A Power-Aware Distributed Wi-Fi Access Point Scheduling Algorithm

Asma Enayet, Nusrat Mehajabin and
Md. Abdur Razzaque [*]
Green Networking Research Group
University of Dhaka, Bangladesh
asmaenayet, nusrat.47.cse@gmail.com,
razzaque@cse.univdhaka.edu

Choong Seon Hong [†]
Networking Laboratory, School of Electronics
and Information, Kyung Hee University
1732 Deokyoungdaero, Giheung-gu, Yongin-si,
Gyeonggi-do, 446-701, Korea, South Korea
cshong@khu.ac.kr

ABSTRACT

Smartphone has implemented several power saving strategies for its Wi-Fi radio as the Wi-fi radio consumes significant amount of energy. Previously, Wi-Fi energy saving mechanisms have been designed in such a way that add unfairness to the Wi-Fi network. Prime reasons of the unfairness are retransmissions due to collisions in the channel, unnecessary waiting in high power mode and hidden terminals in the vicinity.

We propose PowerNap an improved distributed energy efficient access point (AP) scheduling algorithm that addresses the above issues. PowerNap schedules the APs to scale down the overlapping among the transmission time of APs in the same vicinity. This trims out the energy consumption of Smartphone and also avoids unfairness. PowerNap schedules the APs in a weighted manner and also supports dynamic rescheduling. The proposed algorithm could be implemented via software installation in APs. Implementation of the PowerNap prototype improves Smartphone battery life upto 49- 60%.

Categories and Subject Descriptors

H.4 [**Mobile networking**]: Miscellaneous; D.2.8 [**Mobile networks**]: Energy savings—*Algorithms, performance measures*

General Terms

Algorithms, Wi-Fi, Smartphone

Keywords

Wi-Fi Network, Energy Optimization, Distributed Scheduling Algorithm, Traffic Measurement

1. INTRODUCTION

Today, Smartphones support numerous applications, added processing power, and ease of the Internet access. They require extended battery life to satisfy the user needs. More explicitly, users demand longer battery life in wireless handheld devices such as Smartphones to receive uninterrupted services on the go without compensating the number of running applications. Thus, it has always been a critical goal for Smartphones to optimize energy. Improved user experience and productivity will result from this optimization. This certainly necessitates optimizing the energy hungry processes so that the power consumption is minimized. The major power consuming components are display, sensors, communication and computation circuitries [7, 8]. Widespread mobile networking technologies, for instance 3G, GSM and Wi-Fi, necessitate power draining at high frequency. For Wi-Fi, a large portion of consumed power is wasted at the Access Point (AP) level due to network contention [8, 5]. In this work, our goal is to diminish this loss of energy by scheduling the AP's transmission periods in such a way that their time slot overlapping and the traffic contentions are reduced.

The optimization of network activity regarding Smartphone with a view to prolong battery life has several state-of-the-art implementations and solutions. The 802.11 PSM (power saving mode) [4] is implemented as the default power conservation strategy for Wi-Fi with the limitation of being able to only economize energy if the network activity is idle for a certain predefined duration. Also, PSM does not account for situations where multiple clients may wake up at the same time to use the channel, which results in heavy contention in the network. This problem has been addressed in NAPman [8] that prioritizes the packets and provides a virtual copy of an AP to every connected user. Thus, NAPman supports multiple clients under an AP without collision through virtualization. However, the virtualization of one AP does not care for the existence of multiple physical APs in the same vicinity. Therefore, traffic contention will be increased in the environment, wherein, there are many overlapping APs. Another power saving approach is Catnap [2] that introduces operating system (OS) buffer to store the packets from wired part of the network and postpone the delivery to the wireless section until the deadline. But it imposes a constraint that the wireless bandwidth has to be larger than the wired to be able to save energy. A different approach towards power saving in Smartphones via packet aggregation is found in

---

[*] Dept. of Computer Science & Engg., University of Dhaka
[†] Dept. of Computer Engineering, Kyung Hee University

ICUIMC(IMCOM)'13 , January 17-19, 2013, Kota Kinabalu, Malaysia

Low Energy Data-packet Aggregation Scheme (LEDAS) [6]. LEDAS aggregates the upper layer packets and construct them into MAC layer bursts. An AP scheduling algorithm which takes into account the possibility of multiple APs and multiple clients attached to them is SleepWell [5]. It avoids rush hours, that is, it arranges the turn off an AP when every other AP is out of work. But hidden terminals add unfairness to the system and introduce overlapping in the transmission time of the APs. Hence, the time slots of different users are multiplexed and a Wi-Fi Smartphone keeps awake even if it has no part in transmission.

