i-2}) = H(t_{i-2}, t_{i-1}, t_i) - H(t_{i-2}, t_{i-1}). \quad (4)$$

Fig. 7 shows the CDFs of the mean marginal entropy and the mean conditional entropy, for $M = 1$ to 5, over 1,000 users in 3G trace. From the CDFs, it can be seen that, in all the application scenarios, the more previous IBTs being given, the smaller entropy we will get. It implies that the uncertainty about IBTs decreases when knowing historical IBTs.

5 SmartCut Design

5.1 Overview

With the observation of temporal correlation of IBTs, it is possible for a mobile device to infer its future traffic, which can be utilized to establish a more energy-efficient RRC state transition strategy. To this end, we design the SmartCut scheme which runs on mobile devices like smartphones. There are three key techniques integrated in the scheme: *estimating future IBTs*, *cutting unnecessary tails*, and *remedying prediction errors*. Specifically, SmartCut adopts the ARMA model to capture the temporal correlation of IBTs extracted from recent mobile traffic on a mobile device, based on which it estimates a number of next IBTs (i.e., the arrival time of the next data blocks). With such information, SmartCut actively cuts the tails if the expected IBTs are larger than the promotion delays and promotes the radio interface in advance before the real data transmission begins. It is very likely, however, that SmartCut makes an inaccurate IBT estimation, which can lead to false promotions or getting data transmission delayed. In order to cope with problem, SmartCut also trains another ARMA model on the variations of previous estimations. In addition to estimating IBTs, SmartCut also estimates the corresponding variations. Upon IBT estimation errors, SmartCut leverages the variation estimation to reduce the influence. In this section, we elaborate the design details of SmartCut.

5.2 Estimating Future IBTs

Treating the series of IBTs as a signal in the temporal dimension, we can apply time series analysis models to capture temporal correlation of IBTs and further predict future IBTs. For example, Fig. 8 shows the IBT prediction error ratio as a function of different application traffic of 1,000 random chosen users using three well-known prediction algorithms, i.e., ARMA [19], EWMA [20] and Holt-Winters [21]. The IBT prediction error ratio is defined as the ratio of IBT prediction error to the actual IBT value. Fig. 8 shows that Holt-Winters constantly performs best and ARMA has similar prediction error as EWMA.

For demonstration, in this paper, we choose to use ARMA model for three reasons. First, in general, ARMA is rich enough to capture a large variety of linear temporal dependencies. Second, as the model runs on personal UEs,

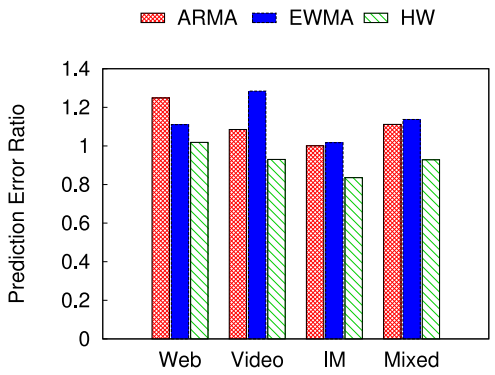


Fig. 8. Predicting IBTs using different models.

it has to have low computational cost in order to reduce the extra energy consumption. Last, ARMA has the capability to predict for a longer future. The ARMA model consists of two parts: an autoregressive part (AR) and a moving average part (MA). Typically, the model is referred as a two-tuple ARMA (p,q), where p is the order of the autoregressive part and q is the order of the moving average part. It can be defined as follows:

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q}, \quad (5)$$

where $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$ are the parameters of the model, ϕ_0 is a constant component and $\varepsilon_t, \ldots, \varepsilon_{t-q}$ are white noise error terms.

The orders p and q can be determined by conducting Akaike information criterion (AIC) test [22], which is a measure of the relative goodness of fit of a statistical model. Using least-square estimation, the AIC can be written as:

$$AIC = nlog\sigma^2 + (p+q+1)logn, \quad (6)$$

where n is the number of samples, σ^2 is the variance of the residuals after fitting the model, p and q are undetermined orders of the model. The AIC test will choose the values of p and q which maximize the Equation (6).

