Übersleep: An innovative mechanism to save energy in IEEE 802.11 based WLANs

Sudarshan S, Rajan Prasad, Abhishek Kumar, Rahul Bhatia, Bheemarjuna Reddy Tamma
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
Email:{cs10b036, cs10b031, cs10b002, cs10b030, tbr}@iith.ac.in

***Abstract*—Perpetually rising energy costs are forcing research communities to focus their efforts on a reduction of the global CO_2 footprint, and since Information and Communication Technologies (ICT) account for a significant percentage of the overall global energy consumption, energy efficiency is becoming increasingly important in the operation of ICT infrastructure, especially in enterprise and data center networks. Simultaneously, the proliferation of devices such as smartphones and tablets, which have to use battery powered wireless radio adapters, and whose battery life is dependent on the power consumption of the radio adapters used for wireless communication indicates that efficient radio power saving strategies are needed to increase the battery life of such devices. This paper extends an earlier proposal by [1], and examines the performance of the same on IEEE 802.11n networks. It also extrapolates the results to estimate performance on 802.11ac WLANs, the planned successor to 802.11n WLAN in 5 GHz unlicensed band. The results indicate that this approach can be used to save over 59% of the RX energy (*i.e.,* energy consumed by Wi-Fi radio while operating in receive mode) without any loss whatsoever in throughput unlike 802.11 power save mode and that this approach is completely backward compatible.**

Keywords*—Energy efficiency; 802.11; WLANs; Power Saving; Green Communication Protocols

I. INTRODUCTION

Energy efficiency has become crucial for all industries as a way to cut down on recurring costs. According to [2], the annual electricity consumed by networking devices in the U.S. is 6.06 Terra Watt hours, thereby presenting a strong case for reducing the energy consumed by networking devices.

Today, since IEEE 802.11 based Wi-Fi radios are being shipped with video game consoles, smart phones, tablets, printers, cameras, peripheral devices, and all laptop computers, and since the amount of power consumed by 802.11 radios (Wi-Fi cards) is a significant component of total power consumed in these wireless devices [3], with Wi-Fi NIC power consumption well over one watt in some scenarios [4], reducing the energy consumed by IEEE 802.11 radios is an increasingly important problem to solve. Implementing mechanisms for power saving in all shipped devices can lead to significant decrease in power consumption, not to mention significantly longer battery life on portable and hand held devices. While there are mechanisms for power saving built into the 802.11 standards itself, these affect throughput significantly and can lead to loss of packets, eventually even leading to increased power consumption [5]. Thus the need for a new power saving mechanism that saves power without leading to reduced performance of networks.

Numerous power saving techniques using rate adaptation, (timed/non timed) sleeping, dynamic adjustments of the RF chains or a combination of the aforementioned techniques have been suggested in the literature. However, most of them save power only for a subset of the categories of network usage, e.g. [6] saves power during VoIP sessions using “Adaptive Multi sleep” and “Non-Adaptive Multi Sleep” and [7] saves power for short TCP file downloads and web browsing using so-called “Opportunistic Power Save Mode”. These techniques have been developed keeping in mind a particular kind of network traffic and may not generalise to today’s heterogeneous traffic including parallel VoIP sessions and TCP traffic. Some other techniques such as [8] and [9] attempt to trade off network performance in terms of achievable throughput and latency for energy saving. The proposed approach, on the other hand, seeks to save energy without affecting the throughput irrespective of the kind of network usage.

It is to be noted that in this paper, the concepts STA, node and client are used interchangeably to refer to a wireless 802.11 adapter, of which a device may contain several.

II. 802.11N AND 802.11AC

The recent IEEE 802.11n standard is able to offer wireless bitrates of up to 600 Mbps [4], taking advantage of the spectral efficiency of the OFDM introduced in 802.11g and extending it using multiple RF chains to create multiple spatial streams, increasing the amount of information sent by sending multiple spatial streams at the same time, but unfortunately, keeping on multiple RF chains leads to increased power consumption [4]. This further increases the strain on the batteries of devices using this new standard, leading to an urgent need to save power since 802.11ac, the proposed successor to 802.11n, allows up to 8 spatial streams.