In this paper, we design a distributed network access point scheduling algorithm, named PowerNap, for a dense deployment environment that saves energy consumptions of Wi-Fi client devices. Our algorithm is fully distributed; every AP exploits single hop neighbor traffic information only and is capable of taking transmission decisions for itself. A dynamic rescheduling is also employed in the algorithm for robustness. As there is no central controller, the algorithm is scalable. We use a workload estimation based algorithm so the other network parameters such as heterogeneity, latency, etc. do not affect the execution of the algorithm. The key contributions of our work are summarized below:

- We design a distributed network access point transmission scheduling algorithm for saving power of Smartphone.
- We use weighted fair share of transmission opportunities for all the APs according to their workloads.
- The unused portion of timeslots in a transmission round is also distributed in a weighted manner.
- Our performance evaluation shows that the proposed PowerNap scheduling algorithm outperforms the state-of-the-art works.

The remaining of the paper is organized as follows. The Section 2 contains a study on the related works in this field of research. In Section 3, we have presented the network model and assumptions for our proposed work. The Section 4 gives an explicit insight of our proposed scheduling algorithm PowerNap and the Section 5 presents the performance evaluation of the algorithm. Finally, we conclude the paper in Section 6.

2. RELATED WORKS

Being an enriched area of research, the optimization of energy consumed through Wi-Fi has received many proposals over last few years [11, 10, 9, 1]. The existing mechanisms can be classified into several categories: link status monitoring (PSM), bandwidth depended mechanism (Catnap), scanning using cellular towers (footprint) and AP scheduling algorithm (NAPman, SleepWell) are the significant types.

The 802.11 PSM [4] is the default power saving methodology for all the access points. The approach is to doze off the device when there is no network activity for a predefined duration of time and wake up at low power mode on every beacon interval to check for beacons frames intended for this device. If the user id matches, PSM switches to high power transmission mode. The main shortcoming of this method is that the client Wi-Fi devices stay in high power state for long time without any transmissions, only to check whether it is time to sleep or not.

NAPman (network assisted power management protocol) [8] conserves energy of Wi-Fi devices by creating several virtual APs' of one physical AP and provides a device abstraction that it is connected to an AP having one to one relation. The problem here is that one AP is capable of supporting a certain number of virtual APs; when the number of client devices per AP exceeds the threshold, the service will be disrupted. Furthermore, NAPman only deals with one physical AP, whereas, the energy consumption is worse when multiple APs are present in same wireless vicinity.

The authors of Catnap [2] exploited the situation where the bandwidth of wireless part is greater than that of wired part. It stores the delay tolerant packets into a middlebox to transfer later on deadline, when the AP turns on. This approach is not applicable to networks which have higher wired bandwidth than wireless. And installing storage on every central controller is not an efficient move either.

In SleepWell [5], the authors presented a mechanism to avoid rush hours, i.e., to transfer data when the network is least competitive and to schedule the user's wakeup times accordingly. Multiple access points contend for the channel and meanwhile the users connected to these APs are awake and wasting energy but SleepWell migrate the beacon frames in a manner that they do not overlap with each other. Our concept has the much similarity with the SleepWell excepting the following distinct differences:

- Weighted fair share from start point, i.e., we estimate the workload of all the APs and distribute the transmission time to the contending APs according to their workloads.
- To avoid overlapping, we use the least workload estimated for a certain AP. The estimation process involves the AP itself and its one hop neighborhood APs.
- We also adopt undershooting in the least estimated workload to achieve actual transmission time of the AP in concern.
- We also addressed the existing hidden terminal problem in our algorithm.