In order to train an accurate ARMA model, it is also of great importance to know how much historical data are sufficient to be involved. We use cross validation method to determine the optimal amount of training data. More specifically, we divide the 3G trace into two parts: one for training and the other for testing. Let $MSE(n)$ be the mean squared error of the prediction using past n IBTs for model training. We gradually increase the number of IBTs used for training until we find the optimal number of past IBTs which minimizes the MSE. We will intensively examine the effect of the amount of history data to the accuracy of IBT prediction in the evaluation section.

Besides the determination of training data, since different applications have different data traffic pattern, SmartCut maintains a unique model for each one for the purpose of prediction accuracy.

5.3 Cutting Unnecessary Tails

With the established ARMA models, SmartCut is enabled to predict future data blocks. Based on the estimated IBT values, SmartCut will immediately switch the 3G interface to the IDLE state after a data transmission completes if the expected IBT is larger than the promotion delay. In order for the interface to function properly in future data transmissions, it is also necessary to promote the interface back in advance. For this reason, it promotes the interface back to the FACH state before the estimated arrival time of the next data block. In this way, with an accurate prediction, the promotion can be completed just before the actual data transmission so that users cannot feel any promotion delays and the energy wasted by unnecessary tails are saved.

More specifically, let $\{t_i, i = 1..n-1\}$ denote the series of inter-block times, $\{t'_i, i = 1..n-1\}$ be the predicted values of $\{t_i\}$ and t_{delay} denote the delay for link promotions. (For simplicity, we here do not differentiate the promotions whether it is from IDLE to DCH or from FACH to DCH). Then the *cut-and-promote* scheme is described as follows:

- After a new booting of a smartphone, the 3G interface remains at the IDLE state.
- The very first data block will cause the radio to be promoted, which also brings the user a promotion delay t_{delay}.
- Once a data transmission completes, the interface forecasts the arrival moment of the next data block. Let t'_i be the predicted value of the inter-block time between the current block and the next one. If $t'_i \leq t_{delay}$, the tail gets retained. Otherwise, an immediate demotion from DCH to IDLE will be applied to the interface.
- When at the IDLE state, the 3G interface will be promoted to the DCH state after a time of $t'_i - t_{delay}$, preparing for the upcoming data.

5.4 Remedying Prediction Errors

The performance of the above cut-and-promote scheme significantly relies on the accuracy of the IBT predictions. Ideally, if no prediction errors happen, our scheme can achieve the optimal solution which wastes no power on tails and meanwhile gets no data transmission delayed since the interface has been promoted just before the transmission happens. In practice, due to complicated application behaviors, traffic workload may not be perfectly estimated. For example, the IBT prediction errors can reach the same order of magnitude as that of real IBTs as shown in Fig. 8. SmartCut has to deal with the situation when IBT prediction errors occur.

1) Dealing with Small IBT Estimation Errors

Specifically, there are two types of IBT prediction errors:

Small IBT estimation. Refers to the expected arrival time t'_i is earlier than the actual arrival time t_i, which will lead to a waste of energy since the interface has already stay in high-power-level state waiting for data transmission.

Large IBT estimation. Refers to the expected arrival time is later than the actual arrival time, which will result in a noticeable promotion delay since the 3G interface still remains at the IDLE state and needs time to be promoted.

Let $\{e_i = t_i - t'_i, i = 1..n-1\}$ denote the estimation variations, where t_i is the ith observed IBT and t'_i is the estimation of t_i, and e'_i denote the estimation of e_i. Note that the estimation variations also form a series, which can be

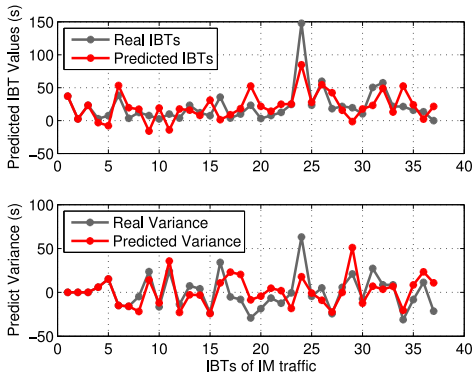


Fig. 9. An example of IBT and its variation prediction.

