

Application Level Energy and Performance Measurements in a Wireless LAN

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

Abstract—We present an experimental evaluation of energy usage and performance in a wireless LAN cell based on a testbed using the 5GHz ISM band for 802.11a and 802.11n. We have taken an application-level approach, by varying the packet size and transmission rate at the protocol level and evaluating energy usage across a range of application transmission rates using both large and small packet sizes. We have observed that both the application’s transmission rate and the packet size have an impact on energy efficiency for transmission in our testbed. We also included in our experiments evaluation of the energy efficiency of emulations of YouTube and Skype flows, and a comparison with Ethernet transmissions.

I. INTRODUCTION

Wireless LAN (WLAN) 802.11 technology is increasingly used to provide connectivity for mobile computing devices. The maturity and widespread deployment of 802.11 infrastructure, and the availability of cheap, integrated chipsets means that it is a popular choice for many devices including smartphones, as well as low-cost consumer devices such as netbooks and hand-held game consoles. Additionally this technology is also increasingly used in non-mobile use cases for providing rural areas with Internet connectivity [2]. While the use of 802.11 is widespread, it has not been designed with energy efficiency as a key priority, albeit, sleep mode is possible. Whilst in the future this situation could improve, with, perhaps, the definition of new energy aware and energy adaptive 802.11 extensions, e.g. [9], [10], [23], the current deployed technology is likely to remain in place and in use for some time. Additionally, in situations in which WLANs are used to provide key infrastructure, providers may be reluctant to jeopardise performance by enabling new energy-saving features. So, it is useful to understand how application-specific adjustments of packet transmissions can effect energy efficiency and performance.

We take the position that it may be possible for applications to adopt energy efficient behaviour even if the underlying WLAN transmission capability is not designed to be energy efficient. By understanding the performance and energy efficiency of the current capability of 802.11 systems, there is the potential for large-scale energy efficiency gains. Hence, we explore the way that the generation of packet flows at the application level impacts upon energy efficiency. We believe it may be possible for applications to dynamically adapt their

packet flow generation in order to improve (or trade off) performance with energy consumption [3]. Ultimately, our ongoing work aims to design *energy-aware* self-adaptation policies for applications.

In this work we have considered the potential for application adaptation capability by determining the energy efficiency of WLAN transmissions. Specifically, we have examined how:

- the (packet) transmission rate; and
- the packet size

used by an application in a 5GHz WLAN, in 802.11a and 802.11n, will impact energy usage. We have conducted experiments on a test-bed using off-the-shelf equipment. We generated different packet-level flows and measured the power consumed during transmission. From various measurements of performance and power, we have drawn conclusions about the energy efficiency of transmissions in a WLAN cell. Our measurements have allowed us to determine an energy usage “envelope”, showing the upper and lower bounds of energy efficiency. We have also emulated YouTube and Skype flows, as the popularity of such application is increasing [6]. We find that they operate in an area of the energy envelope in which energy is not used efficiently and suggest how this may be corrected by enabling self-adaptation. Additionally, we have compared the wireless measurements with measurements on an Ethernet network.

The remainder of this paper is structured as follows. In Section II experiment design and metrics for measuring both performance and energy usage are outlined. Next, we present our observations with some discussion in Section III. In Section IV we look at existing work on performance and energy and make comparisons with findings and methodology presented in our experiments. We summarise our findings and give an outline of future work in Section V.

II. EXPERIMENT DESIGN AND METRICS

We specifically wish to measure the energy usage of WLAN transmissions with 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 specific energy efficient tuning.* Specifically, our constraints are:

- *Standard WLAN configuration.* We used only standard, un-tuned WLAN radio-channel configurations (more details later). While many WLAN NIC drivers do permit

various controls of the hardware, this is not easily accessible or comprehensible for modification by most users.

- *Whole-system power usage.* Rather than just look at the NIC only, e.g. [13], we have considered the energy usage of the system overall in transmissions, in line with our motivation (Section I) of application-level adaptation. For simplicity, we consider that the only information that may be available to an application is the overall power usage of the system as a whole.
- *Packet flow behaviour.* We have assumed that the only part of the system that can be changed by the application is its own packet flow. Future systems may allow more detailed control of underlying radio and NIC sub-systems, but for existing deployed capability such facilities are not widely available today.

We have deliberately chosen the assumptions above to present challenging constraints, but we have shown that even within these constraints, useful information for energy usage can be made available.

One further practical constraint we have used is that of a 5GHz-only testbed. This was because in our local environment, we have exclusive usage of 5GHz and so our experiments were free from interference from other departmental 5GHz deployments. However, we can easily apply our methodology to standards operating in the 2.4GHz band (e.g. IEEE 802.11g) in the future through a simply software configuration change in our testbed.

A. Overview

We have experimentally evaluated energy usage and performance in our 5GHz testbed, with 802.11a and 802.11n, using off-the-shelf equipment. We generated packet flows of various bit-rates and packets sizes, and measured power usage during the packet transmission. Our testbed (Figure 1) consisted of a single client host, a host running a wireless access-point (AP) and experimental control units (only one shown in 1) for monitoring the WLAN environment, providing storage for measurement data, *ntp*¹ services and system configuration. The WLAN hosts were setup in a teaching lab in the University of St Andrews with a distance of $\sim 24 \pm 0.5$ m between the antennae.

Packet generation and performance measurement was conducted using *iperf*² for which the AP was used as the server. A wrapper script executed *iperf* and extracted throughput and loss for individual UDP flows using *iperf server reports*. Power consumption was measured at the client and the AP using a commercial power meter (with an inductive clamp).

B. Workloads: packet flows

We wished to measure the maximum and minimum power efficiency that was achievable. So, we configured UDP flows across a range of bit rates, with both small and large packets sizes³. Our motivation for using UDP was its popularity for

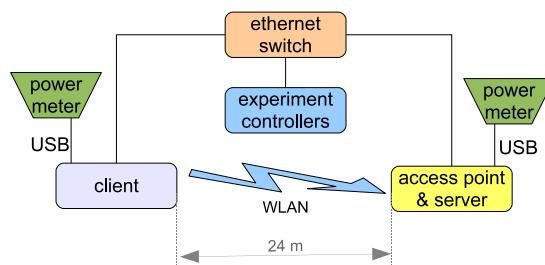


Fig. 1. Schematic of test-bed showing physical connectivity. We used 5GHz only, and the testbed was configured separately for 802.11a and 802.11n (20MHz channels) experiments. The experiment controller uses Ethernet for control messages and shared file-system access. The power meter readings are logged by each node using USB connections from the power meter. The separation between the antenna of the client and access point/server is 24m. Data packets generated by *iperf* were transferred across the WLAN.

