The Effect of the 802.11 Power Save Mechanism (PSM) on Energy Efficiency and Performance During System Activity

Markus Tauber, Saleem N. Bhatti
University of St Andrews, UK
{markus,saleem}@cs.st-andrews.ac.uk

***Abstract*—802.11 WLAN is a popular choice for wireless access on a range of ICT devices. A growing concern is the increased energy usage of ICT, for reasons of cost and environmental protection. The Power Save Mode (PSM) in 802.11 deactivates the wireless network interface during periods of inactivity. However, applications increasingly use *push* models, and so devices may be active much of the time. We have investigated the effectiveness of PSM, and considered its impact on performance when a device is active. Rather than concentrate on the NIC, we have taken a system-wide approach, to gauge the impact of the PSM from an application perspective. We experimentally evaluated performance at the packet level and system-wide power usage under various offered loads, controlled by packet size and data rate, on our 802.11n testbed. We have measured the system-wide power consumption corresponding to the individual traffic profiles and have derived application-specific effective energy-usage. We have found that in our scenarios, no significant benefit can be gained from using PSM.**

I. Introduction

Wireless local area networks (WLANs) are increasingly used in home and office environments. Power save mechanisms that exist in the variants of the IEEE 802.11 WLAN standards are expected to save power and hence to use energy efficiently. The impact that such mechanisms may have on performance, and the performance requirements of individual applications are not considered in their function.

Most existing WLAN power saving mechanisms [1], [2] are based on deactivating the WLAN NIC in periods in which there is no traffic. This includes the generic IEEE 802.11 Power Save Mode (PSM), as well as the 802.11n Spatial Multiplexing Power Save (SMPS) mode and Power Save Multi-Poll (PSMP). We argue that this approach has decreasing potential for offering effective power saving capability.

The 802.11 PSM mechanisms rely on the WLAN NIC becoming idle. However, users increasingly desire *always on* connectivity, with *notification* services based on *push* delivery from applications. The popularity of *cloud* strategies throughout the IT landscape suggests that this trend will continue and increase in importance, e.g. [3]–[5]. Also, media streaming applications and the use of multiple applications on devices such as laptops and smartphones means that there may be little opportunity for the NIC to become idle.

A. Research questions

It is quite likely that the NIC will need to stay active to receive incoming notifications. Even if an informed user is able to make energy efficient configuration of their device, system-wide energy saving features, e.g. system sleep modes, are likely to be more effective, as they provide coordinated control for the user device as a whole in a systematic manner, not just an independent device deactivation for the NIC.

Meanwhile, as 802.11 becomes more widely used, manufacturers continue to produce 802.11 chipsets that are increasingly energy efficient, and so energy savings from the NIC alone are reduced. Nevertheless, as 802.11 WLAN is widely used, even small energy savings may be significant when considered multiplied by the number of users on a global scale.

So, we address two specific questions:

1) *What is the potential for use of PSM?*
We wish to assess by *traffic analyses* of popular traces [6] if use of PSM could reduce power usage. We provide analyses of inter-packet arrival times from recent 802.11 WLAN traces. We find that idle times between the packets are such that normal power save mechanisms based on idle time of the NIC are likely to be ineffective.
2) *What power saving is possible for 802.11 during use?*
Motivated by the scenarios identified in 1) we assess in *testbed experiments* what power savings can be observed when using PSM for active devices. Recent work [7], [8] in this area shows that traffic flow characteristics, such as data rate and packet size of the offered load, have a significant impact on effective energy usage. We extend that work by *considering the impact of PSM on energy efficiency and performance and showing the upper and lower bounds of this impact at the application level*.

Our empirical analyses was conducted on our 802.11n testbed, and is based on measurements of system-wide power consumption, as well as throughput and loss at the packet level. We relate these observables to the experimental parameters specified by the data rate and packet size of the offered load, to show the upper and lower bounds of PSM's impact on energy and performance. As 802.11n is available on the 2.4 and the 5GHz ISM band, each with different physical layer characteristics, we test both configurations.

We find no significant effect due to the use of PSM for active devices, but, as observed previously, energy efficiency is affected strongly by application-specific flow characteristics such as packet size and data rate and hence support and extend the findings reported in [7], [8] by considering a typical power save mechanism.

B. Structure of this paper

In Section II we provide a problem definition and in Section III we explain our methodology. We explain our experimental findings in Section IV. In Section V we discuss implications of our results on current systems followed by an overview of related research in Section VI. We provide concluding remarks and an outline of future work in Section VII.

II. Problem Definition

Firstly, we discuss how PSM works and provide answers to our first research question on the power-saving potential of PSM. We use the basic PSM mechanism of 802.11 for our study, as it is the most widely available, and other mechanisms (such as SMPS and PSMP) work on the same principle.

Note that we are concerned with the use of PSM for devices *during activity*. Thus we investigate the potential for PSM to be effective by looking at traces from a busy network. Clearly, in a scenario when devices are mainly idle, existing mechanisms may offer suitable power-saving capability (we hope to examine this in detail in future work).

A. Operation of PSM

The PSM mechanism deactivates the WLAN NIC and periodically activates it to fetch cached data from the access point. This mechanism is triggered by periodic messages – *beacons* – which are transmitted by an 802.11 access point at a constant interval – the *beacon interval (BI)* – as a management mechanism. Every n beacon intervals, information about (potentially) cached data at the access point is also transmitted – the *delivery traffic indication message (DTIM)*. This interval is called the *DTIM interval*, and n is the *DTIM period*. When a station is not transmitting, it only needs to awaken at the DTIM interval to check for incoming data.