predicted using ARMA models as well. For example, the upper sub-graph in Fig. 9 shows the results of using ARMA model to predict future IBTs of the IM traffic of an arbitrary user. The gray dash line represents the real IBTs observed in the trace and the red solid line is the predicted IBTs. The lower sub-graph in Fig. 9 shows the corresponding results of estimation variations, where the grey dash line represents the variation series and the red solid line represents its estimated value using another independent ARMA model. It can be seen that SmartCut can achieves rather good performance in predicting both IBT values and IBT estimation variations. As a result, we can remedy the IBT prediction errors by utilizing the information of both the predicted IBTs and the corresponding predicted estimation variations presented as follows.

With the cut-and-promote scheme, the interface of a smartphone previously been cut will be promoted back t_{delay} time before the ith expected block arrival time t'_i. When the interface is up and finds no data transmission is required, i.e., $t'_i < t_i$, a small IBT estimation error occurs. In this situation, if do nothing, the 3G interface will remain at a high-power-level state waiting for the arrival of the next block. The duration of the waiting time is thus equal to e_i, which is a positive value. Therefore, giving the predicted values e'_i, we are able to estimate whether or not it is worth waiting. In this case, SmartCut will perform a *positive variation correction* (PVC) operation.

More specifically, we set a threshold t_{th}. If $e'_i \leqslant t_{th}$, the 3G interface will keep waiting. Otherwise, it will be demoted to the IDLE switch again. One of the reasonable values of t_{th} is the mean value of e_i (for all $e_i > 0$). Note that in this situation, e_i is always positive but e'_i could be a negative value. In this case, we treat e'_i as zero. In this paper, we present our correction method adopted in SmartCut. In specific, the 3G interface first calculates the predicted value e'_i. If $e'_i \leq t_{th}$, the interface will remain at the DCH state and keep waiting. Otherwise, an immediate demotion will be carried out.

2) Preventing Large IBT Estimation Errors

If a large IBT estimation error happened, i.e., $t'_i > t_i$, the 3G interface would fail to be promoted back from the IDLE state to the DCH state before the actual data transfer request comes. This will force the upper layer application to wait until the interface is ready to use. In this case, an inevitable large delay is introduced, which severely degrades the user experience and should be avoided. To accurately determine when a large IBT estimation error would occur, however, is very challenging. The reason is that only when the error actually happens the interface can realize that a large IBT estimation error has occurred.

Fig. 10. Prototype and power meter.

In SmartCut, we adopt the *to-avoid-is-better-than-to-save* strategy. The key idea is to utilize the variation prediction to assess the possibility of a large IBT estimation error. Specifically, as mentioned above, when the 3G interface completes a data transmission of the ith data block, SmartCut will estimate the arrival time of the next data block t_i. In addition, SmartCut will also make an estimation on the variation e_i. If the estimated variation value e_i is negative, which indicates the possibility that a large IBT estimation is happening is large. At this point, SmartCut will conduct a *negative variation correction* (NVC) operation. Precisely, the estimated time of the next data block's arrival will be corrected to $t'_i - |e'_i|$. After that, the interface follows the normal procedure of SmartCut.

With the PVC and NVC operations, the consequence of IBT estimation errors can be greatly mitigated. The performance of all these error correction methods will be thoroughly examined in Section 7.

6 Prototype Implementation

In order to verify its feasibility, we implement the prototype of SmartCut on HTC G6 smartphones [23]. The major reason that we choose this smartphone is because it supports the fast dormancy mechanism. With the fast dormancy mechanism, the smartphone first sends a SCRI RRC message to the BS via the control channel. Upon receiving this message, the BS releases the channel assigned to the smartphone and allows the smartphone to demote to the IDLE state. In order for SmartCut to operate properly, in addition to the fast dormancy mechanism, it is also important to effectively switch the 3G interface back to the transmission-ready state. To this end, we let the smartphone send a tiny UDP packet (typically, 40 Bytes) whenever an advance promotion is required, which ensures that the 3G interface is ready at the DCH state. We have also implemented a power meter [24] to measure real-time power consumption on a smartphone (as shown in Fig. 10).