3. NETWORK MODEL

We consider several APs in a wireless vicinity, as shown in Fig. 1, where each AP is in the transmission range of its one hop neighbor AP. Each AP sends periodic update messages to its neighbors.

Thus, we can assume that AP_C is not in the transmission range of AP_A. In this case, AP_A has no knowledge of AP_C, hence it does not include AP_C for fair share calculation. Fig. 2(a) is the transmission round produced by AP_A. We also see that AP_C has no contribution in the fair share but it will definitely have impact on AP_A's fair share as it has contribution in AP_B's fair share since its a neighbor of AP_A.

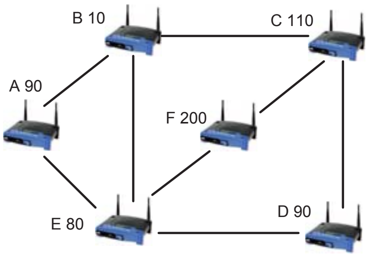
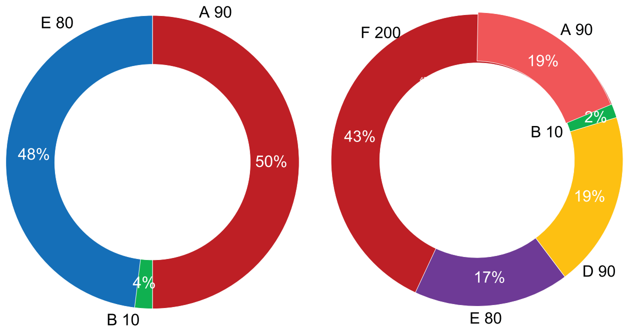


Figure 1: Sample Network Environment.



(a) AP A's transmission round (b) AP E's transmission round

Figure 2: Transmission rounds of access points A and E without any scaling in transmission times.

Here, the fair share for AP_A is 50% of the total transmission period. But, when we look at the transmission round produced by AP_E in Fig. 2(b) we see AP_A has a fair share of 19. So in the first case AP_A will be prepared to transmit for 50 unit of time but will be adding contention to the network and hampering other AP's transmission. This is clearly is a source of energy wastage as the APs' will have be awake in high energy consuming transmission mode during the contention. To minimize this contention we design PowerNap, an access point scheduling algorithm which will intelligently decide the suitable time and time-span for a given AP.

3.1 Design Principle of PowerNap

We designed such an algorithm were one AP will minimally interfere in another's transmission. The APs will be staggered throughout the transmission round to minimally overlap with others transmission.

Our design principles are:

- Choosing the least fair share calculated for a particular AP
- Use of undershooting to avoid unfairness
- Allocating the unused portion with respect to the minimal weighted fair share

We let every neighbor to calculate for the given AP and by selecting the least estimated share we avoid unfairness as much as possible. In this manner an AP can include the impact of APs' who are not visible to the given AP. At last dynamic rescheduling helps the network to cope with the bursty nature of Internet traffic.

In the next section, we give detailed description of the proposed algorithm and its behavior.

4. PROPOSED SCHEDULING ALGORITHM

In this section, we describe the detail operation of the proposed scheduling algorithm PowerNap that minimizes the transmission overlapping among the collocated APs. Every AP transmits beacon frames at regular intervals (typically after 100ms). The beacon frame consists of traffic information (beacon interval, time stamp, and traffic indication map (TIM)) related to that particular AP. In wireless network, the client device is synchronized with the AP using beacon frame's time stamp attribute. Every AP computes its initial workload using the beacon frame's TIM attribute.

The functionality of our proposed PowerNap can be divided into following four phases: initialization, traffic minimization, transmission using undershooting and scaling and dynamic rescheduling. The phases are described in detail in the following subsections.

4.1 Initialization

We define the initial state of every access point when there are data packets for transmission. And these APs are able to estimate their own transmission time demands. The access points prepare themselves for the rest of the processes in this phase. We can also consider it as an information collection phase. For transmission time estimation the AP need environment characteristics e.g. channel characteristics, AP's own capacity etc. and traffic demand of associated clients.