However, n is often set to 1 or 2, even though it can be in the range $[1 \ldots 255]$: clearly, there is a trade-off between performance and delivery latency. The station's NIC is, however, typically configured to receive each beacon (a station is informed about the AP's constant beacon interval during the WLAN association process). This means that even when no traffic is received or transmitted, the maximum period a WLAN NIC can remain deactivated is determined by the beacon interval and not the DTIM interval.

So, during normal operation, the *inter-packet arrival time (ipat)* of a traffic flow determines the useful period a WLAN NIC can remain deactivated.

B. Analysis of inter-packet arrival times (ipats)

We consider the default beacon interval of 100ms in the popular WLAN AP software *hostapd*[1] and compare it with ipat distributions extracted from traces of several days of network activity at SIGCOMM 2008 [6].

Figure 1 shows that the majority of the monitored ipat values (80% or more) are shorter than the 100ms beacon interval. In the Figure, we do not consider values greater than 1 second as those are not relevant for the reasons discussed above. However, for rigour, in Table I we provide some information about the distribution of ipat values in the complete data sets (two measurement points identified by the *monitor id*). The data shows ipat values for a range of percentiles and the percentile where ipat values are in the same range as the beacon interval (100ms) as *BI-%ile*.

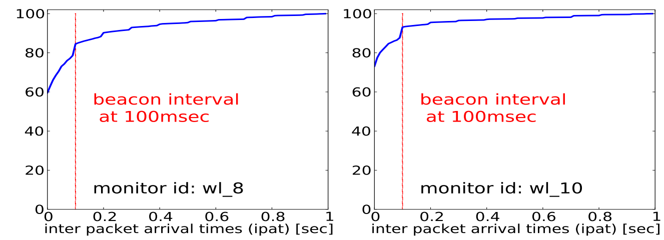

Fig. 1. Cumulative Frequency Distributions of inter-packet arrival time (ipat) from *SIGCOMM08* traces [6], two monitors, wl_8 and wl_10. We find that the majority of ipat values are less than 0.1s (100ms) – the default beacon interval, so PSM has limited opportunity to deactivate the NIC.

TABLE I
IPAT - Distribution Properties (including outliers)

monitor id	BI-%ile	mean	median	95-%ile	99-%ile	max
wl_8	75%	24.10s	<0.01s	2.25s	49.56s	~49h
wl_10	85%	7.37s	<0.01s	0.72s	9.83s	~77h

The datasets had some ipats up to a few days (e.g. due to deactivated devices) but the majority of ipats are less than the default beacon interval.

So, in answer to our first question (Section I-A), PSM does indeed have limited opportunity for energy-saving at the NIC, when nodes are in use, assuming that the *SIGCOMM* trace is representative. We see that the time span between arriving packets at the client (ipat) are mainly smaller than the beacon interval which would, in the absence of heavy traffic, cause the NIC to wake up from its *sleep mode*. However, to support this passive observation we also consider testbed experiments to exam the effects of traffic.

III. Metrics and Approach

Our experiments are based on work described in [7], [8]. However, for this study, we take new measurements using enhanced power-meters and adapting the WLAN NIC kernel module to monitor state changes of the WLAN NIC from *active* to *inactive* to record *sleep times*. We make the following key assumption: *as most users do not have the expertise to fine-tune their equipment, we consider that most deployed systems are used in 'out-of-the-box' configurations, without performance tuning*. So, for our testbed, we chose:

- *Standard WLAN configuration.* We used only standard, un-tuned WLAN setups. While many WLAN NIC drivers and access points (AP) do permit various controls of the hardware, e.g. beacon and DTIM interval, this is not easily accessible for modification by most users.
- *Packet flow behaviour.* To measure application-specific performance (throughput and loss) we use a range of UDP flows specified by packet rate and packet size to represent the upper and lower performance bounds.

[1] http://w1.fi/hostapd/

Due to space constraints, we present a detailed study of results of the 802.11n default configuration at 2.4 and 5GHz only. Note that while our experiments use a specific-experimental testbed and software, the operation of DTIM in IEEE 802.11 is defined as part of the standard. So results with other experimental configurations may produce different overall performance values (e.g. throughput, loss and energy usage), but will have a similar pattern of behaviour and trends if they conform to the standard. Also, we use a single client, as DTIM is controlled via the access point in standard configurations, and we wish to observe the protocol behaviour rather than conduct a performance test of the WLAN cell.

A. Testbed

We have experimentally evaluated energy efficiency and performance in our 802.11n testbed. We generated packet flows of offered loads with various mean bit-rates and packets sizes, and measured system-wide power usage during the packet transmission. Our testbed (Figure 2) consisted of a single client host, a host running a wireless access-point (AP) and experimental control units for monitoring the WLAN environment, providing storage for measurement data, *ntp*[2] services and system configuration. The WLAN hosts were set up in a teaching lab in the University of St Andrews with a distance of $\sim$24$\pm$0.5m between the 2dBi antennas.

Our WLAN card used the popular Atheros[3] chipset. All testbed nodes used Ubuntu 10.04, a minimal server distribution (no desktop service daemons or GUI overhead), with the default kernel 2.6.32-24-generic-pae, and updated WLAN modules compat-wireless-2011-05-02.