Another enhancement introduced in 802.11n (and further enhanced for 802.11ac) is the concept of frame aggregation, which wraps several frames destined to the same destination into a single large frame and sends it in a single burst thus reducing and/or removing the overhead created by the 802.11 DCF MAC protocol which is contention based and leads to collisions, that further impact the data rates achievable, even if the PHY data rate were infinite [10]. This frame aggregation can either be of Ethernet frames from the higher layers (A-MSDU) or can be of 802.11 frames created at this layer (A-MPDU). We illustrate how this very idea can be used to further improve the power saving from approaches like [1] in this paper.

A. 802.11 Power Save Mode

Realising the need for saving power in 802.11-based nodes, many of which run on batteries, the creators of the standard added Power Save Mode (PSM) into the standard. However the original PSM, which relied on quasi-periodic announcements of downlink traffic from the Access Point (AP) immediately followed by PS-Poll packets by all PS-enabled nodes, which contended via the usual Distributed Coordination Function (DCF) plus back-off, had issues with scalability and loss of packets to nodes under moderate to heavy traffic load which when accounted for could even increase overall power consumption. This lead to the introduction of newer and more advanced modes of power saving in 802.11 standards [11], which are discussed below.

The 802.11n introduces a new MIMO specific PSM, called Spatial Multiplexing Power Save (SMPS). This mode of operation allows an 802.11n node to power down all but one of its receive chains, leading to reduction in power consumed, possibly at the expense of reduced throughput, with two sub modes of operation as indicated below.

1) *Static SMPS:* This mode has the node function like a legacy node without MIMO, as once the AP is sent a notification that the node is in static SMPS, it sends only single data stream to the node, which now powers off all but one of its receive chains, saving power due to reduction in the number of receive chains active.
2) *Dynamic SMPS:* This mode is similar to static SM power save except that the node rapidly turns on extra receive chains when it realises that there is a frame about to be sent to it, so that by the time the frame transmission starts, it can listen to all spatial streams.

In addition to the aforementioned modes, 802.11n also includes enhanced versions of the standard 802.11 PSM, called as Power Save Multi Poll (PSMP) which too is of two types:

1) *Unscheduled PSMP:* Unscheduled PSMP is the closest to legacy PSM in 802.11 standard. It extends the earlier proposal, Automatic Power Save Delivery (APSD), and allows a node(s) to inform an AP that frames of some specified QoS levels should be buffered, while frames sent by the node at another set of specified QoS levels are to be considered triggers that will cause the delivery of buffered frames.

Frame control	Duration	Receiver MAC	Rest of the packet
2 bytes	2 bytes	6 bytes	

Fig. 1. The MAC headers of IEEE 802.11 frames mean that a station needs only to read the first 10 bytes to realise that the frame is not intended for it, at which time it would know the duration of the frame as well

Reserved	MPDU length	CRC	Delimiter signature	Frame control	Duration	Receiver MAC	Rest of the packet
4 bits	12 bits	1 byte	1 byte	2 bytes	2 bytes	6 bytes	

Fig. 2. If the incoming frame is an A-MPDU, we need to read 14 bytes for detecting whether the frame is intended for receipt by the current STA due to the additional overhead created by A-MPDU headers

2) *Scheduled PSMP:* Scheduled PSMP is a new addition in 802.11n which allows APs to send schedules to nodes so that they can know when they are going to get downlink traffic from the AP and sleep in inter-traffic intervals.

Though the built-in PSM is indeed capable of, and does save, power, it has certain issues. For example, the spatial multiplexing PSMs can lead to use of a reduced number of spatial streams leading to reduced performance, and the APSD mechanism can lead to frames being buffered for longer than necessary at the AP, which may even lead to buffer overflow and hence packet loss at the AP's packet buffers. In addition, under heavy load, these approaches can cause heavy packet loss, leading to reduced throughput. They also require support at both the AP and the end user nodes to be effective. In fact, as observed by [5], they can even lead to increased power usage at the nodes when these effects are accounted for.