4.1.1 Initial Workload

The term workload defines the amount of network activity within a beacon interval. Every AP gathers information from the clients associated with it. The clients pass its bit rate demand to the AP. After that, the AP calculates its workload with respect to the connected user numbers, their bit rate demands and channel characteristics.

$$IWL_{AP_i} = \Sigma(BR_u)/C_{AP_i}$$

Here IWL = initial workload
i = index of APs
BR = user bit rate
u = index of users connected with that particular AP
C_{AP_i} = Capacity of AP_i

The Initial Workload defines the characteristics of a given AP to its neighbors so every AP shares this Initial Workload with its neighbors.

4.1.2 Workload Information Exchange

An AP shares its initial workload among the adjacent APs to build an understanding of the network environment and characteristics. Here, every AP gets the benefit of estimating the fairness of the network. Even though the estimation is based on one hop neighborhood we will generalize it for the entire network later.

The transmission interval is very small (typically 100ms) so the Initial Workload calculated needs to be scaled down. We use traffic minimization for this purpose.

4.2 Traffic Minimization

Every AP uses the information gathered in the initialization phase for traffic minimization. Here we use a threshold value for the scaling or minimization of traffic demand of every AP. This value must be chosen carefully considering the network environment. We consider saturated traffic and hence weighted fair share is used to enhance fairness of transmission among the AP's. The weight factor is the Initial Workload calculated in the previous phase.

4.2.1 Weighted Fair Share

The term weighted fair share (WFS) defines the share of an AP in transmission round with respect to the APs' workload in one hop neighborhood. The information shared in initialization phase is used to calculate an AP's own Weighted Fair Share. Weighted nature of the workload for using the transmission round let every AP who has queued traffic transmit its share, therefore the network transmission is fair.

The APs deal with huge amount of traffics and one AP is only aware of its one hop neighborhood traffic information and thus the WFS may seem fair locally. However, when we consider the entire network it will introduce unfairness to APs that are situated at more than one hops distance. Therefore, we opt for calculating the Weighted Fair Share for neighbors of an AP too to have a broader perspective of the network.

4.2.2 Minimal Weighted Fair Share

First, we let every AP to calculate its neighbor's WFS. This method minimizes the chance of hidden terminal problem, effecting the fair share calculation adversely. Because the more the AP is familiar with its environment the more it can increase its performance. Next, all the APs send what they calculated for one particular APi, to i. Finally i selects the least of all the weighted fair share from its neighbors and the one calculated by itself.

Because of the unpredictable nature of Internet traffic [3] and the fact that Internet traffic is increasing everyday plus a given access point at any given time would not have knowledge of the entire network; therefore one AP is likely to overlap with other AP's transmission time. To avoid such situation undershooting is used. Undershooting is the process of employing less than the practical value. A carefully chosen threshold will suffice for our purpose. By scaling down the transmission time we can have a fair transmission environment at cost of one set-up transmission round.

4.2.3 Minimal Weighted Fair Share Undershooting

To avoid unfairness we adopt minimal weighted fair share undershooting (MFS) so that every AP gets a proportional chance to transmit in one beacon interval. Without undershooting one AP is likely to spill into another AP's transmission slot even worse deprive another AP from transmission. To improve performance and integrity of the channel we undertook this measure. We set a threshold α. The MFS is deduced to α. Every AP of the same wireless vicinity will use the same threshold. This threshold is chosen based on channel characteristics and traffic pattern in nearby APs.

Algorithm 1 PowerNap Scheduling Algorithm, *for each AP*

```
Initialization:
calculate initial workload for $AP_i$;
$AP_i^{IWL} = \frac{\sum_{\forall u \in AP_i^{user}} BR_u}{AP_i^C}$;
Workload minimization :
Calculate weighted fair share (WFS),
$AP_i^{WFS} = \frac{AP_i^{IWL}}{\sum_{\forall AP \in Adj[AP_i]} IWL}$
for $AP_j \in Adj[AP_i]$ do
    $AP_{ij}^{WFS} = \frac{AP_{ij}^{IWL}}{\sum \forall AP \in Adj[AP_i] AP_j}$
    $send\_workload(AP_{ij}^{WFS})$
end for
$AP_i^{MFS} = min(AP_i^{WFS})$
$minimal\_share = AP_i^{MFS} \times \frac{(100-\alpha)}{100}$
$timer = \frac{1}{minimal\_share}$
Transmission:
if ($timer$ expires AND $free - to - transmit$) then
    send_datapkt();
    end_time= get_time()
end if
if new beacon interval starts then
    $minimal\_share = \frac{minimal\_share \times beacon\_time}{end\_time}$
    send_datapkt()
end if
if minimal_share then
    transmit();
    decr minimal_share;
end if
```

