Surviving Screen-off Battery through Out-of-band Wi-Fi Coordination

Xianjin Xia*, Shining Li*, Yu Zhang*†, Lin Li*, Tao Gu†, Yongji Liu*, Yan Pan*
*School of Computer Science, Northwestern Polytechnical University, China
†School of Computer Science and IT, RMIT University, Australia
jinchenxia@mail.nwpu.edu.cn; {lishining, zhangyu}@nwpu.edu.cn; tao.gu@rmit.edu.au

***Abstract*—This paper identifies two energy saving opportunities of Wi-Fi interface emerged during smartphone's screen-off periods. Exploiting the opportunities, we propose a new power saving strategy, BackPSM, for screen-off Wi-Fi communications. BackPSM regulates client to send and receive packets in batches and coordinates multiple clients to communicate at different slots (i.e., beacon interval). The core problem in BackPSM is how to coordinate client without incurring extra traffic overheads. To handle the problem, we propose a novel paradigm, Out-of-Band Communication (OBC), for client-to-client direct communications. OBC exploits the TIM (Traffic Indication Map) field of Wi-Fi Beacon to create a free side-channel between clients. It is based upon the observation that a client may control $1 \rightarrow 0$ appearing on TIM bit by locally regulating packet receiving operations. We adopt this $1 \rightarrow 0$ as the basic signal, and leverage the time length in between two signals to encode information. We demonstrate that OBC can be used to convey coordination information with close to 100% accuracy. We have implemented and evaluated BackPSM on a testbed. The results show that BackPSM reduces screen-off energy by up to 60%, and outperforms state-of-the-art strategies by 16%–42%.**

I. Introduction

Smartphones stay in stand-by mode (e.g., when the screen is off) for a large fraction of time during the day [3]. Although users are not actively interacting with phones, many apps and services still run in the background [4] [16] [14]. They stay connected to the Internet, updating app status, syncing with cloud servers or waiting for various incoming events such as emails, instant messages, notifications of social networking apps, etc. Such screen-off traffic are important for smartphones to provide good user experience. The energy consumption due to screen-off traffic, however, has become a matter of concern. Recent works [7][4] reveal that screen-off traffic accounts for the total system energy by 29%–58%. Therefore, to preserve battery power, mobile platforms like Android and iOS often prohibit screen-off background traffic under 3G or LTE, which are energy-costly, and only enable it when Wi-Fi is available (e.g., at homes or offices) [1][3][4].

The current Power Save Mode (PSM) of Wi-Fi, however, performs poorly when applying to screen-off traffic. We identify two energy sources in PSM that can be specially optimized for screen-off traffic. *First*, when a PSM client wants to send an upstream packet, it has to wake up the radio for transmission and switch back to sleep after sending. Although incurring expensive radio switch costs, it will be necessary for ordinary user traffic since it adds no delays to packet sending (i.e., good user experience). Whereas for the screen-off traffic which are mainly short, frequent packets [7], this would severely impair the energy saving benefits of PSM. Since screen-off traffic can often tolerate long delays due to absence of user interactions [7], a better way is to amortize such radio switch energy by sending multiple upstream packets in batches. *Second*, since PSM clients wakes up at the same time (every beacon interval) to receive downstream packets, a client may spend long time awake, contending for channel access, if multiple clients reside in the Wi-Fi cell, which can cause up to 4 times more energy consumption [11]. NAPman [15] and SleepWell [11] optimize this contention energy by isolating client traffic into different micro time slices. However, to avoid large traffic delays, they only divide time slice within the time of one beacon interval, which produces limited number of time slices and thus leads to scalability issues for the strategies. Different from ordinary user traffic, the delay-tolerance nature of screen-off traffic enables us to break the limit of one beacon interval on traffic isolation. One can adopt coarse-grained time slice, e.g., use the whole beacon interval as a slice, to achieve better scalability.

In this paper, we address the two emerged opportunities by designing a new power saving strategy, BackPSM, for screen-off Wi-Fi communications. BackPSM regulates client to send and receive packets in batches, and coordinates multiple clients to communicate at different beacon intervals (termed *slots*). As we need to control both upstream sending and downstream receiving, BackPSM must locate at client side, rather than AP-side, which makes coordination among clients the key challenge for our strategy. The conventional idea to enable client coordination is to employ a central controller, such as a Wi-Fi AP [5] or the WLAN back-haul [17], to compute coordination information and disseminate them to all clients. However, this would incur tremendous traffic overheads, defeating the goal of energy saving. Different from existing approaches, in this work, we ask: *is it possible to directly exchange coordination information between clients yet with no extra traffic?*

We answer the question by presenting a novel paradigm for client-to-client direct communications that do not occupy the regular Wi-Fi communication band (named *Out-of-Band Communication, OBC*). OBC exploits the TIM (Traffic Indication Map) field of Beacons, broadcasted periodically by a Wi-Fi

[0]This work is supported by the National Key Technologies R&D Program of China under grant No. 2014BAH14F01, and the National Key Technology Support Program of China under grant No. N2014KE0043.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.
978-1-5090-5336-0/17/$31.00 ©2017 IEEE

AP, to create a free side-channel between clients, termed *TIM channel*. TIM bits are originally designed to notify presence of buffered downstream packet for PSM clients. We observe that a TIM bit will change from '1' to '0' when the corresponding client receives all buffered packets from the AP. This implies that a client may control the appearing of $1 \rightarrow 0$ on TIM bit by locally regulating packet receiving operations. Exploiting the observation, we adopt $1 \rightarrow 0$ of TIM bit as the basic signal and employ the distances between such signals as alphabets to encode information. An OBC-enabled client detects raw signals ($1 \rightarrow 0$) from the TIM bit sequence of a source client, and decodes out embedded information by interpreting the patterns created by signal distances. We demonstrate that, by subtly selecting the encoding symbols (i.e., signal distance), client can decode information with accuracy close to 100%. As OBC encodes information based on regulations of client's normal traffic activities, it incurs no extra packet exchanges on the Wi-Fi communication band.

We apply OBC in our BackPSM to exchange coordination information (client's communication slots and period) between clients. A client acquires traffic patterns of peer clients from the OBC side channel, and properly selects its own slots to avoid collision. With the wide adoption of BackPSM, it allows Wi-Fi clients to coordinate communications in a way akin to distributed TDMA, yet requiring no changes to the protocol.

We have implemented BackPSM on our testbed (10 wireless NICs connecting on Linux PCs and two Nexus 4 phones). We evaluate BackPSM on the testbed using both controlled experiments and real-world traffic traces. Our results show that BackPSM reduces screen-off system energy by up to 60%, as compared to the default SPSM strategy of Nexus 4. BackPSM outperforms state-of-the-art strategies by 16∼42%, and scales well to dense network.

In summary, this paper makes the following contributions.