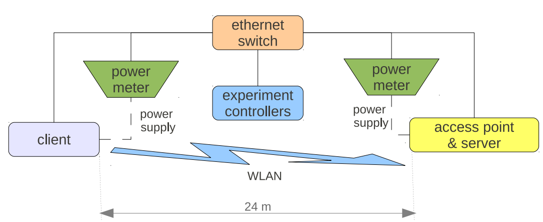

Fig. 2. Schematic of test-bed showing physical connectivity. The testbed was configured separately for PSM on/off with 802.11n (at 2.4 and 5GHz separately, each with 20MHz channels, TX power 17dBm). The experiment controller uses Ethernet for control messages and shared file-system access. Power meter readings are logged by experiment controllers via Ethernet. The separation between the 2 dBi antennas of the client and access point/server is 24$\pm$0.5m. Data packets generated by *iperf* were transferred across the WLAN.

We tested 802.11n at 17dBm (50mW, a typical default, indoor RF power, chosen to avoid measurements being biased by poor RF conditions), and with a 20MHz (default) channel at 2.4 and 5GHz. For this configurations the (modulation coding scheme – MCS – related) channel rates normally range from 52 – 130Mbps. We ran separate experiments with PSM switched on and switched off. This means that all our experimental workloads in Table II are executed twice, once with each of these combinations for each 802.11n configuration.

[2] http://www.ntp.org/
[3] http://www.atheros.com/

B. Experiments

Packet generation and performance measurement for UDP traffic was conducted using *iperf*[4] for which the AP was used as the server. A wrapper script at the client executed *iperf* and extracted throughput and loss for individual UDP flows using *iperf server reports*. The specific packet sizes and bit rates of the UDP workload are given in Table II. Our motivation for using UDP is its popularity for Voice and Video over IP (VoIP and ViIP) applications. Also, UDP allows better control of application-specific offered load compared to TCP, which is modulated by its congestion control behaviour.

TABLE II
GENERIC UDP WORKLOAD.

Packet size	64; 1460 bytes
Bit rate of the offered load	10; 50; 100; 500; 1024 Kbps 5; 10; 50; 100; 150 Mbps

Combining packet size and bit-rate gives 20 configurations; 25 measurements for each gives 500 flows; PSM on/off gives 1000 flows; each flow had a duration of 90 seconds, giving over 25 hours of measurements.

We restricted ourselves to 150Mbps maximum offered load, as few devices or applications use data rates more than 150Mbps (especially low-end devices), and also to avoid bottleneck effects on the PCI bus of our testbed equipment. This scenario will cover many of today's common use cases.

C. Observed variables and metrics

We have measured the observables as described below:

- *Performance:* throughput and loss, as recorded by *iperf*'s server reports on the client, for each UDP flow.
- *Power:* every 0.5 seconds (500ms) we have recorded the power consumption in *Watts* at the AP and client.
- *WLAN NIC sleep times:* state changes of the WLAN NIC, recorded using *syslog(1)* (see Section III-D)

The monitoring intervals for all of the above observables were chosen from preliminary experiments. For power measurements, we used an i-socket power meter[5] for which we found 500ms to be the shortest supported monitoring interval.

For assessing energy efficiency, we define *effective application-specific energy-usage* (E_A) as follows:

$$E_A = \frac{\text{mean power used during transmission of flow}, P_U}{\text{mean throughput of flow}, T_A}$$

E_A has units Joules/Mega-bit (J/Mb):

$$\frac{\text{power in Watts}}{\text{throughput in Mbps}} = \frac{J/s}{Mb/s} = J/Mb$$

A lower value of E_A is better in terms of energy efficiency. To generate values for E_A, for each individual flow, we use the following measurements:

$$E_A = \frac{P_U}{T_A} = \frac{P_A - P_I}{T_A} \tag{1}$$

[4] https://sourceforge.net/projects/iperf/
[5] http://www.i-sockets.com/

- P_A Mean power consumption measured during the transmission of flow [Watts].
- P_I Mean power consumption measured for an idling node [Watts] (was measured to be 65 Watts).
- P_U Mean power used for transmission ($P_A - P_I$).
- T_A Mean throughput measured (using *iperf*) during flow transmission [Mbps].

D. WLAN Kernel Modification

We have modified the *compat-wireless-2011-05-02* kernel module to en-/disable PSM, and to monitor interface *sleep times*. We defined *sleep times* as the time between a state change from *awake* to *sleep* as triggered by the kernel module. The modification of the kernel module was necessary as all of the current Linux distributions we tested used versions of the *ath* kernel module in which a PSM activation did not show any effect, or returned an error when used. (We tested: Open SUSE 11.4, Ubuntu 10.04, Debian 6, Mint 2, BackTrack Linux 5R1, Fedora 15.) PSM was enabled by *hijacking* a TX-power configuration session and calling *ath9k_enable_ps(..)* or *ath9k_disable_ps(..)* in *ath9k_config(..)* in *compat-wireless-2011-05-02/drivers/net/wireless/ath/ath9k/main.c*. Recording of sleep times was achieved by printing timestamps and state changes in function *ath9k_hw_setpower(...)* in *compat-wireless-2011-05-02/drivers/net/wireless/ath/ath9k/hw.c*, and then collected via the *syslog(1)* facility.

IV. Results and Discussion

We can summarise our observations by stating that, in our experiments, PSM had little effect on power usage, or throughput and loss performance. This was true even for the flows with low data rates and large packets, where PSM did have some opportunity to send the NIC to sleep. This was the case with both 802.11n configurations (2.4 and 5GHz). When looking at performance in isolation, however, we see a significant increase of loss at certain date rates if 5GHz is used rather than 2.4GHz.