4.3 Transmission

Now, an AP transmits the queued data packets according to the minimal weighted fair share. We use random back-off time to start the transmission of an AP. The AP with maximum traffic load is allowed to transmit first. This is inversely proportional to the traffic load.

4.3.1 Back-off timer

The inverse of the MFS has been used to select back-off time for an AP. This way, the AP with the most traffic is likely to transmit before others. After the back-off timer expires, the AP scans for a 'free-to-transmit' message. On reception of this message the AP starts to transmit the data packets to its clients.

At the end of any transmission round the channel has the 'free-to-transmit' message of the last AP transmitted in this round. This message will be heard by the first AP of the next transmission round.

4.3.2 Determination of End-Time

After completion of the transmission of one AP, it broadcasts 'free-to-transmit' message. Along with that message the AP also transmit the End-Time. The End-Time is defined as

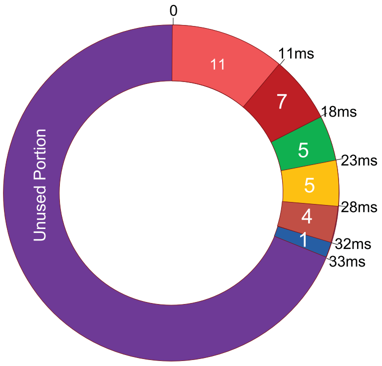


Figure 3: Transmission using the Undershooting of our algorithm

the finishing time of the AP in the current beacon interval. Purpose of the End-Time is for unused-portion allocation.

As we undershoot at the first transmission round there is a good chance of low channel utilization and thus every AP will produce some unused time slots. These unused slots made up the unused portion in a transmission round. The next round is expected to allocate this portion for efficiency.

4.3.3 *Allocation of unused timeslots*

Due to undershooting the beacon intervals will have some 'unused portion' of timeslots. Here, the 'unused-portion' is the set of timeslots that was not used by any APs in the previous beacon interval. To utilize this 'unused-portion' of timeslots we use the 'End-Time' from the last beacon interval. The main concept of the allocation is to consider the last 'free-to-transmit' that was heard before the starting of the current beacon interval. The empty timeslots in between the last 'free-to-transmit' message of the previous beacon interval and the 'start-time' of the current beacon interval represents the unused-portion.

The unused portion is divided among the APs proportionally with respect to the MFS, determined in the previous phase. This method preserves fairness in the network. Because every AP receives the same weighted fraction of their traffic demand.

The execution of our proposed algorithm for Fig.1 generates the following share of nodes, as shown in table 1.

Table 1: Fair Share Selection Model

	A	B	C	D	E	F
A	50	6			44	
B	31	3	38		28	
C	2	27	22		49	
D			39	32	29	
E	19	2		19	17	43
F			28		21	51
selected weight	19	2	27	19	17	43

If there is no change in network, e.g. change in traffic demand of a particular AP, inclusion of new APs and failure or shutting down of the old ones the transmission rounds will go on just the way it did after the unused portion has been allocated. But a real life Wi-Fi network is subject to these changes very often. These dynamic conditions might occur anytime. Dynamic conditions are handled through rescheduling in PowerNap.

4.4 Dynamic Rescheduling

Traffic in a typical Wi-Fi network is an exponential distribution. Three major incidents controls dynamic rescheduling.