- We identify two energy hot spots in PSM. We show that they are hard to be reduced in ordinary cases, but can be further optimized under screen-off traffic.
- We propose a novel Out-of-Band Communication framework, which enables Client-to-Client direct communications with no extra traffic. We believe that OBC may shed light on new potentials for phone-to-phone communication, Wi-Fi coordination and interference mitigation.
- We design and implement BackPSM, a new power saving strategy for screen-off Wi-Fi communications. We evaluate BackPSM using extensive testbed experiments.

II. Background and Motivation

To preserve battery power, smartphones today generally put Wi-Fi radio into a Power Save Mode (PSM) when the network interface is idle. The two widely adopted power management strategies are: *Static PSM (SPSM)* and *Adaptive PSM (APSM)*. Both SPSM and APSM put Wi-Fi radio into sleep (i.e., a low-power state) when there is no traffic, and wake up the radio periodically to listen Beacons, through which the AP informs clients the presence of downstream traffic. The main difference between SPSM and APSM is: SPSM switches a radio back to sleep immediately after completing a traffic activity; while APSM keeps radio active for a certain time, it only sleeps the radio if no traffic is arrived during the time. Therefore, APSM has lower traffic delays, but SPSM may save more energy [15].

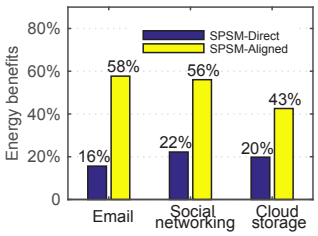


Fig. 1. Energy benefits of SPSM-Direct and SPSM-Aligned over APSM.

Since screen-off traffic can often tolerate long delays due to absence of user interactions [7], one may prefer smartphones adopting SPSM during screen-off periods. However, we observe that there is still room for improvement while applying SPSM to the screen-off traffic. We identify the energy saving opportunities for SPSM in screen-off scenarios as below.

Opportunity 1 – radio switch energy on upstream sending: When an SPSM client sends upstream packet, it has to switch radio into active state for communication and switch back to sleep after packet sending. Such frequent radio switches may degrade the energy benefits of SPSM, especially for the screen-off traffic which are mainly frequent, short packets [7].

We have measured the energy consumption of a smartphone (Nexus 4) with SPSM and APSM under the screen-off traffic of different apps. From the results (SPSM-Direct in Fig.1), we see that the energy gain of SPSM over APSM is only around 20%, far below our expectations. The energy benefit of SPSM is impaired by the expensive radio switch costs of upstream sending, because when we defer upstream sending to the time when a client regularly wakes up to listen Beacon (SPSM-Aligned in Fig.1), the energy benefit increases accordingly to nearly 50%. Ideally, a client should send upstream packets in batches to amortize the radio switch costs. Such a strategy does not work for the ordinary user traffic because it will add delays to packet sending, impairing user experiences. Whereas in the case of delay-tolerant screen-off traffic, we can exploit the idea to achieve significant energy savings.

Opportunity 2 – contention energy on downstream receiving: When multiple SPSM clients reside in the same Wi-Fi cell (or nearby cells working at the same channel), they wake up to receive buffered downstream packets from the AP at the same time (i.e., beginning of a beacon interval) [11]. Contentions between these clients will make them awake for long duration, causing up to 4 times more energy consumption. Motivated by these results, we want our approach to avoid contentions by isolating screen-off devices into different beacon intervals to receive downstream packets.

It is worth to note that the idea of isolating client traffic at different time slices to optimize contention energy has been used by NAPman [15] and SleepWell [11]. However, due to small-delay restrictions of ordinary traffic, they can only divide

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.

limited number of time slices within the time of one beacon interval, and thus face scalability issues when client number becomes large (see Fig.9, Section V-B2). The delay-tolerance nature of screen-off traffic enables us to break the limit of one beacon interval. We exploit coarse-grained time slice, in unit of beacon interval, to isolate client traffic for better scalability.

Coordination problem. By addressing the two opportunities, we tend to design a new power saving strategy, on base of SPSM, for screen-off traffic. The new strategy regulates client to send and receive packet in batches, and coordinate multiple clients to communicate at different beacon intervals in a way like TDMA. We implement the strategy at client side since the control of upstream sending can only be realized at client. Here, the core problem is *how to coordinate clients without incurring extra traffic overheads*. We handle the problem by proposing a novel *out-of-band coordination* scheme, which will be presented in the next section.

III. System Design

A. Screen-off Transmission Model

We divide the screen-off traffic of smartphone into multiple transmission tasks, where a task w typically corresponds to a packet. w can be upstream or downstream. Screen-off traffic is scheduled in time unit of beacon interval, termed *slot*. We denote the slot that task w arrives as $t_a(w)$, and the slot that w is scheduled as $t_s(w)$. We assume that w is delay-tolerant.

We denote the set of clients (i.e., screen-off smartphones) by $\mathcal{C}$. W stands for the set of transmission tasks of all clients. Let $v(w)$ denote the traffic volume (in bytes) of task w, and $c(t)$ the capacity of slot t, i.e., the maximum number of data bytes that can be transferred through AP. The condition

$$\sum_{w \in W \cap \{w | t_s(w) = t\}} v(w) \leq c(t) \tag{1}$$

should be met for any slot.

B. Out-of-band Wi-Fi Coordination Scheme

TIM channel. A Wi-Fi AP uses the TIM (Traffic Indication Map) field of Beacon to inform PSM clients the presence of buffered downstream packets. Assume c_i is a PSM client. A downstream packet of c_i arrives at the AP at t_a. The AP will set c_i's TIM bit to '1' in the next slots. After c_i retrieving the packet at t_s, the AP will clear c_i's TIM bit (i.e., set c_i's TIM bit to 0) at slot t_s+1. Therefore, when the downstream packets of c_i are pending at the AP, c_i can regulate local packet receiving operations to control the appearing of $1 \rightarrow 0$ on TIM bit. Thus, it can further adopt the time length in between two $1 \rightarrow 0$ signals to convey information. A peer client detects $1 \rightarrow 0$ from the TIM bit sequence of c_i, and decodes the embedded information from patterns created by such $1 \rightarrow 0$ signals. Since Beacon (and the TIM field included) is broadcasted by the AP at every slot, all clients can obtain TIM bit sequences of other peers. It provides a free side channel for client-to-client direct communications. Specifically, we call the TIM bit sequence of c_i as c_i's *TIM channel*, as illustrated in Fig.2.

We present the detailed design of our out-of-band coordination scheme as below.