Voice and Video over IP (VoIP and ViIP) applications, and that it allows us to control application specific bit rates accurately: TCP’s congestion control behaviour does not permit such accurate control. The range of control values is summarised in Table I, and these were applied separately for 802.11a and 802.11n, and (see later) Ethernet for comparison. The measurements were used to establish the “energy-efficiency envelope” for 802.11a and 802.11n in our testbed.

TABLE I
UDP CONTROL VARIABLES.

Packet size in offered load ³	64; 1460 bytes
Offered load’s bit rate ³	32; 50; 256; 512 Kbps 1; 5; 10; 15; 20; 22; 24; 26 Mbps

Each packet size was combined with each bit-rate (24 combinations); 25 flows measured with each combination executed for each of 802.11a, 802.11n and Ethernet (600 flows for each); each flow had a duration of 4 minutes (~ 120 hours of measurements).

We wished to have a direct comparison between 11a and 11n, as both operate at 5GHz, so we chose to limit our experiments to the highest real throughput we were able to achieve with 11a on our test-bed – initial (TCP) calibration experiments showed this to be 24 ± 1 Mbps, though 802.11n can achieve higher throughput. Also, we take the position that the areas of interest are likely to be at the lower rates, for now at least, as applications tend to have lower bit rates (e.g. the Skype and YouTube examples we have chosen), and previous work has shown that, for energy efficiency, the dynamics are more noticeable at the lower rates [13].

The 64 byte packet is the smallest size for which we have observed that *iperf* is able to generate server reports. The 1460 byte packet is chosen as that is a common MTU size used for Internet-wide communication (a known legacy of Ethernet), and we wished to avoid the effects of IP-level fragmentation.

The flow duration for all traffic profiles of 4 minutes was chosen based on previous studies of application behaviour: Skype call durations are between 2 – 6 minutes for the majority of the monitored sessions in [4], [5], and average YouTube

¹<http://www.ntp.org/>

²<https://sourceforge.net/projects/iperf/>

³Please note: in contrast to related work on performance we are referring to characteristics of UDP flows in this context for the rest of this document.

sessions are 3–5 minutes [11], [24] (see below).

As well, as the control variables explained above, we also wished to observe where typical applications might fall in this envelope. For simplicity and reproducibility, we have used emulated flows for Skype and YouTube, as summarised in Table II. Traffic representative for Skype (VoIP) streams with a bit rate of 65 Kbps and a packet size of 300 bytes was derived from previous studies [4], [5]), as was traffic representative for YouTube (ViIP) streams with a bit rate of 639 Kbps and a packet size of 1431 bytes (based on [11], [24]). These also served as a sanity check for our envelope. We note that our packet-level emulation of these application flows would not incur on our testbed the load and energy overhead of audio/video encoding/decoding as would a real instance of those applications.

TABLE II
EXAMPLE APPLICATION UDP TRAFFIC EMULATION.

Skype	300 byte packets, 65 Kbps
YouTube	1431 byte packets, 639 Kbps

25 measurements with each flow (50 flows); flow duration of 4 minutes (200 minutes of measurements). Configurations based on [4], [5], [11], [24].

C. Observed variables

In each experiment we have measured the observables as described below, and summarised in Table III:

- *Performance*: throughput and loss, as recorded by *iperf*'s server reports on the client for each UDP flow.
- *Power*: every 30s we have recorded the current power consumption in *Watts* at the AP and client.
- *WLAN rate*: the currently used Modulation and Coding Schemes (MCS), and received signal strength indicator (RSSI), as recorded every 30s at the client using *iwconfig*⁴.
- *WLAN spectrum*: the signal strength (showing channel utilisation) as recorded every 30s interval via the USB spectrum analyser *WiSpy*⁵ at the observer host. This was to confirm that during the measurements, only our test-bed was operating at 5GHz, i.e. to spot possible interference from other sources.

The monitoring intervals for all of the above observables were deduced from preliminary experiments. Motivation for doing this was to avoid excessive disk I/O in cases where the monitoring machinery operated on a host on which we also conducted e.g. power measurements. Hence the intervals were chosen to result in sufficient monitoring sample sets for determining significant differences between experiments without biasing other measurements.

We also conducted control experiments as follows: (i) idling measurement (P_I), where we measured mean power consumption when flows were not being generated (Section II-D); (ii) control measurement of the overhead were conducted with a sub-set of our experimental parameters to investigate the

⁴<http://man.he.net/man8/iwconfig>

⁵<http://www.metageek.net/products/wi-spy/>

TABLE III
OBSERVABLES MEASURED DURING EXPERIMENTS.

Observable attribute	Units	Comment
Performance	throughput (Mbps) loss (%)	<i>iperf</i> server reports
Power	Watts	power meter used on node's power cable
WLAN rate	data rate (Mbps) signal strength (dBm)	recorded using <i>iwconfig</i>
WLAN spectrum	signal strength using <i>WiSpy</i>	at an observer

The first three observables, throughput, loss and power, were used for our energy efficiency evaluations; the others were used as experimental checks.

exhibited overhead in the WLAN (Section III-E); (iii) probe effect measurements where we measured the difference in the power consumption with and without monitoring (Section III-F); and (iv) comparison measurements where the workload was executed via the Ethernet network interface (Section III-G).

D. Energy efficiency

We measured power consumption on client and AP at 30 second intervals. For energy usage, we define *Effective Application-specific energy-usage* (E_A) as follows:

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

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

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

To generate values for E_A , for each individual flow, we use the following measurements:

$$E_A = \frac{P_A - P_I}{T_A} \quad (1)$$

P_A Mean power consumption measured during the transmission of flow [Watts].

P_I Mean power consumption measured for an idling node [Watts].

T_A Mean throughput measured (using *iperf*) during flow transmission [Mbps].

P_I was measured to be 54 Watt when the machine idled, performing only power consumption monitoring.

E. Equipment

The client, server and observer were each identical hardware: a Shuttle X (*XPC Barebone S556G*⁶) with Intel®Pentium®4 CPU 3.00GHz, 1 GB RAM, 112 GB HD. Each was equipped with a wireless LAN equipment⁷ as shown in Figure 2.