III. PROPOSED APPROACH

We propose here an extension to a method described earlier in [1], which leads to improvements in power usage, especially when the node running the algorithm is relatively inactive in an active network - an all too common scenario in today's world where low power devices such as smartphones connect to the same networks to which laptops and netbooks connect - without affecting the performance of the network adapter, with excellent energy saving (in fact, with improved energy saving) under heavy traffic/large number of active nodes in the WLAN system. In addition, this technique can lead to power saving even if implemented only at one end, without needing any modification at the other end. This means that smartphones and/or tablets implementing this approach will save power even on legacy 802.11 WLANs.

An overview of the proposed approach now follows, succeeded by an explanation of the reference implementation, performance evaluation and then the conclusions.

A. Overview

802.11n radios consume significantly less power when in idle mode as compared to when in receive mode, as analysed by [4]. This means that keeping the radios in IDLE

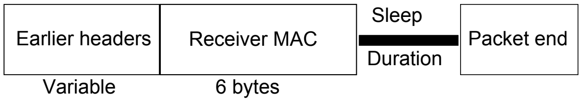

Fig. 3. Übersleep visually demonstrated

mode rather than in RX mode can lead to significant savings of power. The default Wi-Fi mode, CAM (Constantly Awake Mode) leads to power penalties every time there is traffic on the channel, since all nodes (STAs) that sense the channel as busy waste time decoding the packet, only to discard it after wasting further more energy checking the FCS at the end of the frame. Similar to [1], our approach is based upon turning the radio off the instant we realise that the incoming A-MPDU/frame is not meant for the current STA as indicated in Figure 3, something which can be decided after reading the first 10 bytes alone as illustrated in Figure 1 (and for A-MPDUs the first 14 bytes alone as illustrated in figure 2) as opposed to reading the entire frame. We do not implement the alternate frame format proposed therein, however, for reasons that are discussed later in this paper. We believe this is possible to implement on existing commodity hardware, since experiments using a modified version of the ath9k driver illustrated that an Atheros card can be woken up after being given the command to enter the "network sleep" mode (which is essentially a partial sleep of the MAC and PHY layer) in around 50μs.

B. Taking advantage of 802.11n/ac frame aggregation

Frame aggregation mechanism introduced in the 802.11n allows a STA to aggregate together multiple packets destined to the same receiver and send them in one frame burst. Übersleep can take great advantage of this since the aggregates usually occupy the channel for a significant amount of time, allowing all other STAs (to whom this frame is irrelevant) to sleep for a relatively large amount of time. The newly proposed 802.11ac standard further extends the limit on aggregation, and can aggregate frames up to 1 MB in size, thus allowing a STA to enter network sleep for a large amount of time despite the higher data rates provided by this standard. Thus, Übersleep is easily extensible to 802.11ac and possibly to standards to be released in the future, since those upcoming standards are likely to add even more aggregation.

C. Analysis of effect on throughput

This approach is based on STAs turning off their receiver the instant they realise the incoming packet is not intended for them and turning it back on at the expected end duration of the transmission. This approach is reasonable, since the CSMA/CA used in 802.11 means that nodes will not stop transmitting even when there is a collision. However, as [1] describes, there are situations where this approach can lead to packet loss, mainly since the usual mechanism of checking the FCS and discarding corrupt packets is bypassed when the node decides to sleep on a packet, leading to the possibility of corrupt headers being trusted. These cases are examined in this section.

It is to be noted that if the receiver MAC address is corrupted leading to a packet not meant for this node being classified as intended for it then the packet will be decoded and discarded after checking the FCS. The opposite does not lead to packet loss too, since if a packet intended for this node has a corrupt receiver MAC, appearing to be addressed to some other node, then the packet is corrupt anyway and the fact that this node slept during that packet will not matter at all. In fact any packet whose destination MAC appears to be the current STA's MAC is not a source of concern since the node will fully decode the packet and then check the FCS, discarding the packet if corrupt.