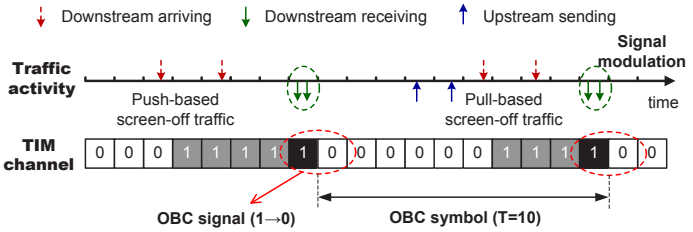


Fig. 2. Illustration of TIM channel.

1) Coordination information encoding: We adopt $1 \rightarrow 0$ of TIM bit as signal and employ the time length in between two signals (i.e., *signal distance*) as symbol to encode information, as shown in Fig.2. We term the set of signal distances used for information encoding as *alphabet set*, denoted by $\mathcal{A}$. For example, if $\mathcal{A} = \{2, 5\}$, we may use length 2 to encode an information bit '0' and 5 for bit '1'. In our context, a client only needs to exchange communication slots and periods with peers. We can simply adopt $\mathcal{A} = \{T\}$, where T corresponds to the communication period of client. The slot where $1 \rightarrow 0$ appears, termed *signal slot*, also carries information, i.e., the communication slot of client.

Next, we address the problem of how to modulate signal $1 \rightarrow 0$ on TIM channel. Assume that client c_i has continuous downstream traffic. We schedule c_i to receive downstream packets from the AP at every T slots. $1 \rightarrow 0$ appears at c_i's signal slots if two conditions are satisfied:

- at least one packet arrived at AP during the last period,
- all packets must be retrieved in the communication slot.

If the first condition is failed, $0 \rightarrow 0$ will occur at the signal slot. We call it a *zero noise*. While the second condition is failed, $1 \rightarrow 1$ will occur and we call it *one noise*. We want to reduce noise rate as low as possible. We achieve this by delicately selecting T for client c_i as below.

Theorem 1: Given that the arriving of downstream traffic is a Poisson process, λ is the average number of packets arrived per slot. The packet size of downstream traffic follows normal distribution $N(\mu, \sigma^2)$. We demand $\frac{1}{\lambda} \leq T \leq \frac{c(t)}{\mu\lambda}$, where $c(t)$ corresponds to the capacity of c_i's communication slot t.

Proof: Let N_T denote the number of downstream packets arrived in one period. Since N_T follows a Poisson distribution, we have $E(N_T) = \lambda T$.

To ensure $1 \rightarrow 0$ appearing at a signal slot, N_T must meet the above two conditions, which are captured by

$$N_T \geq 1 \quad \text{and} \quad \mu N_T \leq c(t).$$

Substituting $E(N_T)$ into the inequalities, we obtain

$$\frac{1}{\lambda} \leq T \leq \frac{c(t)}{\mu\lambda}.$$

Using the Poisson p.d.f. (probability distribution function) of N_T, we can compute the zero noise rate as

$$P(N_T < 1) = P(N_T = 0) = e^{-\lambda T},$$

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.

and the one noise rate

$$P(\mu N_T > c(t)) = \sum_{k=N_c+1}^{\infty} \frac{(\lambda T)^k}{k!} e^{-\lambda T} = 1 - \sum_{k=0}^{N_c} \frac{(\lambda T)^k}{k!} e^{-\lambda T}$$

where, $N_c = \lfloor \frac{c(t)}{\mu} \rfloor$. ■

2) Coordination information detection & decoding: When a client receives information from TIM channel, it involves three procedures: signal detection, symbol demodulation and information decoding. The signal detection procedure detects $1 \to 0$ from TIM channel. Then the symbol demodulation procedure extracts the encoding symbol (i.e., T) of source client from detected signal distances. Once symbol T and the signal slots where $1 \to 0$ appears are determined, information decoding is straightforward: we interpret detected signal slots as the source client's communication slots and T as period. We focus on the first two procedures in the following.

Signal detection. Noises will impact the accuracy of signal detection. Assume that the AP does not drop client's downstream packets. If $1 \to 0$ appears, it would only be caused by source client c_i retrieving all buffered packets from the AP at the slot. It is an authentic signal modulated by c_i. Therefore, the *false positive* rate (i.e., detecting $1 \to 0$ as a signal while it is not) of signal detecting is 0. However, when noise (either zero noise or one noise) occurs at the signal slot, a peer client can not detect $1 \to 0$, resulting in a *false negative* error. We conclude the analysis above with Theorem 2.

Theorem 2: Let P_I, P_{II} denote the false positive and false negative rate of signal detecting, respectively. We have $P_I = 0$, $P_{II} = 1 - \sum_{k=1}^{N_c} \frac{(\lambda T)^k}{k!} e^{-\lambda T}$.

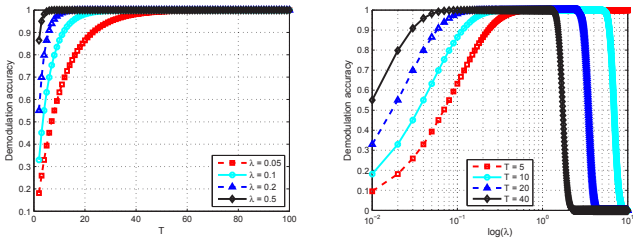
Proof: $P_I = 0$ is straightforward based on the analysis above. According to Theorem 1, we have $P_{II} = P(N_T < 1) + P(\mu N_T > c(t)) = 1 - \sum_{k=1}^{N_c} \frac{(\lambda T)^k}{k!} e^{-\lambda T}$. ■

Symbol demodulation. Let d stand for the distance between detected signals. As false negative error is the only error that may occur in signal detecting, the detected signal distance (d) and the source client's encoding symbol (T) would meet $d = nT, (n = 1, 2, \cdots)$. If no error occurs, $d = T$, otherwise $d > T$. Hence, we may use multiple signal distances to calibrate each other and choose the minimum distance from them as our demodulated symbol. Formally, let d_i represent the i'th latest signal distance, T^* denote the demodulated symbol. We have $T^* = \min_{i=1}^{L}\{d_i\}$, where L is the number of signal distances employed for calibration.

Theorem 3 gives the accuracy rate of this calibration based demodulation scheme when $L = 2$.

Theorem 3: Let p denote the error rate of signal detecting, $p = P_I + P_{II}$. $P(T^* = T)$ stands for the probability that demodulated symbol T^* equals to the source client's encoding symbol (T). When $L = 2$, $P(T^* = T) = 1 - p^2$.

Proof: When $L = 2$, we employ the latest and second latest signal distances (d_1 and d_2) for symbol demodulation. Suppose that $d_1 + d_2 = kT, (k = 2, 3, \cdots)$, i.e., d_1 and d_2 are obtained by $k+1$ times of signal detecting, among which only three signals (i.e., $1 \to 0$) are detected. We use $S_i, (i = 0, 1, \cdots, k)$ to denote the i'th latest signal detecting. Since the symbol demodulation procedure is invoked only when $1 \to 0$ is detected, we demand S_0 to be $1 \to 0$.