Our WLAN card uses the popular Atheros⁸ chipset. As of the beginning of 2011, Atheros is the second largest WiFi

⁶http://www.shuttle.eu/_archive/old/es/www.shuttle.eu/html/index-416.html

⁷http://www.compex.com.sg/Datasheets/WLM200NX_DSv3.2.9.pdf

⁸<http://www.atheros.com/>

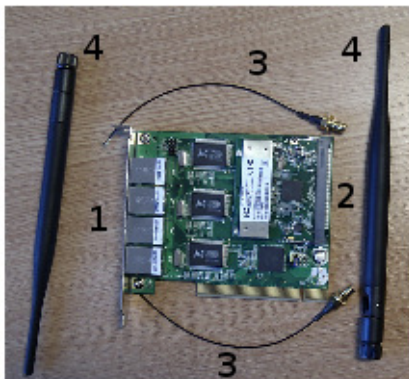


Fig. 2. WLAN components in test-bed nodes. 1. RTL 4 port network card (32-bit PCI), model P811B-4R with RLT8100C chipset, mini PCI slot. 2. COMPEX iWawoport WLM200NX(2T2R) 802.11N a/b/g/n miniPCI card 20d. 3. IPAX/U.FL to ReSMA Chassis Socket – 15cm. 4. Omni-directional antenna, 2.4GHz/5GHz DualBand – 5dB.

chipset developer holding nearly 21% market share. Moreover, Qualcomm, the largest global manufacturer of mobile phone chipsets, has decided to acquire Atheros, which may boost Atheros’ market share and popularity [21].

We have used a *WiSpy DBx* WLAN probe⁹. We have adapted *spectools_raw* (version 2010-04-R1)¹⁰ with a set of custom scripts to post-process spectrum monitoring data. This was mainly used for sanity checks during initial/preliminary runs and for on-demand-analysis and checking of spectrum usage during experiment runs.

To test out-of-the-box configurations, we kept to a minimum any modification of the operating system or driver. We were running a minimal *Ubuntu 10.04* Server installation rather than Desktop installation, as we wanted to avoid additional system/energy load due to a GUI and other desktop processes and daemons. During the setup of our testbed we noticed that the *ath9k* driver suite, part of the default kernel (2.6.32-24) version for *Ubuntu 10.04*, does not provide all the desired monitoring information when using 802.11n and so upgraded the kernel to the newest version available in the *Ubuntu* repository at that time (2.6.35-25). We used the new kernel only at the client as it was incompatible with the bridge-control software suit we used for running our access-point. For running the access-point we have used the *hostapd*¹¹ package, the configuration files which used *hostapd*’s default parameters for every radio and hardware level configuration. All nodes in the testbed were isolated from the rest of the network and the system clocks of all the nodes were synchronised (using NTP) before each individual experimental run. For measuring power, we have used a CC128 power meter¹² (see Figure 3), which has an XML data feed accessible via USB.

⁹<http://www.metageek.net/products/wi-spy/>

¹⁰<http://www.kismetwireless.net/spectools/>

¹¹<http://hostap.epitest.fi/hostapd/>

¹²<http://www.currentcost.com/product-cc128.html>



Fig. 3. Some nodes from our testbed. 1. The *WiSpy DBx* probe with antenna, mounted in a desktop USB hub. 2. The CC128 power meter and its power clamp (behind the CC128 LCD screen).

III. RESULTS AND DISCUSSION

A. Overview

We present the application-specific variables – throughput, loss, effective application-specific energy – E_A (Section II-D) to analyse the effects of the application-level changes in bit rate and packet size. For each variable we plot the mean and the standard error (with 95% confidence) over 25 runs, per packet size for the offered load at each bit rate. Please note that in the majority of the experiments, only very small error bars were calculated, so error bars may not always be easily visible even though they have been plotted. Also, as we have made measurements at discrete values of the control variables, lines on plots should be considered only as a visual aid, and do not represent an interpolation of results.

B. IEEE 802.11a

Figures 4 and 5 show throughput and loss in 802.11a of ranges of UDP traffic patterns ranging from 32 Kbps to 26 Mbps with packet sizes of 64 B and 1460 B. The traffic patterns also include an emulation for Skype and another one for YouTube. The results show that the throughput increases up to the operational limit of about 25 Mbps with large packets and only up to about 3 Mbps with small packets. The bit rate of the offered load at which we observe an increase of the loss rate also corresponds with this operational limit.

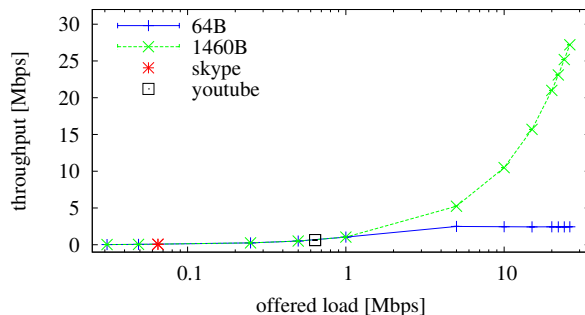


Fig. 4. UDP throughput in IEEE 802.11a.

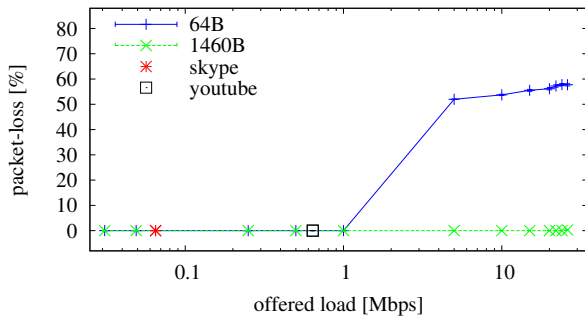


Fig. 5. UDP loss rates in IEEE 802.11a.

The high loss rate at values greater than 10 Mbps offered load with 64 bytes packet size was not due to any radio frequency interference as shown in Figure 6. Even though we have had departmental wide exclusive usage of the 5 GHz band we wished to detect potential interference from other sources due to the general availability of 5 GHz enabled devices. Figure 6 shows that with low bit rates (1 Mbps) only occasional loss-peaks occur which do not correspond with (minor) drops in the link quality, as reported by *iwconfig*.

The signal quality can be seen to progress at similar levels in experiments with higher bit rates (15 Mbps), even though we record higher loss rates. The minor drop in signal quality from about minute 70 to minute 95 has no effect on the progression of the loss rate. Further investigation showed a drop in the received signal strength indication (RSSI) being the reason for the drop in the link quality rather than any interference (potentially due to a temporary obstacle between the antennae).