In order to verify the feasibility of SmartCut, we conduct a set of small-scale experiments, taking into consideration the video streaming, web browsing and instant messaging applications. We first initialize SmartCut to run as a kernel module and, for each application, we let 10 volunteers to use the application for 2 hours. During the

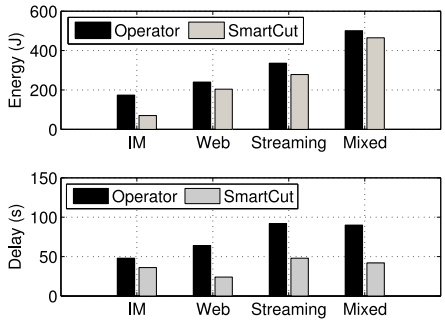


Fig. 11. Total energy consumption and promotion delays when running different applications on prototype system.

process, SmartCut maintains a series of IBTs collected in the first hour for training two ARMA models. From the beginning of the second hour, SmartCut starts to predict IBTs and operate the 3G interface accordingly. Meanwhile, we use the power meter to record instant power consumption. Fig. 11 shows the energy consumption when running different applications with the default RRC model adopted by the 3G network operator and with SmartCut. It can be seen that SmartCut can largely reduce both the energy cost and the delay due to link promotions over all applications. In particular, for Tudou [25], a popular video streaming provider in China, SmartCut can save up to 17.6 percent of the total energy comparing to the current fix tail timers scheme adopted by the network operator. It can also be seen that, surprisingly, IM applications (QQ [26] and MSN [27] installed on our prototype) have the most improvement of energy efficiency among all application scenarios, which is 58.8 percent. The reason is that, although the human interactions can introduce unpredictable factors into the traffic workload, those applications always periodically send keep-alive packets to their servers, which shows stronger traffic patterns for better IBT estimation. Note that as we do measure the total energy consumption of the smartphone, the power consumption caused by training ARMA models and making IBT predictions is included. As the computational cost of ARMA models is rather low, the corresponding power consumption is negligible. At the same time, SmartCut can also help reduce the amount of time delayed by inefficient radio usage. For example, SmartCut can reduce more than 67 percent of delays while browsing web pages.

From the prototype implementation and experiments, we learn that SmartCut is very effective in terms of reducing power consumption caused by long tails. It is also a lightweight solution and transparent to upper layer applications. Besides that, we also notice that SmartCut performs better when the traffic workload of an application has regular pattern. We further investigate the performance of SmartCut via extensive trace-driven simulations in the following section.

7 Evaluation

7.1 Methodology

To further investigate the efficacy of SmartCut, we conduct extensive simulations based on our trace. We randomly choose a one-month 3G data set including 1,000 users, which contains of data traffic generated by all 3G applications. In realizing the significance of video streaming, web browsing and instant messaging in the traffic distribution, we focus on their traffic to study. We compare SmartCut with several alternative schemes:

1) *Always-on*. In this scheme, the 3G interface remains awake for ever no matter whether there is data to transfer or not. That is to say, after a transmission at the DCH state, the interface simply switches to the FACH state awaiting the next packet.
2) *Always-off*. As an opposite of the Always-on scheme, it always demotes the interface to the IDLE state once a transmission completes. Therefore, this scheme costs no energy on tails. One simple way to realize the Always-off scheme is to use the fast dormancy mechanism.
3) *Fixed-tail*. This scheme is adopted by network operators by default, in which, after each transmission, the 3G interface retains a fixed tail (5 seconds for T_1 and 10 seconds for T_2). As shown in Fig. 6, there is much energy wasted on unnecessary tails.

We evaluate all above radio usage schemes using the following metrics:

1) *Waste energy ratio*. It is defined as the ratio of the extra energy to the total energy. This metric implies how much energy is caused in order for the interface to be ready for data transmission, which includes the energy consumption on both the tails and promotions. The larger the ratio is, the less energy-efficient the scheme is.

2) *Un-delayed packet ratio*. It is defined as the ratio of the number of un-delayed packets to the total number of packets. For time-sensitive applications, packets delayed due to the inappropriate link state of the 3G interface can significantly degrade user experience. We use this metric is to quantify user experience with certain radio usage scheme. We consider those delays caused by link promotions, which are rather long (e.g., one and a half second from FACH to DCH and 2 seconds from IDLE to DCH) and cause unpleasant interruptions for data transmission. Therefore, we consider the number of such promotion delays instead of absolute amount of time and normalize the value with the total number of packets.