(a) (b)

Fig. 3. Relationships between demodulation accuracy (when $L = 2$) and λ, T. (a): demodulation accuracy vs. T with T ranges from 2 to 100 slots; (b) demodulation accuracy vs. log(λ), λ ranges from 0.01 to 10 frames/slot.

Theoretically, k can be any integer that is no smaller than 2. When $k = 2$, both S_1 and S_2 produce $1 \to 0$. We have

$$P(k = 2) = (1 - p)^2.$$

When $k = 3$, S_3 is $1 \to 0$, one of S_1, S_2 is $1 \to 0$ and the other is noise. Therefore,

$$P(k = 3) = (1 - p) \cdot C_2^1 (1 - p)p = 2p(1 - p)^2.$$

Generally, when $k = n$, S_n is $1 \to 0$, one of S_1, S_2, $\cdots$, S_{n-1} is $1 \to 0$ and the others are noises. We have

$$P(k = n) = (1-p) \cdot C_{n-1}^1 (1-p)p^{n-2} = (n-1)p^{n-2}(1-p)^2.$$

Note that $T^* = \min\{d_1, d_2\}$. When $k = 2$, $d_1 = d_2 = T$. Let $P(T^* = T|k = 2)$ denote the probability that $T^* = T$ under the condition of $k = 2$. We have

$$P(T^* = T|k = 2) = 1.$$

Generally, when $k = n, (n \geq 3)$, we can obtain $T^* = T$ only if $d_1 = T$ or $d_2 = T$, which requires either S_1 or S_{n-1} to be $1 \to 0$. Therefore, the success ratio under condition $k = n$ is

$$P(T^* = T|k = n) = \frac{C_2^1}{C_{n-1}^1} = \frac{2}{n-1}, \quad n \geq 3.$$

Finally, we compute the success ratio as

$$\begin{aligned} P(T^* = T) &= \sum_{n=2}^{\infty} P(T^* = T|k = n) \cdot P(k = n) \\ &= (1-p)^2 + \sum_{n=3}^{\infty} \frac{2}{n-1}(n-1)p^{n-2}(1-p)^2 \\ &= (1-p)^2 \left(1 + 2\sum_{n=3}^{\infty} p^{n-2}\right) \\ &= 1 - p^2. \end{aligned}$$

This completes the proof. ■

We also analyzed the demodulation accuracy when $L > 2$. However, due to page limit, we do not present the results in this paper. Actually, with $L = 2$, we can already achieve high

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.

enough accuracy. Figure 3 illustrates the relationships between demodulation accuracy and λ, T. We show that one can always properly select T for a given λ to achieve near 100% accuracy. The results in Fig.3(b) indicate that a selected T may produce high accuracy only in a certain range of λ, which is consistent with our analysis in Theorem 1. This suggests us to maintain T on-line based on the real-time value of λ, see Section III-C1.

3) Discussion—extending to general out-of-band communications: By adding more symbols into alphabet set $\mathcal{A}$, we can easily extend the above framework to the general Out-of-Band Communications (OBC). OBC enables clients to exchange information directly with each other (i.e., client-to-client communication), without sending extra packets through the regular Wi-Fi communication band. Since each client has a dedicated TIM channel, multiple clients can send and receive concurrently with no interference, providing high throughput. Moreover, we show that OBC is robust to Beacon losses and can generally work with APSM clients.

Handling Beacon loss. The loss of Beacon frame (i.e., TIM bit loss) may impact both the sending and detecting of signal $1 \rightarrow 0$. Observing that the TIM bit of a client changes only when the first downstream packet arrives or after all pending traffic being retrieved, we can restore missed TIM bits using two steps (Assume that the Beacon at slot t is lost. b_t denotes the TIM bit of slot t. We omit the analysis due to page limit.):

- set b_t with b_{t-1} at slot t,
- if t is expected to be a signal slot, reset b_t with b_{t+1} after receiving the Beacon at slot $t+1$.

Working with APSM client. If a client adopts APSM for Wi-Fi power saving, it can also control the change of $1 \rightarrow 0$ on TIM bit. However, different from SPSM clients that modulates signal $1 \rightarrow 0$ by issuing PS-Poll frames to retrieve all pending downstream packets, an APSM client modulates signal $1 \rightarrow 0$ by controlling the sending of Null data frames.

C. BackPSM Power Saving Strategy

Based on the out-of-band coordination scheme, we design a new power saving strategy, named BackPSM, for screen-off traffic. The high-level idea of BackPSM is to locally regulate screen-off traffic into periodic pattern and coordinate with peer clients to transfer in different slots, i.e., like the idea of TDMA. The motivation for periodic upstream sending is to let client process outgoing traffic in batches so as to amortize the radio switch costs. While the goal of periodic downstream receiving is to encode traffic pattern information onto the TIM channel for Wi-Fi coordination (i.e., to realize traffic isolation).

BackPSM divides time into slots. Since Beacons are broadcasted every beacon interval, it provides synchronized timing for slot managing on client. To deal with Beacon loss, we also use a local slot timer for backup. We present the pseudo-code of BackPSM in Algorithm 1. Procedure *Core* implements the main scheduling logic of BackPSM. It is invoked every slot, upon receiving a Beacon frame. *Core* extracts the TIM field of the Beacon and pass it to *PatternDecoding*, which parses the TIM bit sequence of every peer client and demodulates OBC symbols from it (lines 15-18) using the method presented in Section III-B2. *Core* schedules each client to send and receive at its own slots. At the end of every communication slot, *Core* invokes *PeriodManaging* and *SlotMaintaining*. We will detail the strategies of period managing and slot selecting below.

```
1:  procedure CORE(Beacon)
2:      Get TIM field of this Beacon;
3:      Call PatternDecoding(TIM field);
4:      if (the current slot is client's communication slot)
5:          Send local outgoing traffic;
6:          if (the TIM bit of this client is set)
7:              Retrieve downstream traffic;
8:              N_T ← number of downstream packets;
9:          Call PeriodManaging(N_T);
10:         Call SlotMaintaining(SOM);
11: end procedure
12: procedure PATTERNDECODING(TIM field)
13:     foreach client c_i ∈ C
14:         Update the TIM bit sequence of c_i;
15:         if (1 → 0 occurs on c_i's TIM channel)
16:             Compute the latest signal distance d_0;
17:             T* ← min{d_1, d_0};
18:             d_1 ← d_0;
19:             foreach t_j in SOM, j = (k + nT*)%T_m
20:             // t_k is the current slot
21:                 U_j ← U_j ∪ {c_i};
22: end procedure
23: procedure PERIODMANAGING(N_T)
24:     if (1 → 1 appears in two successive periods)
25:         T' ← T/2;
26:     else if (0 → 0 appears in two successive periods)
27:         T' ← 2T;
28:     else
29:         λ ← N_T/T;
30:         T' ← the maximum T that makes P_II ≤ δ;
31:     T ← min_{n=1}^{m} {nT_0 | nT_0 ≥ T'};
32: end procedure
33: procedure SLOTMAINTAINING(SOM)
34:     K = {t_k | t_k ∈ SOM ∧ |U_k| = 0};
35:     if (this client (c_i) is entering BackPSM mode)
36:         Randomly select a slot from K;
37:         return;
38:     foreach client c_j ∈ U_i
39:         if (c_i, c_j appeared in any other slot of SOM)
40:             Add c_j to collision set R;
41:     if (|R| > 1)
42:         q ← the order of c_i's client ID in R;
43:         if (q > 1)
44:             Select the (q − 1)'th slot in K;
45: end procedure
```