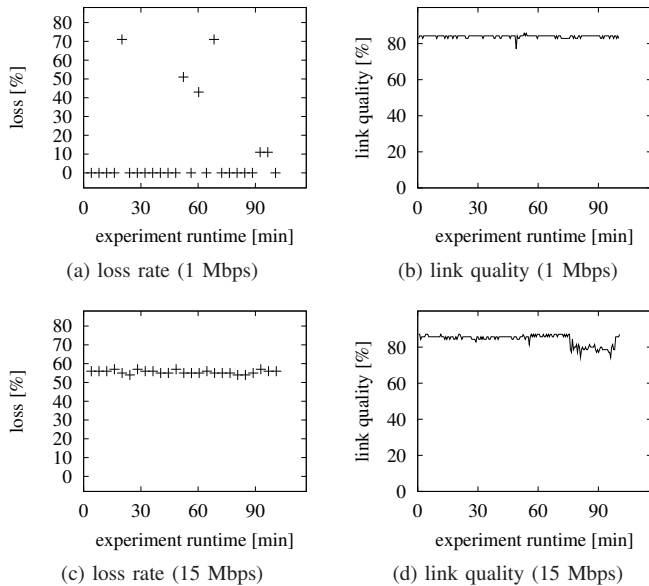


Fig. 6. Progressions of link quality and loss rate over the experimental run time in experiments using 802.11a with 1 and 15 Mbps rates and 64 B packets.

An application's bit-rate has a greater effect than packet size on energy efficiency, E_A . Figure 7 shows that in the ranges in which lower bit-rate ViIP and VoIP applications (like Skype

and YouTube) operate, we observe significantly lower energy efficiency than applications with a higher bit-rate or larger packets.

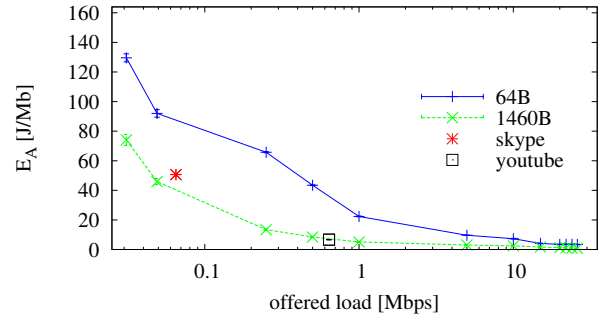


Fig. 7. Application-Specific Energy Usage at the Client with 802.11a.

C. IEEE 802.11n

Our corresponding observations in experiments in which we have used 802.11n are similar to our observations in experiments in which we have used 802.11a (Figures 4 – 7). Thus we are providing all corresponding 11n related measurements (Figures 8 – 11) but focus in our discussion here on the differences between those measurements which are not captured by our direct comparison in Section III-D. There is, for instance, no great difference between small and large packets in throughput up to an offered load of 5 Mbps. We however see that 11n supports higher throughputs than 11a if small packets are used with higher rates of offered load.

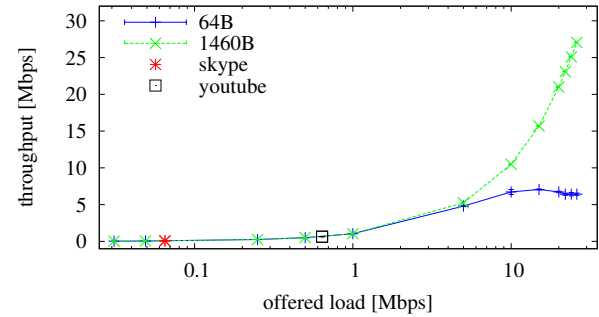


Fig. 8. UDP throughput in IEEE 802.11n.

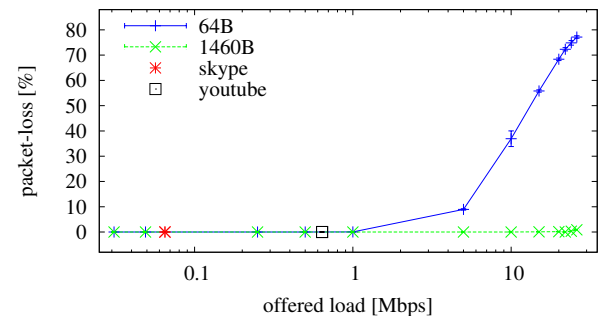


Fig. 9. UDP loss rates in IEEE 802.11n.

In similarity to 802.11a, for 802.11n we see that the high loss rate at throughput values greater than 10 Mbps offered

load with 64 bytes packet size was not due to any radio frequency interference, as shown in Figure 10.

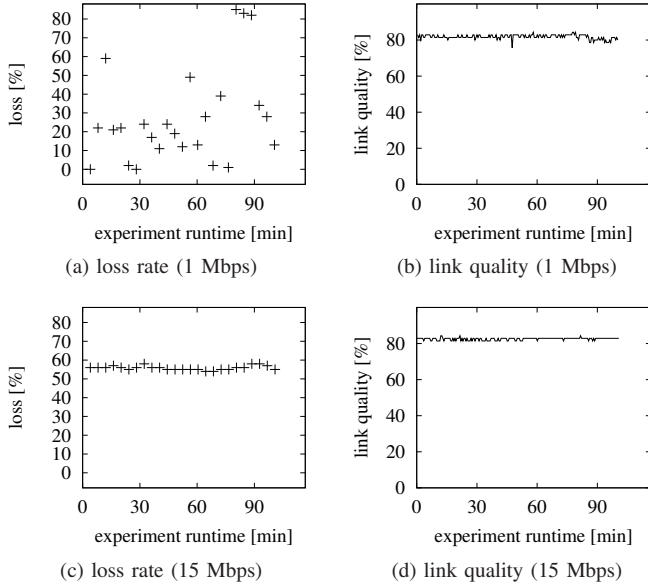


Fig. 10. Progressions of link quality and loss rate over the experimental run time in experiments using 802.11n with 1 and 15 Mbps rates and 64 B packets.