- traffic demand fluctuation.
- arrival of a new AP
- departure of an old AP

Traffic demand may change due to client number variation or simply by clients demand change. Rescheduling can be periodic or triggered by interrupt. All the AP's continuously scans the Wi-Fi spectrum for any demand changes and if the upper limit for rescheduling is met we move to the initialization phase again and repeat the undershooting process. When inclusion or failure of APs occur, the rescheduling is interrupt triggered and PowerNap is forced to move to the initialization phase and recompute MFS. So we need to reschedule when there are significant changes in network. We have set a threshold β by which it is determined that after β % change in user data rate PowerNap will be rescheduled.

Suppose we are using the selected weights of table:1 for the transmission phase. First we need to trigger the undershooting mechanism. For undershooting an incremental argument α is used. After the first transmission round with these values we will start scaling the weights.

Let us assume $\alpha = 75\%$. Table:2 is the minimal fair share for the network of Fig. 1.

5. PERFORMANCE EVALUATION

Currently Wi-Fi APs are using 802.11 PSM for power conservation and we can refer to them as legacy systems. This

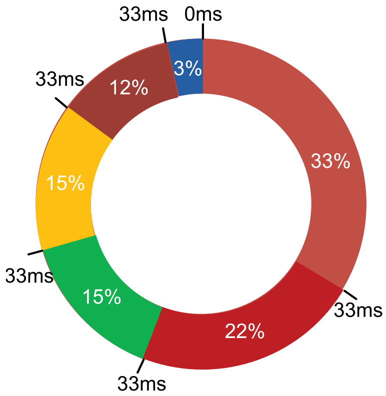
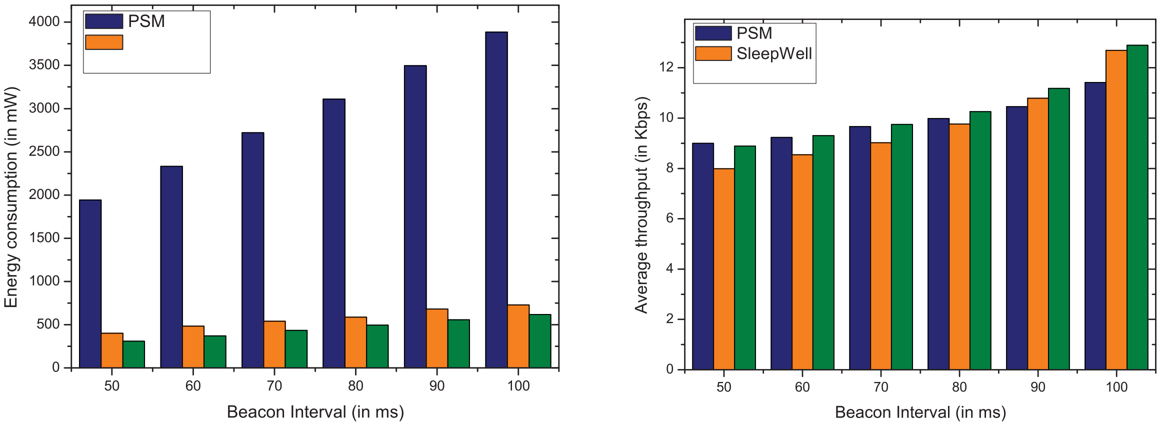
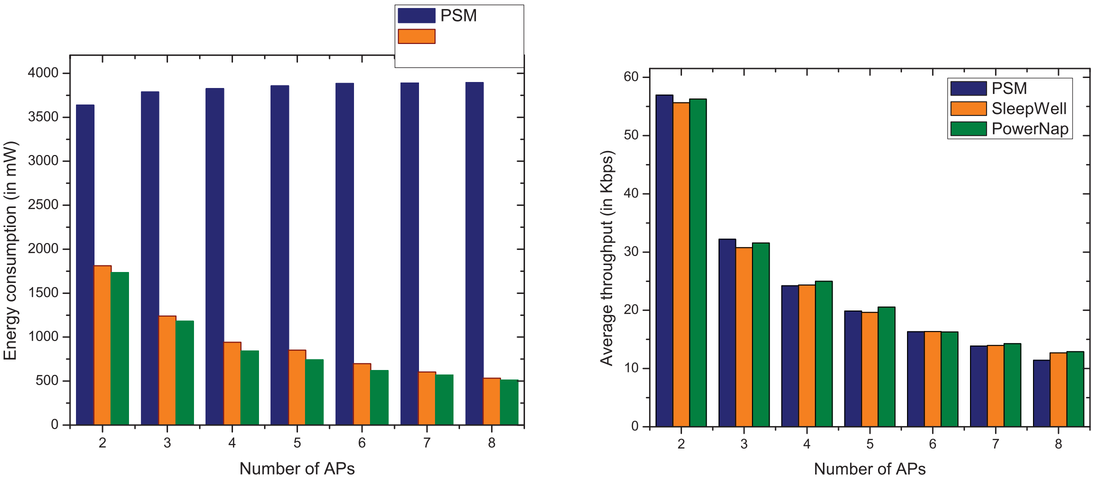