Algorithm 1: The BackPSM scheduling algorithm

1) Managing period: In BackPSM, T is used as not only the period to schedule client traffic, but also the basic symbol for out-of-band coordination. The setting of T should ensure high reliability of OBC. We set T based on Theorem 2. We introduce δ, an upper bound on the error rate of OBC signal

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.

detection, and demand $P_{II} \leq \delta$. Hence, we compute T as $T = \max\{T|P_{II}(T) \leq \delta\}$. Recalling from the expression of P_{II} given by Theorem 2, we need parameter λ (i.e., the average number of downstream packets arrived at the AP per slot) to solve $P_{II}(T) \leq \delta$. We measure N_T, the number of packets received by client at the communication slot, and estimate λ with $\lambda = \frac{N_T}{T}$.

However, N_T only represents the number of packets arrived at the AP when $1 \rightarrow 0$ appears at the signal slot. If $1 \rightarrow 1$ (one noise) appears, it implies that more packets are still pending at the AP, and thus received packet number N_T is smaller than the number of arrived packets. We remedy it by letting a client communicate more frequently, i.e., cutting period T by half. Similarly, we double the period when zero noise occurs. Therefore, we adjust the period of a client on-line as below.

$$T = \begin{cases} \frac{T}{2}, & \text{one noise occurs for} \\ & \text{two successive periods,} \\ 2T, & \text{zero noise occurs for} \\ & \text{two successive periods,} \\ \max\{T|P_{II}(T) \leq \delta\}, & \text{otherwise.} \end{cases} \tag{2}$$

Specifically, to facilitate coordination among clients, we set up a lower bound T_0 and an upper bound T_m, $(T_m = mT_0)$, on T, and demand $T = nT_0, n = 1, \cdots, m$. We round T in Eq. (2) to the nearest nT_0 to obtain the final period.

2) Selecting slot: After acquiring traffic patterns of other peers through OBC (see procedure *PatternDecoding*), a client can select the slot that is not occupied by any peers to communicate. However, if multiple BackPSM clients join at the same slot, they would obtain the same knowledge on traffic patterns of existing clients and thus may choose the same slot, resulting in collisions. We employ a hash-table like idea to resolve slot collision. The implementation of our slot selecting strategy relies on a data structure named Slot Occupation Map (SOM), which is described as below.

Slot Occupation Map (SOM). Figure 4 shows the structure of SOM. It is an array with T_m elements. Each element stands for a slot. We formally represent an SOM as $SOM = \{t_i|0 \leq i \leq T_m - 1\}$. For each slot t_i, U_i denotes the set of clients that communicate at t_i. If client $c_j \in U_i$, t_i is occupied by c_j. We set the length of SOM as T_m because T_m is the maximum length of a client's communication period. We can record the traffic patterns of all clients within T_m slots. We dynamically map the current time slot (t) to slot t_k in SOM as below:

$$k = t\%T_m. \tag{3}$$

By doing so, we are able to exploit a fixed SOM structure to represent the traffic patterns of all clients at any different time. For example, as shown in Fig.4, $T_m = 8$, assume the current time is mapped to t_5 in SOM. If client c_4 communicated in the current slot and $T = 4$, we can record c_4's traffic pattern by adding c_4 to U_1 and U_5, i.e., t_1, t_5 are occupied by c_4.

A client uses SOM to track peer clients' traffic patterns in procedure *PatternDecoding* (lines 19, 21). SOM is passed to procedure *SlotMaintaining* to facilitate client's slot selecting:

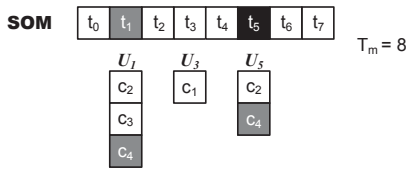


Fig. 4. Illustration of Slot Occupation Map (SOM): c_2 collides with c_4.

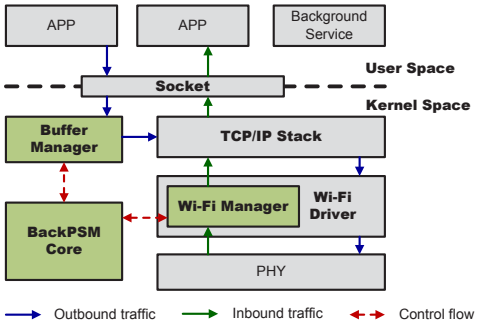


Fig. 5. The system architecture of BackPSM.

When a new BackPSM client joins, it randomly selects one slot from the candidate set $\mathcal{K} = \{t_k|t_k \in SOM \wedge |U_k| = 0\}$. It performs collision detection (lines 38-40) every period, at the end of communication slot. If slot collision is detected, a client computes the set of colliding clients (R). The client with the smallest ID stays in the current slot. The other $|R|-1$ clients re-select slot in the order of client ID (lines 41-44).

3) Overhead analysis: The main computation overhead of BackPSM comes from decoding the out-of-band coordination information, i.e., procedure *PatternDecoding*, which is invoked every slot. Let T_i denote the period of client c_i $(c_i \in \mathcal{C})$. In *PatternDecoding*, we parse the TIM bit of every peer client (say c_i) and record the traffic pattern in SOM when a signal $1 \rightarrow 0$ is detected, which occurs every T_i slots. Therefore, the computation complexity is $\sum_{i=1}^{|\mathcal{C}|}(1 + \frac{1}{T_i} \cdot \frac{T_m}{T_i})$. Since $T_0 \leq T_i \leq T_m$ (T_0 and T_m are constant), we have $(1 + \frac{1}{T_m})|\mathcal{C}| \leq \sum_{i=1}^{|\mathcal{C}|}(1+\frac{T_m}{T_i^2}) \leq (1+\frac{T_m}{T_0^2})|\mathcal{C}|$. The computation complexity of *PatternDecoding* is $O(|\mathcal{C}|)$. We also derived the computation complexity of *PeriodManaging* and *SlotMaintaining* as $O(\frac{1}{T_i})$ and $O(\frac{|\mathcal{C}|}{T_i})$. Therefore, the overall computation complexity of BackPSM is $O(|\mathcal{C}|)$, where $\mathcal{C}$ is the set of BackPSM clients.