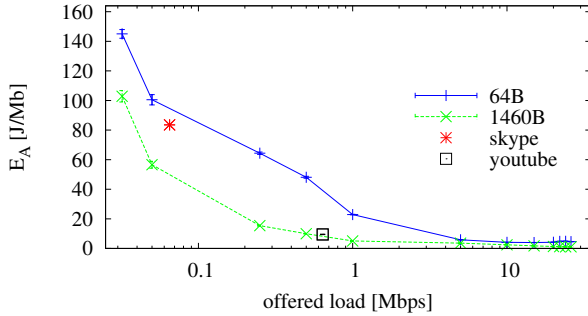


Fig. 11. Application-Specific Energy Usage at the Client with 802.11n.

D. IEEE 802.11a vs. IEEE 802.11n

To analyse the difference between 802.11a and 802.11n performance, we compute Δ values for energy, throughput and loss and plot them in the following section (please note that we omit Skype and YouTube measurements for brevity). We define a Δ value as the relative differences of an “802.11a value” to its “802.11n counterpart” for throughput and energy, and for Δ loss we use simply the difference of $loss_{802.11n} - loss_{802.11a}$ as loss is already a relative metric. As with all our analysis we are comparing energy, throughput and loss up to 802.11a’s operational limit and ignore higher throughputs which are only achievable by 802.11n. Figure 12 shows that both 802.11 flavours result in a similar throughput when an application uses large packets. If, however, small packets are used, 802.11a achieves up to 60% less throughput than 802.11n.

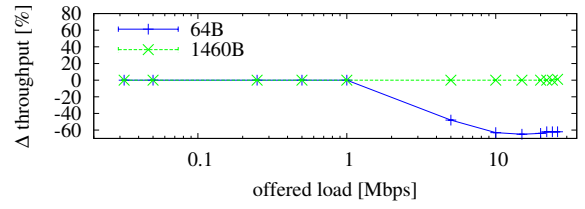


Fig. 12. Differences of throughput in IEEE 802.11a and IEEE 802.11n. The positive (negative) values show where 802.11a is better (worse) than 802.11n.

As with throughput, we observe no difference between loss in 802.11n and 802.11a if large packets are used (Figure 13). Small packets however result in up to 20% more loss in 802.11n than in 802.11a with bit rates greater than 20 Mbps up to the operational limit, but less at lower bit rates.

We can summarise that, as expected, 802.11n provides better performance than 802.11a (especially when taking the higher operational limit of 802.11n into account). If, however, an application sends data only with lower bit rates, no difference can be observed. The same applies up to the operational limit of 11a, if only large packet sizes are used.

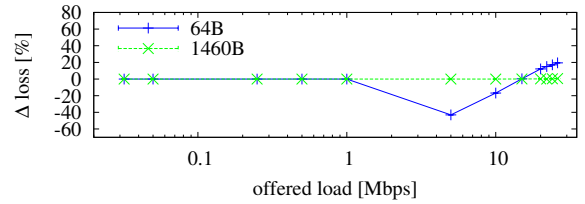


Fig. 13. Differences of loss in IEEE 802.11a and IEEE 802.11n. The positive (negative) values show where 802.11a is better (worse) than 802.11n.

In Figure 14 (the line at 0% is a visual aid), we see that for most points, 802.11a results in better energy efficiency than 802.11n. A peak, during which 802.11a resulted in up to 80 % more energy usage than 802.11n, can be identified for higher bit rates. As the value of E_A decreases with increasing bit rate (in contrast to the progression of performance measurements), this peak corresponds to an absolute difference of about 3 J/Mb whereas the 11% difference with 32 Kbps correspond to an absolute difference of 11 J/Mb. Hence, we can summarise that 802.11n results in lower energy efficiency in most cases compared to 802.11a, at the bit rates tested.

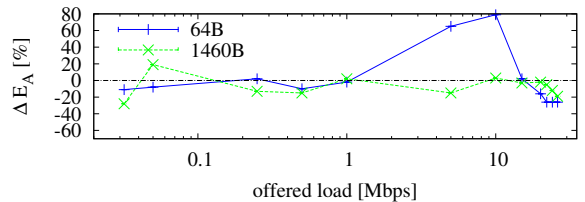


Fig. 14. IEEE 802.11 energy usage. The positive (negative) values show where 802.11a is better (worse) than 802.11n. The dashed line at 0% is included as a visual aid.

As the majority of related work on performance in WLAN has focussed on the link layer and on the bit rates due

to the used Modulation and Coding Scheme (MCS), we summarise here our findings with respect to the different usage of MCS bit rates in 802.11a and 802.11n. The MCS bit rates were chosen in our experiments dynamically by the driver software. In 802.11n, the MCS changed frequently, in contrast to 802.11a where nearly all of the observed MCS bit rates in all experiments were 54 Mbps (this corresponds to the modulation providing the highest data rates at the link layer in 802.11a). No significant changes in our RF setup or environment were monitored, either in experiments with 802.11a or with 802.11n. The highest MCS bit rate we have observed in experiments with 802.11n was 130 Mbps, even though the average RSSI suggests that higher data rates would have been possible. Additionally we have observed that with low application specific bit rates, high MCS bit rates were used, but lower MCS bit rates were observed with increasing bit rate of the offered load. We also observed that with large packet sizes, the frequency of selection of slower MCS for high application specific bit rates, is lower than the frequency of selection of those same MCS bit rates with small packets (Figure 15).

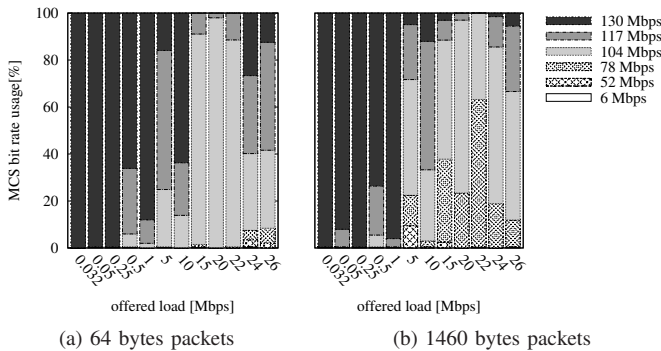


Fig. 15. MCS bit rates at the client in experiments with 802.11n.

In related work, e.g. [13], high MCS bit rates are identified to result in higher power consumption than lower ones. This correlates with our findings for 802.11n (Section III-C). In our observations, system wide energy usage for 802.11a, high MCS bit rates are selected, and similar system wide effects on power consumption and energy usage can be observed.