A. Overview, Data Aggregated over Offered Load

To explain our observations, we show in Figure 3 an overview of throughput, loss, and the power used for operation (P_U), we omit E_A for readability (the wide range of values – see later – appear as outliers, i.e. the distribution of values is skewed). Figure 3 shows the observables aggregated over all application-specific data rates. That means all throughput measurements for all data rates of the offered load are aggregated and their distribution is shown in a boxplot, and compared for PSM-on and PSM-off. The same treatment is applied to the observables for loss and power usage. In all graphs and tables we present our observations individually for 2.4 and 5 GHz.

We see in our aggregation of observables in Figure 3 that packet size has a greater effect on throughput and loss than PSM has. This is supported by the statistical analysis shown in Table III. Table III shows the mean and the standard deviation ($\mu \pm \sigma$) of the distributions of the aggregated observables. Figure 3 shows outliers and supports results as suggested by ($\mu \pm \sigma$) in Table III for loss with 64B packets, both with PSM on and PSM off. We see a higher degree of variation between PSM on and off at 2.4GHz: we believe this is due to the different physical characteristics and environment, but have not explored this in detail.

TABLE III
Distribution Properties of Main Observables

802.11n at 2.4GHz	64B packets		
		($\mu \pm \sigma$) PSM off	($\mu \pm \sigma$) PSM on
	throughput [Mbps]	4.22 ± 4.05	3.90 ± 3.78
	loss [%]	16.05 ± 20.00	15.38 ± 19.36
	P_U [W]	26.67 ± 14.58	32.01 ± 15.16
	1460B packets		
		($\mu \pm \sigma$) PSM off	($\mu \pm \sigma$) PSM on
	throughput [Mbps]	21.45 ± 29.63	20.29 ± 27.79
	loss [%]	1.30 ± 2.61	1.11 ± 2.18
	P_U [W]	19.77 ± 13.72	20.92 ± 14.95
802.11n at 5GHz	64B packets		
		($\mu \pm \sigma$) PSM off	($\mu \pm \sigma$) PSM on
	throughput [Mbps]	3.23 ± 2.98	3.32 ± 3.05
	loss [%]	24.42 ± 27.51	20.97 ± 26.09
	P_U from (1) [W]	28.30 ± 14.51	34.76 ± 15.19
	1460B packets		
		($\mu \pm \sigma$) PSM off	($\mu \pm \sigma$) PSM on
	throughput [Mbps]	20.79 ± 28.70	19.47 ± 26.84
	loss [%]	3.33 ± 5.97	2.61 ± 4.29
	P_U from (1) [W]	20.60 ± 13.71	22.74 ± 15.39

Most distributions are skewed. PSM had little impact.

B. Detailed Analysis, Progressions over Offered Load

To illustrate the effects of all flow characteristics on all observables we present Figures 4–8, in which we have: (i) offered load on the horizontal axis (Table II); (ii) used standard error bars for the raw observable under consideration, but in some cases, the error bars may be too small to be visible; (iii) show the differences of the observables (Δ observable) over the offered load due to PSM.

In Figure 4 we see that, as expected, the sleep times depend on the traffic patterns – when PSM was switched off, obviously, no sleep times are recorded. We see that due to the ipat value with large packets, at low offered load, PSM has time to send the interface into sleep mode. As the data rate increases, the ipat value decreases and leaves less opportunity for PSM to operate. This effect is amplified with a decreasing packet size due to the reduced ipat values. Thus, Figure 4 shows the upper and lower bounds of expected interface sleep times in 802.11n with PSM switched on. Figure 4 also shows that PSM has decreasing opportunity to operate at the high offered load rates as the NIC is busy.

As we see an impact on sleep times due to data rate and packet size, we have made a comparative analysis of the PSM on/off modes depending on the flow characteristics which we will call a *delta (Δ)* analysis. Figures 5–6 shows: (i) Δ throughput which was computed as the normalised relation of $throughput_{PSMon}/throughput_{PSMoff}$; (ii) loss is already a normalised value, so we have computed Δ loss as the difference of $loss_{PSMon} - loss_{PSMoff}$. For the difference in energy usage we plot (iii) ΔE_A derived like Δ throughput.

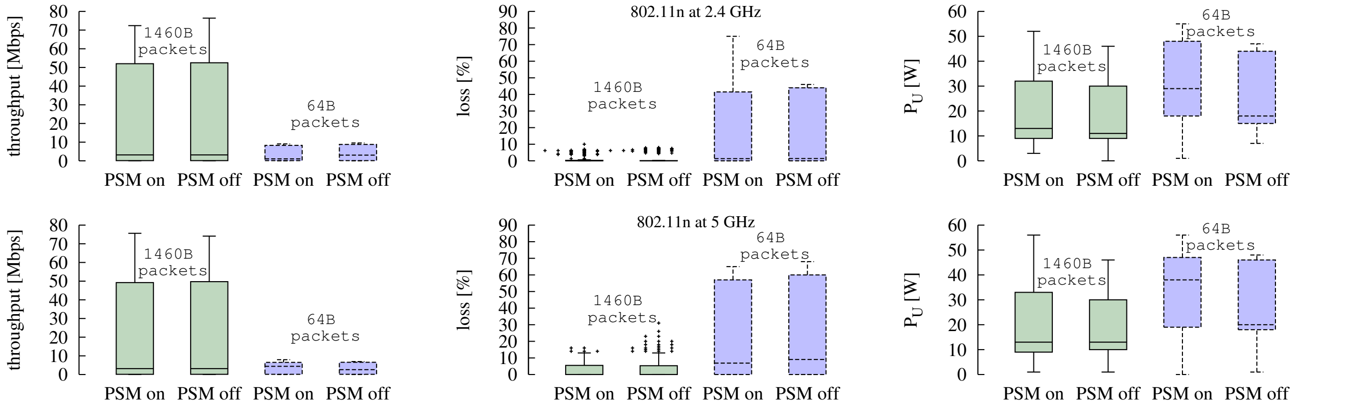

Fig. 3. 802.11n, 20 MHz Channel – aggregated throughput, loss and P_U measurements, with and without PSM, 2.4GHz at the top 5GHz at the bottom row.