The memory overhead of BackPSM mainly comes from data structure SOM. The required storage space of SOM is: $T_m + \sum_{i=0}^{T_m-1}|U_i| = T_m + |\mathcal{C}|$. Although $PatternDecoding$ parses the TIM bit of all peers, it does not need to store these TIM bits. Instead, we only record the latest detected signal distance (d_1) and the position of latest $1 \rightarrow 0$ signal, which consumes $O(1)$ storage space. *PeriodManaging* and *SlotMaintaining* also require just $O(1)$ storage. Therefore, the space complexity of BackPSM is $O(T_m + |\mathcal{C}|)$.

The overheads of $O(|\mathcal{C}|)$ time complexity and $O(T_m + |\mathcal{C}|)$ space complexity suggest that both the BackPSM strategy and our OBC scheme are lightweight. It is suitable to apply them on screen-off smartphones for power saving.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.

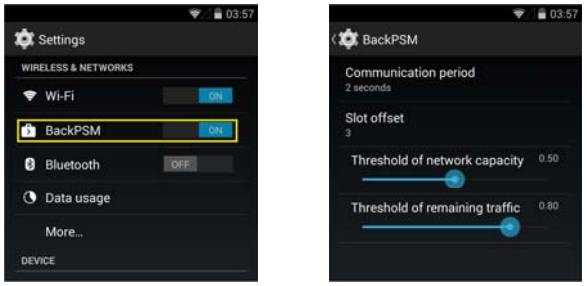


Fig. 6. BackPSM configuration UI.

IV. System Implementation

We have implemented BackPSM on two platforms: a smartphone platform (LG/Google Nexus 4) and a Linux-based PC platform (Ubuntu 14.04). The devices are shown in Fig.7. We employ the Ralink rt2800usb wireless NIC as Wi-Fi interface for PC. For the smartphone platform, Nexus 4 is equipped with Atheros WCN3660 Wi-Fi chipset. The OS version is Android 4.2 (kernel version: 3.1). BackPSM is implemented in kernel space. By default, smartphone starts BackPSM when screen is off for five minutes. We allow a user to manually enable or disable BackPSM in the Android system settings, as shown in Fig.6. We also provide a UI to configure the working parameters of BackPSM.

Figure 5 illustrates the system architecture of BackPSM. It includes the driver modification *Wi-Fi Manager* and two kernel components, *Buffer Manager* and *BackPSM Core*.

BackPSM Core implements the main scheduling algorithm of BackPSM. It receives out-of-band coordination information and controls two other components to send and receive screen-off traffic. *Buffer Manager* receives outgoing traffic from upper layer apps and transfers them in batches. *Wi-Fi Manager* is a component resident in the Wi-Fi driver. It delivers received Beacons to BackPSM Core and exposes downstream receiving to the control of BackPSM Core.

To ensure client sending upstream packets in its own slots, Buffer Manager intercepts the packets sent by upper layer apps before they are passed to the TCP/IP stack. Buffer Manager stores outgoing packets in an FIFO queue. When the client's slot starts, BackPSM Core sends a message to Buffer Manager. The later delivers pending packets to the TCP/IP stack to send them out in batches.

In the current implementation of Wi-Fi driver, received Beacons are processed by a *beacon_handler* that drops Beacon after processing. We modify it to pass Beacon to BackPSM Core after conventional processing. We also modify *beacon_handler* for downstream control. By default, *beacon_handler* issues PS-Poll to retrieve downstream packets from AP once detecting a TIM bit set. We modify the condition as '*TIM bit is set* $\wedge$ *the current slot is a communication slot*'.

V. Evaluation

A. Methodology

Devices. Our testbed consists of one server, one AP, and 12 Wi-Fi clients (two Nexus 4 phones and 10 wireless NICs connecting on 5 PCs). We employ the TP-LINK TL-WDR7500 wireless router as AP. Clients are placed within one meter of the AP to ensure good quality of Wi-Fi signal, as shown in Fig 7. We use an Agilent N6750A power meter to measure the realtime voltage and current of Nexus 4 phone, see Fig 7. To minimize measurement noise, we remove unnecessary applications, disable irrelevant hardware components, turn off screen and all network interfaces except Wi-Fi.

Fig. 7. left: testbed setup; right: Nexus 4 phone connected to power meter.

Traffic traces. We examined BackPSM using real screen-off traffic. Traffic traces are collected in our campus Wi-Fi. We find 20 volunteers with different smartphone use habits. We install a program on their phones to log traffic when phone is screen-off but connecting to Wi-Fi. For privacy, we anonymize each traffic record. We observe that 95% of collected traces are from Email, Social Networking Apps, Cloud Storage and News. We choose traces of a typical 30-minute session, and replay the traffic sequences in our testbed.

Comparison. To demonstrate the advantage of Back-PSM for screen-off traffic, we compare BackPSM with state-of-the-art strategies. Specifically, we use APSM and SPSM as two benchmarks. Since BackPSM operates on client, we compare it with BSD [9] and SAPSM [13], two representative client-side strategies in the literature. Finally, to evaluate the performance of BackPSM on contention energy optimization, we compare BackPSM with NAPman [15] in single AP and SleepWell [11] in multi-AP scenarios. We repeat each experiment for 5 times and compute the average as results.

B. Controlled Experiments

We first use controlled experiments to evaluate BackPSM. Unless otherwise stated, we set up client to send data request (500 Bytes) every 80ms. Upon receiving such a request, the server responses immediately with a 1024-Byte packet.

1) Power Consumption vs. Traffic Latency: We first compare BackPSM with two benchmark strategies, i.e., APSM and SPSM. Figure 8(a) plots a snapshot of power consumption for the three strategies in a network with 10 clients. As expected, APSM consumes the highest power, with an average of 633.5 mW. The next is SPSM, 532.4 mW on average. BackPSM has the lowest power consumption of 358.6 mW. As compared to APSM and SPSM, BackPSM reduces the screen-off system power by 43.4% and 32.6%, respectively.

Figure 8(b) compares the traffic latency of three strategies. Traffic latency is computed as the time from client sending request to that of the response being received. The average latency of BackPSM is 2447.2 ms. It is one order of magnitude larger than that of SPSM (484.3 ms), and two orders larger than APSM (19.1 ms).

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.