E. Overhead

In order to analyse the effects of the MAC protocol overhead in the individual experiments we have added traffic monitoring and repeated experiments with offered load values of 50 Kbps, 1 and 15 Mbps using packet sizes of 64 and 1460 bytes for IEEE 802.11a and IEEE 802.11n. Our traffic monitoring is based on *tcpdump*¹³ which was used on our observer host to monitor traffic on interface *wlan0*. Motivation for capturing traffic on an observer host rather than the client was to avoid disk I/O due to *tcpdump* which could have biased power

measurement. The monitoring host was placed within $\sim 1\text{m}$ of the client machine in order to capture all packets sent out by the client without causing any potential near field interference. Due to unsatisfactory monitoring capabilities of Ubuntu in combination with the used wireless driver software, we have used the Backtrack-Linux distribution (version 4 R2) and the latest wireless LAN driver module at the time of carrying out this work¹⁴.

To analyse the effects of the overhead in 802.11a and 802.11n we have computed the *goodput ratio* as the ratio of the payload sent to the network by *iperf* to the total transmitted traffic as monitored by *tcpdump* on *wlan0*. (The goodput ratio can be defined as the useful data delivered to users as a fraction of the total amount of data transmitted on the network.) A high goodput ratio indicates an efficient use of the radio spectrum and lower interference with other users [16].

TABLE IV
GOODPUT RATIOS

Packet Size	Bit-rate Offered Load	Goodput-Ratio 802.11a	Goodput-Ratio 802.11n
64 bytes	50 Kbps	0.601	0.601
	1 Mbps	0.603	0.603
	15 Mbps	0.604	0.604
1460 bytes	50 Kbps	0.970	0.971
	1 Mbps	0.972	0.972
	15 Mbps	0.971	0.971

Table IV shows that with smaller packets the transport medium is used less efficiently than with large packets due to the incurred overhead. This matches with our findings with respect to E_A : we see higher application specific energy values, $E_A(J/Mb)$, for transmissions with small packets than with large ones.

F. Analysis of Probe-Effects on Monitoring Power

Besides monitoring power consumption during the experiments reported above, we also had other monitoring and control scripts running on our client machine. In order to determine if those other monitoring and control scripts had a probe effect on monitoring power we have conducted some control measurements. This consisted of running our experiments in two specific situations which we refer to as *normal monitoring* and *power monitoring only*. The former represented a normal experiment in which UDP flows were executed and all observables (as outlined in Section II) were monitored. In the latter situation we conducted an experiment in which we did not run any monitoring and control scripts, other than power monitoring, and we executed the UDP workload as in the normal experiments. For each comparison we have used two 1 Mbps flows with 64 and 1460 byte packets for 802.11a and 802.11n.

Figure 16 shows consistent power consumption patterns, in line with our reported findings, with negligible impact due to any probe effect due to our monitoring and control scripts.

¹³<http://www.tcpdump.org>

¹⁴compat-wireless-2012-03-31 from <http://wireless.kernel.org>

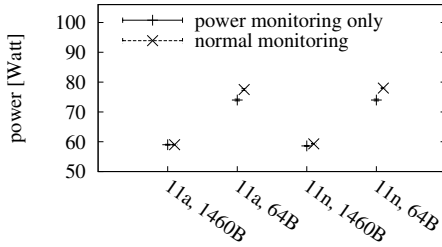


Fig. 16. The effect of monitoring on power measurements.

G. Measurements on Ethernet

To give a relative comparison of our 802.11a and 802.11n measurements with Ethernet, we have repeated our WLAN experiments as outlined in Section II but sent the *iperf* flows over the Ethernet control channel in our setup. We were able to observe similar effects as in the WLAN experiments. Overall, throughput with small and large packets resulted in similar UDP throughput, as shown in Figure 17 (we omit the loss graph, as hardly any loss was monitored on Ethernet). We also observed similarities to the effects observed in WLAN with respect to the energy metric E_A as shown in Figure 18.

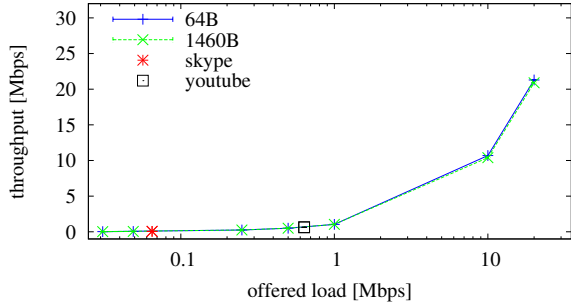


Fig. 17. UDP throughput in wired Ethernet.

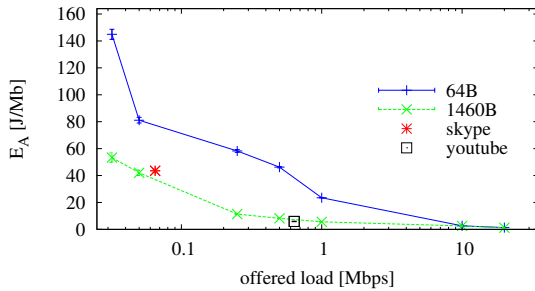


Fig. 18. Application-specific Energy Usage at the Client in wired Ethernet.

H. Energy / Performance Trade-Offs

Our original motivation is to investigate how energy usage can be built into policies for self-adaptation of applications. Our experimental findings allow us to plot an energy-efficiency envelope which shows the upper and lower bounds of energy efficiency. In Figure 19 we overlay the envelope with throughput measurements, and Figure 20 we overlay with loss measurements. We use the higher energy values between 11n

and 11a to plot a *worst-case energy envelope*. Please note that we omit higher data rates for readability, but show the area in which typical VoIP and ViIP (such as Skype and YouTube) operate. We see that if applications send data up to 1 Mbps offered load, no difference in performance can be observed between 802.11a and 802.11n. However, the effect of the change in packet size changes the performance by an order of magnitude. With greater bit rates there is still potential for achieving significantly better energy trade-off if the application is aware that either 802.11a or 802.11n is in use, and an appropriate packet size or bit rate can be configured. Applications like Sype and YouTube operate in the part of the envelope that has potential of improving the energy efficiency by adapting either the offered load or packet size. Additionally, link-level packet aggregation techniques, e.g. A-MPDU and/or A-MSDU [12], may offer energy efficiency benefits, but that is left for future work.