Figure 4: Transmissions in the second round



(a) Power consumption versus beacon intervals (b) Throughput versus beacon intervals

Figure 5: Power consumption and throughput performance comparison for varying beacon intervals.



(a) Power consumption versus number of APs (b) Throughput versus number of APs

Figure 6: Energy consumption and throughput performance comparison for varying number of APs.

Table 2: Minimal Fair Share Model using α% decrement

AP	Minimal Fair Share
A	4.75
B	0.5
C	6.75
D	4.75
E	4.25
F	10.75

section provides an elaborated comparison of PowerNap APs with those of IEEE 802.11 PSM [4] and SleepWell [5]. There we can see the impact of hidden terminals on the energy consumption of client side. In this section, we evaluate the performance of our proposed scheduling algorithm PowerNap in a wireless vicinity such as a University network, where all APs can affect each others transmission. Initially, we use 6 APs to measure the power consumption. We can support a maximum 8 APs to transmit in one beacon interval because of the driver interrupt. We gave every AP random number of users each with dynamic traffic load, to test the robustness of our algorithm.

Table:3 shows the amount of energy consumptions for various modes for 802.11 PSM.

Table 3: Various Power Modes of PSM

Mode	Energy (in mW)
Deep sleep	10
Light sleep	120
Wakeup from deep sleep	250
High power transmission	600
Overhear	400

By customizing the time spent in each mode PowerNap minimizes the energy cost. In transmit mode, the APs transmit data packets and this time depends on the clients bit rates and the workload. The sleep mode consumes the least energy as the entire Wi-Fi radio spectrum runs on low power. Only some internal processing is done during this phase such as timer monitoring and routine tasks. The overhear mode can be considered as a source of energy wastage because even after finishing the high power transmission, it keeps the circuitry up and listens to every packet the associated AP is transmitting. In listening mode, a client only listens for beacon frames and wakes up periodically and receives beacon traffic irrespective of contending APs.

In PowerNap, the AP allocates the unused portion with respect to the minimal fair share, if no dynamic rescheduling is introduced the algorithm is likely to converge in its first few transmission rounds. Fig.4 shows that the convergence occurs at the second transmission round. Since the Internet data communication follows an exponential distribution, dynamic rescheduling will be required.

PowerNap aims to put the clients on listening mode as much as possible rather than the overhear mode. Hence, it saves battery power of the nodes. In PSM, a client goes to overhear mode right after high power transmission and stays that way for a long time expecting there will be a high power transmission again shortly and if it does not receive for a certain amount of time it will go to listening mode. But, in PowerNap, a client will only be awaken during the turn of its associated AP. Moreover, one client will receive one transmission chance in one beacon interval. This makes it easier for a client to go to listen mode right after the high power transmission is finished. The comparison aspects are discussed below.

The performance evaluations are done on C++ programs, where the beacon intervals and the number of APs are varied. The traffic is generated randomly from users under the concerned AP. The power consumption and throughput performances of our proposed PowerNap, SleepWell and IEEE 802.11PSM have been evaluated.

5.1 Comparative Power Saving

From the Fig. 6(a) we see the comparison among 802.11 PSM, SleepWell and PowerNap. With varying number of APs we see our algorithm saves substantial amount of energy. We can also observe when the network is dense, 802.11 PSM and SleepWell consumes more energy whereas our algorithm is more suitable for a dense network. As both PSM and SleepWell will suffer from unnecessary retransmissions because of hidden terminals but PowerNap targets to discover all the APs affecting its transmission hence PowerNap shows better performance.