The potential for packet loss is mainly due to packets whose MAC appears to be different from the current node's MAC since these packets do not have their FCS cross checked, and their headers are trusted implicitly. It is to be noted here that if the bits of Frame Control (FC) field of 802.11 MAC header are incorrect and if an 802.11 DATA packet is misinterpreted as a control packet it will be decoded, checked and discarded which means there is no damage done. Similarly, if a control packet is misinterpreted as an 802.11 DATA packet due to corrupt FC bits there is no damage done since in the worst case the node sleeps on the packet which does not matter since the packet is corrupt in any case.

Thus the only case that is actually significant here is the case where the FC field is correct and the Duration field of 802.11 MAC header is the one that is corrupt. In this case, one of two things may happen:

1) The actual Duration field is more than the observed duration field: in this case either the node may not sleep when it could have actually slept or the node may sleep for lesser time than it could. While these cases lead to reduction in power saved they do not lead to packet loss.

2) The actual Duration field is less than the observed duration field: this is the case that is most worrying since the node sleeps for longer than necessary meaning it may not initiate transmissions when it potentially can and can even miss incoming transmissions. This is the only case that actually leads to packet loss. However as observed by [1] this case is so unlikely that we believe taking the small risk of this happening is better than introducing a new frame format which will not only break compatibility with existing Wi-Fi equipment but will also mean that in order for a node to save power all nodes on the network have to implement Übersleep, especially considering that nodes will usually re-transmit any packets dropped in this manner. Another problem with the Duration field is that for Data packets the Duration field contains the duration for the complete frame exchange (i.e., DATA-ACK), therefore, it cannot be used to determine sleep time duration. We utilize the PLCP header fields, SIGNAL Field and PSDU LENGTH Word, that carry information about the transmission rate and length of the MAC frame, respectively, for determining the time duration at

which the Wi-Fi radio can be turned back on.

D. Comparison with built-in 802.11 PSMs

The proposed approach's greatest benefit is that nodes can save power without significantly losing out on throughput by implementing Übersleep, even when all other nodes in the WLAN are legacy devices and do not support Übersleep. This is in stark contrast to the built-in PSMs in 802.11, which require support on the AP, which may not be available, especially in Small office/Home office (SoHo) use. In addition, the proposed power save scheme is not mutually exclusive with 802.11 PSMs, and can even be safely combined with it, since our approach does not in any way modify or shape traffic on the channel. The passivity of our approach also means that unlike 802.11 PSMs which force buffering of broadcast packets the moment even one node in the WLAN enters PSM, the throughput of other nodes on the channel will not be affected at all, unless they are transmitting to/from the current node. In addition, as analysed earlier, even the throughput of flows to/from nodes implementing Übersleep is not significantly affected.

Algorithm 1 The algorithm that is used by a receiver node implementing Übersleep

```
if Incoming packet is a control packet then
    Do not sleep on this packet
else
    if incoming packet is an A-MPDU then
        Read the first 14 B to check the receiver MAC ID
    else
        Read the first 10 bytes to check the receiver MAC ID
    end if
    if Incoming packet is not destined for this node AND
    Sleep+Wake time < duration of packet AND
    Energy can be saved by sleep then
        Sleep through the duration of this packet
    end if
end if
```

E. Experimental setup

Due to poor support for 802.11n/ac, and especially for MPDU aggregation (which significantly boosts savings from the proposed approach), in most network simulators, we used an innovative approach to measure potential power savings due to the Übersleep scheme.