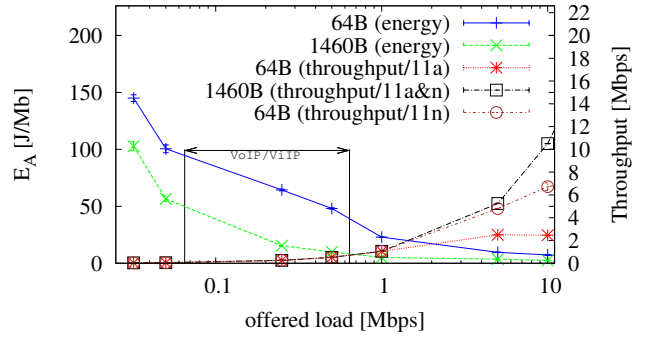


Fig. 19. Worst Case Energy Envelope for Throughput

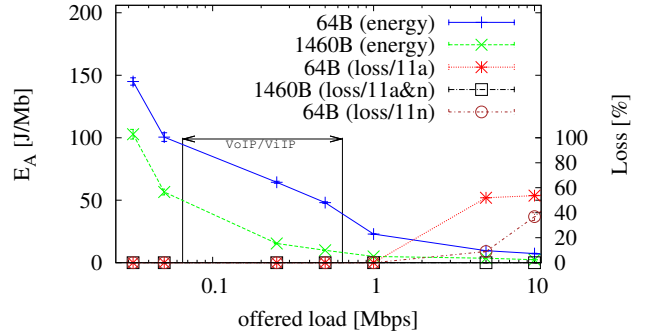


Fig. 20. Worst Case Energy Envelope for Loss

VoIP applications like Skype are of low data rates and use small packets to reduce end-to-end delay and to recover from loss in certain situations, but our results show that flows using low data rates and small packets are not energy-efficient. If the application operates in a scenario with low delay and/or low loss is observed, the operational parameters of the application could be adapted, e.g. by increasing the packet size and/or offered load, to improve energy-efficiency. If packet loss is found to be high, this could be corrected by re-adjusting the flow construction to achieve a better energy/performance trade-off based on the energy-efficiency envelopes.

IV. RELATED WORK

Halperin *et al* [13] 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 measure the power consumption directly at the NIC and ignore system wide effects. They refer to bit rates resulting from the chosen modulation and coding scheme (MCS) rather than a measured application-specific data rate. Their methodology is not designed to measure application-specific, system-wide energy usage. Our application-level approach is in support of our motivation on understanding energy usage to enable applications to self-adapt for optimising energy usage.

In [7], [8] 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, and measure a range of transmission power settings, transmission rate and packet size. Relevant to our work is that they also conclude that large packets use energy more efficiently than small ones. In contrast to our work, the authors focus at the NIC's power usage rather than the system as a whole.

Kuo [14] 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 are more energy efficient than small ones, and that high data rates also result in greater energy efficiency than low rates.

Suong *et al* [17] 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 small packets will result in an increased probability of collision. They correlate this to energy usage simply by defining all collisions as a waste of energy, and hence the probability for collisions is taken as an energy efficiency metric.

Experimental results of an investigation of different IEEE 802.11 standards are provided in [22]. The analysis focuses mainly on coverage, RSSI and interference. The authors provide results of measurements of application-specific performance. Those measurements are conducted using *ping* to determine loss, and file transfers to determine throughput. No information about energy usage is given.

The authors in [18] report on performance measurements at the packet level (using *iperf*), of 802.11a and 802.11g. They use a point-to-point configuration which normally results in better performance, as data is transported over a dedicated link not a shared link. The paper includes throughput graphs which suggest that throughputs up to an operational limit of 23 Mbps were achieved, which approximates well with our observations of a single client for 802.11a.

In [10], the authors compare 802.11n measurements with analytical models in order to improve the models which are used for the analysis of the effects of various MIMO configurations on the performance at the packet level. Their analysis does not provide information about application-specific performance or energy consumption.

In [19], the authors analyse the performance gains of IEEE 802.11n by experimentally evaluating the maximum throughput due to features of IEEE 802.11n, such as channel bonding and various MIMO/SISO configurations. In their evaluation, they use *iperf* for generating UDP streams with mainly fixed, large packet sizes, and provide insight into the effects of various configurations of IEEE 802.11n. In contrast to our work, they do not include energy measurements.

The authors of [23] analyse energy efficiency and performance in WLAN, and introduce an energy efficient MAC protocol. This MAC protocol is an adaptation of the existing IEEE 802.11 protocol which dynamically adapts the interval of the Announcement Traffic Indication Message.

The evaluation of energy usage and performance via an analytic model is presented in [9]. The authors propose a modification of the RTS/CTS operation which would put a waiting station into sleep mode for an expected transmission duration of a currently ongoing transmission. Due to the unavailability of this technology currently, we did not consider it in our experiments, as our goal was to consider current systems with out-of-the-box configurations. Therefore we also omitted technologies which are available but partially standardised, available as options that are not widely used, or in a prototype state (e.g. vendor-specific), or can only be enabled by users with expert knowledge. Those include the 802.11 Power Save Mode (PSM) [20], Unscheduled Automatic Power Save Delivery (U-APSD) [15], and WMM Power Save (WMM-PS) [1].

V. CONCLUSIONS AND FUTURE WORK

Our motivation was to investigate the potential for energy-aware application-level performance adaptation in WLANs. So, the goal of the work reported in this paper was to establish the links between performance and energy usage as relevant to application-level flows.

As a first step towards this goal, we have shown in our experimental evaluation with off-the-shelf equipment, that application-specific adjustments to packet size and offered load can allow trade-offs between performance and energy efficiency. We can summarise that the lower an application's data rate and the smaller its packets, the lower its energy efficiency. We have shown that popular applications like Skype and YouTube operate with a relatively low energy efficiency. Our observations show that there is an *energy-efficiency envelope*, which gives scope for energy-aware adaptation for application packet flows in order to achieve greater energy efficiency. However, how an application could actually do this is for future work.

We used an experimental testbed for our investigations, using IEEE 802.11a and IEEE 802.11n at 5GHz, up to 26

Mbps with 64 byte and 1460 byte packets. We see that (as expected) the overall the performance in an 802.11a network is lower than in an IEEE 802.11n network. However, we also see that no significant difference exists in performance up to the maximum application-specific bit rate supported by 802.11a if applications are tuned to use large packet sizes. A clearer differentiation can be seen with respect to energy efficiency, as IEEE 802.11a gave better results in the majority of our test cases. This means that depending on an application's requirements, an overall benefit may be achieved by using 802.11a instead of the newer IEEE 802.11n, e.g. if the high data rates of 802.11n are not required.