Changing the beacon interval will lead to increased transmission overhead. But PowerNap does not have an iterative initialization process like PSM or SleepWell, so the transmission overhead in PowerNap is substantially low. Fig. 5(a) shows the average power consumption of smartphones for varying beacon intervals. We also observe that with the increasing beacon intervals the energy consumed by all the approaches is increased linearly. We also observe that the energy consumption is much less in PowerNap compared to SleepWell and IEEE 802.11PSM.

5.2 Network Throughput

With fixed beacon interval (100ms) and varying the number of APs we measure the network throughput of PowerNap, the IEEE 802.11 PSM and SleepWell as shown in Fig. 6(b). The figure shows that our proposed PowerNap can achieve almost equal throughput given by SleepWell and IEEE 802.11 PSM. However, we can achieve immense battery life saving.

Also in case of varying beacon intervals and number of APs is fixed to 8, Fig. 5(b) shows that the throughput is almost same as 802.11 PSM and SleepWell. In both PowerNap and IEEE 802.11PSM, the beacon interval has a significant contribution in network throughput since the increased value of beacon interval produces less overhead and thus the network throughput is increased.

Basically, our algorithm increases the listen time and shorten the overhear time of a user. During listen time an AP checks if the timer is matured and during overhear time an AP can learn about the entire environment. In standard PSM, after waking up every client scans for packets for the entire beacon interval but in PowerNap one user overhears the channel only when the AP associated with it is active for transmission. Similarly, the SleepWell incurs much overhead for repeated migration operations and thus it keeps the users in overhear mode for longer period of time.

6. CONCLUSION

Due to increased popularity of Wi-Fi coverage of Smartphones, the battery power drains rapidly. Wi-Fi usage consumes significant portion of total devices energy. In this paper, we consider a dense Wi-Fi network, where the access points compete for the channel and introduce waiting and retransmissions, wasting the battery of Smartphone. To solve this problem we designed an AP scheduling algorithm in order to decrease energy consumption due to contention among the APs for the channel. We simulated our algorithm in a dense Wi-Fi vicinity with varying number of APs. The evaluation results show that our algorithm saves up to 49-60% energy in varying cases of dynamic rescheduling. Our algorithm is incrementally deployable because no client side changes are required for our algorithm.

Acknowledgment

This research was supported by Next-Generation Computing Development Program through the National Research Foundation of Korea (NRF) funded by the ministry of Education, Science and Technology (2012-0006421). Dr. A. Razzaque and Dr. CS Hong are the corresponding authors of this work.

7. REFERENCES

[1] M. Anand, E. B. Nightingale, and J. Flinn. Self-tuning wireless network power management. *Wirel. Netw.*, 11(4):451–469, July 2005.

[2] F. R. Dogar, P. Steenkiste, and K. Papagiannaki. Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 107–122, New York, NY, USA, 2010. ACM.

[3] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A first look at traffic on smartphones. In *Proceedings of the 10th ACM*

SIGCOMM conference on Internet measurement, IMC '10, pages 281–287, New York, NY, USA, 2010. ACM.

[4] IEEE. Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1–1184, 12 2007.

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

[6] R. Palit, K. Naik, and A. Singh. Impact of packet aggregation on energy consumption in smartphones. In *IWCMC*, pages 589–594, 2011.

[7] G. Perrucci, H. F. Fitzek, and J. Widmer. Survey on energy consumption entities on smartphone platform. In *IEEE 73rd Vehicular Technology Conference: VTC2011 - Spring*, 2011.

[8] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu. Napman: network-assisted power management for wifi devices. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 91–106, New York, NY, USA, 2010. ACM.

[9] E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 160–171. ACM, 2002.

[10] J. Sorber, N. Banerjee, M. D. Corner, and S. Rollins. Turducken: Hierarchical power management for mobile devices. In *In Mobisys*, pages 261–274, 2005.

[11] M. A. Viredaz, L. S. Brakmo, and W. R. Hamburgen. Energy management on handheld devices. *ACM Queue*, pages 44–52, 2003.