3) *Energy utility*. It is defined as the number of un-delayed packets divides the total energy consumed. To thoroughly assess whether a scheme is good, only considering the the amount of energy saved is not sufficient. For example, one scheme adopting the fast dormancy mechanism can always cut off the tails once data transmission completes. Doing so definitely has the maximum gain with respect to power saving but it also causes all upcoming data transfer being delayed. For time-sensitive applications, the number of un-delayed data transfers also counts. With this metric, we can perfectly depict the tradeoff between energy consumption and user experience.

In the following sections, we investigate SmartCut and other alternative schemes over all users in the trace. For each experiment, we use the cross validation result presented in Section 5.2 and divide the trace into two parts, i.e., *training* and *testing*. In particular, we further divide the data set for training into two parts. One is used to train the

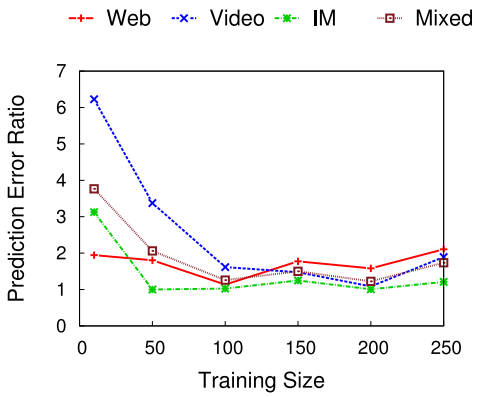


Fig. 12. IBT prediction error ratio versus the number of history IBTs.

ARMA model for IBT prediction and the other one is used to train the ARMA model for variations.

7.2 Effect of History Data and Stability of ARMA Models

In order to investigate how much history traffic are sufficient for training the ARMA model and how stable the trained models can be used over time. We gradually increase the number of past IBTs from one to 250 with an increment of 50 to find the optimal training data size over all traffic of all users.

Fig. 12 shows the prediction error ratio, defined as the ratio of the error of the predicted IBT to the corresponding real IBT, as a function of the number of history IBTs. As it can be seen, when the training size increases to 100, the prediction error decreases by 20, 76, and 75 percent for web browsing, video streaming, and IM traces, respectively. However, as the training size continues to increase, the prediction error ratio may increase due to the overfitting problem. In our following simulations, we take 200 IBTs to train the model.

To study how steady the trained ARMA models are, we conduct an experiment to examine the prediction error ratio of future IBTs. Fig. 13 shows the prediction error ratio as a function of the number of future IBTs to be estimated. It can be seen that the prediction error increases slowly at the prediction depth of the next 60 IBTs but dramatically after that. It suggests that SmartCut needs to update the model constantly to adapt the traffic changes of users or the network condition.

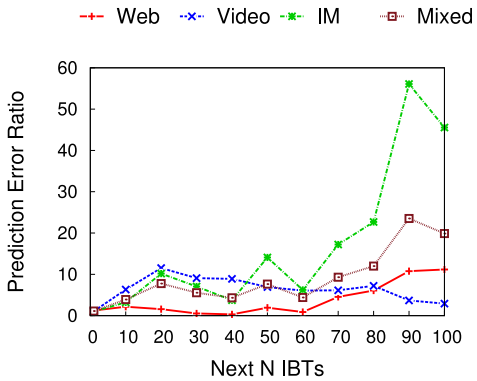


Fig. 13. IBT prediction error ratio versus the number of future IBTs to be estimated.

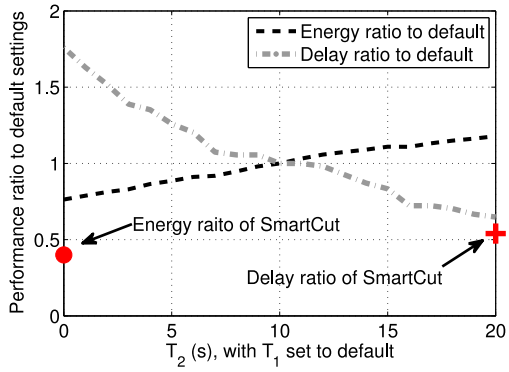


Fig. 14. The ratio of energy consumption and the number of un-delayed packets to the default results with T_1 varying.