A private infrastructure Basic Service Set (BSS) was set up and nodes were associated with it. Then on all these nodes, TCP and/or UDP flows were setup using *iperf* in both uplink and downlink directions. The data that they generated and all other management and control frames were captured using an *AirPcap-Nx* [12] wireless capture device. Capturing through monitor mode on Ubuntu with an Atheros wireless driver lead to an erroneous capture as the wireless driver de-aggregated the A-MPDUs into MPDUs and Block ACKs into regular ACKs before they were passed to *WireShark*, which made it impossible for us to study the effect of frame aggregation and block ACKs on Übersleep. Similar results were obtained when we performed a capture with a Broadcom driver on Linux and with a Broadcom driver on Mac OS X as well, forcing us to use AirPcap. The *TShark* [13] protocol analyser was used to pre-process the captured packets and extract all required information into a CSV file, simultaneously discarding unnecessary headers from higher layers. Finally, a python script was used to process this file and estimate the consumption of power by a reference receiver and compare it with one that implements Übersleep, using the assumption that the power consumed during sleep activation and deactivation is the same as the full receive power and that the power consumed during the remainder of the sleep is equal to the IDLE power of $100mw$ as indicated by [4]. [4] was also used to estimate the consumption of power for a receiver in RX mode against the number of spatial streams in the packet being received.

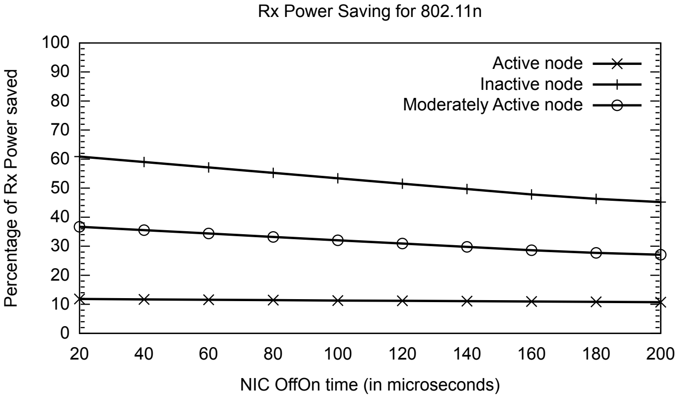

Fig. 4. Power saved by 802.11n nodes implementing Übersleep

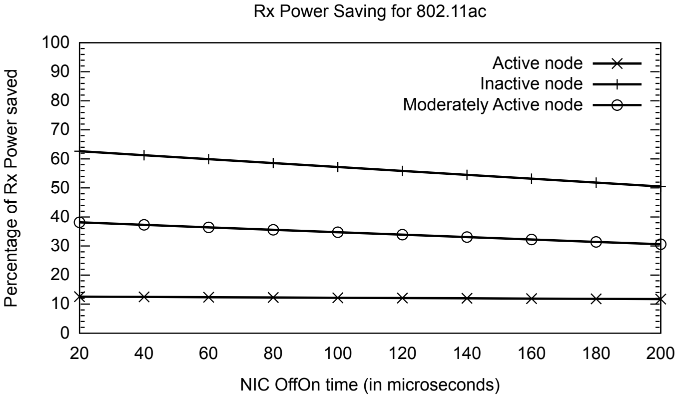

Fig. 5. Power saved by 802.11ac nodes implementing Übersleep (extrapolated)

F. Results

The proposed approach was implemented in the manner described above, and the results indicated that Übersleep on a receiver whose sleep+wake time is 40μs can save up to 60% of RX power (we estimate the percentage of power saved by the receiver side alone).

The results obtained on one test data set for three different nodes with different levels of activity clearly indicate that a significant amount of RX power can be saved in 802.11n radios, and that the power saved is not lowered too

TABLE I. The energy savings obtained using 802.11n/ac in percentage for various activity levels of receivers assuming sleep+wake time as 40 μs.

Type of node	Power Savings(n,%)	Power Savings(ac,%)
Active Node	59.007	61.29
Moderately Active Node	35.54	37.28
Inactive Node	11.69	12.47

TABLE II. The time savings obtained using 802.11n/ac in percentage for various activity levels of receivers assuming sleep+wake time as 40 μs.