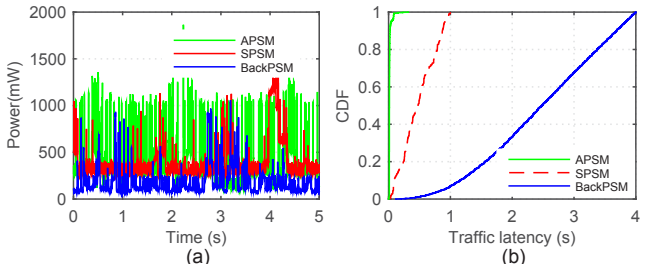


Fig. 8. Comparison of BackPSM with APSM and SPSM in a network of 10 clients: (a) Power consumption of Nexus 4 phone (i.e., screen-off system power); (b) Distribution of traffic latency.

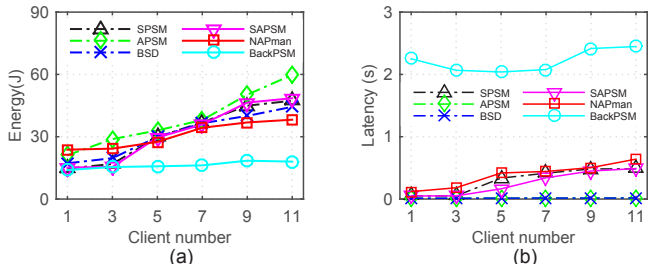


Fig. 9. Comparison of BackPSM with state-of-the-art strategies under different network scales: (a) Energy consumption; (b) Traffic latency.

The results above verify the energy benefit of BackPSM. But this benefit comes at the cost of much longer traffic latency. We argue that long traffic latency is acceptable for screen-off traffic, due to absence of user interactions.

2) Scaling to Dense Wi-Fi: We next compare our BackPSM with state-of-the-art strategies. In the experiments, we increase client number from 1 to 11, with a step of 2. Figures 9(a) and (b) exhibit the average energy consumption and traffic latency of these strategies under different network scales. We see that as client number increases from 1 to 11, the energy of state-of-the-art strategies increase dramatically (increased by 1 to 2 times), while BackPSM experiences only a slight increase. In network of 6 clients, the energy benefits of BackPSM over other strategies range from 52.8% to 61.1%. The benefit grows as network becomes dense. But the energy gain of BackPSM comes at the cost of long traffic latency, as shown in Fig.9(b).

3) Incremental Deployment: In practice, BackPSM clients may coexist with clients not employing BackPSM. In this experiment, we explore how BackPSM affects the performance of both kinds of clients. We set up 10 clients and vary the ratio of client adopting BackPSM from 0% to 100%, here the settings of 0% and 100% are used for benchmark comparison. The client not employing BackPSM will use SPSM instead. Figure 10(a) shows the results. As we can see, the energy of both BackPSM client and non-BackPSM client decrease dramatically as more clients adopt BackPSM. It indicates that deploying BackPSM is beneficial for both kinds of client.

C. Trace-driven Experiments

This section evaluates BackPSM using real-world traces of screen-off traffic. We examine the energy benefits of BackPSM in different real scenarios.

1) Energy Savings in Real-World: We compare BackPSM with state-of-the-art strategies using traces of synthetic screen-off traffic. In the experiments, we replay the same traces on each client and measure the energy drain of a Nexus 4 phone. Figure 10(b) shows the results. We observe that, as compared to SPSM and BSD, the energy benefit of BackPSM is not appealing when client number is small. In this case, the energy consumption of BackPSM is comparable to that of SPSM and BSD. BackPSM exhibits its advantage on energy saving when network becomes crowded. When client number reaches 6 and 10, BackPSM saves 16.1∼42.5% more energy, as compared to other strategies. This validates the effectiveness of BackPSM in real scenarios.

2) Coexisting with Foreground Traffic: In practice, screen-off traffic often coexist with foreground traffic, i.e., the traffic from active devices (called *foreground client*). In this experiment, we study the impact of foreground traffic on BackPSM performance. We setup 10 devices in the network and vary the ratio of foreground client from 0% to 80%. We employ APSM for foreground clients, and BackPSM for screen-off clients. We measure the energy drain of BackPSM client under different network configurations. Figure 10(c) shows the results. We see that BackPSM energy increases only a little while the ratio of foreground client increases. This implies that foreground traffic does not impact much on BackPSM.

3) Multi-AP Scenario: Finally, we setup trace-driven experiments to evaluate BackPSM in the multi-AP scenarios. We increase the number of APs from 1 to 3, with each AP serving four clients. We use SleepWell, the best energy saving strategy for multi-AP network in the literature, as comparison. Figure 10(d) compares the energy performance of two strategies. We see that client energy with SleepWell increases from 28.6J to 45.2J, while that of BackPSM increases from 16.0J to 24.6J. BackPSM outperforms SleepWell by up to 45.6%.

VI. Related Work

Energy saving in screen-off traffic. Recent works [7][3] [4] have highlighted the energy issues of smartphones during screen-off periods. Prior efforts [1] [16] mainly focus on the energy consumption of screen-off communications in 3G/LTE. Peng et al.[12] study the power consumption of Wi-Fi interface when smartphone is in suspend mode. They propose to smartly filter out unwanted broadcast Wi-Fi traffic for energy saving. In contrast, our work addresses the wanted (i.e., useful) traffic.

Energy saving in Wi-Fi. To optimize energy consumption of Wi-Fi, extensive researches have contributed on aggressive sleeping [10], sleep and wake up scheduling [9][13], and adaptive clock rate control of Wi-Fi radio [19]. More recent works (Scheduled PSM [5], SOFA [18], NAPman [15] and SleepWell [11]) address the energy waste problem caused by contentions during downstream receiving. These strategies isolate client traffic into different micro time slots within one beacon interval to avoid contentions. They all locate at the AP to centrally coordinate traffic slots for clients. In contrast, our strategy locates at client side and employs a novel scheme to enable client coordination, with no extra traffic overheads.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.

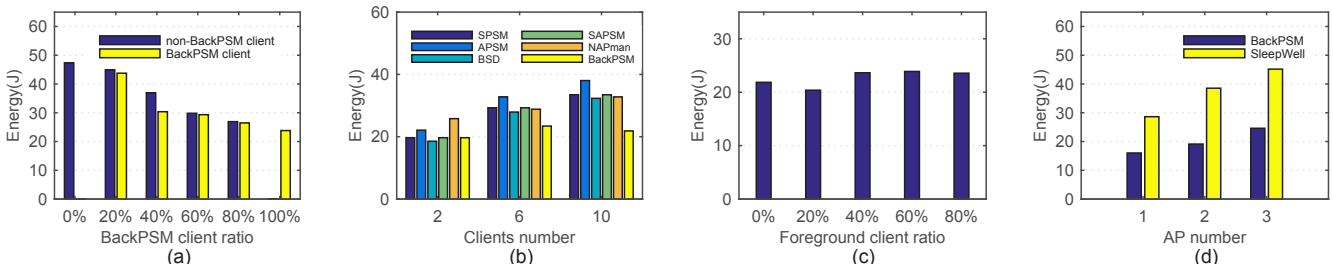


Fig. 10. (a) Energy consumption of BackPSM client vs. non-BackPSM client under different ratio of clients employing BackPSM; (b) Energy comparison of BackPSM with state-of-the-art strategies under real-world screen-off traffic; (c) Energy consumption of BackPSM client under different ratio of foreground clients; (d) BackPSM vs. SleepWell under different number of Wi-Fi APs.