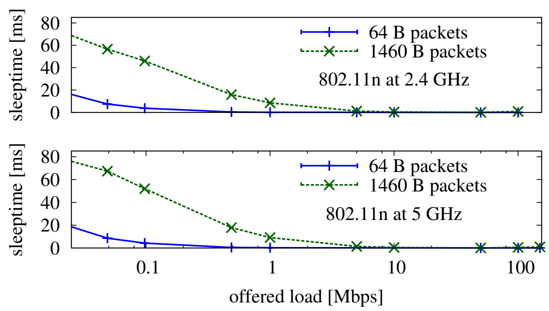

Fig. 4. Sleep times of WLAN NIC with PSM in IEEE 802.11n at 2.4GHz (top) and 5GHz (bottom). Each point is the mean period of WLAN NIC inactivity/sleeptime at a specific workload configuration. We see decreasing opportunity for de-activating the WLAN NIC, due to inactivity, with increasing data rate and decreasing packet size.

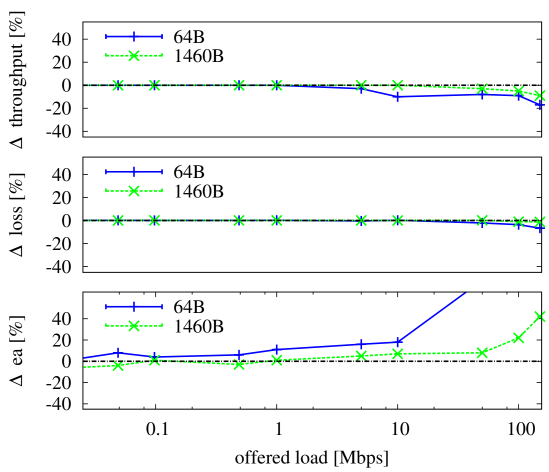

Fig. 5. Differences in the throughput and loss with PSM on/off, at 2.4GHz. Horizontal zero line is a visual aid. +ve values show where PSM on gave higher values.

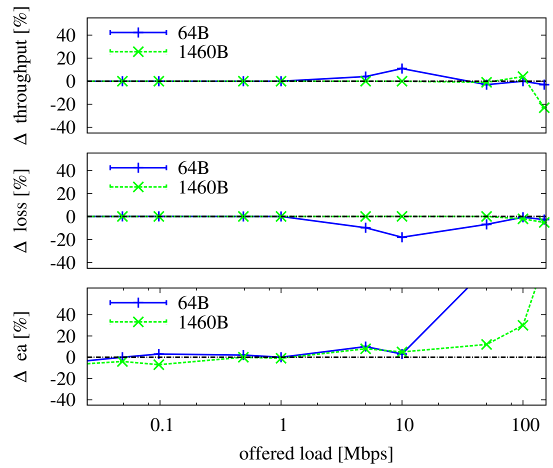

Fig. 6. Differences in the throughput and loss with PSM on/off, at 5GHz. Horizontal zero line is a visual aid. +ve values show where PSM on gave higher values.

The performance metrics show only minor differences, despite large sleeptimes at low data rates (Figure 4). E_A appears to be very different at high data rates. However, the absolute values of E_A decrease by an order of magnitude with increasing data rate of the offered load, as shown in Figure 7–8 in which we plot raw progressions of E_A, throughput and loss. The left columns show results measured with PSM on, and the right column is measured with PSM off. Figure 7–8 for E_A show clearly the *effective energy usage envelopes*, the region between the lines plotted for small packets and large packets. Real applications will operate within the plotted envelopes. Using such a plot to compare performance and E_A helps to identify operational regions in which applications use energy more efficiently due to an amortisation effect of the system-wide energy usage. As discussed previously [7], [8], applications at low data rates (e.g. Voice-over-IP) would be more energy efficient if they change their flow characteristics when the network conditions permit. For instance Skype operates typically below 100Kbps and uses small packets to compensate for loss. During periods in which no/low loss occurs, a higher data rate or packet size (e.g. perhaps due to use of a different codec), could lead to a more efficient energy usage. A similar argument applies to streaming applications like YouTube, which operate between 0.5 and 1 Mbps. Figure 7 and 8 show that this assumption also holds when considering power save mechanisms like PSM.