Type of node	Time Savings(n,%)	Time Savings(ac,%)
Active Node	68.44	69.34
Moderately Active Node	42.85	43.82
Inactive Node	14.42	15.22

much by an increase in the sleep+wake time of the radio (although it does depend on the level of activity of the node). The results obtained on extrapolating aggregated frame sizes and data rates to those of 802.11ac indicate that our approach can generate significant saving even in this new and upcoming standard.

Figures 4 and 5 summarize our results, with results for 802.11ac extrapolated using normalisation factors for aggregate size and transmit rate. We show the power saving on three different nodes, with different levels of activity in the trace file (the less active nodes saving more power) in Table I (for a representative sleep+wake time).

We extrapolated the results for 802.11ac. The extrapolation that we did was based on the following points and the multiplication factor was calculated accordingly:

1) **Packet size:** 802.11ac increased the maximum packet size to 1MB (as compared to 64KB in 802.11n). So the sizes of all packets were multiplied by a factor of 1024/64.
2) **Rate factor:** 802.11ac also significantly increased the datarate when compared with 802.11n. Thus we increased the datarate in our calculations, which in-turn reduced packet air time. We used a Cisco white paper [14] available for 802.11ac to estimate the factor by which datarates increased to be 6930/600.
3) **Power Consumption:** We were not able to find any data about the increase or decrease in power consumption values. Hence in this paper we have made the assumption that the power consumption remains same for the components in 802.11ac as compared to 802.11n.

Furthermore, we believe that the actual power saving that Übersleep achieves for 802.11ac may be higher than our calculations because all 802.11ac packets are necessarily A-MPDUs.

We also estimated the percentage of time that a node could spend in sleep mode, and the results obtained on one test data set for three different nodes with different levels of activity clearly indicate that a significant amount of RX time can be saved in 802.11n radios. This measure counts the time for the radio to sleep and awake as saved as well. Since most frames are larger than about 150μs, the percentage of RX time saved seems to be independent of the sleep+wake time because the radio ends up picking exactly the same frames to activate Übersleep. It can also be seen that under the assumption the radio consumes power equal to the idle power when in sleep mode and consumes power equal to RX power while in the process of entering/exiting sleep, the power savings are consistently lesser than the time savings. The variance of this against the sleep+wake time of the radio was also analyzed and is summarized in Figure 6.

When we extrapolated the data rates to 802.11ac and simultaneously extrapolated the aggregate size, we observed that similar savings can be expected for 802.11ac as well, with results indicating that this approach is extensible to 802.11ac as well. The results extrapolated for 802.11ac are plotted in Figure 7.