Related work on this topic, carried out mainly in the context of the data link layer, draws partially similar conclusions. The novel contribution of our work is that we provide a rigorous experimental analysis using the new 802.11n standard at 5GHz, and that we focus on the application layer and its impact on system wide energy usage. Also, we have emulated flows for popular applications like Skype and YouTube.

A. Future Work

We are aware that our investigation does not consider multiple clients and uses only a limited range of traffic patterns. This is something we would like to address in future work items. Future work items will also include the evaluation of other IEEE 802.11 standards (e.g. 802.11g at 2.4GHz), and other options such as channel bonding in IEEE 802.11n.

We have deliberately used off-the-shelf equipment and software with 'out-of-the-box' configurations so that our results can be seen as widely applicable. Our observations with respect to driver/kernel issues and IEEE 802.11n (Section II) and, for example, the work of Pelechrinis *et al* [19] is a motivation to use more advanced software in future work.

Overall, this work represents a first step towards our eventual goal of enabling application-level, energy-aware self-adaptation strategies, through design of appropriate policies for self-adaptive applications. Such applications could use technology in a more sustainable manner than is currently possible, by allowing application-level adaptation to improve energy efficiency even where the underlying systems may not necessarily have their own energy-efficiency mechanisms.

(Software scripts, tools and information about the experimental setup can be obtained from the authors on request.)

ACKNOWLEDGEMENTS

Lito Kriara (University of Edinburgh, UK) provided comments and information on related work. Prof Krishna M. Sivalingam and Narendran Krishnan (IIT Madras, India) provided useful discussion and motivational background. This work has been supported mainly by the EPSRC-funded project IU-ATC (<http://www.iu-atc.com/>).

REFERENCES

[1] B. Alfonsi. In brief: Wi-Fi certification program aims to boost battery life. *IEEE Distributed Systems Online*, 7(1), Jan 2006.

[2] G. Bernardi, P. Buneman, and M. K. Marina. Tegola Tiered Mesh Network Testbed in Rural Scotland. In *Proceedings of ACM Mobicom 2008, Workshop on "Wireless Networks and Systems for Developing Regions" (WiNS-DR)*, 2008.

[3] S. N. Bhatti and G. Knight. Enabling QoS adaptation decisions for Internet applications. *Computer Networks*, 31:669–692, 1999.

[4] D. Bonfiglio, M. Mellia, and M. Meo. Revealing Skype Traffic: When Randomness Plays with You. In *Proceedings of SIGCOMM 07*, 2007.

[5] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying Skype User Satisfaction. In *Proceedings of SIGCOMM 06*, August 2006.

[6] Cisco Systems. Cisco visual networking index: Global mobile data traffic forecast update, 2009-2014, February 2010.

[7] 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. Technical Report TKN-02-010, Telecommunication Networks Group, Technische Universitt Berlin, Berlin, Germany, June 2002.

[8] J.-P. Ebert, B. Burns, and A. Wolisz. A trace-based approach for determining the energy consumption of a WLAN network interface. In *Proceedings of European Wireless 2002*, pages 230–236, 2002.

[9] M. Ergen and P. Varaiya. Decomposition of Energy Consumption in IEEE 802.11. In *IEEE International Conference on Communications, 2007. ICC '07*, pages 403–408, 2007.

[10] E. Gelal, K. Pelechrinis, I. Broustis, S. Krishnamurthy, S. Mohammed, A. Chockalingam, and S. Kasera. On the Impact of MIMO Diversity on Higher Layer Performance. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, pages 764–773, 2010.

[11] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (IMC '07)*, 2007.

[12] B. Ginzburg and A. Kesselman. Performance analysis of A-MPDU and A-MSDU aggregation in IEEE 802.11n. In *IEEE Sarnoff Symposium, 2007*, pages 1–5, May 2007.

[13] D. Halperin, B. Greensteiny, A. Shethy, and D. Wetherall. Demystifying 802.11n power consumption. In *Proceedings of the 2010 International Conference on Power Aware Computing and Systems, HotPower'10*, Berkeley, CA, USA, 2010. USENIX Association.

[14] W.-K. Kuo. Energy efficiency modelling for IEEE 802.11a distribution coordination function system without finite retry limits. *Communications, IET*, 1(2):165–172, 2007.

[15] H. Liu and Y. Wang. An Implementation Scheme of Power Saving Mechanism for IEEE 802.11e. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 1, pages 1178–1181, Dec 2008.

[16] Network Working Group. RFC 5166, Metrics for the Evaluation of Congestion Control Mechanisms, 2008. <http://tools.ietf.org/html/rfc5166>.

[17] S. H. Nguyen, H. L. Vu, and L. L. H. Andrew. Packet Size Variability Affects Collisions and Energy Efficiency in WLANs. In *2010 IEEE Wireless Communications and Networking Conference (WCNC 2010)*, pages 1–6, 2010.

[18] J. Pacheco de Carvalho, H. Veiga, N. Marques, C. Pacheco, and A. Reis. A contribution to performance measurements of IEEE 802.11 a, b, g laboratory WEP point-to-point links using TCP, UDP and FTP. In *International Conference on Applied Electronics (AE 2010)*, 2010.

[19] K. Pelechrinis, I. Broustis, T. Salonidis, S. Krishnamurthy, and P. Mohapatra. Design and Deployment Considerations for High Performance MIMO Testbeds. In *Wireless Internet Conference, WICON 2008*, 2008.

[20] D. Qiao and K. Shin. Smart power-saving mode for IEEE 802.11 wireless LANs. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1573–1583, March 2005.

[21] Qualcomm Press Release. Qualcomm to Acquire Atheros, Leader in Connectivity & Networking Solutions, Jan 2011. <http://www.qualcomm.com/news/releases/2011/01/05/qualcomm-acquire-atheros-leader-connectivity-networking-solutions>.

[22] S. Sendra, P. Fernandez, C. Turro, and J. Lloret. IEEE 802.11a/b/g/n Indoor Coverage and Performance Comparison. In *6th International Conference on Wireless and Mobile Communications (ICWMC)*, 2010.

[23] S.-L. Wu and P.-C. Tseng. An Energy Efficient MAC Protocol for IEEE 802.11 WLANs. *Communication Networks and Services Research, Annual Conference on*, 0:137–145, 2004.

[24] M. Zink, K. Suh, Y. Gu, and J. Kurose. Watch Global, Cache Local: YouTube Network Traffic at a Campus Network - Measurements and Implications. In *Proceedings of IEEE MMCN 08*, 2008.