7.3 Comparing with Fixed-Tail Scheme

For better understanding the current scheme adopted by the network operator, we conduct an experiment to change both T_1 and T_2 of the fixed-tail scheme and compare the results with SmartCut. We first set T_1 to the default value, i.e., 5 seconds, and vary T_2 from 0 to 20 seconds.

Fig. 14 plots the ratio of total energy consumption and the number of un-delayed packets with T_2 varying to that with both T_1 and T_2 set to the default values used by the operator. It is clear to see that as T_2 increases the ratio of energy consumption also increases but the ratio of un-delayed packets drops. This is because as T_2 increases more energy will be wasted on tails waiting for data transmission but more packets can be transferred right away as the radio is up. We also compare SmartCut with the operator default settings and mark the results on the figure. It can be seen that SmartCut outwits fixed-tail schemes in terms of both energy consumption and un-delayed packets. Similar observations can be found in Fig. 15 where we set T_2 to the operator default and vary T_1.

7.4 Comparing with Alternative Schemes

We compare SmartCut against three alternative schemes: Always-on, Always-off and Fixed-tail.

Fig. 16 plots the waste energy ratio for all the schemes averaged over 1,000 users. It can be seen that in all application scenarios, Always-on obtains a highest extra energy ratio and in contrast, Always-off has the lowest. It can also be seen that for Always-off, even with all the tails being cut

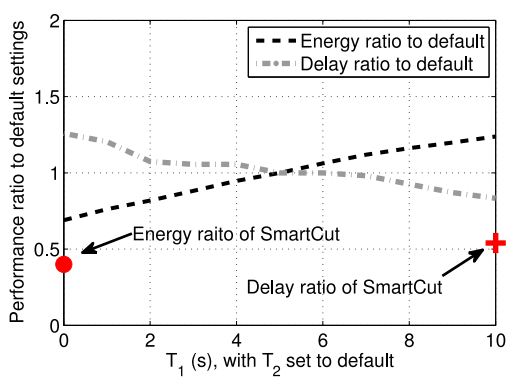


Fig. 15. The ratio of energy consumption and the number of un-delayed packets to the default results with T_2 varying.

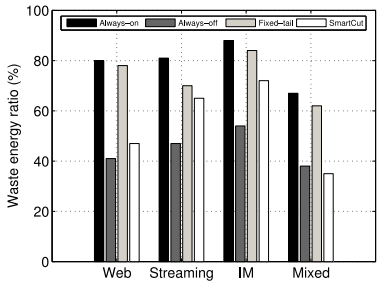


Fig. 16. Comparison of all schemes using waste energy ratio.

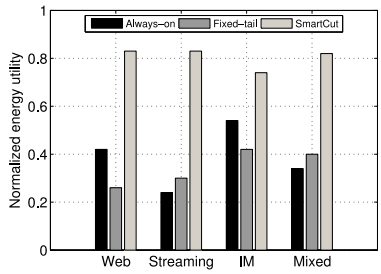


Fig. 18. Comparison of all schemes using metric energy utility.

off, the extra energy remains high (up to 40 percent against the total). It is because that lots of promotions are brought in, which also leads to considerable energy consumption. Note that SmartCut always achieves a more efficient use of energy than Fixed-tail. In general, it shows that SmartCut can save up to 43.08 percent radio energy for applications compared to the Fixed-tail scheme.

Fig. 17 plots the un-delayed packet ratio for all the schemes averaged over 1,000 users. We skip the results of Always-on and Always-off in the figure. The reason is that the former always has a 100 percent un-delayed packet ratio with no packets being delayed and the later always has a zero un-delayed packet ratio with all packets being delayed. It can be seen that, SmartCut has more un-delayed packets than Fixed-tail in streaming but less in the other three types of traffic. The reason may be that the pattern in the traffic workload of video streaming is more distinct than others, which leads to more accurate predictions. Fig. 18 plots the energy utility for all the schemes averaged over 1,000 users. Just as in Fig. 17, we skip the result of Always-off for the reason it always being zero. We can see that among all the schemes, SmartCut achieves the highest value of energy utility. Especially in video streaming, the energy utility of SmartCut is about three times as much as that of Fixed-tail. It is implied that SmartCut makes a much more efficient use of energy to improve the user experience.