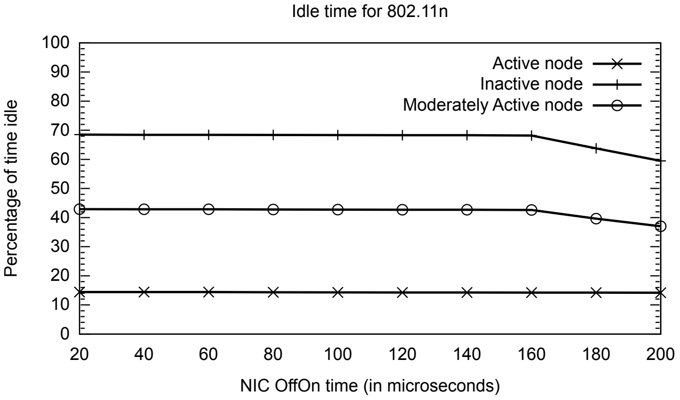

Fig. 6. Time savings for 802.11n radios implementing Übersleep

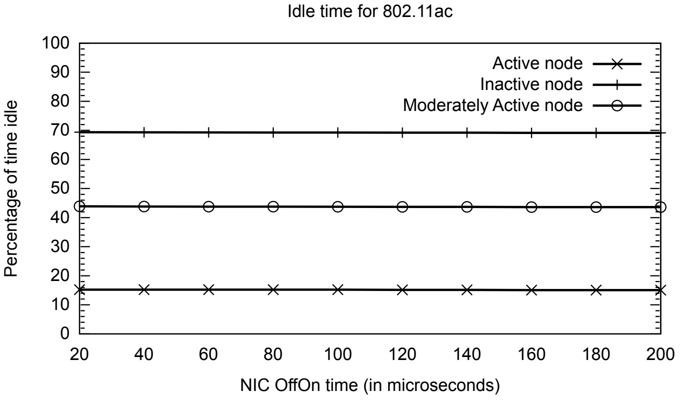

Fig. 7. Time savings extrapolated for 802.11ac radios implementing Übersleep

The results for a particular representative sleep+wake time for both 802.11n and 802.11ac are summarized in Table II.

IV. Conclusions

The proposed Übersleep mode in 802.11n has been successfully tested. As we observed, A-MPDU aggregation boosts the power saved by this approach significantly without any increased latency or loss in throughput. Thus, an extension of Übersleep to 802.11ac mode promises even better power savings. As observed in the graphs obtained by [1], and as expected via probability, we observed that there is no need to add an extra header field since the increased overhead and lack of backward compatibility are greater problems than the extremely rare occurrence of misinterpretation of the packet, that too in such a way that the carrier sense is extended over a longer period than the actual reservation.

ACKNOWLEDGMENT

This work was supported by the Department of Electronics and Information Technology (DeitY), Govt. of India (Grant No.13(6)/2010CCBT).

References

[1] B. Balaji, B. Tamma, and B. Manoj, "A novel power saving strategy for greening ieee 802.11 based wireless networks," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, 2010, pp. 1–5.

[2] M. Gupta and S. Singh, "Greening of the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 19–26. [Online]. Available: http://doi.acm.org/10.1145/863955.863959

[3] J. Manweiler and R. Roy Choudhury, "Avoiding the rush hours: Wifi energy management via traffic isolation," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 253–266. [Online]. Available: http://doi.acm.org/10.1145/1999995.2000020

[4] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n power consumption," in *Proceedings of the 2010 international conference on Power aware computing and systems*, ser. HotPower'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–. [Online]. Available: http://dl.acm.org/citation.cfm?id=1924920.1924928

[5] V. Baiamonte and C.-F. Chiasserini, "Saving energy during channel contention in 802.11 wlans," *Mob. Netw. Appl.*, vol. 11, no. 2, pp. 287–296, Apr. 2006. [Online]. Available: http://dx.doi.org/10.1007/s11036-006-4480-x

[6] S. Chatterjee, D. Falkenthal, and T. Ree, "Exploring adaptive power saving schemes for mobile voip devices in ieee 802.11 networks," in *Digital Telecommunications, 2007. ICDT '07. Second International Conference on*, 2007, pp. 13–13.

[7] P. Agrawal, A. Kumar, J. Kuri, M. Panda, V. Navda, and R. Ramjee, "Opsm - opportunistic power save mode for infrastructure ieee 802.11 wlan," in *Communications Workshops (ICC), 2010 IEEE International Conference on*, 2010, pp. 1–6.

[8] C.-Y. Li, C. Peng, S. Lu, and X. Wang, "Energy-based rate adaptation for 802.11n," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, ser. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 341–352. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348585

[9] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 323–336. [Online]. Available: http://dl.acm.org/citation.cfm?id=1387589.1387612

[10] Y. Xiao and J. Rosdahl, "Throughput and delay limits of ieee 802.11," *Communications Letters, IEEE*, vol. 6, no. 8, pp. 355–357, 2002.

[11] Extricom, "802.11n for enterprise wireless lans," http://www.extricom.com/files/documents/witepapers/_9sY_802.11n_for_the_enterprise_v1.pdf, 2010.

[12] Metageek, "Airpcap- packet analysis for windows," http://www.metageek.net/products/airpcap/, 2012.

[13] "tshark - the wireshark network analyzer," http://www.wireshark.org/docs/man-pages/tshark.html.

[14] "802.11ac- the fifth generation of wi-fi," http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps11983/white_paper_c11-713103.pdf.