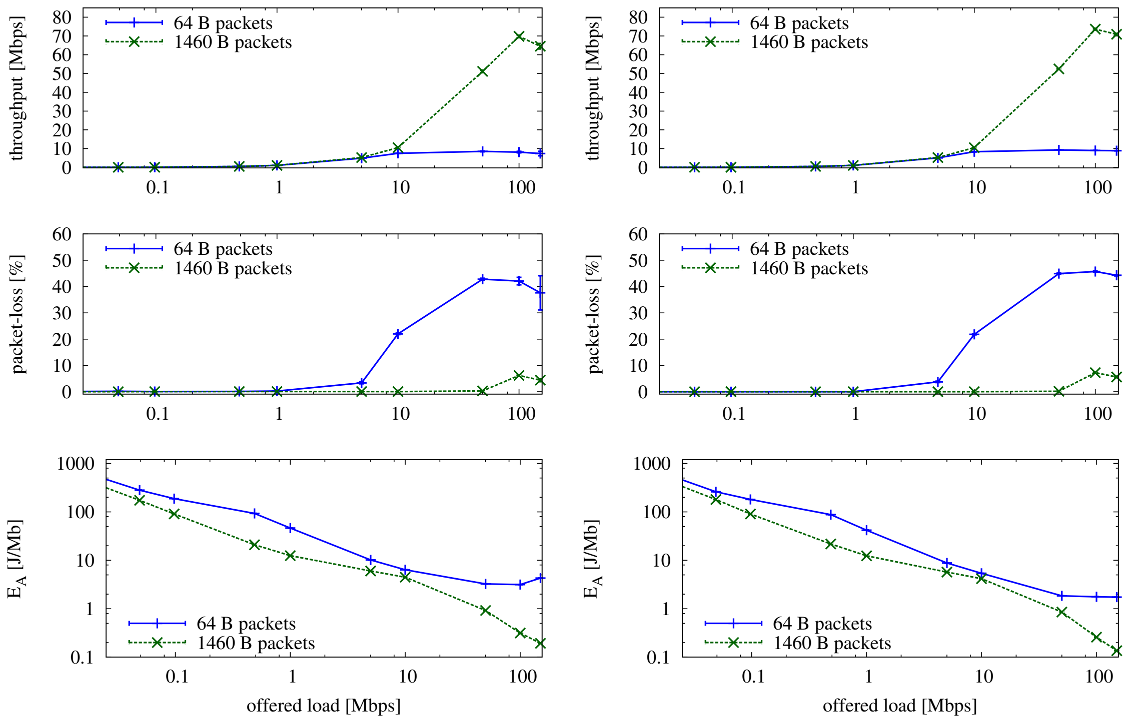

Fig. 7. IEEE 802.11n with 20 MHz Channel and 17dBm TX-power at 2.4GHz, with PSM switched on (left column) and switched off (right column). Each point represents 25 measurements with the same data rate of the offered load and packet size – each UDP flow is of 90s duration.

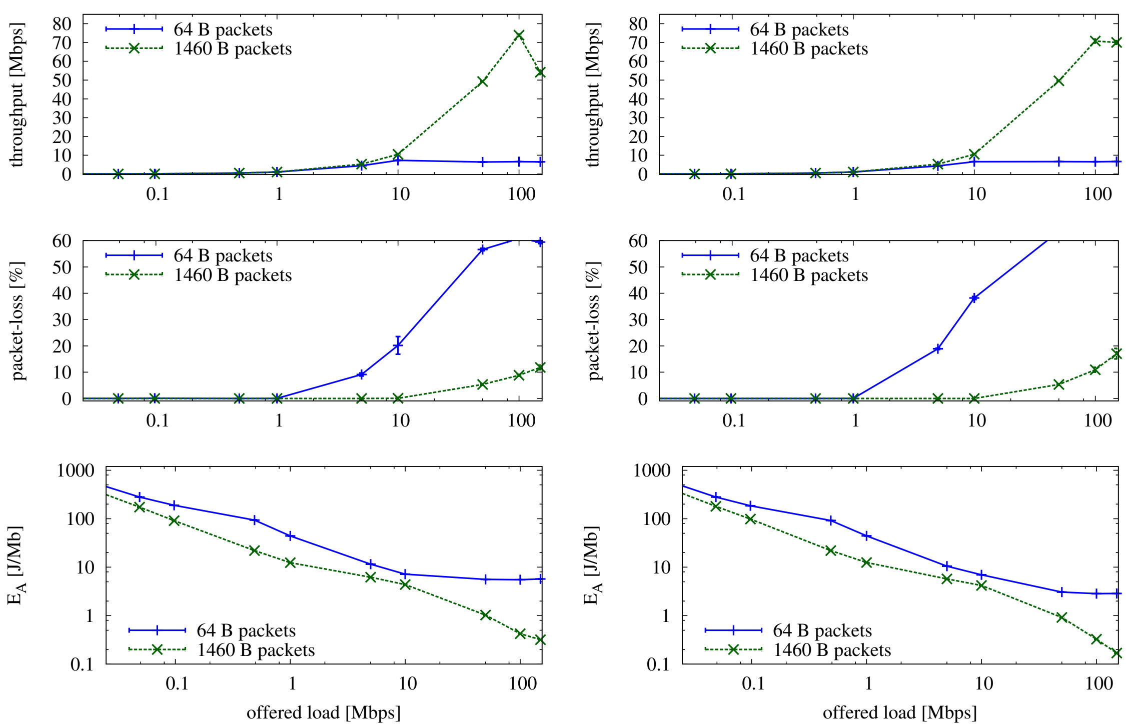

Fig. 8. IEEE 802.11n with 20 MHz Channel and 17dBm TX-power at 5GHz, with PSM switched on (left column) and switched off (right column). Each point represents 25 measurements with the same data rate of the offered load and packet size – each UDP flow is of 90s duration.

V. ANALYSES – MODIFYING DTIM

In our discussion so far it seems clear that during operation (when a flow is being transmitted), PSM is ineffective. The cause is that the DTIM interval, which causes the client to come out of sleep, is too low, based on the traffic patterns observed. Recall that the DTIM is a multiplier for the beacon interval which normally has units milliseconds (ms).

A. Current DTIM values

In a random selection of some equipment from popular WLAN equipment vendors, we list some default settings of beacon interval and DTIM values in Table IV. We found that, the beacon interval was always 100ms, the most common DTIM value was 1, and we found no instance of a DTIM value higher than 3. We have not undertaken a comprehensive survey, but the pattern is clear – default DTIM values are low.