The simulation results show that in most cases, SmartCut can achieve a significant energy efficiency with acceptable impact on user experience.

8 Conclusion and Future Work

In this paper, by analyzing large 3G trace, we have confirmed that the tail effect wastes a significant amount of power. We have found strong temporal correlations existing in 3G traffic workload. Based on those key observations, we have proposed a lightweight scheme, SmartCut, which uses historical 3G traffic data to train ARMA models and further utilizes the predicted arrival time of future data transmission to effectively cut unnecessary tails while having little side-effect to user experience. Both prototype experiment and extensive trace-driven simulation results have shown that SmartCut are energy-efficient. As the power of transmission at the DCH state or FACH state may vary with signal strength, a UE needs to amplify the received signal in order to decode correctly if the UE is near the edge of the signal coverage, this will result in additional energy consumption. In the future, we are planning to study the impact of signal strength to SmartCut.

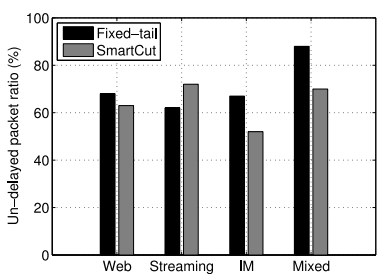


Fig. 17. Comparison of all schemes using un-delayed packet ratio.

Acknowledgments

This research was supported in part by the State High-Tech Development Plan (2013AA01A601), NSFC (No.61170237, 61202375) and Singapore NRF (CREATE E2S2). The authors would like to thank Fei Yu for his contribution in traces collection and system design.

References

[1] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf.*, 2009, pp. 280–293.

[2] F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "Characterizing radio resource allocation for 3G networks," in *Proc. 10th Annu. Conf. Internet Meas.*, 2010, pp. 137–150.

[3] F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "Top: Tail optimization protocol for cellular radio resource allocation," in *Proc. 18th IEEE Int. Conf. Netw. Protocol*, 2010, pp. 285–294.

[4] H. Liu, Y. Zhang, and Y. Zhou, "Tailtheft: Leveraging the wasted time for saving energy in cellular communications," in *Proc. 6th Int. Workshop MobiArch*, 2011, pp. 31–36.

[5] *3gpp ts25.331 radio resource control*. (Jul. 2014). [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/25331.htm

[6] G. Association, "Fast dormancy best practises," Jul. 2011.

[7] J. Huang, F. Qian, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Serv.*, 2012, pp. 225–238.

[8] J. Yeh, J. Chen, and C. Lee, "Comparative analysis of energy-saving techniques in 3GPP and 3GPP2 systems," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 432–448, Jan. 2009.

[9] A. Pathak, Y. Hu, M. Zhang, P. Bahl, and Y. Wang, "Fine-grained power modeling for smartphones using system call tracing," in *Proc. 6th Conf. Comput. Syst.*, 2011, pp. 153–168.

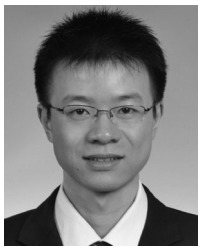
[10] F. Liers and A. Mitschele-Thiel, "UMTS data capacity improvements employing dynamic RRC timeouts," in *Proc. IEEE 16th Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2005, vol. 4, pp. 2186–2190.

[11] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Proc. 10th Annu. Conf. Internet Meas.*, 2010, pp. 281–287.