We adopt coarse-grained slots for screen-off traffic isolation, which comes with advantages of better scalability.

Coordination in wireless communication. Scheduled PSM [5] and OpenTDMF [17] are two works that enable TDMA in today's Wi-Fi and WLAN networks. However, they both adopt a central controller (a Wi-Fi AP or the WLAN back-haul) to coordinate all clients, and thus require either protocol modifications or extra packet exchanges to disseminate coordination information to clients. In contrast, our approach enables clients to coordinate with each other in a distributed manner yet with no extra traffic overheads. Our idea of OBC is partly inspired by studies in cross-technology communication [2] [20] [21] [8], where the focus is how to encode information in a form that can be understood by client employing a different wireless technology. Whereas OBC establishes a side channel for only homogeneous Wi-Fi clients associated to the same AP. OBC brings a new Client-to-Client communication paradigm, i.e., akin to Wi-Fi Direct [6], to a Wi-Fi network, in addition to the traditional AP-Client communication. But different from Wi-Fi Direct, OBC requires no extra traffic on Wi-Fi communication band. As far as we know, this is the first work on 'packet-free' Client-to-Client communications.

VII. Conclusion

The key innovation of this work is the Out-of-Band Communication paradigm (OBC) that enables client-to-client direct communications yet without incurring extra traffic. We apply OBC to BackPSM, our new power saving strategy designed for screen-off Wi-Fi communications. OBC enables clients to exchange traffic patterns with each other and coordinate traffic in a way like distributed TDMA. Our extensive evaluations show that BackPSM reduces screen-off system energy by up to 60% and outperforms state-of-the-art strategies. We believe our work may open up new potentials on Wi-Fi coordination, interference mitigation and Client-to-Client communications.

References

[1] A. Chakraborty, V. Navda, V. Padmanabhan, and R. Ramjee. Coordinating cellular background transfers using loadsense. In *Proc. of MobiCom'13*, pages 63–74, Sep 2013.

[2] K. Chebrolu and A. Dhekne. Esense: communication through energy sensing. In *Proc. of ACM MobiCom'09*, pages 85–96, Sep 2009.

[3] X. Chen, N. Ding, A. Jindal, Y. C. Hu, M. Gupta, and R. Vannithamby. Smartphone energy drain in the wild: Analysis and implications. In *Proc. of ACM SIGMETRICS'15*, pages 151–164, Jun 2015.

[4] X. Chen, A. Jindal, N. Ding, Y. C. Hu, M. Gupta, and R. Vannithamby. Smartphone background activities in the wild: Origin, energy drain, and optimization. In *Proc. of ACM MobiCom'15*, pages 40–52, Sep 2015.

[5] Y. He, R. Yuan, and W. Gong. Modeling power saving protocols for multicast services in 802.11 wireless lans. *IEEE Transactions on Mobile Computing (TMC)*, 9(5):657–671, May 2010.

[6] S. Hu, H. Liu, L. Su, H. Wang, T. Abdelzaher, P. Hui, W. Zheng, and Z. Xie. Towards automatic phone-to-phone communication for vehicular networking applications. In *Proc. of IEEE INFOCOM'14*, pages 1752–1760, Apr 2014.

[7] J. Huang, F. Qian, Z. M. Mao, S. Sen, and O. Spatscheck. Screen-off traffic characterization and optimization in 3g/4g networks. In *Proc. of ACM IMC'12*, pages 357–363, Nov 2012.

[8] S. Kim and T. He. Freebee: Cross-technology communication via free side-channel. In *Proc. of ACM MobiCom'15*, pages 317–330, Sep 2015.

[9] R. Krashinsky and H. Balakrishnan. Minimizing energy for wireless web access with bounded slowdown. In *Proc. of MobiCom'02*, pages 119–130, Sep 2002.

[10] J. Liu and L. Zhong. Micro power management of active 802.11 interfaces. In *Proc. of MobiSys'08*, pages 146–159, Jun 2008.

[11] J. Manweiler and R. Choudhury. Avoiding the rush hours: Wifi energy management via traffic isolation. *IEEE Transactions on Mobile Computing (TMC)*, 11(5):739–752, May 2012.

[12] G. Peng, G. Zhou, D. Nguyen, and X. Qi. All or none? the dilemma of handling wifi broadcast traffic in smartphone suspend mode. In *Proc. of IEEE INFOCOM'15*, pages 1212–1220, Apr 2015.

[13] A. Pyles, X. Qi, G. Zhou, M. Keally, and X. Liuy. Sapsm: Smart adaptive 802.11 psm for smartphones. In *Proc. of ACM UbiComp'12*, pages 11–20, Sep 2012.

[14] S. Rosen, A. Nikravesh, Y. Gao, Z. Mao, F. Qian, and S. Sen. Revisiting network energy efficiency of mobile apps: Performance in the wild. In *Proc. of ACM SIGCOMM IMC'15*, pages 51–60, Oct 2015.

[15] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu. Napman: Network-assisted power management for wifi devices. In *Proc. of MobiSys'10*, pages 91–105, Jun 2010.

[16] F. Xu, Y. Liu, T. Moscibroda, R. Chandra, L. Jin, Y. Zhang, and Q. Li. Optimizing background email sync on smartphones. In *Proc. of ACM MobiSys'13*, pages 55–68, Jun 2013.

[17] Z. Yang, J. Zhang, K. Tan, Q. Zhang, and Y. Zhang. Enabling tdma for today's wireless lans. In *Proc. of IEEE INFOCOM'15*, pages 1436–1444, Apr 2015.

[18] Z. Zeng, Y. Gao, and P. Kumar. Sofa: A sleep-optimal fair-attention scheduler for the power-saving mode of wlans. In *Proc. of ICDCS'11*, pages 87–98, Jun 2011.

[19] X. Zhang and K. Shin. E-mili: Energy-minimizing idle listening in wireless networks. In *Proc. of MobiCom'11*, pages 205–216, Sep 2011.

[20] X. Zhang and K. Shin. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *Proc. of IEEE INFOCOM'13*, pages 3093–3101, Apr 2013.

[21] Y. Zhang and Q. Li. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *Proc. of IEEE INFOCOM'13*, pages 1366–1374, Apr 2013.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on March 24,2024 at 15:37:55 UTC from IEEE Xplore. Restrictions apply.