TABLE IV
A RANDOM SELECTION OF BEACON AND DTIM VALUES FROM VARIOUS POPULAR ACCESS POINT CONFIGURATIONS.

Vendors	beacon interval [ms]	DTIM value
Linksys[a], DLINK[b], DD-WRT[c]	100	1
Cisco[d]	100	2
Netgear[e]	100	3

[a] http://ui.linksys.com/files/WAG300N/1.01.01/help/h_AdvWSettings.htm
[b] http://www.dd-wrt.com/wiki/index.php/Advanced_wireless_settings
[c] http://support.dlink.com/emulators/dwl7100ap/html/help_adv.html
[d] https://supportforums.cisco.com/docs/DOC-4833
[e] http://www.downloads.netgear.com/files/WG103_RM_27Feb09.pdf
All URLs accessed on 06 July 2012.

B. DTIM values likely to remain low

It is possible to change the DTIM value in all but the simplest (or perhaps older) equipment. We have not surveyed the actual DTIM values used in deployed systems, so it is not possible to comment upon how widely the default values, such as those listed above, are used. However, the values of 100ms/1 for beacon/DTIM are very popular in domestic equipment, and we moot that very few domestic users would change them.

Even if the DTIM value is changed, we take the position that DTIM values are unlikely to be high. The reason for this is the purpose of the DTIM. The DTIM, effectively, enables coordination between AP and client systems so that multicast frames and broadcast frames can be delivered in a timely manner. The use of IP multicast is increasing, especially for control channel protocols. For example, popular discovery and configuration protocols, such as UPnP[6], Bonjour[7], ZeroConf[8] and mDNS[9] all use multicast extensively. If DTIM is increased, then so is the latency for the operation of such protocols over WLAN. If use of such protocols continue with IP multicast, then it is unlikely that DTIM will be increased significantly, as it will negatively impact performance.

At the moment, the DTIM value is notified to the client by the AP when it associates with the cell, and is the same for all clients. One possibility for a solution is for client systems to have customised, i.e. client-specific, DTIM values, so that individual clients can make appropriate trade-offs between power-saving and latency for multicast. However, a per-client DTIM would need APs to be modified, as well as client systems. Additionally, even if such a mechanism was explored and found to be useful, it is the subject of a patent [9], so it is not clear how widely usable such a mechanism would be.

C. Future use of DTIM and PSM

It should be noted that our discussion is focussed on the client system when it is in use. We have assumed that client systems may, in today's usage patterns, be active much of the time. However, client systems can, of course, reduce energy usage when they are not active. The PSM mechanism is specific to WLAN. Unless the client system as a whole can also enter a sleep mode, the energy savings may be reduced. However, while the issue with the DTIM that we have highlighted in this paper remains, then the effectiveness of PSM will be limited.

In our experiments here, and in previous work [7], [8], it is clear that taking a system-wide approach, and amortising the energy usage for the system as a whole across the data transfer, results in more effective energy usage (lower values of E_A). This would require applications to be adaptive in their flow construction, changing packet-size and (packet) transmission rate under some application-specific adaptation policy.

Adapting packet size and (packet) transmission rate can be controlled from the application. As software upgrades may be possible to some (but probably not all) existing equipment, this may allow wider deployment of solutions. Additionally, if software upgrades are possible to legacy equipment (equipment which does not have any power saving mechanisms), then those deployments might also benefit from the application adaptation approach.

VI. RELATED WORK

As far as we are aware, this is the first combined empirical evaluation of the effects of PSM on power usage, energy efficiency and performance in 802.11n.

The work extends [7], [8], in which the authors compared energy efficiency in several variants of IEEE 802.11. They provided an initial evaluation of the *effective energy usage envelope* providing the basic model for this study. The new contribution in this paper is to consider the effectiveness of PSM, whose function is representative of the majority of power save mechanisms which aim to save power by putting the interface to sleep during periods of inactivity.

The authors of [10] measure the effects of system-wide *power consumption* of PSM focusing only on power savings in the legacy 802.11b standard. Compared to our work, limited workload options were used. Ideal beacon intervals of 100 – 200 ms were proposed, as well as the suggestion of a DTIM period of 3. They conclude that PSM results in only insignificant power reduction if background traffic is observed.

In [11] the authors analyse the trade-off between delay and power consumption when using PSM via an analytic model by

[6]http://www.upnp.org/
[7]https://developer.apple.com/bonjour/
[8]http://www.zeroconf.org/
[9]http://www.multicastdns.org/

adapting the sleep interval. This is in contrast to our approach as we apply an empirical analysis in our testbed, and also consider the relevance of other performance metrics (loss and throughput).

None of the related work items below consider PSM, but they do contain other relevant work.

Halperin *et al* [12] focus in their analysis on the link layer of IEEE 802.11n and conclude that transmission with higher bit rates and larger packets is more energy efficient than with lower bit rates and smaller packets. In contrast to our measurements, they measured the power consumption directly at the NIC and did not consider system-wide effects or application-level flows.

In [13], [14] the authors present an analysis of link layer measurements of 2.4GHz IEEE 802.11b equipment. They focus on power (and also derive effective energy as J/b) resulting from a range of transmission power settings, transmission rate and packet size. They also conclude that large packets use energy more efficiently than small ones. Again, the authors focus on the NIC's power usage only.

Kuo [15] reports on optimisations of the MAC and PHY layer in order to improve energy usage. The author focuses on the design of an analytic framework for testing adaptations of parameters of the Distributed Coordination Function (DCF). The effects on energy usage of ranges of values for individual parameters are tested in a simulation with DCF's basic mode and its RTS/CTS mode. The effects on application-specific performance are not analysed. The author concludes that large packets and higher data rates are more energy efficient than smaller packets and lower data rates.