[12] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. Padmanabhan, "Bartendr: A practical approach to energy-aware cellular data scheduling," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 85–96.
[13] (2010) *HSPA evolution*. [Online]. Available: www.ericsson.com/res/docs/whitepapers/WP-HSPA-Evolution.pdf
[14] (2010) HUAWEI, *Behavior analysis of smartphones*. [Online]. Available: http://www.huawei.com/ilink/en/download/HW_001545
[15] P. Athivarapu, R. Bhagwan, S. Guha, V. Navda, R. Ramjee, D. Arora, V. Padmanabhan, and G. Varghese, "Radiojockey: Mining program execution to optimize cellular radio usage," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 101–112.
[16] L. Qian, E. W. Chan, P. P. Lee, and C. He, "Characterization of 3G control-plane signaling overhead from a data-plane perspective," in *Proc. 15th ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst.*, 2012, pp. 325–332.
[17] Y. Diao, T. V. Lakshman, Y. Fang, Z. Chen, and R. H. Katz, "Fast and memory-efficient regular expression matching for deep packet inspection," in *Proc. ACM/IEEE Symp. Arch. Netw. Commun. Syst.*, 2006, pp. 93–102.
[18] F. Qian, Z. Wang, Y. Gao, J. Huang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "Periodic transfers in mobile applications: Network-wide origin, impact, and optimization," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 51–60.
[19] *Estimating AR and ARMA models*. (Jul. 2014). [Online]. Available: http://www.mathworks.com/help/ident/ug/estimating-ar-and-arma-models.html
[20] A. Chen and R.-S. Guo, "Age-based double EWMA controller and its application to CMP processes," *IEEE Trans. Semicond. Manuf.*, vol. 14, no. 1, pp. 11–19, Feb. 2001.
[21] J. Yeh, J. Chen, and C. Lee, "Holt-winters forecasting: some practical issues," *The Statistician*, vol. 37, pp. 129–140, 1998.
[22] *Akaike information criterion*. (May 2014). [Online]. Available: http://en. wikipedia.org/wiki/Akaike_information_criterion
[23] *Htc g6*. (Apr. 2014). [Online]. Available: http://en.wikipedia.org/wiki/HTC_Legend
[24] BattOr: Portable Power Monitor for Mobile Phones. (Jul. 2014). [Online]. Available: http://web.stanford.edu/~aschulm/battor.html
[25] (2013). [Online]. Available: http://www.tudou.com/
[26] (2013). [Online]. Available: http://www.imqq.com
[27] (2013). [Online]. Available: http://www.msn.com/

Guangtao Xue received the PhD degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2004. He is an associate professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University, China. His research interests include vehicular ad hoc networks, wireless networks, mobile computing, and distributed computing. He is a member of the IEEE, IEEE Computer Society, and the IEEE Communication Society.

Hongzi Zhu received the PhD degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2009. He is an associate professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University, China. His research interests include vehicular ad hoc networks, wireless networks, mobile computing, and network security. He is a member of the IEEE, IEEE Computer and the IEEE Communication Society.



Zhenxian Hu received the MS degree in communication systems from the China Academy of Telecommunication Technology in 2011. He is currently working toward the PhD degree in the Department of Computer Science and Engineering at Shanghai Jiao Tong University, China. His research interests include computer network protocols and systems, wireless network, mobile computing, and network measurement. He is a student member of the IEEE.

Jiadi Yu received the PhD degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2007. He is an assistant professor in Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. In the past, he has worked as a postdoc at Stevens Institute of Technology, from 2009 to 2011. His research interests include networking, mobile computing, cloud computing, and wireless sensor networks. He is a member of the IEEE.

Yanmin Zhu received the PhD degree from the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology in 2007. He is an associate professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. Prior to joining Shanghai Jiao Tong University, he worked as a research associate with the Department of Computing at the Imperial College London, U.K. His research interests include vehicular networks, wireless sensor networks and mobile computing. He is a member of the IEEE and the IEEE Communications Society.

Gong Zhang is a principal member in Advance Network Technology Research Department at Huawei. His research spans networks, distributed system, and communication system architectures. He has contributed to more than 90 patents. He had been a product development team leader of smart devices in 2002 to pioneer new consumer business for Huawei. He then started to lead the research on future Internet and cooperative communication in 2005. He has been in charge of the Advance Network Technology Research Department, leading research on future network, distributed computing, database system, and data analysis since 2009. His recent research focuses on future network. He is a member of the IEEE.