Suong *et al* [16] introduce a model to analyse the effects of varying packets sizes on collisions. They conclude that a combination of few very large packets and a lot of small packets will result in an increased probability of collision. They correlate this to energy usage simply by defining all collisions as wasted energy, and hence the probability for collisions is used as an energy efficiency metric.

VII. Conclusions and Future Work

We find that PSM is largely ineffective in our testbed experiments. An analysis of WLAN traces in Section II shows that short inter-packet arrival times for typical 802.11 traffic mean that PSM has little opportunity to be effective. In our experiments we see that despite PSM operating as defined, sending the interface to sleep when low data rates are used, the opportunity for it to operate decreases as the offered load increases. So, there may be more opportunity for PSM to operate at low data rates. However, our observations in our experiments show that for the range of offered loads and traffic patterns that we have tested, PSM is observed to be ineffective.

We have evaluated the effectiveness of PSM in 802.11n by considering application-level throughput and loss with measurements of system wide power usage. We have calculated energy efficiency by testing a range of application-level data rates with large and small packets to evaluate performance and energy efficiency envelopes. Our measurements were for an 802.11n testbed under normal office conditions.

As 802.11n allows higher data rate channels, such as 270Mbps, then for a given offered load, such as 10Mbps, the NIC will spend less time transmitting, compared to a lower data rate channel, such as 54Mbps, so higher channel rates may offer greater opportunity for PSM to operate – an investigation for future work. Future work would include testing with different client systems, testing of other 802.11 variants as well as other power save mechanisms. We speculate that it is likely that other mechanisms based on the same principle as PSM will suffer from the same ineffectiveness.

Acknowledgements

This work has been supported mainly by the EPSRC-funded project IU-ATC (http://www.iu-atc.com/).

References

[1] "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 Amendment 5: Enhancements for Higher Throughput," *IEEE Std 802.11n-2009*, pp. c1 –502, 29 2009.

[2] J. Berg, "The 802.11 subsystems for kernel developers," accessed in 2012 from http://wireless.kernel.org/80211books/powersave.html.

[3] IDC Corporate USA, "IDC Cloud Research," accessed 2012 at http://www.idc.com/prodserv/idc_cloud.jsp.

[4] Jeff Barr, "Amazon S3 - 566 Billion Objects, 370,000 Requests/Second, and Hiring!" Oct 2011, http://tinyurl.com/amazone-s3-report-2011.

[5] D. Harris, "Its cloud prediction time: IDC, Gartner (and I) weigh in," Dec 2011, http://gigaom.com/cloud/its-cloud-prediction-time-idc-gartner-and-i-weigh-in/.

[6] A. Schulman, D. Levin, and N. Spring, "CRAWDAD data set umd/sigcomm2008 (v. 2009-03-02)," Downloaded from http://crawdad.cs.dartmouth.edu/umd/sigcomm2008, Mar 2009.

[7] M. Tauber, S. N. Bhatti, and Y. Yu, "Application Level Energy and Performance Measurements in a Wireless LAN," in *Proc. GreenCom2011: IEEE/ACM Intl. Conf. on Green Computing and Comms*, Aug 2011.

[8] M. Tauber and S. N. Bhatti and Y. Yu, "Towards Energy-Awareness in Managing Wireless LAN Applications," in *Proc. NOMS 2012: IEEE/IFIP Network Operations and Management Symposium*, Apr 2012.

[9] J. Wang and T. Nagy, "Maintaining delivery traffic indication message (DTIM) periods on a per wireless client device basis," Aug 2011, US patent number: 8005032. [Online]. Available: http://www.google.com/patents/US8005032

[10] Y. He, R. Yuan, X. Ma, and J. Li, "The IEEE 802.11 Power Saving Mechanism: An Experimental Study," in *Proc. WCNC 2008 - 2008 IEEE Wireless Comms. and Networking Conf.*, Apr 2008, pp. 1362–1367.

[11] S. Baek and B. D. Choi, "Performance analysis of power save mode in IEEE 802.11 infrastructure WLAN," in *Telecommunications, 2008. ICT 2008. International Conference on*, June 2008.

[12] D. Halperin, B. Greensteiny, A. Shethy, and D. Wetherall, "Demystifying 802.11n power consumption," in *Proc. 2010 Intl. Conf. on Power Aware Computing and Systems*, ser. HotPower'10, Berkeley, CA, USA, 2010.

[13] J.-P. Ebert, S. Aier, G. Kofahl, A. Becker, B. Burns, and A. Wolisz, "Measurement and Simulation of the Energy Consumption of a WLAN Interface," Telecommunication Networks Group, Technische Universitt Berlin, Berlin, Germany, Tech. Rep. TKN-02-010, June 2002.

[14] J.-P. Ebert, B. Burns, , A. Wolisz, and A. Wolisz, "A trace-based approach for determining the energy consumption of a wlan network interface," in *Proc. European Wireless 2002*, 2002, pp. 230–236.

[15] W.-K. Kuo, "Energy efficiency modelling for IEEE 802.11a distribution coordination function system without finite retry limits," *Communications, IET*, vol. 1, no. 2, pp. 165 –172, 2007.

[16] S. H. Nguyen, H. L. Vu, and L. L. H. Andrew, "Packet Size Variability Affects Collisions and Energy Efficiency in WLANs," in *WCNC 2010 IEEE Wireless Communications and Networking Conf.*, 2010, pp. 1–6.